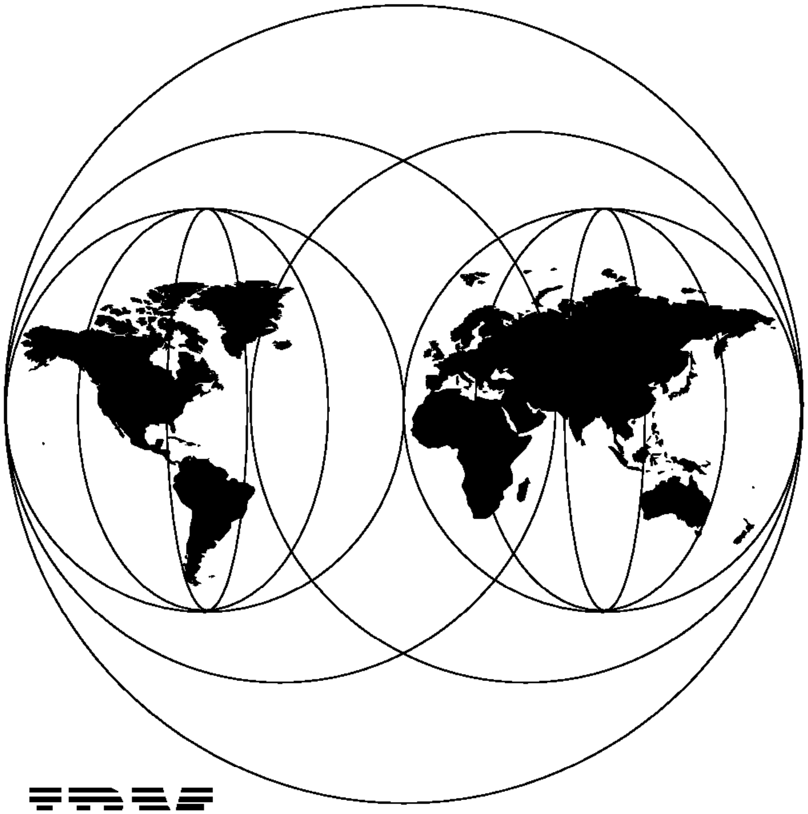
SG24-2080-00

# Technical Presentation for PSSP Version 2.3

December 1997



IBM
International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-2080-00

# Technical Presentation for PSSP Version 2.3

December 1997

**Take Note!**

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 533.

**First Edition (December 1997)**

This edition applies to PSSP Version 2, Release 3 for use with the AIX Version 4, Release 2 Modification 1 operating System.

**Attention**

This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. It is recommended that, when the product becomes generally available, you destroy all copies of this version of the book that you have in your possession.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

# Contents

# Figures

# Tables

# Preface

This redbook offers detailed discussions of the new functions and components in Parallel Software Support Program Version 2 Release 3 (PSSP 2.3) and the new 604e PowerPC High Nodes product, which are major enhancements to the RS/6000 SP product line.

This redbook is for IBM customers, Business Partners, and IBM technical and marketing professionals.

It is in the format of a technical presentation guide, focussing on the following topics:

- PowerPC 604e High Nodes
- SP-8 Switch with High Nodes
- Software Coexistence
- Migration Considerations
- AIX Automounter
- General Parallel File System
- Dependent Node Architecture
- Parallel Environment
- Message Passing Interface

Familiarity with AIX Version 4 and RS/6000 SP is assumed.

## The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Endy Chiakpo** is a Senior Development Manager in the RS/6000 Scalable POWERparallel Lab in Poughkeepsie, New York. He was a Project Leader at the International Technical Support Organization, Pougkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of RS/6000 SP. He holds a B.S. degree in Physics and a Master of Science degree in Electrical Engineering from Syracuse University New York. Before joining the ITSO, Endy worked in the IBM Poughkeepsie Lab in New York, USA.

**Clive Harris** is an RS/6000 and SP Technical Consultant working for IBM′s RS/6000 Business. He is team leader of the EMEA (European) SP Centre of Competence based in the UK. Clive worked in Austin, Texas in the US at the AIX development labs prior to the RISC System/6000 launch, and developed the worldwide product introduction workshop for the RS/6000. He has been with IBM for 12 years and has had a number of roles working with customers and a range of IBM platforms. He started working with IBM′s AIX system in 1989. Clive is now responsible for assisting customers in designing complex AIX SP solutions and runs European programs to develop skills transfer in SP-related matters. He is the author of IBM technical publications (Redbooks) and also a McGraw book, *The IBM RISC System/6000*.

**Alvaro Franco** is an RS/6000 SP specialist at IBM Uruguay. He has eight years of experience in the UNIX field. His areas of expertise include Open Systems, Networking, C programming, Internet issues, and AIX performance. He has taught courses in these fields. He worked for ORT University before joining IBM, where he assisted with Open System and Networking projects.

**Luke Koh** is an Advisory IT Specialist for RS/6000 Technical Support in IBM World Trade Asia Corporation. He holds an honors degree in Information Systems and Computer Science from the National University of Singapore. He joined IBM in 1992 as a System Engineer in Government and Education accounts. He has worked on the RS/6000 SP since the inception of SP1 and is currently responsible for technical marketing and services for the RS/6000 SP for IBM South Asia.

**Koon Meng Tan** is an IT Specialist in IBM World Trade Asia Corporation in Singapore. He holds a degree in Information Systems and Computer Science from the National University of Singapore. He has done parallelization on the SP2 machine, such as Computational Fluid Dynamics and VLSI cell placement, all involving MPI. His current job is providing RS/6000 technical support.

**Poh Yee Tiong** is an Advisory Services Specialist in IBM Singapore. He is responsible for AIX, RS/6000 SP and HACMP/6000 Services and Support for the ASEAN region. He joined IBM in 1990.

**Jan Keymolen** is a System Engineer in Belgium and has more than five years experience as Country Support for RS/6000 and RS/6000 SP software and hardware. His areas of expertise include the Network specialities ATM, SNA, X25, and others. He now works for IBM at a major RS/6000 SP account in Luxembourg.

**Hubert Vacher** is in pre-sales SP Technical Support in West Region Europe. He has 10 years of experience in the UNIX field. He holds a degree in Electronics. His areas of expertise include PC/RT, AIX/370, and AIX/ESA.

Thanks to the following people for their invaluable contributions to this project:

Peter Kes
International Technical Support Organization, Poughkeepsie Center.

IBM PPS Lab Poughkeepsie:
Mike Browne
Joseph Banas
Ken Briskey
Linda Melor
James Gilman
Paul Bildzok
John Simpson
Dave Delia
Stephanie Beals

Jay Benjamin
Ron Linton
Chris Algozzine
John Doxtader
John Divirgilio
Dr. Rama Govindaraju
Bill Ferrante
Bill Wajda
Dr. Bill Tuel

## Comments Welcome

**Your comments are important to us!**

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

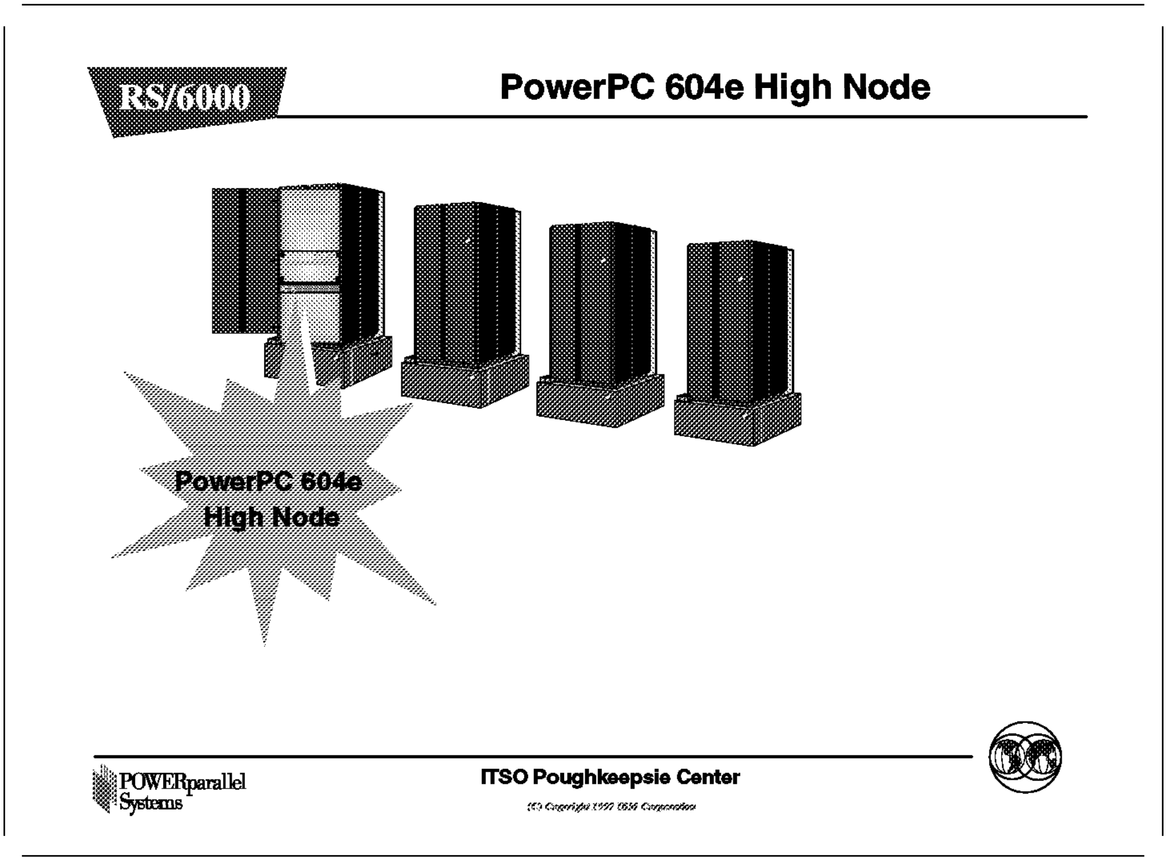
- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 545 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

| | |
|---|---|
| For Internet users | `http://www.redbooks.ibm.com` |
| For IBM Intranet users | `http://w3.itso.ibm.com` |

- Send us a note at the following address:

  `redbook@vnet.ibm.com`

# Chapter 1. PowerPC 604e High Nodes



RS/6000

PowerPC 604e High Node

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

With the announcement of High Nodes, a new era began for the IBM RS/6000 SP. UNIX-based Symmetric Multi-Processing has been around since the 1980s. As the demand for less expensive processing power has grown, IBM has developed the RS/6000 SP for Massive Parallel processing. IBM developed Symmetric Multiprocessors (SMP) in the RS/6000 server range; G30, J30, and R30 were the first models. With the announcement of the High Nodes last year, both Symmetric Multiprocessors (SMP) and Massively Parallel Processors (MPP) are reunited in the same architecture. A new High Node is announced: the PowerPC 604e High Node.

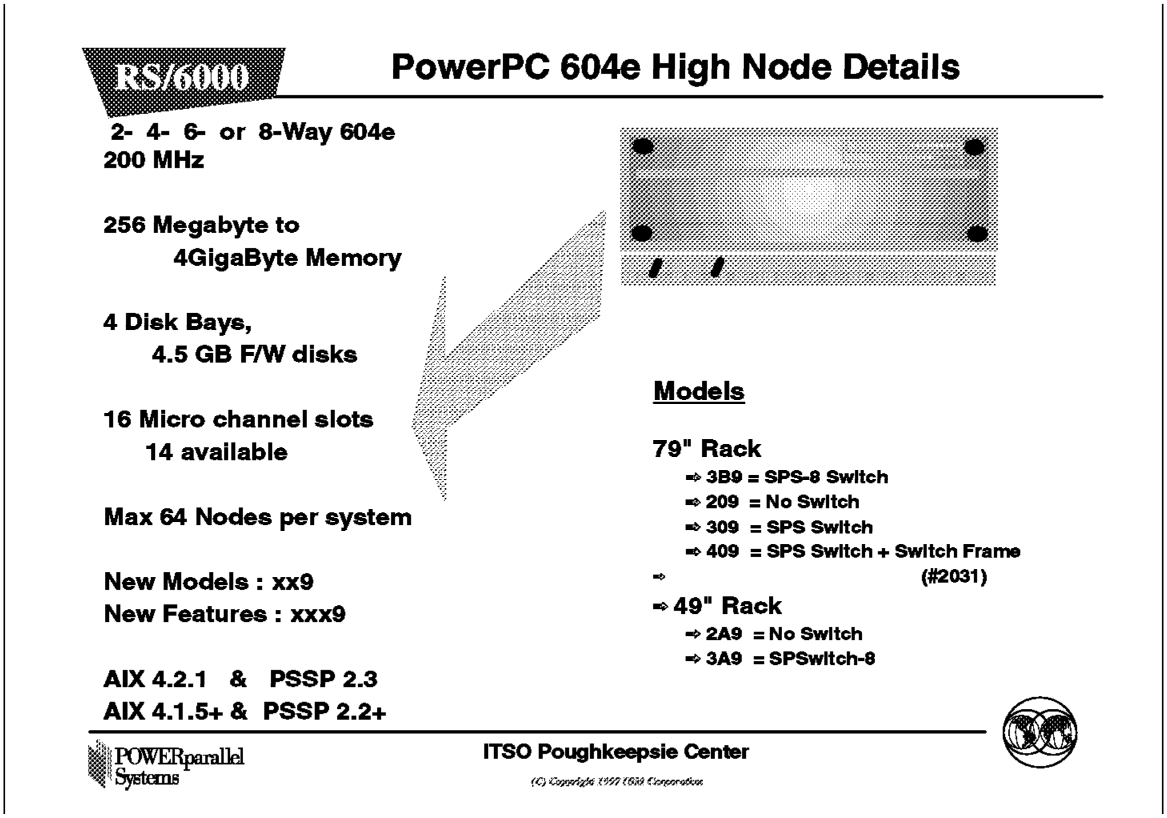
This chapter is organized as follows :

- The first section describes the PowerPC 604e High Node hardware.
- The second section describes the PowerPC 604e High Node differences and limitations in relation to the 604 High Node.
- The third section provide a performance comparison with other nodes on the RS/6000 SP system and discusses future trends.
- The fourth section describes the software requirements for the 604e High Node.
- The last section explains how to install a PowerPC 604 High node.

## 1.1 604e Details

The following is a description of the 604e High Node details. For a detailed explanation of SMPs and IBM implementation, see *IBM RISC System/6000 SMP Servers Architecture and Implementation*, SG24-2583.

### 1.1.1 604e Details and Models



RS/6000 **PowerPC 604e High Node Details**

**2- 4- 6- or 8-Way 604e**
**200 MHz**

**256 Megabyte to**
**4GigaByte Memory**

**4 Disk Bays,**
**4.5 GB F/W disks**

**16 Micro channel slots**
**14 available**

**Max 64 Nodes per system**

**New Models : xx9**
**New Features : xxx9**

**AIX 4.2.1 & PSSP 2.3**
**AIX 4.1.5+ & PSSP 2.2+**

**Models**

**79" Rack**
- ⇨ 3B9 = SPS-8 Switch
- ⇨ 209 = No Switch
- ⇨ 309 = SPS Switch
- ⇨ 409 = SPS Switch + Switch Frame
- ⇨ (#2031)

**⇨ 49" Rack**
- ⇨ 2A9 = No Switch
- ⇨ 3A9 = SPSwitch-8

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

The RS/6000 SP PowerPC 604e High Node is a new member of the High Node family. It is based on the RS/6000 model R50, but without the exact power management of the R50 RS/6000.

Power control is under the supervision of PSSP 2.3. You can have a redundant power supply, but this is an option. The standard 604e High Node comes with one power supply module and one fan module. If you want to have redundant power, you need to order the appropriate #code to replace the fan drawer with a second power supply drawer. The feature code for the second power module is #6293.

As with the 604 High Node, you can have 2, 4, 6, or 8 processors. There are four CPU-cards, and each card has two processors.

The memory is a minimum of 256MB, and a maximum of 4GB.

There are four memory slots, allowing 256MB, 512MB, or 1GB memory cards. A memory card can host a 256MB, 512MB or 1GB amount of Dual Inline Memory

Modules (DIMMS). With this new memory card comes the 256MB DIMM KIT. A 256MB DIMM KIT consist of four 64MB DIMMs.

An upgrade from 604 High Node memory cards is available. If upgrading from 604 High Node to 604e High Node, the memory cards 64MB and 128MB type Starfish are supported. On initial order only 256MB, 512MB and 1GB memory cards are available.

Four disk bays are available. Each disk bay can host one 1-inch high 4.5 GB SCSI-2 F/W disk. This makes a total of 18GB of disk space available internally in the 604e High Node.

One of the existing 16 microchannels is used for the internal SCSI-2 Fast/Wide single ended controller. Another slot is required for the Ethernet adapter. This leaves 14 available microchannel slots.

When a switch is installed in the RS/6000 SP, one microchannel slot is occupied with the Switch Adapter Card.

The 604e High Node is supported by the SPS-8 switch and also by the short frame (also called the low-cost frame or Low Boy).

A maximum of 64 PowerPC High Nodes are supported in an RS/6000 SP.

Five new models are available with the 604e High Node, as follows :

| **Model** | **Characteristics** |
|---|---|
| **209** | A 79″ rack with no switch and a 604e High Node in Position 1. |
| **309** | A 79″ rack with an SPS switch and a 604e High Node in Position 1. |
| **409** | A 79″ rack with an SPS switch and a #2031 SPS switch Frame. This means a rack with only switches inside. |
| **2A9** | A 49″ rack with no switch and an 604e High Node in Position 1. |
| **3A9** | A 49″ rack with an SPS-8 switch and a 604e Node in Position 1. |
| **3B9** | A 79″ rack with an SPS-8 switch and a 604e node in Position 1. |

Figure 1. Models

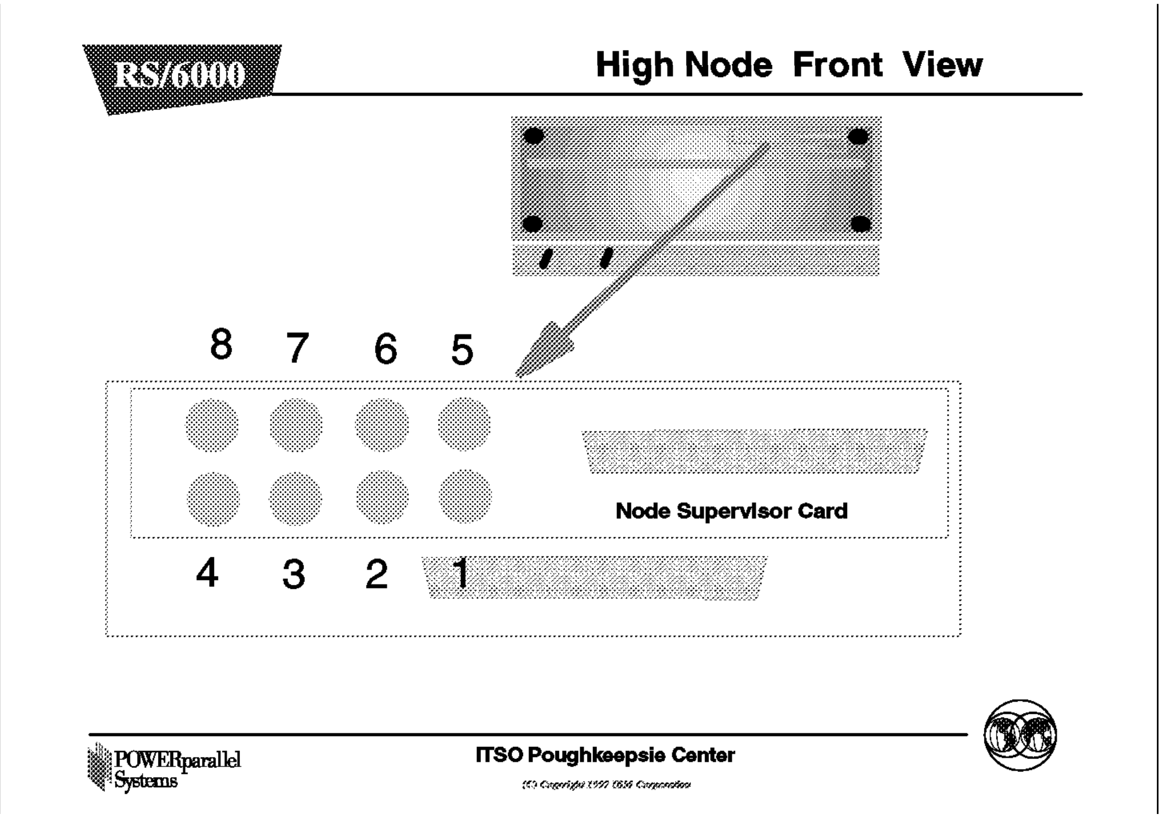
**Important**

The new features for the 604e High Node are:

| | |
|---|---|
| CPU-card(two processors): | #4324 |
| 256MB memory card: | #4165 |
| 512MB memory card: | #4154 |
| 1GB memory card: | #4167 |
| 79-inch expansion frame: | #1009 |
| 79-inch supported from parent switch: | #1019 |
| 49-inch expansion frame: | #1029 |
| Optional power module: | #6293 |
| 4.5GB 1-inch DASD: | #3000 |
| 604e High Node: | #2009 |

AIX 4.1.5 or AIX 4.2.1 and PSSP 2.2 or AIX 4.2.1 and PSSP 2.3 is required for the 604e High Node. See section 1.4, “Software Requirements” on page 17.

### 1.1.2 604e Front and LEDs



RS/6000

High Node Front View

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The 604e High Node has the same front as the 604 High Node: two cables and a Node Supervisor card with two RS232 connectors and eight LEDs; four green LEDs and four yellow LEDS.

The following tables explain what the LEDs mean. Their meaning depends on whether the Node Supervisor card is in Normal Operation state or Supervisor Download State.

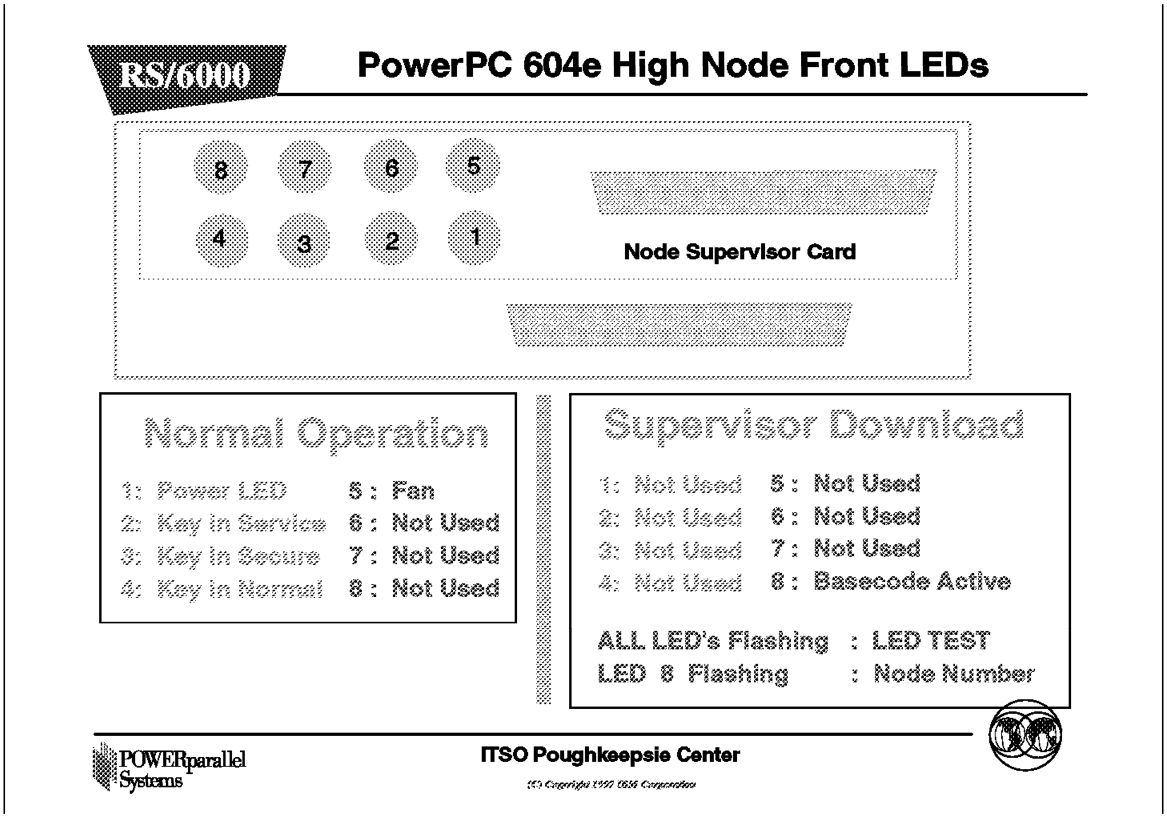
## NORMAL OPERATION STATE

**Green LEDs**

| | |
|---|---|
| **LED 1** | POWER |
| **LED 2** | Key in Service position |
| **LED 3** | Key in Secure position |
| **LED 4** | Key in Normal position |

**Yellow LEDS**

| | |
|---|---|
| **LED 5** | FAN problem, also called Environment Problem. |
| **LED 6** | Not used |
| **LED 7** | Not used |
| **LED 8** | Not used |



## PowerPC 604e High Node Front LEDs

Node Supervisor Card

**Normal Operation**

1: Power LED
2: Key in Service
3: Key in Secure
4: Key in Normal
5 : Fan
6 : Not Used
7 : Not Used
8 : Not Used

**Supervisor Download**

1: Not Used
2: Not Used
3: Not Used
4: Not Used
5 : Not Used
6 : Not Used
7 : Not Used
8 : Basecode Active

ALL LED's Flashing : LED TEST
LED 8 Flashing : Node Number

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

**SUPERVISOR DOWNLOAD STATE**

**GREEN LEDS**

**LED 1** Not used

**LED 2** Not used

**LED 3** Not used

**LED 4** Not used

**Yellow LEDs**

**LED 5** Not used

**LED 6** Not used

**LED 7** Not used

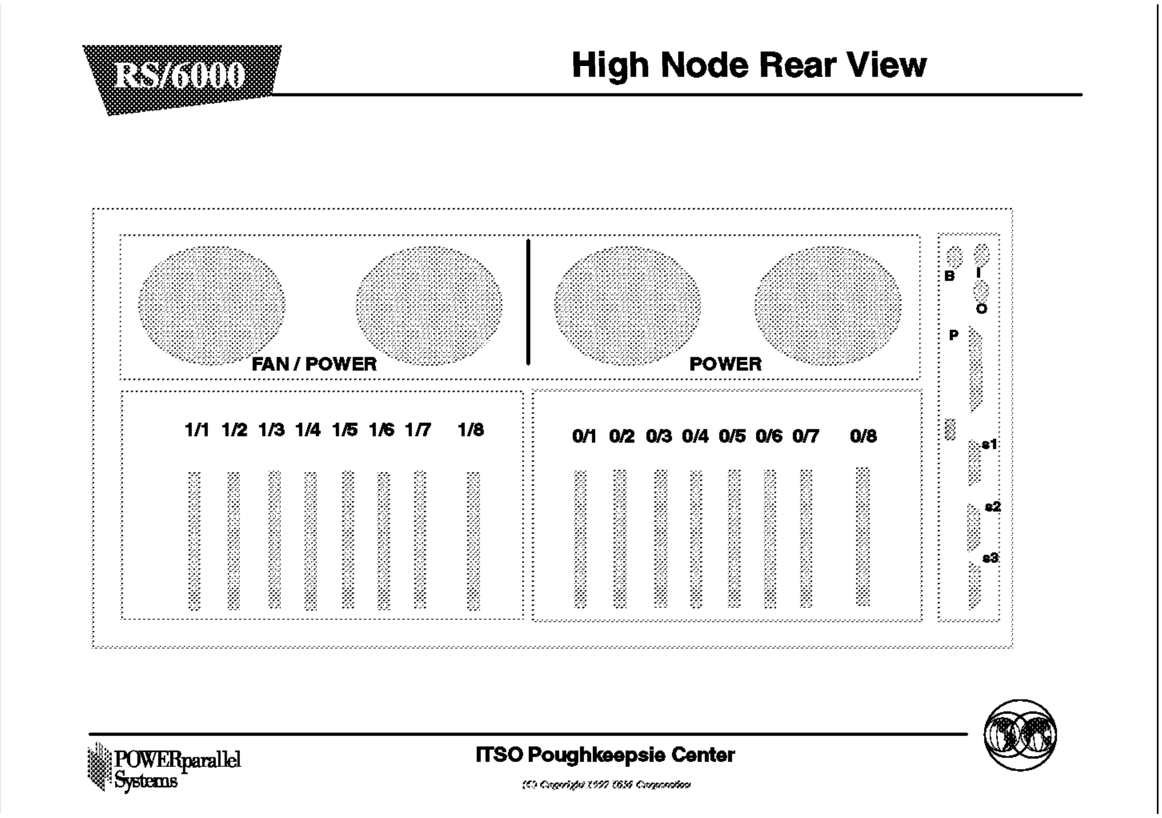
**LED 8** Basecode active means that CWS is downloading Node Supervisor code to this node.

**Note:**

- ALL LEDs Flashing = LED TEST. LED test happens at the end of the Node Supervisor Code download, when the Node Supervisor is rebooted
- LED 8 Flashing = Node Number. At the end of the reboot of the Node Supervisor, LED 8 will flash the address of the node. For example, for node 9 in frame 4, LED 8 will flash 9 times. The rack number is not reflected in the node number LED flashing.

For a detailed explanation of how to download Node Supervisor microcode, see *RS/6000 SP PSSP 2.2 Technical Presentation Redbook* SG24-4868. For more information about SMP, see *IBM RISC System/6000 SMP Servers Architecture and Implementation*, SG24-2583.

## 1.1.3 604e Rear View



RS/6000

High Node Rear View

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The rear of the PowerPC 604e High Node is the same as that of the 604 High Node. At the top left a fan drawer or a second redundant power supply, at the right the standard power supply.

At the bottom we have the two microchannels, microchannel one which has eight slots: slots 1/1 to 1/8 and microchannel zero which has also eight slots: slots 0/1 to 0/8. The 1/1 means: microchannel 1, slot 1. The 0/1 means: microchannel 0, slot 1.

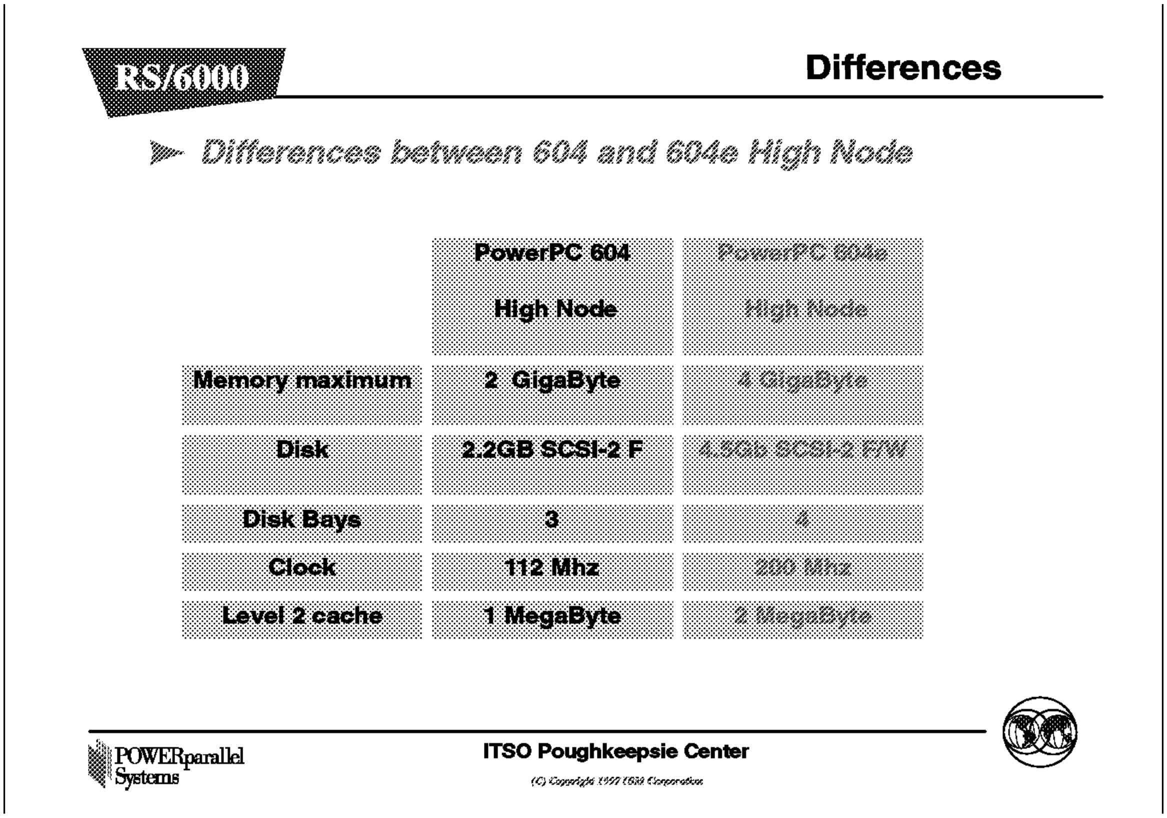
At the right we have the SIB (System Interface Board) card. The SIB has three serial connectors, and one parallel connector, plus three unused RS485 connectors.

**Note:** The unused RS485 connectors are used in the RS/6000 SMP models. One connector is used for battery backup, the other two for the Power Control Interface (PCI), PCI in and PCI out.

## 1.2 Differences and Limitations

The following is a description of the differences between the 604 and the 604e High Node.

### 1.2.1 Differences



RS/6000

**Differences**

- *Differences between 604 and 604e High Node*

| | PowerPC 604<br>High Node | PowerPC 604e<br>High Node |
|---|---|---|
| Memory maximum | 2 GigaByte | 4 GigaByte |
| Disk | 2.2GB SCSI-2 F | 4.5Gb SCSI-2 F/W |
| Disk Bays | 3 | 4 |
| Clock | 112 Mhz | 200 Mhz |
| Level 2 cache | 1 MegaByte | 2 MegaByte |

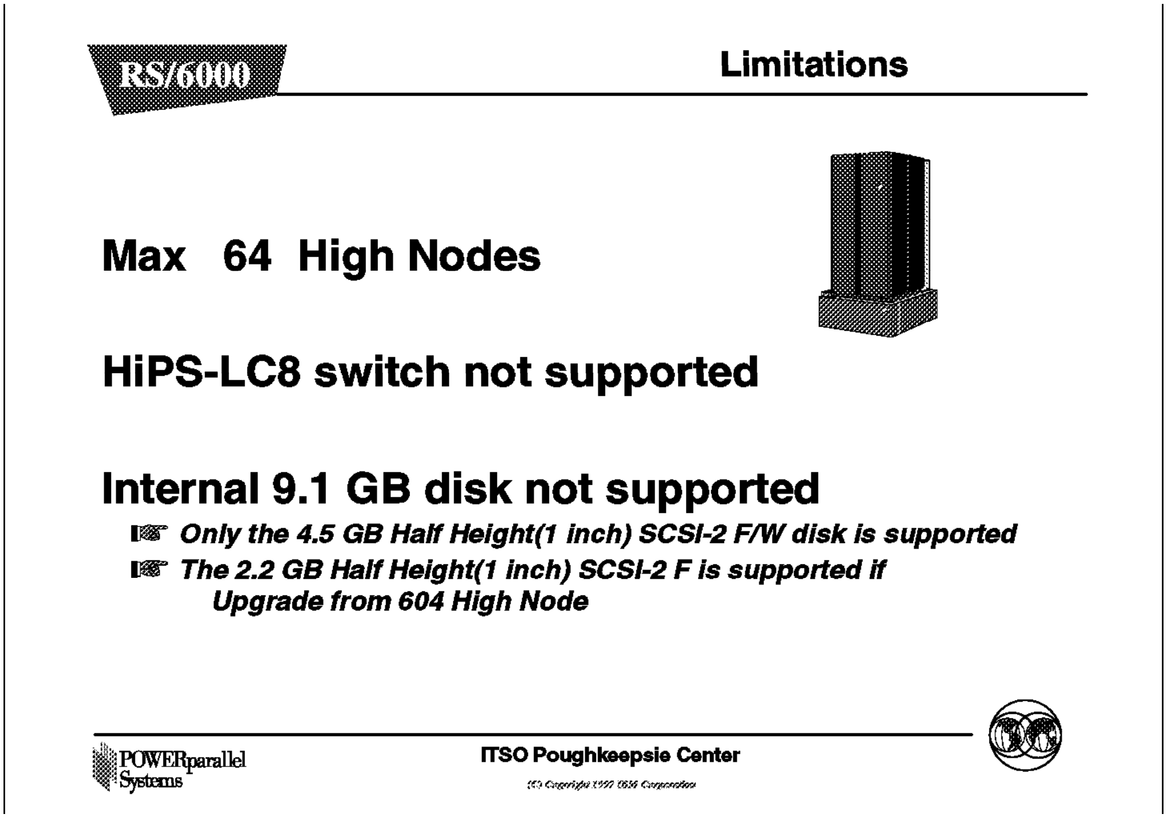
POWERparallel Systems

ITSO Poughkeepsie Center

The differences between the 604 and 604e High Node are as follows:

- Double the memory capacity, 4GB versus 2GB.
- Three times the internal disk space, 18GB versus 6.6GB.
- The CPU clock is 200MHz, almost the double of 112MHz.
- Level 1 cache is doubled, 32KB instead of 16KB.
- Level 2 cache is also doubled, 2MB versus 1MB.

### 1.2.2 Limitations



RS/6000

**Limitations**

**Max 64 High Nodes**

**HiPS-LC8 switch not supported**

**Internal 9.1 GB disk not supported**

- ☞ ***Only the 4.5 GB Half Height(1 inch) SCSI-2 F/W disk is supported***
- ☞ ***The 2.2 GB Half Height(1 inch) SCSI-2 F is supported if Upgrade from 604 High Node***

POWERparallel Systems

**ITSO Poughkeepsie Center**

*(C) Copyright 1997 IBM Corporation*

Up to 64 RS/6000 SP High Nodes in a RS/6000 SP. This could be 604 or 604e High Nodes.

The HiPS LC-8 switch is not supported.

The 9.1GB SCSI-2 disk is not supported. The reason for this is that the four bays are one inch high, while the 9.1GB SCSI-2 disk is two inches high.

The 2.2GB 8-bit SCSI-2 fast disk from the 604 High node is supported when upgrading to a 604e High Node. The 4.5GB 1-inch disk is a 16-bit SCSI-2 F/W device. With the upgrade comes an 8-bit to 16-bit converter for the 2.2GB one inch high disk.

## 1.3 Performances and Trends

This sections list the different performances numbers for POWER2, P2SC, 604, and 604e processors. The first section gives the numbers for the POWER2 and P2SC nodes. The second section covers 604 and 604e processors. The third section explain briefly what is expected in future.

The different benchmarks used are:

.

- SPECint95: SPEC component-level benchmark that measures integer performance. Result is the geometric mean of eight tests that comprise the CINT95 benchmark suite. All of these are written in C language. SPECint_base95 is the result of the same tests in CINT95 with a maximum of four compiler flags that must be used in all eight tests.
- SPECfp95: SPEC component-level benchmark that measures floating point performance. Result is the geometric mean of ten tests, all written in FORTRAN, that are included in the CFP95 benchmark suite. SPECfp_base95 is the result of the same tests in CFP95 with a maximum of four compiler flags that must be used in all ten tests.
- SPECint_rate95: Geometric average of the eight SPEC rates from the SPEC integer tests (CINT95). SPECint_base_rate95 is the result of the same tests as CINT95 with restrictive compiler options.
- SPECfp_rate95: Geometric average of the ten SPEC rates from SPEC floating-point tests (CFP95). SPECfp_base_rate95 is the result of the same tests as CFP95 with restrictive compiler options.
- LINPACK DP: Double precision, n=100 results with AIX XL FORTRAN compiler, with optimization. Units are megaflops (MFLOPS).
- LINPACK SP: Single Precision, n=100 results with AIX XL FORTRAN compiler, with optimization. Units are megaflops (MFLOPS).
- Rel OLTP Perf: Relative OLTP Performance is an estimate of commercial throughput using an IBM analytical model. This model simulates some of the system′s operations of the CPU, caches and memory in an OLTP environment but does not simulate the disk or network I/O operations. Although general database and operating systems parameters are used, the model does not represent specific databases or AIX versions. With these limitations, ROP may be used to compare RS/6000 performance. The Model 250 is the reference system and has a value of 1.0.

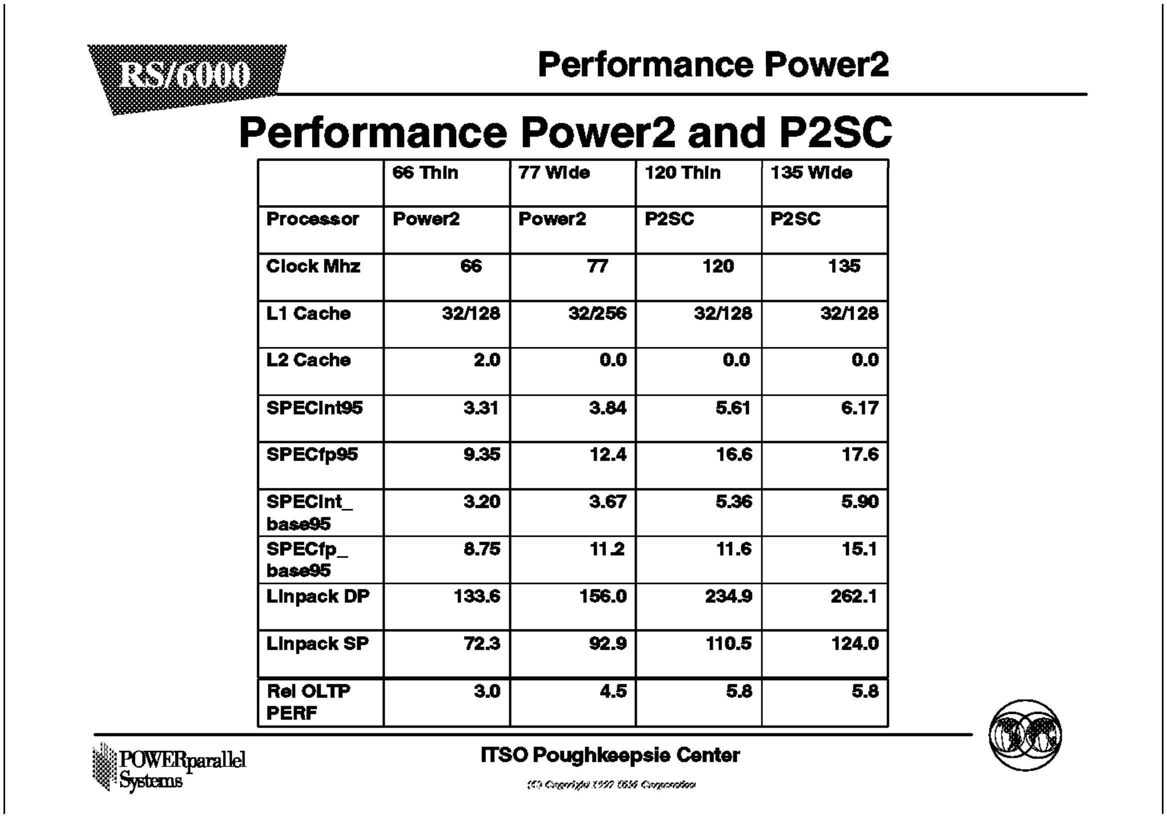
**SPECweb96**

A newcomer is the SPECweb96 performance number. No SPECweb96 performance number is releases yet for the High nodes.

SPECweb96: Maximum number of HTTP operations per second achieved on the SPECweb96 benchmark without significant degradation of response time. The Web server software is Zeus 1.1 from Zeus Technology Ltd.

SPECweb96 is a software benchmark product developed by the Standard Performance Evaluation Corp.(SPEC), a non-profit group. It is designed to measure a system′s ability to act as a World Wide Web server for static pages. A SPECweb96 test bed consists of a server machine that runs the Web server software to be tested and a set number of client machines. The client machines use the SPECweb96 software to generate a workload that stresses the server system, both hardware and software. The workload is gradually increased until the server software is saturated with hits and the response time degrades significantly. The point at which the server is saturated is the maximum number of HTTP operations per second that the Web server software can sustain. That maximum number of HTTP operations per second is the SPECweb96 performance metric that is reported. More information may be found at *www.specbench.org/osg/specweb*

## 1.3.1 Performances POWER2 and P2SC nodes



RS/6000

Performance Power2

Performance Power2 and P2SC

| | 66 Thin | 77 Wide | 120 Thin | 135 Wide |
|---|---|---|---|---|
| Processor | Power2 | Power2 | P2SC | P2SC |
| Clock Mhz | 66 | 77 | 120 | 135 |
| L1 Cache | 32/128 | 32/256 | 32/128 | 32/128 |
| L2 Cache | 2.0 | 0.0 | 0.0 | 0.0 |
| SPECInt95 | 3.31 | 3.84 | 5.61 | 6.17 |
| SPECfp95 | 9.35 | 12.4 | 16.6 | 17.6 |
| SPECInt_ base95 | 3.20 | 3.67 | 5.36 | 5.90 |
| SPECfp_ base95 | 8.75 | 11.2 | 11.6 | 15.1 |
| Linpack DP | 133.6 | 156.0 | 234.9 | 262.1 |
| Linpack SP | 72.3 | 92.9 | 110.5 | 124.0 |
| Rel OLTP PERF | 3.0 | 4.5 | 5.8 | 5.8 |

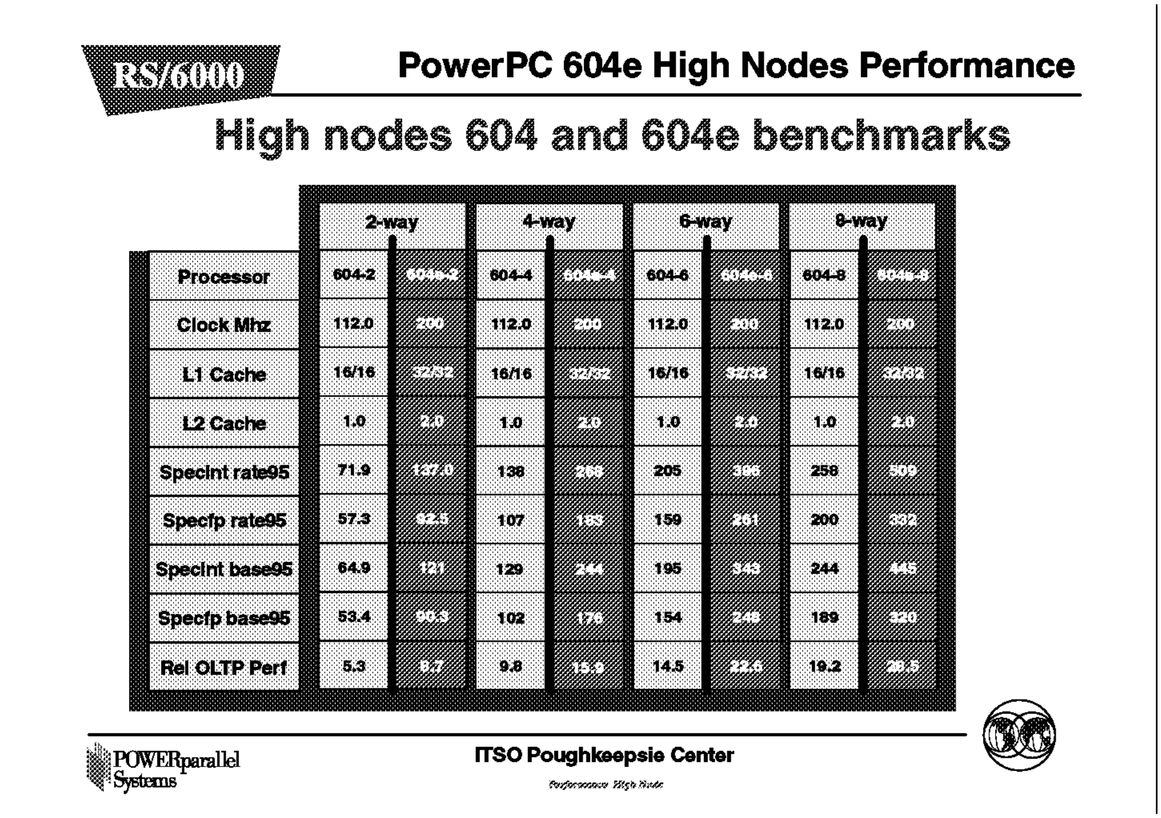
POWERparallel Systems

ITSO Poughkeepsie Center

**Note:** Above performance numbers are not commercial benchmarks.

- Power2 The characteristics of the 66MHz Thin node compared to the 77MHz wide node.
- Power2SuperChip(P2SC) The characteristics of the 120MHz Thin node compared to the 135MHz Wide node.

## 1.3.2 High Nodes Performance



RS/6000

PowerPC 604e High Nodes Performance

High nodes 604 and 604e benchmarks

| | 2-way | | 4-way | | 6-way | | 8-way | |
|---|---|---|---|---|---|---|---|---|
| Processor | 604-2 | 604e-2 | 604-4 | 604e-4 | 604-6 | 604e-6 | 604-8 | 604e-8 |
| Clock Mhz | 112.0 | 200 | 112.0 | 200 | 112.0 | 200 | 112.0 | 200 |
| L1 Cache | 16/16 | 32/32 | 16/16 | 32/32 | 16/16 | 32/32 | 16/16 | 32/32 |
| L2 Cache | 1.0 | 2.0 | 1.0 | 2.0 | 1.0 | 2.0 | 1.0 | 2.0 |
| SpecInt rate95 | 71.9 | 137.0 | 138 | 269 | 205 | 396 | 258 | 509 |
| Specfp rate95 | 57.3 | 92.5 | 107 | 183 | 159 | 261 | 200 | 332 |
| SpecInt base95 | 64.9 | 121 | 129 | 244 | 195 | 343 | 244 | 445 |
| Specfp base95 | 53.4 | 90.3 | 102 | 176 | 154 | 248 | 189 | 320 |
| Rel OLTP Perf | 5.3 | 8.7 | 9.8 | 15.9 | 14.5 | 22.6 | 19.2 | 28.5 |

POWERparallel Systems

ITSO Poughkeepsie Center

Performance High Node

The characteristics of the 604 High node compared to the 604e High node. The performance of the PowerPC 604e High Node is at least 50% higher than that of the PowerPC 604 High Node.

With the PowerPC 604e High node IBM has taken the lead in massively parallel computing.

Performance improvements have been achieved by:

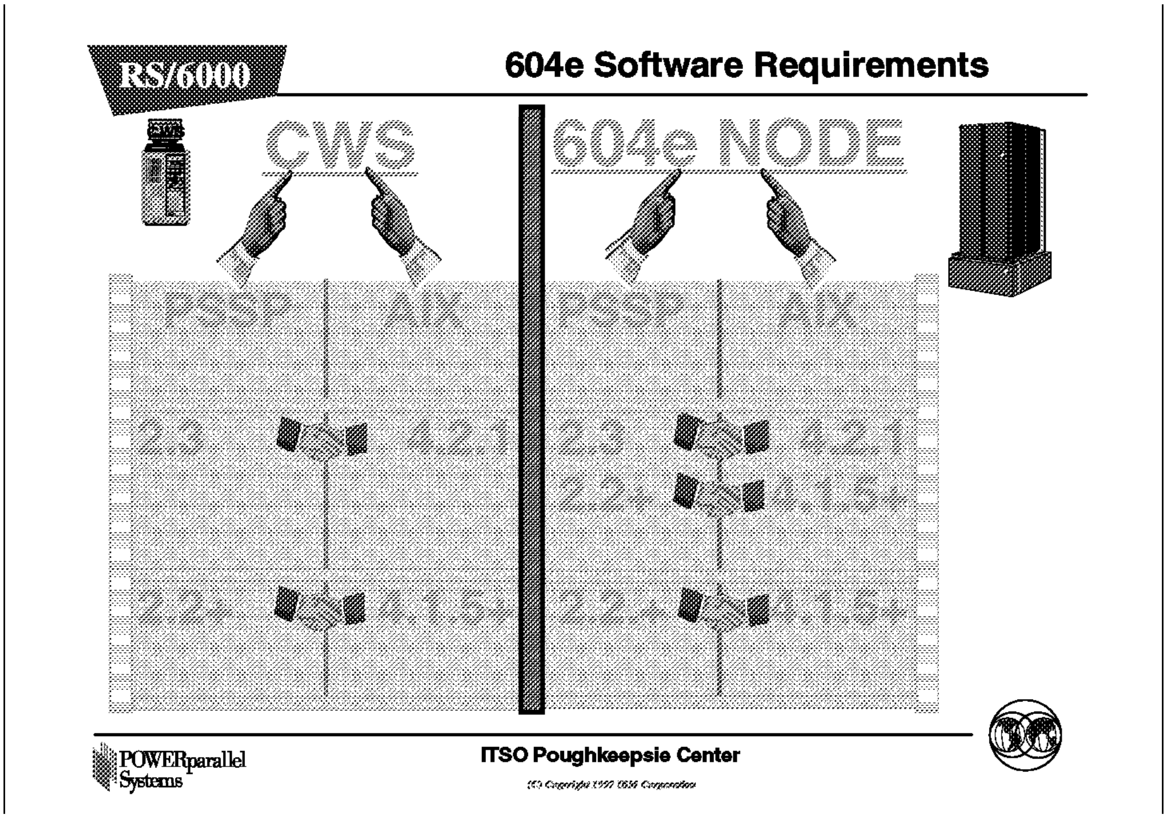
- Increasing memory size.
- Increasing level 1 and level 2 cache sizes.
- Faster CPU

Future improvements will be achieved by:

- Moving to 64 bit technology.
- Speeding up the memory and I/O busses.
- Including multiple I/O busses to spread the I/O load.
- Increasing memory size.
- Increasing level 1 and level 2 cache sizes.

All this results in a new era of very fast computers.

## 1.4 Software Requirements



RS/6000

604e Software Requirements

CWS

604e NODE

PSSP AIX PSSP AIX

2.3 4.2.1 2.3 4.2.1

2.2+ 4.1.5+

2.2+ 4.1.5+ 2.2+ 4.1.5+

POWERparallel Systems

ITSO Poughkeepsie Center

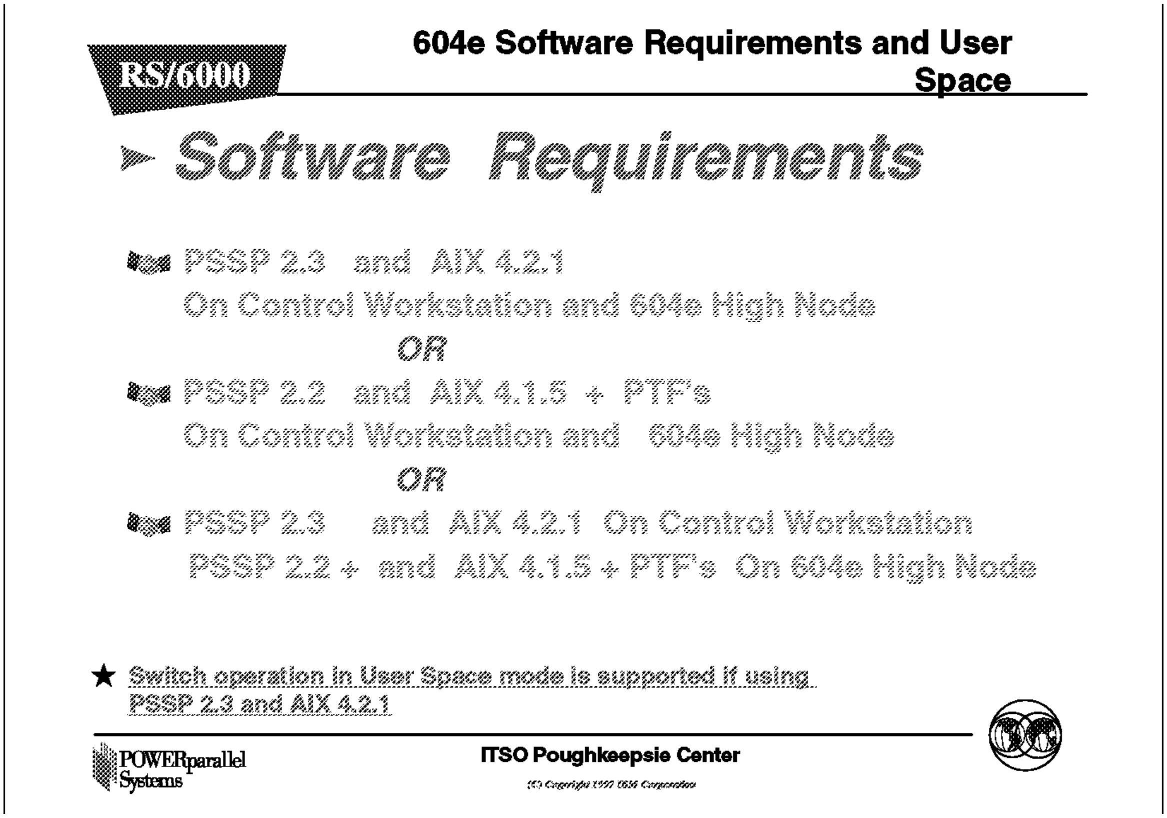
(C) Copyright 1997 IBM Corporation

The above figure shows the correlation between the different levels of AIX and PSSP.

1. PSSP 2.3 and AIX 4.2.1 installed on the CWS:
   - PSSP 2.3 and AIX 4.2.1 on the node.
   - PSSP 2.2 plus PTFs and AIX 4.1.5 plus PTFs on the node.
   - PSSP 2.2 plus PTFs and AIX 4.2.1 on the node.
2. PSSP 2.2 plus PTFs and AIX 4.1.5 plus PTFs installed on the CWS:
   - PSSP 2.2 plus PTFs and AIX 4.1.5 plus PTFs on the node.
   - PSSP 2.2 plus PTFs and AIX 4.2.1 on the CWS.
   - PSSP 2.2 plus PTFs and AIX 4.2.1 on the nodes.

Note that the CWS *must* be at the latest level in the RS/6000 SP complex.

## 1.4.1 Software Requirements and User Space Protocol



RS/6000

604e Software Requirements and User Space

➤ Software Requirements

- PSSP 2.3 and AIX 4.2.1
  On Control Workstation and 604e High Node

  OR

- PSSP 2.2 and AIX 4.1.5 + PTF's
  On Control Workstation and 604e High Node

  OR

- PSSP 2.3 and AIX 4.2.1 On Control Workstation
  PSSP 2.2 + and AIX 4.1.5 + PTF's On 604e High Node

★ Switch operation in User Space mode is supported if using PSSP 2.3 and AIX 4.2.1

POWERparallel Systems

ITSO Poughkeepsie Center

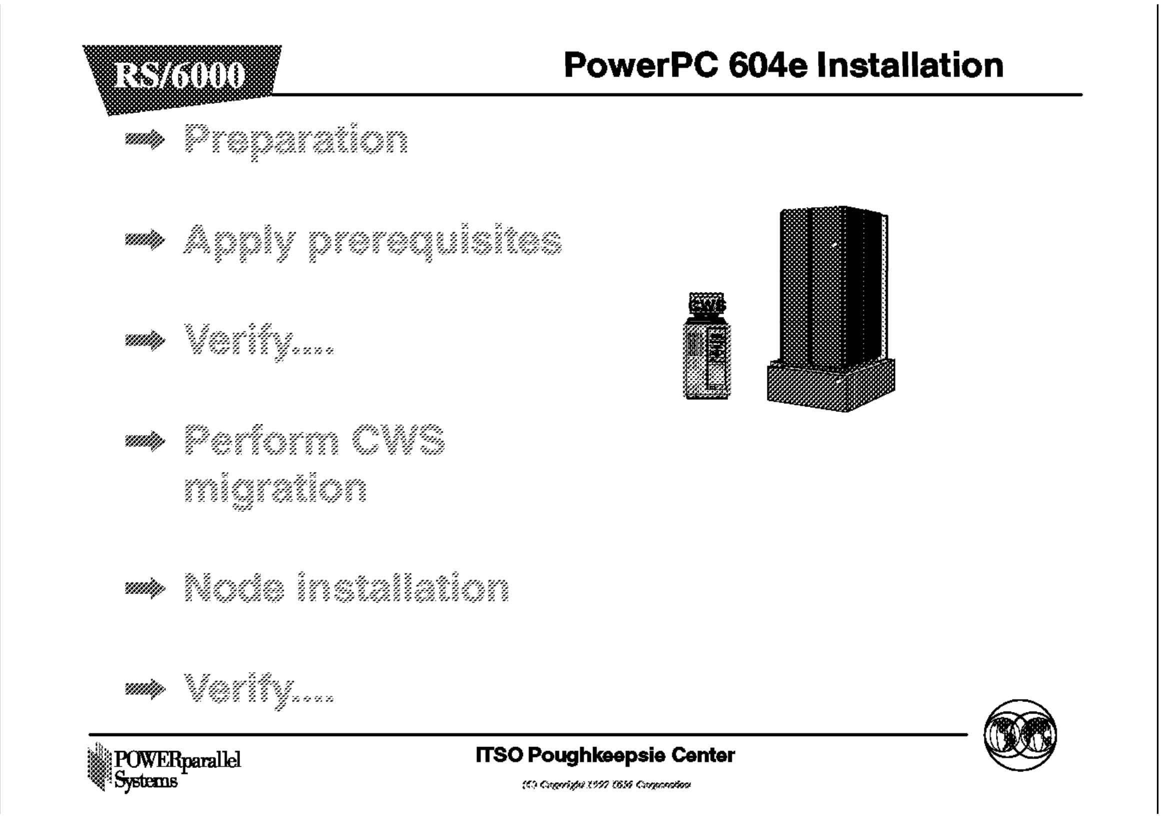
(C) Copyright 1997 IBM Corporation

As shown in the above figure, the PowerPC 604e High node is supported with PSSP 2.3 and AIX 4.2.1 and also with PSSP 2.2 and AIX 4.1.5 plus PTFs or PSSP 2.2 and AIX 4.2.1.

If AIX 4.2.1 and PSSP 2.3 are installed, User Space protocol for the switch is supported.

If PSSP 2.2 and AIX 4.1.5 plus PTFs are installed, User Space protocol for the switch is not supported for the High nodes.

## 1.5 Node Installation



RS/6000

PowerPC 604e Installation

- Preparation
- Apply prerequisites
- Verify....
- Perform CWS migration
- Node installation
- Verify....

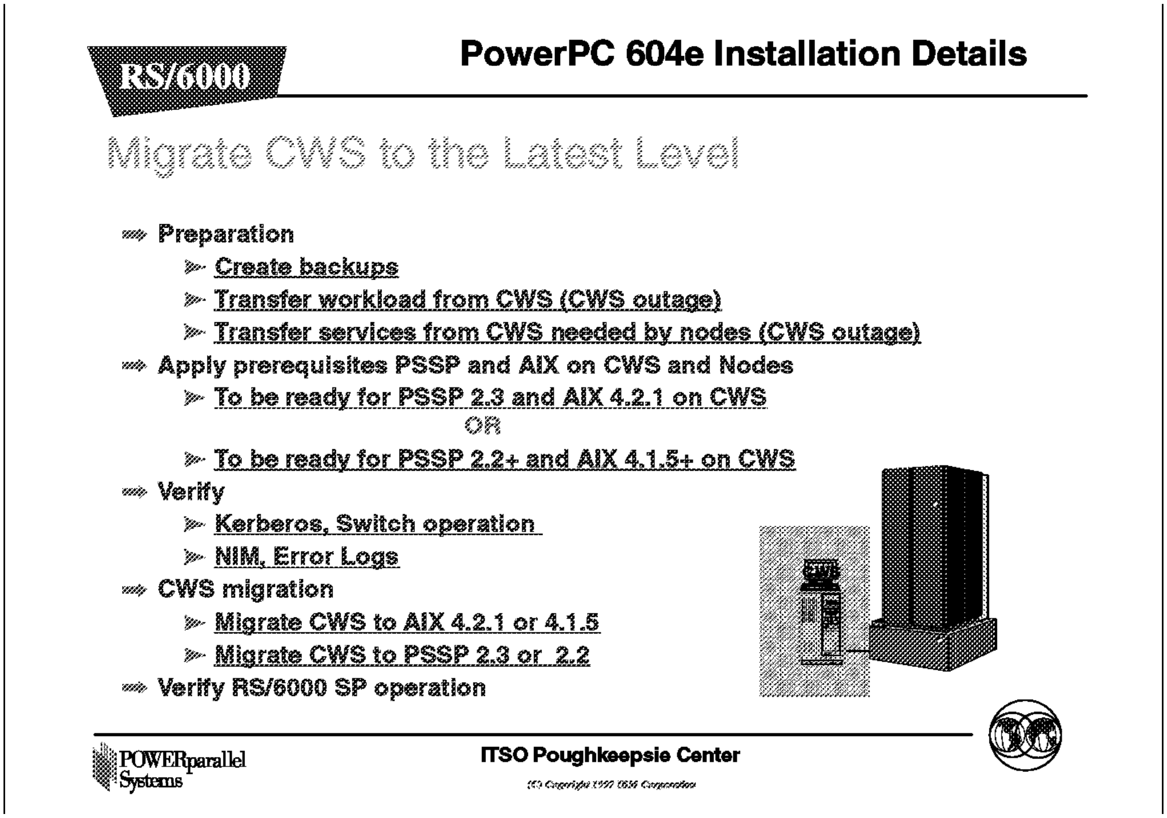
POWERparallel Systems

ITSO Poughkeepsie Center

There are five major steps required to install a 604e High node.

1. First the preparation. Perform system preparation, such as creating backups and transferring the CWS workload and services (if CWS has to be migrated). For example, nameserver service.
2. Install all prerequisites(PTFs, prereq, coreq, ifreq PTFs).
3. After the installation of PTFs, verify that the system is still working as expected.
4. Perform the CWS migration if needed, and install the High node.
5. Verify if the new node can be unfenced. Verify RS/6000 SP operations.

## 1.5.1 Migration Details



RS/6000

**PowerPC 604e Installation Details**

Migrate CWS to the Latest Level

- **Preparation**
  - Create backups
  - Transfer workload from CWS (CWS outage)
  - Transfer services from CWS needed by nodes (CWS outage)
- **Apply prerequisites PSSP and AIX on CWS and Nodes**
  - To be ready for PSSP 2.3 and AIX 4.2.1 on CWS

    OR
  - To be ready for PSSP 2.2+ and AIX 4.1.5+ on CWS
- **Verify**
  - Kerberos, Switch operation
  - NIM, Error Logs
- **CWS migration**
  - Migrate CWS to AIX 4.2.1 or 4.1.5
  - Migrate CWS to PSSP 2.3 or 2.2
- **Verify RS/6000 SP operation**

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

As previously noted the CWS must be at the latest level, so perform this migration first, if necessary.

For example AIX 4.2.1 gives you the possibility to have file up to 64GB.

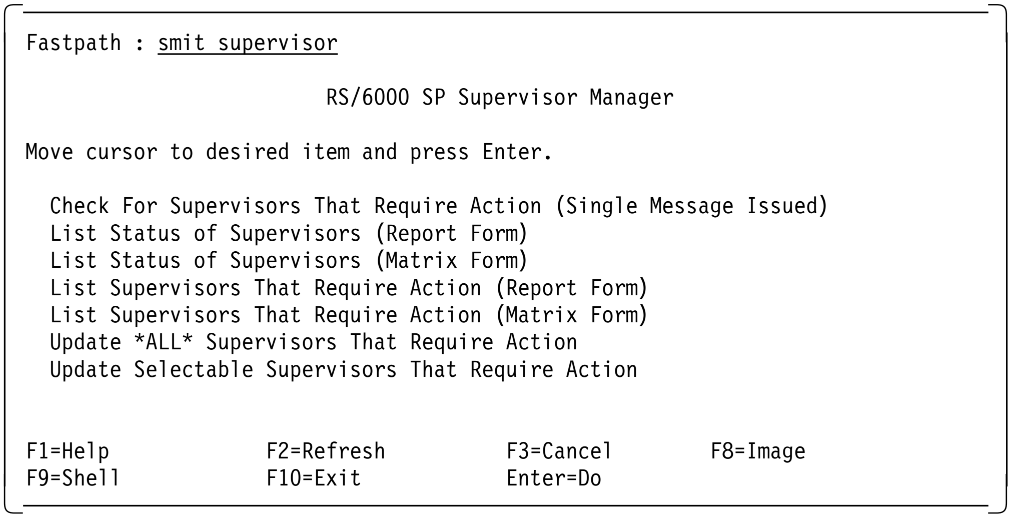
CWS migration:

- Preparation.
  - Make a backup of the CWS. Make a mksysb of the rootvg volume group and backup the other volume groups.
  - Transfer the workload and services from the CWS to another RS/6000 or to a node. For example, nameserver service, NIS service or job scheduler.
- Apply prerequisites. You must apply the latest PTFs for PSSP and AIX prior to migrating the CWS.
- Verify. After applying PTFS, verify if the CWS is working as expected. Verify Kerberos, Switch operation, LPP, errors log. Issue the following commands:
  - `klist` for Kerberos.
  - `spmon -d` for host_responds and switch_responds.
  - `errpt -a|pg` for the error log.
  - `lppchk` to check if there are filesets that needs installation or upgrade.

- CWS migration. Migrate CWS to AIX 4.2.1 and PSSP 2.3 or AIX 4.1.5 and PSSP 2.2.
- Verify RS/6000 SP. Verify if the CWS and the RS/6000 SP are still working as expected. For more details about CWS migration and verification see the section Migration Consideration in this book or *PSSP for AIX Installation and Migration Guide* SG23-3898.

After migrating the CWS, you can install the 604e High node.

If this is the first high node in an existing RS/6000 SP, you probably have to install a new frame supervisor card and download the microcode from the CWS to the frame supervisor card.



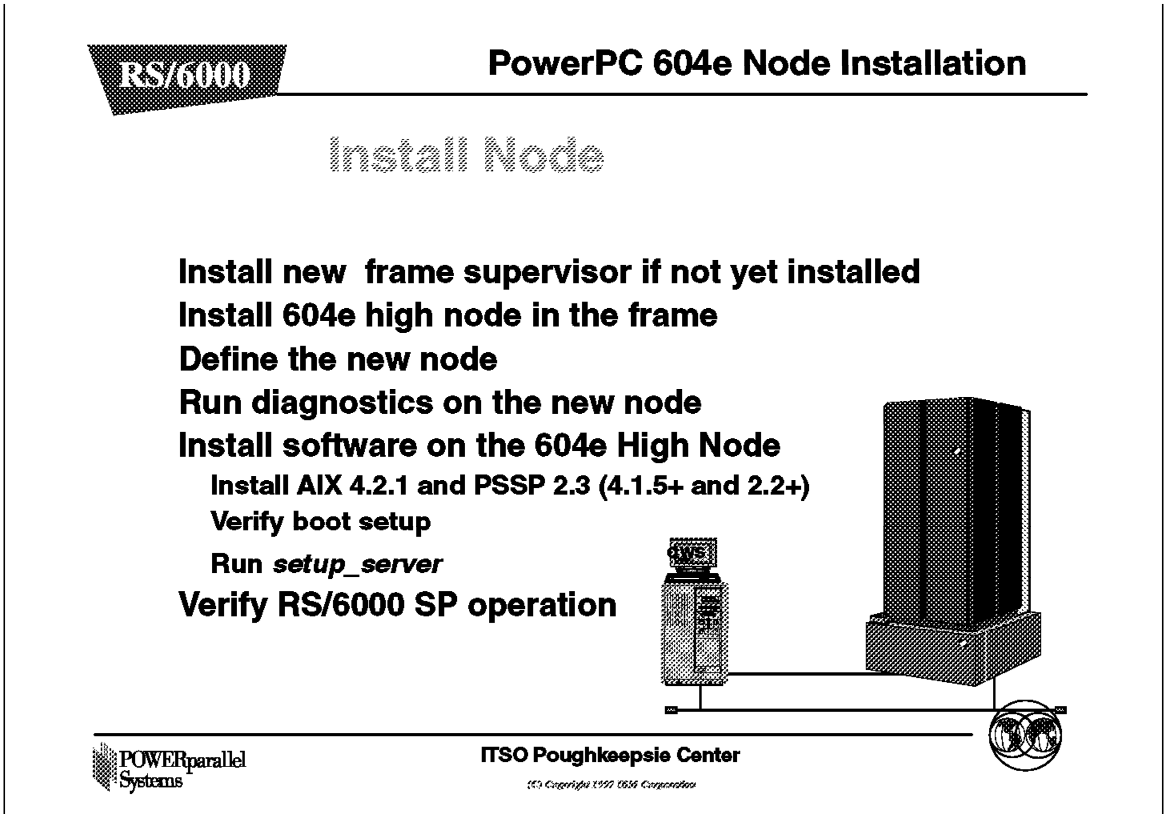
```
Fastpath : smit supervisor

                         RS/6000 SP Supervisor Manager

Move cursor to desired item and press Enter.

  Check For Supervisors That Require Action (Single Message Issued)
  List Status of Supervisors (Report Form)
  List Status of Supervisors (Matrix Form)
  List Supervisors That Require Action (Report Form)
  List Supervisors That Require Action (Matrix Form)
  Update *ALL* Supervisors That Require Action
  Update Selectable Supervisors That Require Action


F1=Help             F2=Refresh          F3=Cancel        F8=Image
F9=Shell            F10=Exit            Enter=Do
```

The above figure gives the SMIT screen used to check and download the microcode for the supervisor card.

Depending on the load of the serial port, this can take up to 30 minutes. Opening an *S1term* gives you a view what is happening on the node supervisor.

For a detailed explanation see *PSSP for AIX Installation and Migration Guide* SG23-3898.

## 1.5.2 Installation Details



RS/6000 PowerPC 604e Node Installation

Install Node

Install new frame supervisor if not yet installed

Install 604e high node in the frame

Define the new node

Run diagnostics on the new node

Install software on the 604e High Node

- Install AIX 4.2.1 and PSSP 2.3 (4.1.5+ and 2.2+)
- Verify boot setup
- Run *setup_server*

Verify RS/6000 SP operation

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The above figure list the steps required to install 604e High node.

Installing the software on the High node:

- Configure the CWS as boot/install server. Configure the install image to be AIX 4.2.1(AIX 4.1.5) or a mksysb image.
- Set code_version and lpp_source to the high node to be:
  - PSSP-2.3 and AIX 4.2.1 or
  - PSSP-2.2 and AIX 4.1.5
- Set boot/install option to overwrite install.
- Verify boot setup, using following commands:
  - splstdata -b
  - splst_version -t
- Perform the command setup_server.
- Network/boot the high node.
- Refresh system partitioning by executing syspar_ctrl -r.
- Verify the RS/6000 SP:
  - Verify host_responds.

  - Verify switch operation.
  - Verify Kerberos.
  - Verify error logs, and so forth.

  For a detailed explanation see *PSSP for AIX Installation and Migration Guide* SG23-3898.

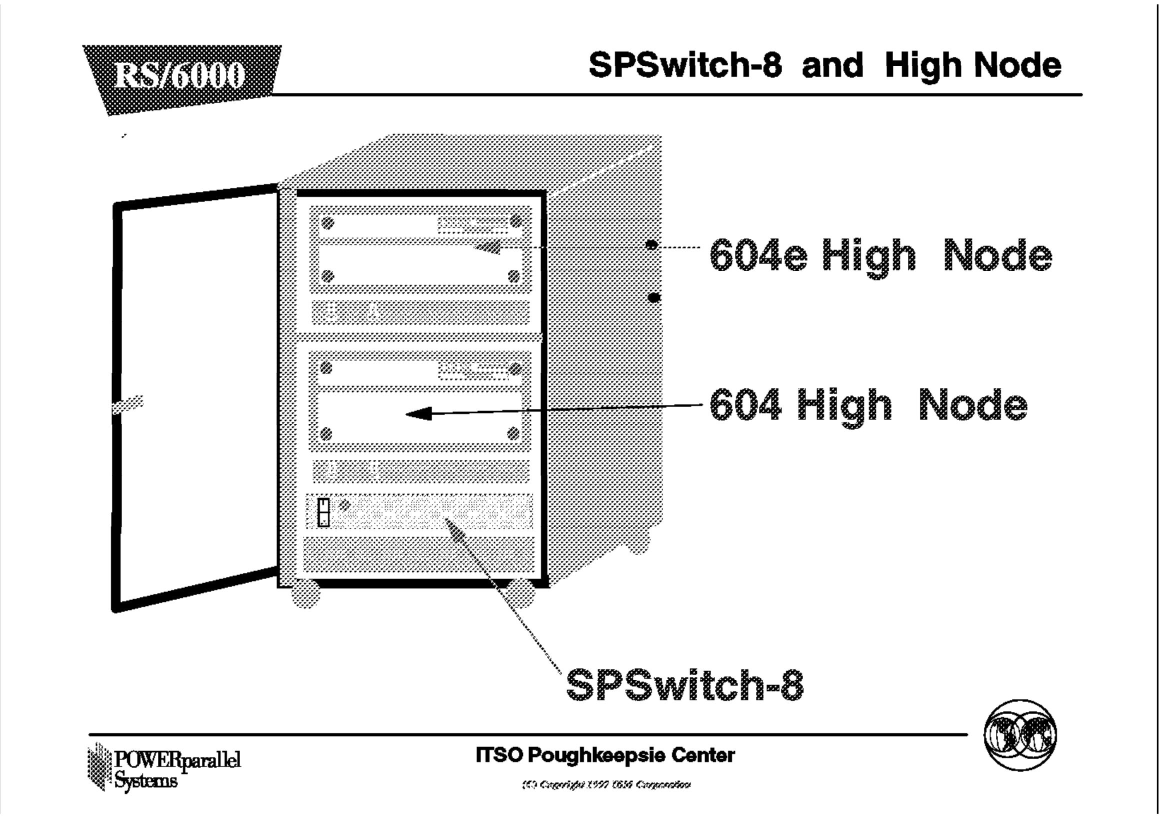
**Note:** When upgrading an 604 High node to an 604e High node, the CPU drawer is replaced. This means a change in CPU-ID. If you have applications who have an encrypted key using the CPU-ID in order to work, then keys will have to be requested from the appropriate vendor.

For example, if your application is SAP based, you may to contact this vendor to obtain new keys that corresponds with your new CPU-ID.

**Note:** The hardware upgrade from an 604 to a 604e High node consist:

- New CPU drawer.
- New Media drawer.

# Chapter 2. SPSwitch-8 and High Nodes



RS/6000

SPSwitch-8 and High Node

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The SP Switch provides low-latency, high-bandwidth communication between nodes, supplying a minimum of four paths between any pair of nodes.

The switch speeds up file transfer, remote procedures, and TCP/IP. With the switch you have a reliable communication network with tremendous throughput.

There are four types of switches:

- The High Performance Switch (HiPS).
- HiPS LC-8 Switch. The HiPS has sixteen ports, the HiPS LC-8 has eight ports.
- The SPSwitch (SPS).
- The SPSwitch-8 (SPS-8). The SPS has sixteen ports, the SPS-8 has eight ports.

The SPS switch and the HiPS switch are not compatible and therefore cannot be mixed.

If you need to add more nodes and more switch ports, you must replace the HiPS switch with SPS switches. The HiPS is out of production.

The SPS connects each SP node to the switch fabric. Switches can interconnect. Up to four SPS switches can be connected together without the use of a supplementary SPS switch for the inter switch connections.

The SPS-8 switch *cannot* be connected to another SPS or SPS-8 switch. The switch has eight node ports, but no inter-switch ports.

The capabilities of the switch are:

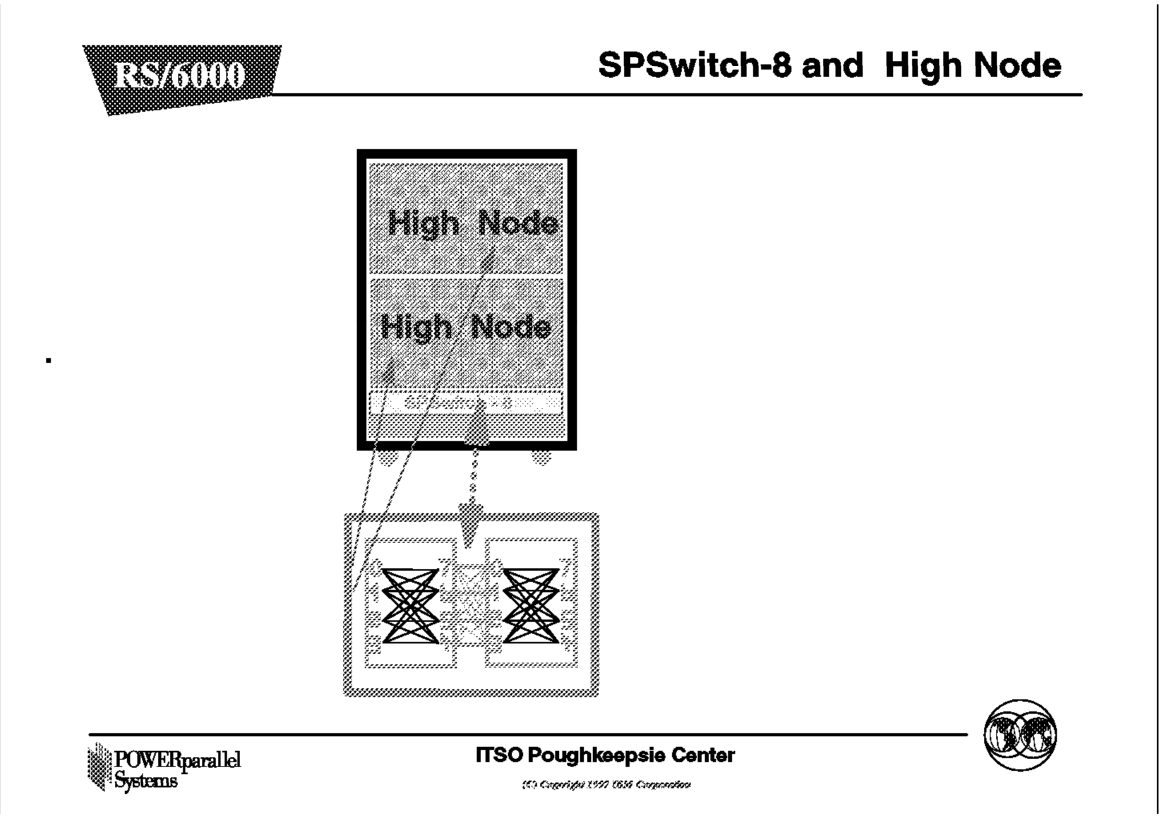
- Interframe connectivity and communication
- Scalability
- Support for Internet Protocol (IP)
- Error detection and fault isolation
- Concurrent maintenance for the nodes (Fencing)
- Constant latency and bandwidth. The throughput does not decrease with the addition of more connections. Other communication networks, such as Ethernet, have a saturation point.

PSSP 2.3 supports the 604e High node on the SPS-8 switch.

For more information, see 2.5, “Switch and Node Support” on page 37 for PSSP and High Node support details.

The following sections discuss the possible configurations and available frames and offer more information about High Nodes.

## 2.1 604e High Node and 49-inch Frame



RS/6000

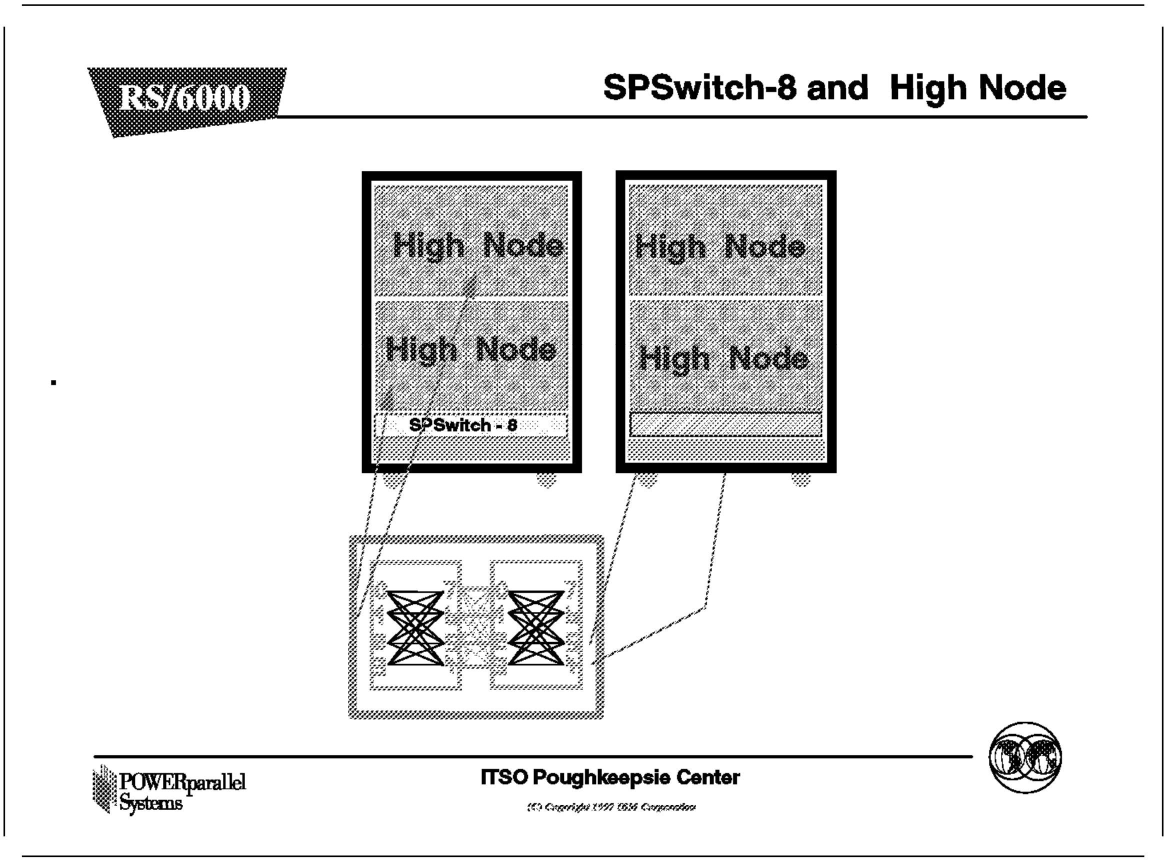
SPSwitch-8 and High Node

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The above and following figures shows that you can have four 49-inch racks with PowerPC 604e High nodes or 604 High nodes and an SPS-8 switch. With PSSP 2.3 or PSSP 2.2 plus PTFs, you have the possibility to connect the 604e High nodes to the SPS-8 switch.



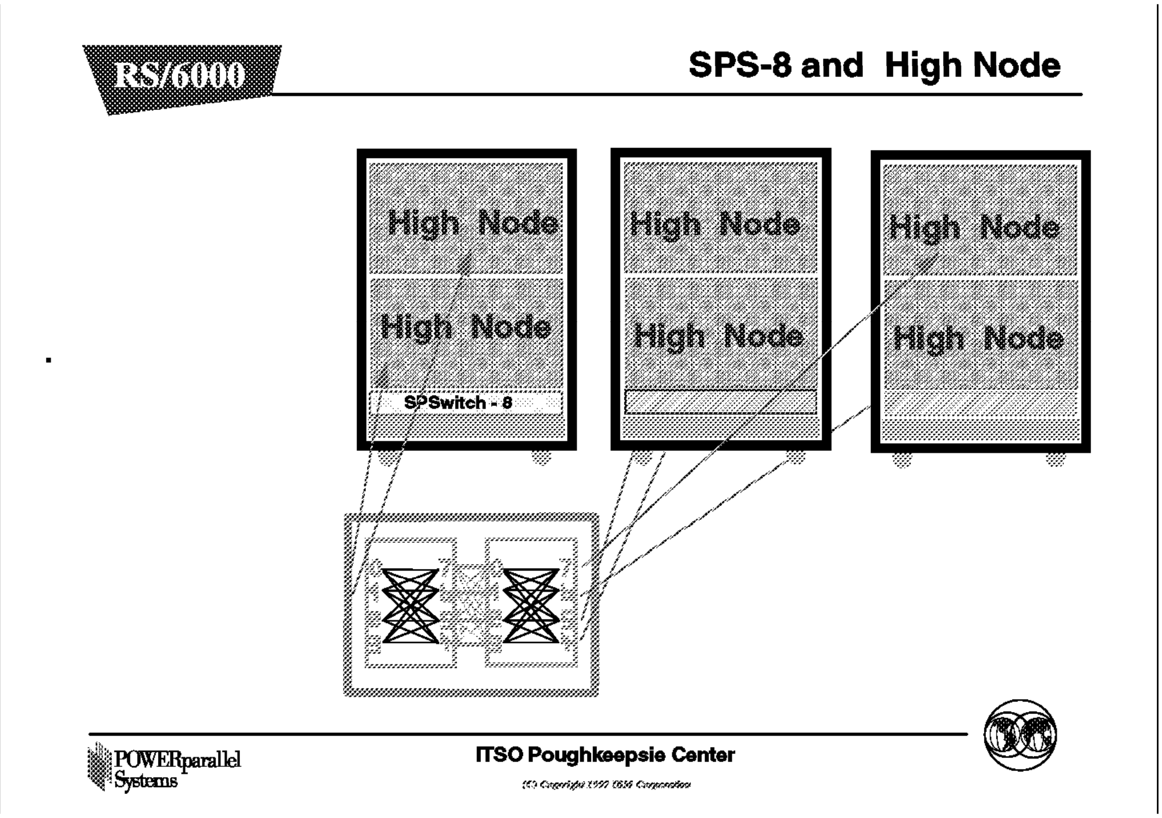
RS/6000

## SPSwitch-8 and  High Node

POWERparallel Systems

ITSO Poughkeepsie Center

This figure shows four High nodes connected to the SPS-8 switch. When connected to different switch chips, you can have two partitions.



SPS-8 and High Node

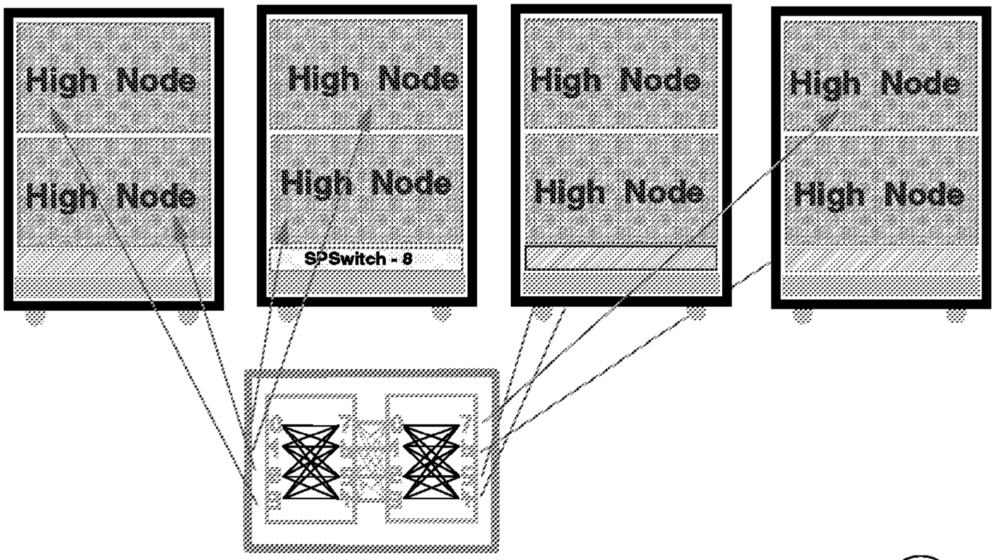
POWERparallel Systems

ITSO Poughkeepsie Center

This figure shows six High nodes connected to the SPS-8 switch.

RS/6000

## SPS-8 and High Node



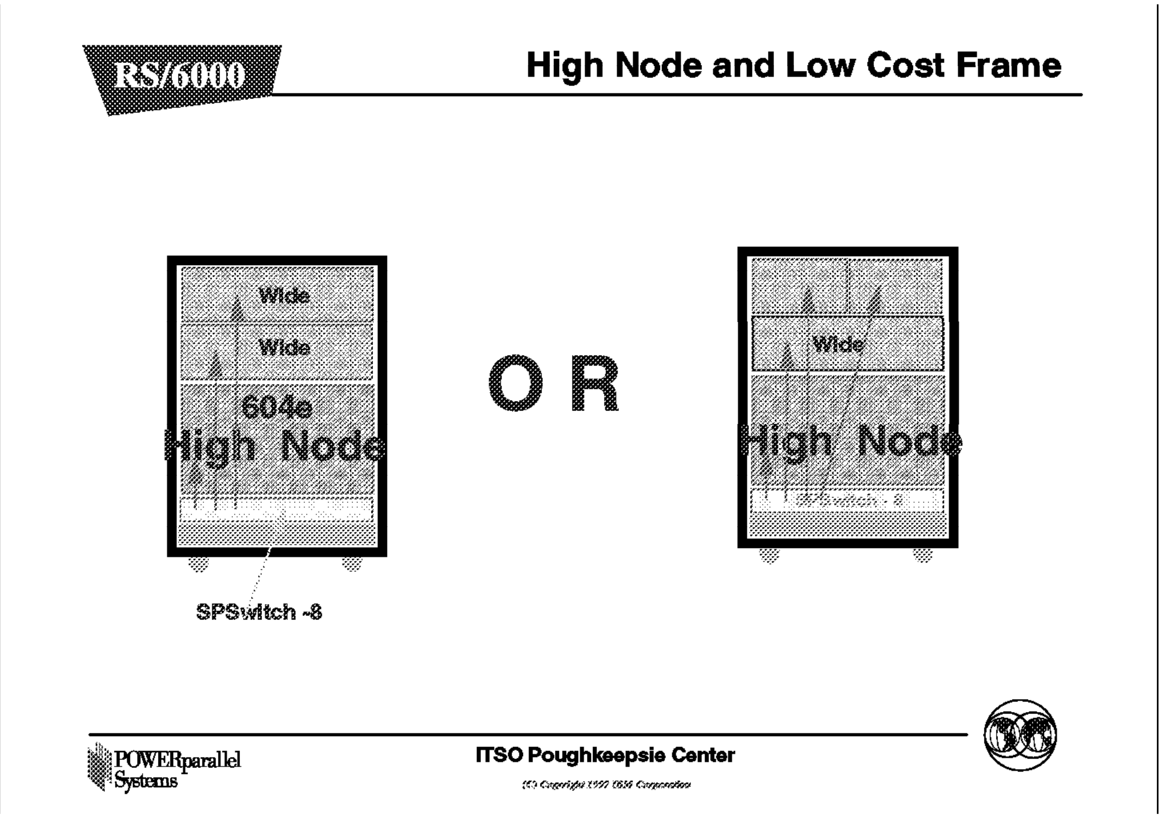
POWERparallel Systems

ITSO Poughkeepsie Center

You can have eight High nodes connected to the SPS-8 switch. This means, a RS/6000 SP system can have maximum four short frames.

**Note:** The HiPS LC-8 Switch is not supported in the short frame.

## 2.2 SPS-8 and High Nodes



High Node and Low Cost Frame

POWERparallel Systems

ITSO Poughkeepsie Center

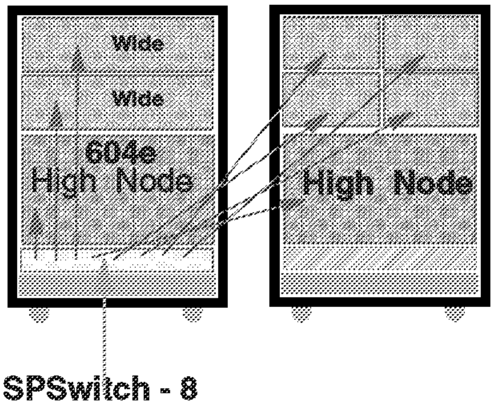
(C) Copyright 1997 IBM Corporation

The above figure illustrates some examples of the possible configurations with the combination of the 604e high nodes and the 49-inch short frame (low cost or low boy frame). With the SP_switch-8 technology, the internals of the switch design contains two switch chips and each chip has four ports that connects to the nodes. This design makes it possible to have two partitions with the 49-inch frame and the SP_switch-8. Unlike the HiPS-8 (LC-8 switch) which the design has one switch chip and system partitioning was note supported.

**Note:** It is important to keep in mind that the introduction of the 604e high nodes in the 49-inch frame makes it possible to have a mixture of the different node types within the frame. However, this combination may create the possibility to have more than eight nodes in a multi Low boy frame configuration. In such a case, it is vital to remember that the recommendation is to have all nodes connected to the switch and the existence of additional nodes beyond eight switch ports is not recommended.

RS/6000

## High Node and Low Cost Frame

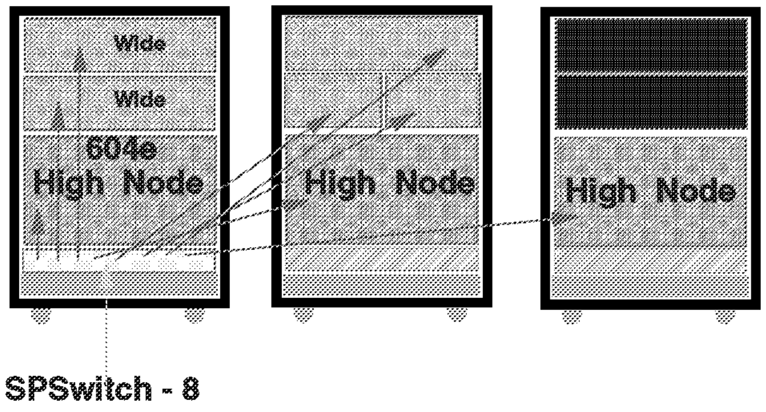


POWERparallel Systems

ITSO Poughkeepsie Center

The 49-inch rack can contain a maximum of eight thin nodes and the 604e high node occupies four thin node slots. It is possible to have a configuration that consist of one 604e High node, two wide nodes and two thin nodes. This configuration will utilize a total of five switch ports and the remaining three switch ports can to used in a second frame to connect three additional nodes to the same switch. This also provides the advantage of having two different partitions.

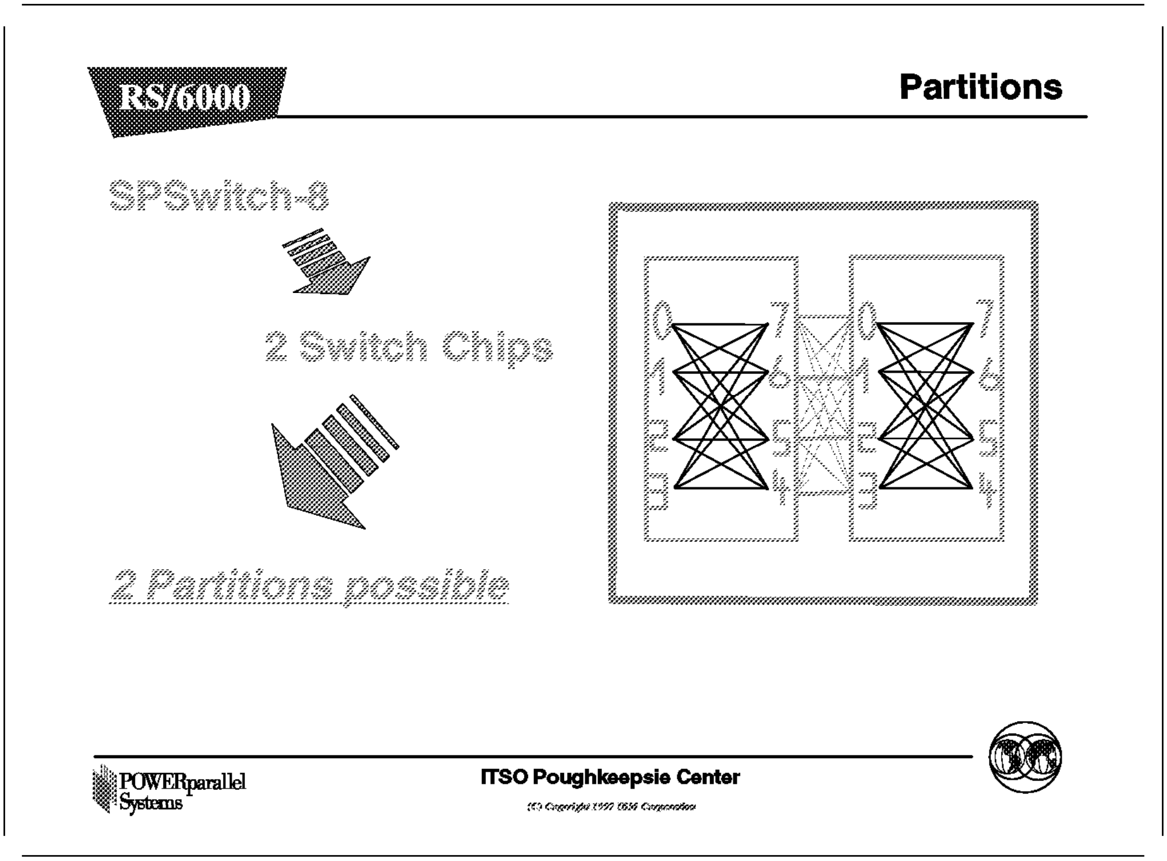
## High Node and Low Cost Frame



ITSO Poughkeepsie Center

The figure above depicts the maximum configuration that is possible with the 49-inch frame and the SP-switch-8. It consist of a combination of two 604e High Nodes, two wide nodes and four thin nodes. With this example, each node is connected to a switch port and all the switch ports will be used.

## 2.3 SPS-8 Switch Chips


RS/6000

Partitions

SPSwitch-8

2 Switch Chips

2 Partitions possible

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The SPS-8 switch has two switch chips, enabling you to use partitions. An SPS-8 switch can have one of two configurations:

- A single partition with a maximum of eight nodes.
- Two partitions with a maximum of four nodes each.

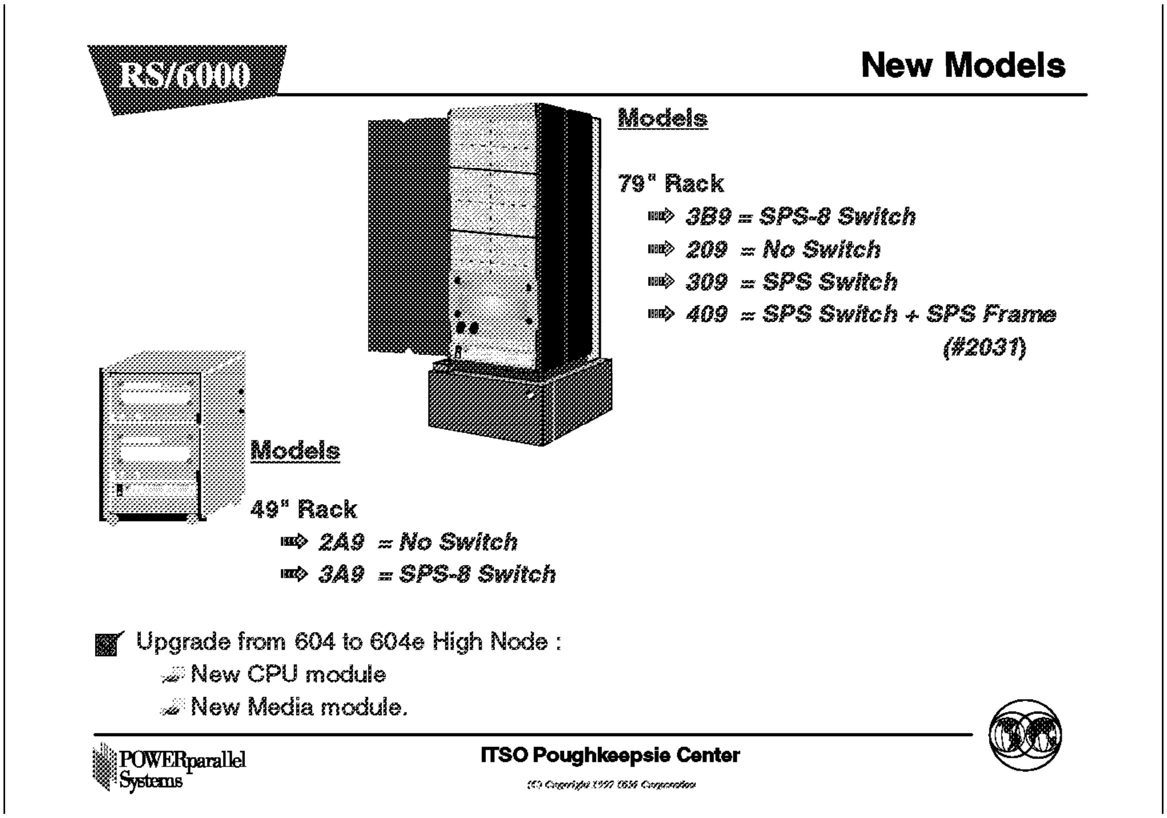
**Note:** The HiPS LC-8 switch had only one chip, so only one partition was possible, in other words, no partitioning.

**Important**

Switchless systems can have partitions. However, if you have a switchless system, and you add a switch, you may have to reconfigure your system partition choice. You may have to reinstall *ssp.top* to remove any special switchless partitions. If you use one of the supported switch partitions, your layout will be usable when you install a switch.

For a detailed explanation of partitioning, see the chapter on partitioning in the *IBM RISC System/6000 SP Scalable POWERparallel Systems planning*, GA22-7281.

## 2.4 New Models



RS/6000

New Models

Models

79" Rack

- 3B9 = SPS-8 Switch
- 209 = No Switch
- 309 = SPS Switch
- 409 = SPS Switch + SPS Frame (#2031)

Models

49" Rack

- 2A9 = No Switch
- 3A9 = SPS-8 Switch

Upgrade from 604 to 604e High Node :

- New CPU module
- New Media module.

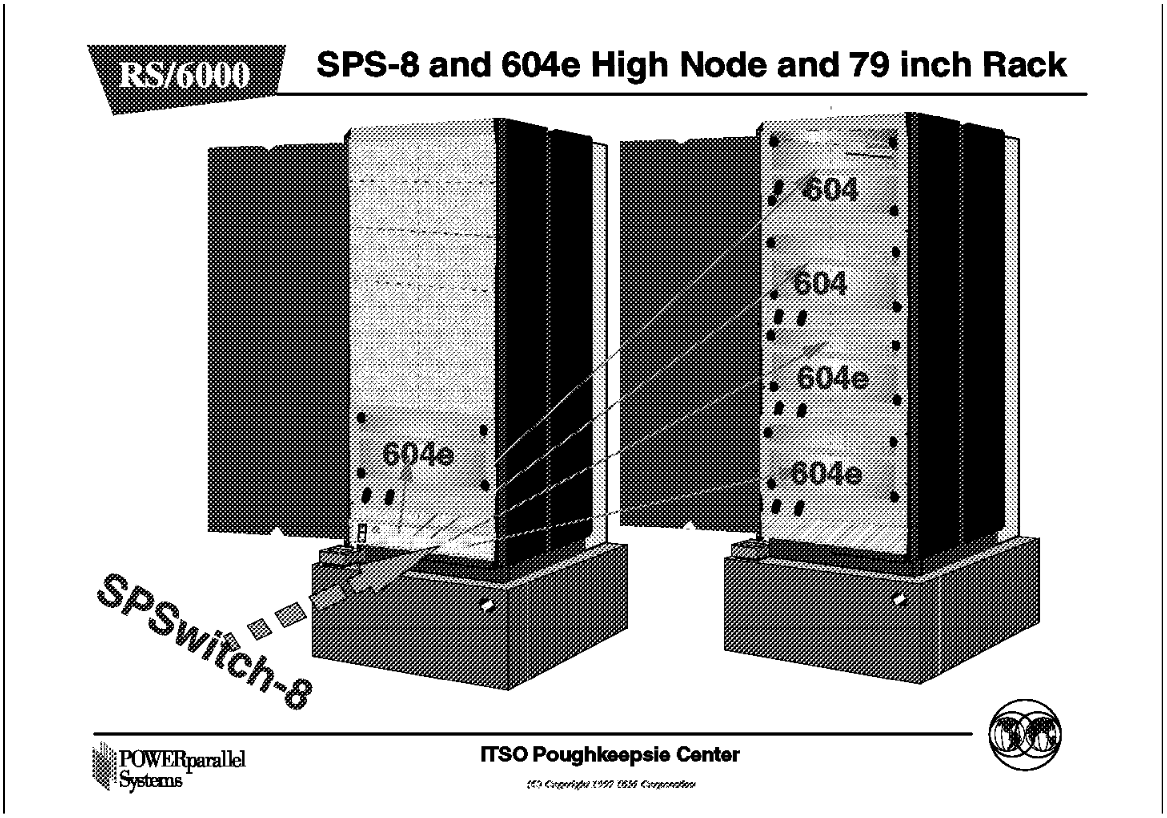
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The above and following picture shows possible configurations and the new models. When the SPS-8 switch is installed in a 79-inch rack, a maximum of two 79 inch-racks can be connected to the SPS-8.

You can install eight High Nodes in the two racks.



RS/6000

SPS-8 and 604e High Node and 79 inch Rack

POWERparallel Systems

ITSO Poughkeepsie Center

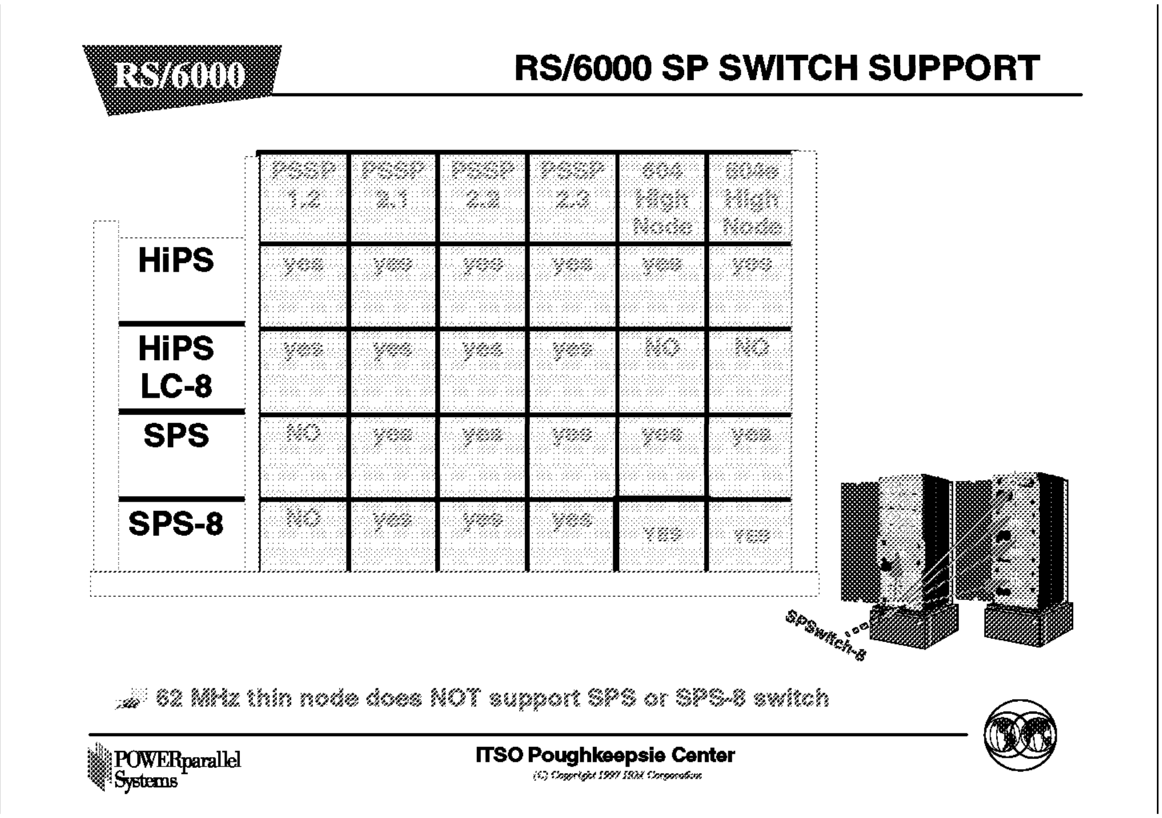
The SPS-8 switch with High node installed in a 79-inch rack is supported with PSSP 2.3. You can have a maximum of two 79-inch racks with eight High Nodes connected to the SPS-8 switch.

Table 1. New SMP Models

| Rack | Model | Description |
|---|---|---|
| 49″ | 2A9 | No Switch |
| 49″ | 3A9 | SPS-8 Switch |
| 79″ | 209 | No Switch |
| 79″ | 309 | SPS Switch |
| 79″ | 409 | SPS + SPS frame #2031 |
| 79″ | 3B9 | SPS-8 Switch |

**Note:** All models have a 604e High Node in Slot 1. Model numbering depends on what kind of node is installed in the first position.

## 2.5 Switch and Node Support



RS/6000

RS/6000 SP SWITCH SUPPORT

| | PSSP 1.2 | PSSP 2.1 | PSSP 2.2 | PSSP 2.3 | 604 High Node | 604e High Node |
|---|---|---|---|---|---|---|
| HiPS | yes | yes | yes | yes | yes | yes |
| HiPS LC-8 | yes | yes | yes | yes | NO | NO |
| SPS | NO | yes | yes | yes | yes | yes |
| SPS-8 | NO | yes | yes | yes | YES | YES |

62 MHz thin node does NOT support SPS or SPS-8 switch

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Following table shows which switch is supported on each version of PSSP and with which type of High Node.

Table 2. SP Switch Matrix

| | PSSP 1.2 | PSSP 2.1 | PSSP 2.2 | PSSP 2.3 | 604 High Node | 604e High Node |
|---|---|---|---|---|---|---|
| HiPS | Y | Y | Y | Y | Y | Y |
| HiPS-LC8 | Y | Y | Y | Y | N | N |
| SPS | N | Y | Y | Y | Y | Y |
| **SPS-8** | N | Y | Y | Y | N | Y |

## 2.5.1 Switch Log Files

CSS trace and log files are found in the /var/adm/SPlogs/css directory on every node and the Control Workstation (CWS). Additionally, the fault_service daemon places entries in the AIX error log.

Log files contain information that relates to the operation of the switch and/or adapter, Ecommands being issued, adapter outages, and so forth.

Trace files are meant to track and/or trace the various pieces of CSS code. These files contain the “good” as well as the “bad” things that are happening or have happened in the communications subsystem.

With PSSP 2.3, log files have more information. This is also applicable for systems at PSSP level 2.2 at a PTF level greater then 6.

### 2.5.1.1 The flt Log File

The /adm/SPlogs/css/flt log file exist on any node that is or was a Primary node. This file is used to log hardware error conditions found on the switch, recovery actions taken by the fault_service daemon and general operations that alter the switch configuration.

Information that can be found in this file:

- Disabled switch chips, nodes, and ports
- Switch initialization error status
- Switch error recovery
- Broadcast service packets failures
- Estart (switch initialization) if it was used as a command
- Estart if it was used as a recovery action
- Primary node takeover
- Eunfence and Efence port operations
- Switch scan failures
- Switch port disabled of the primary node
- Route generation
- Fault_service signals
  - SIGBUS
  - SIGTERM
  - SIGDANGER
- Phase 2 of switch initialization retries

**Note:** Switch initialization has two phases:

- Discovering who is out there, that is, who is connected to the switch fabric and answering.

- The configuring phase, that is, looking up whether the nodes that are responding are corresponding with ones that are in the topology file.

#### 2.5.1.2 The worm.trace File

The worm.trace file is found in the /var/adm/SPlogs/css directory. It exists on every node in the SP system. It contains trace information for the last run of the css0 adapter diagnostics.

The file creation time is Midnight Dec 31 1969. This is because this trace is created during the Power On Self Test or POST. This is due to the fact that the node time is not set when the time diagnostics run.

#### 2.5.1.3 The out.top File

The out.top file is found in the /var/adm/SPlogs/css directory. It exists on every node in the SP system that was or is the primary node. It is basically a copy of the topology file with link and device status filled in.

This file is modified on the primary node every time the switch is initialized, whether that is from executing the Estart command or by the fault_service daemon running it as a recovery action. It contains the current link and device status of the SP system.

An entry in the file looks something like this:

```
s 14 2 tb0 9 0          E01-S17-BH-J32 to E01-N10
```

The above line can be read as follows: switch 1, chip 4, port 2 is connected to switch node 9. The switch is located in frame E01, slot 17. Its bulkhead connection to the node is Jack 32.

The node is also in Frame E01 and its node number is 10.

There is no additional status following this entry, so it can be assumed that everything is okay with the link.

The following entry contains additional status information:

```
s 14 2 tb0 9 0  E01-S17-BH-J32 to E01-N10
  -4R: device has been removed from network = faulty
   (link has been removed from network or mis-wired - faulty)
```

The above example means: device tb0 9 has a device status of -4.

The device status of the node is also displayed in text format as `device has been removed from network - faulty`.

The message guide for PSSP 2.3 contains more information on both link and device status.

### 2.5.1.4 The rc.switch File

The rc.switch.log file is found in the /var/adm/SPlogs/css directory. It can exist on any TB2 or TB3 node on the system. It is created or updated every time `rc.switch` is issued on a node. Additionally, the current rc.switch.log is written to the rc.switch.log.previous file. The file contains the following information:

- Date and time information on when rc.switch was executed
- Date and time information on when rc.switch finished
- The hostname of the node
- The node number
- Adapter_config_status for the node
- The switch_node_number of the node
- The switch chip the node is attached to
- The switch board the node is attached to
- The switch chip port the node is attached to
- The IP_switch_netaddr and IP_switch_netmask
- Is IP_switch_ARP_enabled?
- Is the type of adapter TB2 or TB3?
- The parameters used for ifconfig and fault_service_Worm_RTGxx
- Completion status of rc.switch on this node

**Note:** Switch node numbering starts from zero. Switch node number plus one gives the node number.

### 2.5.1.5 The dtbx.trace File

The dtbx.trace file is found in the /var/adm/SPlogs/css directory. It exists on every node in the SP system. It contains trace information for the last run of the css0 adapter diagnostics. The file creation time of this file is Dec 31, 1969. This is because the trace was created during the Power On Self Test or POST. The node time is not set when diagnostics run.

1. For TB2, this file contains the following diagnostics:
   - Diagnostic setup
   - Clock selection
   - POS testing
   - MSMU testing
   - DRAM testing
   - ECC testing
   - Interrupt testing
   - BiDirectional FIFO testing

- DMA testing
- Completion status

2. For TB3, this file contains the following diagnostics:
   - Diagnostic setup
   - Clock selection
   - Vital Product Data collection
   - POS testing
   - TBIC FIFO testing
   - SRAM testing
   - TBIC self-test
   - TBIC TOD testing
   - Interrupt testing
   - DMA testing
   - Completion status

**Diagnostic setup** This consists of making sure that ODM is configured properly, that is, that device css0 is configured and that diagnostics can get exclusive use of the device.

**Clock selection** There are a number of clocks available to both the TB2 and TB3 adapters. Both adapters have their own internal clock. Also, each adapter has external clock choices.

For TB2, a Data Cable or a discrete wire (Gore cable) clock are available. For TB3, only a Data Cable is available.

For either adapter to complete diagnostics successfully, one of the external clock sources must be available for test purposes. If these clocks are not available, diagnostics are still attempted on the internal clock. If the diagnostics pass on this internal clock, a failure code is returned. This is because, even though the adapter is okay, without an external clock source the card is useless for communicating with other switch adapters.

The clock source selection process for TB2 and TB3 is different.

For TB2, clock selection is as follows:

1. Test the internal clock, if it is not operational, it is assumed the adapter is bad and no further testing is attempted.
2. Select the Data cable, if it is available for testing, write the data_cable file to the /usr/adm/SPlogs/css.
3. If the Data cable is not available, select the Gore clock. If it is available for testing, write the gore_cable file to the /usr/adm/SPlogs/css directory and proceed with the test.
4. If no Data cable or Gore clock is available for testing, select the internal clock. Once the tests have completed, mark the diagnostics as failed because no external clock was available.

For TB3, clock selection is as follows:

1. First test the internal clock. If it is not operational, the adapter is bad and no further testing is attempted.

2. Select the Data cable. If it is available for testing, proceed with the test.

3. If the Data cable is not available, select the internal clock and proceed with the tests. Once the tests have completed, mark the diagnostics as failed because no external clock was available.

**Vital Product Data collection (TB3 only)**

For TB3, the Vital Product Data VPD is read from the adapter EPROM and written to the dtbx.trace file. The VPD includes the following:

- Part number
- EC level
- Serial number
- FRU name
- Manufactures code
- Device description

**POS Testing**

POS testing consists of reading and writing test data to the adapters′ Programmable Option Select registers. It tests both the functionality of specific register bits, as well as patterns where applicable.

**MSMU Testing (TB2 only)**

The Memory and Switch Management Unit is made up of 32 registers as well as three FIFO units. Testing of this “unit” consists of functionally testing these FIFOs and registers.

**TBIC FIFO Testing (TB3 only)**

Test the FIFOs found on the TBIC chip.

**DRAM testing (TB2 only)**

The DRAM is loaded with the microcode and the remaining areas of the memory are tested by writing data.

**SRAM testing (TB3 only)**

Testing SRAM on the TB3 adapter is tested by writing data patterns to the SRAM.

**ECC testing (TB2 only)**

It generates and checks the eight bits of ECC on both data and address.

**TBIC Self Test (TB3 only)**

TBIC self test is a resident function of the TBIC chip.

**TBIC TOD testing (TB3 only)**

The Time-of-Day register on the TBIC chip is tested.

**Interrupt Testing**

During the interrupt test each of the possible interrupts is forced and then checked.

**Bidirectional FIFO testing (TB2 only)**

The FIFO is tested by running test patterns through the FIFOs. The patterns are loaded and unloaded and checked for validity.

**DMA Testing**

Diagnostics are provided to test the DMA functions of both the TB2 and TB3 adapters.

**Completion Status**

The easiest way to determine where to look in the dtbx.trace file is to view the completion status at the bottom of the file. For both TB2 and TB3, the SRN number at the bottom of the file should help you determine where in the file to start looking. To decode these three digit SRNs, use the tables supplied in the Adapter Diagnostic SRNs section.

#### 2.5.1.6 The dtbx.failed.trace File

The dtbx.failed.trace file is found in the /var/adm/SPlogs/css directory. It may or may not exist on a node. It contains trace information for the last failed run of the css0 adapter diagnostics.

The dtbx.trace found in this directory should be used for looking at the last run of the adapter diagnostics. This method of renaming the last failed dtbx.trace to dtbx.failed.trace is the same for both TB2 and TB3.

A file creation time of Midnight Dec 31 1969 means that the file was created during the POST (Power On Self Test). This is because the node time was not set when the time diagnostics were run.

#### 2.5.1.7 The router.log File

The router.log file is found in the /var/adm/SPlogs/css directory. It can exist on any TB3 node in the system. It is created and updated every time Route Generation is run by the fault_service daemon.

Additionally, every time new routes are generated the existing router.log file is copied to the router.log.old file. The information in the file can vary based on the algorithm used for generating the routes. The Primary node router.log contains node-to-node route information. Router.log.old contains service router information.

There are a number of circumstances when both logs contain similar information, such as when Phase2 of the worm process is reinitialized during an Estart.

On secondary nodes both router.log and router.log.old will contain only node-to-node routing for the node. The router.log file contains the following information:

- Date and time when the route table was generated

- The version of the Route Generator that was used
- The algorithm type used in generating the routes
- Either node-to-node routes or service route information for the particular node

  1. Node-to-node entry:

     ```
     ROUTE 4 8 ID(PORT): 100015(6) 100012(4) 100014(3)

     rword 4 8 0x02000000 0x00008364
     ```

     The first line contains the same information as the second, but in a more readable format.

     The first line can be read as follows:

     The route from switch_node_number 4 to switch_node_number 8 is out of switch_node_number 4 to switch_chip_number 100015,

     out of switch_chip_number 100015 on Port 6 to switch_chip_number 100012,

     out of switch chip number 100012 on Port 4 to switch_chip_number 100014,

     out of switch_chip_number 100014 on Port 3 to switch_node_number 8.

  2. Service entry:

     ```
     ROUTE 4 100012 ID(PORT): 100015(6)

     rprocsw 4 100012 0x01000000 0x00000086
     ```

     Again, the second line is identical to the first. Service routes and node-to-node routes are similar. The only difference is that service routes can go to switch chips as well as to nodes.

     The first line can be read as follows:

     The route from switch_node_number 4 to switch_chip_number 100012 is out of switch_node_number 4 to switch_chip_number 100015,

     out of switch_chip_number 100015 on port 6 to switch_chip_number 100012.

### 2.5.1.8 The scan.out File

The scan_out.log file is found in the /var/adm/SPlogs/css directory. It exists on nodes with TB3 adapters. It is created every time the ″TBIC self test″ diagnostic is run as part of css0 diagnostics.

It contains the scan ring information for the TBIC chip, following the completion of self test. This file is not formatted. It is the binary scanned latch information directly from the TBIC. Using an editor to look at it does not give any useful information.

To view the file, some sort of binary editor is required. The information in this file is for engineering purposes only.

### 2.5.1.9 The scan_save File

The scan_save.log file is found in the /var/ adm/SPlogs/css directory. It exists on nodes with TB3 adapters. It is created every time the TBIC self test diagnostic is run as part of css0 diagnostics. To view the file, a binary editor is required. The information in it is for engineering purposes only.

### 2.5.1.10 The topology.data File

The topology.data file is found in the /var/adm/SPlogs/css directory. It exists on any node that is or was the Primary node. The only valid topology.data file is the one on the current Primary node.

The following is a sample of the information contained in it:

```
Number of active node(s) seen by the Worm:
4
Number_of_linksbad: 0
The primary backup node is:
9
The following switch node(s) are active:
1
5
9
13
The topology file used by the Worm: /etc/SP/Jan.1
```

### 2.5.1.11 The css.snap.log File

The css.snap.log file is found in the /var/adm/SPlogs/css directory. It can exist on any TB2 or TB3 node. It is created every time css.snap is run, either manually or by the fault_service daemon. It contains the following information about what happened during the snap operation:

- Date and time at the time of the snap
- Which node it was executed on
- The contents of the /var/adm/SPlogs/css directory prior to css.snap
- Information on the tar and compress operations performed by css.snap
- Information about the ssp.css software product and updates to it (lslpp -i ssp.css)

#### 2.5.1.12 The daemon.stderr file

The daemon.stderr file is found in the /var/adm/SPlogs/css directory. It exists on all nodes in the SP system, whether it is a TB2 or TB3 system. The information is produced by the fault_service daemon when a software error is encountered, such as “open or close file failed.” The file length is usually 0 if no errors have occurred.

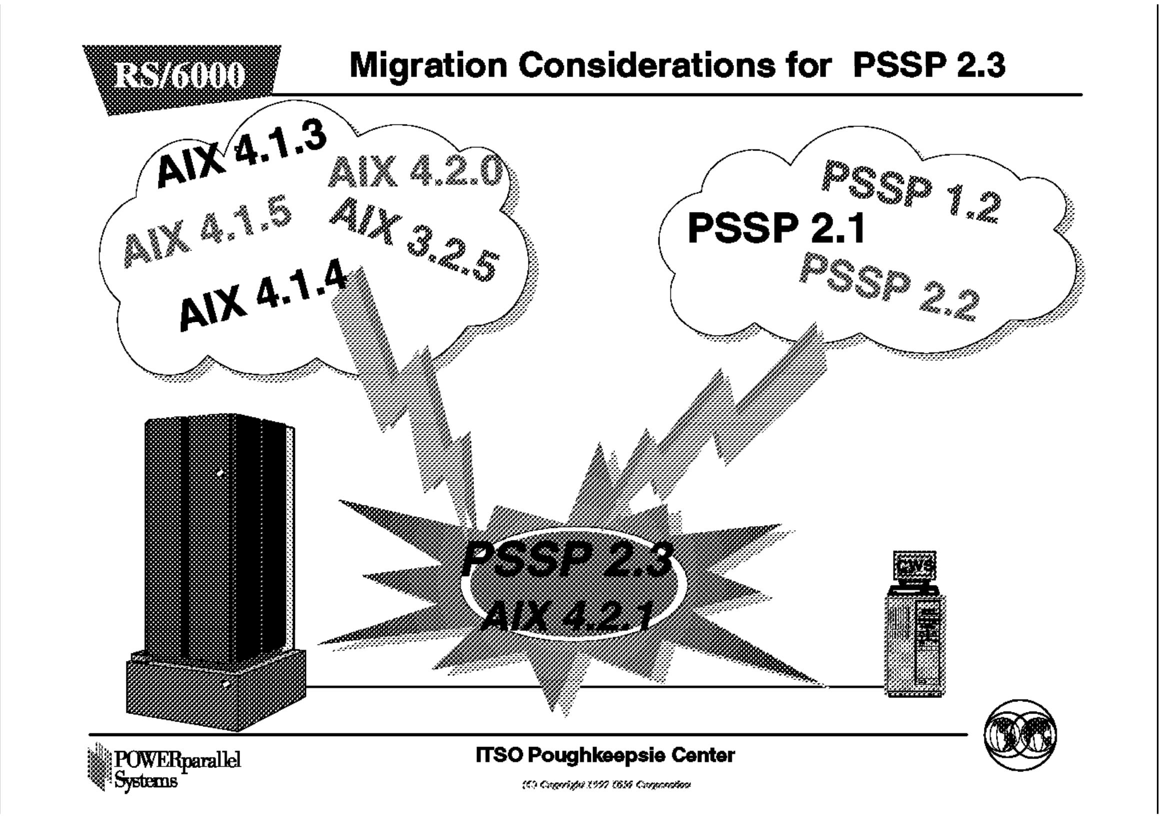
#### 2.5.1.13 The Ecommands.log File

The Ecommands.log file is found in the /var/adm/SPlogs/css directory. It can exist on any TB2 or TB3 node and on the Control Workstation. It is created or updated every time an Ecommand is issued on that particular node or CWS.

The file contains the following information:

- Date and time when a particular Ecommand was used
- Parameters used in the invocation of the Ecommand

# Chapter 3. Migration Considerations for PSSP 2.3



RS/6000 Migration Considerations for PSSP 2.3

POWERparallel Systems

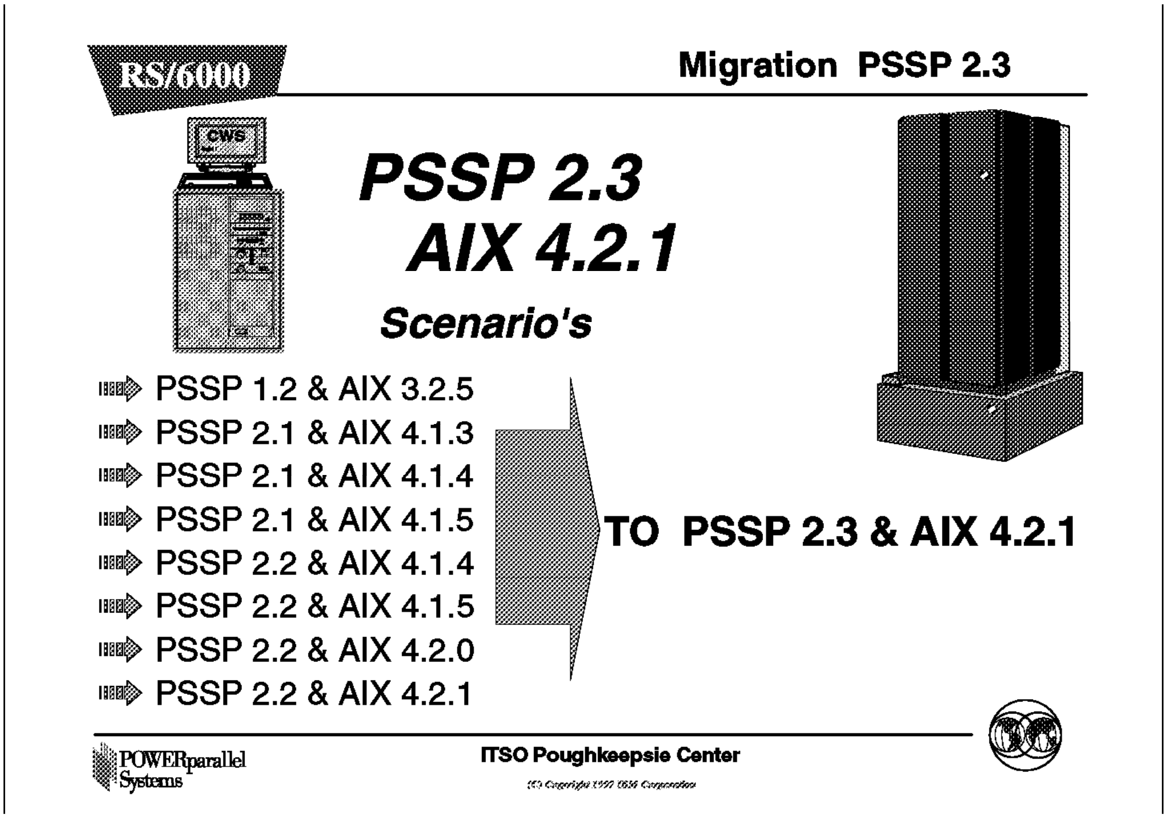
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The concept of RS/6000 SP *migration* has different meanings. The Control Workstation can be migrated, the RS/6000 SP nodes can be migrated, or you can migrate some nodes. If nodes are upgraded to a new version of AIX or PSSP, the Control Workstation must be at the highest software level.

PSSP 2.3 migration is an extension of PSSP 2.2 migration. While the process has not been changed, documentation has been improved.

## 3.1 Overview



RS/6000

Migration PSSP 2.3

PSSP 2.3

AIX 4.2.1

Scenario's

- PSSP 1.2 & AIX 3.2.5
- PSSP 2.1 & AIX 4.1.3
- PSSP 2.1 & AIX 4.1.4
- PSSP 2.1 & AIX 4.1.5
- PSSP 2.2 & AIX 4.1.4
- PSSP 2.2 & AIX 4.1.5
- PSSP 2.2 & AIX 4.2.0
- PSSP 2.2 & AIX 4.2.1

TO PSSP 2.3 & AIX 4.2.1

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This figure shows different scenarios when migrating from one level of PSSP to PSSP 2.3, and from one level of AIX to AIX 4.2.1

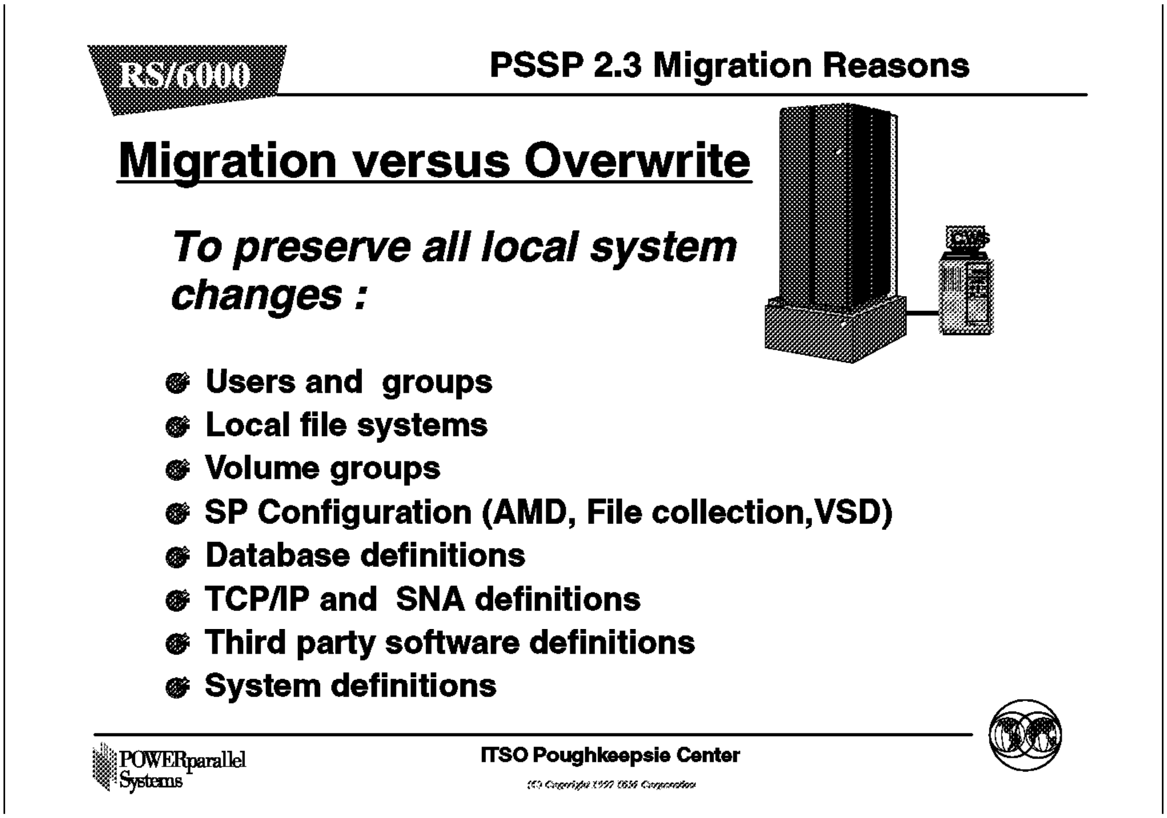
The following sections describe the process of migrating the Control Workstation to AIX 4.2.1. For more information, see the *PSSP Installation and Migration Guide*, GC23-3898, for the AIX 4.2 migration process.

The migration process from PSSP 2.1 and AIX 4.1.5 TO PSSP 2.3 and AIX 4.2.1 is explained in detail.

The migration process from PSSP 2.2 and AIX 4.1.5 to PSSP 2.3 is also explained in detail.

If your operating system is at a level less than AIX 4.1.5, follow the procedures as described in the *PSSP Installation and Migration Guide*, GC23-3898 to upgrade your Control Workstation to AIX 4.2.1.

## 3.2 Reasons for Migration



RS/6000

**PSSP 2.3 Migration Reasons**

**Migration versus Overwrite**

***To preserve all local system changes :***

- **Users and groups**
- **Local file systems**
- **Volume groups**
- **SP Configuration (AMD, File collection,VSD)**
- **Database definitions**
- **TCP/IP and SNA definitions**
- **Third party software definitions**
- **System definitions**

POWERparallel Systems

**ITSO Poughkeepsie Center**

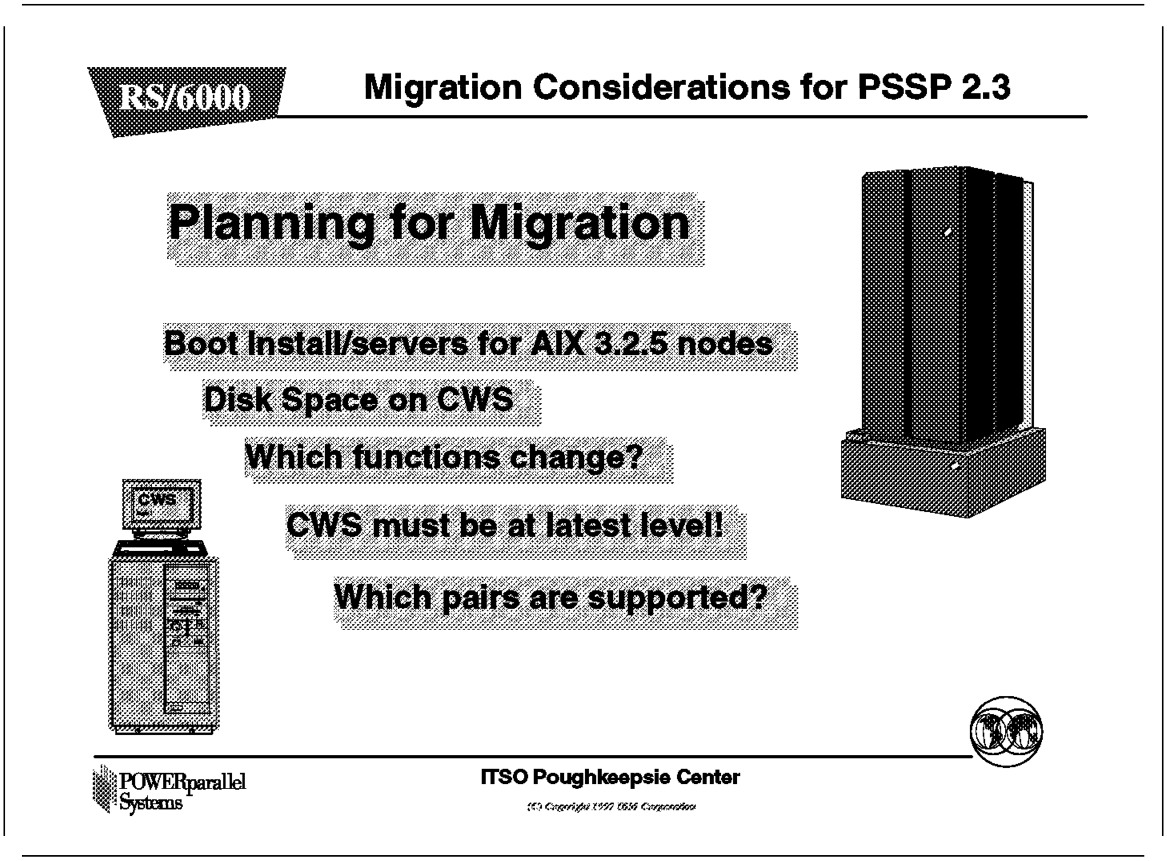
(C) Copyright 1997 IBM Corporation

The main reason for migrating is to preserve all local system changes, such as:

- Users and groups. To preserve the setting for the users, like passwords, profiles, login shells. Group definitions are also preserved when migrating.
- File systems and volume groups. Definitions such as names, parameters, sizes, directories are kept.
- RS/6000 SP setup (AMD, file collections). You do not need to customize AMD.
- Database definitions. User definition and database setting are kept.
- Network setup (TCP/IP, SNA). SNA parameters and the IP setting are retained after migration. The customized no parameters are not changed after migration.
- Third party software definitions and setup. Definitions of OEM software that depends on system settings such as TCP/IP or filesystems are still valid after migration.
- Reduced outage time. When using migration, the outage time of the CWS and node are reduced because all the settings are kept. There is no need to reinitialize and configure the CWS or node.

**Note:** The customer should not stay indefinitely in a mixed environment.

## 3.3 Planning for Migration



RS/6000 **Migration Considerations for PSSP 2.3**

**Planning for Migration**

**Boot Install/servers for AIX 3.2.5 nodes**

**Disk Space on CWS**

**Which functions change?**

**CWS must be at latest level!**

**Which pairs are supported?**

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Before starting the migration of the RS/6000 SP system, you have to plan the following:

- Where your boot install servers for the AIX 3.2.5 and PSSP 1.2 nodes will reside; that is, which nodes are the boot/install servers.
- How much disk space will be needed on the Control Workstation.
- Which functions will change
  - PSSP 2.3 replaces its use of the public domain AMD automount daemon with the AIX automount daemon, which is available as part of NFS. AMD uses the map files to define automounter control. These map files are not compatible with the AIX automounter and must be converted. See section AMD in this redbook for more information.
  - SP Print Management is removed in PSSP 2.3; that is, the SP Print Management System cannot be configured on nodes running PSSP 2.3. IBM recommends the use of Printing System Manager (PSM) for AIX to manage printing on the SP system.

    The SP Print Management System is still supported on nodes running versions of PSSP earlier than PSSP 2.3.
- Which pairs are supported
  - The following PSSP and AIX pairs are supported:

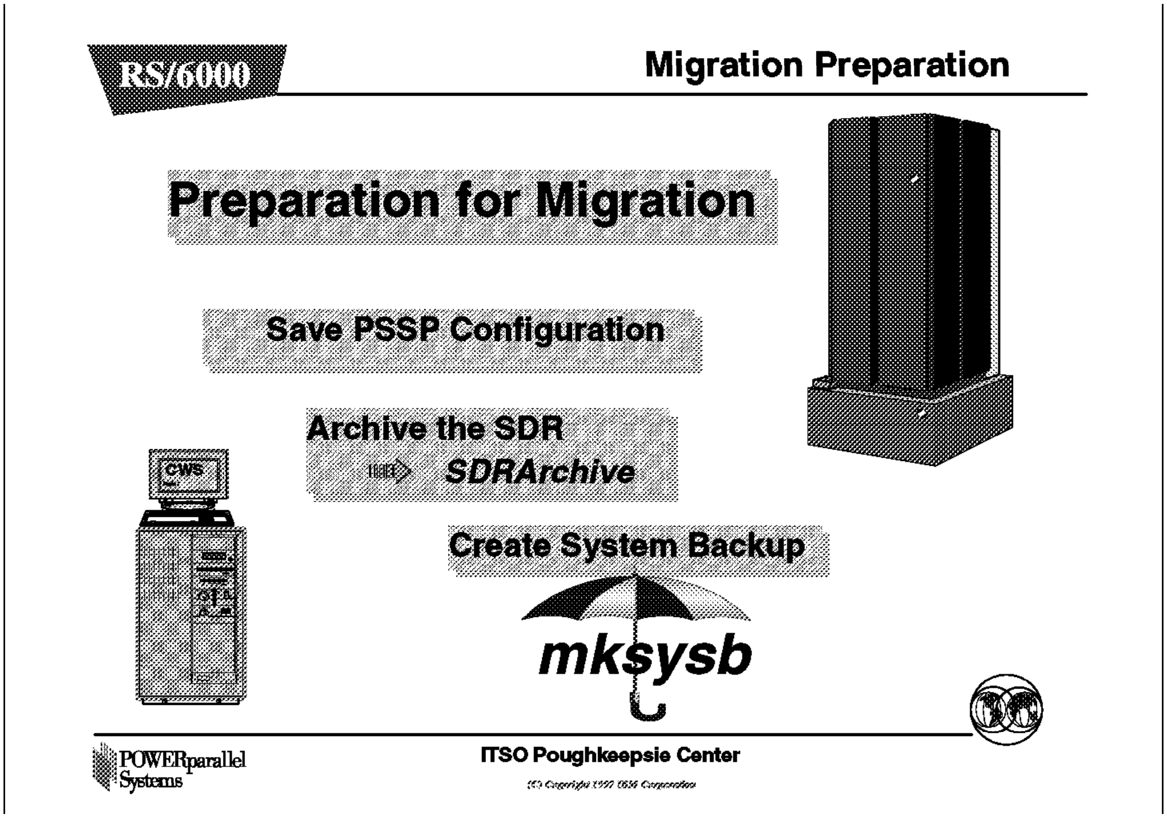
- PSSP1.2 + AIX 3.2.5
- PSSP2.1 + AIX 4.1.3, 4.1.4, 4.1.5
- PSSP2.2 + AIX 4.1.3, 4.1.4, 4.1.5, 4.2.0, 4.2.1
- PSSP2.3 + AIX 4.2.1

When you plan to migrate only some nodes, you must first migrate the Control Workstation to the latest level of PSSP. The Control Workstation must be at the same, or at a higher level of AIX and PSSP as an RS/6000 SP node. If AIX 4.2.1 is installed, you must have PSSP 2.3 or PSSP 2.2+.

**Note:**

- For 604e High Node, you need PSSP 2.3 and AIX 4.2.1 or PSSP 2.2+ and AIX 4.1.5+.
- If you need the AIX 3.2.5 boot/install server, at least two two are recommended; otherwise you cannot reinstall them.

## 3.3.1 Preparation for Migration



RS/6000

Migration Preparation

Preparation for Migration

Save PSSP Configuration

Archive the SDR

SDRArchive

Create System Backup

mksysb

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Migrating your nodes and CWS is a complex task. Preparation is very important. You have to plan the migration by looking at the current configuration and the desired future configuration. Read the “Planning for Migration” chapter in the *Planning Volume 2, Control Workstation and Software Environment Guide,* GA22-7281.

The configuration worksheets found in the *PSSP System Planning Guide* are not necessary for migrating. However, if you are using partitioning or coexistence, it is very helpful to complete the worksheets.

See the Memo for Users for the most up-to-date information on service levels. To retrieve the readme file from the AIX install image, use the command:

```
installp -i -d <dev> all > <filename>
```

Consider creating a production system partition and a test system partition. This enables you to test AIX4.2.1/PSSP 2.3 while running the existing production environment. For more information about system partition, see the *PSSP Planning Guide* and the *PSSP Administration Guide*.

Verify that your backups are valid and up-to-date.

Archive SDR using the command: `/usr/lpp/ssp/bin/SDRArchive`. This script tars the contents of the SDR and puts the tar file in /spdata/sys1/sdr/archives/backup.<datetime>.

Allocate adequate disk space on the Control Workstation for:

- rootvg
- Paging space
- /spdata

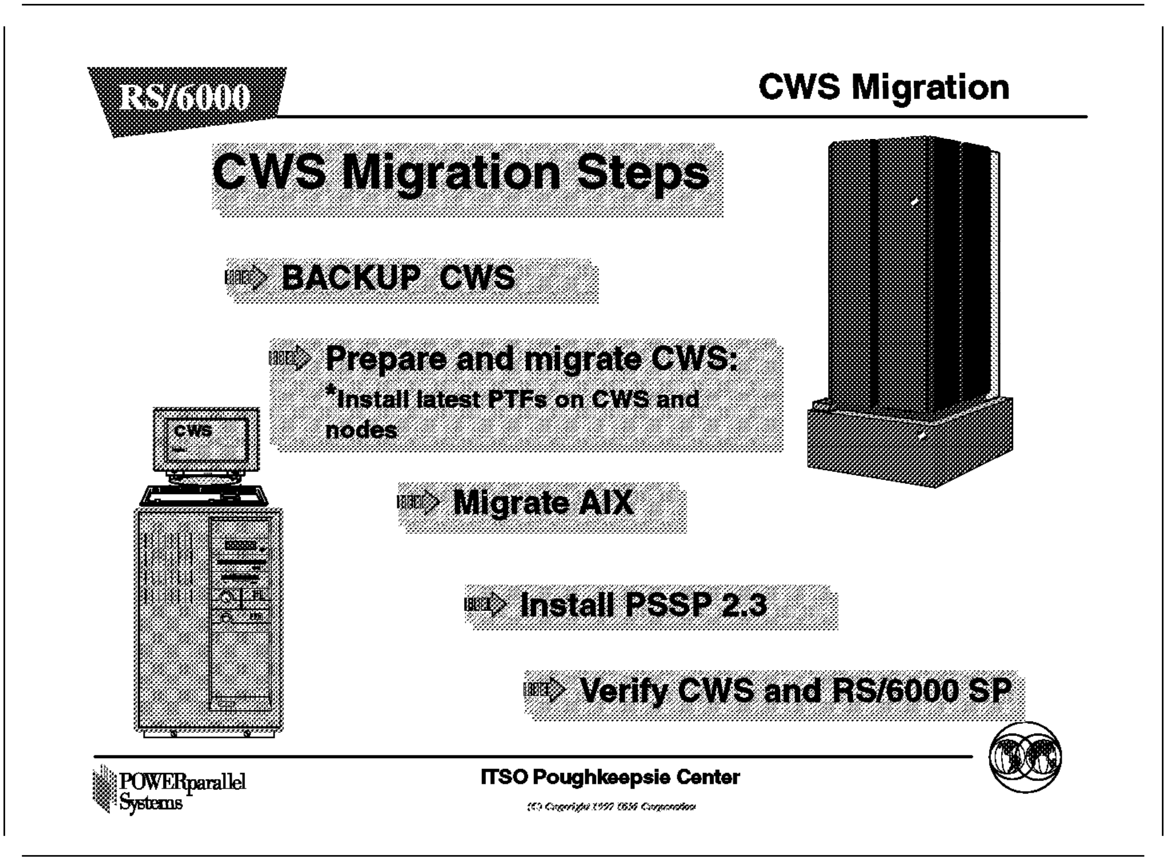
IBM requires a minimum of 4GB of DASD available on the Control Workstation. We recommend allocating 2GB for each AIX and PSSP level in /spdata and 2GB for rootvg.

If you have any PSSP 1.2 nodes, verify that the PSSP 1.2 boot/install nodes have adequate disk space.

Verify that all hostnames and IP addresses are resolvable on the Control Workstation. Do not change them during migration. Migration tasks need root authority. Add the following directories to your .profile, if not already done:

```
/usr/lpp/ssp/bin
/usr/sbin
/usr/lpp/ssp/kerberos/bin
```

## 3.4 Control Workstation (CWS) Migration



RS/6000

CWS Migration

CWS Migration Steps

- BACKUP CWS
- Prepare and migrate CWS:
  *Install latest PTFs on CWS and nodes
- Migrate AIX
- Install PSSP 2.3
- Verify CWS and RS/6000 SP

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

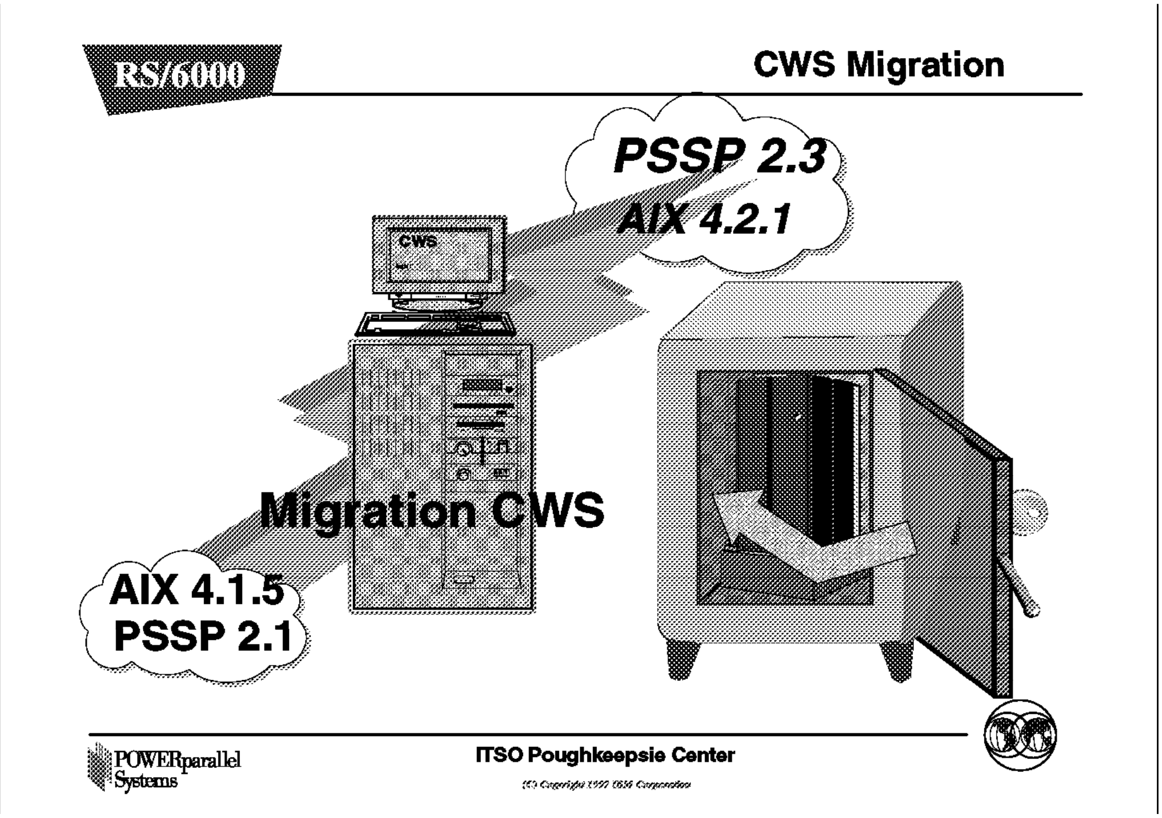
When migrating an RS/6000 SP system, you have to complete five major checkpoints:

1. Back up your Control Workstation.
2. Apply PTFs on the Control Workstation and nodes.
3. Migrate AIX to AIX 4.2.1 on the Control Workstation.
4. Install PSSP 2.3.
5. Verify operations.

You must migrate your Control Workstation to the latest level of AIX and PSSP prior to migrating any node.

The following sections describe how to migrate the Control Workstation.

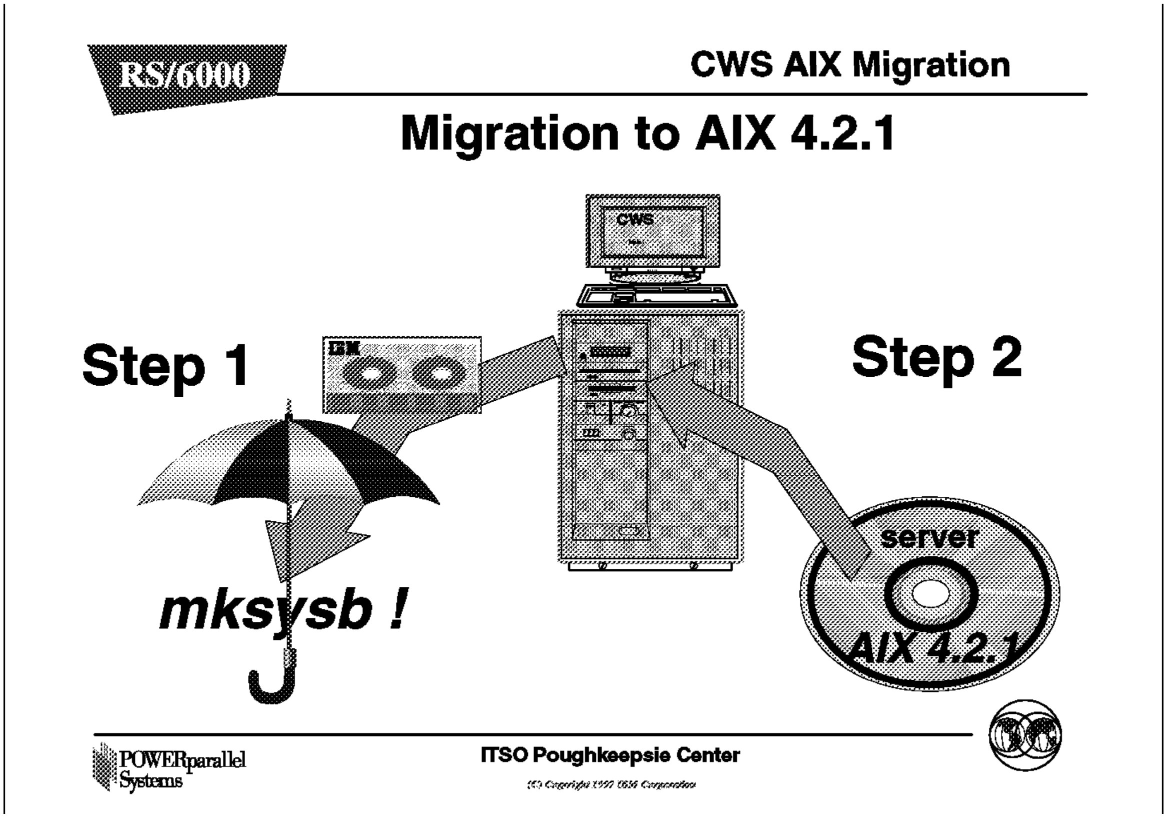
### 3.4.1 CWS AIX Migration and Availability



Availability is very important in a customer environment. Therefore we will minimize disturbing the nodes during the Control Workstation migration to PSSP 2.3 and AIX 4.2.1.

If you are using the Control Workstation as name server, file server, yp server, and so on, you have to move these functions **before** migration to a node or an RS/6000.

## 3.4.2 CWS AIX Migration



RS/6000

CWS AIX Migration

Migration to AIX 4.2.1

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

You should always make a backup of your system. Do the following:

1. Create an mksysb image of the rootvg volume group on the Control Workstation by using the following smit fastpath:
    - Issue `smit mksysb`.
2. Make a backup for every non-rootvg volume groups.
    - Issue `smit savevg`.
        - umount the file systems in that volume group.
        - Issue `varyoffvg`.
        - Issue `exportvg`.
    - Back up file systems on the Control Workstation that have configuration data:
        - Issue `smit backfilesys`.
    - Verify backups; verify if the backup media contains your files.
    - Back up your nodes.
    - Issue `SDRArchive` (if you have PSSP 2.1 or later) as follows:

```
[c201cw]/>SDRArchive
SDRArchive: SDR archive file name is
  /spdata/sys1/sdr/archives/backup/97126.1637
```

3. Boot from AIX 4.2.1 CD or tape, and choose the BOS/Migration Install option.

   **Note:** If you choose BOS overwrite install, you reinstall the Control Workstation. This method does not preserve the file system and the files. You have to reinstall the configuration files.

   - Select option 2: **Change/Show Installation Setting and Install**. Verify that it has been set to `migrate` and that the target disk or disks are correct for installation of rootvg.
   - Select option 1: **System Settings in Installation and Settings menu**.
   - Select option 3: **Migration Install on the Method of Installation menu**.

4. Install the required AIX LPPs and PTFs for PSSP 2.3.

5. Verify AIX migration.

   - Issue `oslevel` to check the AIX level.
   - Issue `oslevel -l 4.2.1.0` to check the files not migrated to AIX 4.2.1.
   - Verify the Control Workstation interfaces. Verify all the network interfaces configured in the Control Workstation

6. Change the maximum number of processes from the default of 40 to 256, with the command:

   ```
   chdev -l sys0 -a maxuproc=256
   ```

7. Verify and, if needed change the network tunables.

   - To list the network options: issue `no -a`. The output follows:

```
output no -a :

                 thewall = 16384
                   sb_max = 163840
                somaxconn = 1024
      clean_partial_conns = 0
        net_malloc_police = 0
                  rto_low = 1
                 rto_high = 64
                rto_limit = 7
               rto_length = 13
              arptab_bsiz = 7
                arptab_nb = 25
               tcp_ndebug = 100
                   ifsize = 8
                 arpqsize = 1
             route_expire = 0
                 strmsgsz = 0
                 strctlsz = 1024
                 nstrpush = 8
                strthresh = 85
                psetimers = 20
              psebufcalls = 20
               strturncnt = 15
             pseintrstack = 12288
                lowthresh = 90
                medthresh = 95
                 psecache = 1
          subnetsarelocal = 1
                   maxttl = 255
                ipfragttl = 60
          ipsendredirects = 1
             ipforwarding = 1
                  udp_ttl = 30
                  tcp_ttl = 60
               arpt_killc = 20
            tcp_sendspace = 65536
            tcp_recvspace = 65536
            udp_sendspace = 32768
            udp_recvspace = 65536
           rfc1122addrchk = 0
           nonlocsrcroute = 1
            tcp_keepintvl = 150
             tcp_keepidle = 14400
                bcastping = 0
                 udpcksum = 1
              tcp_mssdflt = 1448
          icmpaddressmask = 0
             tcp_keepinit = 150
ie5_old_multicast_mapping = 0
                  rfc1323 = 0
         pmtu_default_age = 10
 pmtu_rediscover_interval = 30
        udp_pmtu_discover = 0
        tcp_pmtu_discover = 0
                ipqmaxlen = 100
       directed_broadcast = 1
        ipignoreredirects = 0
           ipsrcroutesend = 1
           ipsrcrouterecv = 0
        ipsrcrouteforward = 1
```

- To modify the network options, issue:

  `no -o <value>=....`

  For example:

  `no -o tcp_sendspace=65536`

- The defaults are as follows:

```
the wall 16384
sb_max   163840
ipforwarding 1
tcp+sendspace  65536
tcp_recvspace  65536
udp_sendspace  32768
udp_recvspace  65536
tcp_mssdflt    1448
```

8. To verify space for /tftpboot, issue the command:

   `df /tftpboot`

   This will show:

   ```
   Filesystem    1024-blocks      Free %Used    Iused %Iused Mounted on
   /dev/lv02          143360     58220   60%       59     1% /tftpboot
   ```

   You need at least 25MB of free space in /tftpboot for each lppsource level supported on the system.

9. Define the spdatavg volume group, if it does not exist:

   - PSSP 2.3 needs the following directory structure:
     - /spdata/sys1/install/images
     - /spdata/sys1/install/ssp
     - /spdata/sys1/install/aix421/lppsource
     - /spdata/sys1/install/pssplpp/PSSP-2.3

       **Note:** If you need the Control Workstation as the NIM server for nodes not at PSSP 2.3, you need to create the appropriate subdirectory, for example: /spdata/sys1/install/pssplpp/PSSP-2.1.

   - Copy the AIX 4.2.1 LPP image into the /spdata/sys1/install/aix421/lppsource directory on the Control Workstation.
   - Copy the PSSP images into the /spdata/sys1/install/pssplpp/PSSP-2.3 directory. You can use the `bffcreate` command or `smit bffcreate`.

     **Note:**

     - For AIX 4.2.0 systems, install the AIX PTFs service for AIX 4.2.1.
     - The latest AIX 4.2.1 service level is needed to support PSSP 2.3. You can install the PTFs directly from the media, or you can load them into the lppsource directory and install them from there.

10. Authenticate the administration user to the Kerberos database with the command:

    `kinit root.admin`

11. Initialize PSSP with the command:

    `install_cw`

    The `install_cw` command:

    - Starts and configures the SDR
    - Starts and configures PSSP daemons

- Establishes network performance tuning parameters for the SP nodes by copying `tuning.default` to `tuning.cust` if `tuning.cust` exists.

12. When AIX migration and the following command

    ```
    install_cw
    ```

    are complete, you must verify if everything is still working. Reboot the Control Workstation to do this.

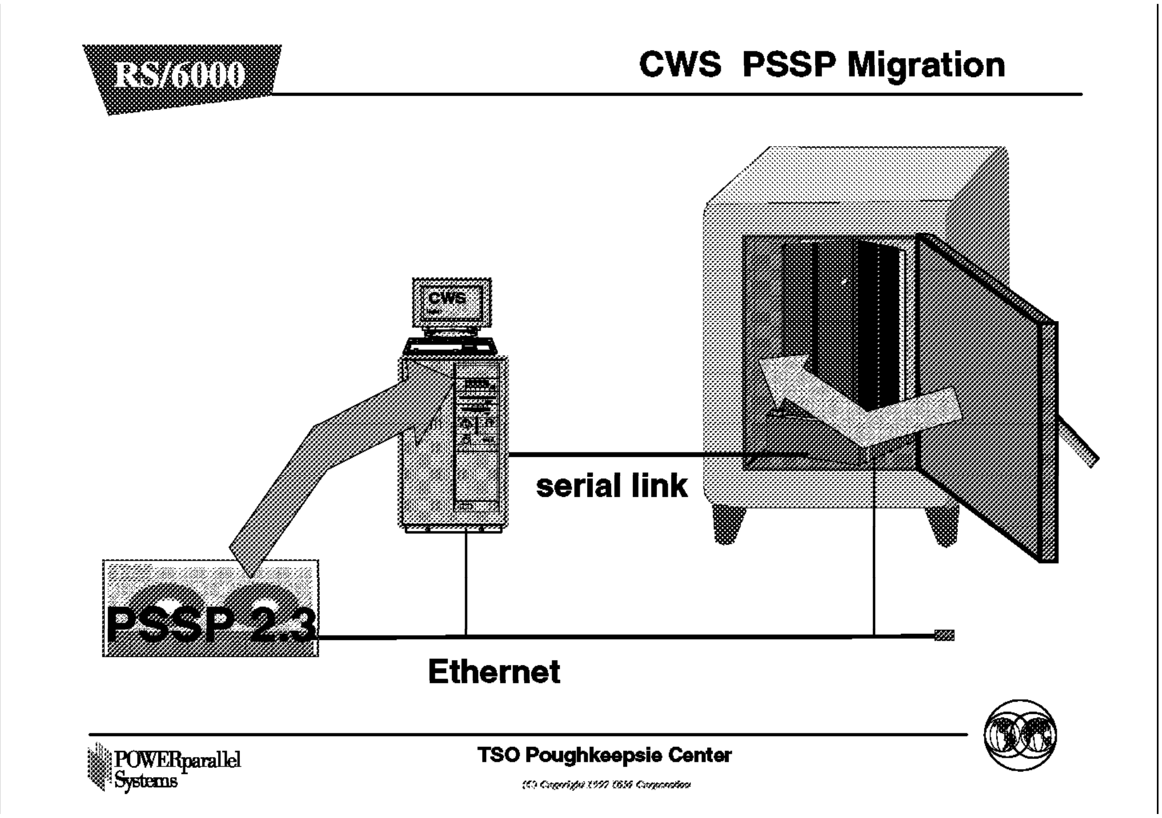
13. Make a new system backup (number two) of the Control Workstation.

**Attention**

If you have problems after migrating the Control Workstation to PSSP 2.3, you can restore this number two system backup. This saves you time, because it is not necessary to return to the first system backup.

If you migrate your HACWS, refer to "Strategy for HACWS Migration" in the *Installation and Migration Guide*.

## 3.4.3 CWS PSSP Migration



RS/6000

CWS PSSP Migration

POWERparallel Systems

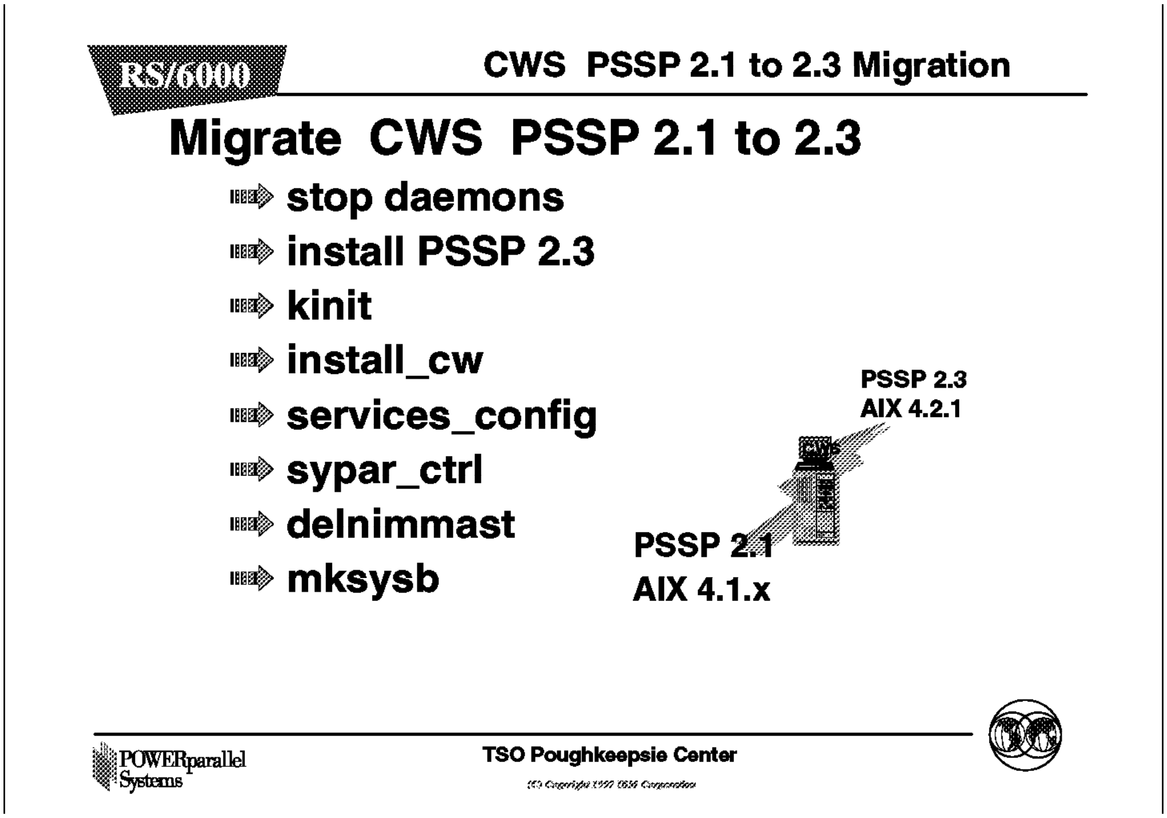
TSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

After migrating the Control Workstation to AIX 4.2.1, you should make a system backup. You can go back to that checkpoint if the following migration does not work.

The following two sections explain in detail how to migrate from PSSP 2.1 and 2.2 to PSSP 2.3.

### 3.4.3.1 CWS Migration from PSSP 2.1 to PSPP 2.3



RS/6000 **CWS PSSP 2.1 to 2.3 Migration**

**Migrate CWS PSSP 2.1 to 2.3**

- stop daemons
- install PSSP 2.3
- kinit
- install_cw
- services_config
- sypar_ctrl
- delnimmast
- mksysb

PSSP 2.3
AIX 4.2.1
CWS
PSSP 2.1
AIX 4.1.x

POWERparallel Systems

TSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Migrating from PSSP 2.1 to PSSP 2.3 involves the following steps:

1. Migrate PSSP, Install the PSSP 2.3 code on the Control Workstation. The PSSP 2.3 file sets have to be installed on top of PSSP 2.1. Copy the PSSP 2.3 images to the /spdata/sys1/install/pssplpp directory, and then you install the new level from there.

   - Check that the latest level of perfagent (AIX PAID) is installed on the Control Workstation prior to installing the pssp.installp package. For AIX 4.2.1, the perfagent.server must be at level 2.2.1.2 or higher.

   - Move existing LPP images. PSSP 2.3 supports more than one SPOT resource on the Control Workstation. If you plan to support PSSP 2.1 nodes at the AIX 4.1.x level, you will need to select a new lppsource to hold the AIX 4.1 file sets. You have to move the existing directory to the new lppsource directory, as follows:

     ```
     mkdir /spdata/sys1/install/aix414
     mv /spdata/sys1/install/lppsource  spdata/sys1/install/aix414/
     ```

     Create a symbolic link from the new directory to the old directory to support PSSP 2.1 as follows:

     ```
     ln -s spdata/sys1/install/aix414/lppsource spdata/sys1/install/lppsource
     ```

   - Stop daemons as follows:

     – `kill -term amd # kill AMD daemon`

- `stopsrc -g hr`
- `stopsrc -g hb`
- `stopsrc -g emon`
- `stopsrc -s sysctld`
- `stopsrc -s splogd`
- `stopsrc -s hardmon`
- `stopsrc -g sdr`

- Install PSSP 2.3, as follows:

```
             Install and Update from LATEST Available Software

  Type or select values in entry fields.
  Press Enter AFTER making all desired changes

* INPUT device / directory for software            /spdata/sys1/install/
* SOFTWARE to install                      [_all_latest]
  PREVIEW only? (install operation will NOT occur)  no
  COMMIT software updates?                          yes
  SAVE replaced files?                              no
  AUTOMATICALLY install requisite software?         yes
  EXTEND file systems if space needed?              yes
  OVERWRITE same or newer versions?                 no
  VERIFY install and check file sizes?              no
  Include corresponding LANGUAGE filesets?          yes
  DETAILED output?                                  no
```

2. Authenticate the administration user to the Kerberos database with the command:

   `kinit root.admin`

3. Initialize PSSP by issuing:

   `install_cw`

4. Verify the Control Workstation with the following scripts:

   - `SDR_test`
   - `SYSMAN_test`
   - `spmon_ctest`
   - `spmon_itest`
   - `jm_install_verify`
   - `jm_verify`
   - `CSS_test`
   - `spverify_test`

5. Set up the site environment LPP source variable by issuing:

   `smitty site_env_dialog`

   Then change the Control Workstation LPP Source Name to `aix421`.

6. Configure PSSP services. The system management environments on the Control Workstation are started with the command:

   `services_config`

7. Start and verify the subsystems, as follows:

- Remove old subsystems (hr, hb, and so on) by issuing:

  ```
  syspar_ctrl -c
  ```

- Add new subsystems via the command:

  ```
  syspar_ctrl -A -G
  ```

- Verify the subsystems via the command:

  ```
  lssrc -a|pg
  ```

  The output will look like this:

  ```
  Subsystem         Group           PID     Status
   endmail          mail            4130    active
   portmap          portmap         4654    active
   inetd            tcpip           4404    active
   snmpd            tcpip           5178    active
   nimesis          nim             3134    active
   biod             nfs             2406    active
   nfsd             nfs             7546    active
   rpc.mountd       nfs             7306    active
   rpc.statd        nfs             6554    active
   rpc.lockd        nfs             7850    active
   sdr.c201cw       sdr             8636    active
   supfilesrv                       9190    active
   qdaemon          spooler         9280    active
   writesrv         spooler         9542    active
   hardmon                          13004   active
   infod            infod           11986   active
   kerberos                         13552   active
   kadmind                          12282   active
   sysctld                          12290   active
   sp_configd                       17934   active
   spmgr                            16916   active
   hb.c201cw        hb              13608   active
   pman.c201cw      pman            12588   active
   pmanrm.c201cw    pman            14394   active
   splogd                           17788   active
   hags.c201cw      hags            19844   active
   hagsglsm.c201cw  hags            18326   active
   hats.c201cw      hats            14326   active
   hr.c201cw        hr              13304   active
   syslogd          ras             6116    active
   ....
  ```

- Start Quiesced Applications:

  Any of the applications that you quiesced prior to migrating your control workstation should be started at this time if they have not been started automatically. For example if your system has a switch issue the following command to restart your switch:

  ```
  Estart
  ```

- Configure Control Workstation as Boot/Install Server

  Verify that the SDR node attribute value for *code_version, lppsource_name*, and *next_install_image* are appropriately set. Use the following command:

  ```
  splstdata -b -G
  ```

  If necessary, you can use the following command to change the node′s attribute values:

  ```
  spbootins -s no -p <code_version> -i <install_image_name>
  -v <lppsource_name> -l <node_list>
  ```

Also, you can use the following command to change the SP attribute values:

```
spsitenv install_image=<install_image_name>
```

The *setup_server* command must be run to properly set up NIM on the control workstation. This can be done by issuing the following command:

```
setup_server 2>& | tee /tmp/setup_server.out
```

This may take some time to complete since it will be creating the NIM master.

- Verify the Control Workstation by using the following scripts:
  - SDR_test
  - SYSMAN_test
  - spmon_ctest
  - spmon_itest
  - jm_install_verify
  - jm_verify
  - CSS_test
  - spverify_test

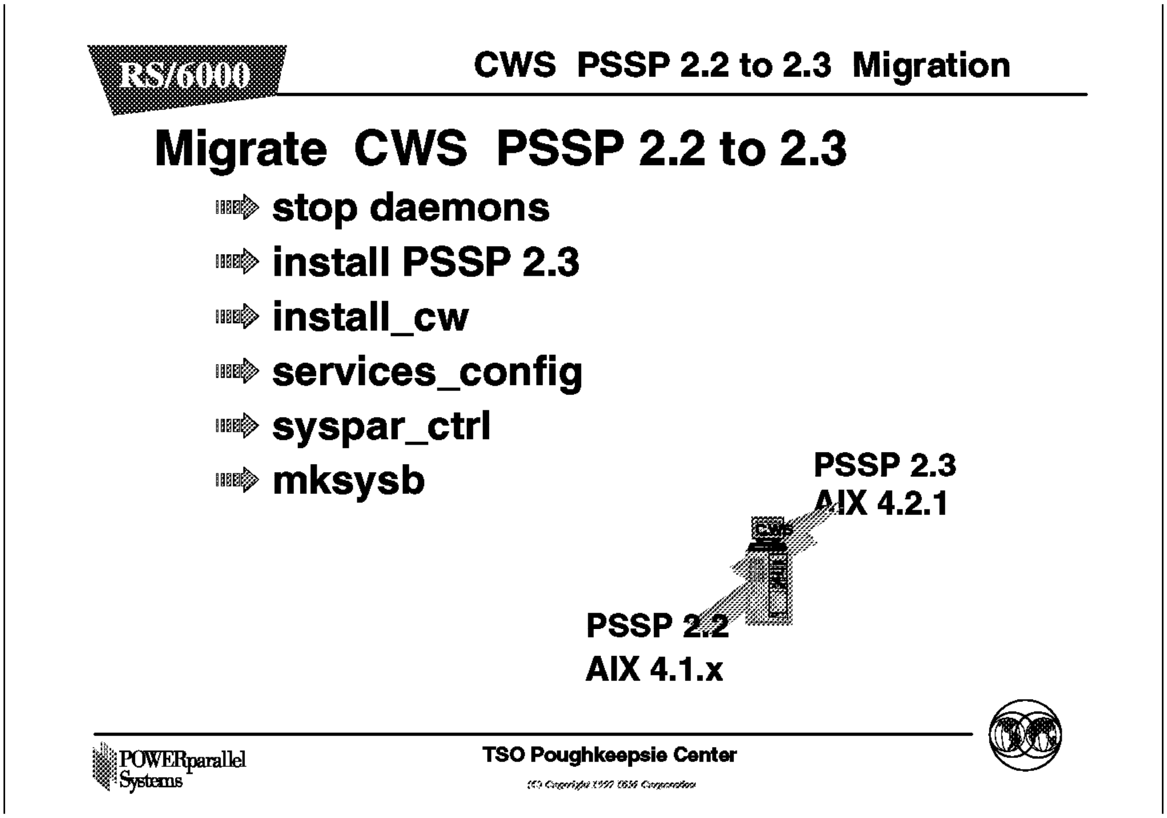
8. Perform NIM master deconfiguration. PSSP 2.3 supports multiple non /usr SPOT resources on the Control Workstation.
   - Deconfigure the NIM by issuing:

     ```
     delnimmast -l 0
     ```

9. Archive the SDR via: SDRArchive
10. Make a system backup of the Control Workstation.

Migration of the Control Workstation is now complete. If you have problems, issue SYSMAN_test and SDR_test again, check errpt, and so on. Restore the system backup if you cannot resolve the problems.

### 3.4.3.2 CWS Migration from PSSP 2.2 to PSPP 2.3



RS/6000

CWS PSSP 2.2 to 2.3 Migration

**Migrate CWS PSSP 2.2 to 2.3**

- stop daemons
- install PSSP 2.3
- install_cw
- services_config
- syspar_ctrl
- mksysb

PSSP 2.3
AIX 4.2.1
CWS
PSSP 2.2
AIX 4.1.x

POWERparallel Systems

TSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Migrating from PSSP 2.2 to PSSP 2.3 involves the following steps:

1. Migrate PSSP. You need to install the PSSP 2.3 code on the Control Workstation The 2.3 file sets have to be installed on top of PSSP 2.2. Copy the PSSP 2.3 images to /spdata/sys1/install/pssplpp directory, and then you can install the new level from there.

    - Check that the latest level of perfagent (AIX PAID) is installed on the Control Workstation prior to installing the pssp.installp package. For AIX 4.2.1, perfagent.server must be at level 2.2.1.2 or higher.

    - Stop daemons, as follows:

        a. `kill -term amd # kill Amd daemon`

        b. `stopsrc -g hr`

        c. `stopsrc -g hb`

        d. `stopsrc -g emon`

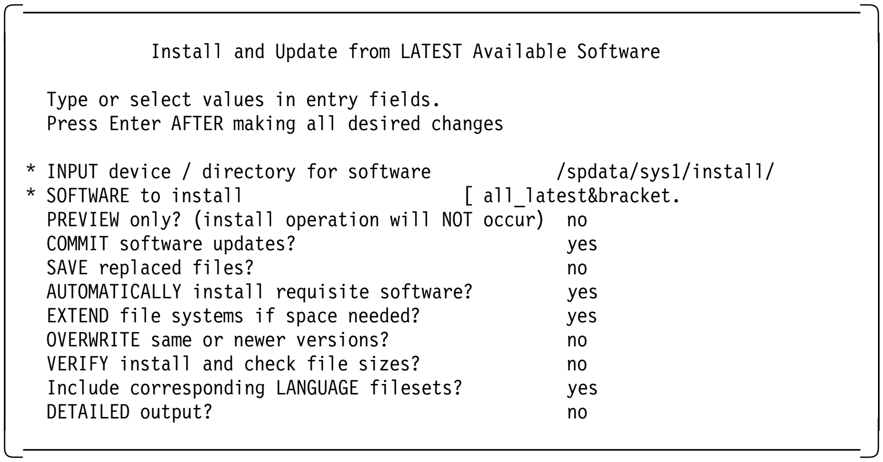
        e. `stopsrc -s sysctld`

        f. `stopsrc -s splogd`

        g. `stopsrc -s hardmon`

        h. `stopsrc -g sdr`

    - Install PSSP 2.3, as follows:



```
           Install and Update from LATEST Available Software

  Type or select values in entry fields.
  Press Enter AFTER making all desired changes

* INPUT device / directory for software            /spdata/sys1/install/
* SOFTWARE to install                          [ all_latest&bracket.
  PREVIEW only? (install operation will NOT occur)  no
  COMMIT software updates?                          yes
  SAVE replaced files?                              no
  AUTOMATICALLY install requisite software?         yes
  EXTEND file systems if space needed?              yes
  OVERWRITE same or newer versions?                 no
  VERIFY install and check file sizes?              no
  Include corresponding LANGUAGE filesets?          yes
  DETAILED output?                                  no
```

2. Authenticate the administration user to the Kerberos database with the following command:

   ```
   kinit root.admin
   ```

3. Initialize PSSP, as follows:

   ```
   install_cw
   ```

4. Verify the Control Workstation by using the following scripts:

   - SDR_test
   - SYSMAN_test
   - spmon_ctest
   - spmon_itest
   - jm_install_verify
   - jm_verify
   - CSS_test
   - spverify_test

5. Set up the site environment LPP source variable with the following command:

   ```
   smitty site_env_dialog
   ```

   Change the Control Workstation LPP Source Name to aix421

6. Configure PSSP services. The system management environments in the Control Workstation are started by issuing:

   ```
   services_config
   ```

7. Start and verify the subsystems, as follows:

   - Remove old subsystems by issuing:

     ```
     syspar_ctrl -c
     ```

   - Add new subsystems by issuing:

     ```
     syspar_ctrl -A -G
     ```

   - Verify the subsystems by issuing:

     ```
     lssrc -a|pg
     ```

```
Subsystem         Group            PID     Status
 endmail          mail             4130    active
 portmap          portmap          4654    active
 inetd            tcpip            4404    active
 snmpd            tcpip            5178    active
 nimesis          nim              3134    active
 biod             nfs              2406    active
 nfsd             nfs              7546    active
 rpc.mountd       nfs              7306    active
 rpc.statd        nfs              6554    active
 rpc.lockd        nfs              7850    active
 sdr.c201cw       sdr              8636    active
 supfilesrv                        9190    active
 qdaemon          spooler          9280    active
 writesrv         spooler          9542    active
 hardmon                           13004   active
 infod            infod            11986   active
 kerberos                          13552   active
 kadmind                           12282   active
 sysctld                           12290   active
 spmgr                             16916   active
 hb.c201cw        hb               13608   active
 pman.c201cw      pman             12588   active
 pmanrm.c201cw    pman             14394   active
 splogd                            17788   active
 hags.c201cw      hags             19844   active
 hagsglsm.c201cw  hags             18326   active
 hats.c201cw      hats             14326   active
 hr.c201cw        hr               13304   active
 syslogd          ras              6116    active
 ....
```

- Start Quiesced Applications:

  Any of the applications that you quiesced prior to migrating your control workstation should be started at this time if they have not been started automatically. For example if your system has a switch issue the following command to restart your switch:

  ```
  Estart
  ```

- Configure Control Workstation as Boot/Install Server

  Verify that the SDR node attribute value for *code_version, lppsource_name*, and *next_install_image* are appropriately set. Use the following command:

  ```
  splstdata -b -G
  ```

  If necessary, you can use the following command to change the node′s attribute values:

  ```
  spbootins -s no -p <code_version> -i <install_image_name>
  -v <lppsource_name> -l <node_list>
  ```

  Also, you can use the following command to change the SP attribute values:

  ```
  spsitenv install_image=<install_image_name>
  ```

  The *setup_server* command must be run to properly set up NIM on the control workstation. This can be done by issuing the following command:

  ```
  setup_server 2>& | tee /tmp/setup_server.out
  ```

  This may take some time to complete since it will be creating the NIM master.

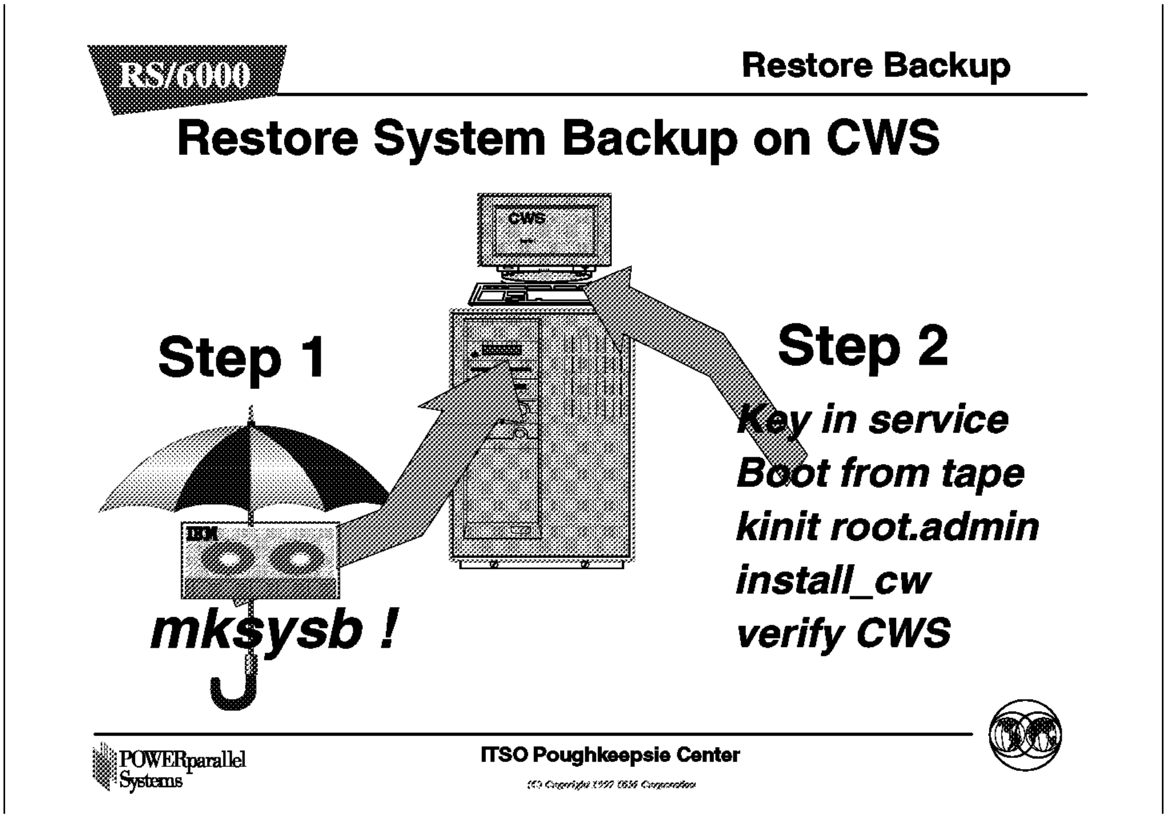
- Verify the Control Workstation with the following scripts:

- SDR_test
- SYSMAN_test
- spmon_ctest
- spmon_itest
- jm_install_verify
- jm_verify
- CSS_test
- spverify_test

8. Archive SDR SDRArchive.
9. Make a system backup of the Control Workstation.

Migration of the Control Workstationis now complete. If you have problems, issue SYSMAN_test SDR_test again, check errpt, and so on. Call you next level of support. Restore the system backup if you cannot resolve the problems. See section 3.4.3.3, “Restore System Backup” on page 70 to restore mksysb.

### 3.4.3.3 Restore System Backup



RS/6000

Restore Backup

Restore System Backup on CWS

Step 1

mksysb !

Step 2

Key in service

Boot from tape

kinit root.admin

install_cw

verify CWS

POWERparallel Systems

ITSO Poughkeepsie Center

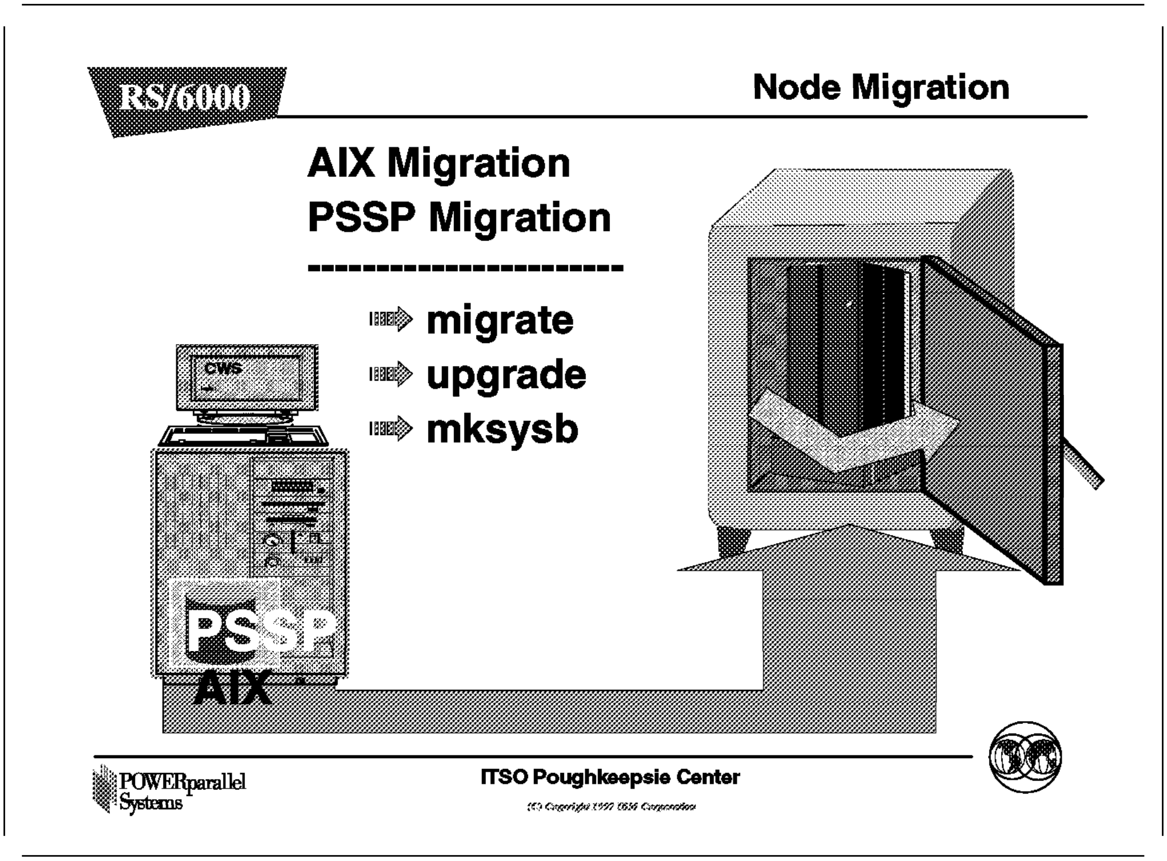
(C) Copyright 1997 IBM Corporation

This section describes how to restore a system backup on the Control Workstation.

