APV QuickSheet

Version: 0.12.0
Date: 5/10/7

Compatibility note: This document is written based upon VIOS 1.3.

CPU allocations

• Shared (virtual) processor partitions can utilize additional resources from the shared processor pool when available. Dedicated processor partitions can only use above the "desired" amount if another CPU is (dynamically) added to the LPAR.
• An uncapped partition can only consume up to the number of virtual processors that it has. A capped partition can only consume up to its allocation limit. Allocations are in increments of 1/100th of a CPU, the minimal allocation is 1/10th of a CPU for each virtual CPU.
• All uncapped partitions using the shared processor pool compete for the remaining resources in the pool. Each partition can be assigned a priority of 0 to 255 that will determine which partition will get the extra CPU resources when CPU resources are limited. The default priority for this value is 128. A partition with a priority of 0 is effectively a capped partition.

Creating virtual devices (in your LPAR profile)

• Virtual Ethernet devices should only have 802.1Q enabled if you intend to run additional VLANs on that interface. (In most instances this is not the case).
• Only one interface should be configured to "Access External Networks" on a VLAN, this should be the virtual interface used for the SEA on the VIOS and not the VIOC. This is the "gateway" adapter that will recieve packets with MAC addresses that are unknown. (This is also known as a "Trunk adapter" on some versions of the HMC.)
• VIOS partitions are unique in that they can have virtual host adapters. Virtual SCSI adapters in VIOC partitions connect to LUNs shared through VIOS virtual host adapters.

VIOS Unix subsystem

• The current VIOS runs on an AIX subsystem. (VIOS functionality is availible for Linux. This QuickSheet only deals with the AIX based versions.)
• The padmin account logs in with a restricted shell. A root shell can be obtained by the `oem_setup_env` command.
• The root shell is designed for intsallation of OEM applications and drivers only. It may be required for a *small* subset of commands. (The purpose of this QuickSheet is to provide a listing of most frequent tasks and the *proper* VIOS commands so that access to a root shell is not required.)
• The restricted shell has access to common Unix utilites such as `awk`, `grep`, `sed`, and `vi`. The syntax and usage of these commands has not been changed in VIOS. (Use "`ls /usr/ios/utils`" to get a listing of available Unix commands.)
• Redirection to a file is not allowed using the standard ">" character, but can be accomplished with the "`tee`" command.
Redirect the output of `ls` to a file
```
ls | tee ls.out
```
Determine the underlying (AIX) OS version (for driver install)
```
oem_platform_level
```
Exit the restricted shell to a root shell
```
oem_setup_env
```

VIO redundancy

• VIOS fault tolerance is provided by configuring pairs of VIOS to serve redundant networking and disk access to client LPARs. In most cases (when using AIX) no additional configuration is required in the VIOC for this capability.
• Both Network and Disk redundancy methods tend to be active / passive. For example, it is not possible to run EtherChannel *within* the system to a VIOS.
• It is important to understand that the performance considerations of an active / passive configuration are not relevant inside the system as all VIOS can utilize pooled processor resources and therfore gain no significant (internal) performance benifit by active / active configurations. Performance benefits of active / active configurations are realized when used to connect to outside / physical resources such as EtherChannel (port aggregation) from the VIOS to a physical switch.

Network redundancy in VIOS

• The two *primary* methods of providing network redundancy in a dual VIOS configuration are NIB (Network Interface Backup) and SEA Failover. (These provide protection from the loss of a VIOS)
• NIB creates a link aggregation of a single virtual NIC with a backup NIC. (Each virtual NIC is connected externally by a different VIOS) This configuration is done in each client OS.
• SEA Failover is a VIOS configuration option that provides two physical network connections, one from each VIOS. No client configuration is required.
• NIB and SEA Failover are not mutually exclusive and can be used together or with link aggregation (EtherChannel / 802.3ad) to a physical device in the VIOS. (Link aggregation of more than one virtual adapter is not supported or necessary from the client.)

Disk redundancy in VIOS

• The two methods for providing disk redundancy in a dual VIOS configuration are LVM mirroring on the client LPAR or MPIO access to an external (SAN) RAID disk presented by both VIO servers.
• LVM mirroring is a client configuration that mirrors data to two *different* disks presented by two different VIOS.
• MPIO is automatically enabled in AIX if the same disk is presented to a VIOC by two different VIOS.
• LVM mirroring (for client LUNs) is `not` recommended `within` VIOS. This configuration would provide no additonal protections over LVM mirroring in the VIOC.

