Migrating to HACMP/6000 2.1

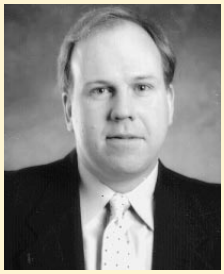
By Thomas Casey and Robert Metcalf

HACMP/6000 Release 2.1 became generally available in December 1993. This article describes the steps and issues involved in migrating an existing HACMP/6000 cluster from Release 1.2 to Release 2.1. In addition, the article provides background for system administrators who are evaluating extending a two-node cluster to three or four nodes.

Building on the strengths of RISC System/6000 servers and the AIX operating system, High Availability Cluster Multi-Processing/6000 (HACMP/6000) provides a set of services that guarantee quick recovery if a critical component fails. HACMP/6000 is designed for database and transaction-processing applications that require highly available, scalable configurations.

Release 2.1 extends HACMP/6000 with features that increase flexibility in both cluster configuration and fallover (fallover occurs when the HACMP/6000 software detects a node failure and reconfigures the cluster to compensate), and make HACMP/6000 easier to configure, customize, and troubleshoot. Sites that have installed an earlier version of the HACMP/6000 software will want to upgrade to Release 2.1 to take advantage of the added functionality provided by this new release:

- Support for four-way clusters
- A new flexible, extensible cluster configuration methodology
- Simplified, centralized cluster configuration
- Cluster verification tool
- Cluster diagnostic tool
- New event customization facility
- More enhancements to the base product



Thomas Casey

Robert Metcalf

Transitioning to Release 2.1 — A Conceptual Background

Release 2.1 includes several new features that change the process of configuring an HACMP/6000 cluster. This section identifies these features and helps system administrators understand what is necessary to migrate an existing cluster to Release 2.1.

Resource-Based Clusters

In earlier releases of HACMP/6000, a cluster could have two active nodes. Those releases provided a set of predefined configurations—such as hot standby or one-sided takeover—that encompassed a range of high-availability solutions. In Release 2.1, a cluster can have up to four active nodes, which dramatically increases the number of possible cluster configurations. Therefore, a new way of configuring a cluster was developed.

Release 2.1 does not provide a set of predefined configurations. It uses a flexible configuration methodology that associates resources with a node or IP address. This allows cluster members to define which resources they own and which they take over from peer nodes.

Types of HACMP/6000 Resources

A focus on resource ownership allows for many cluster configurations and lays the foundation for configuring four-way clusters and more. Release 2.1 defines three types of resources:

- **Owned resources**: Owning a resource denotes a direct relationship between a single node and a resource. When the owning node is active in the cluster, the identified resource

is owned by that node. In concurrent access configurations, a single-disk resource can be owned by several nodes.

- **Takeover resources**: A takeover resource binds to a designated peer node when the owning node detaches from the cluster. Taking over a resource denotes a secondary relationship, active only if the owner node is not available.
- **Rotating resources**: A rotating resource is associated with a specific IP address. It is owned by the node currently assuming that IP address as long as that node has the IP address. If that node detaches from the cluster, then the node assuming the designated IP address becomes the owner of the resource.

Example of a Resource-Based Cluster

The new resource-based methodology is best illustrated through an example. Release 1.2 supports a hot-standby configuration in which a server node provides highly available services and a standby node does no processing—waiting idly for the server to fail. In Release 2.1, there is no hot-standby configuration per se. The relationship among the nodes in a cluster is implicit in the resources defined for the nodes. Figure 1 shows a two-node hot-standby cluster after it has been converted to the resource-based model.

In this setup, a single node owns all highly available resources. This node is the *owner node* (called the server node in Release 1.2). Although the second node owns no resources, the highly available resources are defined to this node as takeover resources. This node, called the *takeover node* or *standby node*, stands idle, waiting for the owner node to detach from the cluster. If the owner node detaches, the takeover node assumes control of the resources owned by the owner node, restarts the applications, and services clients. The takeover node remains active until the owner node rejoins the cluster. At that point, the takeover node releases the highly available resources and returns to an idle state.

This configuration could also be set up using rotating resources—the only difference would be that when the node that had originally detached from the cluster returns, it becomes the standby.

