HP Serviceguard Cluster Configuration for Partitioned Systems

July 2005

Abstract

HP Serviceguard provides an infrastructure for the design and implementation of highly available HP-UX or Linux clusters that can quickly restore mission-critical application services after hardware or software failures. To achieve the highest level of availability, clusters must be configured to eliminate all single points of failure (SPOFs). This requires a careful analysis of the hardware and software infrastructure used to build the cluster. Partitioning technologies such as Superdome nPartitions, available on HP-UX 11i v2 or Linux, and the HP-UX Virtual Partitions (VPARS) present some unique considerations when utilizing them within a Serviceguard configuration. This document discusses these considerations.

Serviceguard A.11.16 is certified on HP-UX 11i v2 update 2, providing the same functionality across all platforms as found on the media dated September 2004. Serviceguard A.11.16 on HP-UX 11i v2 update 2 can be used on clusters up to 16 nodes. The nodes within a single cluster can be HP Integrity servers, HP 9000 servers, or newly supported with this release, combinations of both. Rolling upgrade to Serviceguard A.11.16 on HP-UX 11i v2 update 2 is supported from both Integrity Servers and HP 9000 Servers. For details on specific versions on each server type supported for rolling upgrade refer to the Serviceguard A.11.16 Release Notes Second Edition September 2004.

While not addressed by this white paper, related high -availability products supported on HP-UX 11i v2 include:

- Serviceguard Extension for RAC A.11.16
- Serviceguard Extension for Faster Failover A.01.00
- Serviceguard Extension for SAP B.03.11
- Enterprise Cluster Master Toolkit B.02.11
- Serviceguard Quorum Service A.02.00
- Serviceguard Manager A.04.00

Serviceguard for Linux for Integrity A.11.15 is certified on RedHat Enterprise Linux AS 3 for Itanium and SUSE Enterprise Server 8 for Intel® Itanium2® Processor Family for clusters up to 16 nodes. While this white paper refers to Superdome servers, for Linux, the nPartition configurations and restrictions apply to other HP Integrity servers with nPartition capability. Any differences for Linux are detailed in this white paper. Configurations, restrictions, etc. that are the same as HP-UX are not identified.

While not addressed by this white paper, related high -availability products supported on Linux include:

- Serviceguard Extension for SAP for Linux A.01.00
- Serviceguard for Linux Oracle® toolkit A.01.01
- Serviceguard Quorum Service A.02.00
- Serviceguard Manager A.04.00

Partition configurations

Partitioning technologies such as nPartitions and VPARS provide increased flexibility in effectively managing system resources. They can be used to provide hardware and/or software fault isolation between applications sharing the same hardware platform. These technologies also allow hardware resources to be more efficiently utilized based on application capacity requirements, and they provide the means to quickly redeploy the hardware resources should the application requirements change. Given this capability, it is natural to want to utilize these technologies when designing Serviceguard clusters. Care must be taken, however, as the use of partitioning does present some unique failure scenarios that must be considered when designing a cluster to meet specific uptime requirements.

The partitioning provided by nPartitions—available in HP-UX 11i v2—is done at a hardware level and each partition is isolated from both hardware and software failures of other partitions. VPARS partitioning is implemented at a software level. This provides greater flexibility in dividing hardware resources as shown in Figure 1.