Low level redundancy configuration

• Management and setup of devices requiring drivers and tools not provided by VIOS (ie PowerPath devices) will require use of the root shell availible from the `oem_setup_env` command.
• The `mkvdev -lnagg` and `cfglnagg` commands can be used to set up and manage link aggregation (to external ethernet switches).
• The `chpath`, `mkpath`, and `lspath` commands can be used to manage MPIO capable devices.

Devices

Discover new devices (VIOS equivelant of `cfgmgr`)
```
cfgdev
```
List all adapters on the system
```
lsdev -type adapter
```
List only virtual adapters
```
lsdev -virtual -type adapter
```
List all virtual disks (created with mkvdev command)
```
lsdev -virtual -type disk
```
Find the WWN of the fcs0 HBA
```
lsdev -dev fcs0 -vpd | grep Network
```
List the firmware levels of all devices on the system
`lsfware -all` (The `invscout` command is also available)
List all devices (virtual and physical) by their slot address
```
lsdev -slots
```
List the port speed of the (physical) ethernet adapter eth0
```
lsdev -dev ent0 -attr media_speed
```
List all the possible settings for media_speed on ent0
```
lsdev -dev ent0 -range media_speed
```
Set the media_speed to auto negotiate on ent0
```
chdev -dev ent0 -attr media_speed=Auto_Negotiation
```
Set the media_speed to auto negotiate on ent0 on next boot
```
chdev -dev ent0 -attr media_speed=Auto_Negotiation -perm
```

VIOS Networking

Enable jumbo frames on the ent0 device
```
chdev -dev ent0 -attr jumbo_frames=yes
```
View settings on ent0 device
```
lsdev -dev ent0 -attr
```
Find the default gateway and routing info on the VIOS
```
netstat -routinfo
```
List open (TCP) ports on the VIOS IP stack
```
lstcpip -sockets | grep LISTEN
```
List TCP and UDP sockets listening and in use
```
lstcpip -sockets -family inet
```
List all (virtual and physical) ethernet adapters in the VIOS
```
lstcpip -adapters
```
Set up initial TCP/IP config (en10 is the interface for the SEA ent10)
```
mktcpip -hostname vios1 -inetaddr 10.143.181.207 \
        -interface en10 -start -netmask 255.255.252.0 \
        -gateway 10.143.180.1
```
Show interface traffic statistics on 2 second intervals
```
netstat -state 2
```
Show *verbose* statistics for all interfaces
```
netstat -cdlistats
```
Show the default gateway and route table
```
netstat -routtable
```
Change the default route on en0 (fix a typo from mktcpip)
```
chtcpip -interface en0 \
        -gateway -add 192.168.1.1 -remove 168.192.1.1
```
Change the IP address on en0 to 192.168.1.2
```
chtcpip -interface en0 \
        -inetaddr 192.168.1.2 -netmask 255.255.255.0
```

VIOS management

Shutdown the server
`shutdown` (Optionally include `-restart`)
List the version of the VIOS system software
```
ioslevel
```
List the boot devices for this lpar
```
bootlist -mode normal -ls
```
List all (AIX) packages installed on the system
`lssw` (Equivalent to `lslpp -L` in AIX)
Change the MOTD to an appropriate message
```
motd "Unauthorized access is prohibited."
```
Display a timestamped list of all commands run on the system
```
lsgcl
```
To display the current date and time of the VIOS
```
chdate
```
Change the current time and date to 1:02 AM March 4, 2005
```
chdate -hour 1 -minute 2 -month 3 -day 4 -year 2005
```
Change just the timezone to AST
`chdate -timezone AST` (Visible on next login)
• The `date` command is availible and works the same as in Unix.

User Management

• padmin is the only user for most configurations. It is possible to configure additional users, such as operational users for monitoring purposes.
List attributes of the padmin user
```
lsuser padmin
```
List all users on the system
`lsuser` (The optional parameter "`ALL`" is implied with no parameter)
Change the password for the current user
```
passwd
```

The `lsmap` command

• Used to list mappings between virtual adapters and physical resources.

