Considerations and Sample Architectures for High Availability on IBM @server® pSeries® and System p5™ Servers

March 3, 2006

Table of Contents

Table of Figures

Introduction

Abstract

Because of the breadth and depth of the IBM @server® pSeries® and System p5™ hardware product line and related software products offered by IBM and other vendors, clients often face a daunting task in identifying the right set of System p5 hardware and software to meet their business needs. As a result, clients can experience uptime challenges with their System p5 environments (e.g. maintenance (applying fixes non-disruptively, knowing which fixes are appropriate/safe), configuration for greatest uptime, proper planning and configuration of IBM HACMP™ software to implement high availability clustering, and so on). This paper describes sample architectures for System p5 hardware that emphasize high availability. Traditionally, highly available pSeries environments have been the province of larger customers who have the resources to apply to high availability and have determined that the cost of implementing high availability is less than the cost of lost revenue in the event of a system or workload outage. In the last few years, high availability is becoming a factor for clients with smaller IT shops. In the small/medium enterprise (SME) space, uptime requirements are rising, making manual recovery from a failure, or downtime for maintenance, less acceptable to the business. “I’m not running a bank”-level availability is no longer good enough for a growing number of businesses.

High availability in this context is not the same as fault tolerance. A fault tolerant environment is designed to hide all failures from end users. Application software must generally be cluster-aware in order to achieve fault tolerance. Except in the special case of stateless transaction processing (such as HTML requests to a web server), complete fault tolerance is rarely achieved because it is difficult (and expensive) to guarantee that software is defect-free. (Some would say impossible rather than expensive.) In any case, fault tolerant hardware and software is very expensive.

HACMP software does not provide fault tolerance. That is, it does not eliminate outages. Instead, it reduces the duration of an application outage (due to a hardware or operating system failure) to a few minutes. Because the HACMP software does not attempt to provide fault tolerance, no special support is required within an application. The HACMP software can be used to make the vast majority of open systems applications highly available without requiring any application changes. If an application is able to recover after an AIX® operating system crash and reboot, it can almost certainly be made highly available using HACMP software. In an HACMP environment, user requests may occasionally fail, but with an appropriate IT infrastructure, the HACMP software is able to bring the application back up within a few minutes and users are able to resubmit the failed requests.

This document will start with some basic concepts for designing an IT infrastructure generally and will point out some considerations that influence the overall availability of the infrastructure. Then, example configurations will be discussed, starting with a simple cluster on smaller servers and moving from there to examples on larger environments which employ logical partitions (LPARs).

Useful Background Information

This document does not specify the amount of memory (RAM) or disk storage for each step. Memory and disk storage sizes must be determined by planning for the workload that will be run and examining capacity requirements. In general, more memory or disk storage is better than less; also, keep in mind that the amount needed tends to increase over time. The book cited below talks about this topic in great detail.

- Redbook: IBM eServer pSeries Sizing and Capacity Planning: A Practical Guide
  http://www.redbooks.ibm.com/abstracts/sg247071.html?Open

The IBM System p5 hardware range supports the optional Advanced POWER Virtualization feature, a suite of hardware and software technologies that provide access to the following components:

- Micro-Partitioning technology (LPAR creation and management where individual LPARs are given less than one physical CPU)
- Virtual I/O Server (virtual SCSI, virtual Ethernet, and Integrated Virtualization Manager)
- Partition Load Manager

Further information about virtualization on System p5 hardware can be found at:

- Advanced POWER Virtualization overview
  http://www-03.ibm.com/servers/eserver/pseries/ondemand/ve/

- Redbook: Advanced POWER Virtualization on system System p5
  http://www.redbooks.ibm.com/redpieces/pdfs/sg247940.pdf

Capacity on Demand (CoD) is an important optional feature of the System p5 range. It gives the ability to build in reserve CPU and memory at the factory that remains inactive until needed. Additional information on CoD is available at:

- Capacity on Demand overview
  http://www-03.ibm.com/servers/eserver/pseries/ondemand/cod/

- IBM eServer System p5 550, System p5 570, System p5 590, System p5 595: Working With Capacity on Demand
  http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/ipha2/ipha2.pdf

This document does not attempt to cover IP or storage networking in detail and instead illustrates these concepts at the block diagram level. Detailed information regarding networking and storage area network (SAN) technologies and practices can be found at:

- Redbook: Introduction to Storage Area Networks
  http://www.redbooks.ibm.com/redbooks/pdfs/sg245470.pdf

- Redbook: IBM TotalStorage: SAN Product, Design, and Optimization Guide
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246384.pdf

- Redbook: IP Network Design Guide
  http://www.redbooks.ibm.com/redbooks/pdfs/sg242580.pdf

- Redbook: Extending Network Management Through Firewalls
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246229.pdf

Further reading on HACMP technologies and practices can be found at:

- HACMP Website
  http://www-1.ibm.com/servers/eserver/pseries/ha/

- HACMP resources
  http://www-1.ibm.com/servers/eserver/pseries/ha/resources.html

- HACMP best practices white paper
  http://www-1.ibm.com/servers/eserver/clusters/whitepapers/hacmp_bestpractices.html

- Redbook: Implementing HACMP Cookbook
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246769.pdf

- HACMP V5.3 Concepts and Facilities Guide
  http://publib.boulder.ibm.com/epubs/pdf/c2348646.pdf
  For overview information regarding application availability, refer to Chapter 5.

- HACMP V5.3 Planning and Installation Guide
  http://publib.boulder.ibm.com/epubs/pdf/c2348616.pdf
  For application availability information, refer to Chapter 2 and Appendix B.

Detailed information regarding the IT Information Library (ITIL):

- IBM ITIL White paper
  http://www-1.ibm.com/services/us/imc/pdf/g510-5072-information-technology-infrastructure-library.pdf

- IBM Global Services ITIL Website
  http://www-1.ibm.com/services/us/index.wss/offerfamily/its/a1000429

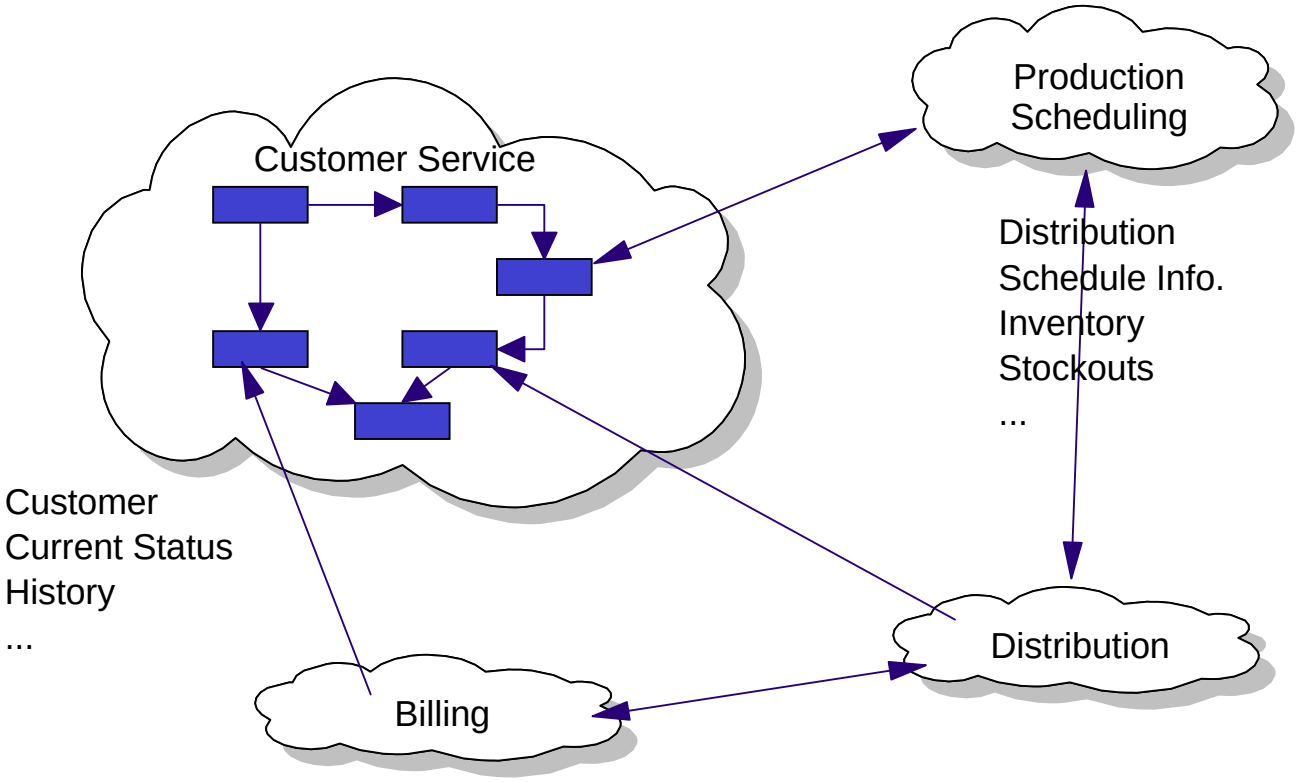
High Availability on pSeries and System p5 Hardware

Overall Infrastructure Considerations

In practice, when designing an IT infrastructure, you would start with a summary of the business context and functional plus non-functional requirements. Though this document is not intended to be a treatise on how to design a modern end-to-end IT infrastructure, the examples shown here are gleaned from best practices and lessons learned from IBM and its varied customers. IBM also brings the necessary design skills to the table when working with a client to deliver a new or updated infrastructure.

