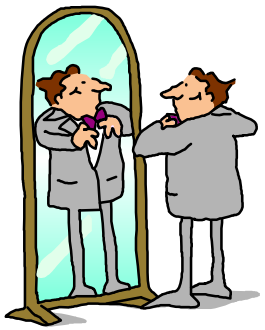
High Availability Architectures

Bruce Spencer

baspence@us.ibm.com

2/10/2001



Limitations and Disclaimers

- **Goal: High Availability Architectures**
  - How to organize systems for HA
  - Not discussed
    - System features (ie, Raid Disk, ECC Memory, JFS)
    - Non system issues: Network, Environment, Applications
- **Guidelines**
  - There are no "hard and fast" rules
  - Each situation is unique
- **Standard Disclaimers**
  - "Opinions expressed here are not necessarily those of my employer, but ought to be."
  - "Your mileage may vary"

Best Practices
Three P's of High Availability

The purpose of the next chart is to show that 72% of downtime is caused by **non-system** factors, such as planned maintenance, application bugs, operator error. To protect against these factors, you must consider people, processes and products, the "**Three P's of High Availability**."

Although this presentation focuses on HA architectures, it can't be stressed enough that architecture is irrelevant without first addressing the "People" and "Process" aspects

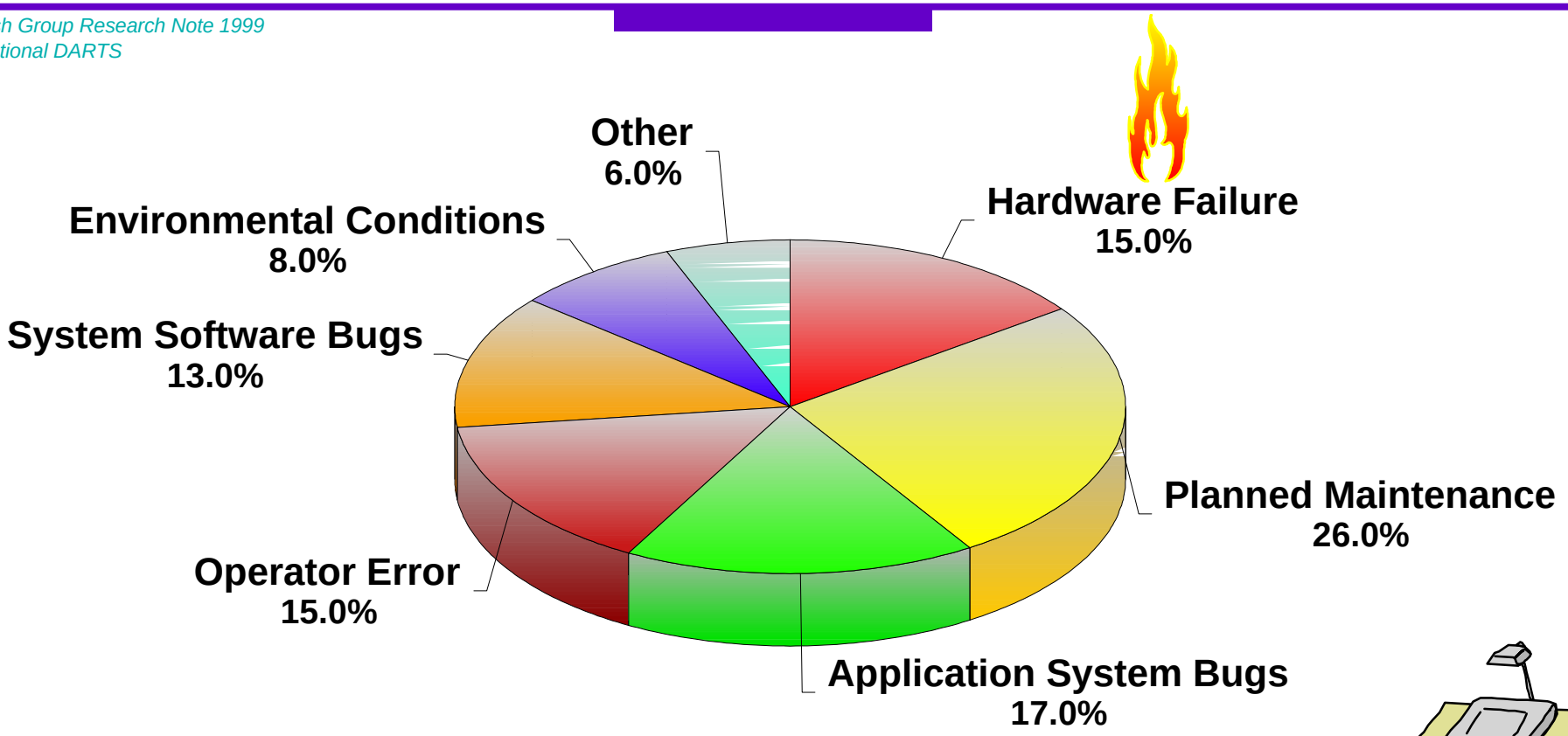
People
- Training
- Documentation
- Responsibility
- Authorization

