Chapter 22
-
Virtual Partitions (vPars)

INDEX

This chapter provides an overview of the Virtual Partitions (vPars) product.

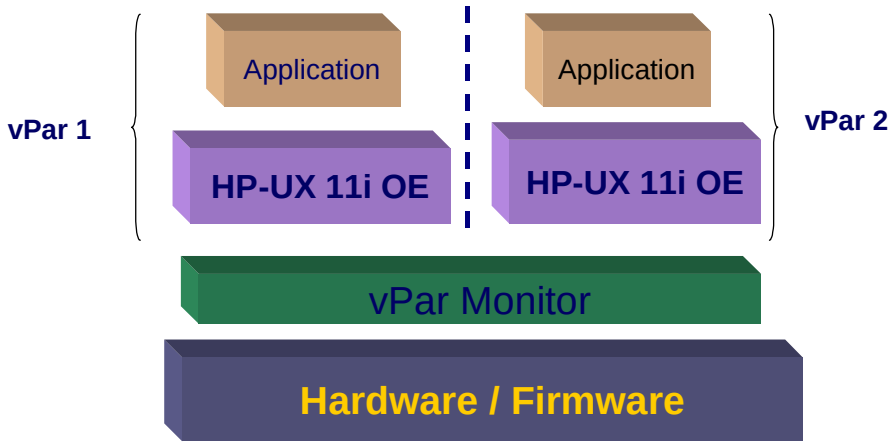
Virtual Partitions (vPars) is not included with the HP-UX Operating System. With this product it is possible to run multiple instances of HP-UX on one Node. The detailed product information can be found at http://docs.hp.com/:

- HP-UX Virtual Partitions Ordering and Configuration Guide
- Installing and Managing HP-UX Virtual Partitions
- Read Before Installing HP-UX Virtual Partitions

Introduction

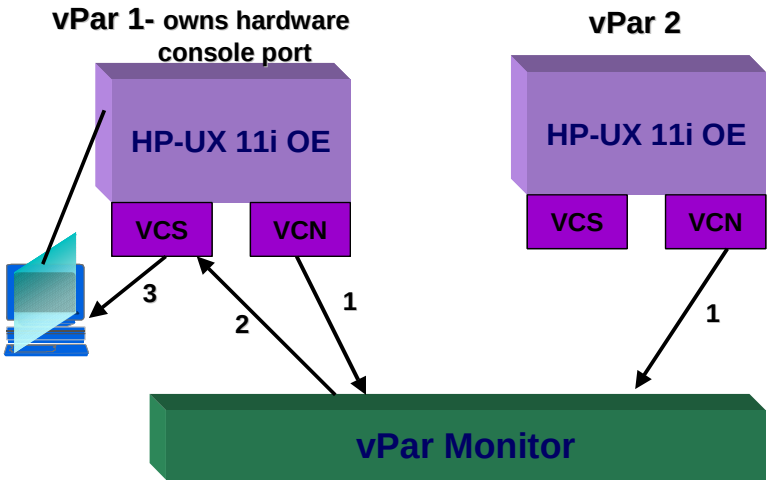
Virtual Partitions (vPars) allows you to run multiple instances of HP-UX on the same node. The instances of HP-UX are fully isolated from each other. Each Partition is assigned its own subset of hardware. Each vPar can have a different patch level to the operating System.

The Virtual Partition needs a Monitor called `vpmon`. This is loaded before starting a partition running HP-UX. The Monitor is loaded on Boot Control Handler (BCH). This Monitor enables the system to run vPars. If the Monitor is not started we can boot the original kernel as well. But remenber we only can run one instance of HP-UX without the vPars Monitor. The following picture shows the Monitor:



The Monitor is called `vpmon` and is stored in `/stand/vpmon` on each partition. The commands for the vPars Monitor are shown in section Monitor Commands or via http://docs.hp.com. To run a vPar the Monitor needs a database where all partition information is stored. The vPars Database is called `/stand/vpdb` and resides on each Virtual Partition. To create or modify a Virtual Partition see section vPars commands or online manual at http://docs.hp.com.

Virtual Console



Each virtual partition has its own virtual console. You can use **Ctrl-A** to cycle between all live vPars.