For the purposes of this document, it is useful to quickly illustrate the different levels of thought that go into designing an IT infrastructure. As with any type of project, defining the requirements and goals is a critical and sometimes difficult first step. One technique used to get a handle on this often daunting task is to create a business context diagram, an example of which is shown below.

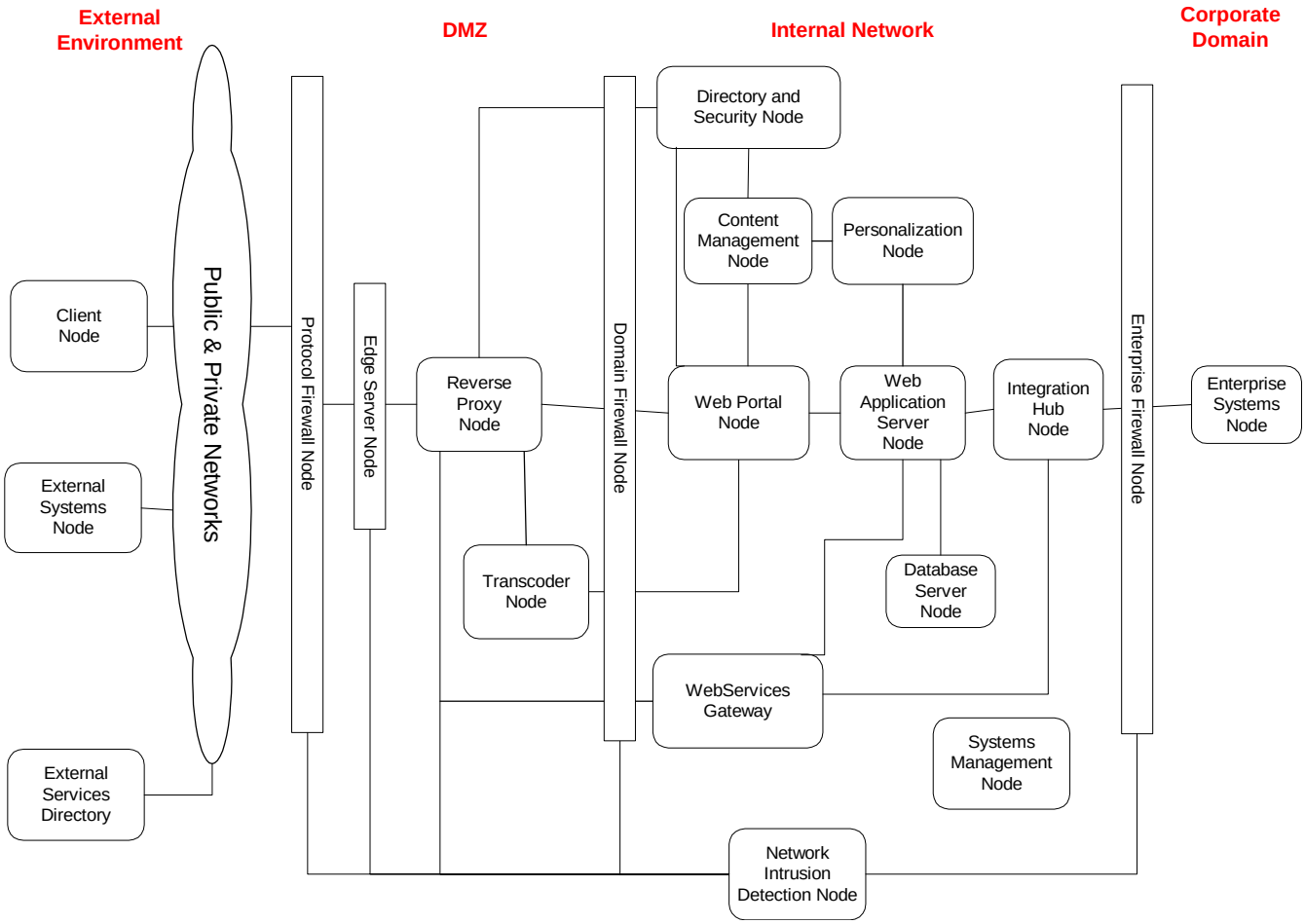
Figure 1 - Sample Business Context Diagram



The business context diagram, along with the requirements, plus use cases, would in turn be used to direct the design of a high-level topology diagram. The diagram below illustrates an infrastructure for a business that conducts transactions over the internet, certainly a common model. Keep in mind that this diagram is fairly abstract; the term

"node" does not refer to a single server but instead is used here to indicate an IT function within the overall infrastructure. This modular approach makes it easier to keep track of the different functions required in the overall infrastructure.

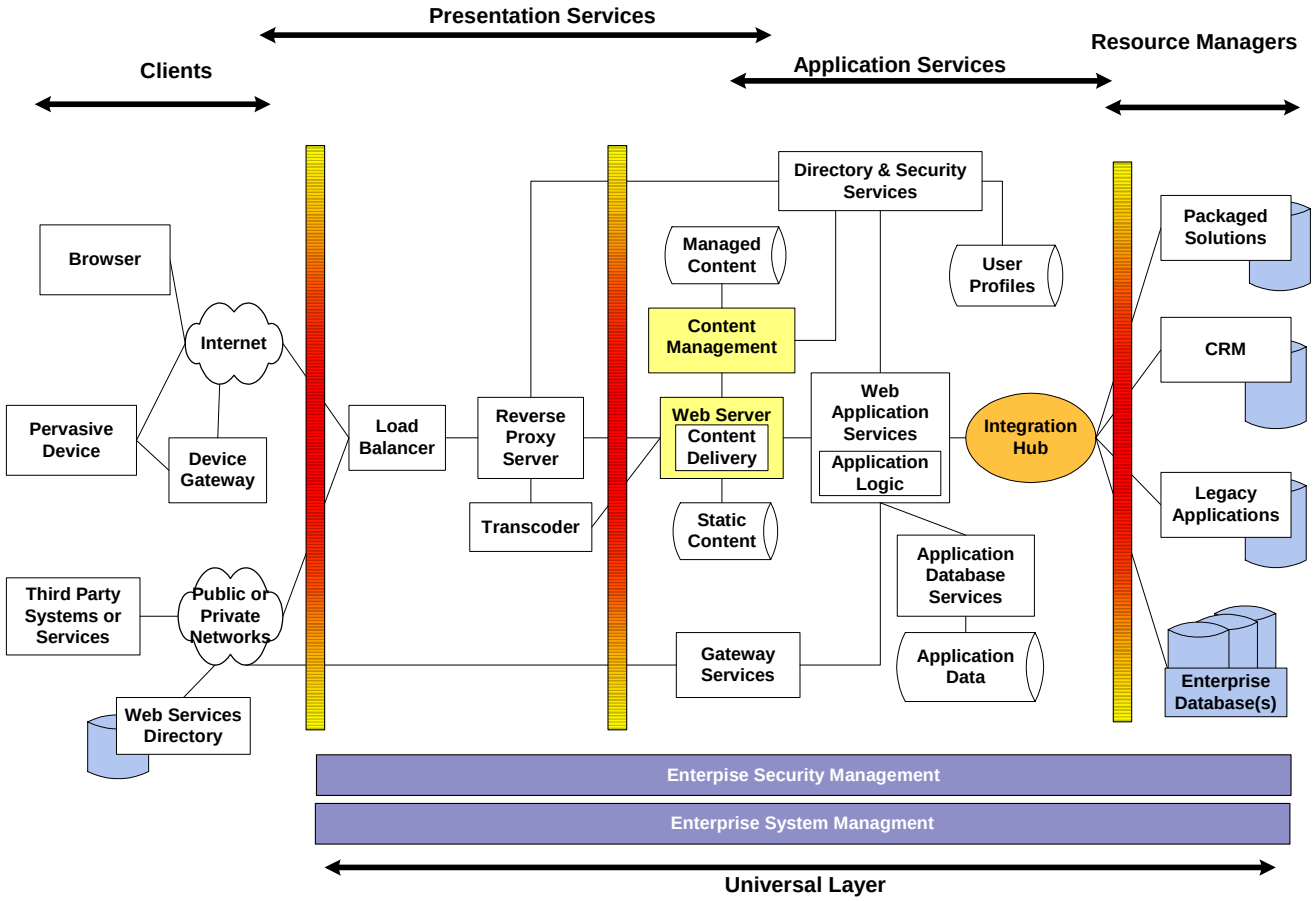
Figure 2 - Conceptual-level System Topology Diagram



A detailed examination of each node in this diagram is outside the purpose of this document, but its inclusion here serves as a useful reminder of the many functions that must be considered when designing a modern IT infrastructure. Note, for example, the inclusion of three levels of firewalls as well as the "directory and security" and "network intrusion detection" nodes. The modern business climate demands a focus on security as indicated in this diagram.

Clearly, the level of detail is insufficient to be used directly as a physical representation of the infrastructure. From the diagram shown above, an IT architect would then produce an architectural overview diagram similar to the example below.