List all (virtual) disks attached to the vhost0 adapter
```
lsmap -vadapter vhost0
```
List only the virtual target devices attached to the vhost0 adapter
```
lsmap -vadapter vhost0 -field vtd
```
This line can be used as a list in a for loop
```
lsmap -vadapter vhost0 -field vtd -fmt :|sed -e "s/:/ /g"
```
List all shared ethernet adapters on the system
```
lsmap -all -net -field sea
```
List all (virtual) disks and their backing devices
```
lsmap -all -type disk -field vtd backing
```
List all SEAs and their backing devices
```
lsmap -all -net -field sea backing
```

The `mkvdev` command

• Used to create a mapping between a virtual adapter and a physical resource. The result of this command will be a "virtual device".

Create a SEA that links physical ent0 to virtual ent1
```
mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 1
```
Create a disk mapping from hdisk7 to vhost2 and call it wd_c1_hd7
```
mkvdev -vdev hdisk7 -vadapter vhost2 -dev wd_c1_hd7
```
Remove a virtual target device (disk mapping) named vtscsi0
```
rmvdev -vtd vtscsi0
```

SEA Setup

• The command used to set up a SEA (Shared Ethernet Adapter) is `mkvdev`.

• IP addresses cannot be configured on either the virtual or the physical adapter used in the mkvdev command. IP addresses are configured either on the SEA itself or another physical or virtual adapter that is not part of a SEA "bridge".

• Best practices suggest that IP addresses for the VIOS should not be created on the SEA but should be put on *another* virtual adapter in the VIOS attached to the same VLAN. This makes the IP configuration independent of any changes to the SEA. (This is not the case when using SEA failover).

• The virtual device used in the SEA configuration should have "Access External Networks" (AKA: "Trunk adapter") checked in its configuration (in the profile on the HMC). This is the *only* interface on the VLAN that should have this checked.

Create a SEA "bridge" between the physical ent0 and the virtual ent1
```
mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 1
```
• Explanation of the parameters:
- `-sea ent0` – This is the physical interface
- `-vadapter ent1` – This is the virtual interface
- `-default ent1` – Default virtual interface to send untagged packets
- `-defaultid 1` – This is the PVID for the SEA interface

• The PVID for the SEA is relevant when the physical adapter is connected to a VLAN configured switch and the virtual adapter is configured for VLAN (802.3Q) operation. All traffic passed through the SEA should be untagged in a non-VLAN configuration.

• This example assumes that separate (physical and virtual) adapters are used for each network. (VLAN configurations are not covered in this QuickSheet).

SEA failover

• Unlike a regular SEA adapter, a SEA failover configuration has a few settings that are different from stated best practices.

• A SEA failover configuration is a situation when IP addresses *should* be configured on the SEA adapter.

• A control channel must be configured between the two VIOS using two virtual ethernet adapters that use that VLAN strictly for this purpose. The local virtual adapter created for this purpose should be specified in the `ctl_chan` attribute in each of the SEA setups.

• Both virtual adapters (on the VLAN with clients) should be configured to "Access External network", but one should have a higher priority (lower number) for the "Trunk priority" option. A SEA failover configuration is the only time that you should have two virtual adapters on the same VLAN that are configured in this manner.

• The following command needs to be run on each of the VIOS to create a simple SEA failover. (It is assumed that interfaces match on each VIOS.)
```
mkvdev -sea ent0 -vadapter ent1 -default ent1 \
       -defaultid 1 -attr ha_mode=auto \
       ctl_chan=ent3 netaddr=10.143.180.1
```
• Explanation of the parameters:
- `-sea ent0` – This is the physical interface
- `-vadapter ent1` – This is the virtual interface
- `-default ent1` – Default virtual interface to send untagged packets
- `-defaultid 1` – This is the PVID for the SEA interface
- `-attr ha_mode=auto` – Turn on auto failover mode
- `(-attr) ctl_chan=ent3` – Define the control channel interface
- `(-attr) netaddr=10.143.180.1` – Address to ping for connect test

• `auto` is the default `ha_mode`, `standby` forces a failover situation

Change the device to standby mode (and back) to force failover `chdev`
```
-dev ent4 -attr ha_mode=standby
chdev -dev ent4 -attr ha_mode=auto
```
See what the priority is on the trunk adapter
```
netstat -cdlistats | grep "Priority"
```

Virtual Disk Setup

• Disks are presented to VIOC by creating a mapping between a physical disk or storage pool volume and the vhost adapter that is associated with the VIOC.