1. Each vPar has its console I/O sent to their **vcn** (Virtual CoNsole) driver. The **vcn** driver then sends the console I/O to the Monitor.
2. From the vPars Monitor, the console I/O is sent to the **vcs** (virtual console slave) driver of the partition that owns the hardware console port.
3. Finally, the **vcs** driver sends the console I/O to the physical hardware console. It is this **vcs** driver that manages the console I/O to the actual hardware console port.

Daemons

The daemon **`vpard`** is started with the script **`/sbin/init.d/vpard`**. It will synchronize the master database and the local disk database for all **"up"** state vPars, at a default rate of every 5 seconds. The **`vpard`** daemon also helps to manage communications for the **virtual console**.

The daemon **`vphbd`** is started with the script **`/sbin/init.d/vparhb`**. It provides a heartbeat status, written to the local disk. The default for **"sleeping"** between heartbeats is 360 seconds (6 minutes). If 10 heartbeats are missed, the vPar is considered **"hung"**. This state is maintained within the master database.

Once the Monitor is booted, the database selected (**`/stand/vpdb`** is default, or, the database specified when booting the Monitor) is copied into memory and becomes the master copy. The master copy is used to compare and synchronize the other **run state** vPars. If the database does not exist on the vPar as it boots, it will be created. The daemon **`vpard`** communicates with the Monitor to ensure the local databases are properly synchronized. Updates occur every 5 seconds by default.

Supported Systems and Requirements for vPars

Virtual Partition comes in two Versions:

- VPARBASE product
- T1335AC product

The VPARBASE product is free of charge and can be downloaded from http://software.hp.com.
The Product T133AC must be ordered and will be delivered on an **extra** CD. This product is not bundled on the Application (DART) CD-ROM.

Limitation to vparbase product:

The VPARBASE product , provides the capability for creating 2 virtual partitions, of which one virtual partition has only 1 CPU. Migration of CPUs between the vPars is not possible.

Supported Systems and required Firmware vPars Version A.02.02

Supported systems:

- rp5470 (formerly L3000)
- rp7400 (formerly N4000)
- SuperDome
- rp8400 (Keystone)
- rp7410 (Matterhorn)
- L1000 and L2000 are not supported with vPars

Firmware Requirements:

- rp5470 (L3000), rp7400 (N4000) need PDC reviosion 41.02 or higher
- rp7410, rp8400 minimal Firmware Version needs 4.0
- SuperDome needs PDC release 35.3 (June 2002)

Requirements for vPars

Product and Patches:
To use vPars on a node you need to install VPARBASE or T133AC.
The product includes all patches that are neccessary to enable vPars. There are newer patches that are called “vpars”. These patches can also be applied when the product is already installed.

With the actual Version of vPars A.02.02 we support also ICOD on vPars Systems. We need to install ICOD Version 5.00 to work with vPars.
For details regarding iCOD refer to the iCOD Chapter.

The A5838A combo LAN/SCSI card is now supported as a boot device in a vPar.
The RAID 4Si card (A5856A) is unsupported with vPars.