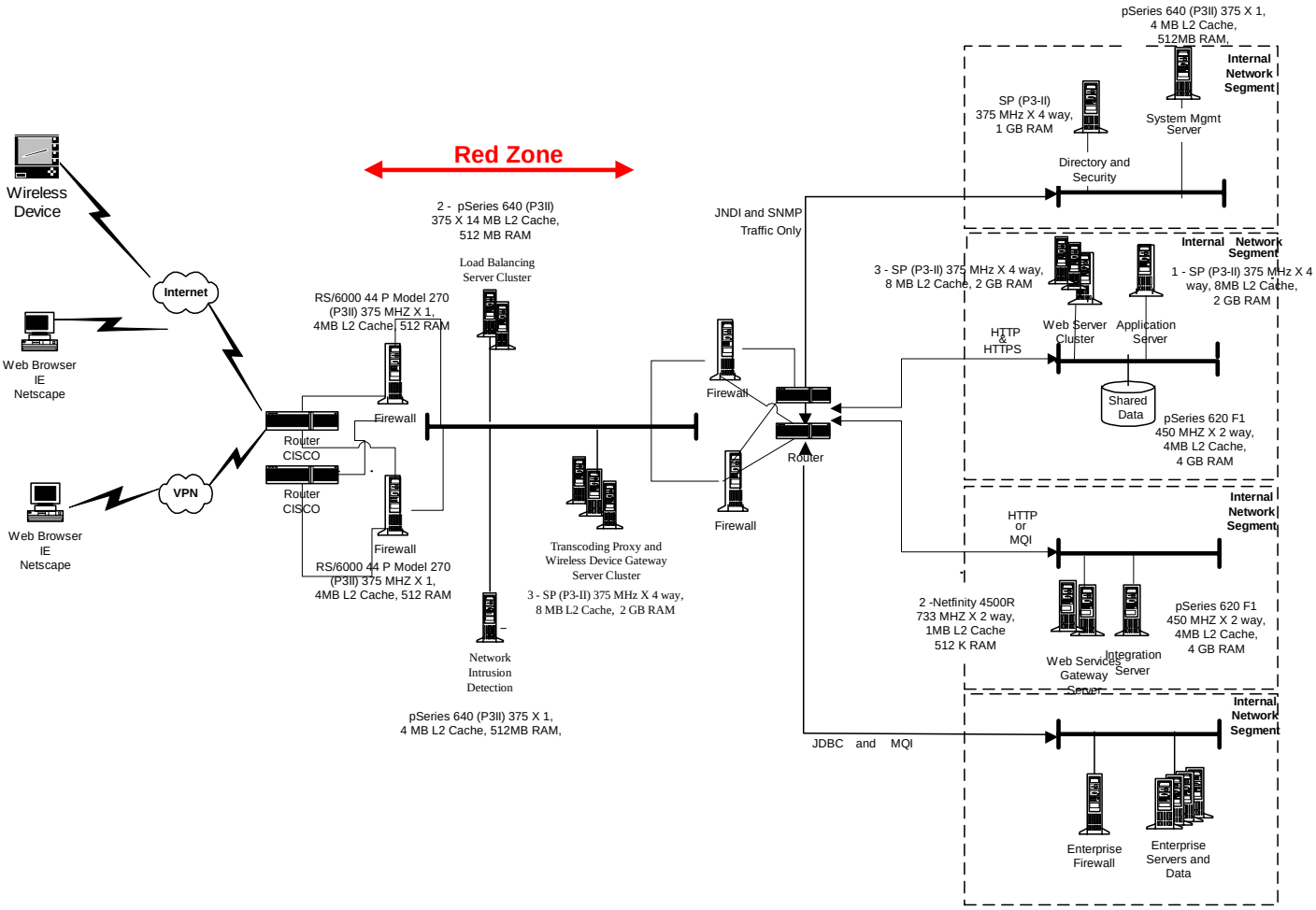
Figure 3 - Sample Architecture Overview Diagram



You can see how the "nodes" shown in the topology diagram are now replaced with indications of potential servers, workloads, and storage. The interconnections between the components suggest how the IP and storage networks might be designed.

From the architecture overview diagram, the IT architect and technical leaders begin to sketch out a physical layout, similar to the example shown below. Don't worry about the specific callouts in this diagram – it is meant only to illustrate what a typical physical layout diagram might look like. You can see how the abstract components in the architectural overview diagram above are now translated into a more physical representation, with example servers, storage, and networking components being illustrated.

Figure 4 - Sample Physical Layout Diagram



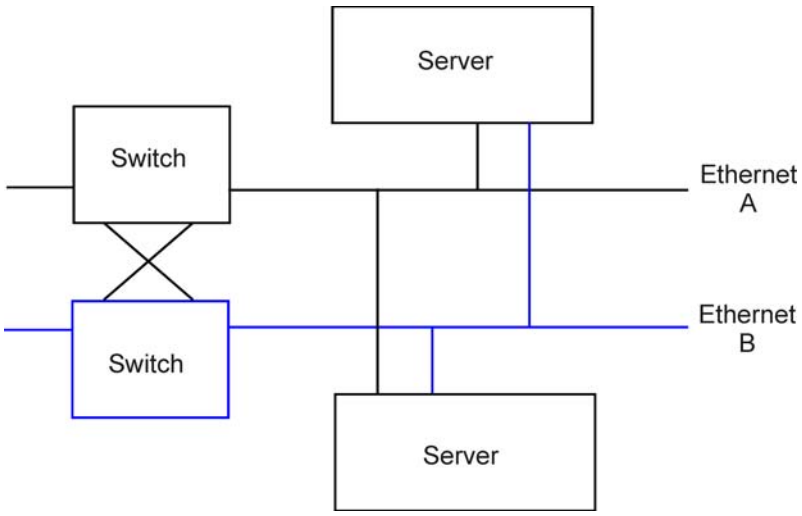
The point of this section is to illustrate the disciplined, step-by-step process that is employed when an IT infrastructure is designed. In particular, when IBM IT architects are engaged by a client's IBM sales team to build a new infrastructure (or to update a client's existing infrastructure), they base their efforts on this process and have access to industry-specific reference architectures on which they build a particular client's solution.

Network Considerations

The sample physical layout diagram shown above indicates a segmented network approach, which is typical for an infrastructure of this scope. What the diagram doesn't illustrate is the need for redundancy throughout the IP network topology. Even if you are not looking to build a highly available infrastructure (in which redundancy at every level is coupled with HA clustering software to create systems which can react automatically in the event of individual component failures), it is still very important to build in network redundancy and segmentation. By doing so, you give yourself the ability to recover more quickly from a network component failure.

Typically, two IP network topologies should be built in parallel, with redundant network hardware (e.g., switches, routers, firewalls) connecting each segment. Further, the network components should be cross-connected so that traffic being handled by one component can be automatically handled by the second in case the first one fails. Each server on the network should have an Ethernet port connected to each Ethernet network. The following diagram illustrates this concept.

Figure 5 - Dual Ethernet Paths



A further note about the server Ethernet connections: many Ethernet adapters now offer dual ports on a single adapter. Keeping the example diagram in mind, it is tempting to use one port on a single adapter for Ethernet A, while the second port is connected to Ethernet B. Be aware that this approach makes the Ethernet adapter a single point of failure. If the Ethernet adapter in the server fails, both of the ports it provides will be unavailable, effectively cutting the server off from either Ethernet network. To avoid this issue, install an Ethernet adapter for each network connection required by a given server.

Storage Considerations

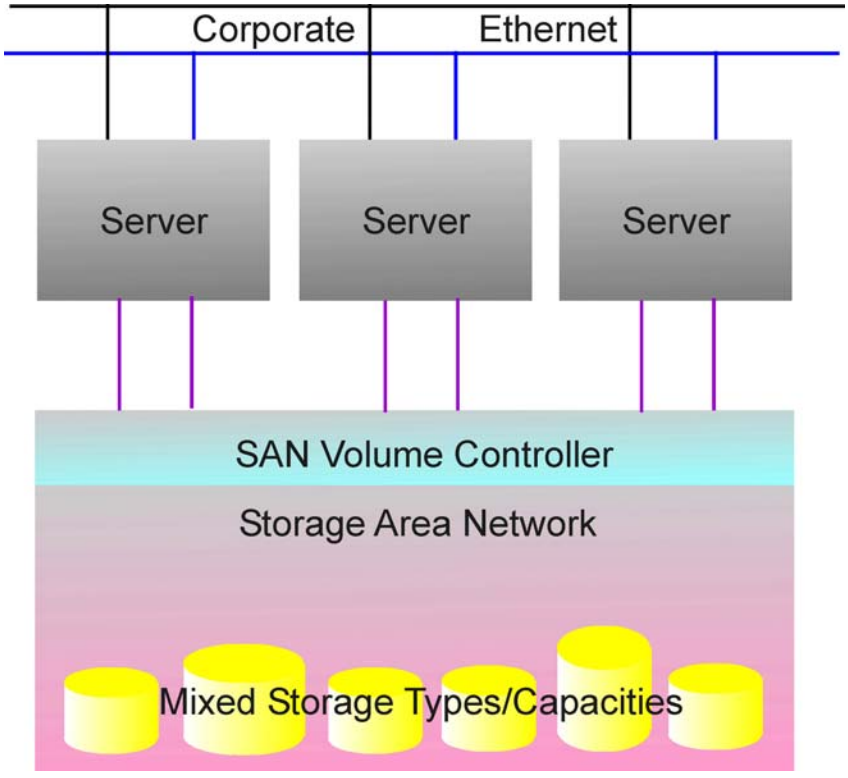
The attachment of storage to servers in the infrastructure brings with it the same concerns about redundancy in terms of access to business data. In years past, external storage devices were directly attached to the server which needed to read and write data on that device. Administrators quickly learned that redundant connections (e.g. SCSI, Fibre Channel (FC)) were required so that the server didn't immediately lose access to its data if a single storage connection failed. This still did not address the problem of providing quick access to the data in the event of a server failure. The workload on the server could be brought up elsewhere, but the storage would have to be cabled to the replacement server, lengthening the duration of the outage.

