Tips for implementing PowerHA in a virtual I/O environment

Chris Gibson February 16, 2010

In this article, get tips on implementing PowerHA in a virtual I/O environment. Take a look at the design and layout for a simple two-node PowerHA cluster, and understand why the virtual network configuration is an important aspect of the PowerHA configuration.

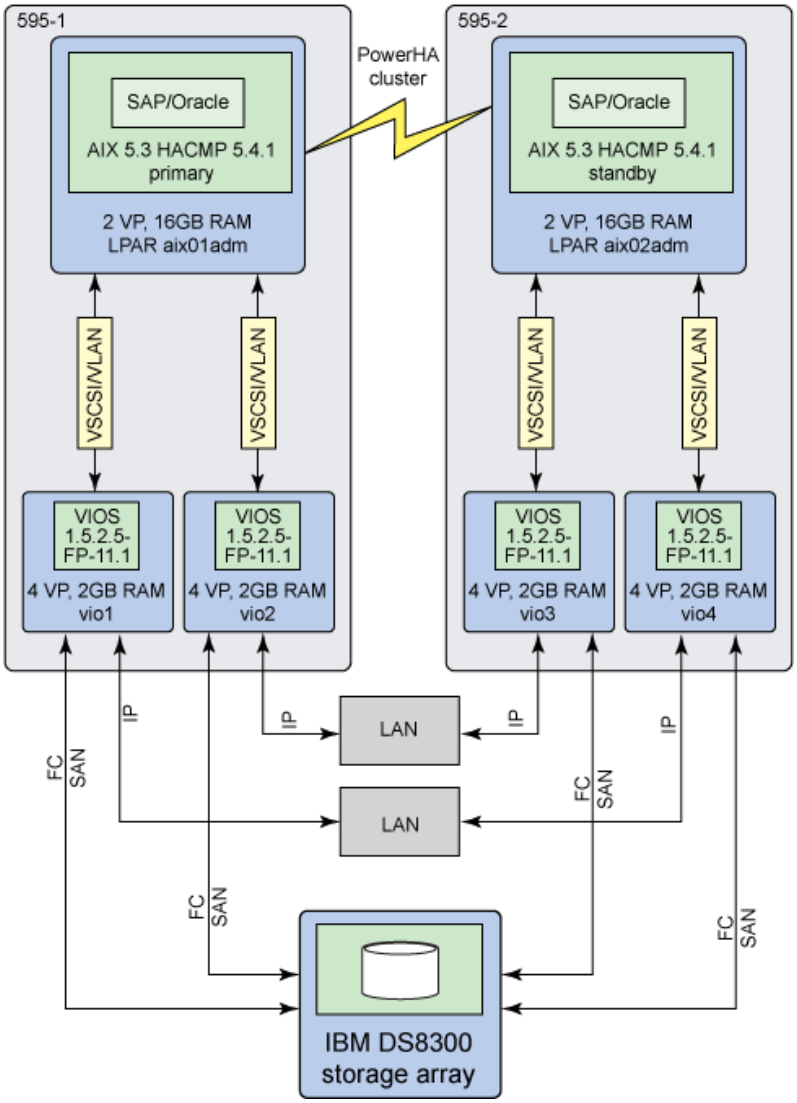
Introduction

In this article, I'll share a few of my tips for building a PowerHA™ cluster within a virtual I/O (VIO) environment. I'll briefly describe an LPAR and VIO server (VIOS) design and layout for a simple two-node PowerHA cluster. However, I won't go into specific PowerHA configuration, as that topic is too large to cover in detail here. For in-depth information, I'll refer you to the official IBM PowerHA documentation (see Related topics). This article also assumes that you have experience with AIX®, VIO, and PowerHA.

Overview

The example environment covered by this article consists of two POWER6® 595 servers. Each 595 is configured with dual VIO servers for redundancy, and a two-node cluster has been built across the two physical frames, that is, one PowerHA node resides on each Power 595 server. The LPARs are running AIX 5.3 TL7 SP5 with PowerHA 5.4.1.3. Each VIOS was built with version 1.5.2.1-FP11.1 across the virtual I/O landscape. Figure 1 shows this configuration.