• Best practices configuration suggests that the connecting VIOS vhost adapter and the VIOC vscsi adapter should use the same slot number. This makes the typically complex array of virtual SCSI connections in the system much easier to comprehend.

• The `mkvdev` command is used to create a mapping between a physical disk and the vhost adapter.

Tasks

Mirror the rootvg in VIOS to hdisk1
```
extendvg rootvg hdisk1
mirrorios hdisk1
```
(The VIOS will reboot when finished)

Disk

Determine if SCSI reserve is enabled for hdisk4
```
lsdev -dev hdisk4 -attr reserve_policy
```
Turn off SCSI reserve for hdisk4
```
chdev -dev hdisk4 -attr reserve_policy=no_reserve
```
Re-enable SCSI reserve for hdisk4
```
chdev -dev hdisk4 -attr reserve_policy=single_path
```
Enable extended disk statistics (in VIOS 1.3)
```
chdev -dev sys0 -attr iostat=true
```
List the parent device of hdisk0
```
lsdev -dev hdisk0 -parent
```
List all the child devices of (DS4000 array) dar0
```
lsdev -dev dar0 -child
```
List the reserve policy for all disks on a DS4000 array
```
for D in `lsdev -dev dar0 -child -field name|grep -v name`
do
lsdev -dev $D -attr reserve_policy
done
```

Storage Pools

• The default storage pool is `rootvg`. If storage pools are used in a configuration then the default storage pool should be changed to something other than `rootvg`.

List the default storage pool
```
lssp -default
```
List all storage pools
```
lssp
```
Create a storage pool called client_boot on hdisk22
```
mksp client_boot hdisk22
```
Make the client_boot storage pool the default storage pool
```
chsp -default client_boot
```
Add hdisk23 to the client_boot storage pool
```
chsp -add -sp client_boot hdisk23
```
List all the physical disks in the client_boot storage pool
```
lssp -detail -sp client_boot
```
List all the backing devices (LVs) in the default storage pool
```
lssp -bd
```
Create a client disk on adapter vhost1 from client_boot storage pool
```
mkbdsp -sp client_boot 20G -bd lv_c1_boot -vadapter vhost1
```
• The `mkbdsp` command does *not* allow you to specify a descriptive name for the virtual device it creates.

Remove the mapping for the device just created, but save the backing device
```
rmbdsp -vtd vtscsi0 -savebd
```
Assign the lv_c1_boot backing device to another vhost adapter
```
mkbdsp -bd lv_c1_boot -vadapter vhost2
```
Completely remove the virtual target device ld_c1_boot
```
rmbdsp -vtd ld_c1_boot
```
Remove last disk from the sp to delete the sp
```
chsp -rm -sp client_boot hdisk22
```

Performance Monitoring

Retrieve statistics for ent0
```
entstat -all ent0
```
Reset the statistics for ent0
```
entstat -reset ent0
```
View disk statistics
```
viostat
```
• The `topas` command is availible in VIOS.

Backup

Create a mksysb file of the system on a NFS mount
```
backupios -file /mnt/vios.mksysb -mksysb
```
Create a backup of all structures of (online) VGs and/or storage pools `savevgstruct vdiskvg` (Data will be saved to `/home/ios/vgbackups`)

List all (known) backups made with `savevgstruct`
```
restorevgstruct -ls
```
Backup the system (mksysb) to a NFS mounted filesystem
```
backupios -file /mnt
```

VIOS Security

List all open ports on the firewall configuration
```
viosecure -firewall view
```
To view the current security level settings
```
viosecure -view -nonint
```
Change system security settings to default
```
viosecure -level default
```
To enable basic firewall settings
```
viosecure -firewall on
```
List all failed logins on the system
```
lsfailedlogin
```

About this QuickSheet

Created by: William Favorite (wfavorite@tablespace.net)
Updates at: http://www.tablespace.net
Disclaimer: This document is a guide and it includes no express warranties to the suitability, relevance, or compatibility of its contents with any specific system. Research any and all commands that you inflict upon your command line.
Distribution: The PDF version is free to redistribute as long as credit to the author and tablespace.net is retained in the printed and viewable versions. LaTeXsource not distributed at this time.