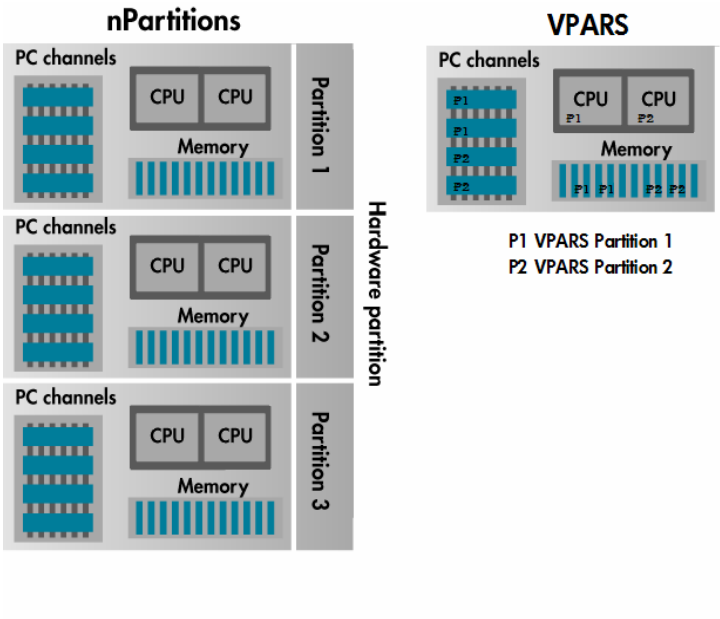


Figure 1. Sample nPartitions and VPARS configurations

VPARS and nPartitions can be combined to create a more complex configuration. This means that VPARS software partitions can be configured within the context of a hardware nPartition. Figure 2 illustrates an example of this configuration where hardware partition 1 contains two VPARS.

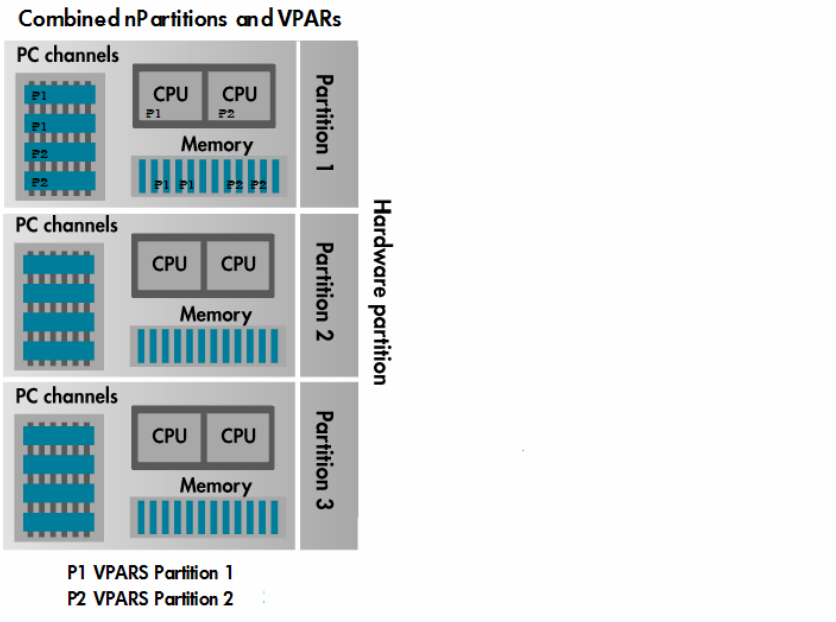


Figure 2. Sample of combined nPartitions and VPARS configurations

Serviceguard design assumptions

To best understand issues related to using partitioning within the cluster, it will be helpful to start with a review of the Serviceguard design philosophy and assumptions.

Hardware redundancy

Serviceguard, like all other high-availability (HA) clustering products, uses hardware redundancy to maintain application availability. For example, the Serviceguard configuration guidelines require redundant networking paths between the nodes in the cluster. This requirement protects against total loss of communication to a node if a networking interface card fails. If a card should fail, there is a redundant card that can take over for it.

As can be readily seen, this strategy of hardware redundancy relies on an important underlying assumption: the failure of one component is independent of the failure of other components. If the two networking cards were somehow related, there could exist a failure event that would disable them both simultaneously. This represents a SPOF and effectively defeats the purpose of having redundant cards. It is for this reason that the Serviceguard configuration rules do not allow both heartbeat networks on a node to travel through multiple ports on the same multi-ported networking interface. A single networking interface card failure would disable both heartbeat networks.

Cluster membership protocol

This same philosophy of hardware redundancy is reflected in the clustering concept. If a node in the cluster fails, another node is available to take over applications that were active on the failed node. Determining with certainty which nodes in the cluster are currently operational is accomplished through a cluster membership protocol whereby the nodes exchange heartbeat messages and maintain a cluster *quorum*.

After a failure that results in loss of communication between the nodes, active cluster nodes execute a cluster re-formation algorithm that is used to determine the new cluster quorum. This new quorum, in conjunction with the previous quorum, is used to determine which nodes remain in the new active cluster.