Process
- Backup/Recovery
- Change Control
- Performance Mgt
- Problem Mgt
- Security
- Monitor/Track

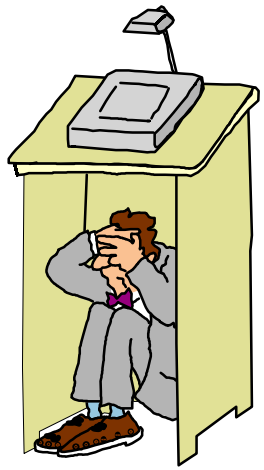
Product
- Application
- Platform Selection
- Architecture
- HA Middleware

Sources of Downtime

rce: Standish Group Research Note 1999
International DARTS



Total
100%

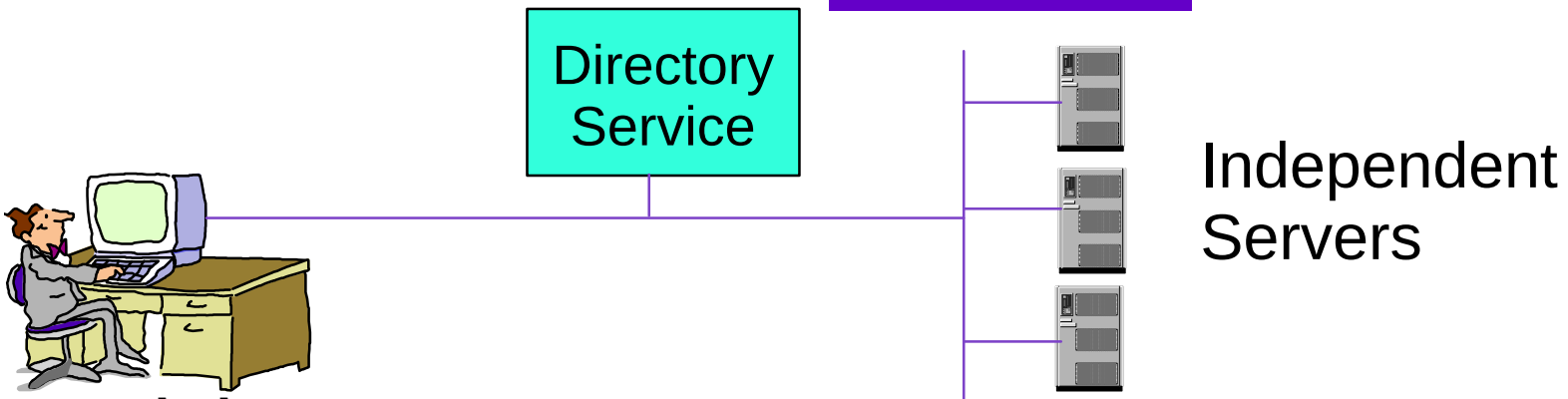


High Availability Techniques

- User Segmentation
- Data Replication
- Client Server
- Load Leveling
- Hot Spare Server
- Alternative Paths