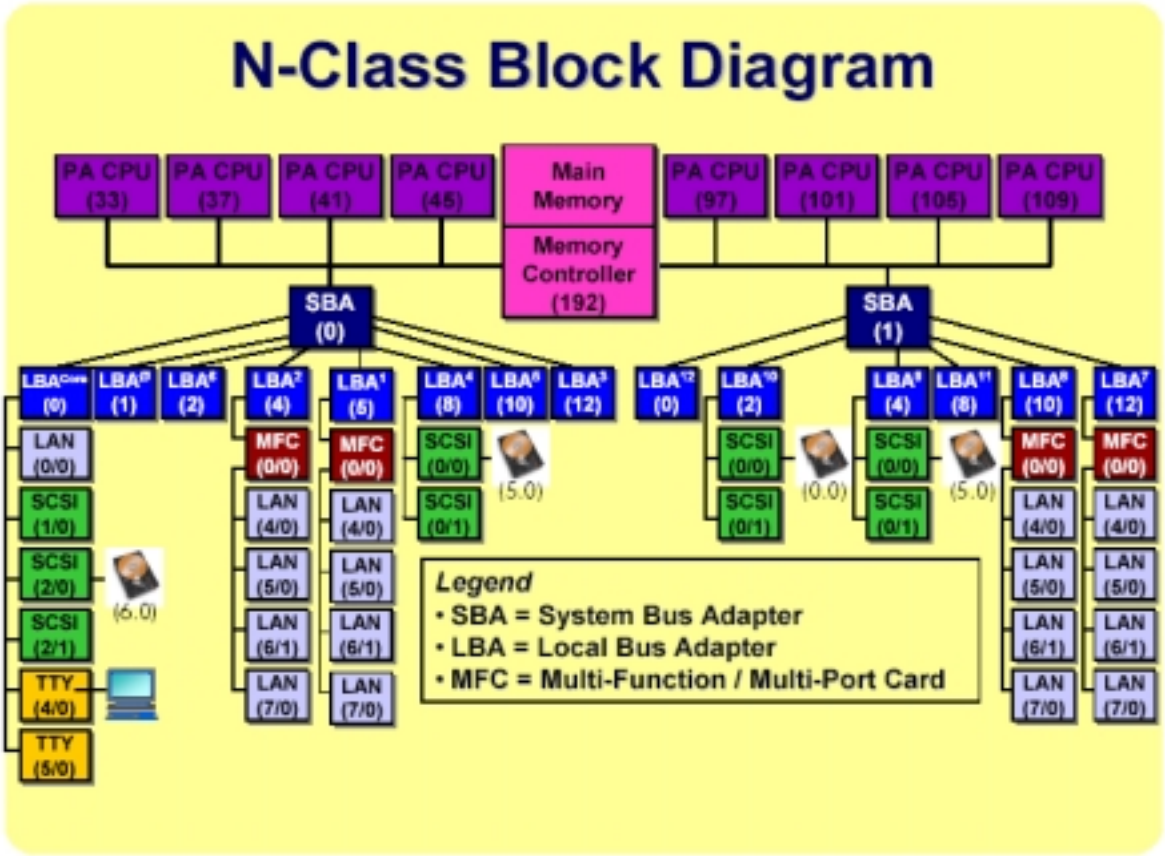
-
- Minimum Requirements for each vPar

- one processor
- enough **physical memory** to run HP-UX 11i and applications
  Minimum requirement is 256MB for L3000 and 512MB on N4000
  Recommended 1GB of memory per available per each installed CPU
- one unique **LAN card** (on a PCI bus/Local Bus Adapter (LBA) that is uniquely owned by that vPar).
- one unique **boot device** (connecting through SCSI or Fiber Channel card, on a PCI bus/Local Bus Adapter that is uniquely owned by that vPar)

Planning a Virtual Partition (an example)

This section gives an overview how to break down a system into partitions. The N-Class block diagramm helps to identify the usage of bus adapters to create a partition plan from the ioscan output.