1. Insert the backup tape into the tape drive.
2. Change the key to the service position. If your Control Workstation is a PCI-based RS/6000, press the F2 key at boot time. This gives a menu. From this menu, select the tape as **boot device** and press Enter. If your Control Workstation is a microchannel-based RS/6000, follow the instructions on the screen.
3. Select the disk(s) to install the rootvg volume group.
4. Log in as root user after successfully completion of the restore.
5. Authenticate as the Kerberos administrator with the `kinit root.admin` command.
6. Issue the `install_cw` command.
7. List the SDR archives.
8. Issue the `sprestore_config <archive_name>` command, where the <archive_name> is the name of your last SDR archive.
9. Check to see if your SDR is correct by using the `spmon -d` command and the `splstdata` command.

10. Perform step Control Workstation migration verification of section Control Workstation migration.

## 3.5 Node Migration



RS/6000

Node Migration

AIX Migration
PSSP Migration

- migrate
- upgrade
- mksysb

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

You can migrate the node to AIX 4.2.1 and PSSP 2.3 in three ways:

1. Perform a migration install. This method preserves:
   - File systems
   - Root volume group
   - System configuration files
   - Logical volumes

   **Note:** /tmp is not preserved with migration install.

2. Perform a mksysb install. With a mksysb install, all instances of the current rootvg are erased. This method installs AIX 4.2.1 and PSSP 2.3 using a previously created mksysb image. This method requires the setup of AIX NIM on the Control Workstation.

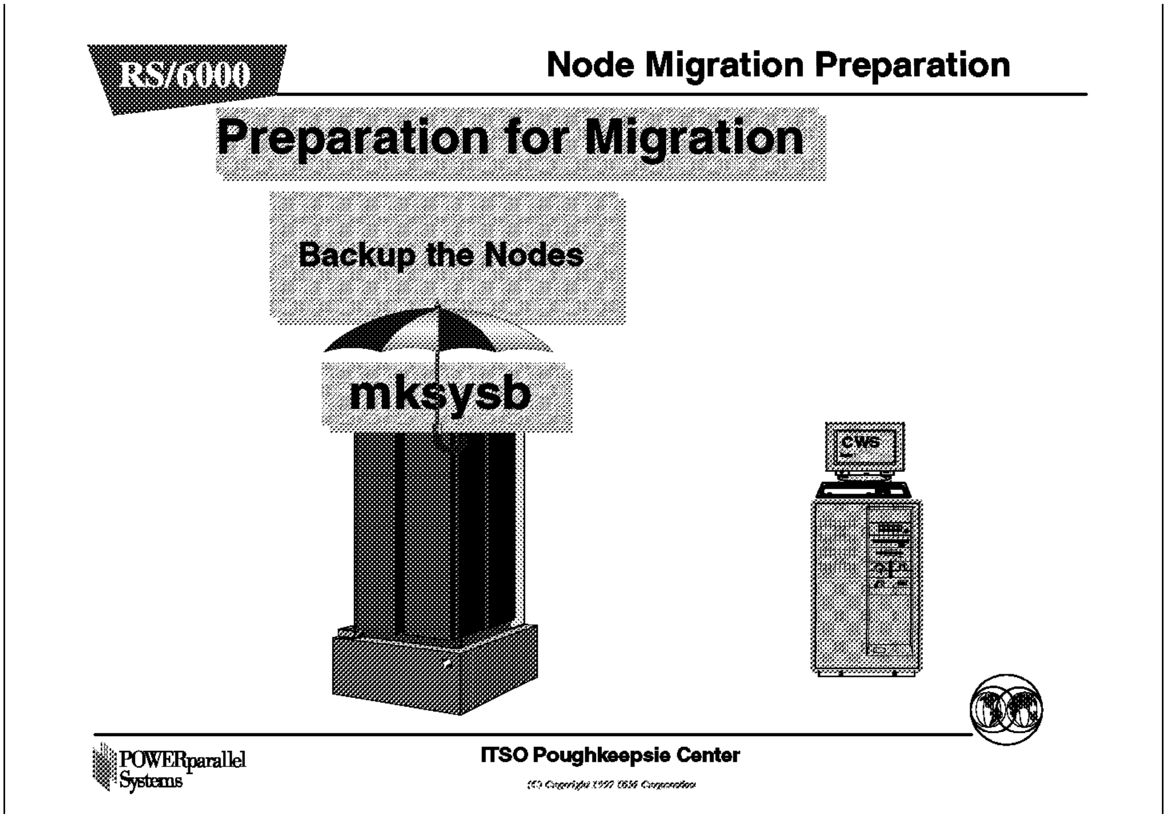
3. Perform an upgrade. When only AIX modification levels are changing, for example in AIX 4.2.0 to AIX 4.2.1, you can use this method. Upgrade preserves the current rootvg and installs AIX PTFs updates using the `installp` command.

The following section explains how to migrate nodes to AIX 4.2.1, and PSSP to PSSP 2.3. Two scenarios are explained:

- PSSP 2.1 to PSSP 2.3, and AIX 4.1.5 to 4.2.1

- PSSP 2.2 to PSSP 2.3, and AIX 4.1.5 to 4.2.1

## 3.5.1 Preparation for Node Migration



RS/6000

Node Migration Preparation

Preparation for Migration

Backup the Nodes

mksysb

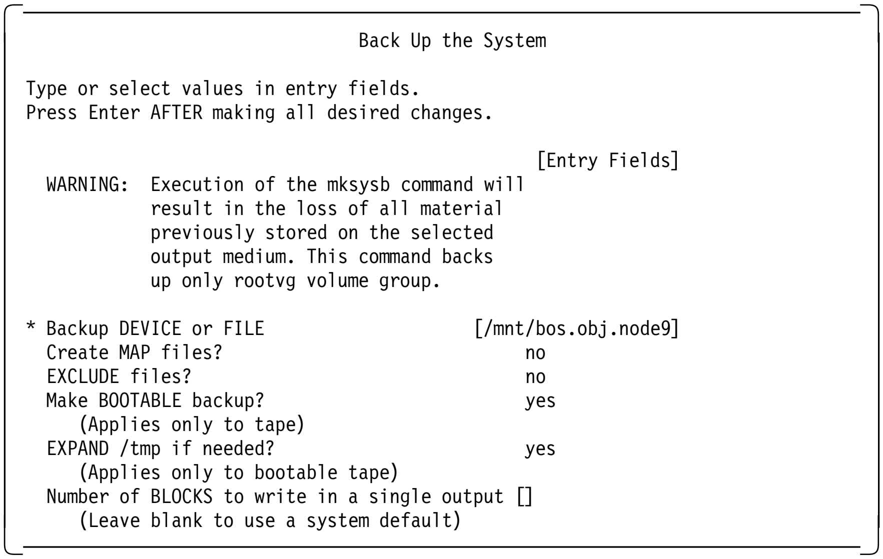
POWERparallel Systems

ITSO Poughkeepsie Center

Before migrating your RS/6000 SP nodes, make a backup of the node.

- Use the `smit mksysb` command to perform the system backup. Set the backup name so that the backup is performed over the network onto the Control Workstation or to a node that is used to do backups.

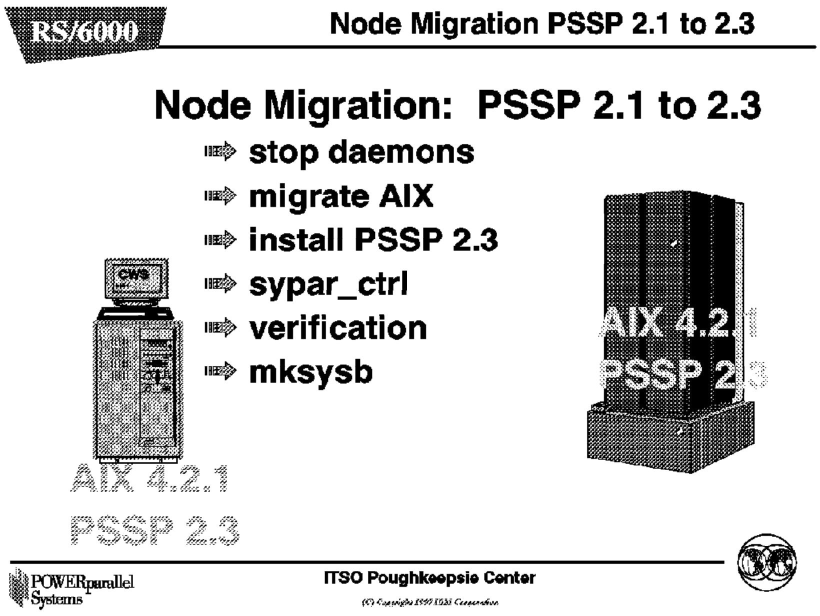
  In the following figure, the /mnt directory is mounted from the Control Workstation using NFS.



```
                                 Back Up the System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                 [Entry Fields]
  WARNING:  Execution of the mksysb command will
            result in the loss of all material
            previously stored on the selected
            output medium. This command backs
            up only rootvg volume group.

* Backup DEVICE or FILE                    [/mnt/bos.obj.node9]
  Create MAP files?                             no
  EXCLUDE files?                                no
  Make BOOTABLE backup?                         yes
     (Applies only to tape)
  EXPAND /tmp if needed?                        yes
     (Applies only to bootable tape)
  Number of BLOCKS to write in a single output []
     (Leave blank to use a system default)
```

- Verify your backups. Verify that the file exists on the medium, whether the backup is made to a file or to a tape.

## 3.5.2 Migrate Node PSSP 2.1 to 2.3



RS/6000

Node Migration PSSP 2.1 to 2.3

**Node Migration: PSSP 2.1 to 2.3**

- stop daemons
- migrate AIX
- install PSSP 2.3
- sypar_ctrl
- verification
- mksysb

CWS
AIX 4.2.1
PSSP 2.3
AIX 4.2.1
PSSP 2.3

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This section explains what steps are needed to migrate an AIX 4.1.5, PSSP 2.1 node to AIX 4.2.1 and PSSP 2.3.

1. Node configuration data: You have to set the appropriate SDR attributes for the node you are migrating. The attributes to be set are:

   - lppsource_name
   - code_version
   - bootp_response

   To set these attributes, use the following command:

   ```
   spbootins -s no -p <code_version> -v <lppsource_name>
   -l <node_list> -r migrate
   ```

   For example, to migrate node 29:

   ```
   [ceedgate]/tmp/>
   spbootins -s no -p PSSP-2.3 -v aix421 -l 29 -r migrate
   ```

2. Verify settings. The settings in the SDR must have the appropriate attributes. Verify these by issuing the following command:

   ```
   splstdata -b
   ```

```
[ceedgate]/tmp/>splstdata -b
List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response     install_disk
last_install_image   last_install_time  next_install_image lppsource_name
                                                                pssp_ver
-----------------------------------------------------------------------


29 ceed1n10.ppd.pok   02608C3D4B7F  17       migrate           hdisk0
           initial            initial            default         aix421
                                                                 PSSP-2.3
```

- response should be set to migrate
- lppsource_name should be set to aix421
- pssp_ver should be set to PSSP-2.3

3. To set up NIM properly on the Control Workstation, the setup_server command must be run. Issue the following command:

   ```
   setup_server 2 > &1 |tee /tmp/setup_server.out
   ```

   The output is saved in the log file setup_server.out. The setup_server command may take a while if this is the first time it is run on the Control Workstation since the Control Workstation was migrated to PSSP 2.3.

4. The nodes that will run PSSP 2.3 are now set in SDR. To make these changes active on the Control Workstation and the nodes, the subsystems have to be refreshed. To do this, issue the following command: syspar_ctrl -r -G,as follows:

   ```
   [ceedgate]/tmp/>syspar_ctrl -r -G

   0513-095 The request for subsystem refresh was completed successfully
   Machines List is already at the latest incarnation
   Refresh not requested
   ```

5. Switch fabric. (Systems without a switch may skip this step.) To isolate nodes from the switch, issue: Efence. This command disconnects the nodes it specifies from the switch fabric.

   Before issuing the Efence command, you have to verify if the node(s) you are migrating are the Primary or Primary Backup Node. Issue the Eprimary command to check, as follows:

   ```
   [c201w]/>Eprimary
   1  - primary
   1  - oncoming primary
   13  - primary backup
   13  - oncoming primary backup
   ```

   If the Primary node or the Primary Backup Node is one of the nodes you are migrating, you have to assign other nodes as Primary or Primary Backup Node by issuing the Eprimary command, as follows:

   ```
   Eprimary -init node_identifier -backup bnode_identifier
   ```

The -init option initializes or reinitializes the current system partition object. The node_identifier specifies the Primary node, and the bnode_identifier specifies the Primary Backup Node.

```
Efence -autojoin 5 9 13
```

In the preceding example, nodes 5, 9 and 13 are fenced, and will join the switch fabric after migration. For more information, see *PSSP Command and Technical Reference Guide*, GC-23-3900.

6. Shut down the node.

7. Network boot each node that you are migrating by issuing the nodecond command, for example:

   ```
   nodecond 1 5 -G &
   ```

   In this example, node 5 of frame 1 will network/boot. After the migration, the bootp_response has been set to disk. Verify this by issuing the command:

   ```
   splstdata -b
   ```

   **Note:** If you use boot.install servers in your system, you need to migrate these before migrating their clients.

   The nodes will be installed when the LEDs are blank and host_responds is active.

8. Verification. See section 3.6, "Node Migration Verification" on page 86 for node verification.

**Information**

syspar_ctrl -r deals with hb (old heartbeat) and hats (topology services).

You need to run syspar_ctrl -r on the CWS when a node's code_version changes. Basically, if a node migrates from PSSP 2.1 to PSSP 2.2, you use spbootins to change the nodes code_version to PSSP 2.3. You then run syspar_ctrl -r on the CWS.

This tells the hats daemon on the CWS and on all the nodes in the current system partition that a new node should be added to the hats group. This also tells the old hb daemon that this node can be removed from its group.

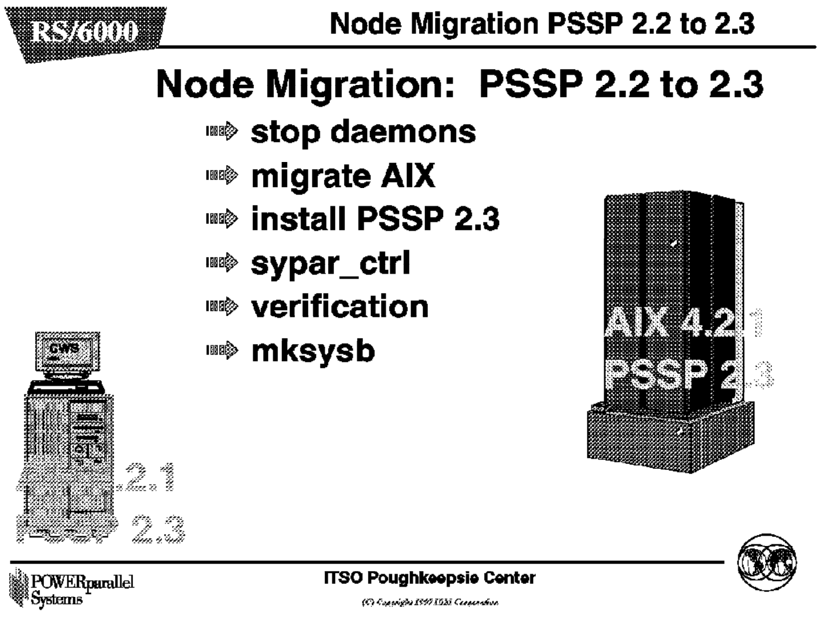
Not until the node is actually running the new level of PSSP, however, should you expect its host_responds to become active (yes). When the new PSSP 2.3 level of hats is started on the node, it tries to join the hats group. It is only allowed to join if it is at PSSP 2.2 or 2.3.

In the refresh that we did just before migrating the nodes, PSSP tells its peer nodes that they should expect to add this node to their group as soon as the node asks to be allowed in (provided it is running a supported level of PSSP).

Always run syspar_ctrl -r on the CWS.

If you have a PSSP 2.2 or 2.3 node, regardless of the host_responds value, a syspar_ctrl -r should refresh it. If all goes well on the node, any host_responds that were set to no should become yes.

### 3.5.3 Migrate Node PSSP 2.2 to 2.3



RS/6000

Node Migration PSSP 2.2 to 2.3

**Node Migration: PSSP 2.2 to 2.3**

- stop daemons
- migrate AIX
- install PSSP 2.3
- sypar_ctrl
- verification
- mksysb

POWERparallel Systems

ITSO Poughkeepsie Center

This section explains what steps are needed to migrate an AIX 4.1.5, PSSP 2.2 node to AIX 4.2.1 and PSSP 2.3.

1. Node configuration data: We have to set the appropriate SDR attributes for the node we are migrating. The attributes to be set are :

   - lppsource_name
   - code_version
   - bootp_response

   To set these attributes, use the following command:

   ```
   spbootins -s no -p <code_version> -v <lppsource_name>
   -l <node_list> -r migrate
   ```

   For example, to migrate node 2, issue the following command:

   ```
   [ceedgate]/tmp/>
   spbootins -s no -p PSSP-2.3 -v aix421 -l 29 -r migrate
   ```

2. Verify settings. The settings in the SDR must have the appropriate attributes. Verify these by issuing the following command:

   ```
   splstdata -b
   ```

```
[ceedgate]/tmp/>splstdata -b
List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response     install_disk
last_install_image   last_install_time  next_install_image lppsource_name
                                                                 pssp_ver
-----------------------------------------------------------------------


29 ceed1n10.ppd.pok   02608C2D4A7F   17        migrate            hdisk0
           initial             initial             default         aix421
                                                                 PSSP-2.3
```

- response should be set to `migrate`
- lppsource_name should be set to `AIX421`
- pssp_ver should be set to `PSSP-2.3`

3. Setup_server. To set up NIM properly on the Control Workstation, the `setup_server` command must be run. Issue the following command:

   ```
   setup_server 2 > &1 |tee /tmp/setup_server.out
   ```

   The output will be saved in the log file setup_server.out.

4. Issue the `syspar_ctrl` command. The nodes that will run PSSP 2.3 are now set in the SDR. To make these changes active on the Control Workstation and the nodes, the subsystems have to be refreshed. To do this, issue command `syspar_ctrl -r -G` as follows:

```
[ceedgate]/tmp/>syspar_ctrl -r -G

0513-095 The request for subsystem refresh was completed successfully
Machines List is already at the latest incarnation
Refresh not requested
```

5. Switch fabric. (Systems without a switch may skip this step.) To isolate nodes from the switch, issue: `Efence`.

   Before issuing the `Efence` command, you have to verify if the node(s) you are migrating are the Primary or Primary Backup Node. Issue the `Eprimary` command to check, and eventually change, the Primary and the Primary Backup node. For more information, see *PSSP Command and Technical Reference Guide* GC-23-3900.

6. Shut down the node.

7. Network boot each node that you are migrating by issuing the following command:

   ```
   nodecond 2 9 -G &
   ```

   In the preceding example, node 9 of frame 2 will network/boot. After the migration, the bootp_response is set to disk. Verify this by issuing the following command:

   ```
   splstdata -b
   ```

   **Note:** If you use boot.install servers in your system, you need to migrate these before migrating their clients.

   The nodes will be installed when the LEDs are blank and host_responds is active.

8. Verification. See section 3.6, “Node Migration Verification” on page 86 for node verification.

### 3.5.4 Migration to PSSP 2.3 Using mksysb

This section explains what steps are needed to migrate an AIX 4.1.5, PSSP 2.2 node to AIX 4.2.1 and PSSP 2.3 via a mksysb install.

1. Node configuration data: You have to set the appropriate SDR attributes for the node you are migrating. The attributes to be set are :

   - lppsource_name
   - code_version
   - bootp_response
   - next_install_image

   To set these attributes, use the following command:

   ```
   spbootins -s no -p <code_version> -v <lppsource_name>
   -i <install_image> -l <node_list> -r install
   ```

   For example, to migrate node 2 when lppsource is in the /spdata/sys1/install/AIX421/lppsource directory, issue:

   ```
   [ceedpart]/tmp>

   spbootins -s no -p PSSP-2.3 -v AIX421 -i bos.obj.ssp.421 -r install -l 2
   ```

2. Verify settings. The settings in the SDR must have the appropriate attributes. Verify these by issuing the following command:

   `splstdata -b`

   ```
   [ceedgate]/tmp/>splstdata -b
   List Node Boot/Install Information

   node#         hostname  hdw_enet_addr srvr     response     install_disk
   last_install_image   last_install_time  next_install_image lppsource_name
                                                                    pssp_ver
   -------------------------------------------------------------------------


   29 ceed1n10.ppd.pok   02608C2DA7EF   17        install            hdisk0
              initial             initial             default        aix421
                                                                   PSSP-2.3
   ```

   - response should be set to `install`
   - lppsource_name should be set to `AIX421`
   - pssp_ver should be set to `PSSP-2.3`
   - next_install_image should be set to `bos.obj.ssp.421`

3. To set up NIM properly on the Control Workstation, the `setup_server` command must be run. Issue the following command:

   `setup_server 2 > &1 |tee /tmp/setup_server.log`

   The output will be saved in the log file `setup_server.log`.

4. Issue the `syspar_ctrl` command. The nodes that will run PSSP 2.3 are now set in the SDR. To make these changes active on the Control Workstation

and the nodes, the subsystems have to be refreshed. To do this, issue the following command on the Control Workstation: `syspar_ctrl -r -G`

5. Switch fabric. (Systems without a switch may skip this step.) To isolate nodes from the switch, use the `Efence` command. Before issuing the command, you have to verify if the node you are migrating is the Primary or Primary Backup Node. Issue the `Eprimary` command to check, and eventually change, the Primary and the Primary Backup node. For more information, see *PSSP Command and Technical Reference Guide*, GC-23-3900.

6. Shut down the node.

7. Network boot each node that you are migrating by issuing the `nodecond` command, for example as follows:

   ```
   nodecond 1 5 -G &
   ```

   In this example, node 5 of frame 1 will network/boot. After the migration, the bootp_response is set to disk. Verify this by issuing the following command:

   ```
   splstdata -b
   ```

   **Note:** If you use boot.install servers in your system, you need to migrate these before migrating their clients.

   The nodes will be installed when the LEDs are blank and host_responds is active.

8. Verification. See section 3.6, “Node Migration Verification” on page 86 for how to perform node verification.

## 3.5.5 Upgrade Node to PSSP 2.3

This section explains what steps are needed to update a node to AIX 4.2.1 and PSSP 2.3. This method applies AIX 4.2.1 PTFs and preserves the rootvg configuration. It upgrades the AIX file sets to AIX 4.2.1. After the upgrade, the script pssp_script installs and updates PSSP 2.3 LPPs on top of current PSSP LPPs. Perform this upgrade as follows:

1. Apply AIX 4.2.1 on the node. You have to first mount the lppsource directory on the node or nodes. Use dsh, as follows:

```
dsh -w c201n01 /usr/sbin/mount c201s:
/spdata/sys1/install/aix421/lppsource /mnt
```

   - c201n01 represents node 1
   - c201s represents the Control Workstation
   - aix421 represents lppsource_name

   Update all LPPs on the node by issuing the following SMIT fastpath on the node:

   `smit update_all`

2. Verify that AIX migration was successful by issuing the following command:

```
[c204cw]/>oslevel -l 4.2.1.0

Fileset                         Actuel level      Maintenance level
-----------------------------------------------------------------
devices.msg.En_US.base.com       4.1.1.0            4.2.1.0
devices.msg.En_US.diag.rte       4.1.1.0            4.2.1.0
devices.msg.En_US.rspc.base.com  4.1.1.0            4.2.1.0
devices.msg.En_US.sys.mca.rte    4.1.1.0            4.2.1.0
```

   The above file sets are not migrated to AIX 4.2.1. You need to order and install the appropriate PTFs to migrate these file sets to AIX 4.2.1.

3. Node configuration data: You have to set the appropriate SDR attributes for the node you are migrating. The attributes to be set are:

   - lppsource_name
   - code_version
   - bootp_response

   To set these attributes, use the following command:

```
spbootins -s no -p <code_version> -v <lppsource_name>
-l <node_list> -r customize
```

   For example, to migrate node 29, issue:

```
[ceedpart]/tmp>
spbootins -s no -p PSSP-2.3 -v aix421 -l 29 -r customize
```

4. Verify settings. The settings in the SDR must have the appropriate attributes. Verify these by issuing the

splstdata -b

command, as follows:

```
[ceedgate]/tmp/>splstdata -b
List Node Boot/Install Information

node#         hostname  hdw_enet_addr srvr     response     install_disk
last_install_image   last_install_time  next_install_image lppsource_name
                                                                 pssp_ver
-----------------------------------------------------------------------


29 ceed1n10.ppd.pok   000000000029   17        customize          hdisk0
           initial             initial            default         aix421
                                                                PSSP-2.3
```

- response should be set to customize
- lppsource_name should be set to aix421
- pssp_ver should be set to PSSP-2.3

5. To set up NIM properly on the Control Workstation, the setup_server command must be run, as follows:

   setup_server 2 > &1 |tee /tmp/setup_server.out

   The output is saved in the log file setup_server.out.

6. The nodes that will run PSSP 2.3 are now set in the SDR. To make these changes active on the Control Workstation and the nodes, the subsystems have to be refreshed. Issue the following command: syspar_ctrl -r -G

7. Switch fabric. (Systems without a switch may skip this step.) Use Efence to isolate nodes from the switch.

   Before issuing the Efence command, you have to verify if the node you are migrating is the Primary or Primary Backup Node. Issue the Eprimary command to check, and eventually change, the Primary and the Primary Backup node. For more information, see *PSSP Command and Technical Reference Guide*, GC-23-3900.

8. Copy the PSSP 2.3 pssp_script file to the /tmp directory of the node(s) you are migrating. To do this, you can use the rcp or ftp command.

9. Execute the pssp_script script that you copied to /tmp on the nodes you are migrating. Do not forget to do a chmod 700 pssp_script. After completion of the script, the bootp_response is set to disk.

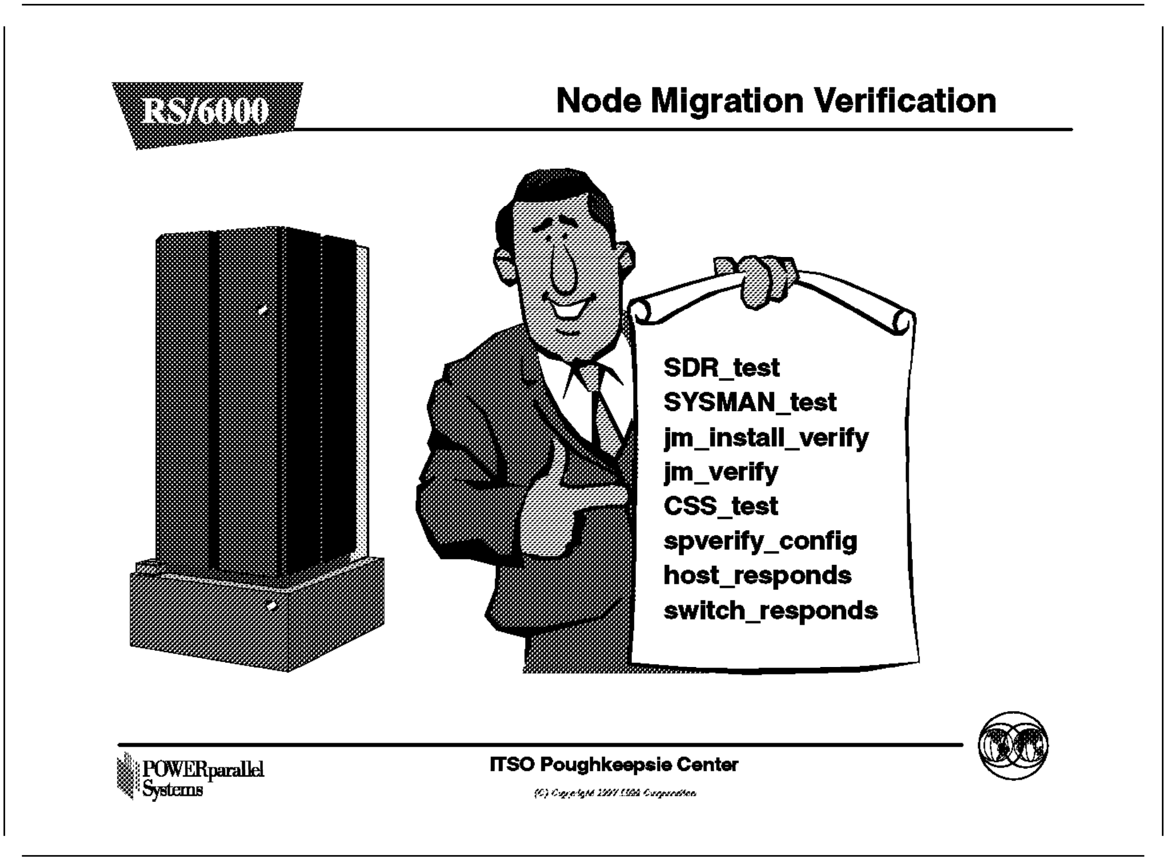
   Verify this by issuing splstdata -b.

10. Reboot the node. You must reboot the node, otherwise the changes to the kernel are not made active.

    The nodes will be installed when the LEDs are blank and host_responds is active.

11. Verification. See section 3.6, "Node Migration Verification" on page 86 for node verification.

## 3.6 Node Migration Verification



Node Migration Verification

SDR_test
SYSMAN_test
jm_install_verify
jm_verify
CSS_test
spverify_config
host_responds
switch_responds

ITSO Poughkeepsie Center

Verify the migrated nodes with the following scripts:

- SYSMAN_test
- jm_install_verify
- jm_verify
- CSS_test
- spverify_test

Verify host_responds and switch_responds with the command `spmon -G -d`:

```
1.  Checking server process
    Process 12748 has accumulated 3 minutes and 43 seconds.
    Check ok

2.  Opening connection to server
    Connection opened
    Check ok

3.  Querying frame(s)
    1 frame(s)
    Check ok

4.  Checking frames

        Controller   Slot 17  Switch   Switch     Power supplies
Frame   Responds     Switch   Power    Clocking   A   B   C   D
----------------------------------------------------------------
  1       yes         yes      on         0       on  on  on  N/A
5.  Checking nodes

--------------------------------- Frame 1 ----------------------------
Frame Node   Node        Host     Switch    Key     Env    Front Pane
Slot  Number Type  Power Responds Responds  Switch  Fail       LEDs
-------------------------------------------------------------------------
1      1    high    on     yes       yes   normal   no   LEDs are blank
5      5    high    on     yes       yes   normal   no   LEDs are blank
9      9    high    on     yes       yes   normal   no   LEDs are blank
15     15   high    on     yes       yes   normal   no   LEDs are blank
1      19   high    on     yes       yes   normal   no   LEDs are blank
5      21   high    on     yes       yes   normal   no   LEDs are blank
9      25   high    on     yes       yes   normal   no   LEDs are blank
```

Verify your applications, network, and so forth. Check the error logs. Make a system backup of the nodes after the migration is done and the nodes are working as expected.

After migrating the Control Workstation to PSSP 2.3, you may be supporting nodes at mixed levels of AIX and PSSP. Once all the nodes have been migrated to AIX level 4.2.1 and PSSP level 2.3, you can remove all NIM resources and files associated with this old level of AIX and PSSP. You may remove the files for AIX not equal to `aix421`, and for PSSP, those that are not equal to PSSP 2.3. For example:

- To remove NIM resources associated with AIX 4.1.5 issue:

  ```
  nim -o remove lppsource_aix415
  ```

- To remove the SPOT and files that NIM generated issue:

  ```
  nim -o spot_aix415
  ```

- To display the mksysb resources:

  ```
  lsnim -t mksysb -l
  ```

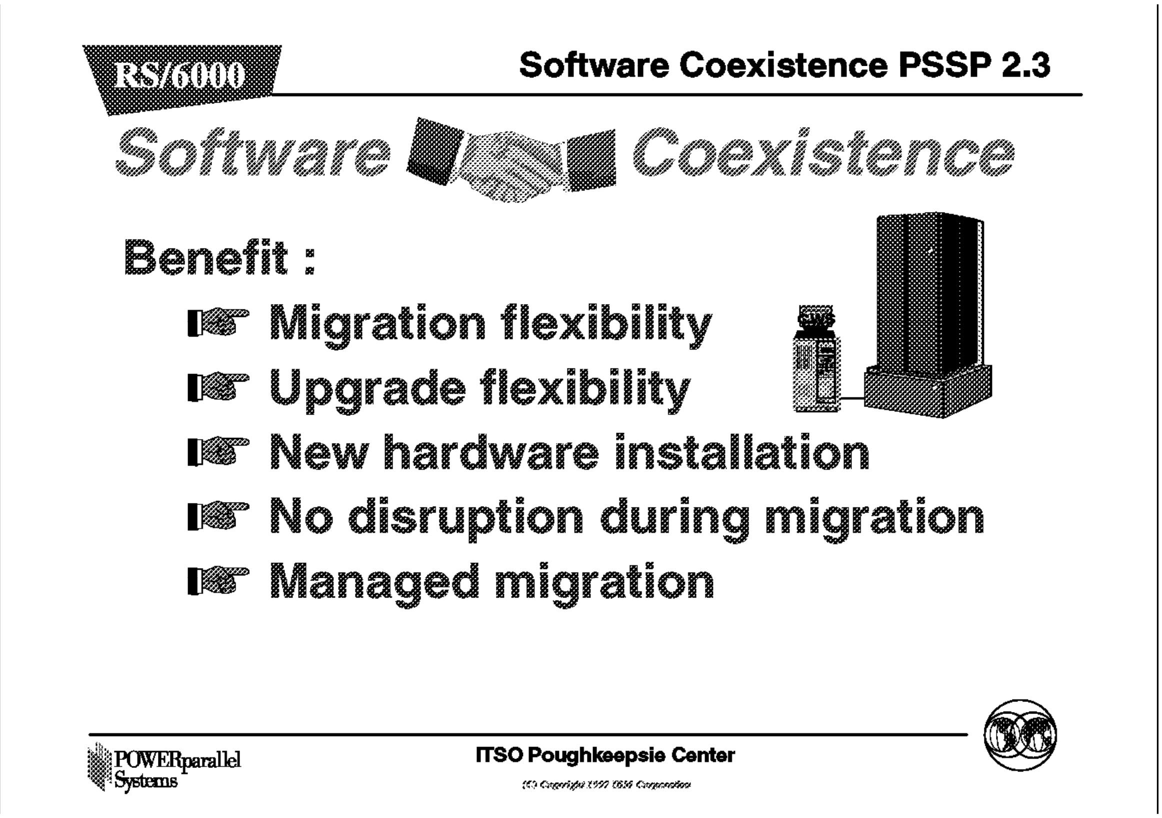
- To remove the mksysb resource:

  ```
  nim -o remove mksysb_415
  ```

- Remove the AIX files associated with AIX 4.1.5 and PSSP 2.2 in the following:

  ```
  /spdata/sys1/install/aix415
  /spdata/sys1/install/images/bos.obj.ssp.415
  /spdata/sys1/install/pssplpp/PSSP-2.2
  ```

# Chapter 4. Software Coexistence



RS/6000

**Software Coexistence PSSP 2.3**

**Software Coexistence**

**Benefit :**

- ☞ Migration flexibility
- ☞ Upgrade flexibility
- ☞ New hardware installation
- ☞ No disruption during migration
- ☞ Managed migration

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

IBM PSSP (Parallel System Support Programs) provides support for installation and management of the RS/6000 SP.

PSSP 2.3 supports multiple levels of AIX and PSSP in the same partition. Installations that benefit from coexistence include those that are using diverse applications on different levels of AIX and PSSP.

With PSSP 2.1, system partition was introduced. With this support, a set of nodes can be viewed as a logical subsystem within the RS/6000 SP. Multiple system partitions can be defined.

This gives a level of isolation and provides a mechanism for testing of new software and software releases. Multiple levels of AIX and PSSP can be used on nodes within the same RS/6000 SP.

In PSSP 2.1, all nodes in an RS/6000 SP partition must be at the same AIX and PSSP. PSSP 2.2 introduced support for two levels of PSSP, with corresponding levels of AIX, in the same partition.

PSSP 2.3 provides support for multiple levels of PSSP. This allows mixed system partitions with nodes running PSSP 1.2, PSSP 2.2 and PSSP 2.3; or PSSP 2.1, PSSP 2.2 and PSSP 2.3 at the same time.

Coexistence is intended to be a migration aid by providing flexibility for RS/6000 SP upgrades.

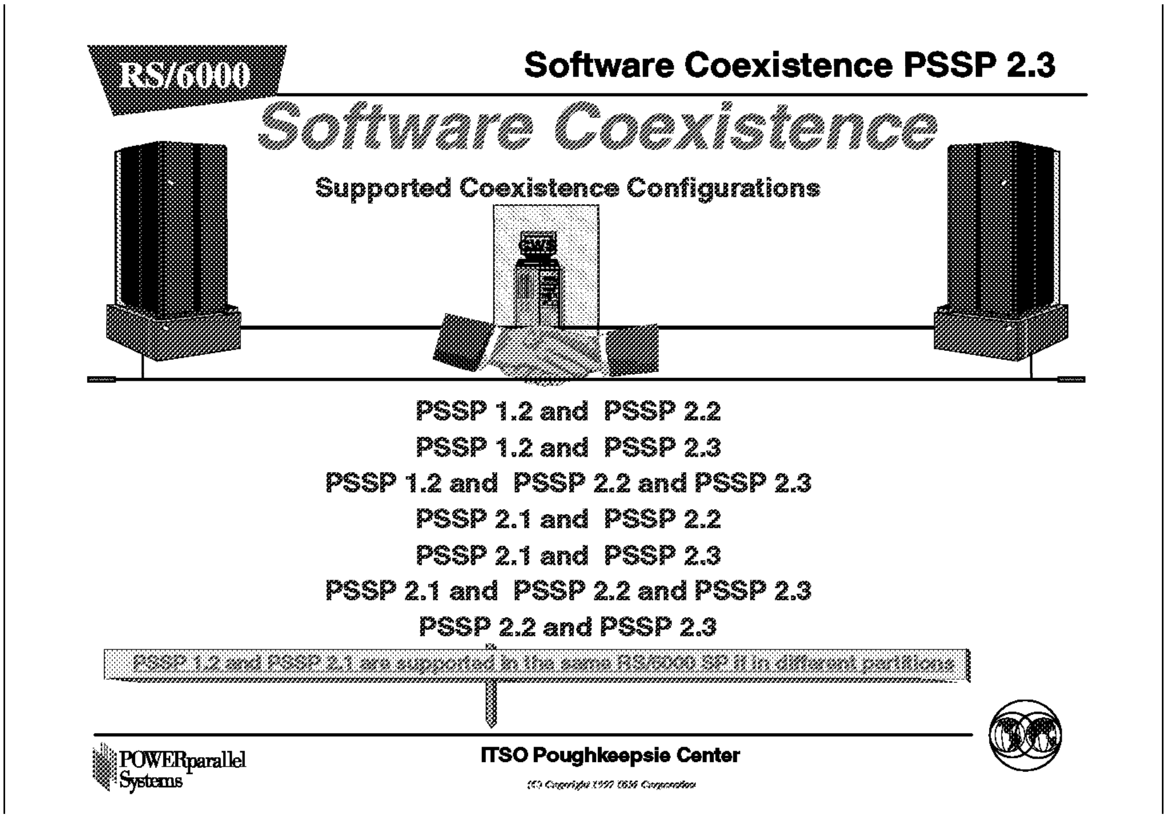
The main benefit of this support is improved upgrade or migration flexibility. This includes:

- Easier introduction of new RS/6000 SP technology
- Addition of new 604e High nodes to existing RS/6000 SP systems
- Upgrade granularity: the ability to add or migrate a single node at a time
- Avoiding disruption to other nodes in the system during the migration upgrade
- Support for customers for whom partitioning is not a solution, for instance small RS/6000 SP systems
- Allowing managed migration of the production application workload onto new software
- Ability to do rolling migration of all nodes running Oracle if high availability is configured while the system is operational

The AIX and PSSP coexistence, or mixed partitions support, should be of particular interest to those installations using the RS/6000 SP for database and commercial processing. These customers can reduce their system down time for maintenance because they can test and migrate without interrupting everything.

This presentation describes the software coexistence support in PSSP 2.3.

## 4.1 Software Coexistence 604e High Node



RS/6000

Software Coexistence PSSP 2.3

Software Coexistence

Supported Coexistence Configurations

PSSP 1.2 and PSSP 2.2
PSSP 1.2 and PSSP 2.3
PSSP 1.2 and PSSP 2.2 and PSSP 2.3
PSSP 2.1 and PSSP 2.2
PSSP 2.1 and PSSP 2.3
PSSP 2.1 and PSSP 2.2 and PSSP 2.3
PSSP 2.2 and PSSP 2.3

PSSP 1.2 and PSSP 2.1 are supported in the same RS/6000 SP if in different partitions

POWERparallel Systems

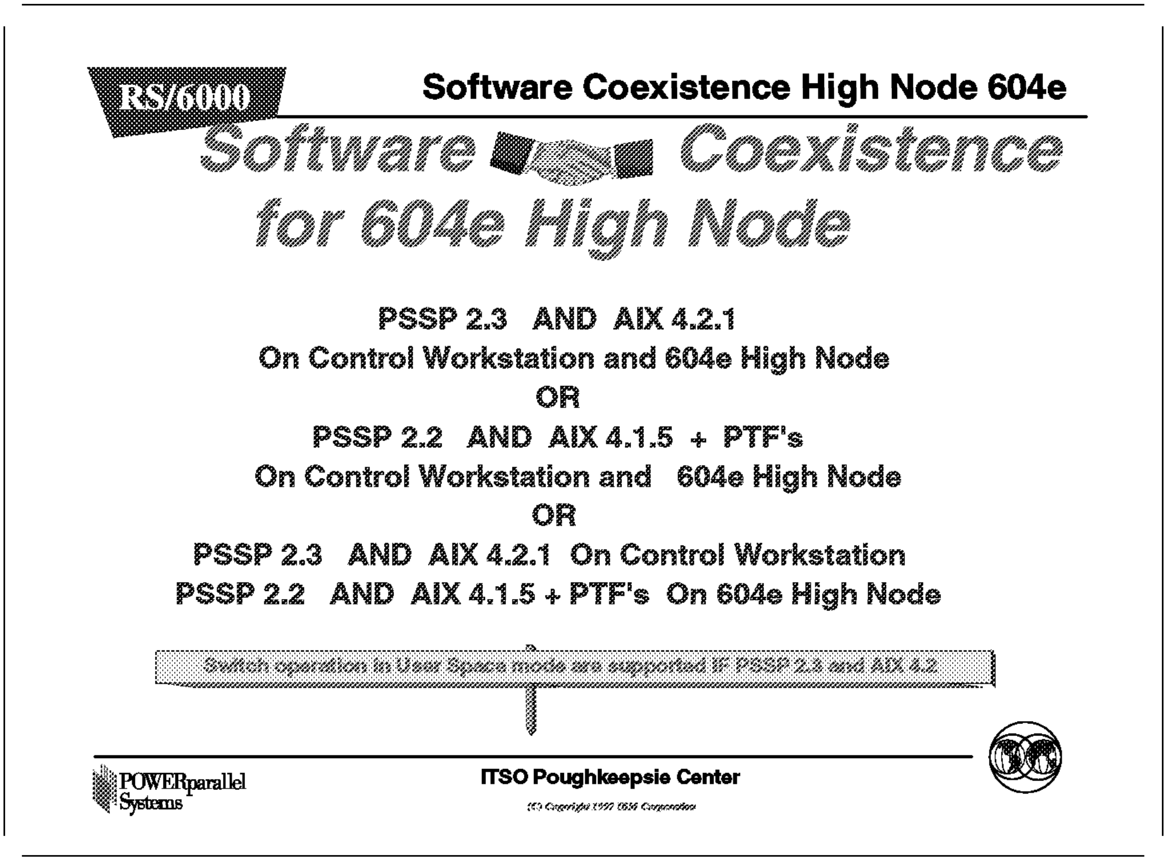
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

- Supported coexistence configurations.
    - PSSP 1.2 and PSSP 2.2
    - PSSP 1.2 and PSSP 2.3
    - PSSP 1.2 and PSSP 2.2 and PSSP 2.3
    - PSSP 2.1 and PSSP 2.2
    - PSSP 2.1 and PSSP 2.3
    - PSSP 2.1 and PSSP 2.2 and PSSP 2.3
    - PSSP 2.2 and PSSP 2.3
  - Each PSSP has his corresponding level of AIX
    - PSSP 1.2 = AIX 3.2.5
    - PSSP 2.1 = AIX 4.1.3, 4.1.4, 4.1.5
    - PSSP 2.2 = AIX 4.1.4, 4.1.5, 4.2.0, 4.2.1
    - PSSP 2.3 = AIX 4.2.1

**Note:** PSSP 1.2 and PSSP 2.1 are supported in the same RS/6000 SP if they are in different partitions.

## 4.2 Software Coexistence 604e High Node



RS/6000

Software Coexistence High Node 604e

Software Coexistence for 604e High Node

PSSP 2.3 AND AIX 4.2.1
On Control Workstation and 604e High Node
OR
PSSP 2.2 AND AIX 4.1.5 + PTF's
On Control Workstation and 604e High Node
OR
PSSP 2.3 AND AIX 4.2.1 On Control Workstation
PSSP 2.2 AND AIX 4.1.5 + PTF's On 604e High Node

Switch operation in User Space mode are supported IF PSSP 2.3 and AIX 4.2

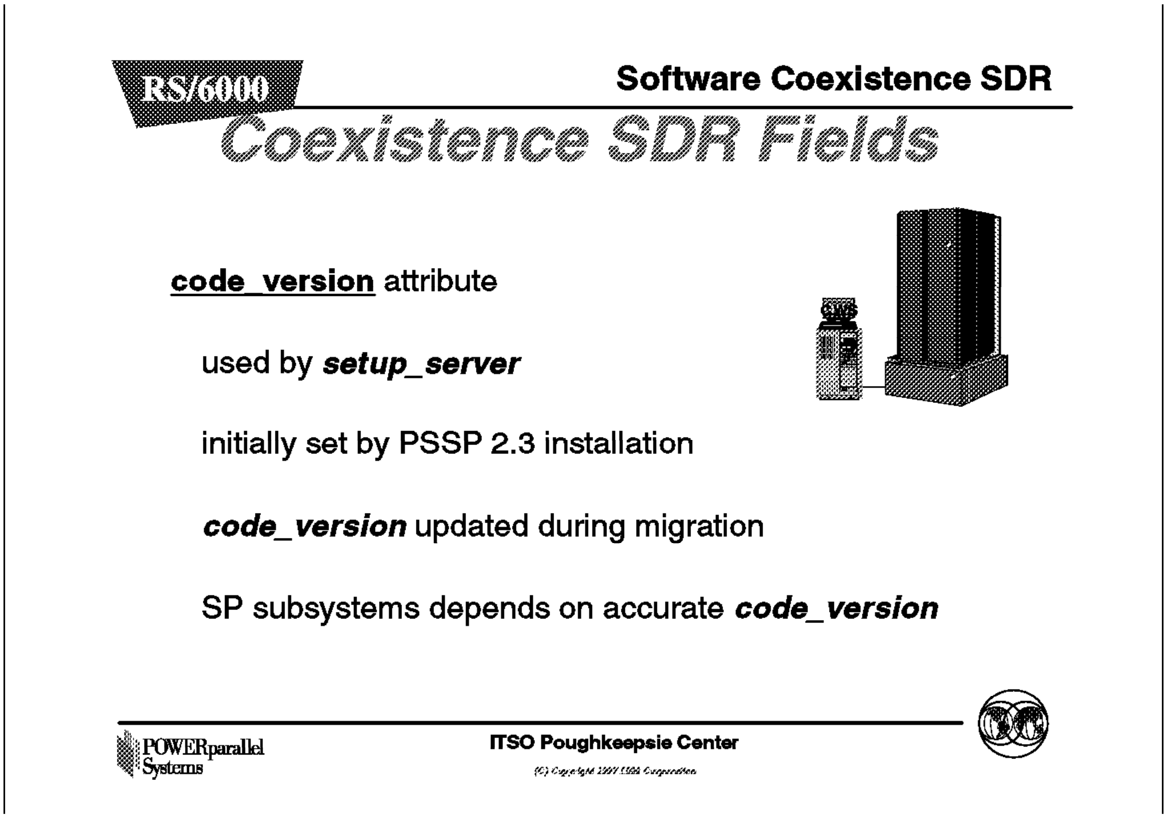
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

It is possible to have 604e High node with PSSP 2.2 and AIX 4.1.5 plus PTFs, and 604e High node with PSSP 2.3 and AIX 4.2.1 in the same RS/6000 SP system. The CWS has to be at the latest level of PSSP and AIX.

## 4.3 SDR Fields



RS/6000

Software Coexistence SDR

Coexistence SDR Fields

**code_version** attribute

used by ***setup_server***

initially set by PSSP 2.3 installation

***code_version*** updated during migration

SP subsystems depends on accurate ***code_version***

CWS
POWERparallel Systems

ITSO Poughkeepsie Center

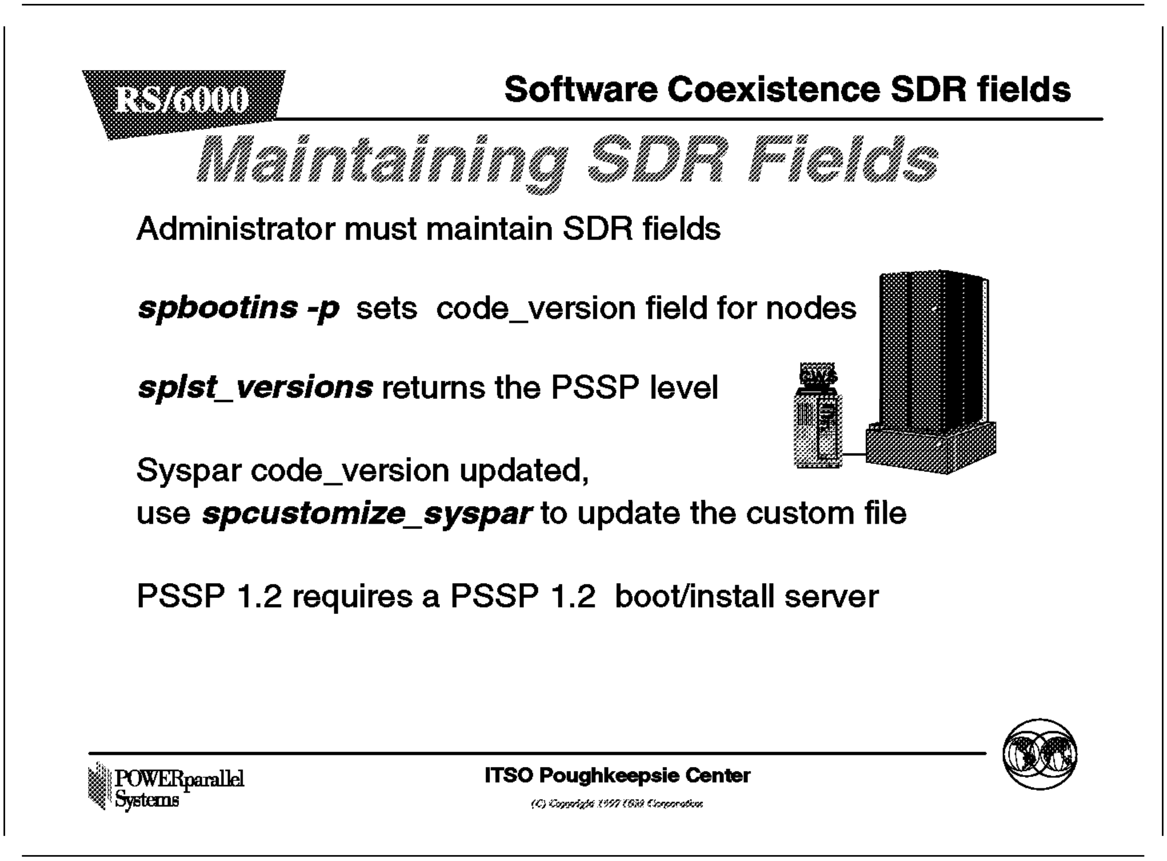
Coexistence SDR fields.

The code_version attribute of the SDR node object was not used in PSSP 2.1. or 1.2. It is used to set the level of PSSP running on the node in PSSP 2.2 and 2.3. The code_version attribute of the SDR Syspar object, which in earlier releases represented the PSSP level of all nodes in the system partition, is in PSSP 2.2 and PSSP 2.3 used to set the earliest PSSP level running in that System partition.

Initialization of the code_version attribute for nodes is done when the CWS is migrated to PSSP 2.3. The installation software propagates the Syspar code_version value. New installations, the code_version attribute is set as part of the creation of the Node objects.

Initially the PSSP 2.3 installation code set these values. Node code_version are updated as the respective nodes within a system partition are migrated to a later release. The field is used by setup_server for installation of a newer release of PSSP. The node code_version represents the level running on the node. RS/6000 SP subsystems (topology services,heartbeat,etc) depends on an accurate state of this SDR attribute for proper operation within a coexistence environment.

## 4.4 Maintaining SDR



RS/6000

Software Coexistence SDR fields

**Maintaining SDR Fields**

Administrator must maintain SDR fields

***spbootins -p*** sets code_version field for nodes

***splst_versions*** returns the PSSP level

Syspar code_version updated,
use ***spcustomize_syspar*** to update the custom file

PSSP 1.2 requires a PSSP 1.2 boot/install server

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

SDR fields setting and retrieving.

The maintenance of the SDR fields is the responsibility of the administrator. It is not an automated process performed as part of node upgrades. The following interfaces are used to set and retrieve the code_version attribute of a node.

- spbootins -p
- splst_versions

The -p option of the spbootins command set the code_version field for the node. You can set the target PSSP level (PSSP 2.3) during a migration or reconfiguration of a node.

Syspar code_version is not set manually. The spbootins command will update this attribute if necessary. When the Syspar code_version is updated, the system partition configuration custom file is not updated . Therefore, the spverify_syspar will report a mismatch. You can update the custom file by invoking spcustomize_syspar. The splst_versions command returns the PSSP level of a specified node, node group, or system partition.

With this output you can determine what PSSP levels are running on specific nodes in a system partition. You can check if you have a mixed system partition.

```
[c201cw]/> splst_versions

PSSP-2.2
PSSP-2.3
```

The output of the command splst_versions gives the PSSP levels installed on the RS/6000 SP in the current partition.

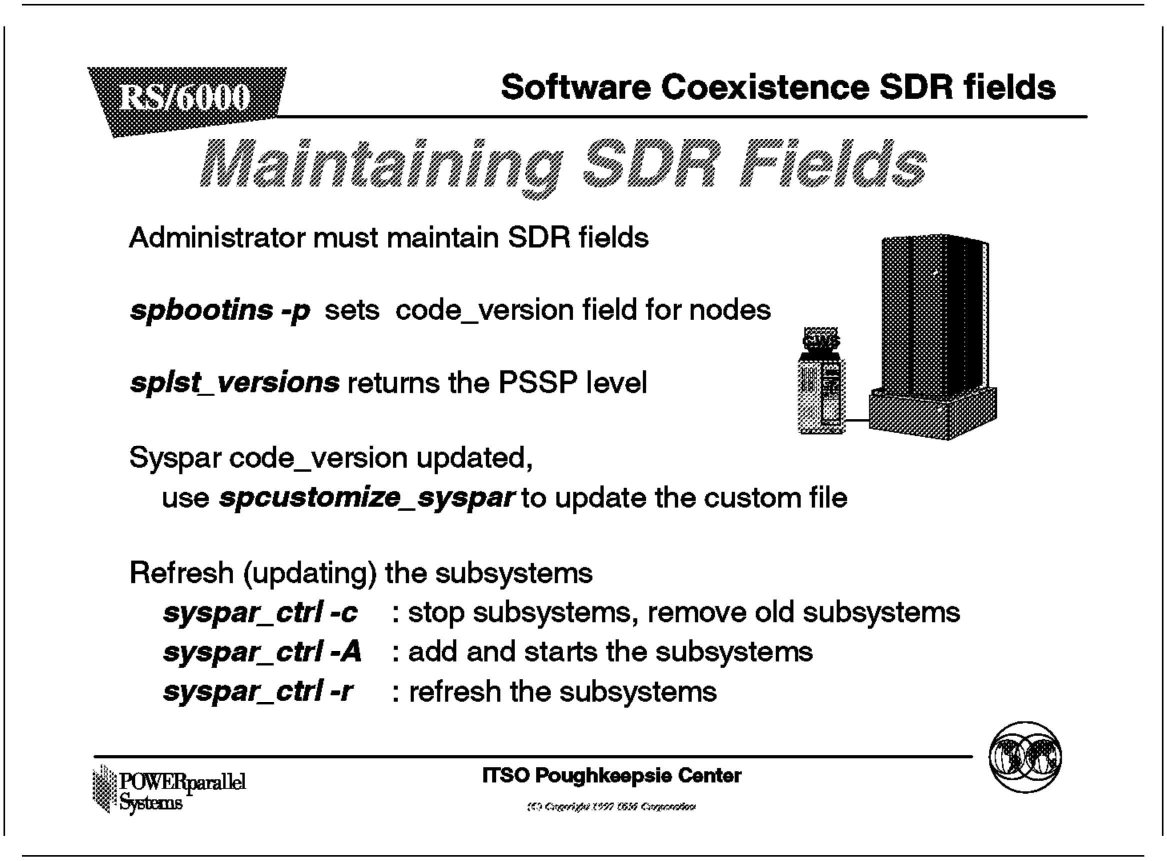
```
[c201cw]/> splst_versions -t
1  PSSP-2.3
5  PSSP-2.3
9  PSSP-2.2
13 PSSP-2.2
```

The output of the command splst_versions -t gives the PSSP levels installed on the RS/6000 SP nodes.

```
[c201cw]/> splstdata -b


                  List Node Boot/Install Information

Node#   hostname  hdw_enet_addr    srvr    response        install_disk
last_install_image   last_install_time  next_install_image lppsource_name
                                                                 pssp_ver
--------------------------------------------------------------------
1 c201n01.ppd.pok.   02608CE908B8    0         disk             hdisk0
 bos.obj.ssp.41 Fri_May__2_22:11:37      bos.obj.ssp.42        aix421
                                                              PSSP-2.3

5 c201n02.ppd.pok.   02608CE908DA    0         disk             hdisk0
 bos.obj.ssp.42 Fri_May__2_19:05:04      bos.obj.ssp.42        aix421
                                                              PSSP-2.3

9 c201n03.ppd.pok.   02608CE908D2    0         disk             hdisk0
 bos.obj.ssp.41 Wed_Apr_30_12:39:38      bos.obj.ssp.41        aix415
                                                              PSSP-2.2

13 c201n04.ppd.pok.   02608CF5056D    0      migrate            hdisk0
 bos.obj.ssp.41 Wed_Apr_30_12:39:26      bos.obj.ssp.42        aix420
                                                              PSSP-2.2
```

The output of the command splstdata -b gives the PSSP code_versions and level of AIX. Be aware that in case the boot response is customize or migrate, splstdata -b shows the PSSP version and AIX level which will be installed after the next net-boot.

## 4.5 The syspar_ctrl Command



RS/6000

**Software Coexistence SDR fields**

**Maintaining SDR Fields**

Administrator must maintain SDR fields

***spbootins -p*** sets code_version field for nodes

***splst_versions*** returns the PSSP level

Syspar code_version updated,
use ***spcustomize_syspar*** to update the custom file

Refresh (updating) the subsystems
***syspar_ctrl -c*** : stop subsystems, remove old subsystems
***syspar_ctrl -A*** : add and starts the subsystems
***syspar_ctrl -r*** : refresh the subsystems

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

Syspar controller manages certain RS/6000 SP subsystems. Including those with dependencies on node_version. It is used for migrating and managing partitions.

If there are no partitions set, you still have a single default system partition. Many RS/6000 SP subsystems operate within the domain of a system partition. When nodes are migrated to PSSP 2.3, the system partition sensitive subsystems need to be updated. This involves the following:

- Stopping and deleting the appropriate subsystem components (daemons) running on the affected node.
- Adding and starting subsystems components.
- Refreshing and updating the subsystems on the CWS. The following is an example:

```
The hats subsystem(PSSP 2.2) or the hb subsystem(PSSP 2.1)
provided by the topology services and
event management infrastructure, is to be updated.
After migration of the CWS to PSSP 2.3, the subsystems running
on the CWS would include hats, haem, hags, hb and hr.
When migrating one of the nodes from PSSP 2.1 to PSSP 2.3, hb is
stopped and removed from that node.
The new subsystems hats , haem, hags are added and started.
On the CWS the subsystems are refreshed to reflect the change in
the node.
```

Uses of syspar_ctrl:

- syspar_ctrl -E returns a list of the subsystems managed by the Syspar Controller.
- syspar_ctrl -c stops and cleans the subsystems.
- syspar_ctrl -A adds and starts the new subsystems.
- syspar -r refreshes the subsystems.

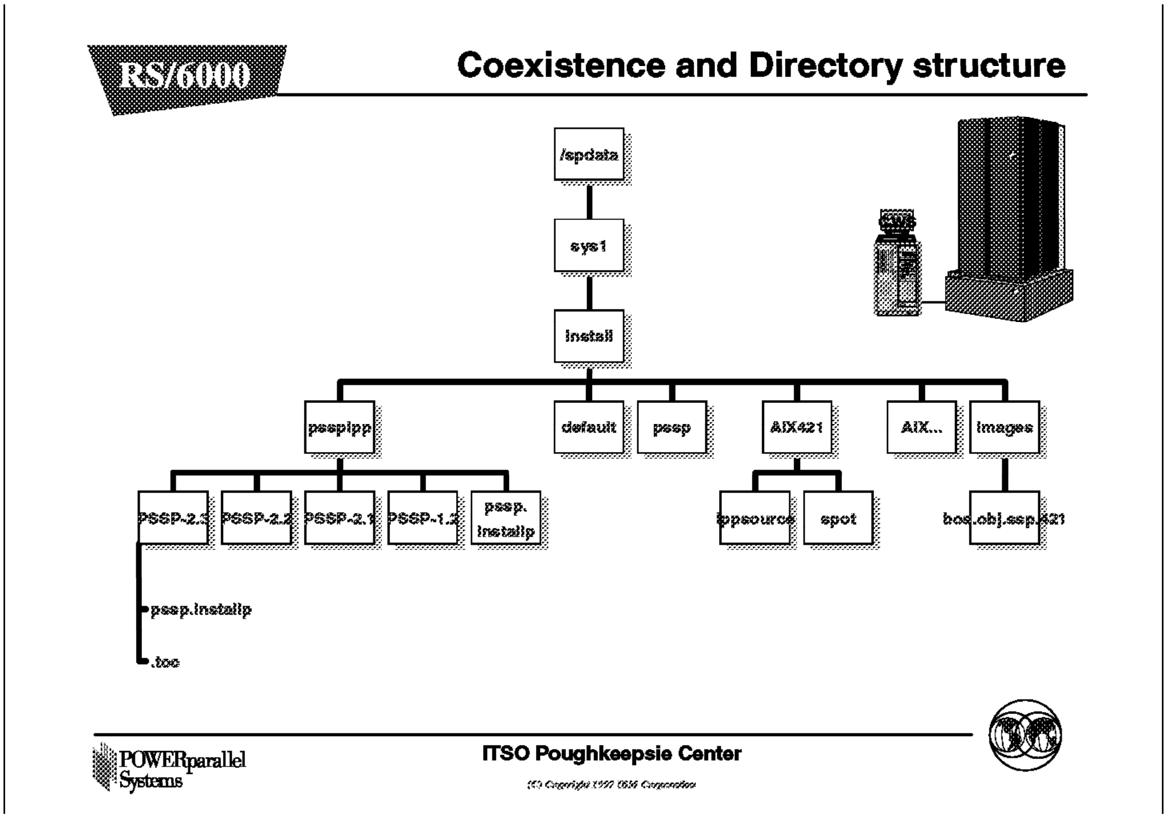
**syspar-ctrl**

The role of the Syspar Controller and how subsystems are managed, is a useful problem isolation tool. If a node is migrated to PSSP 2.3, and you have an inoperative host_responds, you can verify that the subsystems have been started by: lssrc -a. The hats subsystem is responsible for host_responds. The hr subsystem invokes the haem subsystem. The haem subsystem invokes the hags subsystem. The hags subsystem invokes the hats subsytem. You can verify by invoking syspar_ctrl -E and lssrc -a|pg

**Note:** In a migration scenario that involves a reboot, the script rc.sp runs syspar_ctrl -c and syspar_ctrl -A. The administrator is responsible for the refresh operation: syspar_ctrl -r.

When installing or migrating, the component′s name of an LPP could change. If upgrading from PSSP 1.2 to PSSP 2.3, the component name for VSD is different. The csd image must be de-installed before installing the PSSP 2.3 ssp.csd image. Some products have PSSP and AIX dependencies. Ensure that the LPP releases are supported on the new PSSP 2.3 and AIX 4.2.1. The *PSSP System Planning guide* summarizes the LPPs that are supported on different PSSP releases.

## 4.6 Directories



RS/6000

Coexistence and Directory structure

/spdata

sys1

install

pssplpp

default

pssp

AIX421

AIX...

images

PSSP-2.3

PSSP-2.2

PSSP-2.1

PSSP-1.2

pssp. installp

lppsource

spot

bos.obj.ssp.421

pssp.installp

.toc

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

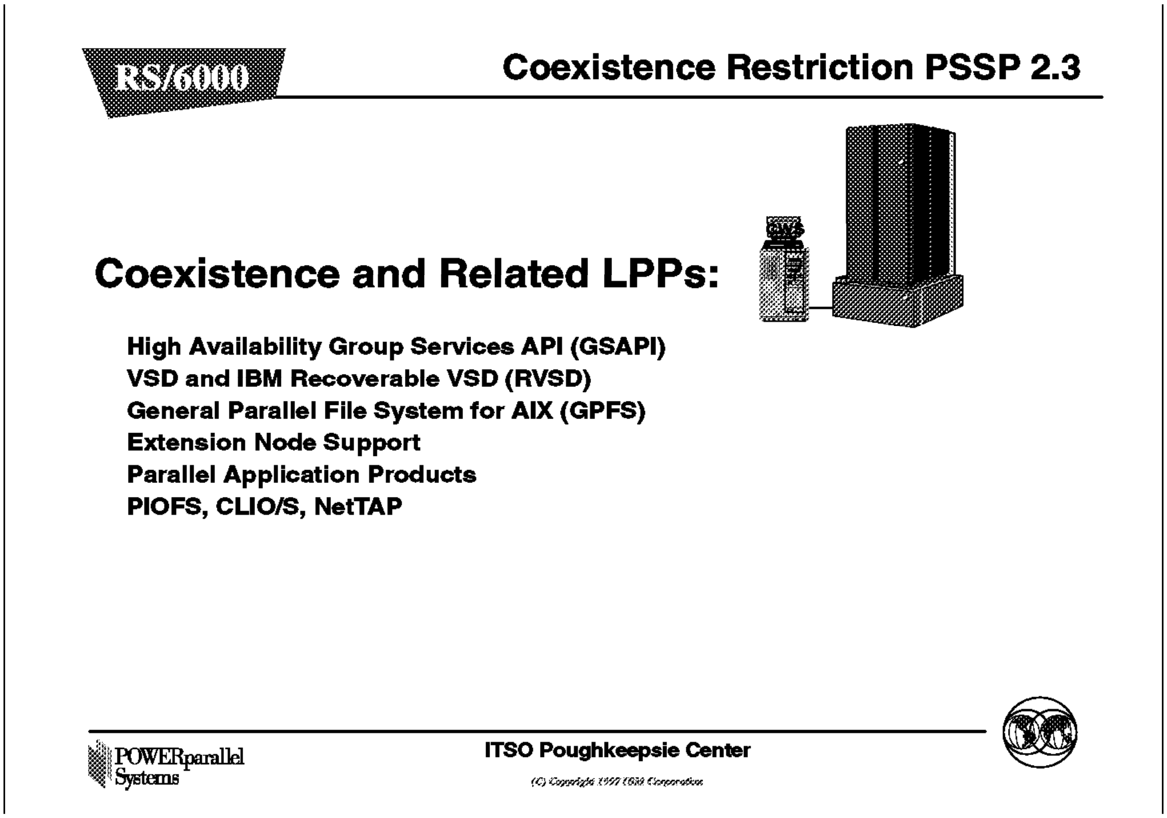
The directory structure to support all releases of PSSP and also to support the AIX versions or releases.

The pssplpp directory contains the directories with the different PSSP versions. In the pssplpp directory is also a pssp.installp directory, this is a symbolic link /spdata/sys1/install/pssplpp/PSSP-2.1/pssp.installp. This way the software coexistence with version 2.1 of PSSP is maintained. This symbolic link is only needed for PSSP 2.1. To support coexistence, you need a psslpp directory for every of PSSP that exist in the RS/6000 SP system.

The default directory is not necessary if you define for example AIX42 directory as the directory for the AIX images and for the NIM SPOT. The definition is entered in the site information frame. Smit fast_path : smitty site_env_dialog.

The images directory has the different system backups. To support coexistence you need an image of every version of AIX that exist in the RS/6000 SP system.

## 4.7 Conclusion



RS/6000

**Coexistence Restriction PSSP 2.3**

**Coexistence and Related LPPs:**

**High Availability Group Services API (GSAPI)**
**VSD and IBM Recoverable VSD (RVSD)**
**General Parallel File System for AIX (GPFS)**
**Extension Node Support**
**Parallel Application Products**
**PIOFS, CLIO/S, NetTAP**

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

#### 4.7.1.1 High Availability Group Services API (GSAPI)

Programmers writing for the GSAPI and also systems administrators with systems using GSAPI need to be aware that all nodes must be at PSSP 2.3 in order to utilize the new PSSP 2.3 GSAPI functions. Systems with mixed PSSP 2.3 and 2.2 nodes can only operate at the PSSP 2.2 level until all nodes are at PSSP 2.3.

The GSAPI function is not available in PSSP 2.1 or PSSP 1.2.

#### 4.7.1.2 VSD and IBM Recoverable VSD (RVSD)

In mixed system partitions containing PSSP 1.2 or PSSP 2.1 nodes, the VSD subsystem in PSSP 2.3 and PSSP 2.2 can coexist with the VSD subsystem at these earlier levels, but VSDs can only be configured on and used by nodes at the same PSSP level.

- PSSP 2.3 and PSSP 2.2 nodes only configure VSDs that are served by PSSP 2.3 and PSSP 2.2 nodes.
- Attempts to configure VSDs on PSSP 1.2 or PSSP 2.1 nodes for VSDs served by PSSP 2.3 and PSSP 2.2 nodes will succeed, but requests to these VSD′s will not be served and will eventually time out.

The VSD subsystem in PSSP 2.3 can interoperate with the VSD subsystem in PSPS 2.2, but the level of function available in this configuration is the PSSP 2.2 level. In order to exploit the new function in the PSSP 2.3 VSD subsystem and RVSD 2.1, all nodes in a system partition must be running PSSP 2.3 and RVSD 2.1. Additionally, when the last PSSP 2.2 node is migrated to PSSP 2.3, all nodes in the system partition must be reconfigured/rebooted to enable the nodes in the system partition to use the PSSP 2.3/RVSD 2.1 level of function.

RVSD includes the following quorum rules/restrictions in a coexistence environment:

- RVSD 2.1 (PSSP 2.3) and RVSD 1.2 (PSSP 2.2) treat nodes running earlier releases of RVSD/PSSP as down. Similarly, earlier releases of RVSD do not recognize RVSD 2.1 or RVSD 1.2 nodes.
- Quorum will be evaluated as followscolon.
  - RVSD 2.1 (PSSP 2.3) and RVSD 1.2 (PSSP 2.2):

    ```
    number_of_PSSP_2.3_or_2.2_VSD_nodes/2 + 1
    ```

    **Note:** Quorum may be overridden by the administrator in RVSD 2.1 and RVSD 2.1.
  - RVSD 1.0 (PSSP 1.2) and RVSD 1.1 (PSSP 2.1):

    ```
    number_of_all_VSD_nodes + CWS)/2 + 1
    ```

    **Note:** Upgrading more than half of the VSD nodes to PSSP 2.3 or PSSP 2.2 causes the VSD group running on earlier releases to become inactive.

#### 4.7.1.3 General Parallel File System for AIX (GPFS)

GPFS is not supported in a coexistence configuration. All nodes within a system partition that requires GPFS must be running PSSP 2.3. GPFS requires RVSD 2.1 and all the within a system partition with GPFS must have RVSD 2.1.

#### 4.7.1.4 Extension Node Support

Extension Node support in PSSP 2.3 functions in a mixed system partition, but the primary node and the primary backup node must be running PSSP 2.3. As the SPS switch is a prerequisite for the Extension Node, AIX 3.2.5/PS SP 1.2 within the system partion is not supported.

#### 4.7.1.5 Parallel Application Products

Parallel applications like IBM Parallel Environment for AIX, or Parallel ESSL for AIX are not supported in a mixed partition. This applies to their use for either IP or user space communication. Parallel applications can only run in a system partition that has all of its nodes at the same PSSP level.

#### 4.7.1.6 Loadleveler

Loadleveler 1.2.1 and 1.2.0 coexistence is supported for serial scheduling in a system partition with nodes running PSSP 2.2 and PSSP 1.2, respectively. Loadleveler 1.2.1 is supported on both PSSP 2.1 and PSSP 2.2. Loadleveler 1.3 (running on PSSP 2.3 or PSSP 2.2), is not compatible with earlier levels of Loadleveler.

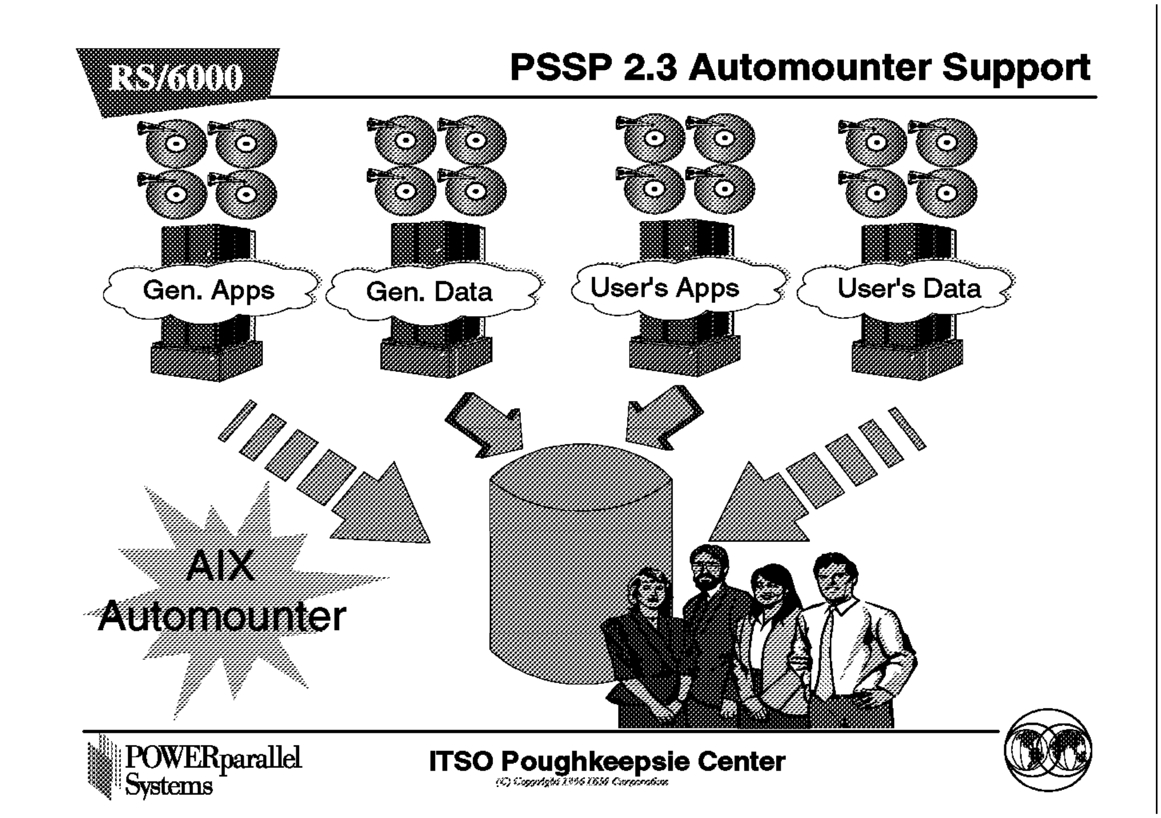
#### 4.7.1.7 PIOFS, CLIO/S, NetTAPE

The following products at the specified release levels support PSSP 2.3, PSSP 2.2 and PSSP 2.1, but are not supported for migration in a mixed partition. The following products require AIX 4.1 (except the NetTAPE products) and are not compatible with releases running on PSSP 1.2:

- Parallel I/O File System 1.2
- IBM Client Input Output/Sockets 2.2
- IBM Network Tape Access and Control System (NetTAPE) for AIX, and IBM NetTAPE Tape Library Connection

# Chapter 5. AIX Automounter

## 5.1 Overview of AIX Automounter



PSSP 2.3 Automounter Support

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

When users or applications work on a standalone RS/6000 machine, they have access to the data that is on the local disk in a familiar and standard way: they see the data as files in the local file systems.

The RS/6000 SP machine is a network of high-performance nodes. Each of these nodes has its own CPU, memory, operating system, and local disks. Each of the disks on the nodes can be used to store both user and application data and binaries. But a problem arises: each node is a separate machine on the network, so the data might be spread over many nodes. Each user and application has to be aware of this and connect to different nodes, depending on what data they want to use.

This fact would make the machine more difficult to use and to write applications for.

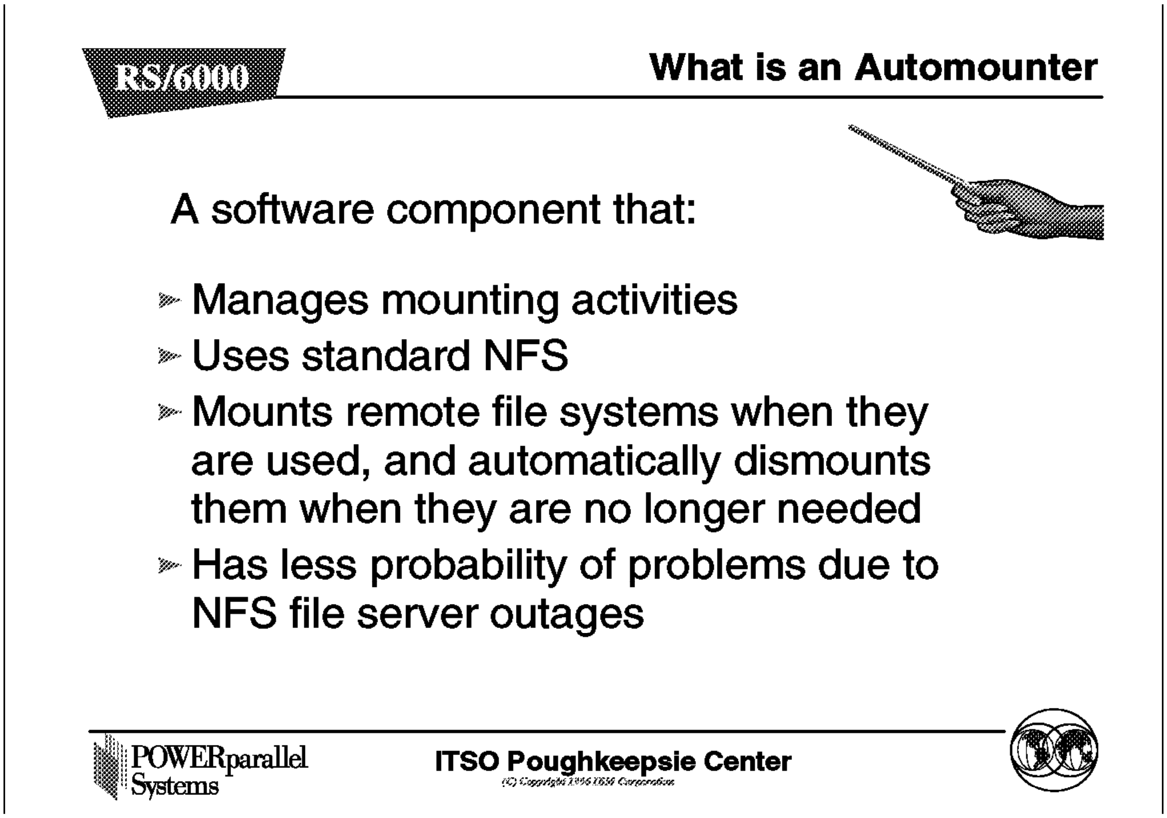
The data on each node would have to be made available to the others in some transparent way, and the data located in remote nodes would have to look as if it were local data to the client machines.

In this way the RS/6000 SP would appear as only one machine to both the end users and the applications, with the data on each node that is to be shared available to all other nodes, creating a global repository of storage available to all the applications and users.

From the system administration point of view, we would like to have a method of making the administration of this sharing of data between the nodes easy and efficient.

In this chapter we discuss a tool that accomplishes this: the AIX Automounter.

## 5.2 What is an Automounter?



RS/6000

What is an Automounter

A software component that:

- Manages mounting activities
- Uses standard NFS
- Mounts remote file systems when they are used, and automatically dismounts them when they are no longer needed
- Has less probability of problems due to NFS file server outages

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

By now, most of us are familiar with working network resources. One of the most used network resources is the disk of file servers. We want certain machines to be able to access other machines′ disk resources as transparently as possible. On AIX, this task is performed by the Network File System (NFS). NFS lets one machine, called the server, make its disk resources available to other client machines on the network. Each client machine can request services from many of those NFS servers.

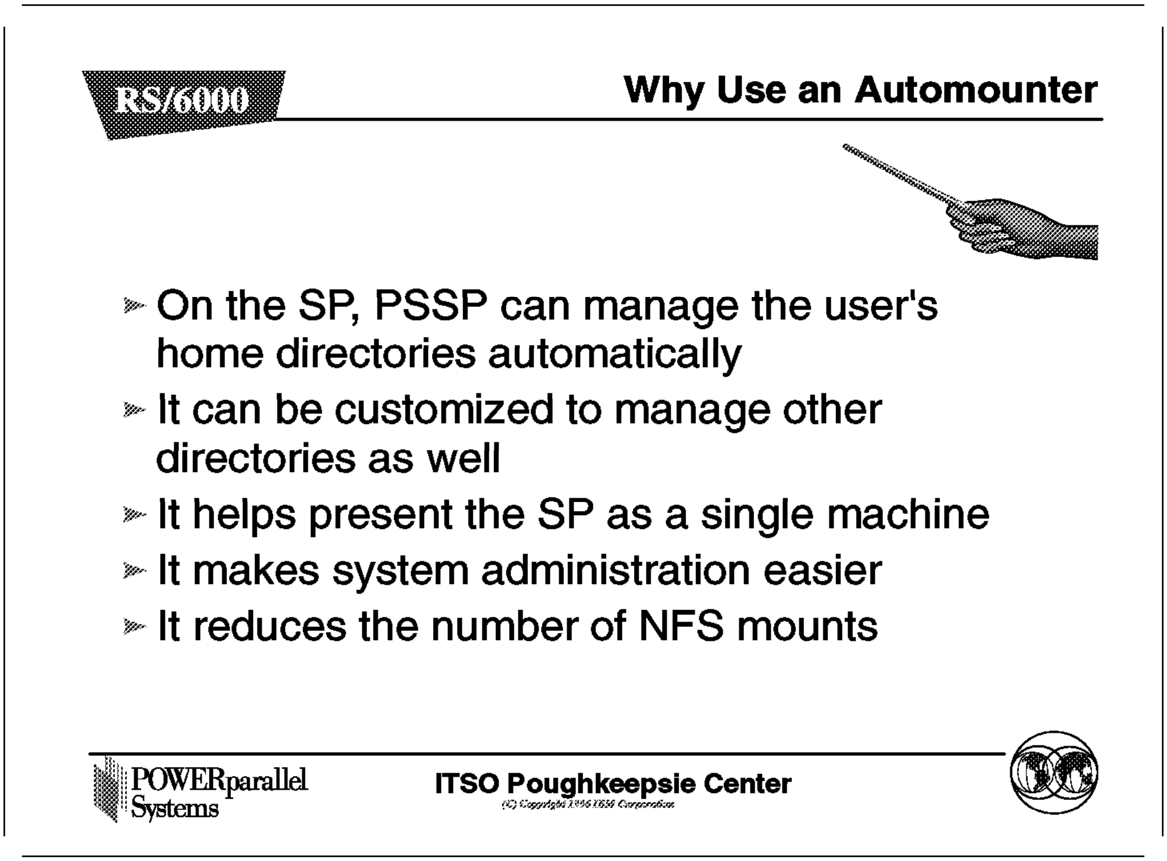
In a plain NFS environment, the system administrator would have to export the desired file systems on the server, and then configure each client machine to mount the remote file system (that is, to make it accessible on that machine). From that point, the client machine can see the data on the remote file system as local.

As the number of client machines grows, the management cost of keeping all the clients up to date can be high. So, a more efficient way of specifying remote mounts on the client should make the system administrator′s task easier, and also be more productive. That would result in improved overall system usability.

The Automounter is a tool that does this. When you access a file or directory on a client machine, and that file system is under control of an Automounter, then this facility will do the required mounts for you.

Also, when there has been no activity for some configurable period of time on the mounted file system, the Automounter will unmount the file system automatically.

For the mounting activity of an NFS file system, the Automounter will use standard NFS mounting facilities. It will reduce the amount of time that a remote file system is mounted to the amount of time that file system is actually needed on the local machine. It will also reduce the number of mounts on a given system, and will help in reducing system problems due to NFS server outages.



RS/6000

## Why Use an Automounter

- On the SP, PSSP can manage the user's home directories automatically
- It can be customized to manage other directories as well
- It helps present the SP as a single machine
- It makes system administration easier
- It reduces the number of NFS mounts

POWERparallel Systems

ITSO Poughkeepsie Center

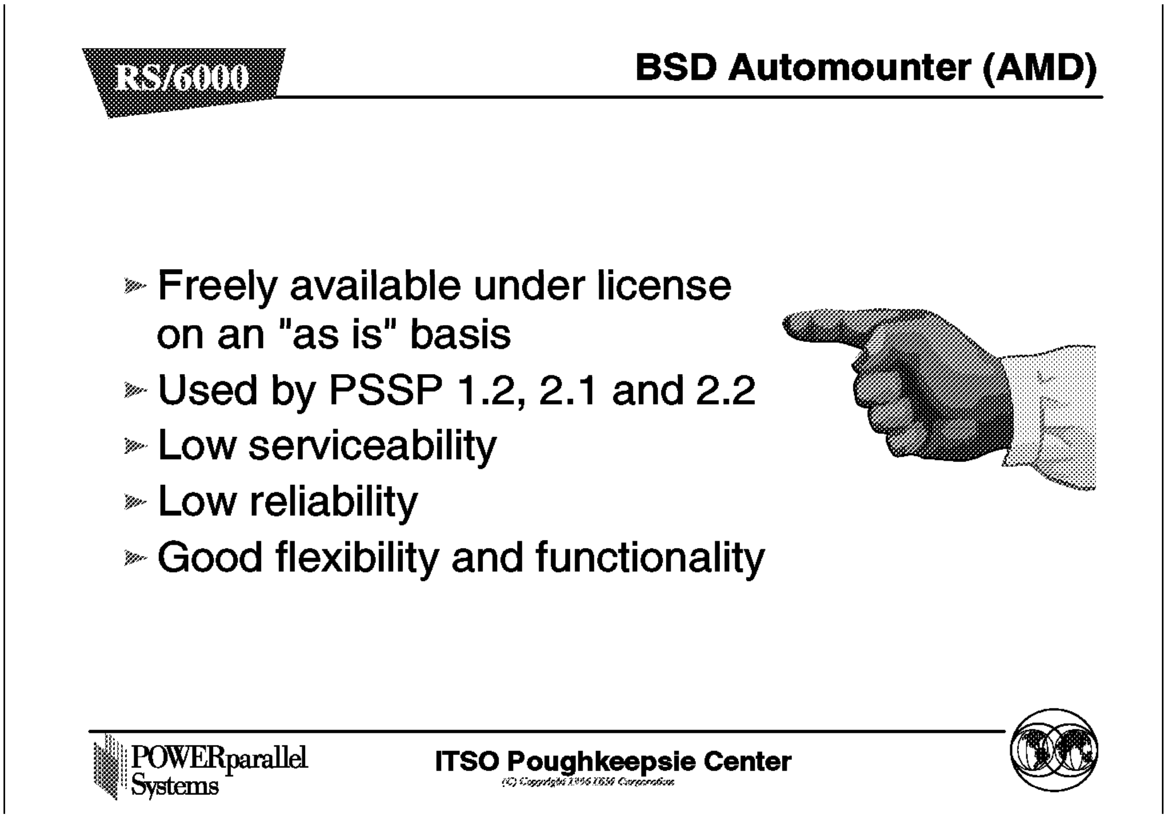
(C) Copyright 1996 IBM Corporation

On the SP, the Automounter is optionally used to manage the mounting of home directories. It can be customized to manage other directories as well. When configured, an Automounter daemon runs on each node and is started when the node is booted. This also applies to the Control Workstation. The mounted directories might be served by any NFS server on the network. If the directory to be mounted appears as a local directory to the machine (as can be the case with AFS), then the Automounter will simply create a symbolic link to that target directory instead of attempting to mount it.

The Automounter manages directories specifically defined in the Automounter configuration files, or map files. These can reside on each node locally, or can be accessed by means of the Network Information Services (NIS). Typically, there is one map file for each file system to be controlled by the Automounter. If SP User Management has been configured, then PSSP will create and maintain a map file to control user home directories under the /u file system.

As mentioned earlier, the Automounter can be customized to manage other directories as well. That helps the end users and applications because they see the same file system structure on all the nodes they have access to. Also, it makes system administration easier, since the system administrator only has to modify map files on the control workstation in order to make changes on a system-wide basis.

## 5.3 New Automounter in PSSP 2.3



RS/6000

### BSD Automounter (AMD)

- Freely available under license on an "as is" basis
- Used by PSSP 1.2, 2.1 and 2.2
- Low serviceability
- Low reliability
- Good flexibility and functionality

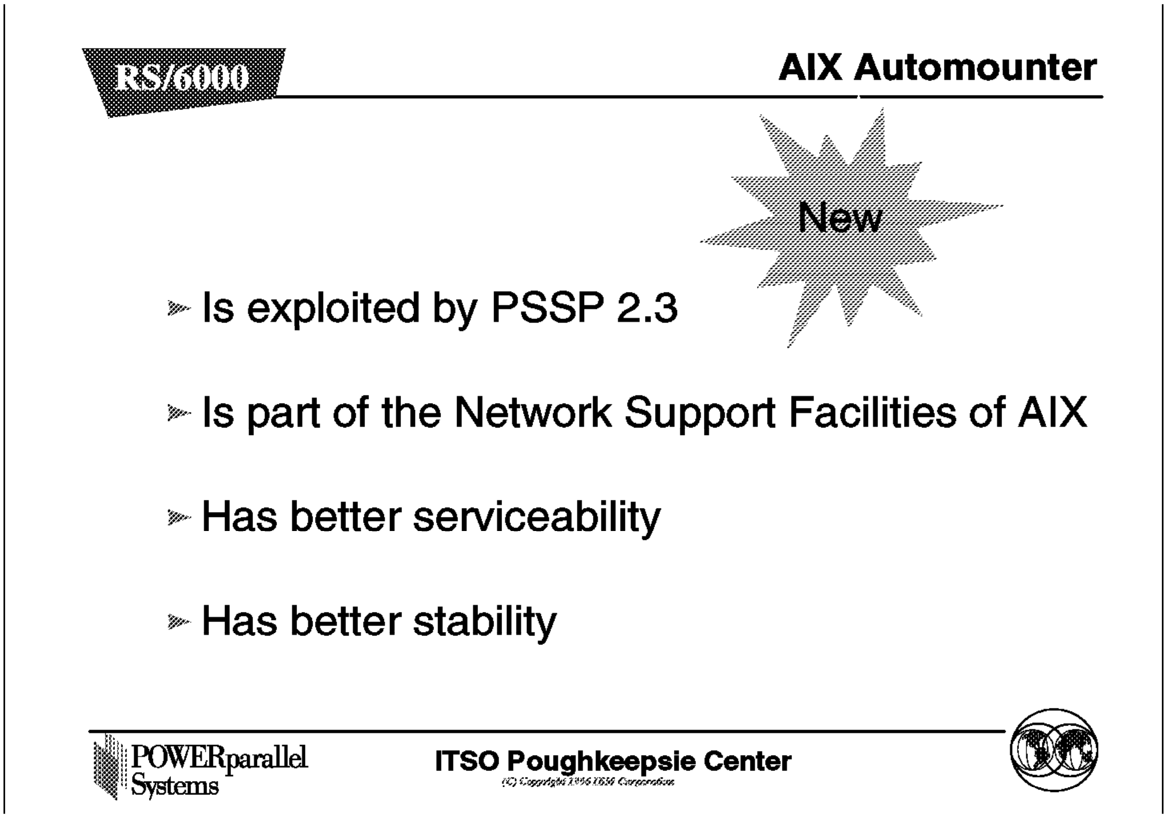
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

In releases of PSSP prior to 2.3, the BSD Automounter (also known as AMD) was exploited. This package offers good flexibility and functionality. It has many options that enable the System Administrator to specify many aspects of how and where the NFS mounts occur.

This package is also freely available under license on an “as is” basis, and has presented many problems in the field. It is hard to service and offers low reliability.



RS/6000

## AIX Automounter

New

- Is exploited by PSSP 2.3
- Is part of the Network Support Facilities of AIX
- Has better serviceability
- Has better stability

POWERparallel Systems

ITSO Poughkeepsie Center

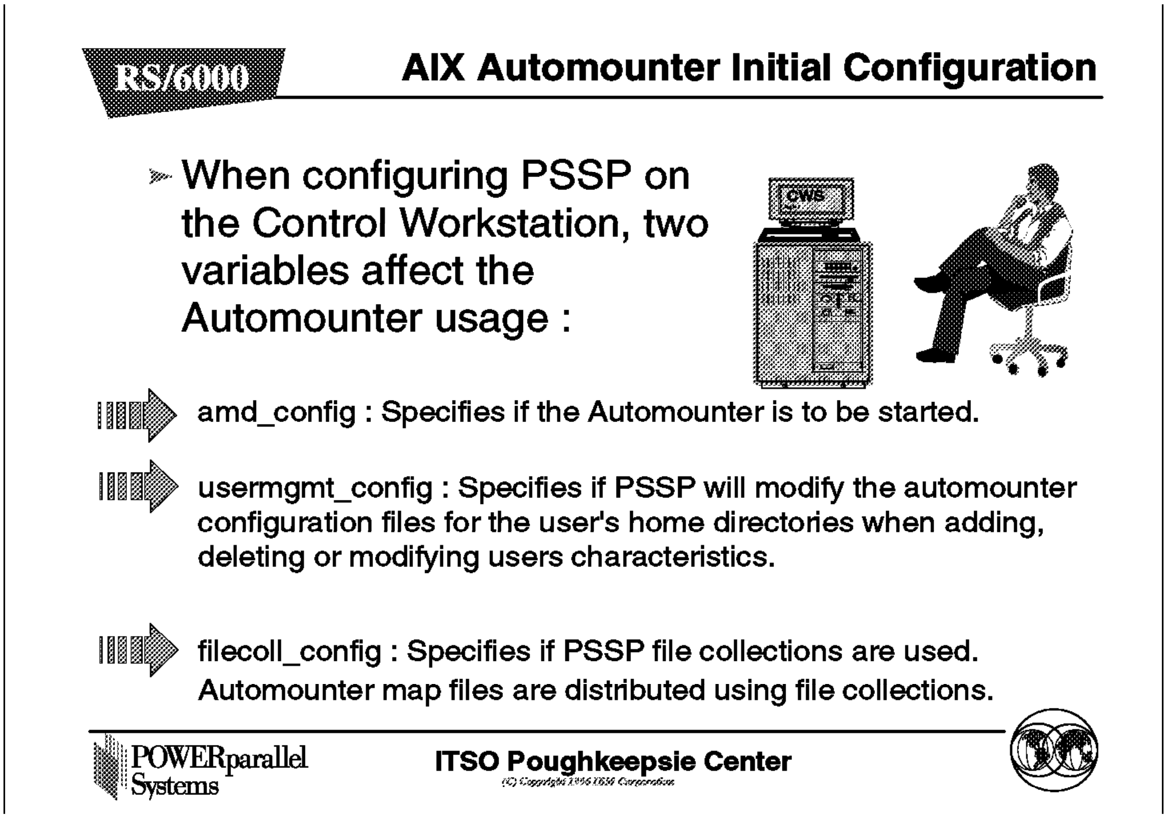
(C) Copyright 1996 IBM Corporation

In PSSP 2.3, the BSD Automounter is replaced by the AIX Automounter, which is the AIX version of the SunOS 4.x Automounter. This software is part of NFS in the Network Support Facilities of the AIX Base Operating System (BOS) Runtime, and is fully supported by AIX. We will refer to the new automounter as the AIX Automounter.

The main goal of this change is to provide better reliability and better serviceability. On a machine like the SP, the automounter plays a key role in making all file systems on which end user and application binaries and data are stored, available to other nodes. It is therefore important that the automounter be very reliable and run without interruptions.

The AIX Automounter has proved to be more stable than AMD. It is also fully supported by AIX, thus giving it better serviceability.

## 5.4 PSSP Configuration



RS/6000

### AIX Automounter Initial Configuration

- When configuring PSSP on the Control Workstation, two variables affect the Automounter usage :

- amd_config : Specifies if the Automounter is to be started.
- usermgmt_config : Specifies if PSSP will modify the automounter configuration files for the user's home directories when adding, deleting or modifying users characteristics.
- filecoll_config : Specifies if PSSP file collections are used. Automounter map files are distributed using file collections.

POWERparallel Systems

ITSO Poughkeepsie Center

During the PSSP installation procedure, the system administrator specifies if the PSSP software will control the automounter use. The site environment variable amd_config is set to either true (PSSP will start the automounter), or false (PSSP will not start the automunter).

**Note**

In PSSP 2.3, the meaning of amd_config is generalized to refer to both the AMD and AIX automounter, no matter which one is installed on the nodes.

This variable could be set by using the `smit enter_data` command on the Control Workstation and selecting from the following SMIT panel:

```
    Site Environment Configuration ==>
         Automounter Configuration {true|false}
```

Figure 2. Setting amd_config

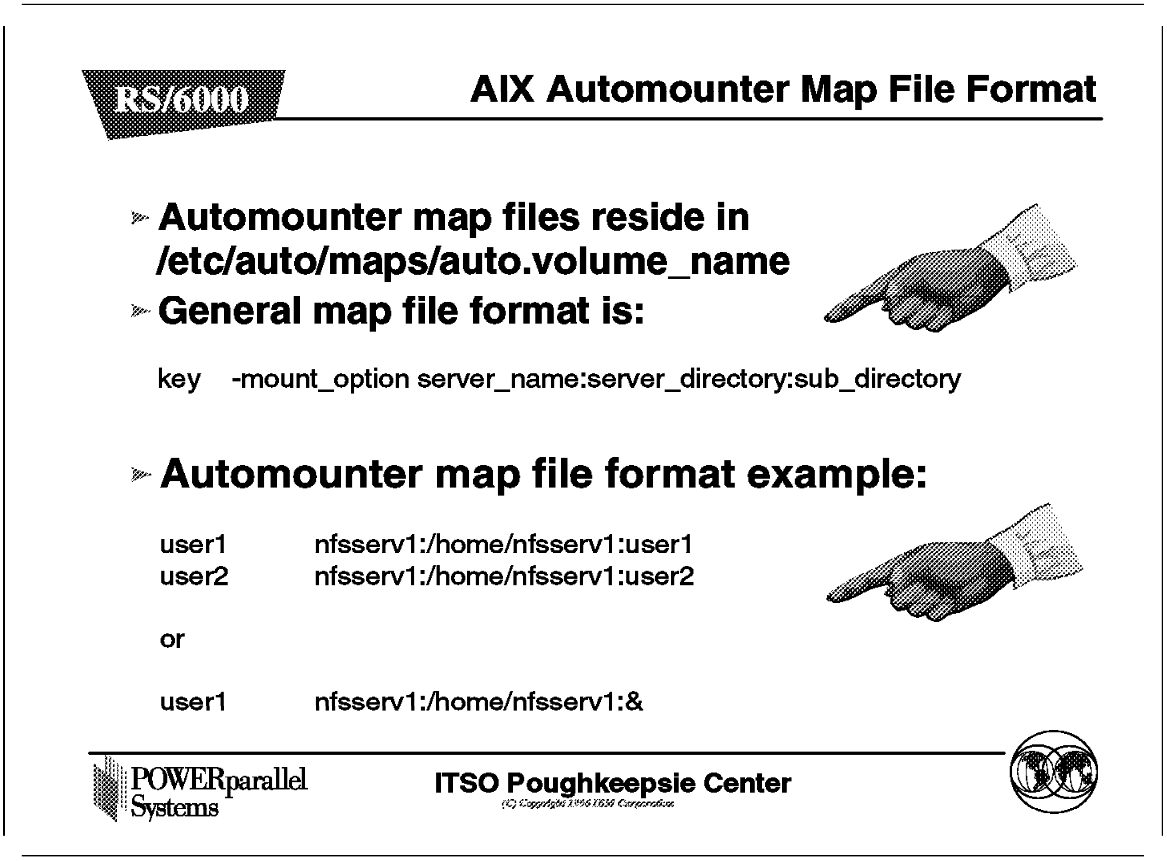
Select the field Automounter Configuration as desired. You could also use the spsitenv command.

Also, if the SP user management services have been configured, then the system administrator can use the smit spmkuser panel to add users to the system, smit sprmuser to delete users from the system, and smit spchuser to change the characteristics of a user.

PSSP then manages the /u file system, where the home directory of the SP users is mounted by the automounter. PSSP adds, deletes, or changes entries on the automounter map files for the /u file system, as you add, delete, or change users.

If the amd_config environment variable is set to false during system configuration, then PSSP will not configure or start the automounter daemon, and if the usermgmt_config environment variable is not set, it will not maintain the maps for the user′s home directories. This setting could be changed at a later time. You would then need to reboot the SP system in order for this change to take effect. Maps for already defined users would not be created, so that information would need to be added manually. This could be done by modifying the maps directly, or by using the mkamdent command.

## 5.5 AIX Automounter Master Map File



RS/6000

**AIX Automounter Map File Format**

- **Automounter map files reside in /etc/auto/maps/auto.volume_name**
- **General map file format is:**

key -mount_option server_name:server_directory:sub_directory

- **Automounter map file format example:**

```
user1     nfsserv1:/home/nfsserv1:user1
user2     nfsserv1:/home/nfsserv1:user2

or

user1     nfsserv1:/home/nfsserv1:&
```

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

The master map file for the automounter is located in /etc/auto.master. This master map file contains definitions for each file system that is to be controlled by the AIX Automounter daemon, the name of the map file containing the directory information, and optional default mount options that would apply for every directory on the map file specified in that line.

As an example, this is the /etc/auto.master file for the /u file system:

```
/u          /etc/auto/maps/auto.u  -rw,hard,rsize=4096,wsize=4096
```

In this file, we are telling the AIX Automounter to manage the /u file system. We specify that the subdirectory information is located in /etc/auto/maps/auto.u and show some NFS default mount options for all the subdirectories in that map.

This master file can also be accessed by means of NIS. By default, the SP invocation of the AIX Automounter disables the use of the auto.master NIS database. In order to be able to use auto.master by means of NIS, you need to do one of the following:

- Create the /etc/auto.master file on the client machine, as follows:

```
+auto.master
```

- You can change the way in which the AIX Automounter is started by not specifying the `-m -f /etc/auto.master` parameters.

Following is a complete example of how to use the /etc/auto.master with NIS:

1. Make sure NIS is not running on the server. For this example, we are going to use the Control Workstation (CWS) as an NIS server. In order to check that the CWS is not an NIS server, issue the `ypwhich` command. If you get the following message, then NIS is not configured in this machine:

```
(root) /> ypwhich
ypwhich: the domainname hasn′t been set on this machine.
```

   If you obtain the name of a domain, then you need to delete the NIS configuration on the CWS. Use the `smit yp` command to do so.

2. Edit the /var/yp/Makefile file and search the following line:

```
all: all.time passwd group hosts ethers networks rpc services protocols \
        netgroup bootparams aliases publickey netid netmasks all.remove
```

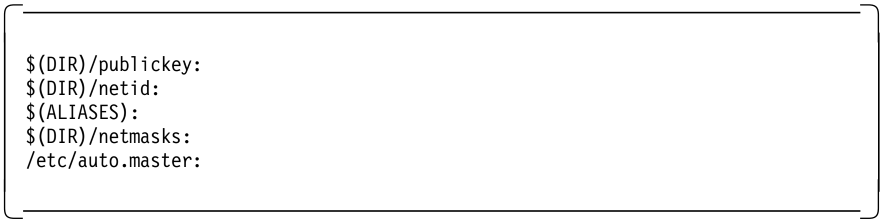
3. Add a line to the `all` stanza for the new auto.master map file as shown:

```
all: all.time passwd group hosts ethers networks rpc services protocols \
        netgroup bootparams aliases publickey netid netmasks all.remove \
        auto.master
```

4. Add the following stanza to the Makefile file:

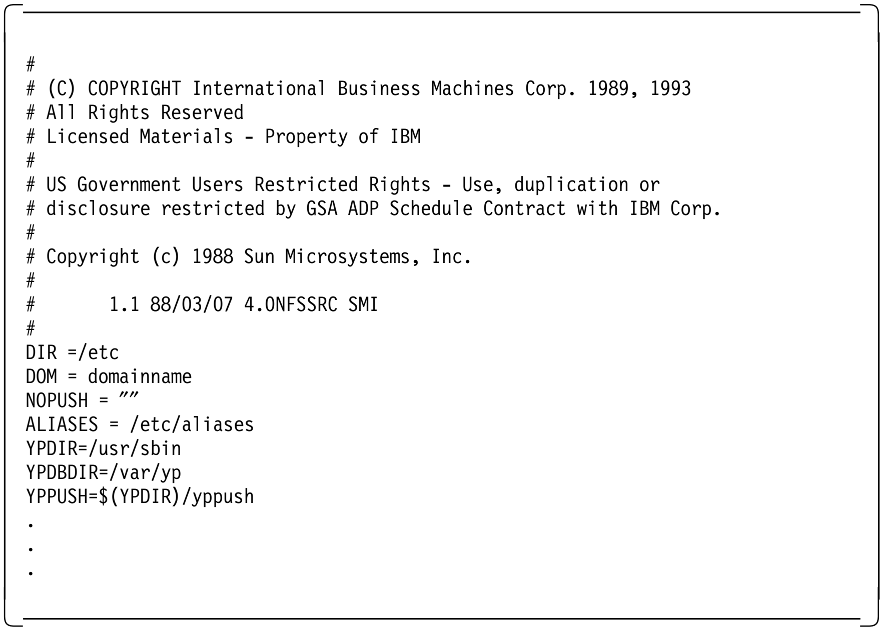
```
auto.master.time: /etc/auto.master
        -@if [ -f /etc/auto.master ]; then \
            $(MAKEDBM) /etc/auto.master $(YPDBDIR)/$(DOM)/auto.master;\
            chmod 600 $(YPDBDIR)/$(DOM)/auto.master.pag; \
            touch auto.master.time; \
            dspmsg cmdnfs.cat -s 56 39 ″updated auto.master\n″; \
            if [ ! $(NOPUSH) ]; then \
                   $(YPPUSH) auto.master; \
                     dspmsg cmdnfs.cat -s 56 40 ″pushed auto.master\n″;\
            else \
                    : ; \
            fi \
        else \
         dspmsg cmdnfs.cat -s 56 41 ″couldn′t find /etc/auto.master\n″;\
        fi
```

5. Add a stanza for /etc/auto.master at the bottom of the Makefile file, as follows:



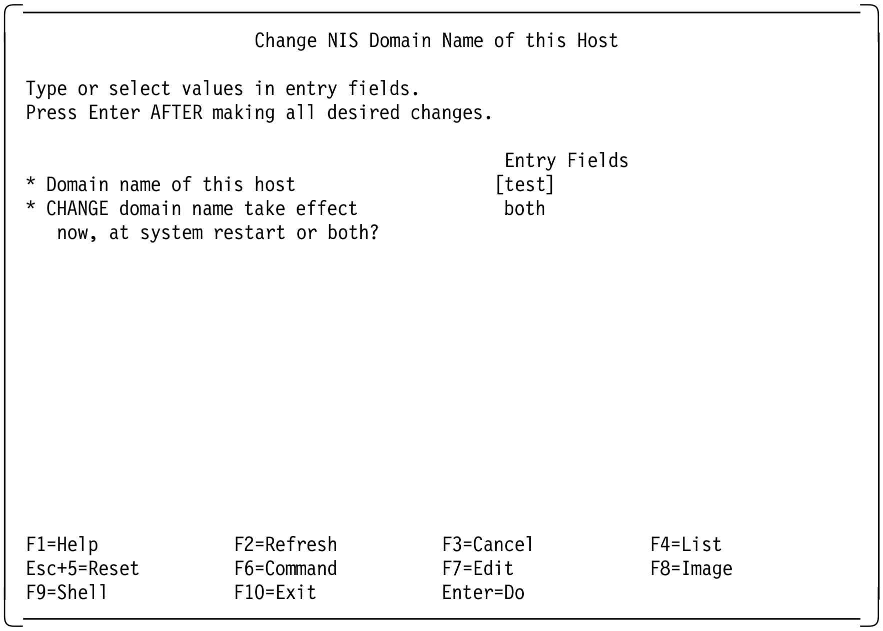
```
$(DIR)/publickey:
$(DIR)/netid:
$(ALIASES):
$(DIR)/netmasks:
/etc/auto.master:
```

You can also write the /etc/auto.master as $(DIR)/auto.master, since DIR is defined as follows in the Makefile file:



```
#
# (C) COPYRIGHT International Business Machines Corp. 1989, 1993
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# Copyright (c) 1988 Sun Microsystems, Inc.
#
#       1.1 88/03/07 4.0NFSSRC SMI
#
DIR =/etc
DOM = domainname
NOPUSH = ""
ALIASES = /etc/aliases
YPDIR=/usr/sbin
YPDBDIR=/var/yp
YPPUSH=$(YPDIR)/yppush
.
.
.
```

6. Now you have to set up the NIS Domain Name for the CWS. In order to do that, issue a smit yp command and select the Change NIS Domain Name of this Host option from the SMIT panel. The following screen is shown:



```
                      Change NIS Domain Name of this Host

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                              Entry Fields
* Domain name of this host                   [test]
* CHANGE domain name take effect              both
   now, at system restart or both?














F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

In this way, the NIS Domain Name is set to test.

7. Next you have to set up the CWS as an NIS Server. In order to do this, issue a smit mkmaster command. The following panel is shown:

```
                   Configure this Host as a NIS Master Server

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     Entry Fields
  HOSTS that will be slave servers                 []
* Can existing MAPS for the domain be overwritten?    yes
* EXIT on errors, when creating master server?        yes
* START the yppasswdd daemon?                         yes
* START the ypupdated daemon?                         yes
* START the ypbind daemon?                            yes
* START the master server now,                        both
   at system restart, or both?







F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

   Choose the options as shown in the example, and hit the Enter key.

8. Issue a cd /var/yp command, and build the auto.master NIS map by executing the following command:

```
make auto.master
```

9. To check if the configuration is working, issue a ypwhich command on the CWS. You should get output similar to the following:

```
sp2cw0/var/yp> ypwhich
loopback.msc.itso.ibm.com
sp2cw0/var/yp>
```

   Also issue a ypcat auto.master command. You should get output similar to the following:

```
sp2cw0/var/yp> ypcat auto.master
/etc/auto/maps/auto.net -soft,intr,retry=3
/etc/auto/maps/auto.u -soft,intr,retry=3
sp2cw0/var/yp>
```

   This output should match the contents of the /etc/auto.master map file. Remove the stanzas that begin with the # character from the /etc/auto.master file.

10. Now is the time to configure one of the nodes as an NIS client. To do this, first stop supper from updating the /etc/auto.master file. You can temporarily do this by issuing crontab -e as root and commenting out the line that does the supper update sup.admin user.admin node.root. After this

procedure completes, uncomment this line. If you want to keep using the NIS auto.master file, then you should erase the /etc/auto.master file from the user.admin file collection.

11. Configure the NIS client. To do this, issue the smit mkclient command. The following panel is shown:

```
                      Configure this Host as a NIS Client

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     Entry Fields
* START the NIS client now,                         both
   at system restart, or both?
```

    Press the Enter key.

12. Check that the configuration is working. Issue the ypwhich command. You should get output similar to the following:

```
sp2n01/> ypwhich
sp2en0.msc.itso.ibm.com
sp2n01/>
```

13. Edit the /etc/auto.master file on the node, and leave it as follows:

```
sp2n01/> cat /etc/auto.master
+auto.master
sp2n01/>
```

14. Check the NIS auto.master map. Issue the ypcat auto.master command. The listing of the /etc/auto.master file of the NIS server should appear, as follows:

```
sp2n01/> ypcat auto.master
/etc/auto/maps/auto.net -soft,intr,retry=3
/etc/auto/maps/auto.u -soft,intr,retry=3
sp2n01/>
```

15. Stop the AIX Automounter daemon by finding its PID and issuing kill -15 as follows:

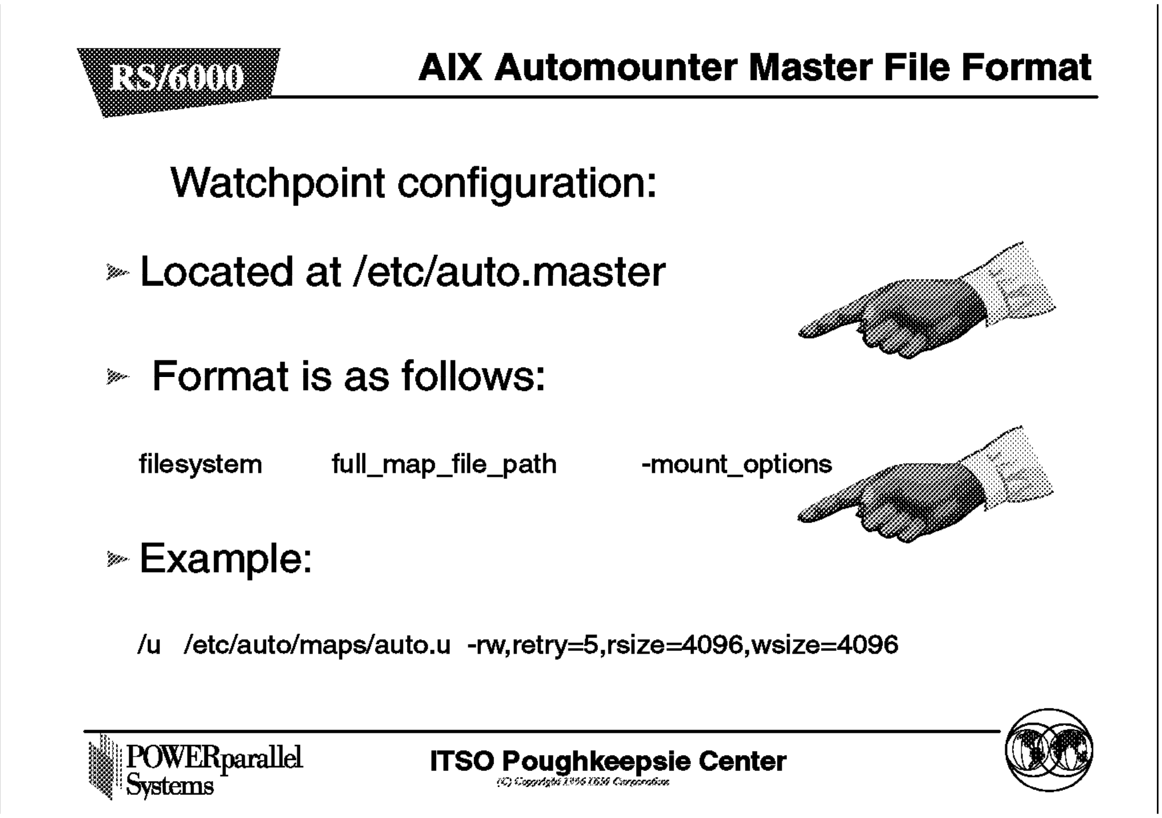
```
sp2n01/> ps auxw | grep -i automount
root     19308  0.0  0.0  236  300      - A    19:26:35  0:00 /usr/sbin/a
utomount -f /etc/auto.master -m -D HOST=sp2n01
root     21754  0.0  0.0  120  152  pts/1 A    17:51:48  0:00 grep -i aut
sp2n01/> kill -15 21754
```

16. Start the AIX Automounter daemon using the /etc/auto/startauto command as follows:

```
/etc/auto/startauto
```

Now you should be able to use the AIX Automounter-managed directories as usual. Note that only the /etc/auto.master file is accessed by NIS. You can use this procedure to make the individual map files of every directory managed by the AIX Automounter available by means of NIS.

### 5.5.1 AIX Automounter Map File



RS/6000

**AIX Automounter Master File Format**

**Watchpoint configuration:**

- Located at /etc/auto.master
- Format is as follows:

```
filesystem    full_map_file_path    -mount_options
```

- Example:

```
/u   /etc/auto/maps/auto.u  -rw,retry=5,rsize=4096,wsize=4096
```

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

In this section we introduce the configuration files for the AIX Automounter, and some examples of configuration files.

The AIX Automounter reads automount map files to determine which directories to handle under a certain mount point. There is usually one map file for every mount point to be controlled. These map files are kept in the /etc/auto/maps directory, while the list of all map files to be used is stored in the /etc/auto.master file. PSSP stores the map files in the /etc/auto/maps directory by convention. The administrator can store the map files in some other directory, as long as the full path is specified. The name of the files can be anything, but the automounter is easier to administer if basic conventions are followed, such as naming each map as auto.*filesystem*, where *filesystem* is the name of the file system that this file describes.

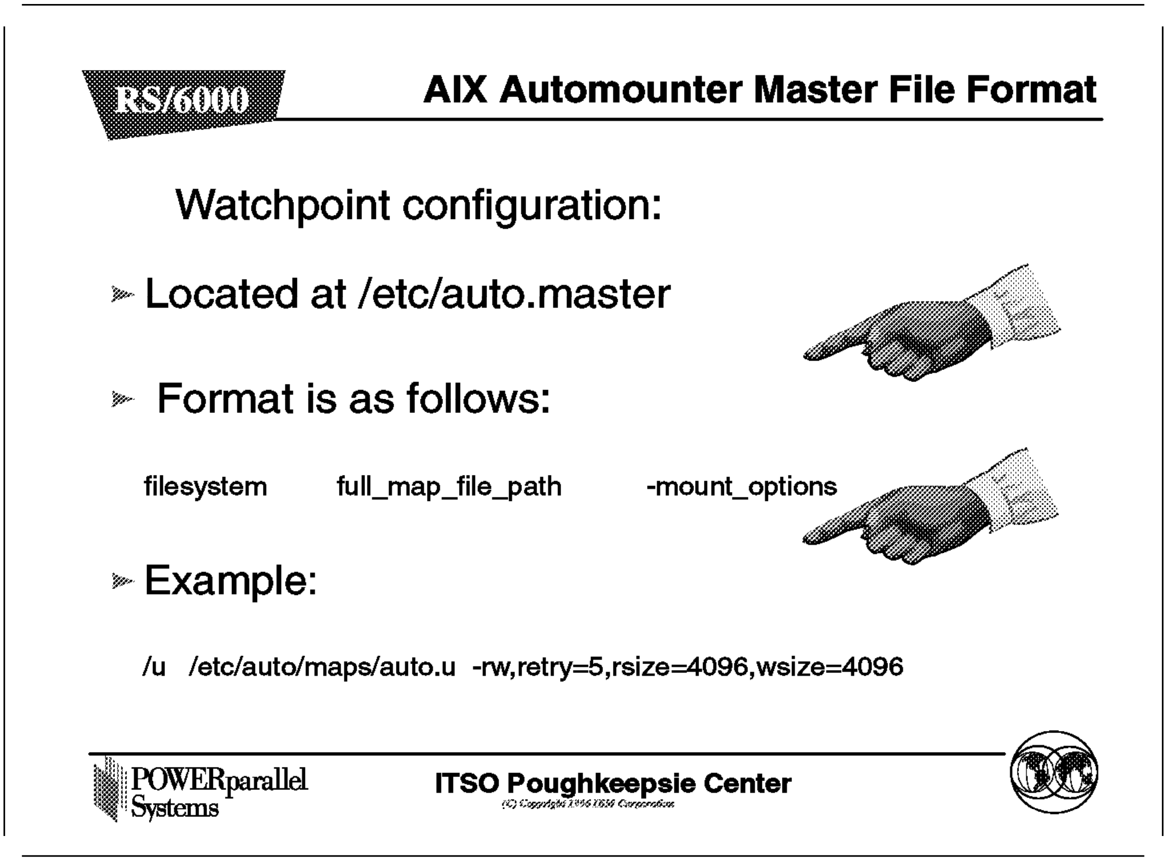
If both amd_config and usermgmt_config are set to true, then the /u file system is automatically controlled by the automounter. PSSP creates an automount map file for the /u file system in the /etc/auto/maps/auto.u file. This file has entries for every user defined by using the `smit spmkuser` command. The format is as follows:

```
key    -mount_options server_name:mount_directory:sub_directory
```

Here, key is the directory that we want to mount within the file system; mount_options is an optional field that is used for the mount operation and that overrides the specifications that might exist for this map in the /etc/auto.master file; server_name specifies which NFS server exports the file system that we want to mount; mount_directory is the directory itself that we want to mount from the NFS server, and sub_directory is an optional path that can be specified under mount_directory.

If server_name matches the local machine name, then the automounter daemon simply creates a symbolic link to the target directory instead of trying to do the NFS mount operation. ************************************************************

### 5.5.2 AIX Automounter Map File Examples


RS/6000

AIX Automounter Master File Format

Watchpoint configuration:

- Located at /etc/auto.master
- Format is as follows:

filesystem        full_map_file_path        -mount_options

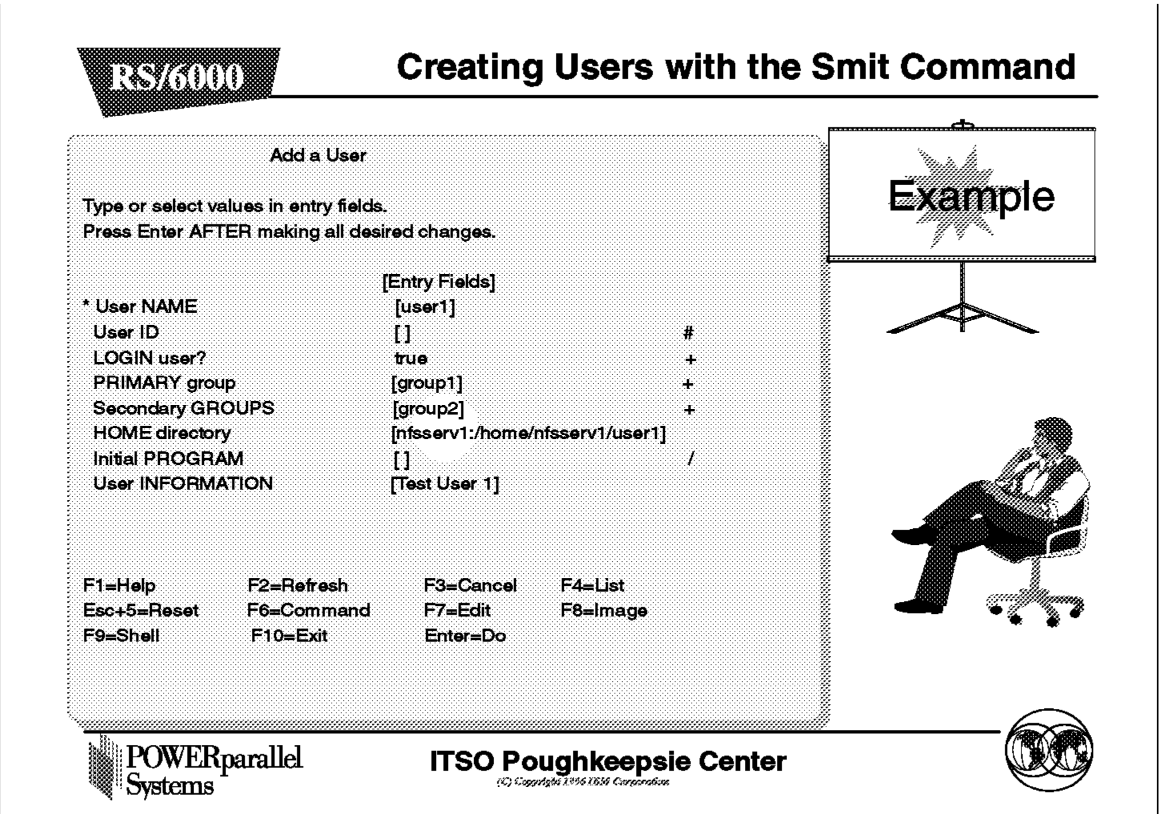
- Example:

/u   /etc/auto/maps/auto.u  -rw,retry=5,rsize=4096,wsize=4096

POWERparallel Systems
ITSO Poughkeepsie Center
(C) Copyright 1996 IBM Corporation

In this section we offer examples of how to use the automounter to manage two different file systems, /u and /net, and discuss the steps for creating your own map files.

## 5.5.3 Creating Users



Creating Users with the Smit Command

```
                    Add a User

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                              [Entry Fields]
* User NAME                    [user1]
  User ID                      []                         #
  LOGIN user?                  true                       +
  PRIMARY group                [group1]                   +
  Secondary GROUPS             [group2]                   +
  HOME directory               [nfsserv1:/home/nfsserv1/user1]
  Initial PROGRAM              []                         /
  User INFORMATION             [Test User 1]



F1=Help         F2=Refresh      F3=Cancel      F4=List
Esc+5=Reset     F6=Command      F7=Edit        F8=Image
F9=Shell        F10=Exit        Enter=Do
```

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

An SP user might be created as shown above.

The default home directory for SP users is /home/hostname/user, where hostname is the short name of the system where that directory resides.

The /etc/auto/maps/auto.u map file for user user1 is:

```
user1          nfsserv1:/home/nfsserv1:user1
```

This could also be written as:

```
user1          nfsserv1:/home/nfsserv1:&
```

In this stanza, the & symbol is replaced by the key value, which in this case is user1.

In this example, the following steps occur:

- If user1 successfully logs onto the machine nfsserv1, then the automounter will automatically create a symbolic link to the local directory, as shown:

  ```
  /u/user1  ->   /home/nfsserv1/user1
  ```

  In this way, the user uses the /home/nfsserv1/user1 directory when accessing the /u/user1 directory.

- When user1 successfully logs onto a machine other than nfsserv1, and tries to access its home directory, the automounter needs to do an NFS mount of the remote directory. The steps for this are the following:

  1. The automounter does the NFS mount of the remote directory if that directory is not already mounted, as follows:

     ```
     mount nfsserv1:/home/nfsserv1 /tmp_mnt/u/user1
     ```

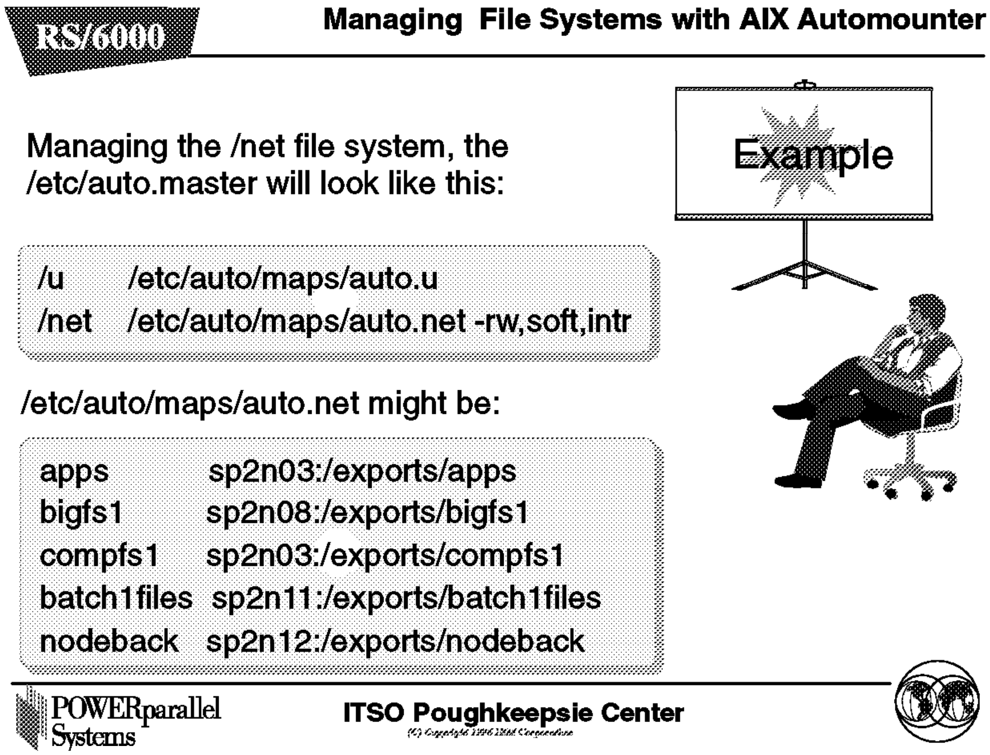
     The directory /tmp_mnt is the local staging or temporary area where all mounts done by the automounter actually take place.

  2. Then the automounter creates a symbolic link to the local directory that was just mounted, as follows:

     ```
     /u/user1  ->   /tmp_mnt/u/user1/user1
     ```

     At this point, the user accesses the same /u/user1, but this time, the actual access will be done over the NFS-mounted /tmp_mnt/u/user1/user1 directory.

## 5.5.4 Managing Other File Systems with AIX Automounter



RS/6000

Managing File Systems with AIX Automounter

Managing the /net file system, the /etc/auto.master will look like this:

```
/u      /etc/auto/maps/auto.u
/net    /etc/auto/maps/auto.net -rw,soft,intr
```

/etc/auto/maps/auto.net might be:

```
apps         sp2n03:/exports/apps
bigfs1       sp2n08:/exports/bigfs1
compfs1      sp2n03:/exports/compfs1
batch1files  sp2n11:/exports/batch1files
nodeback     sp2n12:/exports/nodeback
```

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

As previously mentioned, we can use the AIX Automounter to manage directories other than /u. In this example, we would like the automounter to manage the /net file system, which might be used as the mount point for other general purpose file systems that should be available on all systems.

First, add entries to the /etc/auto.master file that define the /net file system to the AIX Automounter, and the map file for that file system.

The /etc/auto.master file will look like this:

```
/u      /etc/auto/maps/auto.u   -rw,hard,rsize=4096,wsize=4096
/net    /etc/auto/maps/auto.net -rw,soft,intr
```

Then, add definitions for the subdirectories that are to be mounted under the /net file system to the /etc/auto/maps/auto.net file, which might look like the following:

```
apps        sp2n03:/exports/apps
bigfs1      sp2n08:/exports/bigfs1
compfs1     sp2n03:/exports/compfs1
batch1files sp2n11:/exports/batch1files
nodebackup  sp2n12:/exports/nodebackup
```

At this point, refresh the automounter daemon in order to make it read the new /etc/auto.master file. To do that, stop the automounter daemon and start it again.

To stop the daemon, do the following:

1. Make sure the automounter has no subdirectory under its control mounted. To do that, do the following:

   - Issue a mount command, as follows:

```
sp2cw0/> mount
node   mounted        mounted over    vfs    date         options
------ -------------  --------------- ------ ------------ --------------
       /dev/hd4       /               jfs    Apr 29 20:01 rw,log=/dev...
       /dev/hd2       /usr            jfs    Apr 29 20:01 rw,log=/dev...
       /dev/hd9var    /var            jfs    Apr 29 20:01 rw,log=/dev...
       /dev/hd3       /tmp            jfs    Apr 29 20:01 rw,log=/dev...
       /dev/hd1       /home           jfs    Apr 29 20:03 rw,log=/dev...
       /dev/lv02      /spdata         jfs    Apr 29 20:03 rw,log=/dev...
       /dev/lv03      /tftpboot       jfs    Apr 29 20:03 rw,log=/dev...
sp2cw0 (pid9276@/u)   /u              nfs    Apr 29 20:04 ro,ignore
sp2cw0 (pid9276@/net) /net            nfs    Apr 29 20:04 ro,ignore
```

   Check if there is a mount under the /tmp_mnt staging area. These mounts look as follows:

```
sp2cw0/> mount
node   mounted        mounted over    vfs    date         options
------ -------------  --------------- ------ ------------ --------------
       /dev/hd4       /               jfs    Apr 29 20:01 rw,log=/dev...
       /dev/hd2       /usr            jfs    Apr 29 20:01 rw,log=/dev...
       /dev/hd9var    /var            jfs    Apr 29 20:01 rw,log=/dev...
       /dev/hd3       /tmp            jfs    Apr 29 20:01 rw,log=/dev...
       /dev/hd1       /home           jfs    Apr 29 20:03 rw,log=/dev...
       /dev/lv02      /spdata         jfs    Apr 29 20:03 rw,log=/dev...
       /dev/lv03      /tftpboot       jfs    Apr 29 20:03 rw,log=/dev...
sp2cw0 (pid9276@/u)   /u              nfs    Apr 29 20:04 ro,ignore
sp2cw0 (pid9276@/net) /net            nfs    Apr 29 20:04 ro,ignore
sp2n02 /home/sp2n02   /tmp_mnt/u/test1 nfs3  May 03 12:08 soft,intr
```

   Here, /tmp_mnt/u/test1 is an example of a file system mounted on the AIX Automounter staging area.

   - If there is no mount under the /tmp_mnt staging area that the automounter is managing, you can go to the next step.

   - If there are file systems mounted under the /tmp_mnt staging area, stop the processes that are using those file systems and unmount them.

> **Note**
>
> If any active mounts exist when the automount daemon is stopped, these will not be removed the next time the automounter is started. You will need to explicitly unmount those file systems, or wait until the system is rebooted.

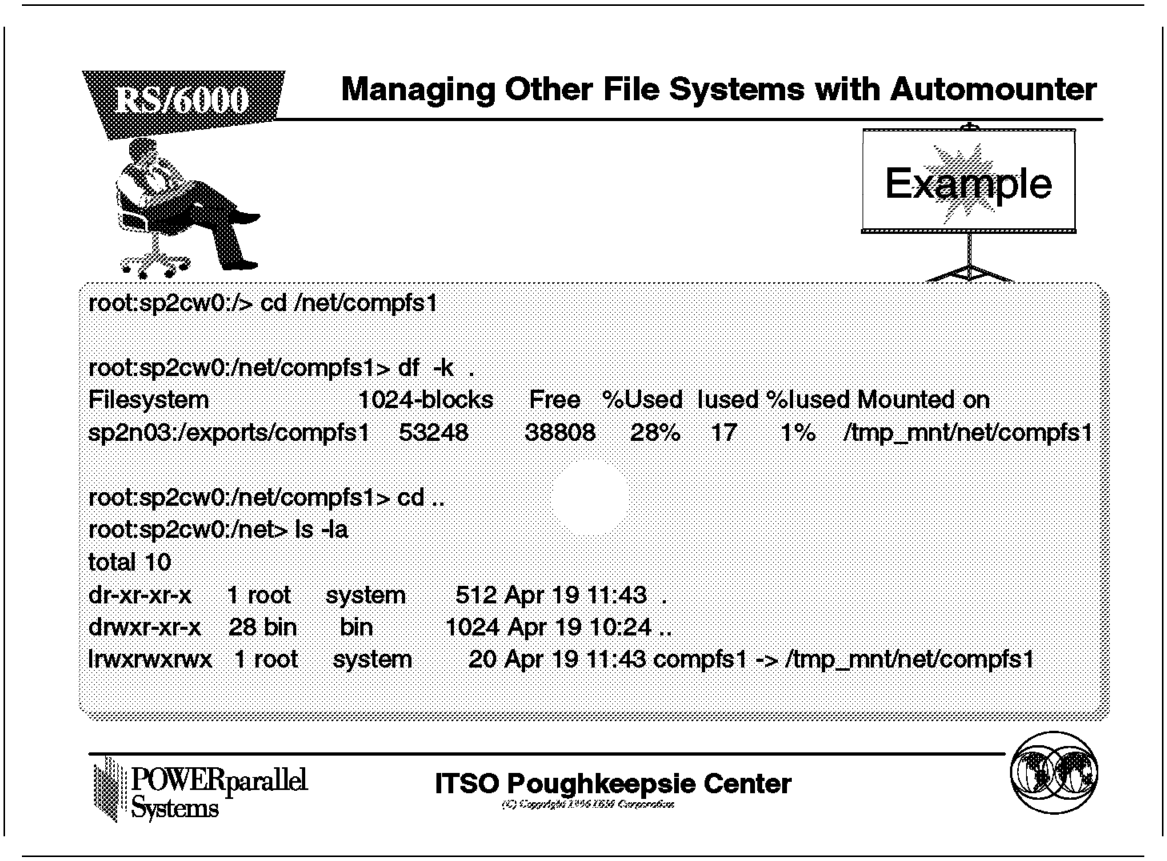
2. Find the PID of the automounter daemon.
3. Stop the automounter by sending a TERM signal.
4. Start the automounter again.

This procedure would look like this:

```
(root)> ps auxwww | grep automount
root      9276  0.0  0.0  296  204      - A      Apr 19  0:00 /usr/sbin/a
utomount -f /etc/auto.master -m -DHOST=sp2cw0
(root)> kill -15 9276
(root)> /etc/auto/startauto
```

You should note that the PID of the AIX Automounter can be also taken from the output of the `mount` command.

A session with one of these file systems is shown in the next figure.



RS/6000

## Managing Other File Systems with Automounter

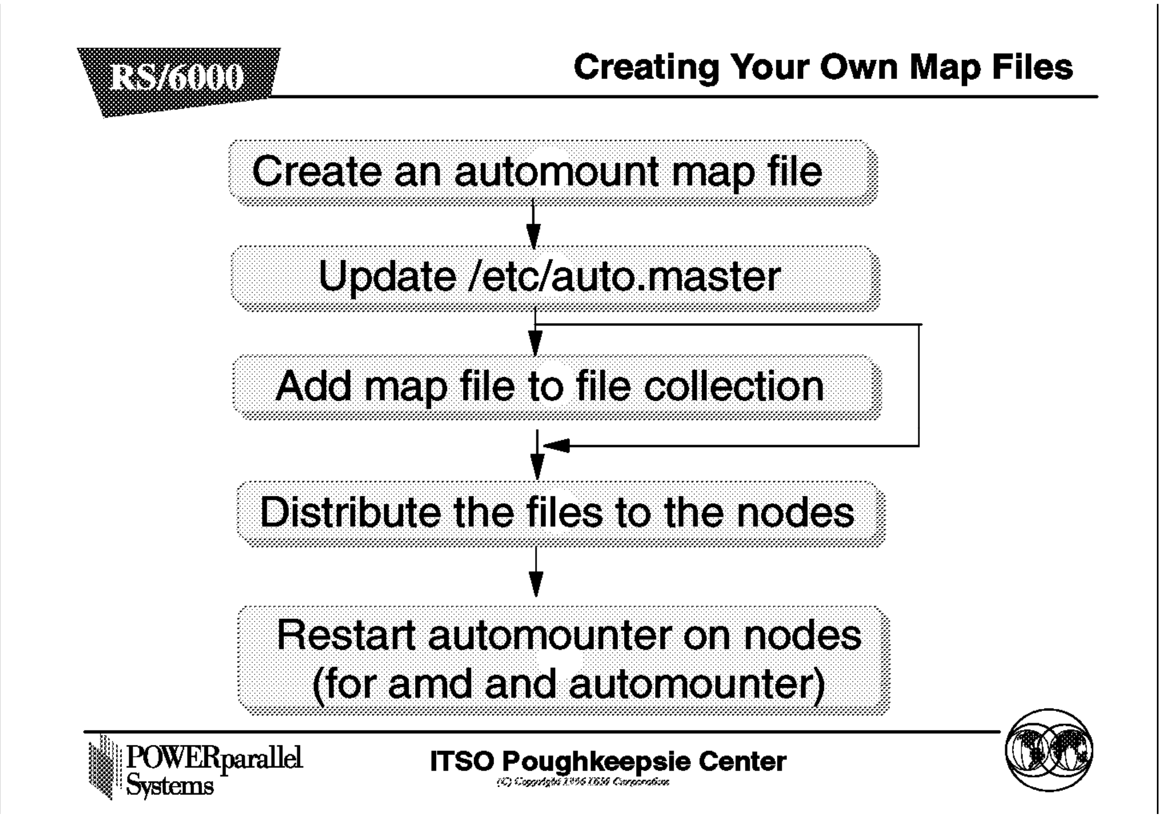
```
root:sp2cw0:/> cd /net/compfs1

root:sp2cw0:/net/compfs1> df  -k  .
Filesystem                 1024-blocks    Free   %Used  Iused %Iused Mounted on
sp2n03:/exports/compfs1   53248       38808    28%    17      1%    /tmp_mnt/net/compfs1

root:sp2cw0:/net/compfs1> cd ..
root:sp2cw0:/net> ls -la
total 10
dr-xr-xr-x    1 root     system       512 Apr 19 11:43  .
drwxr-xr-x   28 bin      bin         1024 Apr 19 10:24 ..
lrwxrwxrwx   1 root     system        20 Apr 19 11:43 compfs1 -> /tmp_mnt/net/compfs1
```

POWERparallel Systems

ITSO Poughkeepsie Center

The above figure illustrates an example of how to manage other file systems with automounter.

### 5.5.5 Creating Your Own Map Files



Creating Your Own Map Files

Create an automount map file

Update /etc/auto.master

Add map file to file collection

Distribute the files to the nodes

Restart automounter on nodes
(for amd and automounter)

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

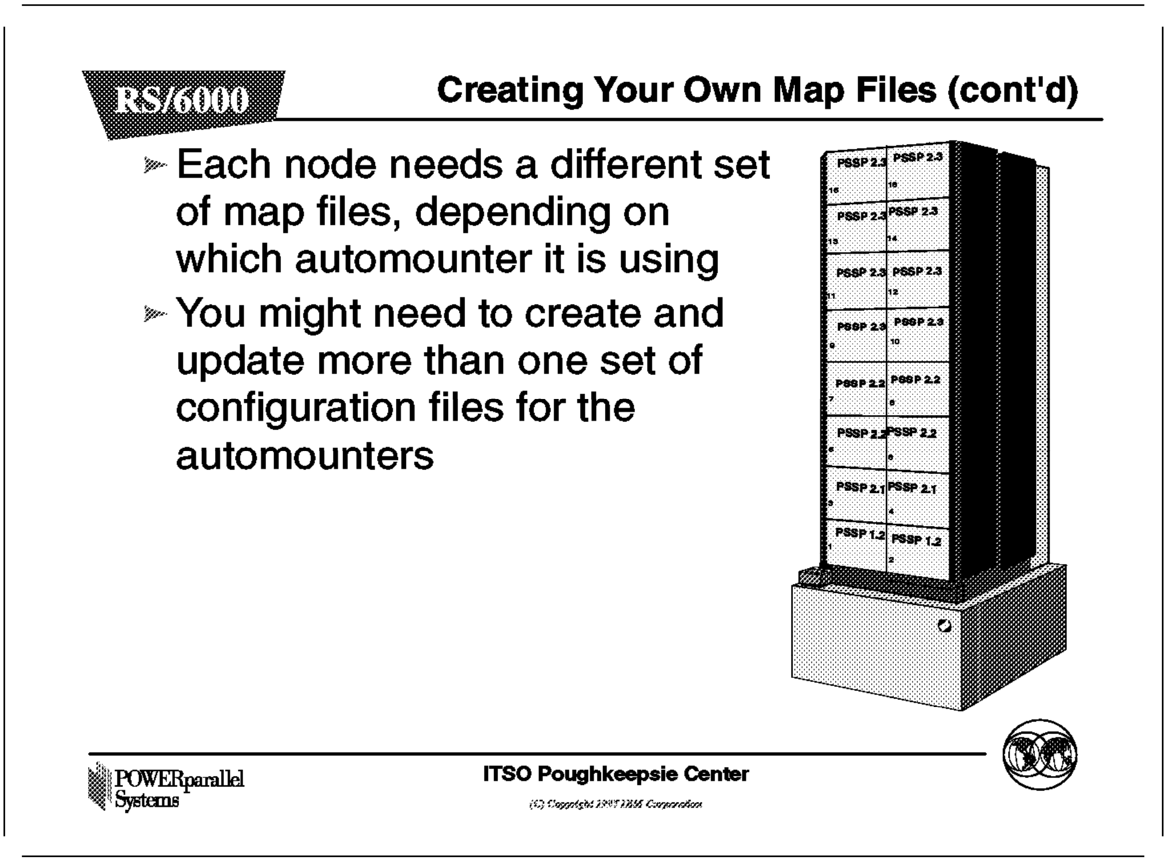
After having seen the previous two examples, let us now discuss the steps for creating your own map files:

1. Create an automount map file at /etc/auto/maps/auto.fs.
2. Update the /etc/auto.master file with the data about the fs.
3. Add the map file to the file collection mechanism, in order to have the files distributed over the nodes.
4. Refresh the automounter on the nodes.

**Note**

Do not try to kill the automounter daemon using `kill -9`. That might cause the automounter-controlled file systems to hang. Use the `kill -15` command instead.

If you have more than one type of automounter in your installation (say AMD and AIX Automounter), you need to create and update the configuration files for all of the automounters. This may be the case if your installation has multiple levels of PSSP.



RS/6000

## Creating Your Own Map Files (cont'd)

- Each node needs a different set of map files, depending on which automounter it is using
- You might need to create and update more than one set of configuration files for the automounters

PSSP 2.3
PSSP 2.3
PSSP 2.3
PSSP 2.3
PSSP 2.3
PSSP 2.3
PSSP 2.3
PSSP 2.3
PSSP 2.2
PSSP 2.2
PSSP 2.2
PSSP 2.2
PSSP 2.1
PSSP 2.1
PSSP 1.2
PSSP 1.2

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The AIX Automounter can be used to access file systems other than NFS, such as AFS or GPFS. The AFS or GPFS file systems already appear as local file systems on each node, so the map file would look like this:

```
user1      $myhostname:/afs/pok.ibm.com/usr2:&
```

In this example, we would like the $myhostname variable to be replaced with the name of the local host where that map resides, so when the key matches user1, the automounter links to the target directory. To do this, it provides a facility that calls predefined variables; so you can start up the automounter like this:

```
/usr/sbin/automounter -D´HOST=`uname -n`´
```

Using this example, the $HOST variable is initialized with the host name of the node where the automounter is running. Then we can specify a map file as follows:

```
user1      $HOST:/afs/pok.ibm.com/usr2:&
```

In this case, if some process tries to access the /u/user1 file system, then the automounter only makes a symbolic link to the AFS file system, as follows:

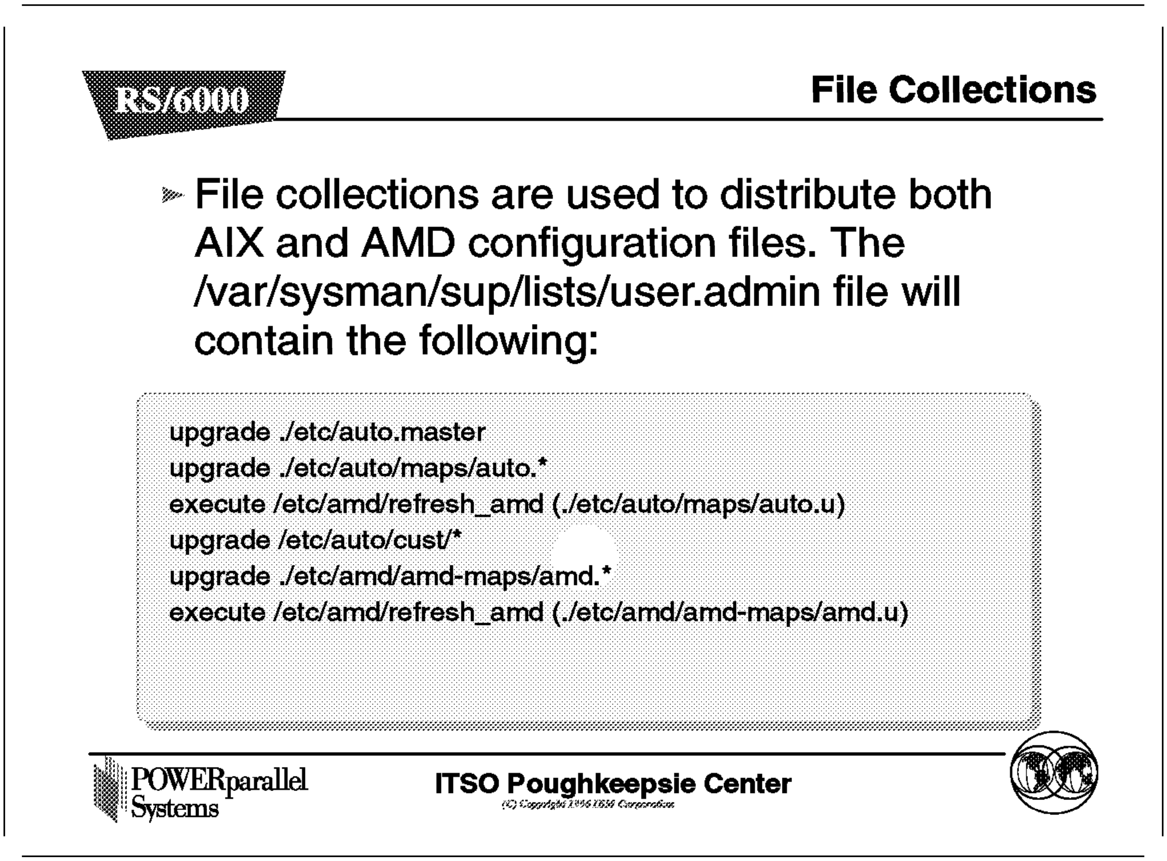
```
/u/user1   ->   /afs/pok.ibm.com/usr2/user1
```

### 5.5.6 Distribution of AIX Automounter Files

If the SP is configured to manage file collections (the site environment variable filecoll_config is “true”), then the AIX automounter map files will be automatically distributed to all the nodes by means of *supper*. The AIX automounter map files are part of the user.admin file collection. If you do customize any of the SP automounter functions, you need to edit the /var/sysman/sup/lists/user.admin file to distribute your own files or remove the ones distributed by default.

**Note**

In the /var/sysman/sup/lists/user.admin file, there is a comment that SP automounter configuration has been added. Do not remove this comment, otherwise the next time the machine gets rebooted, the SP configuration will add the automount entries back again.



RS/6000

## File Collections

- File collections are used to distribute both AIX and AMD configuration files. The /var/sysman/sup/lists/user.admin file will contain the following:

```
upgrade ./etc/auto.master
upgrade ./etc/auto/maps/auto.*
execute /etc/amd/refresh_amd (./etc/auto/maps/auto.u)
upgrade /etc/auto/cust/*
upgrade ./etc/amd/amd-maps/amd.*
execute /etc/amd/refresh_amd (./etc/amd/amd-maps/amd.u)
```

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

On the nodes, the /var/sysman/sup/lists/user.admin file will look like this:

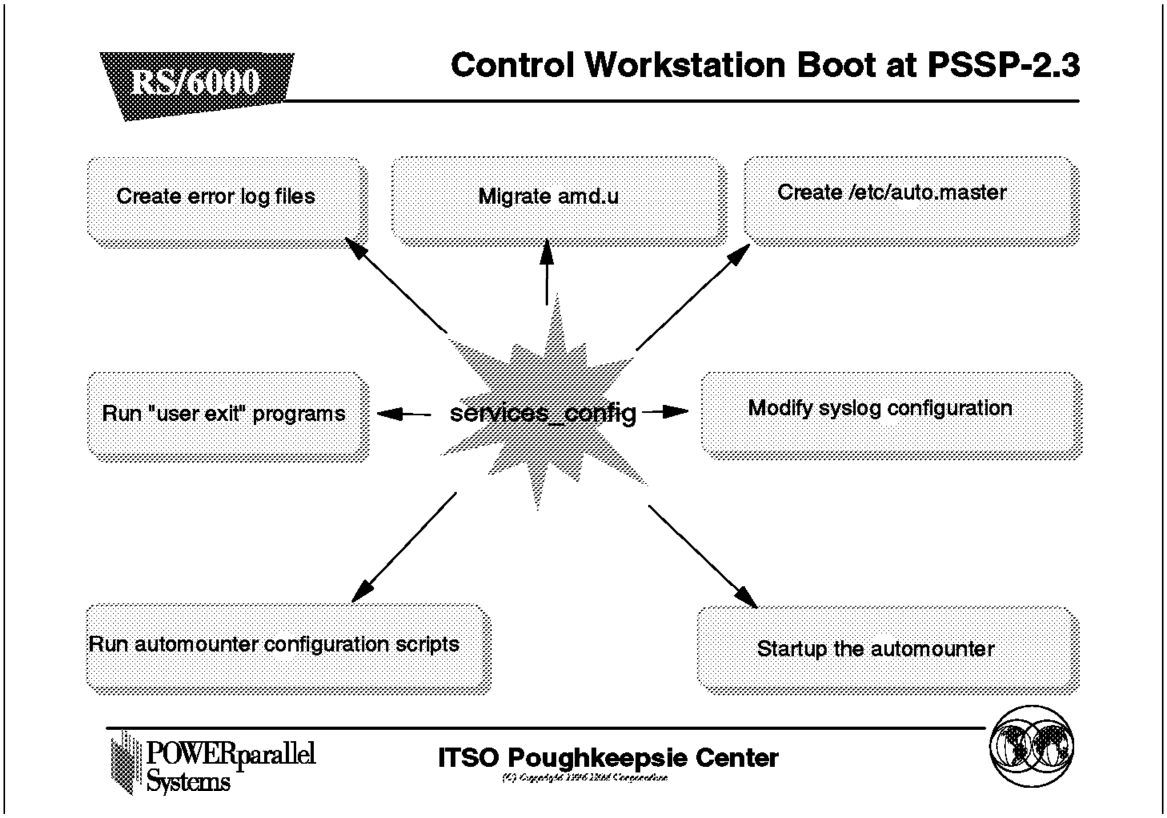
```
upgrade ./etc/auto.master
upgrade ./etc/auto/maps/auto.*
execute  /etc/amd/refresh_amd (./etc/auto/maps/auto.u)
upgrade ./etc/auto/cust/*
upgrade ./etc/amd/amd-maps/amd.*
execute  /etc/amd/refresh_amd (./etc/amd/amd-maps/amd.u)
```

You should add the entries that refer to your files to this file. If the file /var/sysman/sup/user.admin/scan exists on the control workstation, or on any boot/install server, then on you should run the following command on each of those machines:

```
(root)> /var/sysman/supper scan user.admin
```

Then you can do a supper update user.admin or wait for the system to do the update. You can also create other file collections to distribute your own maps; but always check for the scan file.

## 5.6 Migration of Existing AMD Maps



RS/6000

Control Workstation Boot at PSSP-2.3

Create error log files

Migrate amd.u

Create /etc/auto.master

Run "user exit" programs

services_config

Modify syslog configuration

Run automounter configuration scripts

Startup the automounter

POWERparallel Systems

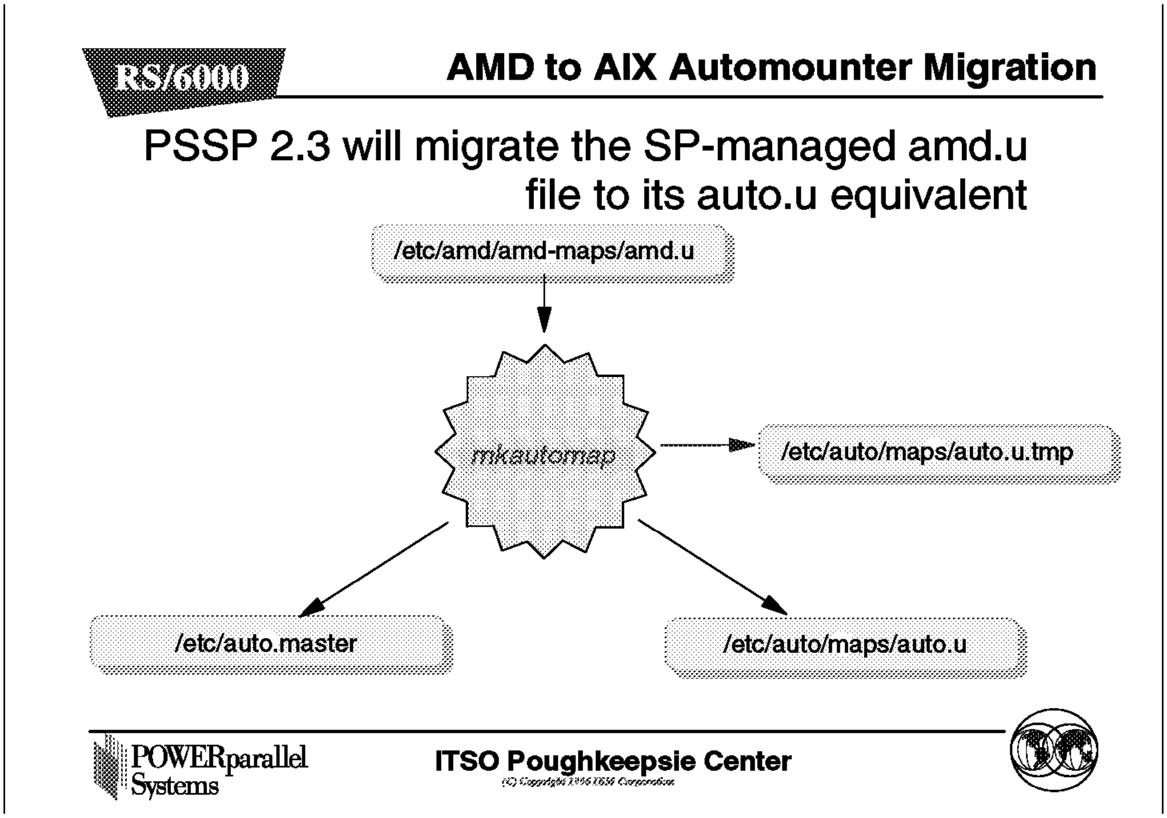
ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

When the Control Workstation first boots up at PSSP 2.3 after being migrated from a prior level of PSSP, rc.sp calls services_config and migrates the /etc/amd/amd-maps/amd.u file to /etc/auto/maps/auto.u.

The user command for this operation is mkautomap. It migrates the amd.u file as managed by prior versions of PSSP. If you have introduced changes to this file, or if you attempt to migrate your own files with this command, errors might occur. Error messages will be written to both standard output and /var/adm/SPlogs/auto/auto.log.

If errors occur, mkautomap leaves a temporary file, /etc/auto/maps/auto.u.tmp, that contains the output of the migration process for all the entries that have been successfully migrated. You can check this file to see if errors occurred while mkautomap was running, and try to correct them. If mkautomap does not succeed, you must migrate the /etc/amd/amd-maps/amd.u file yourself.



RS/6000

## AMD to AIX Automounter Migration

PSSP 2.3 will migrate the SP-managed amd.u file to its auto.u equivalent

/etc/amd/amd-maps/amd.u

mkautomap

/etc/auto/maps/auto.u.tmp

/etc/auto.master

/etc/auto/maps/auto.u

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

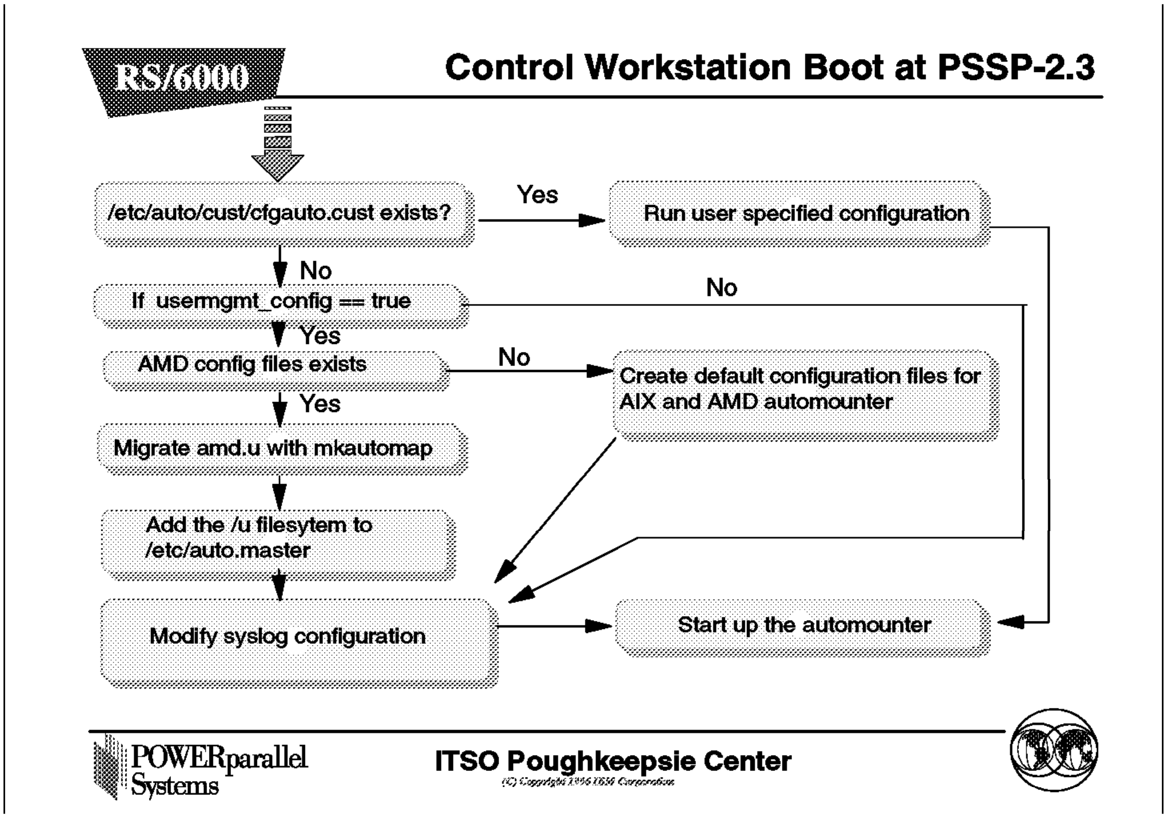
Errors will occur when the source AMD file is using some facility that is not supported on the AIX Automounter.

When there is an error in the migration process, the services_config program continues its execution, but the AIX Automounter daemon is not started.

The tasks that services_config performs for the migration of the AMD map files to their AIX Automounter equivalents include:

- Creating the error log files that the AIX automounter will use
- Creating the /etc/auto.master file
- Migrating the amd.u file to the auto.u file
- Modifying the syslog configuration
- Starting up the automounter
- Running “User Exit” programs (more about this later)

The flow chart of the services_config program when configuring the automounter is shown in the next figure.



## Control Workstation Boot at PSSP-2.3

RS/6000

/etc/auto/cust/cfgauto.cust exists?

Yes → Run user specified configuration

No

If usermgmt_config == true

No

Yes

AMD config files exists

No → Create default configuration files for AIX and AMD automounter

Yes

Migrate amd.u with mkautomap

Add the /u filesytem to /etc/auto.master

Modify syslog configuration → Start up the automounter

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

The flow begins at the top left of the chart with a test to see whether the /etc/auto/cust/cfgauto.cust file exists. This file is called a User Exit. If it exists, it means that the local system administrator has done his/her own version of the configuration step of the automounter, which is the configuration process that would be executed. If the file does not exist, the normal configuration process takes place. For this example, we assume that both amd_config and usermgmt_config are set to true.

In the next step, it checks if there are some AMD configuration files. If there are, then we are migrating from some previous version of PSSP and the /etc/amd/amd-maps/amd.u file is migrated to the /etc/auto/maps/auto.u equivalent by using the mkautomap command. This command will also create the /etc/auto.master file with the description of the /u file system.

Next it modifies the syslog configuration so that the errors from the automounter daemon are directed to the /var/adm/SPlogs/SPdaemon.log file.

Then the automounter is started. The script for the startup of the AIX Automounter can also be replaced by a User Exit. If that is the case, the user-provided mechanism for starting the automounter runs.

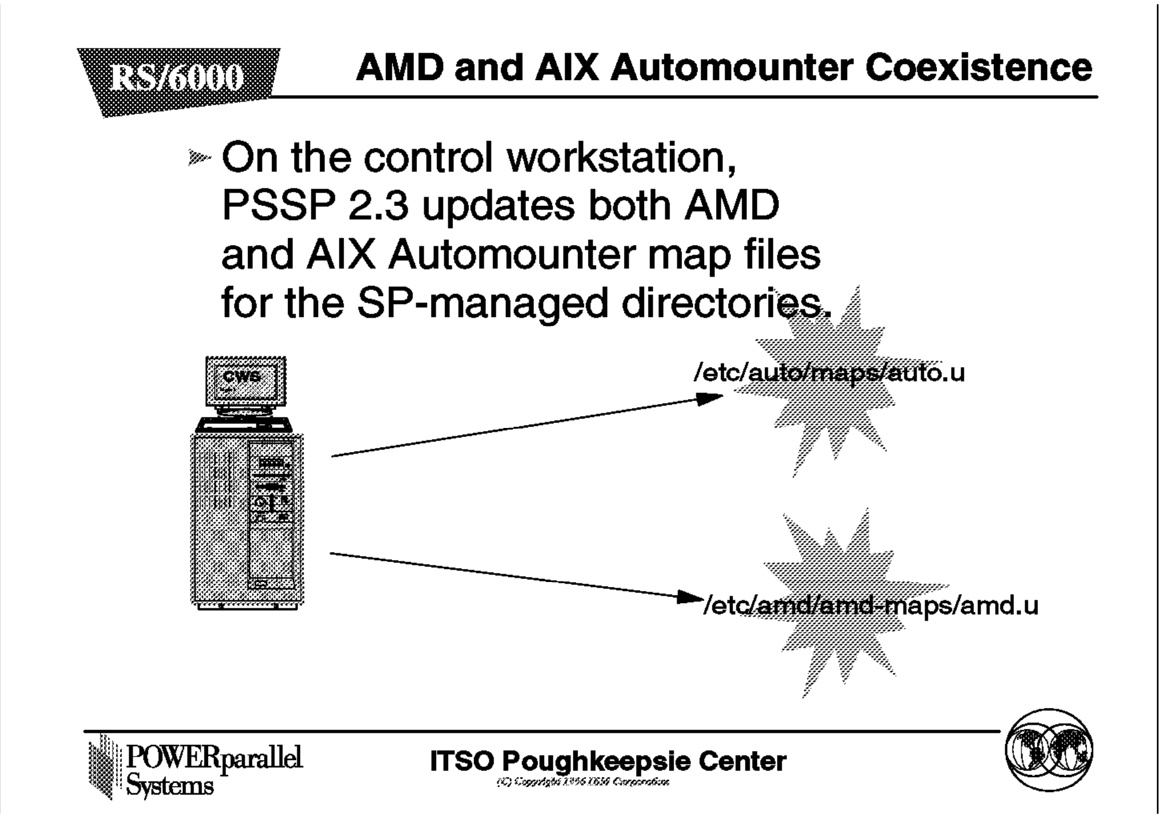
If there were no AMD configuration files, then default AMD and AIX Automounter configuration files are created by the process, and only then does the startup process run. This is the case when you are doing a fresh installation of PSSP 2.3 on the Control Workstation.

## 5.7 Coexistence of the AMD and AIX Automounters

If your system has a mixture of PSSP versions, then the nodes that are running PSSP 2.3 will use the AIX Automounter, while nodes with previous versions will run the AMD Automounter.

On such a system, the Control Workstation should be at the PSSP 2.3 level in order to install and manage PSSP 2.3 nodes. The control workstation will be running the AIX Automounter, but will still have the AMD configuration files and the AIX Automounter files.

Whenever you add, delete, or change SP users by means of the spmkuser, sprmuser, or spchuser commands, the Control Workstation updates both AMD and AIX Automounter map files for the SP-managed /u file system, provided that the site environment variables amd_config and usermgmt_config are set up properly.



RS/6000

## AMD and AIX Automounter Coexistence

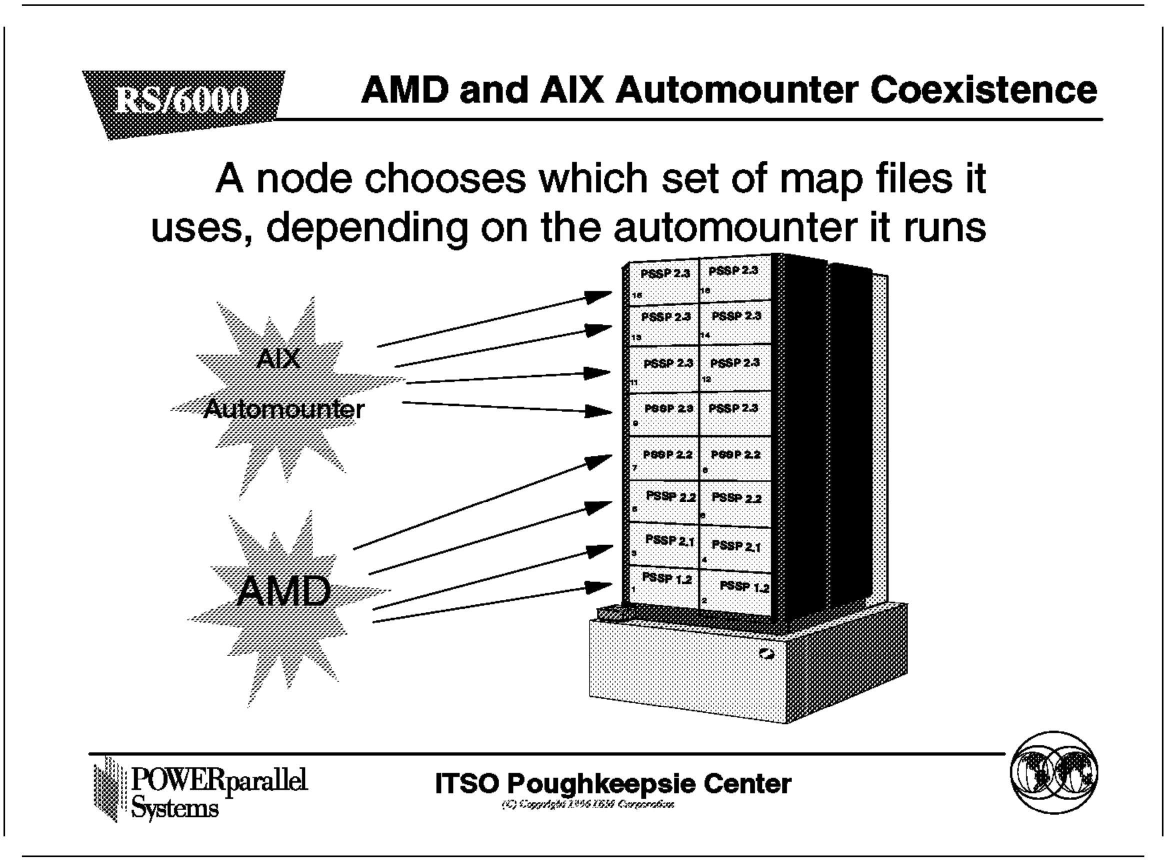
- On the control workstation, PSSP 2.3 updates both AMD and AIX Automounter map files for the SP-managed directories.

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

The Control Workstation distributes both AMD and AIX Automounter configuration files. When a node boots up, it runs services_config. On the PSSP 2.3 nodes, services_config starts the AIX Automounter daemon, which uses the /etc/auto/maps and /etc/auto.master directories for its maps. On the nodes with older levels of PSSP, the AMD daemon is started. It uses the /etc/amd directory structure in order to read its maps. Therefore, nodes with any level of PSSP are able to mount all the file systems, since the configuration files for each automounter reside in a different place. These configuration files are distributed to all nodes, and each automounter uses its own configuration files.



RS/6000

## AMD and AIX Automounter Coexistence

A node chooses which set of map files it uses, depending on the automounter it runs

POWERparallel Systems

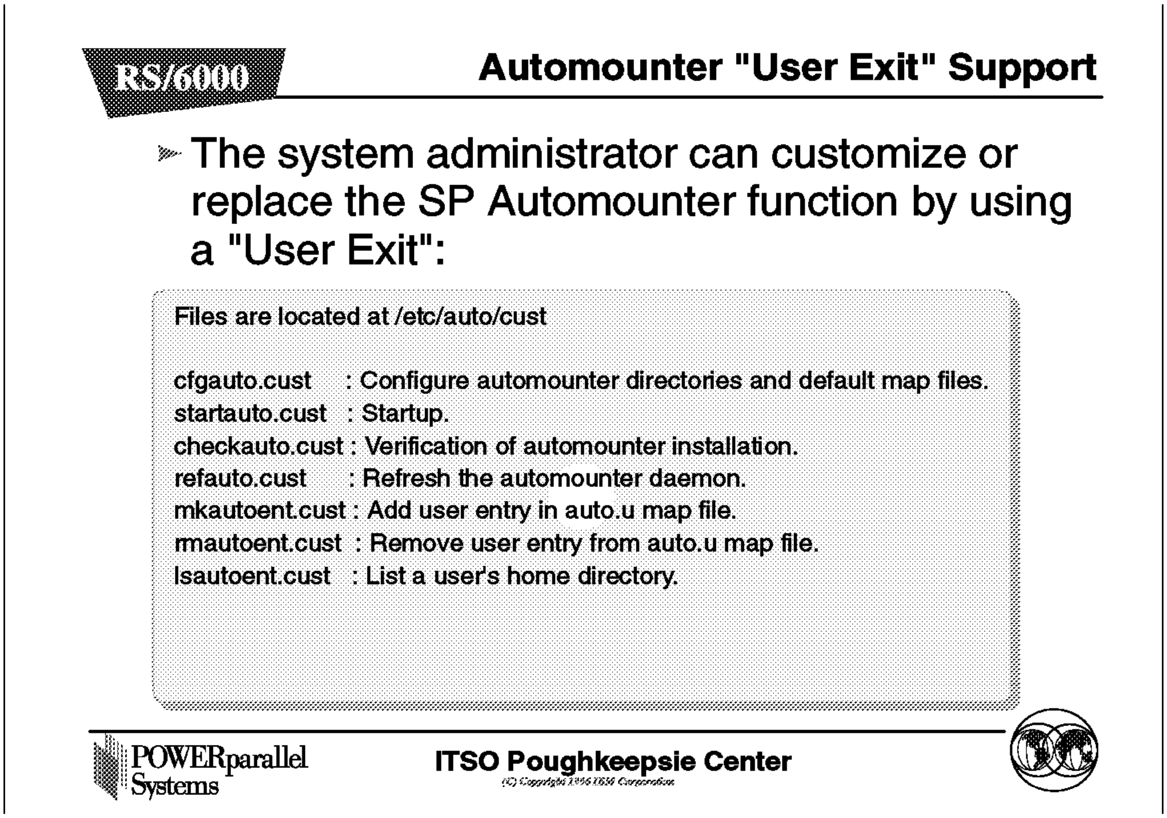
ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

**Note:**

Do not start more than one Automounter on a given node to manage the same file system or you might hang that file system.

## 5.8 User Exit Support



RS/6000

Automounter "User Exit" Support

- The system administrator can customize or replace the SP Automounter function by using a "User Exit":

Files are located at /etc/auto/cust

cfgauto.cust : Configure automounter directories and default map files.
startauto.cust : Startup.
checkauto.cust : Verification of automounter installation.
refauto.cust : Refresh the automounter daemon.
mkautoent.cust : Add user entry in auto.u map file.
rmautoent.cust : Remove user entry from auto.u map file.
lsautoent.cust : List a user's home directory.

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

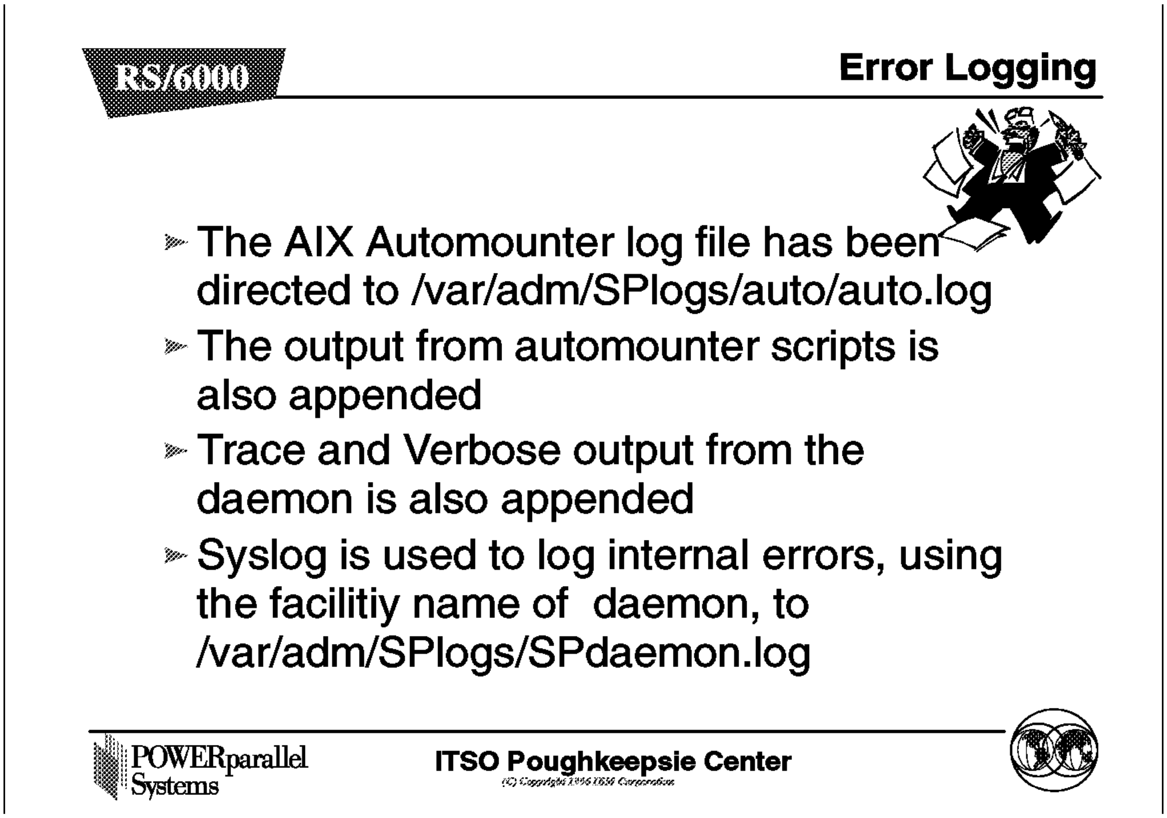
The new AIX Automounter support has been implemented to let system administrators customize part or all of the automounter functionality, with scripts that meet their sites′ needs.

During execution of every AIX Automounter function, the /etc/auto/cust directory is checked to see if a user program exists and can be executed. If it does, then that file is executed instead of the native function.

As an example, the following figure shows a possible startauto.cust for the AMD Automounter.

```
#!/usr/bin/ksh

#Check if the Amd daemon is running
if [[ -n `ps -fe | grep /etc/amd/amd | \
    grep -v grep` ]] ; then
    echo ″startauto.cust: amd daemon is already running.″
    exit 0
fi

# Build amd input list using all amd.* map files
# in /etc/amd/amd-maps.
set -A amdmaps $(ls /etc/amd/amd-maps/amd.*)
let i=0
while (( $i < ${#amdmaps[*]} )) ; do
  amd_argv=“$amd_argv /${amdmaps[$i]##/*/amd.} \
           ${amdmaps[$i]}”
  let i=$i+1
done
# Start the daemon
nice --4 /etc/amd/amd -t 16.120 -x all -l /var/adm/SPlogs/auto/auto.log\
         $amd_argv
exit $?
```

## 5.9 Error Logging



RS/6000

**Error Logging**

- The AIX Automounter log file has been directed to /var/adm/SPlogs/auto/auto.log
- The output from automounter scripts is also appended
- Trace and Verbose output from the daemon is also appended
- Syslog is used to log internal errors, using the facilitiy name of daemon, to /var/adm/SPlogs/SPdaemon.log

POWERparallel Systems

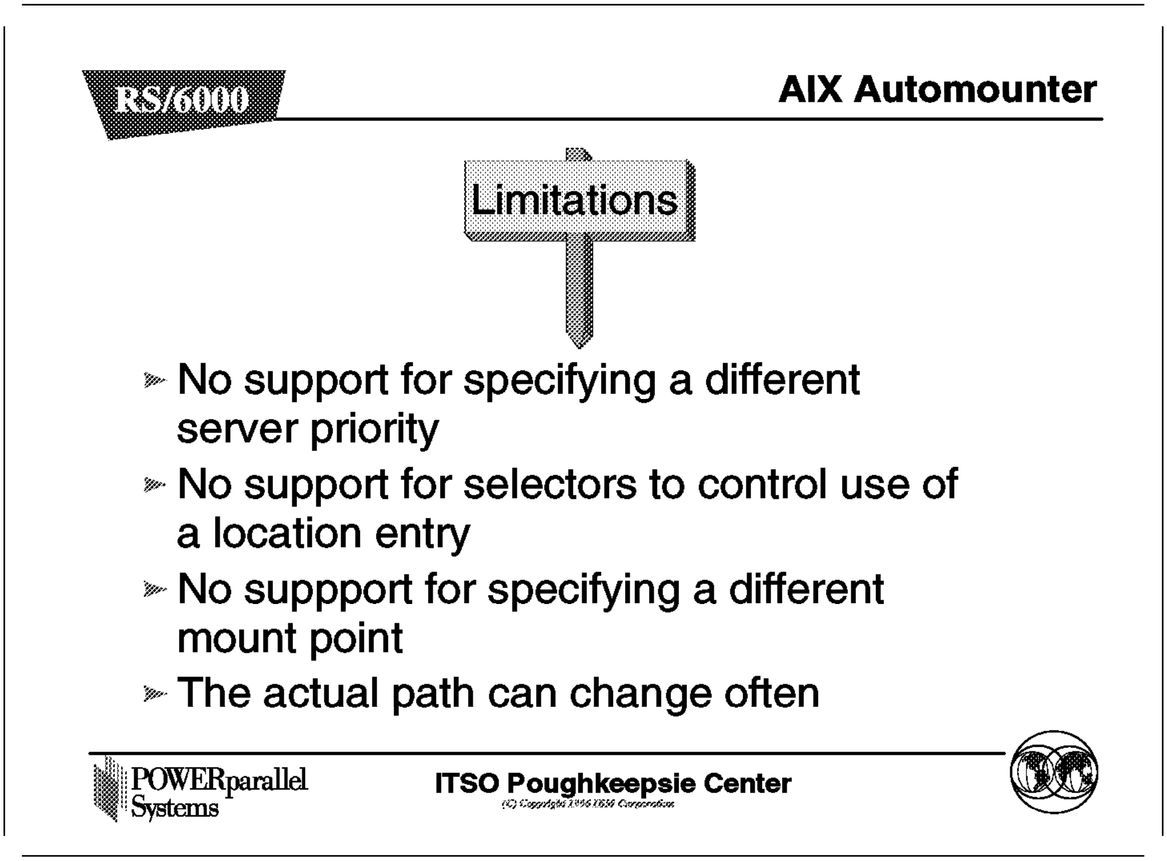
ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

The AIX Automounter facility uses a new log file, which resides in /var/adm/SPlogs/auto/auto.log. The output from the AIX Automounter configuration process and from the scripts that start and refresh the automounter daemon is stored here, as well as standard output and standard error messages. Also, when errors occur during AIX Automounter configuration or startup, they are logged in the /var/adm/SPlogs/SPdaemon.log file.

All internal errors are logged using the syslog and the facility name daemon.

## 5.10 AIX Automounter Limitations



RS/6000

**AIX Automounter**

Limitations

- No support for specifying a different server priority
- No support for selectors to control use of a location entry
- No suppport for specifying a different mount point
- The actual path can change often

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

In this section we point out some limitations of the AIX Automounter, comparing it with facilities offered by the BSD Automounter (AMD). We expect the reader to be familiar with AMD and AMD configuration files.

The limitations are:

- No support for specifying different server priority

  With the AMD Automounter, you can specify an entry like the following for a subdirectory on the file /etc/amd/amd-maps/amd.net:

```
local   host==sp3en;opts=ro,soft,intr;type=link;fs=/exports/local \
        host!=sp3en;type=nfs;rfs=/exports/local;rhost=sp3sw rhost=sp3en
```

  This entry will make the /net/local directory available on the nodes from the NFS server sp3en. If you take a look at the rhost specification, you see that we are using two hosts: sp3sw and sp3en. The sp3sw host corresponds to Node 3 on the SP, using the IP address of the switch interface, while sp3en refers to the same node, but uses the Ethernet IP address.

  When a node does an NFS mount of a directory, the mount operation is done using the IP address of the switch interface, if the switch is up and running. If the switch goes down, or the switch is not up at the moment of the initial mount, the Ethernet interface is used. The order on the rhost field implies a priority for the hosts.

On the Control Workstation, the first entry (sp3sw) will fail, since there is no access from the Control Workstation to the switch interface of the nodes. The second entry, which matches the Ethernet address of the nodes, will succeed and the mount will be done over the Ethernet.

Using AIX Automounter, you do not have such a possibility. On the map file for the automounter you can specify more than one host as follows:

```
local       sp3sw,sp3en:/exports/local
```

The meaning for the AIX Automounter is different, since there is no priority implied by the order of the host names. The first host that answers the NFS mount operation is the host from where the mount gets done.

- No support for specifying a different mount point

  On AMD you can override the mount point for a specific directory by using the fs:=mount_point syntax on the AMD configuration file. This is not supported on the AIX Automounter; the default mount point is always used.

- No support for selectors to control the use of a location entry

  On the AMD Automounter you can specify the use of a location entry by using equivalence operators as shown:

  ```
  keyword==value
  keyword!=value
  ```

  There is no such facility with the AIX Automounter.

- The mount point for a particular directory might change often

  Let us illustrate this issue with an example, using /etc/auto/maps/auto.u:

  ```
  user1      nfsserv1:/home/nfsserv1:&
  user2      nfsserv1:/home/nfsserv1:&
  ```

  Both user1 and user2 work on the client1 machine. So, whenever one of these users wants to access the home directory, a remote mount operation will take place.

  If user1 is the first user to log onto the client1 machine, the following operations will occur:

  1. The client1 machine mounts the remote file system.

     The client1 machine does an NFS mount of the remote /home/nfsserv1 directory from the nfsserv1 machine, which looks like the following:

     ```
     mount nfsserv1:/home/nfsserv1 /tmp_mnt/u/user1
     ln -s /tmp_mnt/u/user1/user1 /u/user1
     ```

If user2 logs onto the client1 machine now, and wants to access his home directory, then the NFS mount is not done, since the remote directory /home/nfsserv1 from server nfsserv1 is already mounted at /tmp_mnt/u/user1. The only thing the AIX Automounter does here is a symbolic link, as follows:

```
ln -s /tmp_mnt/u/user1/user2 /u/user2

/u/user2 -> /tmp_mnt/u/user1/user2
```

So, for user1, the actual path for the home directory is /tmp_mnt/u/user1/user1. For user2, the actual home directory is at /tmp_mnt/u/user1/user2.

2. If the file systems are not being used, they will be unmounted

   If both user1 and user2 disconnect from the client1 machine, and no other process is using either /u/user1 or /u/user2, then after a period of time the AIX Automounter will unmount the remote file system and erase the symbolic links.

   If user2 now logs onto the client1 machine, and accesses his home directory, the mount will look like this:

```
mount nfsserv1:/home/nfsserv1 /tmp_mnt/u/user2
ln -s /tmp_mnt/u/user2/user2  /u/user2
```

   The actual path for user2 will now be /tmp_mnt/u/user2/user2. If user1 now logs into the client1 machine, then the following link will be done, because the NFS mount of the remote nfsserv1:/home/nfsserv1 directory is already done:

```
ln -s /tmp_mnt/u/user2/user1  /u/user1
```

The actual path for the subdirectories can change often, and you cannot be certain in many situations what it is. For users of the C shell, this might be confusing, since the C shell follows the links, and if you issue a pwd command, it will show the actual path and not the path you did the cd to.

Some applications might also get confused because of the changing path.

## 5.11 Command Syntax

In this section we describe the syntax of some commands related to the AIX Automounter.

### 5.11.1 The automount Command

Table 3. Command Line Syntax for Automount

| **Command Line Argument** | **Description** |
|---|---|
| **-M** directory | Specifies a different mount directory. The default is /tmp_mnt. |
| **-m** | Ignore NIS auto.master database. |
| **-tl** seconds | Specifies how many seconds a mount is kept while not in use (default is 300). |
| **-tm** seconds | Specifies the interval between attempts to mount a filesystem (default is 30). |
| **-tw** seconds | Specifies the interval between attempts to unmount a filesystem (default is 60). |
| **-v** | Display verbose msgs. |
| **-T** | Trace NFS calls. |
| **-D** var=value | Set or override map variables. |

### 5.11.2 The mkautomap Command

The mkautomap command generates an equivalent AIX Automounter map file from an AMD map file.

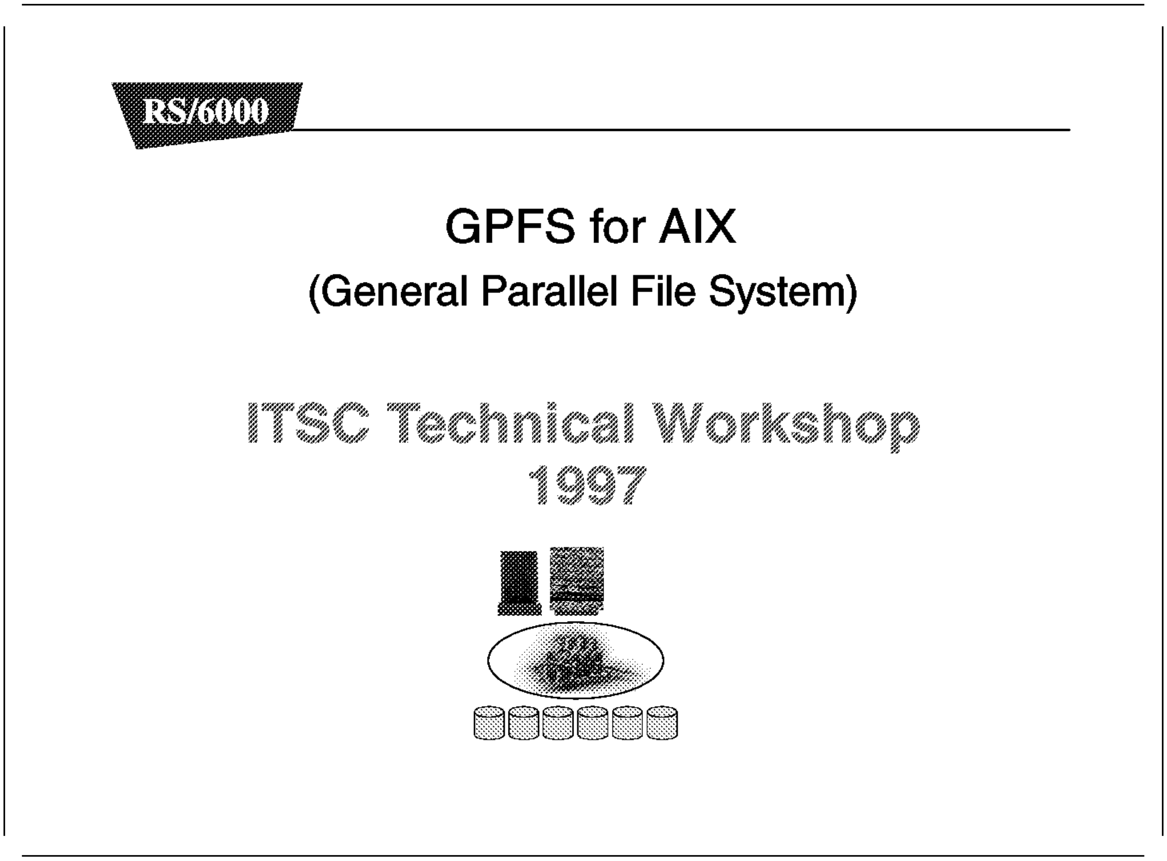
The syntax for the mkautomap command is:

- `mkautomap [-n ] [ -o Automount_map ] [ -f filesystem ] [Amd_map]`

Table 4. Command Line Syntax for Automount

| **Command Line Argument** | **Description** |
|---|---|
| **-n** | Specifies that an entry should not be added in the /etc/auto.master master map file. |
| **-o** Automount_map | Specifies the filename of the automount map where the generated output will be placed. If this file exists, it is replaced. The default output file is /etc/auto/maps/auto.u. |
| **-f** filesystem | Specifies the name of the file system associated with the automounter map files. The default file system if it is not specified, is /u. |
| **Amd_map** | Specifies the filename of the AMD input map file. The default is /etc/amd/amd-maps/amd.u. |

# Chapter 6. General Parallel File System (GPFS)



RS/6000

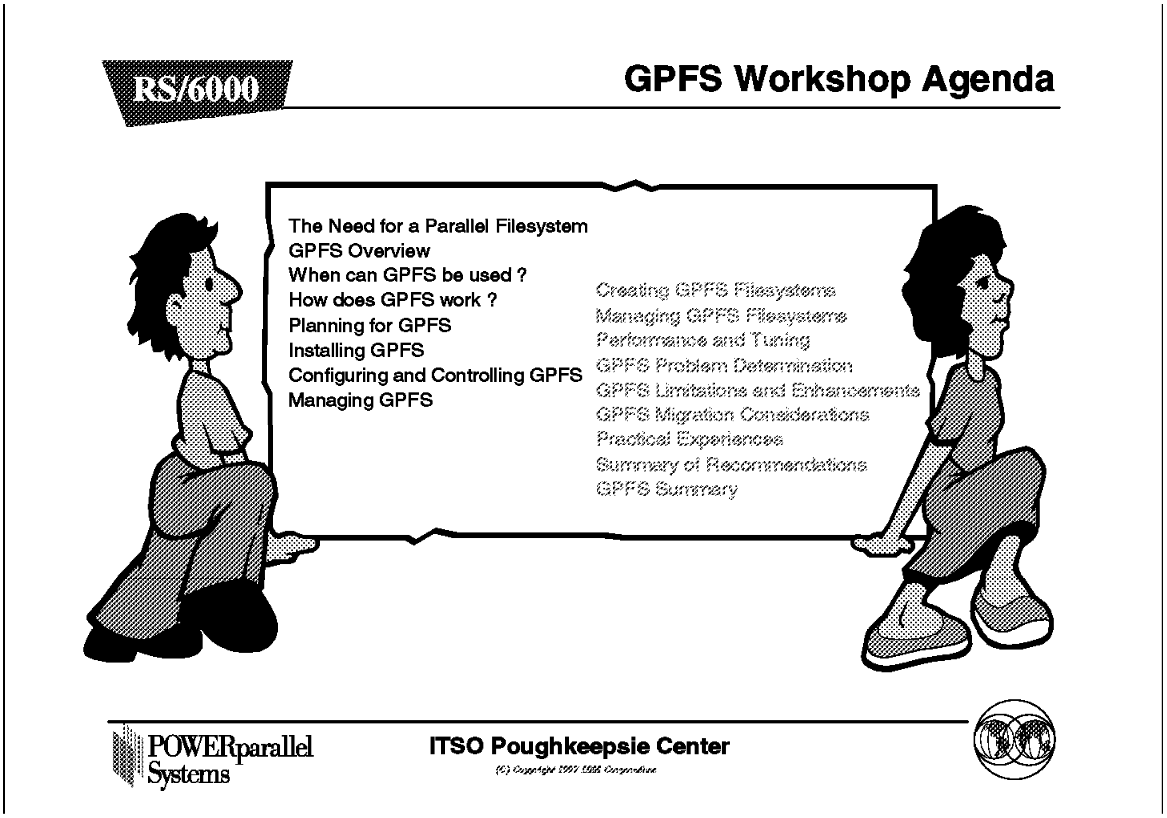
GPFS for AIX

(General Parallel File System)

ITSC Technical Workshop 1997

This chapter discusses the General Parallel File System (GPFS), a software product supported with PSSP 2.3 on the SP.