The algorithm for cluster re-formation generally requires a cluster quorum of a strict majority—more than 50% of the nodes that were previously running. However, exactly 50% of the previously running nodes are allowed to re-form as a new cluster, provided there is a guarantee that the other 50% of the previously running nodes do not also re-form. In these cases, some form of quorum arbitration or tie-breaker is needed. For example, if there is a

communication failure between the nodes in a two-node cluster and each node is attempting to re-form the cluster, Serviceguard must only allow one node to form the new cluster. This is accomplished by configuring a *cluster lock* or *quorum service*.

The important concept to note here is that if more than 50% of the nodes in the cluster fail at the same time, the remaining nodes have insufficient quorum to form a new cluster and fail themselves. This is irrespective of whether or not a cluster lock has been configured. It is for this reason that cluster configuration must be carefully analyzed to prevent failure modes that are common among the cluster nodes. One example of this concern is the power circuit considerations that are documented in *HP 9000 Enterprise Servers Configuration Guide*, Chapter 6 and in the *Serviceguard for Linux Order and Configuration Guide*. Another area where it is possible to have a greater than 50% node failure is in the use of partitioned systems within the cluster. Configuration considerations for preventing this situation are described in the section "Partition Interactions."

Quorum arbitration

Should two equal-sized groups of nodes (exactly 50% of the cluster in each group) become separated from each other, quorum arbitration allows one group to achieve quorum and form the cluster, while the other group is denied quorum and cannot start a cluster. This prevents the possibility of split-brain activity—two sub-clusters running at the same time. If the two sub-clusters are of unequal size, the sub-cluster with greater than 50% of the previous quorum forms the new cluster and the cluster lock is not used.

For obvious reasons, two-node cluster configurations are required to configure some type of quorum arbitration. By definition, failure of a node or loss of communication in a two-node cluster results in a 50% partition. Due to the assumption that nodes fail independently of each other (independent failure assumption), the use of quorum arbitration for cluster configurations with three or more nodes is optional, though highly recommended.

There are several techniques for providing quorum arbitration in Serviceguard clusters:

- On HP-UX 11iV2 through a cluster *lock disk* which must be accessed during the arbitration process. The cluster lock disk is a disk area located in a volume group that is shared by all nodes in the cluster. Each sub-cluster attempts to acquire the cluster lock. The sub-cluster that gets the cluster lock forms the new cluster and the nodes that were unable to get the lock cease activity. A cluster lock disk can be used in Serviceguard clusters of up to four nodes.
- On Linux through a Lock LUN which must be accessed during the arbitration process. The Lock LUN is a logical Unit, usually a "disk" defined in an Array that is shared by all nodes in the cluster. Each sub-cluster attempts to acquire the Lock LUN. The sub-cluster that gets the Lock LUN forms the new cluster and the nodes that were unable to get the lock cease activity. A Lock LUN can be used in Linux Serviceguard clusters of up to four nodes.
- Through an *arbitrator node* which provides tie breaking when an entire site fails, as in a disaster scenario. An arbitrator node is a cluster member typically located in a separate data center. Its main function is to increase the Serviceguard cluster size so that an equal partition of nodes is unlikely between production data centers. This can be used in Serviceguard clusters running HP-UX or Linux.
- Through a *quorum service*, for Serviceguard clusters of any size or type. Quorum services are provided by a quorum server process running on a machine outside of the cluster. The quorum server listens to connection requests from the Serviceguard nodes on a known port. The server maintains a special area in memory for each cluster, and when a node obtains the cluster lock, this area is marked so that other nodes will recognize the lock as "taken." A single quorum server running on either HP-UX or Linux can manage multiple HP-UX and Linux Serviceguard clusters.

Partition interactions

With this background in mind, we next need to examine to what extent the partitioning schemes either meet or violate the *independent failure assumption*.

The partitioning provided by *nPartitions* is done at a hardware level, and each partition is isolated from both hardware and software failures of other partitions. This provides very good isolation between the OS instances running within the partitions. In this sense, nPartitions meets the assumption that the failure of one node (partition) will not affect other nodes. However, within the Superdome infrastructure and other servers supporting nPartitions, there exists a very small possibility of a failure that can affect all partitions within the cabinet. So, to the extent that this infrastructure failure exists, nPartitions violates the independent failure assumption. However, depending on the specific configuration, nPartitions can be used within a Serviceguard cluster.