The benefit of this flexible resource-based methodology will become apparent when extending a cluster to three or four nodes (discussed later in the section "Extending HACMP/6000 Beyond Two Nodes").

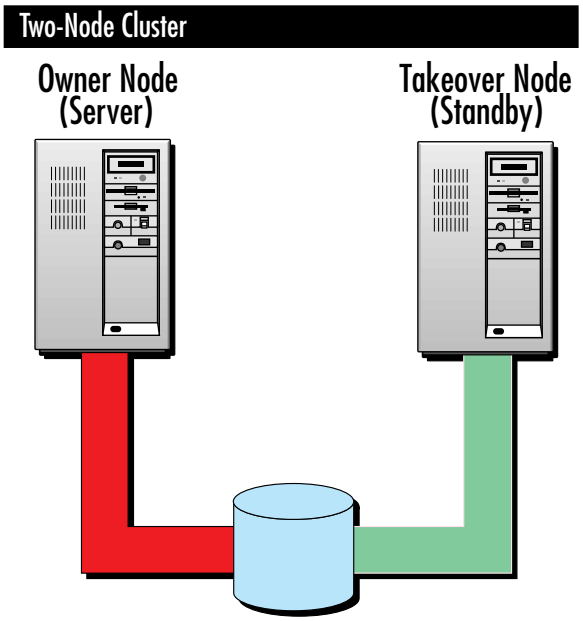


Figure 1. A two-node resource-based cluster

Events

Release 2.1 includes redesigned cluster configuration scripts based on the concept of a *cluster event*. This event represents a change within the cluster that invokes a response from the Cluster Manager (the daemon that monitors cluster status). An example of an event is a `node_down`, where a node detaches from the cluster. An event, in turn, can consist of a series of *subevents*. For example,

Events	Subevents
config_too_long	acquire_service_addr
fail_standby	acquire_takeover_addr
join_standby	get_disk_vg_fs
network_down	node_down_local
network_down_complete	node_down_local_complete
network_up	node_down_remote
network_up_complete	node_down_remote_complete
node_down	node_up_local
node_down_complete	node_up_local_complete
node_up	node_up_remote
node_up_complete	node_up_remote_complete
swap_adapter	release_service_addr
swap_adapter_complete	release_takeover_addr
unstable_too_long	release_vg_fs
	start_server
	stop_server

Figure 2. Cluster events and subevents

a node_down event could include the stop_servers, release_service_addr, and release_vg_fs subevents. For each event and subevent, HACMP/6000 provides a corresponding script of the same name in the /usr/sbin/cluster/samples directory. Figure 2 lists the new 2.1 files in that directory.

Event Scripts Replace topchng.rc and netchng.rc Scripts

The event and subevent scripts replace the topchng.rc and netchng.rc configuration scripts as the Cluster Manager tool for responding to changes in cluster status. If either the topchng.rc or the netchng.rc script has been modified, the system administrator must use the event customization facility provided with Release 2.1 to re-create the necessary functionality.

The /usr/sbin/cluster/clinfo.rc script remains virtually the same as previous versions in both function and intent. It is neither an event nor a subevent—it is a script that runs on HACMP/6000 clients to flush a stale Address Resolution Protocol (ARP) cache.

Extending HACMP/6000 clusters provides a graceful, incremental growth path for mission-critical applications.

Event Customization Facility

Release 2.1 provides a System Management Interface Tool (SMIT) interface that system administrators can use to customize event processing without modifying the event scripts distributed with the product. Since future updates may require that an event script be replaced, it is best to use local customization outside the event script when applying updates.

Using the event customization facility, the system administrator can specify custom commands or scripts to tailor event processing for a site, including:

- Pre-event and post-event commands
- Event notification
- Event recovery and retry

Application Servers Replace node.servers Script

In earlier versions of HACMP/6000, highly available applications were started from the node.servers script, which was called by the topchng.rc configuration script after a change in cluster status. Release 2.1 introduces the concept of *application servers* and views applications as resources, similar to a filesystem or IP address. Release 2.1 provides a SMIT interface for configuring applications made highly available by the cluster, eliminating the need to edit the node.servers script. Configuring an application server associates a meaningful name with a server application, and points the cluster event scripts to the application server's start and stop scripts.