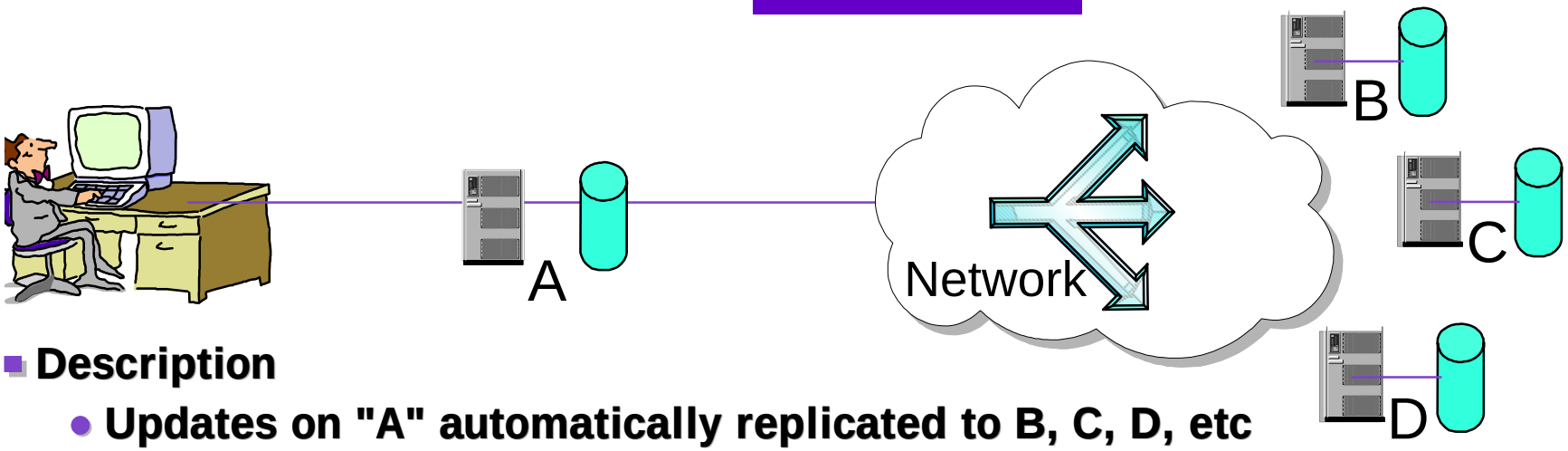
Here are a few techinques I've seen at implemented various locations.

User Segmentation



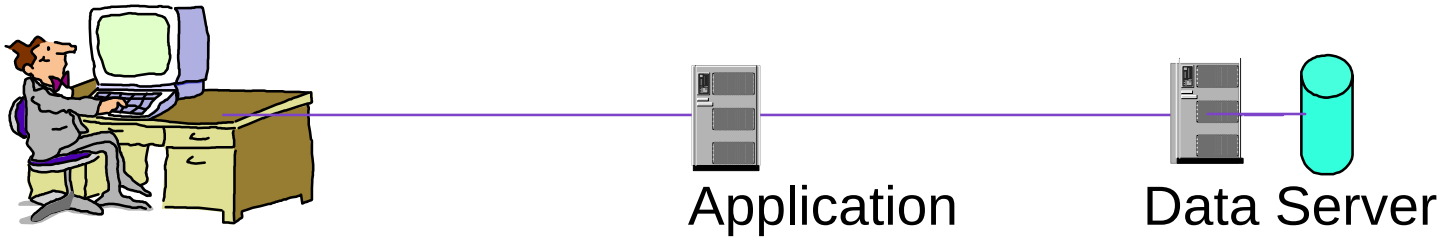
- **Description**
  - **Each user has a "home" server, where their applications and data reside.**
  - **Directory Service points user to their server home (transparent to user)**
  - **Example: Mail Exchanges**
- **Pro's**
  - **Outage limited to "1/n" users (n=number of servers)**
  - **Scaleability**
  - **Unlimited horizontal growth**
- **Con's**
  - **Imperfect load balancing (must analyze and rebalance periodically)**
  - **A server outage will affect some users, but not all**
  - **Data sharing**