The VPARS form of partitioning is implemented at a software level. While this provides greater flexibility in dividing hardware resources between partitions and allows partitioning on legacy systems, it does not provide any isolation of hardware failures between the partitions. Thus the failure of a hardware component being used by one partition can bring down other partitions within the same hardware platform. From a software perspective, VPARS provides isolation for most software failures, such as kernel panics, between partitions. Due to the lack of hardware isolation however, there is no guarantee that a failure, such as a misbehaving kernel that erroneously writes to the wrong memory address, will not affect other OS partitions. Based on these observations, one can conclude that VPARS will violate the independent failure assumption to a greater degree than will nPartitions.

In addition to the failure case interactions, VPARS exhibit a behavior that should also be considered when including a VPARS as a node in a Serviceguard cluster. Due to the nature of the hardware/firmware sharing between VPARS, it is possible for one partition to induce latency in other partitions. For example, during bootup, when the booting partition requests the system firmware to initialize the boot disk, it is possible for other partitions running in the same machine to become blocked until the initialization operation completes. During Serviceguard qualification testing, delays of up to 13 seconds have been observed on systems with a PCI bus and SCSI disks. The ramifications of this type of latency are discussed in the section "Latency Considerations".

Cluster configuration considerations

Using the information from the preceding sections, we can now assess any impacts or potential issues that arise from utilizing partitions (either nPartitions or VPARS) as part of a Serviceguard cluster. From a Serviceguard perspective, an OS instance running in a partition is not treated any differently than OS instances running on non-partitioned nodes. Thus, partitioning does not alter the basic Serviceguard configuration rules as described in *HP 9000 Enterprise Servers Configuration Guide*, Chapter 6 and the *Serviceguard for Linux Order and Configuration Guide*. Details can be obtained through your local HP Sales Representative.

An example of these existing configuration requirements is the need to have dual communication paths to both storage and networks. The use of partitioning does, however, introduce interesting configuration situations that necessitate additional configuration requirements. These are discussed below.

Quorum arbitration requirements

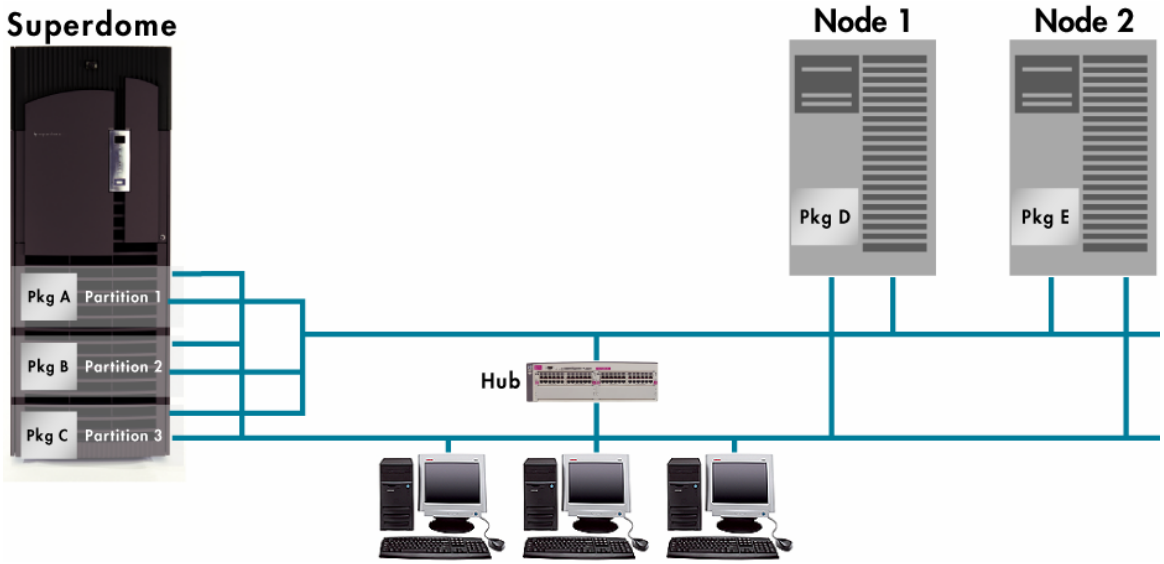
As previously mentioned, existing Serviceguard configuration rules for non-partitioned systems *require* the use of a cluster lock only in the case of a two-node cluster. This requirement is in place to protect against failures that result in a 50% quorum with respect to the membership prior to the failure. Clusters with more than two nodes do not have this as a strict requirement because of the independent failure assumption. However, this assumption is no longer valid when dealing with partitions. Cluster configurations that contain OS instances running within a partition must be analyzed to determine the impact on cluster membership based on complete failure of hardware components that support more than one partition.