Figure 1. PowerHA cluster overview



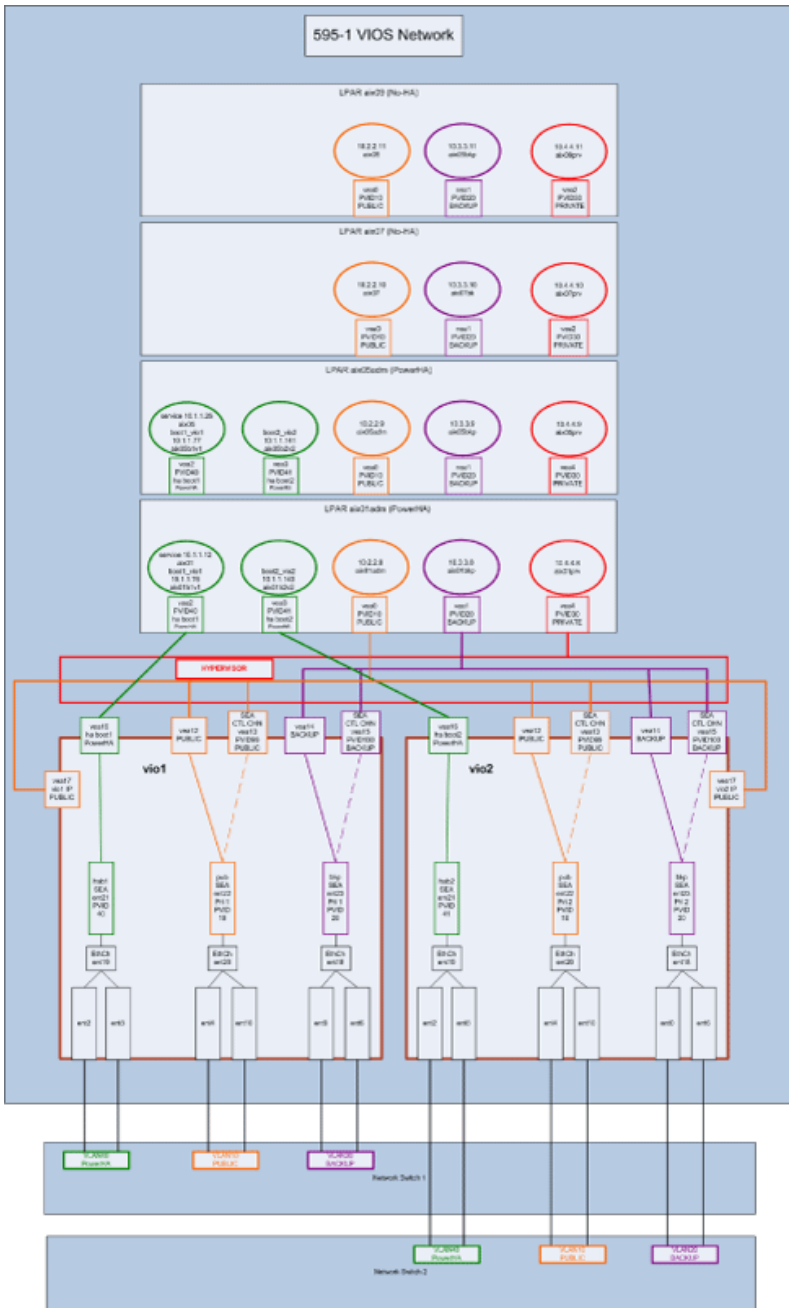
In the following sections, I will briefly touch on the virtual network and virtual (shared) storage configuration for the cluster nodes. In particular, I will highlight these areas:

- PowerHA boot and service network and addresses
- Shared Ethernet Adapter (SEA) configuration for the PowerHA network
- Shared volume group considerations

Virtual network

The virtual network configuration is an important aspect of the PowerHA configuration. Figure 2 shows how the VIOS network is configured; in this example, on one 595 frame. The VIOS network configuration is duplicated on the second frame. (Click to view a larger image.)