Extending HACMP/6000 Beyond Two Nodes

Release 2.1 supports three- and four-way clusters in both concurrent and nonconcurrent access environments. System administrators should examine the benefits of three- and four-node clusters to determine whether it is beneficial to extend the cluster. Three- and four-way clusters provide the following benefits to HACMP/6000 sites.

- **Scalability**: Extending HACMP/6000 clusters to four nodes provides a graceful, incremental growth path for mission-critical applications by allowing the workload to be spread among multiple processors sharing common disk resources. It also allows users to execute and store data on multiple systems with minimal changes to the application.
- **Increased performance**: A cluster's ability to provide high performance is directly related to its ability to scale. Clustering provides a form of parallel processing that splits the workload among more processing resources without adding excessive overhead.
- **Enhanced availability**: As the number of nodes increases, the overall level of availability provided by the cluster also increases.

Before extending a cluster to three or four nodes, system administrators should evaluate the following disk and network considerations.

A four-way HACMP/6000 cluster can use industry-standard Small Computer Systems Interface-2 (SCSI-2) differential disk devices, including the IBM 7135-110 RAIDiant disk array, or the new IBM 9333-011 and IBM 9333-501 serial disk subsystems. SCSI-2 differential devices are supported only in nonconcurrent configurations, while the IBM 7135-110 RAIDiant disk array and the 9333-011 and 9333-501 serial disk subsystems are supported in both concurrent and nonconcurrent configurations.

Eight initiators or targets can connect to a SCSI-2 differential bus that is terminated on the bus by Y cables. The most likely configuration for IBM 7203 SCSI disks and IBM 9334 devices in a four-way cluster is separate dual-ported chains. Each disk and adapter on the same SCSI bus requires a unique SCSI address. The 7135-110 RAIDiant disk array can support four cluster

Four-Node Configuration

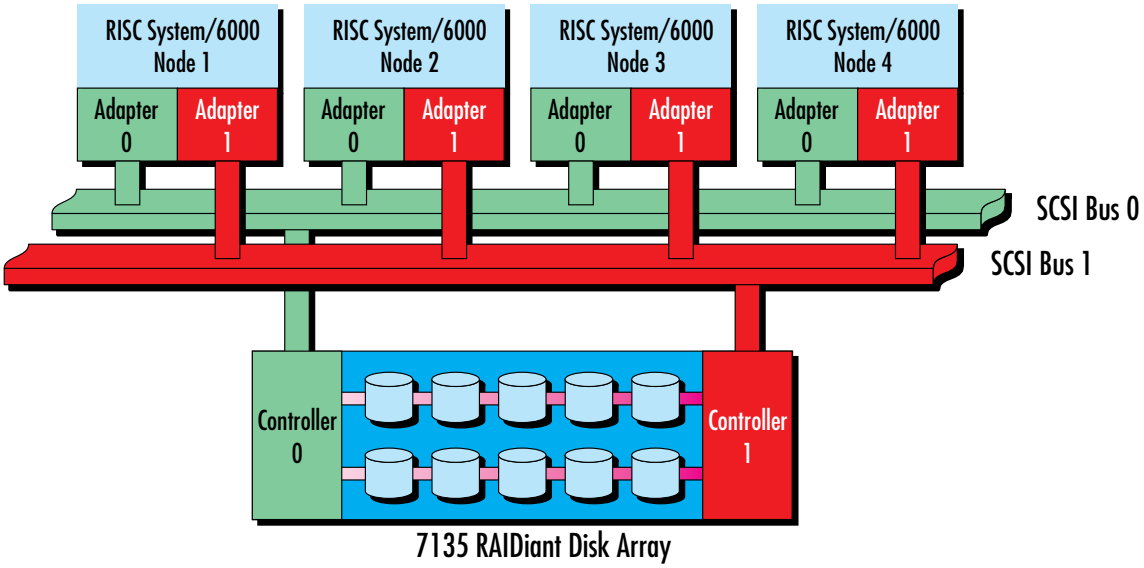


Figure 3. Four-node RAIDiant configuration

nodes directly connected to it. Typically, the 7135-110 RAIDiant disk array is configured with two controllers, each on its own SCSI bus, as shown in Figure 3. Four cluster nodes can attach to a single 9333-011 or 9333-501 disk subsystem.