Rule 1. Configurations containing the potential for a loss of more than 50% membership resulting from a single failure are not supported. These include configurations with the majority of nodes as partitions within a single hardware cabinet. This implies that when there are two cabinets, the partitions must be symmetrically divided between the cabinets.

For example, given three systems as shown in figure 3, creating a five-node cluster with three nPars (or hard partitions) in one and no partitioning in each of the other systems would not be supported because the failure of the partitioned system would represent the loss of greater than 50% of quorum (3 out of 5 nodes). Alternatively, the cluster would be supported if the systems without nPartitions each contained two VPARS, resulting in a seven-node cluster.

Exception: All cluster nodes are running within partitions in a single cabinet (such as the so-called *cluster in a box* configuration). The configuration is supported as long as users understand and accept the possibility of a complete cluster failure. This configuration is discussed in the section "Cluster in a box."

Figure 3. Unsupported configuration

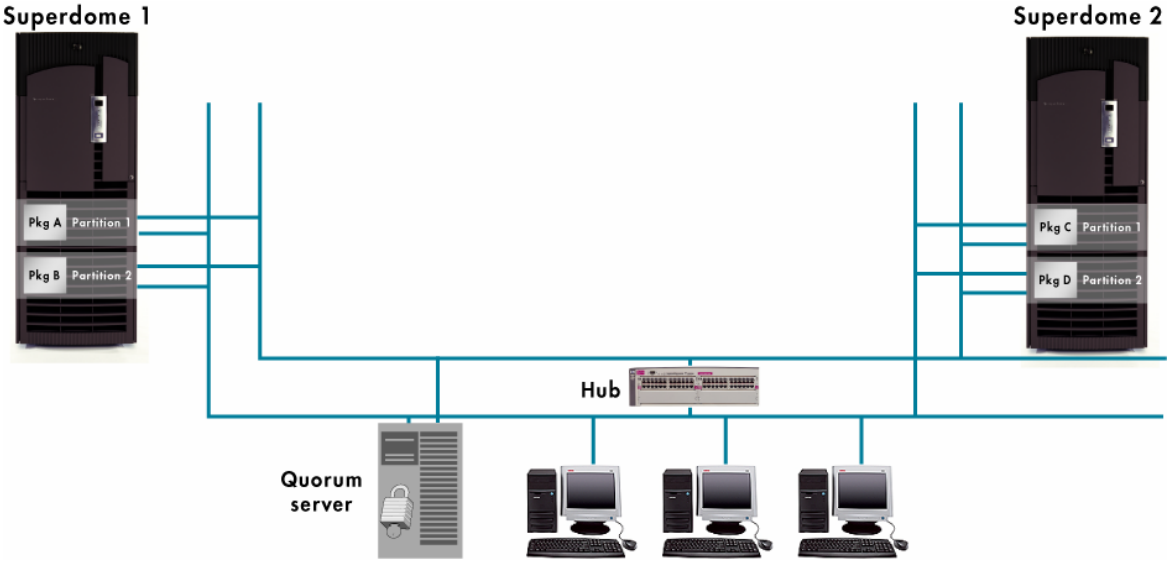


Rule 2. Configurations containing the potential for a loss of exactly 50% membership resulting from a single failure require the use of quorum arbitration. This includes:

- Cluster configurations where the nodes are running in partitions that are wholly contained within two hardware cabinets
- Cluster configurations where the nodes are running as VPARS partitions that are wholly contained within two nPartitions.

For example, to be supported, a four-node cluster consisting of two nPartitions in each of two Superdome cabinets would require a quorum arbitration device. In figure 4, there are two Superdomes, each with two partitions. Each partition is running one package. In this example the quorum arbitration is provided by a quorum server.