There is also the issue of protecting against storage device failures. Today, storage devices, especially at the higher end, have significant redundancy built in, but even so, one must plan for the remote possibility that the entire storage device could fail, or could require a disruptive firmware upgrade. While that particular storage device is down, how can its data be accessed so that the servers depending on the data can continue working?

Enter the Storage Area Network (SAN) – in gross terms, applying the concept of networking to the storage world. SANs are well-documented and now pretty common, so this document will not spend any time describing SAN concepts. Further reading on the subject is cited in the Useful Background Information section above. As with Ethernet networks, it is possible and indeed strongly recommended that SANs are designed with redundancy at every level (storage connections, switches, directors) so that individual component failures do not cripple the SAN. In the diagram below, note the redundant connections between each server and the SAN fabric. Avoid the temptation to use a single dual-port adapter to achieve the dual connections, as the adapter itself becomes a single point of failure – if it malfunctions, both storage connections will be unavailable. Instead, install a dedicated I/O adapter for each storage connection.

In the traditional SAN model, care must be taken to map different classes of storage devices to different storage needs. Often, administrators must deal with cases where some storage devices are underutilized, while others are running out of available space. Storage virtualization is an interesting way to address this problem. Conceptually the idea is to decouple servers from the storage they are using.

Figure 6 - Storage and IP Networking Block Diagram



To use a bit of AIX terminology, all the server sees is an hdisk – it doesn't know (or care) what physical storage device is providing that disk resource. IBM's TotalStorage SAN Volume Controller (SVC) is a hardware/software product that adds this storage virtualization capability. While it is optional when building an IT infrastructure, it should be strongly considered when there is a need for an adaptable, resilient storage infrastructure that can maximize the use of existing storage capacity across a mix of storage devices.

Despite the advantages offered by advanced storage technologies in the areas of manageability, resiliency, and virtualization, consideration must be given to the protection of critical data. In addition to the mirroring and backup of business data (e.g. database contents), consider also protecting critical software programs from the operating

system up through the middleware/application stack. Modern storage systems offer mirroring capabilities, and in the case of the AIX 5L™ operating system running on System p5 servers, options are available for data mirroring and backup as well.

IT Process Considerations

You cannot achieve high availability without discipline in the infrastructure design and implementation activities, nor will high availability be possible if there are single points of failure in the IP or storage networks. However, even if you do those things correctly, high availability will remain out of reach if the IT processes governing the operation and maintenance of the infrastructure are ill-defined or not well-documented. Additionally, if the IT processes are well defined and documented, high availability will still be unattainable if the IT staff is not well-trained on those processes and is not maintaining the IT processes over time.

Often, businesses suffer outages due to deficiencies in the design of the infrastructure, or due to single points of failure in the network environments, or due to immaturity in the business's IT processes. It is tempting to throw more technology at the problem, such as purchasing an HA clustering software package and hurriedly implementing HA clustering as the cure-all against further outages. Unfortunately, taking this approach without first addressing any underlying infrastructure or operational issues actually tends to make the problem worse, because you've now added another layer of software into the mix that must be managed along with the rest of the infrastructure.

Here are a few basic IT process lessons learned:

- IT processes must be clearly defined and documented. Trying to run even a small IT environment "by feel" is doomed to fail.
- IT staff must be trained on the IT processes, and the processes must be maintained as the environment changes over time. Too often, a business will take the initiative to get control of their processes and go through the hard work of defining and documenting everything. Copies are printed, bound, and handed out. The copies either disappear under other paperwork or get put on a bookshelf. In a surprisingly short period of time, the actual processes deviate from the written processes.
- Institute regularly scheduled process tests, and use the test results to identify and correct inaccuracies or weaknesses in the process. This also forces the operations staff to review and touch the processes, increasing their process familiarity.
- IT roles and responsibilities must be clearly defined. For example, consider defining an availability manager role. An availability manager becomes the "traffic cop" for all changes to the environment and the focal point for resolution in the event of an outage.
- A central database is needed to document all aspects of the environment (e.g. standard software catalog, individual machine firmware and operating system levels), so that everyone has access to the same, up-to-date information. Too often, everyone keeps their own copy of some piece of data about the environment, which leads to contradictory views of the state of the environment.

- IT processes that deal with change control are particularly important in an HA environment. Often, clients fail to realize that making changes in a HA cluster can easily disrupt the operation of the cluster, especially during a failover. Robust change control processes which address the special needs of the HA cluster environment are critical; equally critical is staff familiarity with, and adherence to, these change control processes.

The IT Information Library (ITIL®) is a standard framework for use as a base in defining IT processes and using them to manage an IT infrastructure. While at first glance the totality of the ITIL framework can seem overwhelming, best of breed IT shops invest the time and effort in embracing ITIL and basing their own IT processes on it. Sources for further reading about ITIL are cited in the Useful Background Information section in the Introduction.

Single Points of Failure (SPOF)

The elimination of single points of failure (SPOF) from an IT infrastructure is one of the core tenets of high availability. Any single hardware or software resource which does not have a redundant backup can potentially become the source of a significant IT outage – if it fails, its function is unavailable until the failure can be corrected. In the sections above, we discussed the need for redundancy in the IP and storage networks. The same thinking must be applied to the server and software resources in the infrastructure, and the following sections will show sample configurations for achieving this.

Clustering for High Availability

After identifying and eliminating single points of failure in networks, storage, and servers, the question then becomes "how do I recover from a component failure more quickly and consistently?" In the System p5 realm, IBM offers the HACMP product, which is used to design and deploy high availability clusters that monitor selected hardware and software resources and take action automatically to "failover" the resources to designated backup locations.

Let's be clear: the term "high availability" does not imply the elimination of failures. Instead, high availability is often defined to be: *the attribute of a system to provide service during defined periods, at acceptable or agreed upon levels, and the masking of unplanned outages from end users. It employs automated failure detection, recovery, bypass reconfiguration, testing, problem and change management.* A bit of a mouthful, to be sure, but the point is that a highly available system is designed to help minimize the impact of a failure on end users. High availability is not the same thing as fault tolerance. In an HA environment, actions are taken when a failure is detected to move the work of the failed component to a redundant component. Time is required to stop affected workloads and restart them on backup resources. The time needed to accomplish this "failover" can vary significantly, depending on the type of failure (say, network adapter versus serious server failure) and the application characteristics. HA environments such

as described in this document do not promise or deliver zero downtime and are therefore not suitable for truly life-critical workloads.

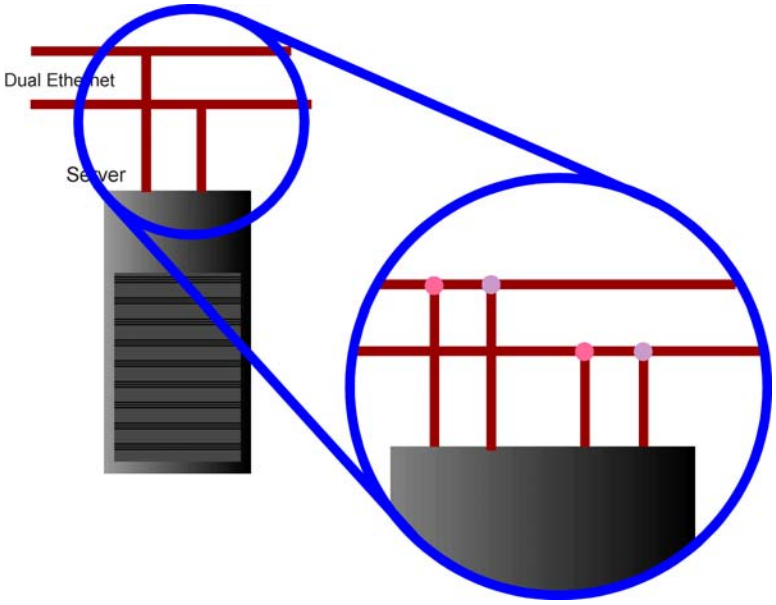
At the server level, HA clustering requires redundancy in the IP and storage connections from the server to the rest of the environment, along with a similarly configured backup server onto which the workload can be moved in the event of a failure on the primary server.

Notes about Cluster Diagrams Used in this Document

In the following sections, sample cluster configuration diagrams are shown to reinforce the concepts discussed in the accompanying text. For visual clarity, the Ethernet and storage network connections are simplified.

In the case of Ethernet network depictions, you will note that a single connection is implied between each cluster node and each individual Ethernet network. This was done to improve the visual clarity of the illustrations. You should assume that the connection line between a cluster node and a single Ethernet network represents at least two individual Ethernet connections, with each connection supported by a like number of Ethernet adapters installed in the cluster nodes. The following diagram illustrates this concept.