As the number of nodes increases, the amount of communication between nodes increases as well. This is especially true in concurrent access environments, which generate large volumes of lock manager traffic. When extending a cluster to three or four nodes, it is beneficial to add a high-speed network such as a Fiber-optic Distributed Data Interchange (FDDI) for internode communication. This prevents traffic from becoming a bottleneck for clients trying to reach the cluster. At the very least, lock manager traffic should be routed over a separate, private network.

Migrating to Release 2.1

This section describes the tasks that system administrators must complete to migrate an existing HACMP cluster from Release 1.2 to 2.1. These tasks include the following:

- Using the `clconvert` utility to convert cluster and node configuration information into the appropriate 2.1 Object Data Manager (ODM) object classes, and to redefine the cluster configuration from the 1.2 role-based model to the 2.1 resource-based model
- Using SMIT to extend the cluster to three or four nodes (if desired)
- Using the event customization facility to re-create any custom processing added to the `topchng.rc` or `netchng.rc` configuration scripts distributed with Release 1.2
- Using the SMIT interface (which replaces the `node.servers` file) for application servers to register the start and stop scripts for applications to be made highly available by HACMP/6000
- Populating the `/usr/sbin/cluster/clhosts` file

Preparations for Upgrading to Release 2.1

Before upgrading a Release 1.2 installation to Release 2.1, the system administrator must complete the following steps:

1. For each node, archive the `/usr/sbin/cluster` directory to a readily accessible place on disk so that it is easy to retrieve and compare localized script and configuration files. Also, capture and store the output from the `/usr/sbin/cluster/clgetenv` command as a snapshot of the node's view of its HACMP/6000 environment.
2. If the 1.2 installation is applied but not committed, commit it so that 2.1 can be installed over 1.2.

Clustering splits the workload among more processing resources without adding excessive overhead.

3. Do a `mksysb` of each node. If the site is running the HACMP/6000-supplied Concurrent Logical Volume Manager (CLVM) software, see the special instructions in the *HACMP/6000 2.1 Release Notes* about `mksysb` and CLVM.
4. If running the CLVM, revert to the standard AIX LVM before upgrading.

The Upgrade Process

After completing the prerequisite tasks described above, the system administrator can upgrade the 1.2 environment to 2.1. During the upgrade, HACMP/6000 runs the new `/usr/lpp/cluster/tools/clconvert` utility, which performs the following functions:

- Converts cluster and node configuration information into the appropriate 2.1 ODM object classes
- Redefines the cluster configuration from the 1.2 role-based model to the 2.1 resource-based model

Once the decision has been made to extend the cluster, adding the actual nodes is simple.

Conversion Tool

Installing Release 2.1 over a committed 1.2 installation causes an automatic update of the 1.2 configuration information to 2.1 format. The `/usr/sbin/cluster/cluster.cf` and `/etc/objrepos/hacmp6000` (node information) files are used to populate the appropriate 2.1 object classes. System administrators can back this out using the standard SMIT interface to modify or delete configuration information.

The conversion fails if either the `/usr/sbin/cluster/cluster.cf` or `/etc/objrepos/hacmp6000` files are not available. If there is already configuration information in one of the Release 2.1 ODM object classes, it is overwritten during the installation.

If the original `cluster.cf` file is correct and the `/etc/objrepos/hacmp6000` file exists, the new ODM classes are updated. The following output should appear in the `smit.log` and on the console:

```
Adding new HACMPcommand ODM entries.
Adding new HACMPevent ODM entries.
clconvert successful.
```

This can be verified by running the `cllsif` command for cluster topology information and the `clgetres -A` command for node information.

Note: To prevent the conversion tool from being run automatically, the system administrator can rename the `/usr/sbin/cluster/cluster.cf` and `/etc/objrepos/hacmp6000` files before installing Release 2.1.

Moving Configuration Information to the ODM

In earlier versions of the HACMP/6000 software, cluster configuration information was stored in the `/usr/sbin/cluster/cluster.cf` file, a stanza-based ASCII file. The easiest way to access this file was through the `/usr/sbin/cluster/cllsif` command. Changes were made to the file using the `smit hacmp` fastpath. Node information was accessible through the `/usr/sbin/cluster/clgetenv` command, which examined the `/etc/objrepos/hacmp6000` file.