## 6.1 GPFS Workshop Agenda



RS/6000

GPFS Workshop Agenda

The Need for a Parallel Filesystem
GPFS Overview
When can GPFS be used ?
How does GPFS work ?
Planning for GPFS
Installing GPFS
Configuring and Controlling GPFS
Managing GPFS

Creating GPFS Filesystems
Managing GPFS Filesystems
Performance and Tuning
GPFS Problem Determination
GPFS Limitations and Enhancements
GPFS Migration Considerations
Practical Experiences
Summary of Recommendations
GPFS Summary

POWERparallel Systems

ITSO Poughkeepsie Center

The needs and requirements for a parallel file system are discussed first.

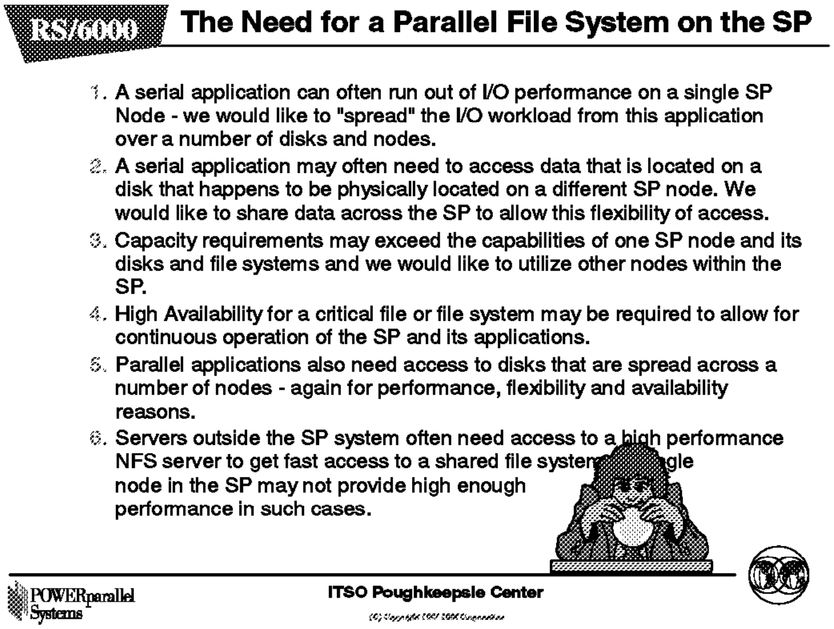
We then discuss some of the technologies that are used by GPFS but are not new in PSSP, such as Virtual Shared Disk. This section is meant as a refresher for those who are already familiar with these topics, and as a quick primer for those who are new to them.

We then look in more detail at how GPFS works, how it should be installed and configured, and how it should be managed.

Other sections cover performance, limitations, and other aspects of GPFS.

Finally, we discuss practical experiences and provide some recommendations and guidance.

## 6.2 The Need for a Parallel File System on the SP



### The Need for a Parallel File System on the SP

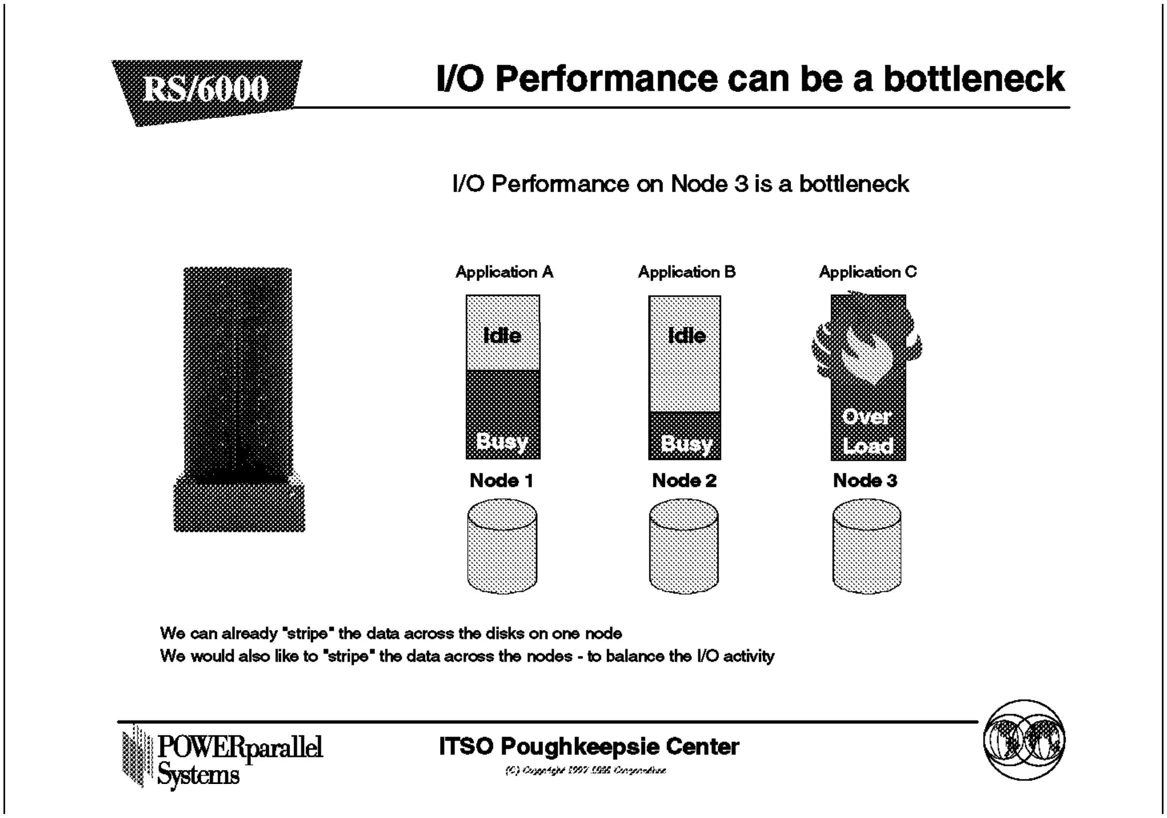
1. A serial application can often run out of I/O performance on a single SP Node - we would like to "spread" the I/O workload from this application over a number of disks and nodes.
2. A serial application may often need to access data that is located on a disk that happens to be physically located on a different SP node. We would like to share data across the SP to allow this flexibility of access.
3. Capacity requirements may exceed the capabilities of one SP node and its disks and file systems and we would like to utilize other nodes within the SP.
4. High Availability for a critical file or file system may be required to allow for continuous operation of the SP and its applications.
5. Parallel applications also need access to disks that are spread across a number of nodes - again for performance, flexibility and availability reasons.
6. Servers outside the SP system often need access to a high performance NFS server to get fast access to a shared file syste[illegible]gle node in the SP may not provide high enough performance in such cases.

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This section deals with the requirements for a parallel file system. These are only partially satisfied today with products such as PIOFS, Virtual Shared Disk, NFS, JFS, and DFS.

### 6.2.1 I/O Performance Can Be a Bottleneck



I/O Performance can be a bottleneck

I/O Performance on Node 3 is a bottleneck

We can already "stripe" the data across the disks on one node
We would also like to "stripe" the data across the nodes - to balance the I/O activity

POWERparallel Systems

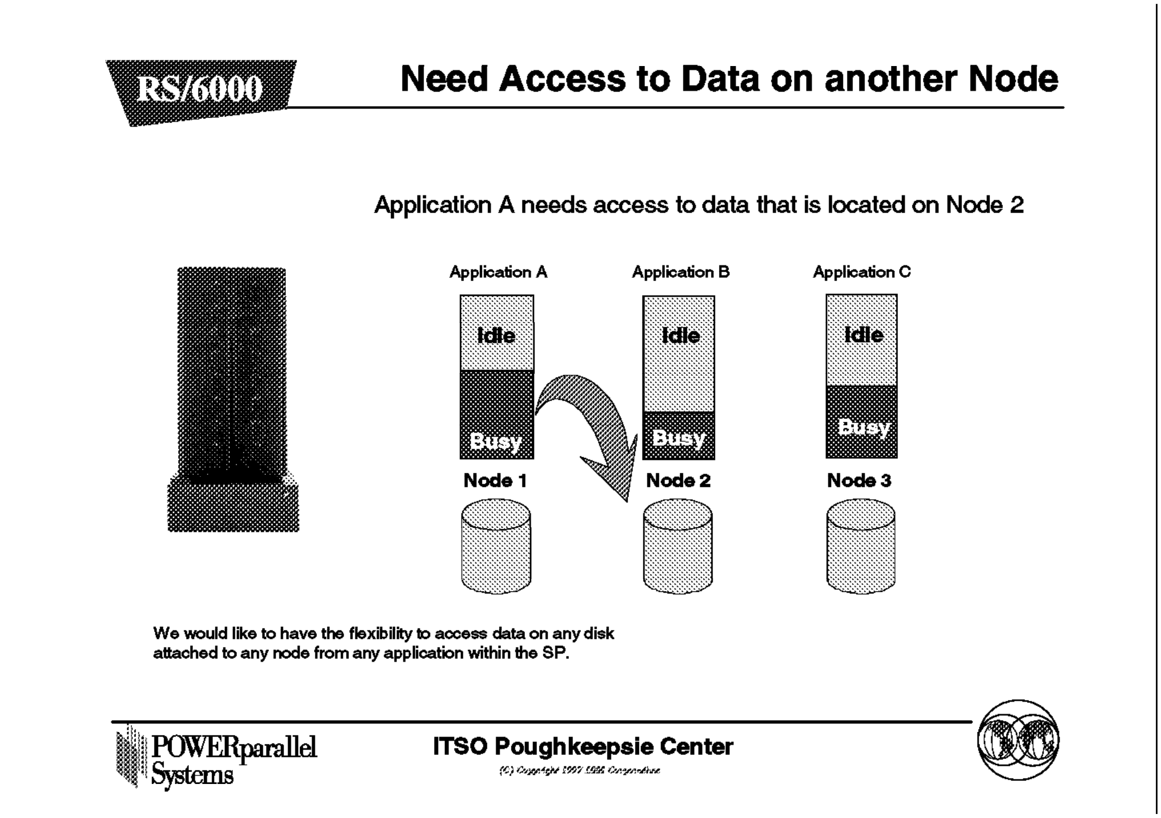
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In this example, our application results in a lot of heavy I/O to Node 3. As a result, Node 3 becomes overloaded.

We would like to be able to "spread" this I/O activity over other nodes. We can already spread the I/O over multiple disks on one node, but the requirement is to also spread the I/O activity over other nodes. This leads to the requirement, of course, to have access to the disks on other nodes within the SP. This is only possible *directly* today with VSD (Virtual Shared Disk), but VSD does not support general I/O activity. In particular, VSD does not support file systems. A solution based on NFS (Network File System) will always point to a single node as the source for that data and, in addition, in many environments, NFS is not a high-performance solution.

## 6.2.2 Need Access to Data on Other Nodes



RS/6000

**Need Access to Data on another Node**

Application A needs access to data that is located on Node 2

We would like to have the flexibility to access data on any disk attached to any node from any application within the SP.

POWERparallel Systems

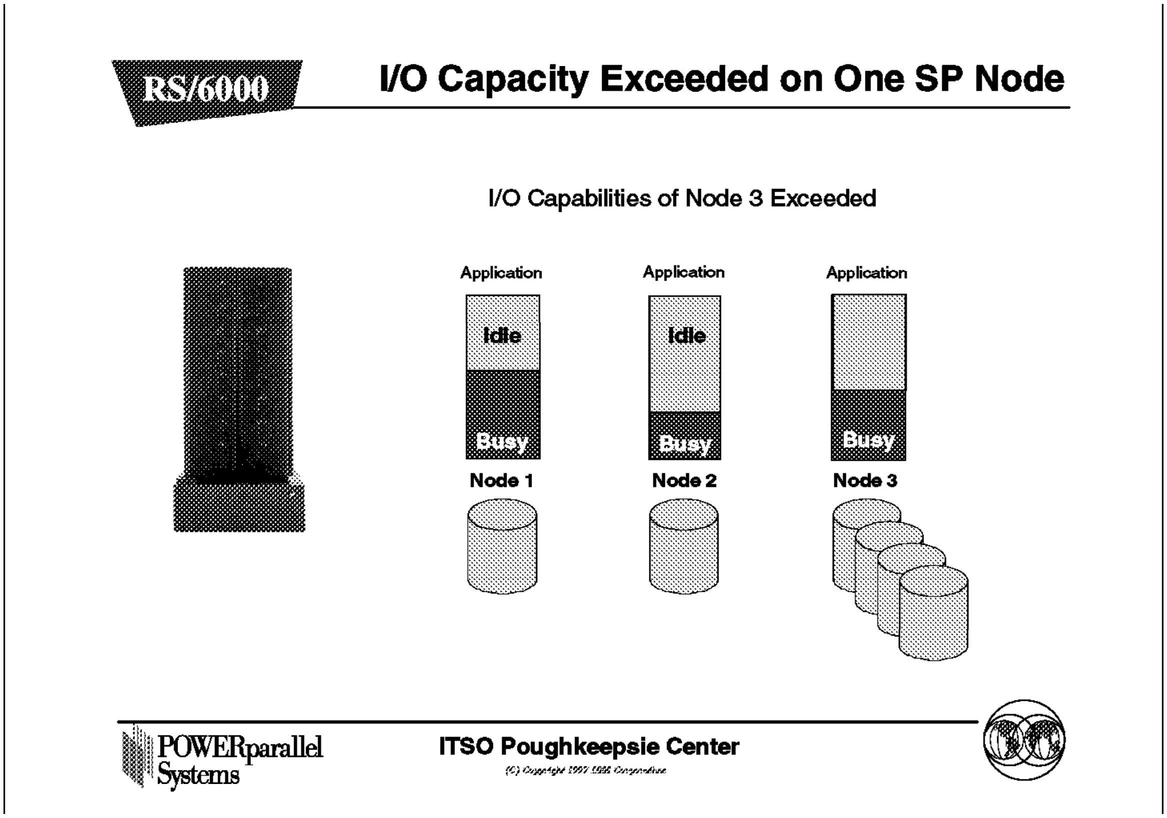
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

From a pure flexibility point of view, it is often very helpful to have access to data that resides on other nodes within the SP so that an application can run on any node within the SP, yet have access to data that it needs.

Once again, NFS could provide such a solution, but its performance is not adequate in many cases. In particular, the write function of NFS can be slow.

## 6.2.3 I/O Capacity Exceeded on One SP Node



RS/6000

**I/O Capacity Exceeded on One SP Node**

I/O Capabilities of Node 3 Exceeded

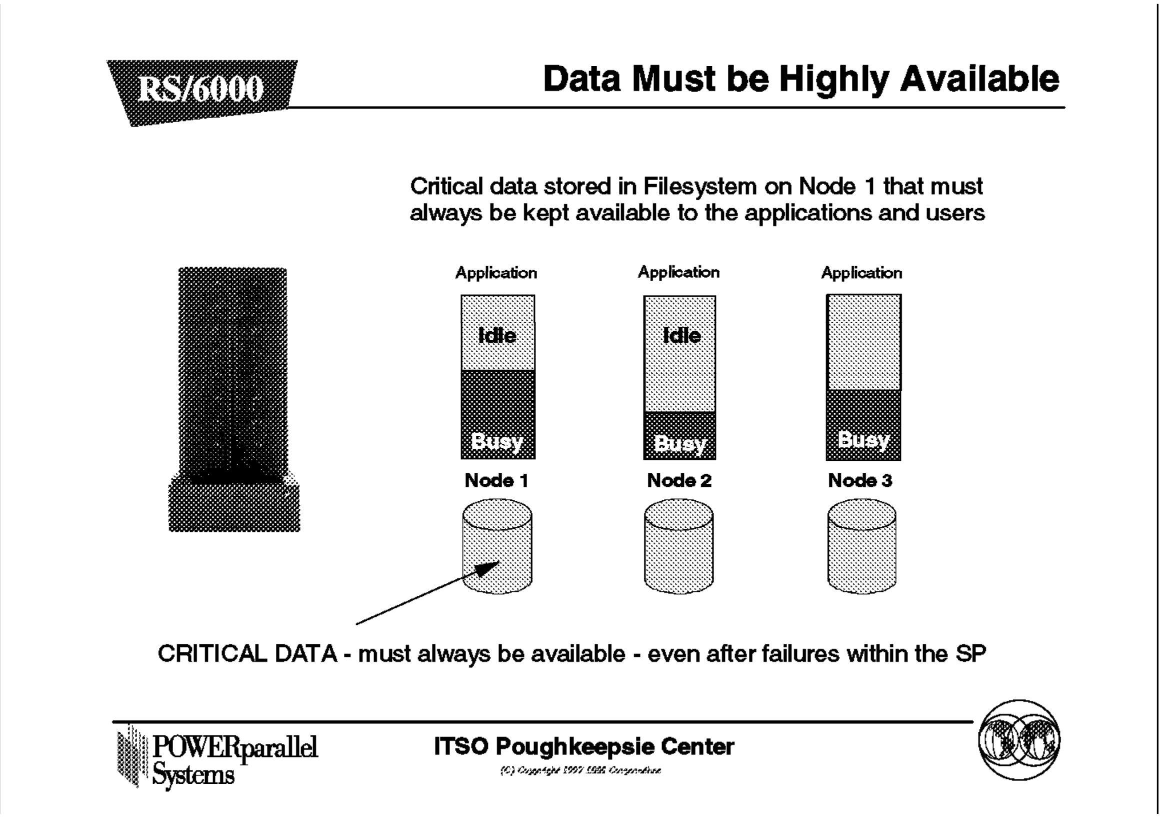
POWERparallel Systems

ITSO Poughkeepsie Center

In this example, the capacity of node 3 has been exceeded from an I/O point of view. For example, adapter slots may have been filled and there is no room for more adapters to connect additional disks.

We would like the ability to access disks that are attached to another node in the SP to take advantage of spare capacity elsewhere.

### 6.2.4 Data Must Be Highly Available



## Data Must be Highly Available

Critical data stored in Filesystem on Node 1 that must always be kept available to the applications and users

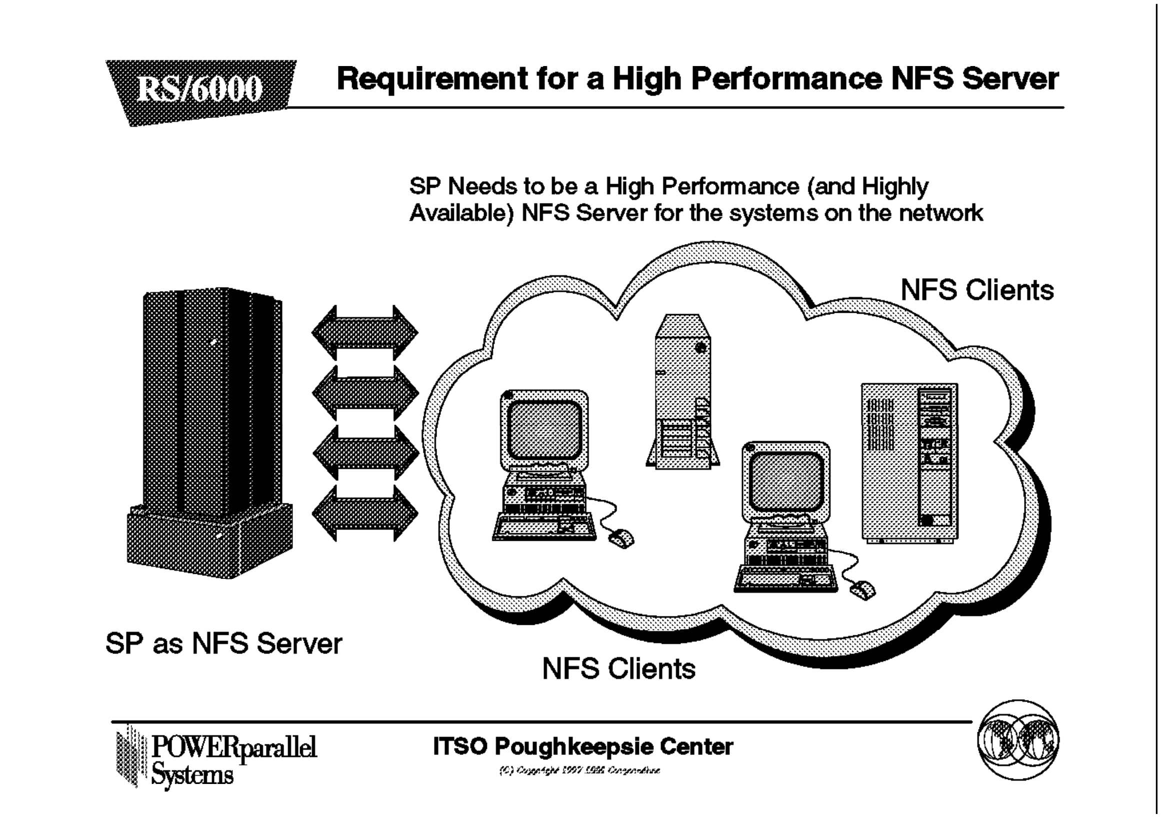
CRITICAL DATA - must always be available - even after failures within the SP

ITSO Poughkeepsie Center

A common requirement is to have critical data on a system that is kept highly available. Solutions do exist to provide this today with NFS, but they have limitations.

As we will see, the GPFS solution is an ideal option in such circumstances.

## 6.2.5 High Performance NFS Server



RS/6000

Requirement for a High Performance NFS Server

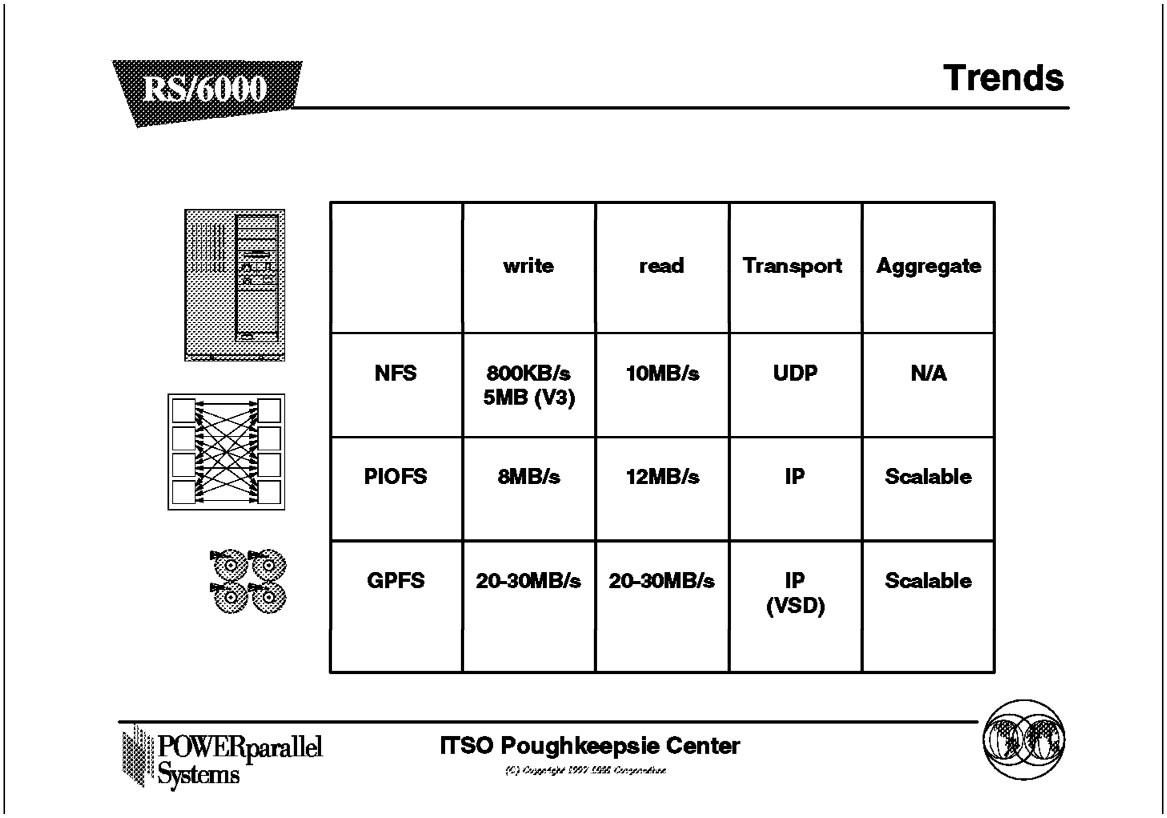
SP Needs to be a High Performance (and Highly Available) NFS Server for the systems on the network

POWERparallel Systems

ITSO Poughkeepsie Center

Some customers already have a network of systems that require access to an NFS server or servers. The requirements for such servers, if they are providing data access to a large number of servers, are usually that they must provide high performance, and often high availability in addition.

## 6.2.6 File System Trends



RS/6000

**Trends**

| | write | read | Transport | Aggregate |
|---|---|---|---|---|
| NFS | 800KB/s<br>5MB (V3) | 10MB/s | UDP | N/A |
| PIOFS | 8MB/s | 12MB/s | IP | Scalable |
| GPFS | 20-30MB/s | 20-30MB/s | IP<br>(VSD) | Scalable |

POWERparallel Systems

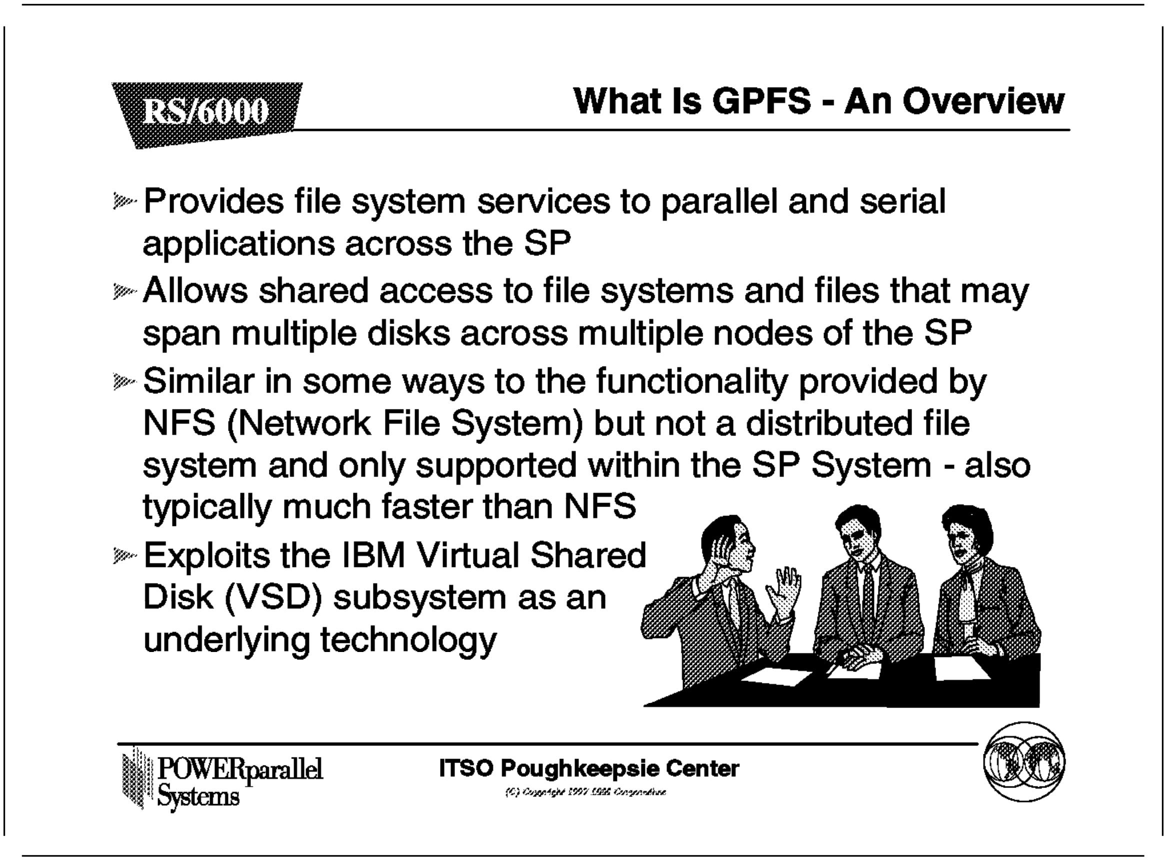
ITSO Poughkeepsie Center

(C) Copyright 1997 1996 Corporation

The trends in terms of file systems are shown here. Each of these solutions has advantages and disadvantages. The performance estimates are indicative of performance that might be obtained under optimal circumstances and with an optimal configuration.

Depending on the application, performance may well be very different from that shown here.

## 6.3 What is GPFS - An Overview



RS/6000

What Is GPFS - An Overview

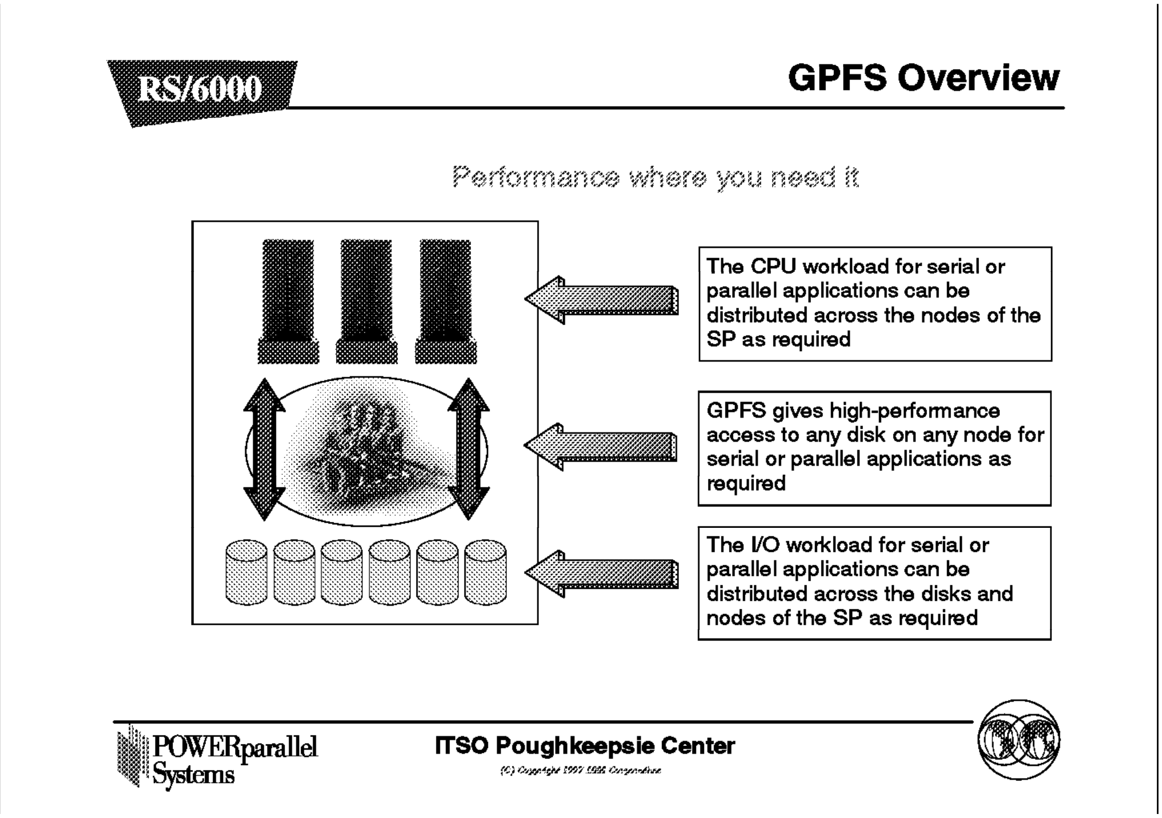
- Provides file system services to parallel and serial applications across the SP
- Allows shared access to file systems and files that may span multiple disks across multiple nodes of the SP
- Similar in some ways to the functionality provided by NFS (Network File System) but not a distributed file system and only supported within the SP System - also typically much faster than NFS
- Exploits the IBM Virtual Shared Disk (VSD) subsystem as an underlying technology

POWERparallel Systems

ITSO Poughkeepsie Center

GPFS is a software product that is available on the SP. The software provides the functionality as shown above, but it should be remembered that GPFS is only supported on the SP system. Systems external to the SP cannot run the GPFS software.

## 6.3.1 GPFS Overview



RS/6000

GPFS Overview

Performance where you need it

The CPU workload for serial or parallel applications can be distributed across the nodes of the SP as required

GPFS gives high-performance access to any disk on any node for serial or parallel applications as required

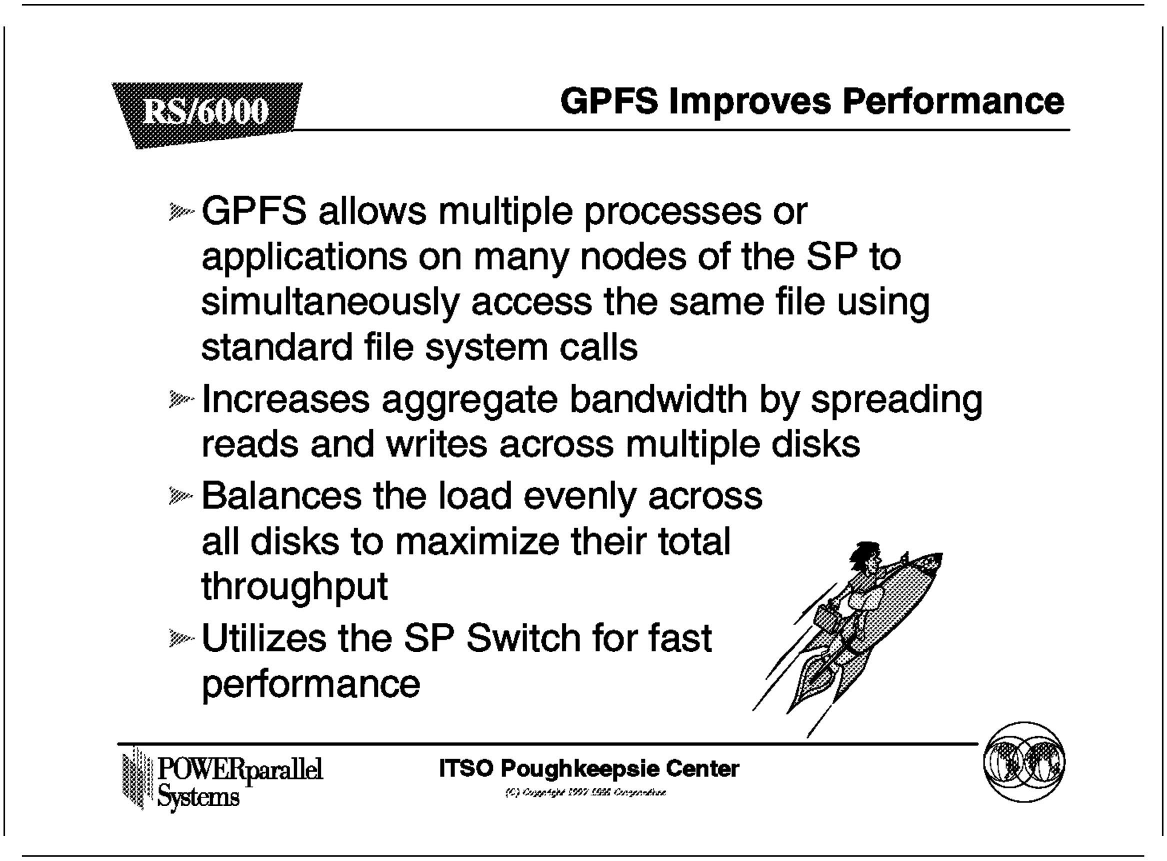
The I/O workload for serial or parallel applications can be distributed across the disks and nodes of the SP as required

POWERparallel Systems

ITSO Poughkeepsie Center

GPFS provides a truly parallel I/O solution for use within the SP. It can be exploited by both serial and parallel applications. It will allow for better balanced performance in many application environments, and will allow an application to exploit a number of nodes for I/O performance, as well as for CPU performance.

## 6.3.2 GPFS Improves Performance



RS/6000

GPFS Improves Performance

- GPFS allows multiple processes or applications on many nodes of the SP to simultaneously access the same file using standard file system calls
- Increases aggregate bandwidth by spreading reads and writes across multiple disks
- Balances the load evenly across all disks to maximize their total throughput
- Utilizes the SP Switch for fast performance

POWERparallel Systems

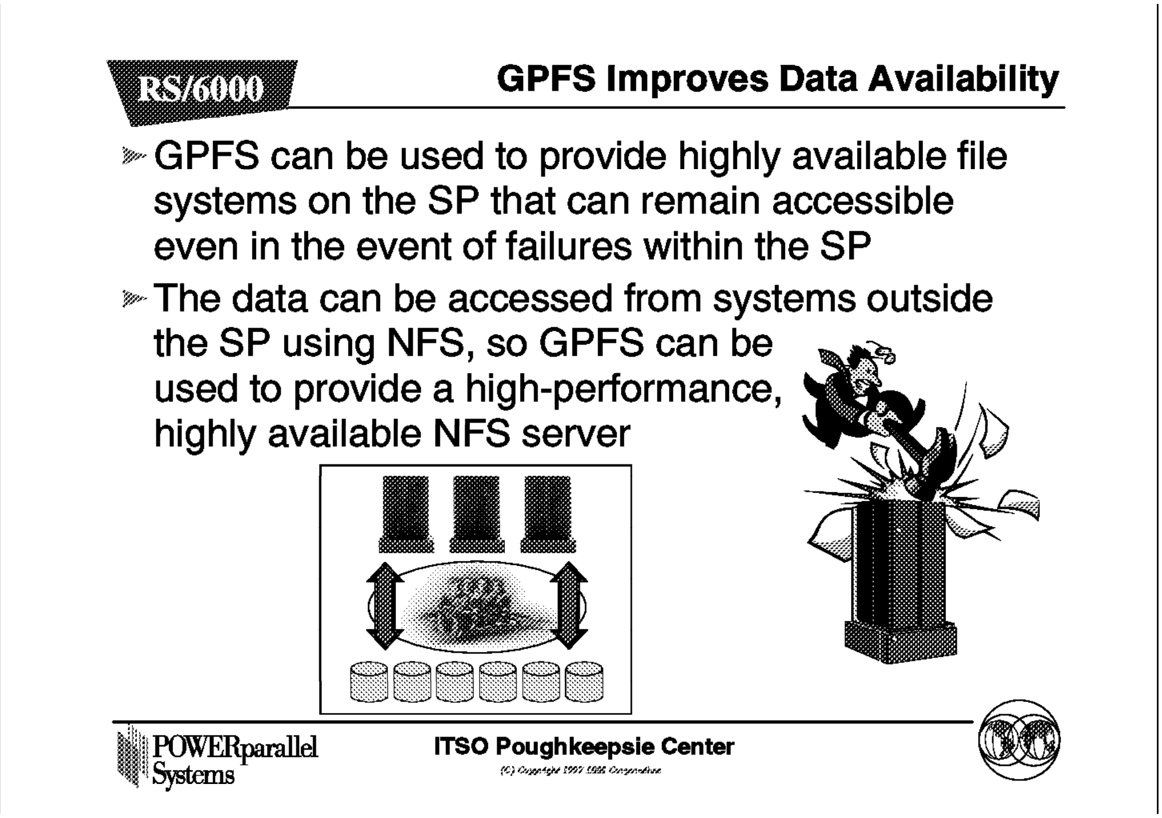
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The GPFS software provides fast access across the SP Switch to disks attached to remote nodes within the SP. This results in extended flexibility and better performance.

Concurrent access can be supported with the application providing locking in much the same way that NFS provides locking.

### 6.3.3 GPFS Improves Data Availability



RS/6000

**GPFS Improves Data Availability**

- GPFS can be used to provide highly available file systems on the SP that can remain accessible even in the event of failures within the SP
- The data can be accessed from systems outside the SP using NFS, so GPFS can be used to provide a high-performance, highly available NFS server

POWERparallel Systems

ITSO Poughkeepsie Center

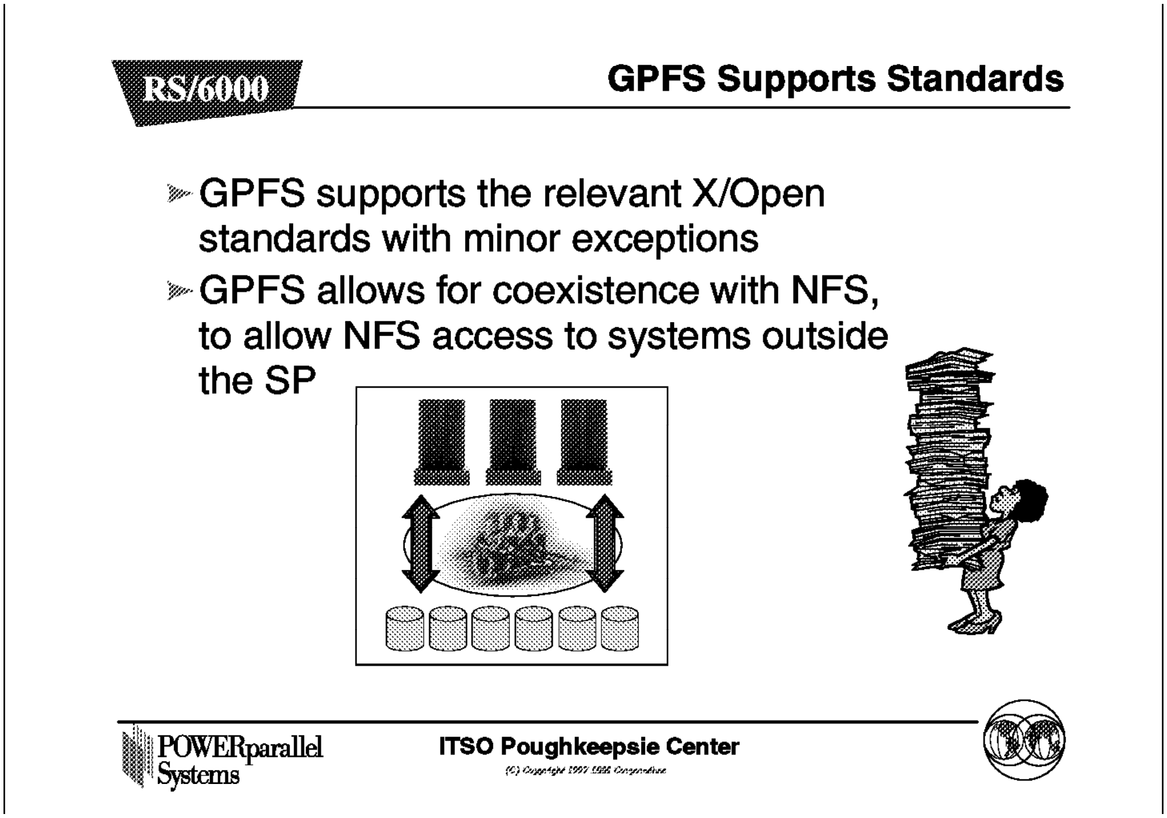
(C) Copyright 1997 IBM Corporation

As we will see, GPFS uses the VSD technology to access remote disks. An extension for VSD is Recoverable Virtual Shared Disk or RVSD. This can provide disk takeover in the event of failure.

GPFS works closely with RVSD and a highly available solution can be provided.

There are a number of design options in this area that will be discussed in this chapter. In addition to the usual options of disk mirroring or RAID disk subsystems, GPFS allows for RVSD and replication of data within GPFS.

## 6.3.4 GPFS Supports Standards


RS/6000

GPFS Supports Standards

- GPFS supports the relevant X/Open standards with minor exceptions
- GPFS allows for coexistence with NFS, to allow NFS access to systems outside the SP

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

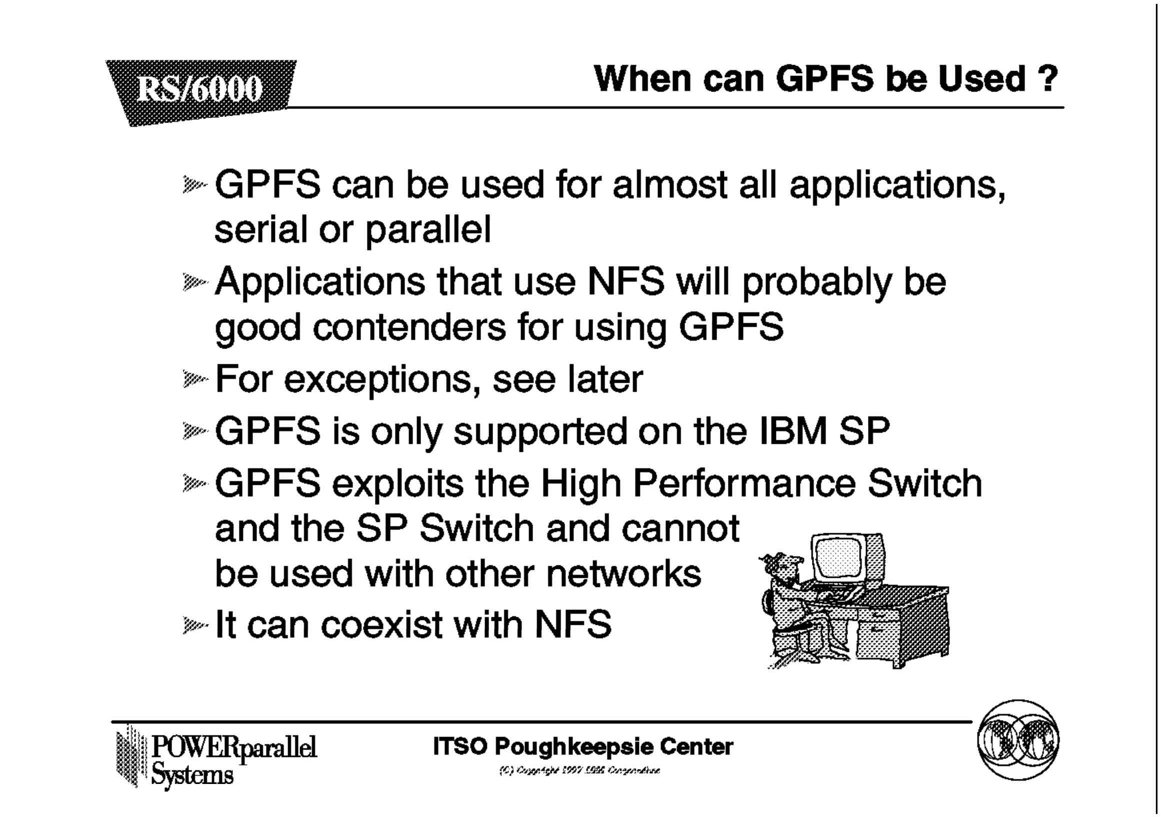
In most cases, GPFS supports the relevant standards for file systems as defined by X/Open. There are some minor exceptions that will be described later.

In many respects, a GPFS file system running within the SP will be seen as a normal file system; it will not normally be obvious that this is a GPFS file system.

Additional commands are delivered with GPFS for supporting GPFS file systems, but many standard AIX file system commands, such as mount or df, will work as expected.

A GPFS file system, once created, can be exported like any AIX file system, and can therefore be mounted on client systems either within the SP, or outside the SP, using normal NFS commands.

### 6.3.5 When Can GPFS Be Used?



RS/6000

When can GPFS be Used ?

- GPFS can be used for almost all applications, serial or parallel
- Applications that use NFS will probably be good contenders for using GPFS
- For exceptions, see later
- GPFS is only supported on the IBM SP
- GPFS exploits the High Performance Switch and the SP Switch and cannot be used with other networks
- It can coexist with NFS

POWERparallel Systems

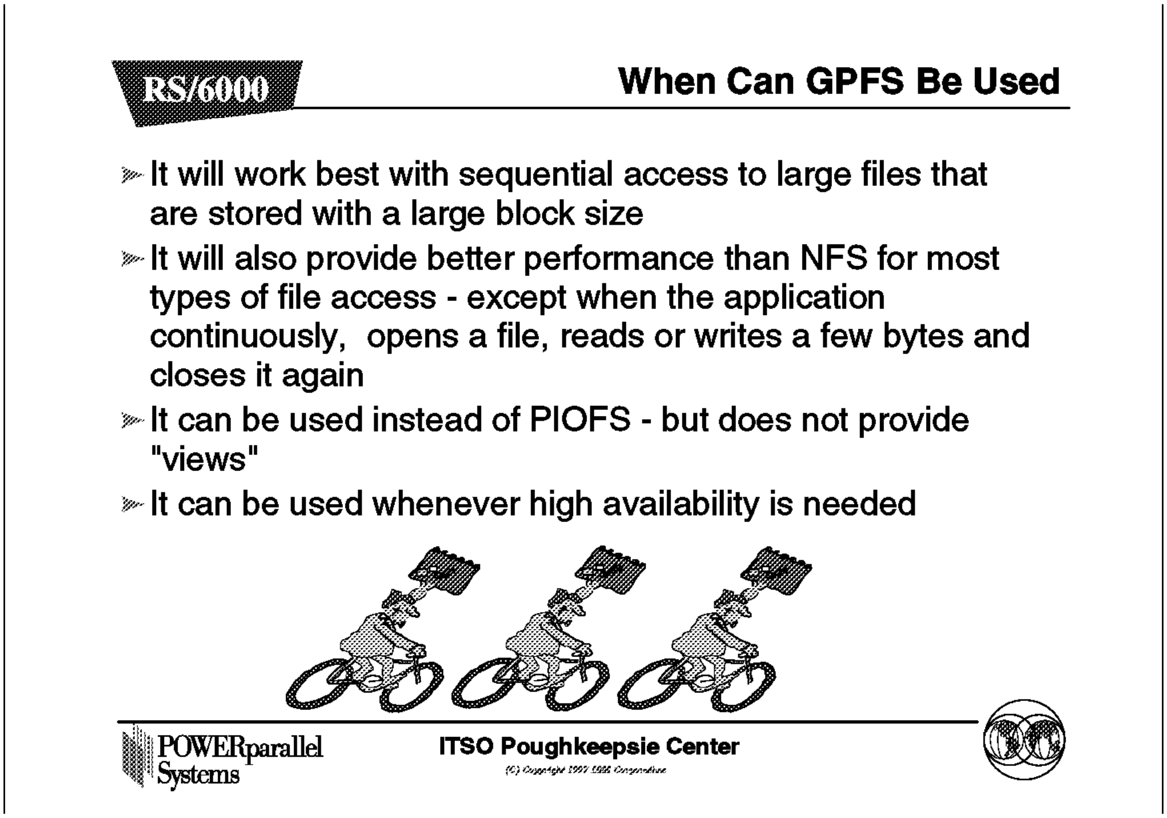
ITSO Poughkeepsie Center

GPFS can be used in most cases. There are a few cases where it may not provide additional advantages over other solutions such as NFS, but in most cases, SP customers are likely to want to implement and exploit GPFS.

It will be applicable for customers running either commercial applications, or scientific and technical applications.

GPFS is only supported on the IBM SP. In particular, it requires a fast network, namely the SP Switch, and uses security facilities within the SP for node-to-node communications.

### 6.3.5.1 When can GPFS be Used?



RS/6000

When Can GPFS Be Used

- It will work best with sequential access to large files that are stored with a large block size
- It will also provide better performance than NFS for most types of file access - except when the application continuously, opens a file, reads or writes a few bytes and closes it again
- It can be used instead of PIOFS - but does not provide "views"
- It can be used whenever high availability is needed

POWERparallel Systems

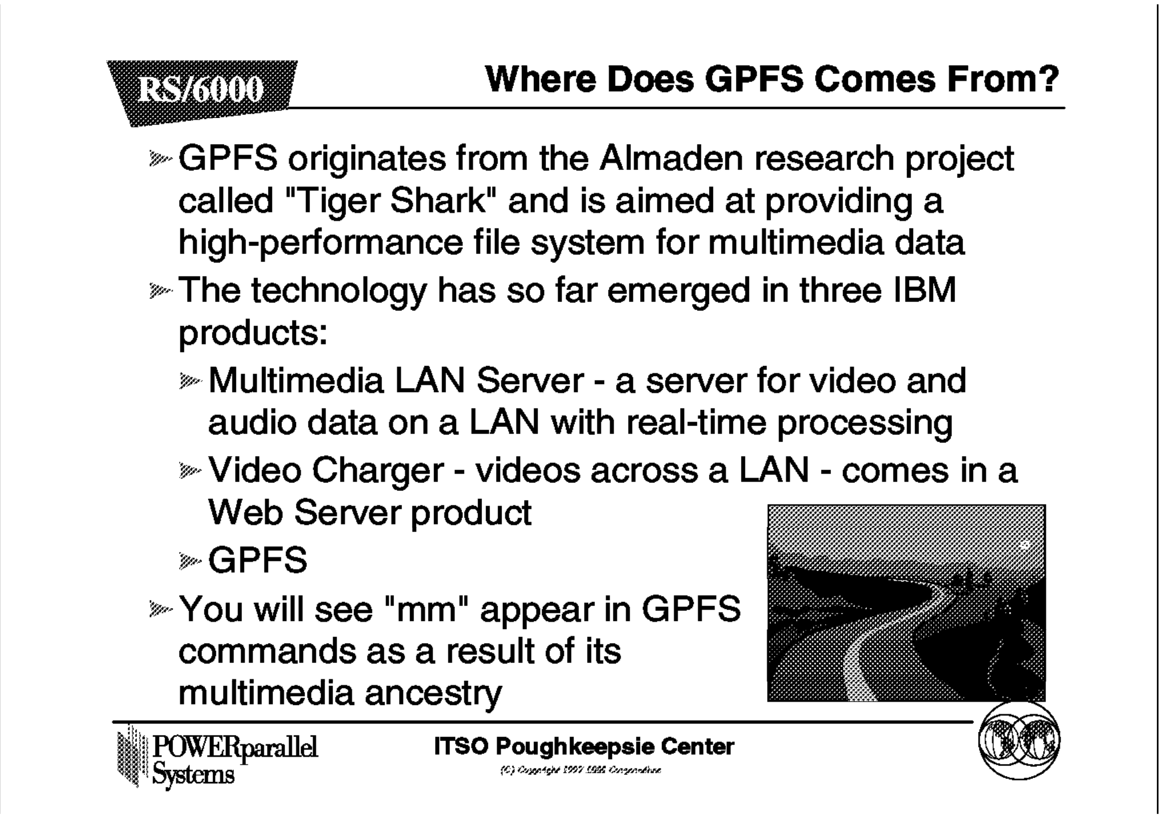
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

GPFS can be used in many cases, but the best performance will be seen under certain circumstances.

GPFS may be viewed as similar to PIOFS, but there are differences. A comparison between these two solutions will be shown later in this chapter. In most aspects, GPFS is superior to PIOFS.

## 6.4 How Does GPFS Work?



RS/6000

**Where Does GPFS Comes From?**

- GPFS originates from the Almaden research project called "Tiger Shark" and is aimed at providing a high-performance file system for multimedia data
- The technology has so far emerged in three IBM products:
  - Multimedia LAN Server - a server for video and audio data on a LAN with real-time processing
  - Video Charger - videos across a LAN - comes in a Web Server product
  - GPFS
- You will see "mm" appear in GPFS commands as a result of its multimedia ancestry

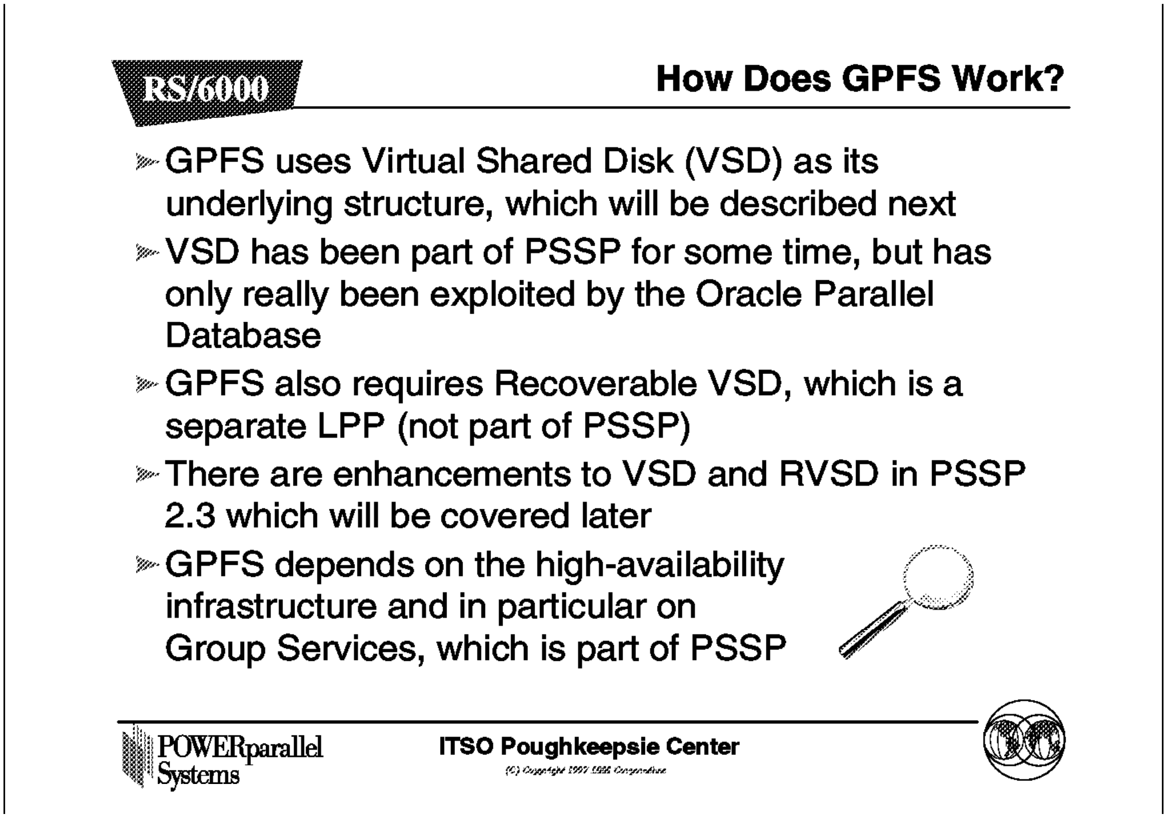
POWERparallel Systems

ITSO Poughkeepsie Center

GPFS originates from the Almaden Research Center. Because of its history as a multimedia file system (mmfs), many of the commands start with the letters mm. This technology has been used in three products to date. GPFS is one of them. On the SP today, only GPFS is supported.

Even if the other products were supported, it is not technically possible to run more than one of these products on the same system.

## 6.4.1 How does GPFS Work?



RS/6000

How Does GPFS Work?

- GPFS uses Virtual Shared Disk (VSD) as its underlying structure, which will be described next
- VSD has been part of PSSP for some time, but has only really been exploited by the Oracle Parallel Database
- GPFS also requires Recoverable VSD, which is a separate LPP (not part of PSSP)
- There are enhancements to VSD and RVSD in PSSP 2.3 which will be covered later
- GPFS depends on the high-availability infrastructure and in particular on Group Services, which is part of PSSP

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

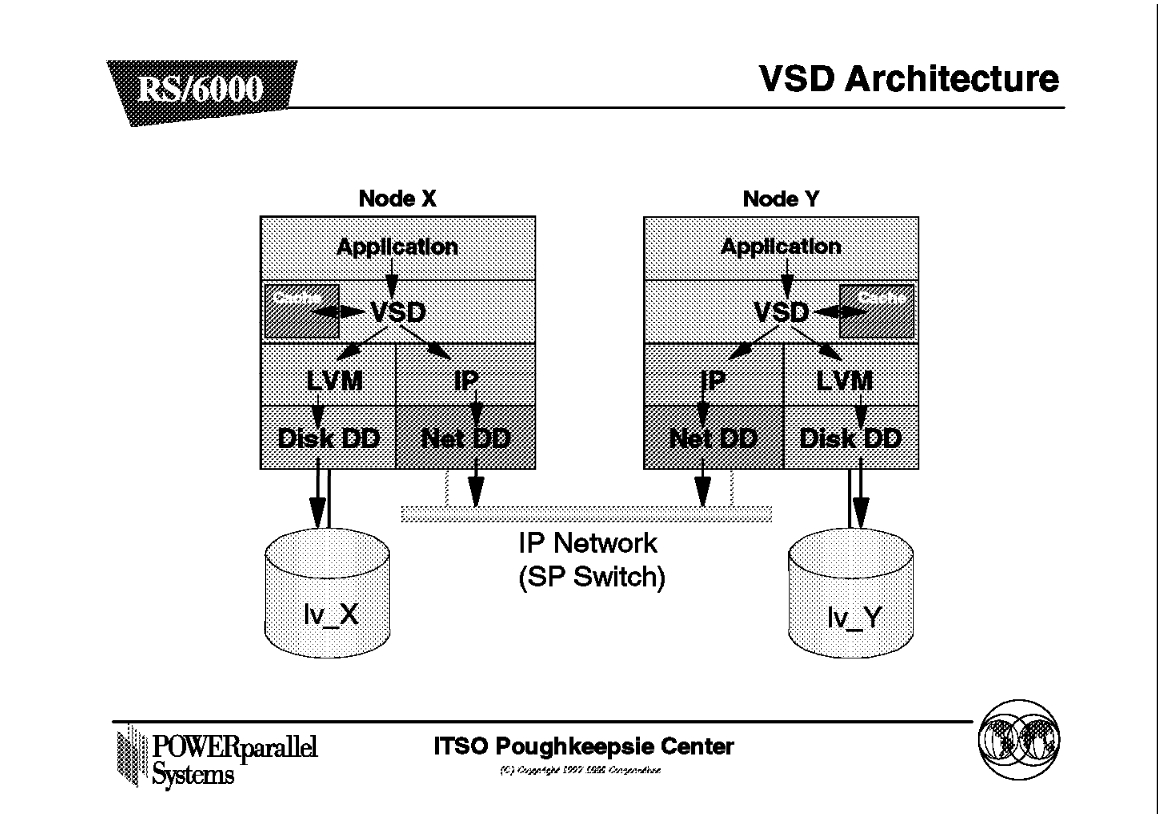
For the access to remote disks, GPFS uses the tried and tested Virtual Shared Disk (VSD) that has been part of PSSP for a long time.

VSD will be explained in more detail in this chapter.

GPFS is only supported with PSSP 2.3, and a new version of VSD ships with PSSP 2.3.

Similarly, Recoverable VSD is used by GPFS. It is required even if twin-tailing of disks is not required. GPFS uses the new node fencing capability that requires RVSD.

## 6.4.2 VSD Architecture



RS/6000

VSD Architecture

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This diagram shows the structure that allows VSD to gain access to remote disks within the SP system.

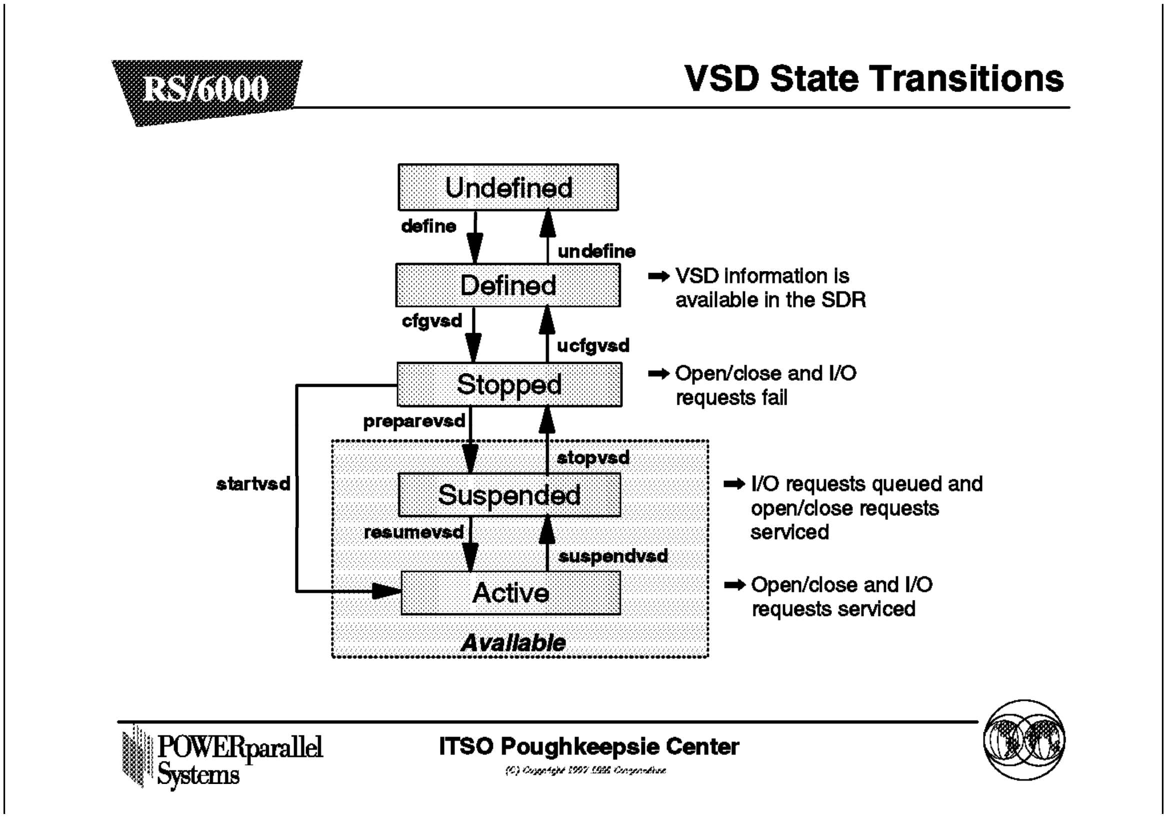
Each disk in an SP is physically cabled to only one node and can only be accessed directly by that node.

VSD allows access across the SP switch to this disk from a remote node. This is achieved by the application communicating with the VSD device driver rather than a disk device driver. The VSD device driver can reroute the I/O request to a remote node if required. Local disk activity occurs in the usual way after going through the VSD device driver.

The VSD software only gives access. It does not provide a locking mechanism to ensure integrity of the data. In addition, a VSD defines only a Logical Volume, or raw device, and not a file system.

An application such as Oracle Parallel Server is required to provide a global locking mechansim.

## 6.4.3 VSD States



## VSD State Transitions

➡ VSD information is available in the SDR

➡ Open/close and I/O requests fail

➡ I/O requests queued and open/close requests serviced

➡ Open/close and I/O requests serviced

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

A VSD device, which in many ways appears like a disk, can be in one of a number of states, as shown here.

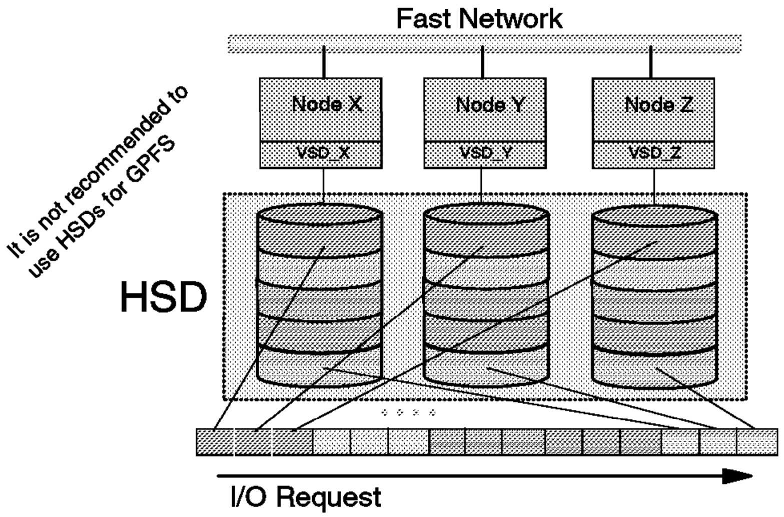
The various states define what operations can be performed on that VSD.

For example, a VSD can be suspended in the event of a failure so that recovery can make it available on another node, where the VSD can be resumed. This will lead to less disruption in the event of a failure from the application point of view.

### 6.4.4 HSDs

RS/6000

HSD Architecture



It is not recommended to use HSDs for GPFS

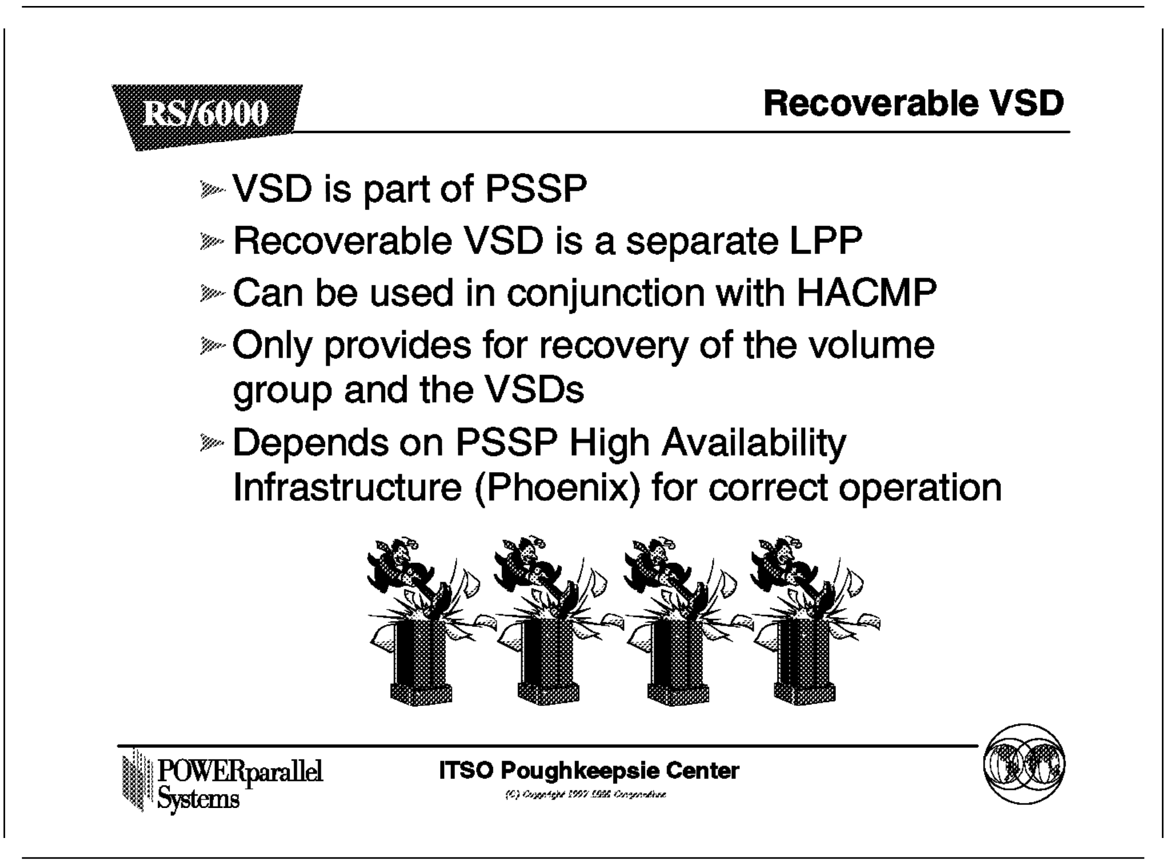
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 1998 Corporation

A Hashed Shared Disk or HSD is a striped version of a VSD. It is recommended that you do not use these with GPFS. GPFS allows for striping itself, and there is no requirement for HSDs. Two levels of striping is not likely to be a good solution.

## 6.4.5 Recoverable VSD



RS/6000

**Recoverable VSD**

- VSD is part of PSSP
- Recoverable VSD is a separate LPP
- Can be used in conjunction with HACMP
- Only provides for recovery of the volume group and the VSDs
- Depends on PSSP High Availability Infrastructure (Phoenix) for correct operation

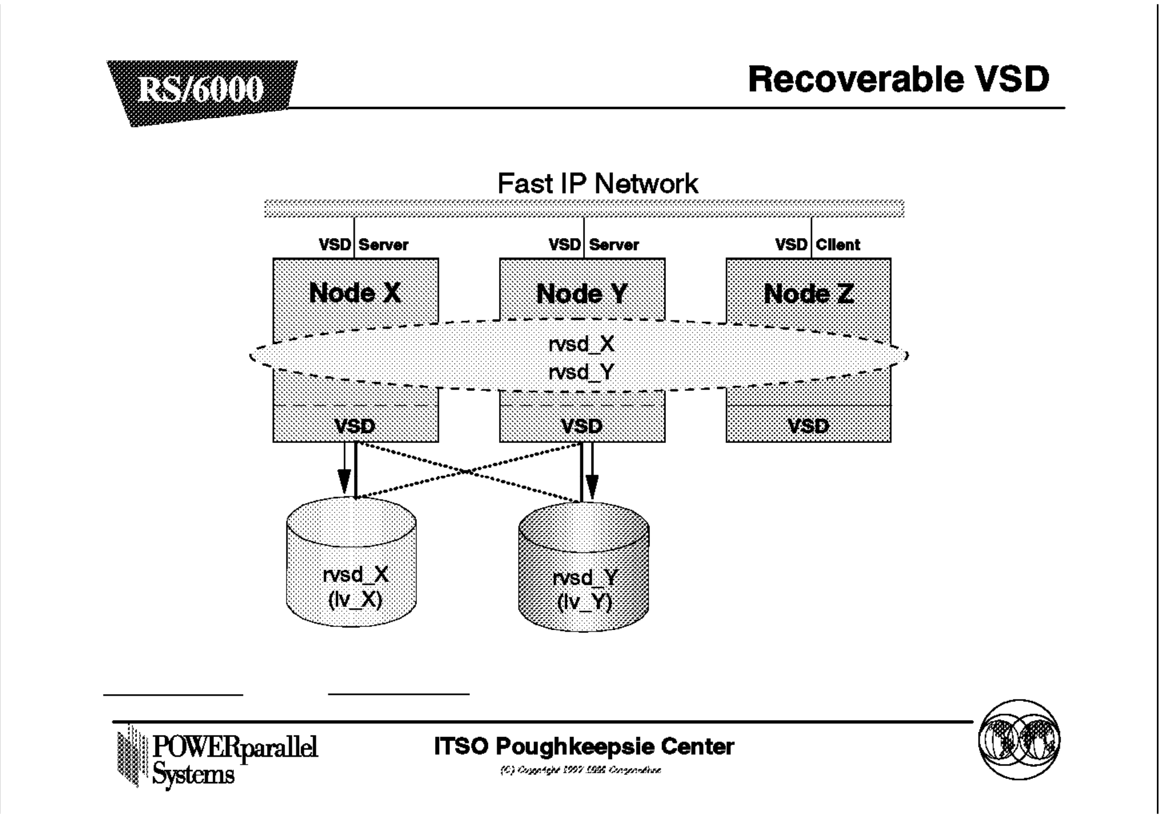
POWERparallel Systems

ITSO Poughkeepsie Center

An optional addition that can be used with VSD is Recoverable VSD or RVSD. RVSD is a separate Licensed Program Product (LPP). It is not part of PSSP. It is normally used in conjunction with twin tailed disks to give high availability for a volume group and the associated VSDs. It is required for GPFS.

RVSD will often be used in conjunction with GPFS, but it is required even if this is not the case.

### 6.4.5.1 Recoverable VSD



RS/6000

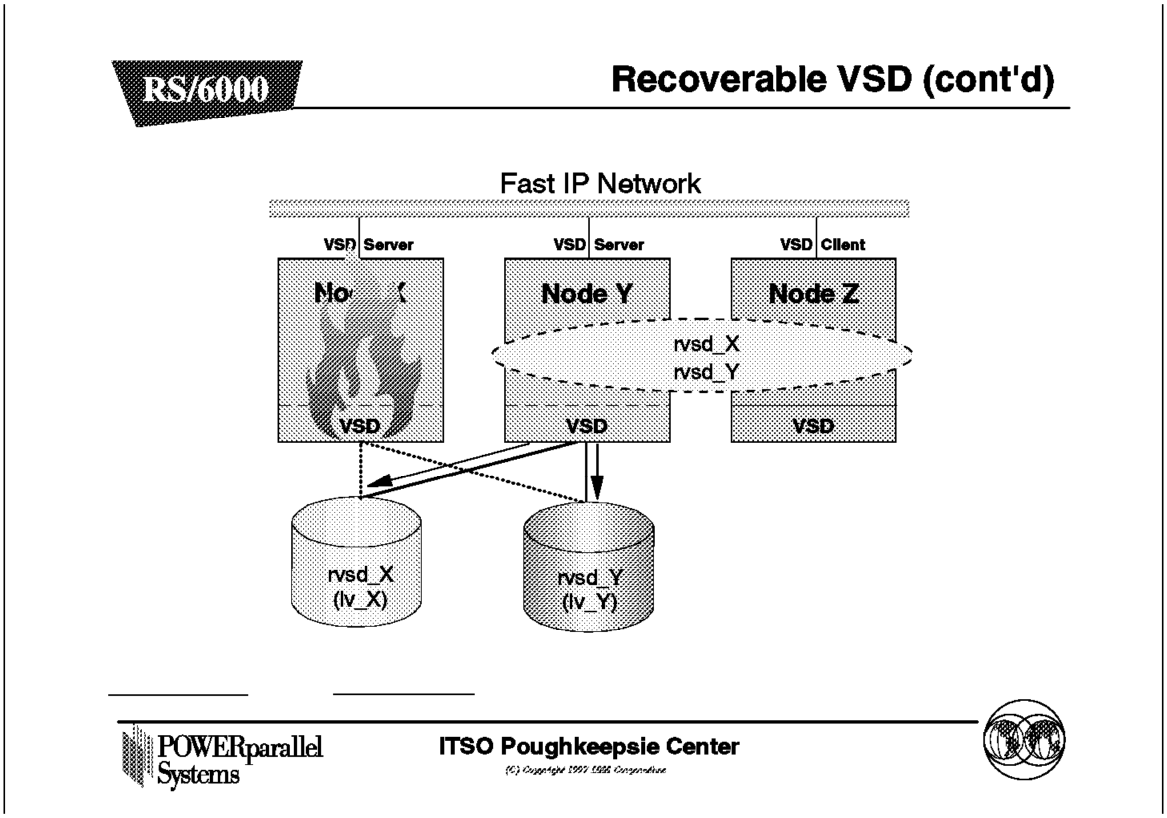
Recoverable VSD

POWERparallel Systems

ITSO Poughkeepsie Center

This diagram shows the normal operation of a group of nodes with VSD. RVSD is being used to protect the VSD server Node X. In the event of failure of Node X, Node Y will act as the secondary server and take over the volume group.

### 6.4.5.2 Recoverable VSD



RS/6000

Recoverable VSD (cont'd)

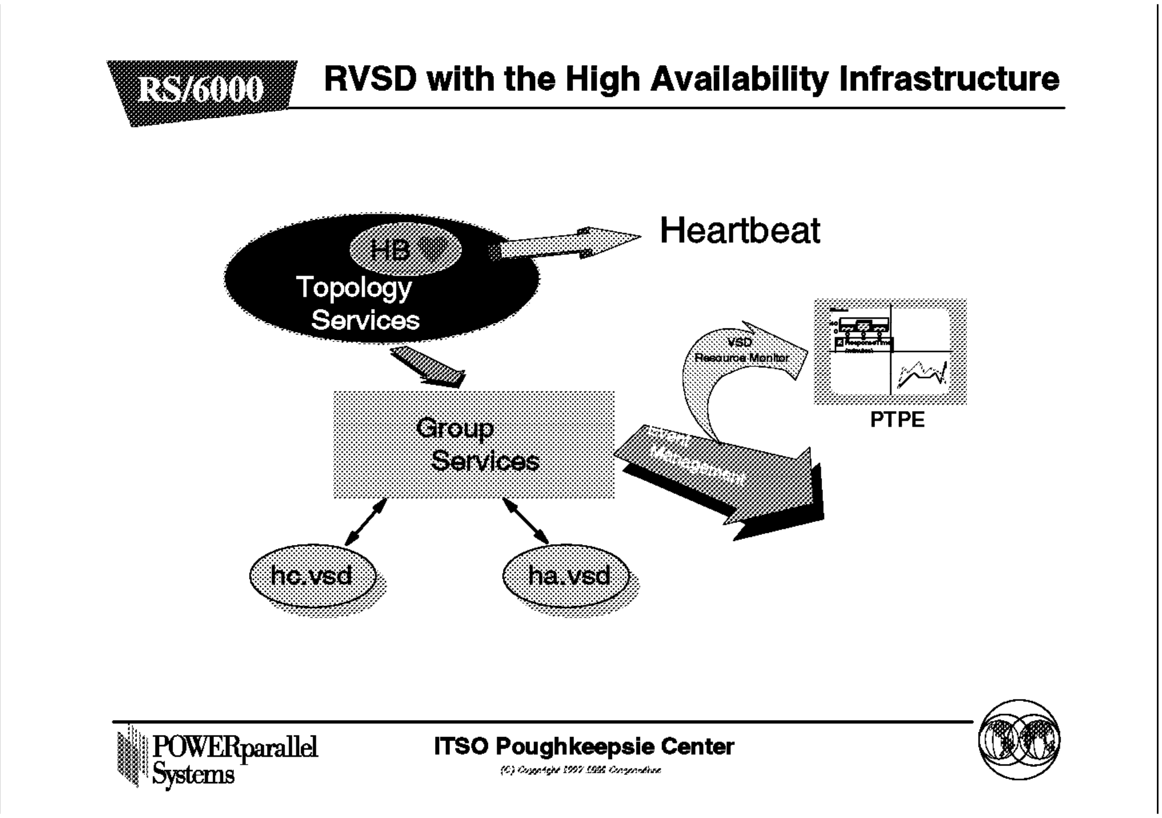
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In this diagram, we see that Node X has failed. Node Y has taken ownership of the volume group that is twin-tailed and will varyon the volume group and make the VSDs available. In addition, RVSD will communicate with Group Services to inform other applications, such as Oracle, of the failure and recovery.

#### 6.4.5.3 Recoverable VSD



RS/6000 **RVSD with the High Availability Infrastructure**

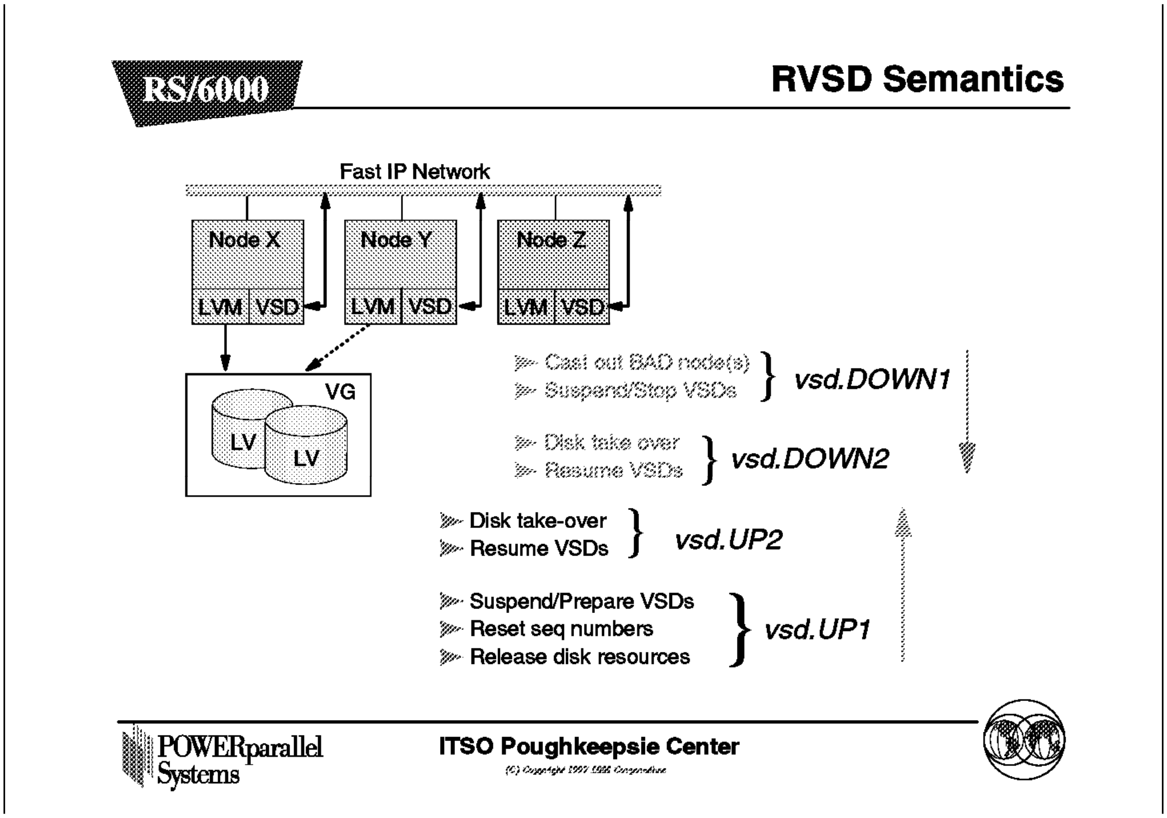
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 [illegible] Corporation

Group Services is part of the high availability infrastructure and has membership groups related to RVSD. The failure of a node, for example will be communicated to RVSD and VSD via Group Services, and recovery can safely be completed before the VSD is resumed and made available again.

### 6.4.5.4 Recoverable VSD



RS/6000

**RVSD Semantics**

- Cast out BAD node(s)
- Suspend/Stop VSDs

} *vsd.DOWN1*

- Disk take over
- Resume VSDs

} *vsd.DOWN2*

- Disk take-over
- Resume VSDs

} *vsd.UP2*

- Suspend/Prepare VSDs
- Reset seq numbers
- Release disk resources

} *vsd.UP1*

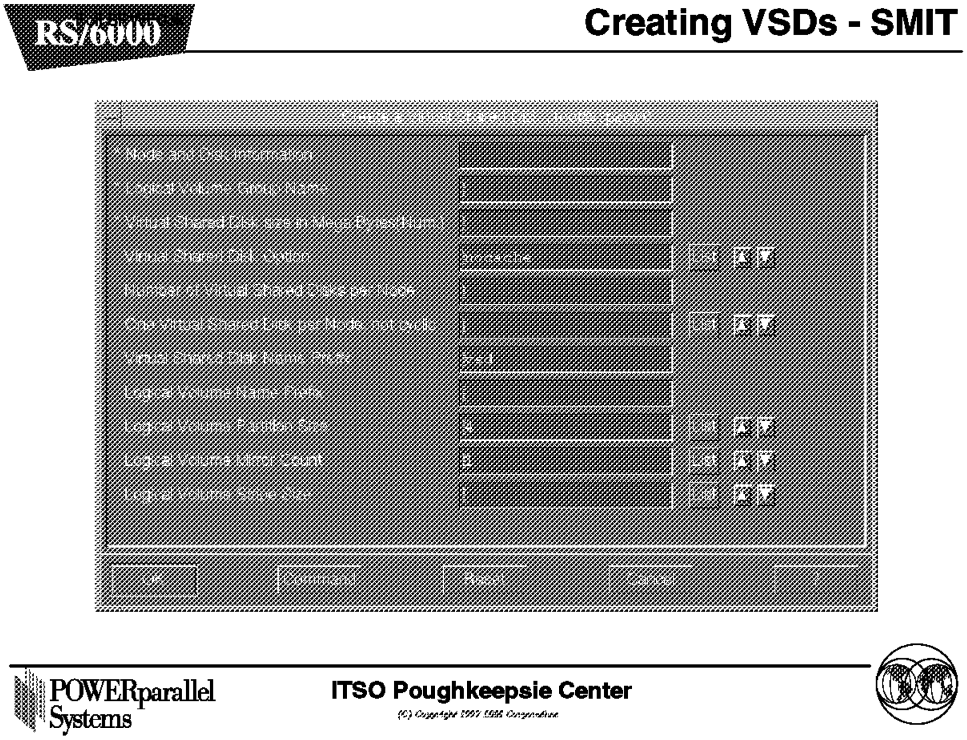
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

A sophisticated process is followed by RVSD in the event of failure to cater to a second or third failure during recovery. Under no circumstances should the data be corrupted, so such a process is necessary to ensure successful recovery.

### 6.4.6 Creating VSDs



RS/6000

**Creating VSDs - SMIT**

* Node and Disk Information

* Logical Volume Group Name

* Virtual Shared Disk size in Mega Bytes(Num.)

Virtual Shared Disk Option: nocache

Number of Virtual Shared Disks per Node

One Virtual Shared Disk per Node: not cyclic

Virtual Shared Disk Name Prefix: vsd

Logical Volume Name Prefix

Logical Volume Partition Size: 4

Logical Volume Mirror Count: 1

Logical Volume Stripe Size

ITSO Poughkeepsie Center

(C) Copyright 1997 1998 Corporation

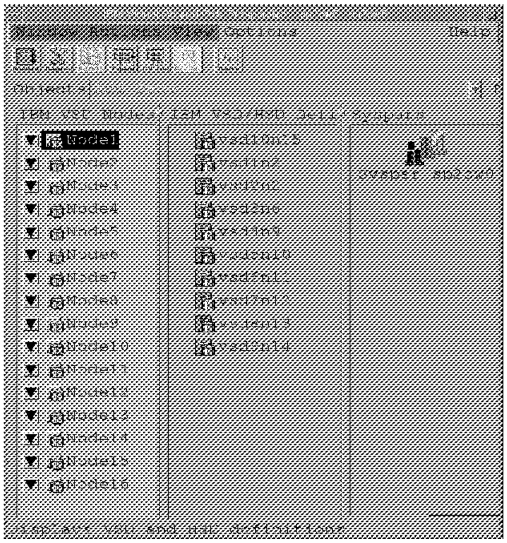
It is quite straightforward to create VSDs on logical volumes within the SP system. GPFS will do this for you; using GPFS interfaces and commands is the preferred route.

If, however, you are familiar with creating and managing VSDs, it is supported that you create your own VSDs and then define them to GPFS.

## 6.4.7 Managing VSDs

RS/6000

### Managing VSDs - Perspectives



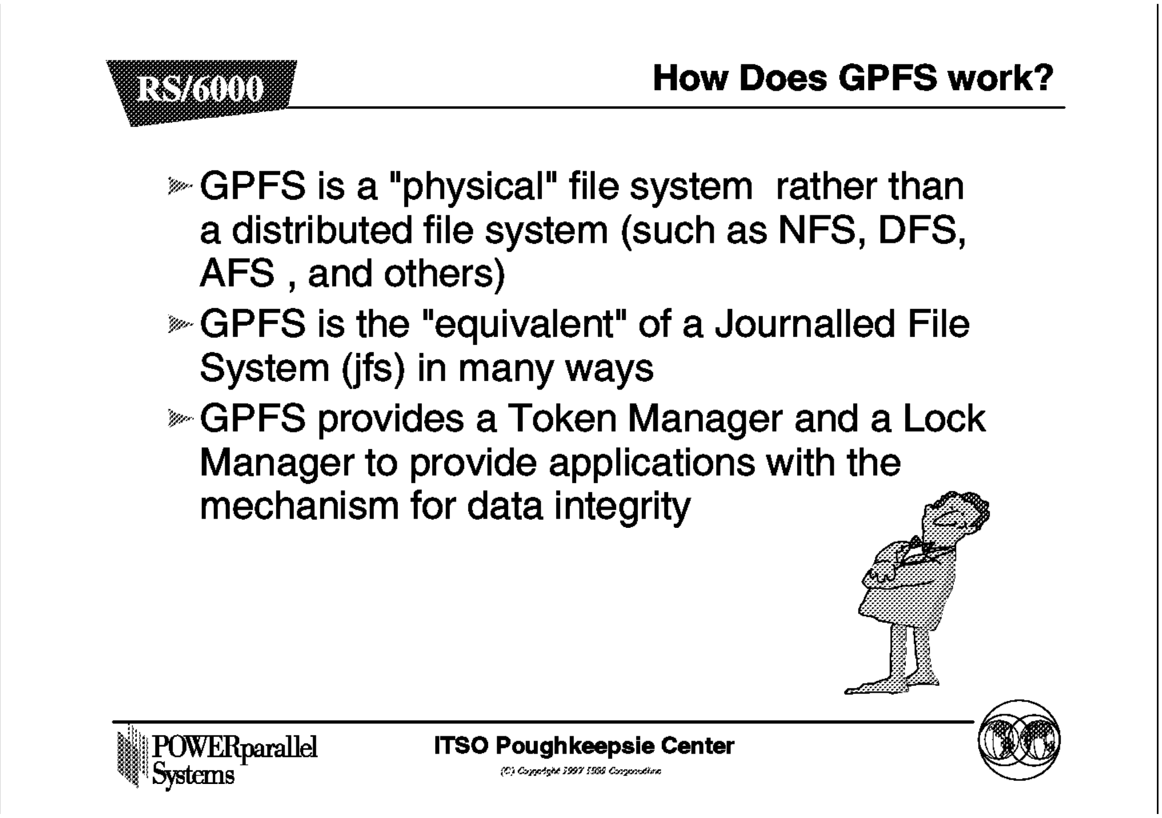
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The PSSP Graphical User Interface, Perspectives, should be used for managing your nodes and VSDs.

### 6.4.8 How Does GPFS Work?



RS/6000

How Does GPFS work?

- GPFS is a "physical" file system rather than a distributed file system (such as NFS, DFS, AFS , and others)
- GPFS is the "equivalent" of a Journalled File System (jfs) in many ways
- GPFS provides a Token Manager and a Lock Manager to provide applications with the mechanism for data integrity

POWERparallel Systems

ITSO Poughkeepsie Center

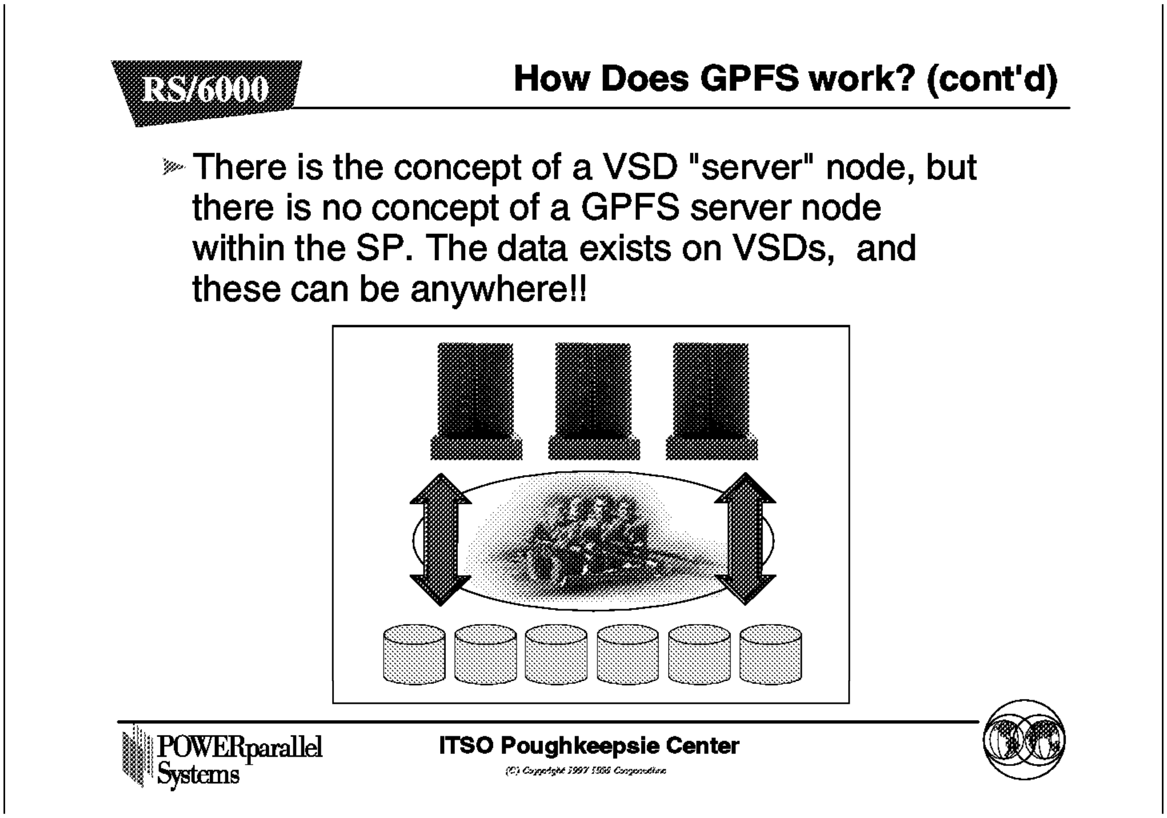
(C) Copyright 1997 IBM Corporation

GPFS may look similar to NFS in some ways, but is actually very different. It does not have a single server and lots of clients, but, instead, is a parallel file system that is normally striped across a number of nodes.

As we shall see, it has a number of facilities to provide for high availability. It is not a Journaled File System (jfs) in AIX terminology, but provides the same functions for recovery through a different method.

It provides a locking mechanism for applications to prevent data corruption.

### 6.4.8.1 How does GPFS Work?



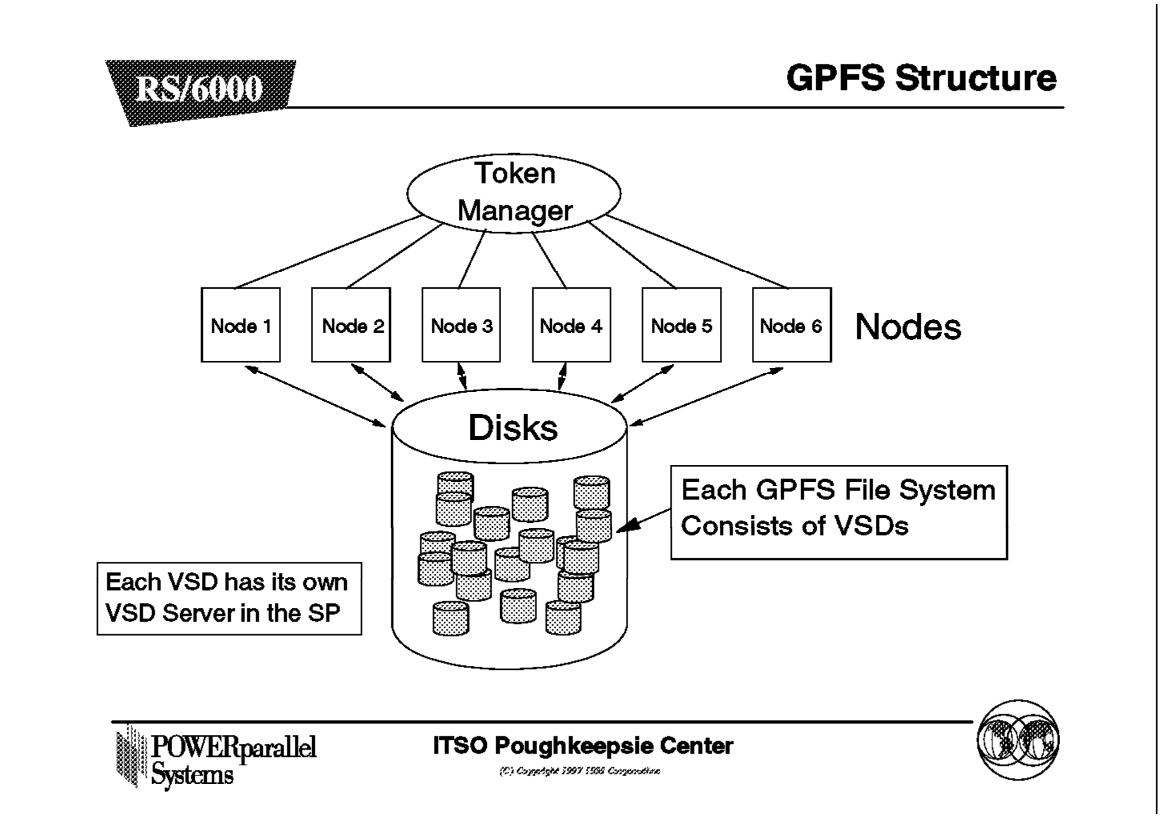
**How Does GPFS work? (cont'd)**

- There is the concept of a VSD "server" node, but there is no concept of a GPFS server node within the SP. The data exists on VSDs, and these can be anywhere!!

ITSO Poughkeepsie Center

Any GPFS node can be a VSD server. Equally any node can have access to the data. So there is no concept of a GPFS server node. A node that is a server for some disks is actually a VSD server node.

### 6.4.9 GPFS Structure



RS/6000

GPFS Structure

Each GPFS File System Consists of VSDs

Each VSD has its own VSD Server in the SP

POWERparallel Systems

ITSO Poughkeepsie Center

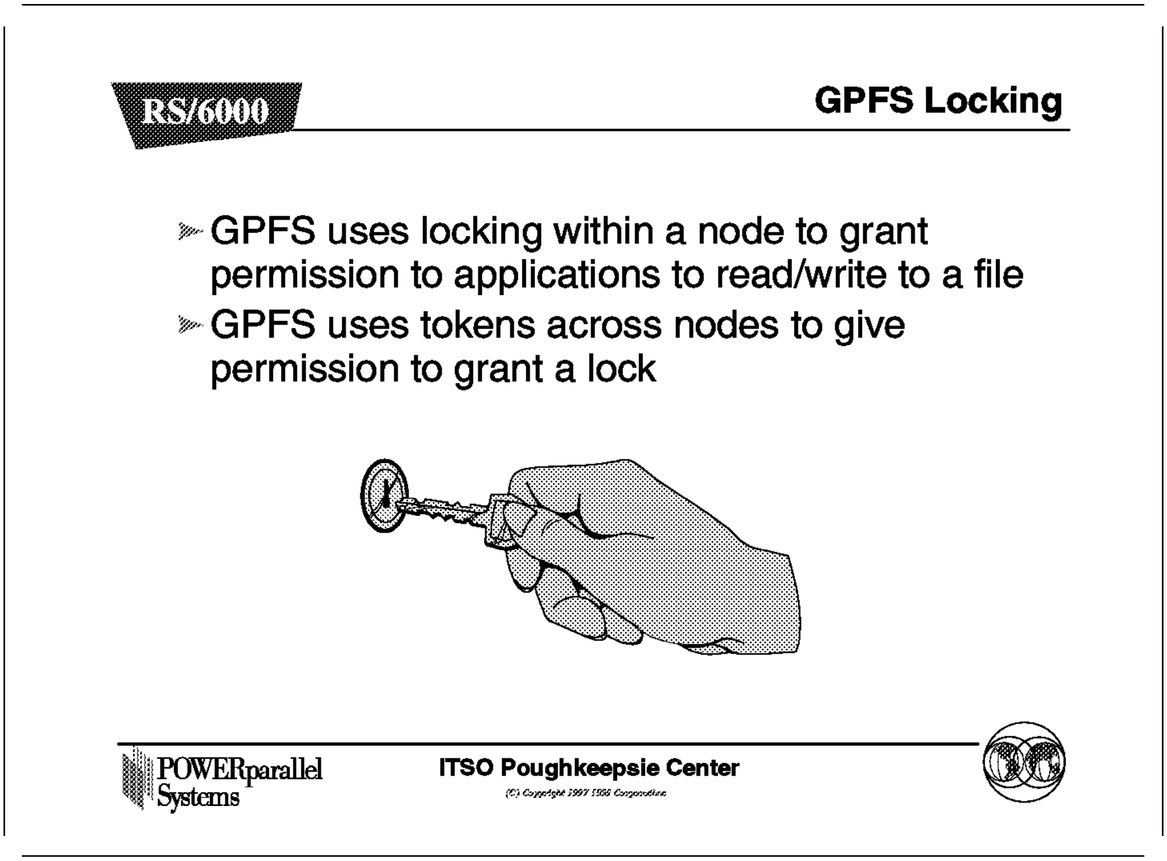
(C) Copyright 1997 IBM Corporation

This diagram shows the overall structure of GPFS. Each GPFS file system is made up of a number of disks defined as VSDs. Each VSD can reside on any server node within the SP system. The file system writes data by striping across all of these VSDs.

In addition, any GPFS node can mount this file system and have access to the same data.

A Token Manager controls the locking within the GPFS file system.

## 6.4.10 GPFS Locking



RS/6000

GPFS Locking

- GPFS uses locking within a node to grant permission to applications to read/write to a file
- GPFS uses tokens across nodes to give permission to grant a lock

POWERparallel Systems

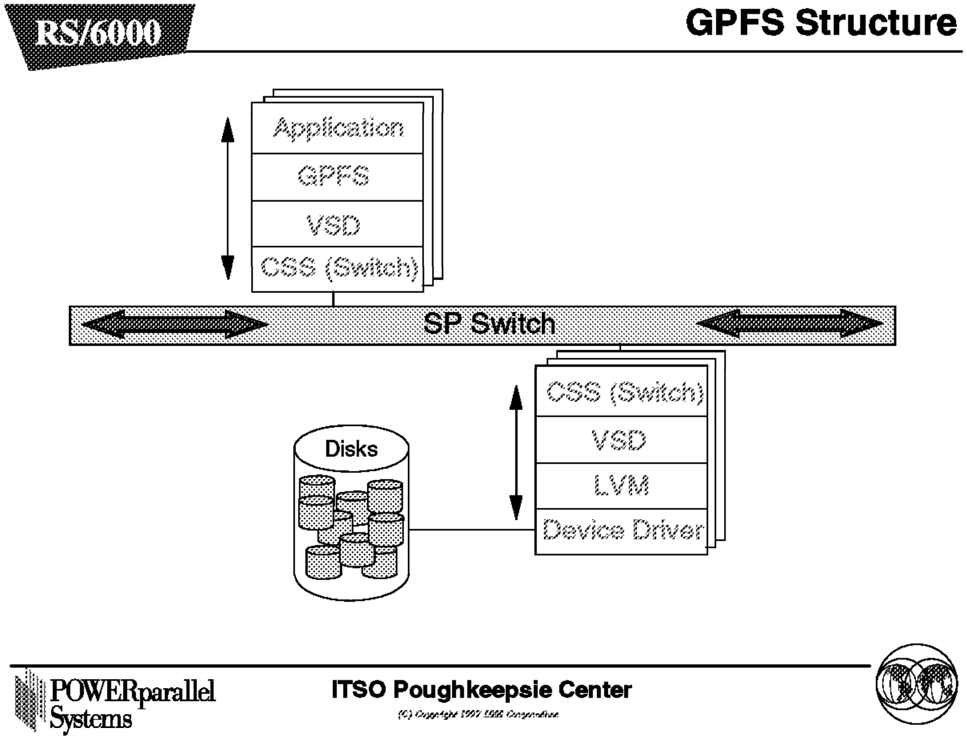
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The locking process will be discussed in more detail later in this chapter. GPFS provides the locking mechanism that VSD does not have.

This allows GPFS file systems to be seen in almost every way as normal file systems.

### 6.4.11 GPFS Structure



RS/6000

GPFS Structure

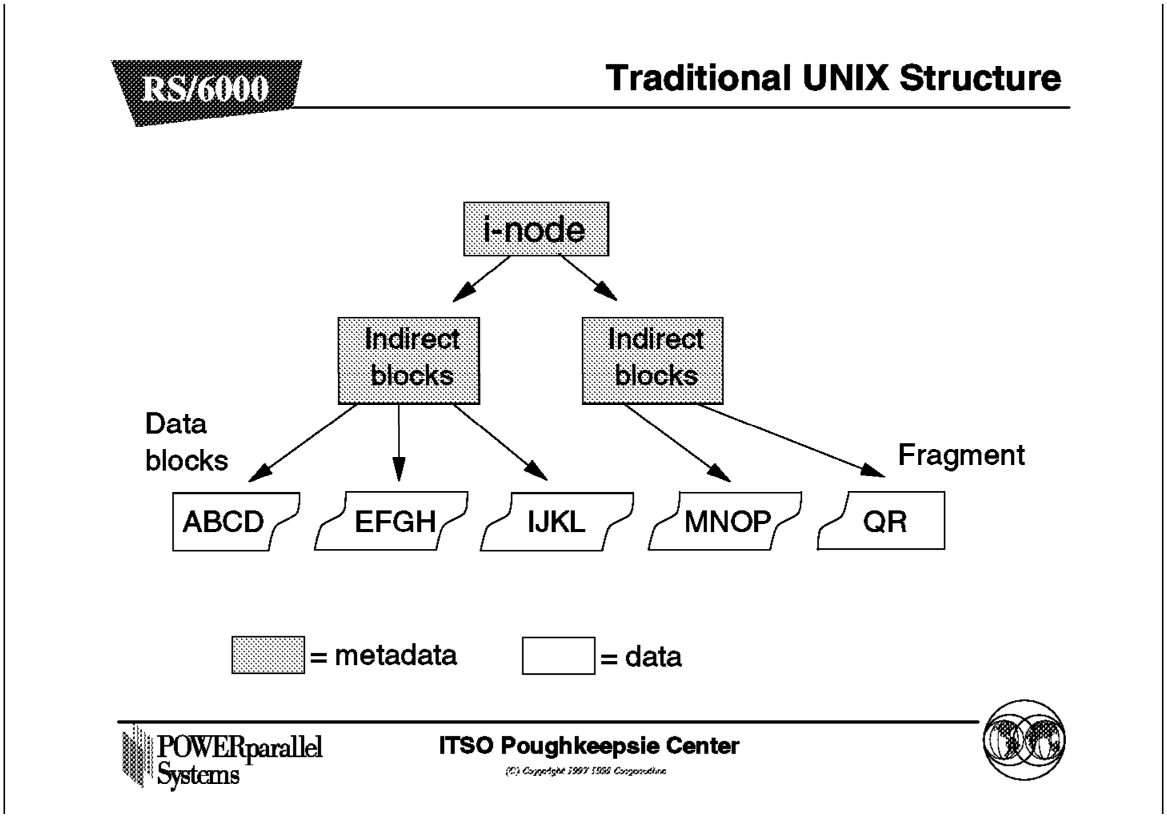
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

GPFS uses VSD to provide local or remote access to disks, as shown.

## 6.4.12 Traditional UNIX Structure



RS/6000

Traditional UNIX Structure

i-node

Indirect blocks

Indirect blocks

Data blocks

Fragment

ABCD

EFGH

IJKL

MNOP

QR

▨ = metadata

□ = data

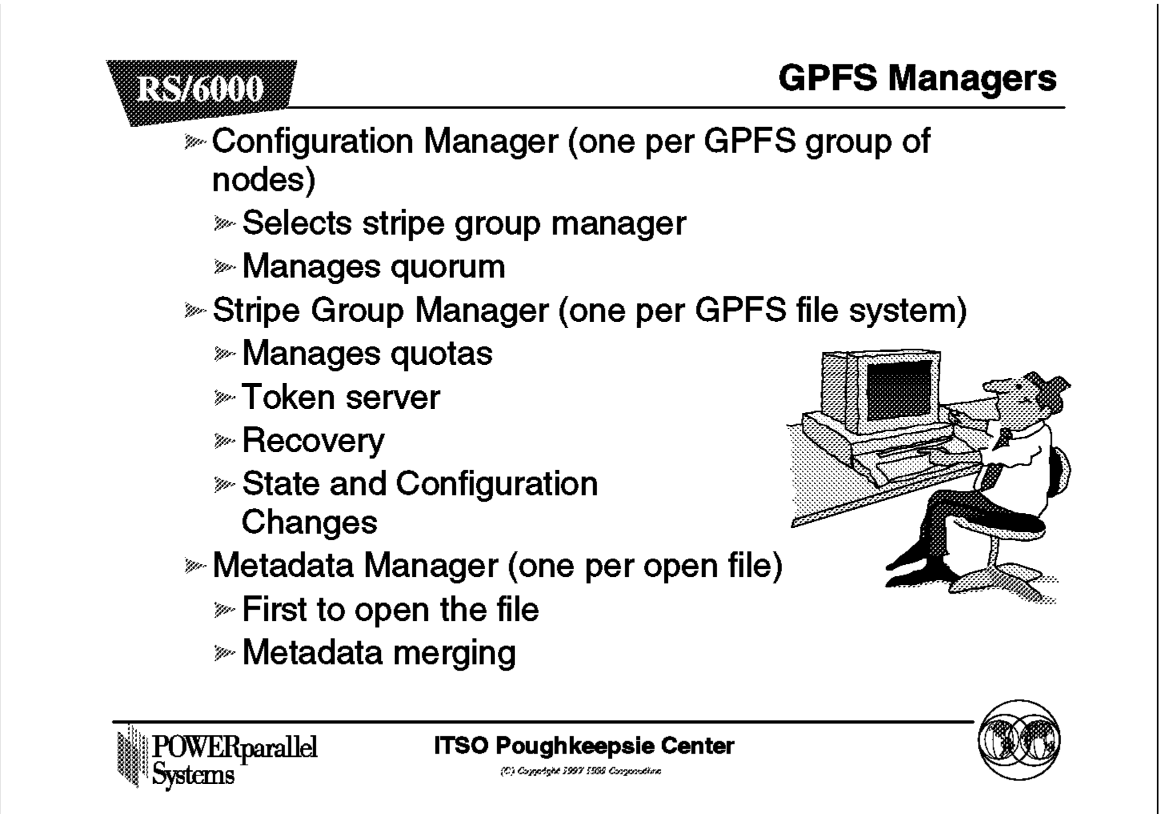
POWERparallel Systems

ITSO Poughkeepsie Center

GPFS uses the traditional way of structuring file systems, as adopted in the UNIX world.

An i-node is a pointer to the actual data on disk. When the amount of data is larger, the i-node will in fact point to another data structure, called an indirect block, that in turn will point to the actual addresses of the data on disk.

### 6.4.13 GPFS Managers



RS/6000

**GPFS Managers**

- Configuration Manager (one per GPFS group of nodes)
  - Selects stripe group manager
  - Manages quorum
- Stripe Group Manager (one per GPFS file system)
  - Manages quotas
  - Token server
  - Recovery
  - State and Configuration Changes
- Metadata Manager (one per open file)
  - First to open the file
  - Metadata merging

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

To control the operation of GPFS, it internally assigns managers or nodes that control the running of GPFS. The normal user will be unaware of these managers, but to understand how GPFS works, we will examine them in some detail in this chapter.

One Configuration Manager runs within any particular pool of GPFS nodes.

Multiple pools of nodes can be run side by side within an SP system. This capability is unrelated to SP system partitions.

The Configuration Manager has responsibility for monitoring and managing quorum. It also selects a Stripe Group Manager for each GPFS filesystem. If this node fails, it will be replaced by another and that node will take over the responsibility of the Configuration Manager. The first GPFS node to start GPFS will assume the role of the Configuration Manager.

The PSSP High Availability Infrastructure, will work closely with GPFS to help manage recovery in the event of failure.

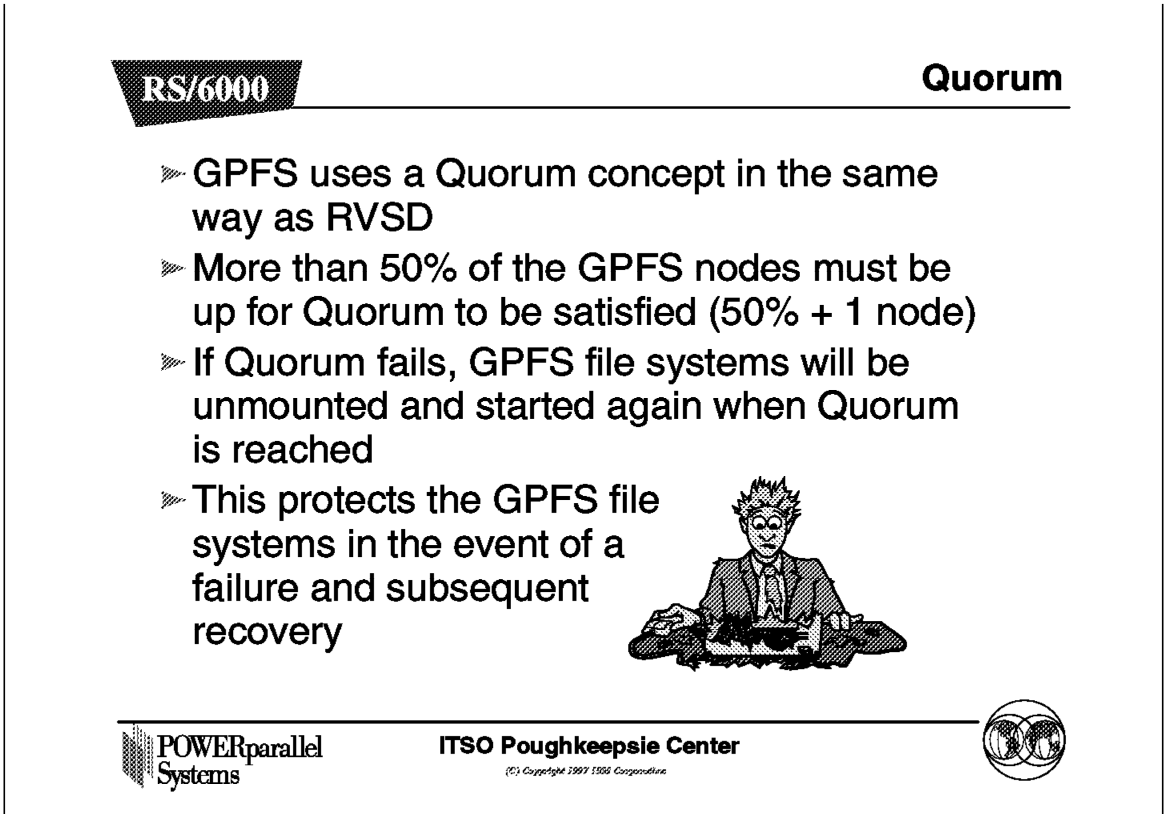
The Stripe Group Managers are distributed across the GPFS nodes. There will be one Stripe Group Manager per GPFS file system. The term Stripe Group really describes a GPFS File System.

Which node becomes Stripe Group Manager will be determined by the Configuration Manager for each file system as it is created.

The Stripe Group Manager is responsible for the locking of files across the GPFS system. This will be described later.

Finally, a Metadata Manager will be assigned for each open file.

### 6.4.14 Quorum



RS/6000

**Quorum**

- GPFS uses a Quorum concept in the same way as RVSD
- More than 50% of the GPFS nodes must be up for Quorum to be satisfied (50% + 1 node)
- If Quorum fails, GPFS file systems will be unmounted and started again when Quorum is reached
- This protects the GPFS file systems in the event of a failure and subsequent recovery

POWERparallel Systems

ITSO Poughkeepsie Center

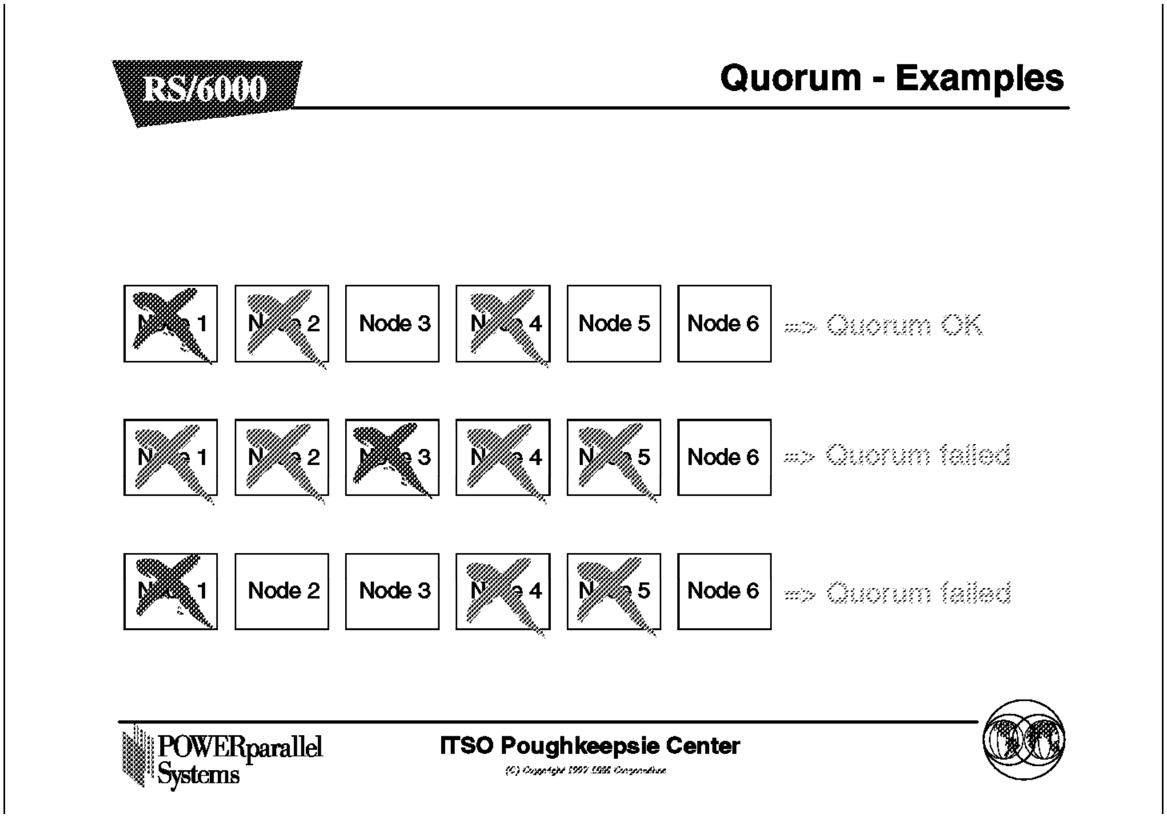
(C) Copyright 1997 IBM Corporation

Quorum is used by GPFS to make sure that no unexpected actions occur during a failure within the SP system.

Quorum ensures that the GPFS group of nodes does not split into two separate groups following a failure of some kind. If there were two separate groups of nodes within the GPFS pool of nodes, there would be two Token Managers controlling file locking. This could have disastrous consequences, with data getting corrupted.

To ensure that this cannot happen and that recovery is carefully controlled and guaranteed, Quorum will not allow GPFS to use a file system if less than half of the nodes are not operational.

## 6.4.15 Quorum Examples



RS/6000

### Quorum - Examples

~~Node 1~~ ~~Node 2~~ Node 3 ~~Node 4~~ Node 5 Node 6 => Quorum OK

~~Node 1~~ ~~Node 2~~ ~~Node 3~~ ~~Node 4~~ ~~Node 5~~ Node 6 => Quorum failed

~~Node 1~~ Node 2 Node 3 ~~Node 4~~ ~~Node 5~~ Node 6 => Quorum failed

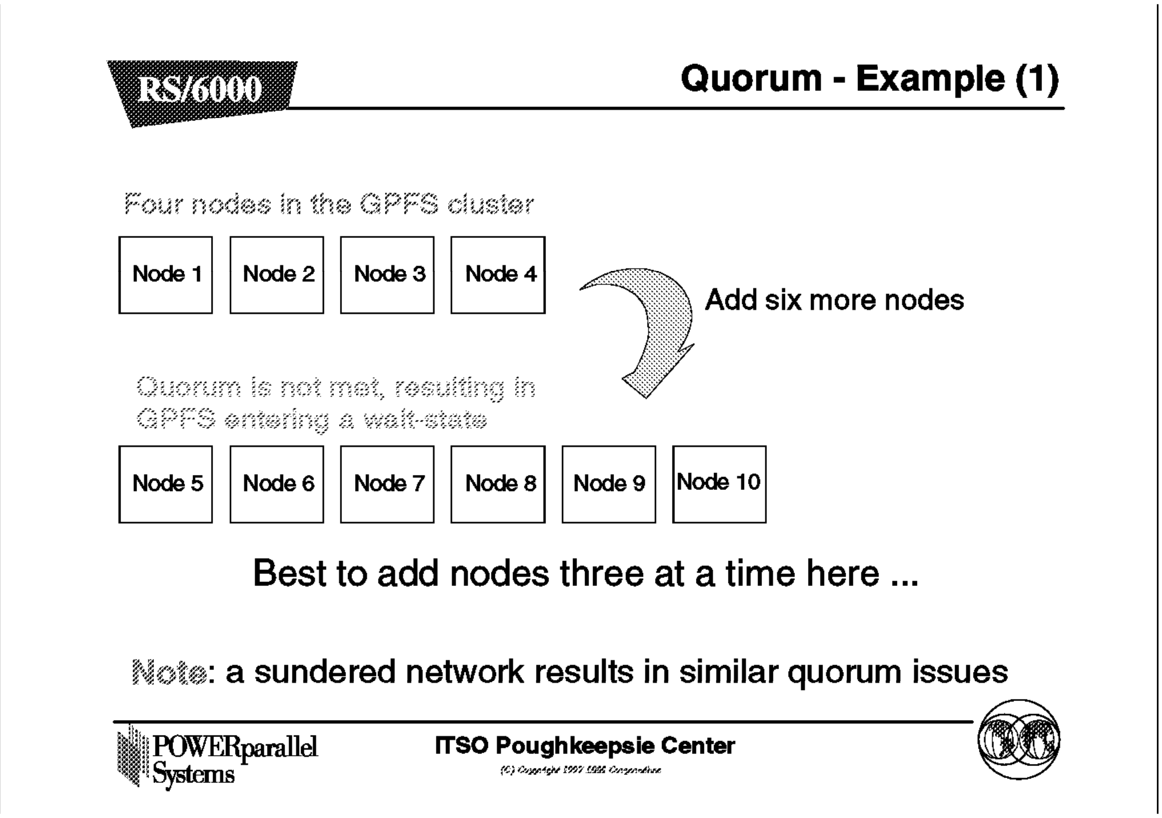
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In these examples, Quorum will be lost even when 50% of the nodes are available, because 50%+1 of the nodes need to be operational for GPFS to allow the file system to be accessed.

### 6.4.15.1 Quorum Examples



RS/6000

**Quorum - Example (1)**

Four nodes in the GPFS cluster

Node 1 Node 2 Node 3 Node 4

Add six more nodes

Quorum is not met, resulting in GPFS entering a wait-state

Node 5 Node 6 Node 7 Node 8 Node 9 Node 10

**Best to add nodes three at a time here ...**

Note: a sundered network results in similar quorum issues

POWERparallel Systems

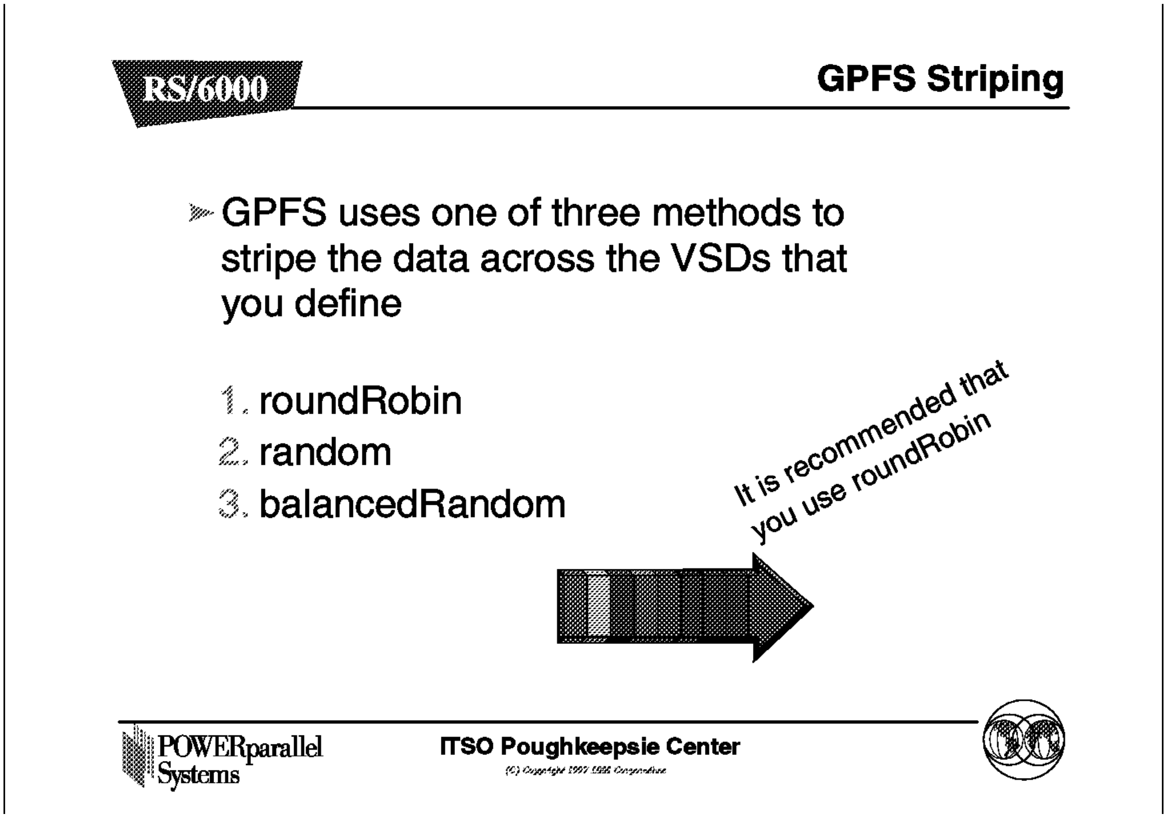
ITSO Poughkeepsie Center

At the time of testing GPFS and creating this chapter, the GPFS Quorum worked in the way described here.

Future enhancements might allow the Quorum to be updated as new nodes are added to GPFS, and for the new nodes to be integrated into the GPFS pool without breaking Quorum.

This would be achieved by ensuring that the new nodes are running correctly first, before integration. Quorum would be satisfied as these operational nodes are then brought into the GPFS pool and activated.

## 6.4.16 GPFS Striping



RS/6000

GPFS Striping

- GPFS uses one of three methods to stripe the data across the VSDs that you define

1. roundRobin
2. random
3. balancedRandom

It is recommended that
you use roundRobin

POWERparallel Systems

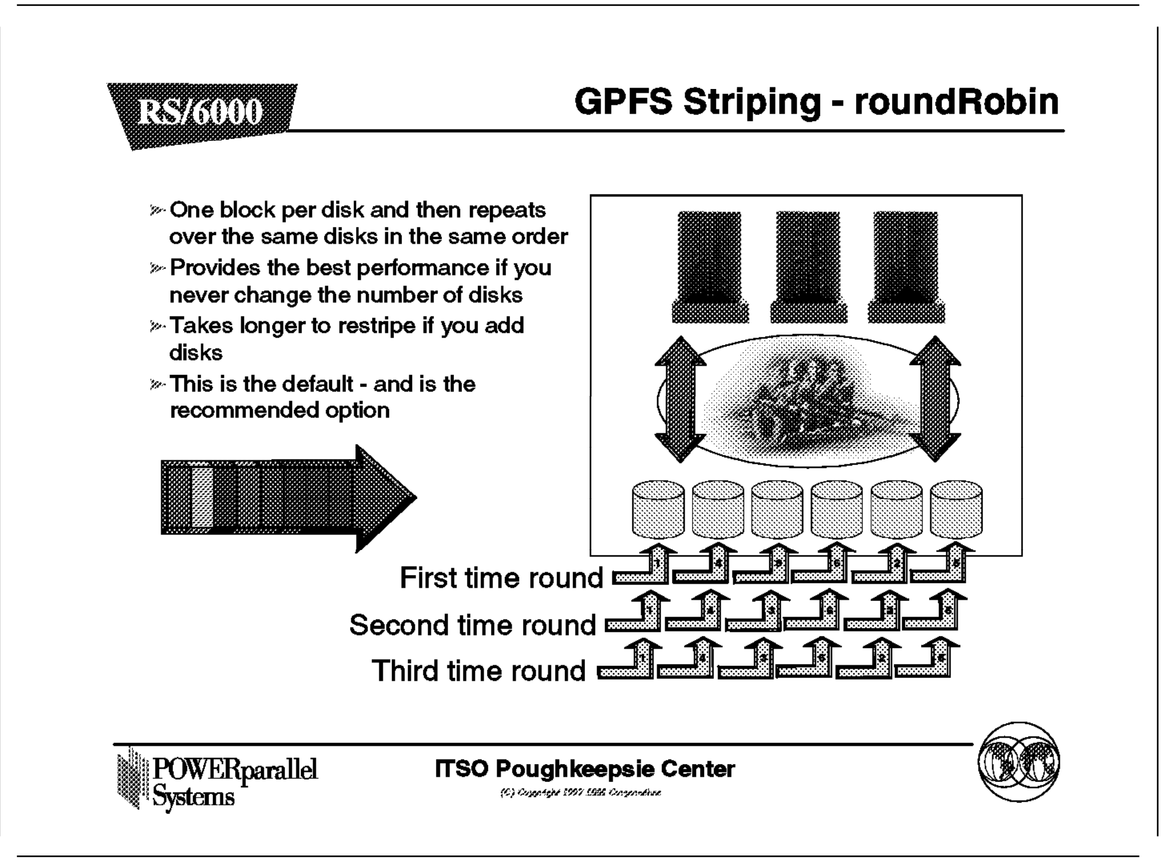
ITSO Poughkeepsie Center

A number of options for GPFS striping exist but the default method, roundRobin, should be the method that you normally use. This is the default selected by GPFS.

It will also be an advantage in most cases, as we will see later, that we use the same number of disks on each node in the GPFS file system and that each of those disks has the same performance and capacity characteristics.

Striping will not use a disk that is offline while the file system is created.

### 6.4.16.1 GPFS Striping



## GPFS Striping - roundRobin

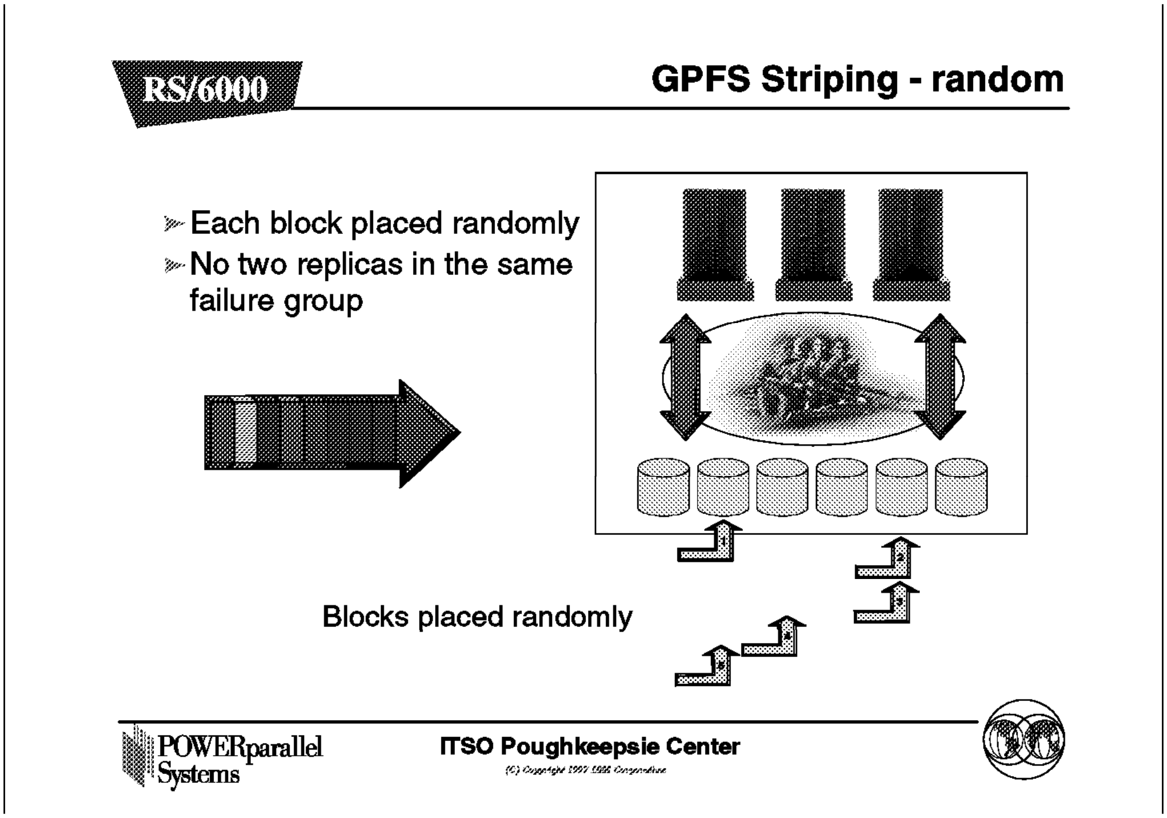
- One block per disk and then repeats over the same disks in the same order
- Provides the best performance if you never change the number of disks
- Takes longer to restripe if you add disks
- This is the default - and is the recommended option

In the case of roundRobin striping, data is written in turn to each disk until all disks have received a block. Then another round of the disks begins, again writing a block each time, and accessing the disks in the same order.

This is the preferred method of striping.

It is clear that equal capacity disks are best in this case.

### 6.4.16.2 GPFS Striping



RS/6000

GPFS Striping - random

- Each block placed randomly
- No two replicas in the same failure group

Blocks placed randomly

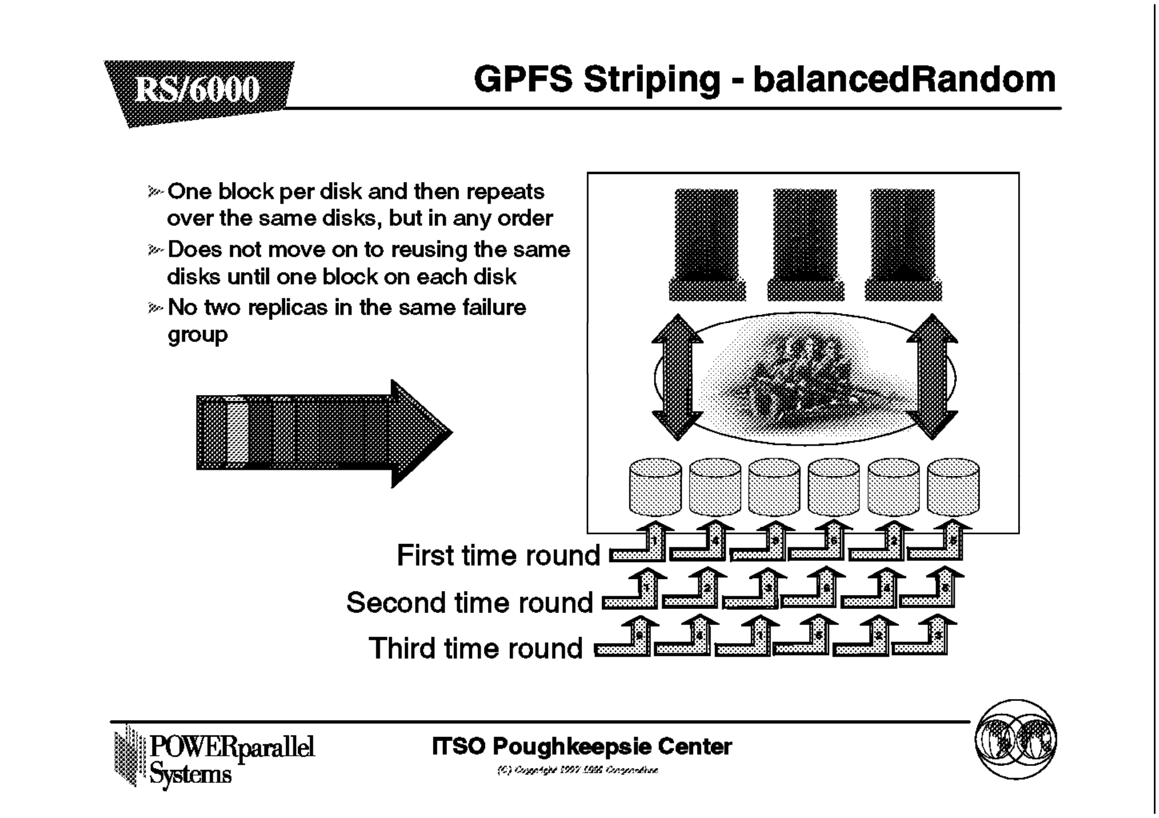
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In the case of random striping, data blocks are placed randomly across the disks in the GPFS nodes. If failure groups are defined the replicated data is stored in separate failure groups to cater for failures.

### 6.4.16.3 GPFS Striping



RS/6000

GPFS Striping - balancedRandom

- One block per disk and then repeats over the same disks, but in any order
- Does not move on to reusing the same disks until one block on each disk
- No two replicas in the same failure group

First time round

Second time round

Third time round

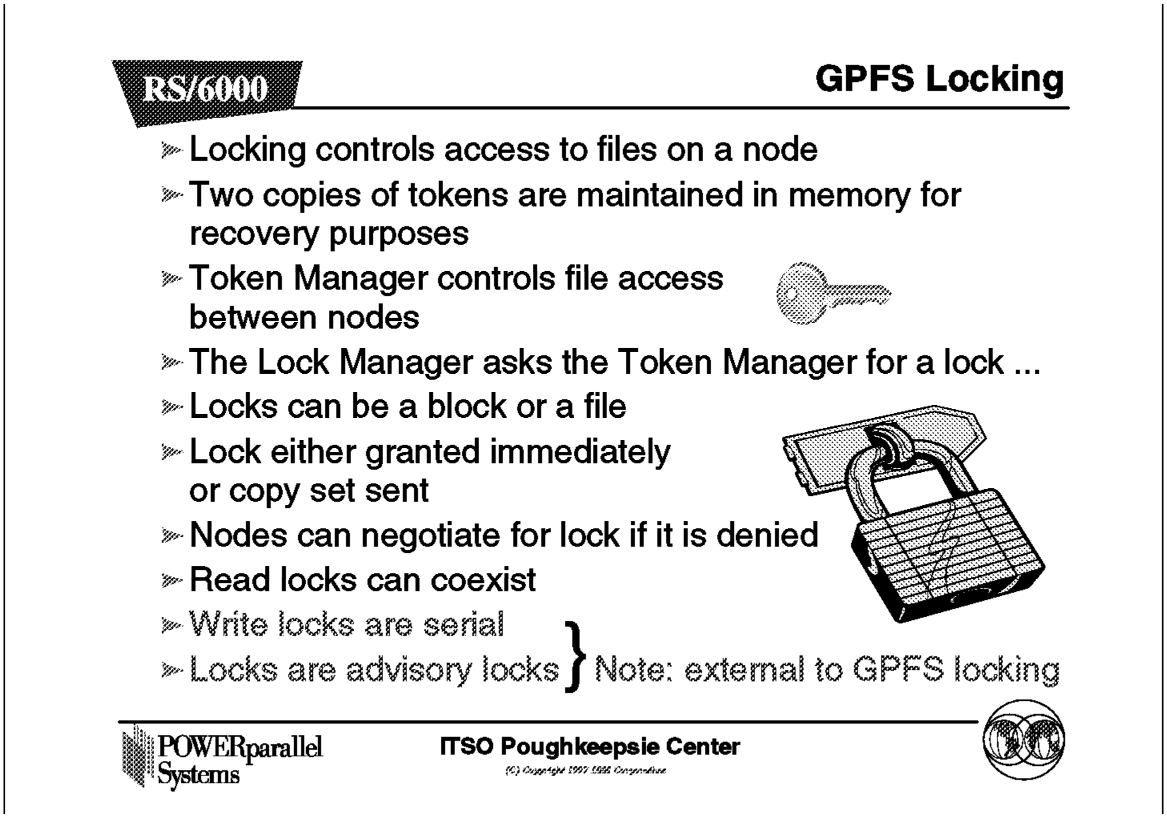
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The balancedRandom striping method writes blocks randomly, but does not return to the same disk until all available disks have been used.

## 6.4.17 GPFS Locking



RS/6000

GPFS Locking

- Locking controls access to files on a node
- Two copies of tokens are maintained in memory for recovery purposes
- Token Manager controls file access between nodes
- The Lock Manager asks the Token Manager for a lock ...
- Locks can be a block or a file
- Lock either granted immediately or copy set sent
- Nodes can negotiate for lock if it is denied
- Read locks can coexist
- Write locks are serial
- Locks are advisory locks

} Note: external to GPFS locking

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Locking controls access to files from a process point of view on any particular node within the GPFS pool. Two copies of the locks are maintained within the GPFS system to allow for recovery in the event of failure. A Lock Manager runs on each node for the purpose of controlling locking on that node.

Each of the two copies of the token are kept in memory but in separate *Failure Groups* within the SP. Failure Groups will be discussed later, but for the moment, we can assume that these copies of locks are kept on separate GPFS nodes.

If the Lock Manager on a particular node is asked for access to a file that exists elsewhere within the SP, then a token needs to be requested from the other node that has the file. This is achieved through the use of tokens. There is one Token Manager for each GPFS file system and this Token Manager (or Stripe Group Manager) runs on one of the nodes within the GPFS pool of nodes.

If other nodes already have the token when it is requested, then the list of nodes in question is passed back to the requesting Lock Manager. This list is called a *copy set*.

It is the responsibility of the requesting Lock Manager to negotiate with nodes in the list to obtain the token.

Under different circumstances, the locking mechanism within GPFS locks different ranges of data.

The request asks for a required range, and also the desired range of data. Depending on who else has locks on sections of the file, a lock may be granted for a larger section of the file. This would improve performance, for example, in the case of a sequential read of a whole file.

There are eight stages to granting a lock, and these cater for contention as well as recovery in the event of failure.

The locks can be read locks or write locks. Read locks are required, for example, so that an application can know whether it can delete a file. It may well be that a file that is being read cannot be deleted.

Multiple concurrent read locks can be granted, whereas write locks are sequential.

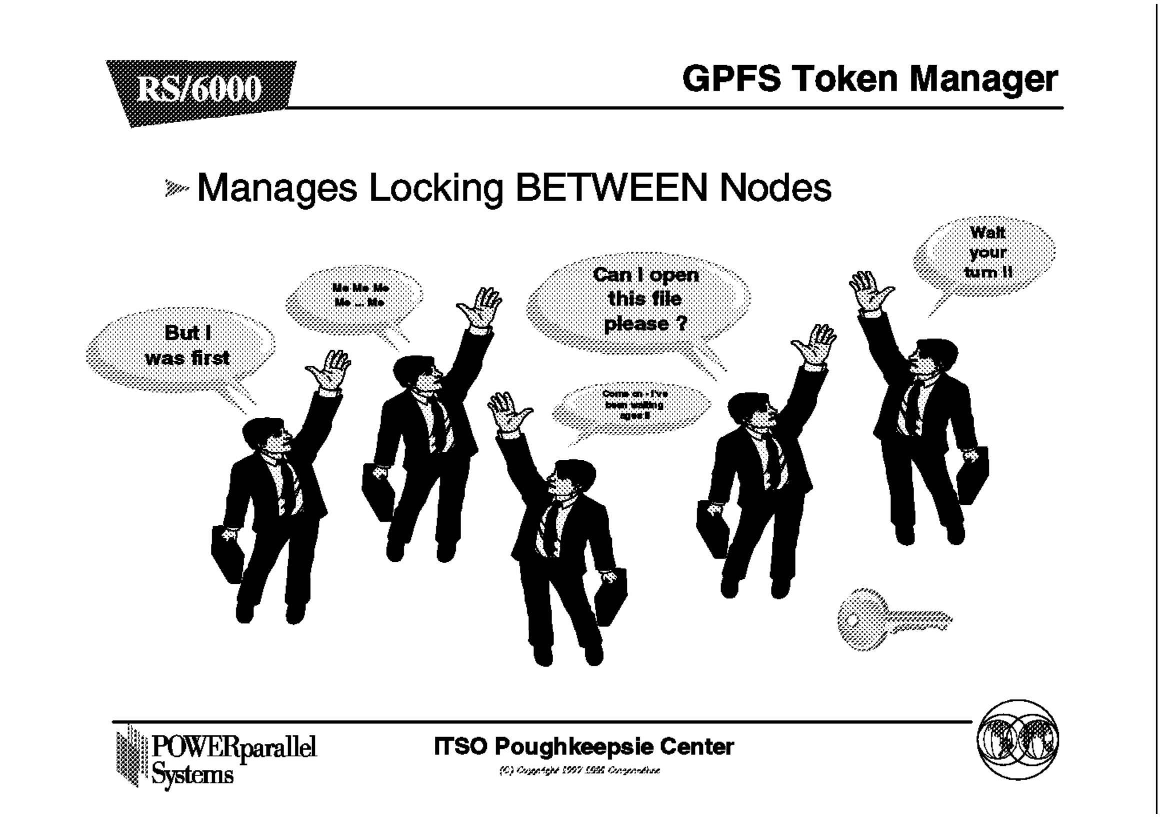
In the event of failure of a node, its logs are replayed before any locks are released to ensure integrity of the data.

In the event of a Stripe Group Manager failure, all the tokens that exist on other nodes can be retrieved to enable the new Stripe Group Manager to have up-to-date information.

The above discussion refers to the internal locking mechanisms within GPFS and not the application locking, such as lockf calls, which are external to GPFS.

The statement that locks are advisory refers to the external locking mechanism.

### 6.4.18 GPFS Token Manager



RS/6000

GPFS Token Manager

- Manages Locking BETWEEN Nodes

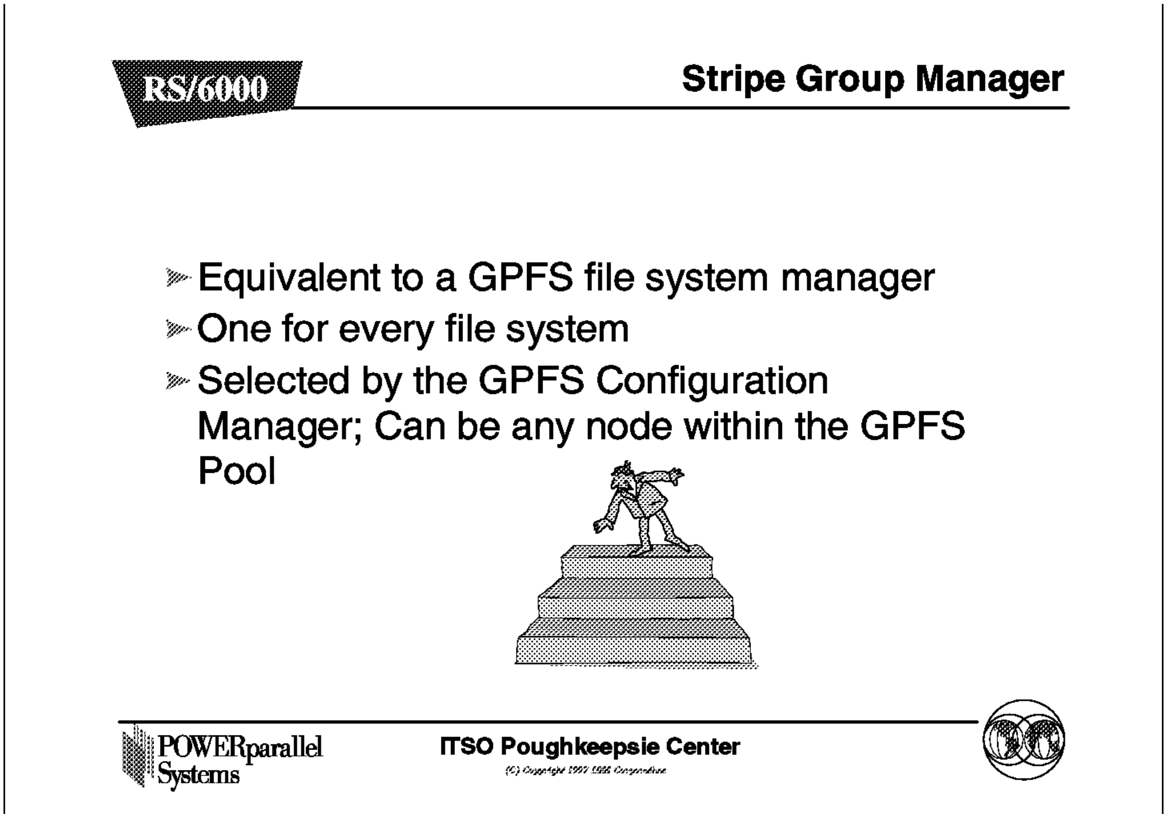
But I was first
Me Me Me Me ... Me
Can I open this file please ?
Come on - I've been waiting ages!!
Wait your turn !!

POWERparallel Systems

ITSO Poughkeepsie Center

As described above, the Token Manager is responsible for looking after the system-wide locking aspects and grants tokens to the Lock Managers on each node as requested and when available.

### 6.4.19 GPFS Stripe Group Manager



RS/6000

**Stripe Group Manager**

- Equivalent to a GPFS file system manager
- One for every file system
- Selected by the GPFS Configuration Manager; Can be any node within the GPFS Pool

POWERparallel Systems

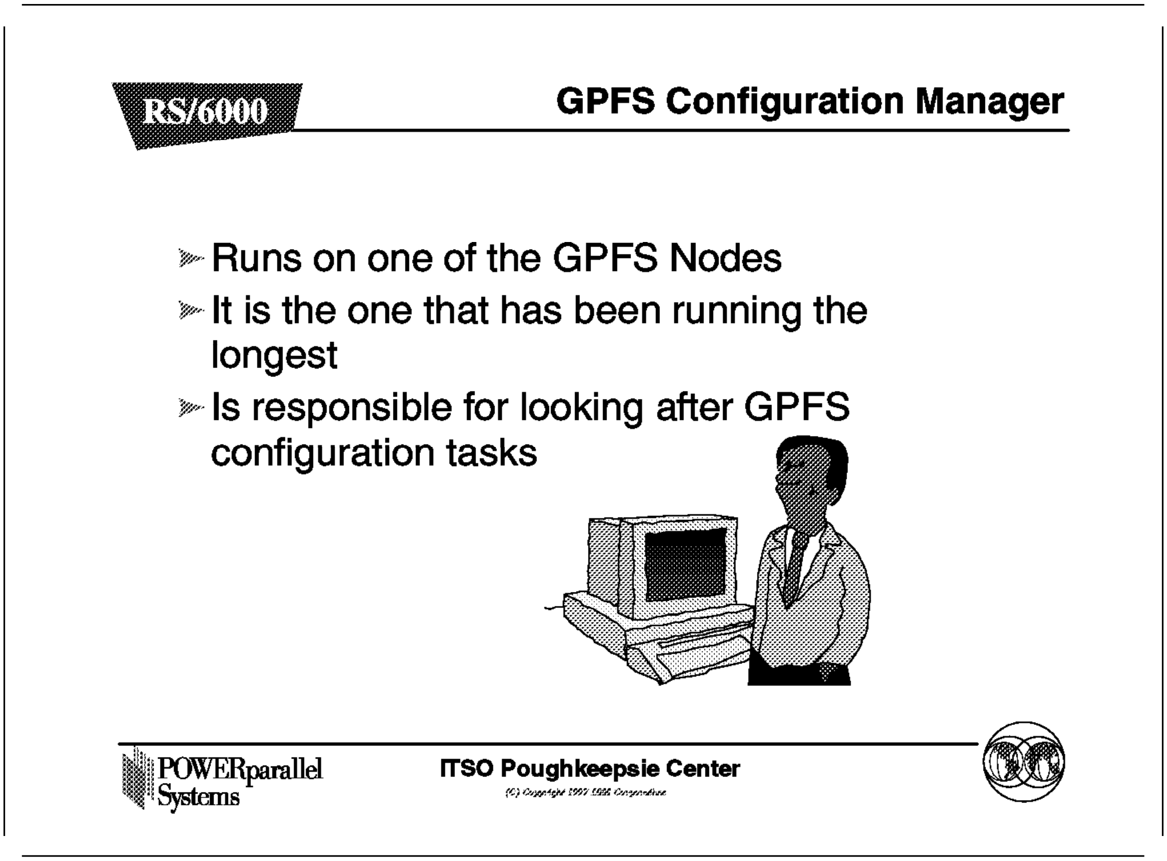
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

A Stripe Group Manager is running for each GPFS file system. Only one can run per file system, because it is responsible for handling requests for tokens.

The Configuration Manager decides which node will act as Stripe Group Manager for each GPFS file system. Other file systems will probably use a different node, so that the workload involved in Token Management is distributed.

## 6.4.20 GPFS Configuration Manager



RS/6000

GPFS Configuration Manager

- Runs on one of the GPFS Nodes
- It is the one that has been running the longest
- Is responsible for looking after GPFS configuration tasks

POWERparallel Systems

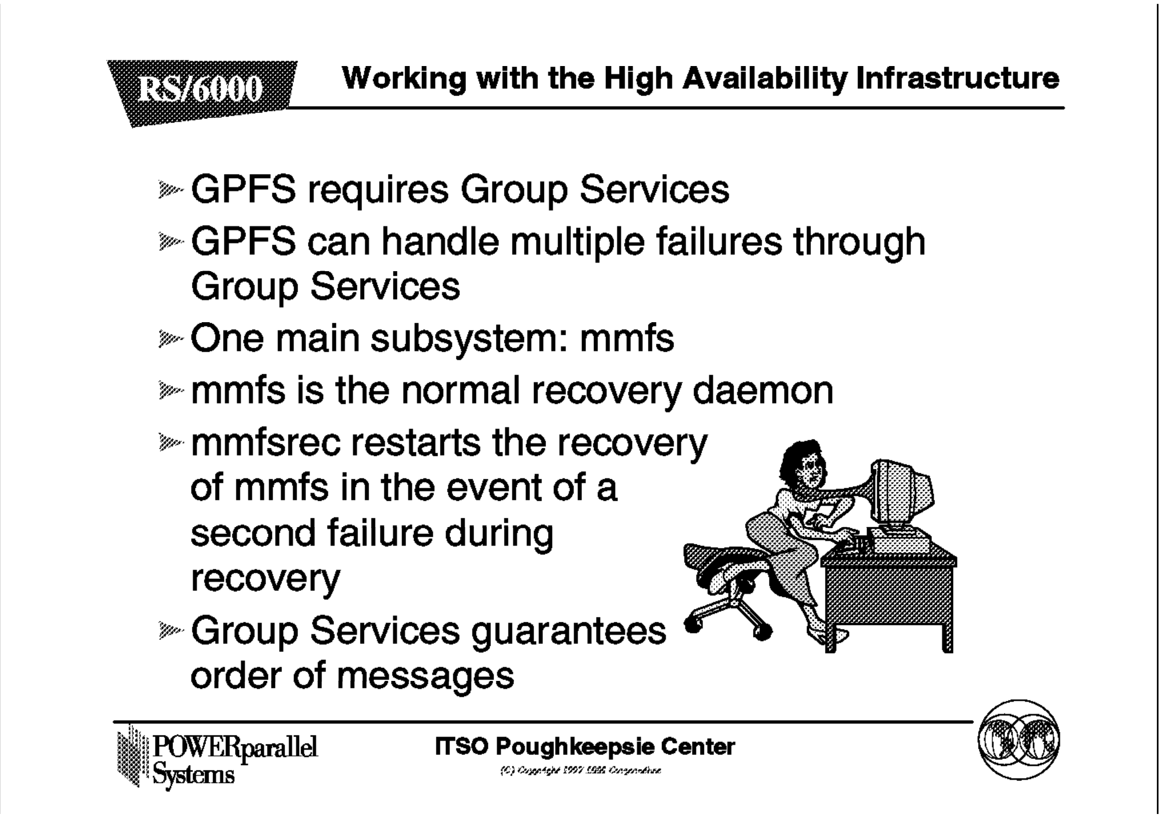
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

There is just one Configuration Manager for the entire GPFS set of nodes. If the node fails, another node will take over this role.

The first node to join the GPFS group, or the first one to start mmfs, will be the Configuration Manager. There is little overhead on the node in most circumstances. The workload increases in case of a failure as recovery takes place.

### 6.4.21 GPFS working with High Availability Infrastructure



RS/6000

Working with the High Availability Infrastructure

- GPFS requires Group Services
- GPFS can handle multiple failures through Group Services
- One main subsystem: mmfs
- mmfs is the normal recovery daemon
- mmfsrec restarts the recovery of mmfs in the event of a second failure during recovery
- Group Services guarantees order of messages

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Both the mmfs and mmfsrec groups are registered with Group Services. The mmfs subsystem is the normal recovery mechanism. In the event of a second failure during the recovery, mmfsrec is used to ensure correct recovery.

To monitor the attributes of the hags group, you can use the command `lssrc -l -s hags`.

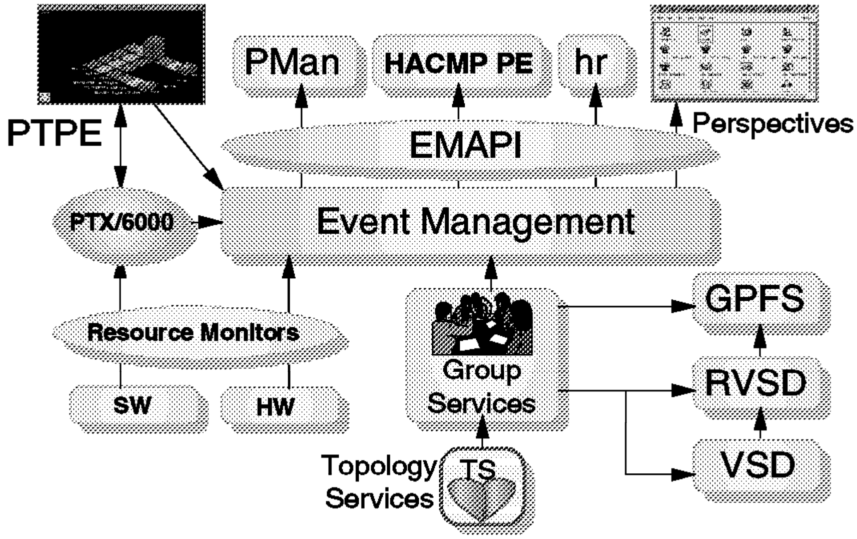
During recovery by mmfs, there is a four-phase process where votes take place to ensure that all nodes are recovering correctly and are in step.

In the event of a second failure, the voting cannot proceed, and in that case, mmfsrec steps in to take action based on the second failure. Recovery can continue with the new information about the second failure now available.

## 6.4.22 High Availability Infrastructure

RS/6000

High Availability Infrastructure

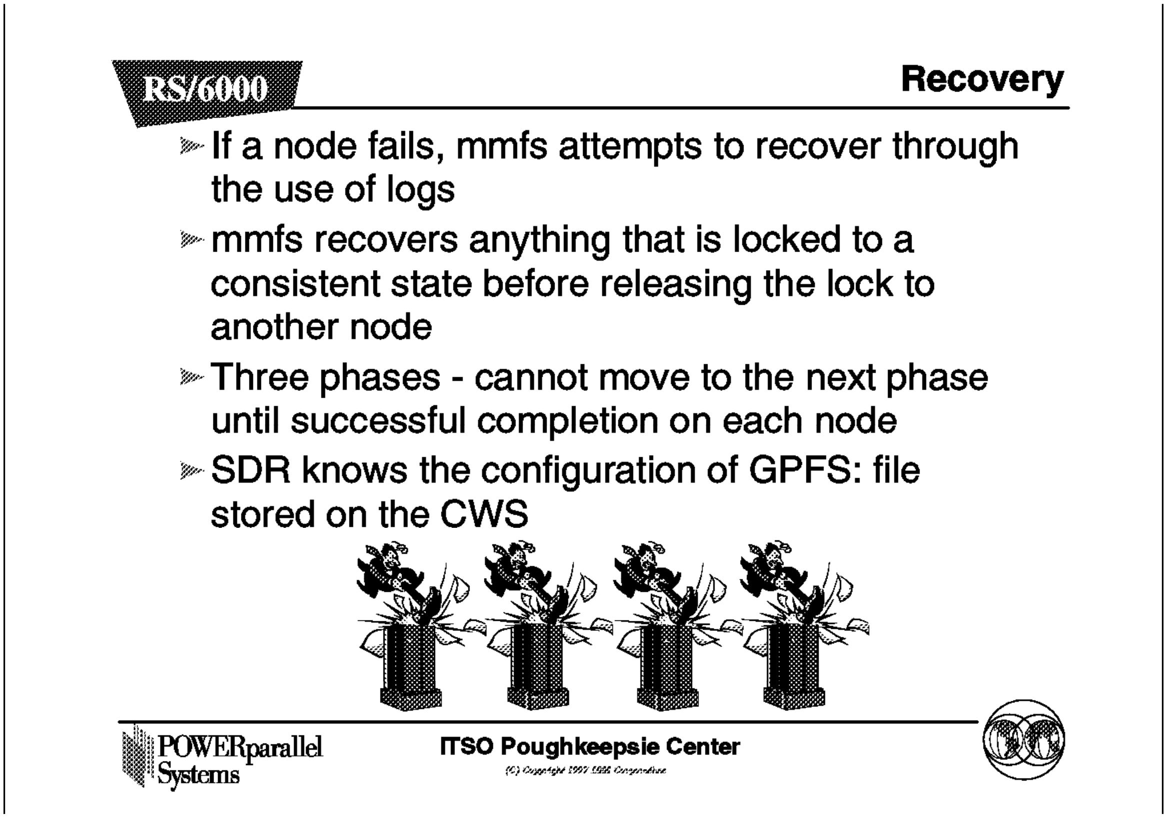


POWERparallel
Systems

ITSO Poughkeepsie Center

GPFS requires Group Services. RVSD and GPFS also works closely with the entire high availability infrastructure.

### 6.4.23 GPFS Recovery



RS/6000

Recovery

- If a node fails, mmfs attempts to recover through the use of logs
- mmfs recovers anything that is locked to a consistent state before releasing the lock to another node
- Three phases - cannot move to the next phase until successful completion on each node
- SDR knows the configuration of GPFS: file stored on the CWS

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Whenever a node boots, it checks the two GPFS files that are stored in the SDR to see if configuration changes were applied to the GPFS configuration while the node was down. If this is the case, the changes will be applied to the node at that time.

This allows configuration changes to be made while not all nodes are available. For example, new nodes or disks could be added to the GPFS configuration while some nodes are powered off.

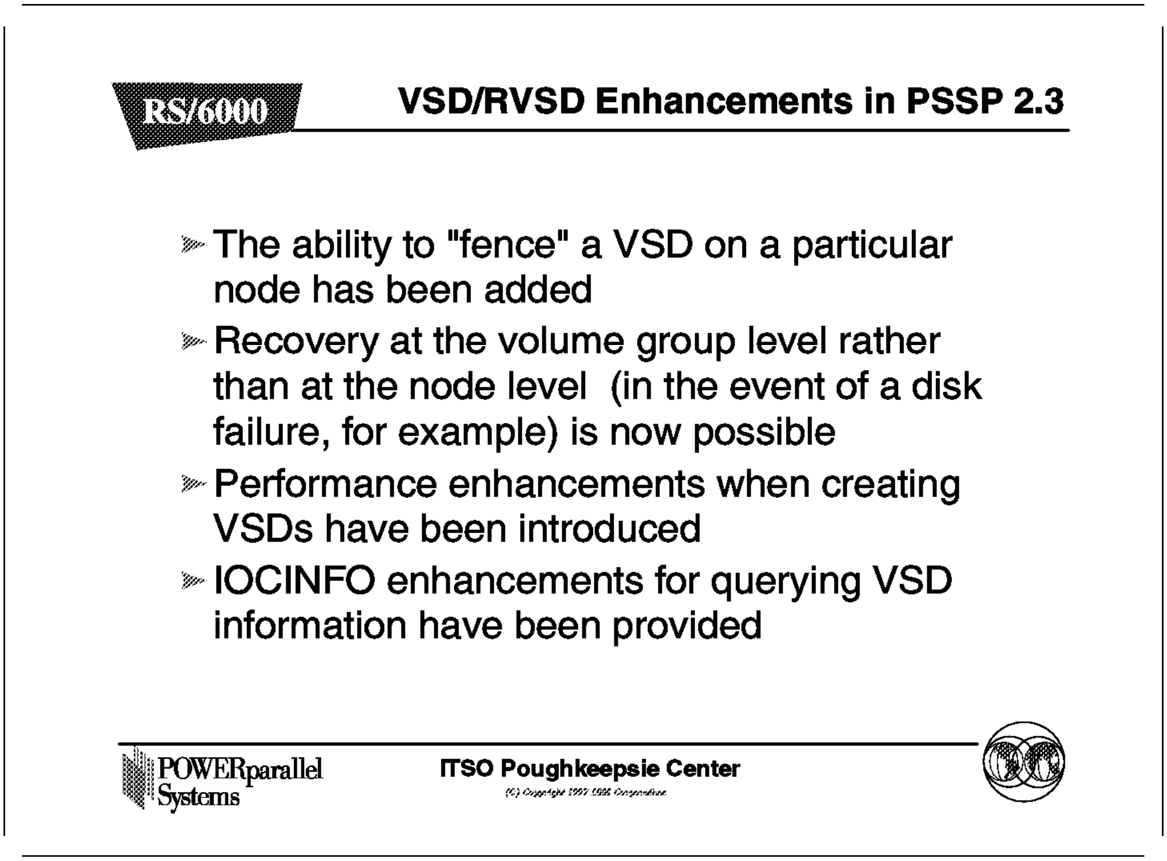
The files in the SDR that are created by GPFS are as follows:

/spdata/sys1/sdr/partitions/9.180.40.16/files/mmsdrcfg1

/spdata/sys1/sdr/partitions/9.180.40.16/files/mmsdrfs

These files should not be edited. They should be backed up when you back up your SDR, and can be used in the event of failure to recover.

### 6.4.24 VSD/RVSD Enhancements in PSSP 2.3



RS/6000

VSD/RVSD Enhancements in PSSP 2.3

- The ability to "fence" a VSD on a particular node has been added
- Recovery at the volume group level rather than at the node level (in the event of a disk failure, for example) is now possible
- Performance enhancements when creating VSDs have been introduced
- IOCINFO enhancements for querying VSD information have been provided

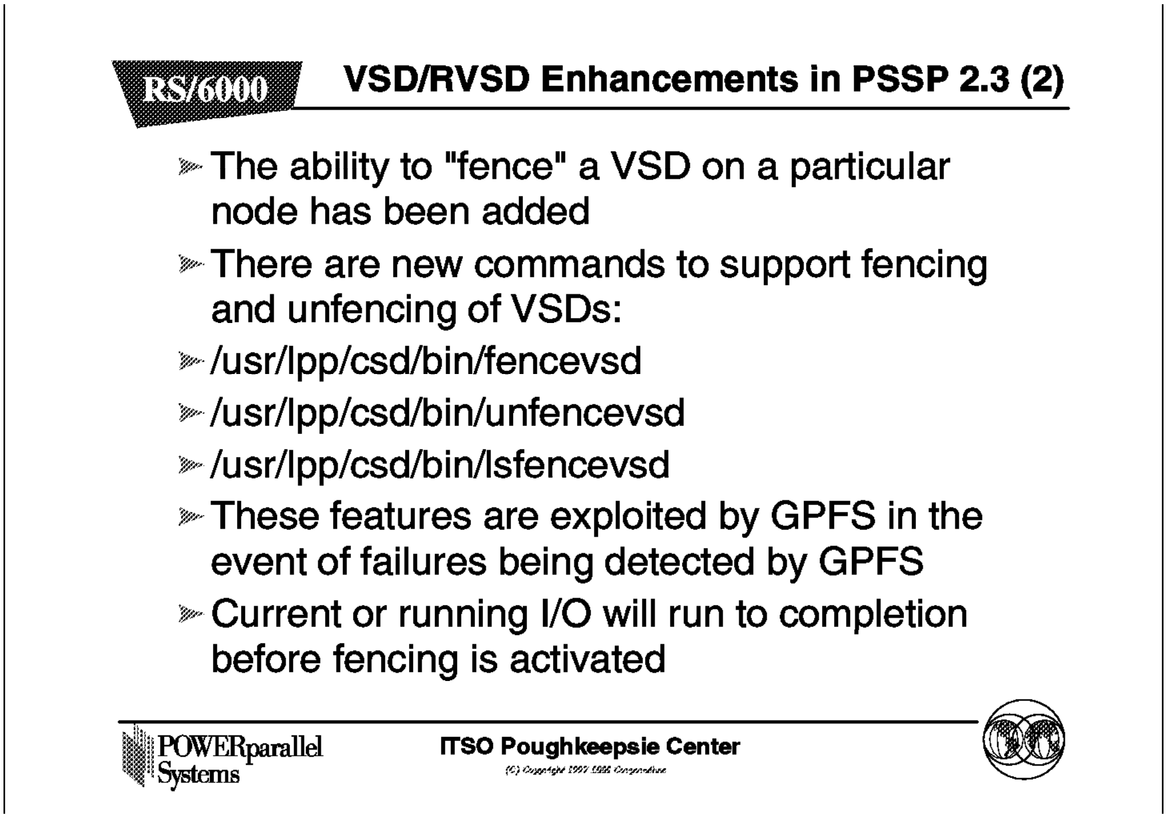
POWERparallel Systems

ITSO Poughkeepsie Center

There are a number of new functions in VSD Version 2.3 and in RVSD when operating in a PSSP 2.3 environment. Some of these functions are used by GPFS and are required.

The four major enhancements are listed here. Each will be discussed in detail.

### 6.4.24.1 VSD/RVSD Enhancements in PSSP 2.3



RS/6000

## VSD/RVSD Enhancements in PSSP 2.3 (2)

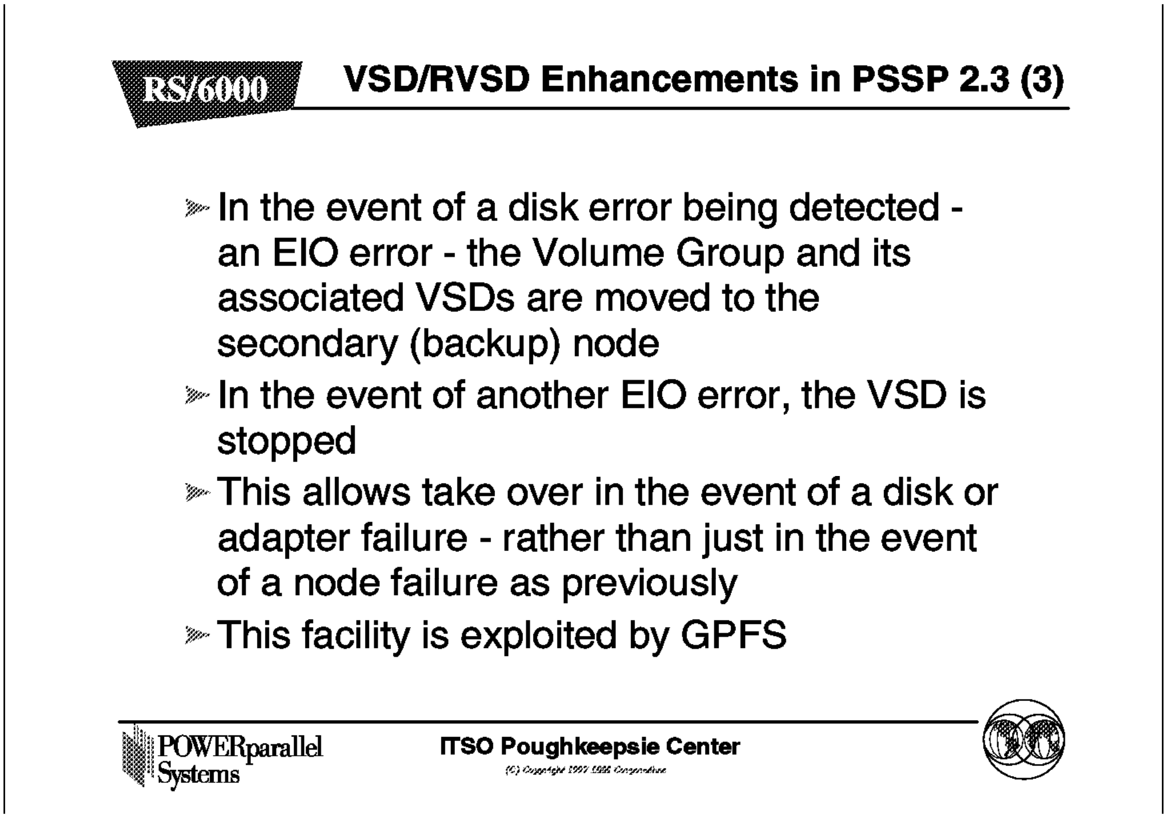
- The ability to "fence" a VSD on a particular node has been added
- There are new commands to support fencing and unfencing of VSDs:
- /usr/lpp/csd/bin/fencevsd
- /usr/lpp/csd/bin/unfencevsd
- /usr/lpp/csd/bin/lsfencevsd
- These features are exploited by GPFS in the event of failures being detected by GPFS
- Current or running I/O will run to completion before fencing is activated

POWERparallel Systems

ITSO Poughkeepsie Center

A VSD on a node within the SP can be fenced and I/O will stop after the current I/O is completed. There are new commands to fence or unfence a node. These facilities are used by GPFS to isolate VSDs in the event of failures.

### 6.4.24.2 VSD/RVSD Enhancements in PSSP 2.3



RS/6000

VSD/RVSD Enhancements in PSSP 2.3 (3)

- In the event of a disk error being detected - an EIO error - the Volume Group and its associated VSDs are moved to the secondary (backup) node
- In the event of another EIO error, the VSD is stopped
- This allows take over in the event of a disk or adapter failure - rather than just in the event of a node failure as previously
- This facility is exploited by GPFS

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In the event of an EIO error, RVSD notes the error and is the first to take action based on this information.

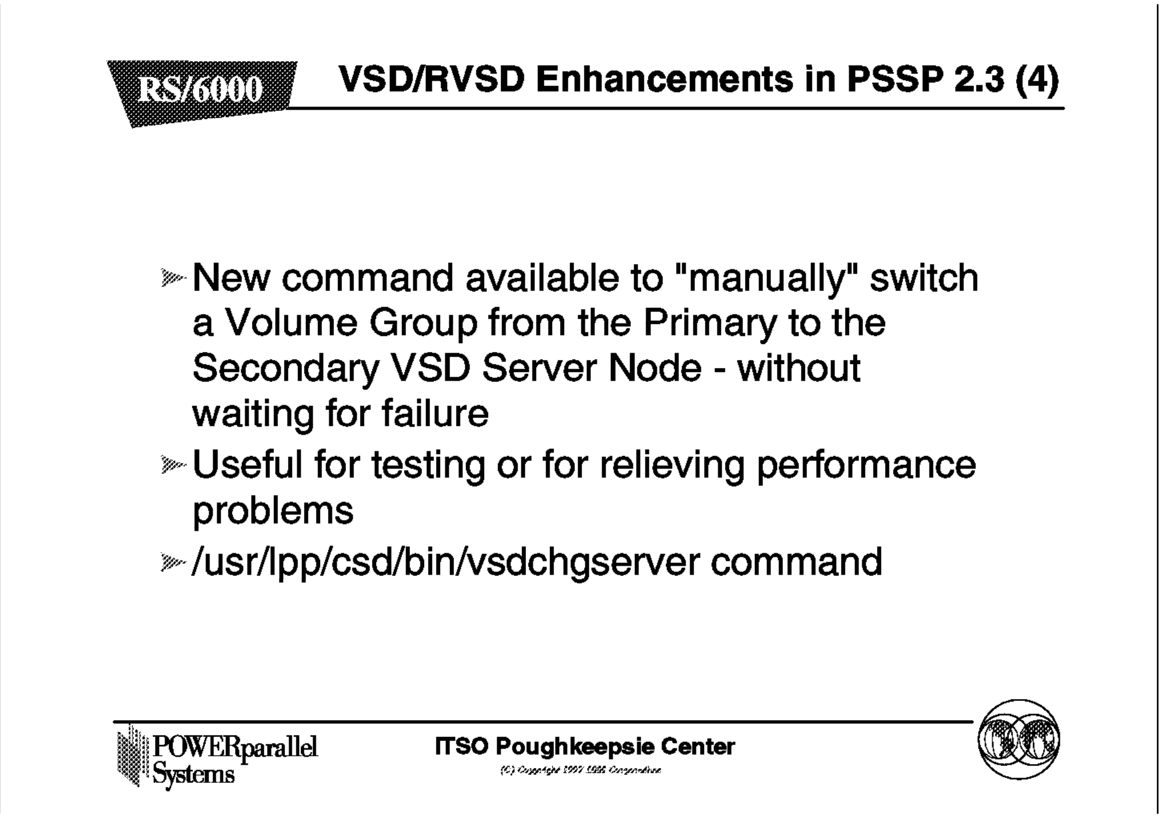
If the disk in question is twin-tailed, then the volume group is moved to the secondary node by GPFS and the VSDs are served by this node. In the event of a second EIO error, RVSD suspends the VSD.

GPFS is not aware of these errors at this time. However, in the event of not having a twin-tailed disk, GPFS is made aware of further EIO errors, and when it sees four such errors, it stops using the VSD in question.

There are three levels of checking/recovery:

- The disk hardware level
- The RVSD level
- The GPFS level

### 6.4.24.3 VSD/RVSD Enhancements in PSSP 2.3



RS/6000 **VSD/RVSD Enhancements in PSSP 2.3 (4)**

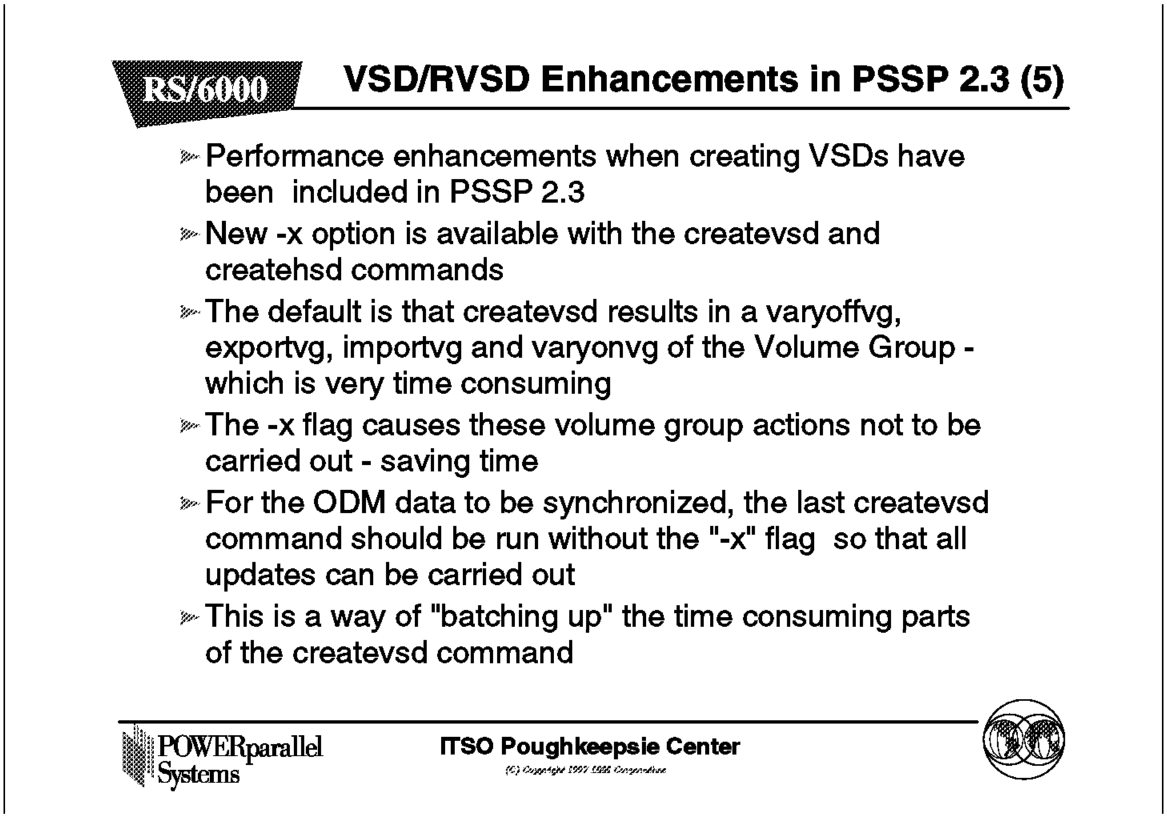
- New command available to "manually" switch a Volume Group from the Primary to the Secondary VSD Server Node - without waiting for failure
- Useful for testing or for relieving performance problems
- /usr/lpp/csd/bin/vsdchgserver command

POWERparallel Systems

**ITSO Poughkeepsie Center**

On a request from the primary node, a volume group can be manually moved to the secondary node using RVSD.

### 6.4.24.4 VSD/RVSD Enhancements in PSSP 2.3



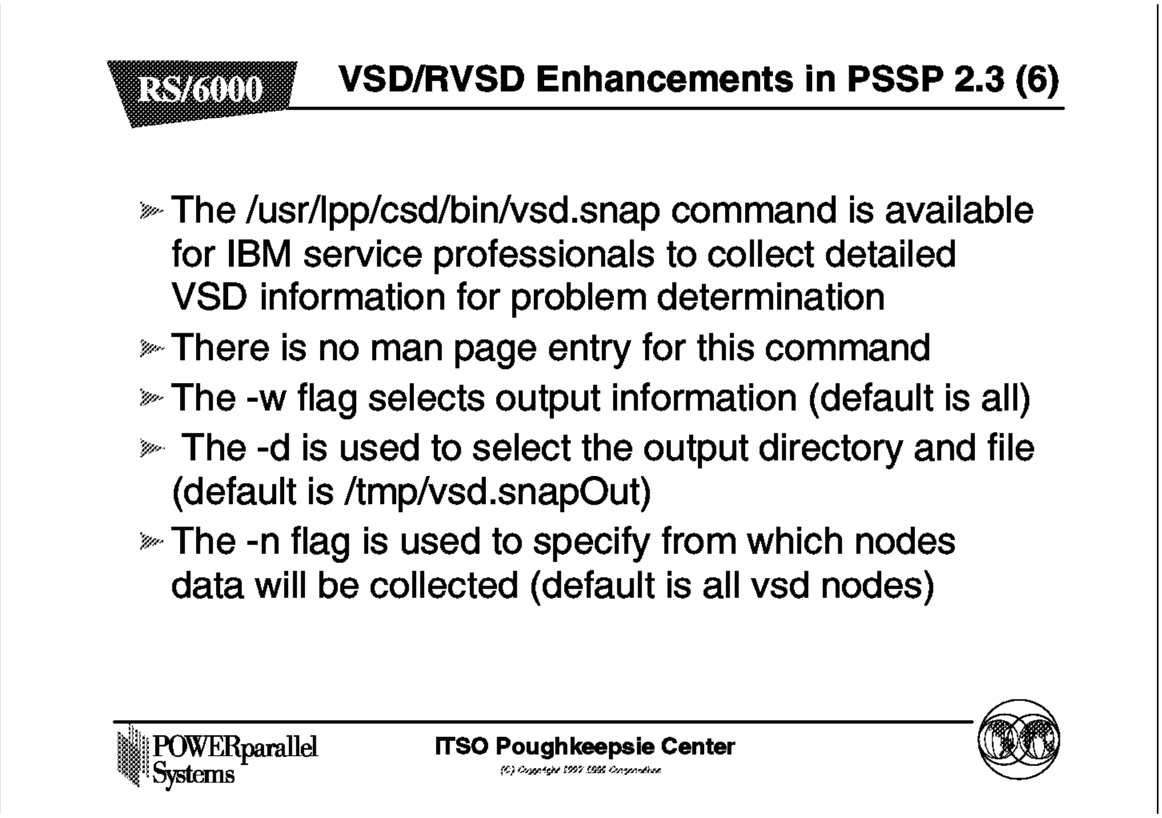
## VSD/RVSD Enhancements in PSSP 2.3 (5)

- Performance enhancements when creating VSDs have been included in PSSP 2.3
- New -x option is available with the createvsd and createhsd commands
- The default is that createvsd results in a varyoffvg, exportvg, importvg and varyonvg of the Volume Group - which is very time consuming
- The -x flag causes these volume group actions not to be carried out - saving time
- For the ODM data to be synchronized, the last createvsd command should be run without the "-x" flag so that all updates can be carried out
- This is a way of "batching up" the time consuming parts of the createvsd command

ITSO Poughkeepsie Center

Creating VSDs can be quite time consuming due to the fact that Volume Group actions are activated at the creation of each VSD. This new option allows you to postpone the Volume Group actions until later, thereby improving performance when creating a large number of VSDs.

### 6.4.24.5 VSD/RVSD Enhancements in PSSP 2.3



RS/6000

## VSD/RVSD Enhancements in PSSP 2.3 (6)

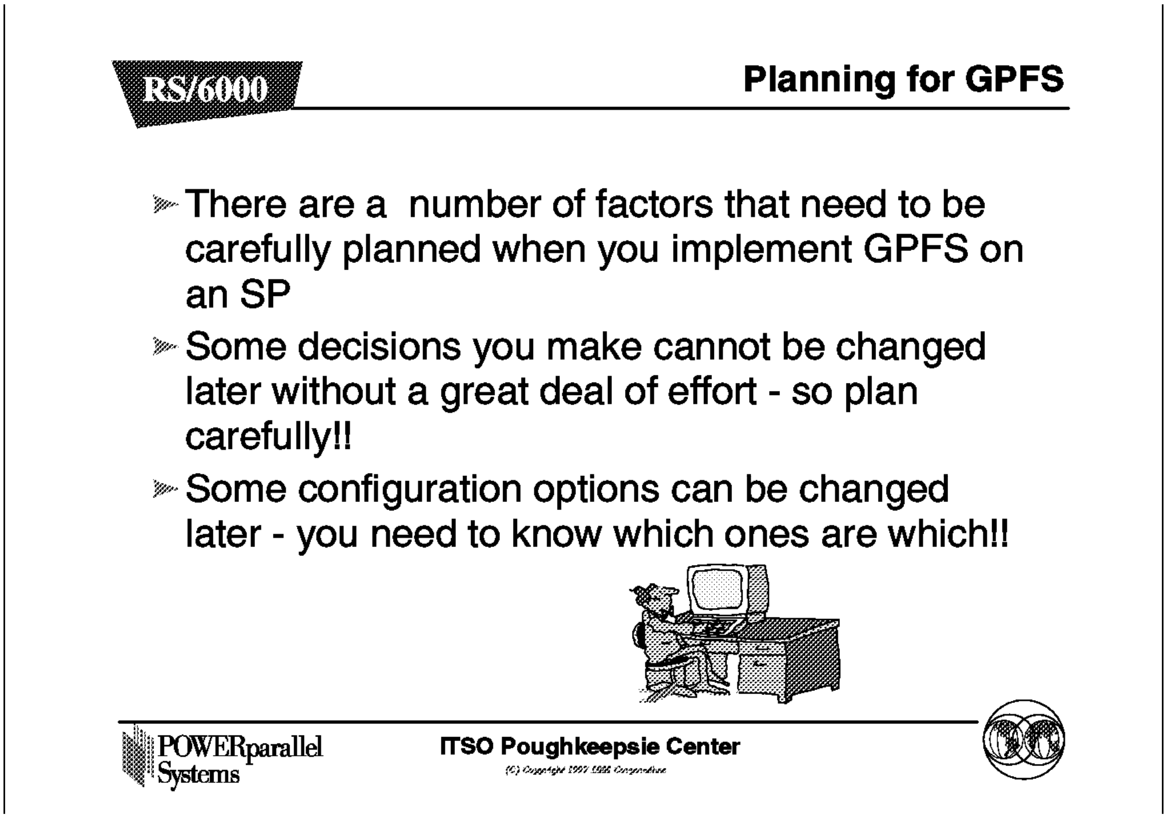
- The /usr/lpp/csd/bin/vsd.snap command is available for IBM service professionals to collect detailed VSD information for problem determination
- There is no man page entry for this command
- The -w flag selects output information (default is all)
- The -d is used to select the output directory and file (default is /tmp/vsd.snapOut)
- The -n flag is used to specify from which nodes data will be collected (default is all vsd nodes)

POWERparallel Systems

ITSO Poughkeepsie Center

The vsd.snap command allows IBM professionals to collect detailed data about VSD status and activity.

## 6.5 Planning for GPFS



RS/6000

### Planning for GPFS

- There are a number of factors that need to be carefully planned when you implement GPFS on an SP
- Some decisions you make cannot be changed later without a great deal of effort - so plan carefully!!
- Some configuration options can be changed later - you need to know which ones are which!!