Figure 4. A 4-node cluster from two Superdomes with quorum server



Cluster configuration and partitions

Given the configuration requirements described in Rule 1 and Rule 2, a few interesting observations can be made of clusters utilizing partitioning:

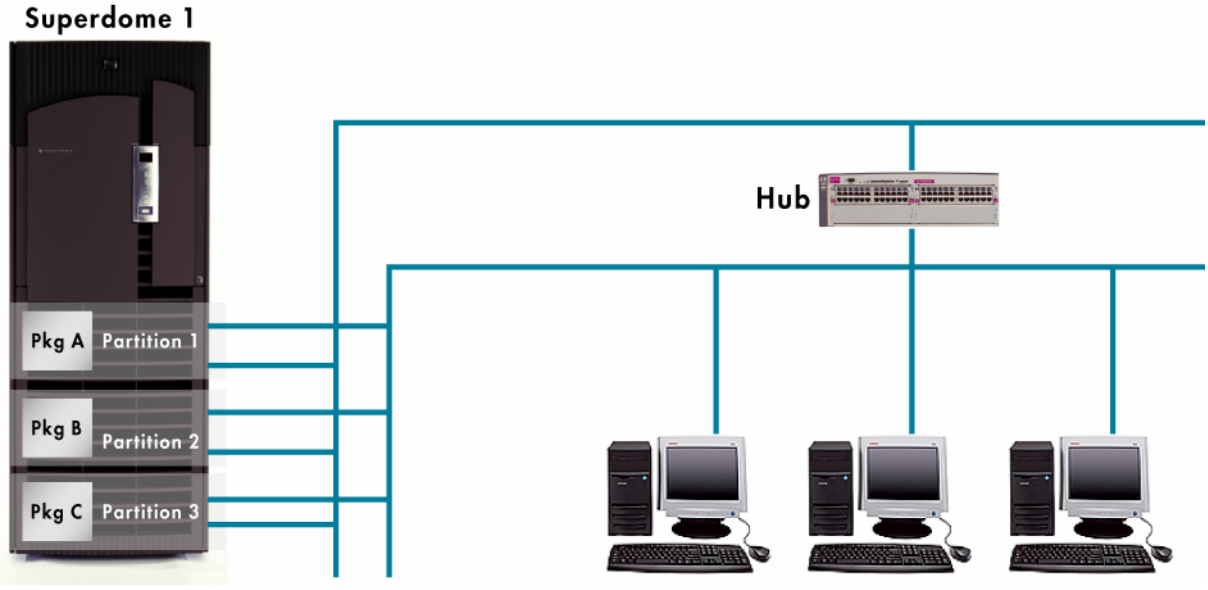
- If it is determined that a cluster lock is needed for a particular configuration, the cluster must be configured so the cluster lock is isolated from failures affecting the cluster nodes. This means that the lock device must be powered independently of the cluster nodes (such as hardware cabinets containing the partitions that make up the cluster).
- Clusters wholly contained within two hardware cabinets and that utilize the cluster lock disk for quorum arbitration are limited to either two or four nodes. This is due to a combination of the existing Serviceguard rule that limits support of the cluster lock disk to four nodes and Rule 1.
- Cluster configurations can contain a mixture of VPARS, nPartitions, and independent nodes as long as quorum requirements are met.
- For a cluster configuration to contain no single points of failure, it must extend beyond a single hardware cabinet, and comply with both the quorum rules and the Serviceguard configuration rules described in *HP 9000 Enterprise Servers Configuration Guide*, Chapter 6 and the *Serviceguard for Linux Order and Configuration Guide*.

Cluster in a box

One unique possible cluster configuration enabled by partitioning is the so-called cluster in a box. In this case, all the OS instances (nodes) of the cluster are running in partitions within the same hardware cabinet. While this configuration is subject to single points of failure, it may provide adequate availability characteristics for some applications and is thus considered a supported Serviceguard configuration. Users must carefully assess the potential impact of a complete cluster failure on their availability requirements before choosing to deploy this type of cluster configuration.

A cluster in-a-box configuration consisting exclusively of VPARS is susceptible to a wider variety of possible failures, that could result in a complete cluster failure, than is a cluster made up exclusively of nPartitions.