Figure 2. VIOS network overview



As shown in Figure 2, there are PowerHA and non-HA LPARs as clients of the same VIOS pair. You'll also notice multiple SEAs, that is, one per VLAN and usage type: *PUBLIC*, *BACKUP*, and *PowerHA*. Each VLAN has a unique IP range: *PUBLIC* 10.2.2, *BACKUP* 10.3.3 and *PowerHA* 10.1.1. There's also an interface on each LPAR, on the 10.4.4 network that is used for internal (private) communication between the LPARs over the POWER Hypervisor virtual network.

The HA nodes communicate with the outside world through VLAN40 (PVID40/41), which is the *PowerHA* network. The non-HA LPARs communicate through VLAN10 (PVID10), over the *PUBLIC*

network. There's also another SEA in each VIOS, on VLAN20, which is used as a dedicated VLAN for backups over the network, hence the network name *BACKUP*.

Shared Ethernet Adapter failover (SEA FO) is configured for both the PUBLIC and BACKUP networks. There is no SEA FO for the PowerHA network. If a SEA fails on a VIOS, for the PowerHA network, then the service IP will move to the other boot adapter, served by the redundant VIOS.

There's no VLAN tagging in use for any of the SEAs. There's no need, as there is only a handful of VLANs to deal with in this network. However, your requirements may differ.

When viewing the PowerHA cluster network, with the `cltopinfo` command, the Network definitions on each node are as follows:

Listing 1. Network definitions

```
# cltopinfo
Cluster Name: CLUSTER-A
Cluster Connection Authentication Mode: Standard
Cluster Message Authentication Mode: None
Cluster Message Encryption: None
Use Persistent Labels for Communication: No
There are 2 node(s) and 3 network(s) defined

NODE aix01adm:
        Network net_diskhb_01
                aix01adm_hdisk1_01    /dev/hdisk1
        Network net_ether_01
                aix01adm  10.2.2.8
        Network net_ether_02aix01   10.1.1.12
                aix01b2v1  10.1.1.76
                aix01b1v2  10.1.1.140

NODE aix02adm:
        Network net_diskhb_01
                aix02adm_hdisk1_01    /dev/hdisk1
        Network net_ether_01
                aix02adm  10.2.2.15
        Network net_ether_02
                aix01   10.1.1.12
                aix02b1v3  10.1.1.77
                aix02b2v4  10.1.1.141

Resource Group HARG1
        Startup Policy  Online On Home Node Only
        Fallover Policy Fallover To Next Priority Node In The List
        Fallback Policy Never Fallback
        Participating Nodes aix01adm aix02adm
        Service IP Label aix01
```

As you can see, the service and boot adapters are all in the same subnetted (segmented) IP network, where `b1v1` defines the first boot adapter (b1) associated with the first VIOS (v1) and so on. The service address is the `hostname` without `adm` appended to it.

Listing 2. Service and boot adapters

```
Service address: aix01  10.1.1.12
       Netmask 255.255.255.192
       IP range 10.1.1.1 - 62

boot1 address:  aix01b1v1  10.1.1.76
       Netmask 255.255.255.192
       IP range 10.1.1.65 - 126

boot2 address:  aix01b2v2  10.1.1.140
       Netmask 255.255.255.192
       IP range 10.1.1.129 - 190

boot1 address:  aix02b1v3  10.1.1.77
       Netmask 255.255.255.192
       IP range 10.1.1.65 - 126

boot2 address:  aix02b2v4  10.1.1.141
       Netmask 255.255.255.192
       IP range 10.1.1.129 - 190
```

Typically, when configuring a SEA on a VIOS, you would deploy SEA Fail Over to ensure network connectivity was protected in the event of a VIOS failure. However, in this PowerHA environment, the approach is different. SEA FO is not used for the PowerHA network. This way, PowerHA is aware of, and controls, network failure and failover. In this case, there is one SEA for the PowerHA network in each VIOS. If a VIOS fails, the service address moves to the boot adapter served by the redundant VIOS.