Figure 7 - Detail of Redundant Ethernet Connections



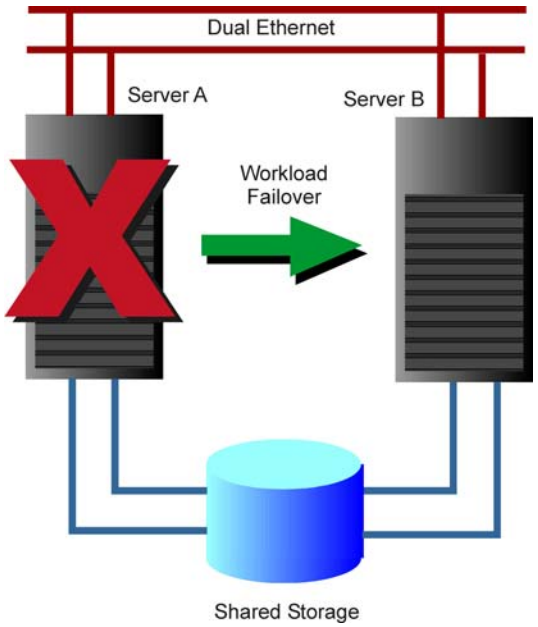
On the storage side, the following diagrams all depict duplicated, direct connections between the cluster nodes and shared storage. In practice, with the use of SANs and (optionally) the IBM SAN Volume Controller, redundant connections between each cluster node and each SAN fabric would be needed to avoid SPOFs in the node to storage connections.

Importance of Non-IP Networks

The HACMP software uses packets transmitted between cluster nodes to determine each node's health. While it can use IP networks for this purpose, it is strongly recommended that HACMP clusters be set up with at least one non-IP network for this so-called heartbeating. This will prevent a cluster which uses the IP network for heartbeat traffic from becoming partitioned in the case of total IP network failure. In a partitioned cluster, each node in the cluster is operational, but because no method exists for the HACMP software to transmit heartbeat packets among nodes, each node believes it is the only remaining node in the cluster and attempts to take over workloads from what it believes are failed nodes. This condition must be avoided as it can lead to data corruption and cluster failures. The HACMP software can send heartbeat traffic over serial and storage connections; these connections are known as non-IP networks. In the following discussions of cluster configurations, always assume that at least one non-IP network is configured for HACMP cluster heartbeat traffic.

Classic Two-Node HA Cluster

Figure 8 - The Classic 2-node HA Cluster



The diagram above illustrates the classic two-node HA cluster. Note the redundant connections to the IP network and the shared, external storage. Recall the admonishment in the Network Considerations section against the use of a single, dual-port Ethernet adapter to establish dual network connections to each server. Similarly, the use of a single, dual-port I/O adapter (e.g. Fibre Channel) to connect each server to the shared storage should also be avoided in favor of a dedicated adapter for each storage connection. For this example, assume that a single workload runs on the primary cluster node, server A, with server B acting as the standby node. When a failure occurs on the primary node, the HA clustering software takes action to move the workload over to the standby node. Because the workload data is stored on the shared storage device, either node can access the data. Typically, in a "hot standby" (also called "active-passive") cluster such as this, the workload is moved back to the primary node once the failure there has been corrected.

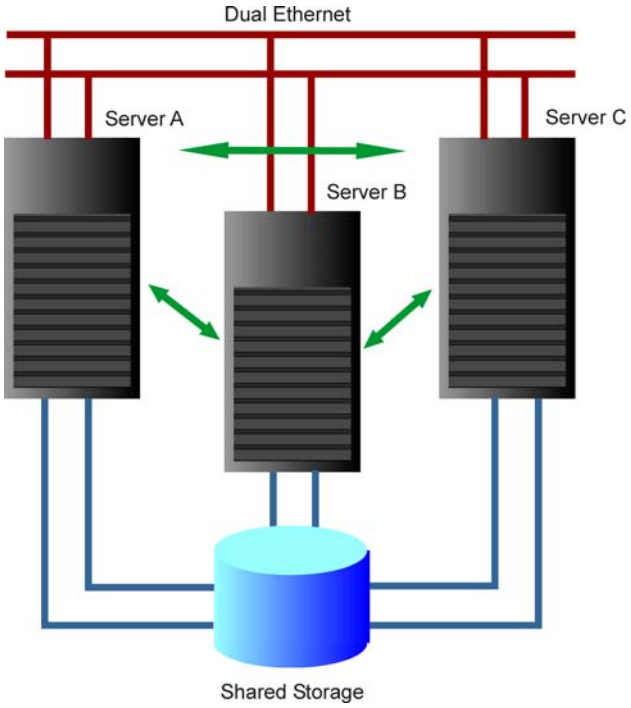
Alternatively, it is possible to build a "mutual takeover" (or "active-active") 2-node cluster. In this case, both server A and server B are running production workloads. In the event of a failure on server A, the workload is moved to server B, and vice versa.

In this classic 2-node scenario (it should be noted that approximates 80% of deployed HACMP clusters are still of the 2-node variety, even though the HACMP product supports up to 32 nodes in a cluster), the capacity of the servers in the cluster must be accounted for. If you choose to build a hot standby cluster, the standby node must have sufficient capacity to run the production workload normally hosted on the primary node. In this simple example, you would typically configure two identical servers. In a mutual

takeover configuration, each server in the cluster must be configured with sufficient capacity to run both workloads simultaneously in the event of a failure on either cluster node. With traditional non-partitioned, fixed-capacity servers, this leaves half of the total processing capacity in the cluster idle during normal operation, which is inefficient. We will discuss how System p5 hardware's Dynamic Logical Partitioning (Dynamic LPAR) and optional Capacity on Demand (CoD) feature can be used to address this issue.

Three-Node Mutual Takeover Cluster

Figure 9 - 3-node Mutual Takeover Cluster



The three-node mutual takeover cluster is becoming a more commonly deployed HA cluster configuration. It consists of three servers connected to shared storage, and each of the three servers is running a production workload. Depending on how the takeover relationships are defined, a failure on one node could result in the workload being restarted on either of the other two. In the extremely unlikely case of two failures on different nodes, all the workloads would then failover to the remaining node. Applications running on each node must store their data, including application configuration data, on the shared storage device, so that no matter which node the application is running on, the data is always available.

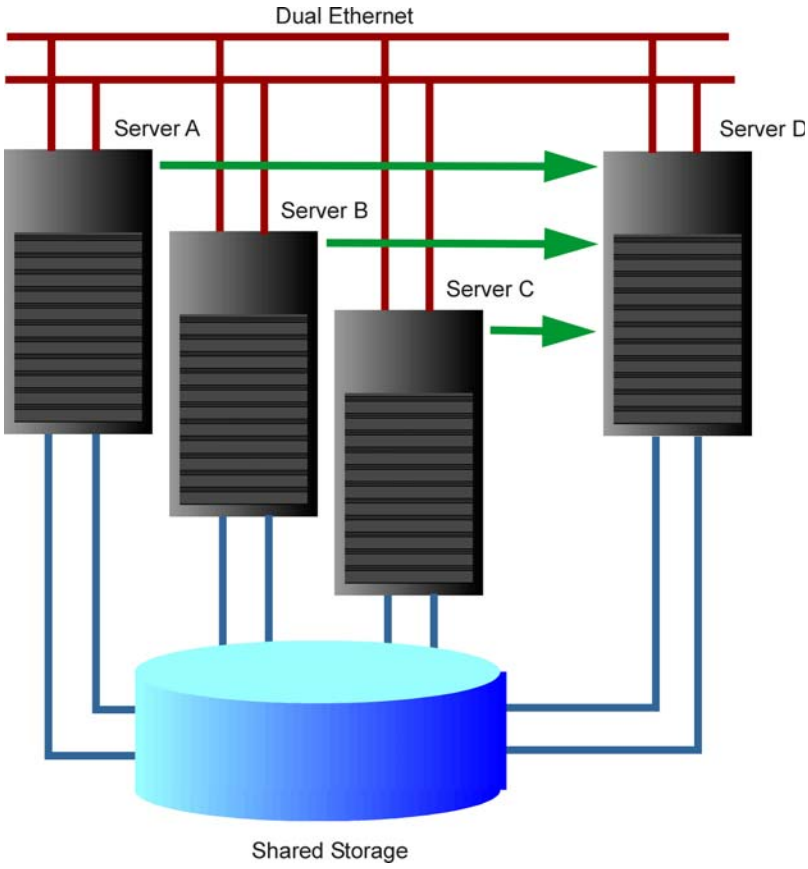
This configuration increases the level of protection versus a 2-node cluster. As noted above, even if two nodes suffer failures, the workloads can all be move to, and run on, the remaining node. On the other hand, the issue of efficiently utilizing the processing resources in the cluster is magnified, as one would want to build each single node so that it has sufficient capacity to run all of the workloads in the cluster. When there is no failure condition, each node would therefore be running at less than 1/3 of its total capacity.

"n+1" Cluster

In this HA cluster model, some number (usually 2 to 4) of nodes in the cluster are running production workloads, and all of them are configured to failover to a single standby server. This type of configuration was created as a way to address the issue of low processing resource utilization on the cluster nodes when running under normal conditions. The idea here is that, while the single standby node is sitting idle during normal operation, the other nodes in the cluster can all be used for production workloads.