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

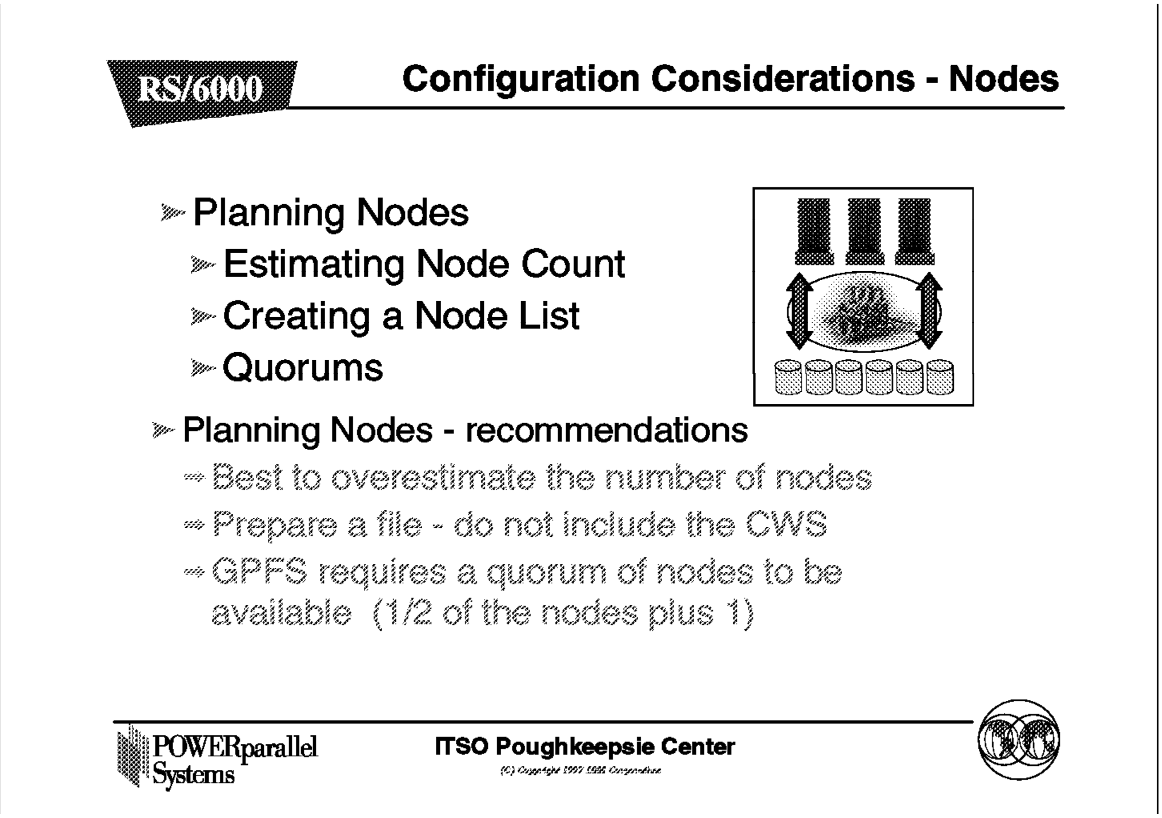
GPFS has many options and needs careful planning in order to be implemented successfully.

You will create a GPFS system first, and then create the GPFS file system to store your data.

Decisions that you make at this time cannot be changed later. This means that you really want to get it right the first time. In the worst case, you will have to back up a GPFS file system, recreate it with different parameters, and then reload the data. This is not recommended.

A summary of recommendations can be found later in the chapter.

## 6.5.1 GPFS Configuration Considerations



RS/6000

Configuration Considerations - Nodes

- Planning Nodes
  - Estimating Node Count
  - Creating a Node List
  - Quorums
- Planning Nodes - recommendations
  - Best to overestimate the number of nodes
  - Prepare a file - do not include the CWS
  - GPFS requires a quorum of nodes to be available (1/2 of the nodes plus 1)

POWERparallel Systems

ITSO Poughkeepsie Center

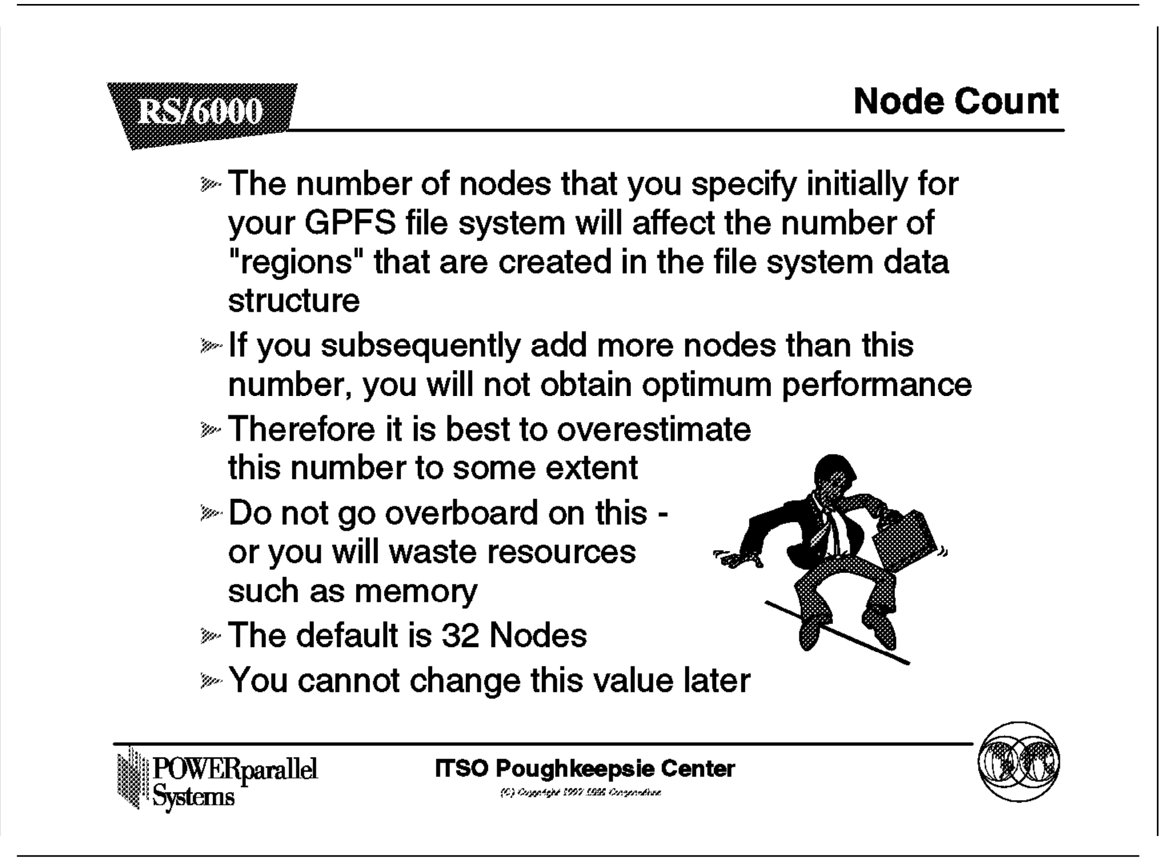
(C) Copyright 1997 IBM Corporation

As you create your GPFS system, you will be asked to supply information about which nodes are in the GPFS pool.

As you create an individual GPFS file system, you need to supply other information about nodes, such as how many nodes will participate in this GPFS file system. The default is 32, and you should not underestimate this number. Make sure that you include at least as many nodes as the maximum that you expect in the GPFS pool.

This information is used to create `allocation regions` that will affect the performance of the file system.

## 6.5.2 Node Count



RS/6000

Node Count

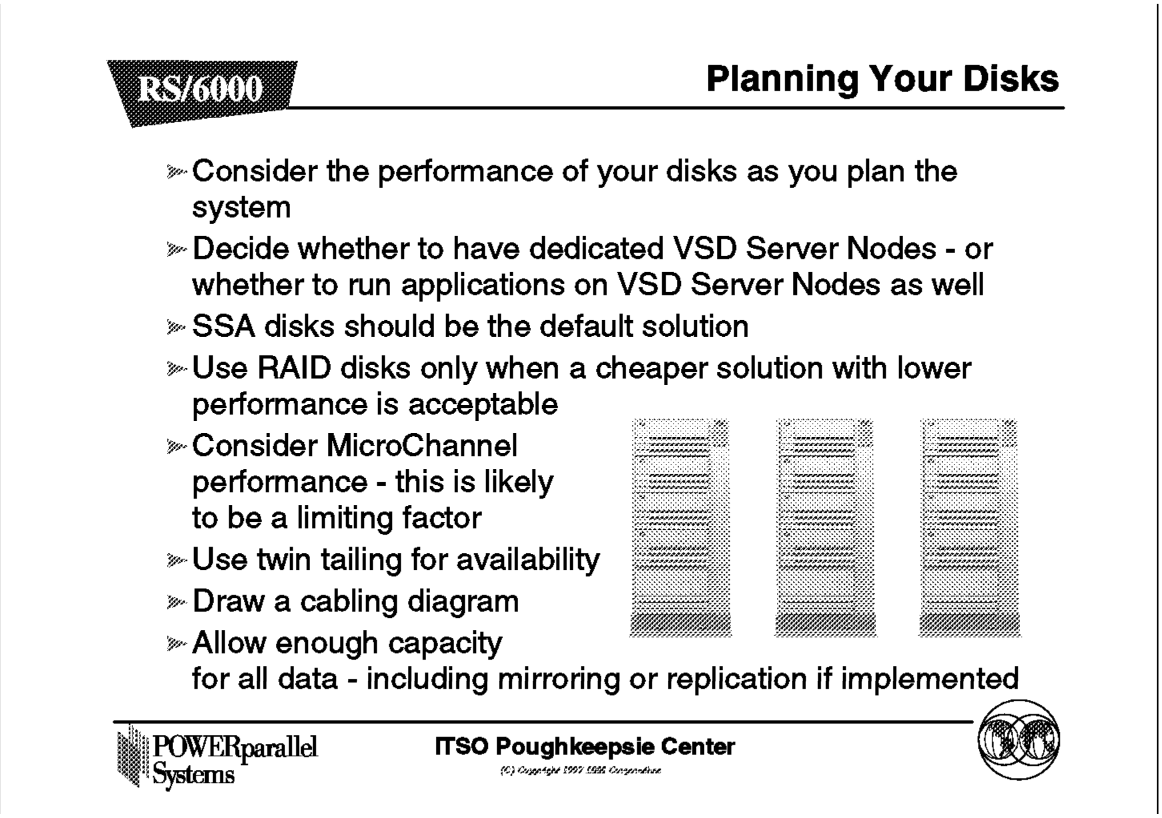
- The number of nodes that you specify initially for your GPFS file system will affect the number of "regions" that are created in the file system data structure
- If you subsequently add more nodes than this number, you will not obtain optimum performance
- Therefore it is best to overestimate this number to some extent
- Do not go overboard on this - or you will waste resources such as memory
- The default is 32 Nodes
- You cannot change this value later

POWERparallel Systems

ITSO Poughkeepsie Center

This value cannot be changed later.

### 6.5.3 Planning Your Disks



**Planning Your Disks**

- Consider the performance of your disks as you plan the system
- Decide whether to have dedicated VSD Server Nodes - or whether to run applications on VSD Server Nodes as well
- SSA disks should be the default solution
- Use RAID disks only when a cheaper solution with lower performance is acceptable
- Consider MicroChannel performance - this is likely to be a limiting factor
- Use twin tailing for availability
- Draw a cabling diagram
- Allow enough capacity for all data - including mirroring or replication if implemented

ITSO Poughkeepsie Center

GPFS allows you to maximize I/O activity, but you will need to carefully plan your disks.

As a general rule, it is best to have an equal number of identical disks on each node in the GPFS pool. This gives you best performance and space utilization.

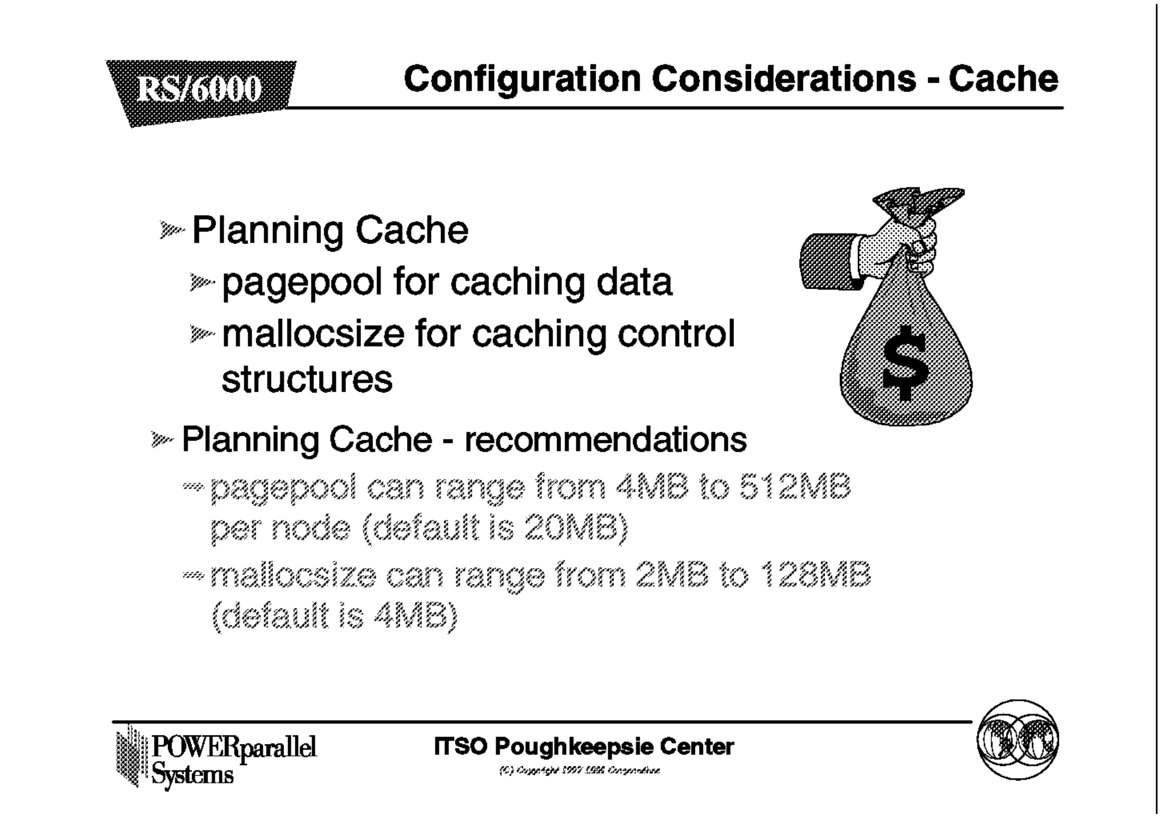
You should also put one VSD per disk only. This is the default option within GPFS. Do not have a VSD spanning multiple disks, and do not have multiple VSDs on one disk.

In many cases, it is an advantage to create VSD servers as separate nodes that do not run other applications.

If you run other applications, you will have an performance impact on the VSD activity and vice versa.

In addition, this limits your operational flexibility. For example, in the case where you would like to reboot a node, you may not be able to do so without impacting the other applications.

## 6.5.4 GPFS Considerations - Cache



RS/6000

Configuration Considerations - Cache

- Planning Cache
  - pagepool for caching data
  - mallocsize for caching control structures
- Planning Cache - recommendations
  - pagepool can range from 4MB to 512MB per node (default is 20MB)
  - mallocsize can range from 2MB to 128MB (default is 4MB)

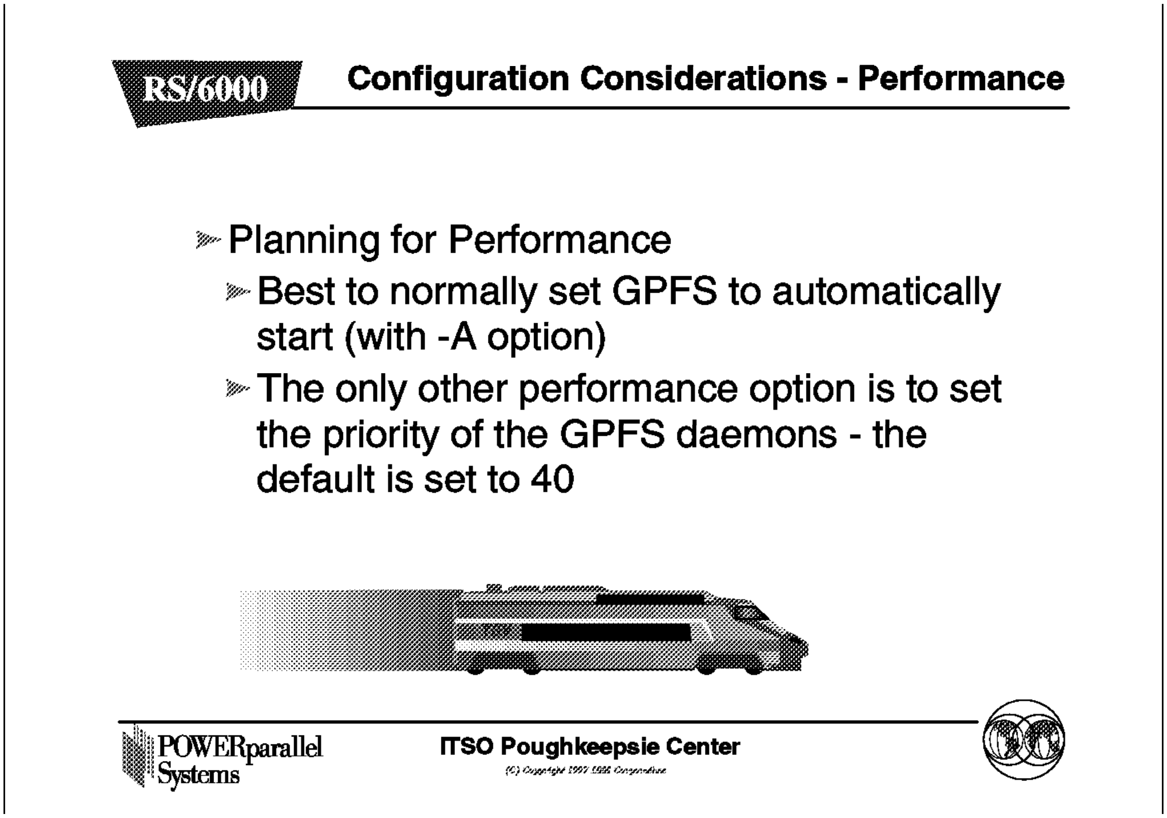
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Use the default values until you have reason to do otherwise.

### 6.5.5 GPFS Considerations - Performance



RS/6000

**Configuration Considerations - Performance**

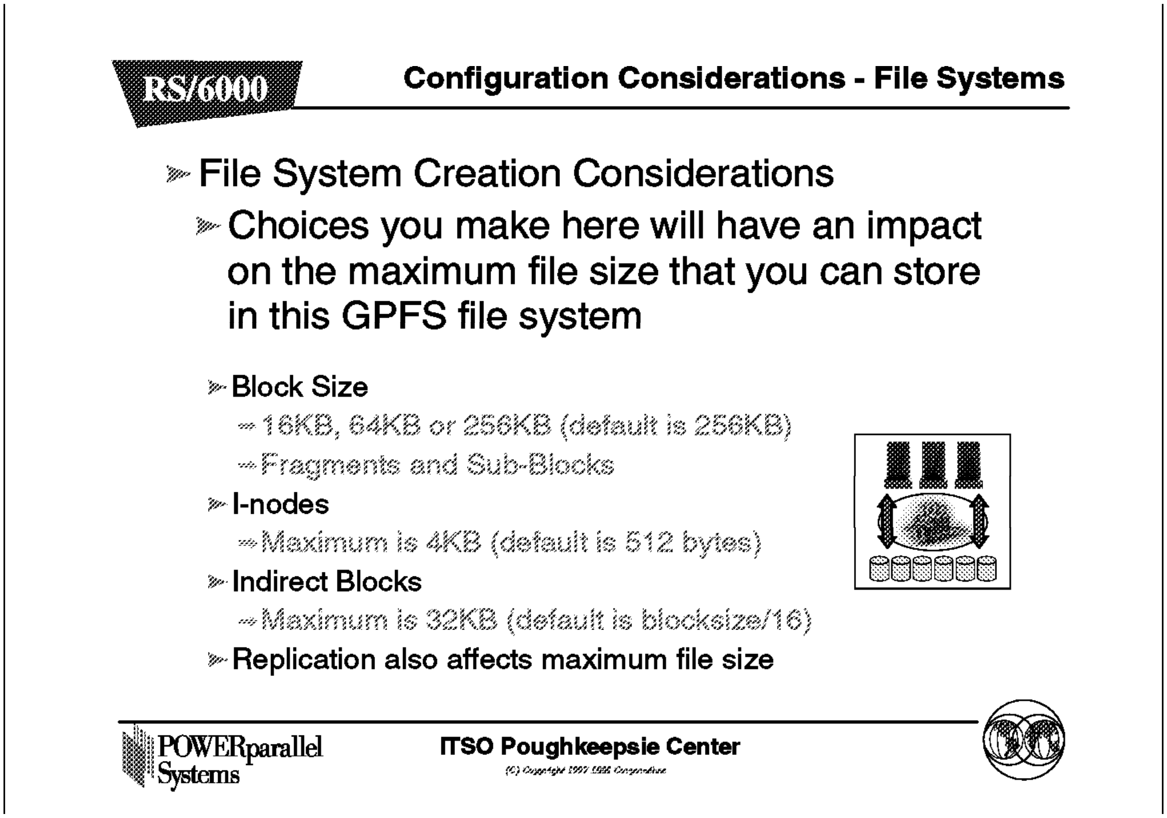
- Planning for Performance
  - Best to normally set GPFS to automatically start (with -A option)
  - The only other performance option is to set the priority of the GPFS daemons - the default is set to 40

POWERparallel Systems

**ITSO Poughkeepsie Center**

It is recommended to automatically start GPFS.

### 6.5.6 Configuration Considerations - File Systems



RS/6000

**Configuration Considerations - File Systems**

- File System Creation Considerations
  - Choices you make here will have an impact on the maximum file size that you can store in this GPFS file system
    - Block Size
      - 16KB, 64KB or 256KB (default is 256KB)
      - Fragments and Sub-Blocks
    - I-nodes
      - Maximum is 4KB (default is 512 bytes)
    - Indirect Blocks
      - Maximum is 32KB (default is blocksize/16)
    - Replication also affects maximum file size

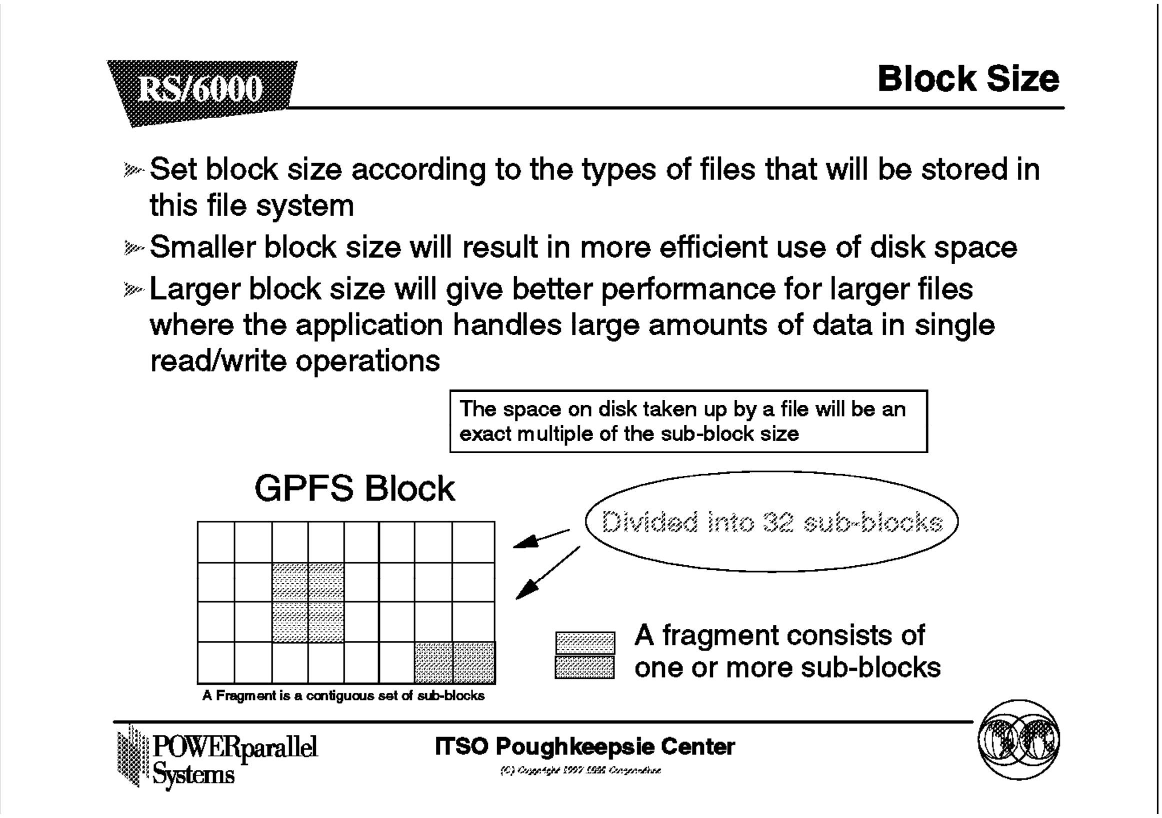
POWERparallel Systems

ITSO Poughkeepsie Center

This is a critical area where you will need to make decisions that cannot be changed. Decisions you make will affect the size of the largest file that you can store in this particular GPFS file system.

This will be discussed in more detail.

## 6.5.7 GPFS Blocksize



RS/6000

**Block Size**

- Set block size according to the types of files that will be stored in this file system
- Smaller block size will result in more efficient use of disk space
- Larger block size will give better performance for larger files where the application handles large amounts of data in single read/write operations

The space on disk taken up by a file will be an exact multiple of the sub-block size

GPFS Block

Divided into 32 sub-blocks

▨ A fragment consists of one or more sub-blocks

A Fragment is a contiguous set of sub-blocks

POWERparallel Systems

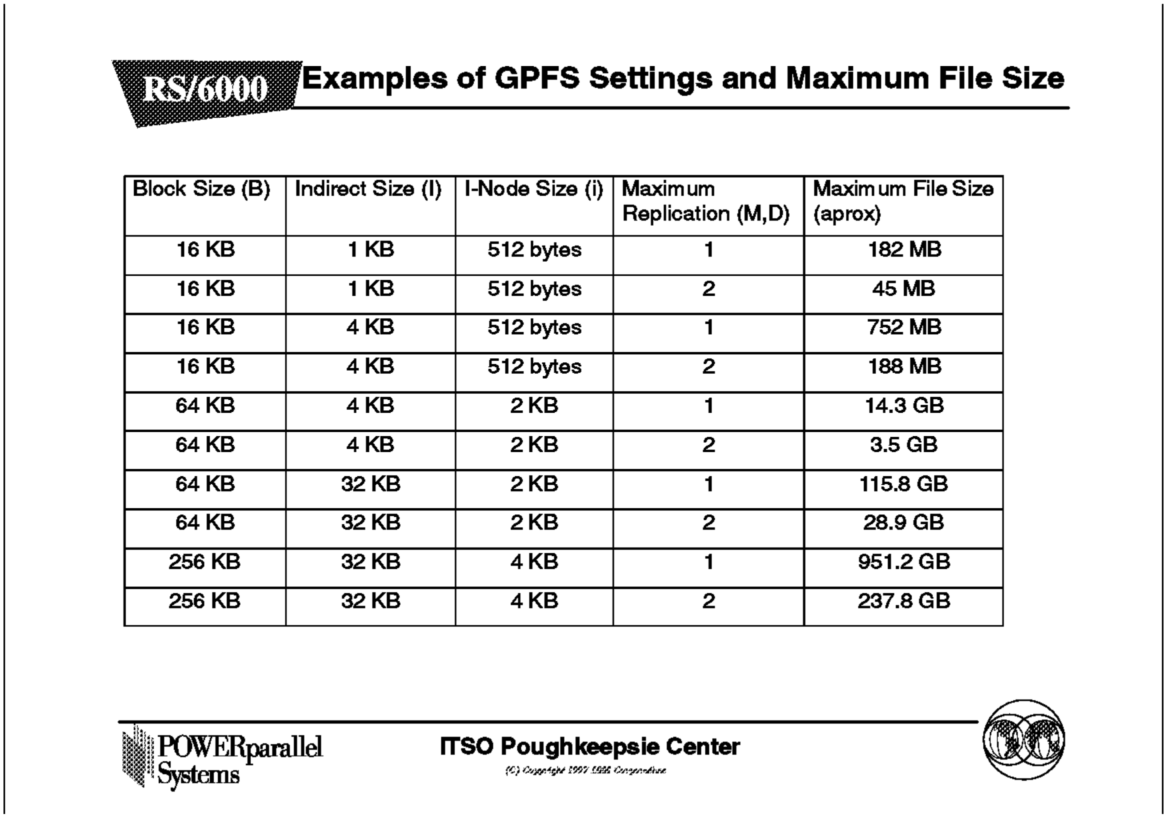
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

You will select the block size for your GPFS file system. This defines the size of the blocks that are written as you stripe over each of the disks in the Stripe Group.

This also defines some other parameters as shown here. The sub-block size results in the smallest area of disk that will is for a file. For small files, this has an impact on the space wasted by unused areas of disk.

## 6.5.8 Examples of GPFS Settings



RS/6000

### Examples of GPFS Settings and Maximum File Size

| Block Size (B) | Indirect Size (I) | I-Node Size (i) | Maximum Replication (M,D) | Maximum File Size (aprox) |
|---|---|---|---|---|
| 16 KB | 1 KB | 512 bytes | 1 | 182 MB |
| 16 KB | 1 KB | 512 bytes | 2 | 45 MB |
| 16 KB | 4 KB | 512 bytes | 1 | 752 MB |
| 16 KB | 4 KB | 512 bytes | 2 | 188 MB |
| 64 KB | 4 KB | 2 KB | 1 | 14.3 GB |
| 64 KB | 4 KB | 2 KB | 2 | 3.5 GB |
| 64 KB | 32 KB | 2 KB | 1 | 115.8 GB |
| 64 KB | 32 KB | 2 KB | 2 | 28.9 GB |
| 256 KB | 32 KB | 4 KB | 1 | 951.2 GB |
| 256 KB | 32 KB | 4 KB | 2 | 237.8 GB |

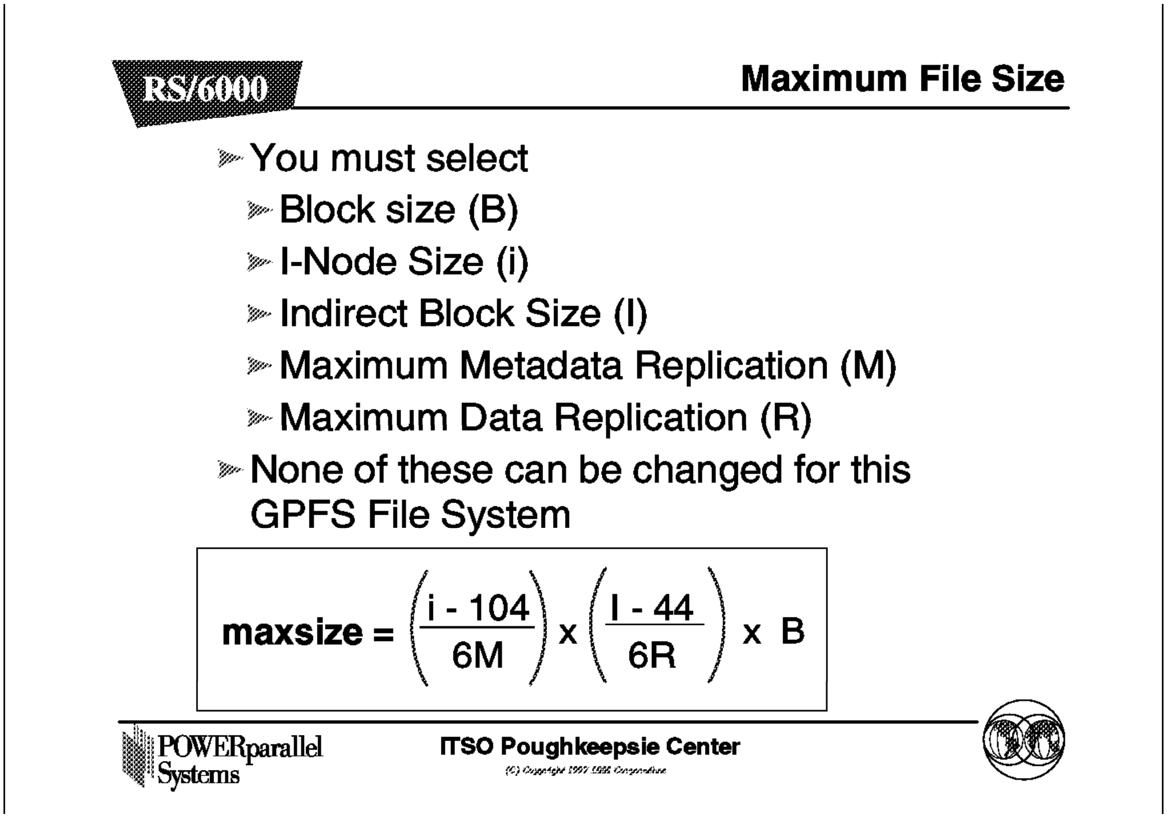
POWERparallel Systems

ITSO Poughkeepsie Center

This table shows some examples of the impact of the decisions you make on the largest file size that you can use in GPFS. You can see the impact of selecting larger block sizes, indirect size, i-node size and replication.

The formula on the next page allows you to calculate these values.

### 6.5.9 GPFS Maximum File Size



RS/6000

**Maximum File Size**

- You must select
  - Block size (B)
  - I-Node Size (i)
  - Indirect Block Size (I)
  - Maximum Metadata Replication (M)
  - Maximum Data Replication (R)
- None of these can be changed for this GPFS File System

$$\textbf{maxsize} = \left(\frac{i - 104}{6M}\right) \times \left(\frac{I - 44}{6R}\right) \times B$$

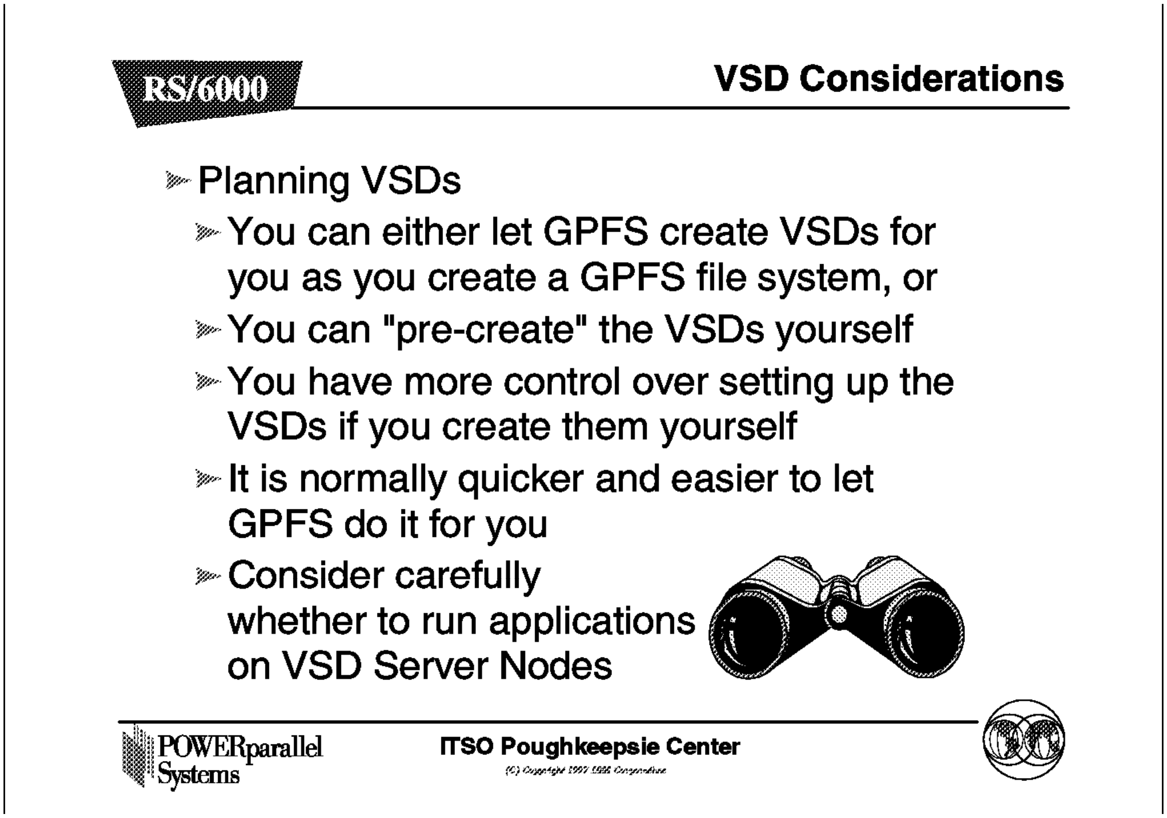
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

This formula helps you determine maximum file size based on the other decisions you make about this filesystem.

### 6.5.10 VSD Considerations



RS/6000

**VSD Considerations**

- Planning VSDs
  - You can either let GPFS create VSDs for you as you create a GPFS file system, or
  - You can "pre-create" the VSDs yourself
  - You have more control over setting up the VSDs if you create them yourself
  - It is normally quicker and easier to let GPFS do it for you
  - Consider carefully whether to run applications on VSD Server Nodes

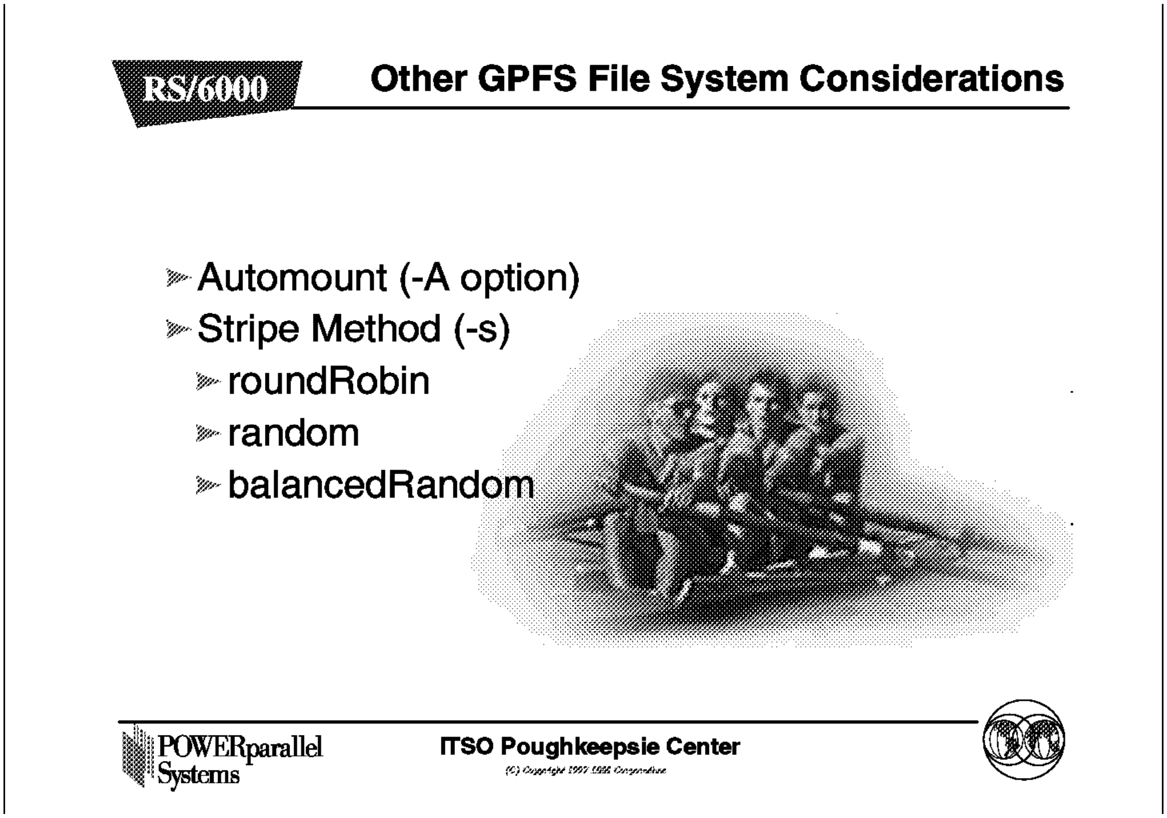
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

It is strongly recommended that under normal circumstances, you let GPFS create the VSDs that you need.

### 6.5.11 Other File System Considerations



RS/6000

**Other GPFS File System Considerations**

- Automount (-A option)
- Stripe Method (-s)
  - roundRobin
  - random
  - balancedRandom

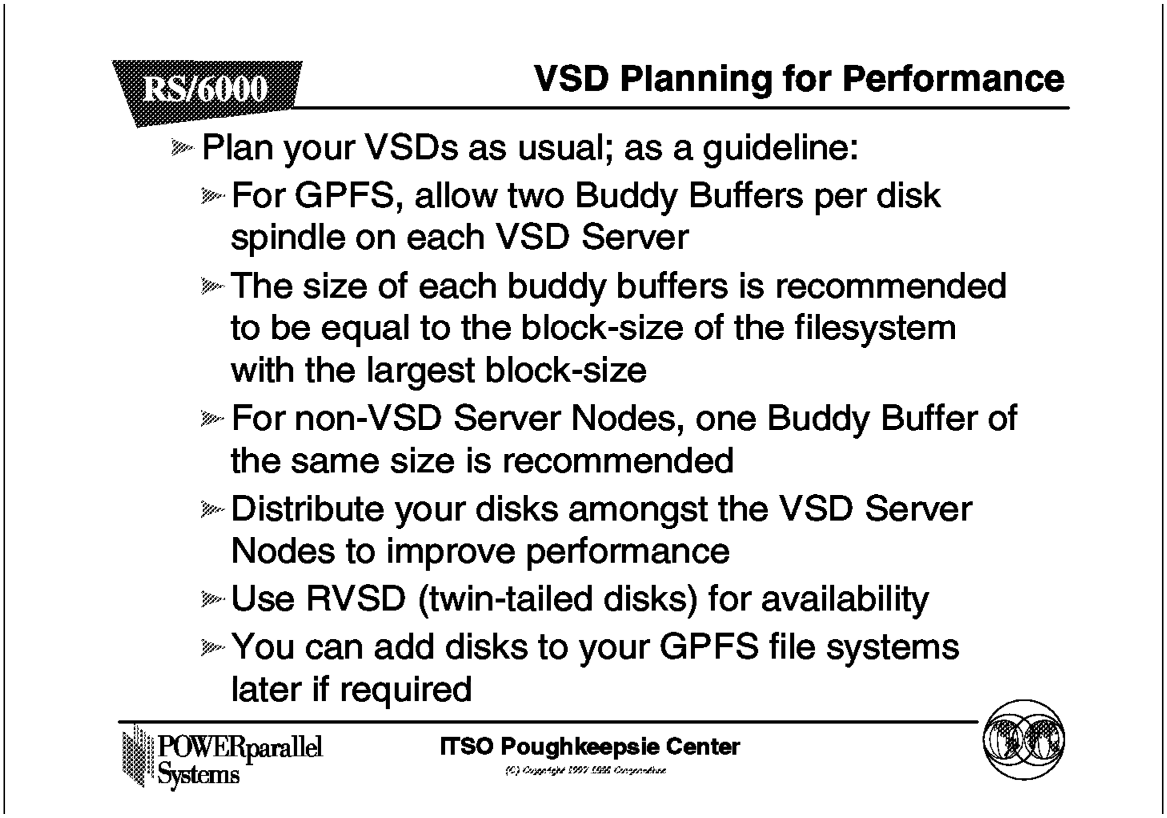
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 [illegible] Corporation

The default striping method is roundRobin and will normally be the option that you use. Do not use another method without good reason.

### 6.5.12 VSD Planning for Performance



RS/6000

**VSD Planning for Performance**

- Plan your VSDs as usual; as a guideline:
  - For GPFS, allow two Buddy Buffers per disk spindle on each VSD Server
  - The size of each buddy buffers is recommended to be equal to the block-size of the filesystem with the largest block-size
  - For non-VSD Server Nodes, one Buddy Buffer of the same size is recommended
  - Distribute your disks amongst the VSD Server Nodes to improve performance
  - Use RVSD (twin-tailed disks) for availability
  - You can add disks to your GPFS file systems later if required

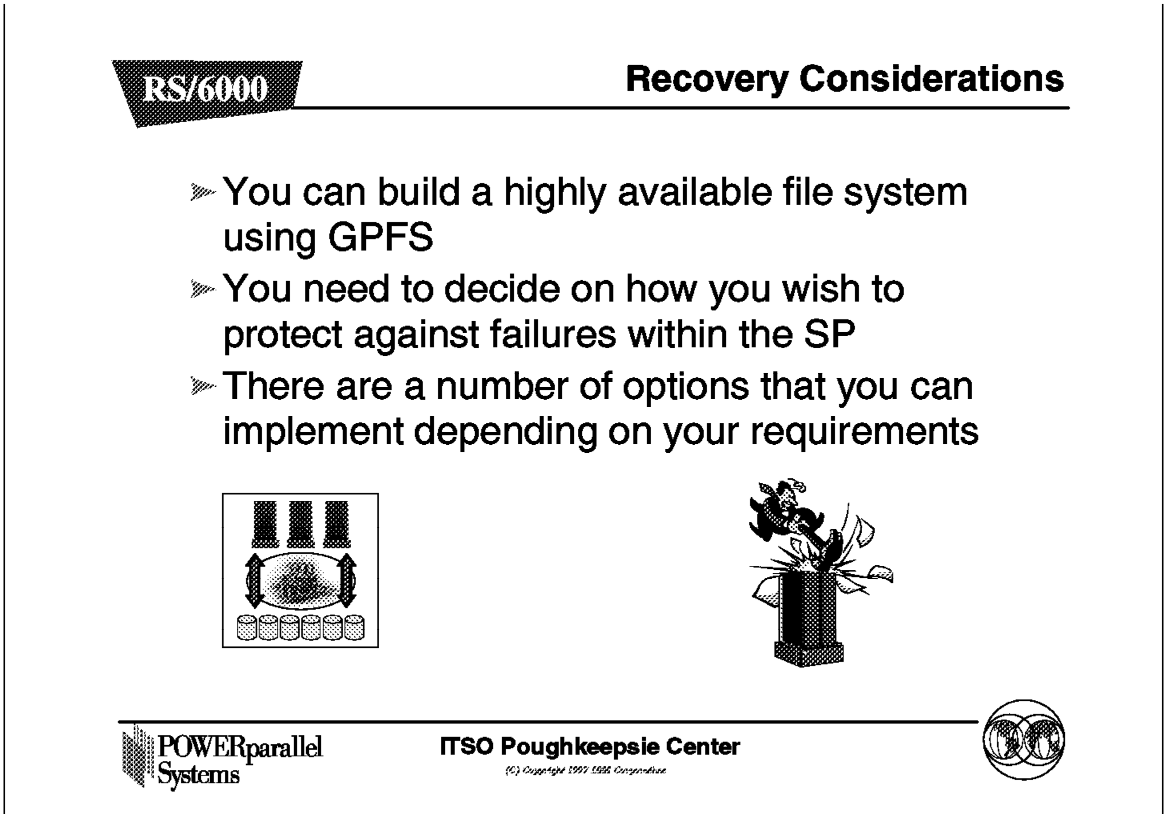
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

As a starting point, use these buddy buffer settings for VSD.

### 6.5.13 GPFS Recovery Considerations



RS/6000

Recovery Considerations

- You can build a highly available file system using GPFS
- You need to decide on how you wish to protect against failures within the SP
- There are a number of options that you can implement depending on your requirements

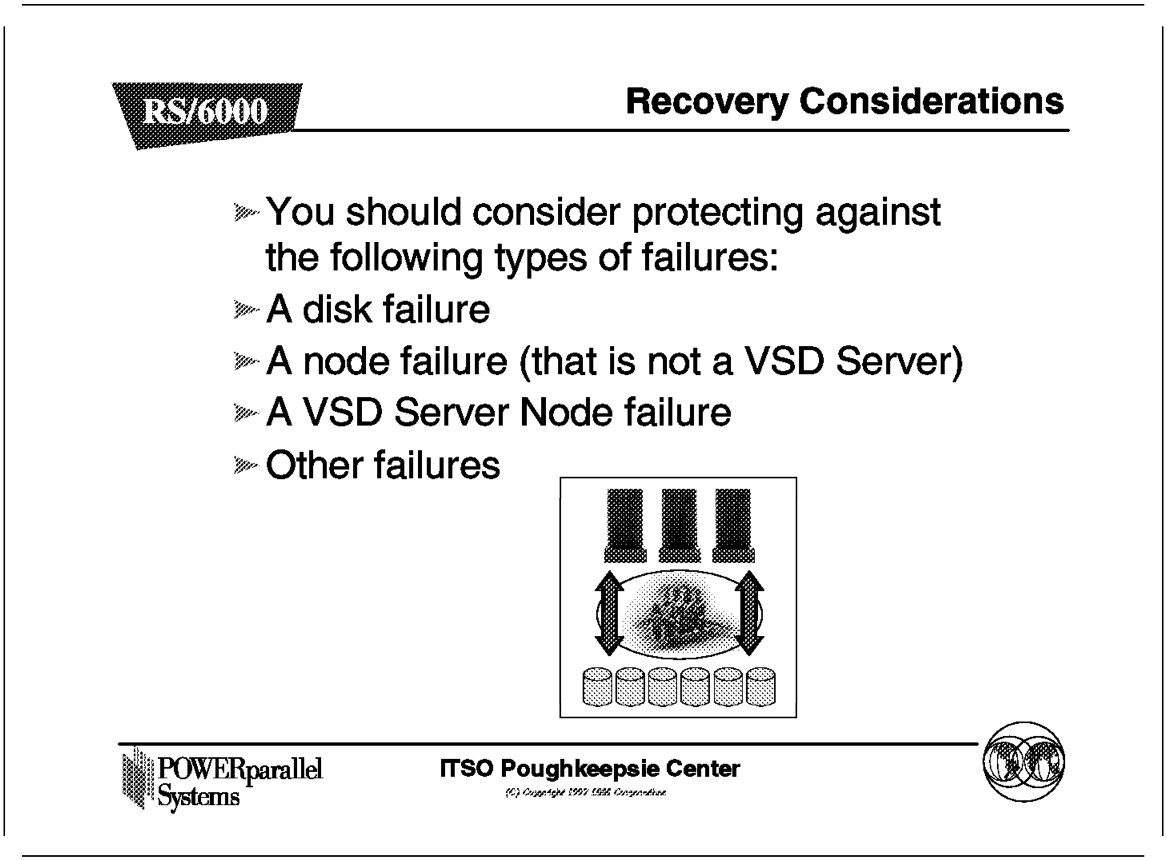
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

GPFS is very strong in the area of availability. You will need to plan your GPFS system carefully to handle failures that might occur.

## 6.5.14 GPFS Recovery



RS/6000

Recovery Considerations

- You should consider protecting against the following types of failures:
- A disk failure
- A node failure (that is not a VSD Server)
- A VSD Server Node failure
- Other failures

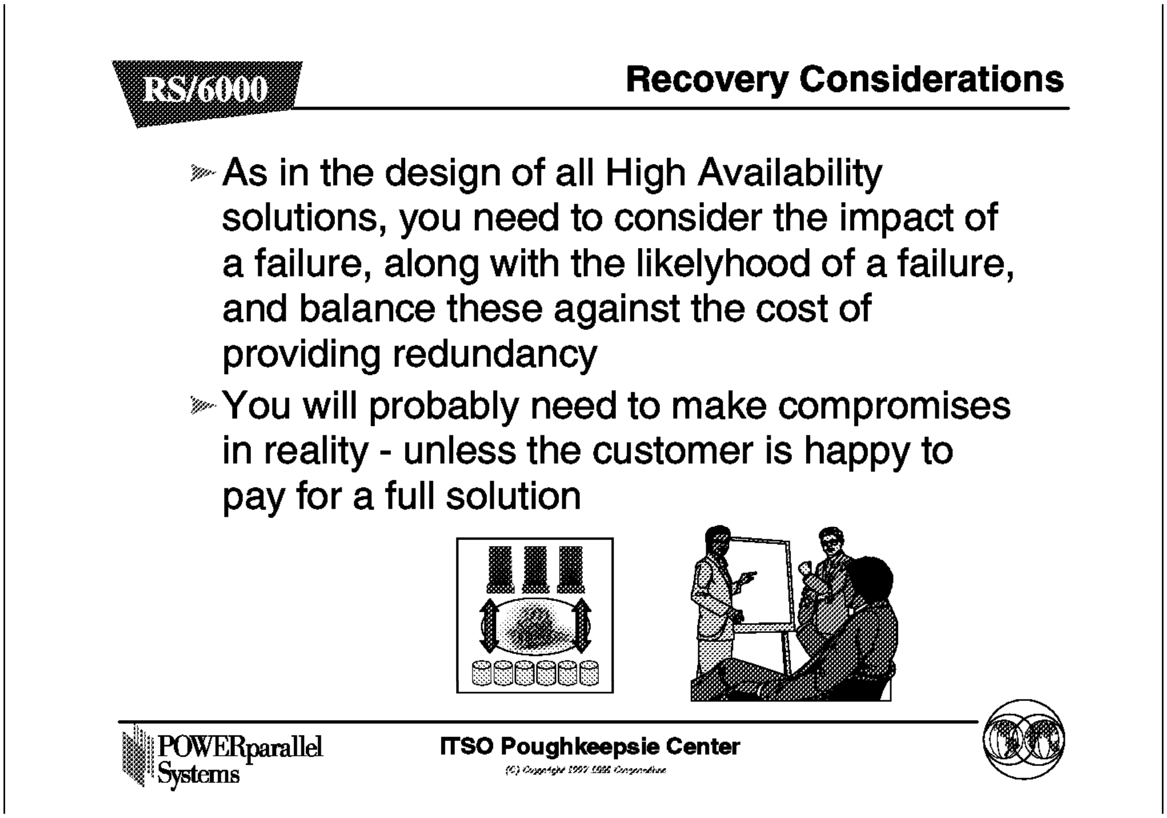
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The way in which you protect against these failures is different in each case and you may often have multiple options. These are discussed next.

### 6.5.14.1 GPFS Recovery



RS/6000

**Recovery Considerations**

- As in the design of all High Availability solutions, you need to consider the impact of a failure, along with the likelyhood of a failure, and balance these against the cost of providing redundancy
- You will probably need to make compromises in reality - unless the customer is happy to pay for a full solution

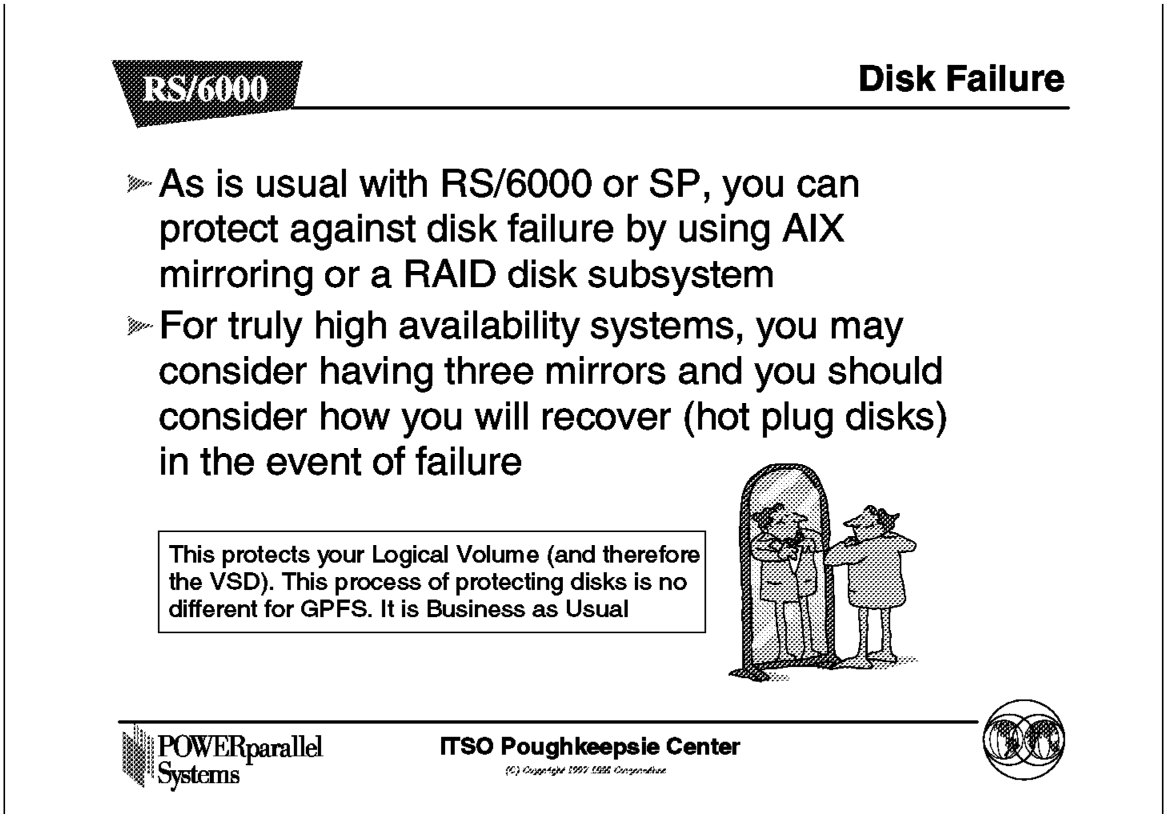
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The process for designing your system for availability is no different in the case of GPFS. You will analyze the various potential failures, protection solutions, costs, and balance these to come to a final decision.

## 6.5.15 Disk Failure



RS/6000

**Disk Failure**

- As is usual with RS/6000 or SP, you can protect against disk failure by using AIX mirroring or a RAID disk subsystem
- For truly high availability systems, you may consider having three mirrors and you should consider how you will recover (hot plug disks) in the event of failure

This protects your Logical Volume (and therefore the VSD). This process of protecting disks is no different for GPFS. It is Business as Usual

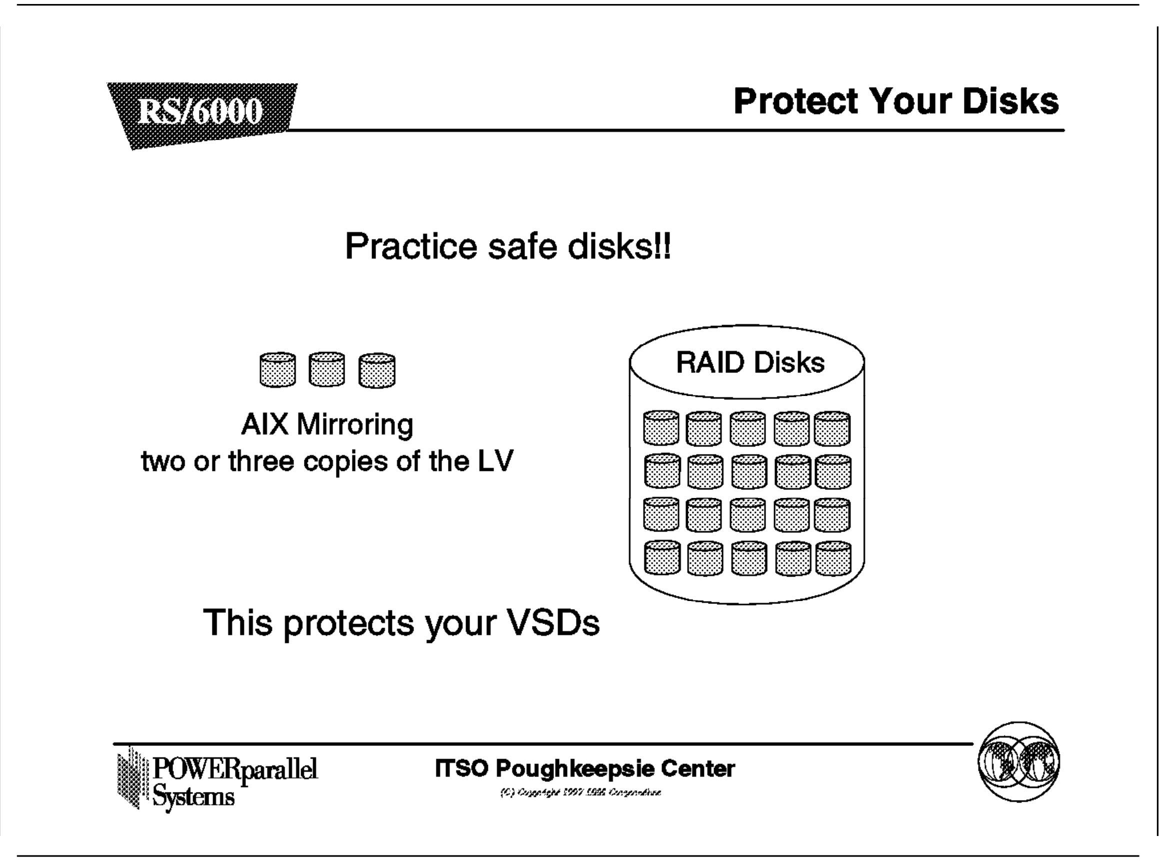
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Disk mirroring or RAID disks should always be considered. The default solution would normally be SSA disks, with a loop attached to two nodes in the GPFS pool, and using AIX mirroring of the Logical Volume.

### 6.5.16 Protect Your Disks



RS/6000

Protect Your Disks

Practice safe disks!!

AIX Mirroring
two or three copies of the LV

RAID Disks

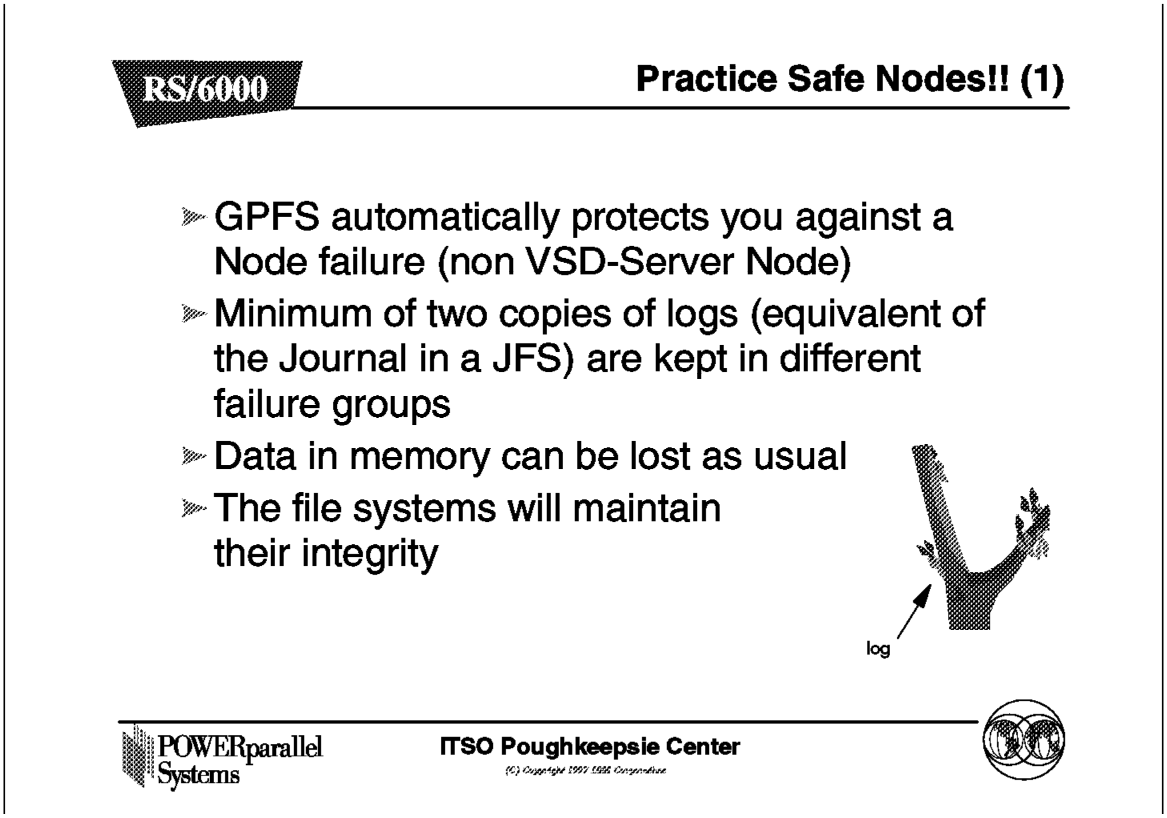
This protects your VSDs

POWERparallel Systems

ITSO Poughkeepsie Center

A RAID disk will provide for high availability, but will not provide the high performance of mirroring. RAID solutions will, however, often be cheaper.

### 6.5.17 Practice Safe Nodes



RS/6000

**Practice Safe Nodes!! (1)**

- GPFS automatically protects you against a Node failure (non VSD-Server Node)
- Minimum of two copies of logs (equivalent of the Journal in a JFS) are kept in different failure groups
- Data in memory can be lost as usual
- The file systems will maintain their integrity

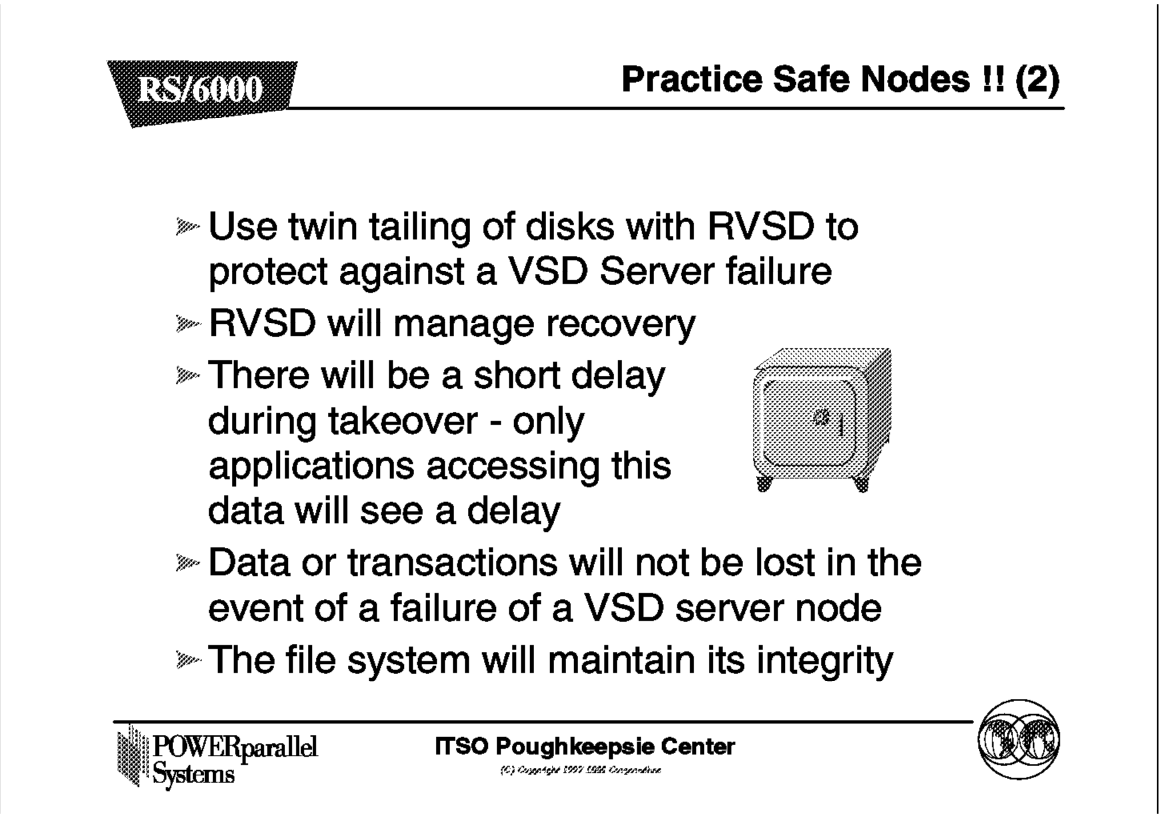
POWERparallel Systems

ITSO Poughkeepsie Center

GPFS maintains two copies of logs in separate failure groups within the SP system. These are used for recovery in the event of a power failure, for example, of one node.

GPFS provides equivalent function to the Journaling in JFS.

### 6.5.17.1 Practice Safe Nodes



RS/6000

**Practice Safe Nodes !! (2)**

- Use twin tailing of disks with RVSD to protect against a VSD Server failure
- RVSD will manage recovery
- There will be a short delay during takeover - only applications accessing this data will see a delay
- Data or transactions will not be lost in the event of a failure of a VSD server node
- The file system will maintain its integrity

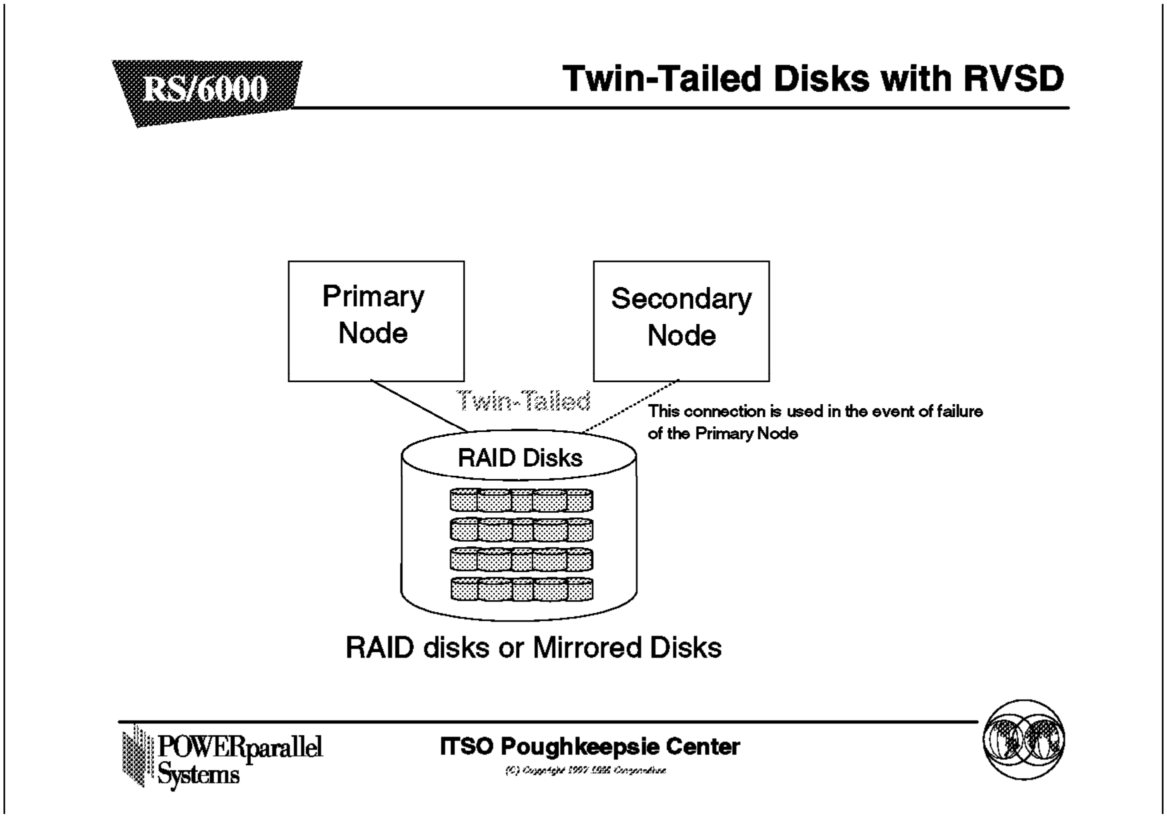
POWERparallel Systems

ITSO Poughkeepsie Center

To protect against a VSD server failure, you should use twin tailing and define RVSD to automatically recover the Volume Group.

Twin tailing your disks should normally be considered if availability is an issue, as it often is. As your file system is spread across a number of nodes, the chances of a failure should lead you to twin tailing your disks in most circumstances.

## 6.5.18 Twin-Tailed Disks



RS/6000

Twin-Tailed Disks with RVSD

Primary Node

Secondary Node

Twin-Tailed

This connection is used in the event of failure of the Primary Node

RAID Disks

RAID disks or Mirrored Disks

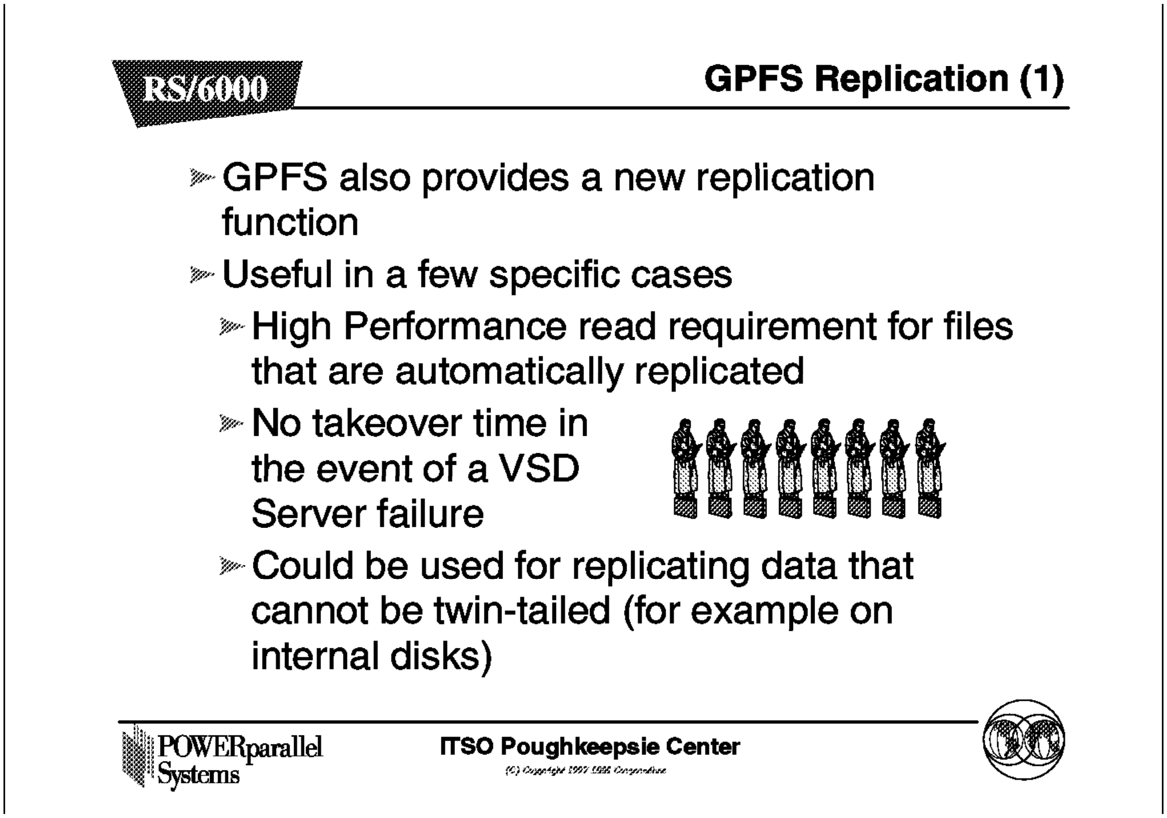
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The recovery process here is the normal RVSD process working in conjunction with the high availability infrastructure.

### 6.5.19 GPFS Replication



RS/6000

**GPFS Replication (1)**

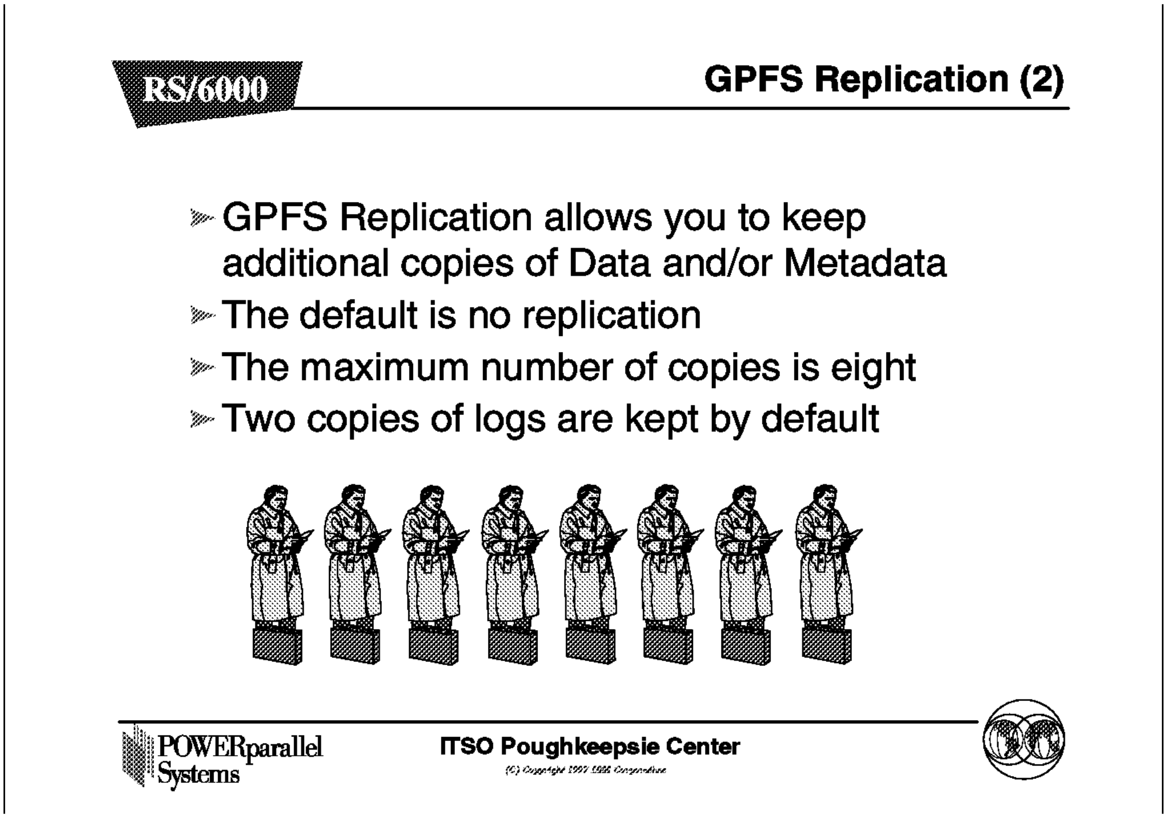
- GPFS also provides a new replication function
- Useful in a few specific cases
  - High Performance read requirement for files that are automatically replicated
  - No takeover time in the event of a VSD Server failure
  - Could be used for replicating data that cannot be twin-tailed (for example on internal disks)

POWERparallel Systems

ITSO Poughkeepsie Center

The Replication function in GPFS provides some overlap with the protection that we have already discussed. Replication will not normally be your first choice. The only obvious cases where it might prove useful are listed here.

### 6.5.19.1 GPFS Replication



RS/6000

**GPFS Replication (2)**

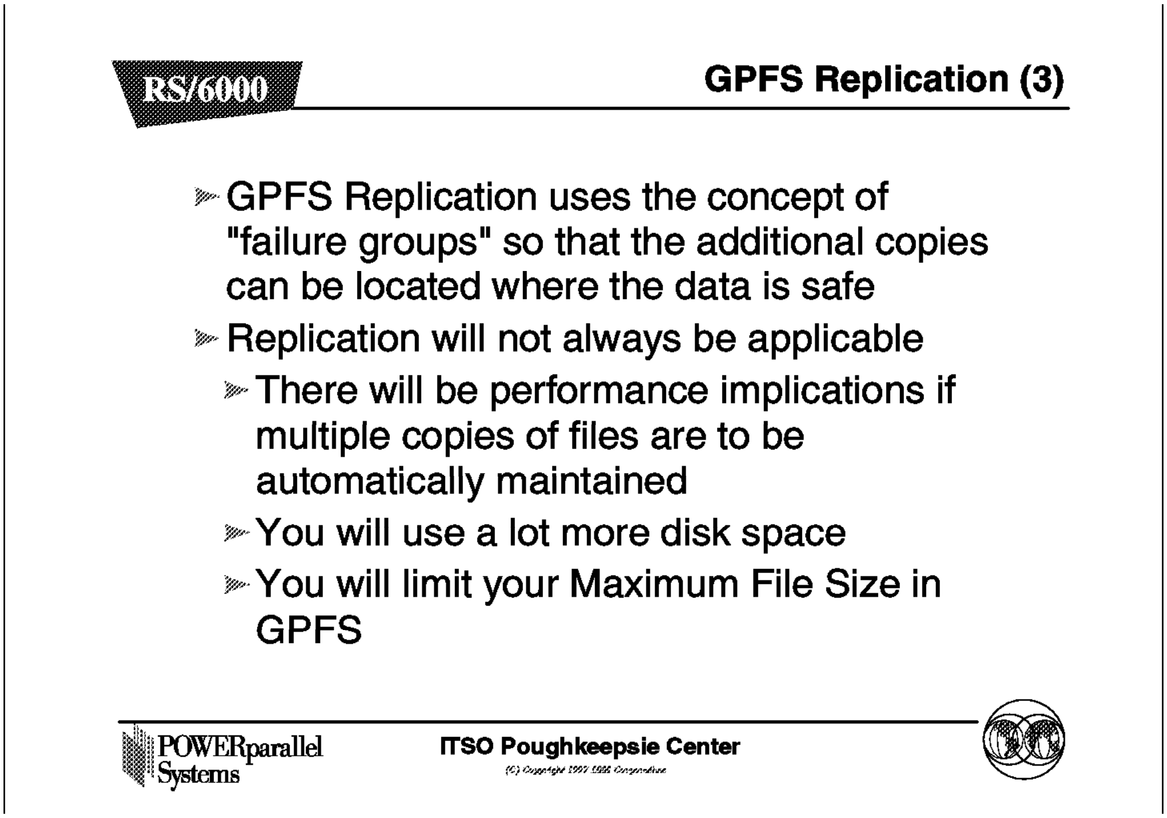
- GPFS Replication allows you to keep additional copies of Data and/or Metadata
- The default is no replication
- The maximum number of copies is eight
- Two copies of logs are kept by default

POWERparallel
Systems
ITSO Poughkeepsie Center

The default for replication is no replication. However, two copies of logs are kept anyway as already discussed.

You can choose at the file level whether you want to replicate a file and/or its metadata (i-node information).

### 6.5.19.2 GPFS Replications



RS/6000

GPFS Replication (3)

- GPFS Replication uses the concept of "failure groups" so that the additional copies can be located where the data is safe
- Replication will not always be applicable
  - There will be performance implications if multiple copies of files are to be automatically maintained
  - You will use a lot more disk space
  - You will limit your Maximum File Size in GPFS

POWERparallel Systems

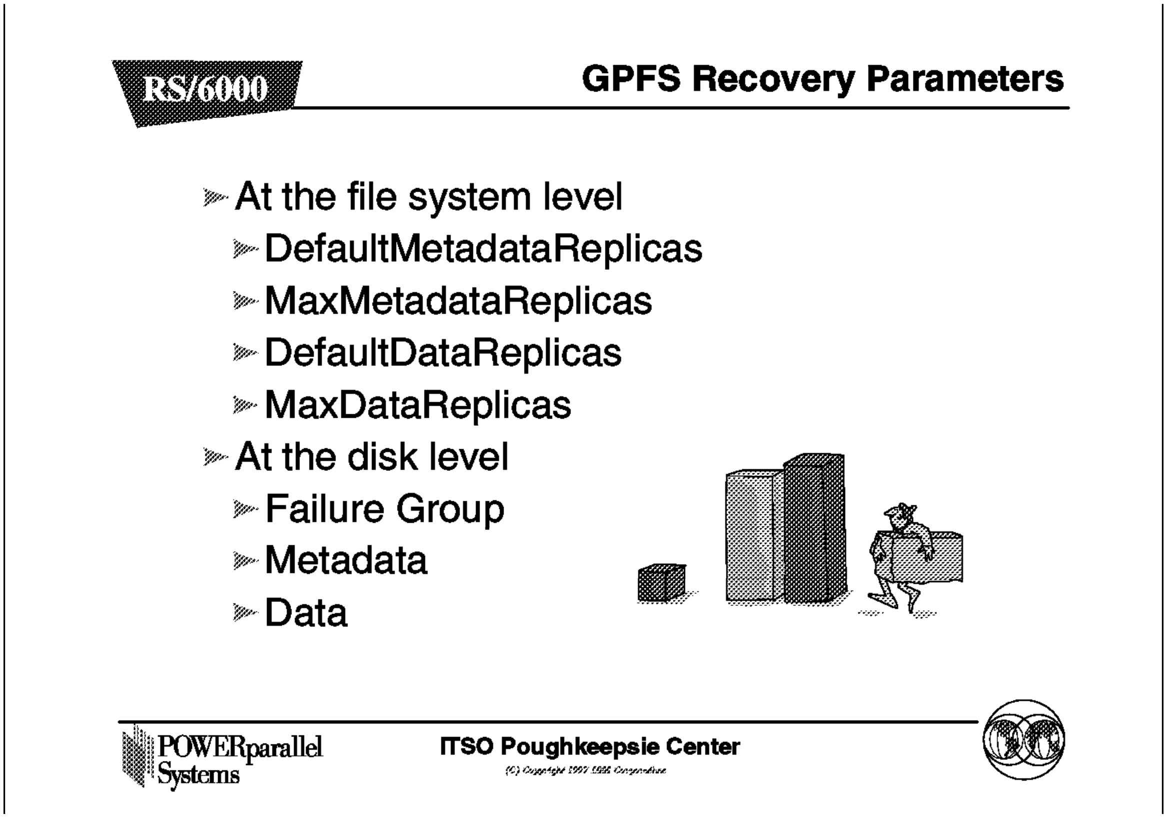
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

It would not make sense to have another copy of the data and put it on the same disk, or even on a disk attached to the same node, as there would be occasions when we could not get to the extra copy.

GPFS uses the concept of failure groups to make sure that data is protected from this kind of thing. Failure groups will be discussed in more detail later. The default failure group is at the node level, and GPFS will only put replica data in a different failure group.

### 6.5.20 GPFS Recovery Parameters



RS/6000

**GPFS Recovery Parameters**

- At the file system level
  - DefaultMetadataReplicas
  - MaxMetadataReplicas
  - DefaultDataReplicas
  - MaxDataReplicas
- At the disk level
  - Failure Group
  - Metadata
  - Data

POWERparallel Systems

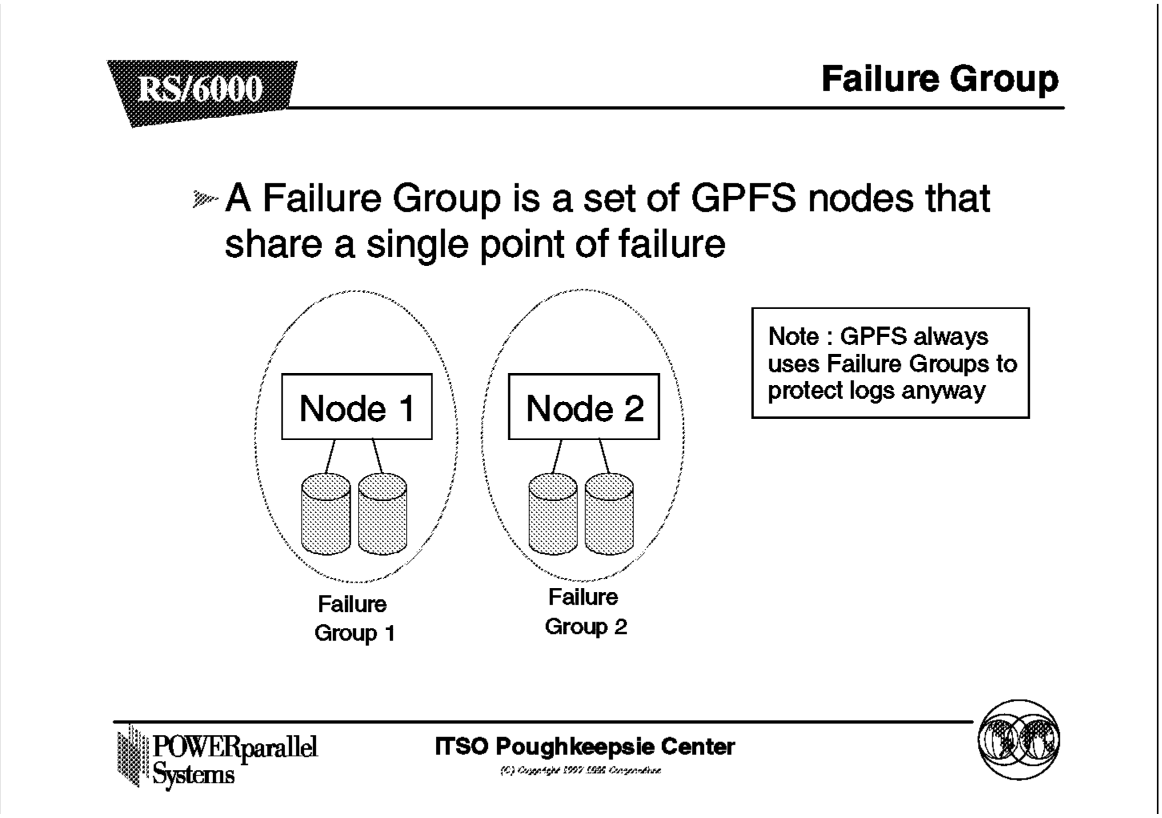
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

You have a great deal of flexibility. However, you cannot change the default or maximum replica settings for a file system once selected. You can change the replication settings at a file level.

A newly created file will always adopt the default settings.

## 6.5.21 GPFS Failure Group



RS/6000

**Failure Group**

- A Failure Group is a set of GPFS nodes that share a single point of failure

Node 1

Node 2

Failure Group 1

Failure Group 2

Note : GPFS always uses Failure Groups to protect logs anyway

POWERparallel Systems

ITSO Poughkeepsie Center

GPFS by default will assign a failure group at the node level for each disk in a GPFS file system.

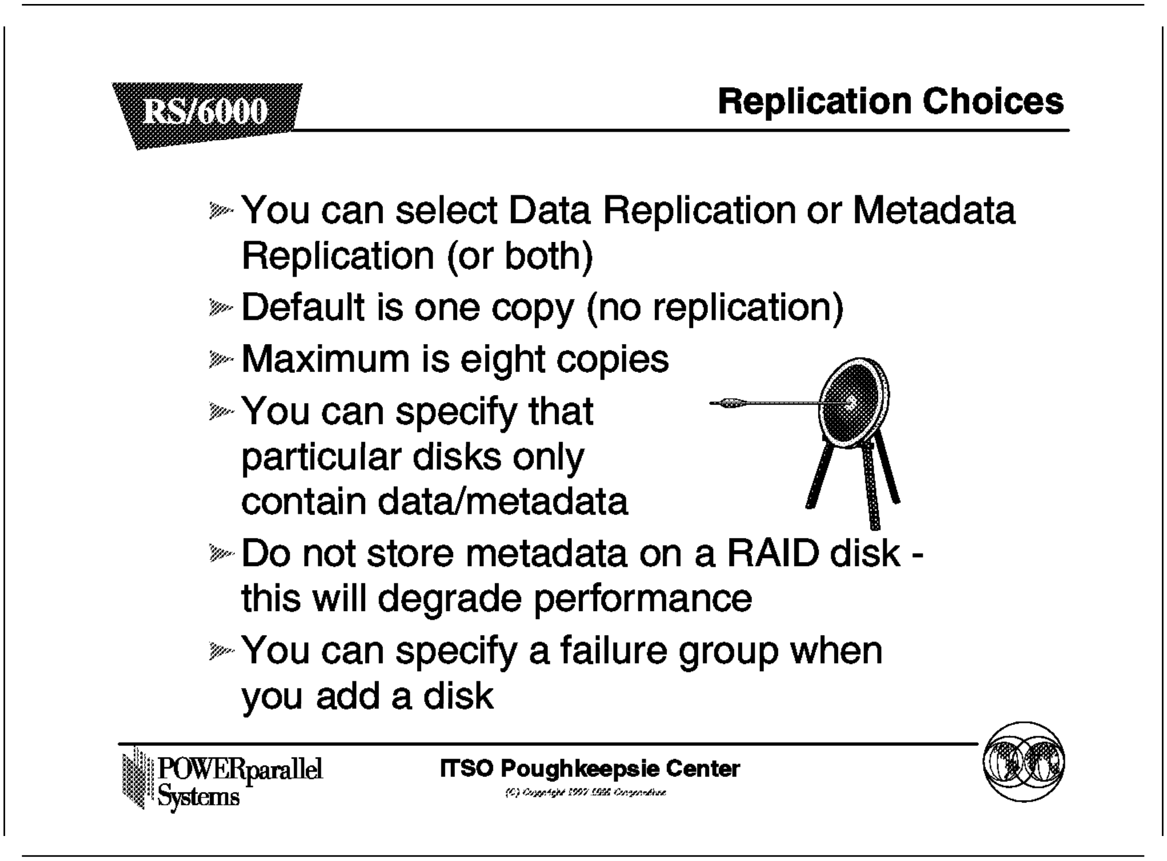
Typically, each node can be seen as a single point of failure, from the GPFS point of view, and, therefore, constitutes a failure group.

As a result, GPFS will only put replicas of data and metadata into a different failure group when these are configured.

A failure group is defined by a number which can be assigned by the user. The default number is the node number plus 1000. For example, a disk on node 7 will be placed in the failure group 1007.

Setting a value of -1 for the failure group says that any considerations with regard to failure groups will be ignored.

### 6.5.22 Replication Choices



RS/6000

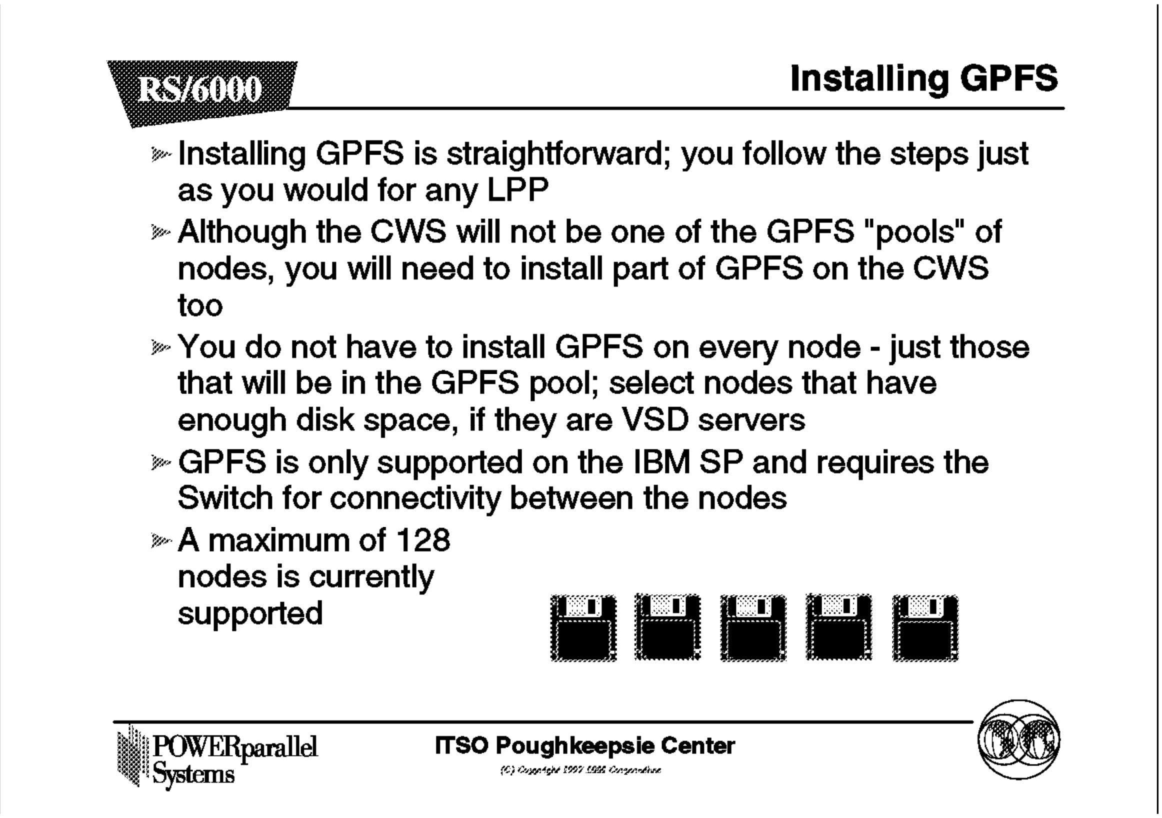
**Replication Choices**

- You can select Data Replication or Metadata Replication (or both)
- Default is one copy (no replication)
- Maximum is eight copies
- You can specify that particular disks only contain data/metadata
- Do not store metadata on a RAID disk - this will degrade performance
- You can specify a failure group when you add a disk

POWERparallel Systems
ITSO Poughkeepsie Center

Replication of data takes up a lot more space, whereas replication of metadata is less costly in terms of disk space.

## 6.6 Installing GPFS



RS/6000

**Installing GPFS**

- Installing GPFS is straightforward; you follow the steps just as you would for any LPP
- Although the CWS will not be one of the GPFS "pools" of nodes, you will need to install part of GPFS on the CWS too
- You do not have to install GPFS on every node - just those that will be in the GPFS pool; select nodes that have enough disk space, if they are VSD servers
- GPFS is only supported on the IBM SP and requires the Switch for connectivity between the nodes
- A maximum of 128 nodes is currently supported

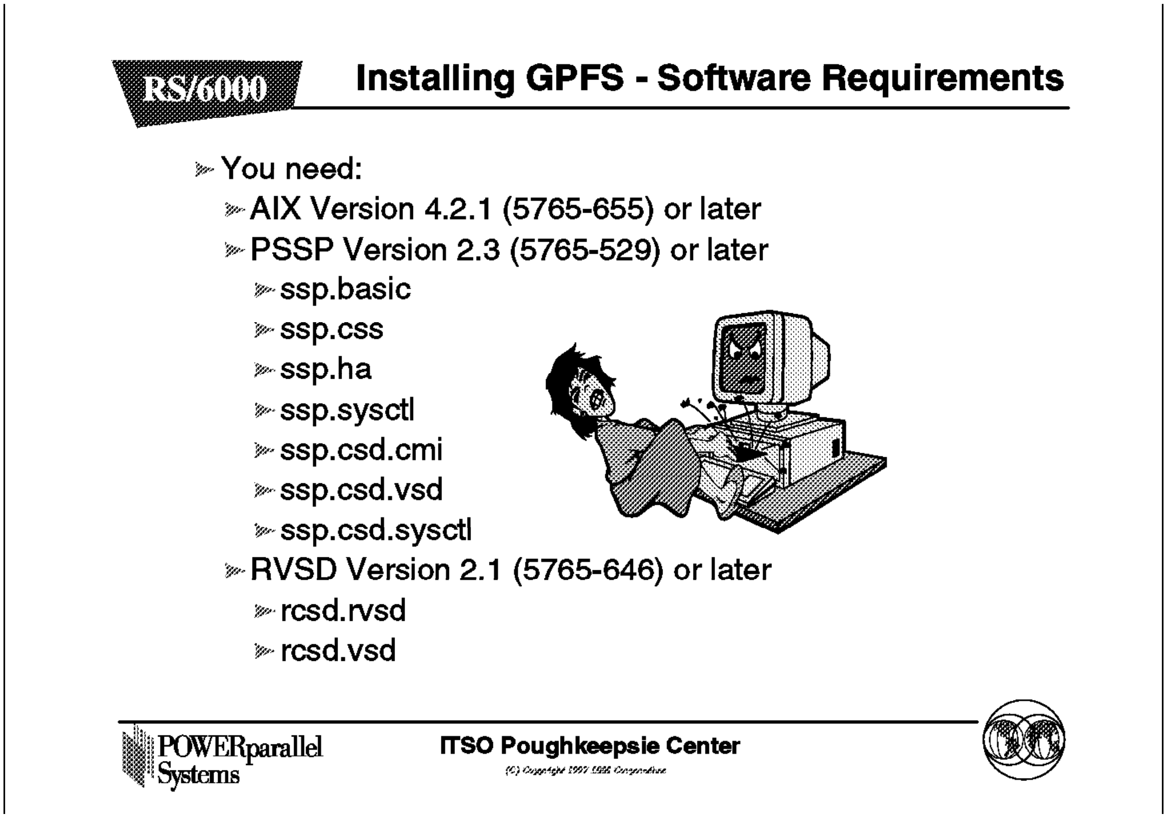
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The installation of GPFS is similar to the installation of any LPP.

## 6.6.1 Installing GPFS - Software Requirements



RS/6000

**Installing GPFS - Software Requirements**

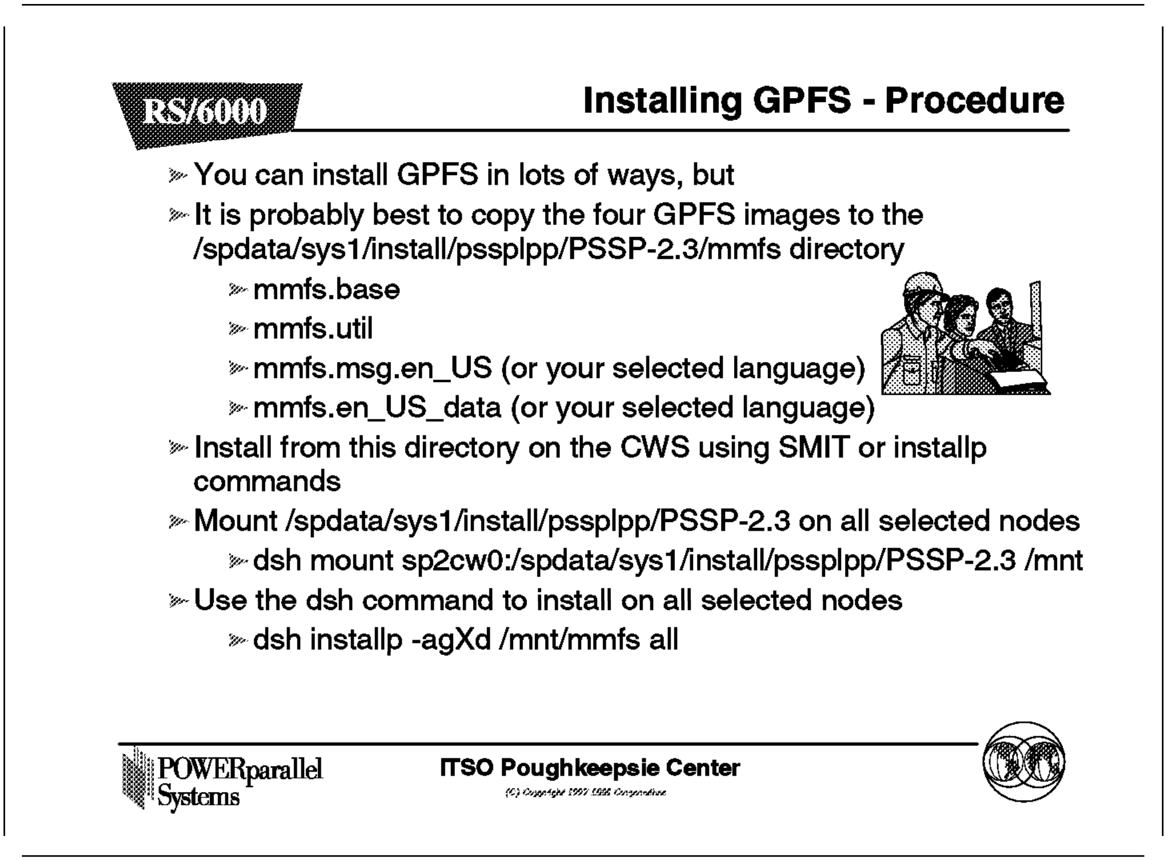
- You need:
  - AIX Version 4.2.1 (5765-655) or later
  - PSSP Version 2.3 (5765-529) or later
    - ssp.basic
    - ssp.css
    - ssp.ha
    - ssp.sysctl
    - ssp.csd.cmi
    - ssp.csd.vsd
    - ssp.csd.sysctl
  - RVSD Version 2.1 (5765-646) or later
    - rcsd.rvsd
    - rcsd.vsd

POWERparallel Systems

ITSO Poughkeepsie Center

GPFS is only supported with these software combinations.

### 6.6.2 Installing GPFS - Procedure



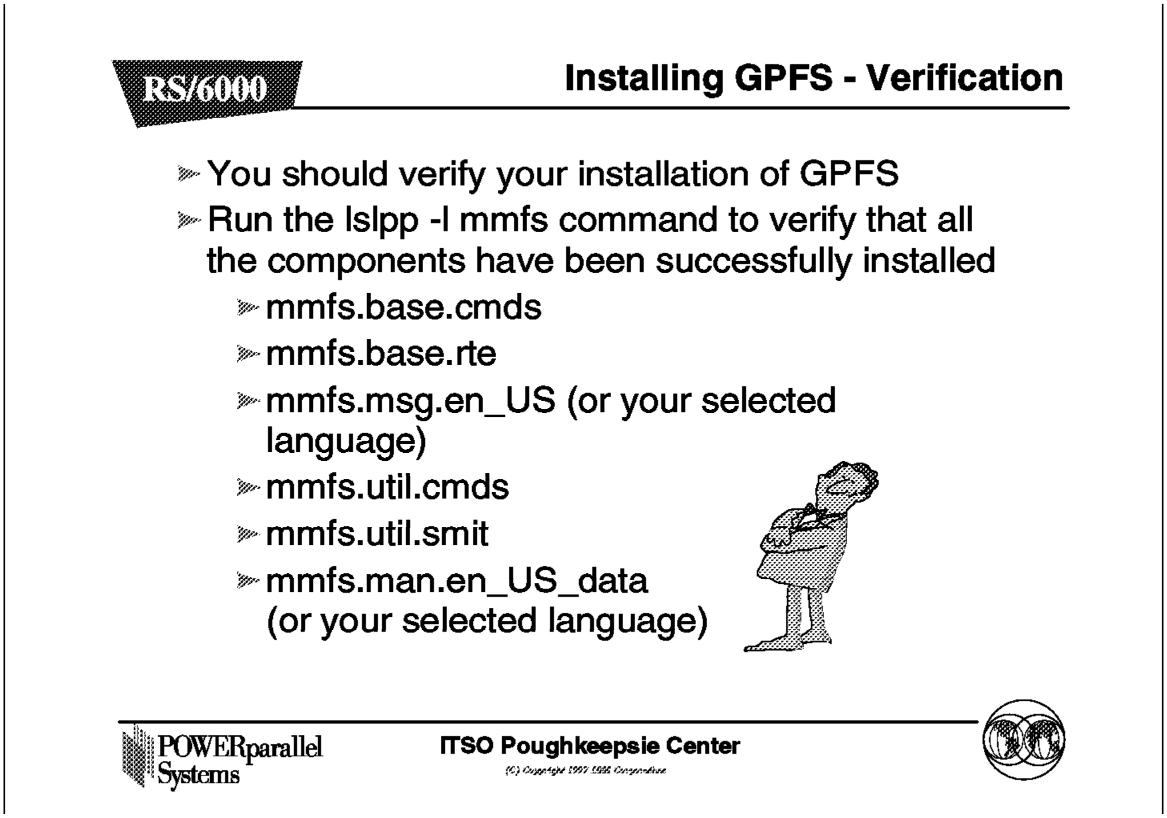
**Installing GPFS - Procedure**

- You can install GPFS in lots of ways, but
- It is probably best to copy the four GPFS images to the /spdata/sys1/install/pssplpp/PSSP-2.3/mmfs directory
  - mmfs.base
  - mmfs.util
  - mmfs.msg.en_US (or your selected language)
  - mmfs.en_US_data (or your selected language)
- Install from this directory on the CWS using SMIT or installp commands
- Mount /spdata/sys1/install/pssplpp/PSSP-2.3 on all selected nodes
  - dsh mount sp2cw0:/spdata/sys1/install/pssplpp/PSSP-2.3 /mnt
- Use the dsh command to install on all selected nodes
  - dsh installp -agXd /mnt/mmfs all

ITSO Poughkeepsie Center

Use the standard SP tools to install GPFS across the nodes in the SP.

## 6.6.3 Installing GPFS - Verification



RS/6000

**Installing GPFS - Verification**

- You should verify your installation of GPFS
- Run the lslpp -l mmfs command to verify that all the components have been successfully installed
  - mmfs.base.cmds
  - mmfs.base.rte
  - mmfs.msg.en_US (or your selected language)
  - mmfs.util.cmds
  - mmfs.util.smit
  - mmfs.man.en_US_data (or your selected language)

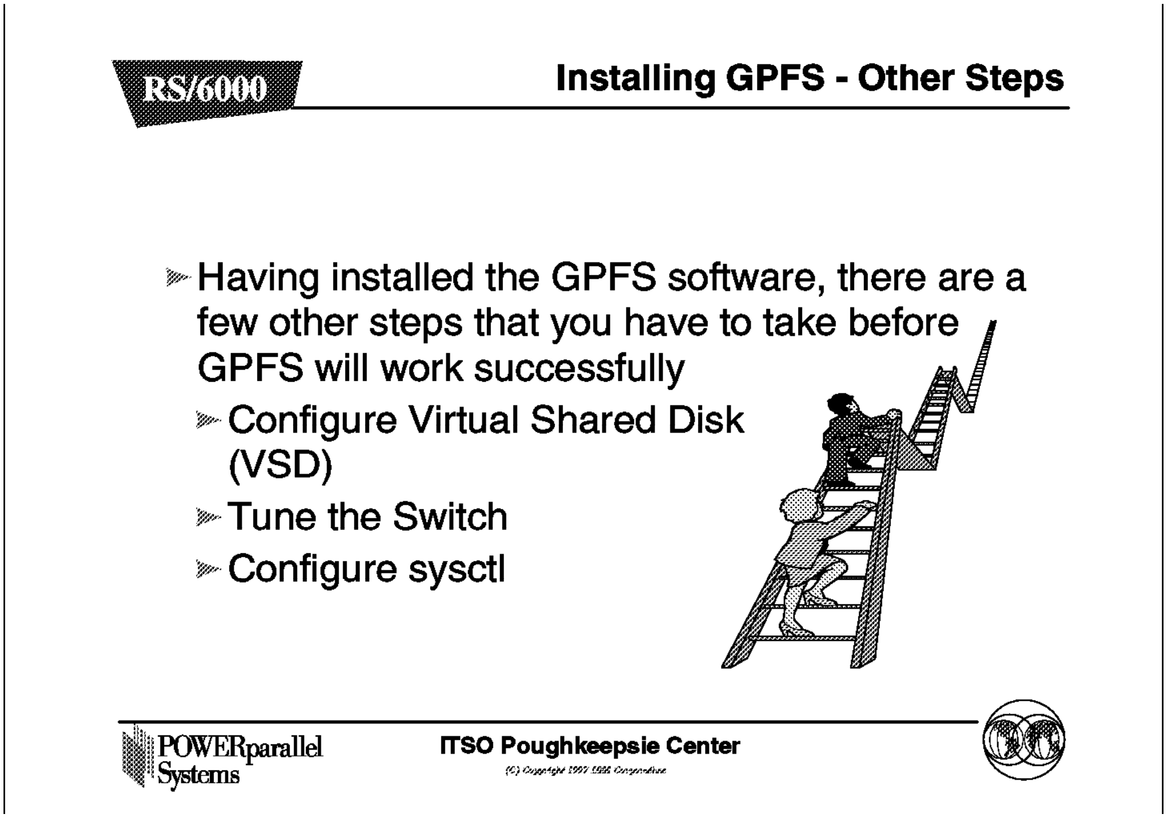
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Check that your installation was successful.

## 6.6.4 Installing GPFS - Other Steps



RS/6000

Installing GPFS - Other Steps

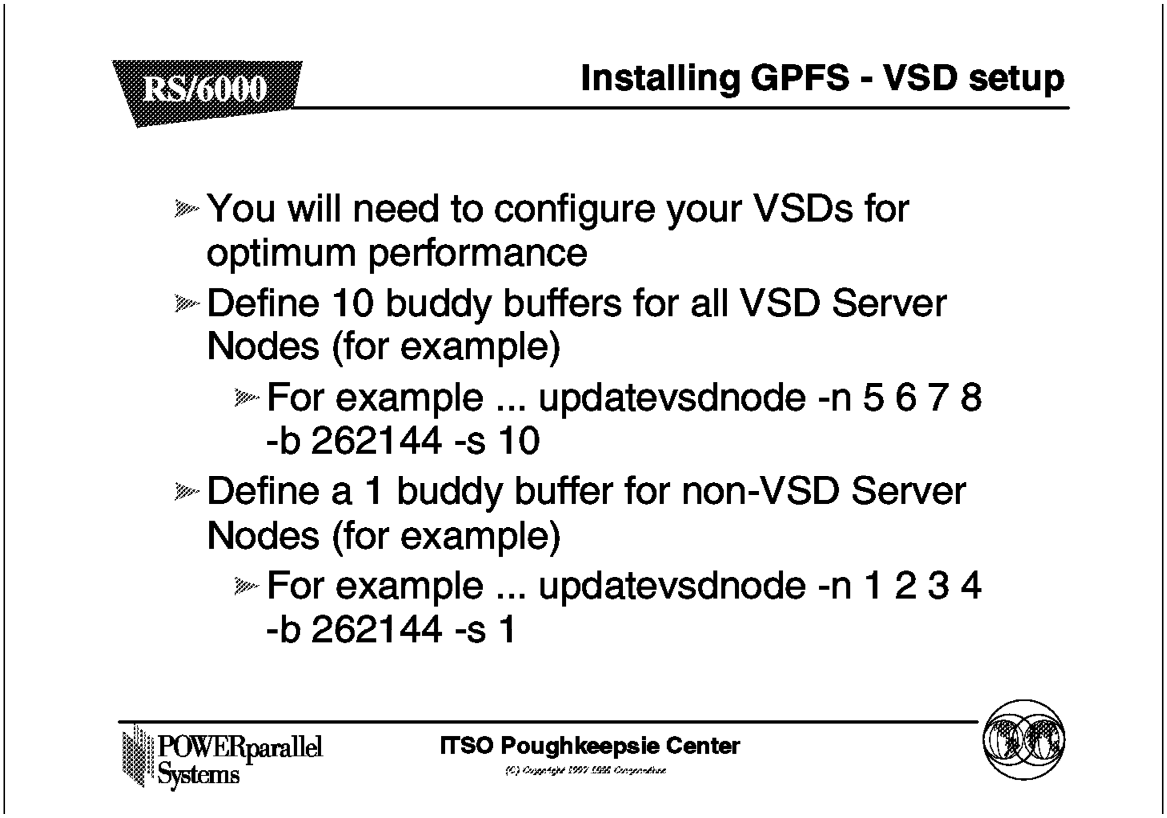
- Having installed the GPFS software, there are a few other steps that you have to take before GPFS will work successfully
  - Configure Virtual Shared Disk (VSD)
  - Tune the Switch
  - Configure sysctl

POWERparallel Systems

ITSO Poughkeepsie Center

There are a few simple but very important steps that you will need to take before you can use GPFS.

## 6.6.5 Installing GPFS - VSD Setup



RS/6000

**Installing GPFS - VSD setup**

- You will need to configure your VSDs for optimum performance
- Define 10 buddy buffers for all VSD Server Nodes (for example)
  - For example ... updatevsdnode -n 5 6 7 8 -b 262144 -s 10
- Define a 1 buddy buffer for non-VSD Server Nodes (for example)
  - For example ... updatevsdnode -n 1 2 3 4 -b 262144 -s 1

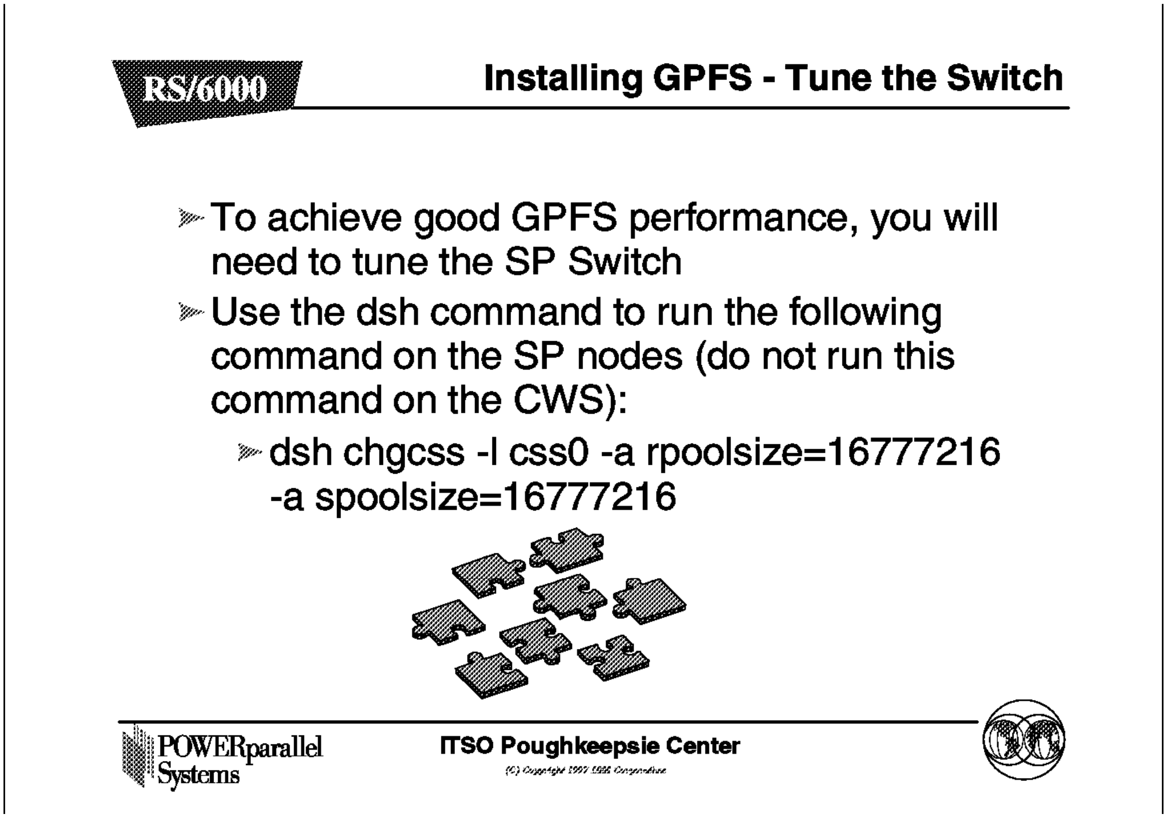
POWERparallel Systems

ITSO Poughkeepsie Center

Tune your VSDs in the normal way. Refer to the VSD documentation for full details.

Suggested starting values, in the absence of other guidance, are given here.

### 6.6.6 Installing GPFS - Tune the Switch



RS/6000

**Installing GPFS - Tune the Switch**

- To achieve good GPFS performance, you will need to tune the SP Switch
- Use the dsh command to run the following command on the SP nodes (do not run this command on the CWS):
  - dsh chgcss -l css0 -a rpoolsize=16777216 -a spoolsize=16777216

POWERparallel Systems

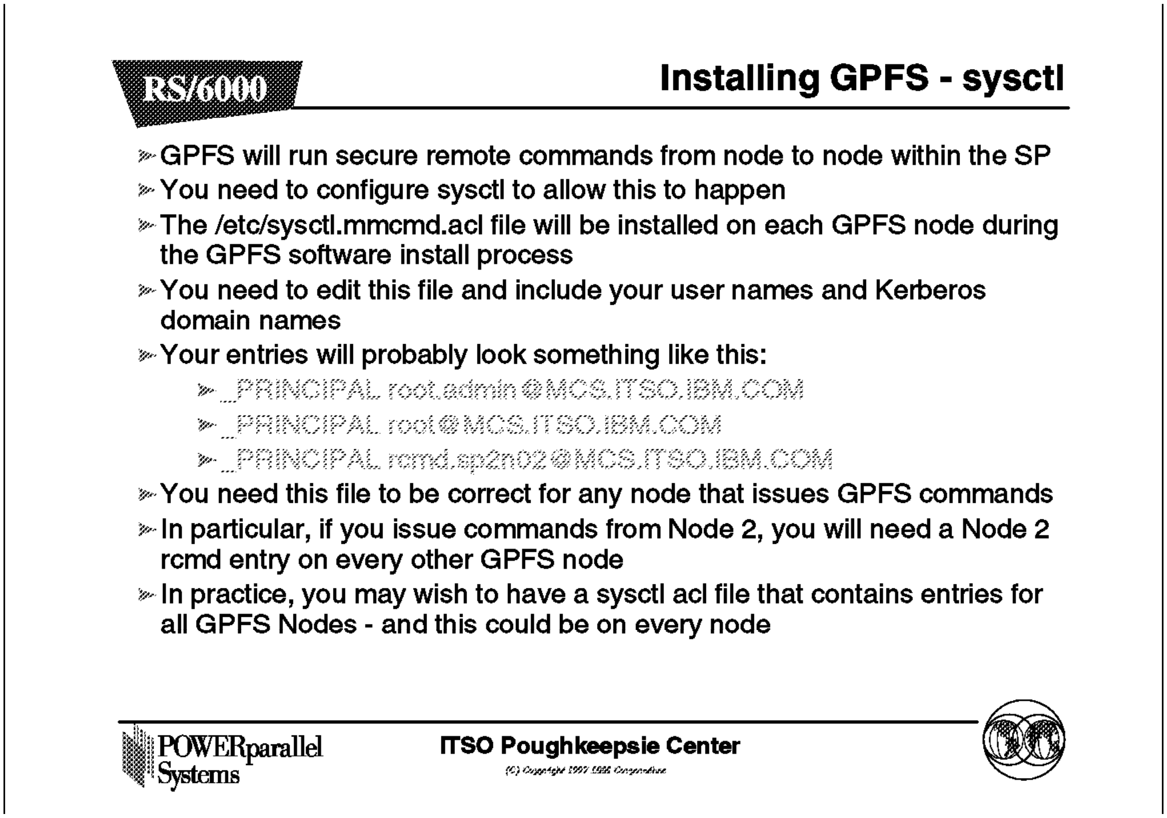
ITSO Poughkeepsie Center

Tune your switch and TCP/IP in the normal way. Refer to the PSSP documentation for full details.

Suggested starting values, in the absence of other guidance, are given here.

It is recommended that you attend the SP Performance and Tuning class if you are not familiar with these procedures.

### 6.6.7 Installing GPFS - sysctl



RS/6000

**Installing GPFS - sysctl**

- GPFS will run secure remote commands from node to node within the SP
- You need to configure sysctl to allow this to happen
- The /etc/sysctl.mmcmd.acl file will be installed on each GPFS node during the GPFS software install process
- You need to edit this file and include your user names and Kerberos domain names
- Your entries will probably look something like this:
  - _PRINCIPAL root.admin@MCS.ITSO.IBM.COM
  - _PRINCIPAL root@MCS.ITSO.IBM.COM
  - _PRINCIPAL rcmd.sp2n02@MCS.ITSO.IBM.COM
- You need this file to be correct for any node that issues GPFS commands
- In particular, if you issue commands from Node 2, you will need a Node 2 rcmd entry on every other GPFS node
- In practice, you may wish to have a sysctl acl file that contains entries for all GPFS Nodes - and this could be on every node

POWERparallel Systems

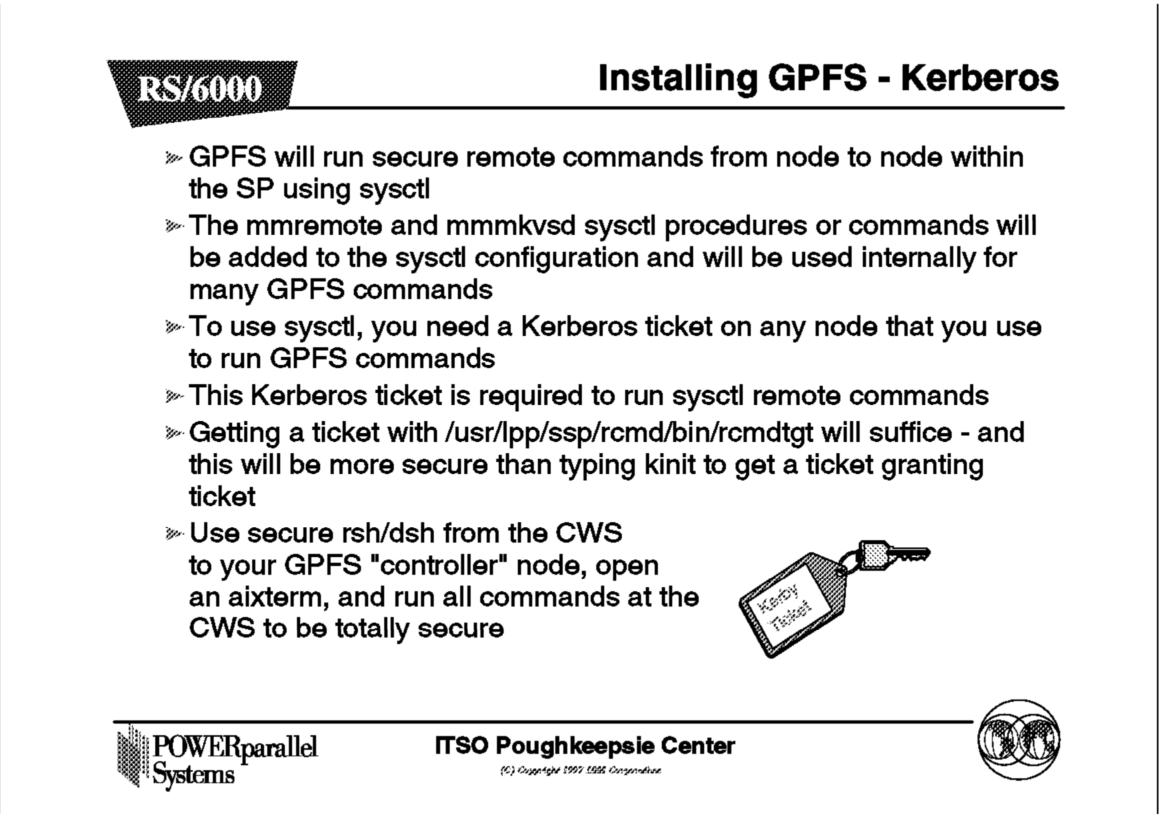
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

GPFS will not work without sysctl. Check carefully that sysctl is configured and working properly before going any further.

GPFS will exhibit strange errors if you do not have sysctl configured.

### 6.6.8 Installing GPFS - Kerberos



RS/6000

**Installing GPFS - Kerberos**

- GPFS will run secure remote commands from node to node within the SP using sysctl
- The mmremote and mmmkvsd sysctl procedures or commands will be added to the sysctl configuration and will be used internally for many GPFS commands
- To use sysctl, you need a Kerberos ticket on any node that you use to run GPFS commands
- This Kerberos ticket is required to run sysctl remote commands
- Getting a ticket with /usr/lpp/ssp/rcmd/bin/rcmdtgt will suffice - and this will be more secure than typing kinit to get a ticket granting ticket
- Use secure rsh/dsh from the CWS to your GPFS "controller" node, open an aixterm, and run all commands at the CWS to be totally secure

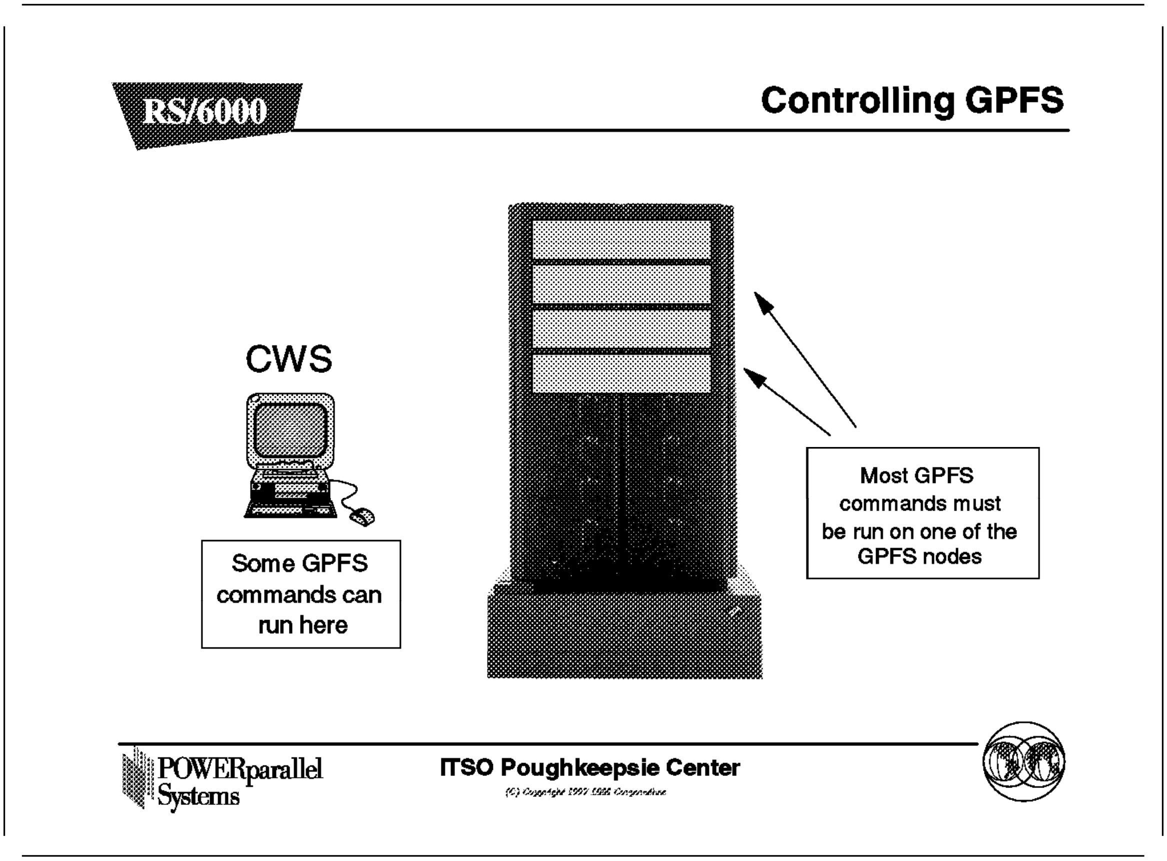
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

To use sysctl, you need a Kerberos ticket. Rather than type in a kinit command on the command line when working on a GPFS node, a remote ticket granting ticket will be a better option.

## 6.7 Configuring and Controlling GPFS



RS/6000

**Controlling GPFS**

CWS

Some GPFS commands can run here

Most GPFS commands must be run on one of the GPFS nodes

POWERparallel Systems

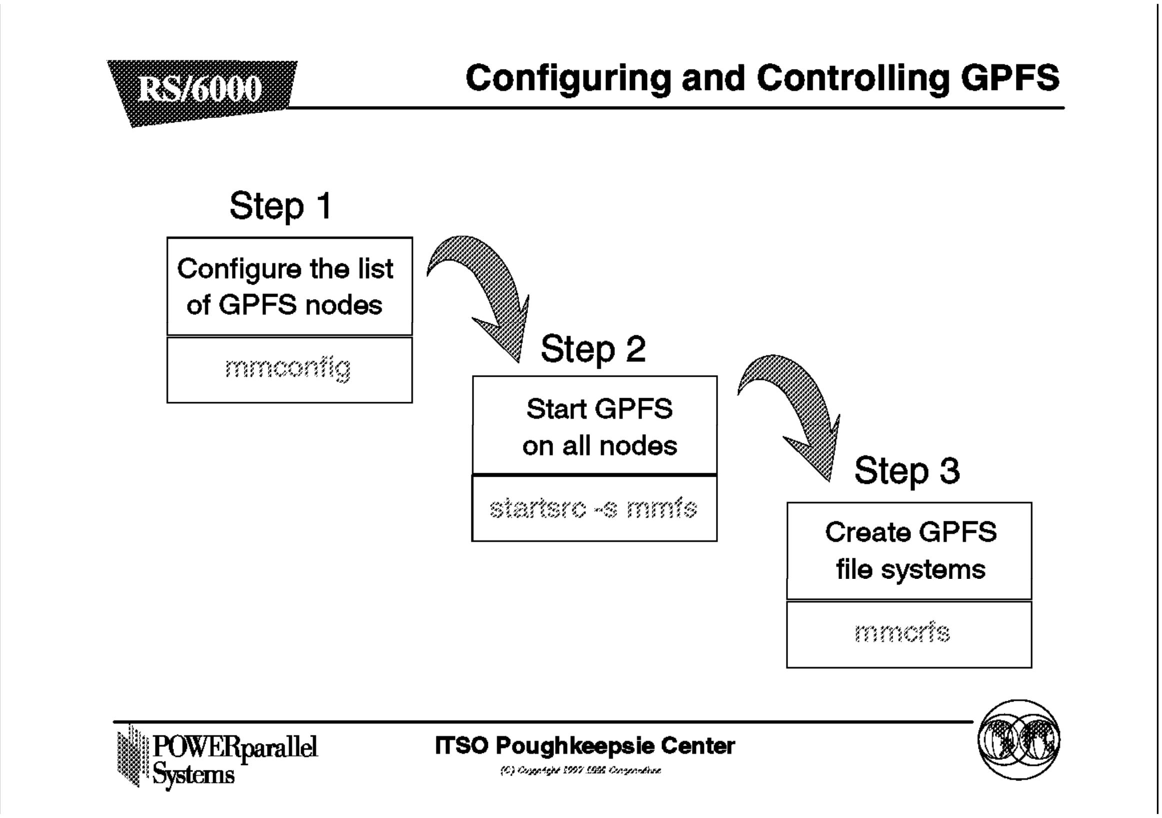
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

One of the things to note about GPFS is that much of the system management cannot be performed directly at the CWS. You need to be on a GPFS node to execute most GPFS commands.

You can use rsh/dsh from the CWS to achieve this.

### 6.7.1 Configuring and Controlling GPFS



RS/6000

Configuring and Controlling GPFS

Step 1

Configure the list of GPFS nodes

mmconfig

Step 2

Start GPFS on all nodes

startsrc -s mmfs

Step 3

Create GPFS file systems

mmcrfs

POWERparallel Systems

ITSO Poughkeepsie Center

To get going with GPFS there are three distinct things you need to do, from a high-level point of view, shown in the figure.

There is a ″one off″ configuration setup command that you will have to run to tell GPFS which nodes are in the GPFS pool. You also provide some other configuration information.

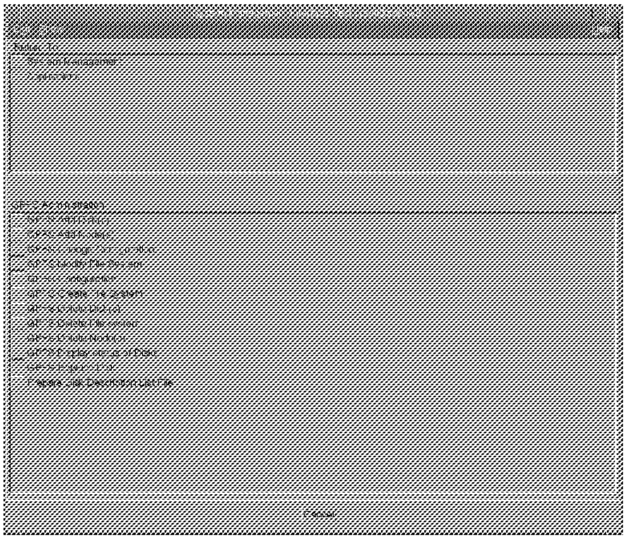
Once this is complete, you can start GPFS (mmfs) on all GPFS nodes.

You are now ready to create and mount GPFS file systems across your GPFS nodes within the SP.

## 6.7.2 GPFS Main SMIT Panel

RS/6000

GPFS Main SMIT Panel



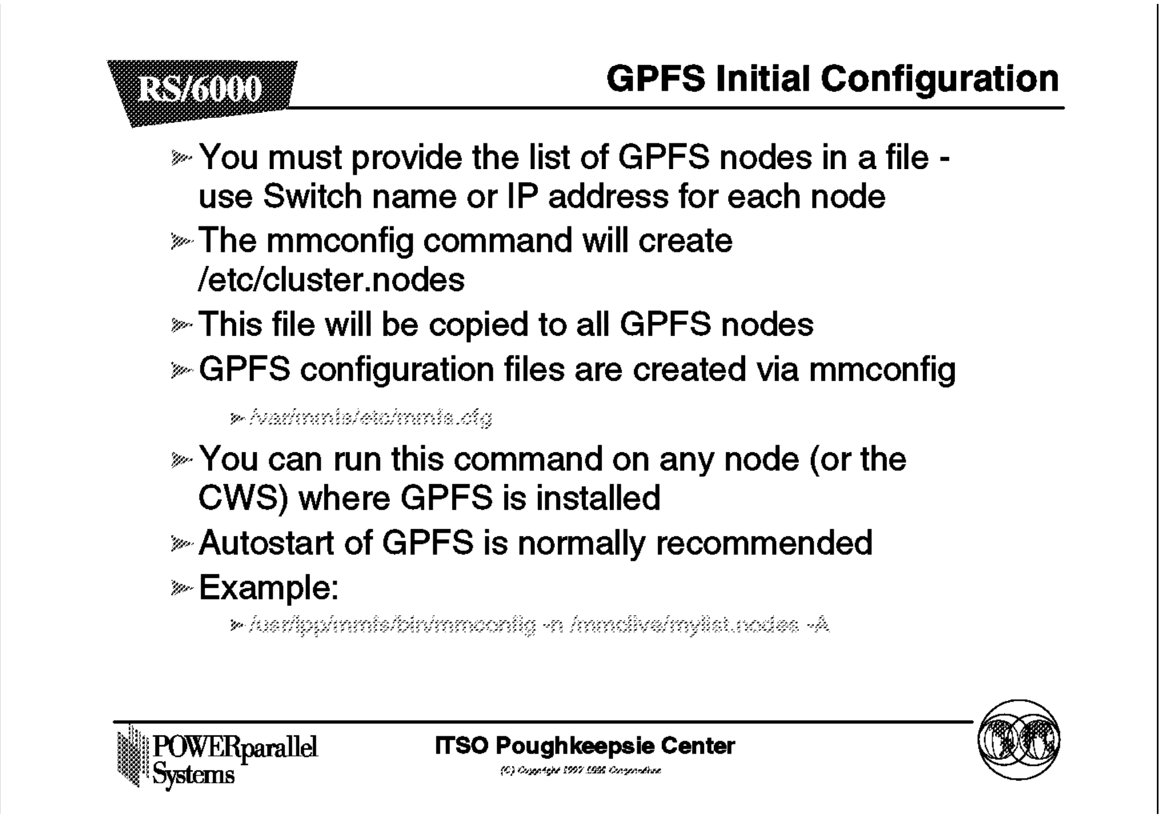
POWERparallel Systems

ITSO Poughkeepsie Center

Many of the tasks that you will wish to perform in GPFS can be achieved either through the use of SMIT panels, or via commands that you issue on the command line.

Most commands or SMIT commands will need to be run on one of the GPFS nodes, not the CWS.

### 6.7.3 GPFS Initial Configuration



GPFS Initial Configuration

- You must provide the list of GPFS nodes in a file - use Switch name or IP address for each node
- The mmconfig command will create /etc/cluster.nodes
- This file will be copied to all GPFS nodes
- GPFS configuration files are created via mmconfig
  - */var/mmfs/etc/mmfs.cfg*
- You can run this command on any node (or the CWS) where GPFS is installed
- Autostart of GPFS is normally recommended
- Example:
  - */usr/lpp/mmfs/bin/mmconfig -n /mmdive/mylist.nodes -A*

ITSO Poughkeepsie Center

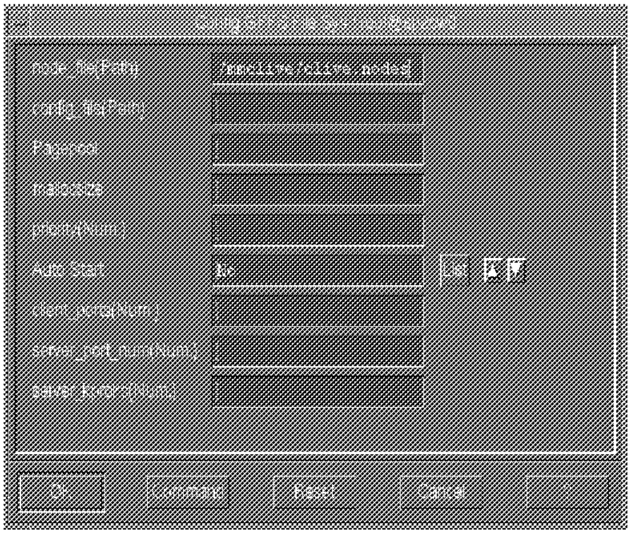
(C) Copyright 1997 IBM Corporation

You must choose a number of nodes for each GPFS file system. This number will be used to create allocation regions for accessing files in the file system. Choosing a number that is too low (less than the number of nodes that may be used in the future) will potentially impact performance.

## 6.7.4 GPFS Configuration Using SMIT

RS/6000

GPFS Configuration Using SMIT



node_file(Path) /mmalive/alive.nodes

config_file(Path)

Pagepool

mallocsize

priority(Num.)

Auto Start No

client_port(Num.)

server_port_num(Num.)

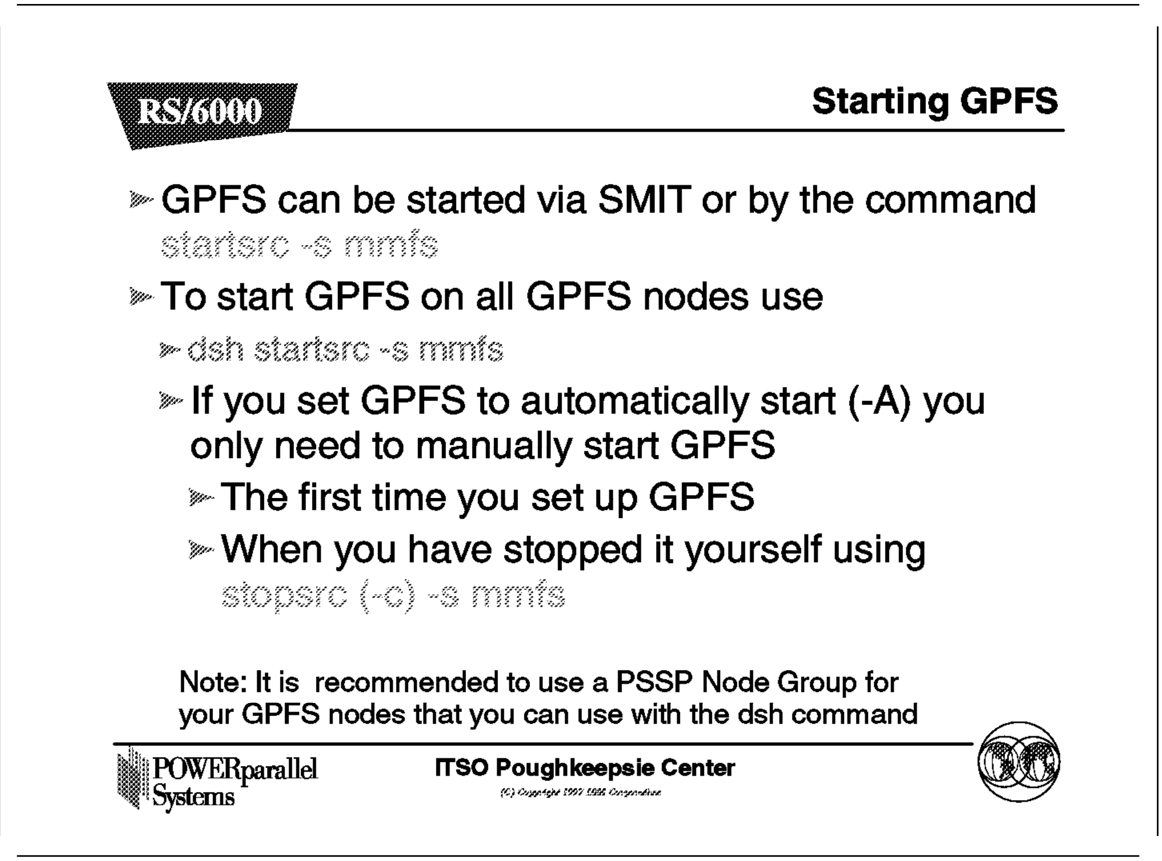
server_kprocs(Num.)

POWERparallel Systems

ITSO Poughkeepsie Center

This SMIT panel shows you the information that you need to provide while creating your initial GPFS configuration.

## 6.7.5 Starting GPFS



RS/6000

**Starting GPFS**

- GPFS can be started via SMIT or by the command startsrc -s mmfs
- To start GPFS on all GPFS nodes use
  - dsh startsrc -s mmfs
  - If you set GPFS to automatically start (-A) you only need to manually start GPFS
    - The first time you set up GPFS
    - When you have stopped it yourself using stopsrc (-c) -s mmfs

Note: It is recommended to use a PSSP Node Group for your GPFS nodes that you can use with the dsh command

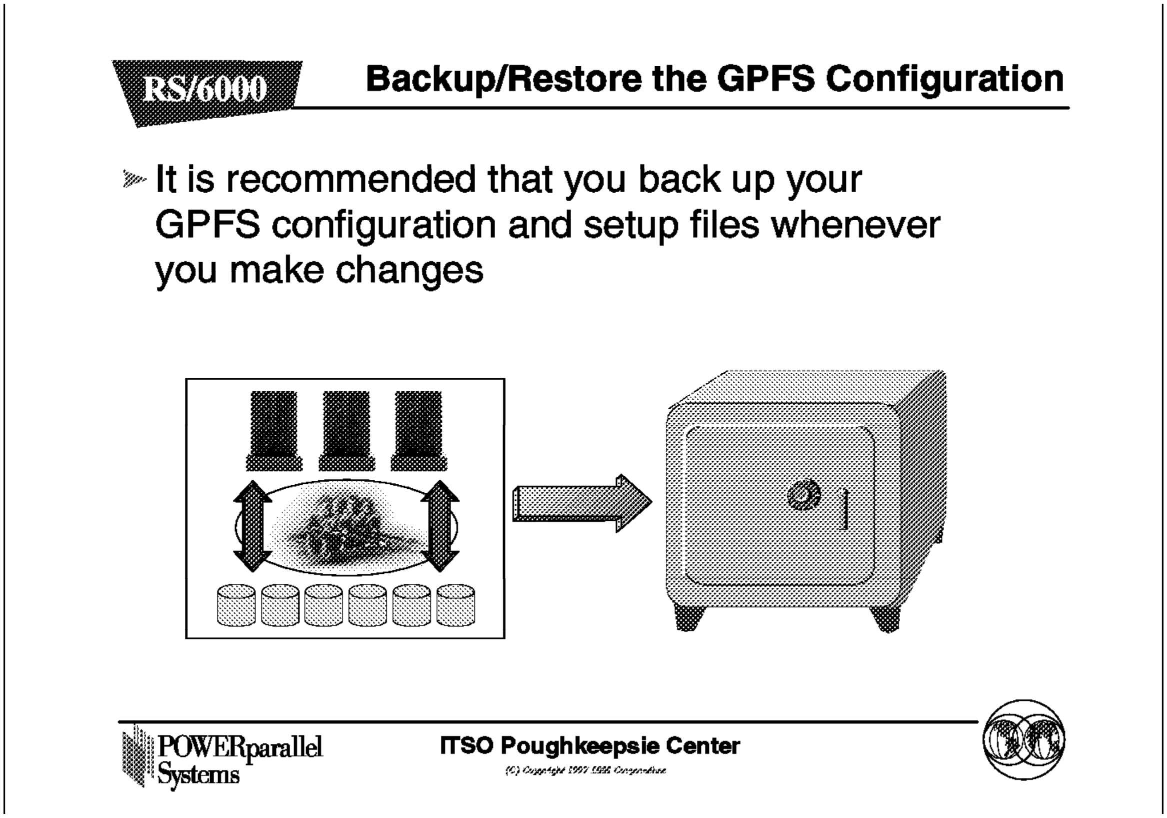
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Having configured GPFS, you can now start GPFS on your GPFS nodes. Your CWS will not be one of your GPFS pools and you will not run GPFS on the CWS.

## 6.7.6 Backup/Restore the Configuration



RS/6000

### Backup/Restore the GPFS Configuration

- It is recommended that you back up your GPFS configuration and setup files whenever you make changes

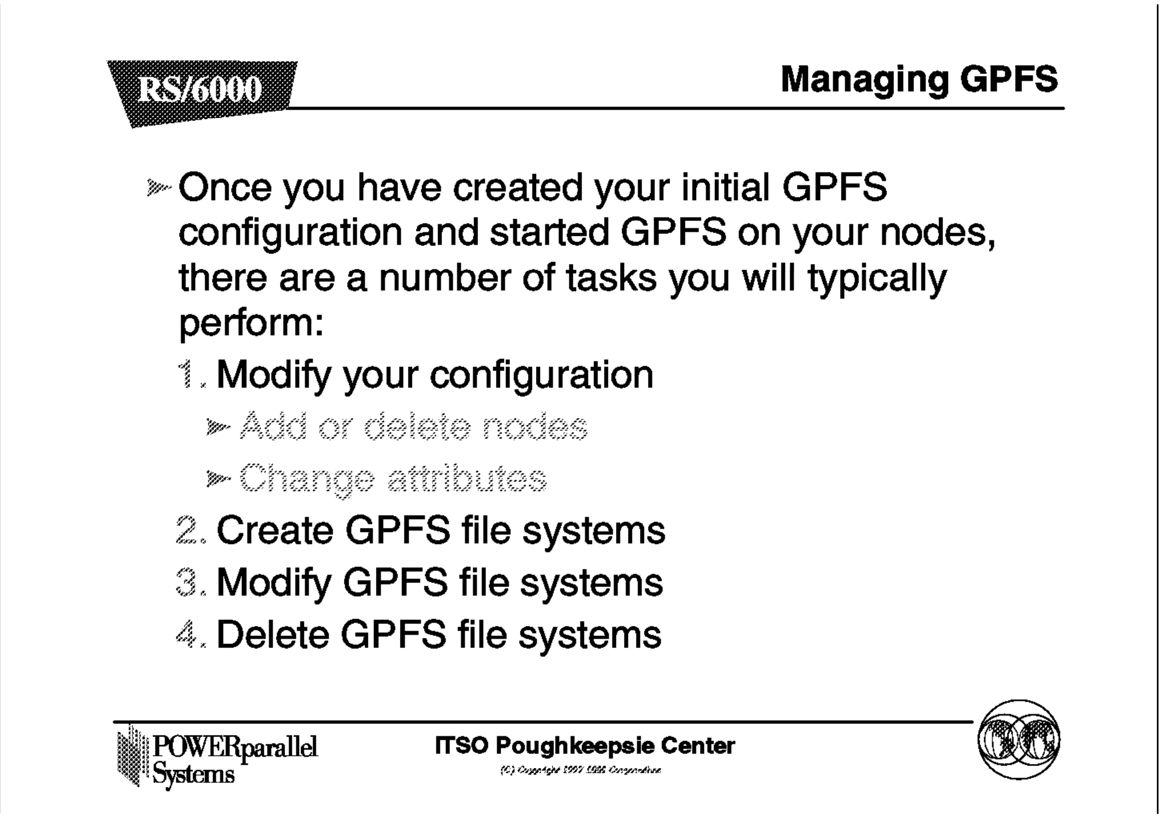
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Once you have a working GPFS system, and as you add GPFS file systems, make sure that you back up any data and configuration files for recovery purposes.

## 6.8 Managing GPFS



RS/6000

### Managing GPFS

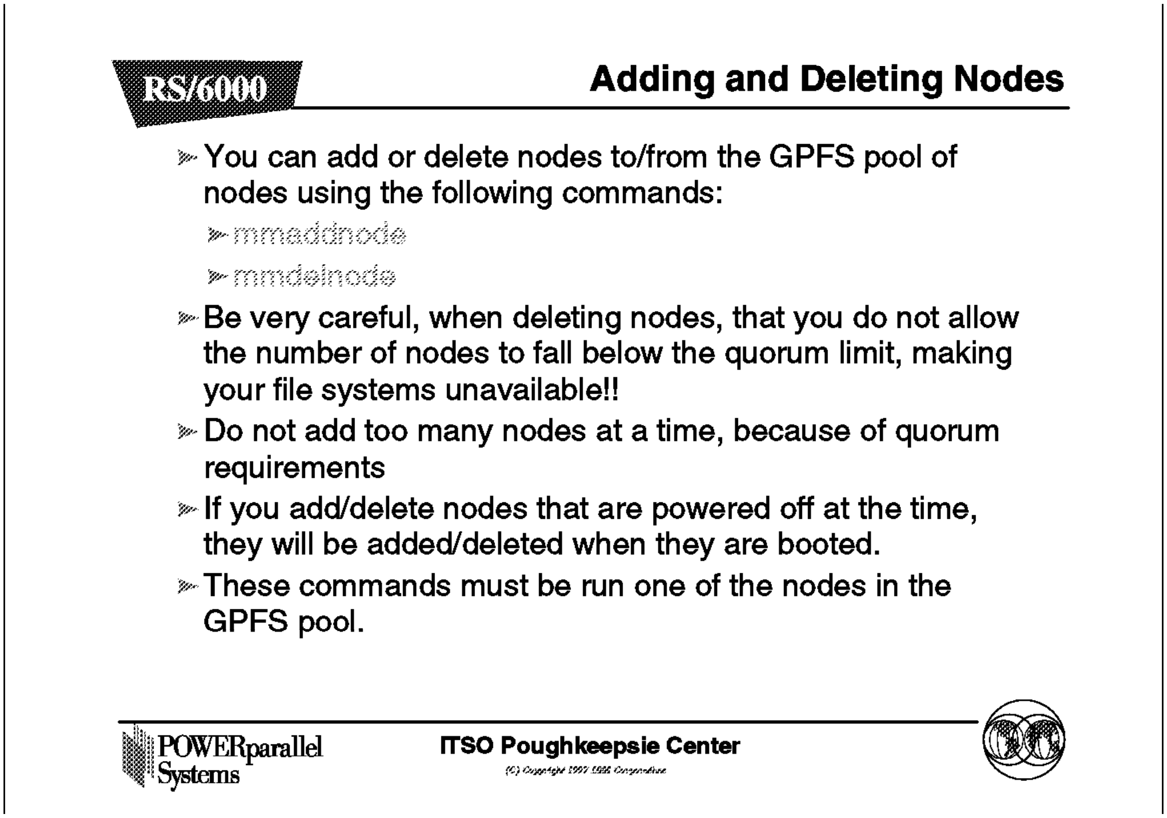
- Once you have created your initial GPFS configuration and started GPFS on your nodes, there are a number of tasks you will typically perform:
  1. Modify your configuration
     - Add or delete nodes
     - Change attributes
  2. Create GPFS file systems
  3. Modify GPFS file systems
  4. Delete GPFS file systems

POWERparallel Systems

ITSO Poughkeepsie Center

We will consider the options in this area a later. These are typically the kinds of tasks that you will want to perform.

## 6.8.1 Adding and Deleting Nodes



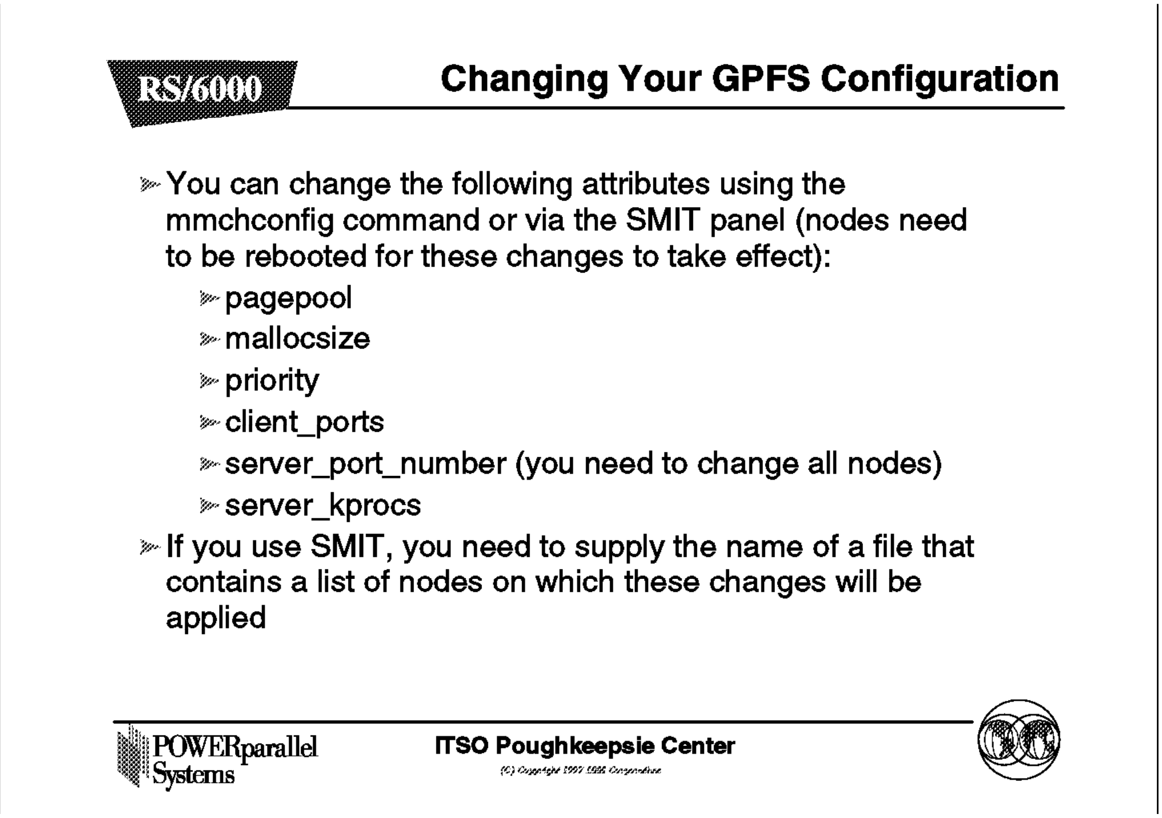
**Adding and Deleting Nodes**

- You can add or delete nodes to/from the GPFS pool of nodes using the following commands:
  - mmaddnode
  - mmdelnode
- Be very careful, when deleting nodes, that you do not allow the number of nodes to fall below the quorum limit, making your file systems unavailable!!
- Do not add too many nodes at a time, because of quorum requirements
- If you add/delete nodes that are powered off at the time, they will be added/deleted when they are booted.
- These commands must be run one of the nodes in the GPFS pool.

ITSO Poughkeepsie Center

If you add nodes into the GPFS pool and these nodes have disks so that they can act as VSD servers, be aware that new files will be striped across these additional disks. Old files will not be striped across these new disks unless you restripe the file system, which is not recommended. It is better to start with the correct number of nodes.

### 6.8.2 Changing Your GPFS Configuration



RS/6000

## Changing Your GPFS Configuration

- You can change the following attributes using the mmchconfig command or via the SMIT panel (nodes need to be rebooted for these changes to take effect):
  - pagepool
  - mallocsize
  - priority
  - client_ports
  - server_port_number (you need to change all nodes)
  - server_kprocs
- If you use SMIT, you need to supply the name of a file that contains a list of nodes on which these changes will be applied

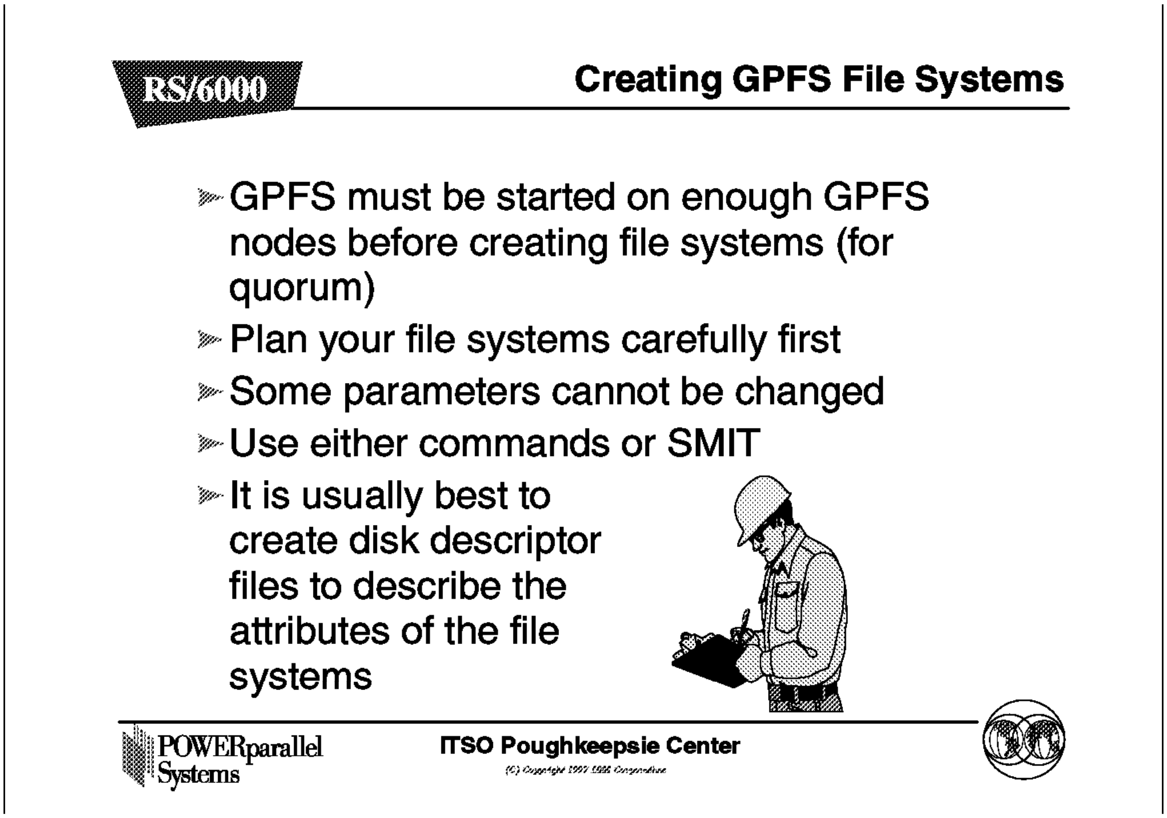
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Some aspects of the GPFS overall configuration can be changed.

## 6.9 Creating GPFS File Systems



RS/6000

### Creating GPFS File Systems

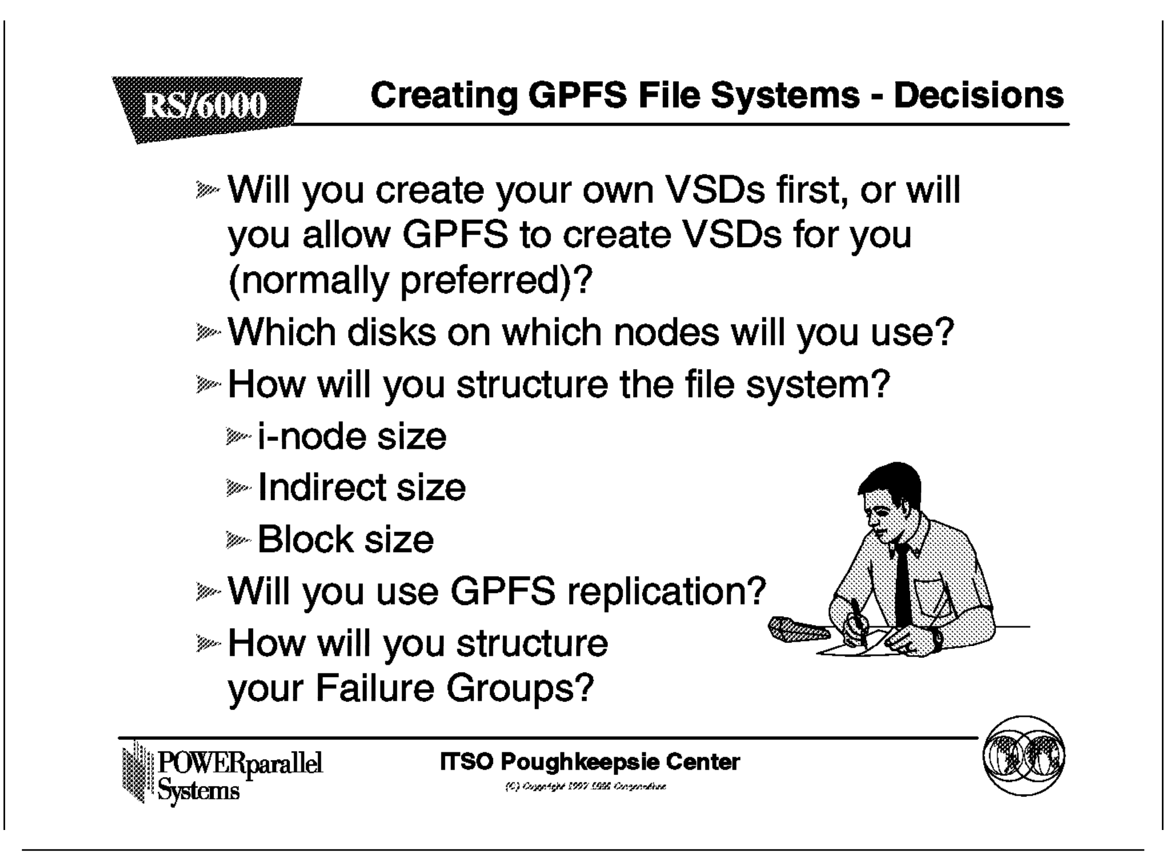
- GPFS must be started on enough GPFS nodes before creating file systems (for quorum)
- Plan your file systems carefully first
- Some parameters cannot be changed
- Use either commands or SMIT
- It is usually best to create disk descriptor files to describe the attributes of the file systems

POWERparallel Systems

ITSO Poughkeepsie Center

You are now ready to create a GPFS file system to store data on the GPFS nodes. Plan what you will do carefully. Many decisions cannot be reversed.

### 6.9.1 Creating GPFS File Systems - Decisions



RS/6000

**Creating GPFS File Systems - Decisions**

- Will you create your own VSDs first, or will you allow GPFS to create VSDs for you (normally preferred)?
- Which disks on which nodes will you use?
- How will you structure the file system?
  - i-node size
  - Indirect size
  - Block size
- Will you use GPFS replication?
- How will you structure your Failure Groups?

POWERparallel Systems

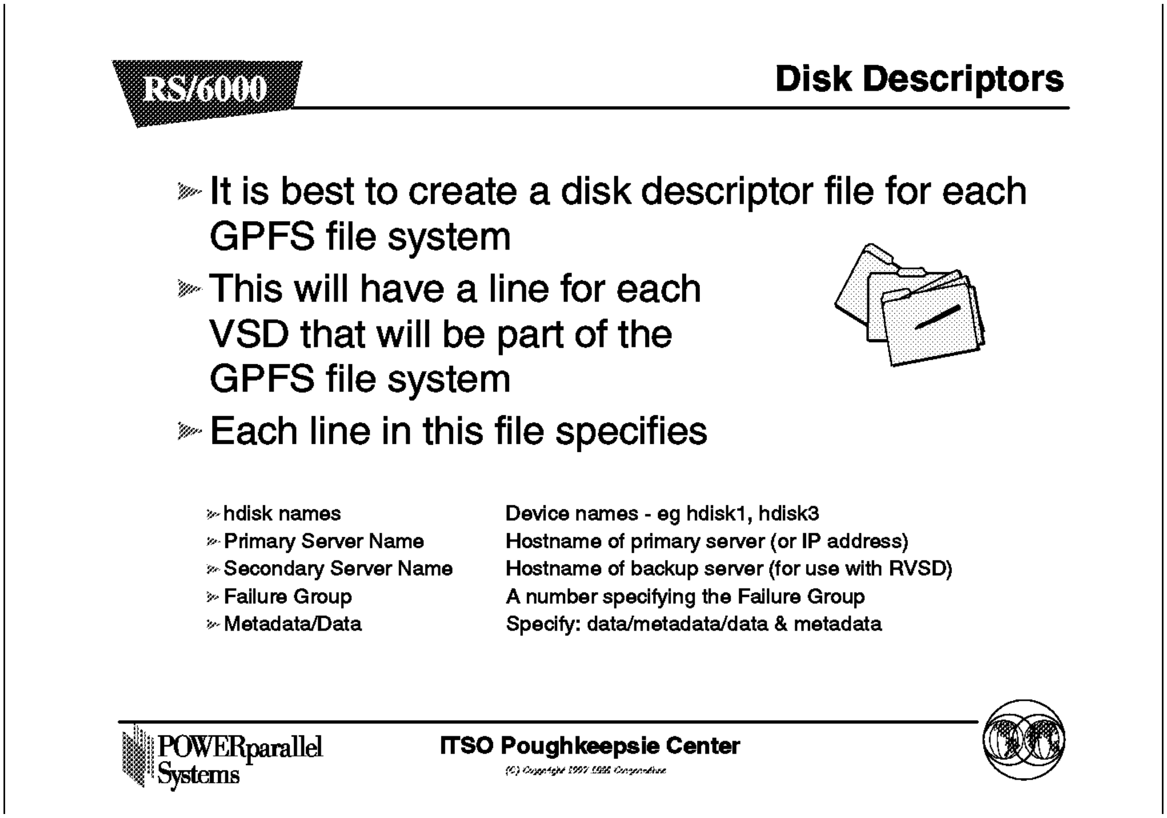
**ITSO Poughkeepsie Center**

(C) Copyright 1997 1998 Corporation

Unless you have good reasons to do otherwise, let GPFS create the VSDs for you.

Do not use replication as your first choice for providing availability.

### 6.9.2 Disk Descriptors



RS/6000

## Disk Descriptors

- It is best to create a disk descriptor file for each GPFS file system
- This will have a line for each VSD that will be part of the GPFS file system
- Each line in this file specifies

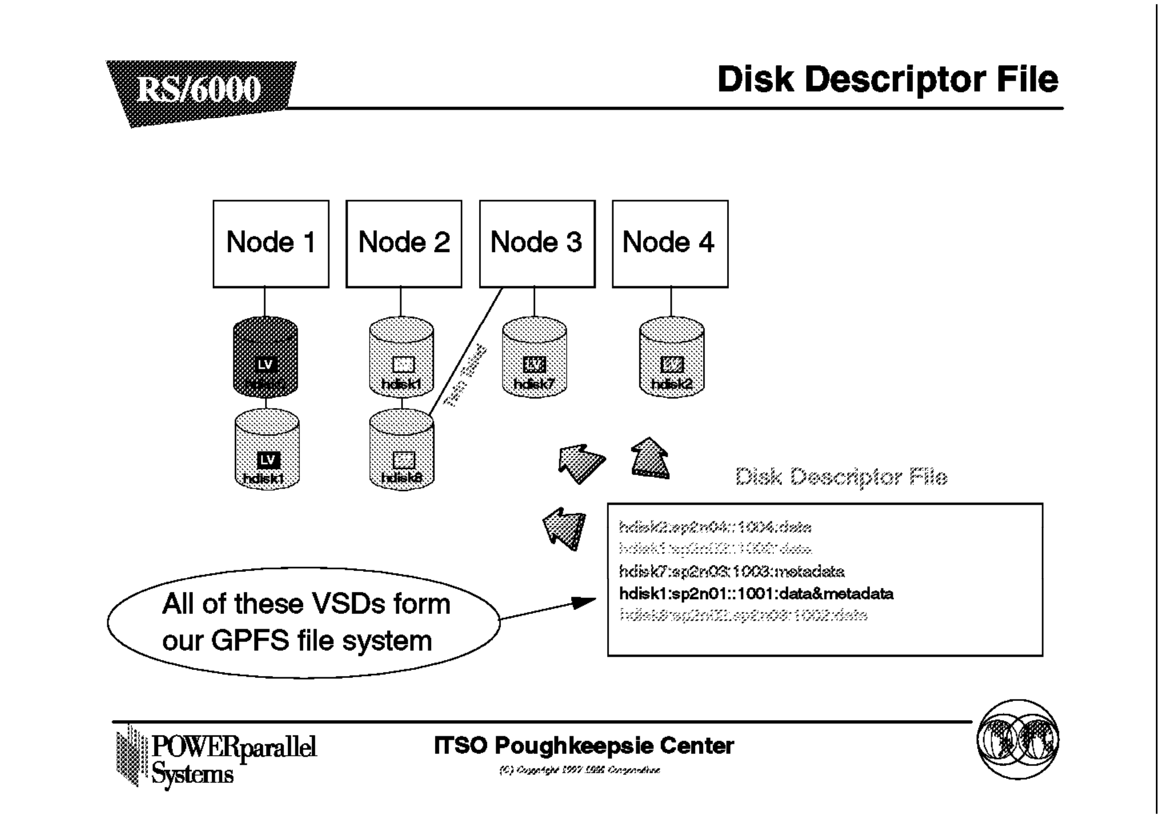
| | |
|---|---|
| hdisk names | Device names - eg hdisk1, hdisk3 |
| Primary Server Name | Hostname of primary server (or IP address) |
| Secondary Server Name | Hostname of backup server (for use with RVSD) |
| Failure Group | A number specifying the Failure Group |
| Metadata/Data | Specify: data/metadata/data & metadata |

POWERparallel Systems

ITSO Poughkeepsie Center

The information that you provide in a disk descriptor file is at the heart of a GPFS file system. It defines in detail exactly how the VSDs will be created. This information will be used by the create VSD commands.

### 6.9.2.1 Disk Descriptors



Disk Descriptor File

Disk Descriptor File

```
hdisk2:sp2n04::1004:data
hdisk1:sp2n02::1002:data
hdisk7:sp2n03:1003:metadata
hdisk1:sp2n01::1001:data&metadata
hdisk8:sp2n02:sp2n03:1002:data
```

All of these VSDs form our GPFS file system

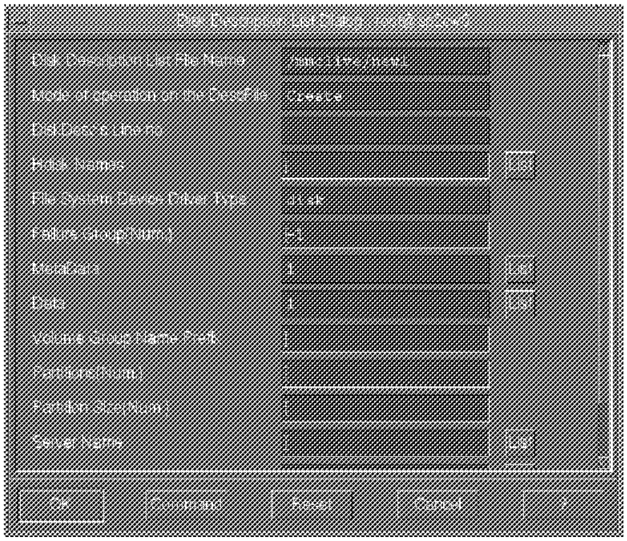
In this example, our disk descriptor file can hold one line for each VSD that goes to make up our GPFS file system.

We have a lot of flexibility. This example is not one that we would normally choose to implement because it has an imbalance of disks.

## 6.9.3 Disk Descriptors - SMIT

RS/6000

**Disk Descriptor SMIT panel**



Disk Description List File Name: /smclive/new1

Mode of operation on the DescFile: Create

Disk Desc's Line no:

Hdisk Names:

File System Device Driver Type: disk

Failure Group(Num.): -1

MetaData: 1

Data: 1

Volume Group Name Prefix:

Partitions(Num.):

Partition Size(Num.):

Server Name:

POWERparallel Systems

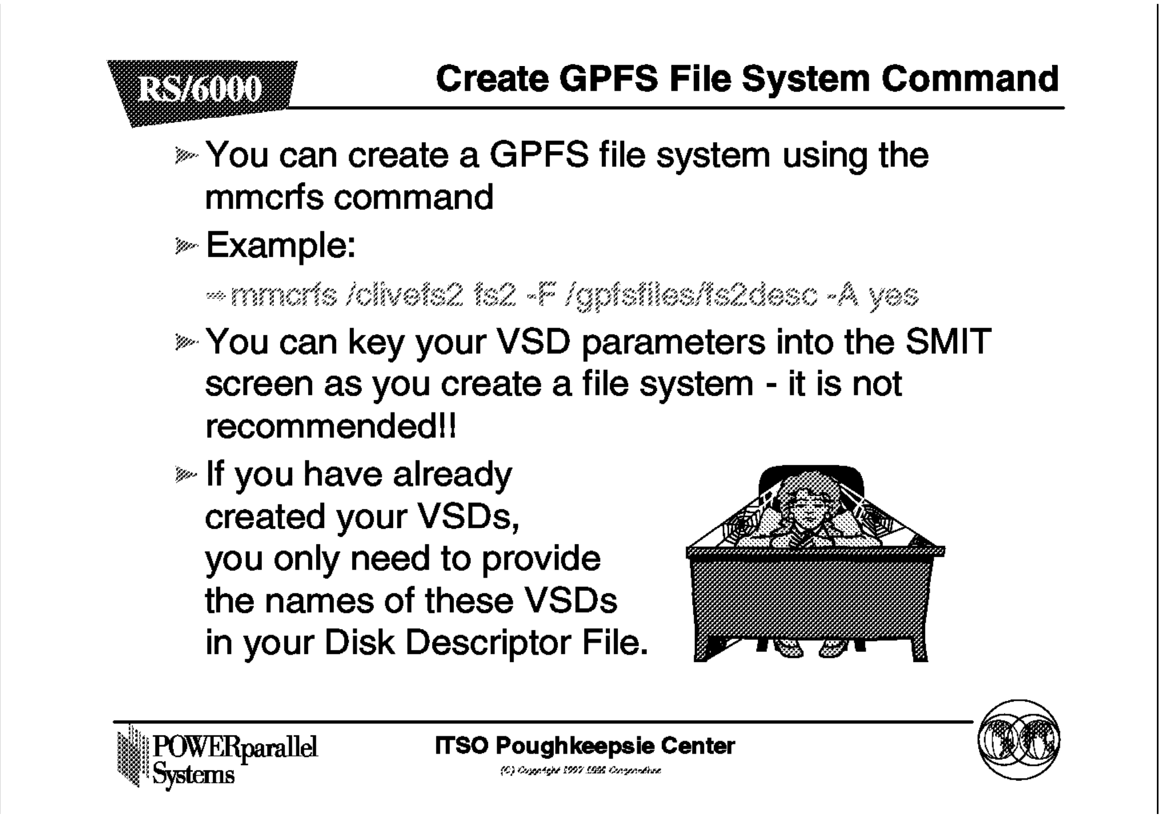
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

You can either create your disk descriptor files through SMIT as shown here, or you can edit the files by hand. The format of the files is important, with each parameter in the correct sequence and separated by a colon.

You can leave a field blank; the default value will be used if one exists.

### 6.9.4 Create Filesystems - Commands


RS/6000

**Create GPFS File System Command**

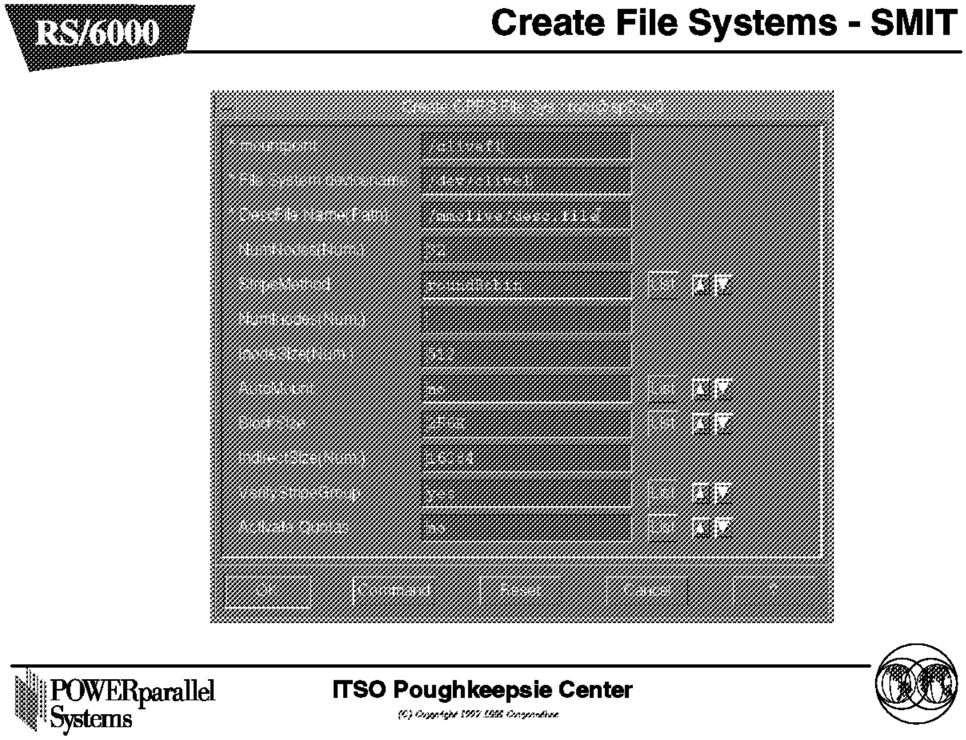
- You can create a GPFS file system using the mmcrfs command
- Example:
  - mmcrfs /clivefs2 fs2 -F /gpfsfiles/fs2desc -A yes
- You can key your VSD parameters into the SMIT screen as you create a file system - it is not recommended!!
- If you have already created your VSDs, you only need to provide the names of these VSDs in your Disk Descriptor File.

POWERparallel Systems

ITSO Poughkeepsie Center

The command for creating a GPFS file system is shown here.

## 6.9.5 Creating File Systems - SMIT



### Create File Systems - SMIT

Create GPFS File System

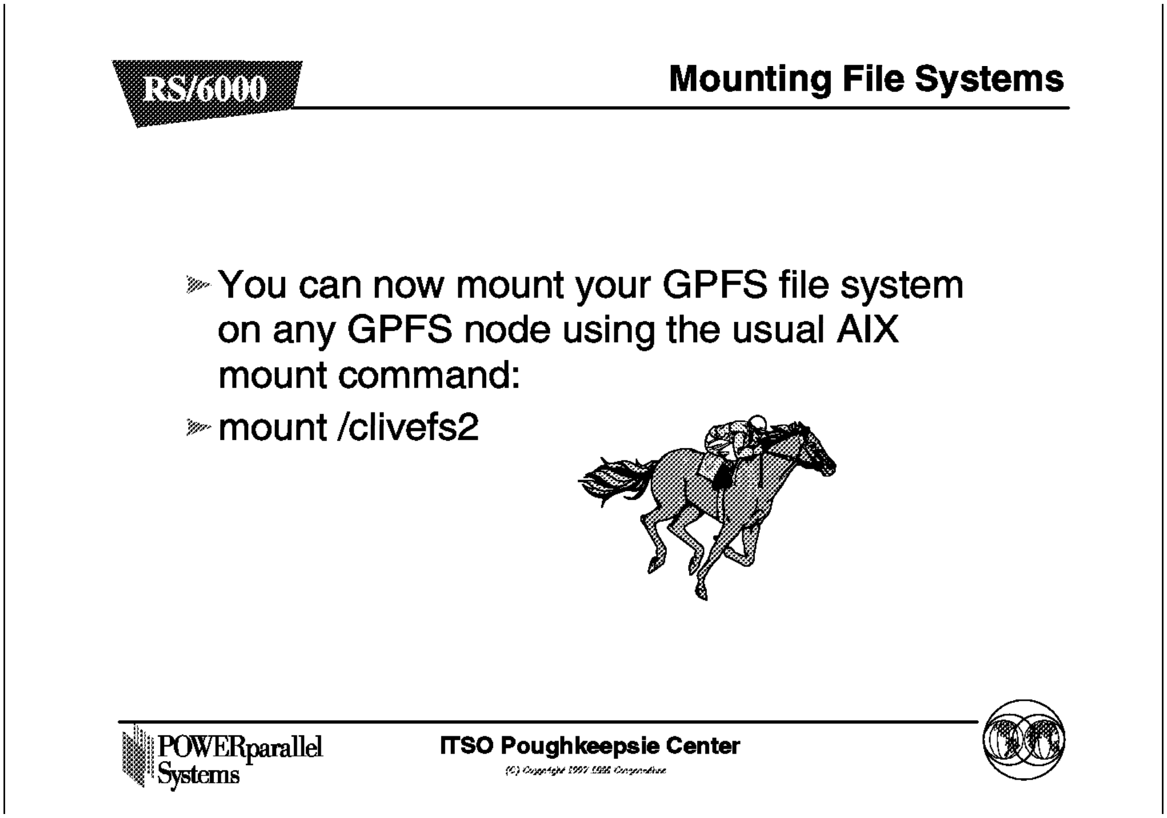
| | |
|---|---|
| * mountpoint | /clivefl |
| * File System devicename | /dev/clive1 |
| * DescFile Name(Path) | /mmclive/desc.file |
| NumNodes(Num) | 32 |
| StripeMethod | roundRobin |
| NumInodes(Num) | |
| InodeSize(Num) | 512 |
| AutoMount | no |
| BlockSize | 256K |
| IndirectSize(Num) | 16384 |
| VerifyStripeGroup | yes |
| Activate Quotas | no |

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Once the disk descriptor file is complete, you can either create your file system using the commands as shown previously, or you can use SMIT as shown here.

### 6.9.6 Mounting File Systems



RS/6000

Mounting File Systems

- You can now mount your GPFS file system on any GPFS node using the usual AIX mount command:
- mount /clivefs2

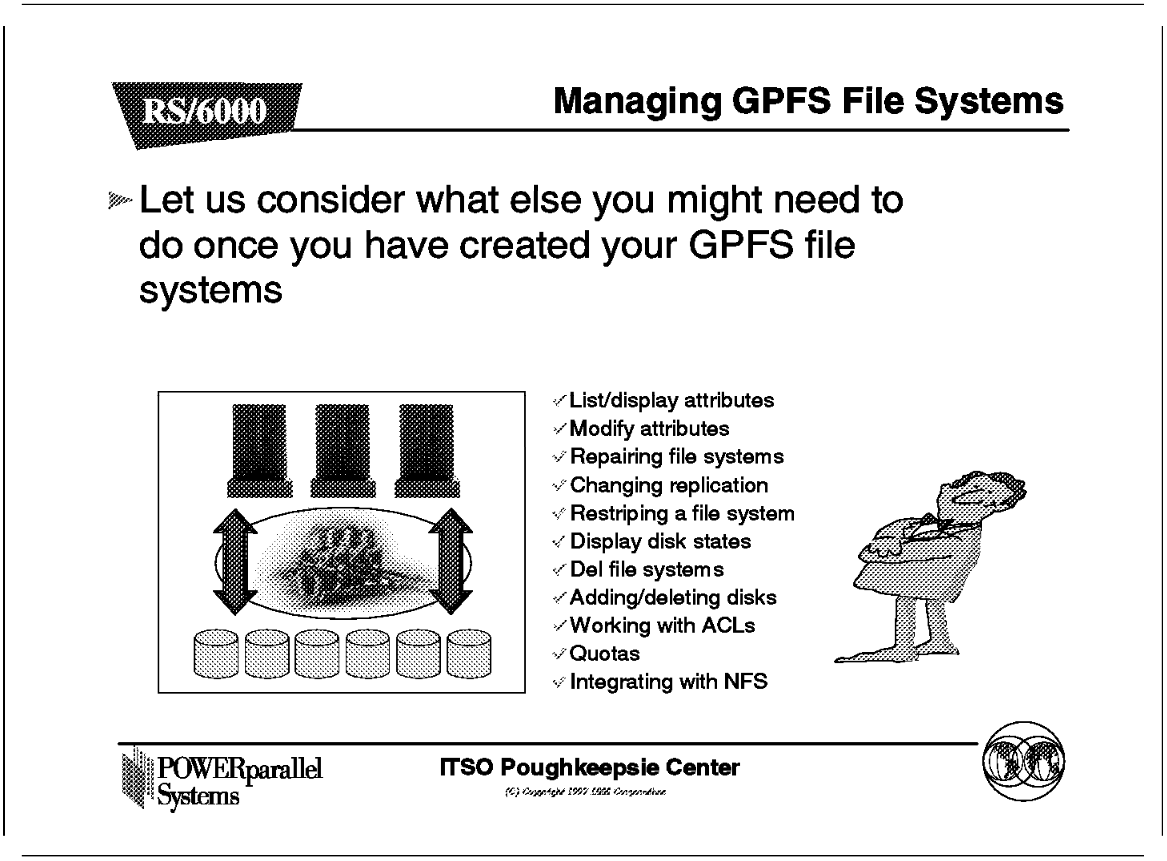
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The GPFS file system, once created, can be mounted on any GPFS node in the normal way with the mount command.

## 6.10 Managing GPFS File Systems



RS/6000

**Managing GPFS File Systems**

- Let us consider what else you might need to do once you have created your GPFS file systems

- ✓ List/display attributes
- ✓ Modify attributes
- ✓ Repairing file systems
- ✓ Changing replication
- ✓ Restriping a file system
- ✓ Display disk states
- ✓ Del file systems
- ✓ Adding/deleting disks
- ✓ Working with ACLs
- ✓ Quotas
- ✓ Integrating with NFS

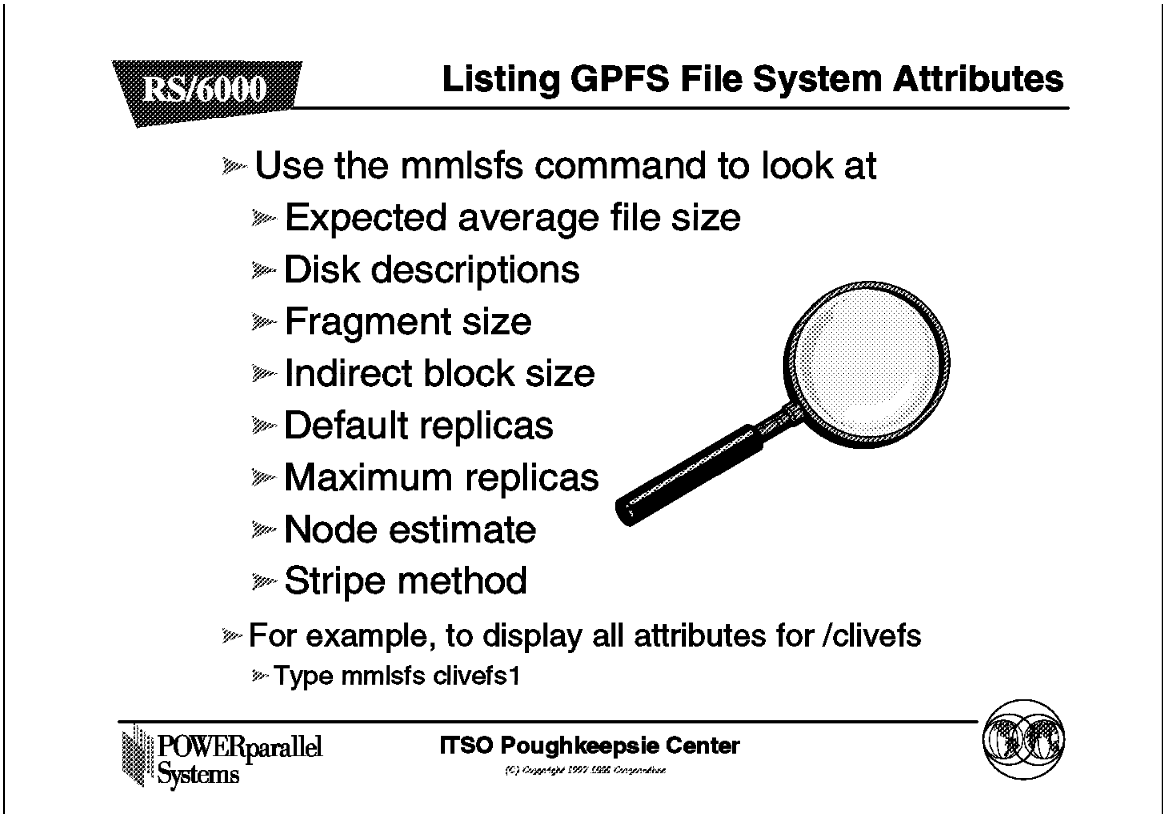
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Once you have created your GPFS file systems, there are a number of tasks that you may wish to perform.

### 6.10.1 Listing GPFS File System Attributes


RS/6000
POWERparallel Systems

## Listing GPFS File System Attributes

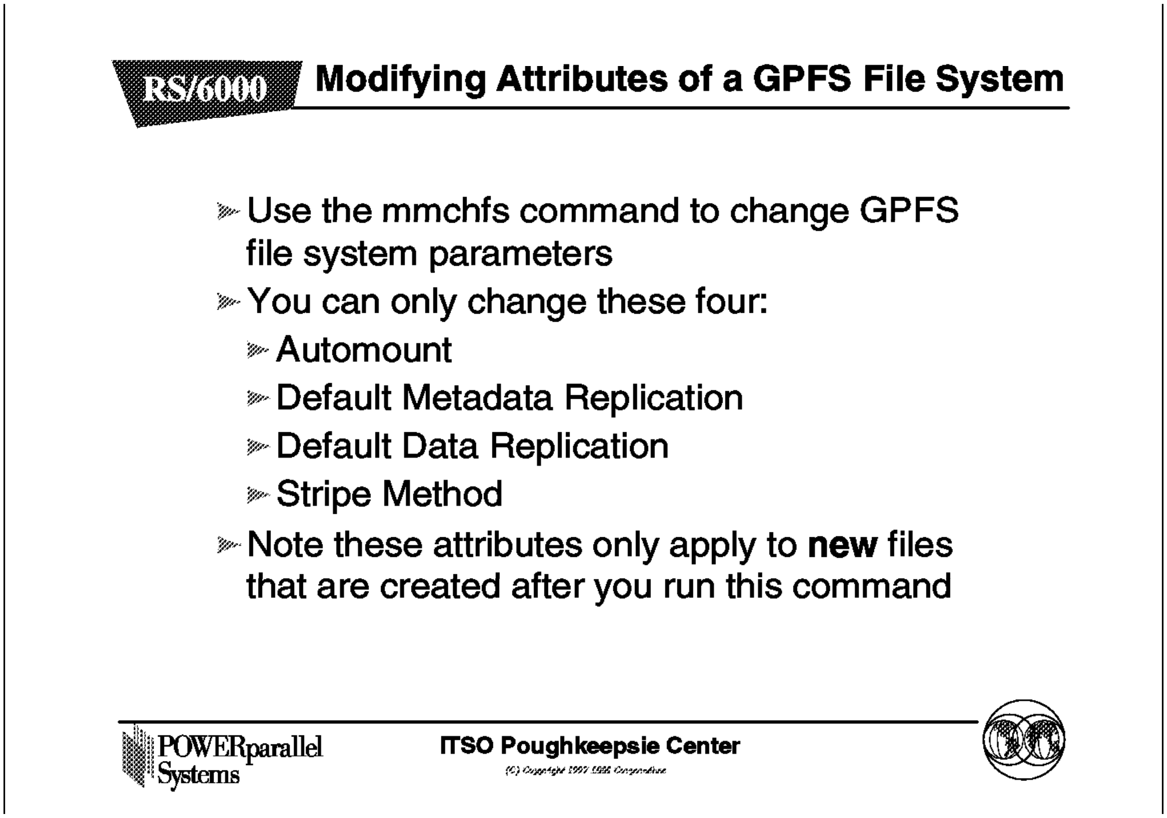
- Use the mmlsfs command to look at
  - Expected average file size
  - Disk descriptions
  - Fragment size
  - Indirect block size
  - Default replicas
  - Maximum replicas
  - Node estimate
  - Stripe method
- For example, to display all attributes for /clivefs
  - Type mmlsfs clivefs1

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

You can list file system attributes.