The main driver for this approach is the way the PowerHA cluster communicates in a virtual network environment. If SEA FO was configured and a failure occurred, HA would have no way of detecting the failure. Likewise, if all communication at the physical layer was lost, HA would continue to think the network was OK, as it is still able to route traffic across the virtual LAN on the Hypervisor.

This is why it is important to configure the netmon.cf file on all nodes in the cluster.

This file instructs HA on how to determine when it has lost connectivity with the network or its partner HA nodes. If this file is not configured appropriately, network failures could go undetected by PowerHA.

The netmon.cf file and VIO

There are two APARS that I recommend you review in relation to configuring the netmon.cf file in a VIO environment. You'll soon understand why this file is important and when it should be implemented.

APAR IZ01331 describes the scenarios of using VIO with PowerHA clusters and the challenges faced in detecting network failures. For example, if an "*entire CEC is unplugged from the network, the PowerHA node on that Frame does not detect a local adapter down event, because traffic being passed between the VIO clients (on the same frame) looks like normal external traffic from the perspective of the LPAR's OS*."

To get around this problem, the netmon.cf file is used to allow customers to declare that a given adapter should only be considered up if it can ping a set of specified targets.

If the VIOS has multiple physical interfaces on the same network or if there are two or more PowerHA nodes using one or more VIOS in the same frame, PowerHA will not be informed of (and hence will not react to) individual physical interface failures.

In the extreme case where all physical interfaces managed by VIO Servers have failed, the VIOS will continue to route traffic from one LPAR to another in the same frame, the virtual ethernet interface used by PowerHA will not be reported as having failed, and PowerHA will not react.

Each node in the cluster has a custom netmon.cf file that lists all the IP addresses it must be able to ping for it to mark an interface *up* or *down*. For example, `aix01adm` resides on Frame 1 (595-1) and `aix02adm` resides on Frame 2 (595-2). If all network connectivity was lost for all physical interfaces on all VIOS on 595-1, then `aix01adm` would still continue functioning, as it would still be able to route packets over the virtual network. For this node (and others) to detect the problem, you populate the netmon.cf file with addresses it should be able to reach on specific interfaces. If it can't, then those interfaces are marked as down and PowerHA is able to react accordingly.

APAR IZ01874 clarifies how to choose IP addresses for the netmon.cf file. This file should contain remote IP addresses and host names that are not in the cluster configuration that can be accessed through the PowerHA network interfaces. These addresses must be preceded by `!REQD`.

Some good choices for targets are name servers (DNS servers) and gateways (routers), or reliable external IP addresses (such as NTP servers) that will respond to a ping. You can use the following ping command to verify that a ping will be answered on a specific interface:

```
# ping -S <Boot IP address> <IP addr in netmon.cf>
```

Where `<Boot IP address>` is the IP address configured on the boot interface. For example,

Listing 4. ping command response on a specific interface

```
# ping -c5 -S aix01b1v1 aix02b1v3
PING bxaix04b1v1: (10.1.1.77): 56 data bytes
64 bytes from 10.1.1.77: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 10.1.1.77: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 10.1.1.77: icmp_seq=2 ttl=255 time=0 ms
64 bytes from 10.1.1.77: icmp_seq=3 ttl=255 time=0 ms
64 bytes from 10.1.1.77: icmp_seq=4 ttl=255 time=0 ms

----aix02b1v3 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
```

Listing 5 shows some netmon.cf samples from two nodes, on two different physical frames.