Figure 4. Cluster-in-a-box configuration



I/O considerations

Serviceguard does not treat OS instances running in a partition any differently than those running on an independent node. Thus, partitions do not provide any exemptions from the normal Serviceguard connectivity rules

(such as redundant paths for heartbeat networks, and to storage) nor do they impose any new requirements. There are a couple of interesting aspects related to partitioned systems that should be noted:

- While not strictly a "partitioning" issue, the Superdome platform that supports nPartitions contains its interface cards in an I/O chassis, and there can be more than one I/O chassis per partition. Since the I/O chassis represents a potential unit of failure, the nPartitions redundant I/O paths must be configured in separate I/O chassis. Generally speaking, Superdome provides enough I/O capacity that Serviceguard's redundant path requirement should not constrain the use of partitioning within the cluster.
- VPARS on the other hand must share essentially one node's worth of I/O capacity. In this case, the redundant path requirement can be a limiting factor in determining the number of partitions that can be configured on a single hardware platform.

For example, assume we would like to create a cluster-in-a-box configuration using a Fibre Channel–based storage device. The redundant path requirement means that each partition would need two Fibre Channel interface cards for storage. Each partition would also need a minimum of two network interface cards for the heartbeat LANs. Assuming that combination Fibre Channel/network cards are not used, each partition would require a minimum of four interface cards. To support a 2 partition cluster-in-a-box the system would need to have a total of eight I/O slots.

The use of "combination" cards that combine both network and storage can help in some situations. However, redundant paths for a particular device must be split across separate interface cards (for example, using multiple ports on the same network interface card for the heartbeat LANs is not supported).

Latency considerations

As mentioned previously, there is a latency issue, unique to VPARS that must be considered when configuring a Serviceguard cluster to utilize VPARS.

There are certain operations performed by one partition (such as initializing the boot disk during bootup) that can induce delays in other partitions on the same hardware platform. The net result to Serviceguard is the loss of cluster heartbeats if the delay exceeds the configured NODE_TIMEOUT parameter. If this should happen, the cluster starts the cluster re-formation protocol and, providing the delay is within the failover time, the delayed node simply rejoins the cluster. This results in cluster re-formation messages appearing in the syslog(1m) file along with diagnostic messages from the Serviceguard cluster monitor (cmcld) describing the length of the delay.

For this reason, it is recommended that clusters containing nodes running in a VPARS partition, be carefully tested using representative workloads to determine the appropriate NODE_TIMEOUT parameter that eliminates cluster reformations caused by VPARS interactions. NOTE: This does not eliminate the cmcld diagnostic messages that record delays of greater than three seconds.

Other Linux Differences

There are some restrictions listed in this document that are considered strong recommendations for Linux configurations. If these restrictions are violated then some failures will cause the failure of a node when only an interface or network card has failed. For example, Serviceguard for Linux will allow the use of just one dual channel Fibre Channel interface cards for storage connectivity as long as the customer is willing to accept that the failure of this card will cause the entire server to fail.

Serviceguard for Linux does not require that redundant I/O paths be configured in separate I/O chassis although it is strongly recommended. If redundant paths are configured in a single I/O chassis, then failure of that chassis will result in the failure of the server.

Summary and conclusion

With careful consideration of hardware redundancy, elimination of single points of failure, use of arbitration (as needed), and appropriate I/O and networking configuration, Superdome with either HP-UX 11i v2 or Linux and Serviceguard provide you with great protection against unavailable software and hardware.

HP-UX 11i release names and release identifiers

With HP-UX 11i, HP delivers a highly available, secure, and manageable operating system that meets the demands of end-to-end Internet-critical computing. HP-UX 11i supports enterprise, mission-critical, and technical computing environments. HP-UX 11i is available on both PA-RISC systems and Itanium-based systems.

Each HP-UX 11i release has an associated release name and release identifier. The uname (1) command with the -r option returns the release identifier. The following table shows the releases available for HP-UX 11i.

Table 1. HP-UX 11i releases

Release name	Release identifier	Supported processor architecture
B.11.11	HP-UX 11i v1	PA-RISC
B.11.20	HP-UX 11i v1.5	Intel Itanium
B.11.22	HP-UX 11i v1.6	Intel Itanium
B.11.23	HP-UX 11i v2 update 2	Intel Itanium and PA-RISC

Linux support

Serviceguard for Linux is supported on various enterprise version distributions from Red Hat and SUSE. This support is detailed in the *HP Serviceguard for Linux Certification Matrix*. The *Serviceguard for Linux Order and Configuration Guide* has configuration restrictions for Linux as well as some examples. Both are available on the HP website. http://www.hp.com/go/sglx

07/05

invent