### 6.10.2 Modifying Attributes of a GPFS File System



RS/6000 **Modifying Attributes of a GPFS File System**

- Use the mmchfs command to change GPFS file system parameters
- You can only change these four:
  - Automount
  - Default Metadata Replication
  - Default Data Replication
  - Stripe Method
- Note these attributes only apply to **new** files that are created after you run this command

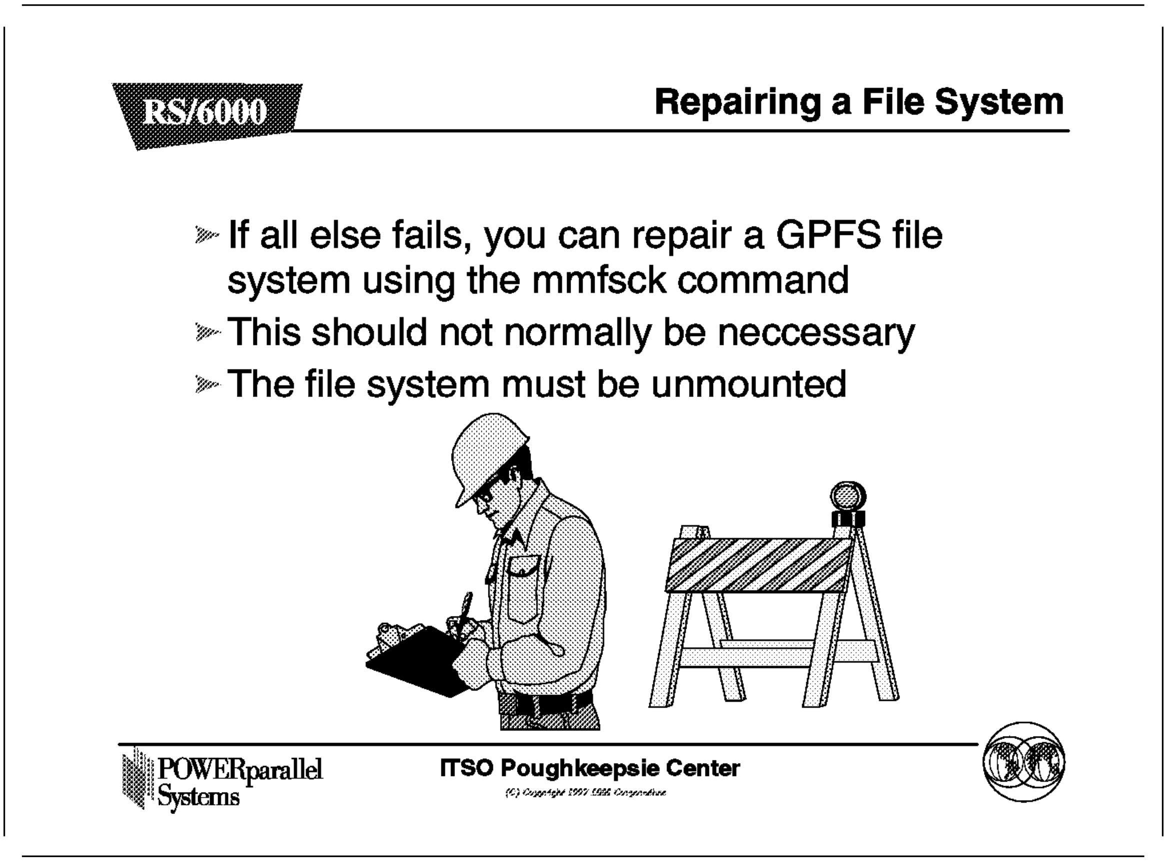
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

You can change a limited number of file system attributes.

### 6.10.3 Repairing a GPFS File System



RS/6000

## Repairing a File System

- If all else fails, you can repair a GPFS file system using the mmfsck command
- This should not normally be neccessary
- The file system must be unmounted

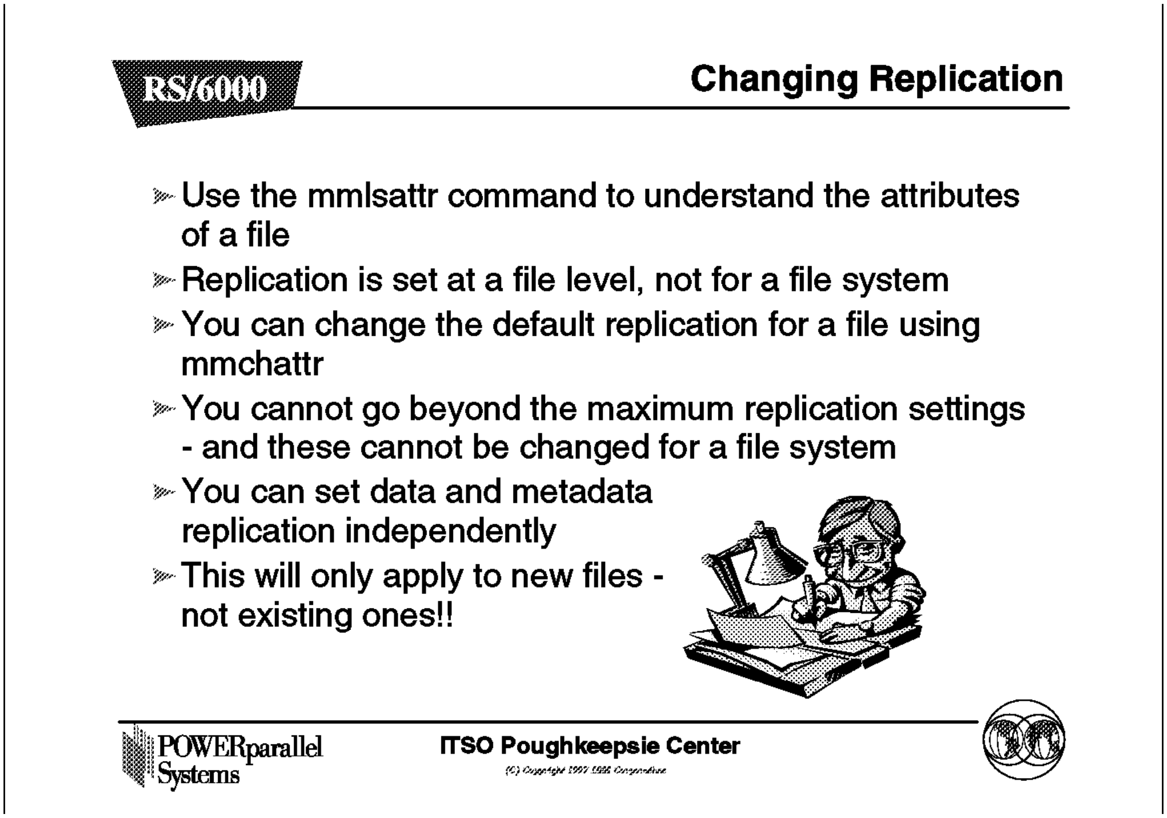
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 1998 Corporation

You should not normally need to recover by running mmfsck, but the facility is there if required. Under normal circumstances, including failures, GPFS should repair any file systems itself when required.

## 6.10.4 Changing Replication



RS/6000

Changing Replication

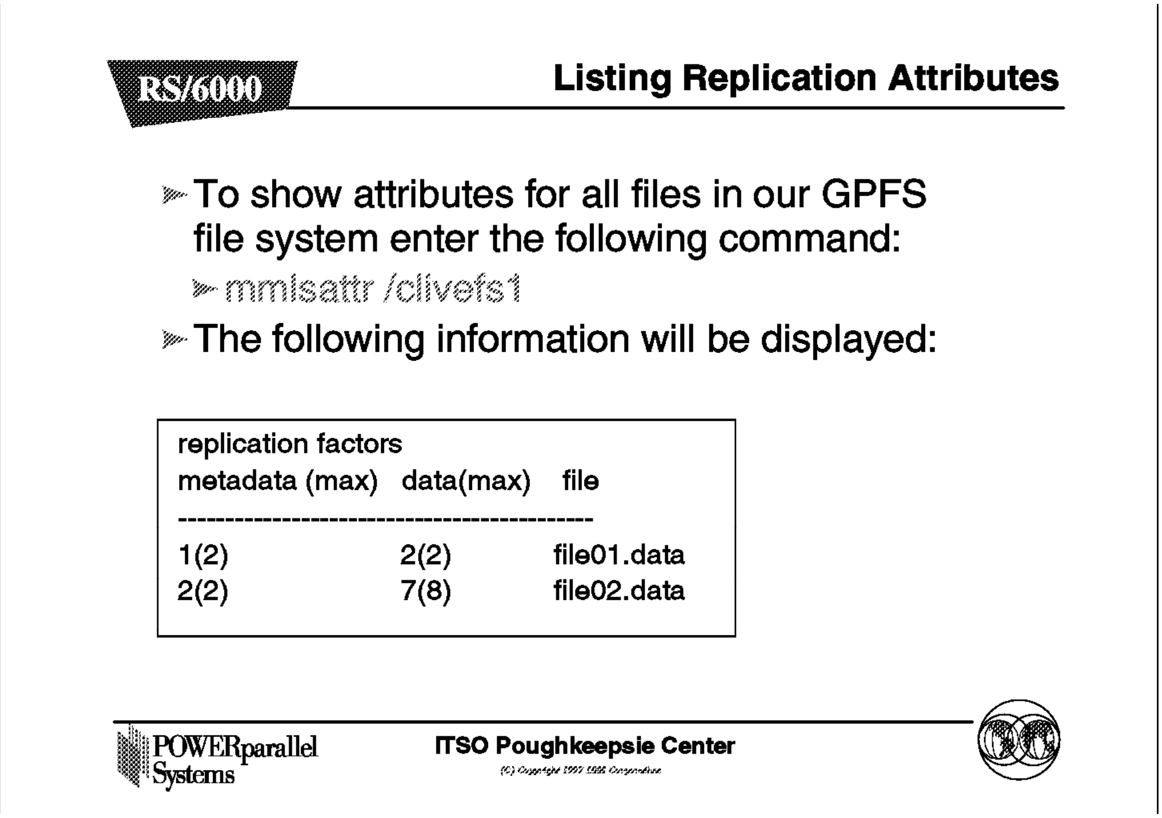
- Use the mmlsattr command to understand the attributes of a file
- Replication is set at a file level, not for a file system
- You can change the default replication for a file using mmchattr
- You cannot go beyond the maximum replication settings - and these cannot be changed for a file system
- You can set data and metadata replication independently
- This will only apply to new files - not existing ones!!

POWERparallel Systems

ITSO Poughkeepsie Center

When you create a new file, it will always be created with the default replication settings. You can subsequently change the settings, but if you can, you may find it useful to touch a file and change the replication settings before using it.

### 6.10.5 Listing Replication Attributes



RS/6000

**Listing Replication Attributes**

- To show attributes for all files in our GPFS file system enter the following command:
  - mmlsattr /clivefs1
- The following information will be displayed:

```
replication factors
metadata (max)   data(max)    file
-------------------------------------------
1(2)             2(2)         file01.data
2(2)             7(8)         file02.data
```

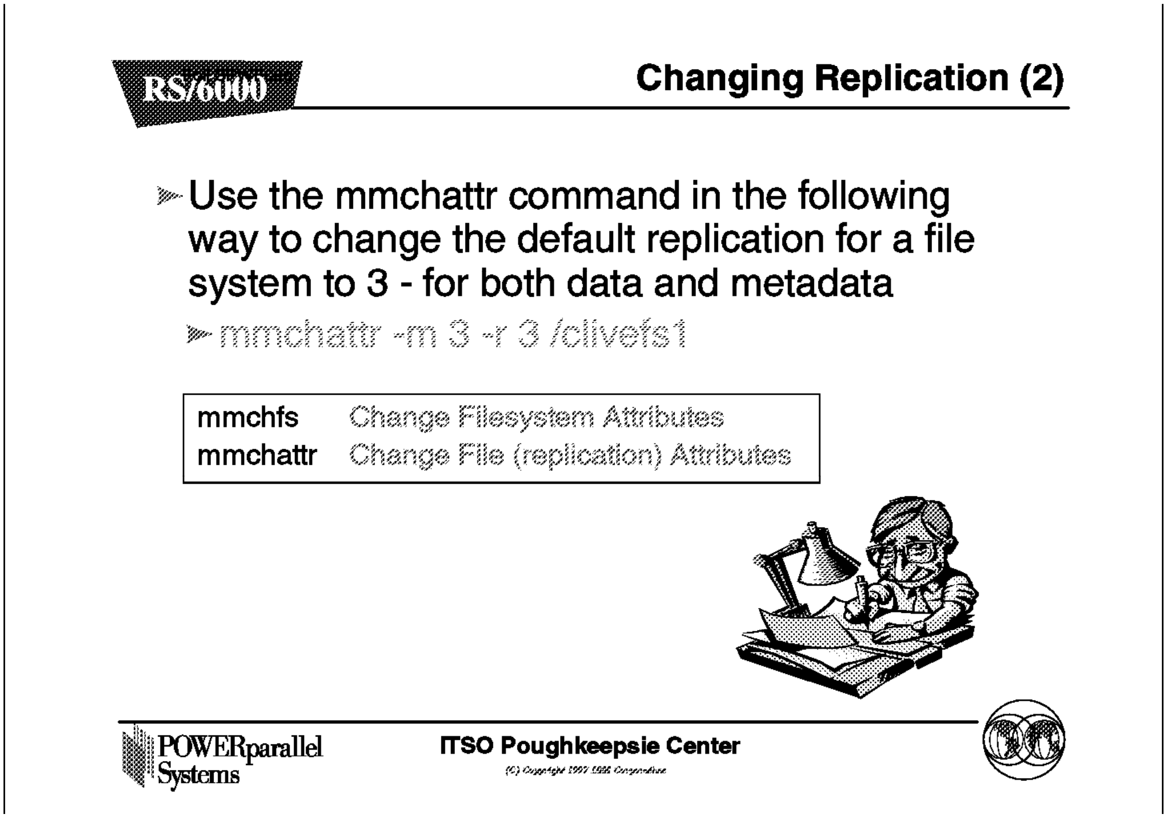
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This command allows you to list the attributes for a particular file.

## 6.10.6 Changing Replication



RS/6000

### Changing Replication (2)

- Use the mmchattr command in the following way to change the default replication for a file system to 3 - for both data and metadata
  - mmchattr -m 3 -r 3 /clivefs1

| | |
|---|---|
| mmchfs | Change Filesystem Attributes |
| mmchattr | Change File (replication) Attributes |

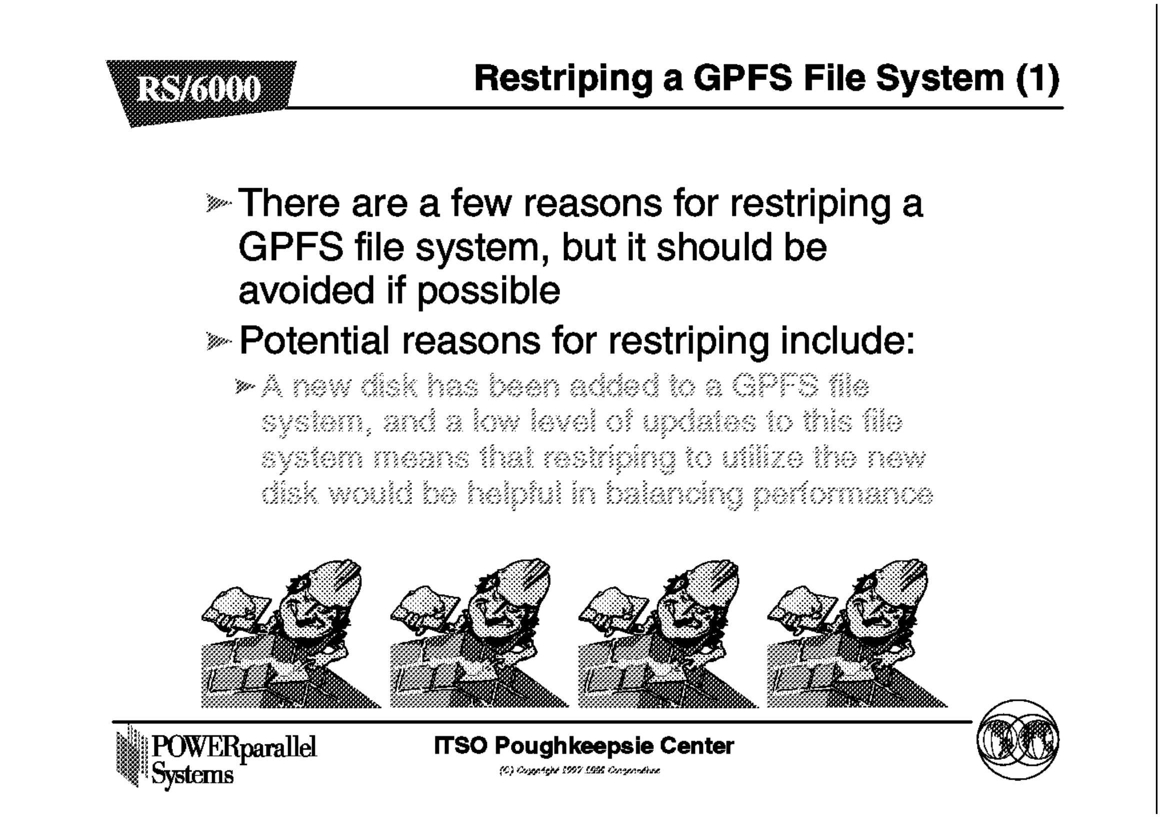
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Here is an example of changing the replication settings for a file.

## 6.10.7 Restriping a GPFS Filesystem



RS/6000

**Restriping a GPFS File System (1)**

- There are a few reasons for restriping a GPFS file system, but it should be avoided if possible
- Potential reasons for restriping include:
  - A new disk has been added to a GPFS file system, and a low level of updates to this file system means that restriping to utilize the new disk would be helpful in balancing performance

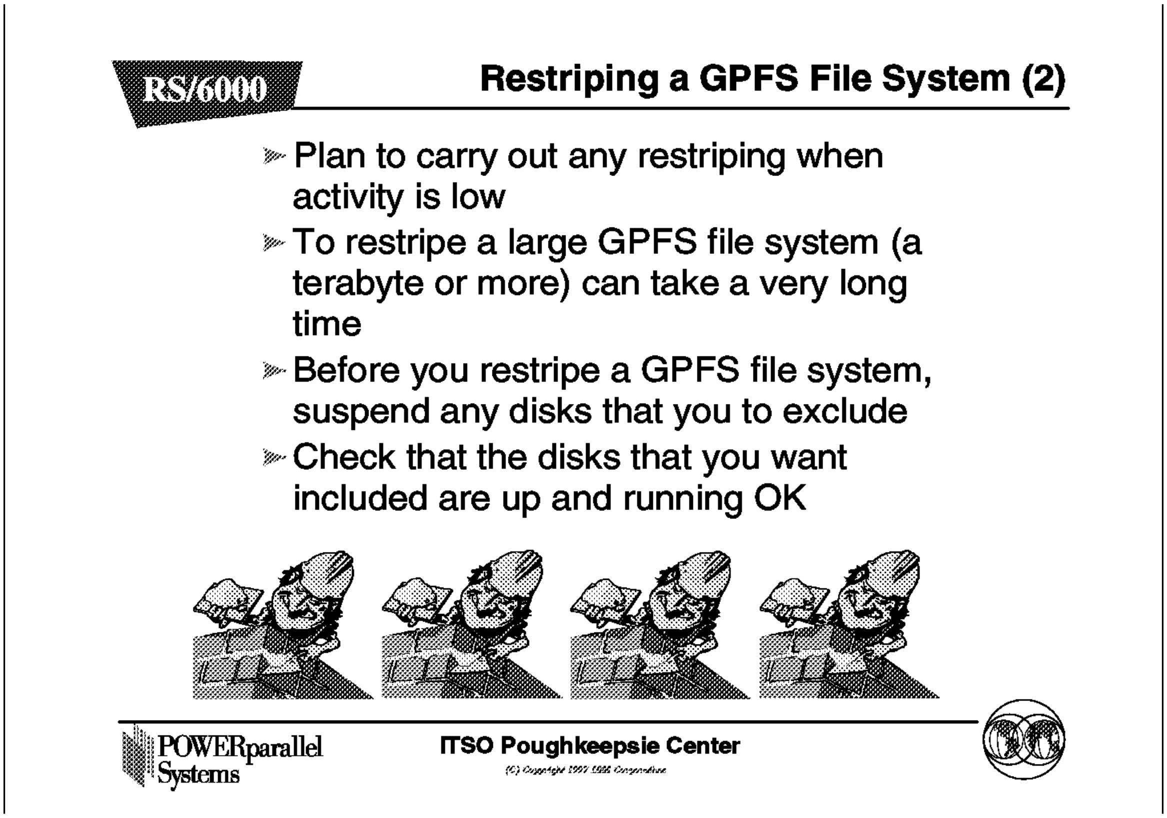
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Restriping is an intensive process and should be avoided if possible.

### 6.10.7.1 Restriping a GPFS Filesystem



RS/6000

**Restriping a GPFS File System (2)**

- Plan to carry out any restriping when activity is low
- To restripe a large GPFS file system (a terabyte or more) can take a very long time
- Before you restripe a GPFS file system, suspend any disks that you to exclude
- Check that the disks that you want included are up and running OK

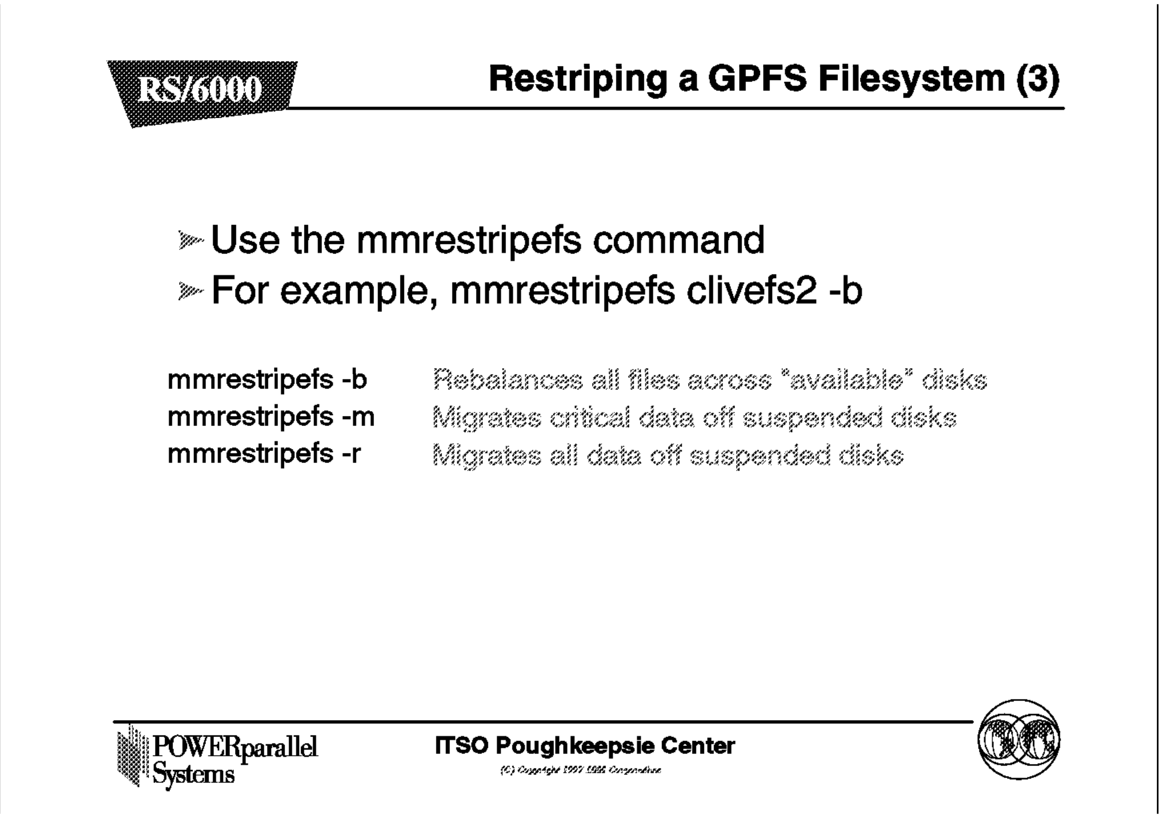
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

If you must restripe a file system, do it at a time when system activity is low.

### 6.10.7.2 Restriping a GPFS Filesystem



RS/6000

## Restriping a GPFS Filesystem (3)

- Use the mmrestripefs command
- For example, mmrestripefs clivefs2 -b

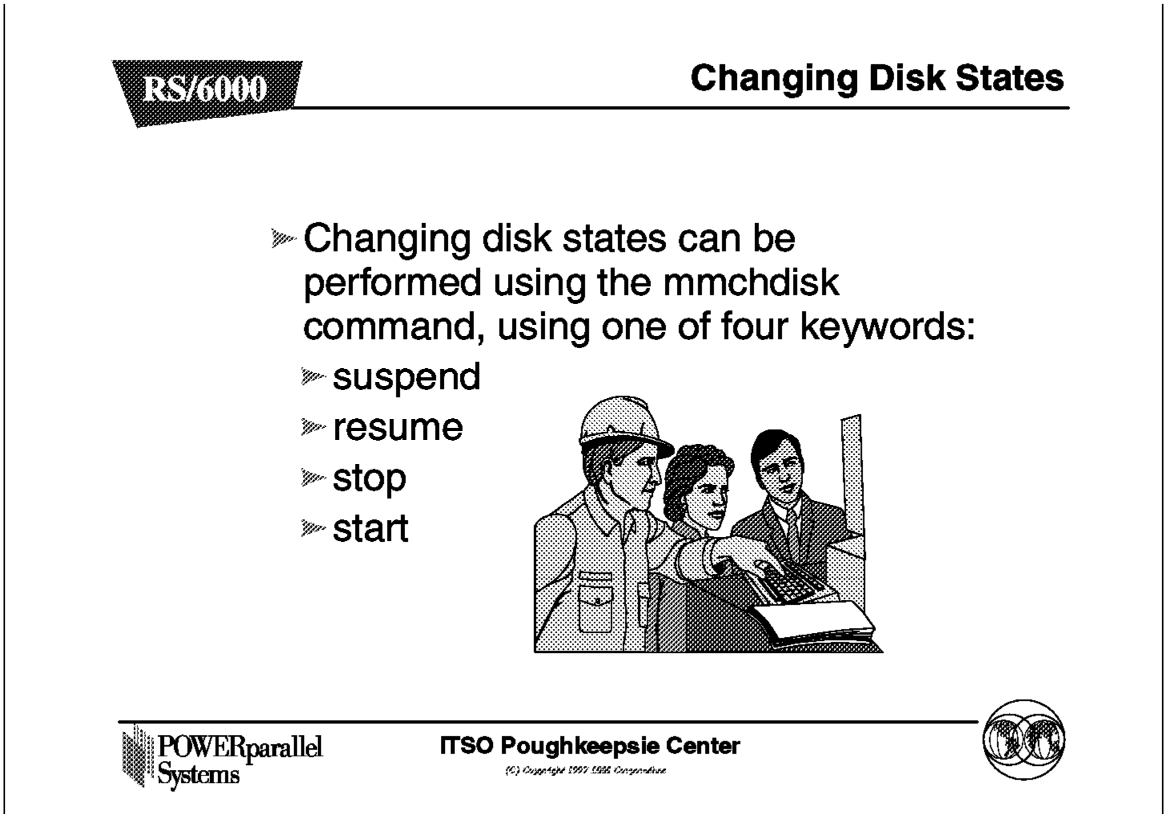
| | |
|---|---|
| mmrestripefs -b | Rebalances all files across "available" disks |
| mmrestripefs -m | Migrates critical data off suspended disks |
| mmrestripefs -r | Migrates all data off suspended disks |

POWERparallel Systems

ITSO Poughkeepsie Center

Here are some examples of commands to restripe a file system.

### 6.10.8 Changing Disk States



RS/6000

## Changing Disk States

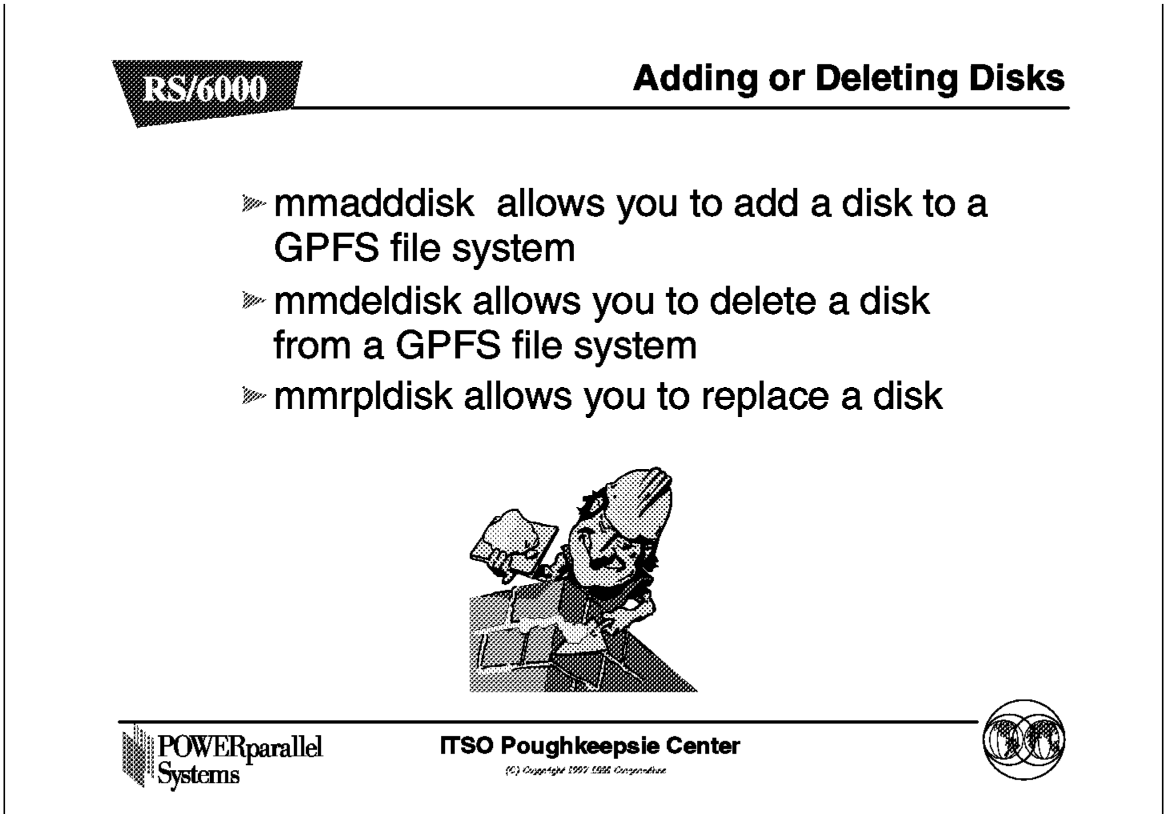
- Changing disk states can be performed using the mmchdisk command, using one of four keywords:
  - suspend
  - resume
  - stop
  - start

POWERparallel Systems

ITSO Poughkeepsie Center

You can change the state of disks when required.

### 6.10.9 Adding or Deleting Disks



RS/6000

**Adding or Deleting Disks**

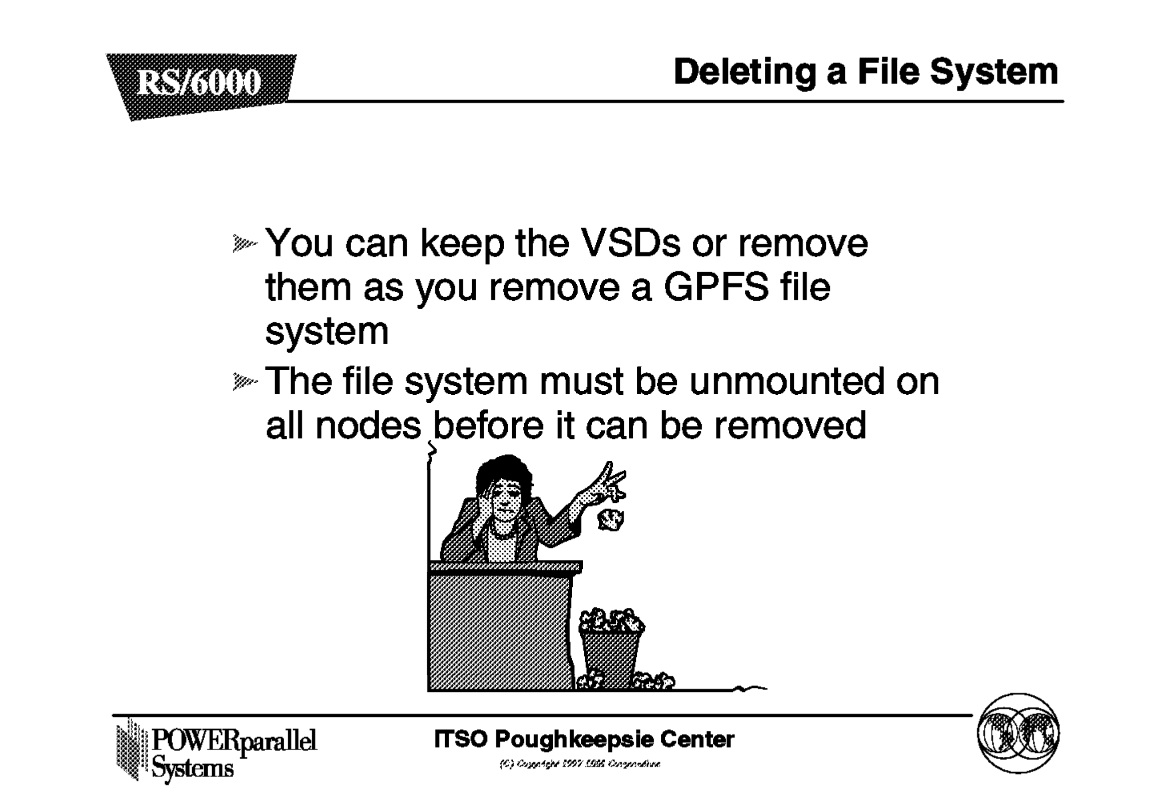
- mmadddisk  allows you to add a disk to a GPFS file system
- mmdeldisk allows you to delete a disk from a GPFS file system
- mmrpldisk allows you to replace a disk

POWERparallel Systems

**ITSO Poughkeepsie Center**

You can add or delete disks to the GPFS system. You will be defining VSDs in much the same way as you did when you created your file system.

## 6.10.10 Deleting File Systems



RS/6000

**Deleting a File System**

- You can keep the VSDs or remove them as you remove a GPFS file system
- The file system must be unmounted on all nodes before it can be removed

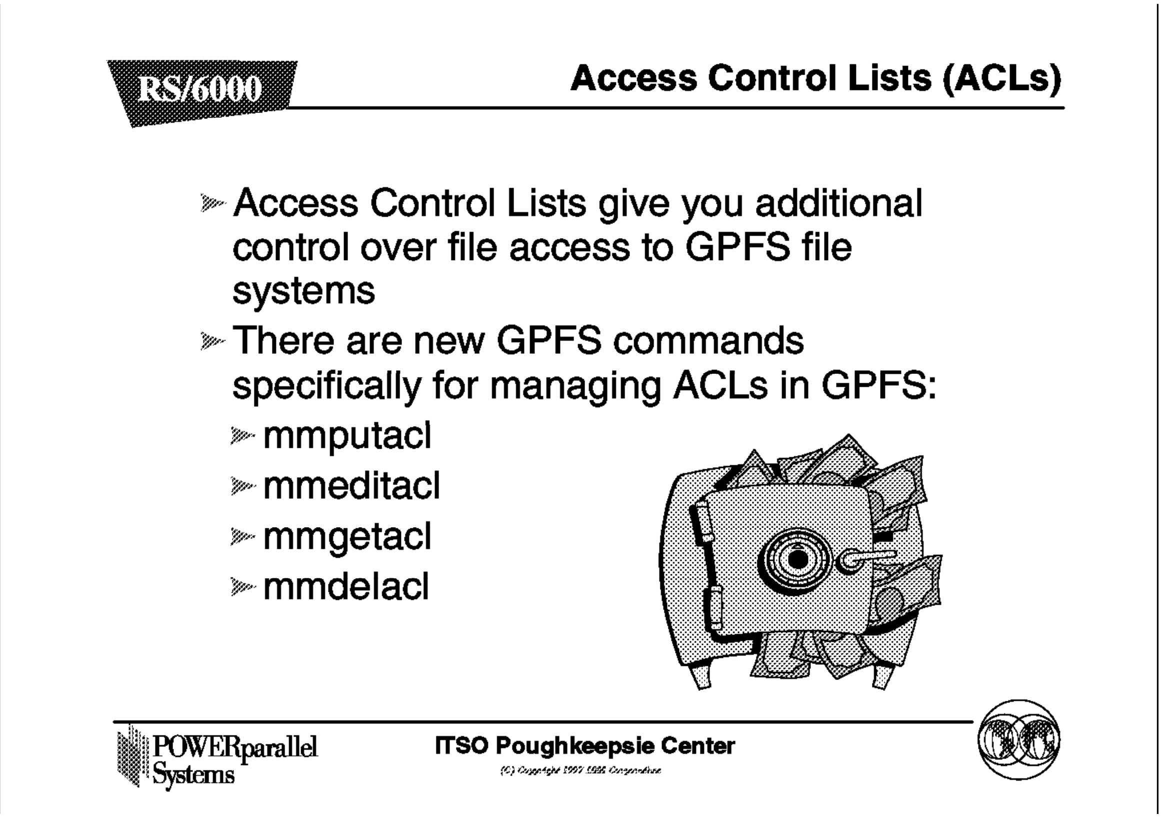
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

File systems can be removed when no longer required.

### 6.10.11 Access Control Lists



RS/6000

**Access Control Lists (ACLs)**

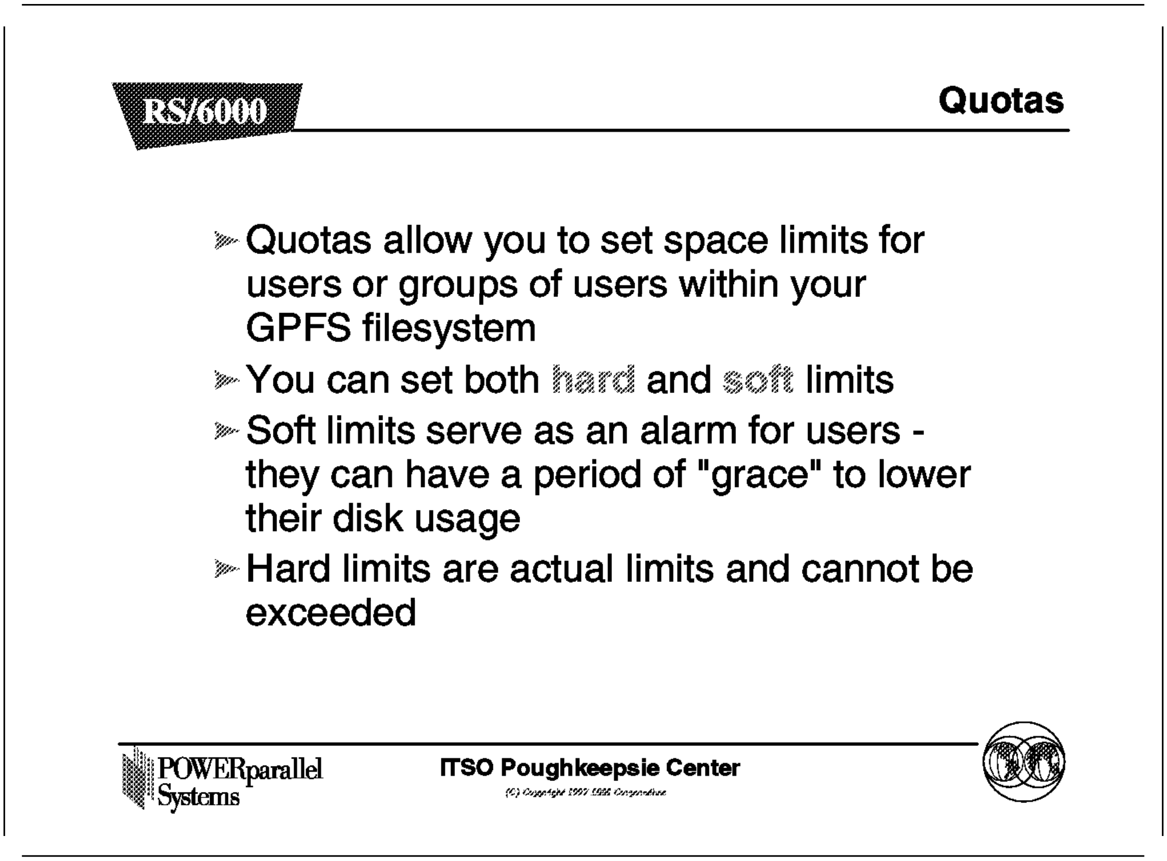
- Access Control Lists give you additional control over file access to GPFS file systems
- There are new GPFS commands specifically for managing ACLs in GPFS:
  - mmputacl
  - mmeditacl
  - mmgetacl
  - mmdelacl

POWERparallel Systems

**ITSO Poughkeepsie Center**

As with standard AIX, ACLs give you additional control over standard file permissions to allow you to give more secure access to files and file systems to users or groups of users.

## 6.10.12 Quotas



RS/6000

**Quotas**

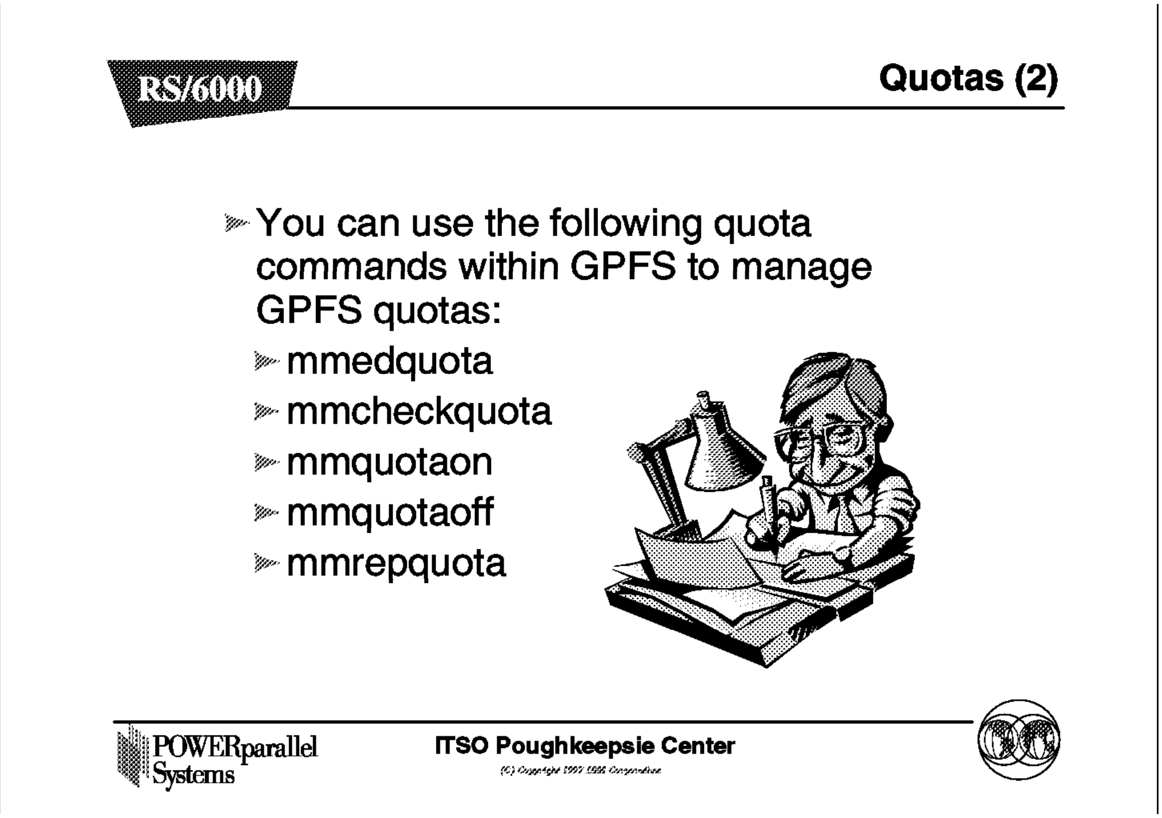
- Quotas allow you to set space limits for users or groups of users within your GPFS filesystem
- You can set both hard and soft limits
- Soft limits serve as an alarm for users - they can have a period of "grace" to lower their disk usage
- Hard limits are actual limits and cannot be exceeded

POWERparallel Systems

ITSO Poughkeepsie Center

You can set limits or quotas for the space that users or groups of users can use within the GPFS file system.

### 6.10.13 Quotas



RS/6000

**Quotas (2)**

- You can use the following quota commands within GPFS to manage GPFS quotas:
  - mmedquota
  - mmcheckquota
  - mmquotaon
  - mmquotaoff
  - mmrepquota

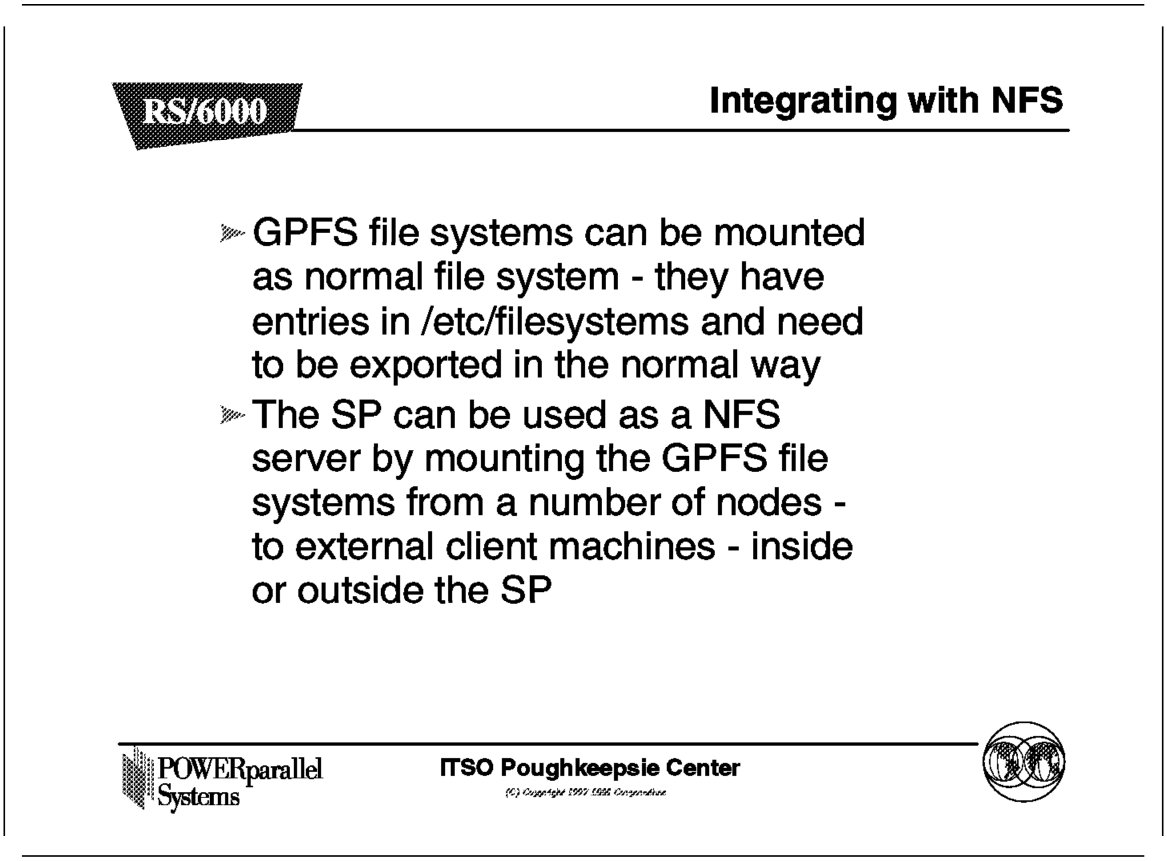
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

There are mm commands to allow you to manage quotas.

### 6.10.14 Integrating with NFS



RS/6000

Integrating with NFS

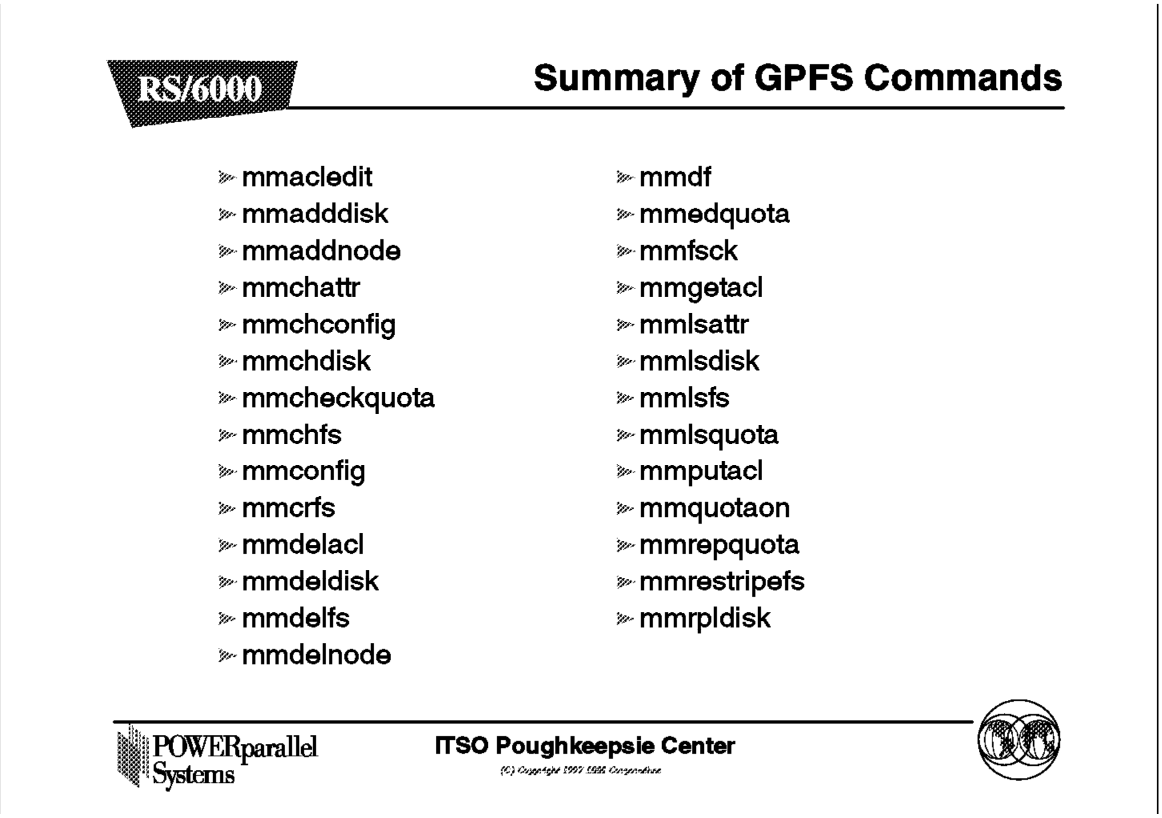
- GPFS file systems can be mounted as normal file system - they have entries in /etc/filesystems and need to be exported in the normal way
- The SP can be used as a NFS server by mounting the GPFS file systems from a number of nodes - to external client machines - inside or outside the SP

POWERparallel Systems

ITSO Poughkeepsie Center

You can integrate GPFS with NFS.

## 6.10.15 GPFS Command Summary



RS/6000

**Summary of GPFS Commands**

- mmacledit
- mmadddisk
- mmaddnode
- mmchattr
- mmchconfig
- mmchdisk
- mmcheckquota
- mmchfs
- mmconfig
- mmcrfs
- mmdelacl
- mmdeldisk
- mmdelfs
- mmdelnode
- mmdf
- mmedquota
- mmfsck
- mmgetacl
- mmlsattr
- mmlsdisk
- mmlsfs
- mmlsquota
- mmputacl
- mmquotaon
- mmrepquota
- mmrestripefs
- mmrpldisk

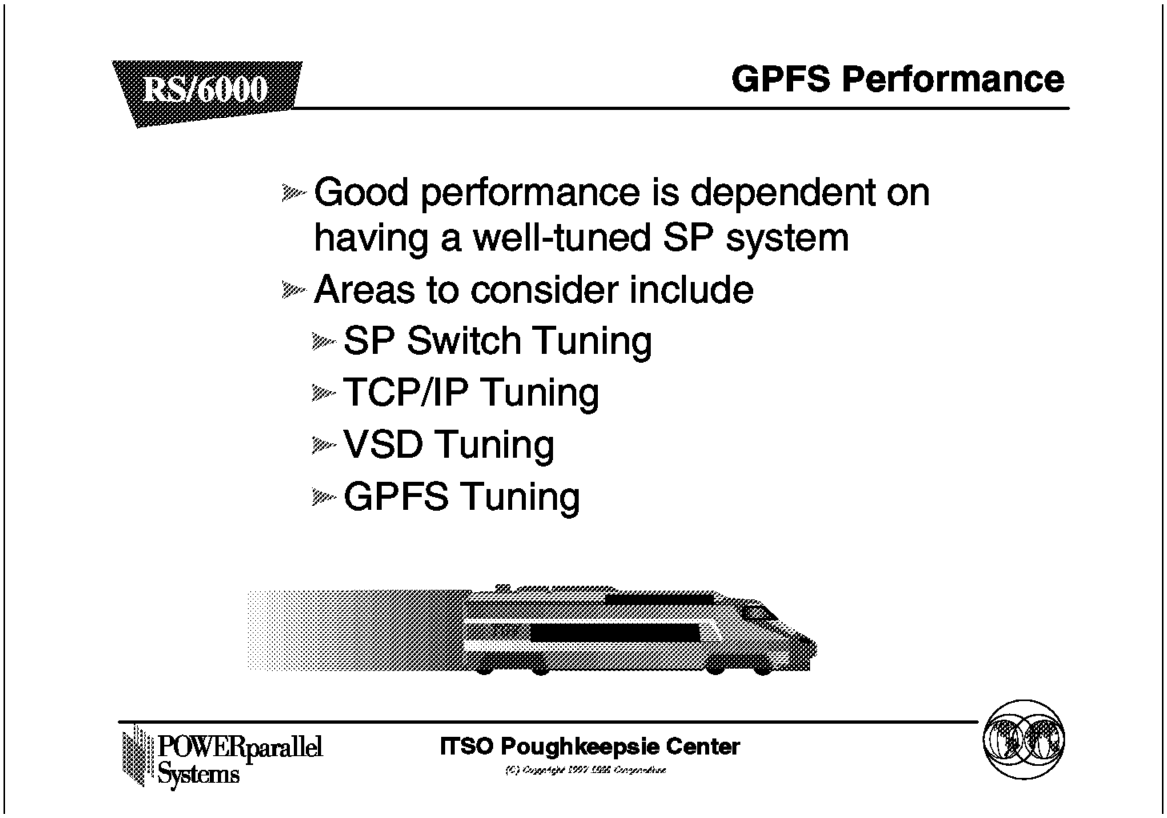
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 1998 Corporation

Here is a summary of all of the GPFS (mm) commands.

## 6.11 GPFS Performance



RS/6000

GPFS Performance

- Good performance is dependent on having a well-tuned SP system
- Areas to consider include
  - SP Switch Tuning
  - TCP/IP Tuning
  - VSD Tuning
  - GPFS Tuning

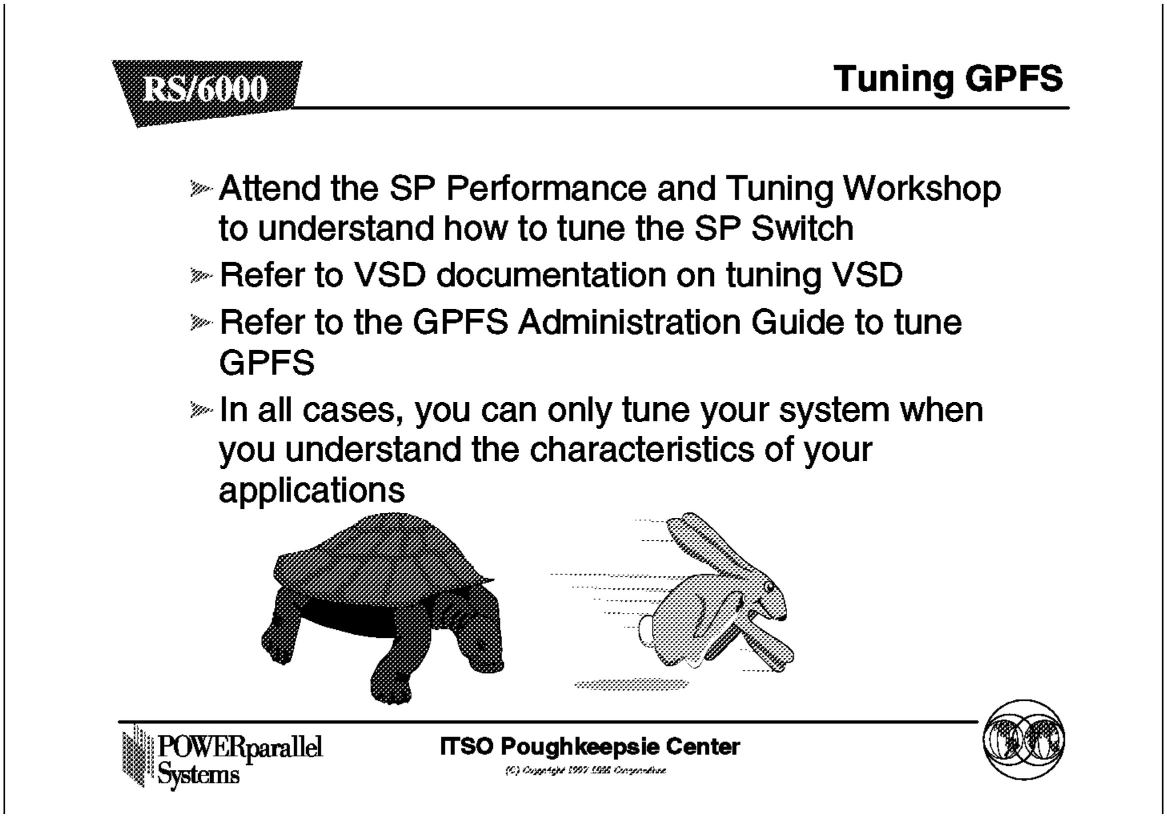
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

You must tune your SP system and Switch to gain optimum performance.

### 6.11.1 Tuning GPFS



RS/6000

**Tuning GPFS**

- ➢ Attend the SP Performance and Tuning Workshop to understand how to tune the SP Switch
- ➢ Refer to VSD documentation on tuning VSD
- ➢ Refer to the GPFS Administration Guide to tune GPFS
- ➢ In all cases, you can only tune your system when you understand the characteristics of your applications

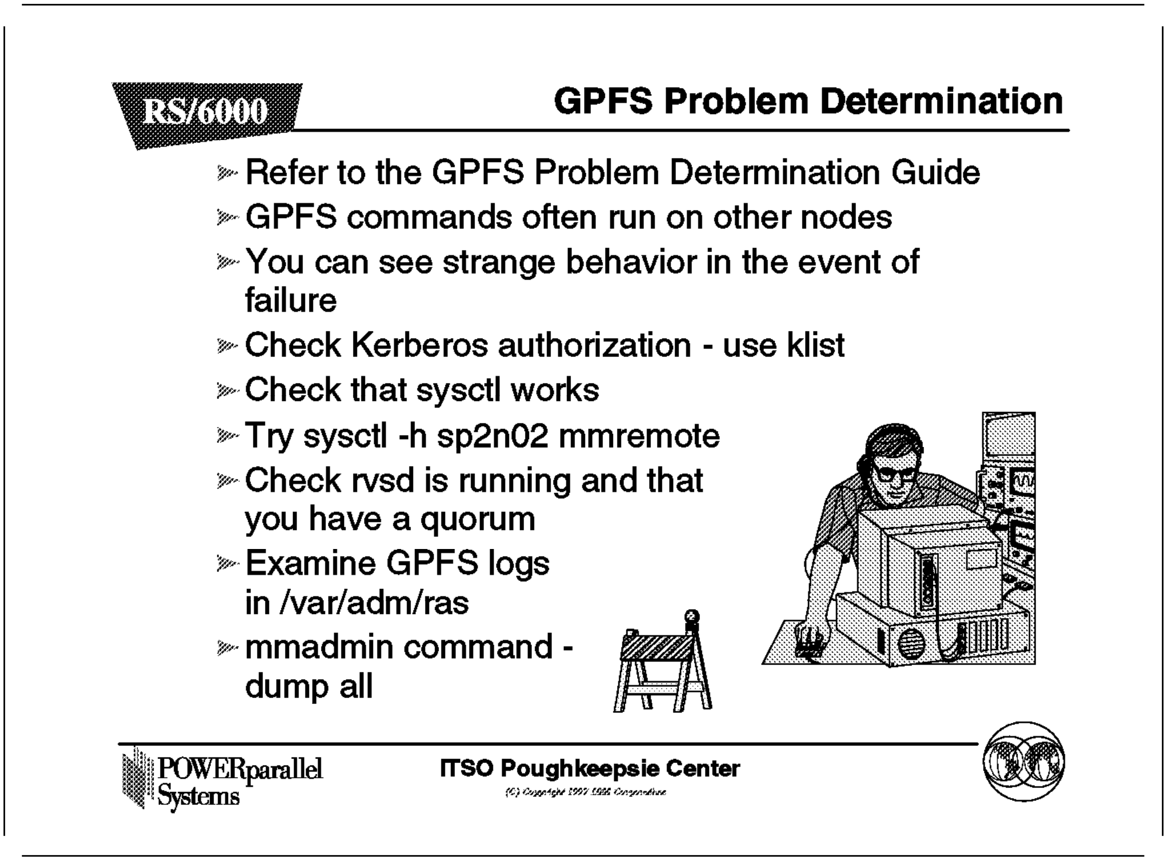
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 1998 Corporation

Without further information, you can use the values given in this workshop as a starting point.

## 6.12 GPFS Problem Determination



RS/6000

GPFS Problem Determination

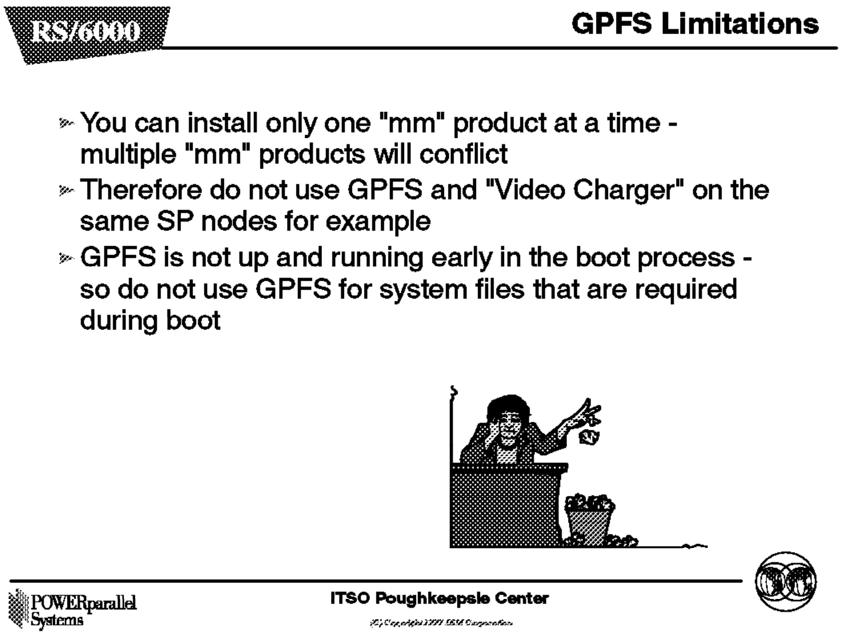
- Refer to the GPFS Problem Determination Guide
- GPFS commands often run on other nodes
- You can see strange behavior in the event of failure
- Check Kerberos authorization - use klist
- Check that sysctl works
- Try sysctl -h sp2n02 mmremote
- Check rvsd is running and that you have a quorum
- Examine GPFS logs in /var/adm/ras
- mmadmin command - dump all

POWERparallel Systems

ITSO Poughkeepsie Center

Check your security setup with Kerberos and sysctl first. This can lead to strange errors.

## 6.13 GPFS Limitations



RS/6000

GPFS Limitations

- You can install only one "mm" product at a time - multiple "mm" products will conflict
- Therefore do not use GPFS and "Video Charger" on the same SP nodes for example
- GPFS is not up and running early in the boot process - so do not use GPFS for system files that are required during boot

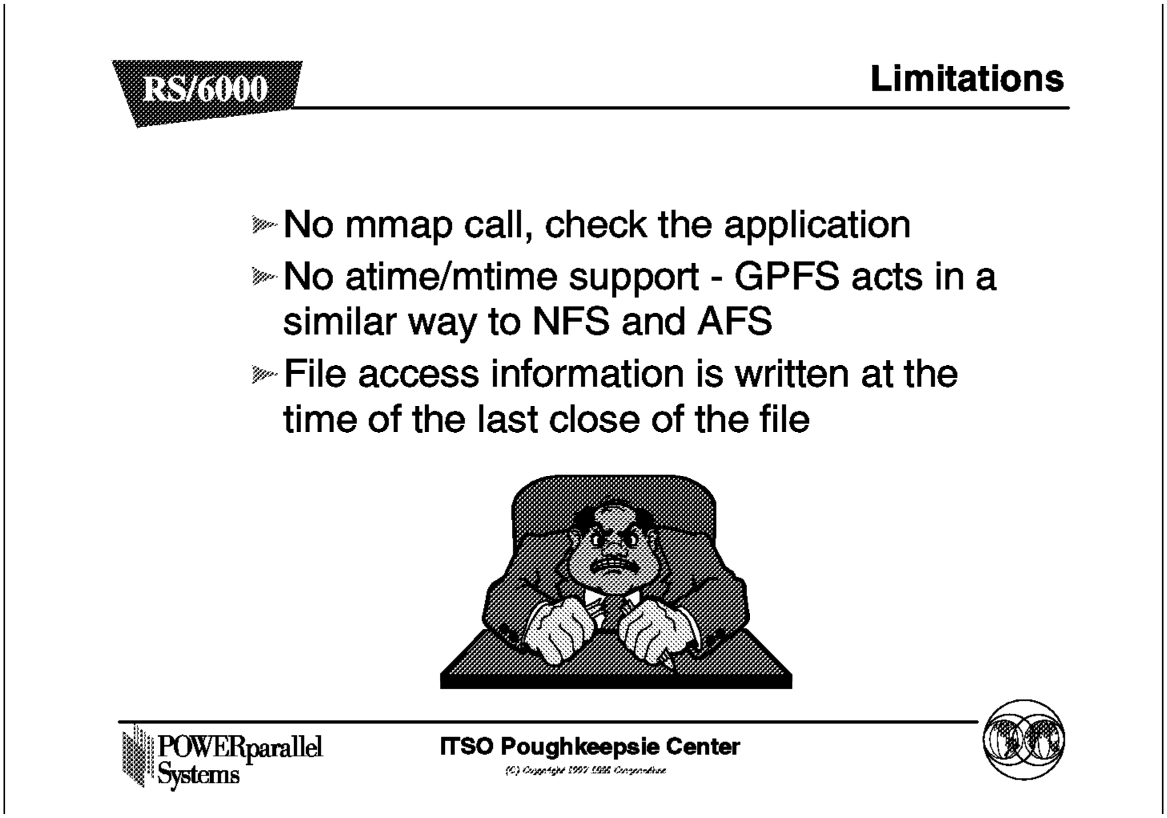
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

There are some limitations to GPFS that you should understand.

### 6.13.1 GPFS Limitations



RS/6000

**Limitations**

- No mmap call, check the application
- No atime/mtime support - GPFS acts in a similar way to NFS and AFS
- File access information is written at the time of the last close of the file

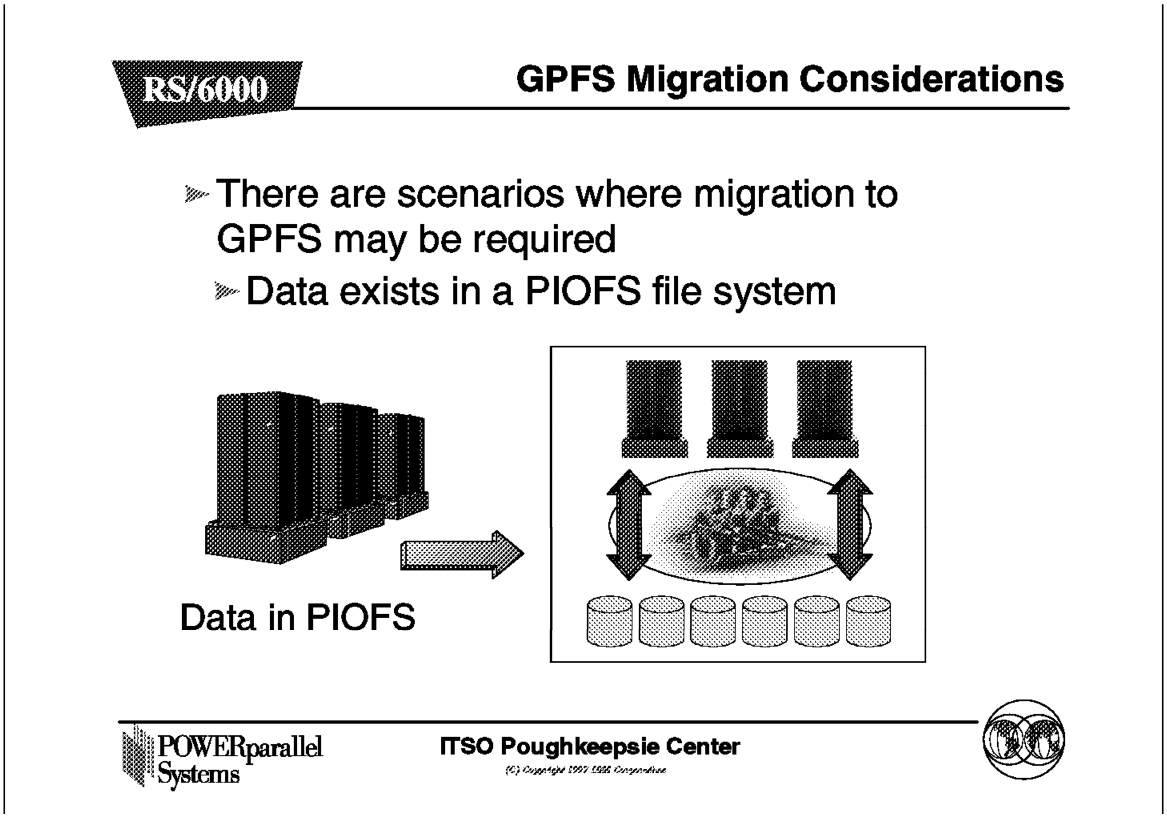
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Check that your application does not use mmap before deciding that GPFS is the correct solution.

## 6.14 GPFS Migration Considerations



RS/6000

GPFS Migration Considerations

- There are scenarios where migration to GPFS may be required
  - Data exists in a PIOFS file system

Data in PIOFS

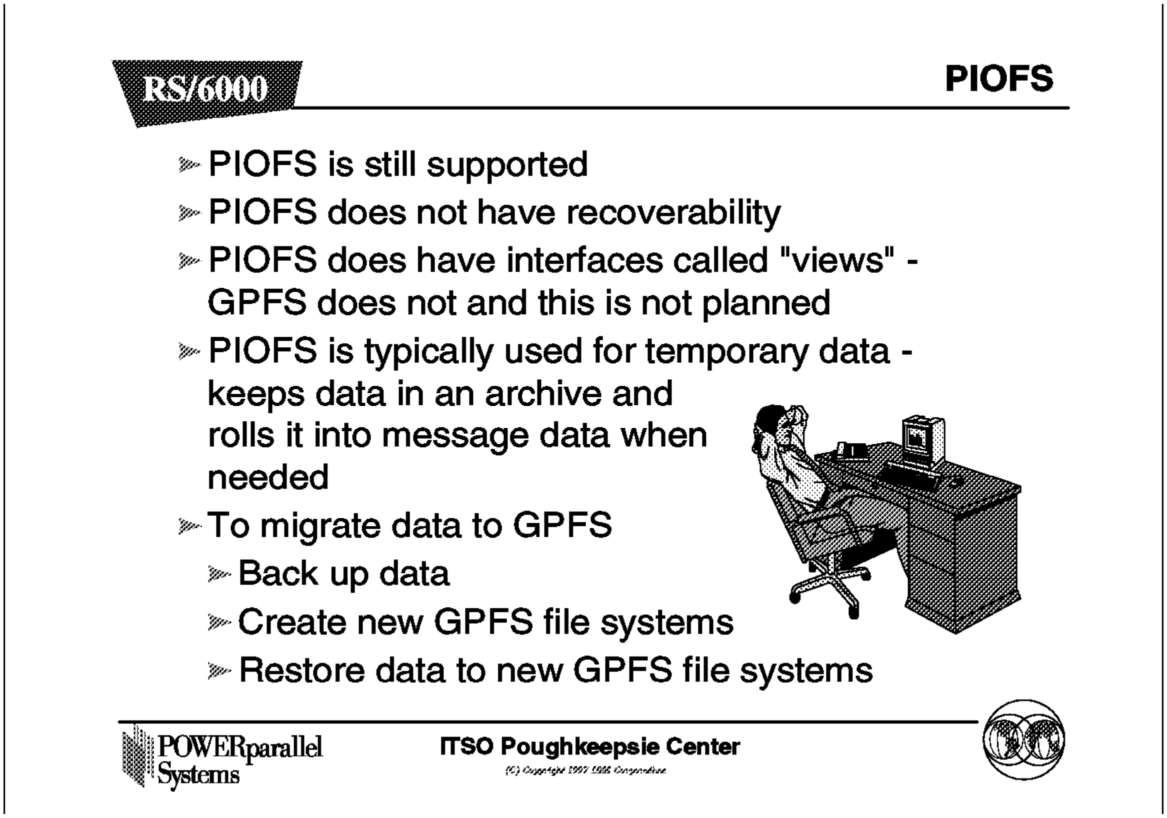
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Some customers may wish to migrate from PIOFS to GPFS.

## 6.14.1 PIOFS



RS/6000

**PIOFS**

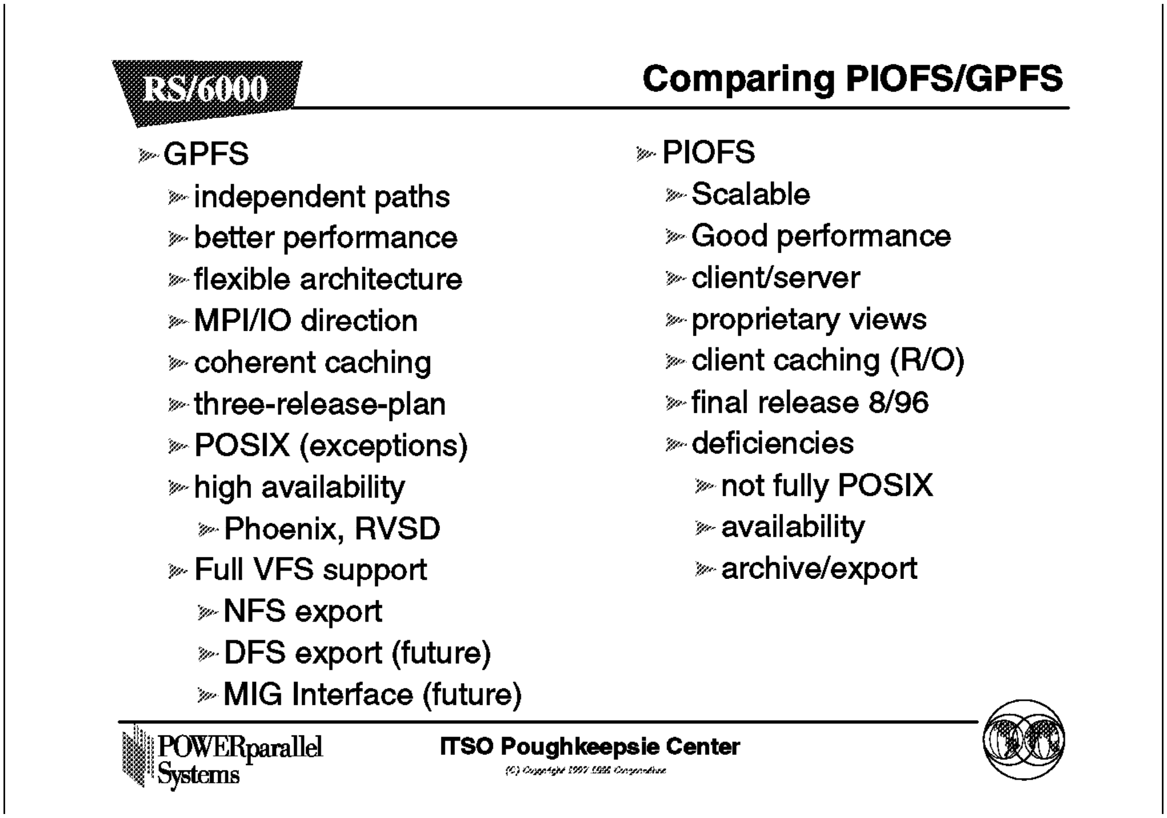
- PIOFS is still supported
- PIOFS does not have recoverability
- PIOFS does have interfaces called "views" - GPFS does not and this is not planned
- PIOFS is typically used for temporary data - keeps data in an archive and rolls it into message data when needed
- To migrate data to GPFS
  - Back up data
  - Create new GPFS file systems
  - Restore data to new GPFS file systems

POWERparallel Systems

ITSO Poughkeepsie Center

PIOFS is still available, but GPFS has some additional functionality.

### 6.14.2 PIOFS Comparison with GPFS



RS/6000

**Comparing PIOFS/GPFS**

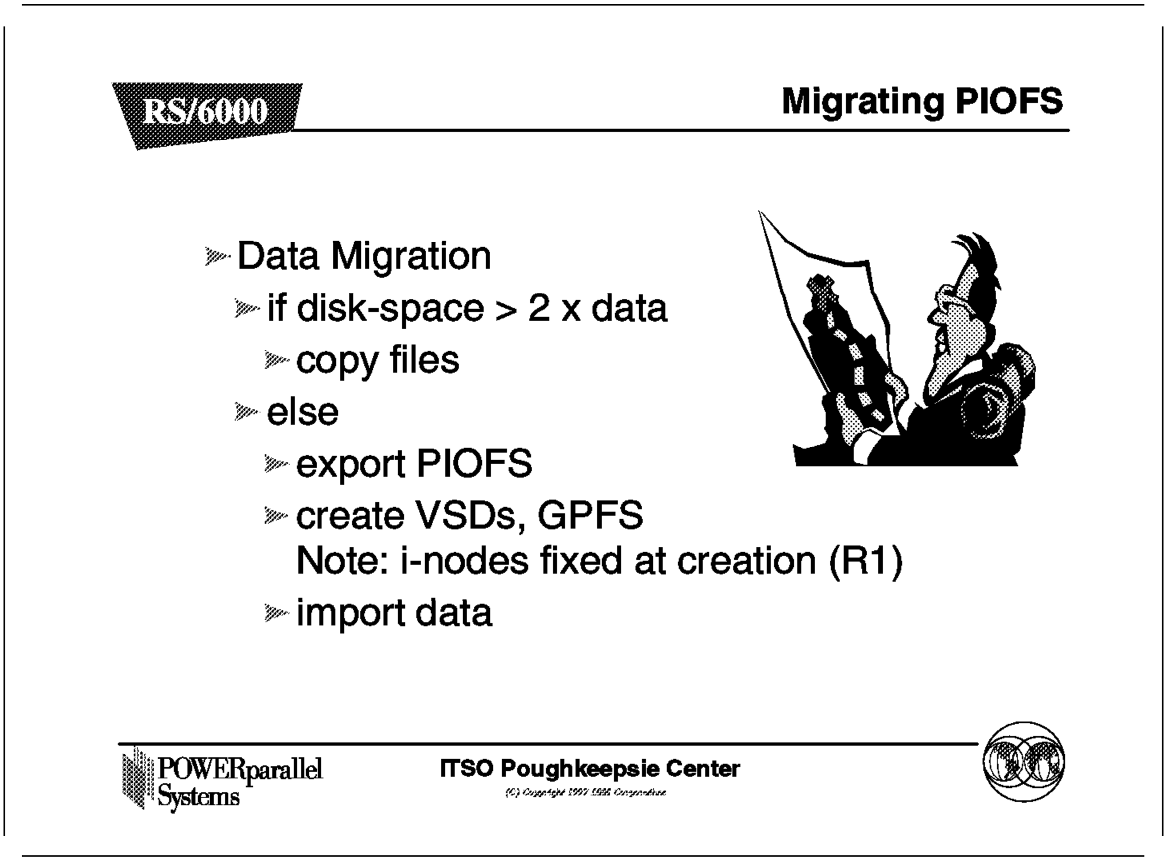
- GPFS
  - independent paths
  - better performance
  - flexible architecture
  - MPI/IO direction
  - coherent caching
  - three-release-plan
  - POSIX (exceptions)
  - high availability
    - Phoenix, RVSD
  - Full VFS support
    - NFS export
    - DFS export (future)
    - MIG Interface (future)
- PIOFS
  - Scalable
  - Good performance
  - client/server
  - proprietary views
  - client caching (R/O)
  - final release 8/96
  - deficiencies
    - not fully POSIX
    - availability
    - archive/export

POWERparallel Systems

ITSO Poughkeepsie Center

Here is a summary comparing GPFS with PIOFS.

### 6.14.3 PIOFS Migration to GPFS



RS/6000

Migrating PIOFS

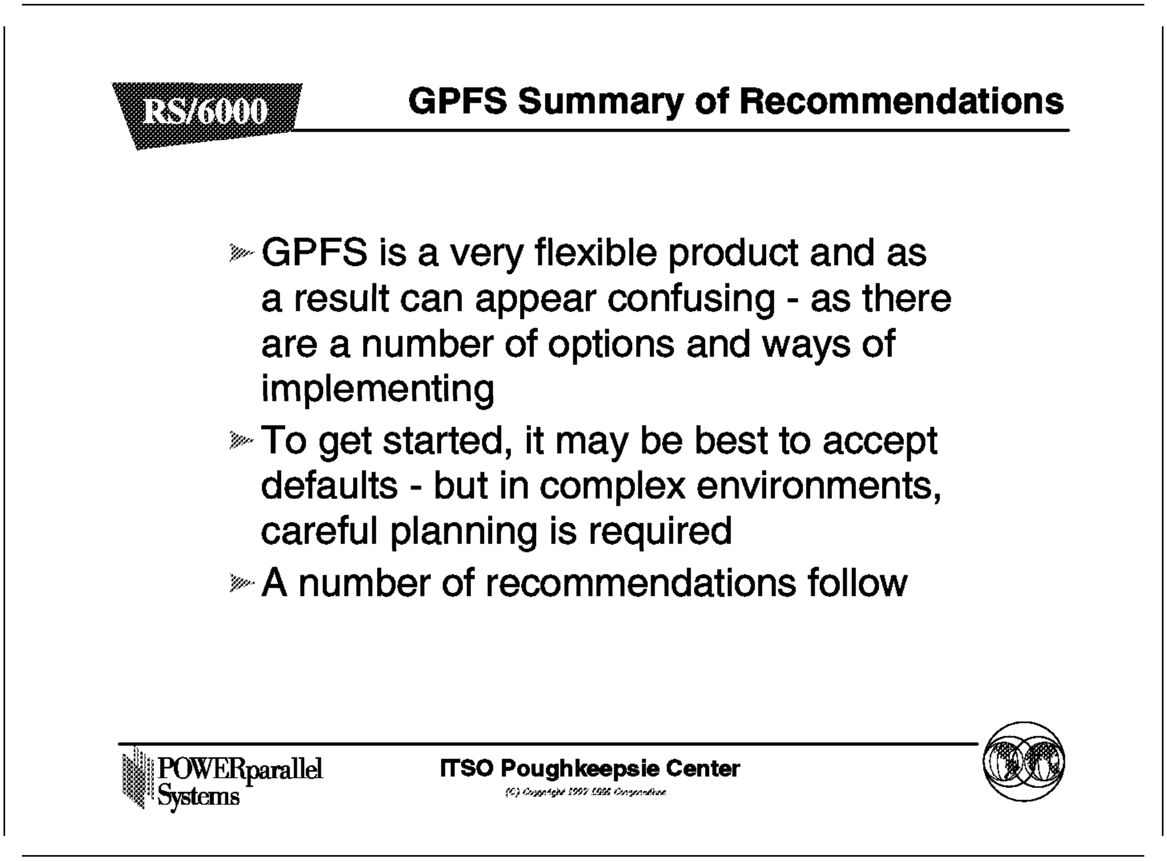
- Data Migration
  - if disk-space > 2 x data
    - copy files
  - else
    - export PIOFS
    - create VSDs, GPFS
      Note: i-nodes fixed at creation (R1)
    - import data

POWERparallel Systems

ITSO Poughkeepsie Center

Consider these factors when migrating from PIOFS.

## 6.15 Summary of Recommendations



RS/6000

GPFS Summary of Recommendations

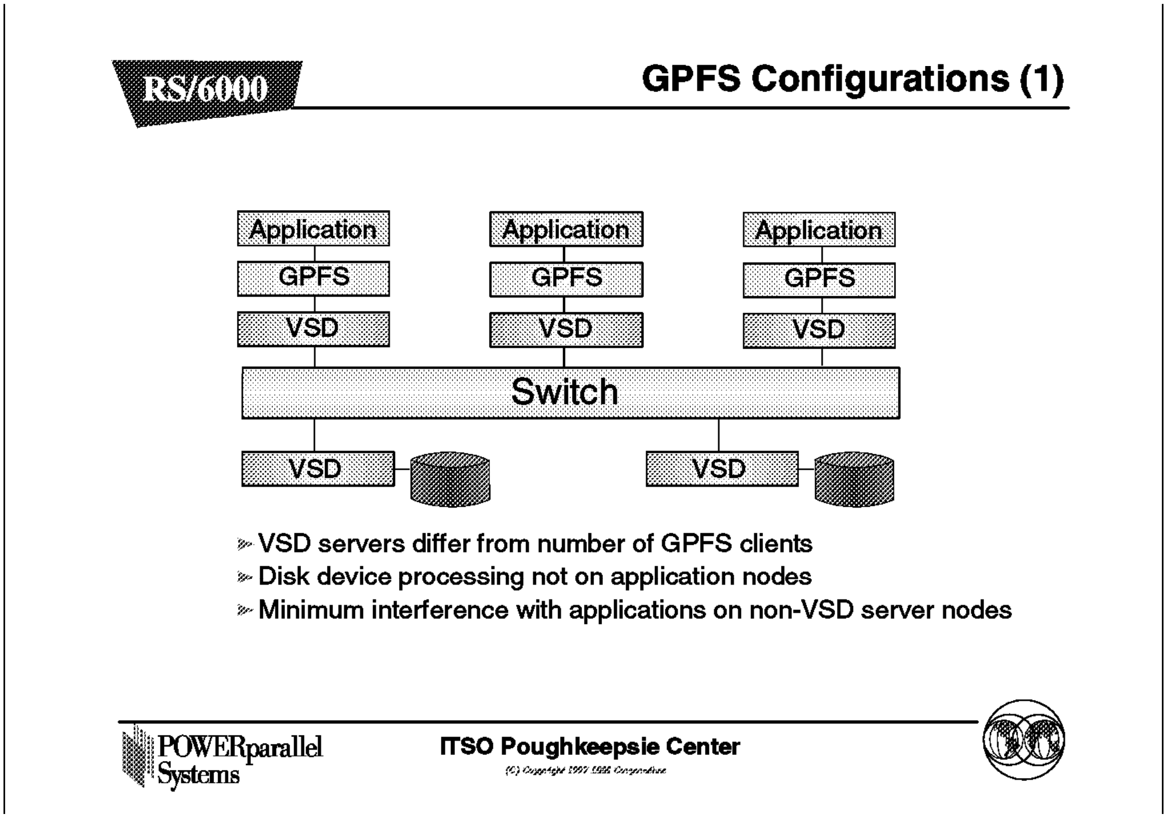
- GPFS is a very flexible product and as a result can appear confusing - as there are a number of options and ways of implementing
- To get started, it may be best to accept defaults - but in complex environments, careful planning is required
- A number of recommendations follow

POWERparallel Systems

ITSO Poughkeepsie Center

There are lots of options within GPFS. Keep your design simple and plan your implementation carefully.

## 6.15.1 Recommended Configurations



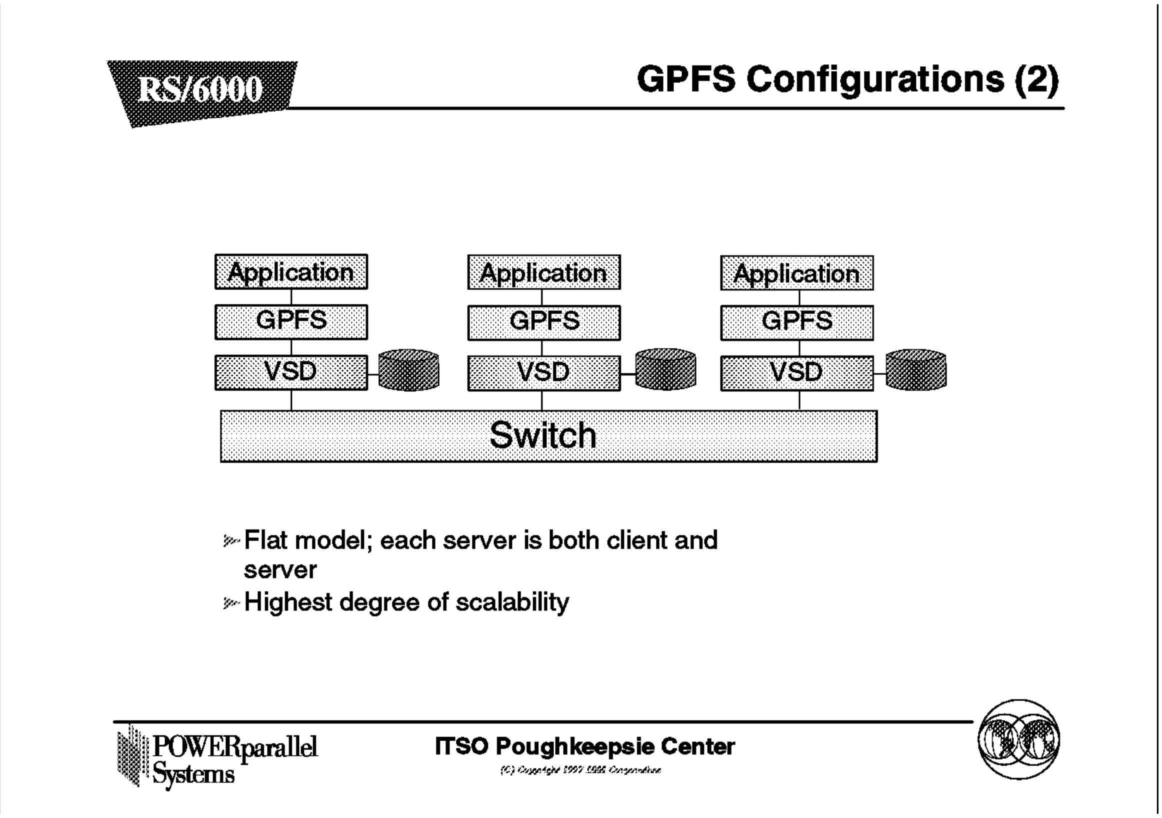
**GPFS Configurations (1)**

- VSD servers differ from number of GPFS clients
- Disk device processing not on application nodes
- Minimum interference with applications on non-VSD server nodes

ITSO Poughkeepsie Center

Here is one example, which is a preferred solution if such a solution can be cost justified. The VSD servers are on separate, dedicated nodes.

### 6.15.1.1  Recommended Configurations



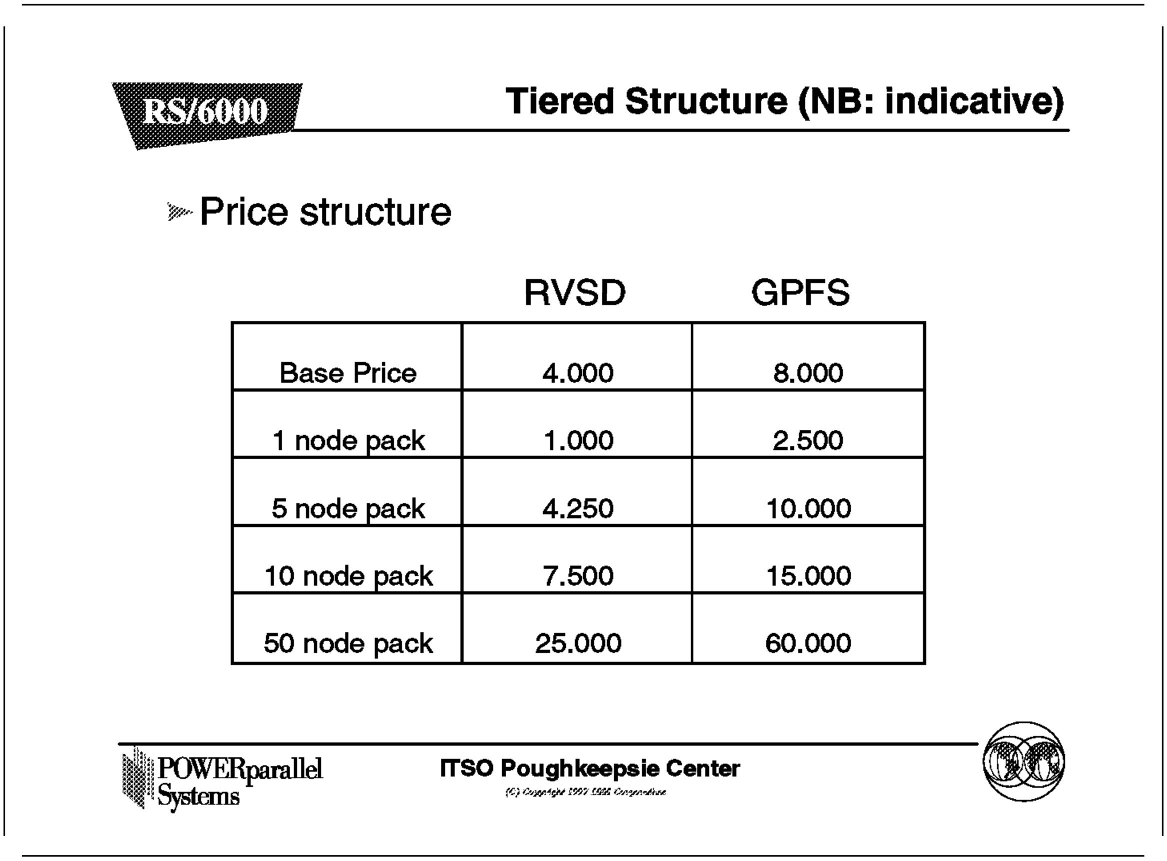
GPFS Configurations (2)

- Flat model; each server is both client and server
- Highest degree of scalability

ITSO Poughkeepsie Center

In this example, all nodes are VSD servers and also run the applications. Such a solution would be acceptable for a parallel application that has the same performance requirements across the nodes and exhibits a balanced workload.

## 6.16 GPFS Summary - Pricing Structure



RS/6000

Tiered Structure (NB: indicative)

- Price structure

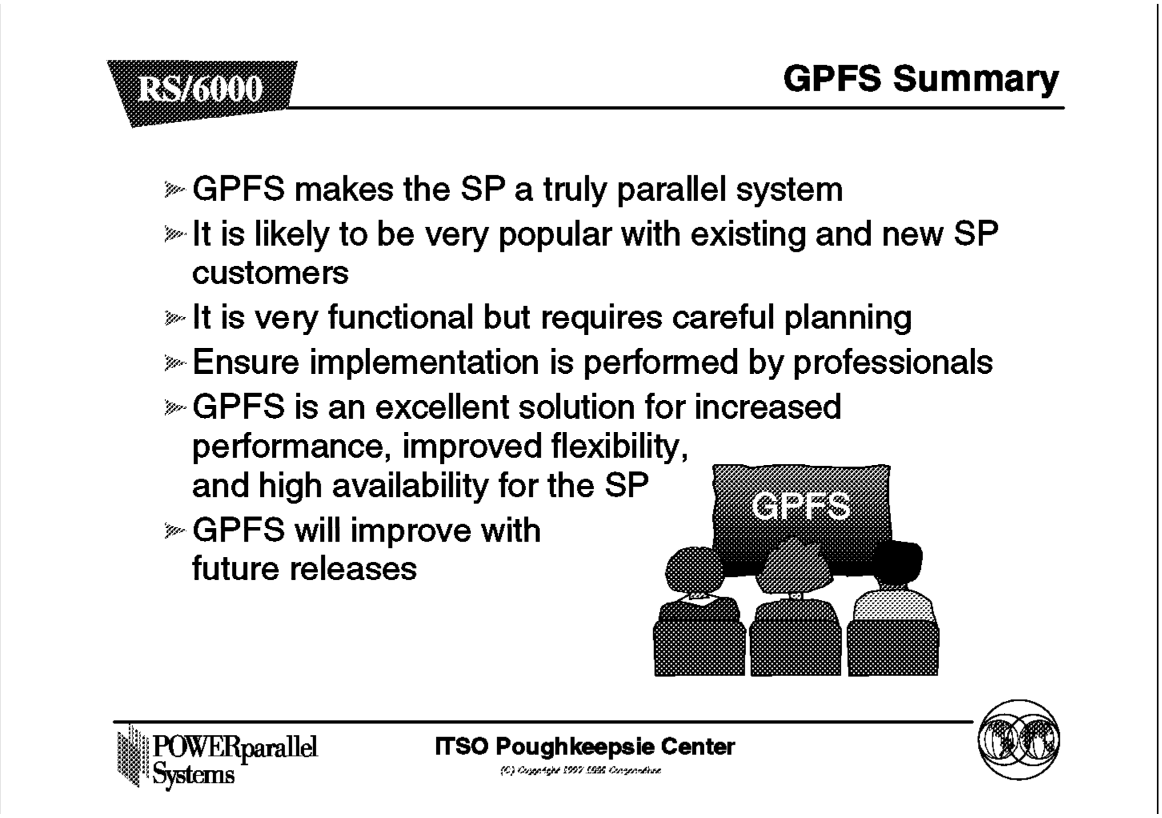
| | RVSD | GPFS |
|---|---|---|
| Base Price | 4.000 | 8.000 |
| 1 node pack | 1.000 | 2.500 |
| 5 node pack | 4.250 | 10.000 |
| 10 node pack | 7.500 | 15.000 |
| 50 node pack | 25.000 | 60.000 |

POWERparallel Systems

ITSO Poughkeepsie Center

Here are some examples of the pricing structure.

### 6.16.1 GPFS Summary



RS/6000

**GPFS Summary**

- GPFS makes the SP a truly parallel system
- It is likely to be very popular with existing and new SP customers
- It is very functional but requires careful planning
- Ensure implementation is performed by professionals
- GPFS is an excellent solution for increased performance, improved flexibility, and high availability for the SP
- GPFS will improve with future releases

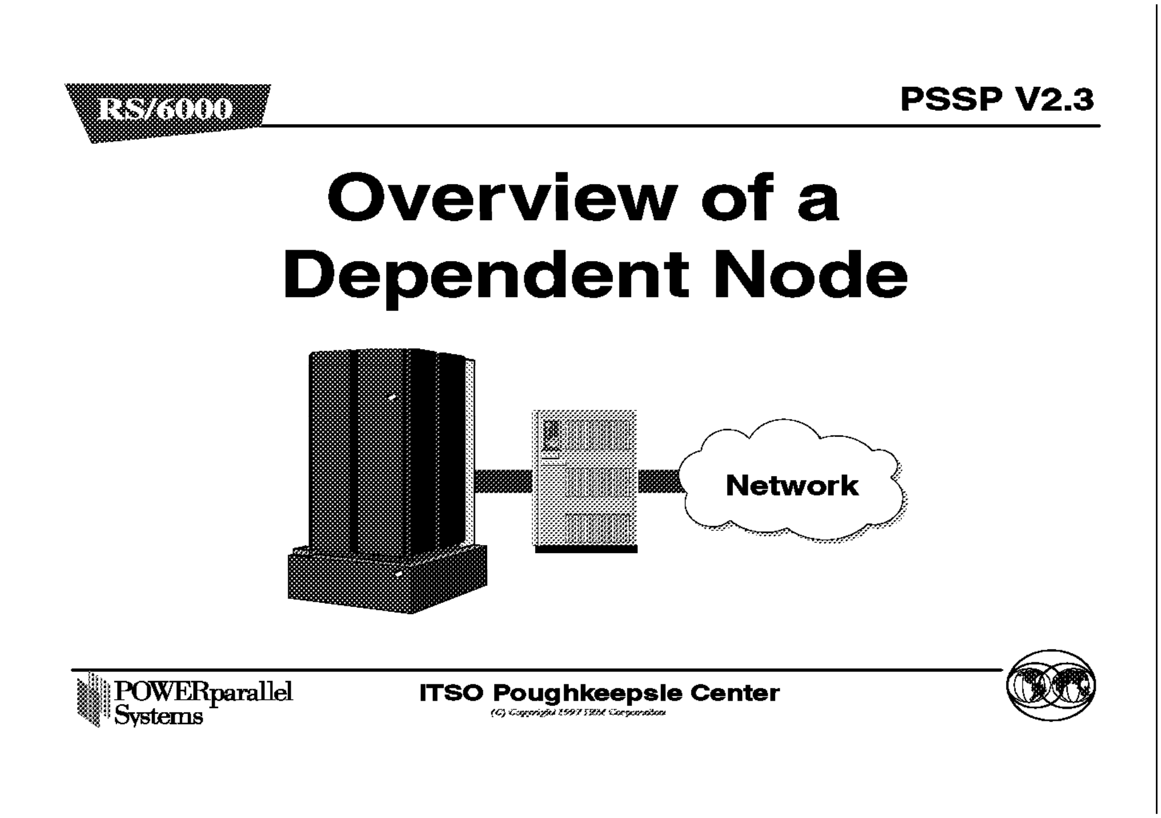
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In summary, GPFS seems a very good addition to the SP system. It provides excellent solutions for increased I/O performance, flexibility and availability.

# Chapter 7. Overview of a Dependent Node



RS/6000

PSSP V2.3

Overview of a Dependent Node

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

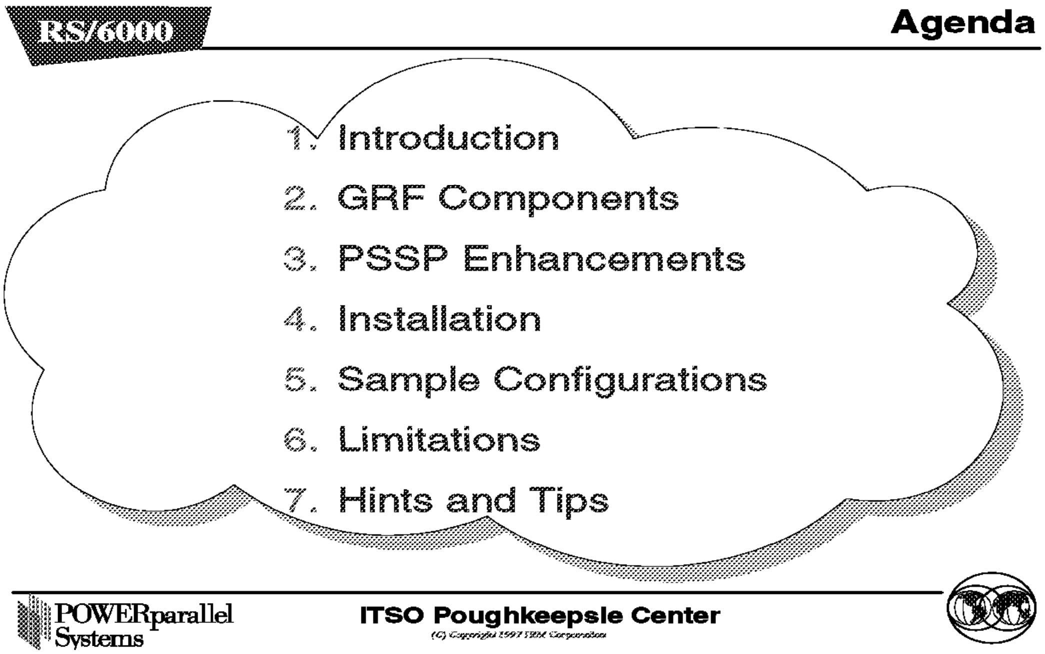
This chapter provides an overview of a dependent node in RS/6000 SP. We start by defining the dependent node, and the reasons for its design.

Next, we define a router, and introduce the one known as GRF. The GRF has a media card that attaches to the SP Switch. Together, they form the dependent node. We compare the routing process with and without the GRF.

We briefly describe the enhancements to the RS/6000 SP due to the introduction of the dependent node, and discuss tasks such as planning and installation using coexistence and partitioning with the dependent node.

To support the above, we introduce several sample GRF configurations.

We end by discussing some limitations of the dependent node, and by giving some hints and tips, both from our experience and about common problems.



RS/6000

Agenda

1. Introduction
2. GRF Components
3. PSSP Enhancements
4. Installation
5. Sample Configurations
6. Limitations
7. Hints and Tips

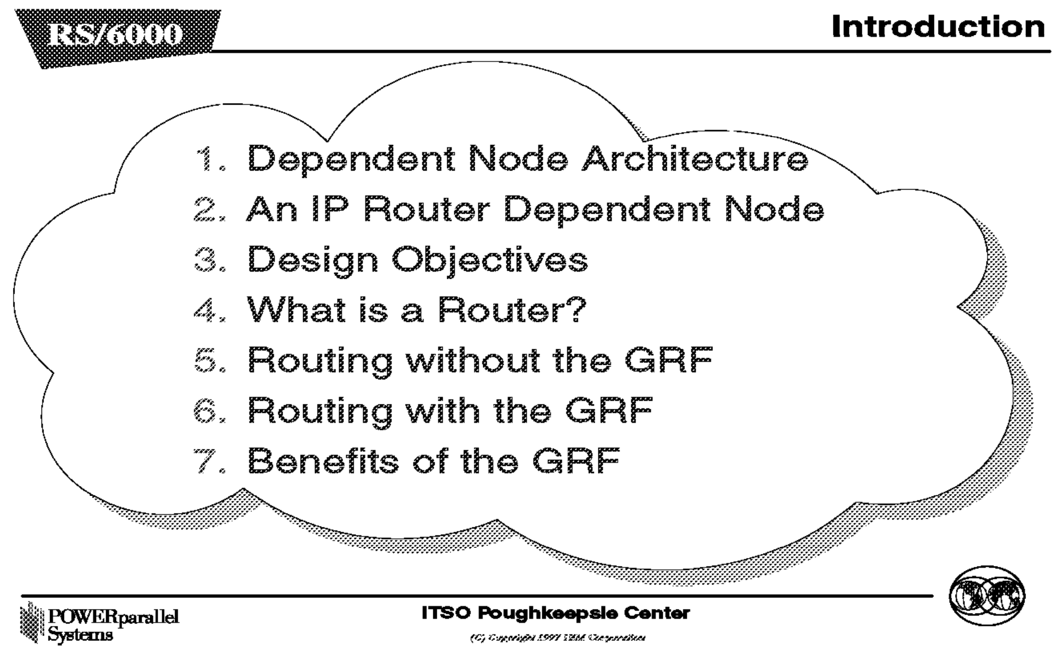
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This figure shows the agenda for this chapter.

## 7.1 Introduction



RS/6000

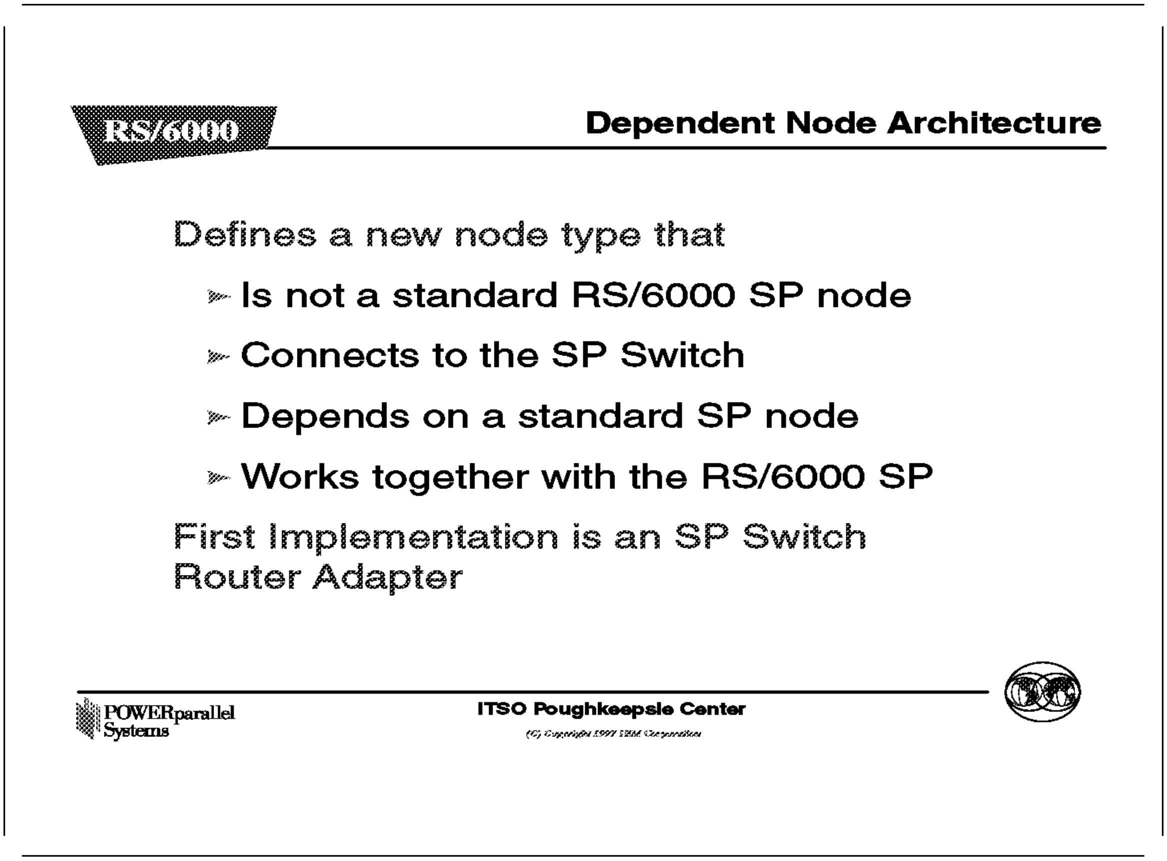
**Introduction**

1. Dependent Node Architecture
2. An IP Router Dependent Node
3. Design Objectives
4. What is a Router?
5. Routing without the GRF
6. Routing with the GRF
7. Benefits of the GRF

POWERparallel Systems

ITSO Poughkeepsie Center

This figure shows the agenda for the introduction to this chapter.



RS/6000

## Dependent Node Architecture

Defines a new node type that

- Is not a standard RS/6000 SP node
- Connects to the SP Switch
- Depends on a standard SP node
- Works together with the RS/6000 SP

First Implementation is an SP Switch Router Adapter

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

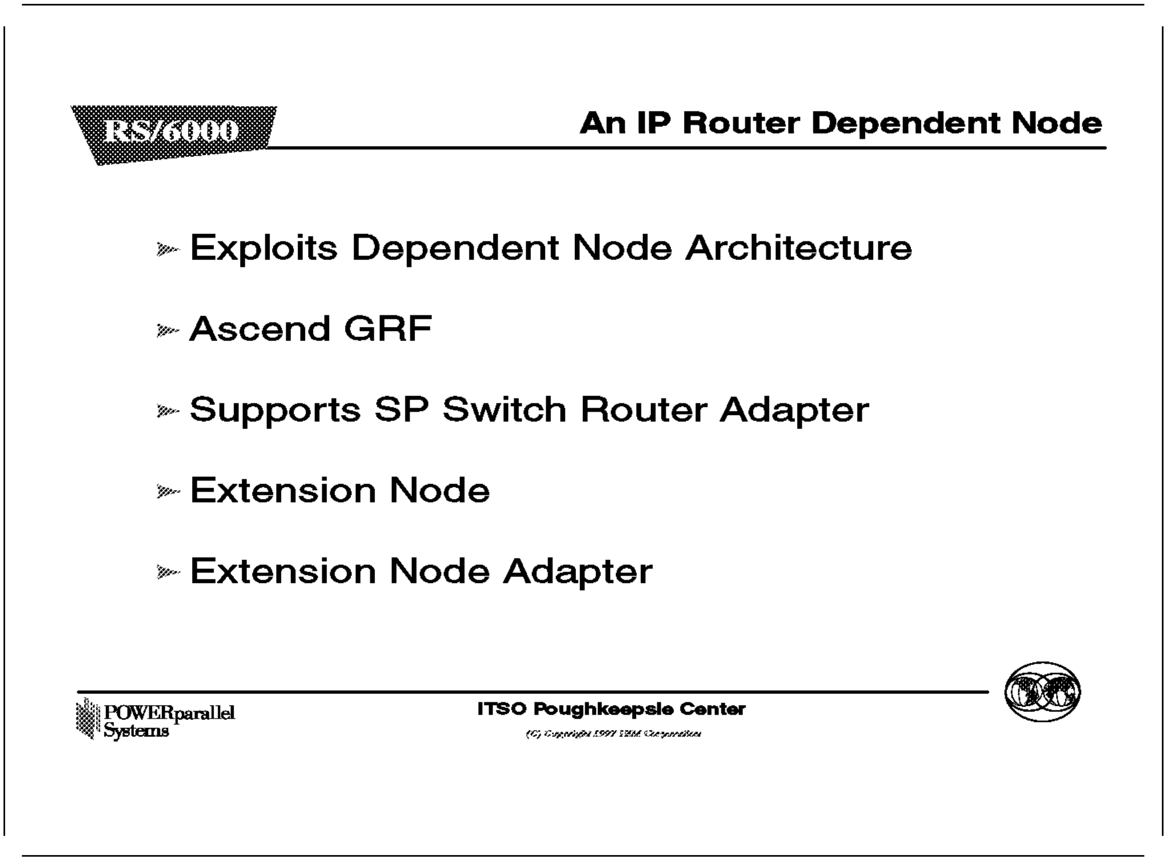
The Dependent Node Architecture refers to a processor or node, possibly not provided by IBM, for use with the RS/6000 SP.

Since this is not a regular RS/6000 SP node, not all the functions of the node can be performed on it. It relies on normal RS/6000 SP nodes to do some of its work, which is why it is called dependent. For example, it does not include all the functions of the complete fault service (worm) daemon, as other RS/6000 SP nodes with access to the SP Switch do.

The objective of this architecture is to allow the other processors or hardware to easily work together with the RS/6000 SP, extending the scope and capabilities of the system.

The Dependent Node connects to the RS/6000 SP Switch.

The SP Switch Router Adapter in the Ascend GRF is the first product to exploit the Dependent Node Architecture.


RS/6000

## An IP Router Dependent Node

- Exploits Dependent Node Architecture
- Ascend GRF
- Supports SP Switch Router Adapter
- Extension Node
- Extension Node Adapter

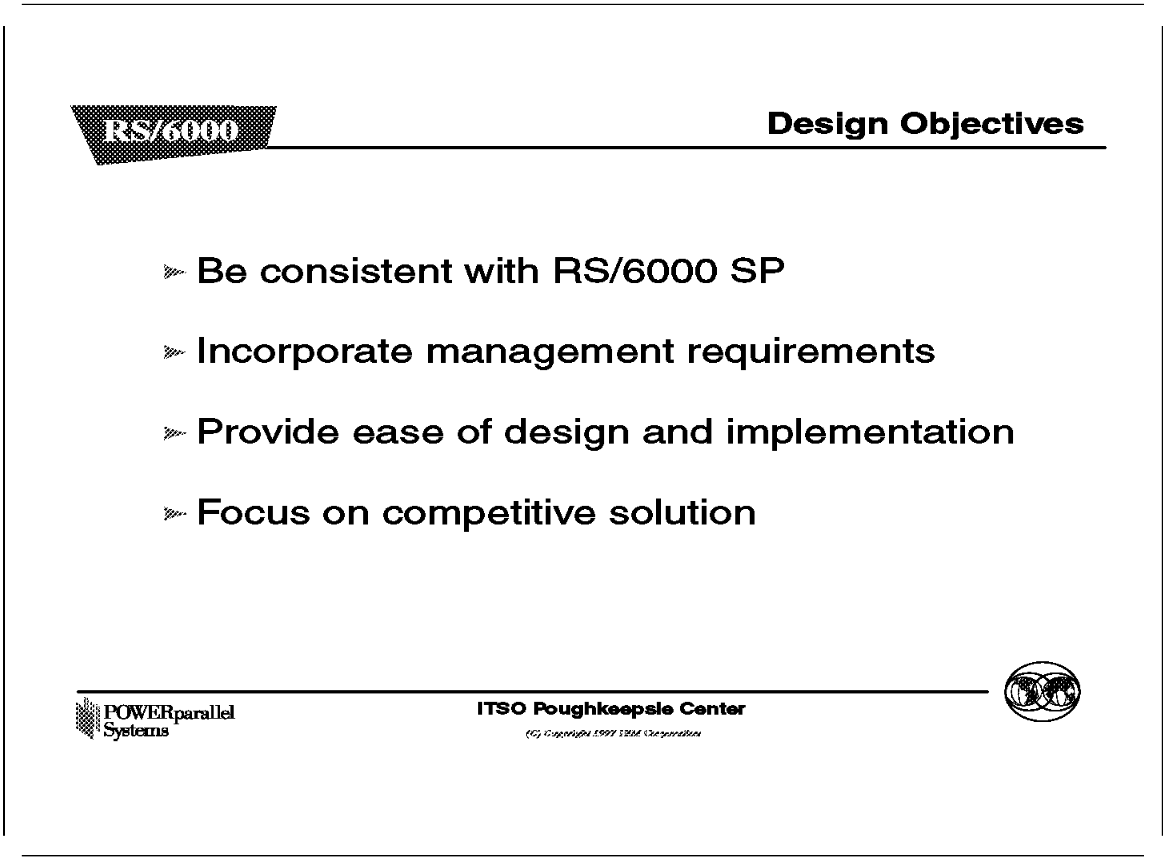
POWERparallel Systems
ITSO Poughkeepsie Center
(C) Copyright 1997 IBM Corporation

The first dependent node is actually a new SP Switch Router Adapter in a router. The purpose of this adapter is to allow the GRF, manufactured by Ascend, to forward SP Switch IP traffic to other networks. The GRF was known as the High Performance Gateway Node (HPGN) during the development of the adapter. IBM remarkets model of the GRF that connect to the SP Switch as the SP Switch Router model 04S and 16S (9077-04S and 9077-16S). These models are not available directly from Ascend. The rest of the book refers to the SP Switch Router as the GRF.

The distinguishing feature of the GRF, when compared with other routers, is that it has an SP Switch Router Adapter and, therefore, can connect directly into the SP Switch.

The RS/6000 SP software treats this adapter as an extension node. It is a node, because it takes up one port in the SP Switch and is assigned a node number. It is described as an extension, because it is not a standard RS/6000 SP node, but an adapter card that extends the scope of the RS/6000 SP.

Though *extension node* represents the node appearance of the adapter, it does not define the connection. An *extension node adapter* is used for that purpose. Each extension node has an extension node adapter to represent its connection to the SP Switch.



RS/6000

## Design Objectives

- Be consistent with RS/6000 SP
- Incorporate management requirements
- Provide ease of design and implementation
- Focus on competitive solution

POWERparallel Systems

ITSO Poughkeepsie Center

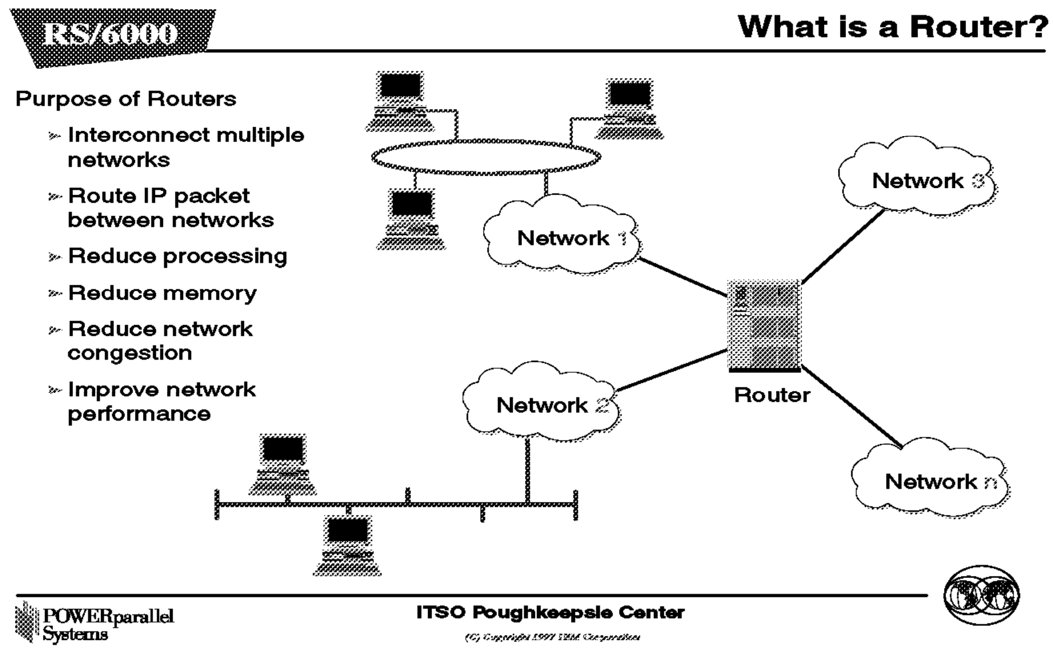
(C) Copyright 1997 IBM Corporation

Because the dependent node is part of the RS/6000 SP, it must be packaged and have some roles consistent with other RS/6000 SP nodes.

Changes must be made to the RS/6000 SP to incorporate management requirements for the dependent node.

Ease of design and implementation were important factors in the design of its support. This was accomplished by limiting the amount of switch-control protocol for the dependent node.

New SDR classes were created to manage dependent nodes. This was done to minimize the scope of the change and the exposure to side effects that dependent nodes may cause if they were represented as standard nodes in the SDR.



RS/6000

## What is a Router?

Purpose of Routers

- Interconnect multiple networks
- Route IP packet between networks
- Reduce processing
- Reduce memory
- Reduce network congestion
- Improve network performance

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

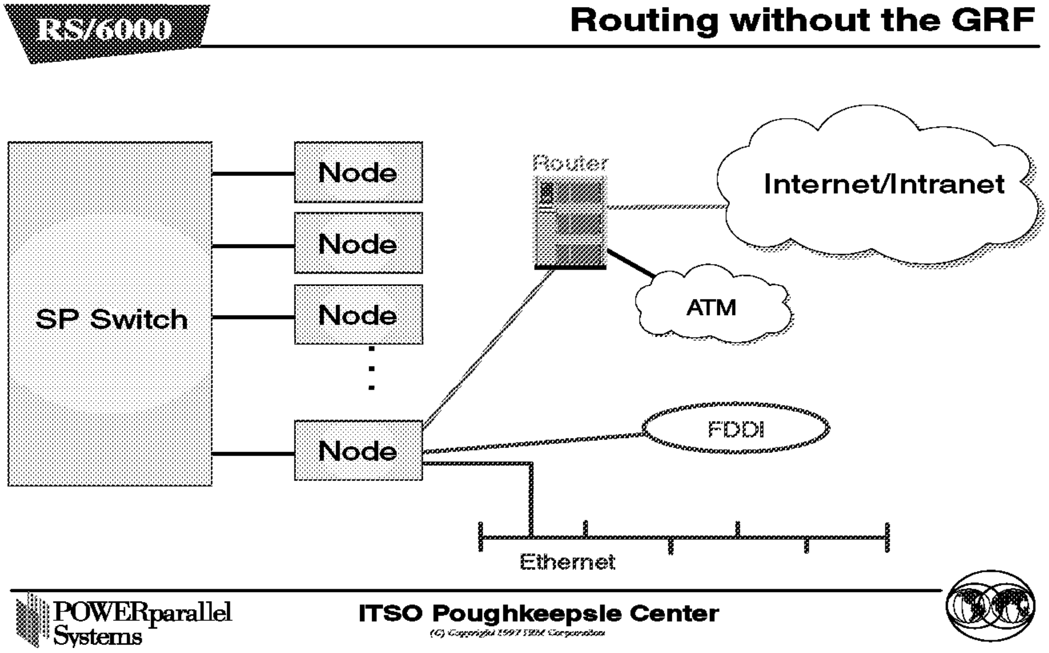
Routers serve a unique purpose in the world of networks. They interconnect networks so that Internet Protocol (IP) traffic can be routed between the systems in the networks.

Routers help to reduce the amount of processing required on local systems, since they perform the computation of routes to remote systems. A system can communicate with a remote system not in the local network by passing the message (or packets) to the router. The router works out how to get to the remote system and forwards the message appropriately.

Storing routes on the system takes up memory. Because it does not have to store routes to systems not in its own subnet, the route table uses less storage space, and thereby frees up memory for other work.

The use of routing reduces network traffic, because routers encourage subnetting, and subnetting creates a smaller network of systems. By having smaller networks, network traffic congestion is reduced and overall network performance is improved.

Benefits of reduced network congestion are better network traffic control and improved network performance.



RS/6000

## Routing without the GRF

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Before the GRF was available, there were only two ways for IP traffic from remote systems to reach the RS/6000 SP nodes:

1. You could put an additional IP adapter into every RS/6000 SP node.
2. You could designate one or two nodes to act as a router (as shown in this figure).

The first case was usually not chosen because of the cost involved. The following points explain why this option is expensive:

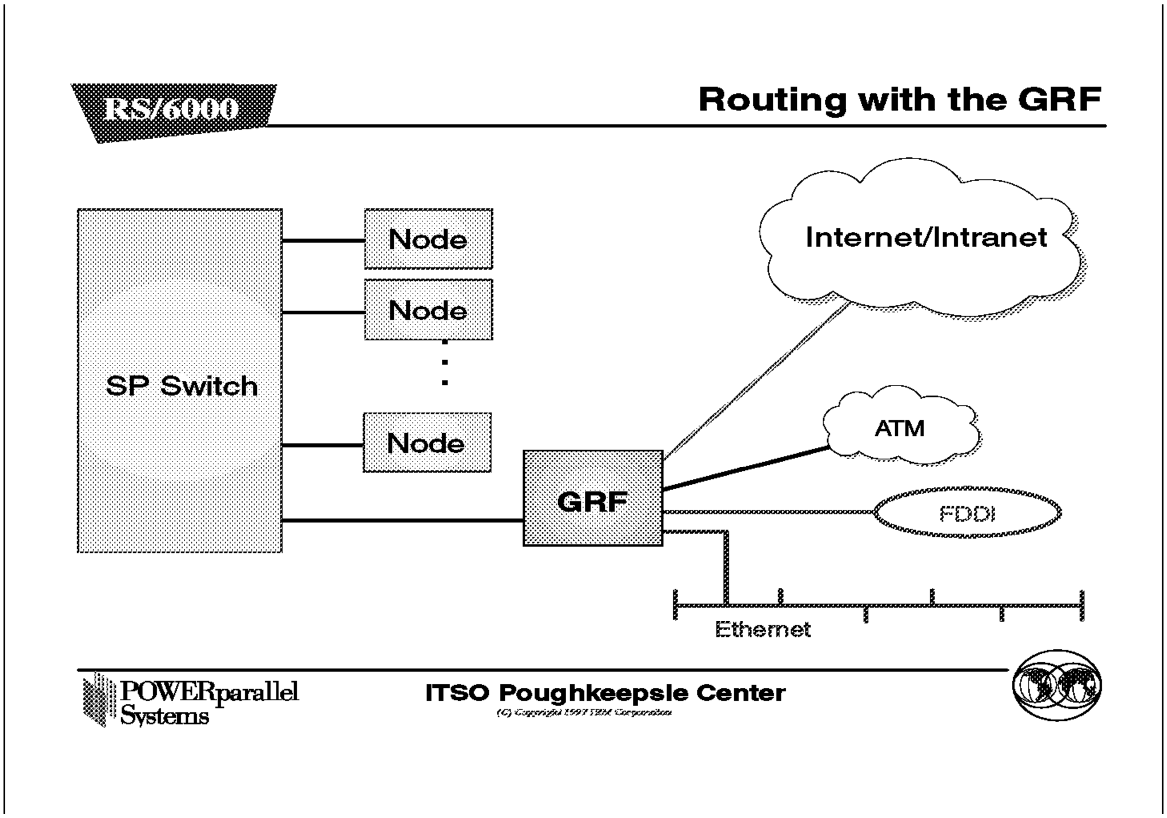
- Purchasing multiple IP adapters for each RS/6000 SP node can be expensive.
- The number of I/O slots in the RS/6000 SP node is limited. In addition, these slots are required to perform other tasks for the system, such as connecting to disk or tape. Using these I/O slots to connect IP adapters restricts the functions of the RS/6000 SP node.

The second case has proven to be very expensive as well. The RS/6000 SP node was not designed for routing. It is not a cost-effective way to route traffic for the following reasons:

- It takes many CPU cycles to process routing. The CPU is not a dedicated router and is very inefficient when used to route IP traffic (this processing can result in usage of up to 90%.).
- It takes a lot of memory to store route tables. The memory on the RS/6000 SP node is typically more expensive than router memory.

- The system I/O bus in the RS/6000 SP node is limited. The CPU on a node can only drive it at less than 80MB per second, which is less than what a high-end router can do.

For these reasons, the performance of routers in handling IP traffic from remote systems to the RS/6000 SP nodes was limited.



RS/6000

## Routing with the GRF

POWERparallel Systems

ITSO Poughkeepsie Center

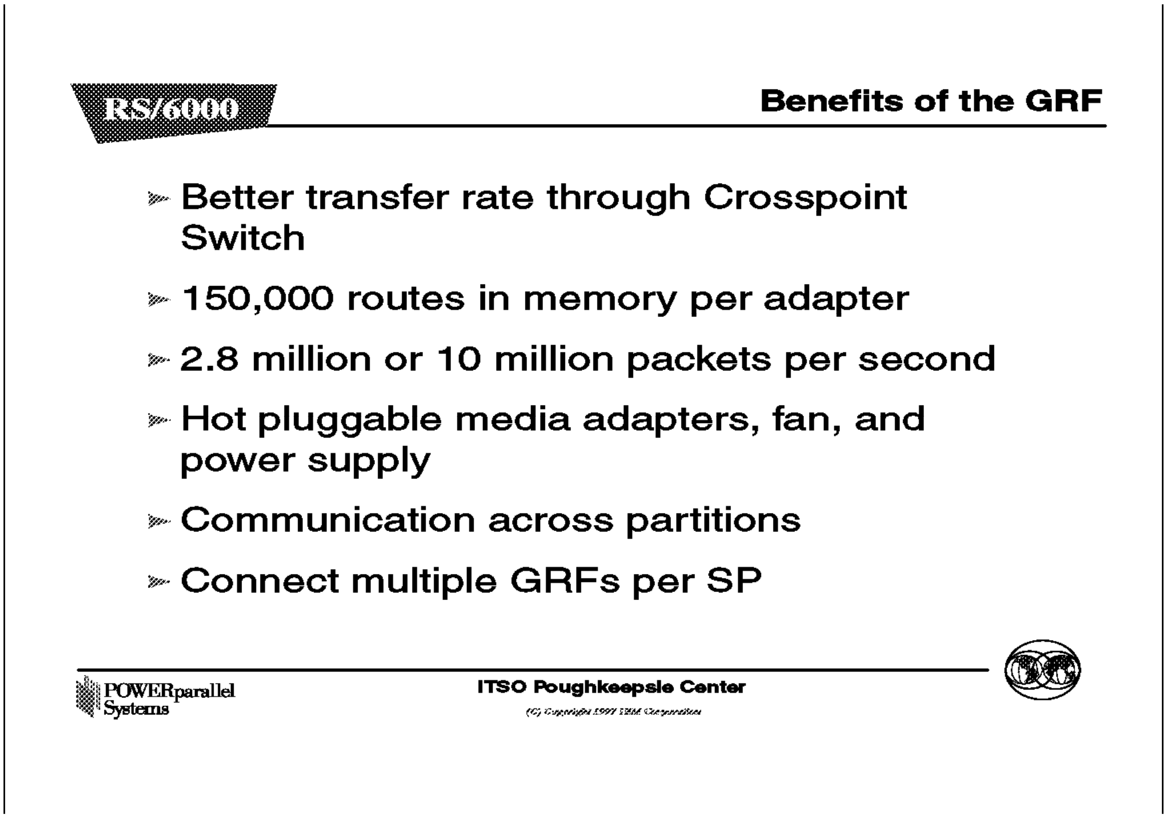
The GRF is a dedicated, high-performance router. Each SP Switch Router Adapter can route up to 30,000 packets per second and up to 100MB per second into the SP Switch network.

The GRF uses a Crosspoint Switch instead of an I/O bus to interconnect its adapters. This switch is capable of 4 to 16Gb per second and gives better performance than the MCA bus. Due to the high bandwidth that is available, communication between media adapters is improved.

Other advantages of using GRF are as follows:

- Availability of a redundant power supply
- Availability of a redundant fan
- Availability of a hot swappable power supply
- Availability of a hot swappable fan
- Availability of hot swappable media adapters (to connect to networks)
- Scalability of up to 4 or 16 media adapters depending on GRF models

Perhaps the greatest advantage of using the GRF is improved price/performance. As previously mentioned, the GRF is a dedicated router, and as such it is much more cost effective to route IP traffic to the RS/6000 SP nodes than another RS/6000 SP node in many high network throughput configurations.



RS/6000

**Benefits of the GRF**

- Better transfer rate through Crosspoint Switch
- 150,000 routes in memory per adapter
- 2.8 million or 10 million packets per second
- Hot pluggable media adapters, fan, and power supply
- Communication across partitions
- Connect multiple GRFs per SP

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The Crosspoint Switch is a *non-blocking crossbar*. This architecture is faster than an RS/6000 SP node, in which media adapters communicate through a microchannel bus.

To take advantage of the fast I/O provided by the Crosspoint Switch, fast route table access time is required. The GRF can store up to 150,000 routes in memory, while an RS/6000 SP node can store only hundreds. This means that the GRF is able to retrieve a route faster than an RS/6000 SP node.

The GRF is able to route up to 2.8 million packets per second for the 4-slot model and 10 million packets per second for the 16-slot model.

All the media adapters on the GRF are hot pluggable. This differs from using an RS/6000 SP node as your router. Should any network adapter on the RS/6000 SP node fail, the node has to be brought down to replace the faulty adapter. As a result, other unaffected network adapters will be brought down as well. The effect of bringing down the router will impact all the networks in the location.

Each RS/6000 SP is allowed to connect to multiple SP Switch Router Adapters. It does not matter whether these adapters are on different GRFs. Connecting multiple SP Switch Router Adapters to either different partitions in an RS/6000 SP or to different RS/6000 SPs allows them to communicate with each other and the other GRF media adapters via the SP Switch. A more detailed discussion of this is found in the Coexistence figure in the PSSP Enhancement section.

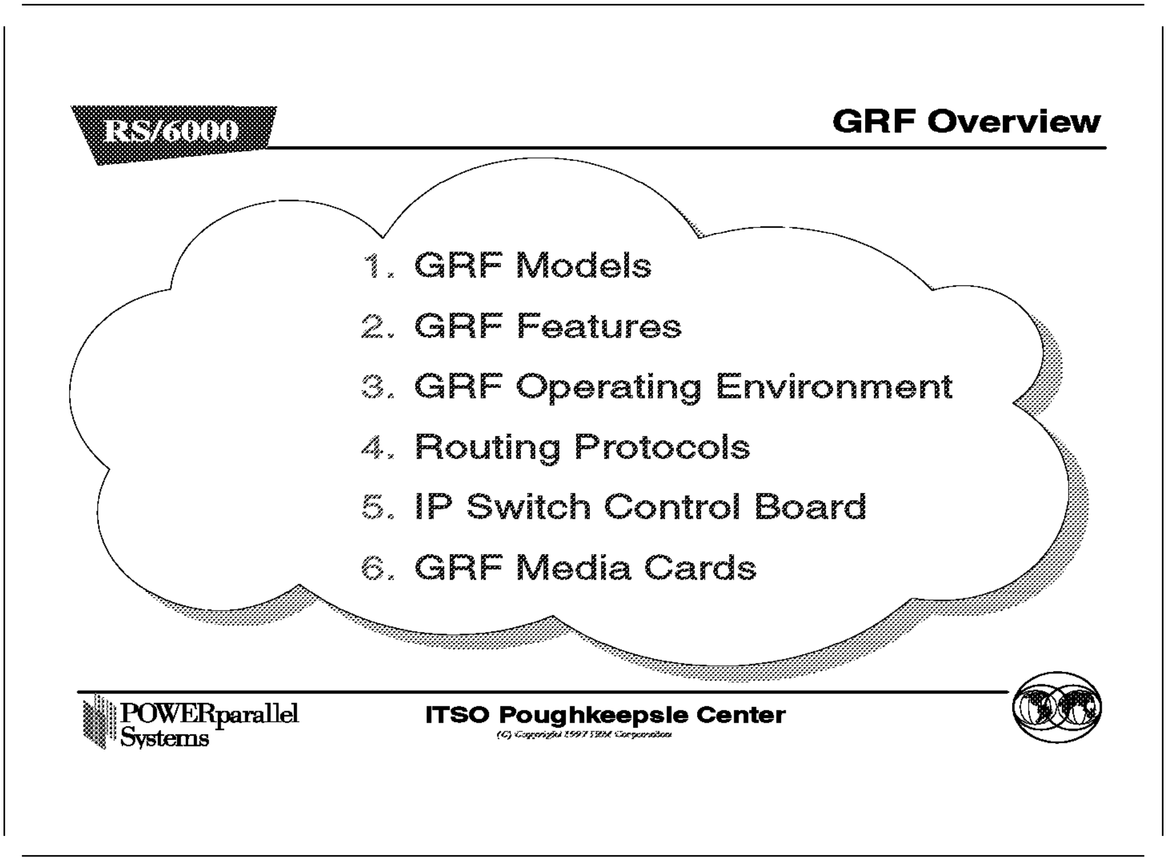
**Attention**

The SP Switch Router model 04S can support four media cards such as FDDI or ATM. The SP Switch Router model 16S can support 16. In either case, multiple SP Switch Router Adapters may be installed in the SP Switch Router. Check the final version of the SP product documentation to determine the maximum number of SP Switch Router Adapters supported in each model.

**Note:** The number of packets that the GRF can route per second depends on the following:

- The type of media adapter
- The size of the packet

## 7.2 GRF Overview



RS/6000

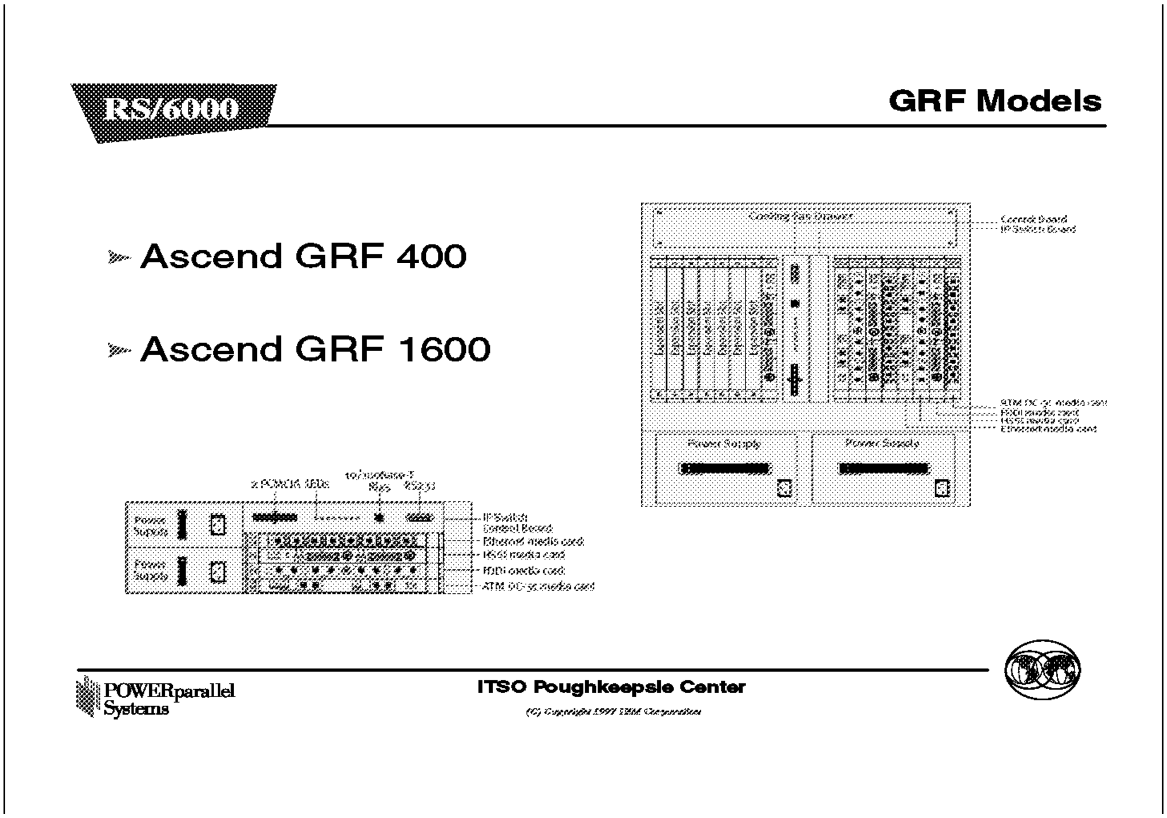
GRF Overview

1. GRF Models
2. GRF Features
3. GRF Operating Environment
4. Routing Protocols
5. IP Switch Control Board
6. GRF Media Cards

POWERparallel Systems

ITSO Poughkeepsle Center

This section describes the major components of the GRF.



RS/6000

## GRF Models

- Ascend GRF 400
- Ascend GRF 1600

POWERparallel Systems

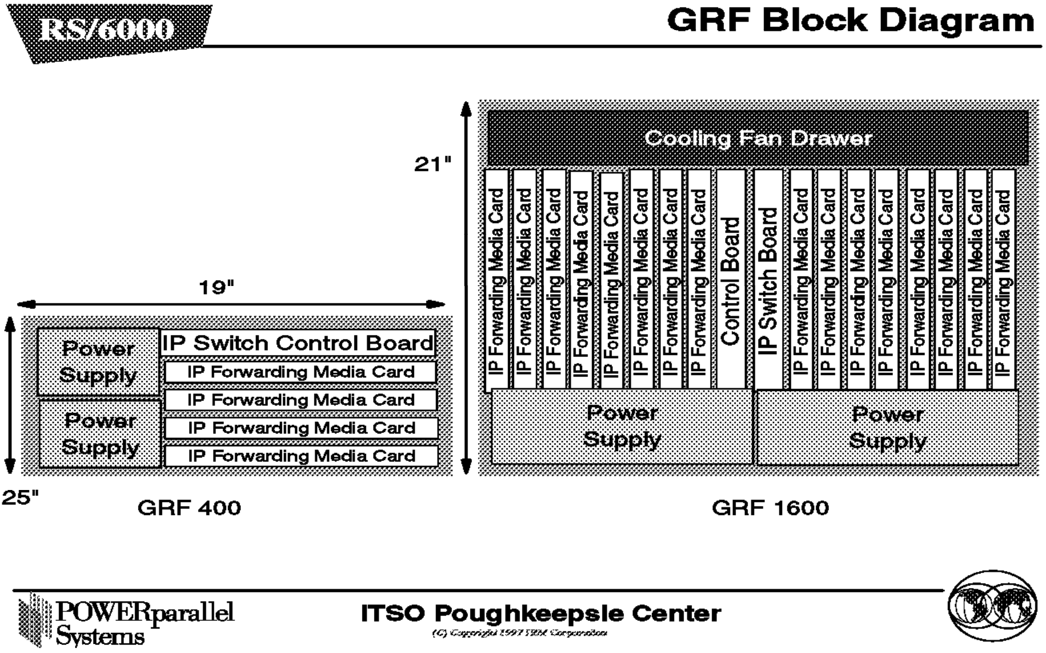
ITSO Poughkeepsie Center

The GRF 400 can accommodate up to four media adapters.

The GRF 1600 can accommodate up to 16 media adapters.

Each adapter allows the GRF to connect to one or more networks.

Each of the models has an additional slot for the IP Switch Control Board, which is used to control the router.



This figure shows the two GRF models: the 4-slot and the 16-slot model. Detailed descriptions of each follow.

## 7.2.1 GRF 400

**Part** — **Description**

**Cooling Fans**

These are located at the right side of the chassis and cannot be accessed without bringing down the GRF. There is no redundant fan built into this model, and since the fans can only be accessed by bringing down the GRF, this model is *not* hot swappable.

**Media Cards**

There are four media card slots on this chassis. They are slotted horizontally and are located at the bottom of the chassis.

**IP Switch Control Board**

The IP Switch Control Board is located at the top of the four media slots and is also slotted horizontally.

**Power Supply**

The left side of the chassis is reserved for the two power supplies that are required for redundancy. The failed power supply can be hot swapped out of the GRF chassis. The second power supply is optional for this model.

### 7.2.2 GRF 1600

**Part** **Description**

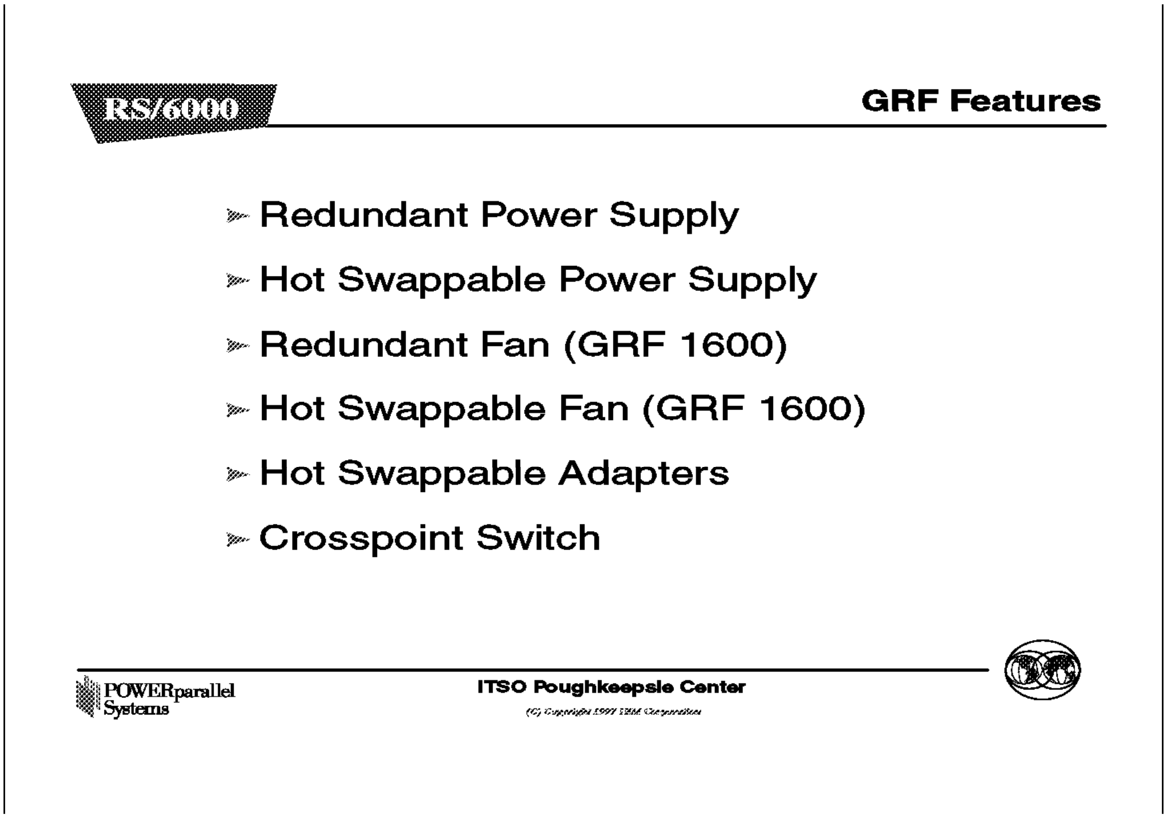
**Cooling Fans** These are located at the top of the chassis, and can be accessed separately from the other parts of the GRF. The redundant fans built into the system are therefore hot swappable.

**Media Cards** There are 16 media card slots on this chassis. They are slotted vertically. Eight of the cards are on the left side of the chassis, eight are on the right.

**IP Switch Control Board**

The IP Switch Control Board is located in the middle of the 16 media slots and is also slotted vertically.

**Power Supply** The base of the chassis is reserved for the two power supplies that are required for redundancy. The failed power supply can be hot swapped out of the GRF chassis.



RS/6000

## GRF Features

- Redundant Power Supply
- Hot Swappable Power Supply
- Redundant Fan (GRF 1600)
- Hot Swappable Fan (GRF 1600)
- Hot Swappable Adapters
- Crosspoint Switch

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

GRF has the following features:

- Redundant Power Supply

  Should any power supply fail, a message is sent to the control board. The power supply will automatically reduce its output voltage if the temperature exceeds 90°C or 194°F. If the voltage falls below 180V, the GRF will automatically shut down.

- Hot Swappable Power Supply

  The faulty power supply can be replaced while the GRF is in operation.

- Redundant Fan

  For the GRF 1600 model, if one fan breaks down, a message is sent to the control board.

  For both models, when the temperature reaches 53°C or 128°F, an audible alarm sounds continuously, and a message is sent to the console and logged into the message log. If the temperature exceeds 57.5°C or 137°F, the GRF will do an automatic system shutdown.

- Hot Swappable Fan

  For the GRF 1600 model, the cooling fan can be replaced while the GRF is in operation.

- Hot Swappable Adapters

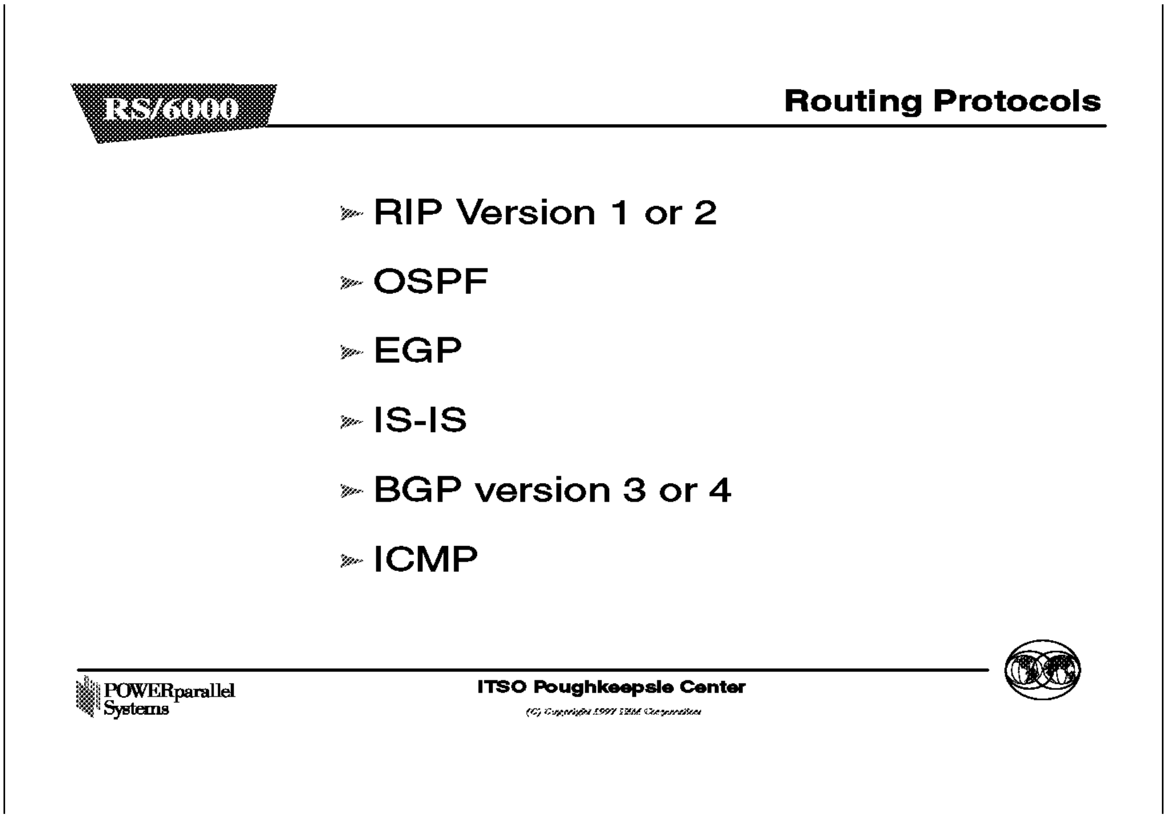
There are two types of adapters on the GRF: the Media Adapters and the IP Switch Control Board.

The media adapters are independent of each other, and can be replaced or removed without affecting any other adapter or the operation of the GRF.

However, the IP Switch Control Board is critical to the GRF. Should this board be unavailable, the router will fail.

- Crosspoint Switch

  The Crosspoint Switch is a 16x16 (16Gb per second) or 4x4 (4Gb per second) crossbar switch for the GRF 1600 and GRF 400, respectively. It is the I/O path used when the media adapters need to communicate with each other.



RS/6000

## Routing Protocols

- RIP Version 1 or 2
- OSPF
- EGP
- IS-IS
- BGP version 3 or 4
- ICMP

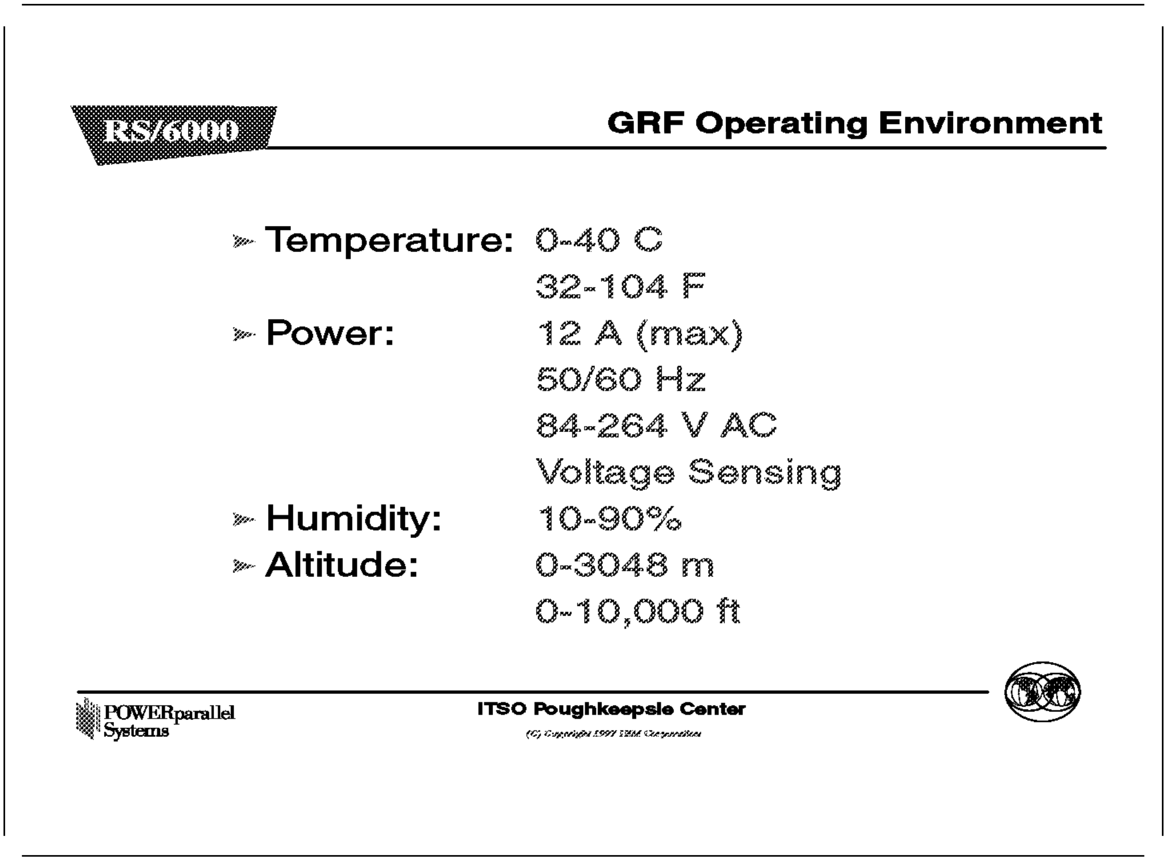
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In addition to static routes, various routing protocols are available on the GRF, as follows:

| | |
|---|---|
| **RIP** | Routing Information Protocol Version 1 or 2 (RIP 1 or 2) |
| **OSPF** | Open Shortest Path First |
| **EGP** | Exterior Gateway Protocol |
| **IS-IS** | Intermediate System to Intermediate System (an OSI gateway protocol) |
| **BGP** | Border Gateway Protocol Version 3 or 4 (BGP 3 or 4) |
| **ICMP** | Internet Control Message Protocol |



RS/6000

## GRF Operating Environment

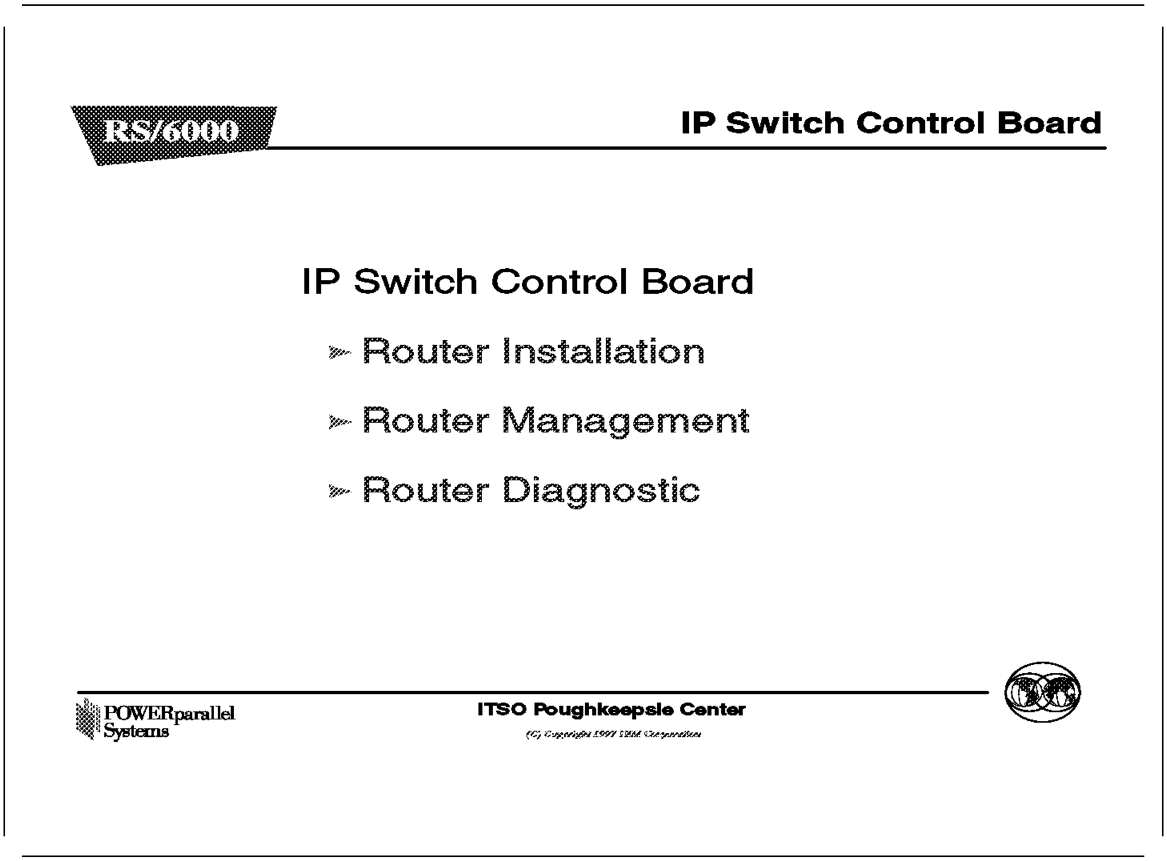
| | |
|---|---|
| Temperature: | 0-40 C<br>32-104 F |
| Power: | 12 A (max)<br>50/60 Hz<br>84-264 V AC<br>Voltage Sensing |
| Humidity: | 10-90% |
| Altitude: | 0-3048 m<br>0-10,000 ft |

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

As mentioned in the previous figure on GRF Features, the operating temperature should not exceed 53°C or 128°F. Even though there is a buffer between the operating temperature and the warning temperature, it is best to keep the temperature within the operating level in order to minimize the possibility of damage to GRF components.



RS/6000

IP Switch Control Board

IP Switch Control Board

- Router Installation
- Router Management
- Router Diagnostic

POWERparallel Systems

ITSO Poughkeepsie Center

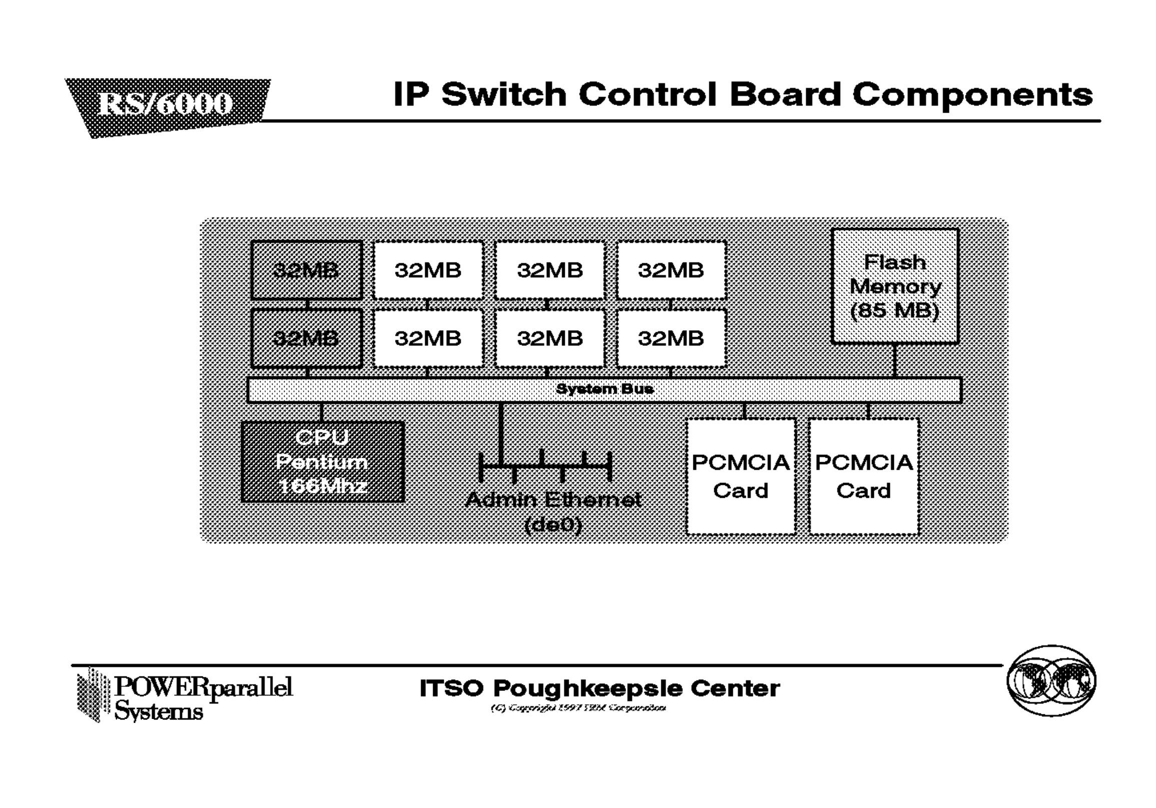
(C) Copyright 1997 IBM Corporation

The control board, also known as the IP Switch Control Board, is accessed through Telnet or a locally attached VT100 terminal. The IP Switch Control Board is supplied with the GRF and is necessary for its operation. The VT100 terminal is not supplied with the GRF. It is only required for the installation of the GRF. After installation, all future access to the GRF is through Telnet to the IP Switch Control Board′s administrative Ethernet.

The IP Switch Control Board is identified as slot 66 in the GRF. The CPU in the IP Switch Control Board is a 166MHz Pentium processor and runs a variant of BSD UNIX as its operating system. Thus, the GRF administrator is assumed to be proficient in UNIX.

The IP Switch Control Board is used to install, boot, and configure the router and its media adapters.

It is also used for the logging of messages, the dumping of memory and status, and to perform diagnostic checking of both the GRF and the media adapters.



IP Switch Control Board Components

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Let us examine the IP Switch Control Board in more detail.

Following are descriptions of its components as shown in the figure:

**Item** **Description**

**Memory**

The IP Switch Control Board comes standard with 64MB of memory (the two shaded blocks of 32MB of memory in the top left hand corner).

The IP Switch Control Board memory can be upgraded to 256MB, in increments of 64MB (the six white blocks of memory).

Each column of 64MB of memory is split into two parts. The system uses the bottom half of the memory (32MB) for file system storage. The top half is used for applications such as the SNMP agent, the gated daemon, and for the operating system.

**Flash memory**

This memory (the 85MB ATA flash memory on the system) is used to store the operating system information and the configuration information for the GRF.

**System bus**

Used by the IP Switch Control Board components to communicate with each other.

**Pentium processor** This 166MHz processor drives the IP Switch Control Board and the GRF. As mentioned in the earlier figure, this processor runs a variant of BSD UNIX, and so it is useful for the GRF administrator to have UNIX management skills.

**Administrative Ethernet** This Ethernet is known to the GRF as de0. This port supports the 10BaseT or the 100BaseT Ethernets, and switches between them automatically, depending on the type of network used.

To use 10Base2 or 10Base5, the user must add a transceiver (supplied by the user).

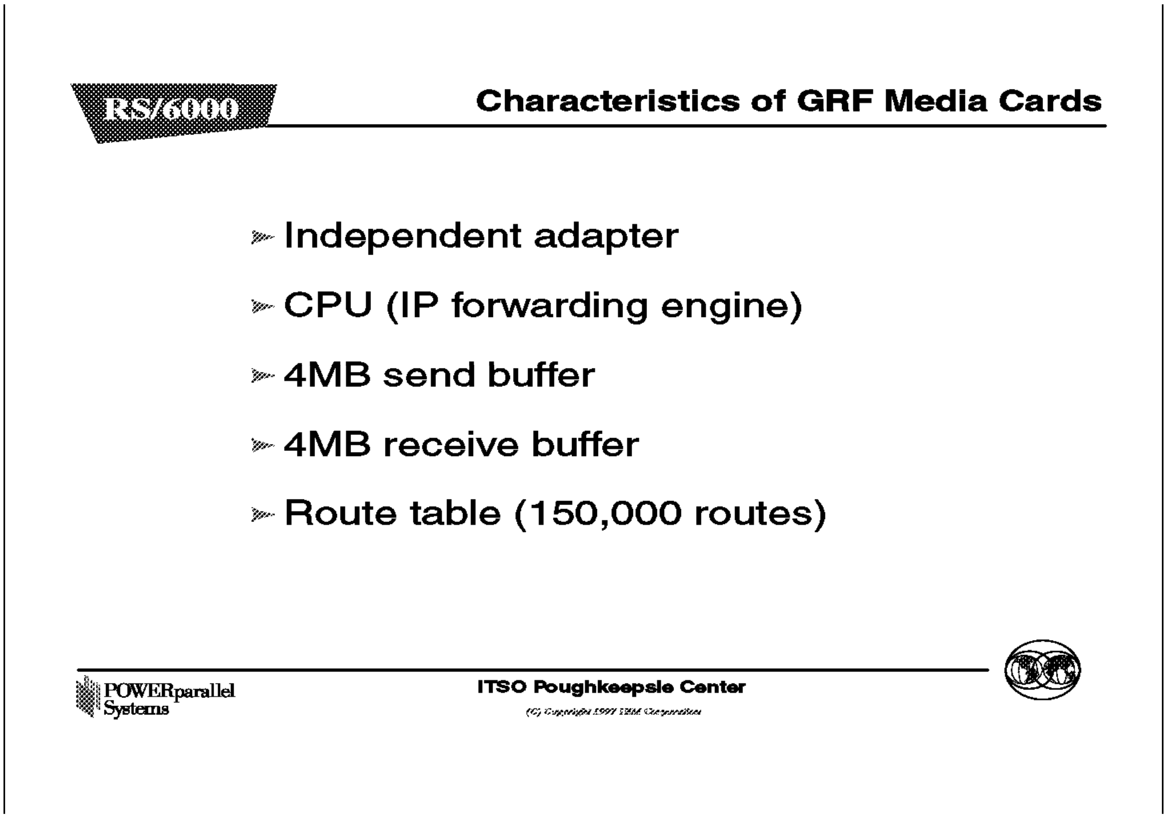
**PCMCIA cards** The two white blocks at the bottom right hand corner of the figure are PCMCIA slots.

There are two types of PCMCIA cards:

- The PCMCIA 85MB flash memory card, available as an optional device, is used to back up the system. It is similar to a tape drive on a normal system.
- The PCMCIA modem card, also available as an optional device, allows the user to dial into the GRF through a modem to administer it remotely.

  **Note:** For the initial setup, the console must be available locally, not through the modem.

Additionally, the RS232 port (which is not shown in the figure) allows you to connect the VT100 console by using an RS232 null modem cable. The console and cable must be supplied by the user.



RS/6000

## Characteristics of GRF Media Cards

- Independent adapter
- CPU (IP forwarding engine)
- 4MB send buffer
- 4MB receive buffer
- Route table (150,000 routes)

POWERparallel Systems

ITSO Poughkeepsie Center

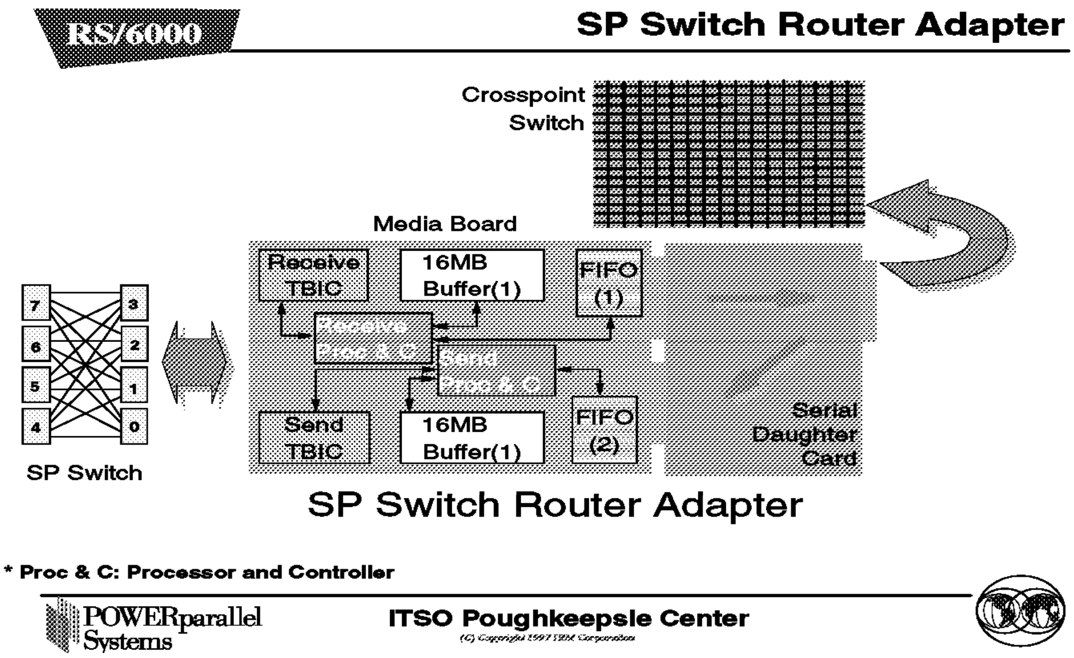
(C) Copyright 1997 IBM Corporation

All GRF media cards (media adapters) are self-contained and independent of other media adapters.

Each media card has an onboard processor that is responsible for IP forwarding on the media adapter.

Each media card has two independent memory buffers, a 4MB send buffer and a 4MB receive buffer. These buffers are necessary to balance the speed differences between the media adapters, because they have different transfer rates.

Each onboard processor has local memory that can contain a local route table with up to 150,000 entries, to be used for routing on the media adapter. Because these route entries are in local memory, access to them is very fast. When the media adapter is started up, it gets its initial route entries from the IP Switch Control Board.



RS/6000

SP Switch Router Adapter

* Proc & C: Processor and Controller

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The GRF supports a number of media adapters. This figure describes the SP Switch Router Adapter in detail. This adapter allows the GRF to connect directly into the SP Switch.

The SP Switch Router Adapter is made up of two parts:

- The media board
- A serial daughter card

The serial daughter card is an interface for the media board into the Crosspoint Switch. This switch is the medium by which the different GRF (media) adapters talk to each other.

The purpose of the media board is to route IP packets to their intended destination through the GRF. The SP Switch Router Adapter described here is used for routing IP packets to and from the SP Switch to other systems connected directly or indirectly to the GRF. A brief description of the components on the media board follows.

**Receive TBIC** This component receives data segments from the SP Switch and notifies the Receive Controller and Processor that there is data to be transfered to the buffer.

**Receive Controller and Processor**

This component recognizes the SP Switch segments and assembles them into IP packets in the 16MB buffer. Up to 256 simultaneous IP datagrams can be handled simultaneously. When a complete IP packet has been received, the Receive Controller sends the packet to the FIFO (1) queue for transfer to the serial daugther card.

**Buffer (1)** This component is segmented into 256 64KB IP packet buffers. It is used to reassemble IP packets before being sent to the FIFO queue, as switch data segments may arrive out of order and interleaved with segments belonging to different IP packets.

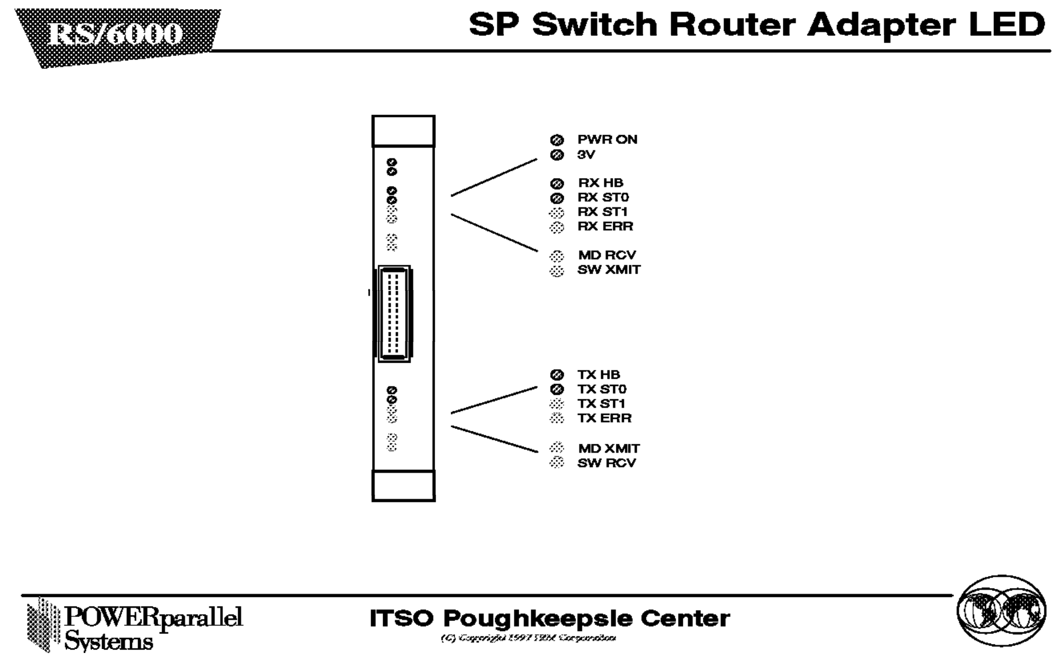
**FIFO (1)** This component is used to transfer complete IP packets to the serial daughter card and even the flow of data between the SP and the GRF backplane.

**FIFO (2)** This component receives IP packets from the serial daughter card and transfers them to the Buffer (2).

**Buffer (2)** This buffer is used to temporarily store the IP packet while its IP address is examined and a proper SP Switch route is set up to transfer the packet through the SP Switch.

**Send Processor and Controller** This component is notified when an IP packet is received in the FIFO (2) queue and sets up a DMA transfer to send the packet to Buffer (2). The Send Processor looks up the IP address in the packet header and determines the SP Switch route for the packet, before notifying the Send Controller to send the packet to the Send TBIC from Buffer (2).

**Send TBIC** This component receives data from Buffer (2) and sends it in SP Switch data segments to the SP Switch.



RS/6000

SP Switch Router Adapter LED

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

LED activities during operations are listed in Table 5, Table 6 on page 314, and Table 7 on page 314.

Table 5. SP Switch Router Adapter Media Card LEDs

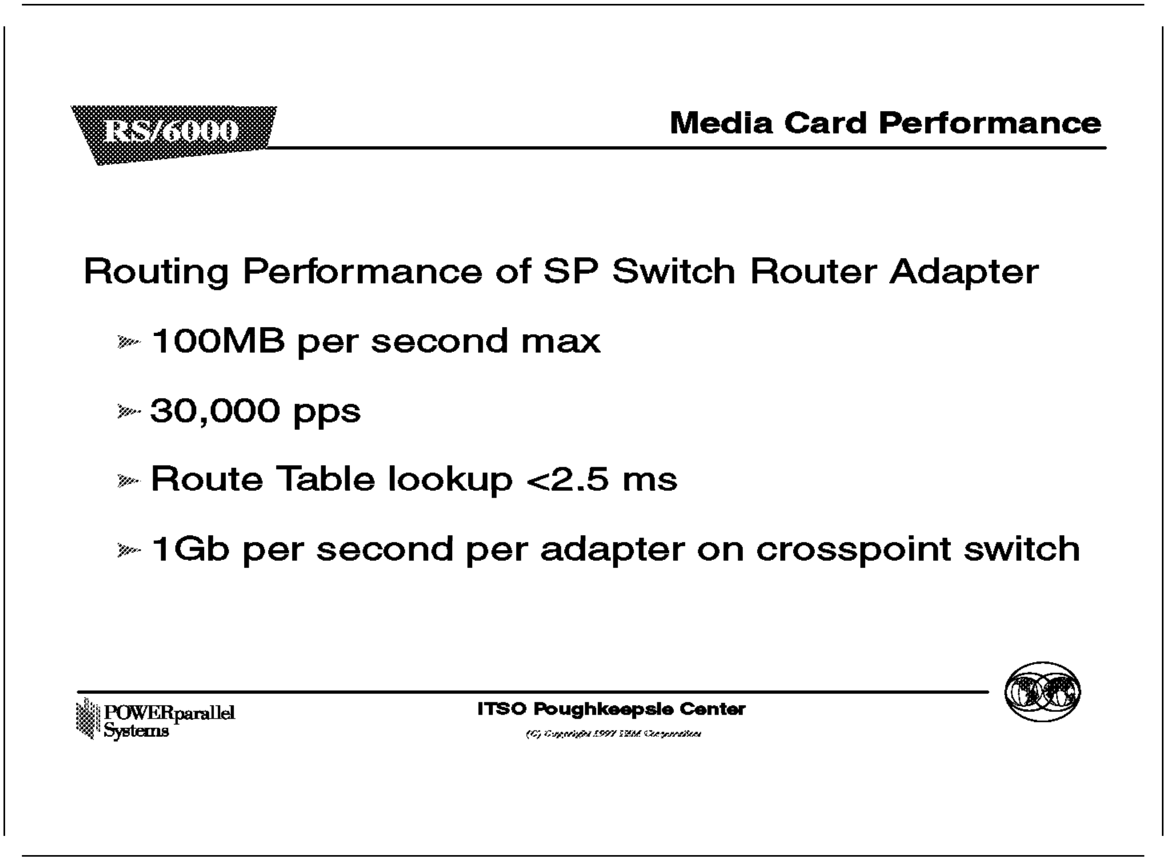
| LED | Description |
|---|---|
| PWR ON | This green LED is on when 5 volts are present. |
| 3V | This green LED is on when 3 volts are present. |
| RX HB | This green LED blinks to show the heartbeat pattern for the receive side CPU. |
| MD RCV | This amber LED turns on when data is received from its media port (RS/6000 SP Switch). |
| SW XMIT | This amber LED turns on when data is sent to the Crosspoint Switch (through the serial daughter card). |
| TX HB | This green LED blinks to show the heartbeat pattern for the transmit side CPU. |
| MD XMIT | This amber LED turns on when data is transmitted from its media port (RS/6000 SP Switch). |
| SW RCV | This amber LED turns on when data is received from the Crosspoint Switch (through the serial daughter card). |

Table 6. SP Switch Router Adapter Media Card LEDs (cont′d)

| RX/TX ST0 (green) | RX/TX ST1 (amber) | RX/TX ERR (amber) | Description |
|---|---|---|---|
| on | on | on | STATE_0 for hardware initialization. |
| off | on | on | STATE_1 for software initialization. Port waiting for configuration parameters. |
| on | off | on | STATE_2 for configuration parameters in place. Port waiting to be connected. |
| off | off | on | STATE_3 for port is connected and link is good. The media adapter is ready to be online. |
| on | off | on | STATE_4 for port is online and running/routing. |

Table 7. SP Switch Router Adapter Media Card LEDs During Bootup

| RX/TX HB (green) | RX/TX ST0 (green) | RX/TX ST1 (amber) | RX/TX ERR (amber) | Description |
|---|---|---|---|---|
| on | on | on | on | All LEDs are lit for 0.5 seconds during reset as part of onboard diagnostics. |
| off | off | off | on | Error condition: checksum error is detected in flash memory. |
| on | off | on | off | Error condition: SRAM fails memory test. |
| on | off | off | on | During loading, HB & ST1 flash as each section of the code loads. |



RS/6000

**Media Card Performance**

**Routing Performance of SP Switch Router Adapter**

- **100MB per second max**
- **30,000 pps**
- **Route Table lookup <2.5 ms**
- **1Gb per second per adapter on crosspoint switch**

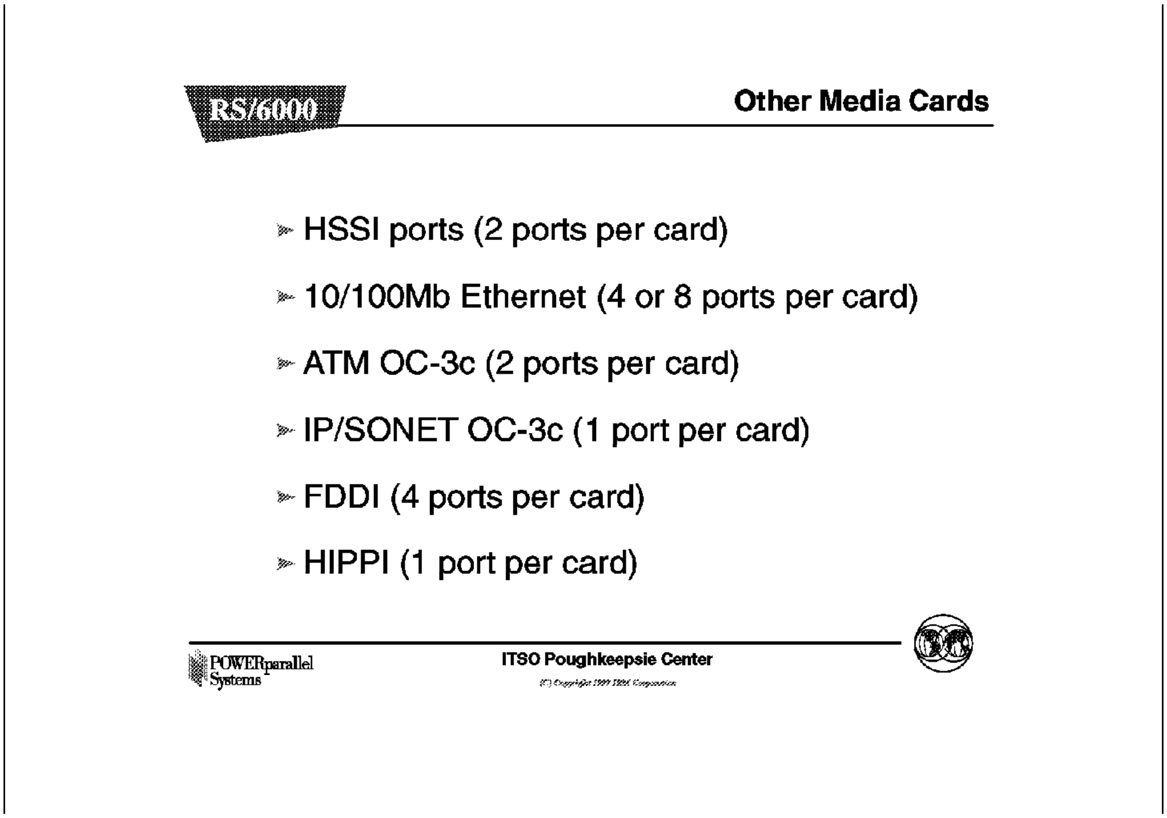
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

The SP Switch Router Adapter has the following performance characteristics:

- It is able to transfer up to 100MB per second. The limiting factor is the Crosspoint Switch connection bandwidth.
- It is able to transfer up to 30,000 packets per second. At 20,000 packets per second, each packet needs to be at 5KB in order to achieve the 100MB per second transfer rate mentioned above.
- As mentioned in the previous figure on the Characteristics of the GRF Media Card, each adapter stores its own route tables in memory. Therefore, route table lookup is very fast, that is, less than 2.5 ms.
- Finally, each media adapter has a 1Gb per second dedicated link into the Crosspoint Switch. That is why the 4-port and 16-port models have an aggregate bandwidth of 4Gb and 16Gb per second, respectively, for the Crosspoint Switch.



RS/6000

## Other Media Cards

- HSSI ports (2 ports per card)
- 10/100Mb Ethernet (4 or 8 ports per card)
- ATM OC-3c (2 ports per card)
- IP/SONET OC-3c (1 port per card)
- FDDI (4 ports per card)
- HIPPI (1 port per card)

POWERparallel Systems

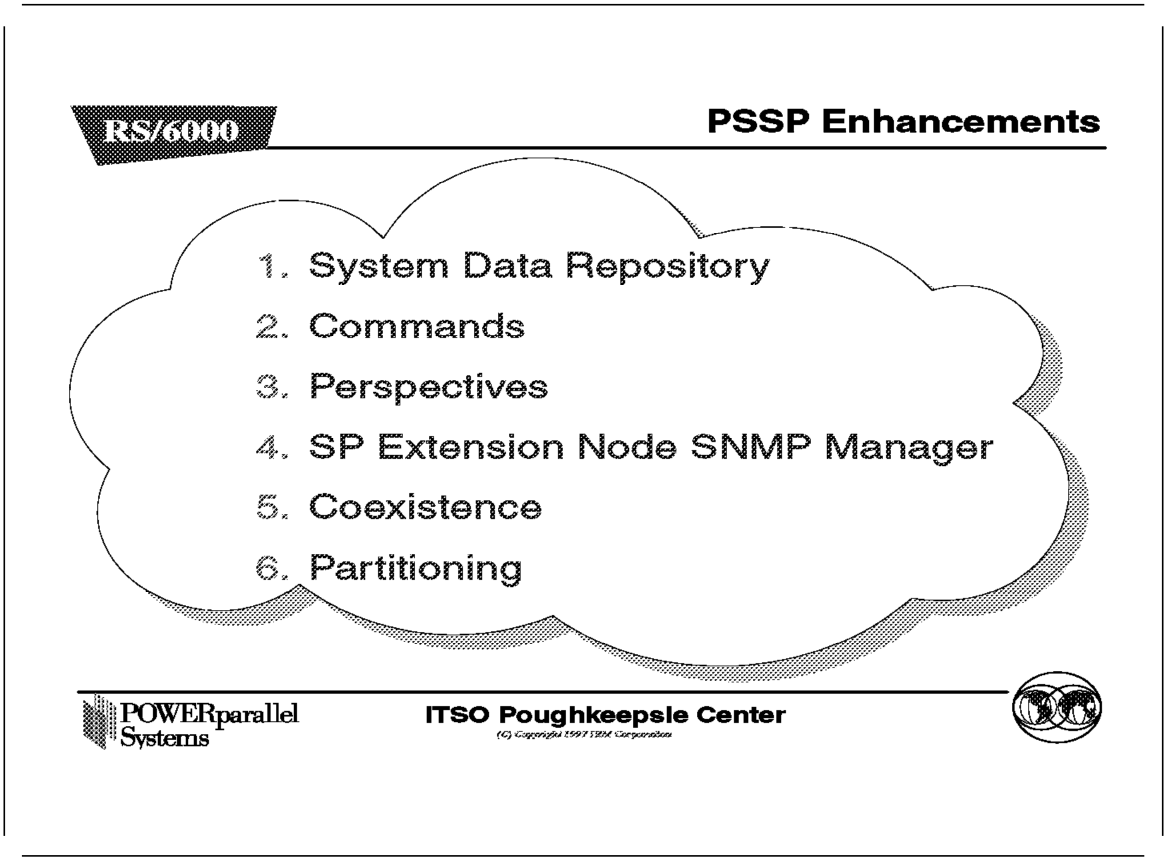
ITSO Poughkeepsie Center

The following are other media cards and adapters currently supported on the GRF:

- The High Speed Serial Interface (HSSI) is a dual-ported media adapter that can connect to two serial networks simultaneously. Each port is capable of up to 45Mb per second.
- The 10/100Mb Ethernet media adapter consists of eight 10/100BaseT Ethernet ports. All ports support only utp cables. Other types of cables require the user to supply the appropriate transceivers.
- The ATM OC-3c media adapter allows the user to connect up to two connections into the ATM network at 155Mb per second.
- The IP/SONET OC-3c is a single-ported card that allows the user to connect to a digital network using a transmission format known as Synchronous Optical Network protocol (SONET). This standard is increasingly popular in the telecommunications industry.
- The FDDI media card provides four ports in the card. These ports allow the media card to be connected into the Fiber Distributed Data Interchange (FDDI). The four ports can be configured such that they support the following:
  - Two dual-ring FDDI networks
  - One dual-ring and two single-ring FDDI networks
  - Four single-ring FDDI networks

- The HIPPI media adapter is a single-port card that allows the GRF to connect to a High Performance Parallel Interface (HIPPI) network at speeds of up to 800 or 1600Mb per second. After deducting the overhead, this medium can support connections of up to 100 Megabytes per second.

## 7.3 PSSP Enhancements



RS/6000

PSSP Enhancements

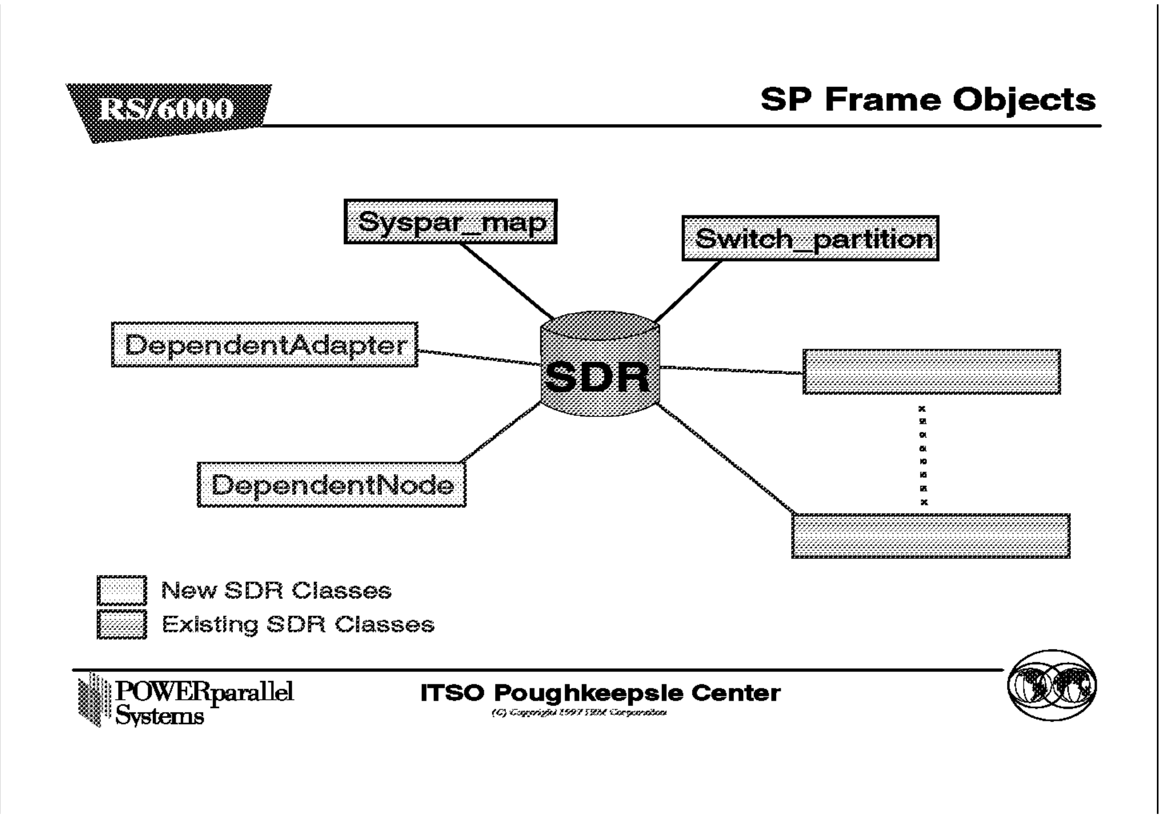
1. System Data Repository
2. Commands
3. Perspectives
4. SP Extension Node SNMP Manager
5. Coexistence
6. Partitioning

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This section discusses the enhancements made to Parallel Systems Support Programs (PSSP) to accommodate the Dependent Node Architecture.