Often, a judgment call is made when sizing the standby node, such that the node is not built with enough capacity to handle all of the workloads from the other cluster nodes. The risk of all the production nodes in the cluster failing simultaneously is deemed to be sufficiently small that the standby server need be sized to handle only the two most critical workloads simultaneously.

Figure 10 - n+1 Cluster Configuration



Standardization and Simplification

As you can see, there are several different possible HA cluster topologies, and with the HACMP product (which can cluster up to 32 nodes) it is possible to create complicated cluster configuration that go well beyond the examples shown above in terms of intricacy. Unfortunately, experience has shown that complex clusters are much more difficult to manage. With more nodes and complicated failover relationships, it becomes more likely that mistakes will be made in the initial configuration of the cluster and less likely that the possible failover scenarios will be fully and frequently tested. Ultimately, this can lead to situations where a failover scenario occurs which has not been tested, and as a result the failover does not function as expected. The same issues tend to occur when lots of small, uniquely-configured clusters are deployed. It's very hard for even the most proficient IT staff to keep in mind the nuances of each HA cluster when each one is slightly (let alone significantly) different from the others.

The recommendations to counter these situations are pretty intuitive:

- Develop one (or at most, a very few) standard cluster configuration that different workloads are plugged into. By doing this, it becomes much easier to manage and maintain the clusters, because your IT staff knows what to expect. It also pays dividends in testing, because an additional, identical cluster can be set up to test changes and upgrades that will apply to all of the production clusters.
- Keep the cluster configuration as simple as possible. This reduces the number of failover scenarios that have to be learned and tested, which means that there's a much better chance that each scenario will be thoroughly tested by and familiar to the IT staff prior to an actual failure, increasing the likelihood that a real failover will occur as planned.

HA Clustering with System p5 Hardware

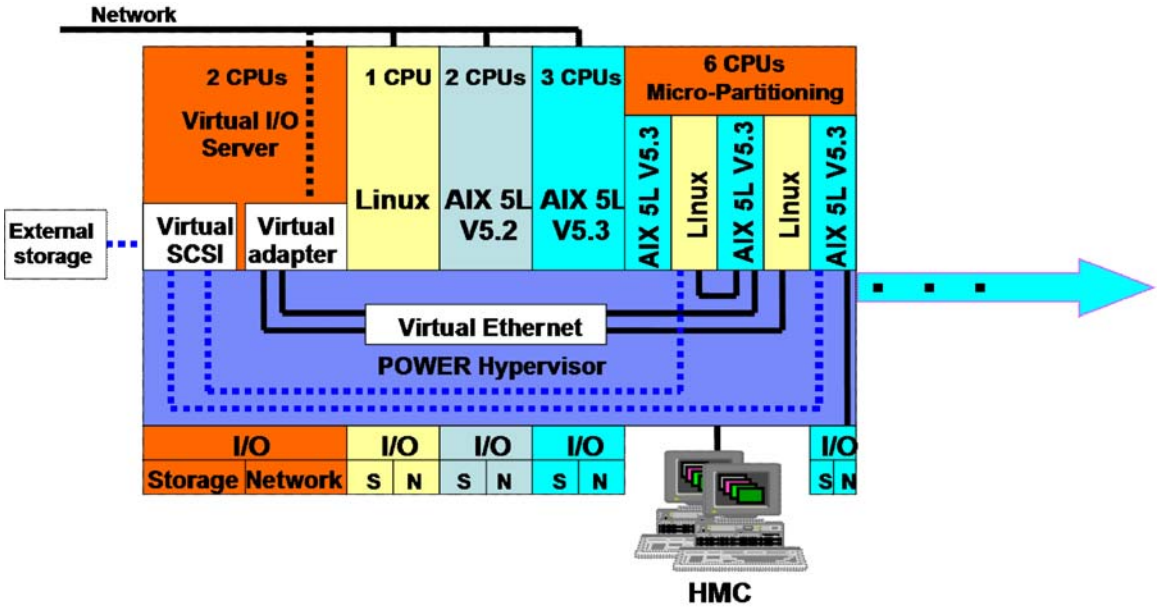
Up to this point, we've discussed at a high level general considerations that one must take into account when considering high availability clustering, from infrastructure-wide design and redundancy to example HA cluster configurations. An infrastructure incorporating the practices and technologies discussed so far typically has different server types deployed (Intel processor-based, commercial UNIX® operating system such as the AIX 5L operating system on System p5 hardware, and often in large environments, mainframes on the backend). We will now look at how System p5 servers fit in this type of infrastructure, pointing out how System p5 hardware's virtualization and capacity management features bring additional value and flexibility. Let's quickly recap these features here:

- Logical Partitioning (LPAR)/dynamic LPAR (dynamic LPAR) – the ability to subdivide a System p5 server's total CPU and memory capacity into discrete partitions (LPARs), each of which can run an operating system instance. Resources can be reallocated across the LPARs without rebooting the machine or partitions.
- Capacity on Demand (CoD) – an optional capacity and cost management feature which allows reserve CPU and memory resources to be installed at the factory in an inactive state. When conditions require it (e.g. due to a workload peak or when manually activated), the reserve resources are activated and assigned to either new or existing LPARs. After the exceptional condition has passed, the resources can then be deactivated and put in reserve again. The customer pays only for the extra resources used during the exceptional condition.
- Micro-Partitioning technology – part of the optional Advanced POWER Virtualization feature, Micro-Partitioning gives the administrator the ability to configure LPARs that receive less than one full CPU's worth of processing power (down to as little as 10% of 1 CPU). This technology gives increased flexibility to the administrator and helps optimize and maximize the use of the server's resources.
- Virtual I/O Server (VIOS) – part of the optional Advanced POWER Virtualization feature, the VIOS provides the ability to share physical Ethernet and disk connections across LPARs by providing each LPAR virtual Ethernet and disk links and mapping those to existing physical hardware resources. Inter-

LPAR communications can take place entirely within the System p5 Hypervisor layer, reducing external IP and storage network traffic and reducing the number of physical I/O adapters required. In an HA environment where VIOS is employed, seriously consider setting up two VIOS servers so that a failure in a single VIOS server does not impact virtual LAN or disk traffic.

- Partition Load Manager – Management software which the administrator uses to balance LPAR resource needs
- System p5 servers support the use of dual Hardware Management Consoles (HMC's), a practice endorsed in particular for environments where high availability is important. As with other single points of failure, the idea here is to protect against the inability to manage a System p5 server due to a single HMC failure.

Figure 11 - System p5 Advanced POWER Virtualization Diagram



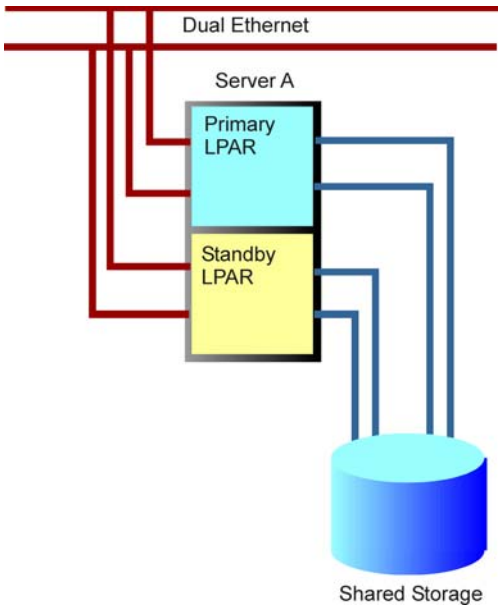
Aside from the thumbnail descriptions given above, this document assumes the reader is familiar with System p5 hardware's unique virtualization and capacity management features. Further reading is cited in the Useful Background Information section of the Introduction.

In addition to System p5 hardware and virtualization features that should be considered in relation to availability, the AIX operating system also plays an important role in availability. Features such as multi-path I/O (MPIO) and Etherchannel provide redundancy in the operating system's handling of disk and network traffic. The AIX operating system also provides support for the insertion or removal of "hot-plug" adapters while the operating system is running.

A popular System p5 choice for implementation in a medium to large IT infrastructure such as has been discussed so far, is the System p5 570. This particular machine brings

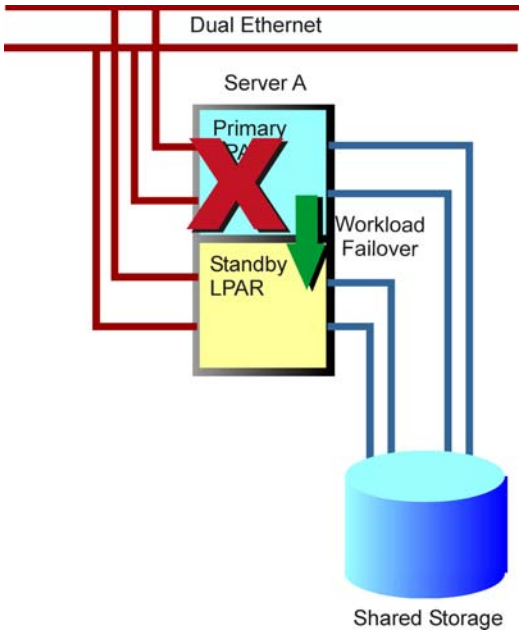
outstanding price/performance, expandability (2-16 CPU's) and a modular packaging approach to mid-size environments. Its big brothers, the System p5 590 and System p5 595, are high-end servers with up to 32 (590) or 64 (595) 64-bit POWER5 CPUs. Customers also deploy some of the lower-end System p5 servers, such as the System p5 520 and System p5 550 for smaller, standalone workloads.