HACMP/6000 2.1 moves this configuration information into the ODM, creating the following seven new object classes in the `/etc/objrepos` directory:

```
HACMPadapter
HACMPcluster
HACMPcommand
HACMPevent
HACMPnetwork
HACMPresource
HACMPserver
```

Files Saved During the Upgrade

The system administrator can now install the new release, as described in Release 2.1 documentation. The following files are saved, with an `.OLD` extension:

```
clinfo.rc.OLD
cluster.cf.OLD
netchng.rc.OLD
node.servers.OLD
node.vars.OLD
topchng.rc.OLD
```

Any customized subdirectories (such as `/usr/sbin/cluster/local`) are left untouched by the installation procedure. The system administrator uses the content of the old scripts to customize the 2.1 environment. It is especially important to review the `topchng.rc`, `netchng.rc`, and `node.servers` scripts, which may have been customized at the site for Release 1.2.

Extending the Cluster to Three or Four Nodes

Once the decision has been made to extend the cluster, adding the actual nodes is simple. Release 2.1 provides a centralized installation facility that allows all the configuration information for additional nodes to be entered into SMIT screens on

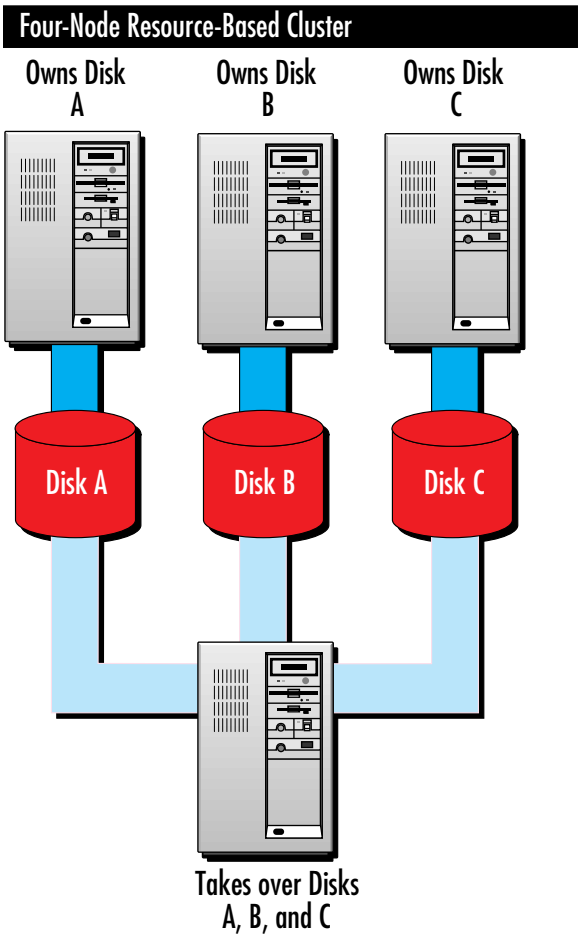


Figure 4. A four-node resource-based cluster

a single node. Then, again using the SMIT interface, the system administrator can execute a single command to propagate the updated cluster configuration to all nodes in the cluster. This simplified, centralized cluster configuration facility, coupled with the new resource-based cluster model, allows system administrators to easily extend an existing two-node cluster to three or four nodes.

In the two-node cluster example shown earlier in the article, a single-standby node backs up a single-server node. This configuration can easily be extended to a cluster in which two or three server nodes are backed up by a single standby. To do this, the system administrator would go to a single node in the cluster, define the new nodes and the resources that they owned, then define the resources owned by the new nodes as "takeover" resources to the existing standby node. Figure 4 shows the new configuration.

Customizing Event Processing

When migrating from Release 1.2 to 2.1, system administrators use the event customization facility to re-create any functions that have been added to the `topchng.rc` or `netchng.rc` configuration scripts.

For example, suppose that the operations manager needs to know immediately—day or night—whenever a fallover occurs. In Release 1.2, the HACMP/6000 Cluster Manager calls `topchng.rc 1 down` when the primary node fails. During the installation, the system administrator can modify the `topchng.rc` script so that when the primary server fails, a script called `wakeup` runs on the secondary. The `wakeup` script can use Kermit to auto-dial the pager of the operations manager and pass on a specific number combination.