Listing 5. netmon.cf samples

```
HOST: aix01adm 595-1
--------------------
# Care is required when modifying this file!
```

```
# The nature of the VIO/PowerHA environment means the contents
# of netmon.cf on each cluster node is different.
# IP labels/addresses on virtual interfaces in any VIO client LPAR
# within this server frame, must be excluded from this file!
!REQD aix01b1v1 10.2.2.1
!REQD aix01b2v2 10.2.2.1
!REQD aix01b1v1 10.1.1.1
!REQD aix01b2v2 10.1.1.1
!REQD aix01b1v2 10.1.1.77
!REQD aix01b2v2 10.1.1.141
!REQD aix01b1v1 aix02b1v3
!REQD aix01b2v2 aix02b2v4
!REQD aix01b1v1 10.1.9.2
!REQD aix01b2v2 10.1.9.3
10.2.2.1
10.1.9.2
10.1.9.3
ntp-srvr
ntp-srvr

HOST: aix02adm 595-2
--------------------
# Care is required when modifying this file!
# The nature of the VIO/PowerHA environment means the contents
# of netmon.cf on each cluster node is different.
# IP labels/addresses on virtual interfaces in any VIO client LPAR
# within this server frame, must be excluded from this file!
!REQD aix02b1v3 10.2.2.1
!REQD aix02b2v4 10.2.2.1
!REQD aix02b1v3 10.1.1.1
!REQD aix02b2v4 10.1.1.1
!REQD aix02b1v3 10.1.1.76
!REQD aix02b2v4 10.1.1.140
!REQD aix02b1v3 aix01b1v1
!REQD aix02b2v4 aix01b2v2
!REQD aix02b1v3 10.1.9.2
!REQD aix02b2v4 10.1.9.3
10.2.2.1
10.1.9.2
10.1.9.3
ntp-srvr
ntp-srvr
```

If you take one line as an example,

```
!REQD aix01b1v1 aix02b1v3
```

The `!REQD` tag specifies that the adapter (aix01b1v1) will only be considered up if it can ping the target (aix02b1v3). The `aix01b1v1` entry specifies which interface to use for the test, that is, aix01b1v1 resolves to 10.1.1.76, which is the address on the en2 interface. This interface will be considered up if it is able to ping the target, aix02b1v3.

Listing 6. Determining adapter hostname

```
# host aix01b1v1
aix01b1v1 is 10.1.1.76

# ifconfig en2
en2:
 flags=1e080863,480<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),CHAIN>
        inet 10.1.1.76 netmask 0xfffffc0 broadcast 10.1.1.127
        inet 10.1.1.12 netmask 0xfffffc0 broadcast 10.1.1.63
         tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
```

en2 will be used to connect to aix02b1v3, which is an interface on its partner node on 595-2. If it cannot communicate, the interface en2 (aix01b1v1) will be marked as down. Do not include any nodes in this file that exist on the same frame. All entries should be for systems that reside outside of the physical frame to ensure the detection of real, physical network failures to the outside world on the physical (not virtual) network.

Be careful not to specify an interface name in the netmon.cf file, such as:

```
!REQD en2 10.1.1.10
```

Including the interface name will not work in a VIO environment. The last time I checked, there was a Design Change Request (DCR) in with the HA development team to overcome this issue. Some customers have experienced a slow takeover due to the way RSCT (netmon) determines if the second field in netmon.cf is an IP/hostname or the name of an interface. In some cases, netmon will attempt to resolve the IP address of the hostname, for example, `$ host en2`, which will fail. IBM development is working on a new algorithm to prevent interface names from being treated as host names, especially for obvious formats such as enX. For now it's best to eliminate the use of the interface name, for example, en*X*, in the netmon.cf file.

It's recommended to only use the netmon.cf method if it is appropriate in your VIO environment. Using this method changes the definition of a so-called good adapter from,"Am I able to receive any network traffic?" to "Can I successfully ping certain addresses? (regardless of how much traffic I can see)".

This can make it more likely for an adapter to be falsely considered down. If you must use this new function, I recommend that you include as many targets as possible for each interface you need to monitor.