Data Replication



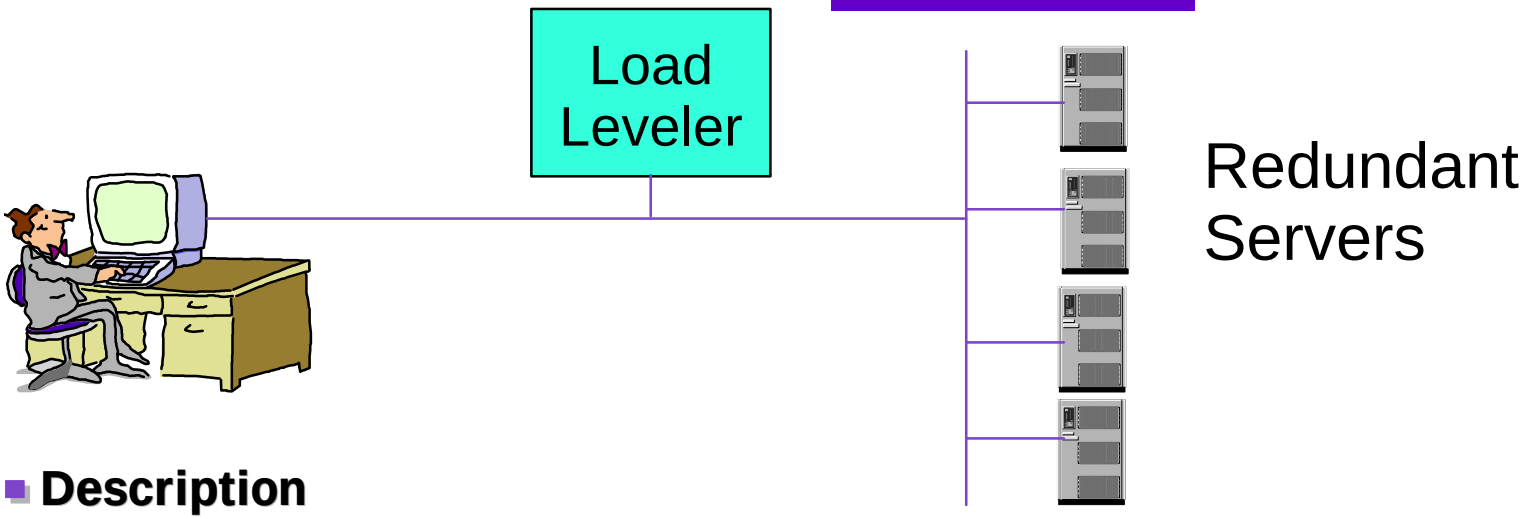
- **Description**
  - **Updates on "A" automatically replicated to B, C, D, etc**
  - **"Any to any" replication is not generally used due to difficulty in maintaining data integrity.**
  - **Usually configured with one primary server and one or more secondary servers. All updates start at primary. Primary replicates to secondary(s). Secondary servers available for Read-Only.**
- **Pro's**
  - **Protected against up to "n-1" server failures**
- **Con's**
  - **Response time (significant delay in 2 phase commit process)**
  - **Sympathy hangs (primary can't determine if secondary slow or down)**
  - **Affected user must reconnect to another server**

Client Server



- **Description**
  - **Protects against database outage**
  - **If database is down, updates are queued on application server(s)**
  - **Queued updates sent to data server when it is back on-line**
- **Pro's**
  - **Higher scalability by separating application from database**
  - **Database failure can be transparent to users.**
- **Con's**
  - **Application must be written to support Client Server**

Load Leveling



- **Description**
  - **"Load Leveler" distributes users evenly across servers**
  - **"Load Leveler" can sense server outage, detours around**
  - **Typically used with Application servers in Client Server**
- **Pro's**
  - **Unlimited Horizontal Growth**
  - **Outage limited to "1/n" users (n=servers)**
- **Con's**
  - **Imperfect Load Balancing (the "AOL" effect)**
  - **Affected user must reconnect**

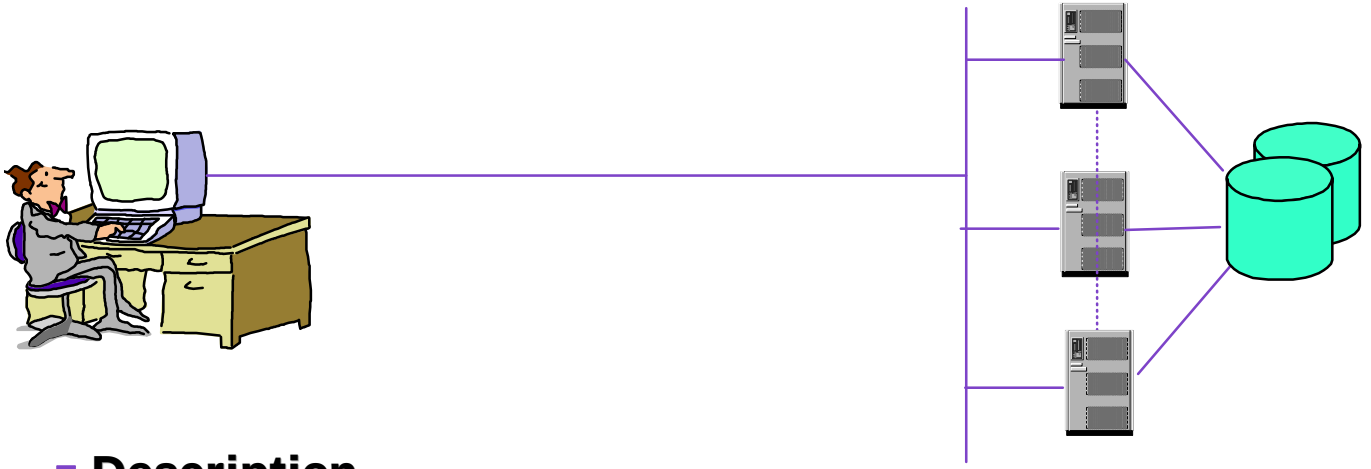
"AOL Effect": Loadlevelers are often modified DNS name servers that distribute load by handing out different IP addresses for each request. However, large ISP's like AOL cache IP addresses, resulting in congestion on one server.