In Release 2.1, the system administrator can avoid editing the event scripts by defining the `wakeup` script as a post-event associated with the `node_down_complete` event on the takeover node. By doing this, the takeover node automatically runs the `wakeup` script after it finishes processing the first node's failure.

No HACMP/6000 scripts need to be edited to add this in Release 2.1. Instead, the system administrator would use the `smit hacmp` fastpath to complete the following steps:

1. Select the `node_down_complete` event.
2. Enter the full path for the `wakeup` script in the post-event command entry.

Defining Application Servers

When migrating to Release 2.1, system administrators must use the SMIT interface for application servers to register the application start and stop scripts identified in the `node.servers` file. For example, suppose that a 1.2 cluster was used to make a database highly available as a back-end server. The `node.servers` script calls the `/usr/sbin/cluster/local/start_db_server` script during a `node_up` event to start the database and the `/usr/sbin/cluster/local/stop_db_server` script to stop the database during a `node_down` event. In Release 2.1, system administrators complete the following steps:

1. Using the `smit hacmp` fastpath:
   - Give the application server the logical name `db_server`.
   - Cite the path for the start script as `/usr/sbin/cluster/local/start_db_server`.

- Cite the path for the stop script as `/usr/sbin/cluster/local/stop_db_server`.

2. On the local node, define the application server as an owned resource.
3. On the takeover node, cite the application server as a takeover resource.
4. Be sure that each node's version of this resource is entered and configured correctly for that node (that is, path, local variables, hostname considerations, and so on).

Editing the clhosts File

In Release 2.1, the Cluster Information (`clinfo`) daemon uses a new file, `/usr/sbin/cluster/clhosts`, to store the hostnames or address of any HACMP/6000 server with which `clinfo` can communicate. When migrating from 1.2 to 2.1, system administrators must add all service labels or addresses of HACMP/6000 servers to this file on HACMP/6000 clients. An example of a `clhosts` file is shown in Figure 5.

```
clam_en0      # clam service
mussel_en0    # mussel service
oyster_en0    # oyster service
100.50.10.1   # shrimp service
```

Figure 5. A clhosts file

Thomas Casey, CLAM Associates, Inc., 101 Main Street, Cambridge, MA 02142. Internet: tom@clam.com. Mr. Casey is the manager of CLAM's technical writing group. He has a BS from Trinity College and an MS from Emerson College, both in Boston.

Robert Metcalf, CLAM Associates, Inc., 101 Main Street, Cambridge, MA 02142. Internet: bobmet@clam.com. Mr. Metcalf is a senior member of CLAM's technical support staff. He has a BS from Suffolk University and an MS from Simmons College, both in Boston.

First PowerPC-Based Notebook

The RISC System/6000 N40, a lightweight color notebook computer, combines the power of the PowerPC 601™ microprocessor and AIX in the industry's first PowerPC-based notebook workstation. Running at 50 MHz, the N40 achieves a SPECint92™ benchmark rating of 41.7, a SPECfp92™ rating of 51, and an Xmark rating of 2.58, making it more powerful than any notebook computer and many desktop workstations.

The 6.9-pound N40 features a 9.4-inch active-matrix color screen that offers wide-angle viewing in 256 colors. The N40's video memory supports up to a 1280 x 1024 image, which can be viewed through a pan-and-zoom feature on the display or through an externally connected monitor. Also featured is IBM's TrackPoint II™ pointing device, which eliminates the need for an external mouse.

Communications and networking features include external ports for Ethernet network support; SCSI-2 diskette drive support; and support for PCMCIA adapters. Other standard features include a removable disk drive with a 340 MB capacity; main memory support from 16 MB to 64 MB; an external display port supporting 1280 x 1024 resolution and up to 256 colors; ports for an external mouse, keyboard, and AppleTalk® printers; and a built-in speaker and microphone. The N40 also features Tadpole's Nomadic Computing Environment™, providing users with rapid save and resume capabilities, power management, portability tools, and other UNIX mobile computing innovations.

The N40 is available now, at $11,995. ■