Virtual (shared) storage

The IBM technical documentation relating to PowerHA and Virtual SCSI (VSCSI) clearly defines the supported storage configuration in a VIO environment. The shared volume group (VG) must be defined as "Enhanced Concurrent Mode." In general, Enhanced Concurrent Mode is the recommended mode for sharing volume groups in PowerHA clusters. In this mode, the shared volume groups are accessible by multiple PowerHA nodes, which results in faster failover (disk takeover) in the event of a node failure. All volume group administration on these shared disks is done from the PowerHA nodes, not from the VIOS.

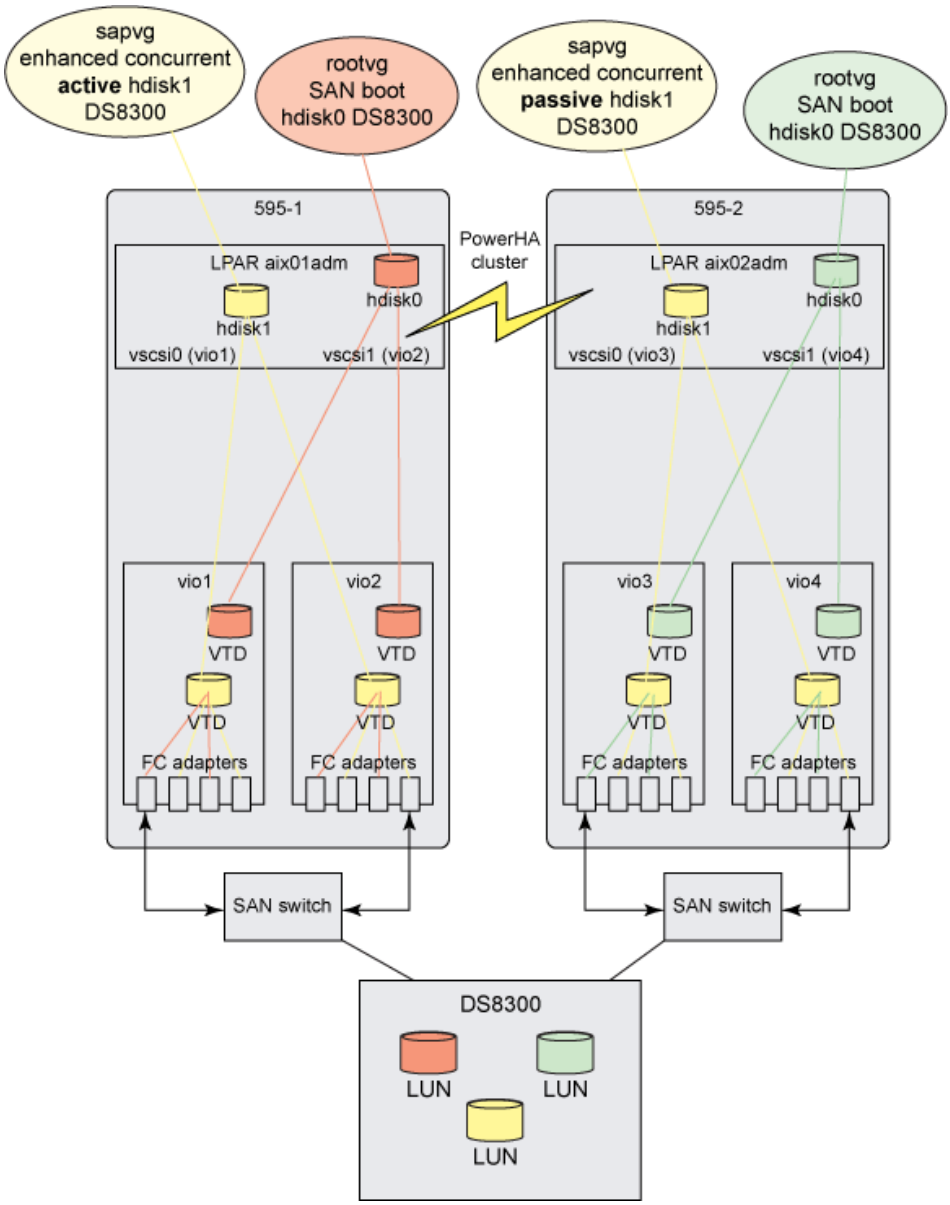
In the example environment, running `lspv` on the primary node confirms the shared volume group is in concurrent mode.