Hot Backup

- Description
  - If the primary server fails, "hot backup" takes over.
  - Typically used with database servers
- Pro's
  - Avoids data synchronization issues as with replication
- Con's
  - Outage during failover (<5 minutes with HACMP)
  - "In flight" transactions lost
  - Higher complexity
  - Added cost of "hot spare" server

Hot backup tend to be the prefered database HA method, despite the list of "con's"

Special Case: Oracle OPS



Redundant Servers, Shared Disk

- **Description**
  - **All Oracle servers have concurrent R/W access to shared disks**
  - **Client connections spread evenly across servers**
  - **If a server fails, affected clients reconnect to an available server**
- **Pro's**
  - **Synchronized data**
  - **Failure limited to "1/n" users (n=servers)**
- **Con's**
  - **Scaleability (Data Link Manager bottleneck)**
  - **Complexity, skill level**

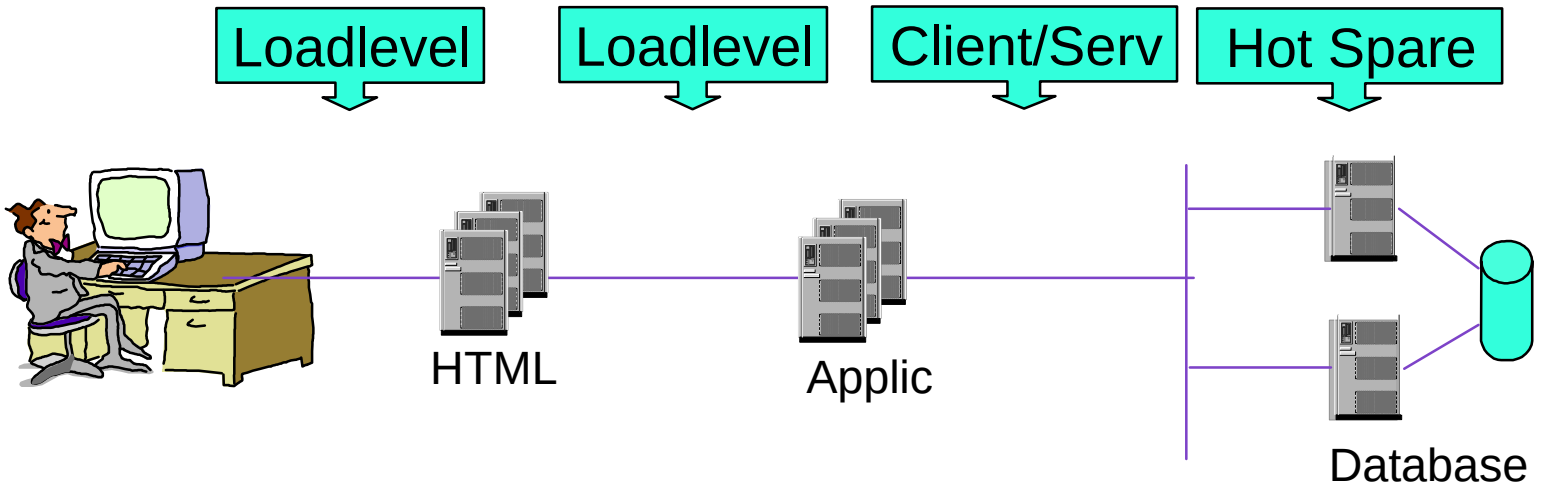
Alternate Paths

- **Have a contingency plan**
- **A web reservation system I'm familar with has the following access paths:**
  1. **Web access**
  2. **Voice Response Unit**
  3. **Fax**
  4. **Operator**