RS/6000

SP Frame Objects

▨ New SDR Classes

▨ Existing SDR Classes

POWERparallel Systems

ITSO Poughkeepsie Center

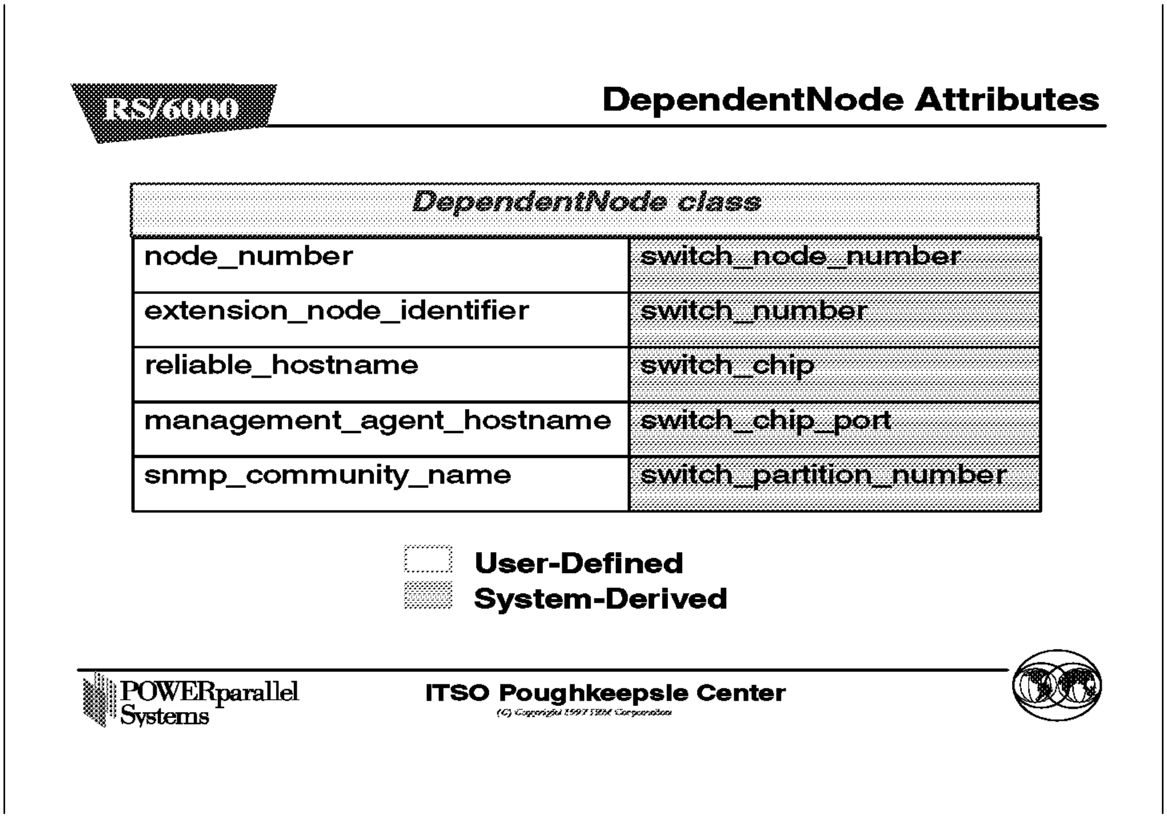
(C) Copyright 1997 IBM Corporation

The following two classes have been added to the System Data Repository (SDR):

- DependentNode
- DependentAdapter

These classes are described in detail in the next two figures.

Changes were made to the Syspar_map and Switch_partition classes, described in the Additional Attributes figure.



RS/6000

## DependentNode Attributes

| DependentNode class | |
|---|---|
| node_number | switch_node_number |
| extension_node_identifier | switch_number |
| reliable_hostname | switch_chip |
| management_agent_hostname | switch_chip_port |
| snmp_community_name | switch_partition_number |

▢ **User-Defined**
▨ **System-Derived**

POWERparallel Systems

**ITSO Poughkeepsie Center**
(C) Copyright 1997 IBM Corporation

This figure shows the attributes of the DependentNode class, described in detail below.

**Attribute** **Description**

**node_number**
User-supplied node number representing the node position of an unused SP Switch port to be used for the SP Switch Router Adapter.

**extension_node_identifier**
This is a 2-digit slot number that the SP Switch Router Adapter occupies on the GRF. Its range is from 00 to 15.

**reliable_hostname**
The hostname of the administrative Ethernet, de0, is the GRF′s hostname. Use the long version of the hostname when DNS is used.

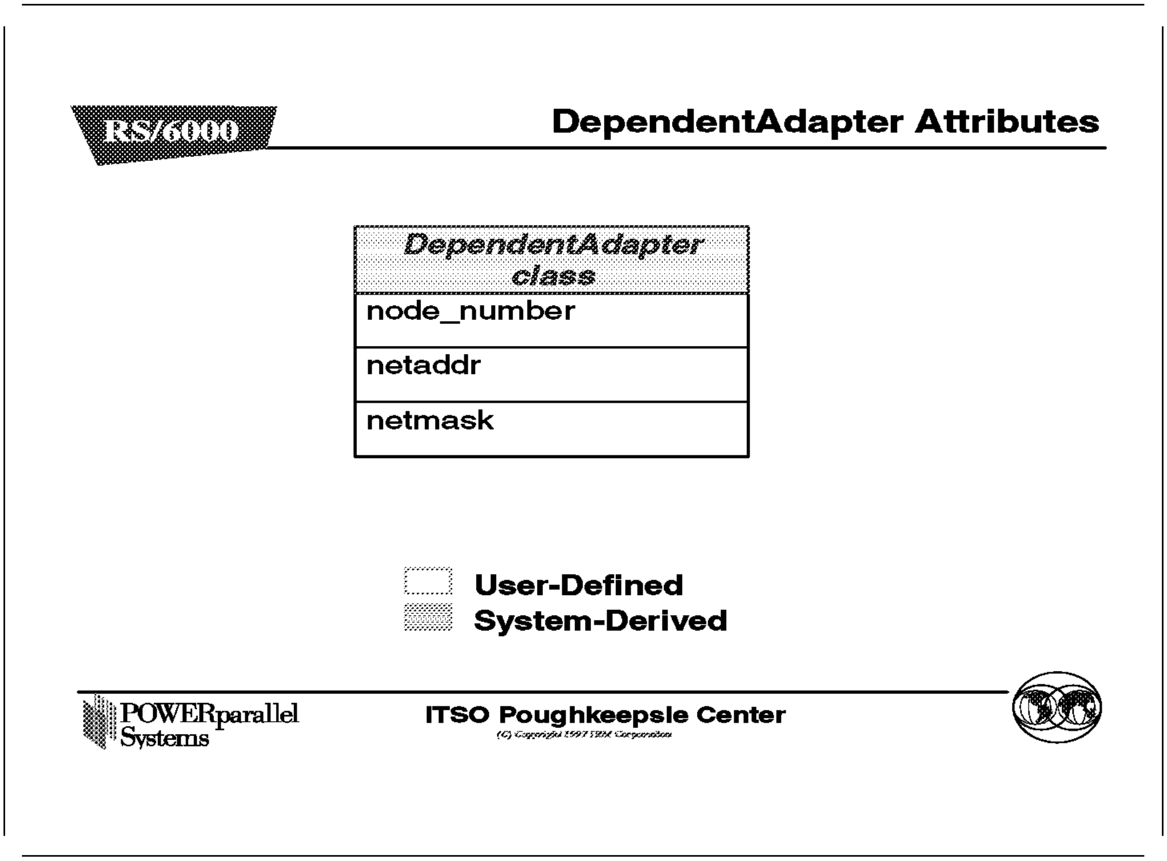
**management_agent_hostname**
This attribute is the hostname of the SNMP agent for the GRF. For the GRF dependent node, this is the same as the reliable_hostname.

**snmp_community_name**
This field contains the SNMP community name that the SP Extension Node SNMP Manager and the GRF′s SNMP Agent will send in the corresponding field of the SNMP messages. This value must match the value specified in the /etc/snmpd.conf file. If left blank, a default

name found in the SP Switch Router Adapter documentation is used.

The following attributes are derived by the RS/6000 SP system when the SDR_config routine of `endefnode` is invoked.

| **Attribute** | **Description** |
|---|---|
| **switch_node_number** | The switch port that the dependent node is attached to. |
| **switch_number** | The switch board that the dependent node is attached to. |
| **switch_chip** | The switch chip that the dependent node is attached to. |
| **switch_chip_port** | The switch chip port that the dependent node is attached to. |
| **switch_partition_number** | The partition number to which the dependent node belongs. |



RS/6000

## DependentAdapter Attributes

| *DependentAdapter class* |
|---|
| node_number |
| netaddr |
| netmask |

□ **User-Defined**

▨ **System-Derived**

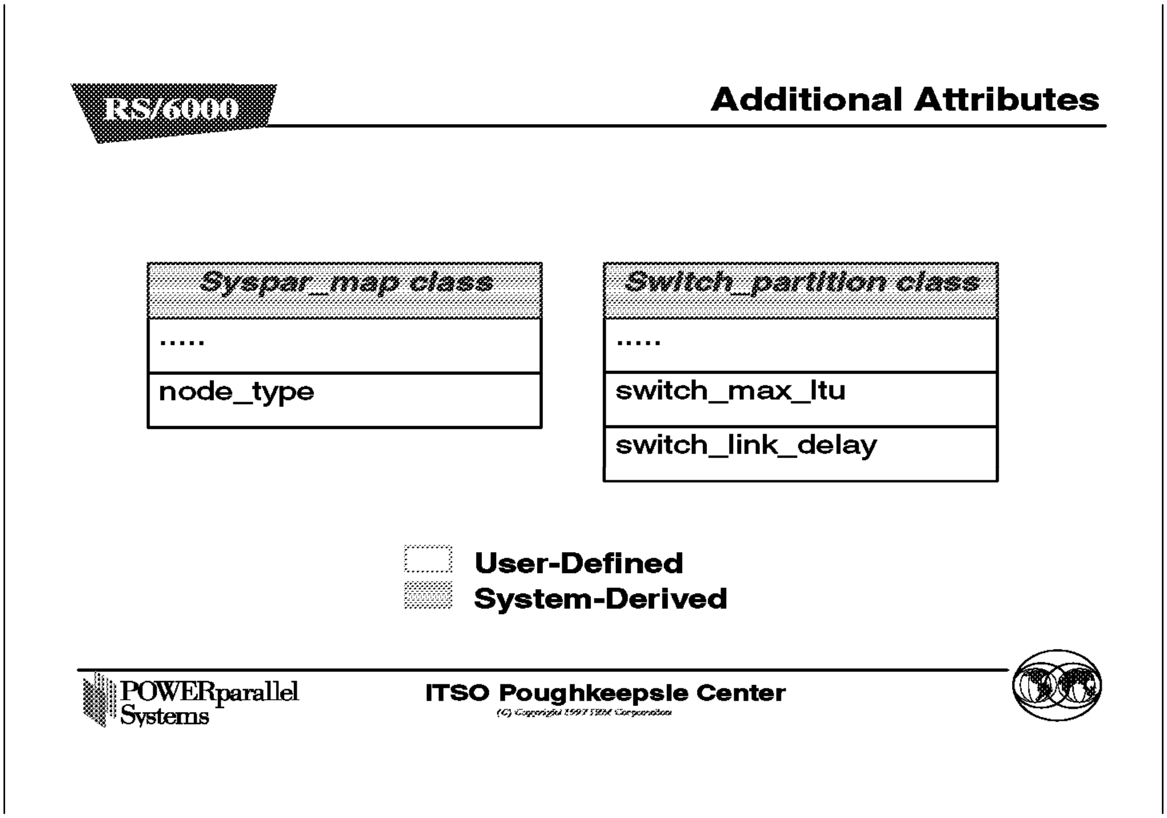
POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

This figure shows the attributes of the DependentAdapter class, described in detail below.

| **Attribute** | **Description** |
|---|---|
| **node_number** | User-supplied node number representing the node position of an unused SP Switch port to be used by the SP Switch Router Adapter. |
| **netaddr** | This is the IP address of the SP Switch Router Adapter. |
| **netmask** | This is the netmask of the SP Switch Router Adapter. |



RS/6000

## Additional Attributes

| Syspar_map class |
|---|
| ..... |
| node_type |

| Switch_partition class |
|---|
| ..... |
| switch_max_ltu |
| switch_link_delay |

□ **User-Defined**
▨ **System-Derived**

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

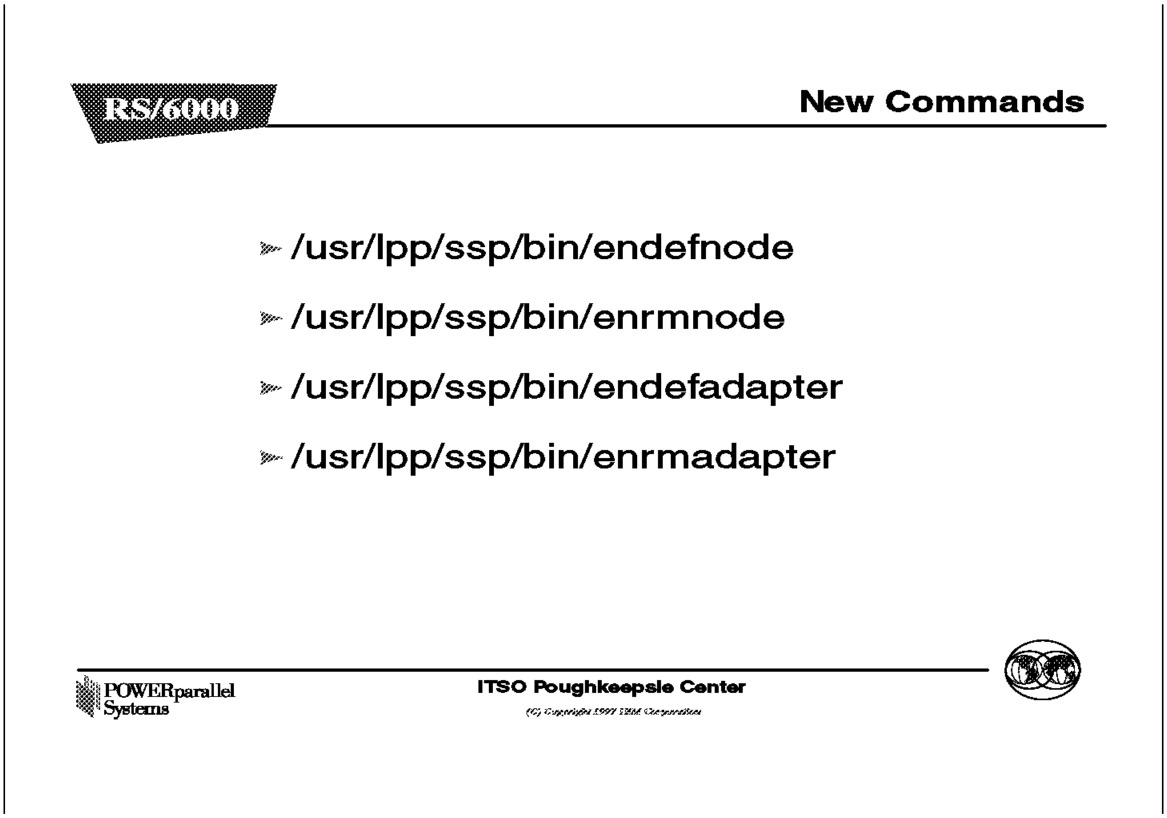
This figure shows the additional attributes of the Syspar_map and Switch_partition classes, described in detail below.

**Attribute** **Description**

**node_type** This attribute is set to `dependent` for GRF and `standard` for all other RS/6000 SP nodes.

**switch_max_ltu** Specifies the maximum packet length of data on the SP Switch; the default is 1024. Do not change this value for any reason.

**switch_link_delay** Specifies the delay for a message to be sent between the two furthest points on the switch; the default is 31. Do not change this value for any reason.



RS/6000

## New Commands

- /usr/lpp/ssp/bin/endefnode
- /usr/lpp/ssp/bin/enrmnode
- /usr/lpp/ssp/bin/endefadapter
- /usr/lpp/ssp/bin/enrmadapter

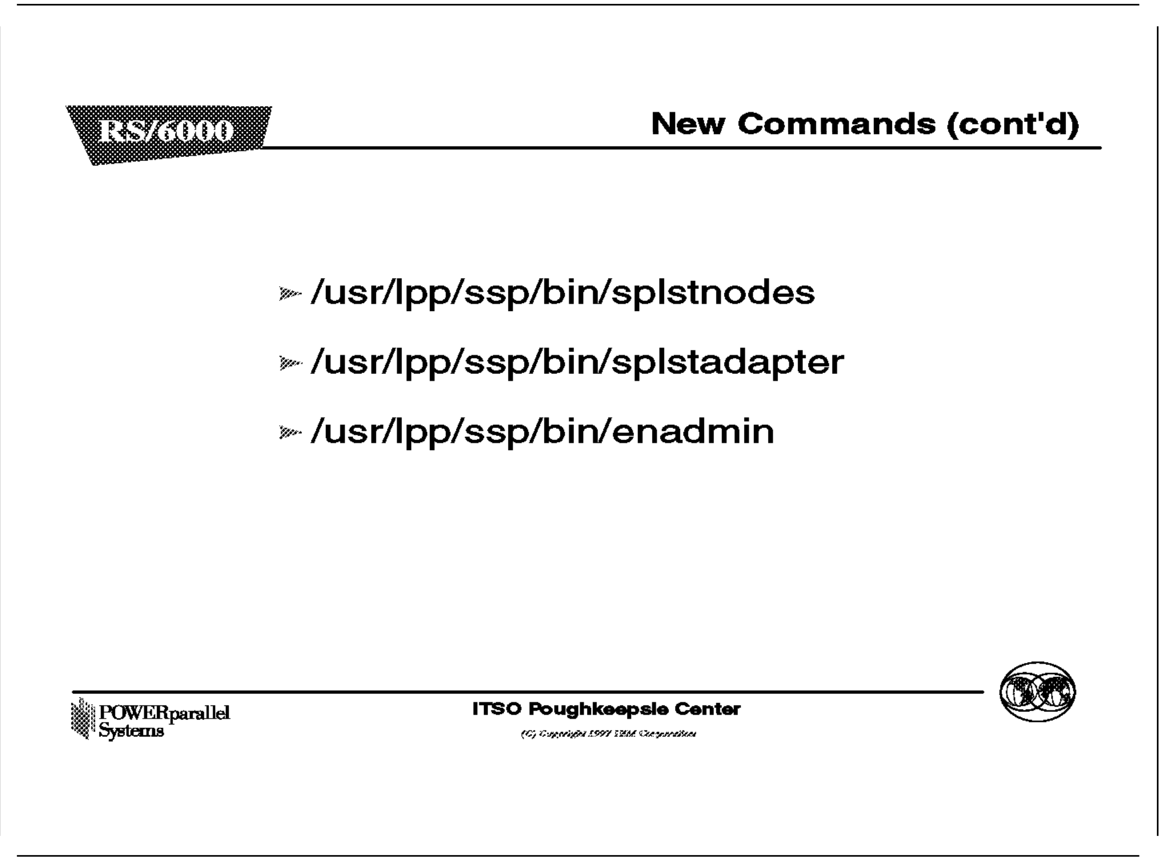
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This figure shows four new commands that were added to manage the extension node. They have the same characteristics, which are as follows:

- Part of the ssp.basic fileset
- Must only be executed on the Control Workstation
- Can only be executed by the root user
- Only affect the current active partition
- Only affect the SDR, unless the `-r` option is specified (this option is not applicable to `enrmadapter`)
- Return code of 0 if successful, 1 if failed



RS/6000

New Commands (cont'd)

- /usr/lpp/ssp/bin/splstnodes
- /usr/lpp/ssp/bin/splstadapter
- /usr/lpp/ssp/bin/enadmin

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This figure shows three more commands that were added to manage the extension node.

The first two commands, `splstnodes` and `splstadapter`, have the following characteristics:

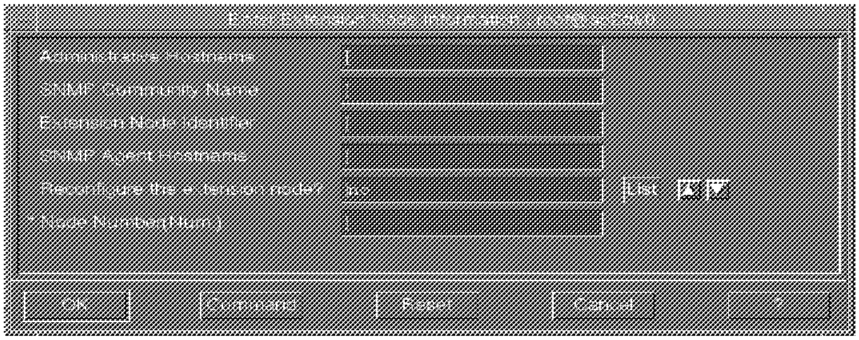
- Part of the ssp.basic fileset
- Can be executed on any standard RS/6000 SP node
- Can be executed by any user
- Will only affect the current active partition unless the `-G` option is used

The `enadmin` command is used to change the administrative state of a dependent node in the GRF; it has the following characteristics:

- Part of the ssp.spmgr fileset
- Must only be executed on the Control Workstation
- Can only be executed by the root user
- The `-r` option from `endefnode` and `endefadapter` triggers `enadmin -a reconfigure`, while the `-r` option from `enrmnode` triggers `enadmin -a reset`.
- Return code of 0 if successful, 1 if failed

RS/6000

## endefnode



Enter Extension Node Information : root@sp2cw0

Administrative Hostname

SNMP Community Name

Extension Node Identifier

SNMP Agent Hostname

Reconfigure the extension node? no

* Node Number(Num.)

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The endefnode command can be executed using smit. The fast path for smit is enter_extnode. This command is used to add or change an extension node in the SDR DependentNode class. Its options are shown in Table 8.

Table 8 (Page 1 of 2). endefnode Options

| Flag | SMIT Option | Description |
|---|---|---|
| -a | Administrative Hostname | This is the hostname of GRF, and the IP name of the GRF′s administrative Ethernet, de0. Use long names if DNS is used in the network. |
| -c | SNMP Community Name | This field contains the SNMP community name that the SP Extension Node SNMP Manager and the GRF′s SNMP Agent will send in the corresponding field of the SNMP messages. This value must match the value specified in the /etc/snmpd.conf file on the GRF. If left blank, a default name found in the SP Switch Router Adapter documentation is used. |
| -i | Extension Node Identifier | This field contains the two-digit slot number of the SP Switch Router Adapter on the GRF. The value for this field is from 00-15 and is shown on the slots of the GRF. |
| -s | SNMP Agent Hostname | This field refers to the hostname of the processor running the SNMP Agent for the GRF. In the current version of the GRF, this value is equivalent to that of the Administrative Hostname. |

Table 8 (Page 2 of 2). endefnode Options

| Flag | SMIT Option | Description |
|---|---|---|
| -r | Reconfigure the extension node | This field specifies whether the enadmin command is to be activated after the endefnode command completes. It is placed here so that the user does not have to explicitly issue the enadmin command. If the specification is yes, the -r option is part of the command. If the specification is no, the -r option is not part of the command. |
| | Node Number | This is the node number the extension node logically occupies in the RS/6000 SP. |

This command adds attribute information for the extension node. The endefadapter command adds IP information, such as IP address and netmask for the extension node. Together, these two commands define the extension node.

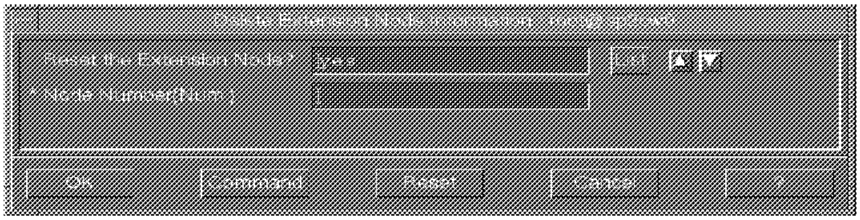
**Attention**

Please note that this command only affects the SDR, unless the -r option is used. The -r option should be issued only if endefadapter has been executed for the extension node.

When the GRF is properly configured and powered on, with the SP Switch Router Adapter inside, it periodically polls the Control Workstation for configuration data. The -r option or enadmin command is not required to activate the polling here.

RS/6000

## enrmnode



Delete Extension Node Information

Reset the Extension Node? yes

* Node Number(Num.)

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The enrmnode command is used to remove an extension node from the SDR DependentNode class and can be executed using smit. The fast path for smit is delete_extnode.

Table 9. enrmnode Options

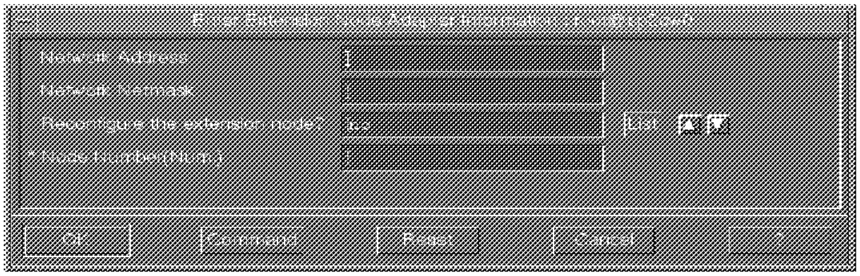
| Flag | SMIT Option | Description |
|---|---|---|
| -r | Reset the extension node | Specifies whether the enadmin command is to be activated after the enrmnode command completes. With this option the user does not have to explicitly issue the enadmin command. If the specification is yes, the -r option is part of the command. If the specification is no, the -r option is not part of the command. |
| | Node Number | This is the node number the extension node logically occupies in the RS/6000 SP. |

**Attention**

- Please note that this command only affects the SDR unless the -r option is used.
- This command should be issued with a -r flag, because the enadmin command is not available for the extension node after enrmnode is executed, since the extension node has been removed from the SDR.

RS/6000

## endefadapter



Enter Extension Node Adapter Information [illegible]

Network Address

Network Netmask

Reconfigure the extension node? no

* Node Number(Num.)

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The endefadapter command is used to add or change the extension node adapter IP information in the SDR DependentAdapter object, and can be executed using smit. The fast path for smit is enter_extadapter.

Table 10. endefadapter Options

| Flags | SMIT Option | Description |
|---|---|---|
| -a | Network Address | Specifies the IP address of the extension node. |
| -m | Network Netmask | Specifies the netmask for the extension node. |
| -r | Reconfigure the extension node | Specifies if the enadmin command is to be activated after the endefadapter command completes. With this option, the user does not have to explicitly issue the enadmin command. If the specification is yes, the -r option is part of the command. If the specification is no, the -r option is not part of the command. |
|  | Node Number | This is the node number the extension node logically occupies in the RS/6000 SP. |

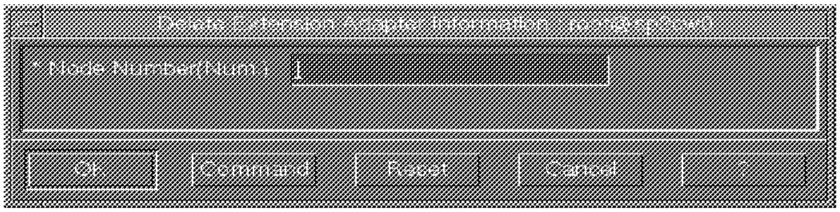
**Attention**

Please note that this command only affects the SDR unless the `-r` option is issued. The `-r` option should be issued only if the `endefnode` has been executed for the extension node.

When the GRF is properly configured and powered on, with the SP Switch Router Adapter inside, it periodically polls the Control Workstation for configuration data. The `-r` option or `enadmin` command is not required to activate the polling here.

RS/6000

## enrmadapter


Delete Extension Adapter Information : root@sp2cw0

* Node Number(Num.) 1

POWERparallel Systems

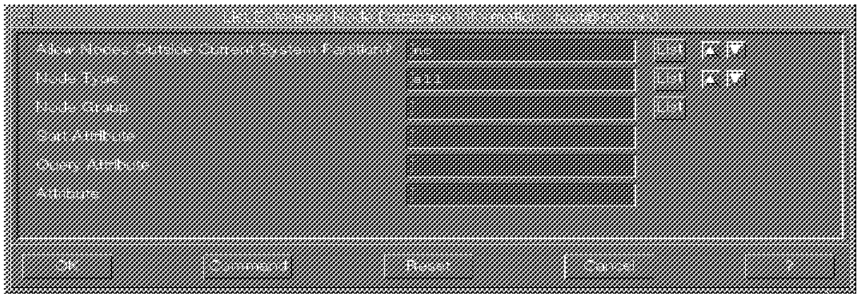
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The enrmadapter command is used to remove the SDR DependentAdapter object, and can be executed using smit. The fast path for smit is delete_extadapter.

RS/6000

## splstnodes



List Extension Node Database Information : [illegible]

Allow Nodes Outside Current System Partition? no

Node Type all

Node Group

Sort Attribute

Query Attribute

Attribute

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The splstnodes command is used to list the node attributes of all nodes in the SDR, and can be executed using smit. The fast path for smit is list_extnode.

Table 11 (Page 1 of 2). splstnodes Options

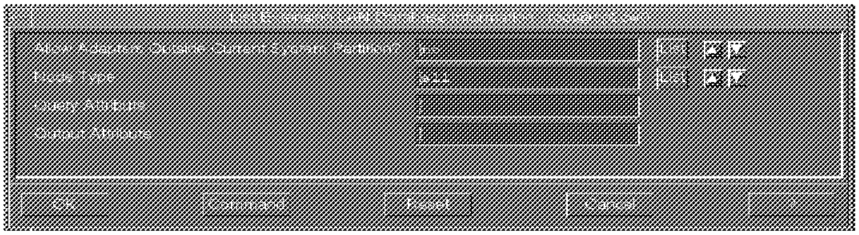
| Flags | Description |
|---|---|
| -h | Outputs usage information. |
| -G | Ignores partition boundaries for its output. |
| -x. | Inhibits header record in output. |
| -d <delimiter> | Uses the <delimiter> between its attributes in the output. |
| -p <string> | Uses the <string> value in the output in place of an attribute that has no value. |
| -s <attr> | Sorts the output using the <attr> value. In SMIT, this field is known as Sort Attribute. |
| -t <node-type> | Uses standard to list RS/6000 SP nodes, or dependent. If none is specified, it displays both. In SMIT, this field is known as Node Type. |
| -N <node_grp> | Restricts the query to the nodes belonging to the node group specified in <node_grp>. If the <node_grp> specified is a system node group, the -G flag is implied. |
| <attr==value> | This operand is used to filter the output, such that only nodes with attributes that are equivalent to the value specified are displayed. In SMIT, this field is known as Query Attribute. |

Table 11 (Page 2 of 2). splstnodes Options

| Flags | Description |
|---|---|
| <attr> | This is a list containing attributes that are displayed by the command. If none is specified, it defaults to node number. This list of attributes can be found in the DependentNode class. In SMIT, this field is known as Attribute. |

RS/6000

## splstadapters



List Extension LAN Database Information

Allow Adapters Outside Current System Partition? no

Node Type all

Query Attribute

Output Attribute

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

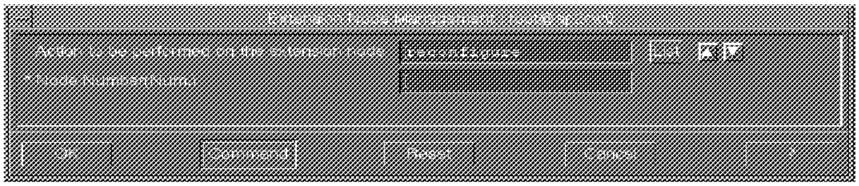
The splstadapers command is used to list the adapter attributes of all nodes in the SDR, and can be executed using smit. The fast path for smit is list_extadapter.

Table 12. splstadapters Options

| Flags | Description |
|---|---|
| -h | Outputs usage information. |
| -G | Ignores partition boundaries for its output. |
| -x. | Inhibits header record in output. |
| -d <delimiter> | Uses the <delimiter> between its attributes in the output. |
| -p <string> | Uses the <string> value in the output in place of an attribute that has no value. |
| -t <node-type> | Uses standard to list RS/6000 SP nodes, or dependent. If none is specified, it displays both. In SMIT, this field is known as Node Type. |
| <attr==value> | This operand is used to filter the output, such that only nodes with attributes that are equivalent to the value specified are displayed. In SMIT, this field is known as Query Attribute. |
| <attr> | This is a list containing attributes that are displayed by the command. If none is specified, it defaults to node number. This list of attributes can be found in the Adapter and DependentAdapter class. In SMIT, this field is known as Output Attribute. |

RS/6000

## enadmin



Extension Node Management

Action to be performed on the extension node: reconfigure

* Node Number(Num.)

POWERparallel Systems

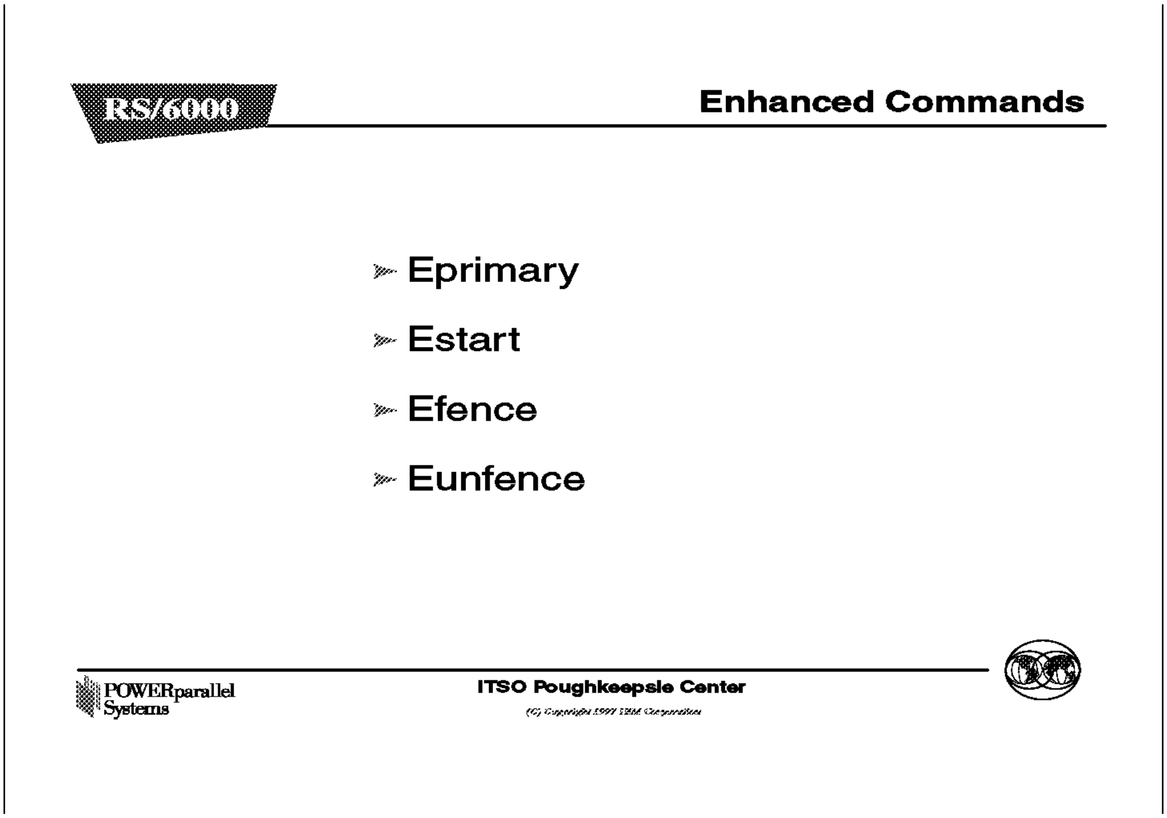
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The enadmin command is used to change the status of the SP Switch Router Adapter in the GRF, and can be executed using smit. The fast path for smit is manage_extnode.

Table 13. enadmin Options

| Flags | SMIT Option | Description |
|---|---|---|
| -a | Actions to be performed on the extension node. | Either reset or reconfigure. A reset is sent to the extension node SNMP Agent to change the target node to a down state (not active on the SP Switch). A reconfigure is sent to the extension node SNMP Agent to trigger reconfiguration of the target node, which causes the SNMP Agent to request new configuration parameters from the SP Extension Node SNMP Manager, and to reconfigure the target node when the new parameters are received. A more detailed explanation of this is found in the SNMP Flow figure in the Installation section. |
|  | Node Number | This is the node number the extension node logically occupies in the RS/6000 SP. |



RS/6000

## Enhanced Commands

- Eprimary
- Estart
- Efence
- Eunfence

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The following commands have been modified due to the introduction of the dependent node:

- `Eprimary`

  This command has been modified so that dependent nodes will not be able to act as a Primary or Primary Backup node for the SP Switch in the partition. The dependent node does not run the RS/6000 SP Switch codes like standard RS/6000 SP nodes and, therefore, does not have the ability to act as the Primary or Primary Backup node.

- `Estart`

  This command functions as usual with the dependent node in the RS/6000 SP.

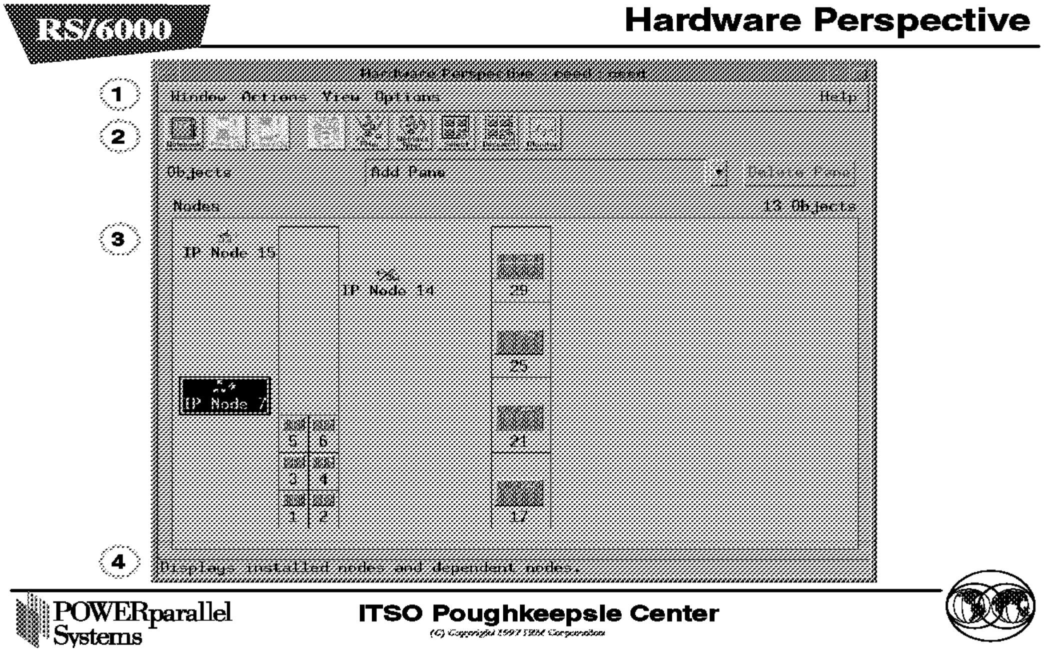
- `Efence`

  This command functions as usual with the dependent node in the RS/6000 SP. In addition, the dependent node can be fenced from the SP Switch with autojoin like any other standard RS/6000 SP node.

- `Eunfence`

  This command functions as usual with the dependent node in the RS/6000 SP. In addition, the dependent node can rejoin the SP Switch network with this command, if that node was previously removed from the switch network due to failures or `Efence`.

## Hardware Perspective



Displays installed nodes and dependent nodes.

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

IP Node is used in Perspectives as a convenient and short descriptive term easily displayed in the GUI that conveys the role and functions of the dependent node. Currently, this is the only dependent node.

In the following figures, we show the changes made to Perspectives because of the introduction of the IP Node. The changes are restricted to the Hardware and System Partition Aid Perspectives.

This figure shows the Hardware Perspectives, which can be started using the command `perspectives` and selecting the **Hardware** icon. Alternatively, it can be started directly via the command `sphardware`.

The Hardware Perspective consists of the following four parts:

1. Menu bar
2. Toolbar
3. Nodes pane (Frame or Icon View)
4. Information area

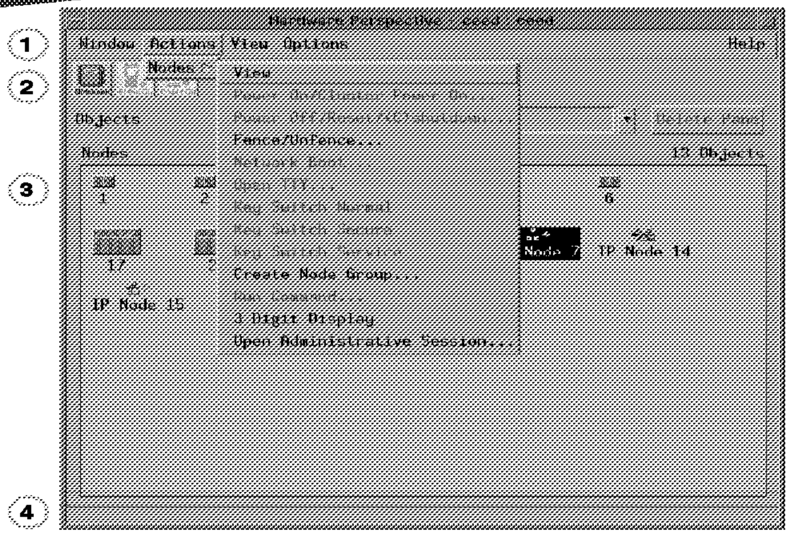
The most obvious change is the addition of the IP Node icon as seen in the Nodes pane. (The figure above shows the Frame View.) The default label for this icon is IP Node <node number>.

The IP Node icon is also located on the side of the frame, where a standard node with that node number would be. In this figure the IP Nodes are 7, 14 and 15.

When switch_responds is monitored, it shows the IP Node in two states: green when working with the SP Switch; marked with a red cross when fenced or not operating due to hardware or configuration problems. In the figure, IP Node 7 and 15 are working, while IP Node 14 is down.

RS/6000

## Action Menu



POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In this figure, we see that IP Node 7 is selected in the Nodes pane, and **Actions→Nodes** as selected in the menu bar (1). We see that only the following five actions are available:

- View

  This will bring up the IP Node′s hardware notebook, shown in the next figure.

- Fence/Unfence...

  This will bring up another window to allow us to either fence or unfence an IP Node. If we are fencing the IP Node, we can use the option of autojoin.

- Create Node Group...

  This will bring up another window to allow us to add the RS/6000 SP nodes to a Node Group. This action does not affect the IP Node, even though it is selectable.

- Three-Digit Display

  This will bring up a window to show the three-digit display of all RS/6000 SP standard nodes in the current partition. This action does not apply to the IP Node, even though it is selectable.

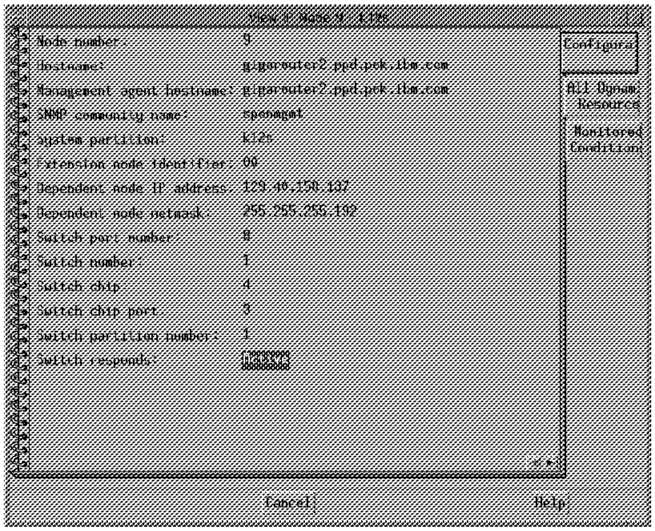
- Open Administration Session...

  This action will open a window that is a Telnet session to the GRF, using the reliable_hostname attribute specified in the DependentNode class.

In addition, the Nodes pane in this figure shows the Icon View. In this view, the IP Node icons are always located after all the standard RS/6000 SP node icons. The effects of monitoring the IP Nodes and the icon labels are the same as those of Frame View, mentioned in the previous figure.

RS/6000

## Hardware Notebook



Node number: 9
Hostname: gigarouter2.ppd.pok.ibm.com
Management agent hostname: gigarouter2.ppd.pok.ibm.com
SNMP community name: spenmgmt
System partition: k12s
Extension node identifier: 00
Dependent node IP address: 129.40.150.137
Dependent node netmask: 255.255.255.192
Switch port number: 8
Switch number: 1
Switch chip: 4
Switch chip port: 3
Switch partition number: 1
Switch responds: Active

POWERparallel Systems

ITSO Poughkeepsie Center

This figure shows the IP Node hardware notebook. This notebook can be triggered by selecting the **Notebook** icon on the Hardware Perspective toolbar (2), or selecting **Action**→**Nodes**→**View** in the menu bar (1).

The notebook has three tabs: Configuration, All Dynamic Resource Variables, and Monitored Conditions. This figure shows the Configuration tab.

These are the attributes listed in the Configuration tab:

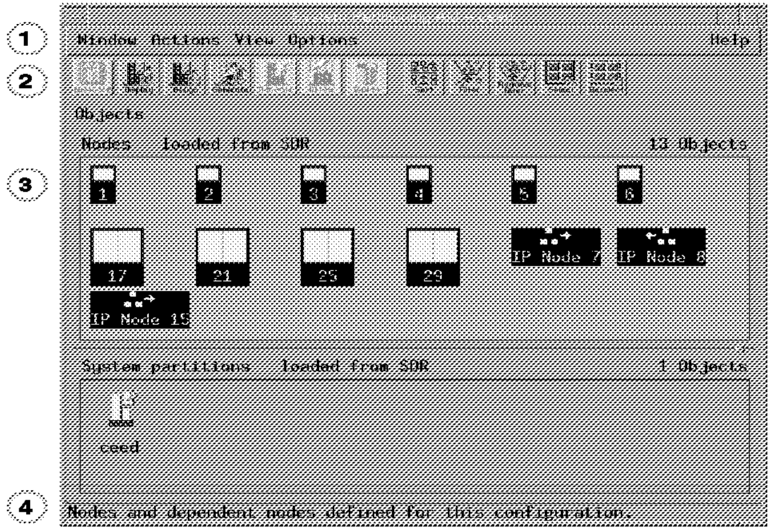
- Node number
- Hostname
- Management agent hostname
- SNMP community name
- System partition
- Extension node identifier
- Dependent node IP address
- Dependent node netmask
- Switch port number
- Switch number
- Switch chip
- Switch chip port

- Switch partition number
- Switch responds

The All Dynamic Resource Variables tab only shows the state of the *Switch Responds*, and the Monitored Conditions tab only shows the value of the *Switch Responds* if it is being monitored.

RS/6000

## System Partition Aid Perspective



Nodes and dependent nodes defined for this configuration.

POWERparallel Systems

ITSO Poughkeepsie Center

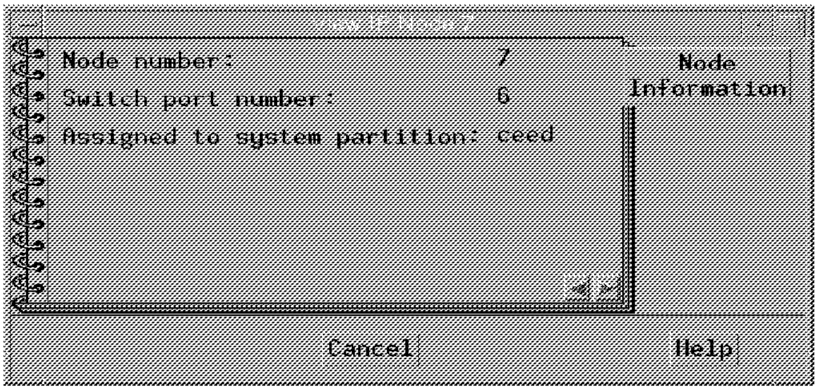
(C) Copyright 1997 IBM Corporation

The System Partition Aid Perspective window has two panes, the Nodes pane and the System partitions pane. The Nodes pane (3) in the figure above shows the Icon view. Notice that the IP Nodes are displayed after all the standard RS/6000 SP nodes. Also, the node numbers of the IP Nodes are listed below their icons.

The IP Nodes can only be assigned to a partition here. This is done either by using the **Assign** icon in the toolbar (2), or by selecting **Action→Nodes→Assign Nodes to System Partition** on the menu bar (1). Except for the System Partition Notebook, discussed in the next figure, all other actions, though selectable, do not apply to the IP Node.

RS/6000

## System Partition Aid Notebook



Node number: 7
Switch port number: 6
Assigned to system partition: ceed

POWERparallel Systems

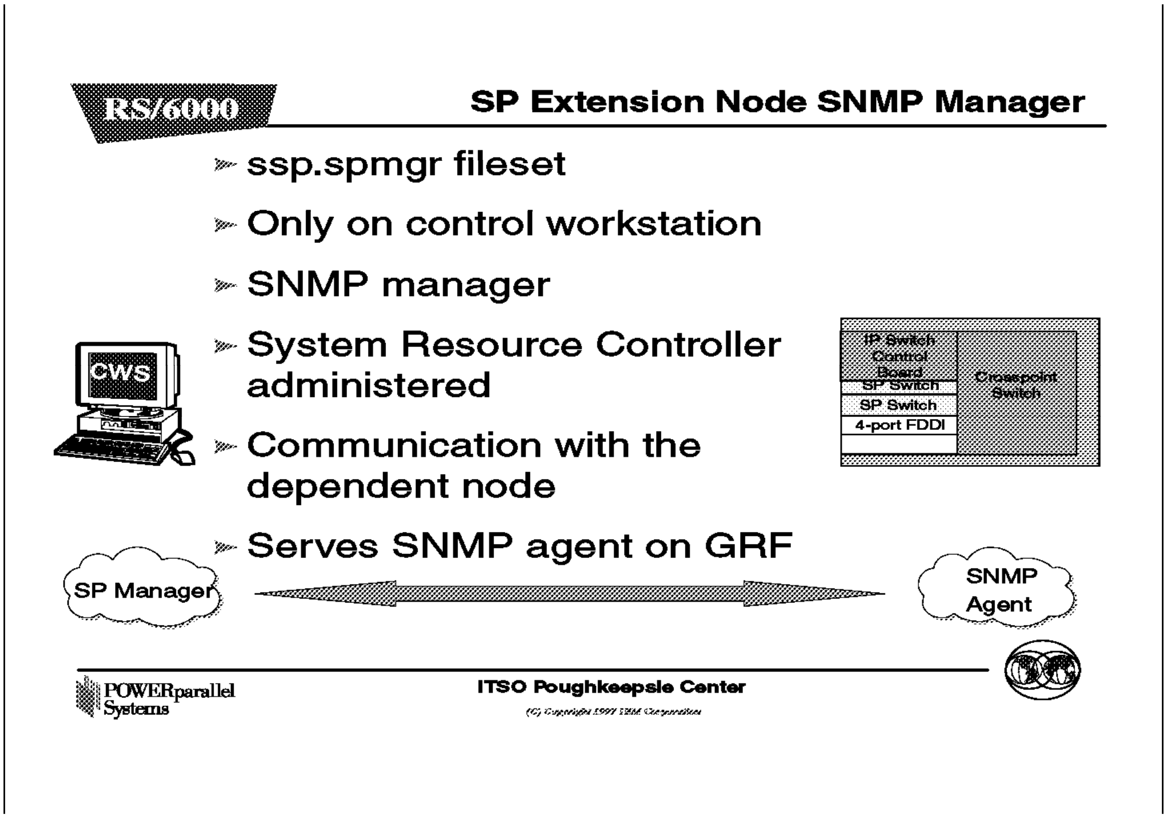
ITSO Poughkeepsie Center

This figure shows the IP Node System Partition Notebook. This notebook can be triggered by selecting the **Notebook** icon on the Hardware Perspective toolbar (2), or selecting **Action**→**Nodes**→**View** on the menu bar (1).

The notebook only has the Node Information tab shown in the figure above.

These attributes are listed in the Node Information tab.

- Node number
- Switch port number
- Assigned to system partition



RS/6000

SP Extension Node SNMP Manager

- ssp.spmgr fileset
- Only on control workstation
- SNMP manager
- System Resource Controller administered
- Communication with the dependent node
- Serves SNMP agent on GRF

CWS
IP Switch Control Board
SP Switch
SP Switch
4-port FDDI
Crosspoint Switch
SP Manager
SNMP Agent

POWERparallel Systems

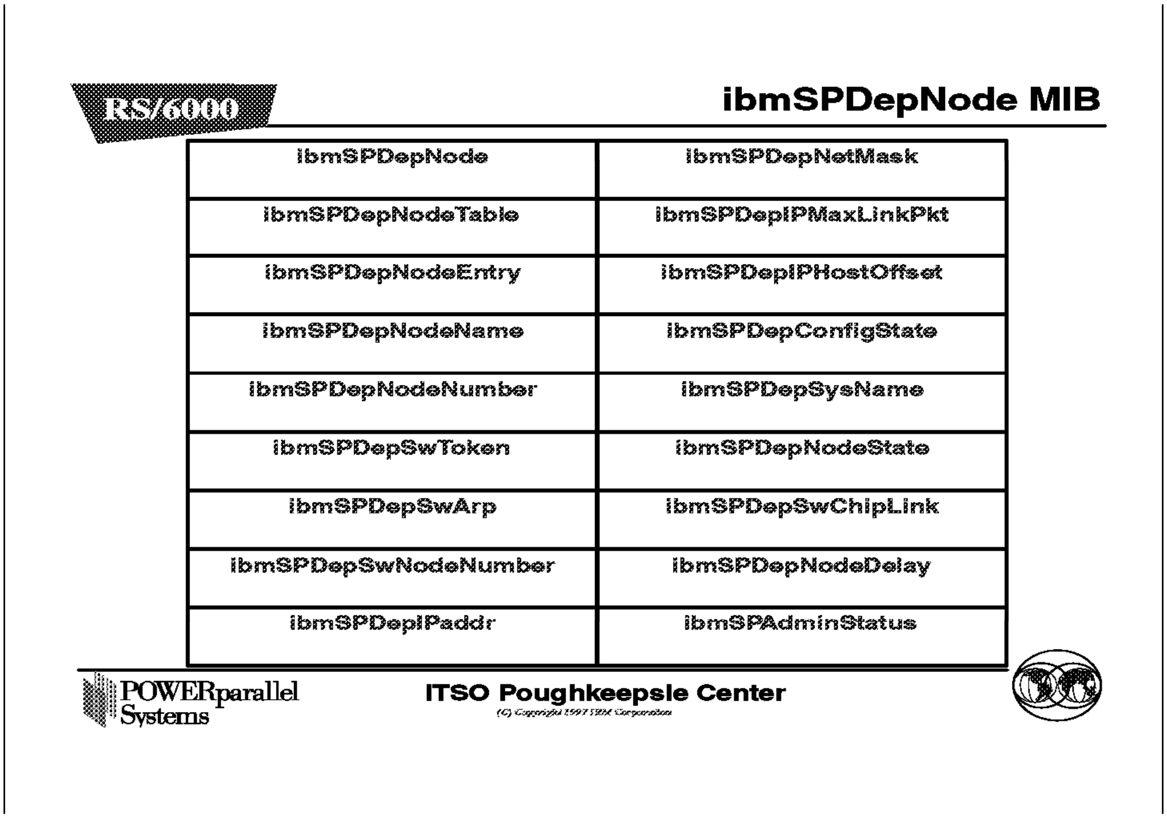
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The SP Extension Node SNMP Manager is contained in the ssp.spmgr fileset of PSSP. This fileset must be installed on the Control Workstation in order for the GRF to function as an extension node.

The SP Extension Node SNMP Manager is an SNMP manager administered by the System Resource Controller. The purpose of the SNMP manager is to communicate with the SNMP agent on the GRF. The SNMP Manager and the Agent adhere to Version 1 of the SNMP protocol. The SNMP Manager sends configuration data for an extension node to the SNMP agent on the GRF. The SNMP agent applies the configuration data to the SP Switch Router Adapter represented by the extension node. The SNMP agent also sends asynchronous notifications in the form of SNMP traps to the SNMP Manager when the extension node changes state. The following commands are available to control the SP Extension Node SNMP Manager:

- `startsrc`
- `stopsrc`
- `lssrc`
- `traceson`
- `tracesoff`



RS/6000

## ibmSPDepNode MIB

| | |
|---|---|
| ibmSPDepNode | ibmSPDepNetMask |
| ibmSPDepNodeTable | ibmSPDepIPMaxLinkPkt |
| ibmSPDepNodeEntry | ibmSPDepIPHostOffset |
| ibmSPDepNodeName | ibmSPDepConfigState |
| ibmSPDepNodeNumber | ibmSPDepSysName |
| ibmSPDepSwToken | ibmSPDepNodeState |
| ibmSPDepSwArp | ibmSPDepSwChipLink |
| ibmSPDepSwNodeNumber | ibmSPDepNodeDelay |
| ibmSPDepIPaddr | ibmSPAdminStatus |

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

IBM has defined a dependent node SNMP Management Information Base (MIB) ibmSPDepNode. This MIB contains definitions of objects representing configuration attributes of each dependent node and its state. The GRF Agent maintains the state and configuration data for each dependent node using the MIB as a conceptual database.

The MIB defines a single table of up to 16 entries representing the adapter slots in the GRF. When a slot is populated by an SP Switch Router Adapter, the entry in the table, accessed using the extension node identifier, contains the configuration attribute and state values for the adapter in the slot. Also included in the MIB are the definitions of trap messages sent by the GRF Agent to the SP Extension Node SNMP Manager. A copy of the MIB is contained in the file /usr/lpp/ssp/config/spmgrd/ibmSPDepNode.my on the Control Workstation.

Other SNMP managers in the network can query this MIB table to validate the configuration and status of the dependent node and GRF. However, only an SNMP manager using the correct SNMP community name can change the values in the MIB table.

Below is a listing of its entries.

**Entry** **Definition**

**ibmSPDepNode** Object identifier for the dependent node in the MIB database.

**ibmSPDepNodeTable** Table of entries for dependent nodes.

**ibmSPDepNodeEntry** A list of objects comprising a row and a clause in the ibmSPDepNodeTable. The clause indicates which object is used as an index into the table to obtain a table entry.

**ibmSPDepNodeName** The extension_node_identifier attribute in the DependentNode class.

**ibmSPDepNodeNumber** The node_number attribute in the DependentNode class.

**ibmSPDepSwToken** A combination of switch_number, switch_chip and switch_chip_port attributes from the DependentNode class.

**ibmSPDepSwArp** The arp_enabled attribute in the Switch_partition class.

**ibmSPDepSwNodeNumber** The switch_node_number attribute in the DependentNode class.

**ibmSPDepIPaddr** The netaddr attribute in the DependentAdapter class.

**ibmSPDepNetMask** The netmask attribute in the DependentAdapter class.

**ibmSPDepIPMaxLinkPkt** The switch_max_ltu attribute in the Switch_partition class.

**ibmSPDepIPHostOffset** This attribute stores the difference between the host portion of a node′s IP address and its corresponding switch node number. When ARP is disabled on the SP Switch network, this offset is subtracted from the host portion of IP address to calculate the switch node number.

**ibmSPDepConfigState** The six config states of the dependent node are: notConfigured, firmwareLoadFailed, driverLoadFailed, diagnosticFailed, microcodeLoadFailed, and fullyConfigured, for use in configuring the adapter.

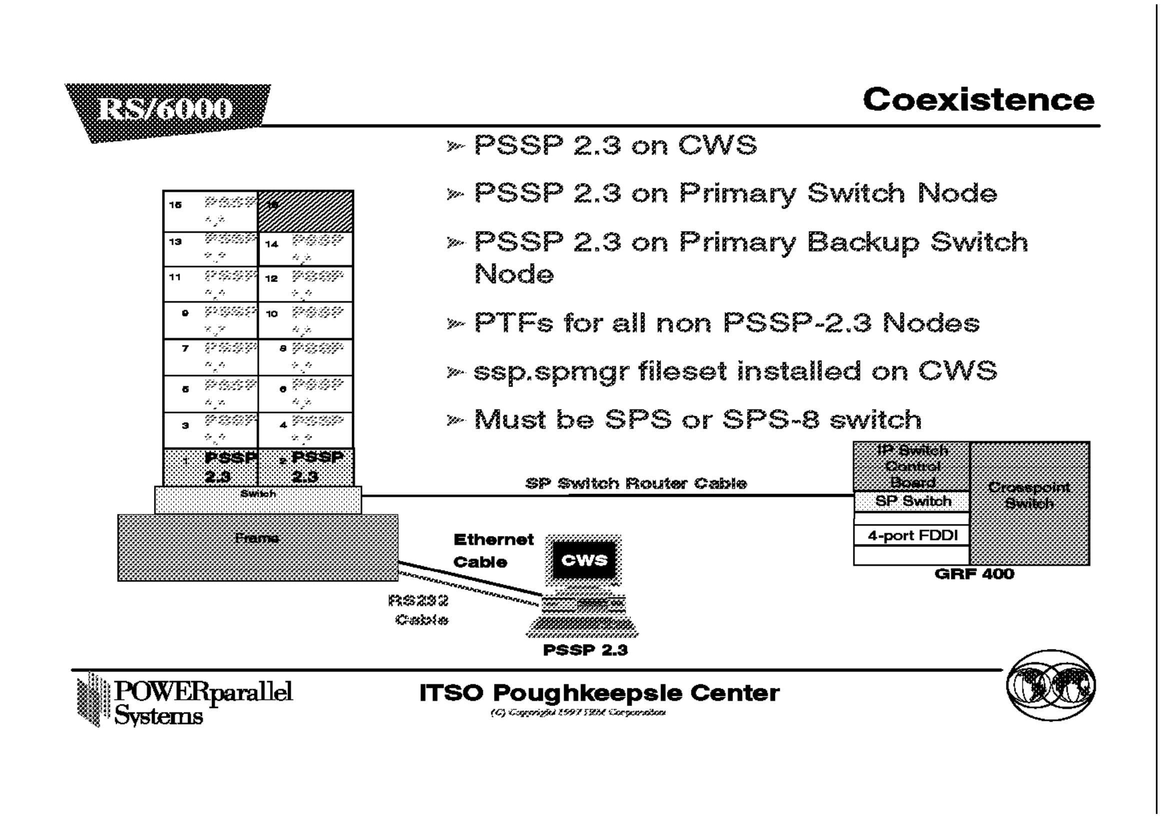
**ibmSPDepSysName** The syspar_name attribute in the Syspar class.

**ibmSPDepNodeState** The value of nodeUp or nodeDown, to show the status of the dependent node.

**ibmSPDepSwChipLink** The switch_chip_port attribute in the DependentNode class.

**ibmSPDepNodeDelay** The switch_link_delay attribute in the Switch_partition class.

**ibmSPDepAdminState** The value of up, down, or reconfigure, indicating the desired state of the dependent node. If the dependent node is not in its desired state, the SNMP agent on the GRF will trigger the appropriate action to change its state.



RS/6000

## Coexistence

- PSSP 2.3 on CWS
- PSSP 2.3 on Primary Switch Node
- PSSP 2.3 on Primary Backup Switch Node
- PTFs for all non PSSP-2.3 Nodes
- ssp.spmgr fileset installed on CWS
- Must be SPS or SPS-8 switch

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This figure shows a single-frame RS/6000 SP in a single partition with a connection to the GRF. Nodes 1 and 2 are installed with PSSP 2.3. The other nodes are installed with any other version of PSSP that can coexist with PSSP 2.3 to represent coexistence. Also, note that Node 16 is empty, because the SP Switch port for this node is used by the SP Switch Router Adapter in the GRF.

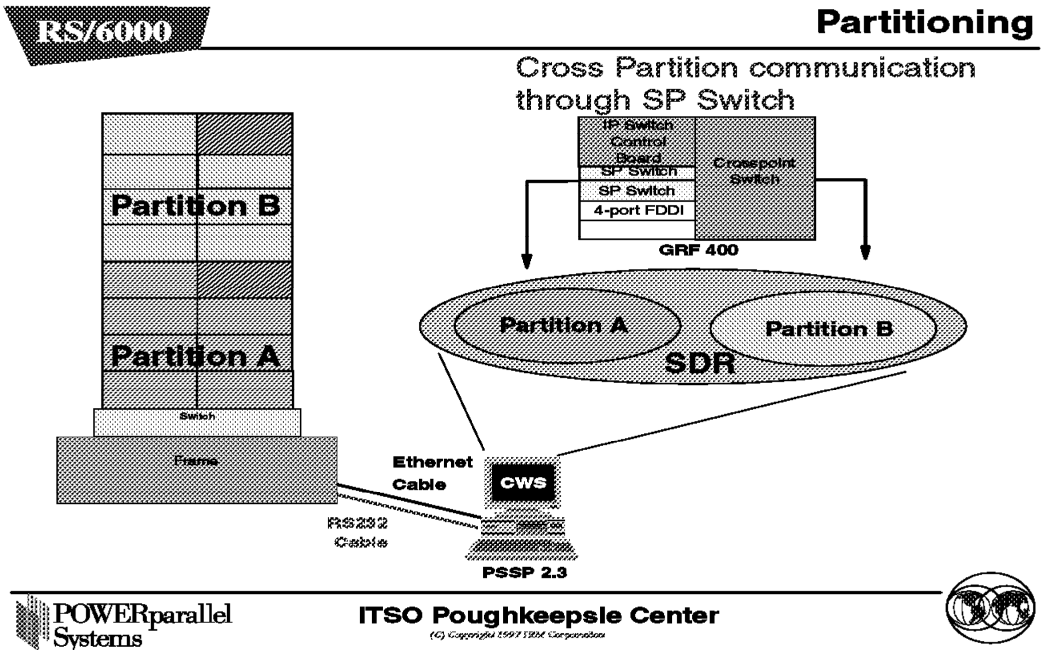
The dependent node is only supported in PSSP 2.3. To use it with non-PSSP 2.3 nodes requires the use of coexistence. The following conditions are required for the dependent node to communicate with non-PSSP 2.3 nodes using coexistence:

- The Control Workstation must be at PSSP 2.3 to manage dependent nodes.
- The Primary node of the SP Switch must be at PSSP 2.3, as the Primary node needs to perform some tasks for the dependent node, and these functions are only available in PSSP 2.3.
- The Primary Backup node of the SP Switch should be PSSP 2.3, so that if the Primary node fails, the dependent node can continue to function in the RS/6000 SP when the Backup node takes over.
- All non-PSSP 2.3 RS/6000 SP nodes in the partition need to maintain the right level of fixes (PTF) in order for coexistence with PSSP 2.3 to take place.
- The ssp.spmgr fileset must be installed on the Control Workstation.
- Because the SP Switch Router Adapter will only work with the 8-port or 16-port SP Switch, make sure that the switch used in the RS/6000 SP is not a High Performance Switch (HiPS).

- There must be at least one free SP Switch port to install the SP Switch Router Adapter.

**Important**

When the Primary Switch node fails, the Primary Backup Switch node will take over as the new Primary switch node. The new Primary Backup Switch node, selected from the current partition, can be a non-PSSP 2.3 node, even though another PSSP 2.3 node may exist in that partition. The only way to ensure that the new Backup Switch node is a PSSP 2.3 node is to manually check the RS/6000 SP system, and reset it to a PSSP 2.3 node if one exists.



## Partitioning

Cross Partition communication through SP Switch

ITSO Poughkeepsie Center

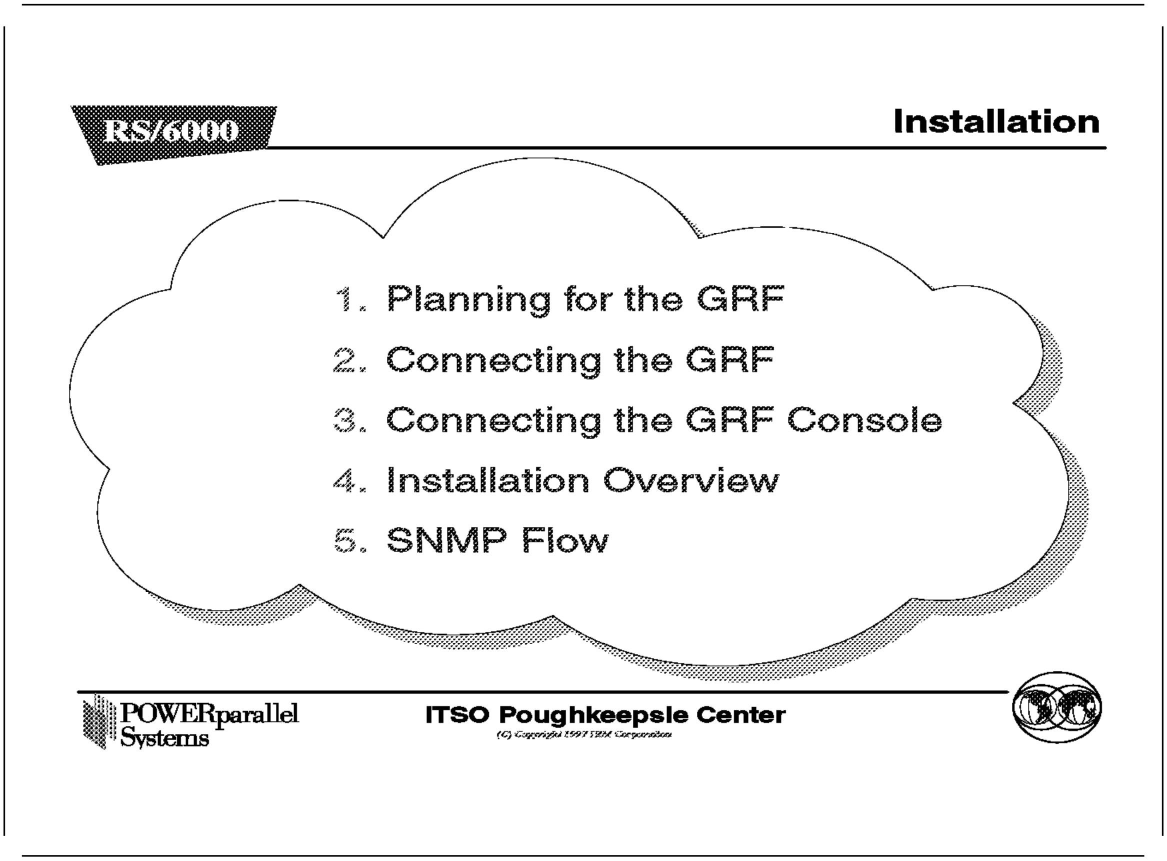
(C) Copyright 1997 IBM Corporation

This figure shows a single-frame RS/6000 SP broken into two partitions. Each partition has seven standard RS/6000 SP nodes and one dependent node. Only seven nodes are allowed in each partition, as a single-frame RS/6000 SP has only 16 SP Switch ports, and two of them are used for the SP Switch Router Adapter, one for each partition.

Normally, RS/6000 SP nodes in different partitions cannot communicate with each other through the SP Switch. The GRF plays a unique role here by allowing RS/6000 SP nodes to communicate across partitions, when each partition contains at least one SP Switch Router Adapter, and these adapters are interconnected by TCP/IP.

The requirements for partitioning are the same as those for coexistence, with the addition of having at least one free SP Switch port per partition, to connect to the SP Switch Router Adapter. A more detailed discussion of this situation is given in the Partition Installation figures of the Sample Configuration section.

## 7.4 Installation



RS/6000

Installation

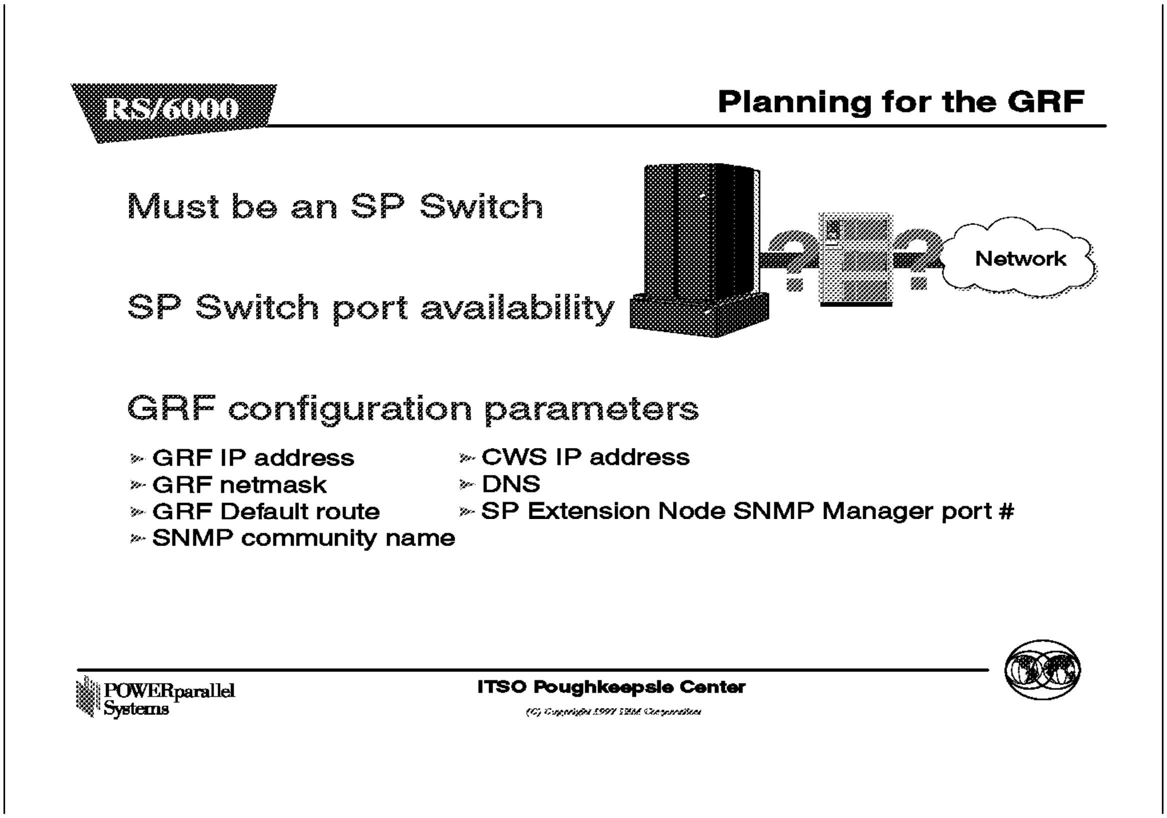
1. Planning for the GRF
2. Connecting the GRF
3. Connecting the GRF Console
4. Installation Overview
5. SNMP Flow

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This section offers an overview of the installation and planning process.



RS/6000

## Planning for the GRF

Must be an SP Switch

SP Switch port availability

GRF configuration parameters

- GRF IP address
- GRF netmask
- GRF Default route
- SNMP community name
- CWS IP address
- DNS
- SP Extension Node SNMP Manager port #

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Before acquiring any model of the SP Switch Router, ensure that there are SP Switch ports available in the designated partition, and the switch used in the RS/6000 SP is the 8-port or 16-port SP Switch.

Next, ensure that the following parameters are defined:

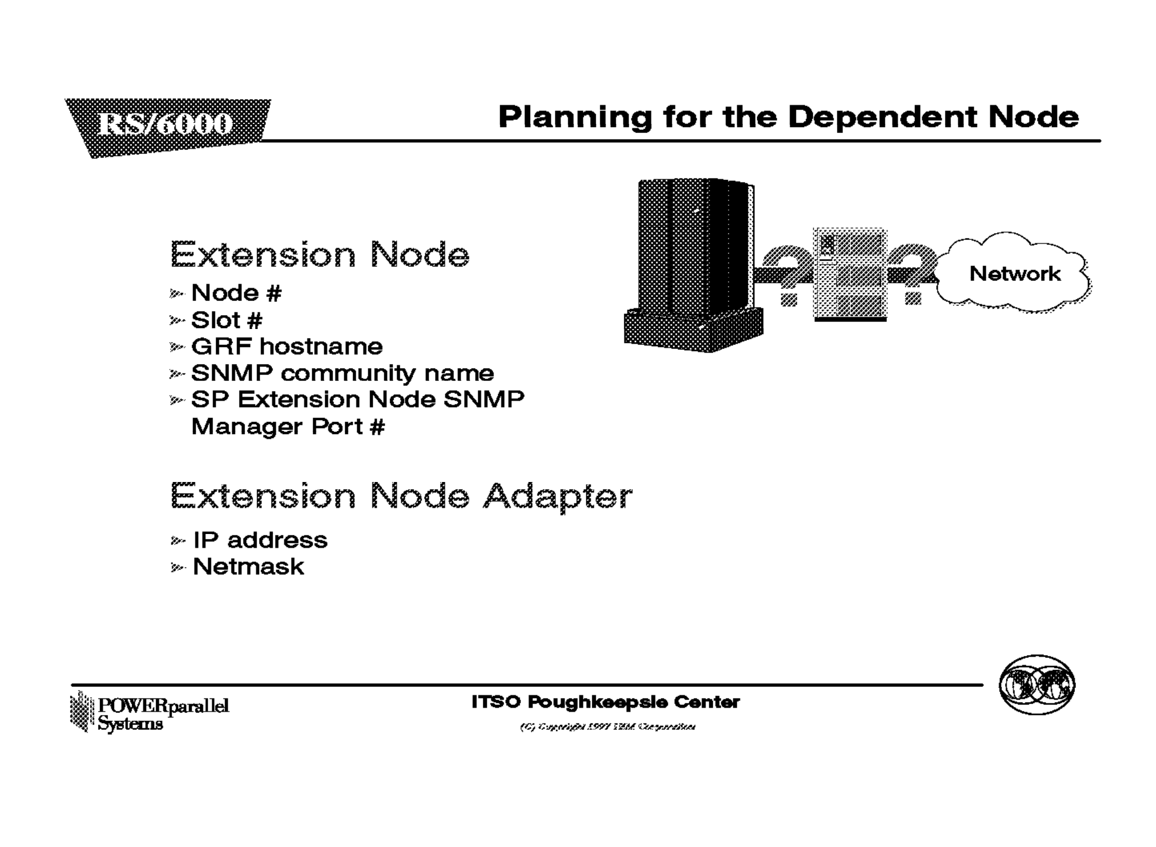
| **Parameters** | **Descriptions** |
|---|---|
| **GRF IP address** | IP address for GRF administrative Ethernet. |
| **GRF netmask** | Netmask for GRF administrative Ethernet. |
| **GRF Default route** | The default route of the GRF. |
| **SNMP community name** | This attribute describes the SNMP community name that the SP Extension Node SNMP Manager and the GRF′s SNMP Agent will send in the corresponding field of the SNMP messages. This value must match the value specified for the same attribute of the corresponding dependent node definition on the SP system. If left blank, a default name found in the SP Switch Router Adapter documentation is used. |
| **CWS IP address** | The Control Workstation′s IP address. When a GRF contains multiple SP Switch Router Adapters which are managed by different SNMP |

Mangers on different RS/6000 SP CWS, each of the Control Workstation IP address should be defined along with a different community name for each Control Workstation.

**DNS** The DNS server and domain name, if used.

**SP Extension Node SNMP Manager port #**

The SNMP port number used by the SP Extension Node SNMP Manager to communicate with the SNMP agent on the GRF. This port number is 162, when the SP Extension Node SNMP Manager is the only SNMP manager on the Control Workstation. Otherwise, another port number not used in the /etc/services of the Control Workstation is chosen.



RS/6000

## Planning for the Dependent Node

### Extension Node

- Node #
- Slot #
- GRF hostname
- SNMP community name
- SP Extension Node SNMP Manager Port #

### Extension Node Adapter

- IP address
- Netmask

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Next, for each dependent node on the RS/6000 SP, define the following:

**Parameters** **Descriptions**

**Node #** User supplied dependent node number representing the node position of a unused SP Switch port to be used by the SP Switch Router Adapter.

**Slot #** Slot number which the SP Switch Router Adapter is located in the GRF.

**GRF hostname** Hostname for GRF administrative Ethernet. Long hostname is recommended if domain name service (DNS) is used in the network. This represents both the Administrative and SNMP Agent Hostname of the dependent node.

**SNMP community name** This attribute describes the SNMP community name that the SP Extension Node SNMP Manager and the GRF′s SNMP Agent will send in the corresponding field of the SNMP messages. This value must match the value specified in the /etc/snmpd.conf file on the GRF. If left blank, a default name found in the SP Switch Router Adapter documentation is used.

**SP Extension Node SNMP Manager port #**

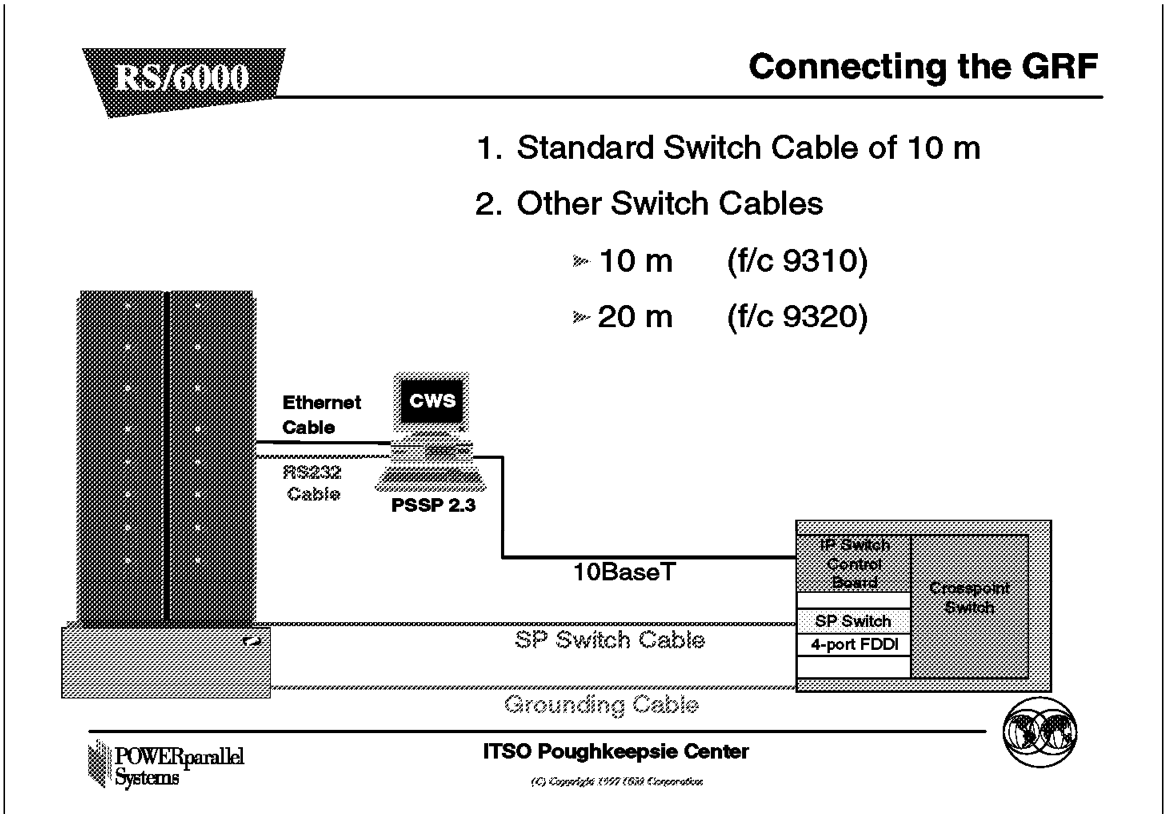
The SNMP port number used by the SP Extension Node SNMP Manager to communicate with the SNMP agent on the GRF. This port number is 162, when the SP Extension Node SNMP Manager is the only SNMP manager on the Control Workstation. Otherwise, another port number not used in the /etc/services of the Control Workstation is chosen.

Then, for the dependent node adapter, define these parameters:

**Parameter** **Descriptions**

**IP address** IP address of this adapter.

**Netmask** Netmask of this adapter. Use the same format as that for standard RS/6000 SP nodes.



RS/6000

## Connecting the GRF

1. Standard Switch Cable of 10 m
2. Other Switch Cables
   - 10 m (f/c 9310)
   - 20 m (f/c 9320)

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

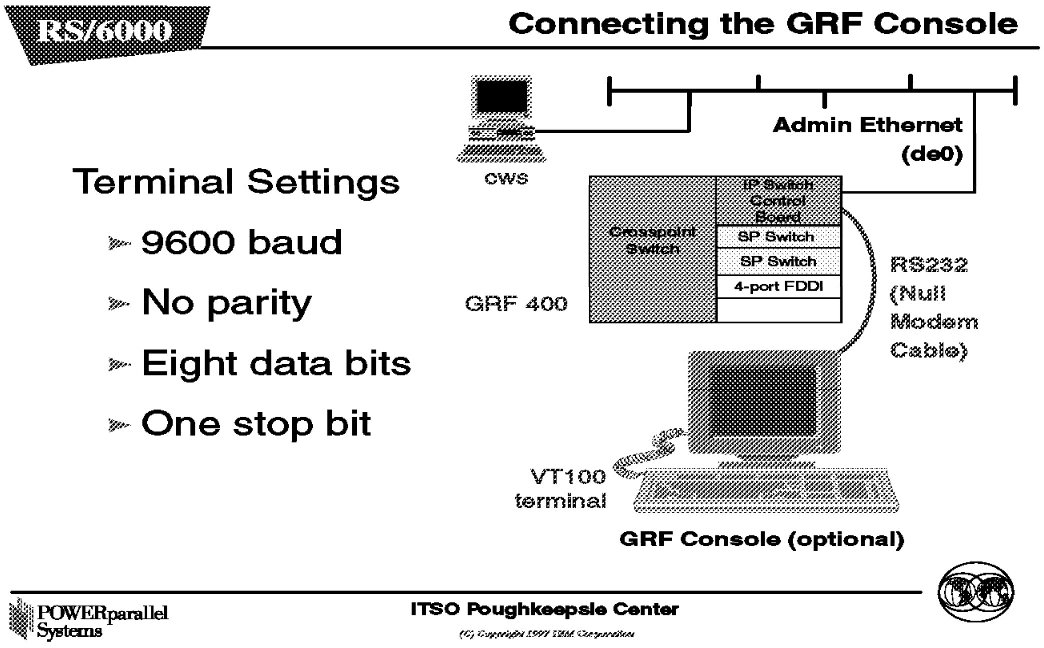
The GRF, when ordered with the SP Switch Router Adapter, comes with two cables, a 10 m SP Switch cable, and a 10 m grounding cable.

- The SP Switch cable connects the SP Switch port to the SP Switch Router Adapter on the GRF.
- The grounding cable connects the GRF chassis to the RS/6000 SP chassis for grounding the GRF.

The 10BaseT Ethernet cable is used to connect the GRF′s administrative Ethernet to the Control Workstation. The customer must supply the 10BaseT connection to the CWS. Alternatively, this Ethernet can be connected to the SP Ethernet by providing the appropriate bridge.

An RPQ is available to provide a 20 m SP Switch cable and grounding cable to extend the distance of the GRF from the RS/6000 SP. However, this cable cannot be wrap tested to check if it is damaged.

An alternative to using the GRF-provided SP Switch cable is to use the standard RS/6000 SP Switch cable. It is identical.



RS/6000

**Connecting the GRF Console**

Terminal Settings

- 9600 baud
- No parity
- Eight data bits
- One stop bit

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

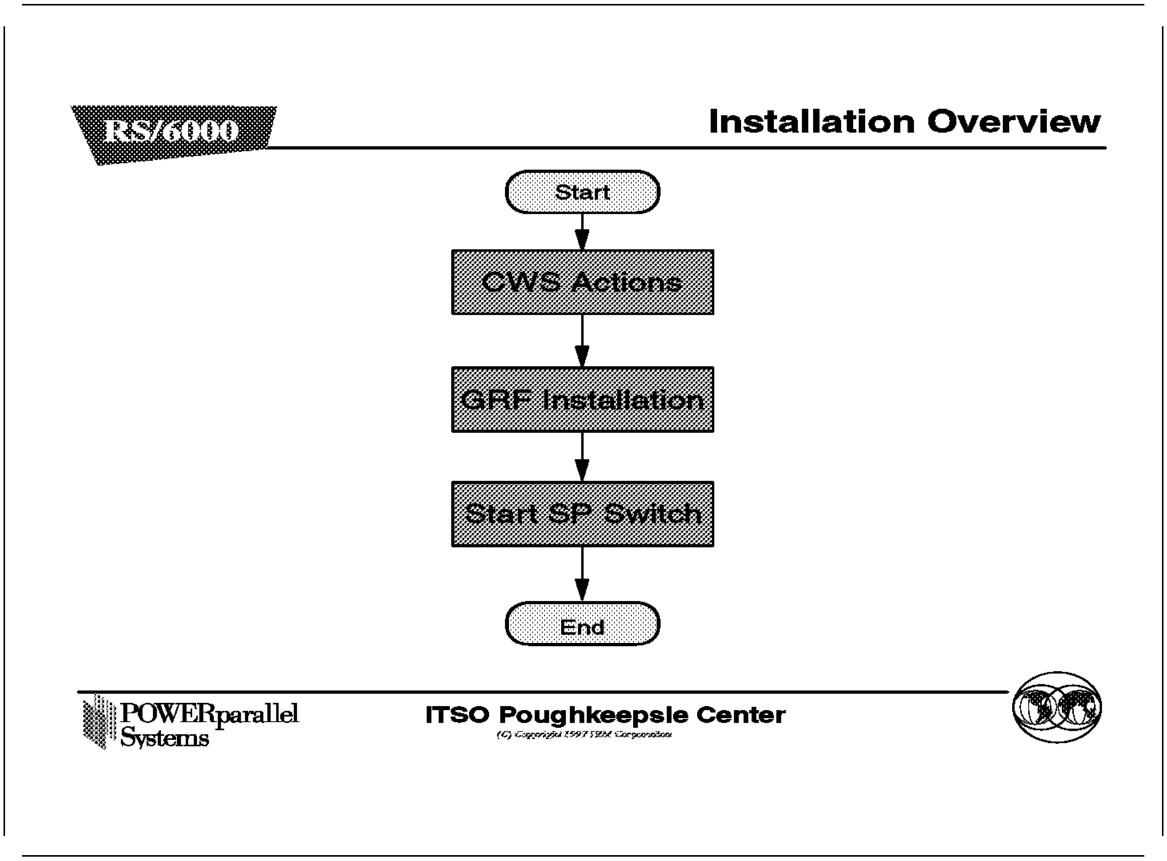
This figure shows how to connect the console to the GRF.

First, you need to supply an RS232 null modem cable and a VT100 terminal. The RS232 null modem cable is used to connect the IP Control Board (9-pin) to the VT100 terminal. The VT100 terminal must have the following settings:

- 9600 baud rate
- No parity
- Eight data bits
- One stop bit

For initial login, the user ID is root and the password is documented in the GRF publications.

Since the VT100 terminal is only required for the initial configuration of the GRF and not for its operation, the user can use a PC to simulate the VT100 terminal.



RS/6000

## Installation Overview

Start

CWS Actions

GRF Installation

Start SP Switch

End

POWERparallel Systems

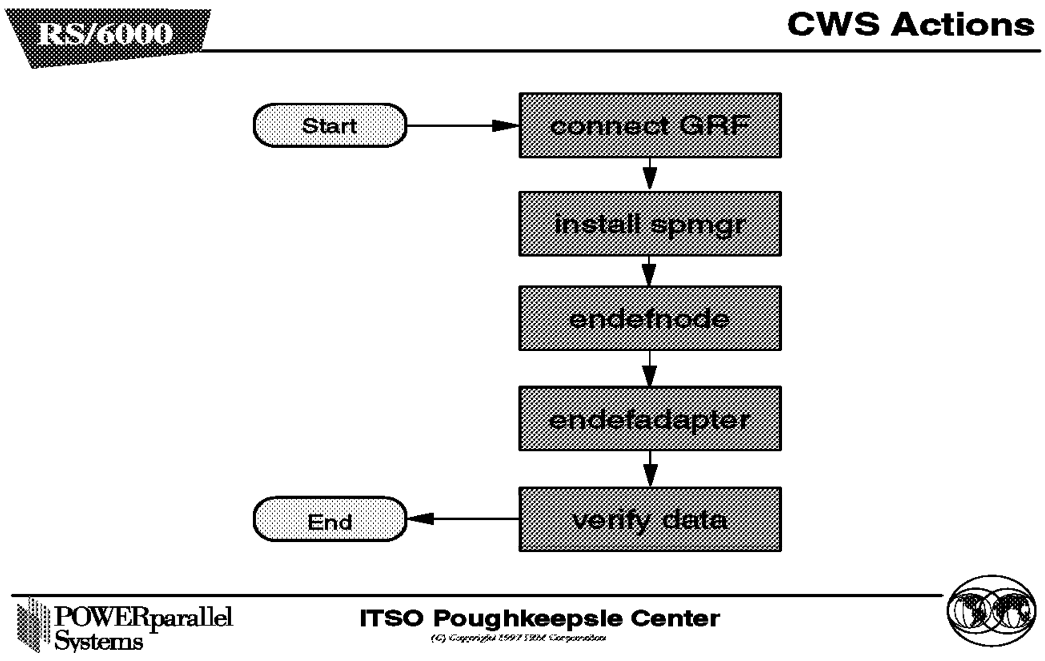
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The installation of the dependent node in the RS/6000 SP involves these three steps:

1. Control Workstation actions
2. SP Switch Router Adapter in the GRF
3. Starting the SP Switch

These steps are discussed in more detail in the next three figures.



The first CWS action is to connect the RS/6000 SP to the GRF. This includes the SP Switch cable, the GRF administrative Ethernet, and the GRF grounding cable.

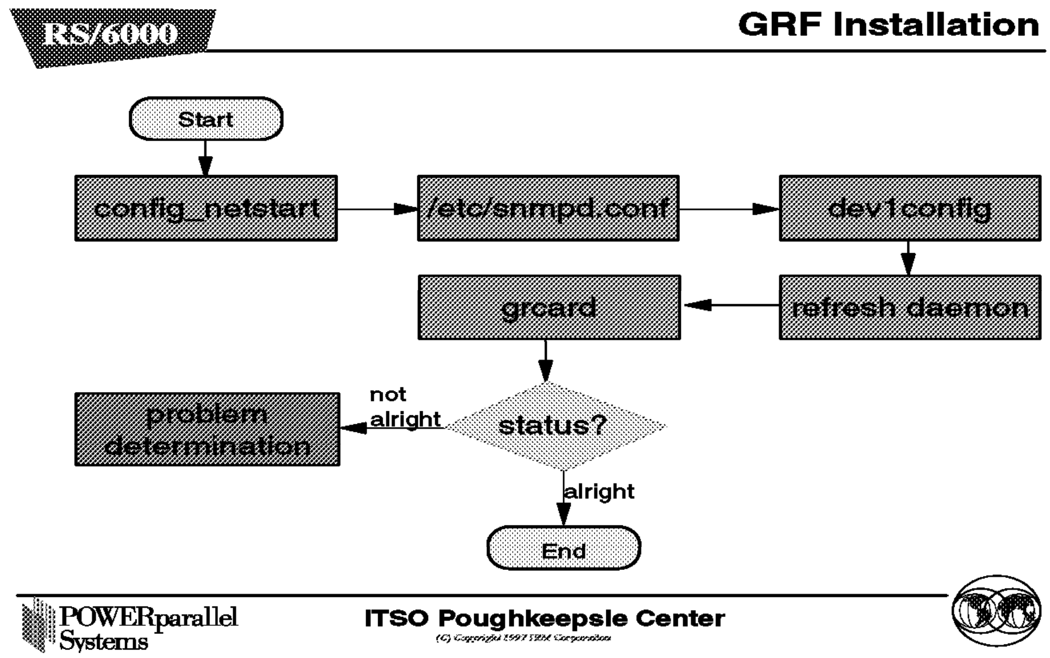
Next, install the ssp.spmgr fileset on the Control Workstation, and ensure that the spmgr daemon is started.

Next, use the commands endefnode and endefadapter to define the dependent node. These commands are described in the PSSP Enhancement section. Execute the two commands for all dependent nodes in the RS/6000 SP.

Finally, verify that the data used to define the dependent nodes were correct using the splstnodes and splstadapters commands.

```
# splstnodes -t dependent node_number switch_node_number \
> switch_chip_port switch_chip switch_number switch_partition_number \
> reliable_hostname management_agent_hostname extension_node_identifier \
> snmp_community_name
node_number  switch_node_number switch_chip_port switch_chip  switch_number
switch_partition_number reliable_hostname management_agent_hostname
extension_node_identifier snmp_community_name
          13           12             1            4            1            1
grf1.ppd.pok.ibm.com grf1.ppd.pok.ibm.com 03           ""
#
# splstadapters -t dependent node_number netaddr netmask
node_number  netaddr      netmask
          13 129.40.47.77 255.255.255.192
```

**Note:** To verify the node number used in the `endefnode` command and the actual switch connection, please refer to the Scalable POWERparallel Switch (SPS) Bulkheads figure in the "Installation of RS/6000 SP Optional Features" chapter of the *RS/6000 SP Maintenance Information Volume 1: Installation and CE Operations* (GC23-3903).


RS/6000

GRF Installation

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

For GRF Installation, perform the following:

1. When the GRF is first powered on, it starts by asking a series of questions in the console to configure itself. These questions can be generated again, to change the GRF configuration, with the command `/etc/sbin/config_netstart` on the GRF. A list of the questions is shown below.

   - `Host name for this machine? [ ]`

     Hostname for the GRF. Use the long name if DNS is used in the Control Workstation.

   - `Do you wish to configure the maintenance Ethernet interface? [yes]`

     Press Enter to take the default, yes. This is necessary to set up the GRF to work with the RS/6000 SP.

   - `IP address of this machine? [ ]`

     IP address of the GRF.

   - `Netmask for this network? [ ]`

     Netmask for the GRF.

   - `IP address for router ('none' for no default route)? [ ]`

     Default route for the GRF. Type `none` if none available. This attribute creates a static route to an external router for routing packets in the administrative Ethernet network.

- `Do you wish to go through the questions again? [yes]`

  Here, the GRF will list all the parameters that you have typed in. Enter no if they are correct. To make corrections, just press Enter.

- `Save a copy of this file as /etc/netstart.bak? [no]`

  Specify yes to get a backup copy of the configuration.

2. Edit `/etc/snmpd.conf` and add the following lines to the end of the file:

   ```
   MANAGER         <Control Workstation IP address>
                   SEND ALL TRAPS
                   TO PORT <SP manager port #>
                   WITH COMMUNITY <SNMP community name>

   COMMUNITY       <SNMP community name>
                   ALLOW ALL OPERATIONS
                   USE NO ENCRYPTION
   ```

   Replace the values in the <brackets> with site-defined parameters. This value is the same as the SNMP Community Name option defined in the endefnode command in the Control Workstation. To prohibit unauthorized SNMP Managers from configuring an extension node, change the existing ′public′ community name access to;

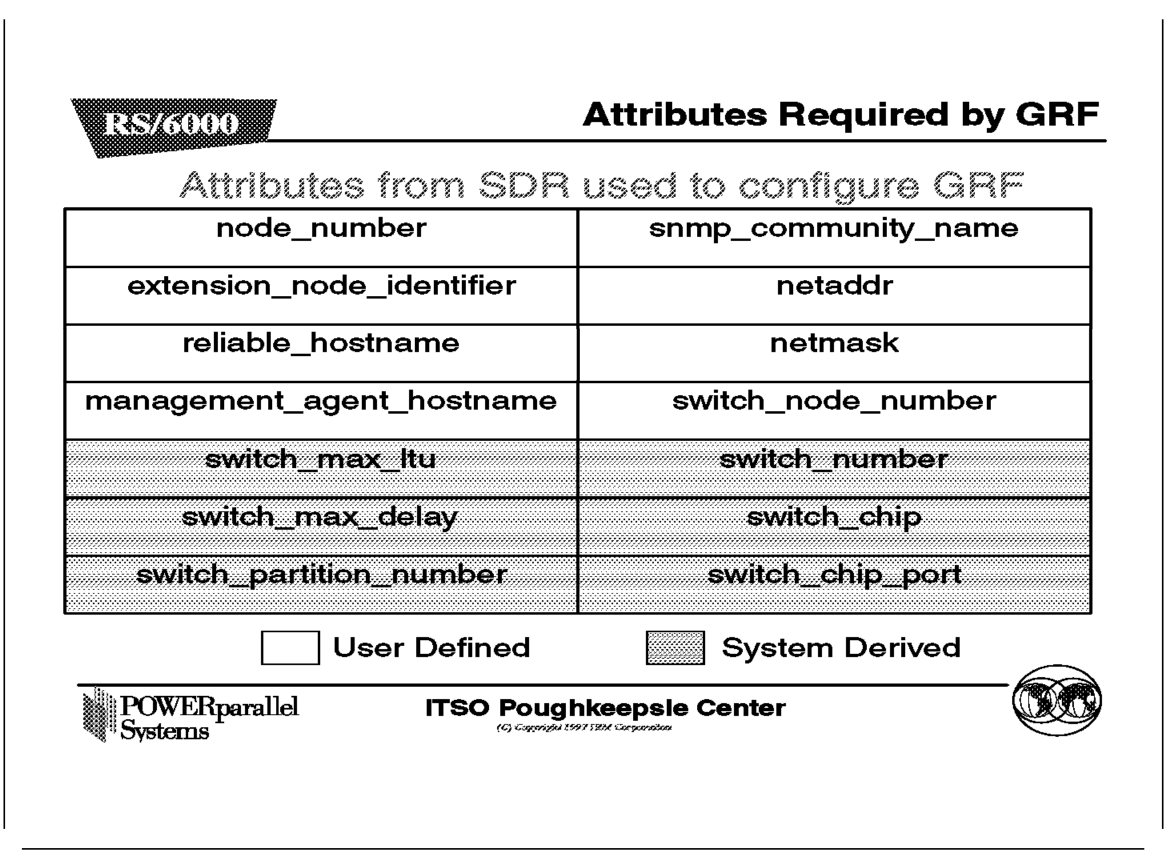
   ```
   COMMUNITY      public
                  ALLOW GET, TRAP OPERATIONS
                  USE NO ENCRYPTION
   ```

3. Execute `dev1config` to configure the dependent node on the GRF. Among other things, this command creates the /etc/grdev1.conf and /etc/grdev1.conf.template files, and also updates the /etc/grinchd.conf and /etc/grifconfig.conf files.

4. Next, on the GRF console, refresh the grinch daemon and the SNMP daemon. Use the `ps ax` and `grep` commands to list the process ID of the daemons. Execute the `kill` command on the two process for the two process to respawn themselves. Below is an example of this process:

   ```
   # ps ax]grep grinch
   15592  ??  S      0:00.51 grinchd -DNAGER         129.40.47.62
   15811  p0  S+     0:00.02 grep grinch
   # kill 15592
   May  3 04:51:00 grf1 root: grstart:  grinchd exited status 143; restarting.
   #
   # ps ax]grep snmp
   15600  ??  S      0:00.14 snmpd /etc/snmpd.conf /var/run/sn mpd.NOV
   # kill 15600
   May  3 04:54:43 grf1 mib2d[15605]: mib2d: terminated by master agent
   May  3 04:54:43 grf1 root: grstart: snmpd exited status 143; restarting.
   May  3 04:54:43 grf1 root: grstart: mib2d exited status 0; restarting.
   ```

5. Finally, type `grcard` on the console. Check the status of the SP Switch Router Adapter. It will show the slot number, the adapter name, and the status of the card. The SP Switch Router Adapter is known as `DEV1_V1` in this listing. If the status is `loading`, it means that it is polling the Control Workstation using the SNMP InfoNeeded trap to request configuration data, and you are done. If not, there is a configuration problem with the SP Switch Router Adapter.



RS/6000

## Attributes Required by GRF

Attributes from SDR used to configure GRF

| | |
|---|---|
| node_number | snmp_community_name |
| extension_node_identifier | netaddr |
| reliable_hostname | netmask |
| management_agent_hostname | switch_node_number |
| switch_max_ltu | switch_number |
| switch_max_delay | switch_chip |
| switch_partition_number | switch_chip_port |

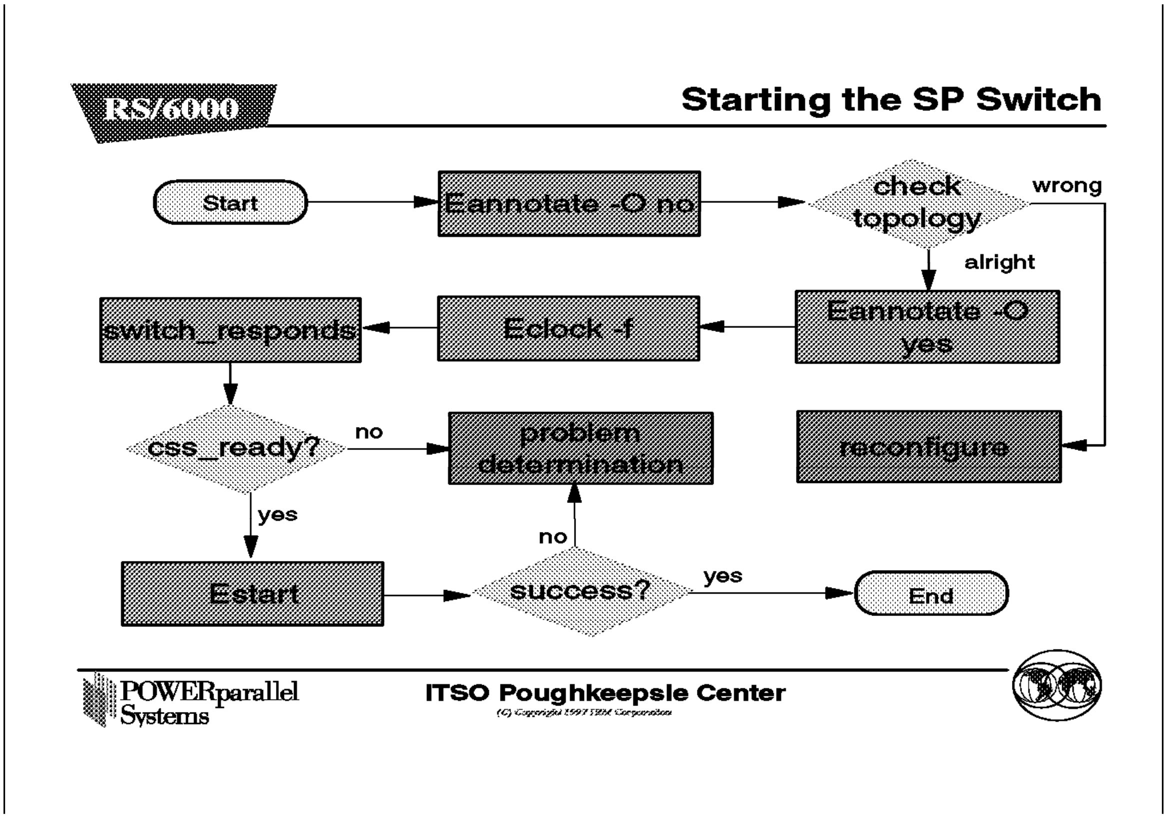
□ User Defined ▨ System Derived

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This table shows the attributes required by the GRF to set up the SP Switch Router Adapter. They are all defined by the endefnode and endefadapter commands. Explanations of the commands and attributes are found in the PSSP Enhancements section.



RS/6000

## Starting the SP Switch

Start

Eannotate -O no

check topology

wrong

alright

Eannotate -O yes

Eclock -f

switch_responds

css_ready?

no

problem determination

reconfigure

yes

no

Estart

success?

yes

End

POWERparallel Systems

ITSO Poughkeepsie Center

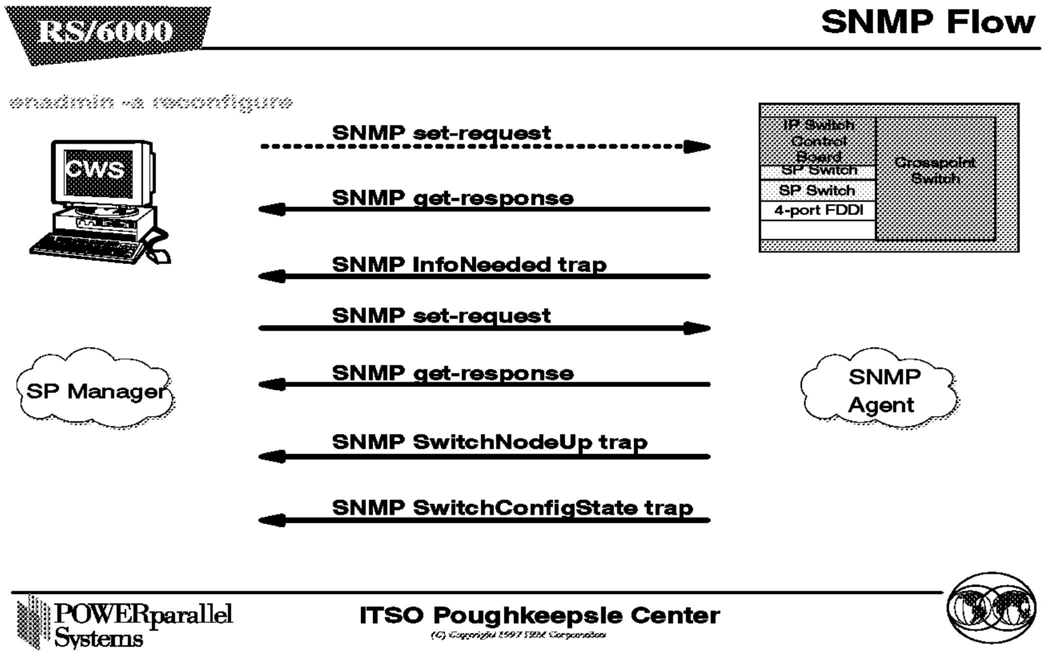
To start the SP Switch, first check the annotated switch file produced by the Eannotator command, without storing the topology file in the SDR.

If the annotated switch topology file shows the dependent node in the RS/6000 SP to be different from that stated in the endefnode command, it means that either the SP Switch Router Adapter is connected to the wrong SP Switch port, or the node number was entered incorrectly in the endefnode command. You need to either reconnect the SP Switch Router Adapter, or rerun the endefnode command to correct the problem before continuing. When updating the configuration with the endefnode or endefadapter command, specify the -r flag, so that the GRF will be notified of the change and poll the Control Workstation for the update.

When the annotated file is correct, execute the Eannotator command again, to store the topology file in the SDR. Run the Eclock command to reset the SP Switch clock, and reset the worm daemon on all standard RS/6000 SP nodes.

Use the SDRGetObjects switch_responds command to check the adapter_config_state attributes for all the dependent nodes.

If all adapter_config_state attributes are css_ready, run the Estart command. If any of the nodes′ adapter_config_state attributes are not css_ready, the Estart will fail for the corresponding node. If any of the dependent nodes′ adapter_config_status is not css_ready, or the Estart fails, perform problem determination using the steps in the Hints and Tips section.



SNMP Flow

The addition of the SP Switch Router Adapter adds four specific traps to the SNMP:

SNMP InfoNeeded
SwitchNodeUP
SwitchNodeDown
SwitchConfigState.

Except for these traps, most other SNMP traps generated by the GRF are ignored by the SP Extension Node SNMP Manager. However, if the user has another SNMP manager in the network, it can query adapter configuration and state information, and monitor the flow of SNMP traps between the GRF Agent and the SP Extension Node SNMP Manager on the Control Workstation.

When the GRF is first powered on, it periodically sends the InfoNeeded SNMP trap to the SP Extension Node SNMP Manager for configuration information. Alternatively, the `enadmin -a reconfig` command will send an SNMP set-request containing an extension node identifier and an administrative status of reconfigure to the GRF to trigger the InfoNeeded trap.

When the Control Workstation receives the InfoNeeded trap, it sends a SNMP set-request containing the extension node identifier and the configuration attributes for that dependent node to the GRF SNMP Agent at UDP port 161. When the GRF Agent has received all the configuration information, it sends an SNMP get-response to the SP Extension Node SNMP Manager on the Control Workstation. The information is then applied to the SP Switch Router Adapter, and the GRF sends two SNMP traps, SwitchNodeUp and SwitchConfigState, to indicate that it is ready.

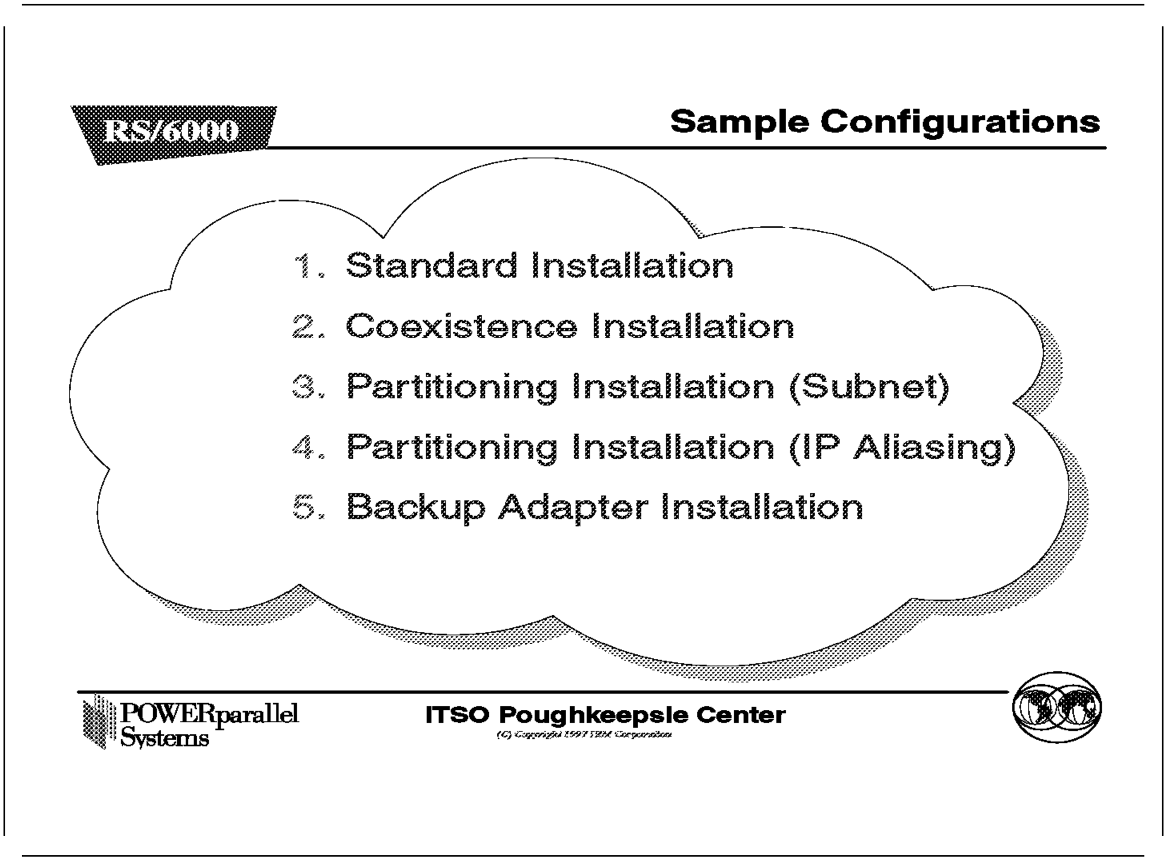
**Notes:**

1. When an Estart or Eunfence is issued, processing is done via link-level service packet exchanges between the dependent node and the Primary node using the SP Switch. The Primary node next sets switch_responds for the dependent node.

2. When Efence is issued, a SwitchNodeDown SNMP trap is sent by the GRF SNMP Agent, and via link-level service packet exchanges between the dependent node and Primary node, the Primary node sets the switch_responds for the dependent node.

3. When the dependent node enables or reenables its SP Switch interface, the GRF SNMP Agent sends a SwitchNodeUp SNMP trap to the SP SNMP Manager. If the Efence command was previously issued with the ′autojoin′ option to remove the dependent node from the SP Switch network, the SNMP Manager will issue the Eunfence command to allow the dependent node to join the SP Swtich network.

## 7.5 Sample Configurations



RS/6000

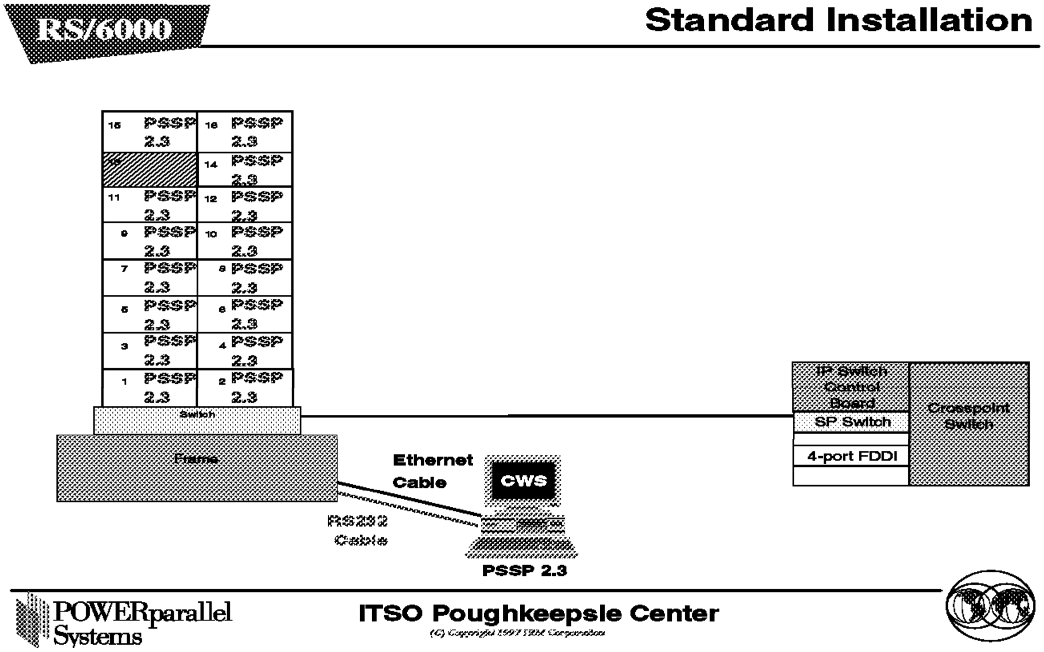
### Sample Configurations

1. Standard Installation
2. Coexistence Installation
3. Partitioning Installation (Subnet)
4. Partitioning Installation (IP Aliasing)
5. Backup Adapter Installation

POWERparallel Systems

ITSO Poughkeepsie Center

This section offers some sample configurations for using the GRF with the RS/6000 SP.



RS/6000

Standard Installation

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This example describes the steps for an installation of GRF into an RS/6000 SP system.

1. First, on the Control Workstation, we install the SP Extension Node SNMP Manager:

```
# installp -a -d /spdata/sys1/install/pssplpp/PSSP-2.3 -X ssp.spmgr
        .
        .
        .
        .
Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
ssp.spmgr                   2.3.0.0         USR         APPLY       SUCCESS
ssp.spmgr                   2.3.0.0         ROOT        APPLY       SUCCESS
```

   Here we assume that the RS/6000 SP software is in the /spdata/sys1/install/pssplpp/PSSP-2.3 directory. We could have similarly installed it from tape (assuming tape drive 0) using `installp -a -d /dev/rmt0 -X ssp.spmgr`.

2. Next, check for its status using `lssrc -s spmgr`. Start the SP Extension Node SNMP Manager on the Control Workstation if it is not started. We turn on tracing on the SP Extension Node SNMP Manager to provide us with more information should the dependent node installation fail. The following

example checks for the activity of the spmgr daemon, starts it, verifies that it is active, and turns tracing on for the daemon:

```
# lssrc -s spmgr
Subsystem         Group            PID     Status
 spmgr                                     inoperative
#
# startsrc -s spmgr
0513-059 The spmgr Subsystem has been started. Subsystem PID is 17574.
#
# lssrc -s spmgr
Subsystem         Group            PID     Status
 spmgr                             17574   active
#
# traceson -ls spmgr
Start trace.
0513-091 The request to turn on tracing was completed successfully.
```

If you intend to run with tracing enabled during production, you can limit the size of the trace table by specifying the maximum size as a ′-s′ switch value to be passed to the spmgr daemon when it is started (for example ′startsrc -s spmgr -a″-s <size>″ ′).

3. Next, we define the dependent node on the Control Workstation with endefnode and endefadapter, using the following parameters:

   - The GRF hostname is grf1.ppd.pok.ibm.com.
   - SP Switch Router Adapter is in Slot 2 of the GRF.
   - The dependent node number is 13.
   - The IP address for the dependent node is 129.40.47.77 (IP address of SP Switch for node 13 in the RS/6000 SP).
   - The netmask for the dependent node is 255.255.255.192 (netmask for the SP Switch on the RS/6000 SP).

   ```
   # endefnode -a grf1.ppd.pok.ibm.com -i 02 -s grf1.ppd.pok.ibm.com 13
   The endefnode command has completed successfully.
   #
   # endefadapter -a 129.40.47.77 -m 255.255.255.192 13
   The endefadapter command has completed successfully.
   ```

4. After setting up the dependent node on the RS/6000 SP, we proceed to set up the GRF. The following questions are asked when the GRF is powered up for the first time or these questions can be activated using the config_netstart command on the GRF:

   - The hostname is grf1.ppd.pok.ibm.com (hostname for the GRF′s administrative Ethernet).
   - Answer yes to configure the maintenance Ethernet.
   - Use 129.40.41.47 (IP address defined for the GRF′s administrative Ethernet).
   - Use 255.255.255.0 (netmask for the GRF′s administrative Ethernet).
   - Use 129.40.47.62 for the default route.
   - Specify no to avoid going through the questions again.
   - Specify yes to save a copy of the configuration in /etc/netstart.bak.

5. Next, we configure the GRF to communicate with the Control Workstation. Append the following lines to /etc/snmpd.conf in the GRF:

```
MANAGER         129.40.47.62
                SEND ALL TRAPS
                TO PORT 162
                WITH COMMUNITY spenmgmt

COMMUNITY       spenmgmt
                ALLOW ALL OPERATIONS
                USE NO ENCRYPTION
```

6. Next, execute dev1config to configure the SP Switch Router Adapter and refresh the SNMP and grinch daemons:

```
# dev1config
#
# ps ax]grep grinch
15592  ??  S      0:00.51 grinchd                129.40.47.62
15811  p0  S+     0:00.02 grep grinch
#
# kill 15592
May  3 04:51:00 grf1 root: grstart:  grinchd exited status 143; restarting.
#
# ps ax]grep snmp
15600  ??  S      0:00.14 snmpd /etc/snmpd.conf /var/run/sn mpd.NOV
# kill 15600
May  3 04:54:43 grf1 mib2d[15605]: mib2d: terminated by master agent
May  3 04:54:43 grf1 root: grstart: snmpd exited status 143; restarting.
May  3 04:54:43 grf1 root: grstart: mib2d exited status 0; restarting.
```

7. Execute the grcard command on the GRF, and check to make sure that the SP Switch Router Adapter, known as DEV1_V1, is running:

```
# grcard
0       ETHER_V1        running
2       DEV1_V1 running
3       HIPPI_V1        running
4       HSSI_V1 running
```

8. We next return to the Control Workstation to start the SP Switch. First, we run the Eannotator and Eclock commands, before starting the SP Switch with an Estart.

```
# Eannotator -F /etc/SP/expected.top.2nsb.0isb.0 \
> -f /etc/SP/ann.2nsb.0isb -O no
```

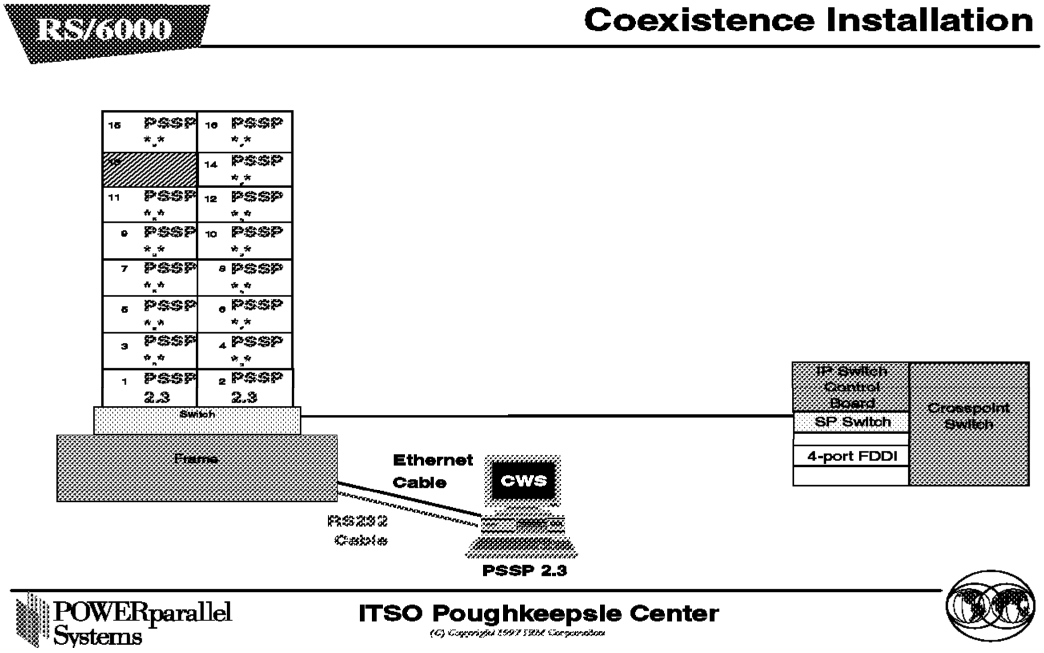
The annotated file is checked for correct dependent node positioning before storing it into the SDR and setting the SP Switch clock, as follows:

```
# more /etc/SP/ann.2nsb.0isb
     .
     .
     .
s 14 1  tb3 12 0           E01-S17-BH-J33 to E01-N13 # Dependent Node
#
# Eannotator -F /etc/SP/expected.top.2nsb.0isb.0 \
> -f /etc/SP/ann.2nsb.0isb -O yes
# Eclock -f /etc/SP/Eclock.top.2nsb.0isb.0
```

9. Finally, we check the SDR class switch_responds to ensure that the adapter_config_status of the dependent nodes is css_ready before starting the SP Switch:

```
# SDRGetObjects -G switch_responds
node_number  switch_responds autojoin     isolated     adapter_config_status
     .
     .
     .
          13            1            0            0 css_ready
#
# Estart
Switch initialization started on ceed1n05.ppd.pok.ibm.com.
Initialized 5 node(s).
Switch initialization completed.
```

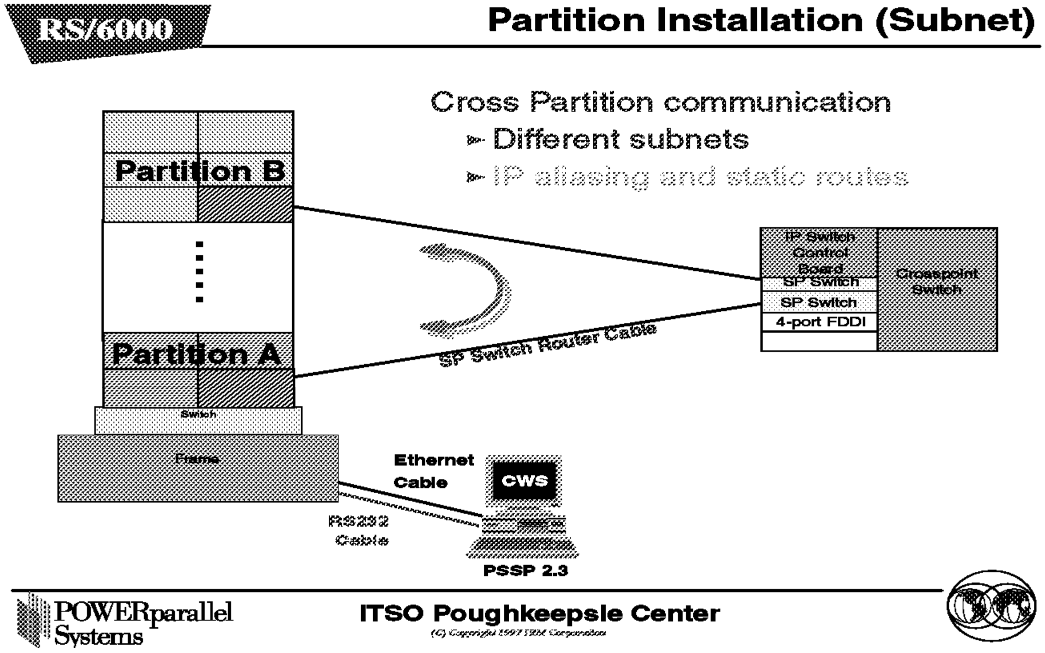
The number of nodes initialized includes both standard and dependent nodes in the RS/6000 SP partition.



The requirements for a dependent node to be installed in a partition with multiple PSSP levels are outlined in the Coexistence figure of the PSSP Enhancements section.

Here, we assume an RS/6000 SP with multiple PSSP levels in a partition, complying with the coexistence requirements, and the nodes in the partition are able to communicate with each other through the SP Switch. If these conditions are met, the installation of the dependent node in this scenario is the same as the standard installation shown in the previous figure.

This installation allows the non-PSSP 2.3 RS/6000 SP nodes to work with the dependent node in the partition.



RS/6000

## Partition Installation (Subnet)

Cross Partition communication
- Different subnets
- IP aliasing and static routes

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In this example, we install the RS/6000 SP such that SP Switch adapters in different partitions have different subnets. We assume the availability of a single frame RS/6000 SP system, with two partitions of eight nodes each. The IP address for the SP Switch network with a netmask of 255.255.255.0 listed below:

- Partition A
  - Node 1: 129.40.47.1 a1
  - Node 2: 129.40.47.2 a2
  - Node 5: 129.40.47.5 a5
  - Node 6: 129.40.47.6 a6
  - Node 9: 129.40.47.9 a9 (dependent node)
- Partition B
  - Node 3: 129.40.48.3 b3
  - Node 4: 129.40.48.4 b4
  - Node 7: 129.40.48.7 b7
  - Node 8: 129.40.48.8 b8
  - Node 11: 129.40.48.11 b11 (dependent node)

Use the instructions of the Standard Installation figure in this section to install the dependent node in each partition.

First, perform Steps 1 and 2.

Perform Step 3 twice, once for dependent node 9 and once for dependent node 11 shown above. Use the IP address and netmask of a9 and b11 instead. Use Slots 02 and 03 of the GRF for each of the dependent node. The endefnode and endefadapter commands should be executed in the each partition. Before executing these commands, set the appropriate partition by executing export SP_NAME=<partition name>.

Next, perform Steps 4, 5, 6 and 7. For Step 7, grcard should show DEV_V1 running on Slot 2 and 3 instead.

For Step 8, the topology files for a partitioned RS/6000 SP are found in the /spdata/sys1/syspar_configs/topologies directory. The correct topology file to use with Eannotator can be listed by the SDRGetObjects Switch_partition topology_filename command. Note that the topology file listed in this manner ends with a dot and a number. This is the version number of the topology file stored in the SDR. When using the Eannotator command, ignore this version number. If you list the topology files in the /spdata/sys1/syspar_configs/topologies directories, you will notice that the partitioned RS/6000 SP topology files end with "isb." Again, perform Step 8 twice, once for each partition.

Finally, perform Step 9 to complete the definition of the dependent nodes. When SDRGetObjects -G switch_responds is performed, check adapter_config_status for both dependent nodes to ensure that both are in the css_ready state. Do Estart twice, once for each partition.

Next, we set up the RS/6000 SP nodes, so that they can communicate with each other across partitions. In partitioning, nodes in one partition do not communicate with nodes in other partitions. They can communicate with the dependent nodes in their own partition. In order for them to communicate across partitions using the GRF, we need to set up routes.

Finally, we set up static routes from each node to enable it to communicate via GRF with the nodes in the other partition. For every node in Partition A, execute the following statement to add a static route to Partition B:

```
# route add -net b3 -netmask 255.255.255.0 a9
a9 net b3: gateway a9
#
```

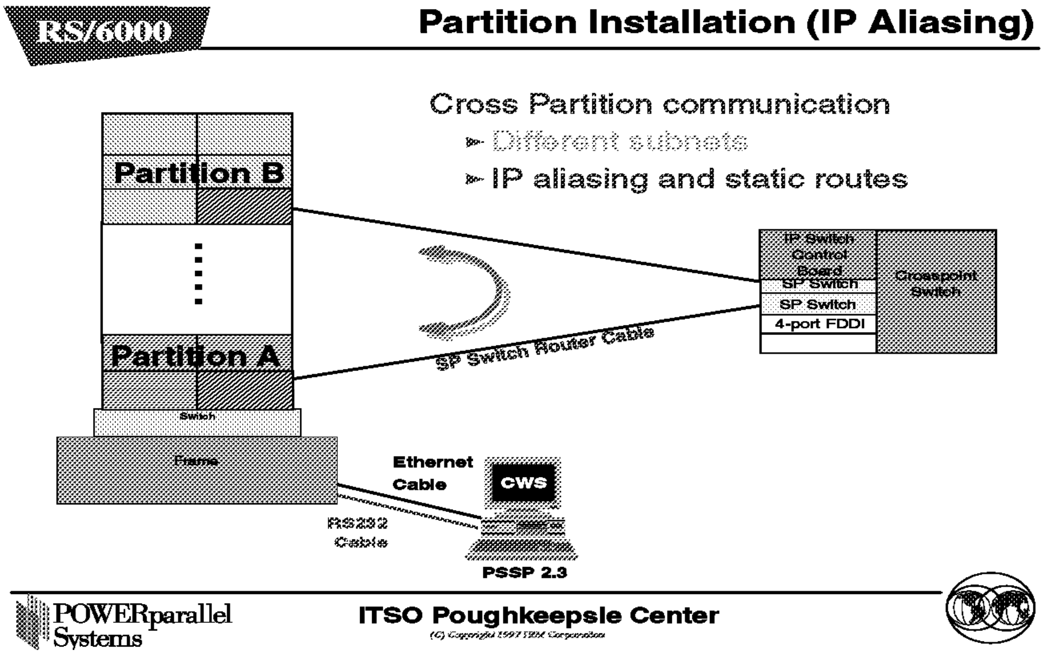
And for every node in Partition B, execute the following statement to add a static route to Partition A:

```
# route add -net a1 -netmask 255.255.255.0 b11
b11 net a1: gateway b11
#
```

Now the RS/6000 SP nodes in Partition A are able to communicate with nodes in Partition B. In order for the routes to be available after a reboot, add them to the /etc/rc.net file. Alternatively, these routes could be set up using dynamic routing protocols, such as RIP or OSPF, that are supported both on the RS/6000 SP and the GRF.

**Attention**

In order for communication to be possible across partitions through the SP Switch, switch_responds must be green for the source node, the destination node and the dependent nodes in each partition.



## Partition Installation (IP Aliasing)

Cross Partition communication

- Different subnets
- IP aliasing and static routes

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Partitions in the RS/6000 SP are separate networks to the system. They should use different subnet masks; this is a requirement when we want them to talk to each other through a single GRF. When both partitions are in the same subnet, the routing table on the GRF will only register one of the routes to the RS/6000 SP. Thus, one of the partitions is not reachable through the GRF.

In this example, we show how to make the partitions talk to each other when they are in a single subnet. The IP addresses of the RS/6000 SP Switch and their aliases with a netmask of 255.255.255.0 are as follows:

- RS/6000 SP
  - Node 1: 129.40.49.1 c1
  - Node 2: 129.40.49.2 c2
  - Node 3: 129.40.49.3 c3
  - Node 4: 129.40.49.4 c4
  - Node 5: 129.40.49.5 c5
  - Node 6: 129.40.49.6 c6
  - Node 7: 129.40.49.7 c7
  - Node 8: 129.40.49.8 c8
  - Node 9: 129.40.49.9 c9 (dependent node)
  - Node 11: 129.40.49.11 c11 (dependent node)

- Partition A (aliases)
  - Node 1: 129.40.47.1 a1
  - Node 2: 129.40.47.2 a2
  - Node 5: 129.40.47.5 a5
  - Node 6: 129.40.47.6 a6
  - Node 9: 129.40.47.9 a9 (dependent node)
- Partition B (aliases)
  - Node 3: 129.40.48.3 b3
  - Node 4: 129.40.48.4 b4
  - Node 7: 129.40.48.7 b7
  - Node 8: 129.40.48.8 b8
  - Node 11: 129.40.48.11 b11 (dependent node)

To set up the above, follow the instructions of the Partition Installation (Subnet) figure in this section. Use the address listed in the RS/6000 SP bullet for the SP Switch above for both partitions instead.

Next, set up the alias on the SP Switch Router Adapters on the GRF by editing the /etc/grifconfig.conf file in the GRF. Here, the adapters are in Slots 02 and 03.

```
      .
      .
      .
gt020     129.40.49.9     255.255.255.0
gt020     129.40.47.9     255.255.255.0
gt030     129.40.49.11    255.255.255.0
gt030     129.40.48.11    255.255.255.0
```

After inserting the two statements on gt020 and gt030, save the file and reset the two adapters. This activates the aliases.

```
# grreset 2
Ports reset: 2
# grreset 3
Ports reset: 3
May 13 00:40:43 classgig kernel: gt020:  GigaRouter DEV1, GRIT address 0:2:0
May 13 00:40:43 classgig kernel: gt020:  GigaRouter DEV1, GRIT address 0:2:0
May 13 00:40:46 classgig kernel: gt030:  GigaRouter DEV1, GRIT address 0:3:0
May 13 00:40:46 classgig kernel: gt030:  GigaRouter DEV1, GRIT address 0:3:0
#
```

Next, set up the alias for the SP Switch adapter on the RS/6000 SP nodes via the ifconfig command. To set up the alias on node 1, use the following command:

```
# ifconfig css0 a1 netmask 255.255.255.0 alias
```

To check whether it was successfull, use the netstat -i command. Execute the above commands on all RS/6000 SP nodes, using the appropriate IP alias.

Finally, on each RS/6000 SP node, add a static route to reach the nodes on the other partition:

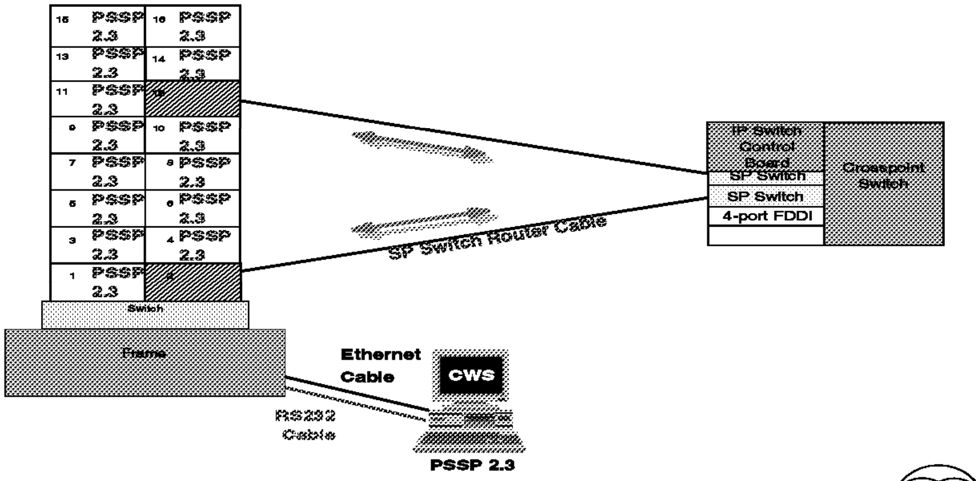
```
# route add -net b3 -netmask 255.255.255.0 a9
a9 net b3: gateway a9
#
```

This is similar to adding the static routes in the previous example.

Now the RS/6000 SP nodes in Partition A can communicate with nodes in Partition B, and vice versa, using the IP aliases. In order for these routes to be available after a reboot, insert the `route add` commands in the /etc/rc.net file. Alternatively, these routes can be set up using dynamic routing protocols, such as RIP or OSPF, that are supported both on the RS/6000 SP and the GRF.

RS/6000

## Backup Adapter Installation



POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In this example, we show how to install two dependent nodes in one partition. As mentioned in the previous examples, when we have more than one media card with the same subnet on the GRF, only one of them is recorded in the GRF′s routing table. Connecting the RS/6000 SP to the GRF in this manner gives us additional availability. Should one of the media cards be unavailable, the other media card will take over. For the RS/6000 SP, the same happens when we connect two SP Switch Router Adapters to the same partition.

For this example to work, the whole network has to run OSPF. On the RS/6000 SP, at least one node on each partition must run OSPF. OSPF must be running on the GRF as well.

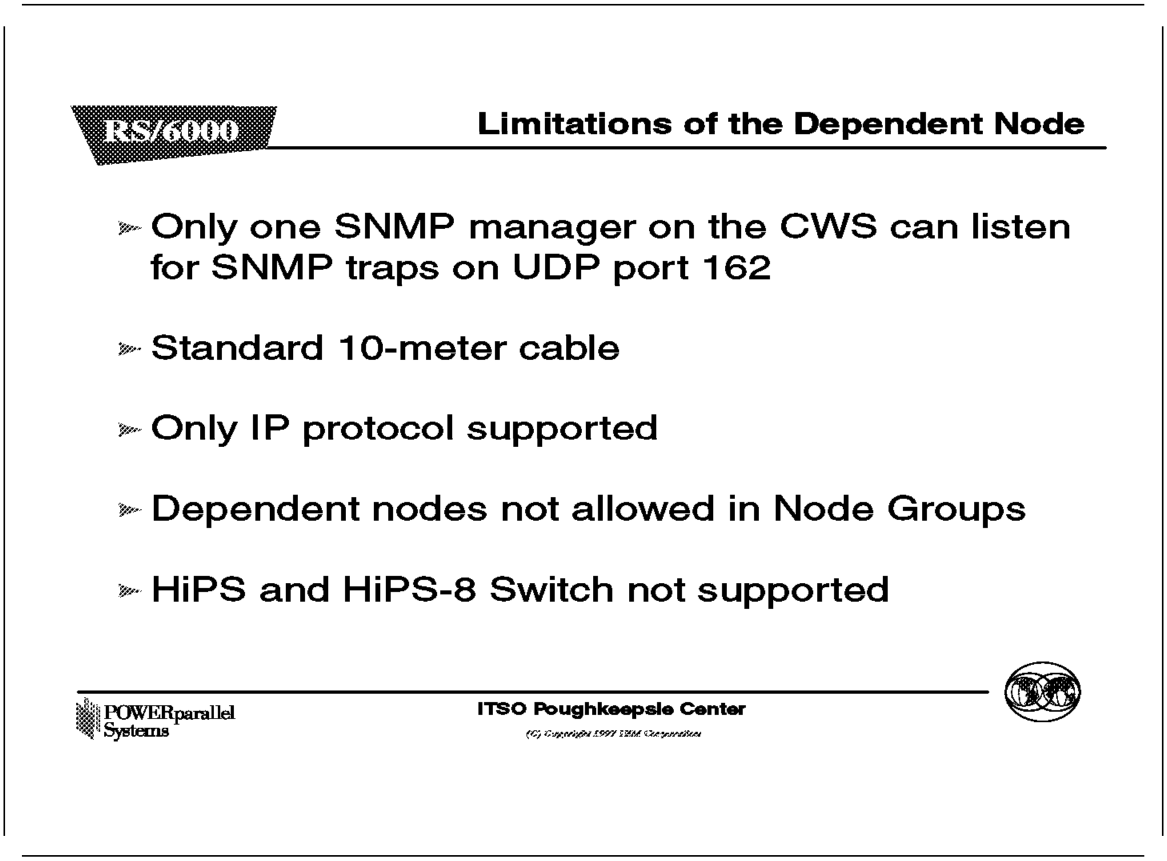
OSPF will configure the routes to the GRF using different weights. Normally, communication between the GRF and the RS/6000 SP uses the SP Switch Router Adapter with a lower-weight route. When that SP Switch Router Adapter is unavailable, the corresponding route is also unavailable, and all IP traffic (except that using static routes), dynamically reroutes to the other route with the active SP Switch Router Adapter. In this manner, it enhances the availability of the RS/6000 SP connection to the GRF in that partition.

**Note:** In the above example, availability can be enhanced by connecting two GRFs, each with an SP Switch Router Adapter, to a single partition. Should the GRF fail in the above example, all routes going through the SP Switch will be unavailable. With two GRFs, when one fails, the other will still be available.

The requirements for this example are exactly the same as those for the Backup Adapter Installation example. OSPF must be running on at least one node of the partition and on both GRFs. In addition, both GRFs must be interconnected by a TCP/IP media like HIPPI or FDDI, and this link must be active.

Lastly, in this example, since there are two GRFs, there are two routing tables available, one on each GRF. Each GRF records the route created by the SP Switch Router Adapter, even though they are in the same subnet. This offers greater flexibility in assigning IP packets between the two routes, and in balancing the IP load.

## 7.6 Limitations



RS/6000

**Limitations of the Dependent Node**

- Only one SNMP manager on the CWS can listen for SNMP traps on UDP port 162
- Standard 10-meter cable
- Only IP protocol supported
- Dependent nodes not allowed in Node Groups
- HiPS and HiPS-8 Switch not supported

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

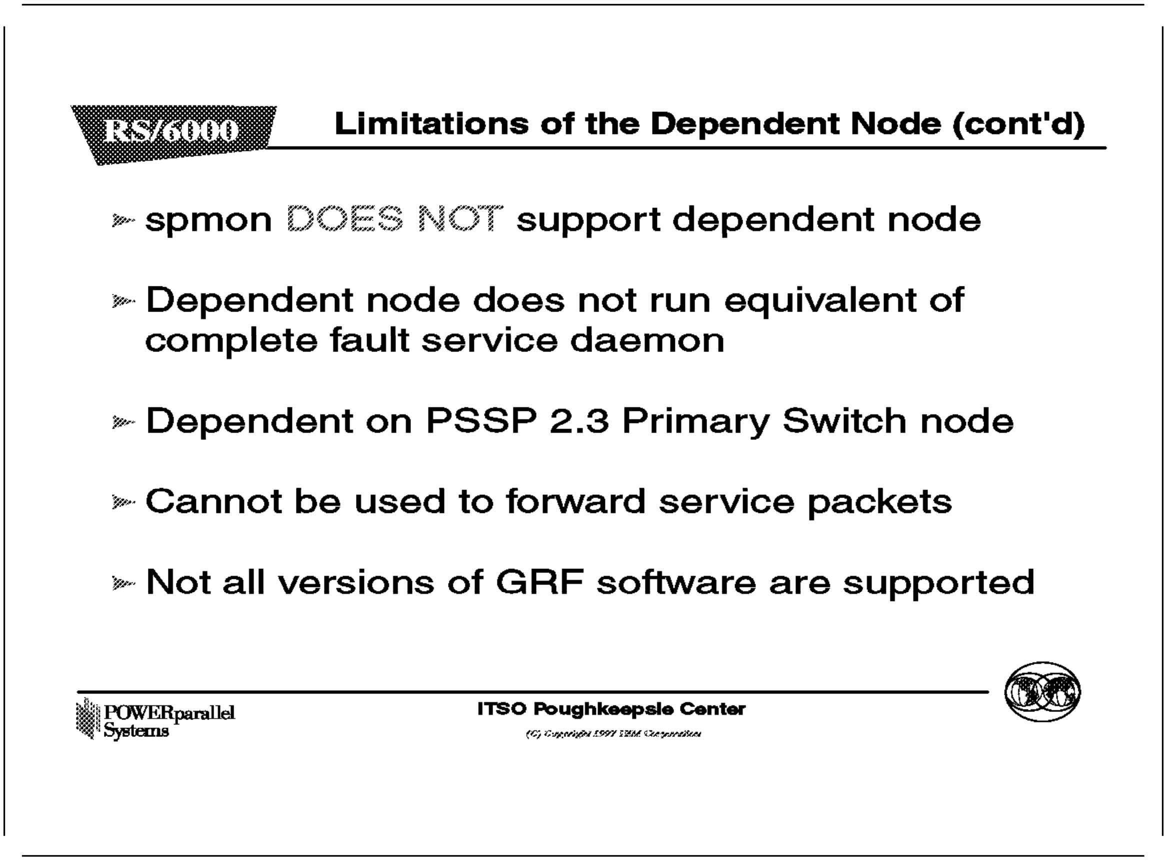
To use the dependent node in an RS/6000 SP requires the SP Extension Node SNMP Manager to be installed in the Control Workstation. The SP Extension Node SNMP Manager requires UDP port 162 in the Control Workstation. Other SNMP managers, such as Netview, also require this port. To allow the two to coexist, the SP Extension Node SNMP Manager must use an alternative udp port. This process is documented in the Installation section.

The standard cable provided to attach the GRF to the RS/6000 SP is only 10 m long. This means that the GRF is within 10 m of the RS/6000 SP, which may not be far enough for some customers. An alternative to provide a longer cable is available through RPQ. The drawback of the longer cable is that it cannot be wrap tested to see if it is faulty.

The GRF only supports TCP/IP routing. Thus, the dependent node does not support any other protocols such as, SNA or user space, commonly associated with the RS/6000 SP.

Dependent nodes are not allowed in Node Groups.

Only the 8-port and 16-port SP Switch is supported. The 8-port and 16-port High Performance Switch, the old SP switch, cannot be connected to the SP Switch Router Adapter on the GRF.



RS/6000

## Limitations of the Dependent Node (cont'd)

- spmon DOES NOT support dependent node
- Dependent node does not run equivalent of complete fault service daemon
- Dependent on PSSP 2.3 Primary Switch node
- Cannot be used to forward service packets
- Not all versions of GRF software are supported

POWERparallel Systems

ITSO Poughkeepsie Center

The spmon command on the RS/6000 SP is not enhanced to support dependent nodes. Dependent nodes can only be viewed from the perspectives command.

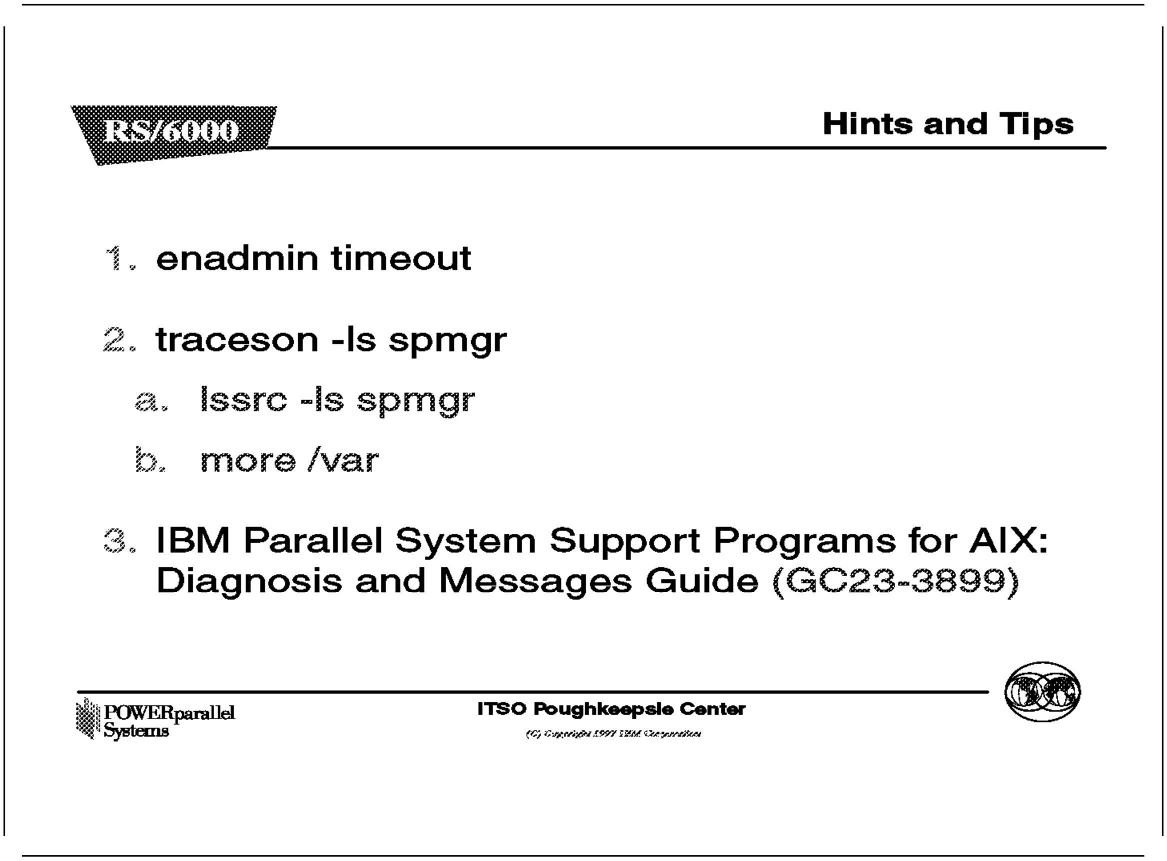
The fault service daemon runs on all switch nodes in the RS/6000 SP, but not on the dependent node. As such, the dependent node does not have the full functionality of a normal RS/6000 SP Switch node.

The dependent node requires the SP Switch′s Primary node to compute its switch routes. If the Primary node is not at PSSP 2.3, the dependent node cannot work with the RS/6000 SP.

In the RS/6000 SP, SP Switch nodes occasionally send service packets from one node to the next to keep track of status and links. Sometimes these packets are sent indirectly through another switch node. As the dependent node is not a standard RS/6000 SP Switch node, it cannot be used to forward service packets to other nodes.

The SP Switch Router comes preloaded with its operating system. To do an upgrade, users will have to download the latest level from the IBM FTP server used for 9077 support. At the time this publication is beig written, that server is expected to be service2.boulder.ibm.com. IBM will provide service updates and new levels of the SP Switch Router software on that server. The only GRF software supported on the SP Switch Router will be those versions that are provided by the IBM FTP server.

## 7.7 Hints and Tips



RS/6000

**Hints and Tips**

1. enadmin timeout
2. traceson -ls spmgr
   a. lssrc -ls spmgr
   b. more /var
3. IBM Parallel System Support Programs for AIX: Diagnosis and Messages Guide (GC23-3899)

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

When installing the dependent node, it is recommended that you turn on tracing for the SP Extension Node SNMP Manager so that valuable information will be available in the snmp log file should the installation fail. To turn on tracing, either specify the `-l` or `-s` flag on the `startsrc` command when the spmgr subsystem is started. Alternatively, use the `traceson -ls spmgr` command if tracing was not specified when the spmgr subsystem was started. To turn it off, use `tracesoff -s spmgr`.

If the output of `enadmin`, `endefnode -r`, or `endefadapter -r` shows a timeout, check the trace file /var/adm/SPlogs/spmgr/spmgrd.log for messages shown in Table 14, to perform the corresponding recovery action.

Table 14 (Page 1 of 2). SNMP Trace File Messages

| Symptom | Recovery |
|---|---|
| `init_io failed: udp port in use.` | If you find this message, then port 162 in the Control Workstation is already in use. Change the `spmgr-trap` port number in `/etc/services` in the Control Workstation, and `/etc/snmpd.conf` in the GRF. |

Table 14 (Page 2 of 2). SNMP Trace File Messages

| Symptom | Recovery |
|---|---|
| 2536-007 An authentication failure notification was received from an SNMP Agent running on the host <hostname> which supports Dependent Nodes. | The SNMP community name in the DependentNode and the GRF do not match. Correct it in the DependentNode using endefnode, or on the GRF by editing the /etc/snmpd.con file. |
| No authentication error message in the trace file. | Correct the dependent node's management_agent_hostname in the DependentNode class by using endefnode. |

Using the command lssrc -ls spmgr, check for the message switchInfoNeeded trap message is not being received. If that is the case, check the IP address of the Control Workstation in the /etc/snmpd.conf file in the GRF. Correct the address and restart the snmp daemon in the GRF.

If lssrc -ls spmgr produces the switchInfoNeeded trap message is being received but not being processed message, check the snmp trace file on the Control Workstation. Table 15 shows the messages found in the trace file and the corresponding recovery action.

Table 15. SNMP Trace File Messages

| Symptom | Recovery |
|---|---|
| Dependent node <ext_id> managed by the SNMP agent on host <CWS hostname> is not configured in the SDR - switchInfoNeeded trap ignored. | Either the wrong dependent node <ext_id> (slot number for the SP Switch Router Adapter in the GRF) or the wrong management_agent_hostname is placed in the DependentNode class. Correct the attributes and check using lssrc -ls spmgr. |
| SDR attribute <attr> in class <class> for dependent node <id> has a null value for SNMP Agent on host <hostname>. | Required attribute value is missing either in the DependentNode or in the DependentAdapter class. Add the missing attributes and check using lssrc -ls spmgr. |
| SDRGetAllObjects() DependentAdapter failed with return code 4. | Same as the previous recovery. |
| Dependent node <ext_id> managed by the SNMP agent on host <hostname> is configured with a bad community name-switchInfoNeeded trap ignored. | The SNMP community names in the DependentAdapter and the GRF do not match. Correct it in the DependentAdapter using endefnode, or on the GRF by editing the /etc/snmpd.conf file. |

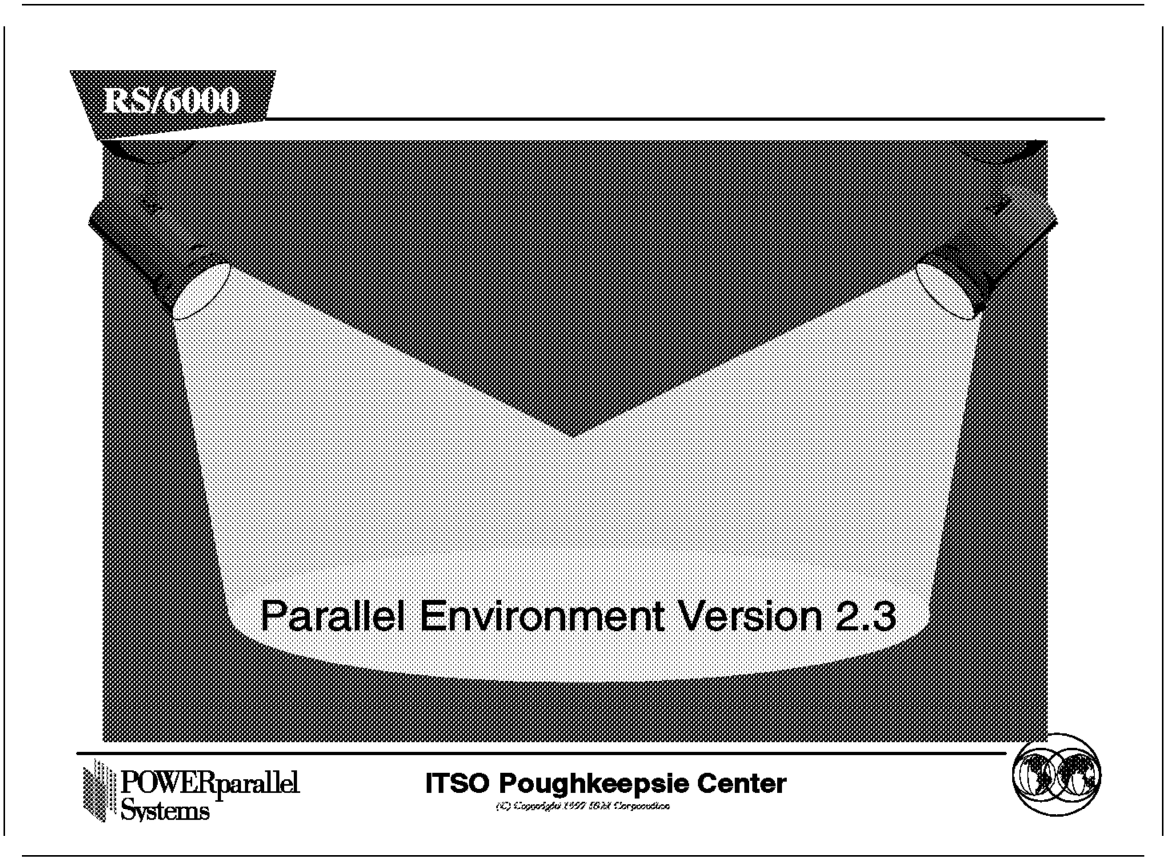
If none of the preceding recovery methods solves the problem, refer to *IBM Parallel System Support Programs for AIX: Diagnosis and Messages Guide*, GC23-3899. Check the *Symptom and Recovery* table in the Diagnosing Switch Problems chapter of GC23-3899 for the proper action. Following is a list of suggestions for performing the recovery:

- If the recovery action is Verify Secondary Nodes, and the failing node is a dependent node, then enter SDRGetObjects switch_responds and check the adapter_config_status of the dependent node. If it is not css_ready, then continue with the following steps.
- Enter SDRGetObjects DependentNode to verify the attributes of the dependent node.
- Login to the GRF to verify the SP Switch Router Adapter attributes by issuing the following command (assume that the SP Switch Router Adapter is in slot 2 of the GRF):

```
# grrmb
GR 66> port 2
GR 02> maint 3
GR 00> [RX]
[RX] Configuration Parameters:
[RX]    Slot Number..........: 2
[RX]    Node Number..........: 7
[RX]    Node Name............: 02
[RX]    SW Token.............: 0001000602
[RX]    Arp Enabled..........: 2
[RX]    SW Node Number.......: 6
[RX]    IP...................: 0x81282f47
[RX]    IP Mask..............: 0xffffffc0
[RX]    Alias IP.............: 0x81283047
[RX]    Max Link Size........: 1024
[RX]    Host Offset..........: 1
[RX]    Config State.........: 1
[RX]    System Name..........: ceedgate
[RX]    Node State...........: 2
[RX]    Switch Link Chip.....: 2
[RX]    Transmit Delay.......: 31
```

- Verify that the SNMP community name in the /etc/snmpd.conf file on the GRF is the same as that in the CWS.
- When all the preceding items are verified, issue an Eunfence to add the dependent node to the RS/6000 SP.

# Chapter 8. Parallel Environment Version 2.3



RS/6000

Parallel Environment Version 2.3

POWERparallel Systems

ITSO Poughkeepsie Center

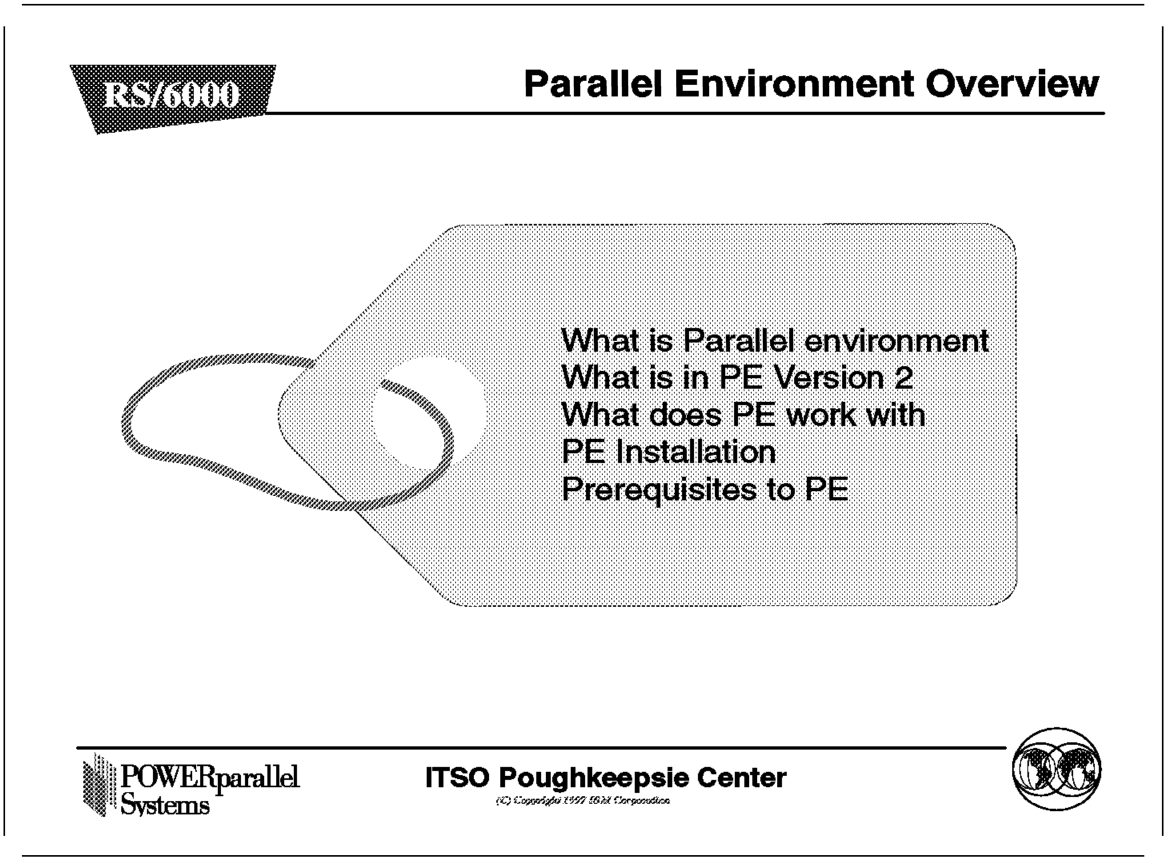
(C) Copyright 1997 IBM Corporation

Parallel Environment (PE) Version 2.3 is the follow-on of Version 2.2.

In this chapter, we present the Parallel Environment architecture, the global concepts, and the enhancements from the previous version. Further, the installation process is discussed. We also introduce the different tools of PE and how to use them to succeed with the different steps of program execution: compilation, execution, tuning and monitoring.

The Message Passing Libraries MPI and LAPI are not covered in this chapter.

## 8.1 Overview



RS/6000

**Parallel Environment Overview**

What is Parallel environment
What is in PE Version 2
What does PE work with
PE Installation
Prerequisites to PE

POWERparallel Systems

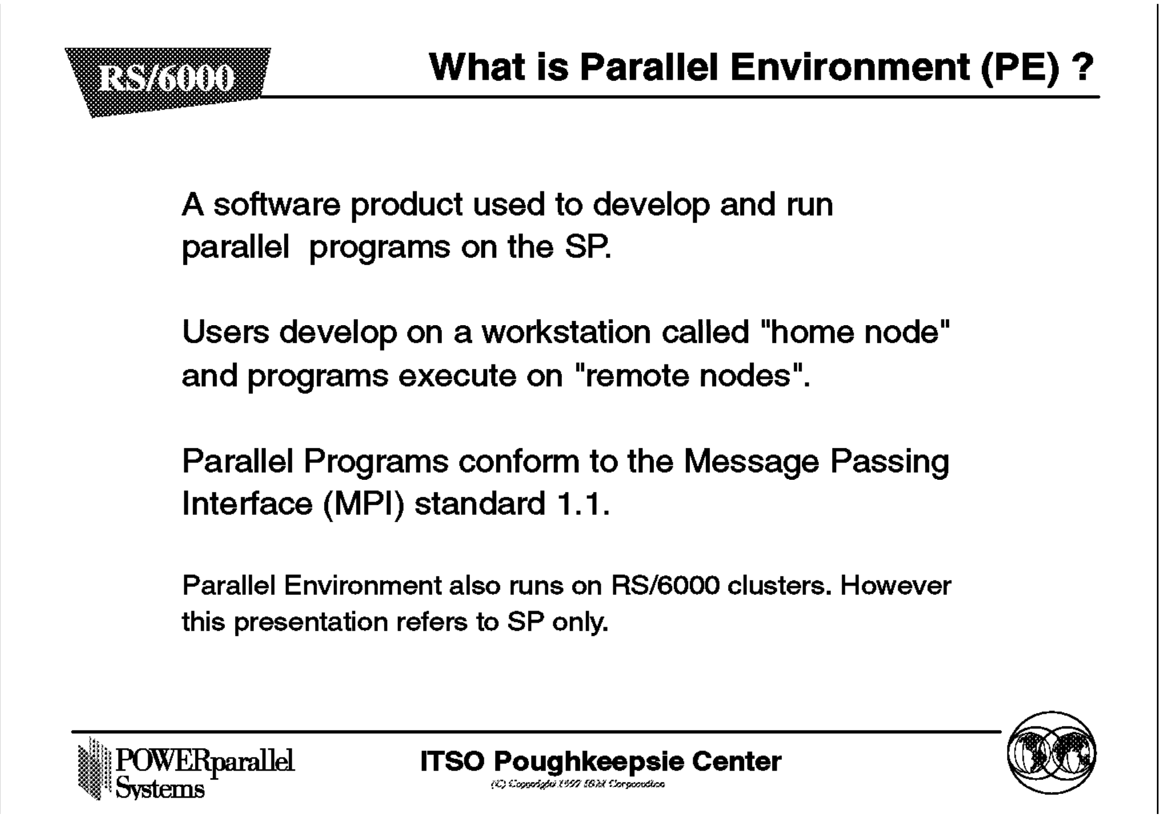
**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

This section presents a global overview of the Parallel Environment: the general concept, the architecture and terminology.

We will introduce the different components, prerequisites and related software, considerations about coexistence and migration.

### 8.1.1 What Is the Parallel Environment (PE)?



RS/6000

**What is Parallel Environment (PE) ?**

A software product used to develop and run parallel programs on the SP.

Users develop on a workstation called "home node" and programs execute on "remote nodes".

Parallel Programs conform to the Message Passing Interface (MPI) standard 1.1.

Parallel Environment also runs on RS/6000 clusters. However this presentation refers to SP only.

POWERparallel Systems

ITSO Poughkeepsie Center

Parallel applications developed for a distributed system are often run under two different major paradigms: shared memory or explicit message passing.

- On a shared memory system, the programmer is able to read/write shared variables among distributed processes running across the system.
- Using the message passing libraries, information is passed between processes on the parallel system via messages communicated in send/receive pairs. Message passing is frequently used on distributed memory parallel systems and workstation clusters.

IBM Parallel Environment is an environment designed for the development and execution of parallel programs on a distributed memory model. Parallel Environment can run on:

- Workstation clusters
- Any configuration of RS/6000 SP
- A mixed system where additional RS/6000 workstations supplement the processors of an RS/6000 SP.

Parallel applications can be developed on any RS/6000 that has PE installed and executed on a network cluster or SP system.

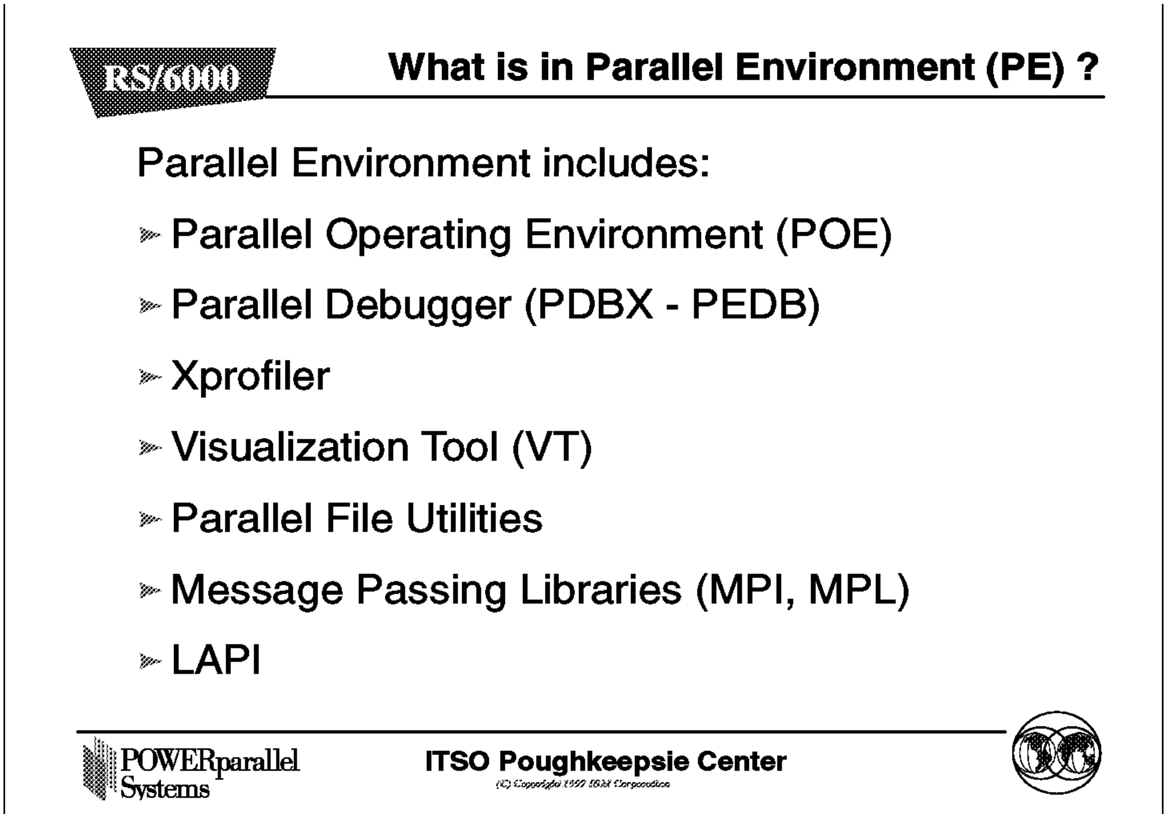
However, Parallel Environment has been optimized to take full advantage of the RS/6000 SP flexible architecture and High Performance Switch (HPS and SP Switch). This presentation refers to RS/6000 SP only.

The Message Passing Interface (MPI) complies with Message Passing Interface standard 1.1.

The Parallel Environment supports the two basic parallel programming models, the Single Program Multiple Data (SPMD) and the Multiple Program Multiple Data (MPMD):

- In the SPMD model, the programs running the parallel tasks of your partition are identical. The tasks, however, work on different sets of data.
- In the MPMD model, each node may run a different program. A typical example of this is the master/slave program. One task (the master) coordinates the execution of the other tasks (the slaves).

### 8.1.2 What Is in Parallel Environment (PE)?



RS/6000

What is in Parallel Environment (PE) ?

Parallel Environment includes:

- Parallel Operating Environment (POE)
- Parallel Debugger (PDBX - PEDB)
- Xprofiler
- Visualization Tool (VT)
- Parallel File Utilities
- Message Passing Libraries (MPI, MPL)
- LAPI

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Parallel Environment includes different components and tools for developing, executing, debugging and profiling parallel programs.

- Parallel Operating Environment (POE)

  POE is a set of tools to compile and run parallel programs from the home node. It is an execution environment designed to hide, or at least smooths, the differences between serial and parallel execution. With POE, you invoke the parallel program from your home node and run its parallel tasks to the remote nodes.

- Visualization Tool (VT)

  This tool consists of a trace generation facility and a trace display system that allow you to visualize performance characteristics of the program and system. VT can be used as an online monitor (for performance monitoring) or to play back traces recorded during a program execution (for trace visualization).

- Parallel debugger

  This tool extends the interface and subcommands of the AIX debuggers. Some subcommands have been modified for use on parallel programs. Command line and X-windows interfaces are both available.

- Xprofiler

Xprofiler extends the usability of the AIX `gprof` command for use on parallel programs.

- Message Passing Interfaces (MPI)

  There are two libraries for MPI:

  - Signal handling, which uses AIX signals and signal handlers
  - Threaded, which uses POSIX threads

  All tasks of a program must use either signal handling or threaded calls, but not a combination of each.

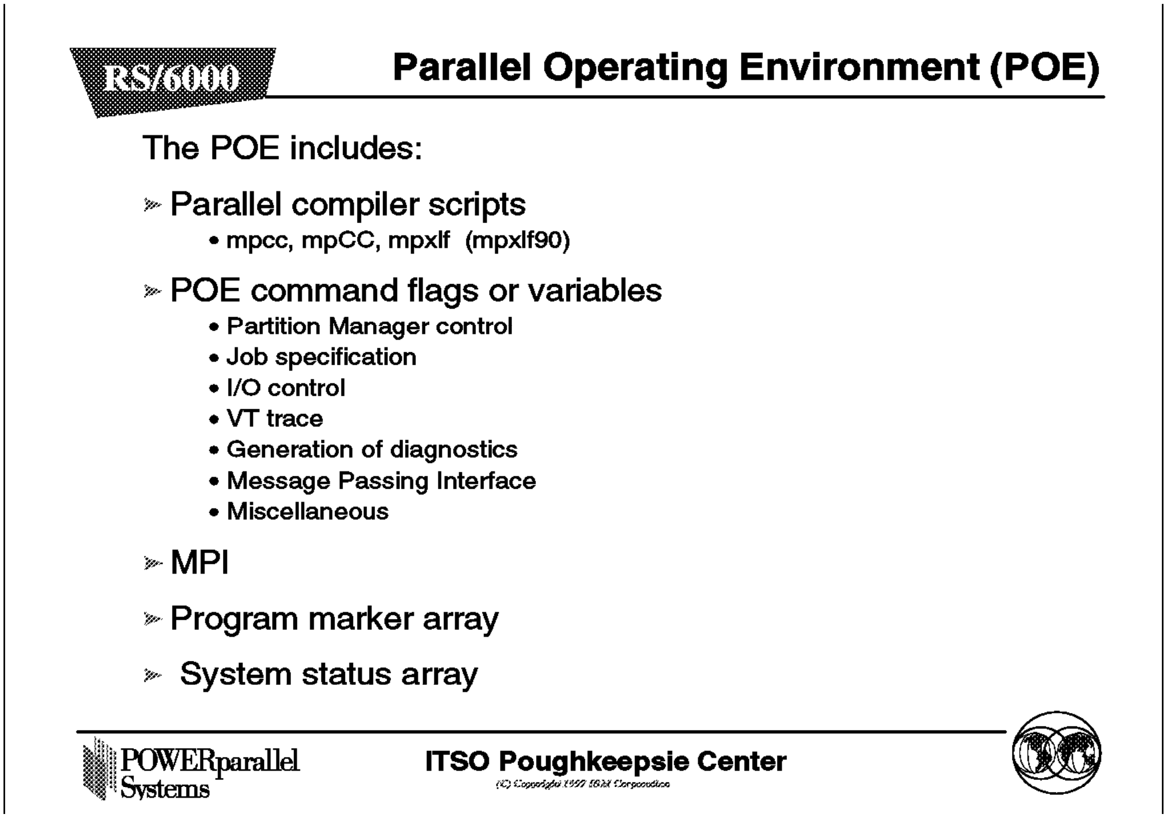
  The Message Passing Library (MPL) is still delivered for compatibility purposes and uses signal handling.

  **Note:** MPL does not support threaded calls.

- LAPI

  LAPI is another kind of message passing interface. This library is not based on standards and is delivered with the PSSP code. LAPI stands for Lowlevel Application Programming Interface.

### 8.1.3 Parallel Operating Environment (POE)



RS/6000

**Parallel Operating Environment (POE)**

The POE includes:

- Parallel compiler scripts
  - mpcc, mpCC, mpxlf (mpxlf90)
- POE command flags or variables
  - Partition Manager control
  - Job specification
  - I/O control
  - VT trace
  - Generation of diagnostics
  - Message Passing Interface
  - Miscellaneous
- MPI
- Program marker array
- System status array

POWERparallel
Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The *parallel compiler scripts* are shell scripts that call the corresponding language compilers while also linking in an interface library to enable communication between your home node and the parallel tasks running on the remote nodes. You dynamically link in a communication subsystem implementation when you invoke the executable.

The poe command invokes the Parallel Operating Environment for loading and executing programs on remote nodes. The operation of POE is influenced by more than 40 environment variables. The flag options on this command are each used to temporarily override these environment variables.

The environment variables and flags that influence the POE command fall into these categories:

- Partition manager control
  The environment variables and flags in this category determine the method of node allocation, message passing mechanism, and the PULSE monitor function.

- Job specification
  The environment variables and flags in this category determine whether or not the partition manager should maintain the partition for multiple job steps, whether commands should be read from a file or STDIN, and how the partition should be loaded.

- I/O control
  The environment variables and flags in this category determine how I/O from the parallel tasks should be handled. These environment variables and flags set the input and output modes, and determine whether or not output is labeled by task id.

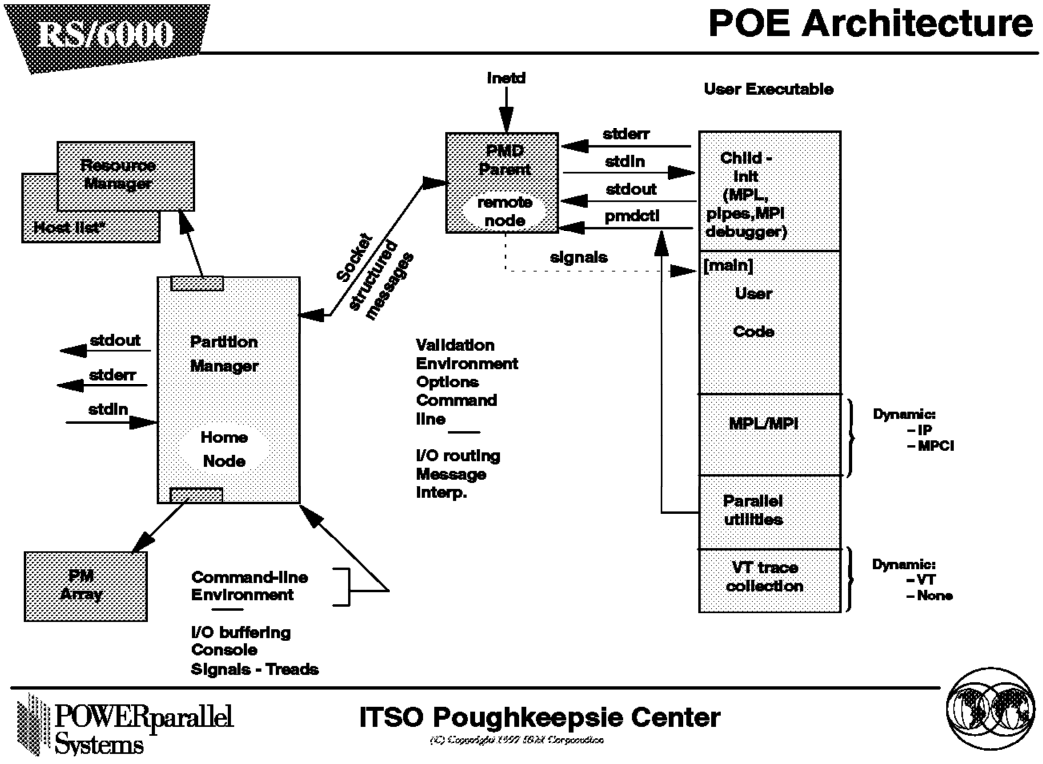
- VT trace collection
  The environment variables and flags in this category determine if and how execution traces are collected for playback using the visualization tool (vt). They determine which type of trace are collected, and how trace storage is handled.

- Generation of diagnostic information
  The environment variables and flags in this category enable you to generate diagnostic information that may be required by the IBM Support Center in resolving PE-related problems.

- Message Passing Interface
  The environment variables and flags in this category enable you to specify values for tuning message passing applications.

- Miscellaneous
  The environment variables and flags in this category enable additional error checking, and set a dispatch priority class for execution.

### 8.1.4 POE Architecture



Application developers compile and run programs from their *home node* using the *Parallel Operating Environment*. The home node can be an RS/6000 SP node or any workstation on the LAN that has PE installed.

With the Parallel Operating Environment, a parallel program is invoked from the home node and runs its parallel tasks on a number of *remote nodes*. The group of parallel tasks is called a *partition*. The user program and data must be accessible by all the remote nodes where the parallel tasks are executed.

When you invoke a program on your home node, POE starts your *partition manager* which allocates the nodes of your partition and initializes the local execution environment for remote tasks. A copy of the partition manager daemon (PMD) is run on each remote node and forks to the user′s executable to initialize the environment. The PMD on the remote nodes is invoked by `inetd` and has entries in /etc/services.

Although you are running tasks on remote nodes, POE allows you to continue using traditional AIX I/O techniques and commands.

PMD manages distribution or collection of standard input (STDIN), standard output (STDOUT), and standard error (STDERR). The Partition Managers communicate with the Socket Structured Messages (SSM). SSM control in and SSM control out are used to exchange the messages of STDIN, STDOUT, and

STDERR by the way of sockets. The Partition Manager daemon puts or discards the header of the SSM.

Depending on the SPMD or MPMD programming model use, you can redirect input, output, pipes, or use shell tools to:

- Determine whether a single task or all parallel tasks should receive input from STDIN.
- Determine whether a single task or all parallel tasks should write to STDOUT. If all tasks are writing to STDOUT, it may be useful to specify that messages be ordered by task ids.
- Specify the level of messages reported to STDOUT.
- Specify that messages to STDOUT and STDERR should be labeled by task ids.

In some cases, depending on the way the redirection are used, I/O buffering should be done with environment variables such as MP_STDINMODE and MP_HOLD_STDIN.

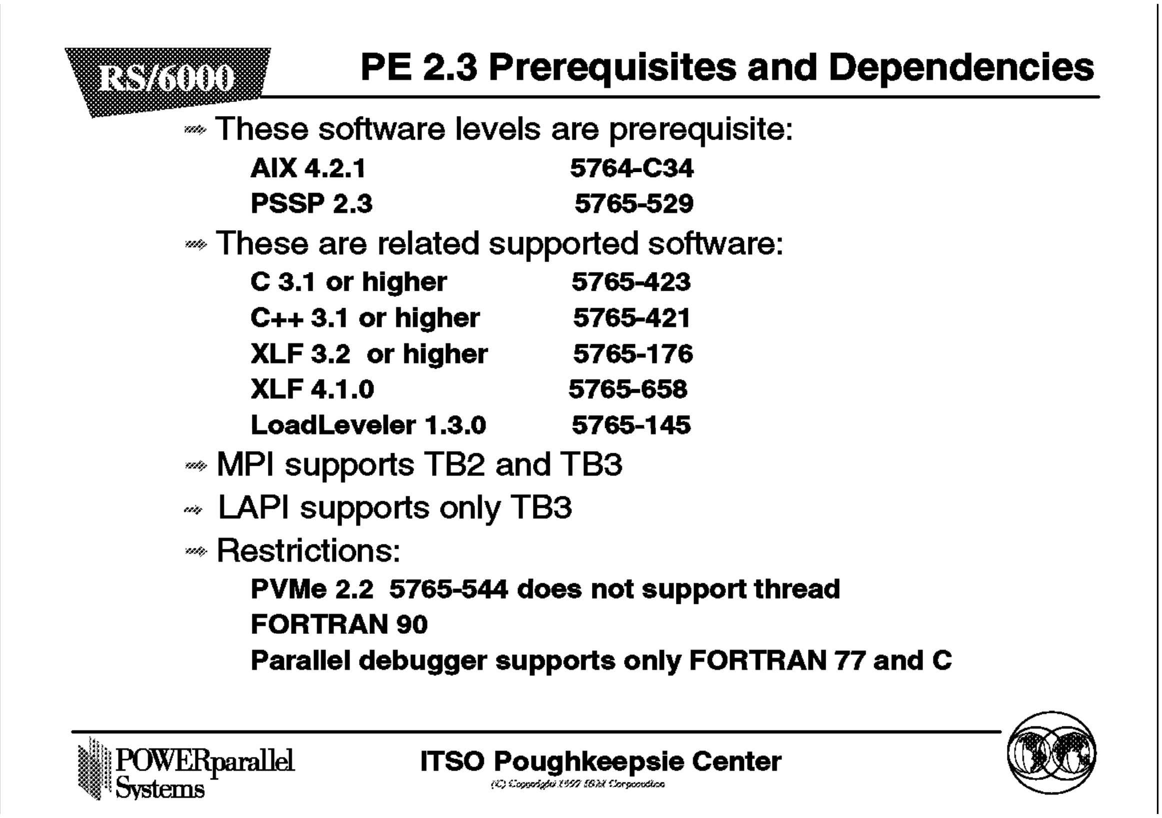
Writes to STDOUT can be synchronous or asynchronous in conjunction with the variable MP_STDOUTMODE.

Depending on your hardware, configuration, or specific need, the partition manager uses a *host list file*, or the System Resource Manager, or both, to allocate nodes. The Parallel Operating Environment does not realize any allocation. The choice of IP or User Space protocol is dynamic and can be set with the MP_EUILIB variable or overridden at POE invocation with `poe -euilib`.

**Note:** The User Space protocol needs the SP switch.

The Visualization Tool trace activation is dynamic and can be set with the variable MP_TRACELEVEL or overridden at POE invocation, `poe -tracelevel` or `-tlevel`.

### 8.1.5 PE 2.3 Prerequisites and Dependencies



RS/6000

**PE 2.3 Prerequisites and Dependencies**

- These software levels are prerequisite:

| | |
|---|---|
| **AIX 4.2.1** | **5764-C34** |
| **PSSP 2.3** | **5765-529** |

- These are related supported software:

| | |
|---|---|
| **C 3.1 or higher** | **5765-423** |
| **C++ 3.1 or higher** | **5765-421** |
| **XLF 3.2 or higher** | **5765-176** |
| **XLF 4.1.0** | **5765-658** |
| **LoadLeveler 1.3.0** | **5765-145** |

- MPI supports TB2 and TB3
- LAPI supports only TB3
- Restrictions:

**PVMe 2.2 5765-544 does not support thread**
**FORTRAN 90**
**Parallel debugger supports only FORTRAN 77 and C**

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

- Parallel Environment Version 2 Release 3 commands and applications are compatible with AIX Version 4.2.1 only, not with earlier versions. The different versions and releases of the Parallel Environment and their related software that are supported are:

Table 16 (Page 1 of 2). Prerequites and Related Software

| PSSP Version | Related Software |
|---|---|
| PSSP 1.1 5765-296 | AIX 3.2.4<br>LL 1.2 5765-145<br>PE 1.2 5765-144<br>PVMe 1.3 5765-246<br>High Performance Switch (HPS) |
| PSSP 1.2 5765-296 | AIX 3.2.4 - 3.2.5<br>LL 1.2 5765-145<br>PE 1.2.1 5765-144<br>PVMe 1.3.1 5765-246<br>High Performance Switch (HPS)<br>SP Switch |

Table 16 (Page 2 of 2). Prerequites and Related Software

| PSSP Version | Related Software |
|---|---|
| PSSP 2.1 5765-529 | AIX 4.1.4<br>LL 1.2.1 5765-145<br>PE 2.1 5765-543<br>PVMe 2.1 1 5765-544<br>C 3.1 or later 5765-423<br>C++ 3.1 or later 5765-421<br>XLF 3.2 or later 5765-526<br>High Performance Switch (HPS)<br>SP Switch |
| PSSP 2.2 5765-529 | AIX 4.1.4 - 4.1.5 - 4.2<br>LL 1.3 5765-145<br>PE 2.2 5765-543<br>PVMe 2.2 5765-544<br>C 3.1 or later 5765-423<br>C++ 3.1 or later 5765-421<br>XLF 3.2 or later 5765-526<br>High Performance Switch (HPS)<br>SP Switch |

- Incompatibilities exist between Fortran 90 and MPI which may affect the ability to use these programs. For more information on the restrictions and implications of using MPI and Fortran 90, refer to the /usr/lpp/ppe.poe/samples/mpif90/README.mpif90 file after POE is installed.

  The mpxlf90 script is delivered as a *sample*. If you want to enable it, you need to perform the following steps:

  1. Copy the mpxlf90 script file:

     cp /usr/lpp/ppe.poe/samples/mpif90/mpxlf90 to /usr/lpp/ppe.poe/bin

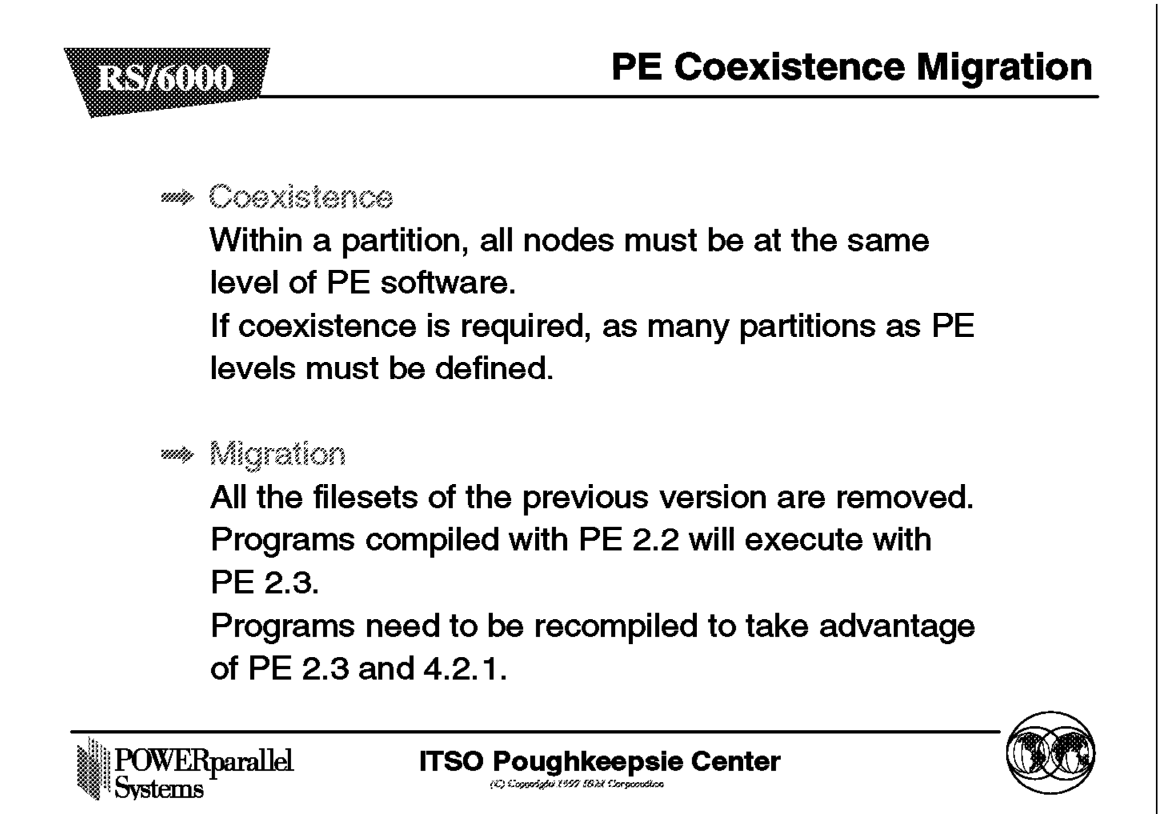
  2. Create a symbolink link to /usr/bin:

     ln -s /usr/bin/mpxlf90 /usr/lpp/ppe.poe/bin/mplf90

  3. Copy the mpilf90.h header file:

     cp /usr/lpp/ppe.poe/samples/mpif90/mpif90 to /usr/lpp/ppe.poe/bin

### 8.1.6 PE Coexistence Migration



RS/6000

**PE Coexistence Migration**

- Coexistence

  Within a partition, all nodes must be at the same level of PE software.