Listing 7. Running lspv on the primary node

```
root@aix01adm / # lspv
hdisk0  00c79a70a6858137          rootvg
hdisk1  00c79a70a20c321c          sapvg           concurrent
```

Figure 3 shows that there are two volume groups on each node. Each node has its own (non-shared) root volume group (`rootvg`).

Figure 3. VIOS VSCSI overview



The primary node has ownership of the shared volume group as it is varied-on and active. I can confirm this by running the `lsvg` command on the primary and taking note of some its characteristics. The `VG STATE` is active, `VG Mode` is Concurrent, `Concurrent` is set to Enhanced-Capable, and `VG PERMISSION` is read/write. The logical volumes in the shared volume group are open.

Listing 8. Running lsvg on the primary node

```
root@aix01adm / # lsvg sapvg
VOLUME GROUP:       sapvg                    VG IDENTIFIER:  00c79a6000004c0000000123a2278720VG STATE:
          active                   PP SIZE:        64 megabyte(s)
```

```
VG PERMISSION:      read/write               TOTAL PPs:      6398 (409472 megabytes)
MAX LVs:            256                      FREE PPs:       1596 (102144 megabytes)
LVs:                13                       USED PPs:       4802 (307328 megabytes)
OPEN LVs:           13                       QUORUM:         2 (Enabled)
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        no
Concurrent:         Enhanced-Capable         Auto-Concurrent: Disabled
VG Mode:            Concurrent
Node ID:            2                        Active Nodes:       1
MAX PPs per VG:     32512
MAX PPs per PV:     4064                     MAX PVs:        8
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable

root@aix01adm / # lsvg -l sapvg
sapvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
oraclelv            jfs2       192     192     1    open/syncd    /oracle
sapmnt_CG1lv        jfs2       144     144     1    open/syncd    /sapmnt
usrsap_CG1lv        jfs2       144     144     1    open/syncd    /usr/sap
oraclestagelv       jfs2       128     128     1    open/syncd    /oracle/stage
sapreorg_CG1lv      jfs2       64      64      1    open/syncd    /oracle/CG1/sapreorg
sapbackup_CG1lv     jfs2       16      16      1    open/syncd    /oracle/CG1/sapbackup
mirrlogA_CG1lv      jfs2       8       8       1    open/syncd    /oracle/CG1/mirrlogA
mirrlogB_CG1lv      jfs2       8       8       1    open/syncd    /oracle/CG1/mirrlogB
origlogA_CG1lv      jfs2       8       8       1    open/syncd    /oracle/CG1/origlogA
origlogB_CG1lv      jfs2       8       8       1    open/syncd    /oracle/CG1/origlogB
sapdata1_CG1lv      jfs2       1600    1600    1    open/syncd    /oracle/CG1/sapdata1
oraarch_CG1lv       jfs2       80      80      1    open/syncd    /oracle/CG1/oraarch
loglv01             jfs2log    1       1       1    open/syncd    N/A
```

File systems on the standby nodes are not mounted until the point of failover, so accidental use of data on standby nodes is not possible. On the standby node, it has access to the shared enhanced-concurrent volume group, but only in a passive, read-only mode. The `VG PERMISSION` is set to passive-only. The logical volumes in the shared volume group are closed.