Putting it Together

Hypothetical Example

Combining HA Techniques to Build a Highly Available Web Site



Backup Charts

High Availability Justification: Typical Cost of Downtime

- **Most users experience 66% more downtime than expected**

- **Average cost of mission critical downtime $10k/min**

Application	Cost/Minute
Call Location	$27,000
Number Portability	$14,400
ERP	$13,000
Supply Chain Management	$11,000
Electronic Commerce	$10,000
Internet Banking	$7,000
Universal Personal Services	$6,000
Customer Service Center	$3,700
ATM/POS/EFT	$3,500
Messaging	$1,000

- **"72% of mission critical applications experience nine hours of downtime per year: Most companies are under-investing in high availability technology"**

Source:
Standish Group Research Note 1998
Penny Wise & Pound Foolish

Selecting an HA Architecture

- **Focus on Business Requirements**
  - Application Requirements
  - Service Level Agreement
  - Data Characteristics
- **What Are You Protecting Against?**
- **Product Selection**
  - Application (Built-in HA Support)
  - HW/SW Reliability
  - HA Middleware
- **Cost Benefit Analysis**
  - There Are Degrees of High Availability
  - Higher Availability/Higher Cost

Guideline: the prefered product selection is an application with high availability designed into it.

Most server HA implementations tend to be an "after thought".

Differences in Availability by Platform

Platform	Outages/Server/Year	Availability
S390 (sysplexed)	10 minutes	99.998%
Tandem	1.7 hours	99.98
AS/400	5.2 hours	99.94
S390 (non-sysplex)	8.9 hours	99.90
VAX	18.9 hours	99.78
Unix (all)	23.6 hours	99.73
NT	224.5 hours	97.44

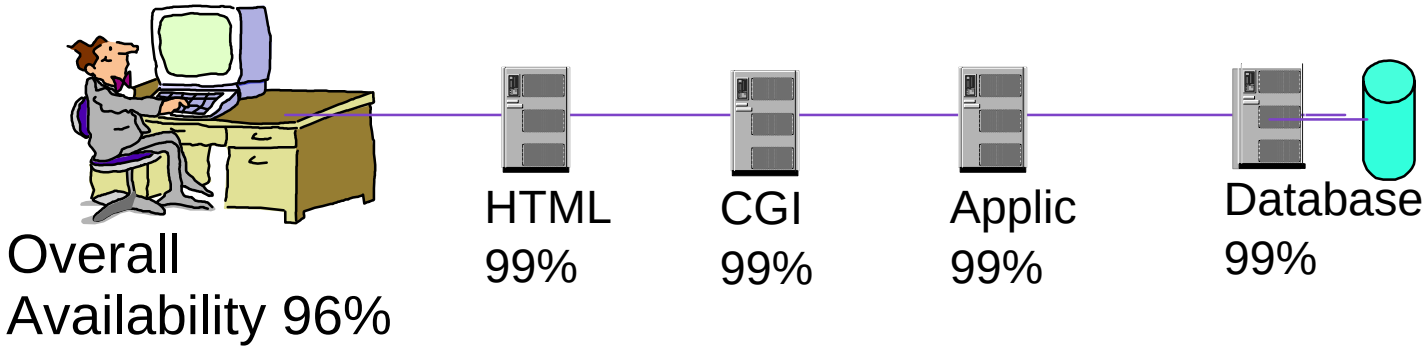
Gartner Report: 10/98

Data Characteristics that Affect HA Architecture

- **Business Value of Data**
- **Processing Requirement**
  - **Real Time vs Batch**
  - **Date Warehouse vs Transaction Processing**
- **Persistence**
  - **Time to Live (ie a current stock quote vs transaction history)**
- **Volatility**
  - **Data Rates**
  - **Read Only vs Read Write**

Guideline: a good practice is to separate data types. For example, mixing critical and non-critical data requires both to be mananaged as critical resources. This adds costs and overhead.

Architecture Recommendation: Reduce Application Path Length



- **In general, the shorter the application path, the higher the availability**
- **Overall Availability = (Avail of Layer 1) x (Avail Layer 2) x…(Avail Layer n)**
- **Hypothetical web site example:**
  - **4 Layers**
  - **Each server is 99% available**
  - **Overall availability is 96%**

Resources

- **Redbooks**
  - http://www.redbook.ibm.com (search "high availability")
- **HACMP Manuals**
  - http://www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixgen/hacmp_index.html