  If coexistence is required, as many partitions as PE levels must be defined.

- Migration

  All the filesets of the previous version are removed.

  Programs compiled with PE 2.2 will execute with PE 2.3.

  Programs need to be recompiled to take advantage of PE 2.3 and 4.2.1.

POWERparallel Systems
ITSO Poughkeepsie Center

Different PE software levels cannot coexist within a partition. The issues concern:

- The PMD, although the PMD′s use the same port number, it has been modified with the PE Version 2.3. Different PMD levels cannot coexist within the same partition.
- Different Job Manager Versions cannot coexist in a same partition. You must define as many partitions as you have different Job Manager levels.
- The Job Manager cannot span across partitions. You must define a Job Manager for each partition. Each partition must have its own Job Manager configuration file /etc/jmd_config.syspar_name.

Partitions have to be defined according the rules of partitioning.

Migration

**Important**

Although a migration from any PE Version to PE Version 2.3 will completely override the earlier fileset, *it is strongly recommended to uninstall the oldfileset.* This will reduce the chance for confusion over old fileset, path name, executable, etc. Use the `PEdeinstallSP` command to uninstall all the PE filesets.

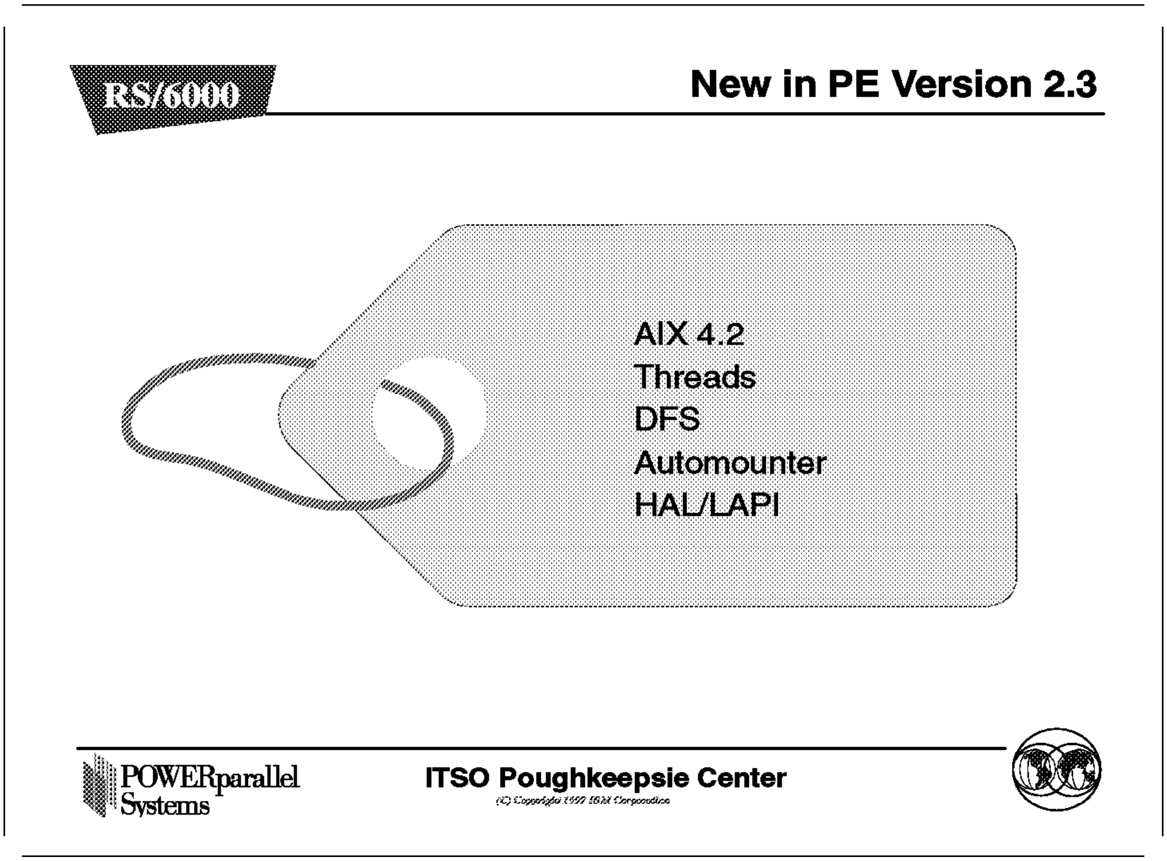
In the case system administrator modified some scripts or configuration files, it may be desirable to save them and redo the modifications on the new version. The modified files could be as follows:

- The compiler scripts
- The configuration file /usr/lpp/ppe.poe/lib/poe.cfg
- The amd script `mpamddir`

PE version 2.3 maintains a binary compatibility with executables compiled with previous versions. There is no need to maintain previous versions of libc or MPI libraries.

- Programs compiled with PE Version 2.2 will execute with PE Version 2.3.
- Programs compiled with PE Version 2.1 will execute with PE Version 2.2 and PE Version 2.3.
- Programs compiled with PE Version 1.2 need to be recompiled with PE Version 2.3

## 8.2 New in PE Version 2.3



RS/6000

New in PE Version 2.3

AIX 4.2
Threads
DFS
Automounter
HAL/LAPI

POWERparallel Systems

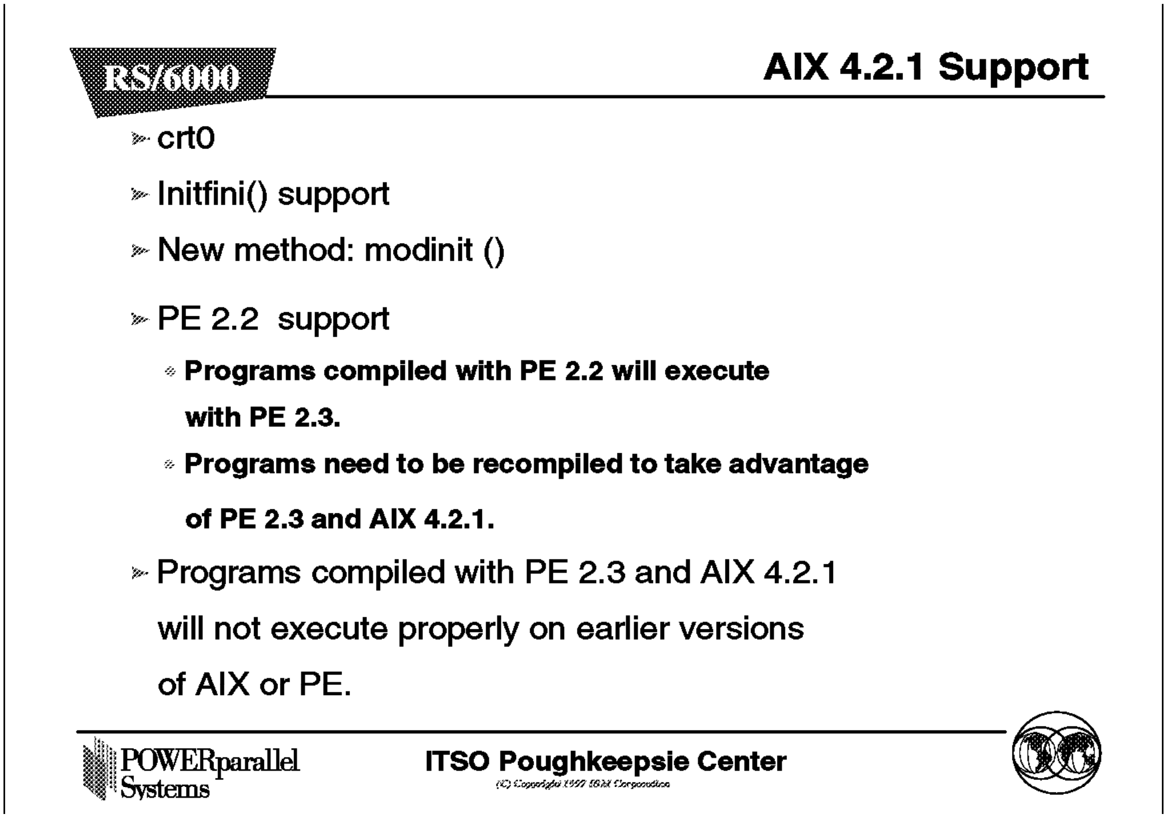
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The major enhancements of this release concern the support of:

- AIX 4.2.1
- Threaded libraries
- Distributed File System (DFS)

## 8.2.1 AIX 4.2.1 Support



RS/6000

**AIX 4.2.1 Support**

- crt0
- Initfini() support
- New method: modinit ()
- PE 2.2 support
  - **Programs compiled with PE 2.2 will execute with PE 2.3.**
  - **Programs need to be recompiled to take advantage of PE 2.3 and AIX 4.2.1.**
- Programs compiled with PE 2.3 and AIX 4.2.1 will not execute properly on earlier versions of AIX or PE.

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

Parallel Environment Version 2.3 supports all the features of AIX 4.2.1 such as files greater then 2GB and some compiler′s features and options:

- crt0

  The binder option initfini allows you to specify some specific initialization routines outside crt0. POE no longer needs to provide its own versions of crt0. It uses those delivered by AIX.

  The crt0 routine is mainly used for:

  – The initialization of the registers
  – Some AIX setup
  – The call to main()

  The object crt0.o is delivered with AIX in /lib and statically bound in the user program. In previous versions, PE delivered its own versions of crt0 in /usr/lpp/poe/lib for MPI specific initialization. This can lead to major issues for compatibility and migration in the following situations:

  – Each time crt0.0 changes in AIX
  – Each time PE changes its own version of crt0

- The C++ language still needs some specific initialization before calling main(). It delivers its own crt0.o versions in /usr/lpp/xlC/lib.
- Binder, initfini option

The AIX Version 4.2 brings a new binder′s option, the initfini, which allows a user to specify a routine that will be executed before main. This routine will be called poe_remote_main(). The routine obtains the values of argc and argv from crt0 and passes them to mp_main(), which then initializes the POE remote child.

The routine is bound to the users executables at compile time. All the POE′s compilation scripts utilize the initfini binder option with the following flag: _FLAGS=″-I$_INCLUDE -binitfini:poe_remote_main ″, so you do not have to specify it.

This binder only applies to AIX Version 4.2 or later and is called as follows: cc -b initfini

initfini: [initial] [:termination] [:priority]

specifies a module initialization and termination function for a module, where initial is an initialization routine, termination is a termination routine and priority is a signed integer, with values from -2,147,483,648 to 2,147,483,647. You must specify at least one value of initial and termination. If you omit both termination and priority, you must omit the colon after initial. If you do not specify priority, 0 is the default. This option only applies to AIX Version 4.2 or later. This option sorts routines by priority, starting with the routine with the smallest priority. It invokes initialization routines in order, and termination routines in reverse order.

**Note:**

- This option invokes routines with the same priority in an unspecified order, but it preserves the relative order of initialization and termination routines. For example, if you specify initfini:i1:f1 and initfini:i2:f2 and i1 is invoked before i2 in an unspecified order, f2 will be called before f1 when the module is unloaded.
- The poe_remote_main() call is a part of the pm_initfini.o object in the libmpi.a library.

- Call modinit()

  With AIX Version 4, the compilers have been modified to support the initfini binder option with the modinit() call in the libc.a and libc_r.a libraries. The POE versions of libc.a and libc_r.a hold this call.

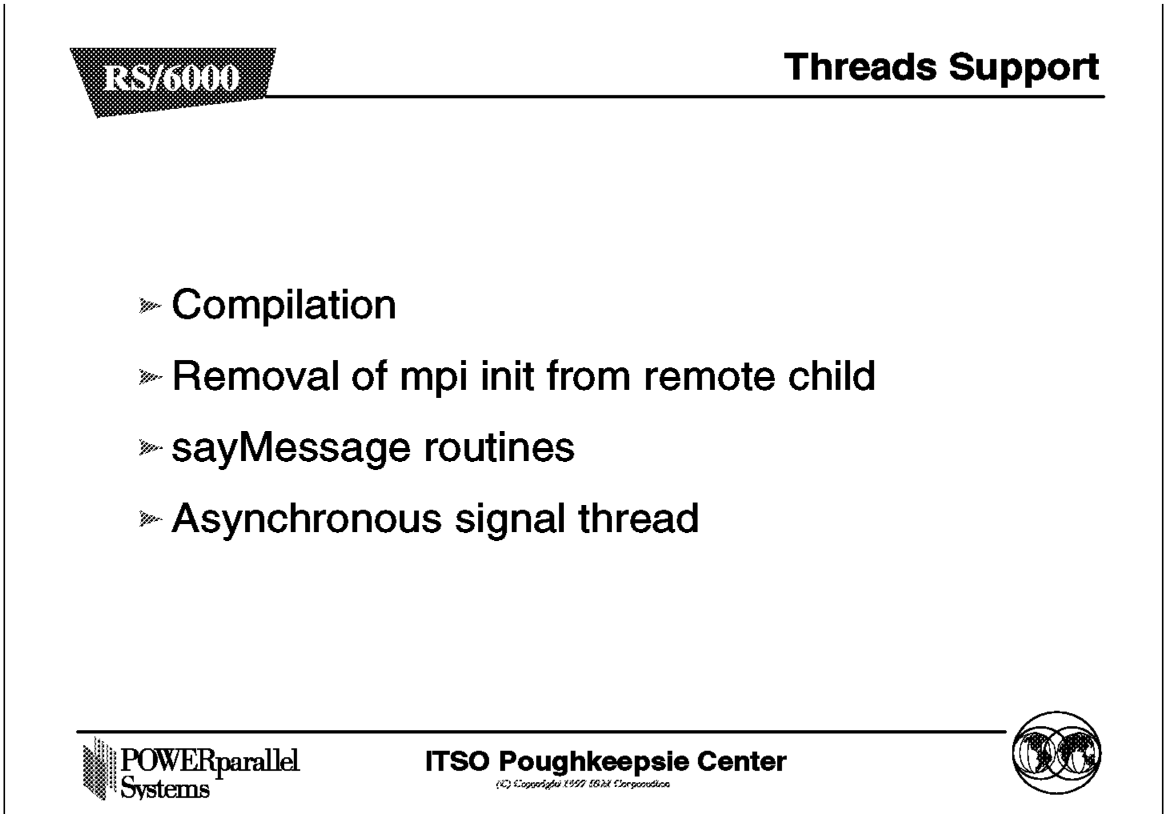
- PE Version 2.2 support

  The compatibility is maintained for the programs compiled with PE Version 2.2. These programs will execute properly with PE 2.3.

  However, to take advantage of PE version 2.3 and AIX 4.2.1, the programs need to be recompiled. A simple relink is not recommended, because the change of crt0 and some calls in the libraries may lead these programs to execute improperly.

- Due to the change of crt0, the programs compiled with PE version 2.3 and AIX Version 4.2.1 will not execute properly on earlier versions.

## 8.2.2 Threads Support



RS/6000

**Threads Support**

- Compilation
- Removal of mpi init from remote child
- sayMessage routines
- Asynchronous signal thread

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

HAL and LAPI are packaged in PSSP and are available only in their threaded version.

The threaded libc_r.a library is built during installation by `makelibc` of the POE from the AIX libc_r.a library, as it is for the signal libc.a library.

PE Version 2.3, therefore, in effect, delivers support of threads and a set of threaded libraries:

```
/usr/lpp/ssp/css/lib/libhal_r.a
/usr/lpp/ssp/css/lib/liblapi_r.a
/usr/lib/libhal_r.a
/usr/lib/liblapi_r.a

/usr/lpp/ssp/css/libip/libmpci_r.a
/usr/lpp/ssp/css/libtb2/libmpci_r.a
/usr/lpp/ssp/css/libtb3/libmpci_r.a

/usr/lpp/ppe.poe/lib/ip/libmpci_r.a
/usr/lpp/ppe.poe/lib/us/libmpci_r.a

/usr/lpp/ppe.poe/lib/libmpi_r.a
/usr/lpp/ppe.poe/lib/libppe_r.a
/usr/lpp/ppe.poe/lib/libvtd_r.a
/usr/lpp/ppe.poe/lib/libc_r.a
/usr/lpp/ppe.poe/lib/profiled/libc_r.a
```

- Removal of MPI_init from remote child

  Initialization of the threaded MPI library is done at the point of invocation of the MPI_Init() call in the user′s program. In the signal library, initialization of the MPI library is done before the user′s main program. This change for the threaded library accommodates programs such as those using LAPI, which may not require the use of the MPI library.

- Compilers

  To support a threaded environment, the user′s program must be compiled using the threaded version of the compilers. POE′s compilation scripts reference the AIX compilers. POE provides the following compilation scripts to support compiling threaded POE programs:

  – mpcc_r
  – mpCC_r
  – mpxlf_r
  – mpxlf90_r

  POE provides a compiler configuration file located in /usr/lpp/ppe.poe/lib/poe.cfg. The corresponding stanzas have been added to the file:

  – cc_r
  – xlC_r
  – xlf_r
  – xlf90_r

  **Notes:**

  1. Although Fortran Version 4.1.0 does not support threads, the compiling script mpxlf_r is delivered with a corresponding stanza xlf_90 in the configuration file. They refer to the non-threaded compiler xlf. However, even with the non-threaded Fortran compiler, programs can take advantage of the threaded message passing libraries.

  2. Incompatibilities exist between Fortran 90 and MPI, which may affect the ability to use these programs. For more information on the restrictions and implications of using MPI and Fortran 90, refer to the /usr/lpp/ppe.poe/samples/mpif90/README.mpif90 file after POE is installed. Although the xlf90 and xlf90_r are present in the poe.cfg configuration file, the POE compiling scripts and the include files are delivered as samples in the /usr/lpp/ppe.poe/samples/mpif90 directory. As for Fortran, the xlf90 compiler is not available in a threaded version. The mpxlf90_r refers to the xlf90 compiler.

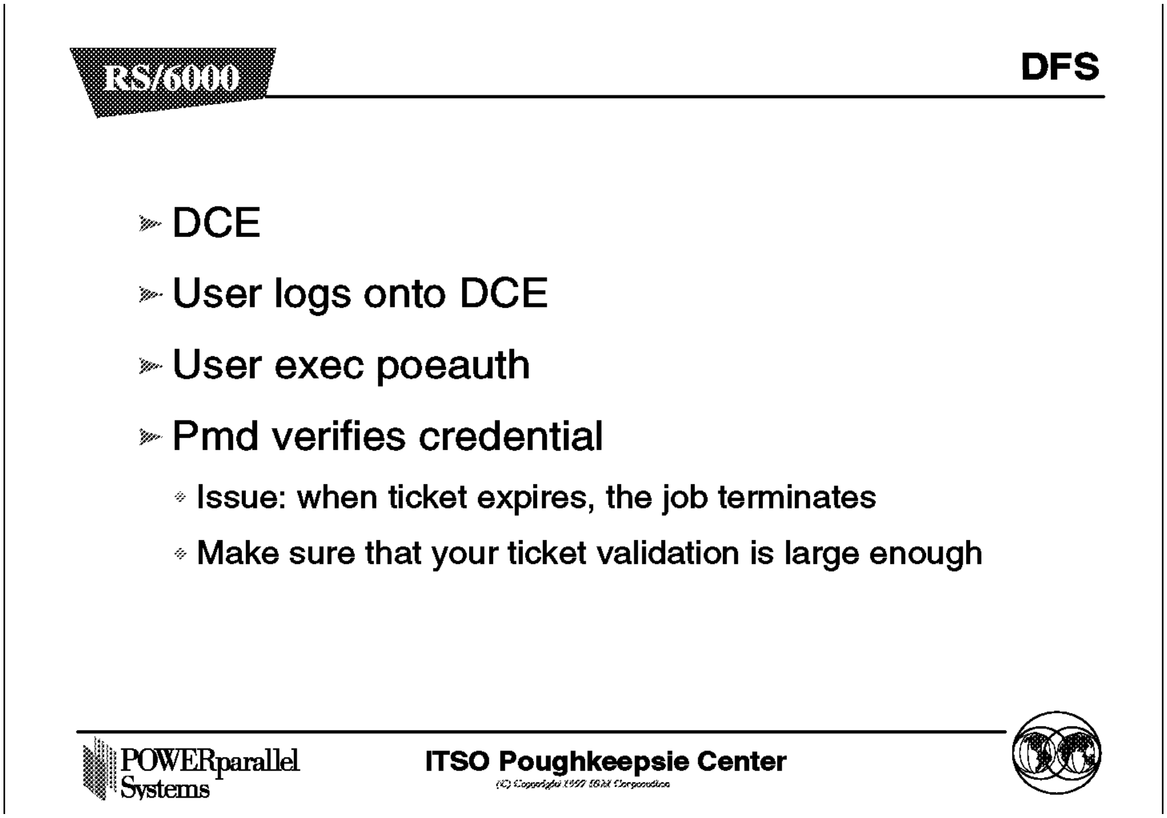
- Asynchronous signal thread

  The following asynchronous signals within the remote child (mp_main) will be handled as a separate thread:

  – SIGQUIT
  – SIGTERM
  – SIGHUP
  – SIGINT
  – SIGTSTP
  – SIGCONT

  If these signals were not handled on a separated thread, any of the threads could receive the signal. This could result in a deadlock.

## 8.2.3 DFS



RS/6000

**DFS**

- DCE
- User logs onto DCE
- User exec poeauth
- Pmd verifies credential
  - Issue: when ticket expires, the job terminates
  - Make sure that your ticket validation is large enough

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

- What is the need for DFS?

  One of the requirements of the Parallel Environment is to have the program, either for SPMD or MPMD models, and data available on the remote nodes where the program run.

  - One of the ways is to copy programs and data, either with the utilities delivered with the Parallel Operating Environment (mcp, mprcp) or with traditional AIX commands. This implies unnecessary extra work for users.
  - An other way is to make these programs and data accessible with a share filesystem:
    - With previous releases of PE, Network File System (NFS) was the only supported share filesystem. (AFS is delivered on an as-is basis in the samples directory). Although NFS used with the Automounter (AMD) is a good solution, there could be some limitations in the case of I/O oriented jobs, specially when heavy writes occur.
    - A more effective way is by using the Distributed File System (DFS), which is now supported by Parallel Environment Version 2.3.

- DCE

  The DFS support does not imply the full support of the Distributed Computing Environment (DCE). DFS is the only part supported by both POE and LoadLeveler. It uses DCE as the underlying mechanism to ensure users are authorized to access files. DCE user credentials would need to be available to all tasks of the parallel job. At the present time, there are no routines available to access the credentials with the PSSP. Full security support could be available with further releases of PSSP. So for POE and LoadLeveler, we will have to provide their own credential routines.

  The DFS use implies the following scenario with three major steps:

  - User logs into DCE.
  - User executes poeauth.
  - Pmd verifies the credential.

  These steps are detailed in the following chapter.

- There is a potential issue when ticket expires.

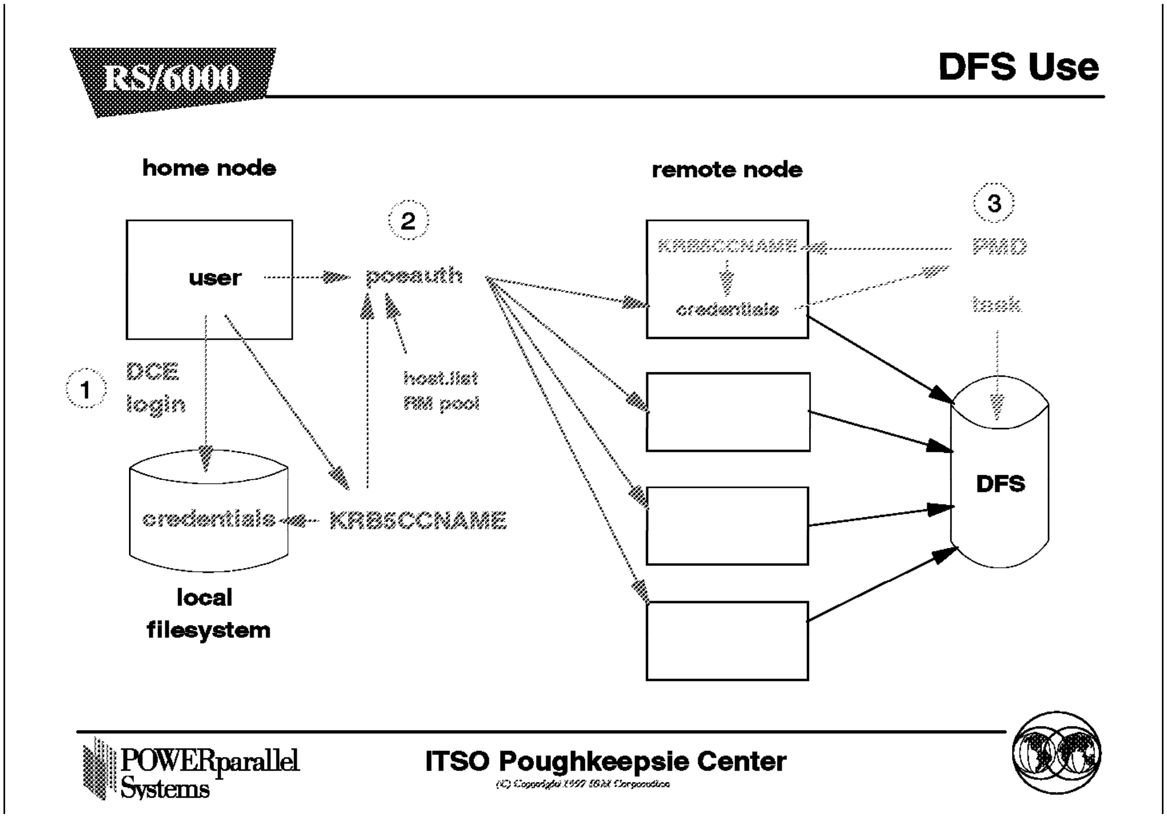
  The pmd is not designed to handle the case where the credentials expires during the job execution. In this case the jobs terminates. So users must make sure the lifetime ticket is large enough.

- For security, you may want to destroy the ticket when the job is finished.

**Important**

Although POE in not dependent on a DFS level, you must install the DFS level supported by the AIX installed on the nodes.

### 8.2.4 DFS Use



RS/6000

DFS Use

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In this section, we will follow the steps to use DFS with POE. As a prerequisite, DFS must be installed and defined on all the nodes where POE must run, and the users must be properly authorized via DCE. The DFS installation and user′s definition are not covered in this presentation.

1. The first step is for the user to login DCE.

   The user must first use dce_login to login on the home node, to establish his DCE *credentials*. This also establishes an environment variable, KRB5CCNAME, which points to the files containing the encrypted credentials. The credentials are stored in a local filesystem of the home node.

2. The second step is to execute the poeauth command.

   The poeauth command is an executable delivered in the POE fileset. Given the number of tasks and a host list file or pool number from the Resource Manager, poeauth will use the KRB5CCNAME environment variable to determine the path of the credentials files and copy them to each remote node in a local filesystem.

   The poeauth command will use message passing routines (MPI_Send and MPI_Recv) to copy the files to the remote tasks, similar to what mcpgath does already. The high level flow of the poeauth command is as follows:

   - Determine if the DCE credentials are available by checking what the KRB5CCNAME environment variable points to.

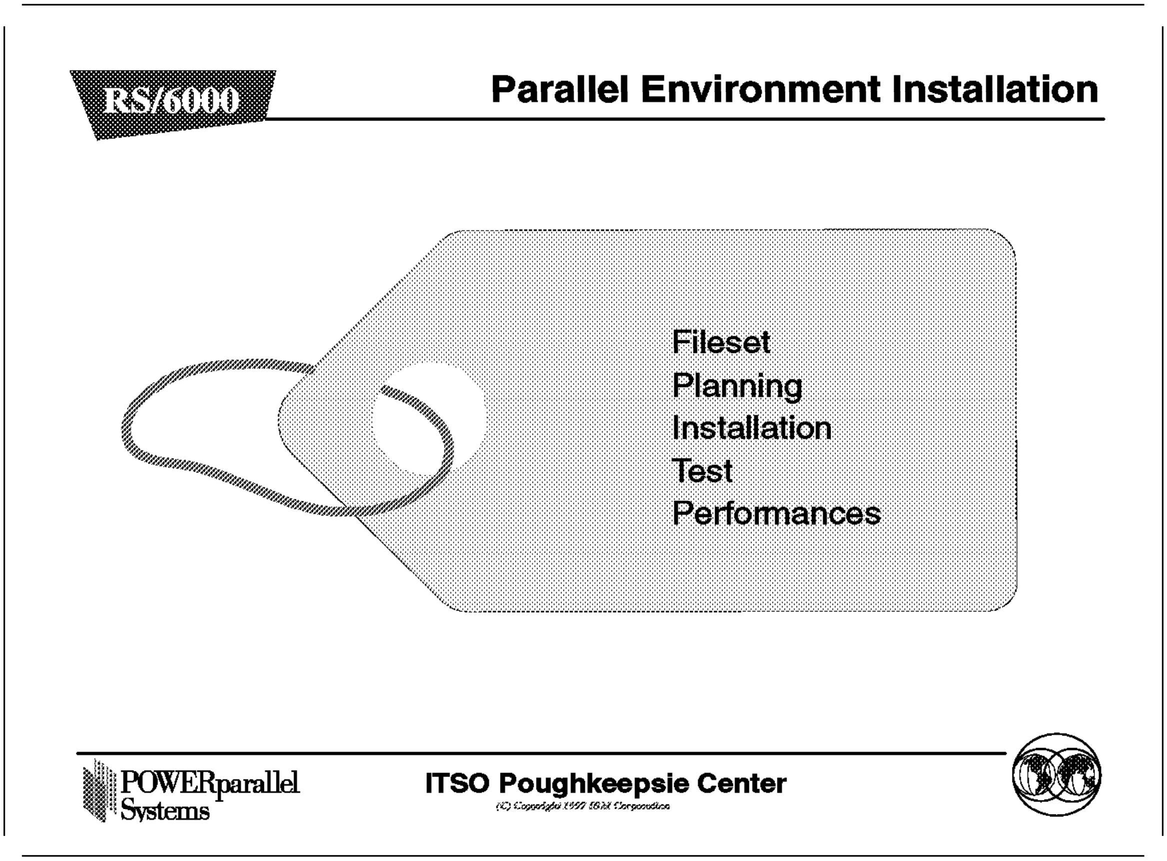
- Initialize the message passing environment and the tasks group, using the node′s names and the number of tasks specified.
- On the sender side (task0), read the KRB5CCNAME environment variable, get a list of the credentials, and send the file with the MPI_send call to each remote task.
- On the receive side (task 1 thru n), receive the contents of the file using the MPI_Recv file, and then write the files on the local file system. Due to a restriction in the AIX implementation of DCE, which relies on a hardcoded path name to access the file, it is impossible to rely on the setting of KRB5CCNAME. The alternative is to store the path name for the files in the /tmp/poedce_master.uid file using the kafs_syscall function, where *uid* is the user′s userid.

3. The third step is related to pmd.

   The pmd daemon reads the content of the credentials and gives DFS access to the remote tasks with the following operations:

   - Check for the existence of /tmp/poedce_master.uid file.
   - Read the contents of the /tmp/poedce_master.uid file.
   - Set and export the value of the KRB5CCNAME environment variable to the actual path name of the credential files.
   - Load poe_dce_shr.o. If the load fails, this indicates DCE is not installed and will terminate. If the load is successful, this means that DCE is installed, and calls are done to appropriate DCE routines.
   - Continue with the current pmd security checks via .rhosts or /etc/hosts.equiv.

## 8.3 Parallel Environment Installation



RS/6000

Parallel Environment Installation

Fileset
Planning
Installation
Test
Performances

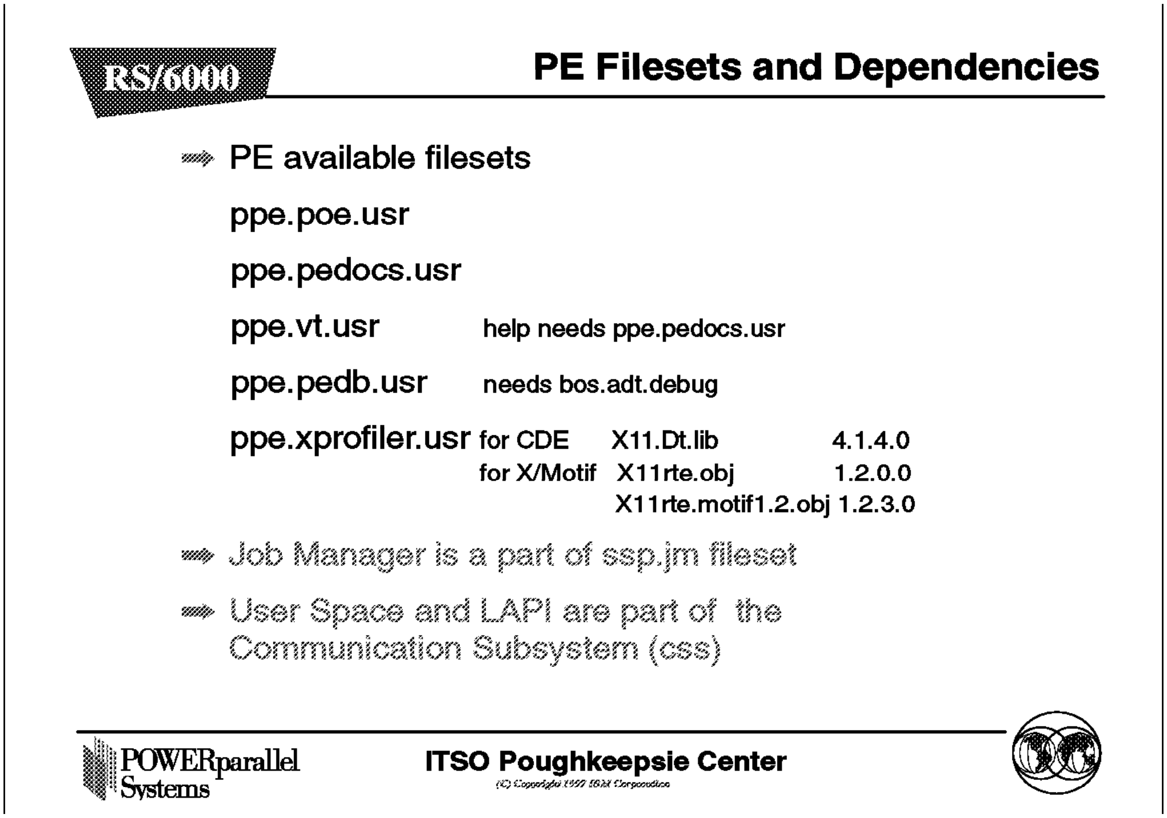
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This section presents the steps to install Parallel Environment filesets from planning to performances considerations.

### 8.3.1 PE Filesets and Dependencies



RS/6000

**PE Filesets and Dependencies**

- PE available filesets

  ppe.poe.usr

  ppe.pedocs.usr

  ppe.vt.usr help needs ppe.pedocs.usr

  ppe.pedb.usr needs bos.adt.debug

  ppe.xprofiler.usr for CDE X11.Dt.lib 4.1.4.0
  for X/Motif X11rte.obj 1.2.0.0
  X11rte.motif1.2.obj 1.2.3.0

- Job Manager is a part of ssp.jm fileset

- User Space and LAPI are part of the Communication Subsystem (css)

POWERparallel Systems

ITSO Poughkeepsie Center

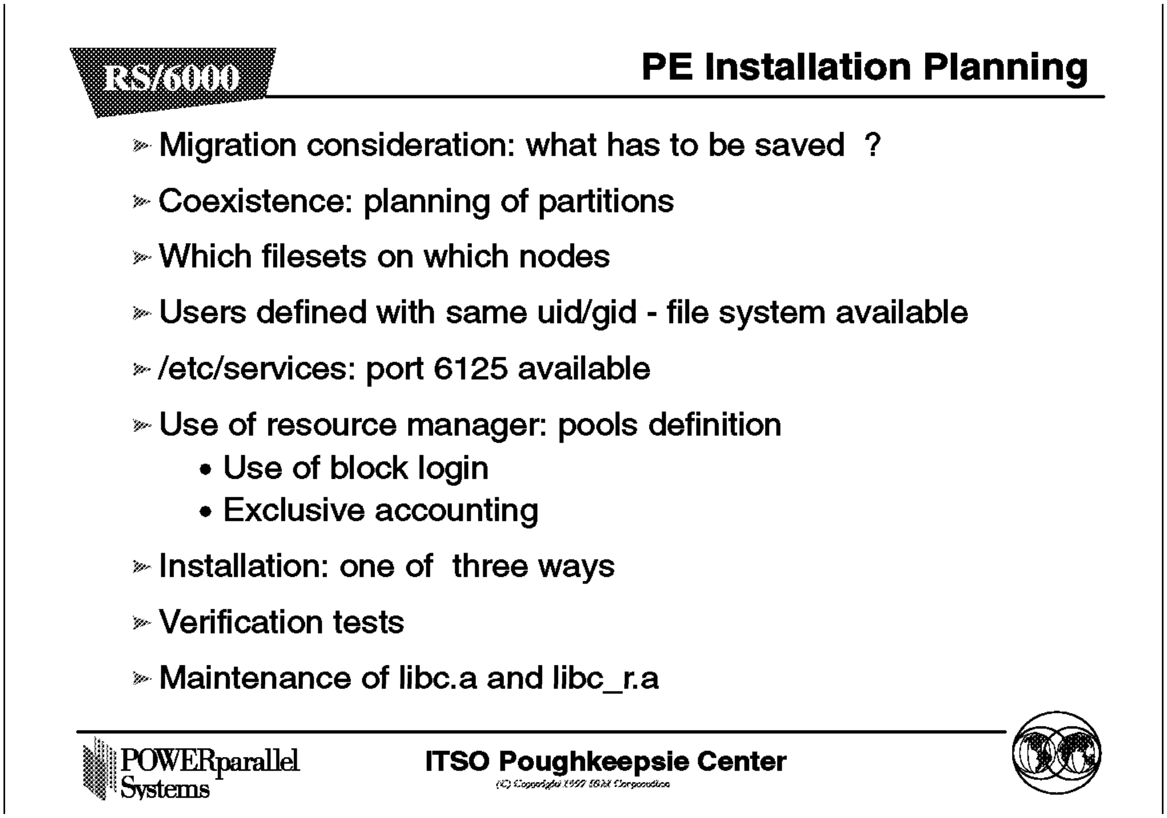
(C) Copyright 1997 IBM Corporation

The ppe.pedocs.usr fileset contains the man pages, and the PE documentation in both *html* and *postscript* format. The size of this fileset is about 40MB in /usr. You may want to save disk space by installing the fileset on one reference node (may be the Control Workstation) and mount the directories with NFS over the nodes.

The bos.adt.debug fileset must be at the level 4.2.0.3 or higher. For more information, refer to the section 8.6.15, “Prerequisites” on page 482

Although LAPI is a parallel programming library, today it is delivered as a part of the ssp.css fileset.

### 8.3.2 PE Installation Planning



RS/6000

PE Installation Planning

- Migration consideration: what has to be saved ?
- Coexistence: planning of partitions
- Which filesets on which nodes
- Users defined with same uid/gid - file system available
- /etc/services: port 6125 available
- Use of resource manager: pools definition
  - Use of block login
  - Exclusive accounting
- Installation: one of three ways
- Verification tests
- Maintenance of libc.a and libc_r.a

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

1. Migration considerations:

   In the case system administrator modified some scripts or configuration files, it may be desirable to save them and redo the modifications on the new version. The modified files could be as follows:

   - The compiler scripts
   - The configuration file /usr/lpp/ppe.poe/lib/poe.cfg
   - The amd script mpamddir

   There is no need to maintain any previous versions of libc or MPI libraries.

2. Migration of related software IBM and OEM:

   You must check that all softwares running on the nodes is supported on the new installed version.

3. Coexistence:

   The number and location of the partitions for a switch have to be defined according to the rules described in the *System Planning Guide*, GC23-3902.

4. User definition:

   The partition manager daemon tries to recreate the user environment on the remote nodes. It implies two steps:

- Users must have an account defined with the same uid/gid on both the home node and on all the remote nodes where the Parallel Environment runs.
- Users must be authorized to run rsh from the home node to each remote node they have to access. This authorization can be allowed in two files on the remote nodes: /etc/hosts.equiv, or the .rhosts file in the user′s home directory.

5. /etc/services - /etc/inetd.conf:

   When POE is installed, it adds entries to the /etc/services and /etc/inetd.conf files.

   - The service pmv2 is added to /etc/services with the default port number 6125. If this port number is not free, the next available port is attributed.

     **Note:** Check that the same port number is available on each node.

   - The daemon pmdv2 is added to /etc/inetd.conf and the /etc/inetd is refreshed automatically.

6. Resource Manager

   The Job Manager fileset ssp.jm must be installed on the Control Workstation and all the nodes where parallel jobs must be managed. It takes about 1.5MB in the /usr directory.

   The file /etc/jmd_config.syspar_name must be updated. The system administrator has to define the number of pools and subpools and the allocations of nodes within them.

   The *exclusive accounting feature* is declared in the *Site Environment Information* database.

7. Installation

   The installation process is detailed in the section 8.3.3, “PE Installation” on page 414

8. Verification tests

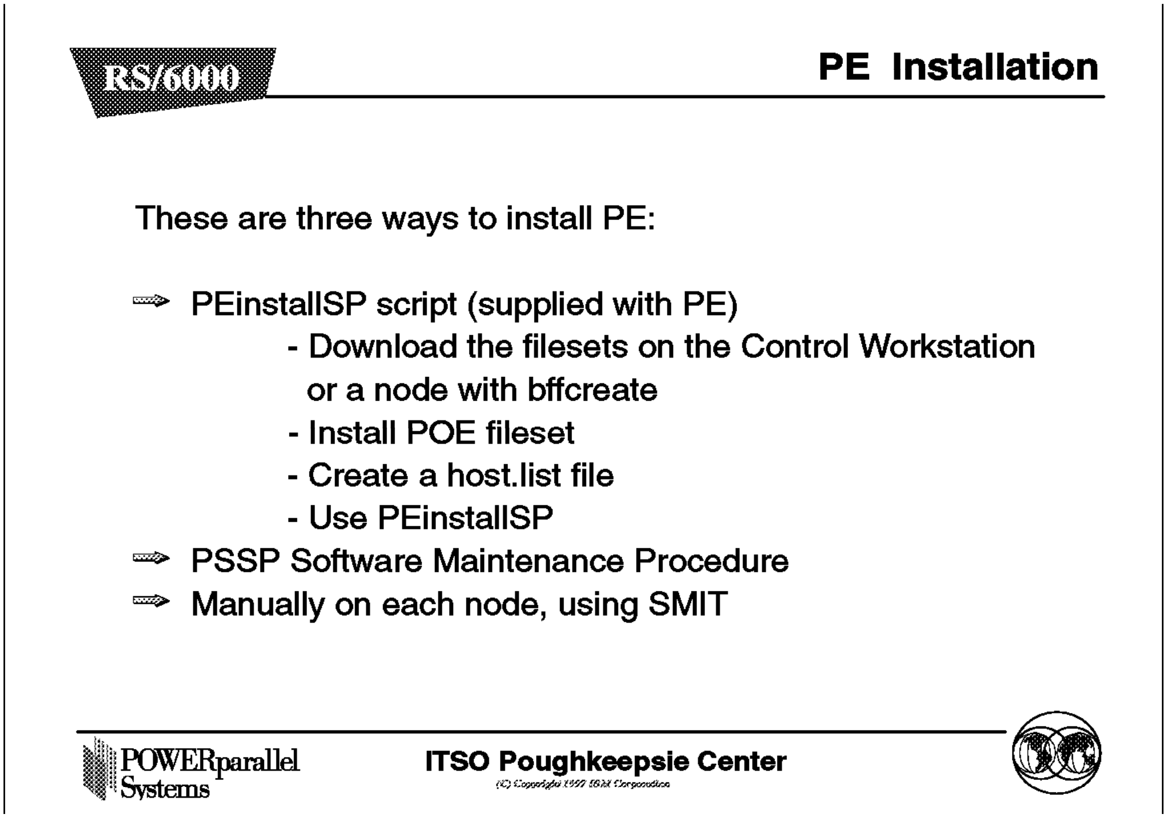
   The verification tests are detailed in the section 8.3.4, “Verification and Test Programs” on page 417

9. Maintenance

   Parallel Operating Environment maintains its own copies of libc.a and libc_r.a to create the entry and exit points when a user′s application is compiled with POE. The /usr/lpp/ppe.poe/lib/profiled/libc.a and libc_r.a are created by extracted and replacing the shr.o module of the AIX /usr/lib/libc.a and libc_r.a.

   As a result, each time service is applied that modifies the AIX lib.c and libc_r.a libraries, the makelibc must be run to recreate the POE libc and libc_r.a.

### 8.3.3 PE Installation



RS/6000

**PE Installation**

These are three ways to install PE:

- PEinstallSP script (supplied with PE)
  - Download the filesets on the Control Workstation or a node with bffcreate
  - Install POE fileset
  - Create a host.list file
  - Use PEinstallSP
- PSSP Software Maintenance Procedure
- Manually on each node, using SMIT

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The following table shows the different fileset's names and their size requirements:

Table 17. Filesets name and size

| Fileset | image_name | Size of the Image in MB | /usr requirements |
|---|---|---|---|
| all the PE filesets | all | 29.5 | - |
| ppe.poe.usr.2.3.0.0 | ppe.poe | 4.2 | 21 |
| ppe.pedb.usr.2.3.0.0 | ppe.pedb | 1.8 | - |
| ppe.vt.usr.2.3.0.0 | ppe.vt | 1.8 | - |
| ppe.xprofiler.usr.2.3.0.0 | ppe.xprofiler | 2.3 | - |
| ppe.pedocs.usr.2.3.0.0 | ppe.pedocs | 19.4 | 40 |

**Important**

Although a migration from any PE Version to PE Version 2.3 will completely override the earlier fileset, *it is strongly recommended to uninstall the oldfileset.* This will reduce the chance for confusion over old fileset, path name, executable, etc. Use the `PEdeinstallSP` command to uninstall all the PE filesets.

You can install PE by using one of three methods. However, regardless of the method you use, as a preliminary step, you have to download all the filesets on the Control Workstation or on one node in the default directory: /spdata/sys1/install/pssplpp/PSSP-2.3.

- The first method of installing the PE is by using the PEinstallSP script:
  1. Install the POE to recover the `PEinstallSP` command.

     If the installation is done on the CWS, the link between the user space library mpci.a and the switch adapter will never be established. (It is normally created by the /usr/lpp/ssp/css/rc.switch when called from /etc/inittab).
     Therefore, you must create the link before installing the POE, The installation steps depend on the correct adapter libraries being linked, as follows:

     – TB2 switch adapter:

       - `ln -s /usr/lpp/ssp/css/libtb2/libmpci.a /etc/ssp/css/libus/libmpci.a`
       - `ln -s /usr/lpp/ssp/css/libtb2/libmpci_r.a /etc/ssp/css/libus/libmpci_r.a`:

     – TB3 switch adapter:

       - `ln -s /usr/lpp/ssp/css/libtb3/libmpci.a /etc/ssp/css/libus/libmpci.a`

       - `ln -s /usr/lpp/ssp/css/libtb3/libmpci_r.a /etc/ssp/css/libus/libmpci_r.a`

  2. Create a `host.list` file with the name of the nodes where the different filesets should be installed. The default name is host.list in the home directory.

  3. Use the `PEinstallSP` script:

     ```
     PEinstallSP image_name [ host_list_file ] [ -f fanout_value ] [-copy |
     -mount ]
     ```

     – `image_name` is mandatory, and represents the name of the installp image.

     **Note**

     The explanation of image_name is confusing. In fact, it is the name of the subdirectory containing the images. Depending of the level to install, you must enter PSSP-2.1, PSSP-2.2 or PSSP-2.3.

     Further, the command concatenates the default source directory /spdata/sys1/install/pssplpp (from either the `-copy` or the `-mount` option) with this image_name to produce the default destination directory /spdata/sys1/install/pssplpp/PSSP-2.3.

     – `host_list_file` is optional, and represents the file containing the list of nodes on which you want to install the fileset. The default file name is `host.list` in the current working directory.

     – `-copy` is the default option. It copies the named image to each node using `rcp`. You are prompted for:

        - The installation image source directory. The default is /spdata/sys1/install/pssplpp.
        - The installation image destination directory. The default is /spdata/sys1/install/pssplpp.

    - -mount: The script issues a mkdir command to create the destination directory, followed by a chmod 777. You are prompted for:

        - The installation image source directory. The default is /spdata/sys1/install/pssplpp.
        - The installation image destination directory. The default is /mnt.

    - PEinstallSP issues a dsh command to execute:

      ```
      installp -aFX -d/image_directory/image_name fileset
      ```

4. With the -mount option, it is more secure to issue a dsh chmod 755 for the nodes where the filesets have been installed. With the -copy, you may want to save disk space and erase the image_name.

**Notes**

- In a general way, the filesets are installed on the Control Workstation. To save disk space and unnecessary work, you can avoid installing the POE fileset (unless needed) by extracting the PEinstallSP command with:

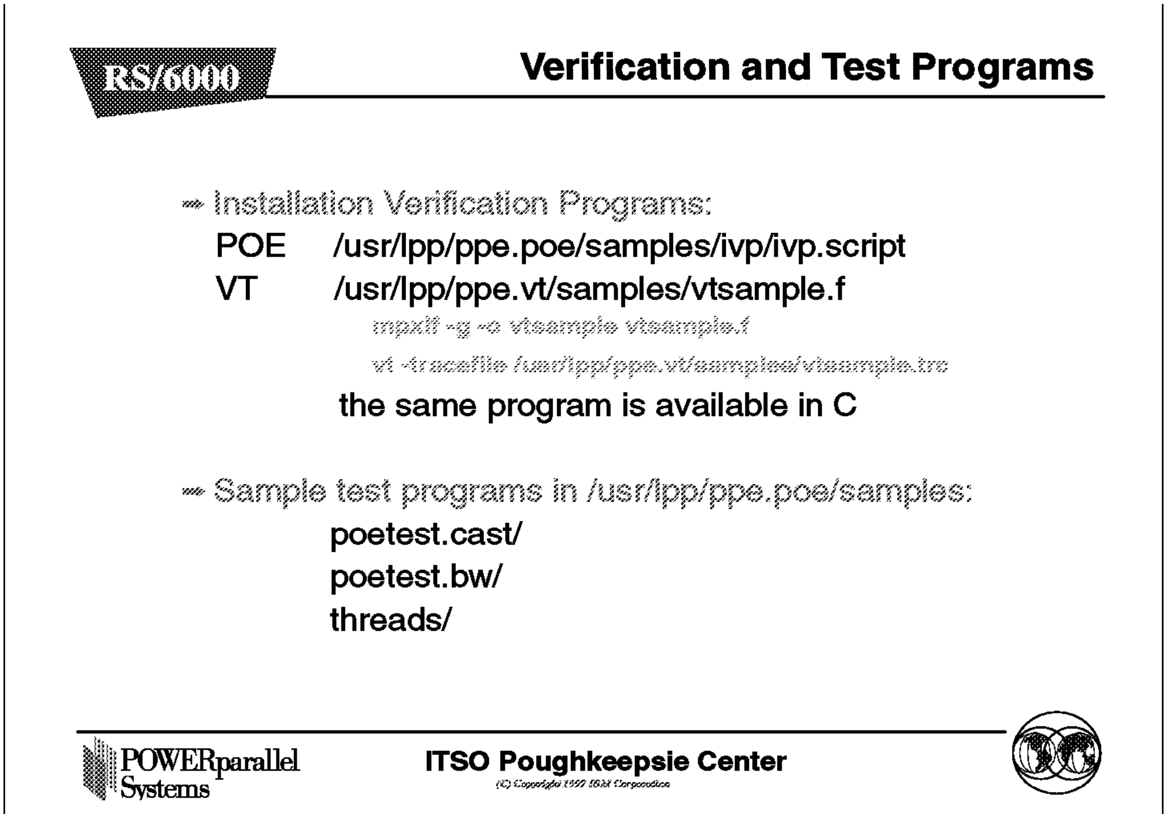
    - `cd /`
    - `restore -xvf /spdata/sys1/install/pssplpp/PSSP-2.3/ppe.poe.usr.2.3.0.0 ./usr/lpp/ppe.poe/bin/PEinstallSP`

- PEinstallSP does not produce any log. As you are installing many nodes at a time with one or more fileset, it is recommanded to use the command: `PEinstallSP [options] 2>&1 | tee peinstall.log`.

- The second method of installing PE is by using the PSSP Software maintenance Procedure.

- The third method of installing PE is by using standard AIX commands. You must mount the directory or copy the files and execute SMIT manually on each node.

### 8.3.4 Verification and Test Programs



RS/6000

**Verification and Test Programs**

- Installation Verification Programs:

  POE /usr/lpp/ppe.poe/samples/ivp/ivp.script

  VT /usr/lpp/ppe.vt/samples/vtsample.f

  ```
  mpxlf -g -o vtsample vtsample.f
  vt -tracefile /usr/lpp/ppe.vt/samples/vtsample.trc
  ```

  the same program is available in C

- Sample test programs in /usr/lpp/ppe.poe/samples:

  poetest.cast/
  poetest.bw/
  threads/

POWERparallel Systems

ITSO Poughkeepsie Center

- Installation Verification Programs (IVP)
  - POE test:

    To test the initial installation of POE, an Installation Verification Program (IVP) script is provided to ensure that the following is true:

    - There is access to a C compiler.
    - The MPI libraries are installed and linked.
    - Certain commands are there and executable (poe,pmdv2,mpcc).
    - Sample programs are compiled and run.
    - Check that the dbx bos.adt.debug fileset is present for the parallel debugger.
  - VT test:

    To test that the VT trace generation mechanism is operating correctly, you can compile a sample program available in C or FORTRAN with the -g flag, and start VT.

    - Press **Load Balance** under **Computation** on the VT view selector panel.

    - Press **Play** on the main control panel.

    If the trace file plays to the end while updating the display, VT is installed successfully.

- Sample test programs:

  Three sample programs are delivered with: README, source code, makefile and scripts files in the directory /usr/lpp/ppe.poe/samples.

  **Subdirectory** **Content**

  **poetest.bw** This is the directory where you can find a Point-to-point bandwidth measurement test. The code needs only two nodes and can run in IP or in user space (us) mode. This sample can be useful in tuning network parameters.

  **poetest.cast** The purpose of this test is to perform a broadcast from task 0 to all the nodes in the partition.

  **threads** Two source programs are delivered to illustrate the use of the MPI message passing library with user-created threads. One is for testing with user threads, the other for testing with a user signal handler.

- Other samples are given in the directories:

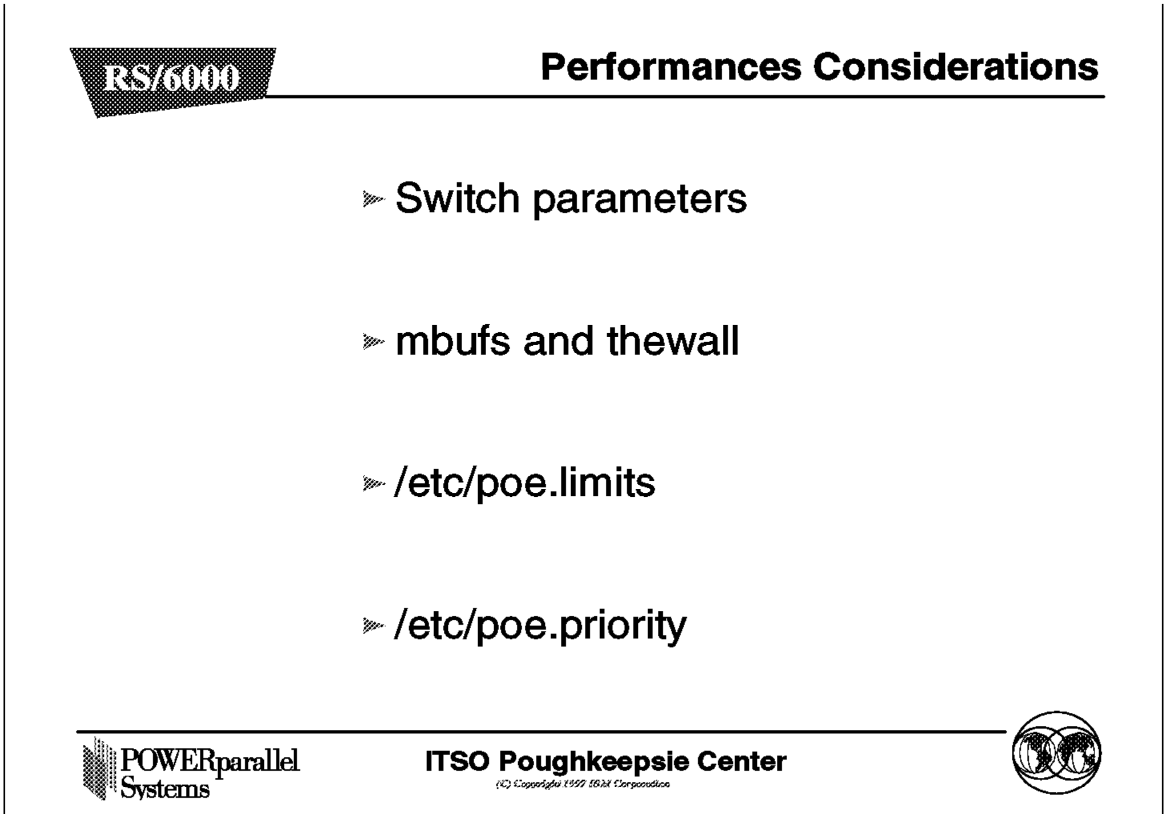
  /usr/lpp/ppe.poe/samples/marker
  /usr/lpp/ppe.poe/samples/mpi
  /usr/lpp/ppe.vt/samples
  /usr/lpp/ppe.pedb/samples
  /usr/lpp/ppe.xprofiler/samples

**Note:** AFS and the parallel fortran 90 compilers (mpxlf90 and mpxlf90_r) are only delivered as samples in the directories:

/usr/lpp/ppe.poe/samples/afs
/usr/lpp/ppe.poe/samples/mpif90

They are not supported.

## 8.3.5 Performance Considerations



RS/6000

**Performances Considerations**

- Switch parameters
- mbufs and thewall
- /etc/poe.limits
- /etc/poe.priority

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

- To have effective performance over the switch, particular attention should be given to the switch parameters. The correct parameters are given in the *Administration Guide,* GC23-3897. The parameters can be distributed with a `dsh no` command. To maintain coherent parameters for all the nodes, it is recommended to use the /etc/rc.net file and distribute it with a `pcp` command or with the *file collections*.
- A POE application may require additional IP buffers (mbufs) under any of the following circumstances:
  - A partition is larger than 128 nodes.
  - A large amount of STDIO (stdin, stdout, stderr) is generated.
  - The home node is running many POE jobs simultaneously, and there is significant IP traffic via mounted or share file system activity.
  - Many large messages are passed via the UDP implementation of the Message Passing Library.

  The additional IP buffers needed are usually evident when repeated requests for memory are denied. Using the `netstat -m` command can tell you when such a condition exists. Under these conditions, you need to increase the value of the `thewall` parameter with the `no` command.

  In AIX version 4.2, the `thewall` default value is 16384.

For more general information on mbufs, see the *AIX Performance and Tuning Guide*.

- The partition manager daemon (pmdv2) on each node examines the /etc/poe.limits file. On each node, the pmdv2 daemon receives the environment values from the home node. If the environment value is not compatible with what is available, it can cause problems on the remote node. For example, if a node only has 64 MB of real memory, a default value of 64 MB for MP_BUFFER_MEM would be too high. This file allows the system administrator to override some defaults for environment variables.
- Priority adjustments:

  Certain applications can benefit from enhanced dispatching priority during execution. POE provides a service for periodically adjusting the dispatching limits of a user′s task between limits. The service is specified by entries in the file /etc/poe.priority with hipriority, lopriority, duty factor, and adjustment period.

**Attention**

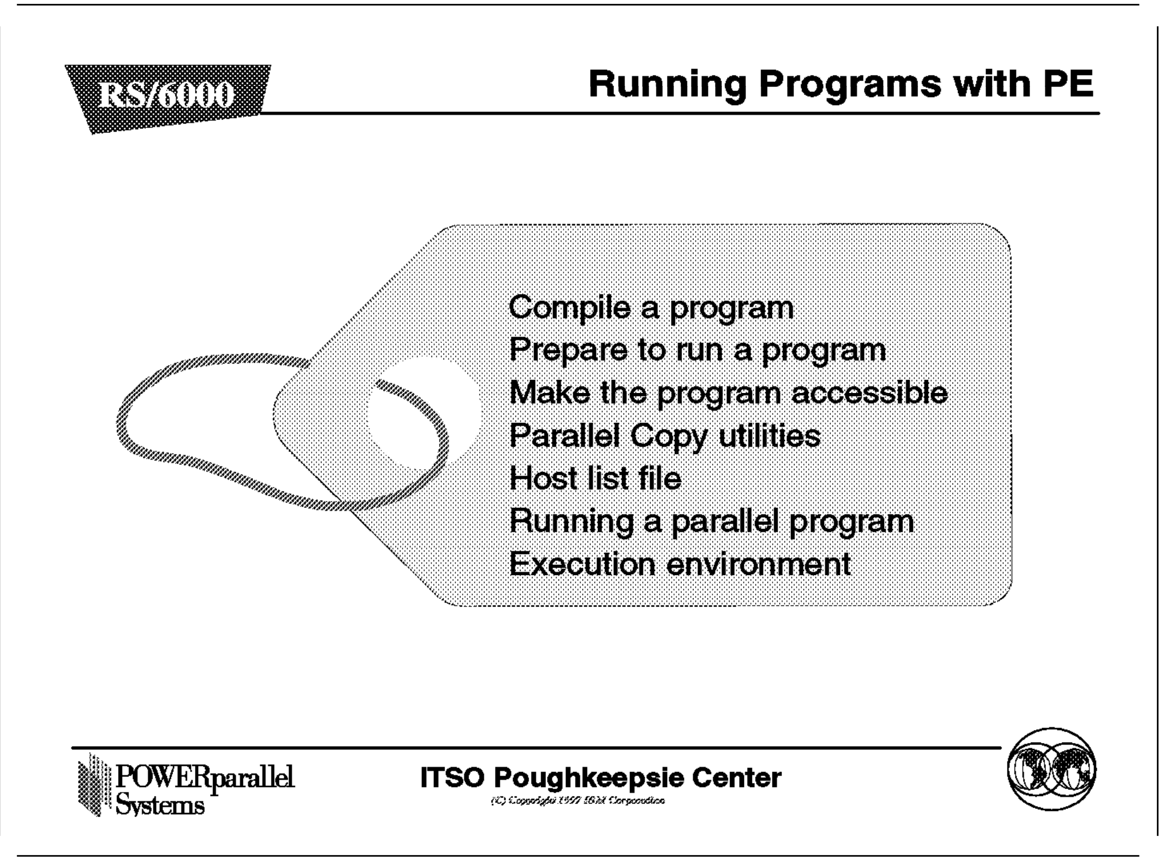
System administrators must evaluate the effect of the priority settings in their own environment. With a priority that is set too low, user jobs will compete with the system processes and may disrupt normal activity. Some examples of this are as follows:

- The system may hang.
- The switch fault recovery may be unsuccessful and the node will be disconnected from the switch.
- Keystrokes may be inhibited.
- If the user is more favored than the network processes, the required IP message passing traffic may be blocked and cause the program to hang.
- Other users′s jobs would never be dispatched.

**Important**

Before any priority adjustments, consult the include file /usr/include/sys/pri.h for definitions of the priorities used for normal AIX operations.

## 8.4 Running Programs with PE



RS/6000

### Running Programs with PE

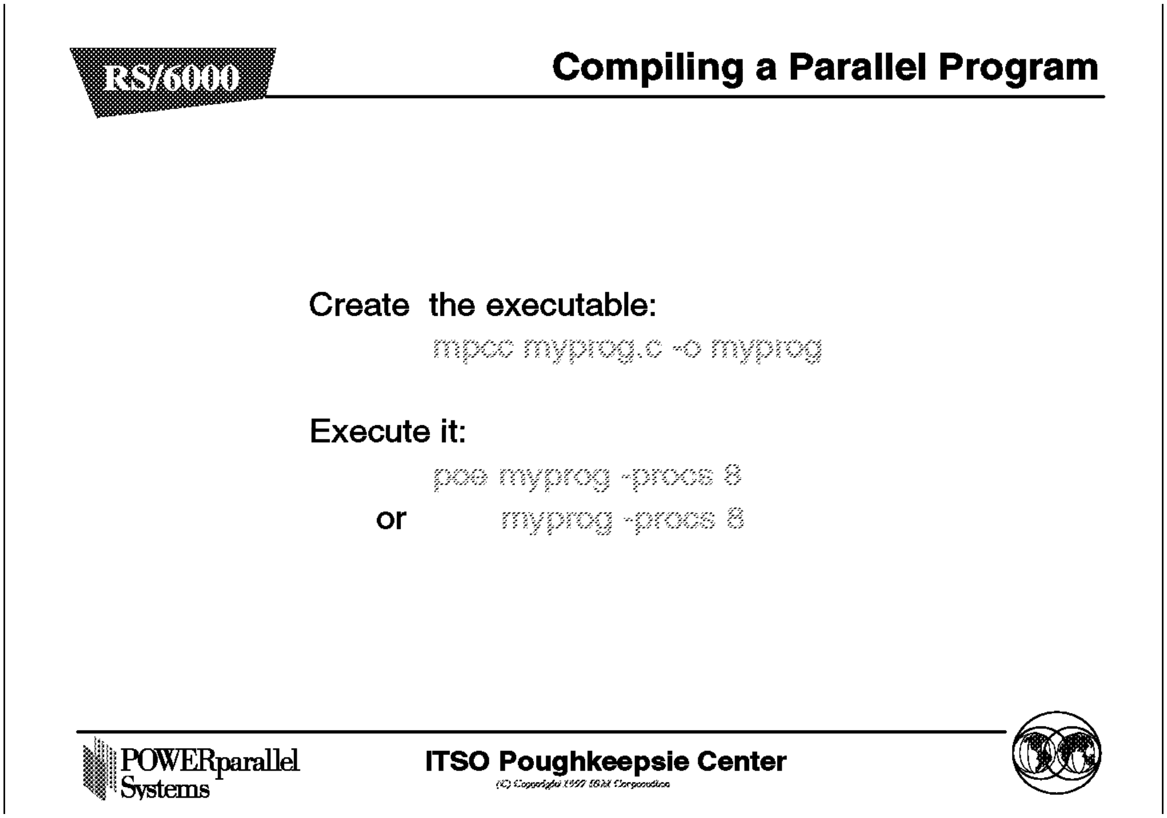
- Compile a program
- Prepare to run a program
- Make the program accessible
- Parallel Copy utilities
- Host list file
- Running a parallel program
- Execution environment

POWERparallel Systems

ITSO Poughkeepsie Center

This chapter presents aspects on the way to run a program with Parallel Environment, the environment execution prerequisites, from the compilation to the node allocation.

## 8.4.1 Compiling a Parallel Program



RS/6000

Compiling a Parallel Program

Create the executable:

```
mpcc myprog.c -o myprog
```

Execute it:

```
    poe myprog -procs 8
or      myprog -procs 8
```

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

As with a parallel application, a parallel C, C++, or Fortran program must be compiled before being run. Instead of using the cc, xlc, or xlf commands, you must use the Parallel Environment commands, mpcc, mpCC, or mpxlf. To compile with the threaded versions of the compilers, use the mpcc_r, mpCC_r, or mpxlf_r commands.

### 8.4.1.1 dynamic executables

These commands not only compile the program, but also link in the partition manager and message passing interface. When the executable is invoked, the libraries will be dynamically linked. The subroutines in these libraries enable the home partition manager to communicate with the parallel tasks, and enable the tasks to communicate with each other.

These commands are script shells that call the appropriate AIX compilers with the necessary flags for the Parallel Environment; for example:

- One of the library paths is set to /usr/lpp/ppe.poe/lib.
- The following line is included to give the path to the POE include file and to support the new initfini binder option:
  `_FLAGS="-I$_INCLUDE -binitfini:poe_remote_main "`

All the flags given to the scripts are passed to the compiler.

The scripts also refer to the POE configuration file /usr/lpp/ppe.poe/lib/poe.cfg. Each script has its own stanzas in this file. For the C compiler, the corresponding stanzas, cc and cc_r, are as follows:

```
Abstract of /usr/lpp/ppe.poe/lib/poe.cfg:

* standard POE C compiler AIX 4.2
* Derived from /etc/xlC.cfg:cc (AIX 4.2)
cc:     use        = DEFcC
        crt        = /lib/crt0.o
        mcrt       = /lib/mcrt0.o
        gcrt       = /lib/gcrt0.o
        libraries  = -lmpi,-lvtd,-lc
        proflibs   = -L/lib/profiled,-L/usr/lib/profile
        options    = -qlanglvl=extended,-qnoro,-qnoroconst

* standard POE c compiler aliased as cc_r (DCE) AIX 4.2
* Derived from /etc/xlC.cfg:cc_r (AIX 4.2)
cc_r:   use        = DEFcC
        crt        = /lib/crt0_r.o
        mcrt       = /lib/mcrt0_r.o
        gcrt       = /lib/gcrt0_r.o
        libraries  = -L/usr/lpp/ppe.poe/lib/threads,-lmpi_r,-lvtd_r,\
 -lc_r,-lpthreads,/usr/lib/libc.a
        proflibs   = -L/lib/profiled,-L/usr/lib/profiled
        options    = -qlanglvl=extended,-qnoro,-qnoroconst,-D_THREAD_SAFE
```

**Important**

- Although there is a mpxlf_r script and the relevant stanza in the configuration file, the Fortran Version 4.1.0 does not support threads. Because xlf_r does not exist, the mpxlf_r script just calls the xlf compiler.
- However, Fortran programs can take advantage of the threaded versions of the communication libraries, MPI and LAPI.
- Incompatibilities exist between Fortran 90 and MPI which may affect the ability to use these programs. For more information on the restrictions and implications of using MPI and Fortran 90, refer to the /usr/lpp/ppe.poe/samples/mpif90/README.mpif90 file after POE is installed. Although the xlf90 and xlf90_r are present in the poe.cfg configuration file, the POE compiling scripts and the include files are delivered as samples in the /usr/lpp/ppe.poe/samples/mpif90 directory.
- As for Fortran, the xlf90 compiler is not available in a threaded version. The mpxlf90_r refers to the xlf90 compiler.

### 8.4.1.2 Static executables

Creating statically bound executables with POE is not recommended. If service is ever applied that affects any of the Parallel Environment libraries, the applications need to be recompiled to create a new executable. This leads to a lot of unnecessary work and may expose you to potential problems.

#### 8.4.1.3 Message catalogs

The PE message catalogs are in English, and located in the following directories:

- /usr/lib/nls/msg/C
- /usr/lib/nls/msg/En_us
- /usr/lib/nls/msg/en_US

Although all the Parallel Environment components and tools support the National Language Support (NLS), if your site is using its own translated message, you could get an error saying that a message catalog is not found. In this case, you have to use the default message catalog:

```
export NLSPATH=/usr/lib/nls/msg/%L/%N
export LANG=C
```

#### 8.4.1.4 Examples

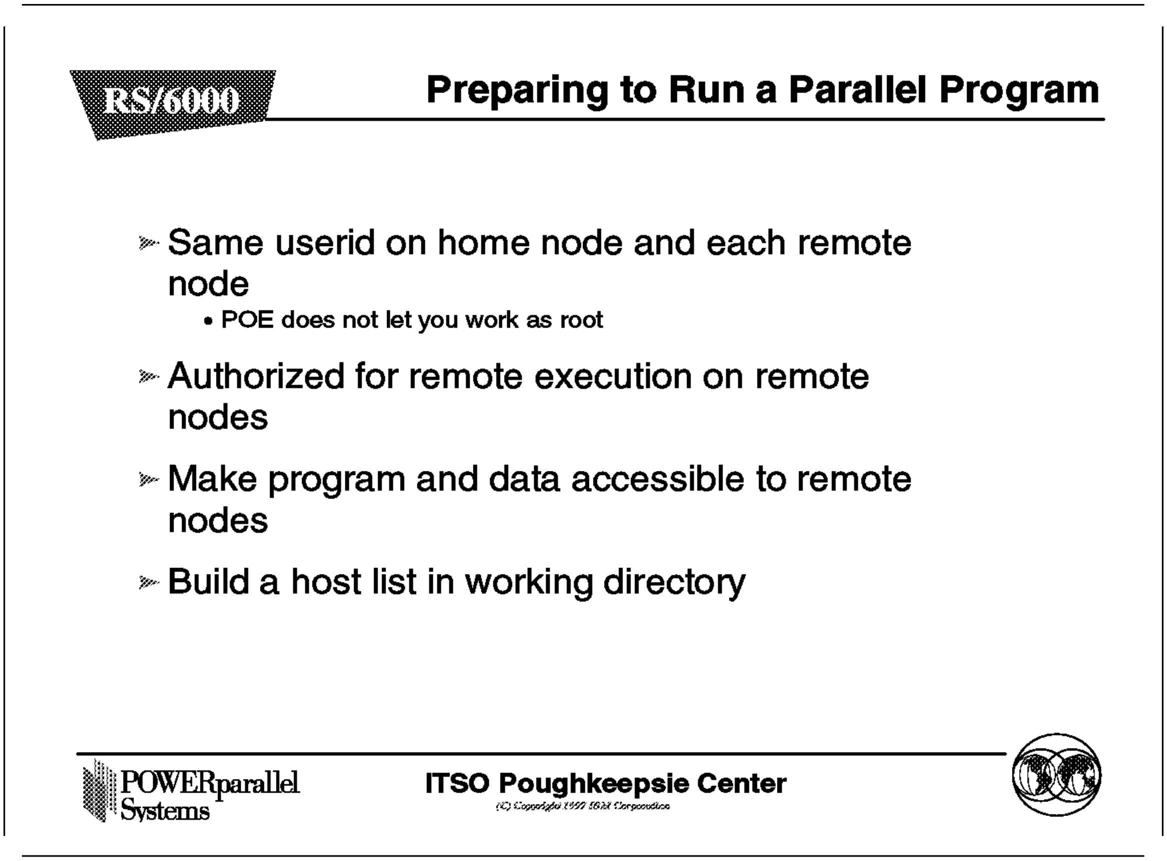
The executable is simply created with the following command:

```
mpcc myprog.c -o pmyprog
```

The following command executes myprog on eight nodes:

```
poe myprog -procs 8
```

## 8.4.2 Preparing to Run a Parallel Program



RS/6000

**Preparing to Run a Parallel Program**

- Same userid on home node and each remote node
  - POE does not let you work as root
- Authorized for remote execution on remote nodes
- Make program and data accessible to remote nodes
- Build a host list in working directory

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Implementing the steps for preparing to run a parallel program is the responsibility of both system administrator and the user.

- Users definition

  This part is system administrator responsibility. Because the partition manager daemon tries to reproduce on the remote node the same user environment as it exists on the home node, the user must be defined an all nodes with the same uid/gid.

- Users authorization

  The partition manager daemon of the home node and those of the remote nodes establish a link between the task 0 on the home node and the tasks 1 thru n of the remote nodes. The task 0 requests remote execution of tasks 1 thru n. Therefore, the users must be authorized for remote execution on the remote nodes. This step can be done at a system administrator level, or at the user level, as follows:

  - The system administrator can give the appropriate authorization in the /etc/host.equiv file.
  - Alternatively, each user can give his own access authorization with the .rhosts file in its home directory.

- Make program and data available

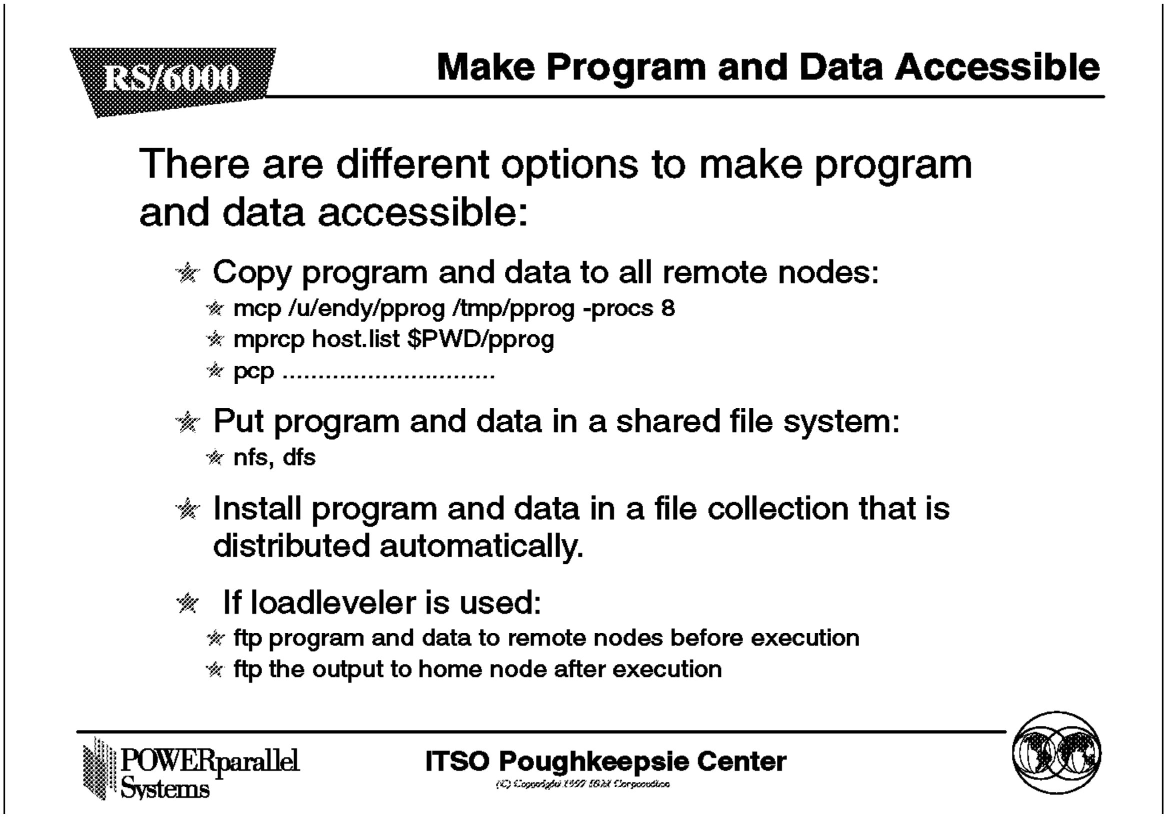
This is a user responsibility. The partition manager loads the user′s executable in the memory of all nodes where the program has to be run. Therefore the program must be accessible to the remote nodes. In the same way, each task could require access to one or more data files either in read mode when the program starts, or in write mode when the program finishes. In the following section, we describe the different ways to make the programs and data available.

- Host list file definition

  Most of the POE commands and tools relies on a list of nodes where they must execute. The list of nodes can be defined at the system level with the Resource Manager configuration file or at a user level with a host file.

  - By default, if no file is given, the commands look for host.list in the current working directory.
  - It is still possible to specify a full path name like: /u/endy/parallel/bin/node_list.

### 8.4.3 Make Program and Data Accessible



RS/6000

**Make Program and Data Accessible**

There are different options to make program and data accessible:

- Copy program and data to all remote nodes:
  - mcp /u/endy/pprog /tmp/pprog -procs 8
  - mprcp host.list $PWD/pprog
  - pcp ..............................
- Put program and data in a shared file system:
  - nfs, dfs
- Install program and data in a file collection that is distributed automatically.
- If loadleveler is used:
  - ftp program and data to remote nodes before execution
  - ftp the output to home node after execution

POWERparallel Systems

ITSO Poughkeepsie Center

This section describes different options for making program and data available on all nodes.

1. One option is to copy the program and data on all remote nodes. Different commands from POE, PSSP, or standard commands can be used.

   - mcp *infile* [*outfile.*] [*POE options.*]
     This command allows you to propagate a copy of a file to multiple nodes. It has be rewritten in the Parallel Environment Version 2.3 with the MPI library. As a POE program, all POE options are available. The most interesting flags for this command are `-procs` to give the number of nodes to copy the file, and `-euilib` to use the communication protocol `ip` or `us`.

     a. The following command copies a file from the current directory to the current directory on 16 nodes, using the user space protocol through the switch:

     ```
     mcp filename -procs 16 -euilib us
     ```

     b. The following command copies a file from the current directory to the /tmp directory on 16 nodes:

     ```
     mcp filename /tmp -procs 16 -euilib us
     ```

   - mprcp `host.list filename`
     This command copies a file from the home node to a list of remote hosts. It is a script shell which uses the rcp and rsh commands. This command

is more suitable to distribute files over workstations outside an RS/6000 SP.

The two parameters are mandatory.

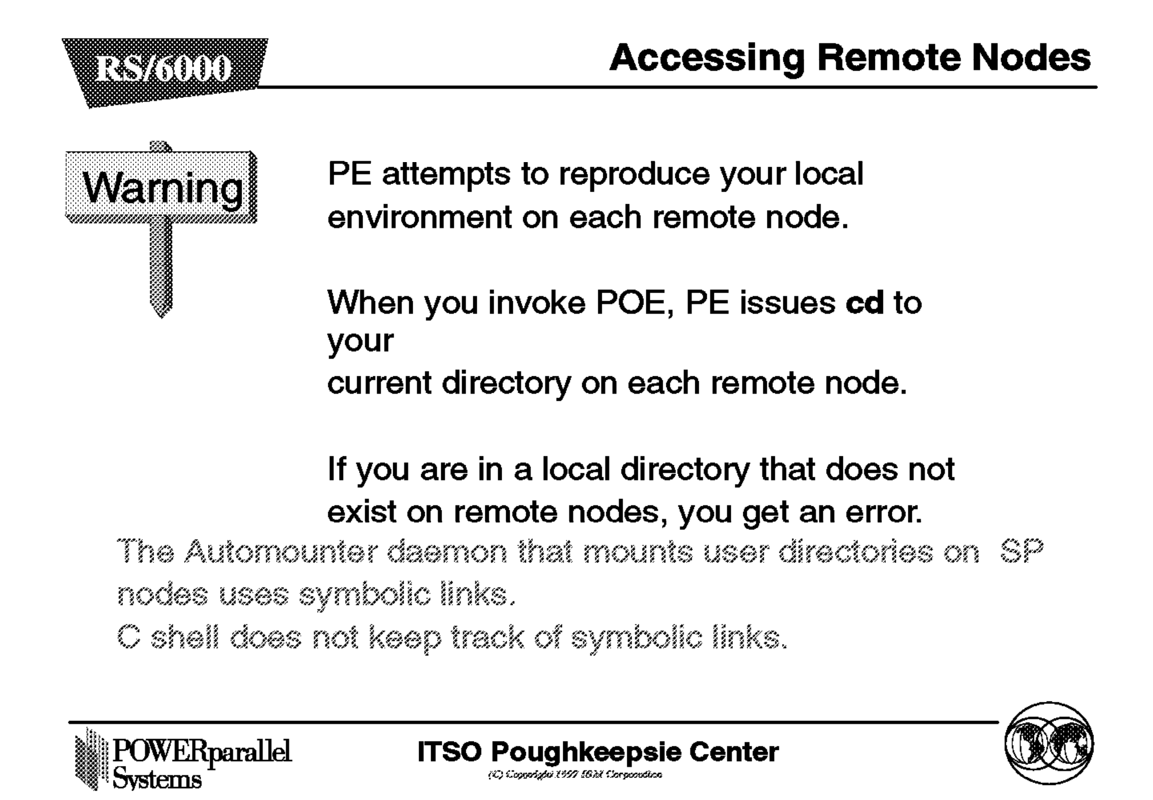
- pcp
This command also allows you to propagate a copy of a file to multiple nodes. It is a part of the PSSP and uses Kerberos authentication. The user must be authorized to log into Kerberos to use pcp.

2. Another option is to make program and data available with a share filesystem if this solution is available on your system. With the Parallel Environment Version 2.3, NFS and DFS are supported. Although this solution does not create extra work for the user, this option may not be suitable in every cases. If large executables need to be loaded quickly, or if programs need to write a large amount of data, the most powerful solution is to make program and data resident on each node.

3. A further option is to make program and data available in a Parallel System Support Programs file collection that is distributed automatically. The system administrator must define a file collection where your files will be distributed automatically. This implies the files still reside on the same node.

4. A different solution can be useful when the system is under the control of LoadLeveler. To submit a job to LoadLeveler, the user can create a job command file which contains information needed by LoadLeveler. In this file, it is possible to include an `ftp` statement to tranfer programs and data to the nodes before the job′s execution.

### 8.4.4 Accessing Remote Nodes



RS/6000

## Accessing Remote Nodes

Warning

PE attempts to reproduce your local environment on each remote node.

When you invoke POE, PE issues **cd** to your
current directory on each remote node.

If you are in a local directory that does not exist on remote nodes, you get an error.

The Automounter daemon that mounts user directories on SP nodes uses symbolic links.
C shell does not keep track of symbolic links.

ITSO Poughkeepsie Center

You might encounter a problem when the automounter is used in conjunction with the C shell.

The automounter is used to mount user directories with symbolic file system links rather than the physical file system links, as they are defined in the amd′s map. While the korn shell keeps track of file system changes, the C shell only maintains the physical file system link. As a result, users that run POE from a C shell may find that their current directory is not known to amd and POE fails.

By default, POE uses the `pwd` command to obtain the name of the current directory. This works for C shell users if the current directory is either:

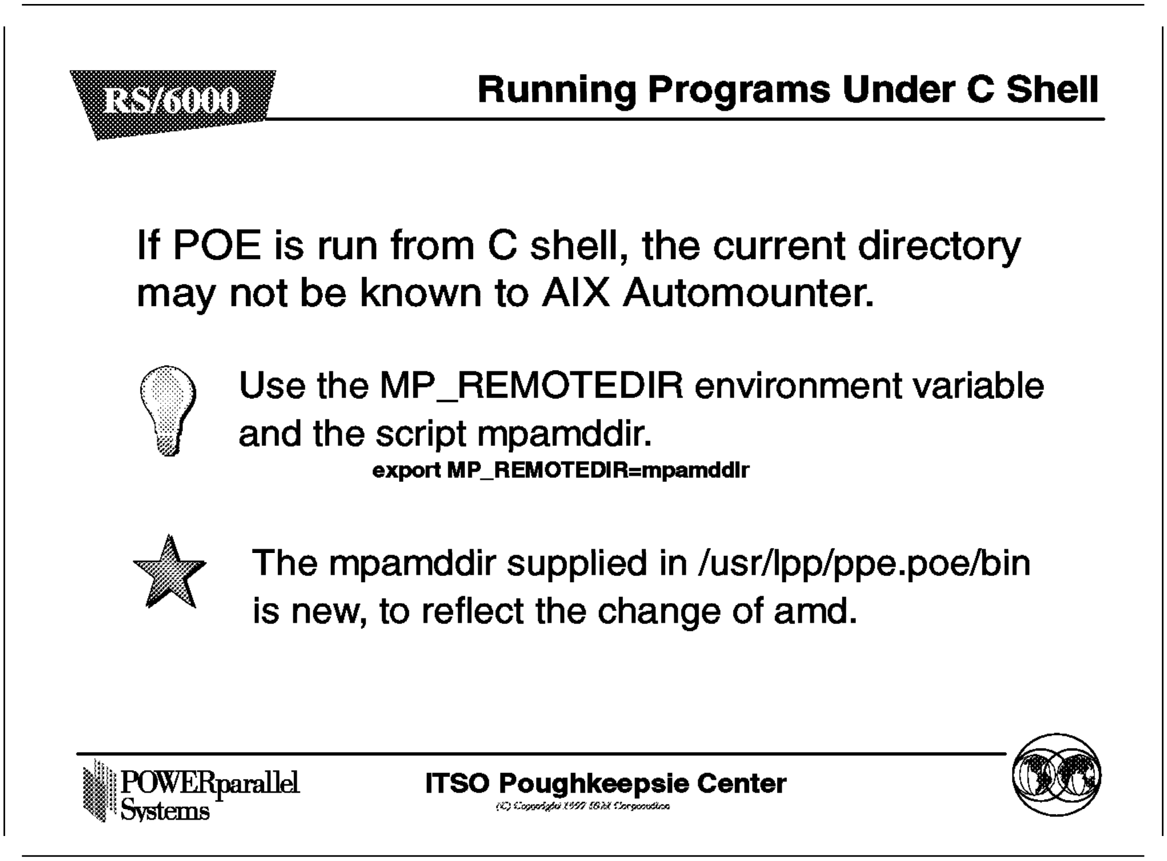
- The home directory
- Not mounted by amd

Assume a user *user1* is created on the file system *filesys1* with an automounter mount point /amd_mount. In the home directory, the `pwd` command will return the following:

- With the korn shell: `/u/user1`
- With the C shell: `/amd_mount/filesys1/user1`

When the remote node receives this path, the automounter finds nothing to mount in its map, so the POE `cd` command will fail.

### 8.4.5 Running Programs Under C Shell



RS/6000

## Running Programs Under C Shell

If POE is run from C shell, the current directory may not be known to AIX Automounter.

Use the MP_REMOTEDIR environment variable and the script mpamddir.

**export MP_REMOTEDIR=mpamddir**

The mpamddir supplied in /usr/lpp/ppe.poe/bin is new, to reflect the change of amd.

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

When POE issues the cd command from a current directory not known by amd, for example a subdirectory, the user directory will not be mounted on the remote nodes. POE will be unable to change to this directory and will fail. In this case, POE provides the MP_REMOTEDIR environment variable to determine the correct amd map. POE recognizes the MP_REMOTEDIR variable as a name of a command or a script that echoes a fully-qualified file name.

MP_REMOTEDIR is run from the current directory from which POE is started.

- If the MP_REMOTEDIR directory is not set, the default command issued is pwd. Assuming you are in the /usr/lpp/ppe.poe directory, when POE is invoked, it issues pwd and gets back /usr/lpp/ppe.poe. This value is sent to the remote nodes which uses it as the current directory.
- If you set MP_REMOTEDIR=″echo /tmp″, POE executes this command, gets back the /tmp value, and sends the value to the remote nodes. The current directory is now /tmp on the remote nodes.

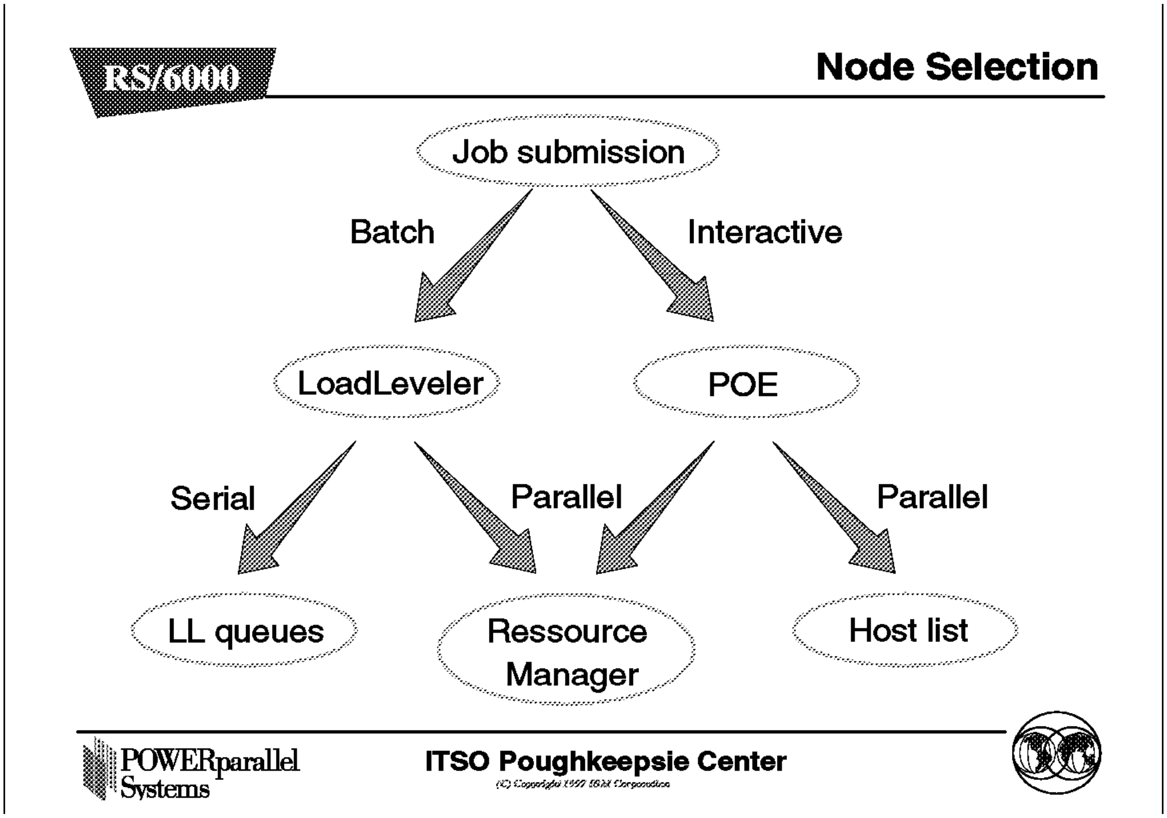
POE provides the `/usr/lpp/ppe.poe/bin/mpamddir` script that:

- Determines if the current directory is a mounted file system or not.
- If it is the case, searches the amd map for this directory.
- Builds, for this directory, a name which is known by amd.

With the setting export MP_REMOTEDIR=mpamddir, POE executes the script, gets a value which is known by amd, and sends this value to the remote nodes. The directory can be mounted by amd and POE can execute the cd command.

**Note:** The mpamddir has been changed to reflect that the Parallel System Support Programs amd has been moved to the AIX automounter. System administrators who modified mpamddir for their own purposes must redo the modifications on the new script.

### 8.4.6 Node Selection



RS/6000

Node Selection

Job submission

Batch

Interactive

LoadLeveler

POE

Serial

Parallel

Parallel

LL queues

Ressource Manager

Host list

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This figure presents an overview of the ways a user can select nodes to run a job. Depending upon the system administration policy, jobs could be run in batch mode, or in interactive mode, or both, as discussed in the following section:

- Batch mode

  If you want to run a program in batch mode, you must send a request to *LoadLeveler*. There are different ways to submit a job through LoadLeveler: interactive, job command file, or graphic interface. One item of requested information is the job′s type, either serial mode or parallel mode. The serial jobs are handled by LoadLeveler, which sends them into LoadLeveler queues.

  **Note:** The Resource Manager does not handle serial jobs.

  When the statement indicates the job must be run in parallel, LoadLeveler sends the request to the Resource Manager. System administrators must add some options in the LoadLeveler configuration file to interface with the Resource Manager. For complete information, refer to the LoadLeveler documentation or to *IBM LoadLeveler Technical Presentation*, ZZ81-0348-00.

- Interactive mode

  The way to submit a parallel job in interactive mode is to run the `poe [ options ]` command from the command line on the home node. Depending on the options given, POE sends the request to the Resource Manager, or to a list of nodes defined in a host list file.

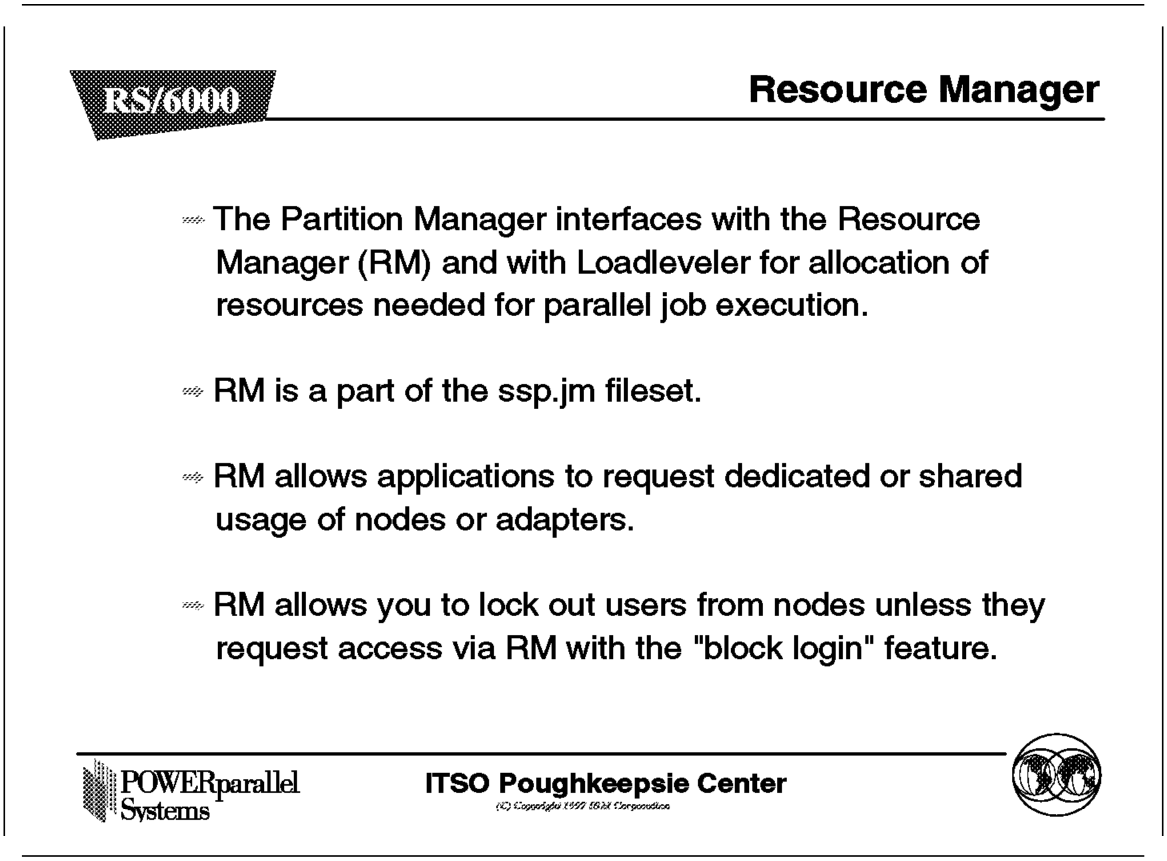
– The Resource manager request is handled with the MP_RESD and MP_RMPOOL environment variables, or at POE invocation, `poe -resd -rmpool`.
– The host list file request is handled with the MP_HOSTFILE environment variable, or at POE invocation, `poe -hostfile`.

**Notes:**

1. The `-hostlist` determines the name of a host file. Any file specifiers are valid. If not set, the default is the host.list file in the current directory.

2. All the environment variables have a corresponding option with the `poe` command. These options given at POE invocation override their associated environment variable.

## 8.4.7 Resource Manager



RS/6000

**Resource Manager**

- The Partition Manager interfaces with the Resource Manager (RM) and with Loadleveler for allocation of resources needed for parallel job execution.
- RM is a part of the ssp.jm fileset.
- RM allows applications to request dedicated or shared usage of nodes or adapters.
- RM allows you to lock out users from nodes unless they request access via RM with the "block login" feature.

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The Parallel Operating Environment does not allocate any nodes differently than those specified in the host list file. The Resource Manager allows you to specify a group of nodes that meet specific criteria like CPU, memory, or disk.

If the Parallel Operating Environment is not used, you do not need to install the Resource Manager.

### 8.4.7.1 Dedicated or shared usage of resources

One consideration with parallel computing is to reduce the execution time of a program. To achieve this, you may need to reserve the exclusive use of resources to some jobs. Resource Manager allows you to request shared or dedicated use of a node or adapter. If the node is shared, the adapter can be shared or dedicated. Requesting dedicated use of an adapter stops other jobs from using this adapter. It does not stop the node from being used by another job using a different adapter.

The switch adapter can run only one application in *user space* (us) mode but it can be used by other jobs being used in IP mode. So, it is necessary to specify dedicated use of the switch adapter in user space if needed.

As a dedicated use of adapter or node limits usage of these resources to one job, the cost of this resource is likely to be different than the cost of shared use. The RS/6000 SP accounting system allows you to handle separately a job requesting exclusive use of a node or adapter. The Resource Manager generates specific accounting records for the user ID of these jobs, so they can be charged differently. The RS/6000 SP accounting is described in detail in the *Administration Guide*, GC23-3897.

#### 8.4.7.2 Login control

Login control is used to dynamically prevent interactive login of users on a node basis. Preventing interactive login of users on nodes running parallel jobs may be desirable for performance purposes.

Login control is intended for use with the Resource Manager. It updates dynamically the /etc/security/user file to disallow all of the following types of interactive access:

- login
- rlogin
- AIX rsh
- AIX rcp
- AIX rexec
- SP rsh
- SP rcp

Login Control unblocks the node access on a request of Resource Manager.

For advanced security, the system administrator can disallow users from using `ftp` on a node by placing users in the /etc/ftpusers file. This file can be kept in a file collection and distributed to the appropriate nodes.

**ATTENTION**

- Since the /etc/security/user file is used by Login Control to state information, this file becomes machine-dependant and should not be overwritten. The /etc/security/user file must not be distributed through file collections or any other mechanism.
- The Login Control utility will not prevent *LoadLeveler* from running jobs submitted by blocked users. *LoadLeveler* logs in as `root` and then switches to the users. Root is never blocked on a node.

Login Control is described in detail in the *Administration Guide*, GC23-3897.

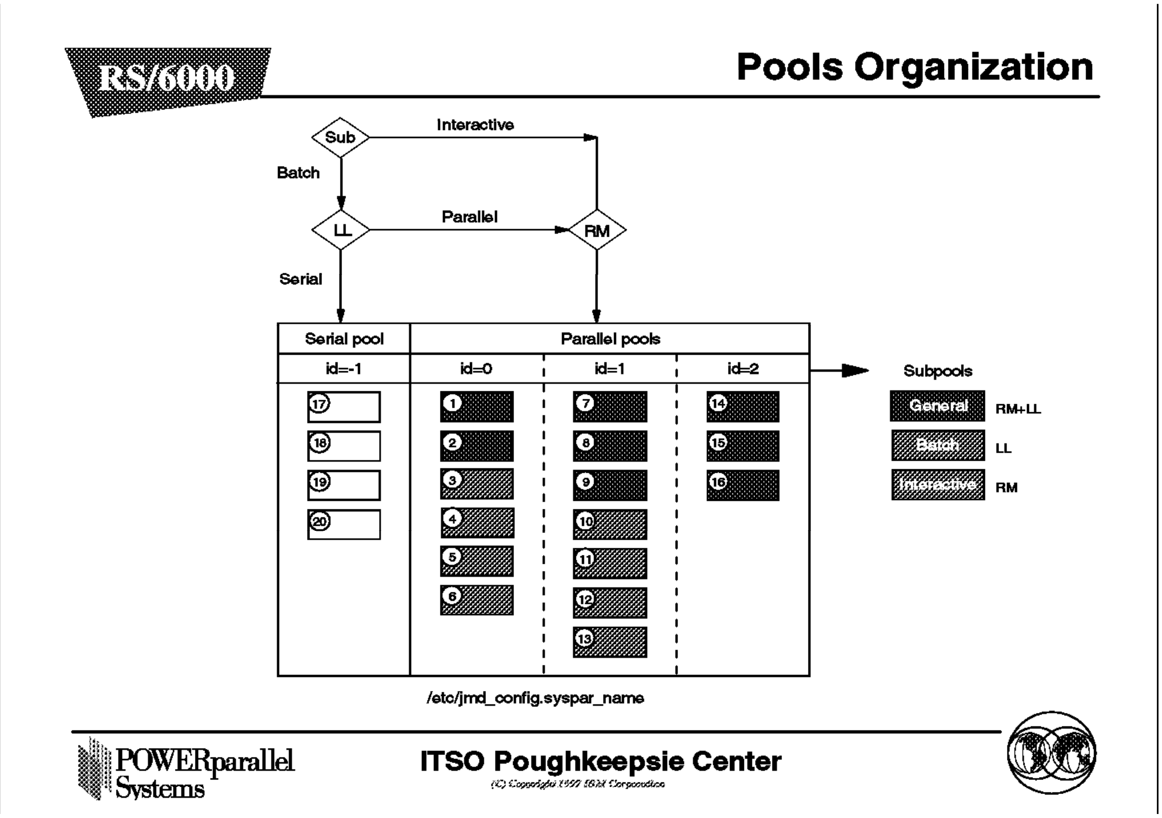
#### 8.4.7.3 Recovery

When the Resource Manager is started, it automatically tries to start a backup on another node on the list. If the primary Resource Manager fails, this backup becomes the primary and starts a backup on another node. Jobs will continue to run unaffected. If the primary and the backup die at the same time, there is no recovery. You must:

- Wait running parallel jobs terminate.
- Reset the user node access with the `spacs_cntrl` command if the access control is configured.
- Restart the Job Manager with the `jm_start` command.

The Resource Manager requires that the System Data Repository (SDR) is operational to get the primary server information. The Resource Manager will no longer be usable, and will eventually die when the Control Workstation is down.

### 8.4.8 Pools Organization



Pools Organization

| Serial pool | Parallel pools | | |
|---|---|---|---|
| id=-1 | id=0 | id=1 | id=2 |
| 17 | 1 | 7 | 14 |
| 18 | 2 | 8 | 15 |
| 19 | 3 | 9 | 16 |
| 20 | 4 | 10 | |
| | 5 | 11 | |
| | 6 | 12 | |
| | | 13 | |

Subpools

| | |
|---|---|
| General | RM+LL |
| Batch | LL |
| Interactive | RM |

/etc/jmd_config.syspar_name

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

LoadLeveler always requests shared use of nodes, or adapters on nodes, unless the switch adapter is used in user space mode. In that case it requests shared use of the node, but dedicated use of the adapter.

The Resource Manager functions, as described , are supported within individual system partitions. The Resource Manager operates within the scope of each system partition.

Each partition may have its own Resource Manager server, but no functionality crosses system partition boundaries. The Resource Manager gets the node and adapter information for each system partition from the *System Data Repository* (SDR). Each system partition has its own configuration database named on the Control Workstation, the /etc/jmd_config.syspar_name file.

- Serial Nodes
  The Resource Manager does not allocate serial nodes; however, you can record:
  - The *batch serial nodes* where the batch serial jobs run
  - The *interactive serial nodes* where the users are allowed to log into
  - The *general serial nodes* where either of the above will be allowed
- Parallel Nodes
  - *Parallel pool* is a group of parallel nodes, since parallel applications indicate their resource requirements by specifying poll numbers.

- *Parallel subpool* is a subgroup of a parallel pool where the jobs are allowed to run in:
    - A *batch subpool* in which parallel jobs are submitted through LoadLeveler.
    - An interactive subpool in which parallel jobs are started interactively.
    - A general subpool in which jobs are submitted in either way.

- Example configuration

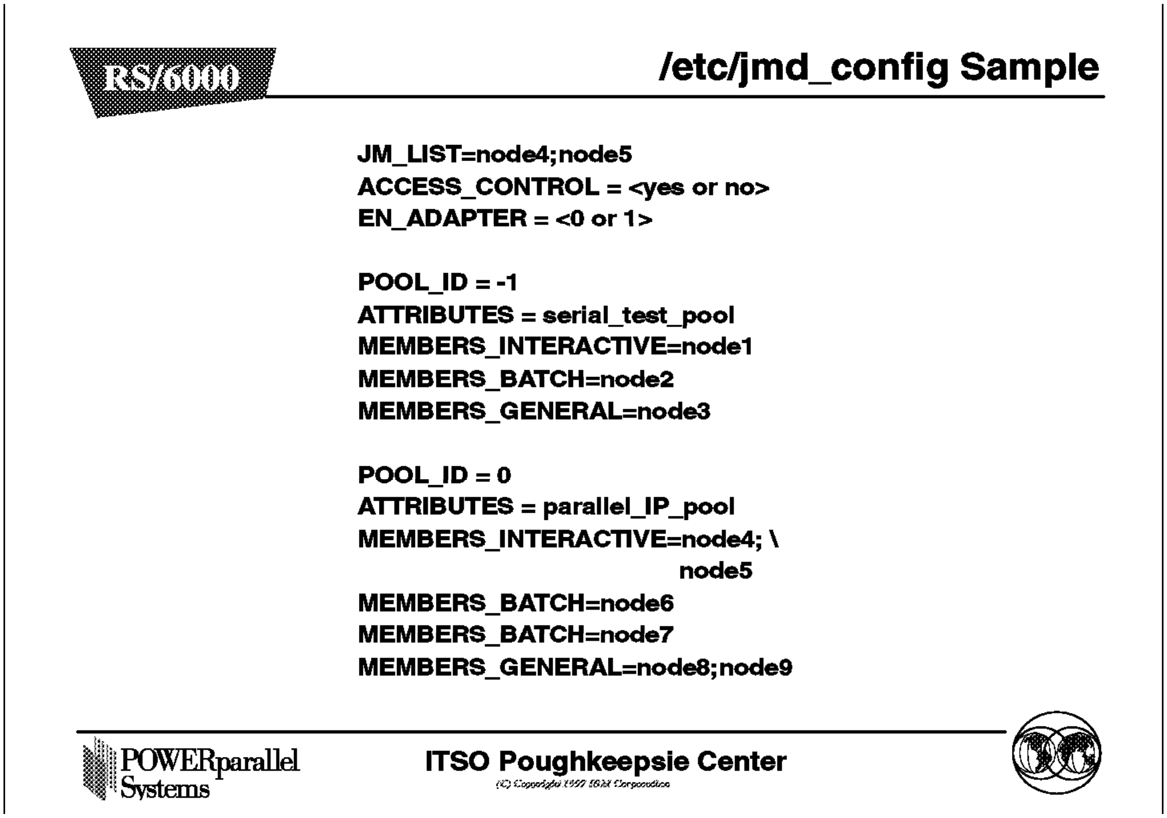
  The system has three parallel pools in the figure preceding:

  - The pool with id 0 has 6 nodes:
    - General subpool: nodes 1-2
    - Batch subpool: nodes 3-4
    - Interactive subpool: nodes 5-6
  - The pool with id 1 has 7 nodes:
    - General subpool: nodes 7-8-9
    - Batch subpool: nodes 10-11
    - Interactive subpool: nodes 12-13
  - The pool with id 2 has 3 nodes:
    - General subpool: nodes 14-15-16
  - If an interactive job requested two nodes from pool 1, It would receive nodes 12 and 13, if they are available.
  - If an interactive job requested four nodes from pool 0, it would receive nodes 1, 2, 5 and 6.
  - If a batch job requested two nodes from pool 2, it would receive two out of nodes 14, 15, 16.
  - if a batch job requested five nodes from pool 1, it would receive nodes 7, 8, 9, 10 and 11.

**Notes:**

- There are additional LoadLeveler configurations to perform interaction with the Resource Manager. Details are described in *Using and Administering LoadLeveler*.
- The supported LoadLeveler version is 1.3.0.
- Any user can use the jm_status command to get information about defined pools or about jobs running on nodes allocated by the Resource Manager

### 8.4.9 /etc/jmd_config Sample



RS/6000

/etc/jmd_config Sample

```
JM_LIST=node4;node5
ACCESS_CONTROL = <yes or no>
EN_ADAPTER = <0 or 1>

POOL_ID = -1
ATTRIBUTES = serial_test_pool
MEMBERS_INTERACTIVE=node1
MEMBERS_BATCH=node2
MEMBERS_GENERAL=node3

POOL_ID = 0
ATTRIBUTES = parallel_IP_pool
MEMBERS_INTERACTIVE=node4; \
                                        node5
MEMBERS_BATCH=node6
MEMBERS_BATCH=node7
MEMBERS_GENERAL=node8;node9
```

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The configuration file /etc/jmd_config.syspar_name is read with the `jm_start` command which starts the Resource Manager server. Whenever the configuration file is changed to modify the node allocation in the pools, the Resource Manager has to be reconfigured with the `jm_config` command. You can modify the configuration by editing the file or with the SMIT menus. The configuration file resides on the Control Workstation and there must be one configuration file for each partition in the system. The suffix *syspar_name* is the *hostname* value of the Control Workstation partition addressed.

If the System Data Repository (SDR) is modified to reflect changes in the nodes or adapters attributes, the Resource Manager must be stopped with `jm_stop`, and restarted with `jm_start`.

The modification (enable or disable) of the SP Exclusive Use Accounting needs only a reconfiguration of a running Resource Manager with `jm_config`.

The statements to be defined in the configuration file are described as follows:

- JM_LIST

  This contains the list of RS/6000 SP host names that are candidates to run the Job Manager server daemon. The primary and secondary Job Manager server location is taken from this list. Any node is suitable to handle the Job Manager server, since The `jmd` daemon does not use CPU resource. You should define at least one primary and one secondary. Since no recovery occurs when the primary and the secondary server die at the same time, it may be desirable to define them on two nodes from different frames.

- ACCESS_CONTROL

  The Job Manager uses the Access Control Management tool (spacs_cntrl command) to allow user access to a parallel node while it is allocated. The Access Control Management tool resides in the Control Workstation (CWS) and has entries in the System Data Repository (SDR).

- EN_ADAPTER

  This selects the Ethernet adapter network that will be used to allocate resources. Valid adapters are either en0 or en1.

- POOL_ID

  This describes the nodes belonging to a same group:

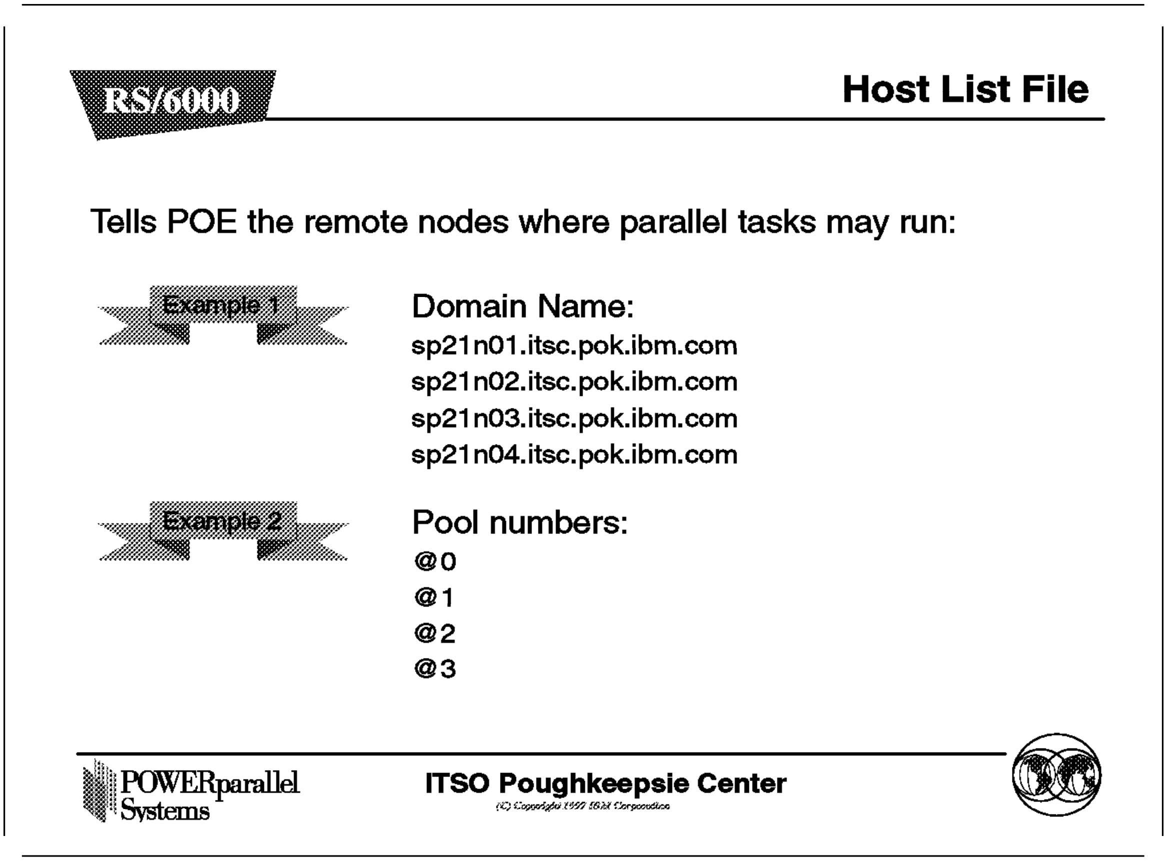
  - One node can be in only one pool and only one subpool within that pool. Since aliases can be used to refer to the same node, you will receive an error if different names refer the same node.
  - There can only be one serial pool. Its id must be -1.
  - Multiple parallel pools can be defined. The id can be 0 or greater.

- ATTRIBUTES

  This is mandatory. The ATTRIBUTES definition must be a string without blanks or ″=″ character.

- The different stanzas available for MEMBERS_INTERACTIVE, MEMBERS_BATCH or MEMBERS_GENERAL pools are described in the preceding figure.

### 8.4.10 Host List File



RS/6000

**Host List File**

**Tells POE the remote nodes where parallel tasks may run:**

Example 1

**Domain Name:**
sp21n01.itsc.pok.ibm.com
sp21n02.itsc.pok.ibm.com
sp21n03.itsc.pok.ibm.com
sp21n04.itsc.pok.ibm.com

Example 2

**Pool numbers:**
@0
@1
@2
@3

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

A *host list* specifies the processor nodes on which the individual tasks of a program should run. The host list file must be created if the program needs a specific node allocation or a non-specific node allocation from one or more pool. When a parallel program is invoked, the Partition Manager checks if there is a host list file specified with:

- The MP_HOSTFILE environment variable. If the variable is set to an empty string (“”) or to the word “NULL,” it means that no host list file should be used. If MP_HOSTFILE is not set, POE looks for a default `host.list` file in the current directory.
- The `poe` command flags `-hostfile` or `-hfile`. These flags override MP_HOSTFILE settings.

The host list file can contain:

- Comment lines beginning with ! or *
- A list of nodes given by name or by IP address
- One or more pool number from the Resource Manager. The pool number must have the prefix `@`.

It cannot contain a *mixture* of node and pool requests, so use one method or the other.

If the file exists, the Partition Manager reads it to allocate the nodes with the following rules:

- Example 1:
  In this example, the nodes are defined with their names. This means the program requests a specific allocation. Depending on the number of tasks needed by the program, the node allocations are as follows:

  1. Four tasks are needed:
     The partition manager allocates task 0 to sp21n01, task 1 to sp21n02, task 3 to sp21n03, and task 3 to sp21n04.

  2. Two tasks are needed:
     The partition manager allocates task 0 to sp21n01 and task 1 to sp21n02. The two remaining entries are ignored.

  3. Six tasks are needed:
     The first four tasks are allocated as in item 1. The remaining tasks (4 and 5) are allocated to the last entry, sp2n04.

  4. Multiple tasks can share the same node by listing the same node multiples times:

     ```
     sp21n01
     sp21n02
     sp21n01
     sp21n02
     sp21n01
     sp21n02
     ```

     In this example, tasks 0, 2, and 4 will share sp21n01, while tasks 1, 3, and 5 will share sp21n02.

- Example 2:

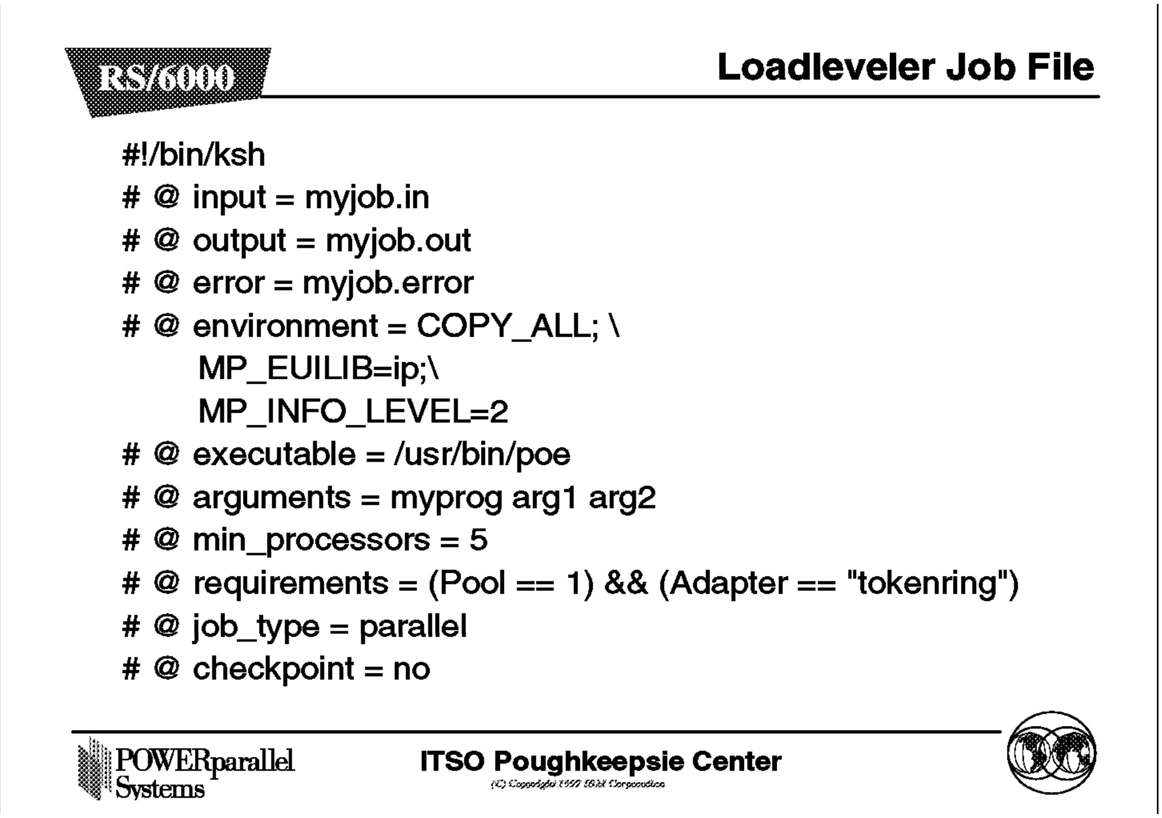
  - The same rules apply. The only difference is that the Partition Manager allocates the first task to a non-specific node from the pool 0, the second task to a non-specific node from the pool 1, and so on.

  - The following example requests four nodes from pool 0 and two nodes from pool 1:

    ```
    @0
    @0
    @0
    @0
    @1
    @1
    ```

    If there are insufficient resources in a requested pool. The Partition Manager returns a message stating this and does not run the program.

## 8.4.11 LoadLeveler Job File



RS/6000

Loadleveler Job File

```
#!/bin/ksh
# @ input = myjob.in
# @ output = myjob.out
# @ error = myjob.error
# @ environment = COPY_ALL; \
      MP_EUILIB=ip;\
      MP_INFO_LEVEL=2
# @ executable = /usr/bin/poe
# @ arguments = myprog arg1 arg2
# @ min_processors = 5
# @ requirements = (Pool == 1) && (Adapter == "tokenring")
# @ job_type = parallel
# @ checkpoint = no
```

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Here is a brief sample of the LoadLeveler job file.

To submit a POE job to LoadLeveler, you need to build a LoadLeveler job file, which specifies:

- The number of nodes to be allocated.
- Any POE options, passed via environment variables using LoadLeveler′s environment statement, or passed as command line options using LoadLeveler′s arguments statement.
- The path to your POE executable (usually /usr/bin/poe).

The following POE environment variables, or associated command line options, are validated but not used for jobs validated via LoadLeveler. These variables have a corresponding statement in the LoadLeveler environment:

- MP_PROCS
- MP_RMPOOL
- MP_EUIDEVICE
- MP_HOSTFILE
- MP_SAVEHOSTFILE
- MP_PMDSUFFIX
- MP_RESD
- MP_RETRY
- MP_RETRYCOUNT
- MP_ADAPTERUSE

- MP_CPU_USE

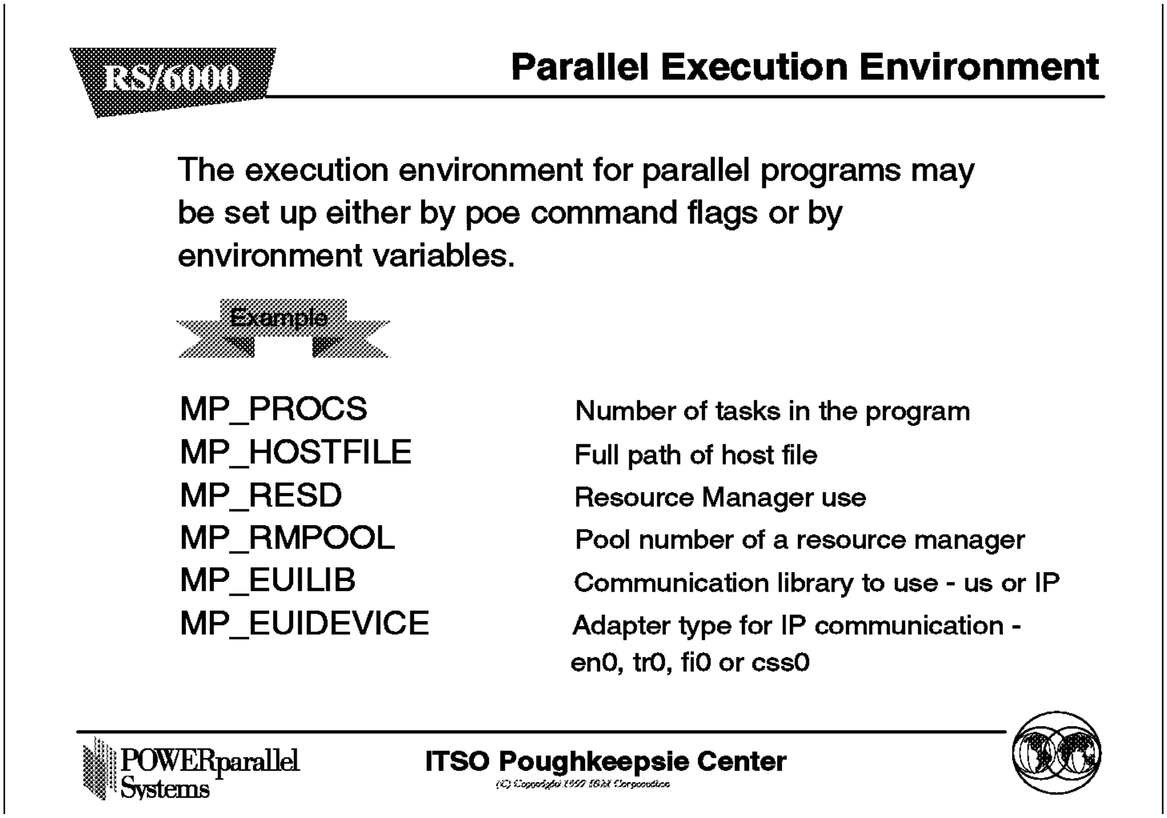
For example, since LoadLeveler has its own pool of nodes defined in the /etc/jmd_config, the environment variables MP_PROCS, MP_RMPOOL, and MP_HOSTFILE are meaningless.

The preceding figure shows a sample of the LoadLeveler job file. It allows you to run myprog on five nodes from pool 1, using a Token Ring adapter for IP message passing. The arguments arg1 and arg2 are passed to myprog.

**Notes:**

1. Parallel Environment Version 2.3 is only compatible with LoadLeveler Version 1.3.0 or later.
2. When LoadLeveler allocates nodes for parallel execution, POE and one of the parallel tasks will be executed on the same node, but it is not guaranteed to be task 0.
3. When LoadLeveler detects a condition that should terminate the parallel job, aSIGTERM is sent to POE. Then POE sends the SIGTERM to each parallel task in the partition. If this signal is caught or ignored by a parallel task, LoadLeveler will terminate the task.
4. Programs that use the US protocol must have the LoadLeveler requirements statement specifying Adapter=″hps_user″.

### 8.4.12 Parallel Execution Environment



RS/6000

**Parallel Execution Environment**

The execution environment for parallel programs may be set up either by poe command flags or by environment variables.

Example

| | |
|---|---|
| MP_PROCS | Number of tasks in the program |
| MP_HOSTFILE | Full path of host file |
| MP_RESD | Resource Manager use |
| MP_RMPOOL | Pool number of a resource manager |
| MP_EUILIB | Communication library to use - us or IP |
| MP_EUIDEVICE | Adapter type for IP communication - en0, tr0, fi0 or css0 |

POWERparallel Systems

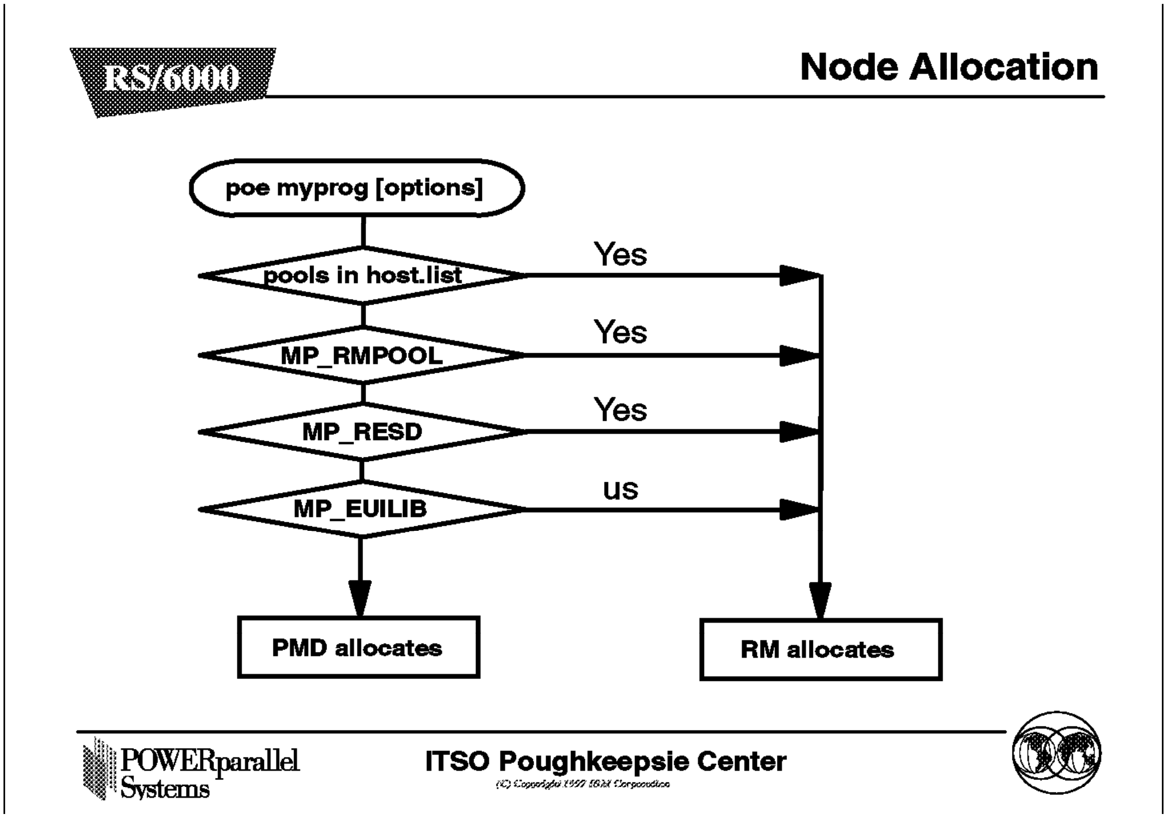
ITSO Poughkeepsie Center

Before invoking a program, the execution environment must be set up. This section details the most important environment variables necessary for program invocation. Each time a parallel program is run, the home partition manager checks these variables to determine the number of tasks required and how to allocate the processor nodes for these tasks.

- MP_PROCS determines the number of program tasks. If not set, the default is 1.
- MP_HOSTFILE determines the name of the host list file to use for node allocation. If the variable is set to an empty string (“”) or to the word “NULL,” it means that no host list file should be used. If MP_HOSTFILE is not set, POE looks for a default host.list file in the current directory.
- MP_RESD determines whether or not the Partition Manager should connect to the Resource Manager to allocate the nodes. MP_RESD only specifies whether or not to use the Resource Manager. The Resource Manager to use must be defined by setting the variable SP_NAME to the name of the Control Workstation. There are as many Control Workstation names as partitions on a system.
- MP_EUILIB specifies the communication subsystem library implementation to use, either the IP communication subsystem or the User Space (US) communication subsystem.

- MP_EUIDEVICE specifies the adapter set used for IP communication among the nodes. This variable is only checked if the IP communication subsystem implementation is used on the Resource Manager. If MP_RESD=no, the value of MP_EUILIB is ignored.
- MP_RMPOOL specifies the number of a Resource Manager pool. This variable is checked only if the Resource Manager is used without a host file.

### 8.4.13 Node Allocation



RS/6000

Node Allocation

poe myprog [options]

pools in host.list — Yes

MP_RMPOOL — Yes

MP_RESD — Yes

MP_EUILIB — us

PMD allocates

RM allocates

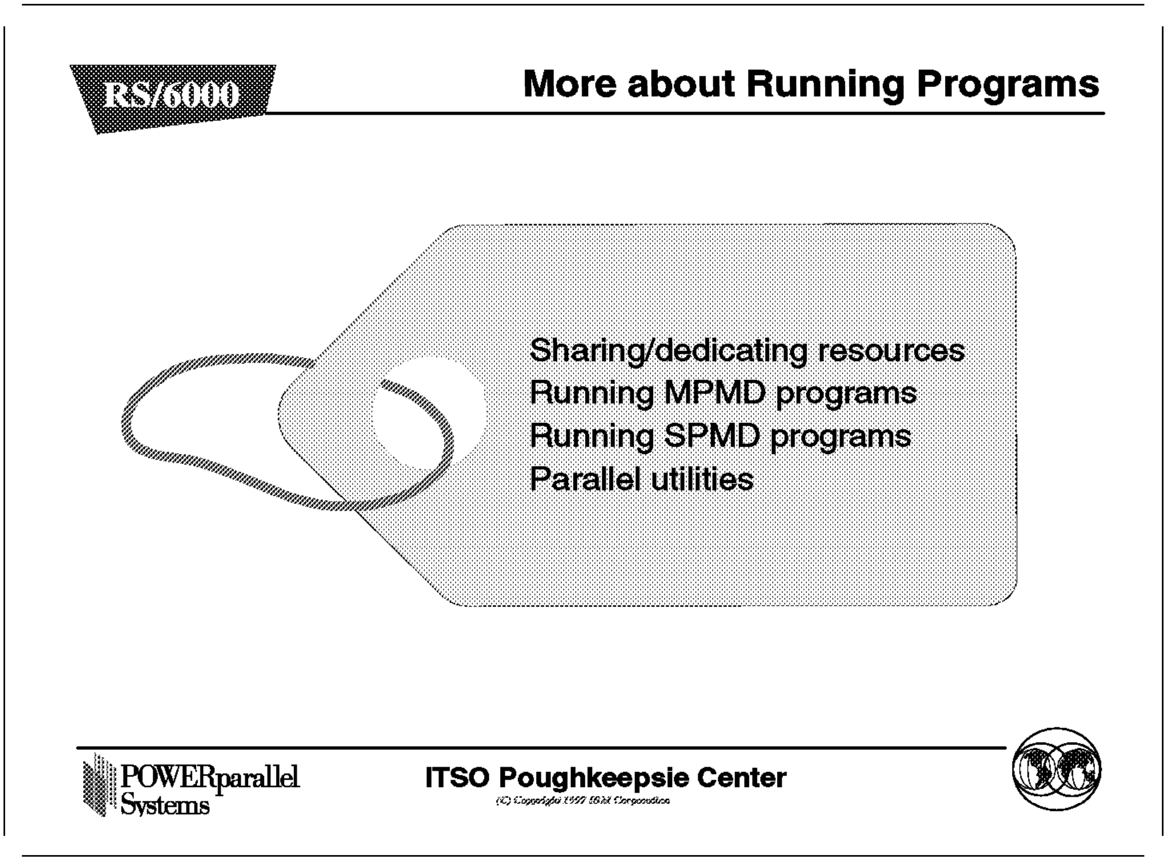
POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This figure briefly summarizes the way node allocation is done between the Partition Manager and the Resource Manager. The MP_EUIDEVICE does not appear in the figure, because it does not influence the node allocation.

## 8.5 More about Running Programs



RS/6000

More about Running Programs

Sharing/dedicating resources
Running MPMD programs
Running SPMD programs
Parallel utilities

POWERparallel Systems

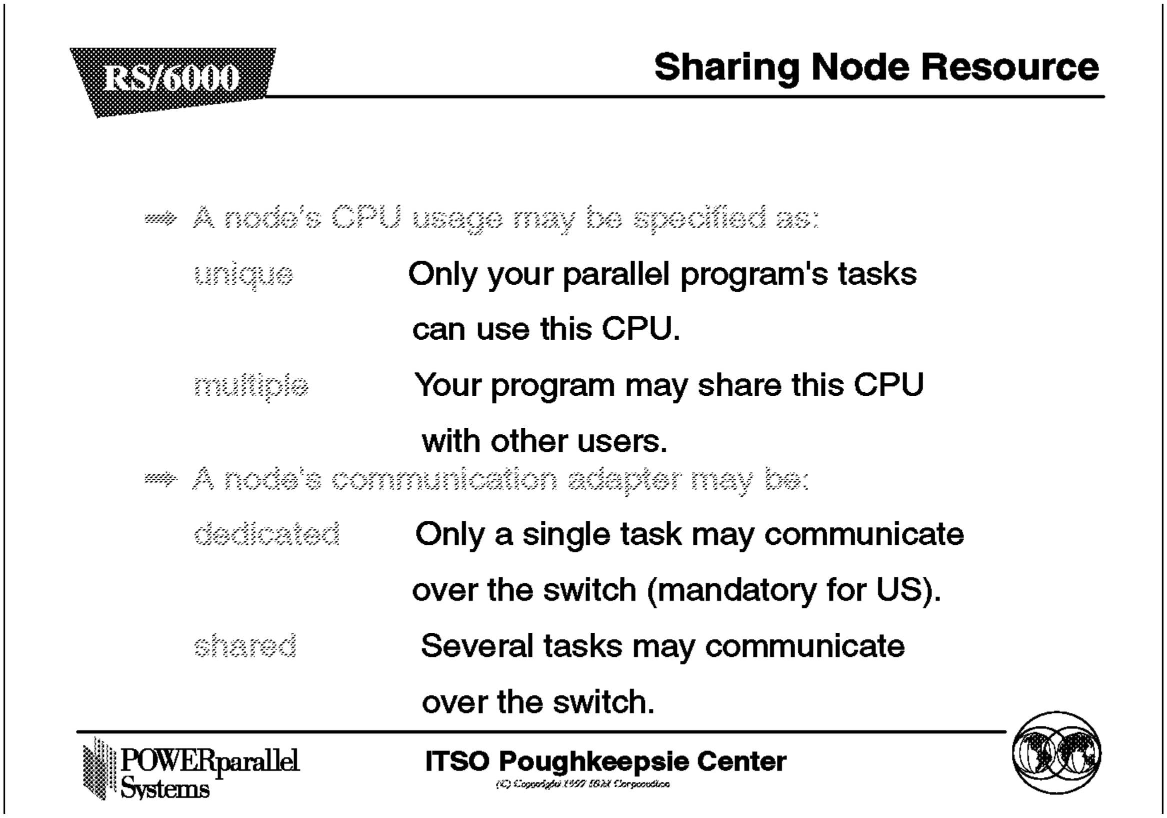
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In this section, we show with some examples how the program execution is controlled with environment variables or POE flags. The topics covered are the control of:

- Node and adapter usage
- SPMD and MPMD applications
- Outputs

### 8.5.1 Sharing Node Resource



RS/6000

**Sharing Node Resource**

- A node's CPU usage may be specified as:

| | |
|---|---|
| unique | Only your parallel program's tasks can use this CPU. |
| multiple | Your program may share this CPU with other users. |

- A node's communication adapter may be:

| | |
|---|---|
| dedicated | Only a single task may communicate over the switch (mandatory for US). |
| shared | Several tasks may communicate over the switch. |

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Let us remember the discussion about the dedicated or shared usage of resources in the section 8.4.7, "Resource Manager" on page 434

One consideration with parallel computing is to reduce the execution time of a program. To achieve this, you may need to reserve the exclusive use of resources to some jobs. The Resource Manager allows you to request shared or dedicated use of a node or an adapter. If the node is shared, the adapter can be shared or dedicated. Requesting dedicated use of an adapter stops other jobs from using this adapter. It does not stop the node from being used by another job using a different adapter if the node is defined as *multiple*.

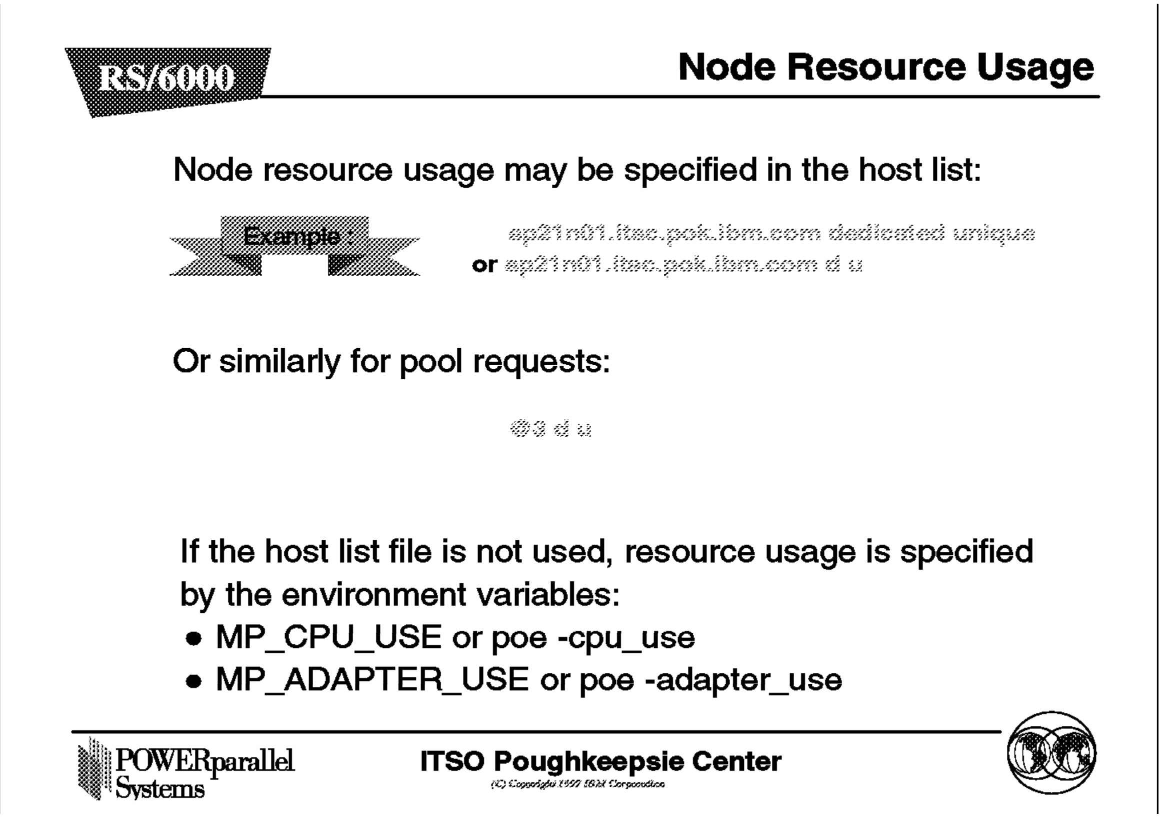
The User Space (US) communication library requires dedicated use of the high performance switch. If you are using the US communication subsystem for communication among processor nodes, POE forces adapter use to be dedicated. If you are using the US and you specify the switch adapter to be shared, the specification is ignored. The US use is determined by the settings: `MP_EUILIB=us` or `poe -euilib`. However, the adapter can be shared with IP communication among the nodes.

In other words, all the nodes should be able to run:

- The IP protocol, or
- The US protocol, or
- The IP and US protocols concurrently

While US message passing programs must use the RM to allocate nodes, IP message passing programs maybe use the RM, but are not required to.

### 8.5.2 Node Resource Usage



RS/6000

Node Resource Usage

Node resource usage may be specified in the host list:

Example : sp21n01.itso.pok.ibm.com dedicated unique
or sp21n01.itso.pok.ibm.com d u

Or similarly for pool requests:

@3 d u

If the host list file is not used, resource usage is specified by the environment variables:

- MP_CPU_USE or poe -cpu_use
- MP_ADAPTER_USE or poe -adapter_use

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The nodes and adapters usage can be specified in the host list file on a node or a pool id, as follows:

- The first word is the node name or pool id.
- The second word represents the adapter usage, dedicated or shared.
- The third word represents the node usage, unique or multiple.

If the host list file is not used, resource usage is specified with the following settings:

- MP_ADAPTER_USE or `poe -adapter_use`
- MP_CPU_USE or `poe -cpu_use`

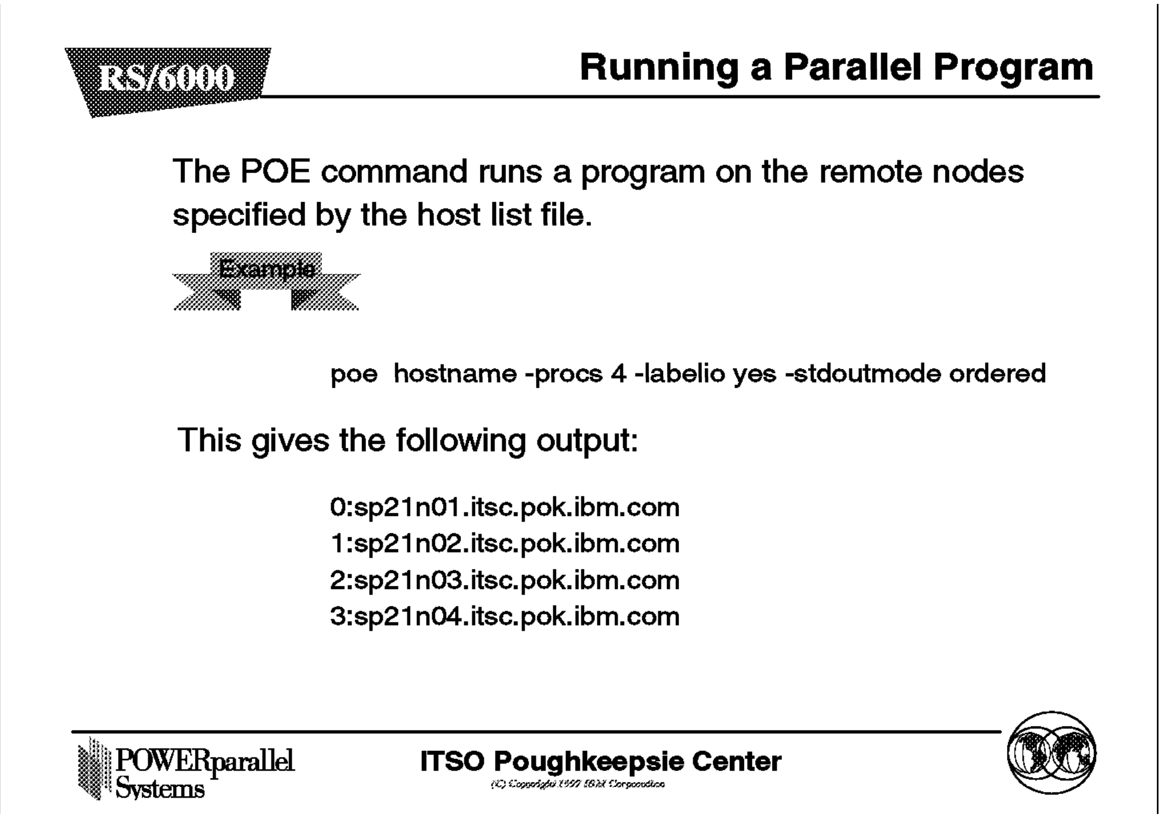
Here are two examples of settings, and the resulting node and switch adapter usage:

1. MP_EUIDEVICE=css0, MP_EUILIB=us, node_1 dedicated multiple:
   These settings imply that the adapter can run only one US mode communication and the CPU can be shared.
2. MP_EUIDEVICE=css0, MP_EUILIB=ip, node_1 shared multiple
   These settings imply that, while the first job is running, a second POE user requests a shared adapter use for IP communication mode and a shared CPU use.

In both cases, the Partition Manager asks the Resource Manager for adapter and CPU usage. Then the Resource Manager allocates the nodes with the necessary requirements. If the second job′s requirements (cpu unique instead of multiple) conflict with the allocation of the first, the Resource Manager refuses the allocation and the second job fails.

**Note:** Programs that use LAPI must set `MP_EUILIB=us` or `poe -euilib us`.

### 8.5.3 Running a Parallel Program



RS/6000

## Running a Parallel Program

The POE command runs a program on the remote nodes specified by the host list file.

Example

```
poe  hostname -procs 4 -labelio yes -stdoutmode ordered
```

This gives the following output:

```
0:sp21n01.itsc.pok.ibm.com
1:sp21n02.itsc.pok.ibm.com
2:sp21n03.itsc.pok.ibm.com
3:sp21n04.itsc.pok.ibm.com
```

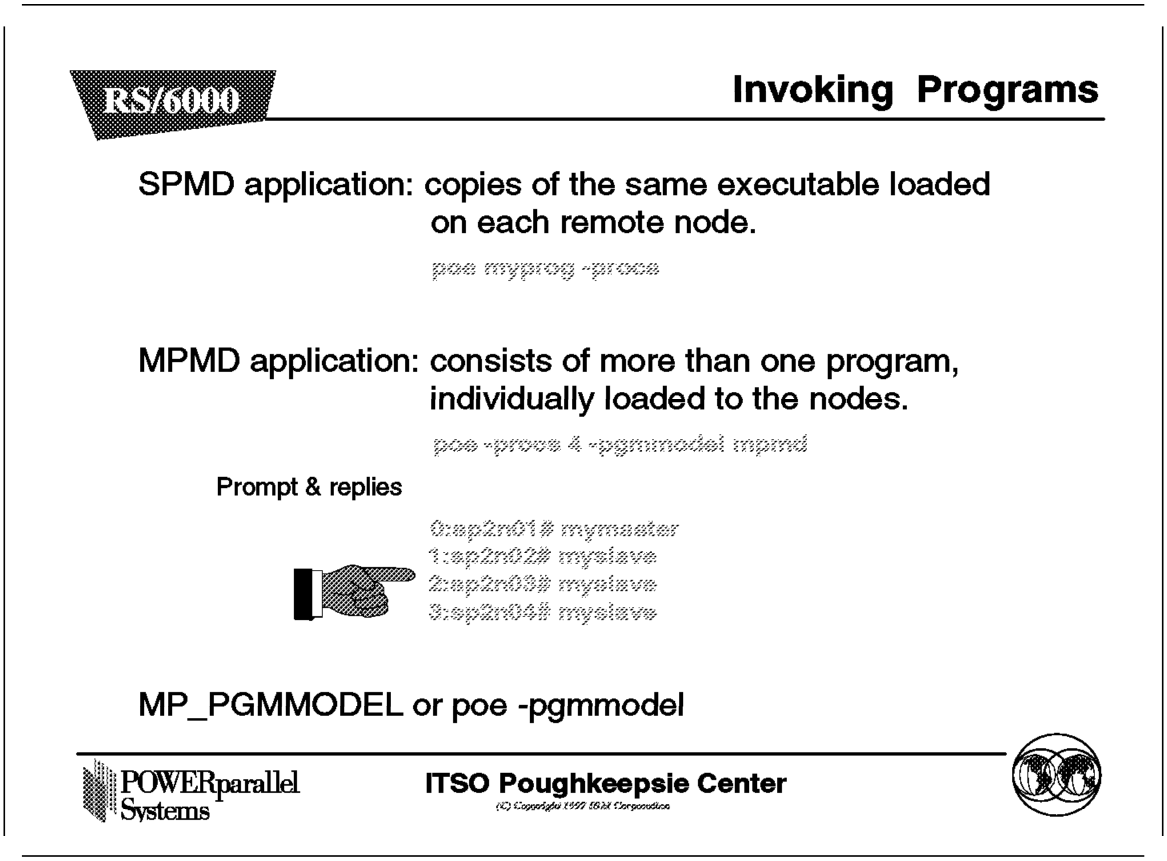
POWERparallel
Systems

ITSO Poughkeepsie Center

The example in the preceding figure shows how to run a program. The POE command runs the command hostname on four processors. The host list file is given by the environment variable MP_HOSTFILE, or if not set, the POE found a host.list file in the current directory with the four nodes. The hostname command has been executed on node 1 as task 0, and returned the host name: sp21n01.itsc.pok.ibm.com.

- The MP_LABELIO environment or the flag -labelio gives the parallel task outputs labeled by task id. This ability can be very useful when the outputs are generated in *ordered* or *unordered* mode. In an unordered mode, if the tasks are not labeled, you would be unable to determine which task sends which output.
- The MP_STDOUTMODE environment variable or the flag -stdoutmode allows you to specify the way outputs are written, as follows:
  - By task id
    task 0 means that only this task will write an output to STDOUT.
  - Ordered
    In this mode, each task writes output data in its own buffer. All the task buffers are later flushed in order of task id, to STDOUT.
  - Unordered
    In this mode, the tasks write their output in an asynchronous mode, as they execute. This mode can be useful to save buffer space.

## 8.5.4 Invoking Programs



RS/6000

**Invoking Programs**

SPMD application: copies of the same executable loaded
on each remote node.

```
poe myprog -procs
```

MPMD application: consists of more than one program,
individually loaded to the nodes.

```
poe -procs 4 -pgmmodel mpmd
```

Prompt & replies

```
0:sp2n01# mymaster
1:sp2n02# myslave
2:sp2n03# myslave
3:sp2n04# myslave
```

MP_PGMMODEL or poe -pgmmodel

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

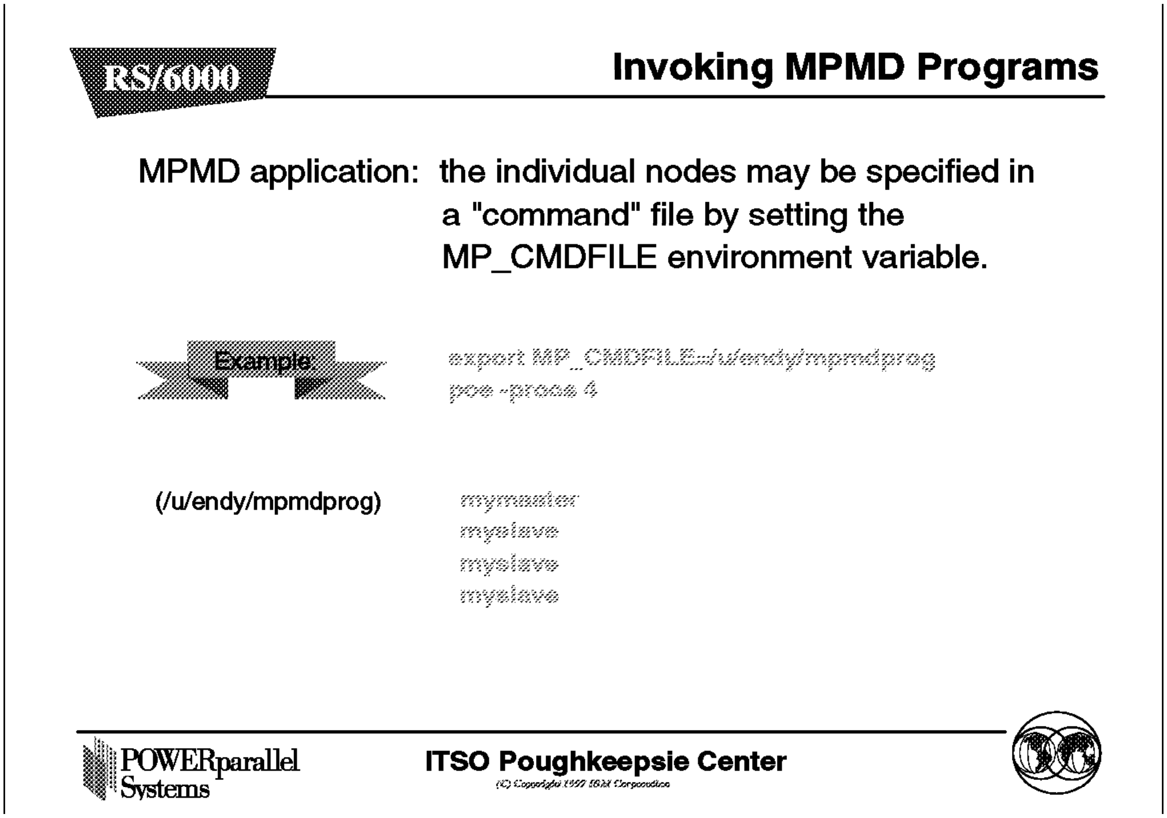
The two different programming models are Single Program Multiple Data (SPMD) and Multiple Program Multiple Data (MPMD). With an SPMD application, a copy of the same executable is sent to, and runs on, each of the processor nodes of your partition. If you are invoking an MPMD application, you are dealing with more than one program and need to individually load the nodes of your partition.

Because the execution differs, the programming model used must be specified with the MP_PGMMODEL environment variable or the `-pgmmodel` flag. The default programming model is SPMD.

In the MPMD example shown in the figure, there are two programs, `mymaster` and `myslave`, designed to run together and communicate via calls to message-passing subroutines. The program `mymaster` is designed to run on one processor node. The `myslave` program is designed to run as separate tasks on any number of other nodes. The `mymaster` program will coordinate and synchronize the execution of the `myslave` tasks.

With the command `poe -procs 4`, a partition of four nodes is established and you are prompted to load the tasks individually on the nodes.

### 8.5.5 Invoking MPMD Programs



RS/6000

**Invoking MPMD Programs**

MPMD application: the individual nodes may be specified in a "command" file by setting the MP_CMDFILE environment variable.

Example:

```
export MP_CMDFILE=/u/endy/mpmdprog
poe -procs 4
```

(/u/endy/mpmdprog)

```
mymaster
myslave
myslave
myslave
```

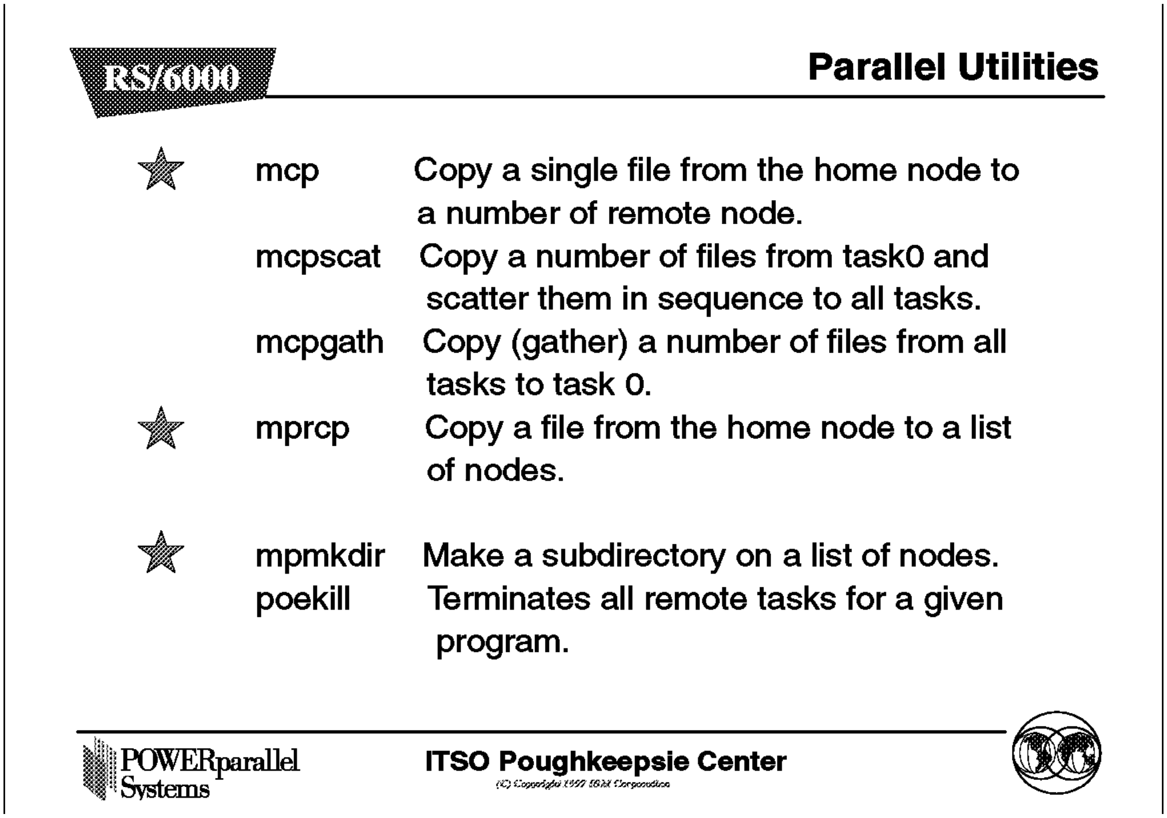
POWERparallel Systems

ITSO Poughkeepsie Center

Rather than loading all the nodes from the keyboard, the MP_CMDFILE environment variable or `-cmdfile` allows you to specify the name of a POE commands file.

The POE commands file /u/endy/mpmdprog lists the individual programs you want to load and run on the four nodes of your partition.

## 8.5.6 Parallel Utilities



RS/6000

**Parallel Utilities**

★ **mcp** Copy a single file from the home node to a number of remote node.

**mcpscat** Copy a number of files from task0 and scatter them in sequence to all tasks.

**mcpgath** Copy (gather) a number of files from all tasks to task 0.

★ **mprcp** Copy a file from the home node to a list of nodes.

★ **mpmkdir** Make a subdirectory on a list of nodes.

**poekill** Terminates all remote tasks for a given program.

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

This section describes the parallel file copy utilities delivered with POE:

- mcp infile [outfile] [POE options]

  This command copies the same file to all tasks. The input file must reside on task 0. You can copy it to a new name on the other task, or to a directory. It accepts a file name as `infile` and a destination file name or directory as `outfile`.

- mcpscat [-i] source ... destination [POE options]

  This command distributes a number of files in sequence to a series of tasks, one at a time. It will use round robin ordering to send the files in a one-to-one correspondence to the tasks. If the number of files exceeds the number of tasks, the remaining files are sent in another round through the tasks.

- mcpgath [-ai] source ... destination [POE options]

  This command is used when you need to copy a number of files from each of the tasks back to a single location, task 0. The files must exist on each task. You can optionally specify to have the task number appended to the file name when it is copied.

**Notes:**

1. All of these utilities are POE programs, therefore they accept any poe command flags as input parameters.
2. These utilities accept the source file names and a destination directory.
3. These utilities use the MPI communication subsystem. The source codes are delivered in the /usr/lpp/ppe.poe/samples/mpi directory, and can be used as programming samples.

- mprcp host_list filename

  This command allows also to copy a single file from one node to a list of nodes. Because it is a simple script using the rcp command, it is more intended for use on workstations on the network, rather than for the RS/6000 SP.

Others utilities are as follows:

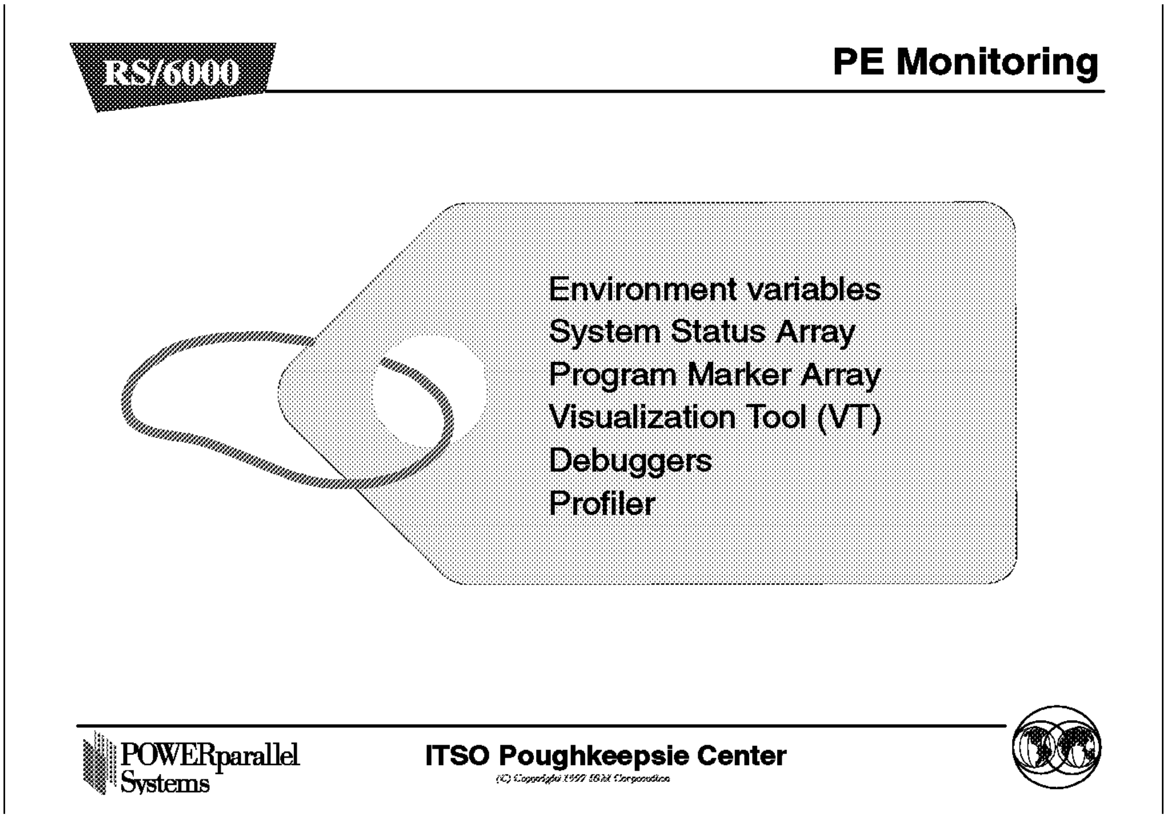
- mpmkdir host_list directory_name

  This script allows you to create directories on remote nodes. It uses the rsh command.

- poekill pgm_name [POE options]

  This script searches for the existence of running programs owned by the user and terminates them via a SIGTERM signals. If run under POE using poe poekill, it uses the standard POE mechanism to identify the remote nodes, host list file, or Resource Manager.

## 8.6 PE Monitoring



RS/6000

PE Monitoring

- Environment variables
- System Status Array
- Program Marker Array
- Visualization Tool (VT)
- Debuggers
- Profiler

POWERparallel Systems

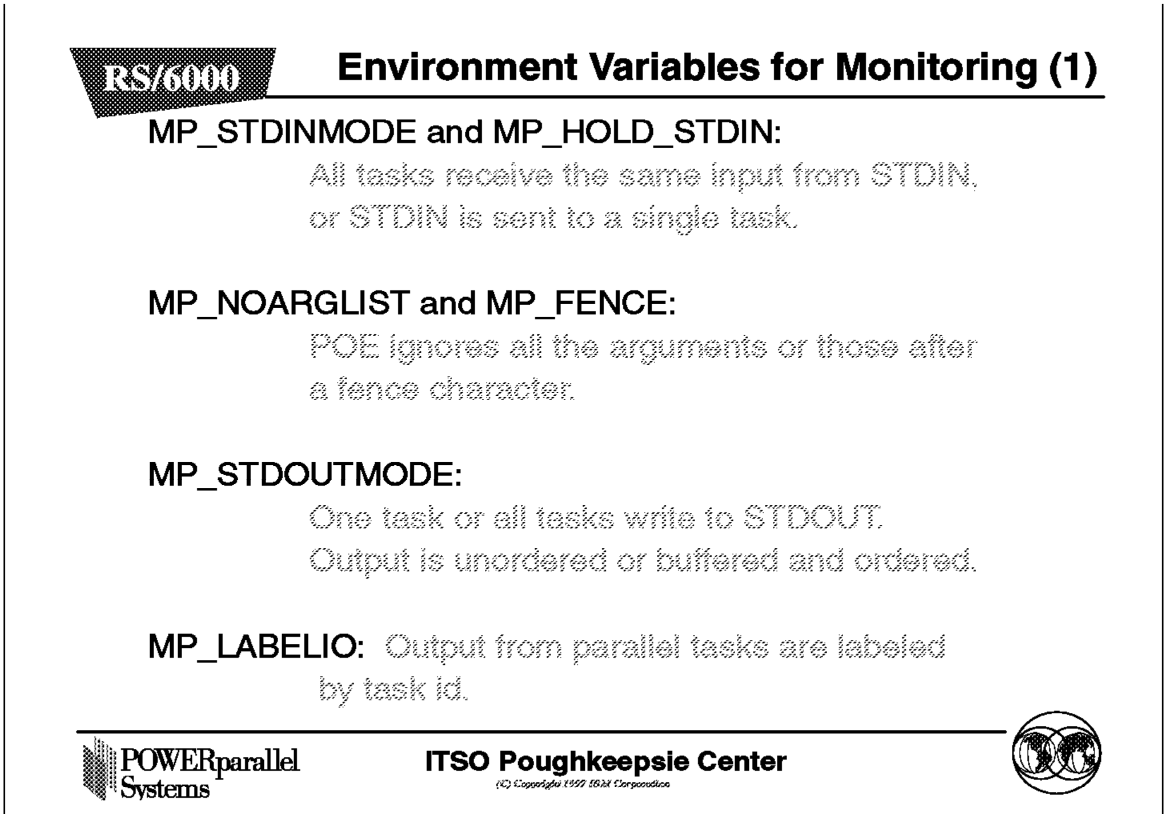
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This chapter presents the tools needed to monitor and debug parallel programs.

We describe some necessary environment variables, as well as tools such as the program marker array, the system status array, the Visualization Tool (VT), and the profiler to monitor and debug programs. Finally, we present the parallel debuggers for debugging purpose.

## 8.6.1 Environment Variables for Monitoring (1)



RS/6000 **Environment Variables for Monitoring (1)**

**MP_STDINMODE and MP_HOLD_STDIN:**

All tasks receive the same input from STDIN, or STDIN is sent to a single task.

**MP_NOARGLIST and MP_FENCE:**

POE ignores all the arguments or those after a fence character.

**MP_STDOUTMODE:**

One task or all tasks write to STDOUT. Output is unordered or buffered and ordered.

**MP_LABELIO:** Output from parallel tasks are labeled by task id.

POWERparallel Systems ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Both this section and the next one describe some necessary environment variables for monitoring. The first part of the environment variables concerns the management of standard input (STDIN) and standard output (STDOUT).

- STDIN
  - MP_STDIN determines how input is managed for the parallel tasks:
    - `All` means that all tasks receive the same input data from STDIN.
    - `None` means that no tasks receive data from STDIN. STDIN will be used by the home node only.
    - A task id, like `task 0`, means STDIN is only sent to the task identified.

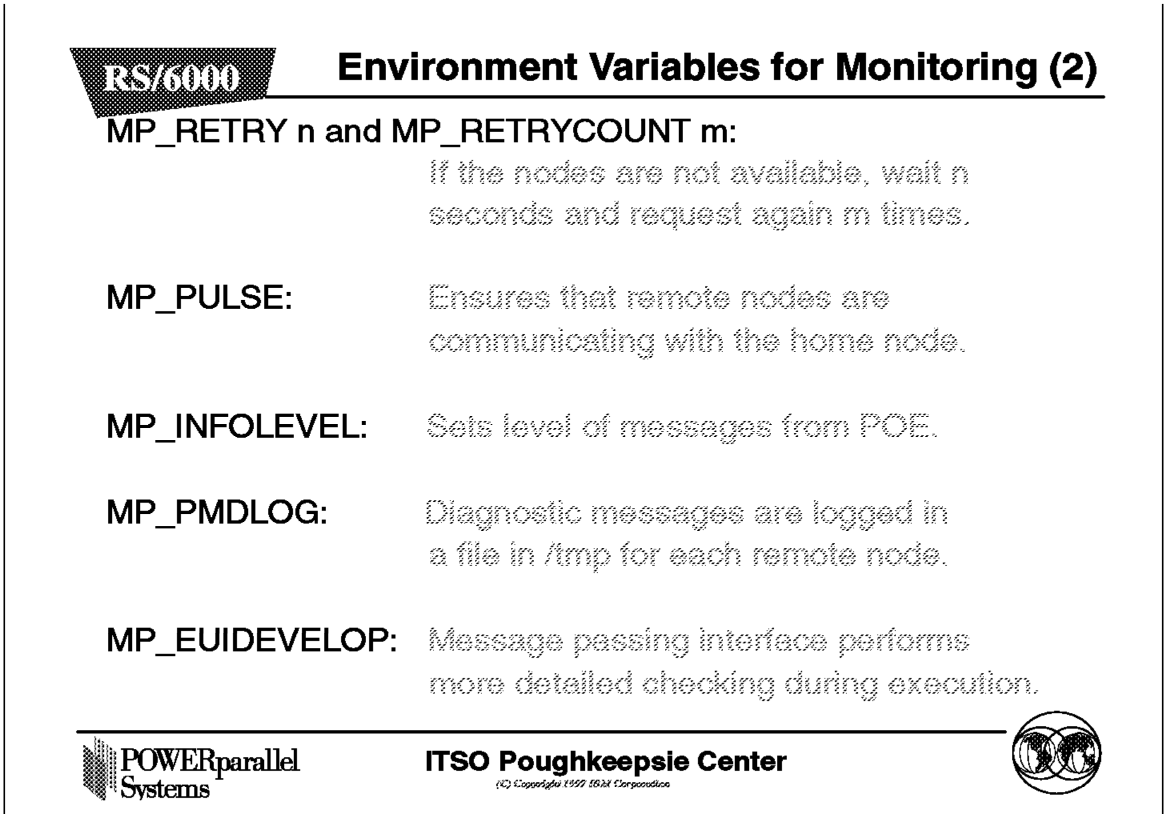
    Usually STDIN refers to the keyboard input. If you use redirection or piping, STDIN could refer to a file or the input from another command.
  - MP_CMDFILE gives a name of a POE commands file used to load the nodes of the partition. If set, POE will read this commands file rather than STDIN.
  - MP_HOLD_STDIN is used to defer the sending of STDIN from the home node to the remote nodes until the message passing library has been initialized. If not set, it could result in a user program hanging.
  - Two other variables can be associated with the setting of MP_STDIN. When you invoke a parallel program, you can specify an argument list with a number of program options and POE flags.

    - MP_NOARGLIST makes POE ignore the entire argument list when the setting is yes.
    - MP_FENCE makes POE ignore a portion of the argument list.
      The following setting, export MP_FENCE=Q, makes POE ignore the portion of the argument list located after the Q character.

- STDOUT
  - MP_STDOUTMODE determines how STDOUT is handled by the parallel task. The valid parameters are as follows:
    - A task id, like task 0, means only the task indicated writes output to STDOUT.
    - Ordered means output data from each parallel task is written to its own buffer. All buffers are flushed later in task order to STDOUT.
    - Unordered means all tasks write output data to STDOUT asynchronously.

    Usually STDOUT refers to the display. In the same way with STDIN, you can use redirection or piping to refer to a file or another command.
  - MP_LABELIO is a variable associated to MP_STDOUT. The parameters are yes or no. It indicates whether or not output from the parallel tasks is labeled by task id. This variable is the most useful when the MP_STDOUT setting is unordered.

## 8.6.2 Environment Variables for Monitoring (2)



RS/6000 **Environment Variables for Monitoring (2)**

MP_RETRY n and MP_RETRYCOUNT m: If the nodes are not available, wait n seconds and request again m times.

MP_PULSE: Ensures that remote nodes are communicating with the home node.

MP_INFOLEVEL: Sets level of messages from POE.

MP_PMDLOG: Diagnostic messages are logged in a file in /tmp for each remote node.

MP_EUIDEVELOP: Message passing interface performs more detailed checking during execution.

POWERparallel Systems ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The second part is related to the execution and debugging environment variables.

- Execution

  Due to system timeout, program execution may hang or fail even if the nodes are available. To avoid this, you can set the following variables:

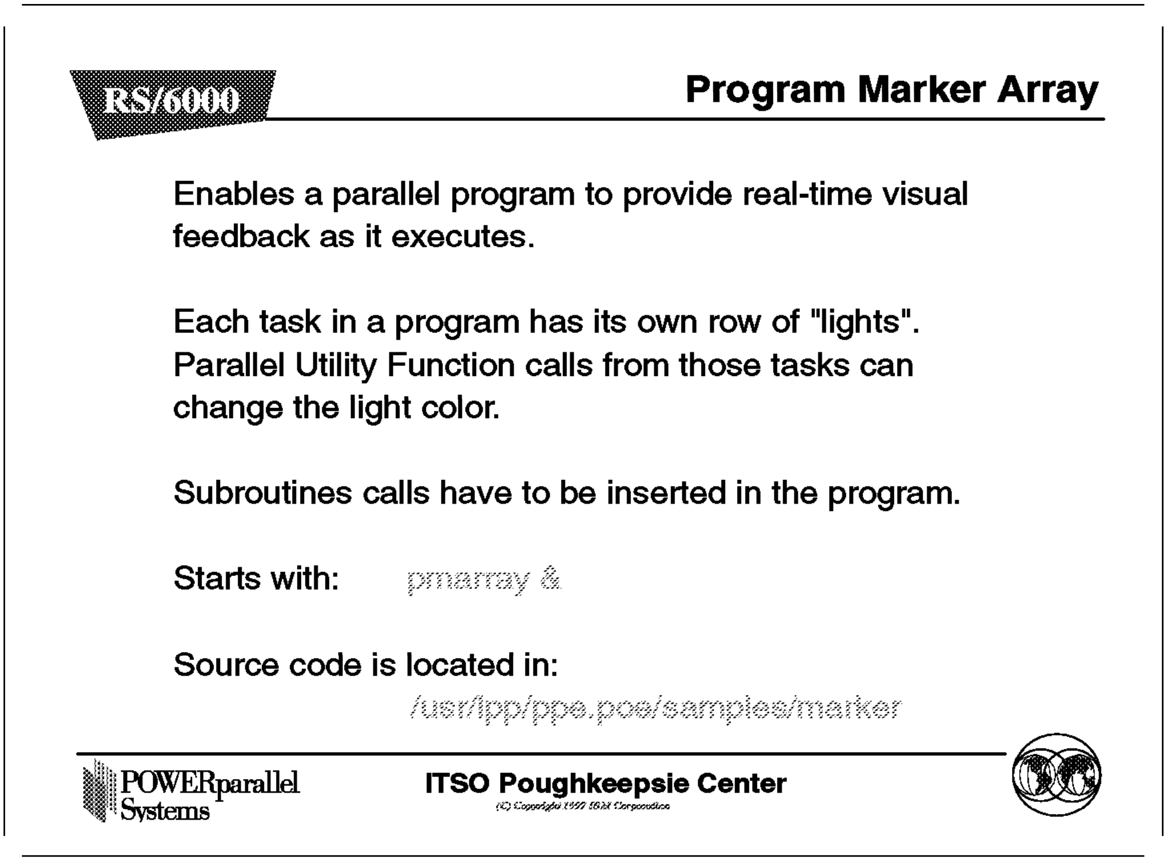
  - MP_PULSE is the interval (in seconds) at which POE checks the remote nodes to ensure that they are communicating with the home node. The default is 600 seconds.
  - MP_RETRYCOUNT is the number of times that the Partition Manager attempts to allocate processor nodes.
  - MP_RETRY is the interval (in seconds) between processor node allocation retries if there are not enough processor nodes available to run a program. MP_RETRY is only valid when the Resource Manager is used for nodes allocation.
  - On the reverse, MP_TIMEOUT is the length of time that POE waits before abandoning an attempt to connect to the remote nodes. The default is 150 seconds.

- Debugging

  When trouble occurs, this set of variables can be used for debugging.

- MP_INFOLEVEL gives the level of message reporting. The default is 1 for warning and error.
- MP_EUIDEVELOP indicates whether or not the message passing interface performs more detailed checking during execution. This additional checking is intended for developing applications, and can significantly slow performances.
- MP_PMDLOG allows you to log diagnostic messages to a file in /tmp. on each of the remote nodes. Typically this variable is set on the request of the *IBM Support Center* to resolve a PE-related problem.

### 8.6.3 Program Marker Array



RS/6000

Program Marker Array

Enables a parallel program to provide real-time visual feedback as it executes.

Each task in a program has its own row of "lights". Parallel Utility Function calls from those tasks can change the light color.

Subroutines calls have to be inserted in the program.

Starts with: pmarray &

Source code is located in:
/usr/lpp/ppe.poe/samples/marker

POWERparallel Systems

ITSO Poughkeepsie Center

The Program Marker Array is one of the tools delivered in Parallel Environment to monitor programs and is a part of the POE fileset. This is a programmable array of small boxes or *lights*, which are associated with parallel tasks. Under program control these lights can change color to provide you with immediate visual feedback as your program executes.

Each task in a parallel program has its own row of lights. Using calls to the Parallel Utility Functions enable a parallel program to control the appearance of the Program Marker Array Window. Calls to MP_MARKER (for Fortran programs) or mpc_marker (for C programs) enables a program to color lights on and/or send output strings to the Window. Calls to MP_NLIGHTS (for Fortran programs) and mpc_nlights (for C programs) enable a program to determine the number of lights displayed per task row.

These calls are included in the *libmpi.a* library.

If the Visualization Trace (VT) tracing is on, the marker information is also added to the trace file as an Application Marker event.

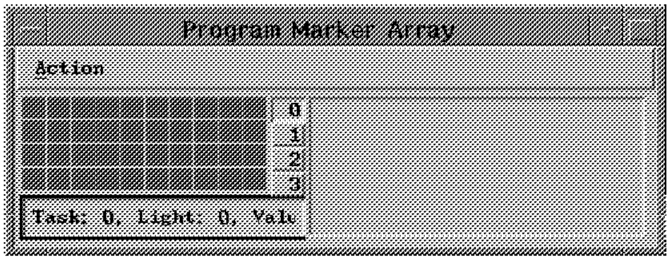
The Program Marker Array display executable runs as a separate task on the initiating processor and is connected via a socket to the Partition Manager. The Program Marker Array program (pmarray) displays the marker messages received by the Partition Manager.

The source for the Program Marker Array program is located in /usr/lpp/poe/samples/marker. The two files are named: PMArray.c and POE_Light.c. A sample program, hello_parallel.c, is also delivered.

### 8.6.4 Program Marker Array Display

RS/6000

**Program Marker Array Display**



Task: 0, Light: 0, Valu

export MP_PROCS=4
export MP_LIGHTS=10
pmarray &

POWERparallel Systems

**ITSO Poughkeepsie Center**

This picture has been obtained with the following settings:

- `export MP_PMLIGHTS=10` (default is 0)
- `export MP_PROCS=4` (default is 1)
- `pmarray &`

MP_USRPORT is the port number to be used for connecting with the Partition Manager. It is only necessary to specify MP_USRPORT if there is a port number conflict. If MP_USRPORT is specified for pmarray, the same port number must be specified for the Partition Manager, the default port number is 9999.

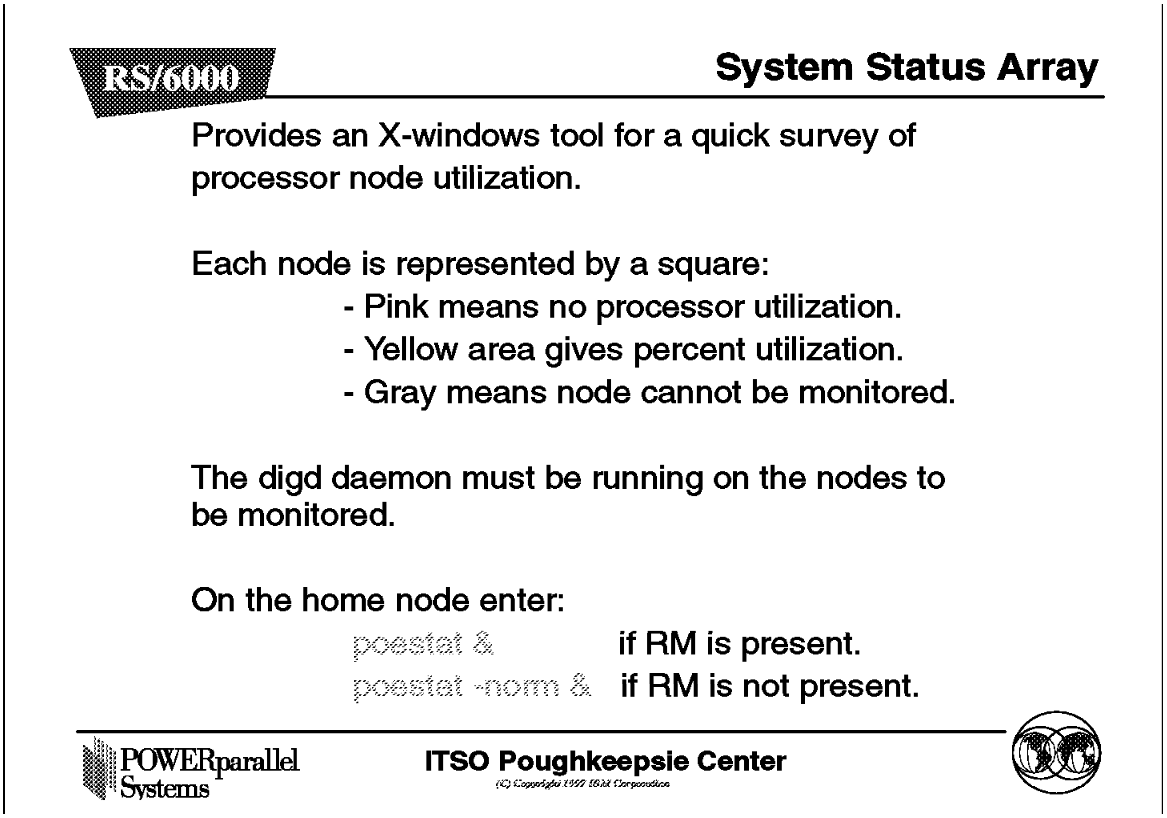
At the right end of the row is a push button labelled with the task number, numbered from 0 top-to-bottom. Pushing one of the task buttons causes the current message text for that task to be displayed in the panel immediately to the right of the push button column. The button turns yellow if there is a pending message for that task.

You can click with the left mouse button on a light to obtain details in the bottom text area:

- The task identifier
- The light number
- The color number value

This information is not updated until you select another light.

### 8.6.5 System Status Array



System Status Array

Provides an X-windows tool for a quick survey of processor node utilization.

Each node is represented by a square:

- Pink means no processor utilization.
- Yellow area gives percent utilization.
- Gray means node cannot be monitored.

The digd daemon must be running on the nodes to be monitored.

On the home node enter:

```
poestat &          if RM is present.
poestat -norm &    if RM is not present.
```

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

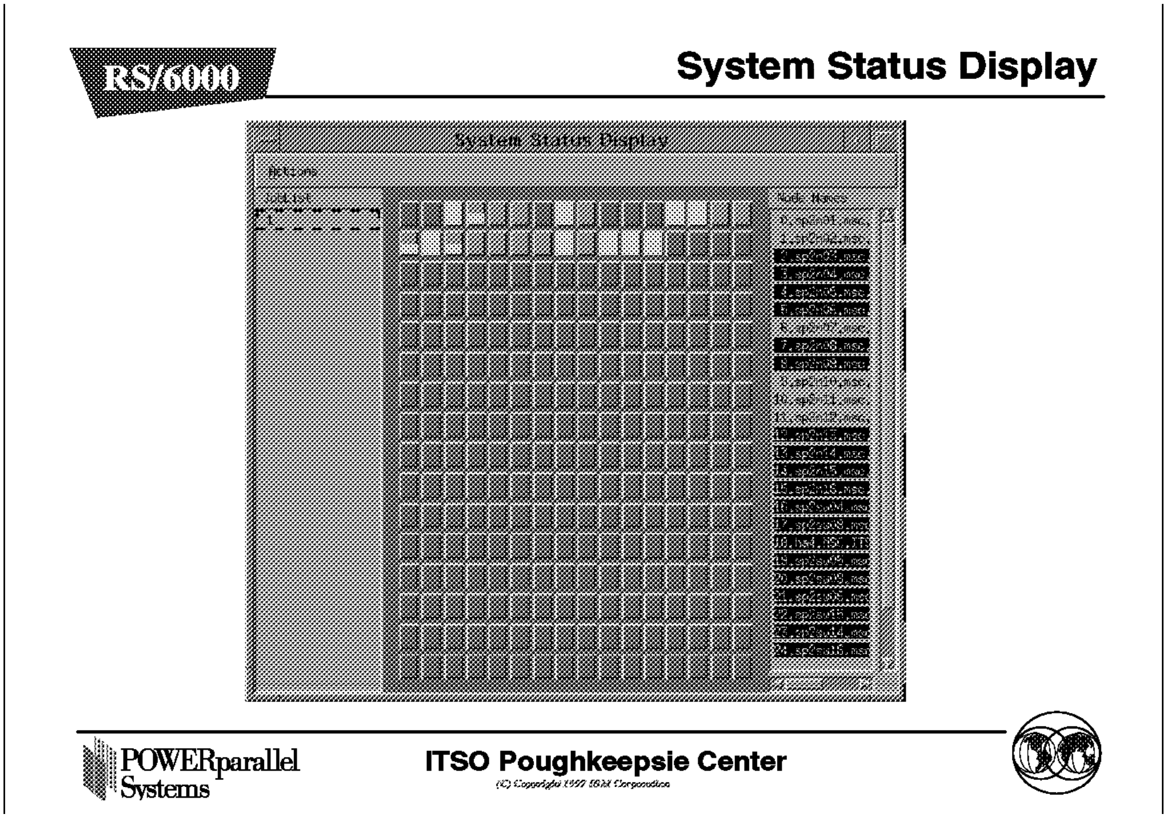
This X-Windows monitoring tool lets you quickly survey the utilization of processor nodes. The array consists of a number of squares, each representing a processor node of your RS/6000 SP system or cluster. The squares are colored pink and yellow to show the instantaneous percent of CPU utilization for each processor node. If a square were to appear all pink, the node would be at 0 percent utilization. If a square were to appear all yellow, it would be at 100 percent utilization. To the right of the array is a node list which contains the name of each node shown in the array. The nodes are listed in the order in which they were contacted, left to right, starting with the top row of the array. Use this list to identify the name of a node represented in the array.

In order to use this tool, the Visualization Tool′s (VT) Statistics Collector Daemon process (digd) needs to be running on each of the nodes you wish to monitor. The daemon feeds the System Status Array with the CPU information it displays. The digd statistics collector daemon can also feed information to the Visualization Tool. If a square on the array appears gray, the node is unavailable for monitoring. It either does not have the Statistics Collector Daemon running, or the array cannot communicate with it.

The digd daemon is installed as a part of the POE fileset (ppe.poe), and is started through the /etc/inittab. The VT fileset do not need to be installed to use the System Status Array.

The poestat command starts the System Status Array and pools for the digd daemon running on the network. The pooling is dynamic.

### 8.6.6 System Status Display



RS/6000

System Status Display

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This window consists of:

- A job list
  This list provides all the jobs currently running on the RS/6000 SP system using data provided by the Resource Manager. If you started the display with the `-norm` option, the System Status Array cannot track jobs and so cannot list them in this area.
- A node matrix
  Each square on the matrix represents one of the processor nodes. The nodes are listed in order, left to right, starting with the top row of the array.

  You can select the nodes and turn monitoring on from:

  – The node matrix
  – The node list
  – The job list

  If the Resource Manager is used, the nodes are displayed in the pool order returned by the `jm_status` command. In this case, you start the System Status Array with the following statements:

  – `export MP_RESD=yes`
  – `poestat &`

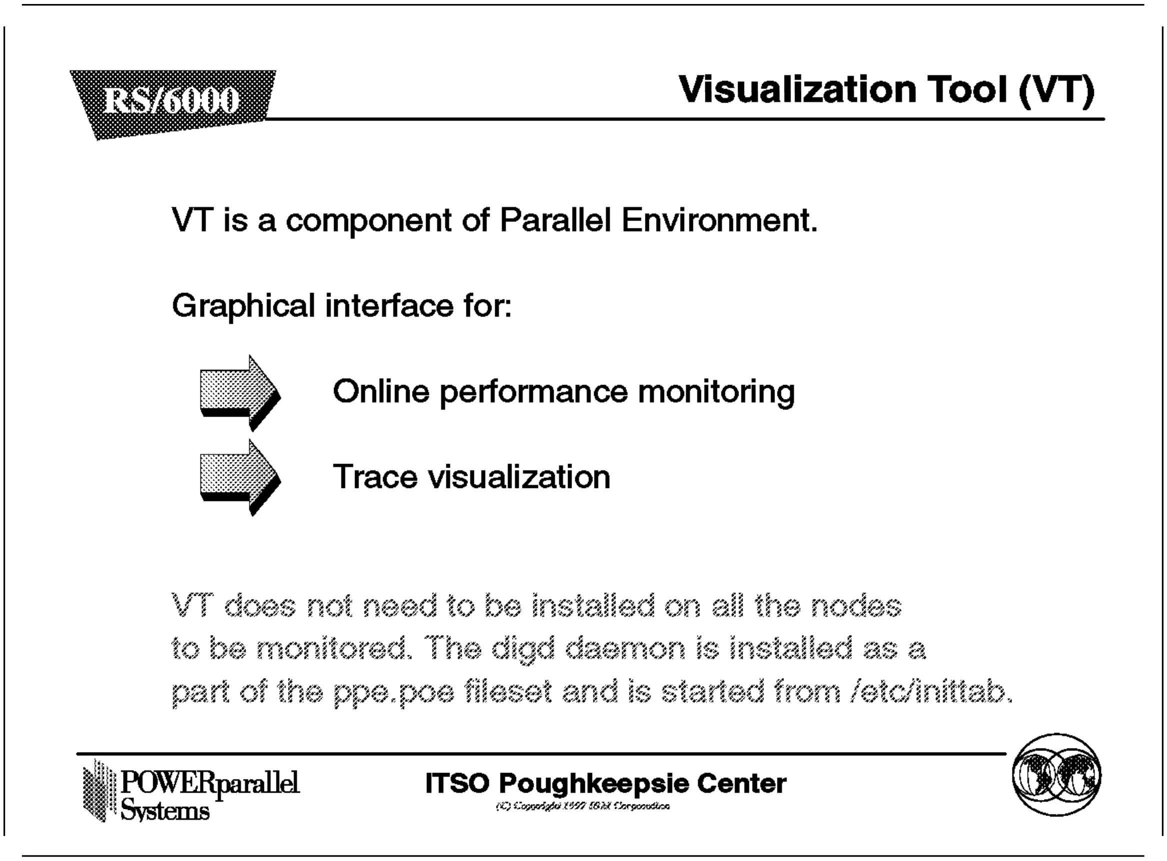
  If the Resource Manager is not running on your system, you must enter the following statements:

- export MP_HOSTFILE=host.list
- poestat -norm &

If MP_HOSTFILE is not set, poestat relies on the default host.list file in the current directory.