Listing 9. Standby nodes

```
root@aix02adm / # lsvg sapvg
VOLUME GROUP:       sapvg                    VG IDENTIFIER:  00c79a6000004c0000000123a2278720
VG STATE:           active                   PP SIZE:        64 megabyte(s)
VG PERMISSION:      passive-only             TOTAL PPs:      6398 (409472 megabytes)
MAX LVs:            256                      FREE PPs:       1596 (102144 megabytes)
LVs:                13                       USED PPs:       4802 (307328 megabytes)
OPEN LVs:           0                        QUORUM:         2 (Enabled)
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        no
Concurrent:         Enhanced-Capable         Auto-Concurrent: Disabled
VG Mode:            Concurrent
Node ID:            1                        Active Nodes:       2
MAX PPs per VG:     32512
MAX PPs per PV:     4064                     MAX PVs:        8
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable

root@aix02adm / # lsvg -l sapvg
sapvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
oraclelv            jfs2       192     192     1    closed/syncd  /oracle
sapmnt_CG1lv        jfs2       144     144     1    closed/syncd  /sapmnt
usrsap_CG1lv        jfs2       144     144     1    closed/syncd  /usr/sap
oraclestagelv       jfs2       128     128     1    closed/syncd  /oracle/stage
```

```
sapreorg_CG1lv      jfs2       64      64      1    closed/syncd  /oracle/CG1/sapreorg
sapbackup_CG1lv     jfs2       16      16      1    closed/syncd  /oracle/CG1/sapbackup
mirrlogA_CG1lv      jfs2       8       8       1    closed/syncd  /oracle/CG1/mirrlogA
mirrlogB_CG1lv      jfs2       8       8       1    closed/syncd  /oracle/CG1/mirrlogB
origlogA_CG1lv      jfs2       8       8       1    closed/syncd  /oracle/CG1/origlogA
origlogB_CG1lv      jfs2       8       8       1    closed/syncd  /oracle/CG1/origlogB
sapdata1_CG1lv      jfs2       1600    1600    1    closed/syncd  /oracle/CG1/sapdata1
oraarch_CG1lv       jfs2       80      80      1    closed/syncd  /oracle/CG1/oraarch
loglv01             jfs2log    1       1       1    closed/syncd  N/A
```

The `bos.clvm.enh` fileset must be installed (on all nodes in the cluster) to support enhanced concurrent volume groups. A new subsystem (gsclvmd) is started with enhanced concurrent volume groups. You can query this subsystem to determine the active enhanced concurrent volume groups.

Listing 10. Querying the gsclvmd subsystem

```
# lssrc -s gsclvmd
Subsystem         Group            PID          Status
 gsclvmd                           327756       active

# ps -fp 462906
UID     PID    PPID   C    STIME    TTY TIME CMD
root  462906  327756   0   Nov 04   -   0:02 /usr/sbin/gsclvmd -r 30 -i 300 -t 300 -c
 00c79a6000004c0000000123a2278720 -v 0

# lssrc -ls gsclvmd
 Subsystem       Group           PID     Status
 gsclvmd         gsclvmd        327756  active

 Active VGs # 1
 vgid                                  pid
 00c79a6000004c0000000123a2278720      462906
```

To enable a shared volume group for enhanced concurrent mode (Fast Disk Takeover), you can use CSPOC.

Listing 11. Enabling enhanced concurrent mode

```
# smit cl_vg
                                                                  Shared Volume Groups

Move cursor to desired item and press Enter.

  List All Shared Volume Groups
  Create a Shared Volume Group
  Create a Shared Volume Group with Data Path Devices
  Enable a Shared Volume Group for Fast Disk Takeover
  Set Characteristics of a Shared Volume Group
  Import a Shared Volume Group
  Mirror a Shared Volume Group
  Unmirror a Shared Volume Group
```

Refer to the IBM technical documentation and PowerHA documentation for more information relating to PowerHA and virtual storage support.

Summary

This is just one approach to this type of configuration. I hope these brief tips provide you with some ideas on how to approach PowerHA in a VIO environment.

Related topics

- Read the technical documentation on PowerHA.
- This online information describes converting volume groups to enhanced concurrent mode.
- The IBM Redbook HACMP 5.3, Dynamic LPAR, and Virtualization is practical case study about HACMP V5.3, Dynamic LPAR, and Advanced POWER Virtualization on pSeries servers.
- Take a look at the announcement for PowerHA support of VIO Server 2.1.