Figure 12 - Two-node Cluster Implemented with LPARs



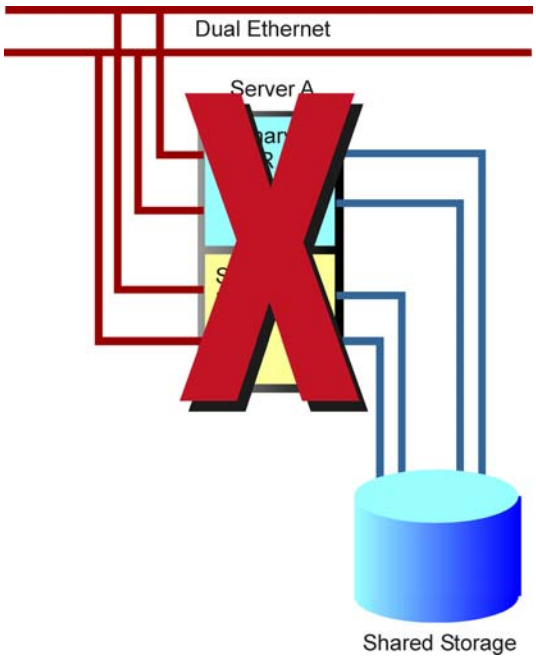
Recall the classic two-node hot standby cluster shown above. The diagram accompanying that example indicates two physical servers, and we pointed out that, except when the workload has to be failed over to the standby server, that standby server is sitting idle. Let's update that example using a System p5 server with two LPARs. Note that putting a 2-node cluster on two LPARs in a single server is not recommended for a production environment and is shown here for educational purposes. The weakness of this approach is discussed below.

Figure 13 - Failover in 2-LPAR Cluster on One Server



In the event of a failure in the primary LPAR, the workload will be failed over to the standby LPAR in the same manner that the workload failed over to the standby server in our classic 2-node cluster example.

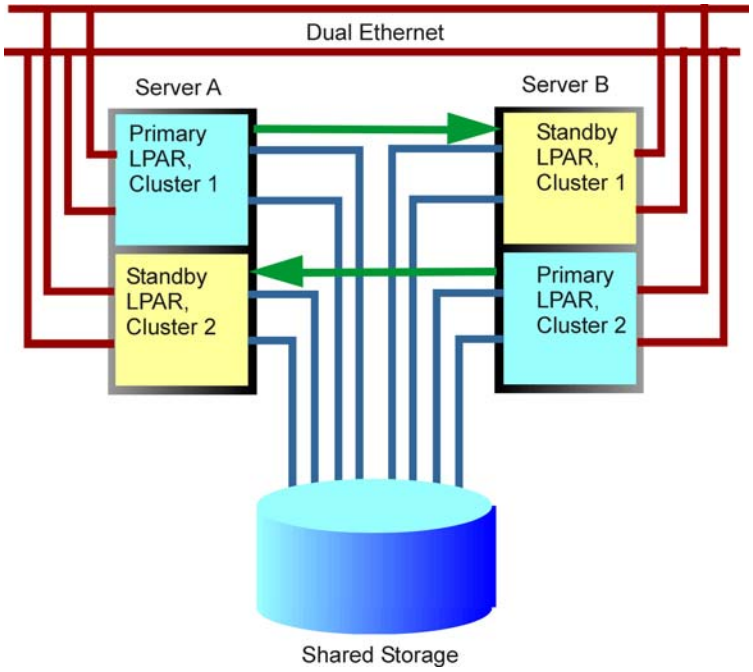
Figure 14 - HA Cluster Down Because Server Failed



There is an issue with consolidating an HA cluster into LPARs on a single physical server. If the entire server fails or needs to be taken offline for a firmware upgrade, the HA cluster is unavailable until the server comes back up. This is a step backwards from the classic model with two physical servers.

This model, then, is not recommended for use in a production environment. However, it is a good way to create development or test clusters while reducing the number of physical servers needed.

Figure 15 - Two, 2-LPAR Clusters Across Two Servers

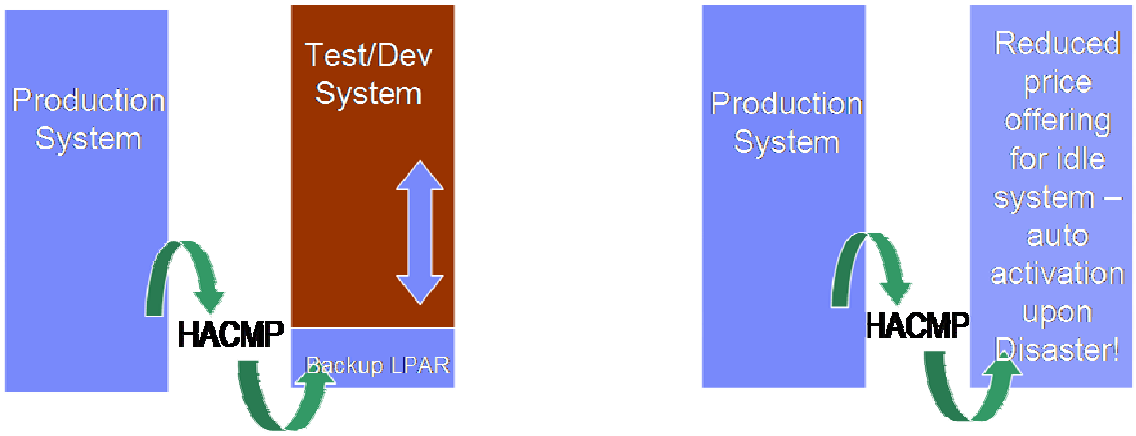


The resulting recommendation, then, is that you must plan for a minimum of two physical servers, even when using partitioned System p5 hardware. However, you can then define two or more 2-node clusters based on LPARs, with the primary and standby LPARs sited on different physical servers. Also, as shown at left, note how each physical server has one primary and one standby LPAR. This protects against the total loss of both primary and standby LPARs for a single cluster in the event of a single server failure.

The configuration shown above implies LPARs of fixed size, so it still does not address the issue of having idle processing resources under normal conditions or the alternative scenario, where the "idle" capacity on the backup node is used to run other production workloads. While this would seem to address the problem of wasted compute power, it often creates a problem when a failure occurs which causes the HACMP software to transfer the workload from the failed node to the standby node. If insufficient capacity has been reserved on the standby node, the HACMP software may not be able to run effectively in order to complete the failover.

Recall that System p5 hardware allows dynamic reallocation of processing resources across LPARs, and the larger System p5 servers (System p5 550 and above) also have Capacity on Demand capability. The HACMP product, which delivers HA clustering on pSeries hardware running the AIX operating system, supports dynamic LPAR and CoD during a failover. The HACMP software uses these technologies to "grow" the standby LPAR on the fly to the size needed to run the production workload, either through reallocation of active resources from other LPARs on the backup server or by tapping CoD to temporarily activate CPU and memory resources that are applied to the backup LPAR.

Figure 16 - HACMP Software's Use of Dynamic LPAR and CoD to Grow Backup LPAR at Failover