In the preceding picture, the host.list file contained the nodes sp2n01 to sp2n16. The nodes whose the square is gray could not be selected. Either the digd daemons were not running, or the nodes were unreachable. These nodes are not selected in the node name area on the right of the window.

### 8.6.7 Visualization Tool (VT)



RS/6000

**Visualization Tool (VT)**

VT is a component of Parallel Environment.

Graphical interface for:

- Online performance monitoring
- Trace visualization

VT does not need to be installed on all the nodes
to be monitored. The digd daemon is installed as a
part of the ppe.poe fileset and is started from /etc/inittab.

POWERparallel Systems

ITSO Poughkeepsie Center

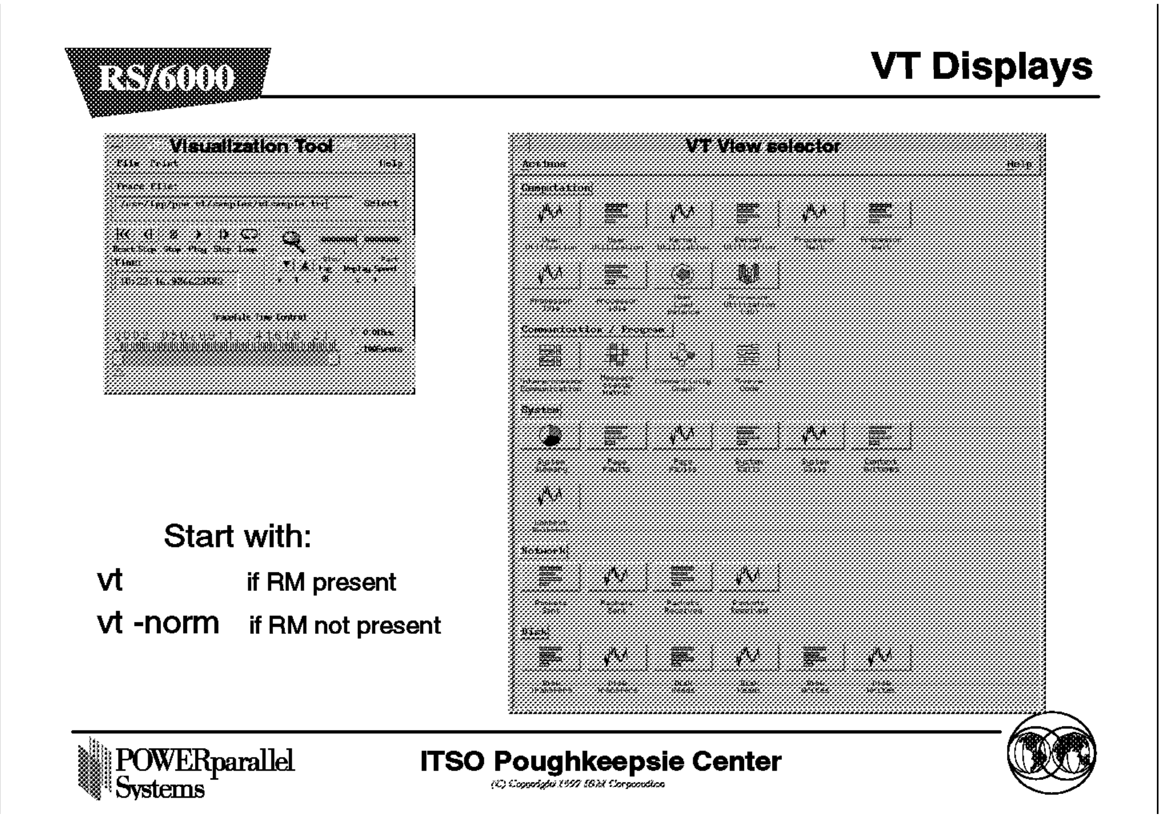
The IBM Parallel Environment, Version 2 Release 3, Visualization Tool (VT) is designed to show graphically the performance and characteristics of a parallel application program using the IBM Message Passing Interface (Program Visualization), and also to act as an online monitor (Performance Monitoring).

- The displays used for Program Visualization show program and system information collected during the application′s execution.
- The displays used for Performance Monitoring show online system activity at a configurable sampling frequency.

VT can be used to play back traces generated during a program′s execution (trace visualization).

VT is based on Motif and X Window System standards. It is a PE fileset (ppe.vt). The VT Tracing System is packaged and installed with the Parallel Operating Environment (POE) fileset (ppe.poe). The same daemon, *digd*, collects AIX information on cluster workstations and RS/6000 SP nodes, with or without the High Performance Switch Option.

## 8.6.8 VT Displays



RS/6000

VT Displays

Start with:

vt if RM present

vt -norm if RM not present

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

VT can be started with:

- The command vt if the RM is present
- The command vt -norm if the RM is not present.

The command raises two windows: the *Visualization tool* which is the master window, and the *VT View selector* window. You can save and load your preferences (colors, time resolution, sampling interval, and so on) with a configuration file that is called from either the -configfile flag or the graphical tool.

Whether you are using VT for trace visualization or performance monitoring, the Selector View offers a collection of displays called *views* These views allow to display activity about:

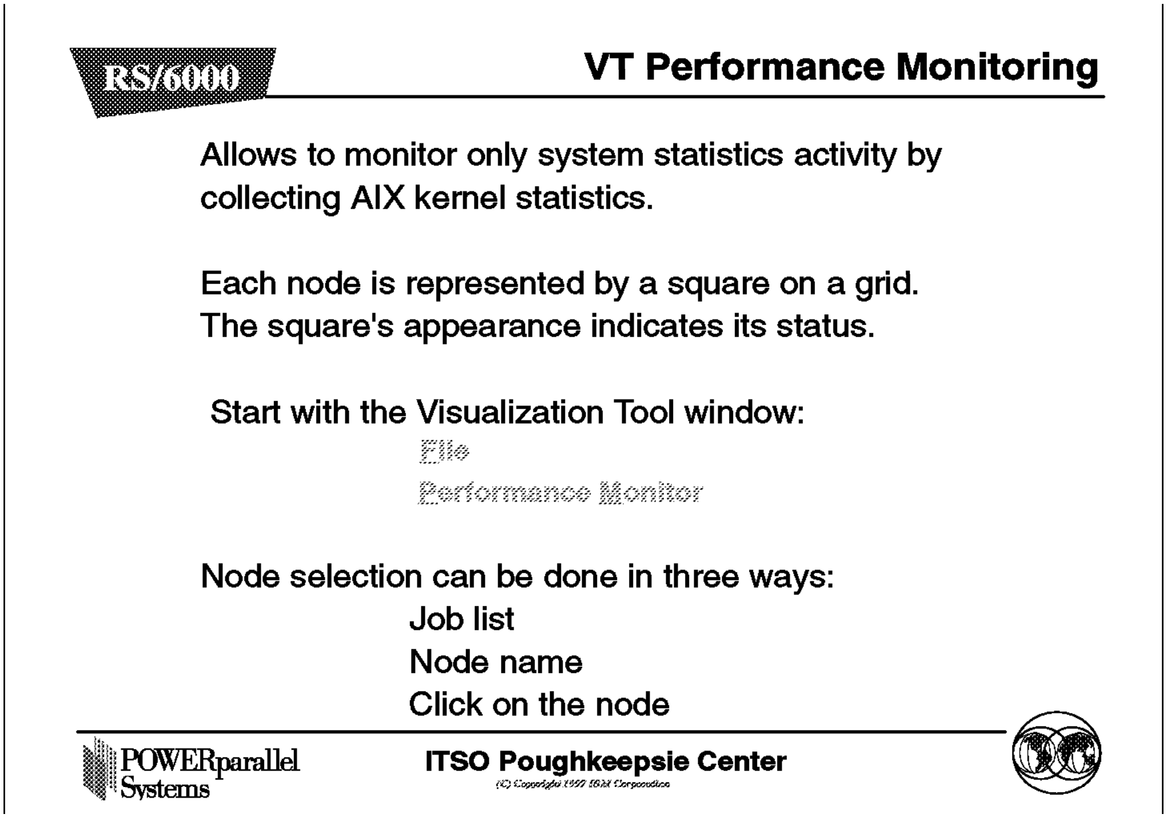
- CPU
- Communication/Program
- System summary
- Network
- Disk

The Communication/Program views are only available in trace mode. Any other view may be used for online performance monitoring.

Using the mouse, you can display details of the displayed information or change the appearance or configuration.

Often, different views take the same information and present it in different ways, such as in a bar chart or strip graph.

### 8.6.9 VT Performance Monitor



RS/6000

VT Performance Monitoring

Allows to monitor only system statistics activity by collecting AIX kernel statistics.

Each node is represented by a square on a grid.
The square's appearance indicates its status.

Start with the Visualization Tool window:
File
Performance Monitor

Node selection can be done in three ways:
Job list
Node name
Click on the node

POWERparallel Systems

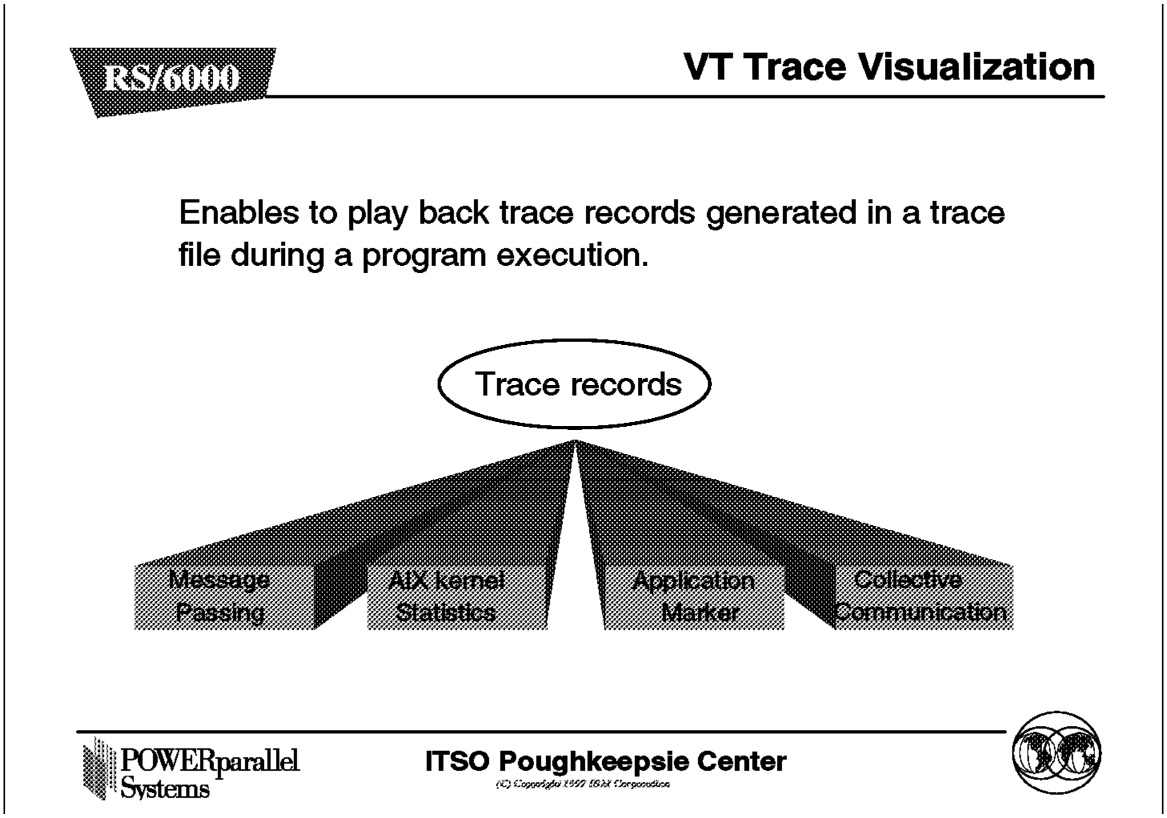
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The Performance Monitoring is an online monitor used to study the operational status and activity of processor nodes.

The window and functionality are similar to those described for the System Status Array in the sections 8.6.5, “System Status Array” on page 466 and 8.6.6, “System Status Display” on page 468.

## 8.6.10 VT Trace Visualization



RS/6000

VT Trace Visualization

Enables to play back trace records generated in a trace file during a program execution.

Trace records

Message Passing

AIX kernel Statistics

Application Marker

Collective Communication

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The trace visualization enables you to play back trace records generated in a trace file during a program′s execution. Every view is available in this mode.

There are four types of trace records:

- Message passing

  Message passing event trace records contain information regarding point-to-point message passing events, such as blocking sends and receives among tasks of your program. Each of these events is the result of a call to a message passing subroutine.

- Collective communication

  Collective communication trace records contain information about communication events involving groups of tasks. Broadcasts and gathers are examples of collective communication trace records. Each of these events is a result of a call to a collective communication subroutine.

- AIX kernel Statistics

  AIX kernel statistics trace records contain a sampling of statistics from the kernel. These include the:

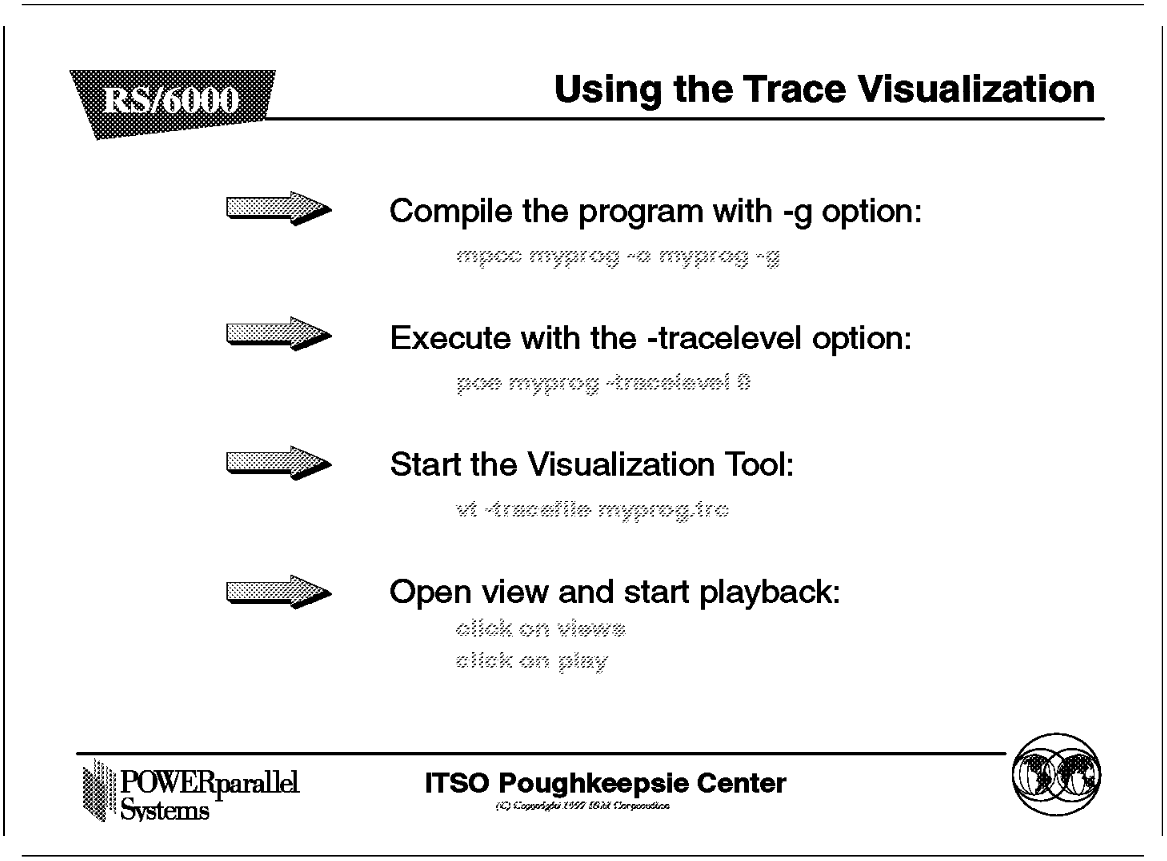
  - CPU utilization (user, kernel, wait, and idle)
  - System calls and pages faults
  - Disk utilization (transfers, reads, and writes)

  - TCP/IP packets received and sent

- Application Marker

  The Program Marker Array information (8.6.2, “Environment Variables for Monitoring (2)” on page 461) is also registered in the trace file and it can be displayed in the *source code* view.

## 8.6.11 Using the Trace Visualization



RS/6000

Using the Trace Visualization

Compile the program with -g option:

mpcc myprog -o myprog -g

Execute with the -tracelevel option:

poe myprog -tracelevel 9

Start the Visualization Tool:

vt -tracefile myprog.trc

Open view and start playback:

click on views
click on play

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

Before the trace file can be used to replay a program execution, you must create it.

First, you can compile your program with the -g flag, which produces an object file with symbol table references needed to take advantages of the source code view. This view lets you see the actual lines of code associated with the trace record events you are visualizing.

**Note:** The -g flag is not required if you do not wish to use the source code view.

Second, to generate a trace file, you need to execute your program with tracing turned on. The MP_TRACELEVEL environment variable or the -tracelevel flag activates the tracing with the values 1, 2, 3, or 9. The default value 0 means tracing is off. By default, trace file are named the same as the program name with the suffix .trc added.

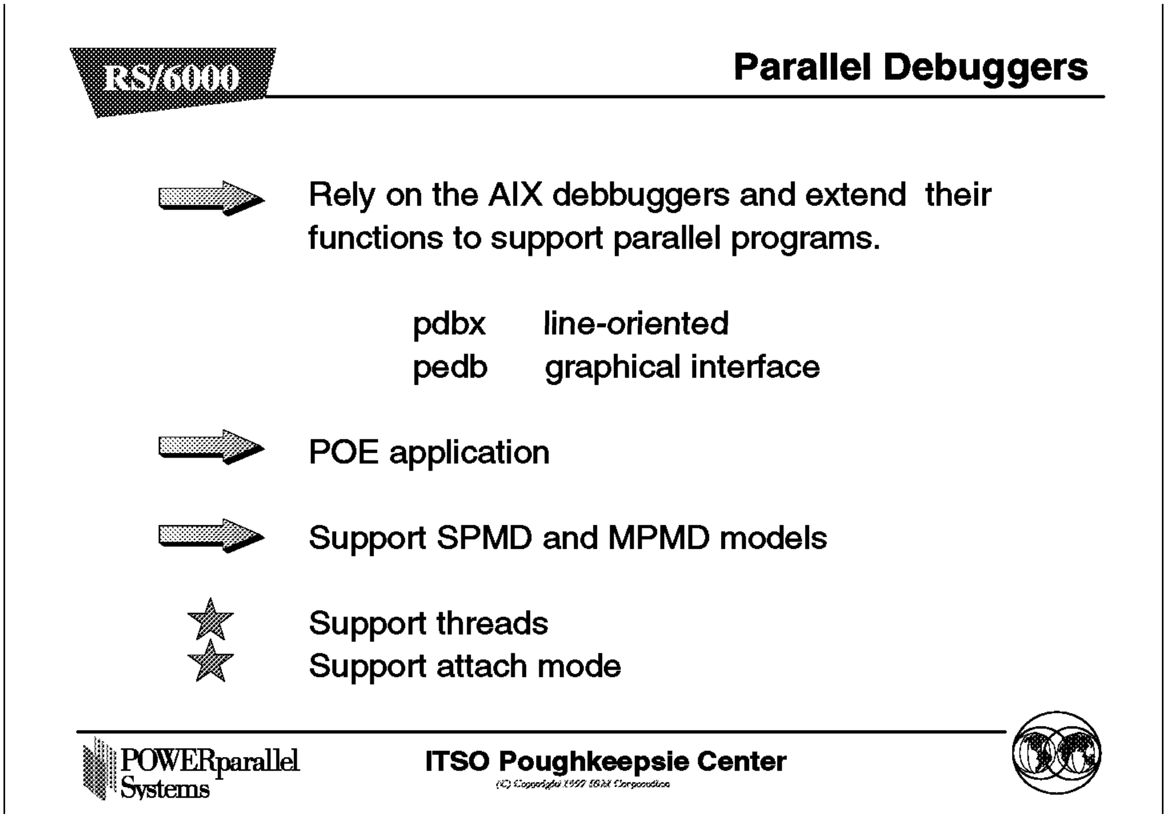
The MP_TRACEFILE environment variable or -tracefile flag allows you to start VT directly with your trace file. If not set, VT starts with a default trace file /usr/lpp/ppe.vt/samples/vtsample.trc. You must select your trace with the **File** icon of the the Visualization Tool window. Then you must click the **views** from the View Selector and **play** from the Visualization Tool.

By clicking on one of the views, Computation, Communication/Program, System, Network and disk, you can raise all the windows contained in this views.

VT uses its own routines to create trace records and does not utilize the AIX trace facility. Some of these routines can be called from your application program, allowing you to generate trace files containing just the type of trace record you are interested in. In the same way, you can avoid generating huge trace files by placing VT_trc_stop() and VT_trc_start() calls in your program. Thus, you can start tracing only for the most interesting parts of the program.

If you have turned off trace record generation for five minutes during normal playback, you will have to wait the five minutes before any of the views are updated. You can get around this by advancing playback to the next trace record by clicking the **Step** button in the Visualization Tool window.

## 8.6.12 Parallel Debuggers



RS/6000

**Parallel Debuggers**

- Rely on the AIX debbuggers and extend their functions to support parallel programs.

  pdbx line-oriented
  pedb graphical interface

- POE application
- Support SPMD and MPMD models
- Support threads
- Support attach mode

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

Two versions of debugger are delivered in Parallel Environment. They rely on their corresponding AIX versions, and support most of their subcommands. Some of them have been modified to support parallel commands.

- The pdbx extends the dbx debugger′s line-oriented interface and subcommands.
- The pedb provides a simplified Motif graphical interface. The pedb debugger was formerly called xpdbx in the previous Parallel Environment version.

An example of an additional subcommand is the group subcommand for pdbx. This allows you to collect a number of tasks under a group name you choose, and then manipulate them as a simple group. The actions available for the group subcommand are: add, change, delete, and list.

When invoking one or the other debugger, it is first necessary to compile the program and set up the execution environment:

- Compiling the program.

  Since they are source code debuggers, you must compile the program with the -g option to produce an object file with debug symbol, source line number, and data type information. As for their AIX versions, it is advisable to not use the -O optimization option. Using the debugger on optimized code may produce inconsistent and erroneous results.

- Copying the source files.

These debuggers are POE applications with some modifications on the *home node* to provide a user interface. Like POE, they require that the applications programs are available to run on each node of the partition. To support source level debugging, the debuggers require the source code to be available on the nodes. To make the source file accessible, you will use the same mechanism as you used for the application program.

- Setting up the execution environment.

  You need to set up some execution environment variables like those described in the previous section and set up a few more in the particular debugging environment.

NORMAL mode and ATTACH mode

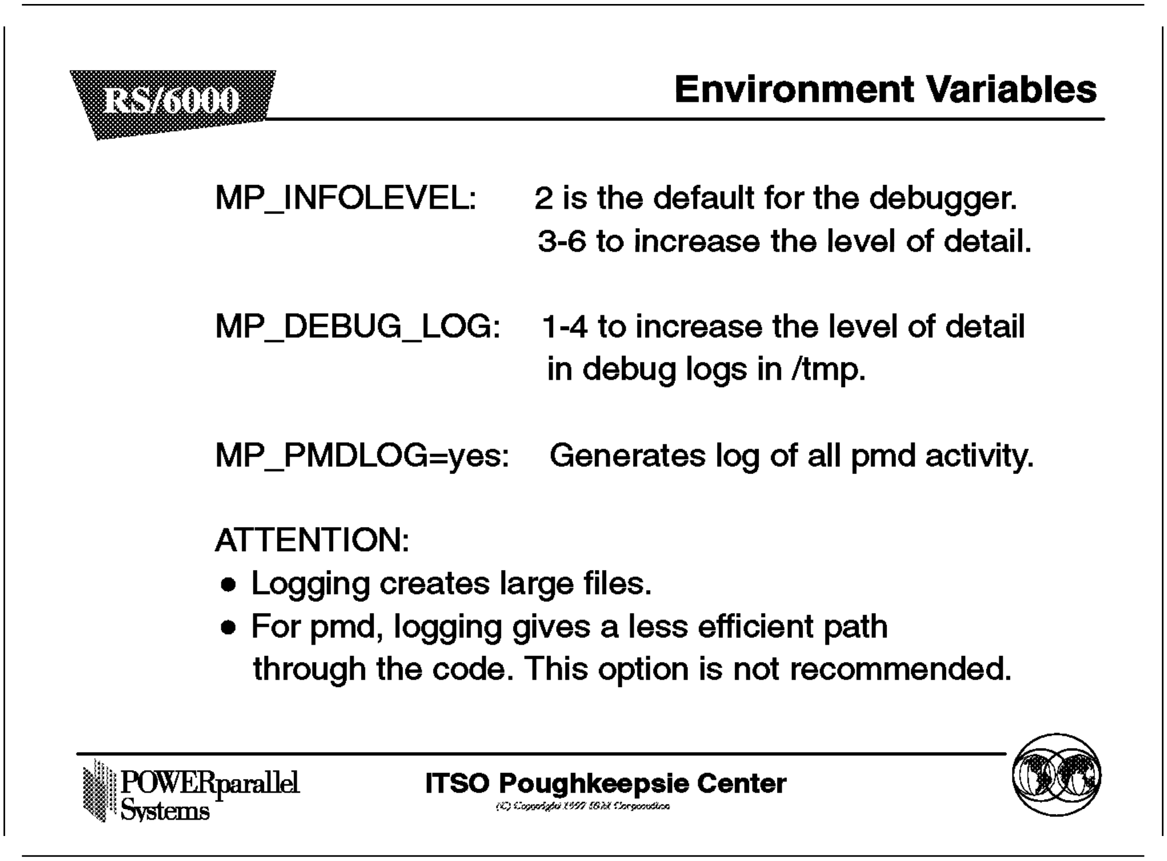
- With the *normal* mode, you enter the name of the program to debug on the debugger command line or with the graphic interface.
- With the *attach* mode, the debuggers allow you to attach the debugger to a parallel application that is currently running. This feature is useful to debug large, long running, or apparently hung applications. The debugger attaches to any subset of a task without restarting the executing parallel program. In this mode, you must specify the appropriate process identifier (PID) of the POE job, so the debugger can attach the correct application processes contained in that job.

While the attach mode was already supported with the previous version of the pedb, it is newly supported for the pdbx debugger.

Threads support

The debuggers bring a full support of threads, with the display of source code, variables, and stack, and the setting of breakpoint or tracepoint. A thread window has been added to the pedb debugger′s window.

## 8.6.13 Environment Variables



RS/6000

**Environment Variables**

MP_INFOLEVEL: 2 is the default for the debugger.
3-6 to increase the level of detail.

MP_DEBUG_LOG: 1-4 to increase the level of detail
in debug logs in /tmp.

MP_PMDLOG=yes: Generates log of all pmd activity.

ATTENTION:

- Logging creates large files.
- For pmd, logging gives a less efficient path through the code. This option is not recommended.

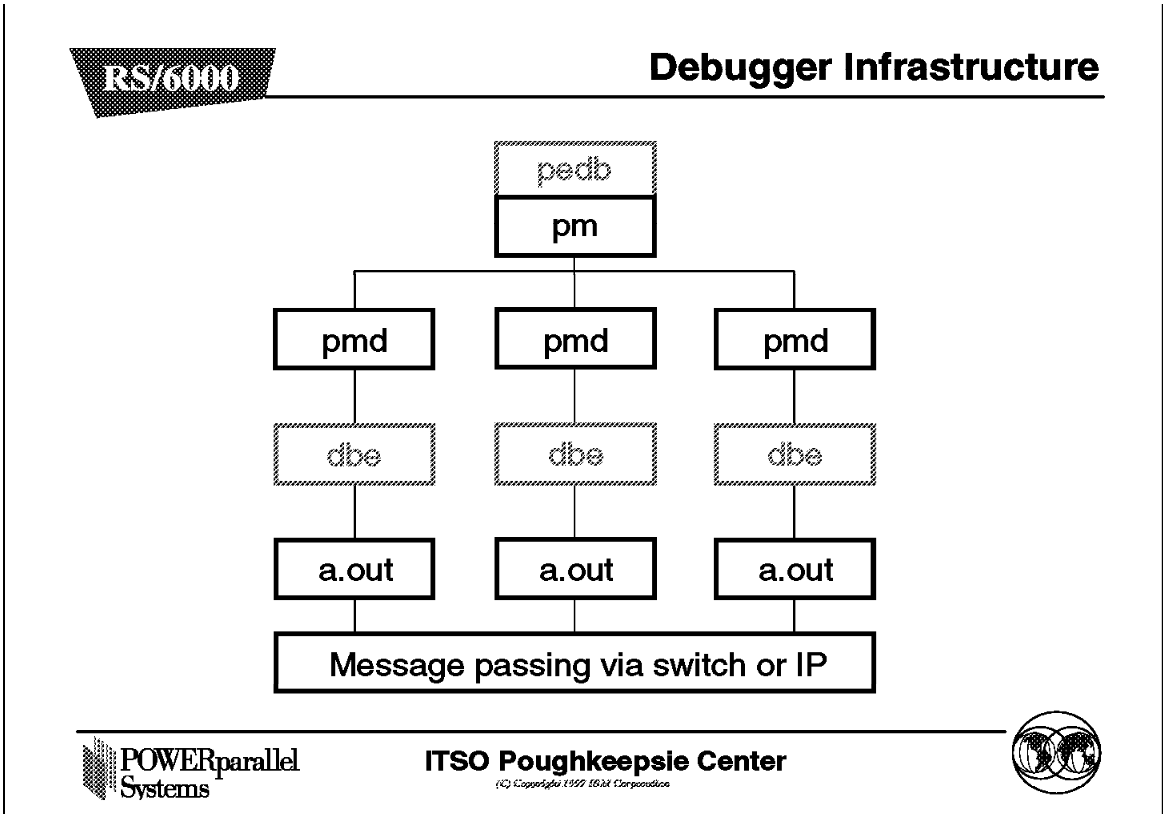
POWERparallel Systems

ITSO Poughkeepsie Center

While the debuggers are POE applications, you must set the necessary environment variables for the program execution. However, depending on the mode (normal or attach) you are working, some of them are invalid. As an example, if you have MP_PROCS set when the debuggers start in attach mode, they ignore the setting. A complete list of valid and invalid variables for both modes is given in one Appendix of the *Operating and Use, Volume 2*.

The preceding figure shows the environment variables that influence the debuggers.

## 8.6.14 Debugger Infrastructure



RS/6000

Debugger Infrastructure

pedb

pm

pmd

pmd

pmd

dbe

dbe

dbe

a.out

a.out

a.out

Message passing via switch or IP

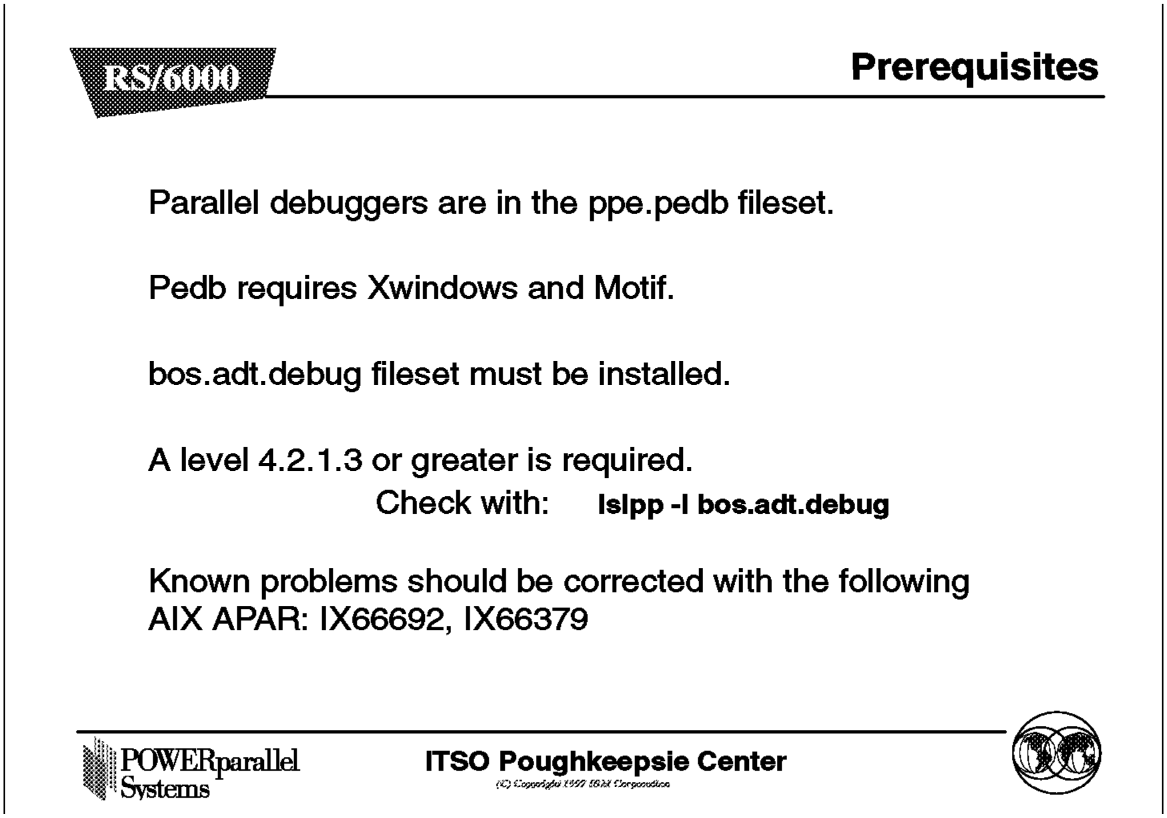
POWERparallel Systems

ITSO Poughkeepsie Center

The preceding figure shows the infrastructure in normal mode for the pdb. The infrastructure is different in attach mode. The infrastructure is the same for both debuggers:

- On the home node, they address the Partition Manager.
- On the remote node, the Partition Manager daemon relies on the AIX debugger to interpret the a.out file.

### 8.6.15 Prerequisites



RS/6000

**Prerequisites**

Parallel debuggers are in the ppe.pedb fileset.

Pedb requires Xwindows and Motif.

bos.adt.debug fileset must be installed.

A level 4.2.1.3 or greater is required.
Check with: **lslpp -l bos.adt.debug**

Known problems should be corrected with the following AIX APAR: IX66692, IX66379

POWERparallel Systems

**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

This a copy of the file /usr/lpp/ppe.pedb/README/pedb.README. It mentions the list of known problems and their related *APAR*.

The following are the function restrictions for Release 2.3 of pedb:

- Pedb tracepoints and single stepping on SMP processors

  When using tracepoints or program stepping using the *step over* and *step into* buttons, tasks running on SMP nodes may never return to the debug ready state. The *halt* button has no effect in these cases, and further debugging of the program on these nodes is impossible. The fixes for these problems are expected to be in a future PTF for the bos.adt.debug component.

- Illegal instruction executing task with held thread

  When holding the interrupted thread and then clicking the **step into** button, the task sometimes incorrectly reports an illegal instruction. The message displayed in the message window will be: `0030-3015 Task: n encountered signal: 4 - Illegal instruction`. Further debugging of this task will be impossible. Apply the fix for AIX APAR IX66692 when available.

- Pedb threads viewer window scroll bar

After manipulating the threads information displayed by using the select display details window numerous times, the data in the threads viewer window sometimes disappears, because the scroll bar becomes inactive. To refresh the threads information, select the **find** option in the threads viewer window and enter **t** in the **text to find field** and then select the first button. This will make the threads viewer scroll bar active and you can scroll the thread data into view. This will be fixed in the GA release of *pedb*.

- Setting breakpoints at first routine in a thread

  After setting breakpoints at the routine passed into pthread_create and then allowing the program to continue multiple times, the debugger eventually hangs. Apply the fix for AIX APAR IX66379 when available.

- Trace [if Condition] with out-of-scope variables

  The debuggers may hang if conditional tracepoints that reference out-of-scope variables are set. It is possible to create the tracepoint, but after it is encountered, the program stops and the debugger must be stopped. It is possible to work around this problem by fully qualifying the variable name specified in the condition. A similar problem with breakpoints is described later. The fix for this problem is expected to be in a future PTF for the bos.adt.debug component.

- Cannot continue after out of scope conditional breakpoint

  When a conditional breakpoint at a line number is specified involving a variable that is out of scope at the time the line number is encountered, the debugger stops at the line even though the condition has not been satisfied. Further execution through single stepping or continuing is not allowed, even if the breakpoint is removed. It is possible to work around this problem by fully qualifying the variable name specified in the condition. A similar problem for conditional tracepoints is described previously. The fix for this problem is expected to be in a future PTF for the bos.adt.debug component.

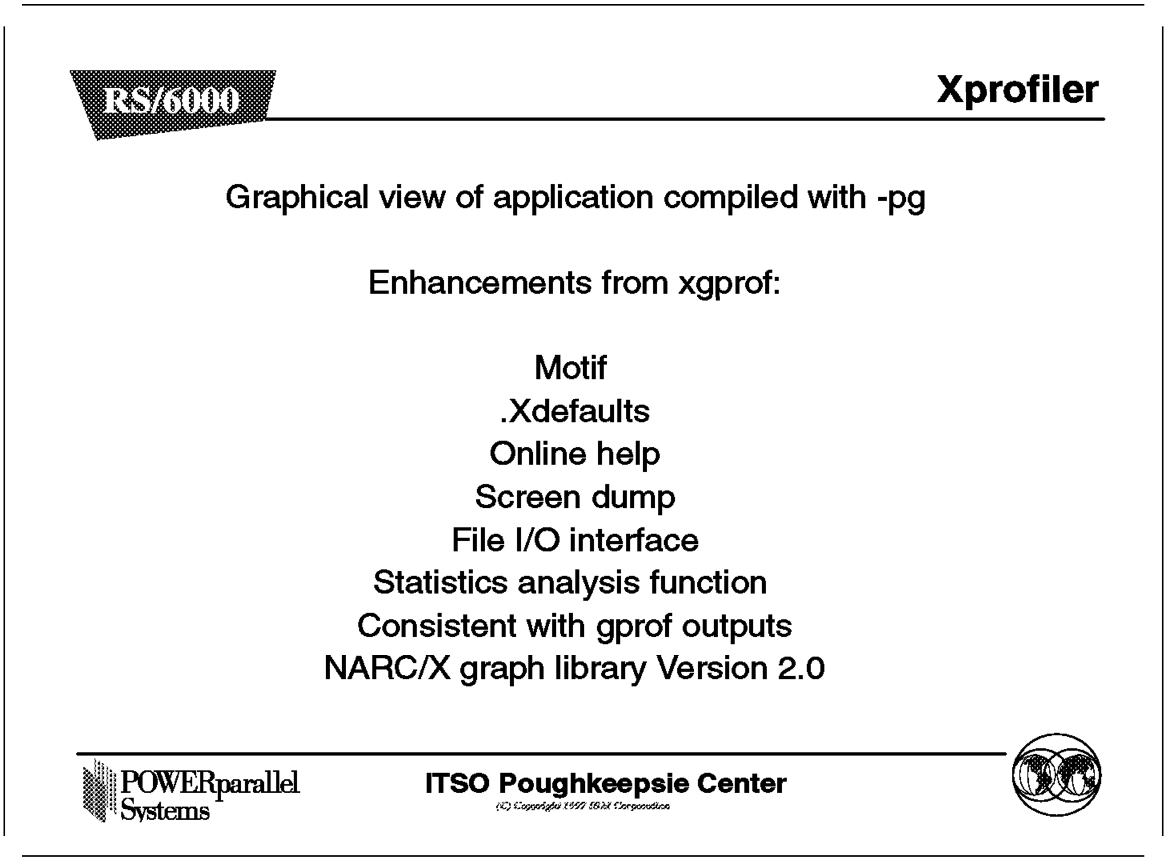
- Incorrectly seeing the `all threads held` message

  After holding the interrupted thread, then continuing execution and then releasing the thread, it is possible to receive the `all threads held` message, so that no further debugging is possible. Apply the fix for AIX APAR IX66692 when available.

- Setting breakpoints near pthread_join

  After setting breakpoints near pthread_join, and then allowing the program to continue multiple times, the debugger eventually hangs. Apply the fix for AIX APAR IX66379 when available.

## 8.6.16 Xprofiler



RS/6000

Xprofiler

Graphical view of application compiled with -pg

Enhancements from xgprof:

Motif<br>
.Xdefaults<br>
Online help<br>
Screen dump<br>
File I/O interface<br>
Statistics analysis function<br>
Consistent with gprof outputs<br>
NARC/X graph library Version 2.0

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

*Xprofiler* is a tool that helps you to analyze your serial or parallel application′s performance quickly and easily. It uses data collected with the -pg compiling option to build a graphical display of the functions within your application. Xprofiler provides quick access to the profiled data, which lets you identify the functions that are the most CPU-intensive and focus on the application′s critical area. However, you can still use the standard AIX profilers prof and gprof to analyze your application.

**Note:** Unlike gprof, Xprofiler lets you profile your program at a source statement level. In this case, the application must be compiled with the -g option.

During the parallel program execution, the outputs produced by the -pg option are written into multiple files, one for each task that is running in the application. To prevent each output file from overwriting the others, POE appends the task ID to each gmon.out file. The current directory must be shared by all remote nodes. Otherwise, the profile data files must be manually transferred to the home node for analysis. The Parallel Environment command mcpgath can also be used to copy the files to the home node, and add the task ID as a suffix to the name of each file.

In order to get a complete picture of your parallel application′s performance, you must indicate all of these gmon.out files when you load the application into Xprofiler. Thus, Xprofiler shows you the sum of the profile information contained in each file.

Xprofiler does not give you information about the specific threads in a multithreaded program. The data Xprofiler presents is a summary of the activities of all the threads.

Xprofiler brings many enhancements to the former version xgprof:

- Motif

  All the graphic interfaces in Xprofiler are reorganized to follow the Motif convention.

- .Xdefaults

  A set of X resources are defined for each graphic display. These resources are kept in the Xprofiler′s resource file Xprofiler.ad, which provides more flexibility for a user environment′s customization.

- Online help

  There are help buttons on all the main dialogs which bring up a help window. The help paragraphs are stored in the xprofiler.sdl file.

- Sreen dump

  Xprofiler provides screen dump functions that allow users to selectively capture the image of a display window and store the data in postscript format for later use, or send the data directly to a printer.

- File I/O interface

  After the Xprofiler′s initialization, the file I/O interface allow users to load a different set of executables and/or gmon.out files.

- Statistics analysis function.

  When more than one gmon.out file is given, the prof and gprof profilers provide a summary profile information of all the input file. There is no statistic analysis. The statistics function in Xprofiler calculate the maximum, minimum, standard deviation, and average performance profile values across all the input gmon.out files.

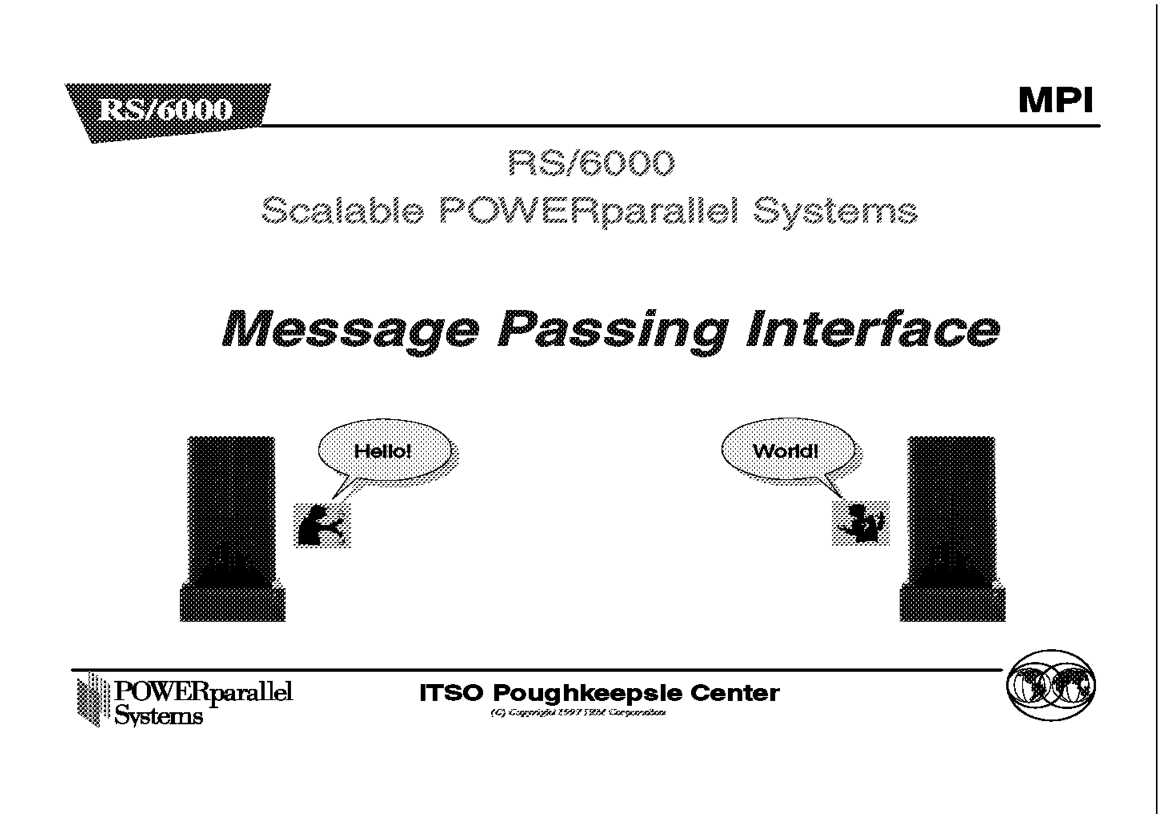
- Consistent with gprof outputs

  Although Xprofiler provides reports similar to the ones generated by gprof, it does not rely on gprof and uses its own routines to produce reports. The intent is to ensure Xprofiler will provide reports consistent with those delivered by gprof.

- NARC/X graph library

  Xprofiler relies on the NARC/X graph library to provide underlying graph capabilities. The current release on AIX platform is NARC 2.2.

# Chapter 9. Overview of MPI



RS/6000 MPI

RS/6000
Scalable POWERparallel Systems

*Message Passing Interface*

Hello!
World!

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This chapter provides an overview of the Message Passing Interface (MPI).

The IBM Parallel Environment for AIX, Version 2 Release 3, includes the following:

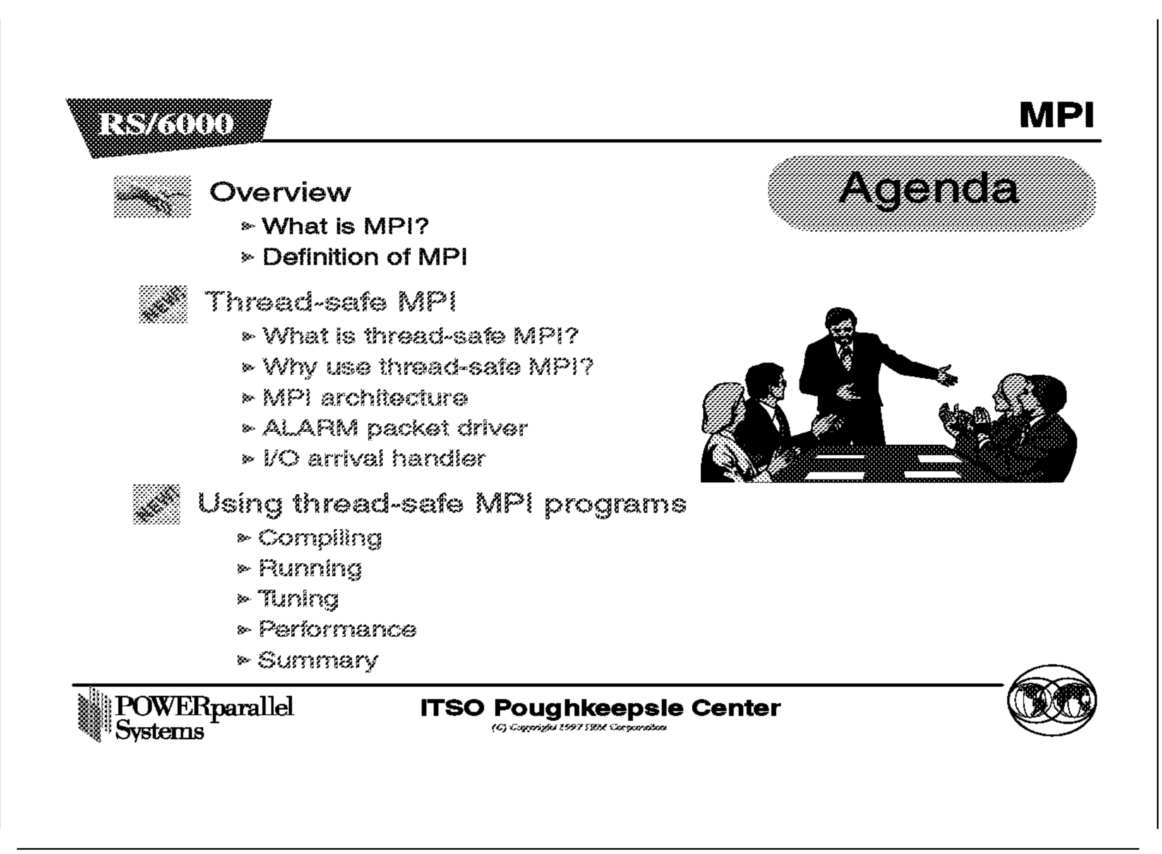
- Parallel Operating Environment (POE)
- Parallel Debugger (PDBX - PEDB)
- Xprofiler
- Visualization Tool (VT)
- Parallel File Utilities
- Message Passing Libraries (MPI, MPL)

IBM Parallel Environment for AIX Version 2.3 continues to provide support for MPL (Message Passing Library), the IBM message passing API. MPL and MPI subroutines can coexist in the same parallel program. However, note that MPL support is only provided in the non-threaded version of the MPI library.

The MPI library includes several IBM extensions (MPE) subroutines. These extensions, though not part of the MPI standard, provide an alternative set of powerful collective nonblocking functions. No callback facilities are defined in the current MPI Version 1.1 standard. The MPI library supplied with the IBM Parallel Environment for AIX, Version 2 Release 3 is compliant with the MPI

Version 1.1 standard. It is up to the developer to choose between the conformity of his code with the MPI and the use of the IBM extensions.

The following presentation is devoted to the changes in the MPI library in the IBM Parallel Environment for AIX, Version 2 Release 3.



RS/6000

MPI

Agenda

- Overview
  - What is MPI?
  - Definition of MPI
- Thread-safe MPI
  - What is thread-safe MPI?
  - Why use thread-safe MPI?
  - MPI architecture
  - ALARM packet driver
  - I/O arrival handler
- Using thread-safe MPI programs
  - Compiling
  - Running
  - Tuning
  - Performance
  - Summary

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

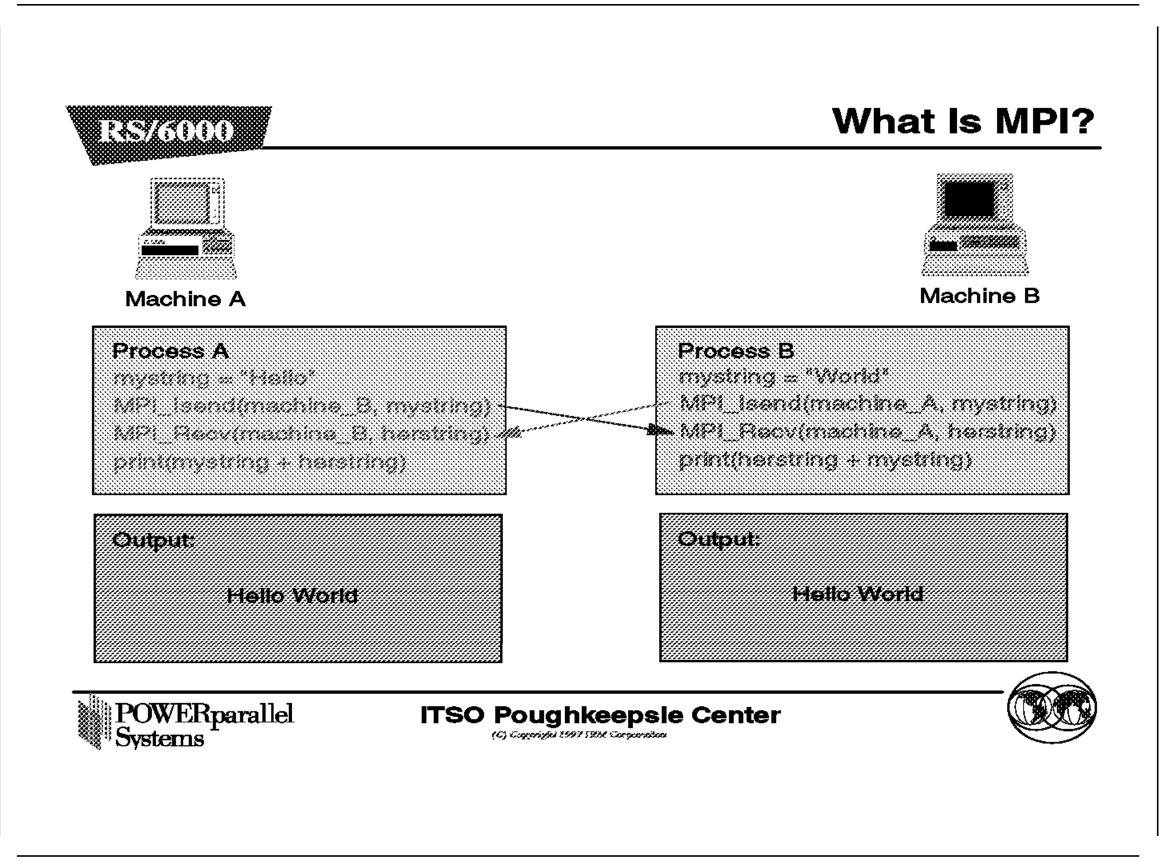
This chapter details how the Message Passing Interface is being implemented in IBM Parallel Environment for AIX, Version 2 Release 3.

We start with an overview, looking at what constitutes an MPI program and the definition of MPI.

That is followed by a discussion of thread-safe MPI, which takes a look at the MPI architecture as implemented in IBM Parallel Environment for AIX, Version 2 Release 3.

Finally, we take a look at how MPI programs are compiled and run and also some tuning parameters which we can specifiy. The presentation will be closed with a summary table.

### 9.1.1 What is MPI?



RS/6000

**What Is MPI?**

Machine A

Machine B

```
Process A
mystring = "Hello"
MPI_Isend(machine_B, mystring)
MPI_Recv(machine_B, herstring)
print(mystring + herstring)
```

```
Process B
mystring = "World"
MPI_Isend(machine_A, mystring)
MPI_Recv(machine_A, herstring)
print(herstring + mystring)
```

Output:

Hello World

Output:

Hello World

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This example illustrates the concept of message passing.

Suppose we have two MPI tasks running on two workstations, or two SP nodes. We have Process A running on Machine A, and similarly Process B running on Machine B.

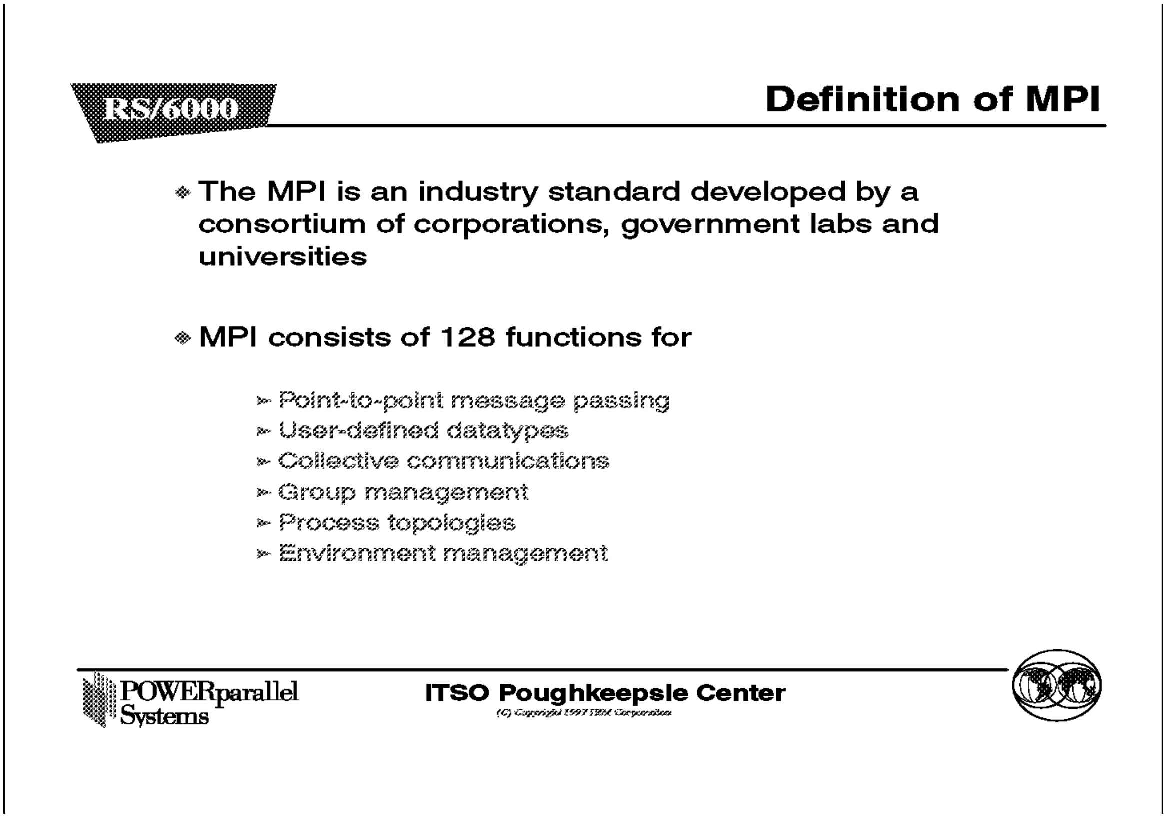
The essential parts of the two processes are shown. In each, a string will first be defined and then sent to the other process.

MPI_Isend here is a nonblocking MPI call, that is as soon as the message (in this case the string) is put onto the network, the process resumes execution without the confirmation of a receive from the target process.

After sending the message, both processes then issue an MPI_Recv, which posts a receive and waits for message arrival. This is a blocking call, which means that execution will be suspended until the MPI call returns, which will only happen when the MPI_Recv receives the correct message.

Finally, both processes print out a string that consists of data received from each other.

### 9.1.2 Definition of MPI



RS/6000

**Definition of MPI**

- The MPI is an industry standard developed by a consortium of corporations, government labs and universities
- MPI consists of 128 functions for
  - Point-to-point message passing
  - User-defined datatypes
  - Collective communications
  - Group management
  - Process topologies
  - Environment management

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

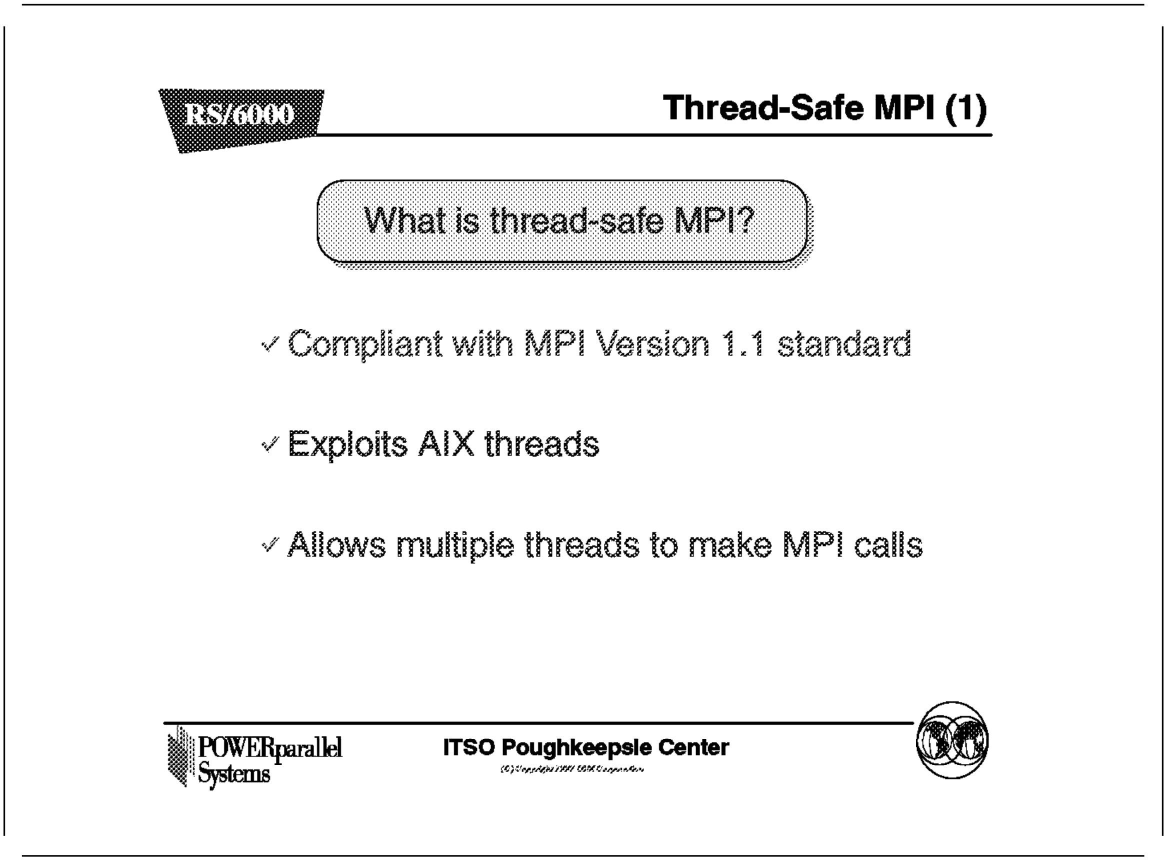
The MPI Standard, as it is copyrighted by the University of Tennessee, was strongly influenced by:

- Work at the IBM T.J. Watson Research Centre
- Intel NX/2
- Express
- nCUBE′s Vertex
- PARMACS
- Chimp
- PVM
- PICL

The MPI standard library provides functions for:

- Point-to-point message passing
- User-defined datatypes
- Collective communications
- Group management
- Process topologies
- Environment management

## 9.2 Thread-Safe MPI



RS/6000

**Thread-Safe MPI (1)**

What is thread-safe MPI?

✓ Compliant with MPI Version 1.1 standard

✓ Exploits AIX threads

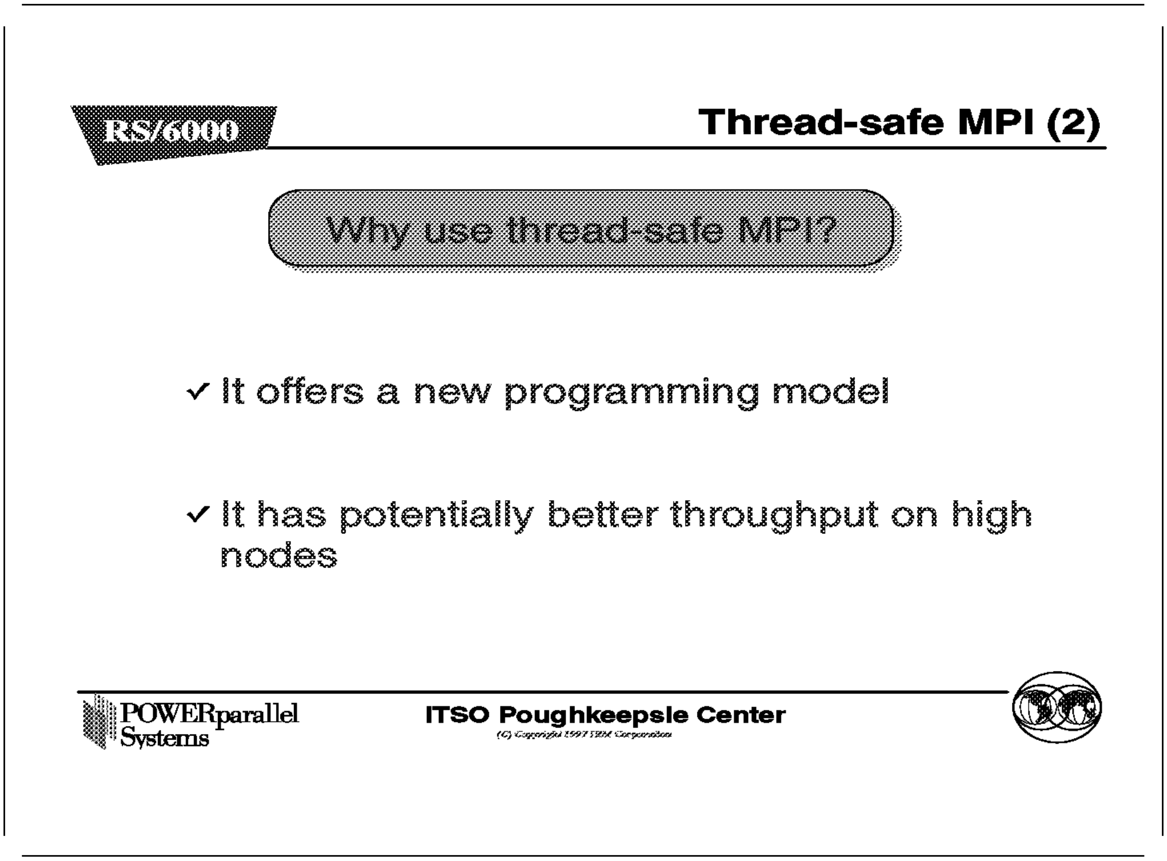
✓ Allows multiple threads to make MPI calls

POWERparallel Systems

**ITSO Poughkeepsle Center**

In order to take advantage of threads, especially on SMP machines, the MPI library is now thread-safe. The most important change to the MPI library is the addition of locks, for global data consistency.

Also, the MPI services are implemented with threads, instead of signals. This is discussed later.



RS/6000

## Thread-safe MPI (2)

**Why use thread-safe MPI?**

- ✓ It offers a new programming model
- ✓ It has potentially better throughput on high nodes

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

On SMP machines, besides the shared memory model of programming, developers now have the alternative of using the message passing model, while at the same time exploiting the threading capabilities of the SMP.

With threading, there is a fine degree of job distribution within a parallel task, because of the added granularity. Thus, a task can have a number of threads, with some or all of the threads doing MPI communications with external tasks, while the rest continue their computation.

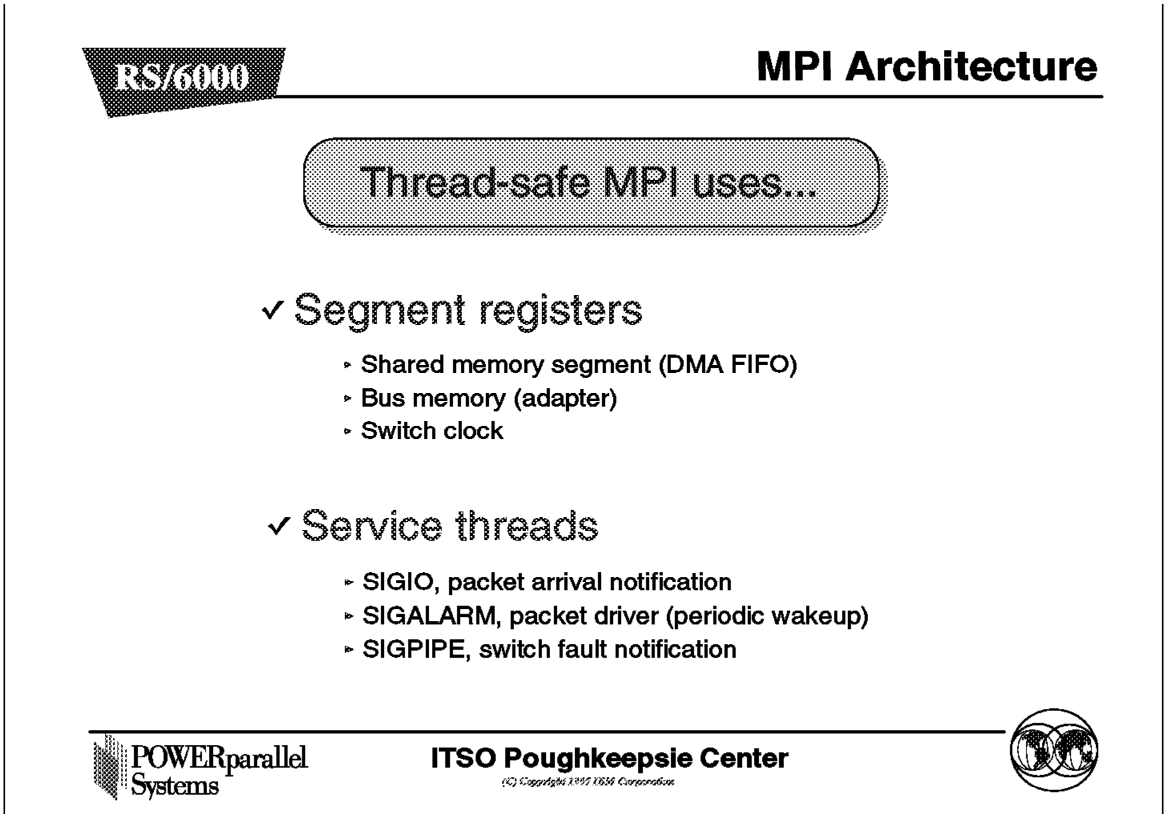
Previously, whenever blocking MPI calls were used in a task, the entire task would block. With the use of threads, one single thread could be assigned to block, while the rest continue their computations. This method of overlapping potentially decreases the execution time of the task.

**Note**

It must be noted here that if the algorithm of the MPI program was not written to utilize threads by overlapping computation and communication, then linking in the thread-safe MPI library offers little or no improvement at all.

Moreover, the I/O performance of a uniprocessor is currently superior to that of an SMP, so unless the improvements gained by programming in threads and overlapping execution far outweigh the slower I/O performance of the SMP, performance would only be, at best, comparable to a uniprocessor.

## 9.2.1 MPI Architecture



RS/6000

**MPI Architecture**

Thread-safe MPI uses...

✓ Segment registers

- **Shared memory segment (DMA FIFO)**
- **Bus memory (adapter)**
- **Switch clock**

✓ Service threads

- **SIGIO, packet arrival notification**
- **SIGALARM, packet driver (periodic wakeup)**
- **SIGPIPE, switch fault notification**

POWERparallel Systems

**ITSO Poughkeepsie Center**

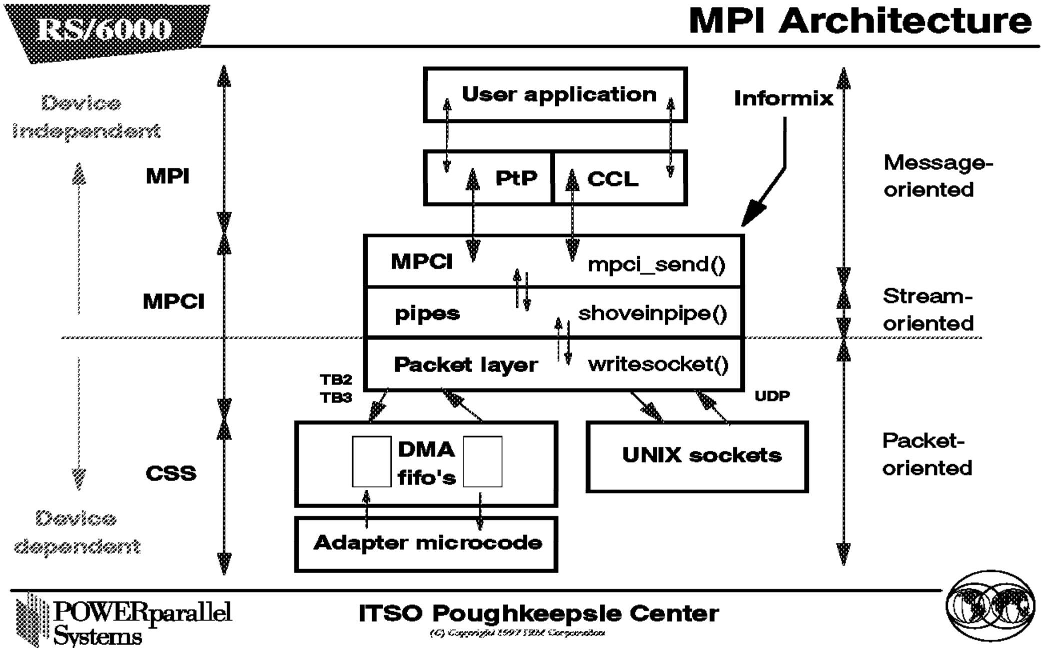
(C) Copyright 1997 IBM Corporation

MPI uses up to three of the sixteen segment registers available in an AIX process to keep track of the DMA window, the microchannel bus adapter memory, and the switch clock.

If LAPI is used, it requires an additional two segment registers.

Instead of using signals for packet notification, communications driving, and switch fault notification, thread-safe MPI implements these services with threads.

For SIGIO and SIGALRM, two threads were dedicated to handling the associated events. The handler for SIGPIPE is now coded in MPI, which periodically checks and relinquishes the adapter if necessary.



The above figure shows the code flow in MPI.

The layer from the pipes up to the user application layer defines the device-independent layer, while from the packet layer downwards to the adapter defines the device-dependent layer. When there is an upgrade or change in the microcode or hardware, only the device-dependent layer needs to be rewritten.

The trace of code flow is as follows:

**Layer** — **Function**

**User application** — The application makes an MPI call, either for point-to-point communications or for group communications.

**MPI library** — MPI functions in turn call MPCI functions to deliver or receive messages. The semantics of message passing is enforced in this layer, including group communications.

**MPCI** — MPCI consists of three sublayers:

1. MPCI
2. pipes
3. packet layer

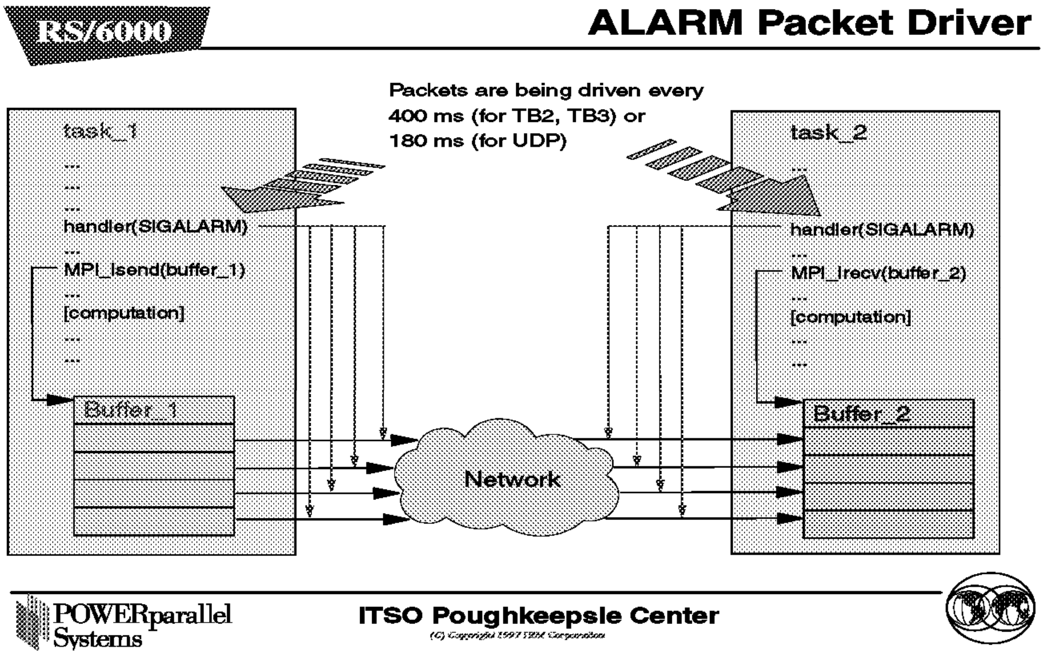
The first sublayer translates the MPI calls into low-level pipe calls. Note that MPCI only provides primitives for point-to-point communications. Group communication semantics is handled by the MPI layer.

**Pipes** Pipes are low-level calls that do not understand the abstract of messages. This layer views all data as streams. Flow control and error recovery are enforced in this layer.

**Packet layer** This layer takes care of moving into the next lower layer by first breaking up and packetizing the stream data into suitable MTU sizes. It also retrieves packets from the lower layer and reassembles them. For the TB2 and TB3 adapters, the packets are sent into or received from the DMA FIFO memory. Data is exchanged with UNIX sockets in the case of UDP.

**CSS** This is the Communication Subsystem layer; it applies only to TB2 and TB3 adapters. This layer includes the DMA FIFOs and the adapter microcode. The DMA FIFO is allocated from the shared memory segment that is attached to the user process. This memory is pinned for DMA by the CSS kernel extensions. By pinned it means that the memory used for DMA FIFOs is not pagable, that is it stays locked in the memory at all times.

## 9.2.2 MPI Service: ALARM Packet Driver



RS/6000

**ALARM Packet Driver**

Packets are being driven every
400 ms (for TB2, TB3) or
180 ms (for UDP)

POWERparallel Systems

ITSO Poughkeepsie Center

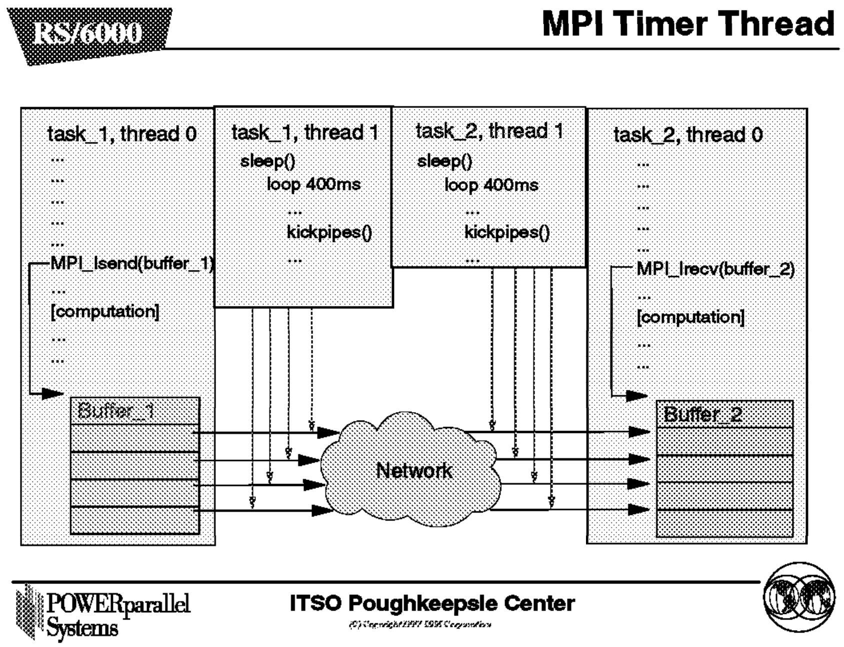
(C) Copyright 1997 IBM Corporation

This figure explains how the ALARM service is implemented in the signal-based MPI library.

Depending on the adapter used, a SIGALRM is issued to each MPI task periodically (400 ms for TB2 and TB3 and 180 ms for UDP). The purpose of this is to drive or proceed with communications. Data is moved out of or into the appropriate buffers at these intervals, so that messages can get sent or received.

Suppose we have two tasks, task_1 and task_2, with task_1 calling MPI to do a nonblocking send to task_2. Here is how MPI communications is driven:

| **Task** | **Code flow** |
|---|---|
| **task_1** | As soon as MPI processes the MPI_Isend call, it initiates the communication and program execution resumes, even though the message is still being sent. However, a SIGALRM will periodically be sent to this task. The signal handler for SIGALRM will then check for pending messages to be sent and process them accordingly. |
| **task_2** | In this task, after the nonblocking MPI_Irecv has returned, program execution proceeds until interrupted by a SIGALRM. The signal handler for this signal will then be triggered and start moving data into the receive buffer. |



RS/6000

## MPI Timer Thread

task_1, thread 0
...
...
...
...
...
MPI_Isend(buffer_1)
...
[computation]
...
...

task_1, thread 1
sleep()
loop 400ms
...
kickpipes()
...

task_2, thread 1
sleep()
loop 400ms
...
kickpipes()
...

task_2, thread 0
...
...
...
...
...
MPI_Irecv(buffer_2)
...
[computation]
...
...

Buffer_1

Network

Buffer_2

POWERparallel Systems

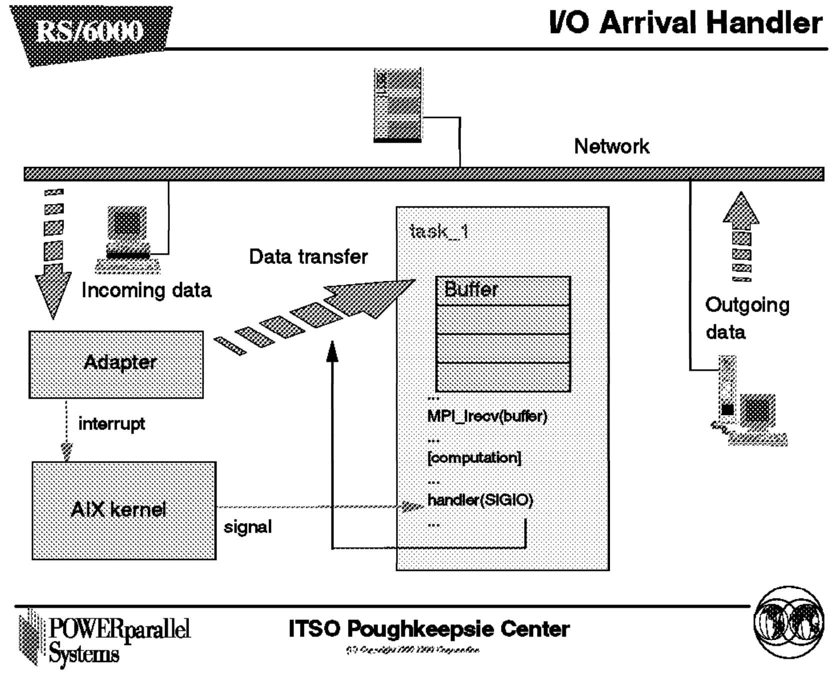
ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

In the thread-based MPI, SIGALRM is no longer used. Instead, we have a separate thread that drives the communications at intervals similar to the signal-based version, that is, every 400 ms for TB2 and TB3, and every 180 ms for UDP.

Assuming that TB2 or TB3 is used, we have the timer thread (thread 1 of both tasks) waking up every 400 ms and issuing a kickpipes call to drive the communications.

### 9.2.3 MPI Service: I/O Arrival Handler



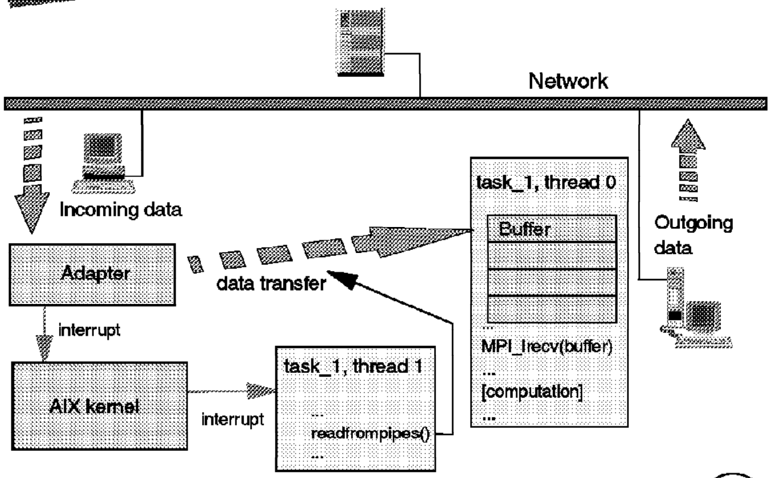
The above figure describes the SIGIO mechanism in the signal-based MPI library.

Whenever there is incoming data destined for a task, a SIGIO signal is sent to the task via the AIX kernel.

The signal handler for SIGIO in the task will then wake up and start moving data into the message buffer.

RS/6000

## MPI Interrupt Thread



POWERparallel Systems

ITSO Poughkeepsie Center

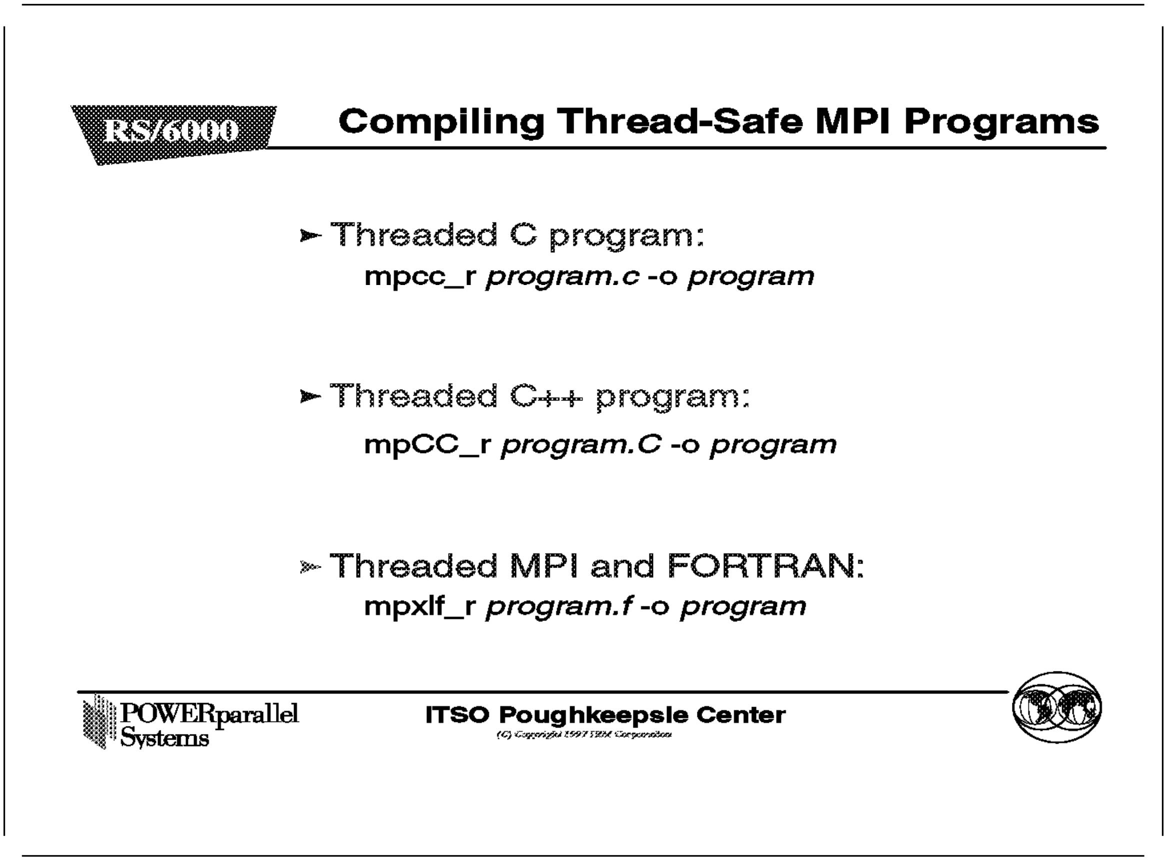
(c) Copyright 1997 IBM Corporation

In the thread-safe MPI library, signals are no longer sent to the active process. Instead, the SIGIO interrupt handler is now replaced by a separate dedicated thread to handle incoming data.

In this figure, thread_1 is the thread containing the interrupt handler code. The AIX kernel extension wakes thread_1, which in turn is responsible for moving the data from the DMA FIFO into the message buffer. At the same time, thread_0, which is the computation thread, continues its execution uninterrupted.

This method of dedicating an MPI service to a separate thread has the advantage of overlapping the communications and computation windows of execution, in effect potentially increasing the speed of execution.

## 9.3 Using Thread-Safe MPI



RS/6000

### Compiling Thread-Safe MPI Programs

- Threaded C program:
  **mpcc_r** ***program.c*** **-o** ***program***

- Threaded C++ program:
  **mpCC_r** ***program.C*** **-o** ***program***

- Threaded MPI and FORTRAN:
  **mpxlf_r** ***program.f*** **-o** ***program***

POWERparallel Systems

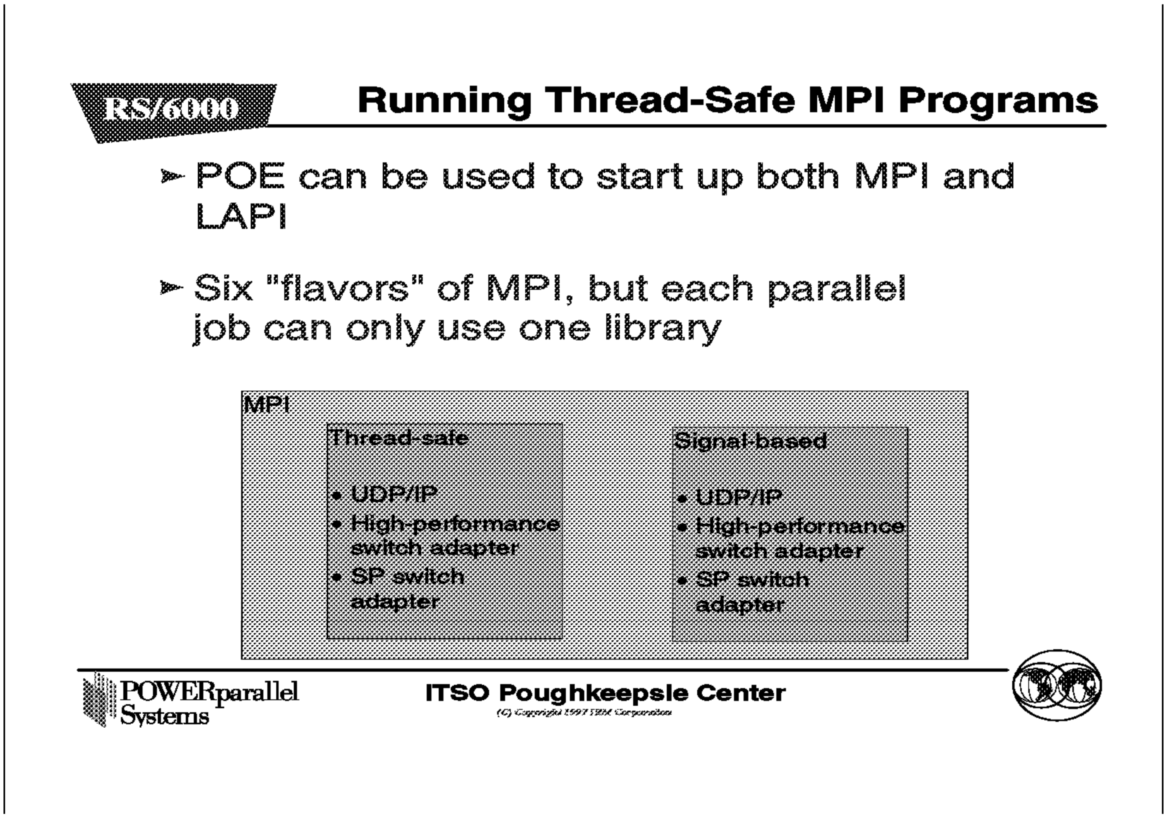
ITSO Poughkeepsie Center
(C) Copyright 1997 IBM Corporation

Instead of using the cc, xlC, or xlf commands, we compile parallel programs using the commands mpcc, mpCC, or mpxlf.

These commands not only compile the program, but also link in the Partition Manager and message passing interface.

To compile threaded C, C++, or Fortran programs, we use the mpcc_r, mpCC_r, or mpxlf_r commands (the prerequisite for threaded MPI with FORTRAN is XLF 4.1.0.1).

A communication subsystem library implementation will be dynamically linked when the executable program is invoked.



RS/6000

## Running Thread-Safe MPI Programs

- POE can be used to start up both MPI and LAPI
- Six "flavors" of MPI, but each parallel job can only use one library

MPI

Thread-safe

- UDP/IP
- High-performance switch adapter
- SP switch adapter

Signal-based

- UDP/IP
- High-performance switch adapter
- SP switch adapter

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

This foil is an illustration of how MPI can be run.

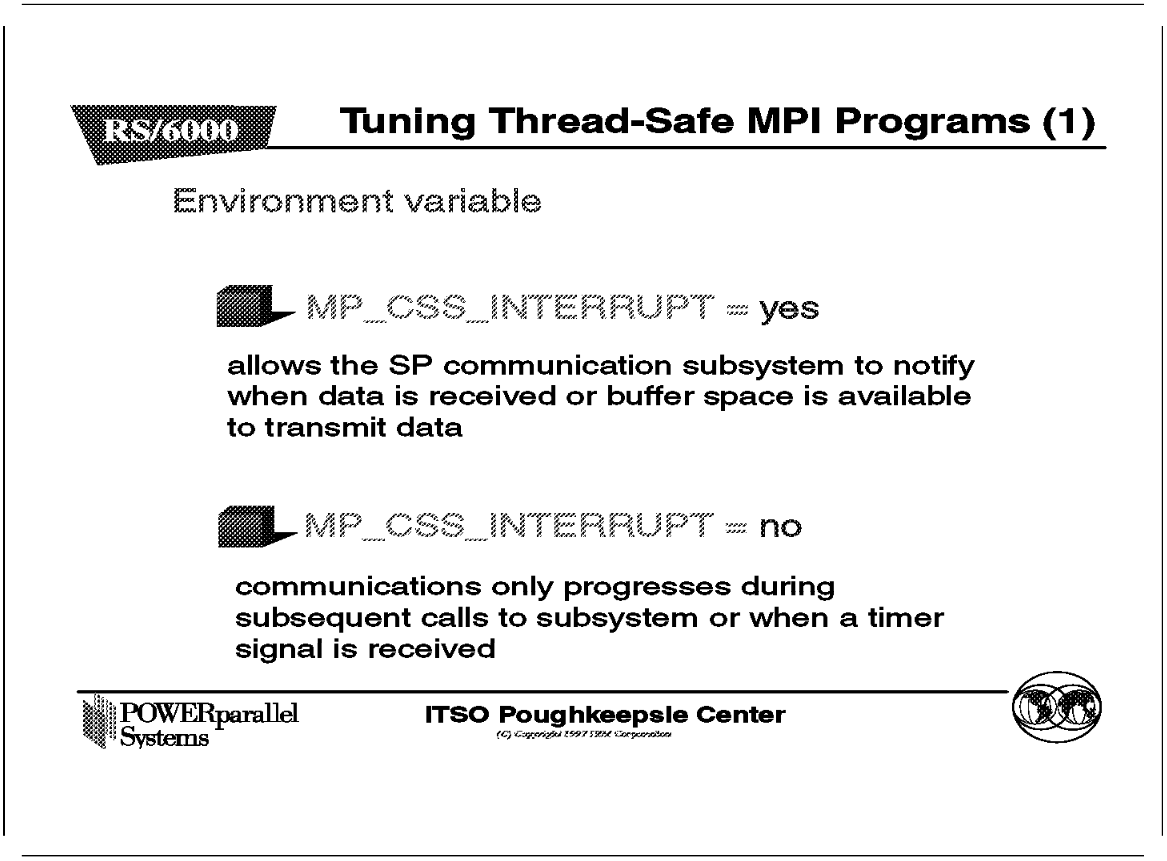
Parallel Operating Environment (POE) must be used to start MPI, whether the parallel job is using the signal-based or thread-based library. In addition, POE must be used to run LAPI tasks. MPI and LAPI calls can coexist in the same task.

There are six “flavors” of MPI:

- Thread-safe MPI, using UDP/IP
- Thread-safe MPI, using the High-Performance Switch adapter
- Thread-safe MPI, using the SP Switch adapter
- Signal-based MPI, using UDP/IP
- Signal-based MPI, using the High-Performance Switch adapter
- Signal-based MPI, using the SP Switch adapter

**Note**

Only one library is permitted to be used in any one parallel job.



RS/6000

## Tuning Thread-Safe MPI Programs (1)

Environment variable

- MP_CSS_INTERRUPT = yes

  allows the SP communication subsystem to notify when data is received or buffer space is available to transmit data

- MP_CSS_INTERRUPT = no

  communications only progresses during subsequent calls to subsystem or when a timer signal is received

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

The MP_CSS_INTERRUPT environment variable may take the value of either yes or no. By default it is set to no. In certain applications, setting this value to yes may improve performance.

To understand how this parameter works, it is important to first understand how the SP communication subsystem regains control from the user space to complete asynchronous requests for communication.

For asynchronous communication calls, the calls to the nonblocking send or receive routines do not actually ensure the transmission of data from one node to the next, but only post the send or receive and then return immediately to the user application to resume execution. Since the SP communications subsystem is a user protocol, it must regain control from the application to complete asynchronous requests for communication.

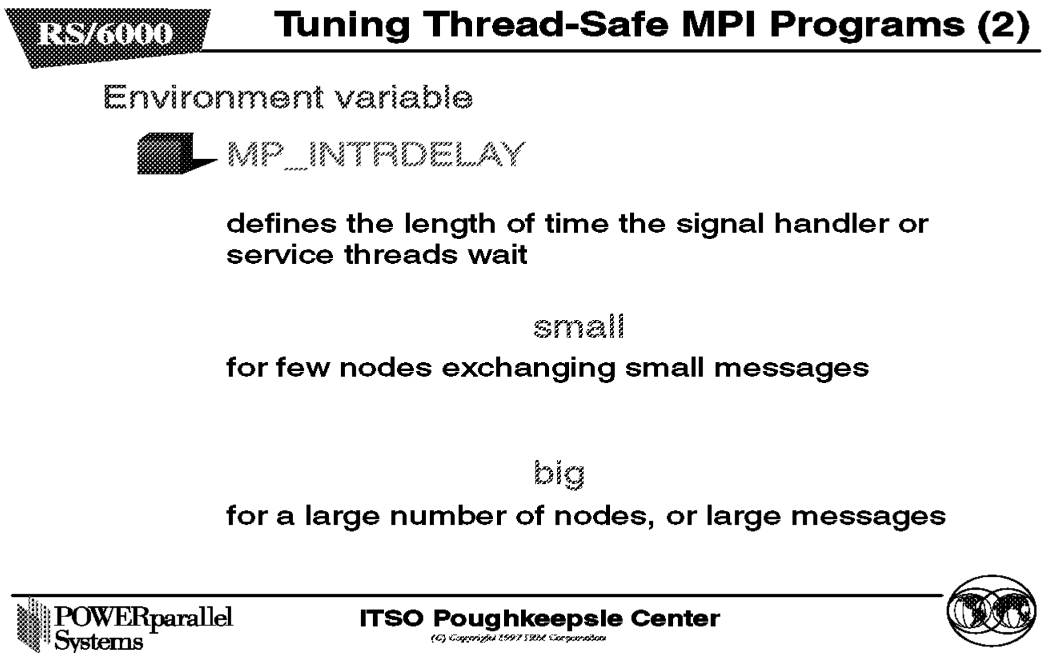
With MP_CSS_INTERRUPT set to no, the communication subsystem will only proceed with communications when:

- Subsequent calls are made to the SP communication subsystem to send, receive or wait on messages.
- A timer signal is received periodically.

With the MP_CSS_INTERRUPT variable set to yes, the communication subsystem device driver sends a signal to the user application when data is received or buffer space is available to transmit data.

This is especially useful for applications that:

- Use nonblocking communications
- Use non-synchronized sets of send or receive pairs
- Do not issue waits for nonblocking send or receive operations, but rather do some computation prior to issuing the waits



RS/6000

## Tuning Thread-Safe MPI Programs (2)

Environment variable

MP_INTRDELAY

**defines the length of time the signal handler or service threads wait**

small

**for few nodes exchanging small messages**

big

**for a large number of nodes, or large messages**

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1997 IBM Corporation

When a node receives a packet and an interrupt is generated, the interrupt handler checks its table for the process identifier of the user process and notifies the process. The signal handler or service threads wait for at least two times the interrupt delay, checking to see if more packets arrive. Waiting for more packets avoids the cost of incurring an interrupt each time a new packet arrives (interrupt processing is very expensive). However, the more packets that arrive, the more the delay time is increased.

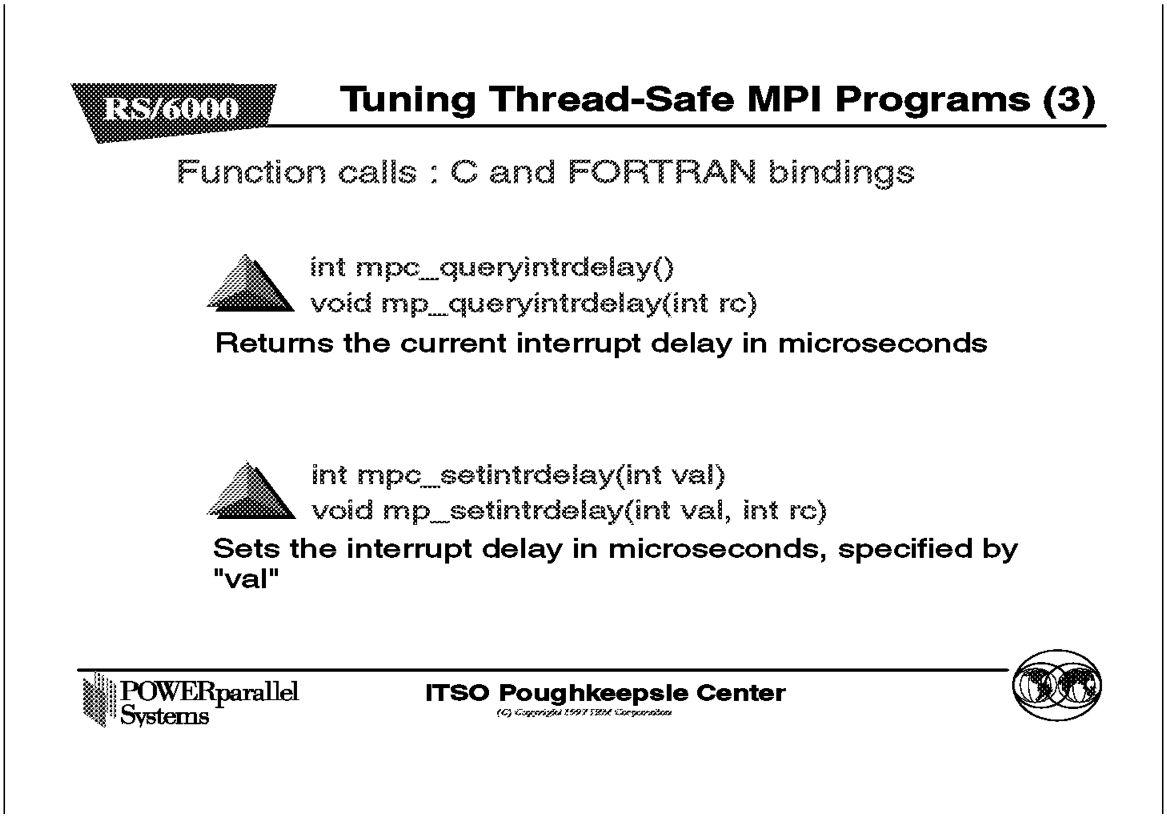
The MP_INTRDELAY environment variable allows you to set the delay parameter for how long the signal handler or service threads wait for more data.

For an application with only a few nodes exchanging small messages, it will help latency if the interrupt delay is set to a small value.

For an application with a large number of nodes or one which exchanges large messages, keeping the interrupt delay large will help bandwidth, as a large delay allows multiple read transmissions to occur in a single read cycle.

The exact value of MP_INTRDELAY is application-dependent and is discovered through experimentation.

The default values are 35 microseconds for the HP Switch adapter, and 1 microsecond for the SP Switch adapter.



RS/6000

## Tuning Thread-Safe MPI Programs (3)

Function calls : C and FORTRAN bindings

int mpc_queryintrdelay()
void mp_queryintrdelay(int rc)

**Returns the current interrupt delay in microseconds**

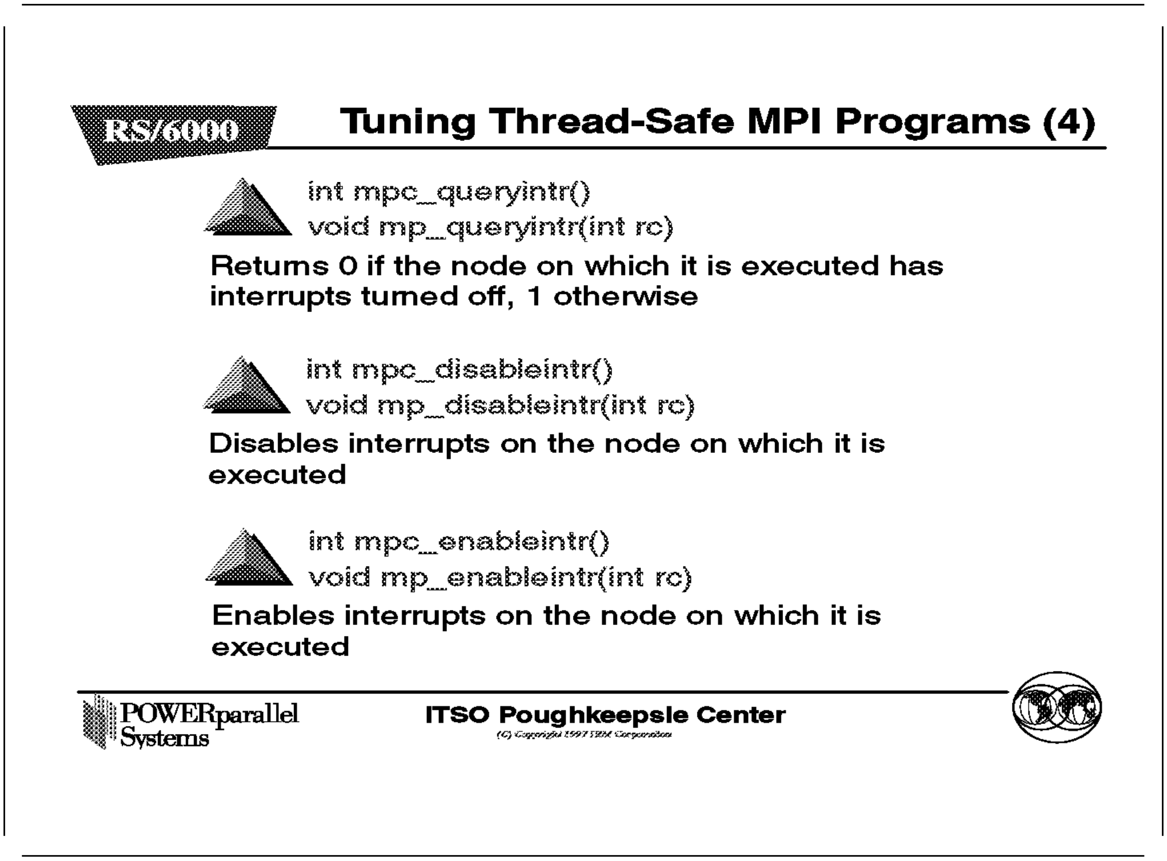
int mpc_setintrdelay(int val)
void mp_setintrdelay(int val, int rc)

**Sets the interrupt delay in microseconds, specified by "val"**

POWERparallel Systems

**ITSO Poughkeepsie Center**

These two function calls allow the programmer to query the current interrupt delay, and also to set it dynamically. The interrupt delay value here is the same delay value that the MP_INTRDELAY would specify.



RS/6000

## Tuning Thread-Safe MPI Programs (4)

- int mpc_queryintr()
  void mp_queryintr(int rc)

Retums 0 if the node on which it is executed has interrupts tumed off, 1 otherwise

- int mpc_disableintr()
  void mp_disableintr(int rc)

Disables interrupts on the node on which it is executed

- int mpc_enableintr()
  void mp_enableintr(int rc)

Enables interrupts on the node on which it is executed

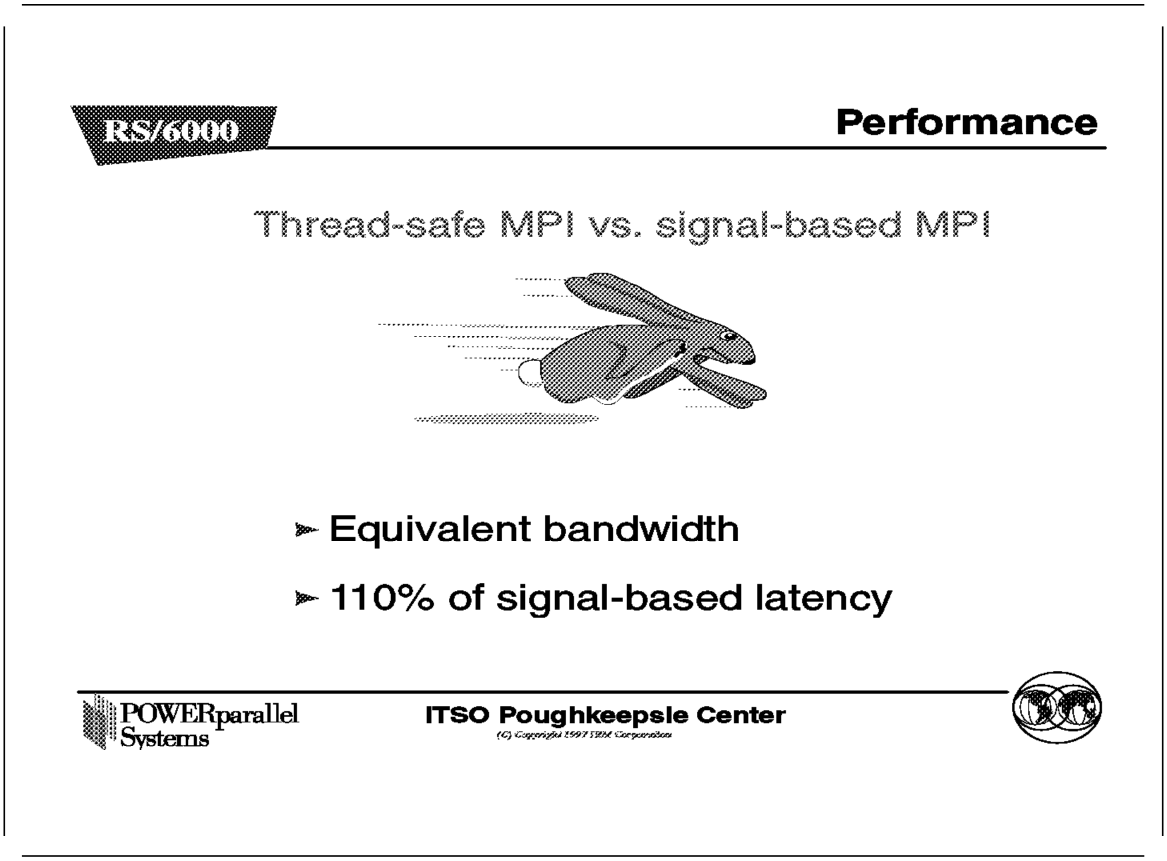
POWERparallel Systems

ITSO Poughkeepsle Center

(C) Copyright 1997 IBM Corporation

The enable and disable interfaces for interrupts override the setting of the MP_CSS_INTERRUPT environment variable.

## 9.3.1 Performance



RS/6000

**Performance**

Thread-safe MPI vs. signal-based MPI

- Equivalent bandwidth
- 110% of signal-based latency

POWERparallel Systems

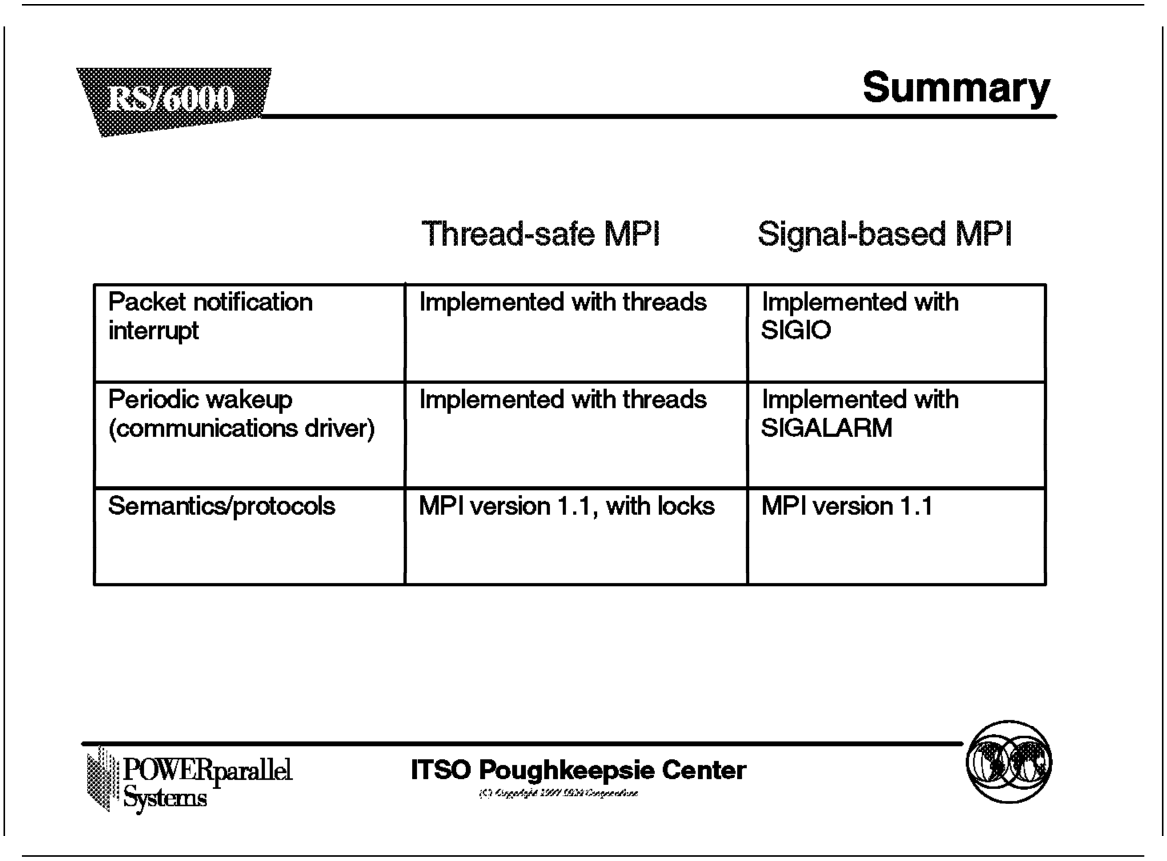
**ITSO Poughkeepsie Center**

(C) Copyright 1997 IBM Corporation

The above figure compares the performance of the thread-safe MPI library to that of the signal-based MPI library.

Both libraries offer about the same amount of bandwidth, but the thread-safe MPI library is about 15% slower in terms of latency than the signal-based MPI.

### 9.3.2 Summary



RS/6000

**Summary**

|  | Thread-safe MPI | Signal-based MPI |
|---|---|---|
| Packet notification interrupt | Implemented with threads | Implemented with SIGIO |
| Periodic wakeup (communications driver) | Implemented with threads | Implemented with SIGALARM |
| Semantics/protocols | MPI version 1.1, with locks | MPI version 1.1 |

POWERparallel Systems

ITSO Poughkeepsie Center

The above figure summarizes the key differences between the thread-safe MPI and signal-based MPI libraries.

MPL provides the function MP_RcvNCall, which allows a user to specify a handler that is to execute when the receive completes.

The execution of the callback function is atomic and asynchronous and can interrupt the user′s main flow of code at any time, but cannot itself be interrupted by user code or by another handler.

The handler semantic of MPL is not consistent with a threaded environment so neither MPL Receive and Call nor its precise semantic are currently available in the thread-safe MPI library or in the standard. However, the signal-based MPI library provides both MPL and MPL Receive and Call.

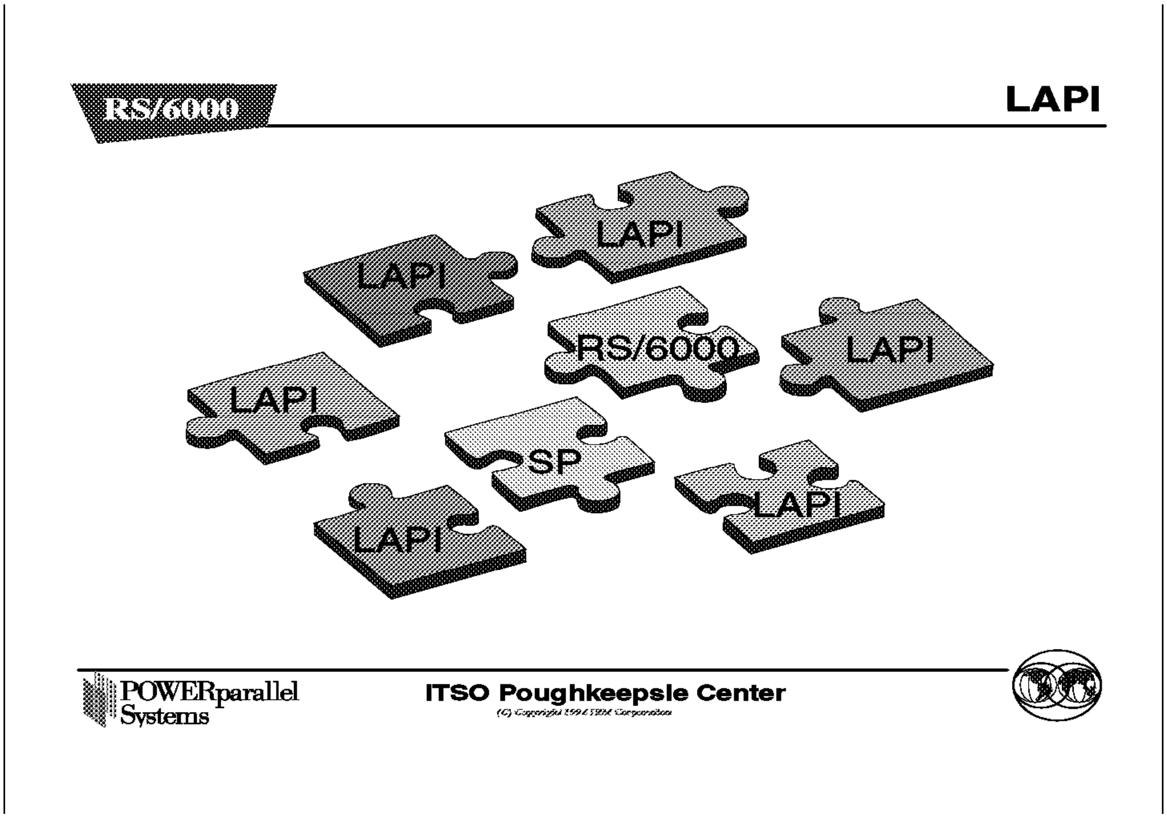
SIGIO, SIGALRM services

MPI no longer uses the SIGIO or SIGALRM signals. Instead, separate threads are created, which handle the conditions that were handled using these signals.

Semantics

No MPI semantics were modified. However, in a multi-threaded program, the implementation of MPI calls is modified such that it is now allowed to call an MPI function while another MPI function is active. The result is that the second MPI function is delayed, instead of failing, until the first function releases its lock. This also permits multiple user threads to simultaneously make MPI calls, such that if one thread sits inside a blocking MPI call, it will not indefinitely delay any other threads. Instead, it will periodically unlock the MPI library in order to allow other threads to make progress in their MPI calls.

# Chapter 10. Overview of LAPI



RS/6000

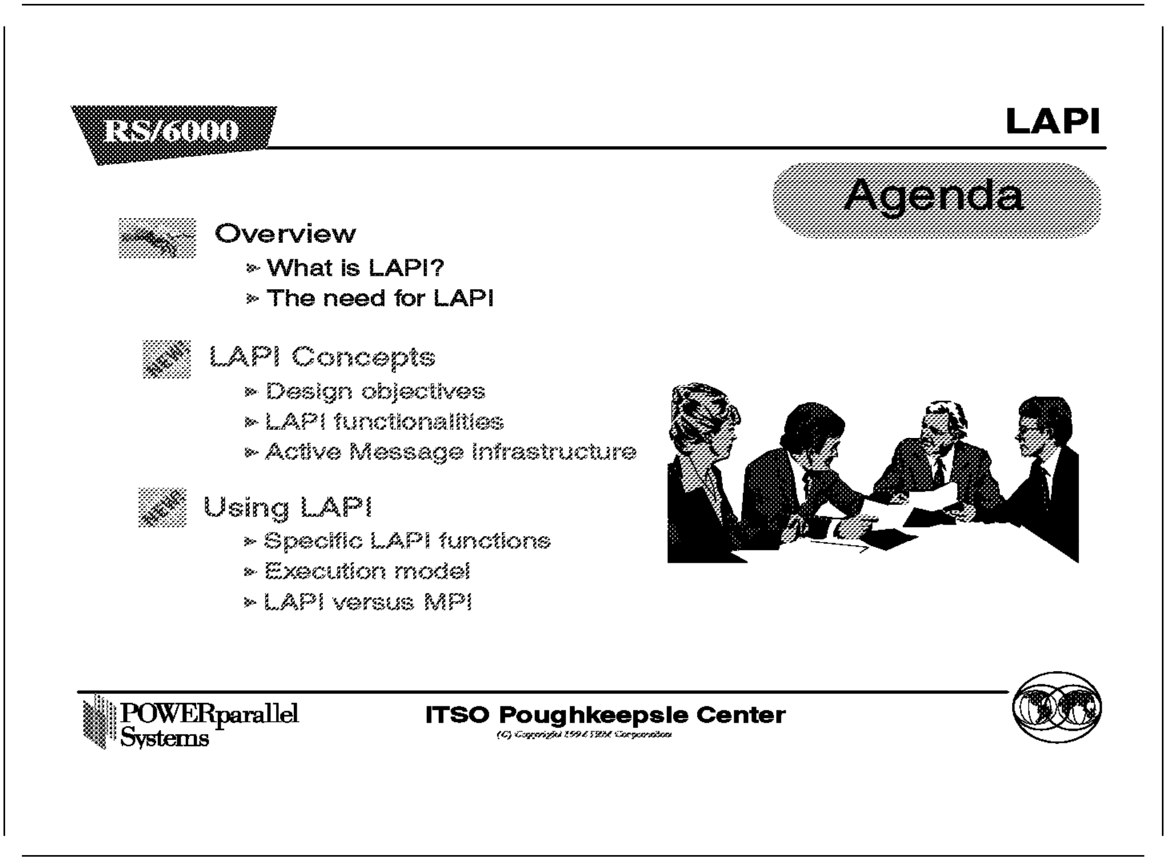
LAPI

POWERparallel Systems

ITSO Poughkeepsle Center

(C) Copyright 1994 IBM Corporation

This chapter discusses the Low-level Applications Programming Interface (LAPI).



RS/6000

## LAPI

Agenda

- Overview
  - What is LAPI?
  - The need for LAPI
- LAPI Concepts
  - Design objectives
  - LAPI functionalities
  - Active Message Infrastructure
- Using LAPI
  - Specific LAPI functions
  - Execution model
  - LAPI versus MPI

POWERparallel Systems

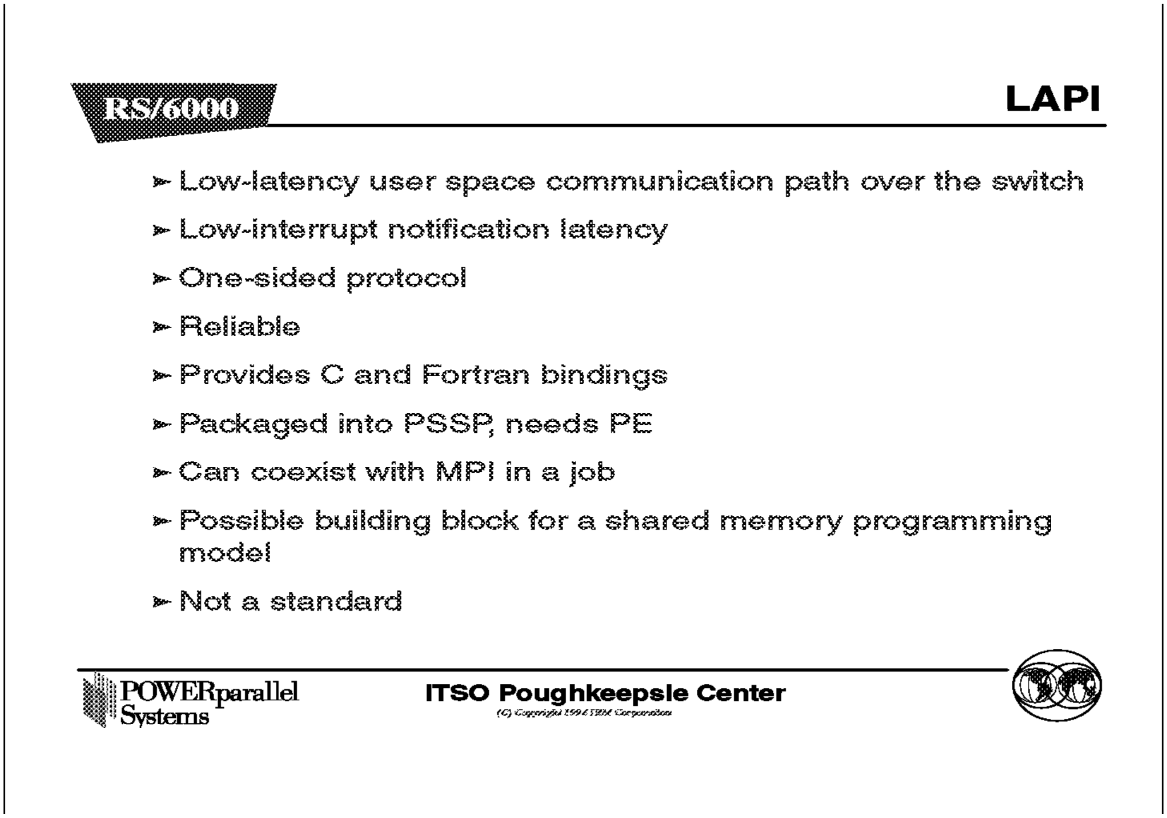
ITSO Poughkeepsle Center

We first take a look at what LAPI is, followed by justifications for its development.

Next, we discuss the design objectives, what LAPI functions provide, and the Active Message infrastructure, which is a very important concept in writing LAPI programs.

We close this chapter by looking at a summary of some specific LAPI calls, the LAPI execution model that discusses how LAPI programs are run, and finally a short comparison between LAPI and the Message Passing Library (MPI).

### 10.1.1 What Is LAPI?



RS/6000

LAPI

- Low-latency user space communication path over the switch
- Low-interrupt notification latency
- One-sided protocol
- Reliable
- Provides C and Fortran bindings
- Packaged into PSSP, needs PE
- Can coexist with MPI in a job
- Possible building block for a shared memory programming model
- Not a standard

POWERparallel Systems

ITSO Poughkeepsie Center

LAPI offers low latencies in terms of communications over the switch and interrupt handling.

The LAPI semantic is unilateral. This means that one process initiates a LAPI operation, and the completion of the operation does not require any other process to take some complementary action. This is unlike MPI′s send and receive, where a send requires a complementary receive with matching parameters to be posted for completion, and vice versa.

LAPI is also reliable, because it provides for flow control and guaranteed delivery of messages. However, it does not provide message ordering. That is the programmer′s responsibility.

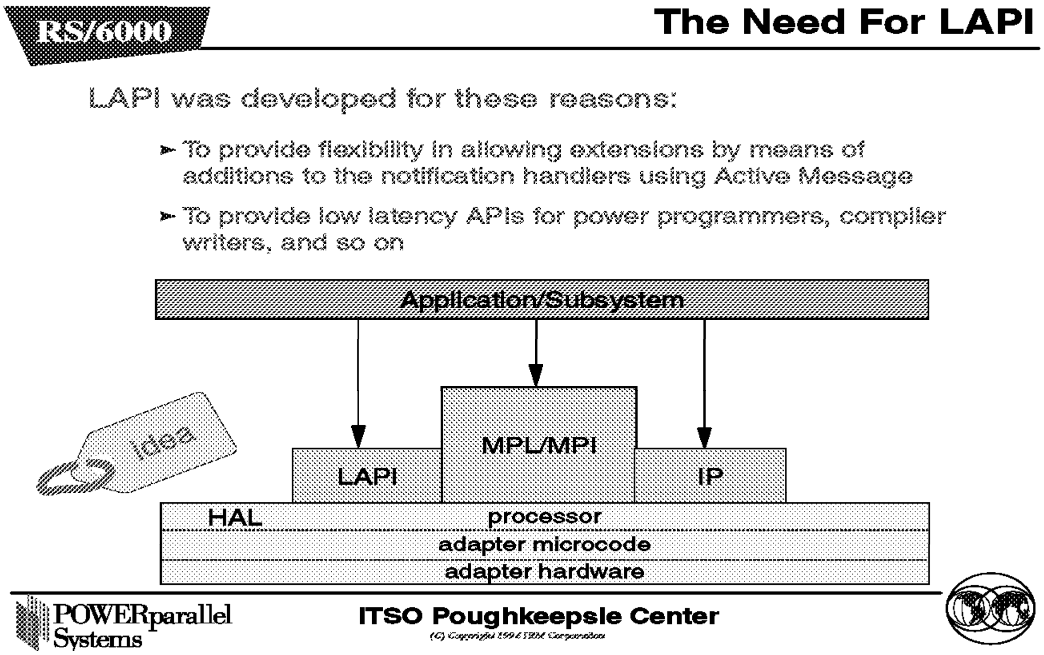
LAPI functions can either be invoked as C or Fortran calls. Header files are provided for both languages.

LAPI is packaged with PSSP, and requires PE. LAPI programs are started by POE, much the same way as MPI programs. In fact, LAPI and MPI calls can exist within the same program, or the same parallel job.

LAPI, with its efficient primitives, also offers itself as a possible building block for a higher-level shared memory programming model.

Lastly, LAPI is not a standard interface like MPI, but only IBM′s own interface.

### 10.1.2 The Need For LAPI



RS/6000

**The Need For LAPI**

LAPI was developed for these reasons:

- To provide flexibility in allowing extensions by means of additions to the notification handlers using Active Message
- To provide low latency APIs for power programmers, compiler writers, and so on

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1994 IBM Corporation

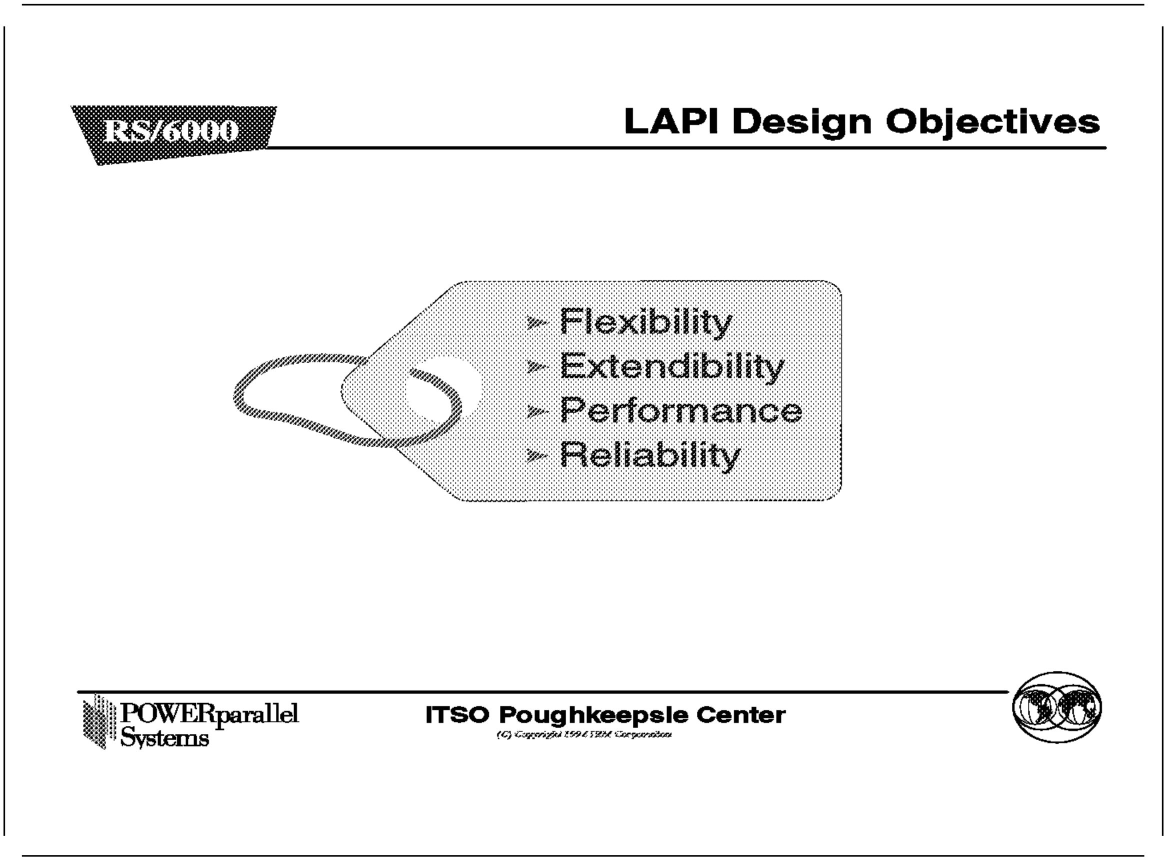
There are three reasons for using LAPI:

- Flexibility in programming
- Good performance for interrupt latency
- Good bandwidth for small and medium messages

The LAPI library provides PUT and GET functions and a general "Active Message" function to allow programmers to supply extensions by means of additions to the notification handlers. This is similar to letting the programmers define their own callback functions, and provides for a good degree of customization and flexibility.

LAPI was also designed to provide optimal communication performance on the SP switch.

## 10.2 LAPI Concepts



RS/6000

LAPI Design Objectives

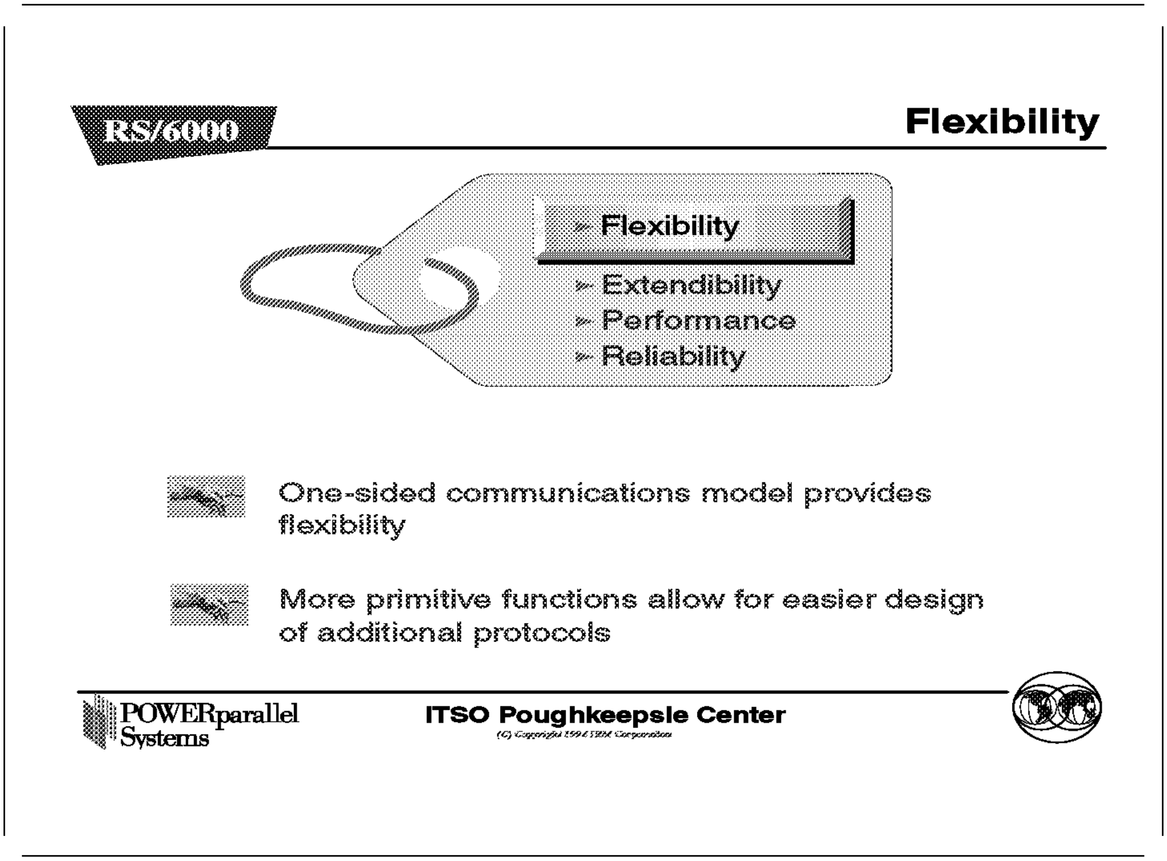
- Flexibility
- Extendibility
- Performance
- Reliability

POWERparallel Systems

ITSO Poughkeepsle Center

(C) Copyright 1996 IBM Corporation

LAPI was designed for performance, flexibility, extendibility and reliability.



RS/6000

## Flexibility

- Flexibility
- Extendibility
- Performance
- Reliability

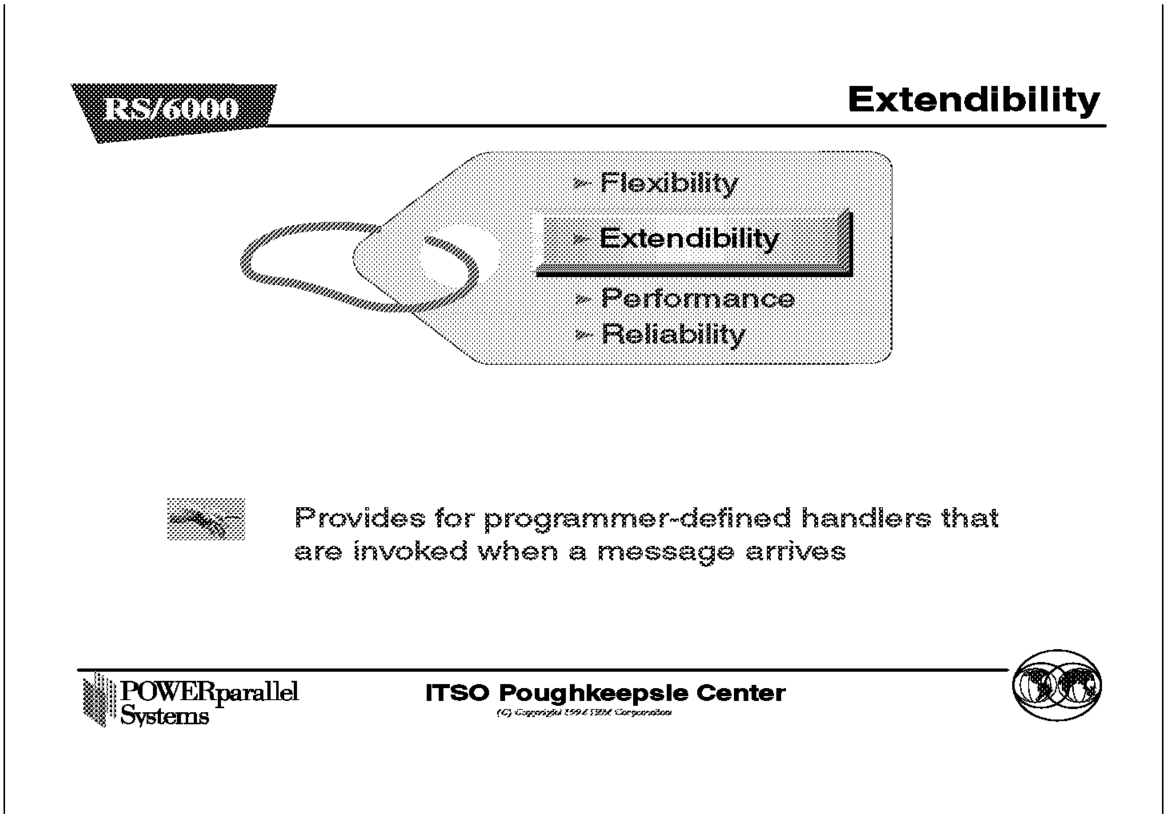
- One-sided communications model provides flexibility
- More primitive functions allow for easier design of additional protocols

POWERparallel Systems

ITSO Poughkeepsie Center

LAPI is designed to cater to a diverse set of users.

A key goal is to define LAPI so that it can be easily extended functionally by the user. This allows users to customize LAPI to their specific environments.



RS/6000

## Extendibility

- Flexibility
- **Extendibility**
- Performance
- Reliability

Provides for programmer-defined handlers that are invoked when a message arrives

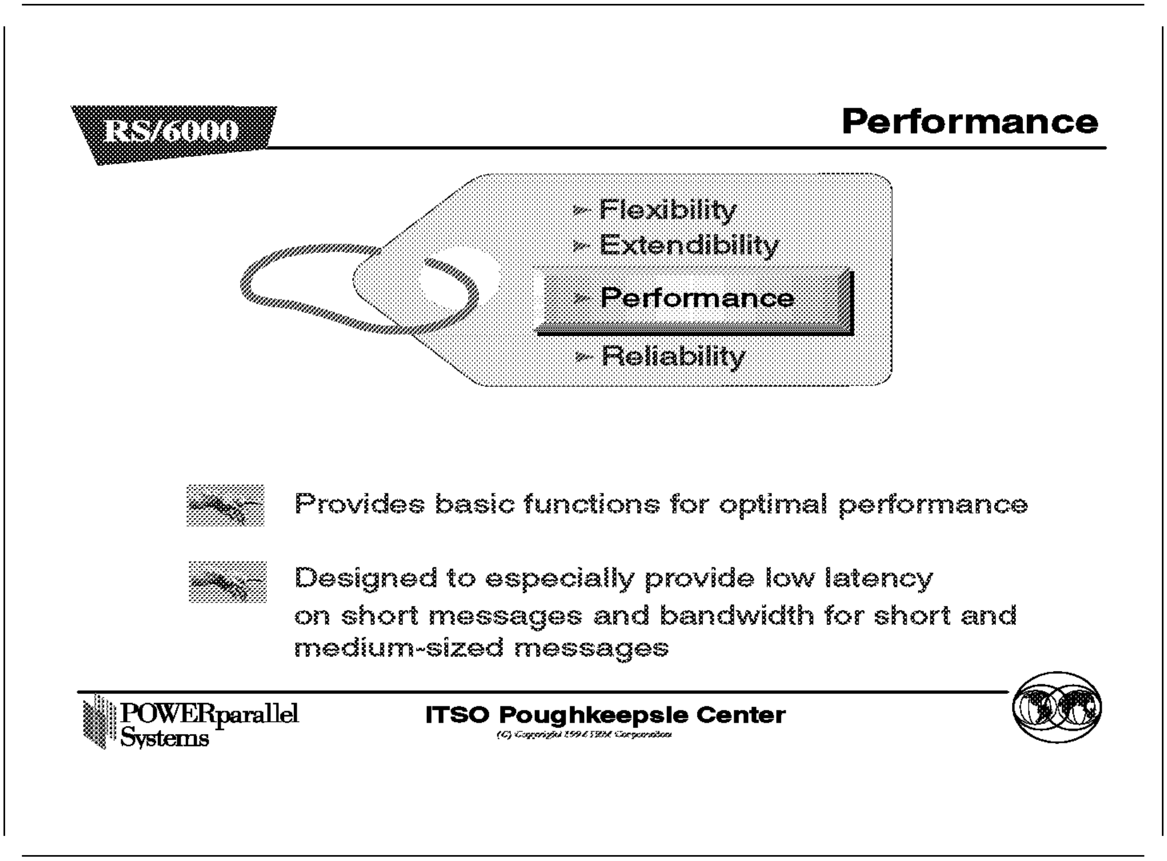
POWERparallel Systems

ITSO Poughkeepsie Center

LAPI should be flexible and expressive enough to accommodate diverse applications and algorithms.

In particular, it should be more flexible than the standard send/receive protocol. The key limitation of the send/receive protocol is that it is bilateral, requiring processes at both source and destination to explicitly participate in the communication.

This makes programming difficult in situations where one of the “participants” is unaware of the identity of the other. Such situations arise in applications that have dynamically changing and unpredictable communications.



RS/6000

## Performance

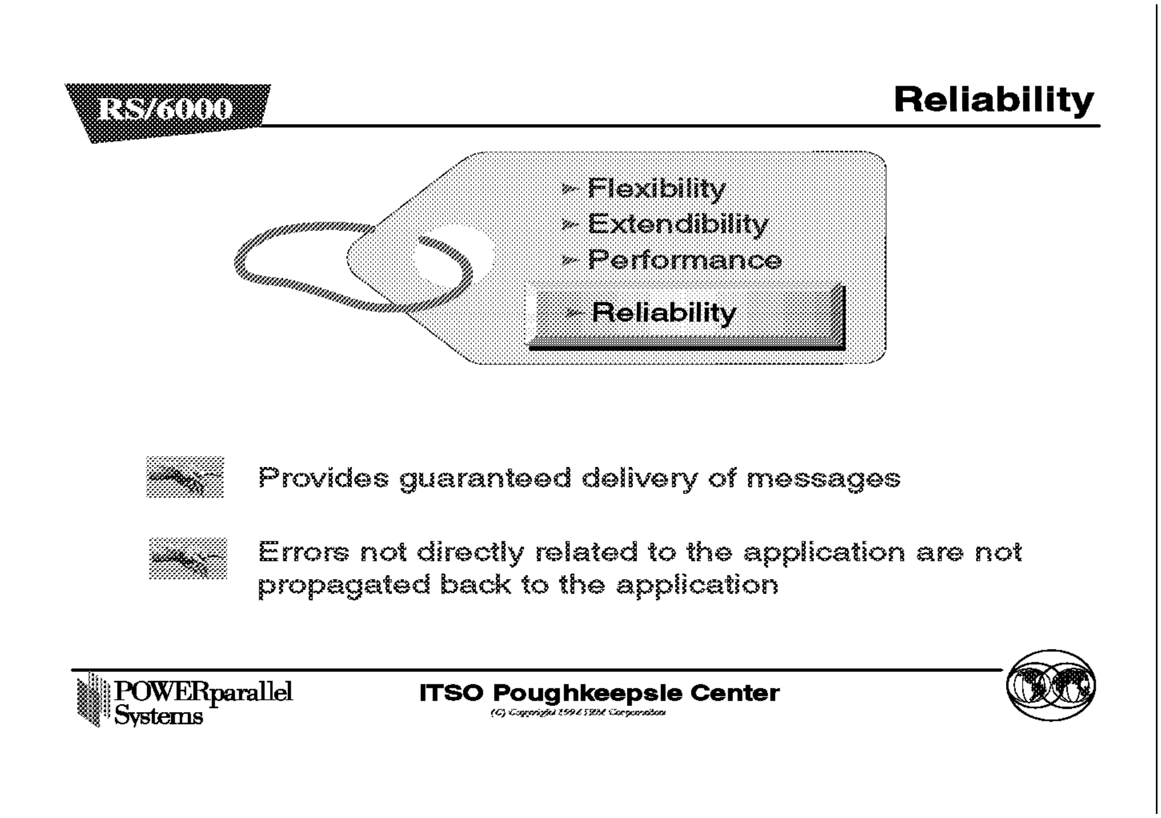
- ➢ Flexibility
- ➢ Extendibility
- ➢ **Performance**
- ➢ Reliability

- Provides basic functions for optimal performance
- Designed to especially provide low latency on short messages and bandwidth for short and medium-sized messages

POWERparallel Systems

ITSO Poughkeepsie Center

The LAPI is designed to allow performance-optimized implementation as a thin layer. Performance is measured by latency, bandwidth, overhead, and ability to overlap (of computation and communication). Of these, latency on short messages is the key.



RS/6000

## Reliability

- ➢ Flexibility
- ➢ Extendibility
- ➢ Performance
- ➢ **Reliability**

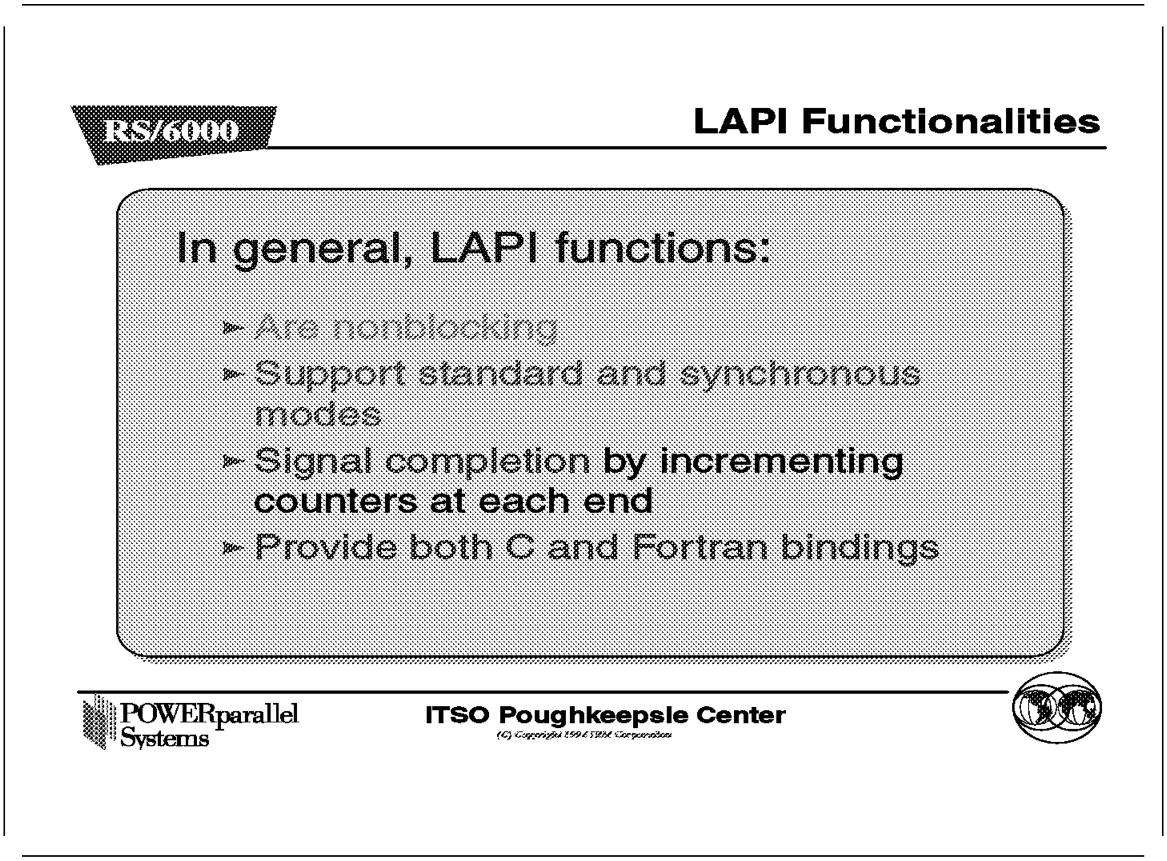
- Provides guaranteed delivery of messages
- Errors not directly related to the application are not propagated back to the application

POWERparallel Systems

**ITSO Poughkeepsle Center**

The LAPI implementation must provide reliable communication. Errors not directly related to the application must not be propagated back to the application.

## 10.2.1 LAPI Functions



RS/6000

LAPI Functionalities

In general, LAPI functions:

- Are nonblocking
- Support standard and synchronous modes
- Signal completion **by incrementing counters at each end**
- Provide both C and Fortran bindings

POWERparallel Systems

ITSO Poughkeepsle Center

(C) Copyright 1994 IBM Corporation

This foil is a summary of the functionalities of LAPI.

LAPI functions are in general nonblocking, that is, they may return before the operation is complete and before the user is allowed to reuse all the resources specified in the call. A nonblocking operation is considered to be complete only after a completion testing function (such as lapi_waitcntr or lapi_getcntr) has indicated that the operation has completed.

To understand the two different modes (standard and synchronous) of LAPI communications and their relationship with the counters, the semantics of the counters must first be explained. The term *origin process* refers to the process that initiated the LAPI call, and *target process* refers to the process that the LAPI call operates on, that is, the destination process.

**Counter** **Semantic**

**org_cntr** This is the origin counter, a variable stored at the origin process. Whenever a LAPI call using this counter is initiated, this value is incremented by one once the data has been copied out of the origin buffer. Incrementing this counter implies that the origin buffer space is safe to reuse.

**tgt_cntr** This is the target counter, a variable stored at the target process. This counter, if used, is incremented by one after data arrives at the target. After it has been incremented, it is safe to access the data in the target buffer space.

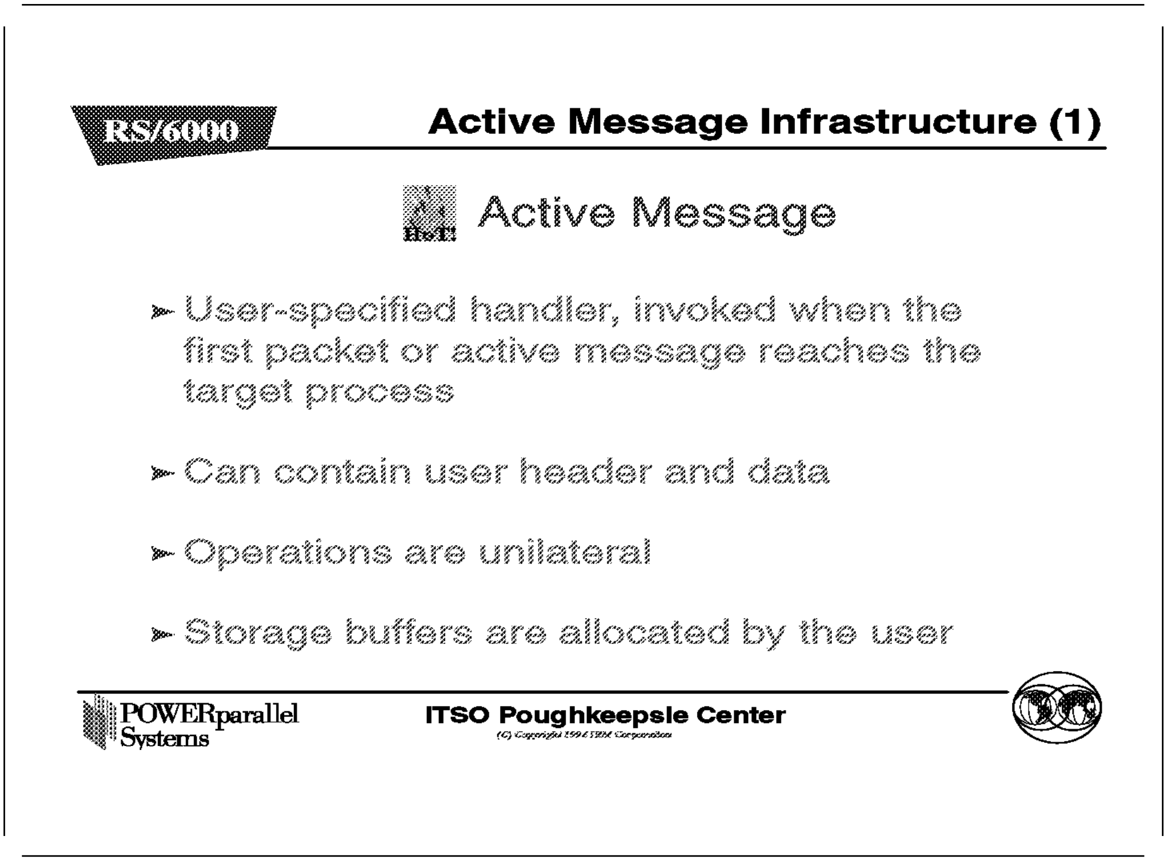
**cmpl_cntr** This is the completion counter. If this counter is used, it is stored at the origin process and is a reflection of tgt_cntr. The completion counter will be incremented at the origin process after tgt_cntr has been incremented at the target process.

With this, we can now define the semantics of the standard and synchronous modes of operation with respect to those counters. Note that the decision to use either mode depends on the programmer, and is enforced by the programmer by checking different combinations of those counters. LAPI does not choose, nor does it enforce, any of the two semantics.

In standard mode, an operation is said to be completed at the origin process when org_cntr has been incremented. Similarly, it is said to be completed at the target process when tgt_cntr has been incremented.

In synchronous mode, an operation is considered to be complete with respect to the origin process when both org_cntr and cmpl_cntr have been incremented. It is considered to be complete with respect to the target process when tgt_cntr has been incremented. The semantic with respect to the target process in synchronous mode is the same as the semantic with respect to the target process in standard mode.

## 10.2.2 Active Message Infrastructure



RS/6000

### Active Message Infrastructure (1)

Active Message

- User-specified handler, invoked when the first packet or active message reaches the target process
- Can contain user header and data
- Operations are unilateral
- Storage buffers are allocated by the user

POWERparallel Systems

ITSO Poughkeepsie Center

(C) Copyright 1996 IBM Corporation

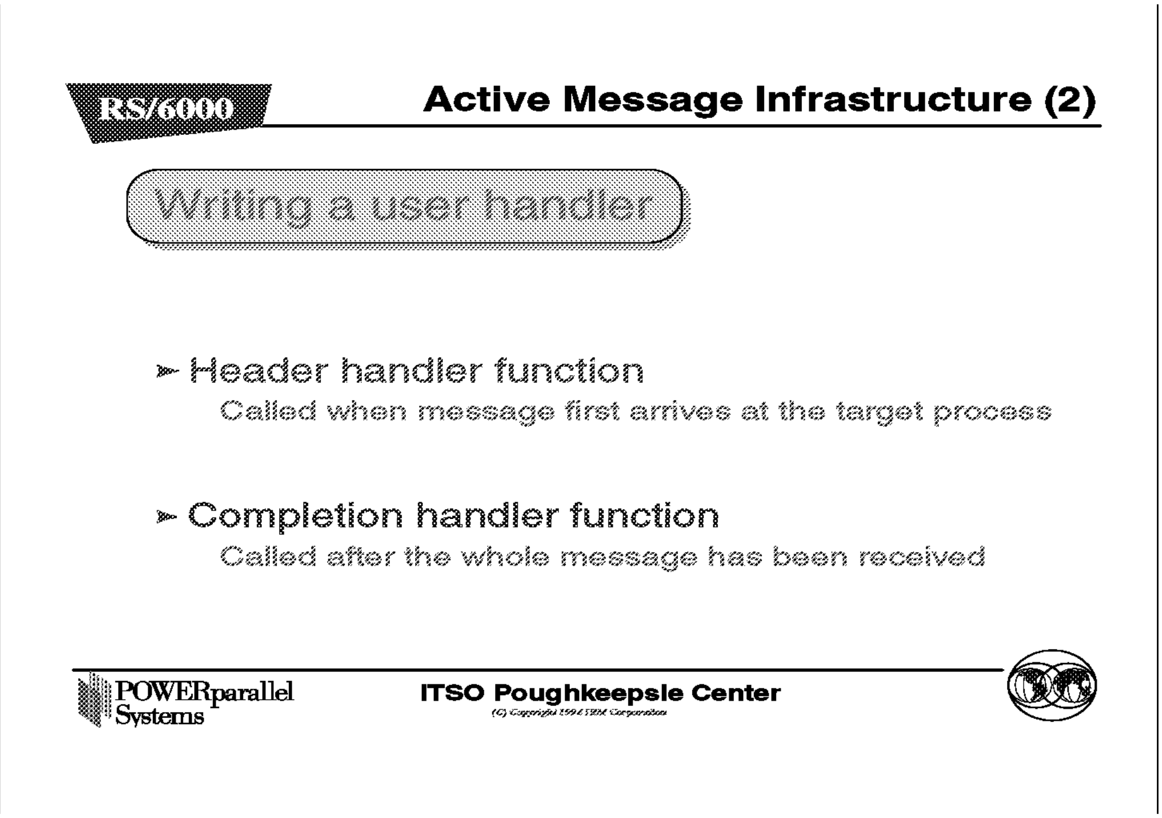
Understanding the Active Message infrastructure is essential to programming in LAPI.

The Active Message function call is a nonblocking call that causes a specified message handler to be invoked and executed in the address space of the target process upon the arrival of the active message. Completion of the operation is signaled if counters are specified.

Optionally, the active message may also bring with it a user header and data from the originating process.

The operation is unilateral in the sense that the target process does not have to take explicit action for the active message to complete.

Buffering is not required because either storage for arriving data (if any) is specified in the active message, or is provided by the invoked handler.



RS/6000

## Active Message Infrastructure (2)

Writing a user handler

- Header handler function
  Called when message first arrives at the target process
- Completion handler function
  Called after the whole message has been received

POWERparallel Systems

ITSO Poughkeepsle Center

(C) Copyright 1994 IBM Corporation

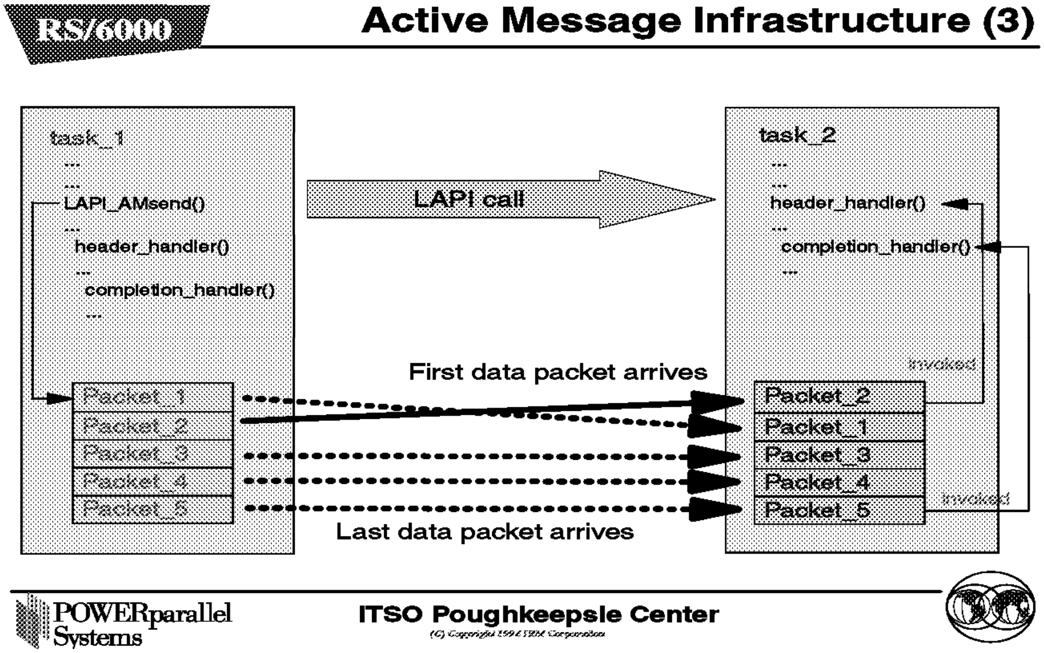
When the active message brings with it data from the originating process, the architecture requires that the handler be written as two separate routines, as follows:

**Function** — **Purpose**

**Header handler**
This is the function that is specified in the active message call. It will be called when the message first arrives at the target process, and provides the LAPI dispatcher (which is the part of the LAPI layer that deals with the arrival of messages and invocation of handlers) with:

- An address where the arriving data must be copied
- The address of the optional completion handler
- The address of an optional user-defined parameter to be passed to the target process

**Completion handler**
This function is called after the whole message has been received. Note that for large messages, the data sent with an active message will arrive in multiple packets. These packets can also arrive out of order.



RS/6000

## Active Message Infrastructure (3)

task_1
...
...
LAPI_AMsend()
...
header_handler()
...
completion_handler()
...

Packet_1
Packet_2
Packet_3
Packet_4
Packet_5

LAPI call

First data packet arrives

Last data packet arrives

task_2
...
...
header_handler()
...
completion_handler()
...

invoked

Packet_2
Packet_1
Packet_3
Packet_4
Packet_5

invoked

POWERparallel Systems

ITSO Poughkeepsle Center

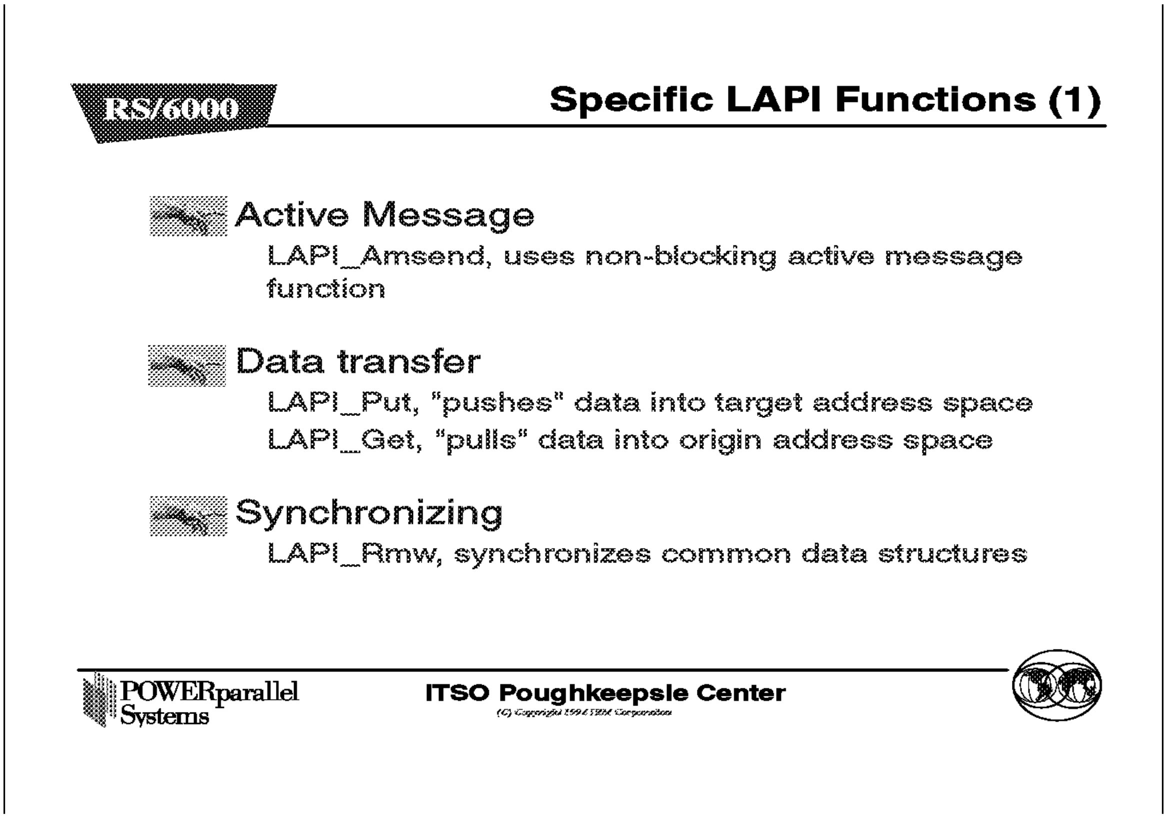
This foil illustrates the relationships between the LAPI_Amsend function call, the header handler, and the optional completion handler.

The completion handler is optionally specified within the header handler which, in turn, is itself one of the parameters of the LAPI_Amsend function call.

When the LAPI_Amsend call is invoked, the message is sent from task_1 to task_2. For the general case, let us assume that the message is large, that is, it takes a few packets to fully transport the message. When the first packet arrives, the LAPI dispatcher at task_2 will invoke the header handler function within the address space of task_2. This is the header handler specified in task_1′s LAPI_Amsend call.

As the packets continue to arrive, when the last packet reaches task_2, the dispatcher at task_2 will invoke the completion handler, which was specified in the header handler. Just as the header handler, the completion handler runs within the address space of task_2.

## 10.3 Using LAPI



RS/6000

### Specific LAPI Functions (1)

- Active Message
  LAPI_Amsend, uses non-blocking active message function
- Data transfer
  LAPI_Put, "pushes" data into target address space
  LAPI_Get, "pulls" data into origin address space
- Synchronizing
  LAPI_Rmw, synchronizes common data structures

POWERparallel Systems

ITSO Poughkeepsle Center

(C) Copyright 1996 IBM Corporation

The following shows some of the LAPI calls and their various purposes:

**Category** **Function**

**Active Message** The function prototype for this call is:

```
int LAPI_Amsend(hndl, tgt, hdr_hdl, uhdr, uhdr_len, udata,
udata_len, tgt_cntr, org_cntr, cmpl_cntr)
```

The active message function (LAPI_Amsend) is a nonblocking call that causes the specific active message to be invoked and executed in the address space of the target process. Completion of the operation is signaled if counters are specified in the call. The LAPI_Amsend function provides three counters (org_cntr, tgt_cntr, and cmpl_cntr), which can be used to provide both standard and synchronous modes. The org_cntr counter is incremented when the origin buffer can be reused, tgt_cntr is incremented when the target buffer can be reused and cmpl_cntr is incremented after the completion handler has completed execution.

**Data transfer** The function prototypes for LAPI_Put and LAPI_Get are:

```
int LAPI_Put(hndl, tgt, len, tgt_addr, org_addr, tgt_cntr,
org_cntr, cmpl_cntr)
```

```
int LAPI_Get(hndl, tgt, len, tgt_addr, org_addr, tgt_cntr,
org_cntr)
```

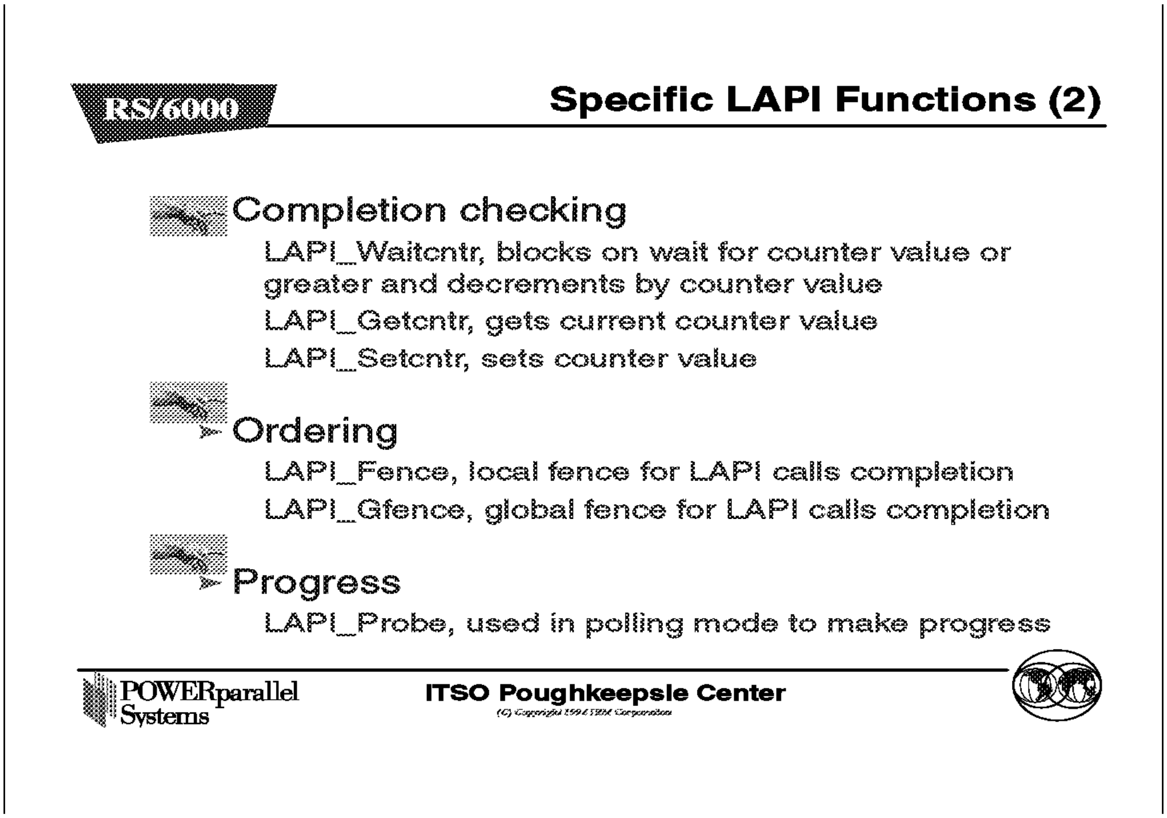
Data transfer functions are nonblocking calls that cause data to be copied from a specified region in the origin address space to the specified region in the target address space (in the case of a LAPI_Put operation), or from a specified region in the target address space to a specified region in the origin address space (in the case of a LAPI_Get operation).

Both standard and synchronous modes are supported by LAPI_Put, but only the synchronous mode is possible in the case of LAPI_Get.

**Synchronizing**

The LAPI_Rmw function is used to synchronize two independent operations, such as two processes sharing a common data structure. The operation is performed at the target process and is atomic, that is, executed to completion and uninterruptable. This operation takes a variable from the origin and performs one of the four selected operations on a variable from the target, and replaces the target variable with the results of the operation. The original value of the target variable is returned to the origin. The four operations are:

- SWAP
- COMPARE_AND_SWAP
- FETCH_AND_ADD
- FETCH_AND_OR



RS/6000

## Specific LAPI Functions (2)

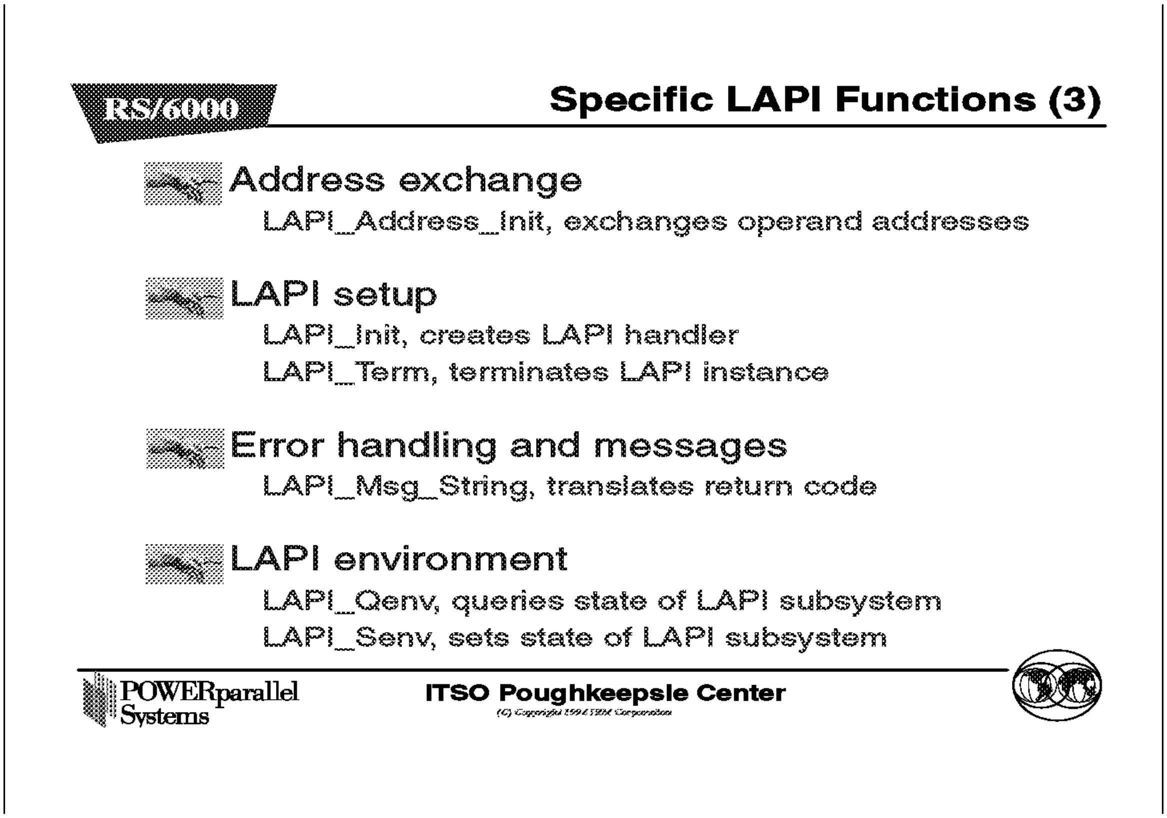
- Completion checking
  - LAPI_Waitcntr, blocks on wait for counter value or greater and decrements by counter value
  - LAPI_Getcntr, gets current counter value
  - LAPI_Setcntr, sets counter value
- Ordering
  - LAPI_Fence, local fence for LAPI calls completion
  - LAPI_Gfence, global fence for LAPI calls completion
- Progress
  - LAPI_Probe, used in polling mode to make progress

POWERparallel Systems

ITSO Poughkeepsle Center

(C) Copyright 1996 IBM Corporation

| **Category** | **Function** |
|---|---|
| **Completion checking** | These functions manipulate the counter values as shown in the figure. |
| **Ordering** | LAPI_Fence and LAPI_Gfence operations provide a fencing capability. LAPI functions initiated prior to these fencing operations are guaranteed to complete with respect to both the origin and target processes before LAPI functions initiated after the fencing operations. LAPI_Fence is a local operation that is used to guarantee that all LAPI functions initiated by the local process are complete. LAPI_Gfence is a collective (global) operation involving all processes in the parallel program. |
| **Progress** | The LAPI_Probe function is used in polling mode to transfer control to the communications subsystem in order to make progress on arriving messages. |



RS/6000

## Specific LAPI Functions (3)

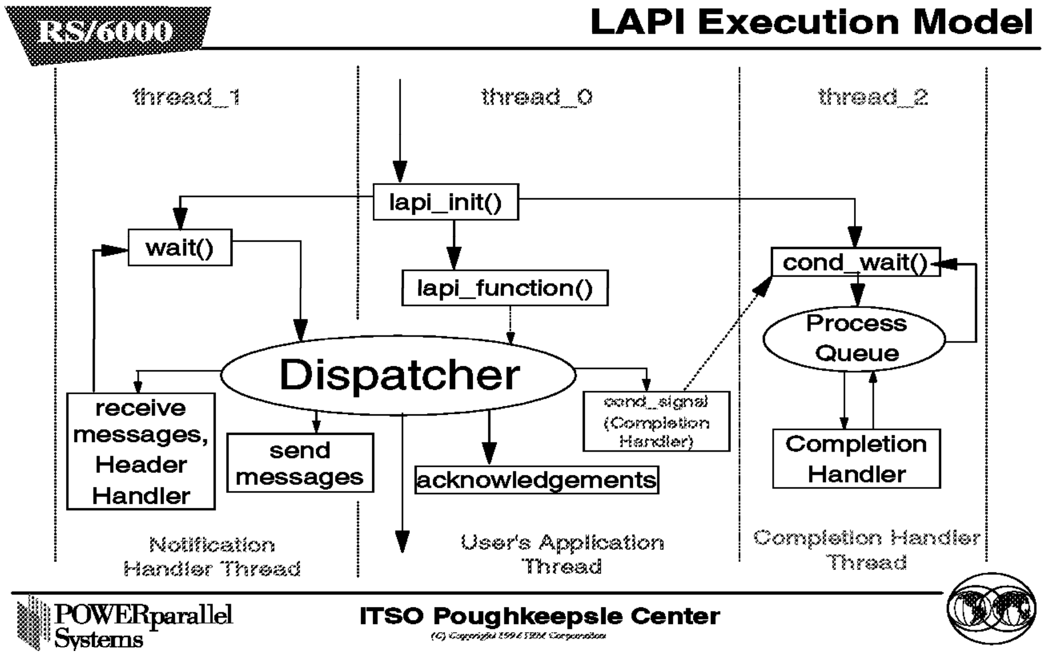
- Address exchange
  - LAPI_Address_Init, exchanges operand addresses
- LAPI setup
  - LAPI_Init, creates LAPI handler
  - LAPI_Term, terminates LAPI instance
- Error handling and messages
  - LAPI_Msg_String, translates return code
- LAPI environment
  - LAPI_Qenv, queries state of LAPI subsystem
  - LAPI_Senv, sets state of LAPI subsystem

POWERparallel Systems

ITSO Poughkeepsle Center

| Category | Function |
|---|---|
| **Address exchange** | The LAPI_Address_Init collective operation allows processes to exchange operand addresses of interest. |
| **LAPI setup** | LAPI_Init and LAPI_Term operations are used to initialize and terminate the communications structures required to effect LAPI communications. |
| **Error handling and messages** | The LAPI_Init function provides a means for the user of LAPI to register an error handler. The LAPI_Msg_String function translates an LAPI call return code value into a message string. |
| **LAPI environment** | The LAPI_Qenv function queries the state of the LAPI communications subsystem, whereas the LAPI_Senv function allows the programmer to specify the value of some of the LAPI communications subsystem′s environment variables.<br>An example is the interrupt state, which is set by specifying INTERRUPT_SET as on (for interrupt mode) or off (for polling mode). The default setting for INTERRUPT_SET is on. |

### 10.3.1 LAPI Execution Model



Once LAPI_Init has been called, two threads in addition to the user′s main thread will be started. One of them, the Notification Handler Thread, sleeps in the kernel. An incoming active message generates an interrupt, which awakens the Notification Handler Thread.

Once woken up, the Notification Handler Thread executes in the user′s address space and invokes the Dispatcher, which does one of the following:

- Receives an incoming message, for example an incoming LAPI_Amsend call. If it is the first physical packet, the Dispatcher will invoke the user-specified header handler, and if it is the last physical packet and there is a user-specified completion handler, the Dispatcher will enqueue the completion handler and wake up the Completion Handler Thread.
- Sends a pending message, for example the user application executed a LAPI_Put.
- Sends an acknowledgement to the origin process, for example to increase the cmpl_cntr at the origin process.

Returning to our example, upon seeing the incoming active message, the Dispatcher will invoke the user-specified header handler. After the header handler has completed, the Notification Handler Thread goes back to sleep.

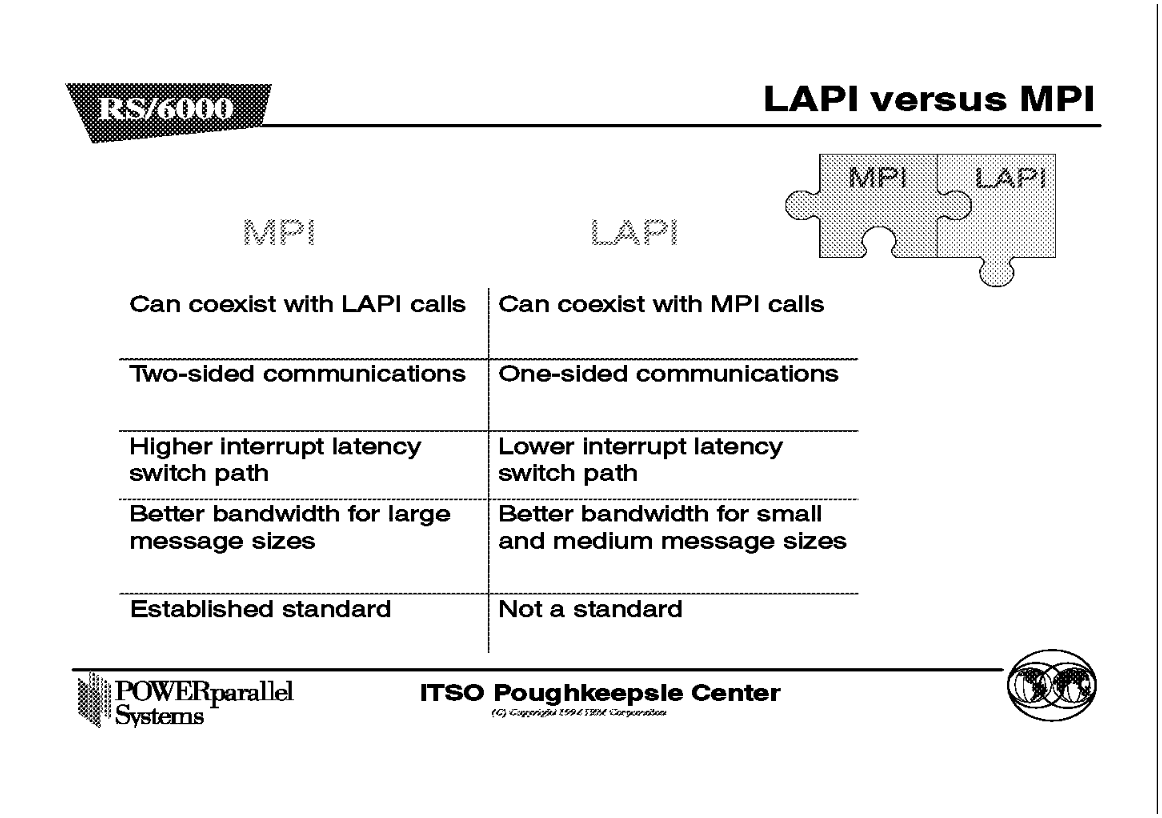
The other thread created by LAPI_Init is the Completion Handler Thread.

If the Dispatcher sees the last packet of the active message arriving, it enqueues the completion handler of the active message, and wakes up the Completion Handler Thread. This thread checks its queue and invokes the enqueued completion handler or handlers.

**Note**

The Dispatcher is controlled by a lock, which enforces that only one Dispatcher is running in a process at any time.

## 10.3.2 LAPI versus MPI


RS/6000

LAPI versus MPI

MPI
LAPI

| MPI | LAPI |
|---|---|
| Can coexist with LAPI calls | Can coexist with MPI calls |
| Two-sided communications | One-sided communications |
| Higher interrupt latency switch path | Lower interrupt latency switch path |
| Better bandwidth for large message sizes | Better bandwidth for small and medium message sizes |
| Established standard | Not a standard |

POWERparallel Systems

ITSO Poughkeepsle Center

(C) Copyright 1996 IBM Corporation

This figure compares MPI and LAPI.

A user program can make MPI and LAPI calls in the same program.

LAPI gives the programmer the ability to do “one-sided communication” (as opposed to MPI, which supports two-sided or collective communication). The one-sided programming model may be a better fit for certain user applications.

LAPI provides a lower-latency path through the switch.

The bandwidth is message size dependent. For small- and medium-sized messages, the LAPI bandwidth is better than that of MPI. For very large messages the MPI bandwidth is slightly better than the LAPI bandwidth.

MPI is a standard interface, whereas LAPI is a nonstandard interface.

# Appendix A. Special Notices

This publication is intended to help IBM customers, Business Partners, IBM System Engineers, and other RS/6000 SP specialists who are involved in Parallel System Support Programs (PSSP) Version 2 Release 3 projects, including the education of RS/6000 SP professionals responsible for installing, configuring, and administering PSSP Version 2 Release 3. The information in this publication is not intended as the specification of any programming interfaces that are provided by Parallel System Support Programs. See the PUBLICATIONS section of the IBM Programming Announcement for PSSP Version 2 Release 3 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
IBM
NetView
Scalable POWERparallel Systems
AIX/6000
LoadLeveler
RS/6000

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

GRF is a trademark of Accent, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

# Appendix B. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 537.

- *PSSP Version 2.2 Technical Presentation*, SG24-4868
- *PSSP Version 2 Technical Presentation*, SG24-4542
- *RS/6000 SMP Servers Architecture*, SG24-2583
- *RS/6000 SP High Availability Infrastructure*, SG24-4838

## B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

| CD-ROM Title | Subscription Number | Collection Kit Number |
|---|---|---|
| System/390 Redbooks Collection | SBOF-7201 | SK2T-2177 |
| Networking and Systems Management Redbooks Collection | SBOF-7370 | SK2T-6022 |
| Transaction Processing and Data Management Redbook | SBOF-7240 | SK2T-8038 |
| AS/400 Redbooks Collection | SBOF-7270 | SK2T-2849 |
| RS/6000 Redbooks Collection (HTML, BkMgr) | SBOF-7230 | SK2T-8040 |
| RS/6000 Redbooks Collection (PostScript) | SBOF-7205 | SK2T-8041 |
| Application Development Redbooks Collection | SBOF-7290 | SK2T-8037 |
| Personal Systems Redbooks Collection | SBOF-7250 | SK2T-8042 |

## B.3 Other Publications

These publications are also relevant as further information sources:

- *PSSP Installation and Migration Guide*, GC23-3898
- *PSSP Diagnosis and Messages Guide*, GC23-3899
- *PSSP Command and Technical Reference*, GC23-3900
- *IBM RS/6000 SP Planning, Volume 1, Hardware and Physical Environment*, GA22-7280
- *IBM RS/6000 SP Planning, Volume 2, Control Workstation and Software Environment*, GA22-7281
- *SP Switch Router Adapter Guide*, GA22-7310

# How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at http://www.redbooks.ibm.com.

## How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER —** to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

  For a list of product area specialists in the ITSO: type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Web Site on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

**Redpieces**

For information so current it is still in the process of being written, look at ″Redpieces″ on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.htm). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

# How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

| | **IBMMAIL** | **Internet** |
|---|---|---|
| In United States: | usib6fpl at ibmmail | usib6fpl@ibmmail.com |
| In Canada: | caibmbkz at ibmmail | lmannix@vnet.ibm.com |
| Outside North America: | dkibmbsh at ibmmail | bookshop@dk.ibm.com |

- **Telephone orders**

| | |
|---|---|
| United States (toll free) | 1-800-879-2755 |
| Canada (toll free) | 1-800-IBM-4YOU |
| | |
| Outside North America | (long distance charges apply) |
| (+45) 4810-1320 - Danish | (+45) 4810-1020 - German |
| (+45) 4810-1420 - Dutch | (+45) 4810-1620 - Italian |
| (+45) 4810-1540 - English | (+45) 4810-1270 - Norwegian |
| (+45) 4810-1670 - Finnish | (+45) 4810-1120 - Spanish |
| (+45) 4810-1220 - French | (+45) 4810-1170 - Swedish |

- **Mail Orders** — send orders to:

IBM Publications
Publications Customer Support
P.O. Box 29570
Raleigh, NC 27626-0570
USA

IBM Publications
144-4th Avenue, S.W.
Calgary, Alberta T2P 3N5
Canada

IBM Direct Services
Sortemosevej 21
DK-3450 Allerød
Denmark

- **Fax** — send orders to:

| | |
|---|---|
| United States (toll free) | 1-800-445-9269 |
| Canada | 1-403-267-4455 |
| Outside North America | (+45) 48 14 2207 (long distance charge) |

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

| | |
|---|---|
| Redbooks Web Site | http://www.redbooks.ibm.com |
| IBM Direct Publications Catalog | http://www.elink.ibmlink.ibm.com/pbl/pbl |

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

**Redpieces**

For information so current it is still in the process of being written, look at ″Redpieces″ on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.htm). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

## IBM Redbook Order Form

**Please send me the following:**

| **Title** | **Order Number** | **Quantity** |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number

- Credit card number

Credit card expiration date Card issued to Signature

**We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.**

# List of Abbreviations

| | |
|---|---|
| ***ARP*** | Address Resolution Protocol |
| ***MIB*** | Management Information Base |
| ***MAC*** | Medium Access Control |
| ***IP*** | Interface Protocol |
| ***IPAT*** | IP Address Takeover |
| ***HWAT*** | Hardware Address Takeover |
| ***CWS*** | Control Workstation |
| ***IBM*** | International Business Machines Corporation |
| ***ITSO*** | International Technical Support Organization |
| ***ATM*** | Asynchronous Transfer Mode |
| ***FDDI*** | Fiber Distributed Data Interface (100Mbit/s fiber optic LAN) |
| ***SP*** | IBM RS/6000 Scalable POWERparallel System (RS/6000 SP) |
| ***TCP*** | Transmission Control Protocol |
| ***TCP/IP*** | Transmission Control Protocol/Internet Protocol |
| ***Clinfo*** | Client Information Program |
| ***IPAT*** | IP-address takeover |
| ***UDP*** | User Datagram Protocol |
| ***LAN*** | Local Area Network |
| ***LVM*** | Logical Volume Manager |
| ***HACMP*** | High Availability Cluster Multi-Processing |
| ***HANFS*** | High Availability Network File System |
| ***HACMP ES*** | High Availability Cluster Multi-Processing Enhanced Scalability |
| ***PTF*** | Program Temporary FIX |
| ***PSSP*** | Parallel System Support Program |
| ***PTPE*** | Performance Toolbox Parallel Extension |
| ***VSD*** | Virtual Shared Disks |
| ***RVSD*** | Recoverable Virtual Shared Disks |
| ***AIX*** | Advanced Interactive Executive |
| ***NFS*** | Network File System (USA, Sun Microsystems Inc.) |
| ***GPFS*** | General Parallel File System |
| ***C-SPOC*** | Cluster Single Point of Control |
| ***SMIT*** | System Management Interface Tool |
| ***NIM*** | Network Interface Module |
| ***DARE*** | Dynamic Automatic Reconfiguration Events |
| ***VSM*** | Visual System Management |
| ***HPS*** | High Performance Switch |
| ***ODM*** | Object Data Manager |
| ***SDR*** | System Data Repository |
| ***SNMP*** | Simple Network Management Protocol |
| ***CPU*** | Central Processing Unit |
| ***EMAPI*** | Event Management Application Programming Interface |
| ***PTX/6000*** | Performance Toolbox/6000 |
| ***LPP*** | Licensed Program Product |
| ***SMUXD*** | SNMP Multiplexor Daemon |
| ***SBS*** | Structured Byte String |
| ***DNS*** | Domain Name Server |
| ***ADSM*** | ADSTAR Distributed Storage Manager |
| ***SCSI*** | Small Computer System Interface |

# Index

# ITSO Redbook Evaluation

Technical Presentation for PSSP Version 2.3
SG24-2080-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@vnet.ibm.com

**Please rate your overall satisfaction** with this book using the scale:
**(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

**Overall Satisfaction** ____________

**Please answer the following questions:**

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

______________________________________________________________________________

______________________________________________________________________________

______________________________________________________________________________

______________________________________________________________________________

What other redbooks would you like to see published?

______________________________________________________________________________

______________________________________________________________________________

______________________________________________________________________________

**Comments/Suggestions: ( THANK YOU FOR YOUR FEEDBACK! )**

______________________________________________________________________________

______________________________________________________________________________

______________________________________________________________________________

______________________________________________________________________________

______________________________________________________________________________

Printed in U.S.A.

SG24-2080-00