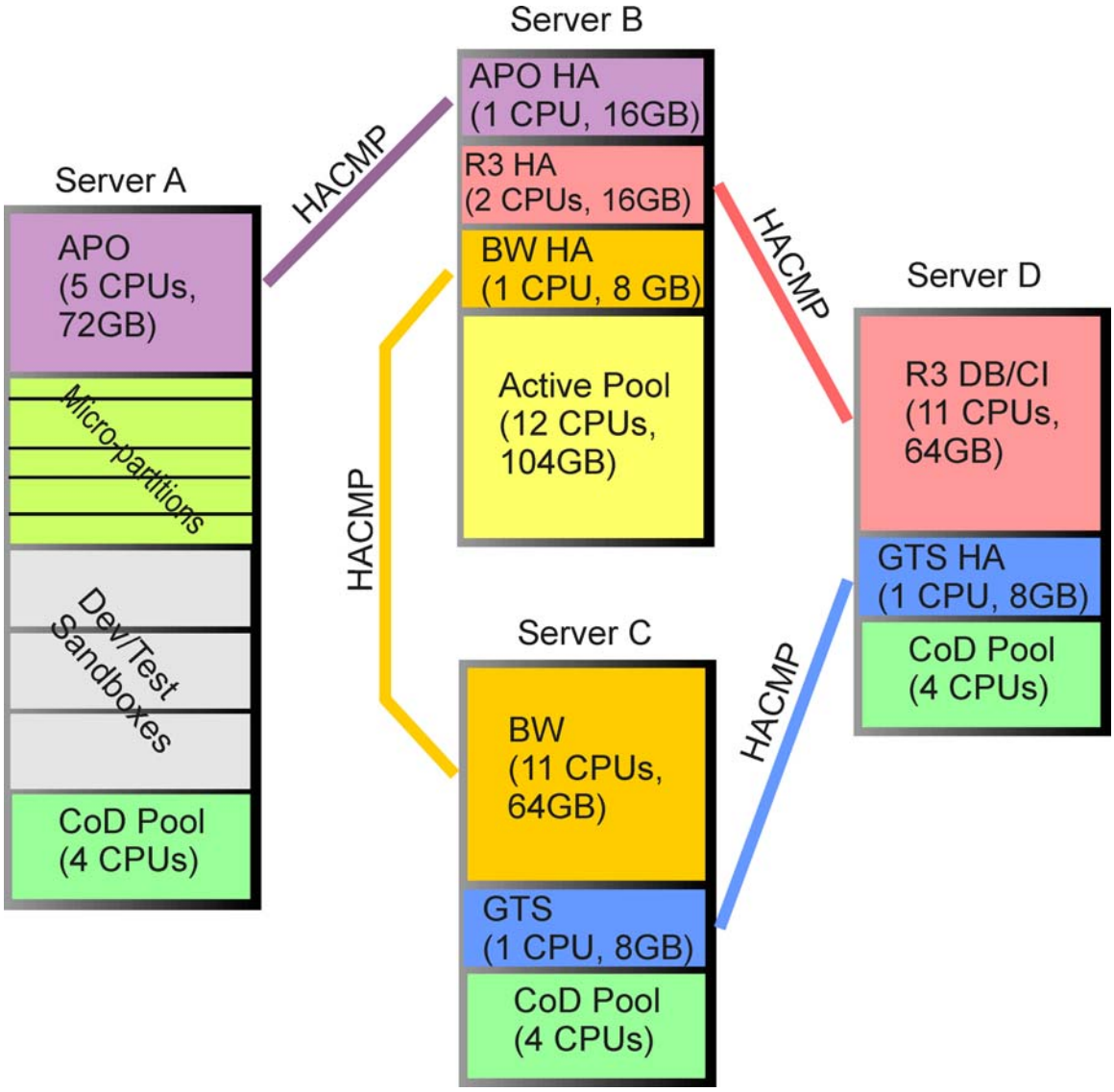
Once the failure on the production system has been corrected, the HACMP software then moves the workload back and reallocates the resources used by the standby LPAR back to their original starting points. In the case where the HACMP software is configured to pull resources from an active LPAR, care must be taken to ensure that the affected LPAR has been set up to run a lower-priority workload and that users of that workload are aware that they may likely see degraded performance during the period of time that the high priority workload is running on the standby LPAR.

By evaluating the workloads to be run on your servers and mapping out LPAR configurations where lower-priority (non-critical) workloads are placed in LPARs on servers alongside business-critical workloads, and by taking advantage of the HACMP software's support for dynamic LPAR and CoD, you can achieve higher rates of utilization while consolidating server footprints, all without compromising the availability of your critical workloads.

Make no mistake; careful planning and design for HA clustering in your enterprise is every bit as important as basic infrastructure design. You must make sure single points of failure are addressed, and ensure that production HA clusters are spread across servers, even if the cluster "nodes" are implemented as LPARs.

Now, let's take a look at a larger HA clustering example, based on a real client implementation. For clarity, IP network and storage connections have been omitted. Assume that each LPAR participating in an HACMP cluster has at least two IP network and storage connections.

Figure 17 - HACMP Clusters Implemented Across Multiple Partitioned System p5 Servers



While there is a lot going on in the diagram above, you will see that it adheres to principles presented in this document:

- All production HA clusters are spread across physical servers. The failure of a single server does not take down both the primary and standby LPARs of any HACMP cluster.
- All clusters take advantage of dynamic capacity allocation. Servers, A, C, and D have Capacity on Demand pools, which can be used by the HACMP software to "bulk up" the standby LPARs in the event of a failover
- Note that Server B does not have a CoD pool; instead, a large "active pool" is defined that can be tapped by the HACMP software for any of the R3, BW, or APO standby LPARs. This was done so that the failover times would be as quick as possible. By having active vs. inactive resources available, time is saved by not needing the HACMP software to activate required resources via CoD. In this

particular case, the client deemed quicker failover times to be worth the cost of having active resources sitting idle.

- In this configuration, all four HACMP clusters are simple two-node, hot-standby configurations.

Considerations for Application Availability

One might get the impression that software like the HACMP product is beneficial mainly in the realm of protecting against infrastructure failures (servers, networks, storage, and operating systems). Such an attitude neglects the importance of planning and configuring for application availability in HACMP clusters, and experience has shown that using HACMP software only to handle infrastructure failures is insufficient to deliver the highest level of application availability. Additionally, many applications and middleware products are now building in some level of high availability, leading the users of these products to incorrectly conclude that those features are good enough to insure high availability generally. What is actually required to maximize availability is the use of complementary HA technologies throughout the hardware/software stack, from built-in redundancy in the hardware, operating system technologies designed to support availability, through HA clustering software such as the IBM HACMP product, and including built-in availability features in middleware and applications.

A continuing focus area of HACMP enhancements is in the area of application availability. The HACMP product provides facilities that allow the cluster administrator to define application servers and application monitors. Through the use of these facilities, HACMP software can detect application-level software failures and take action to alleviate the problems. An application availability reporting tool is also provided so that the administrator can gauge the uptime of applications managed by HACMP software. The recently introduced concept of resource group dependencies allows the administrator to define relationship between resource groups. Instead of writing custom scripts to insure, for example, that an application failover will also take action to restore the application's connection to a back-end database, the application and database can be defined in resource groups with a dependency relationship indicated between the two.

Apart from discussing specific features in the HACMP product that are designed to facilitate application availability, there are a number of key areas that must be considered when incorporating application availability management into an HACMP cluster.

- Automation - making sure your applications start and stop without user intervention
- Dependencies - knowing what factors outside the HACMP product affect the applications
- Interference - knowing that applications themselves can hinder proper function of the HACMP software
- Robustness - choosing strong, stable applications
- Implementation - using appropriate scripts, file locations, and cron schedules.

It is considerably beyond the scope of this paper to discuss these areas in detail. Please refer to the Useful Information section of the Introduction for further reading on this important subject.

IT Processes, Maintenance, and Testing in an HA Environment

In addition to the IT process points mentioned earlier in this document, there are points to consider that are specific, or especially important to, an HA clustered environment.

- Insure that IT processes address the specifics of the HA cluster environment, and schedule regular process drills that include tests of HA-related processes and operational techniques.
- Clearly communicate to operations staff and user community the unique requirements and behaviors of an HA clustered environment.
- Make sure that administrators, managers and users are fully trained as to how the clustered environment operates, and set realistic expectations as to what failures the environment will and will not protect against.
- Limit "superuser" access to clustered systems to specifically appointed HA cluster specialists in IT operations organization. Doing this reduces the chance that an operator not trained to manage the HA cluster environment will accidentally make a change detrimental to the operation of the cluster.

Testing and maintenance practices are very important to the successful operation and stability of a typical IT environment. This importance grows significantly when the environment includes HA clustering. It is all too easy to cite cases where IT shops try to skimp on robust processes as well as stringent testing and maintenance practices, only to pay the price in increased numbers and durations of outages. In these cases, blame is often placed on the HA clustering software, but upon examining the outages, the cause is almost certainly due to changes applied that were not properly tested in the clustered environment, applied inconsistently across the HA cluster nodes, or applied outside of established change processes and roles.

- Following from the recommendation to standardize on a small number of cluster configurations, you must maintain HA test clusters that can be used to verify the efficacy of fixes and upgrades in the clustered environment before consideration is given to applying them to production clusters. Ideally, the test clusters should exactly mimic the production environment; however, this is very often cost-prohibitive. At minimum, maintain test clusters that match the most critical aspects of their production counterparts.
- Insure that a stringent cluster test strategy is developed and used consistently for every change (fix, upgrade, or configuration change).
- The HACMP software provides a cluster test tool that can be set up to run off-hours and log its results for later review. This tool should be run in the test environment after any change is applied.
- While the goal of cluster maintenance should be to keep software levels on cluster nodes identical across the cluster, it is often impractical to institute a change to the cluster en masse. The HACMP product provides facilities to allow rolling migrations and upgrades as well as dynamic cluster reconfiguration. This allows HACMP clusters to run in mixed mode while individual nodes are being upgraded, and additionally enables some cluster changes to take place without forcing a stop and restart of the cluster services.
- Recognize that cluster maintenance is not completely concurrent. That is, there will be changes required that will in turn necessitate the restart of the cluster

services. While workloads can be moved from the node undergoing the change to another node, keep in mind that these movements are visible as brief service interruptions. For non-concurrent changes, schedule once-or-twice yearly maintenance windows, and communicate your maintenance schedules to operations staff and users.

- Plan for maintenance of the cluster software; do not assume that, once the cluster environment is deployed that you can stay on that same release for long periods. A common outcome of a "set it and forget it" approach to HA cluster software is that the old release falls out of support, and if a failure then occurs, the outage is lengthened because support cannot easily be obtained for the cluster software. New HACMP releases typically ship once per year. While new functions are provided in each subsequent release, these releases are also used as vehicles to deliver the latest service updates to the product.
- Make use of the AIX 5L V5.3 operating system's Service Update Management Assistant (SUMA) to automate the discovery and download of AIX fixes and maintenance packages.

IBM Corporation
Systems and Technology Group
Route 100
Somers, New York 10589

Produced in the United States of America
March 2006