N-Class Block Diagramm



ioscan

```
H/W Path       Class                     Description
=================================================
```

```
               root
0              ioa                     System Bus Adapter (803)
0/0                 ba            Local PCI Bus Adapter (782)
0/0/0/0                    lan         HP PCI 10/100Base-TX Core
0/0/1/0                    ext_bus     SCSI C895 Fast Wide LVD
0/0/1/0.7                    target
0/0/1/0.7.0                     ctl    Initiator
0/0/2/0                    ext_bus     SCSI C875 Ultra Wide Single-Ended
0/0/2/0.6                    target
0/0/2/0.6.0                disk   SEAGATE ST39102LC
0/0/2/0.7                    target
0/0/2/0.7.0                     ctl    Initiator
0/0/2/1                    ext_bus     SCSI C875 Ultra Wide Single-Ended
0/0/2/1.7                    target
0/0/2/1.7.0                     ctl    Initiator
0/0/4/0                    tty         PCI Serial (103c1048)
0/0/5/0                    tty         PCI Serial (103c1048)
0/1                 ba                 Local PCI Bus Adapter (782)
0/2                 ba                 Local PCI Bus Adapter (782)
0/4                 ba            Local PCI Bus Adapter (782)
0/4/0/0                    ba          PCItoPCI Bridge
0/4/0/0/4/0                      lan   HP A5506A PCI 10/100Base-TX 4 Port
0/4/0/0/5/0                      lan   HP A5506A PCI 10/100Base-TX 4 Port
0/4/0/0/6/0                      lan   HP A5506A PCI 10/100Base-TX 4 Port
0/4/0/0/7/0                      lan   HP A5506A PCI 10/100Base-TX 4 Port
0/5                 ba            Local PCI Bus Adapter (782)
0/5/0/0                    ba          PCItoPCI Bridge
0/5/0/0/4/0                      lan   HP A5506A PCI 10/100Base-TX 4 Port
0/5/0/0/5/0                      lan   HP A5506A PCI 10/100Base-TX 4 Port
0/5/0/0/6/0                      lan   HP A5506A PCI 10/100Base-TX 4 Port
0/5/0/0/7/0                      lan   HP A5506A PCI 10/100Base-TX 4 Port
0/8               ba                 Local PCI Bus Adapter (782)
0/8/0/0                     ext_bus     SCSI C875 Fast Wide Differential
0/8/0/0.5                      target
0/8/0/0.5.0                  disk   SEAGATE ST39175LC
0/8/0/0.7                      target
0/8/0/0.7.0                       ctl   Initiator
0/8/0/1                      ext_bus    SCSI C875 Fast Wide Differential
0/8/0/1.7                      target
0/8/0/1.7.0                       ctl   Initiator
0/10              ba                    Local PCI Bus Adapter (782)
0/12              ba                    Local PCI Bus Adapter (782)
1              ioa                      System Bus Adapter (803)
1/0               ba                    Local PCI Bus Adapter (782)
1/2               ba                    Local PCI Bus Adapter (782)
1/2/0/0                      ext_bus    SCSI C875 Fast Wide Differential
1/2/0/0.0                      target
1/2/0/0.0.0                       disk  SEAGATE ST39102LC
1/2/0/0.7                      target
1/2/0/0.7.0                       ctl   Initiator
1/2/0/1                      ext_bus    SCSI C875 Fast Wide Differential
1/2/0/1.7                      target
1/2/0/1.7.0                       ctl   Initiator
1/4               ba                 Local PCI Bus Adapter (782)
1/4/0/0                      ext_bus    SCSI C875 Fast Wide Differential
1/4/0/0.5                      target
1/4/0/0.5.0                  disk   SEAGATE ST39175LC
1/4/0/0.7                      target
1/4/0/0.7.0                       ctl   Initiator
1/4/0/1                      ext_bus    SCSI C875 Fast Wide Differential
1/4/0/1.7                      target
1/4/0/1.7.0                       ctl   Initiator
1/8               ba                    Local PCI Bus Adapter (782)
1/10              ba                 Local PCI Bus Adapter (782)
1/10/0/0                ba                PCItoPCI Bridge
1/10/0/0/4/0                     lan  HP A5506A PCI 10/100Base-TX 4 Port
1/10/0/0/5/0                     lan  HP A5506A PCI 10/100Base-TX 4 Port
1/10/0/0/6/0                     lan  HP A5506A PCI 10/100Base-TX 4 Port
```

```
1/10/0/0/7/0                    lan  HP A5506A PCI 10/100Base-TX 4 Port
1/12              ba            Local PCI Bus Adapter (782)
1/12/0/0                ba      PCItoPCI Bridge
1/12/0/0/4/0                    lan  HP A5506A PCI 10/100Base-TX 4 Port
1/12/0/0/5/0                    lan  HP A5506A PCI 10/100Base-TX 4 Port
1/12/0/0/6/0                    lan  HP A5506A PCI 10/100Base-TX 4 Port
1/12/0/0/7/0                    lan  HP A5506A PCI 10/100Base-TX 4 Port
32            pbc               Bus Converter
33            processor         Processor
36            pbc               Bus Converter
37            processor         Processor
40            pbc               Bus Converter
41            processor         Processor
44            pbc               Bus Converter
45            processor         Processor
96            pbc Bus           Converter
97            processor         Processor
100           pbc               Bus Converter
101           processor         Processor
104           pbc               Bus Converter
105           processor         Processor
108           pbc               Bus Converter
109           processor         Processor
192           memory            Memory
```

Refering to the full ioscan we create three vPars:

- Resources assigned to **vpar1** is marked in blue.
- Resources assigned to **vpar2** is marked in green.
- Resources assigned to **vpar3** is marked in orange.

Partition information

The partition plan shows the following:

Name of vpar	vpar1	vpar2	vpar3
Processors	2	2	2
Memory	1 GB	4 GB	2 GB
I/O Path (LBS's)	0/0, 0/4	0/8, 1/1	0/5, 1/4
Boot Path	0/0/2/0.6.0	0/8/0/0.5.0	1/4/0/0.5.0

Console:
The hardware console port is at 0/0/4/0 which uses LBA at 0/0. The LBA is owned by vpar1 so when we create the virtual partitions, vpar1 will be created first.

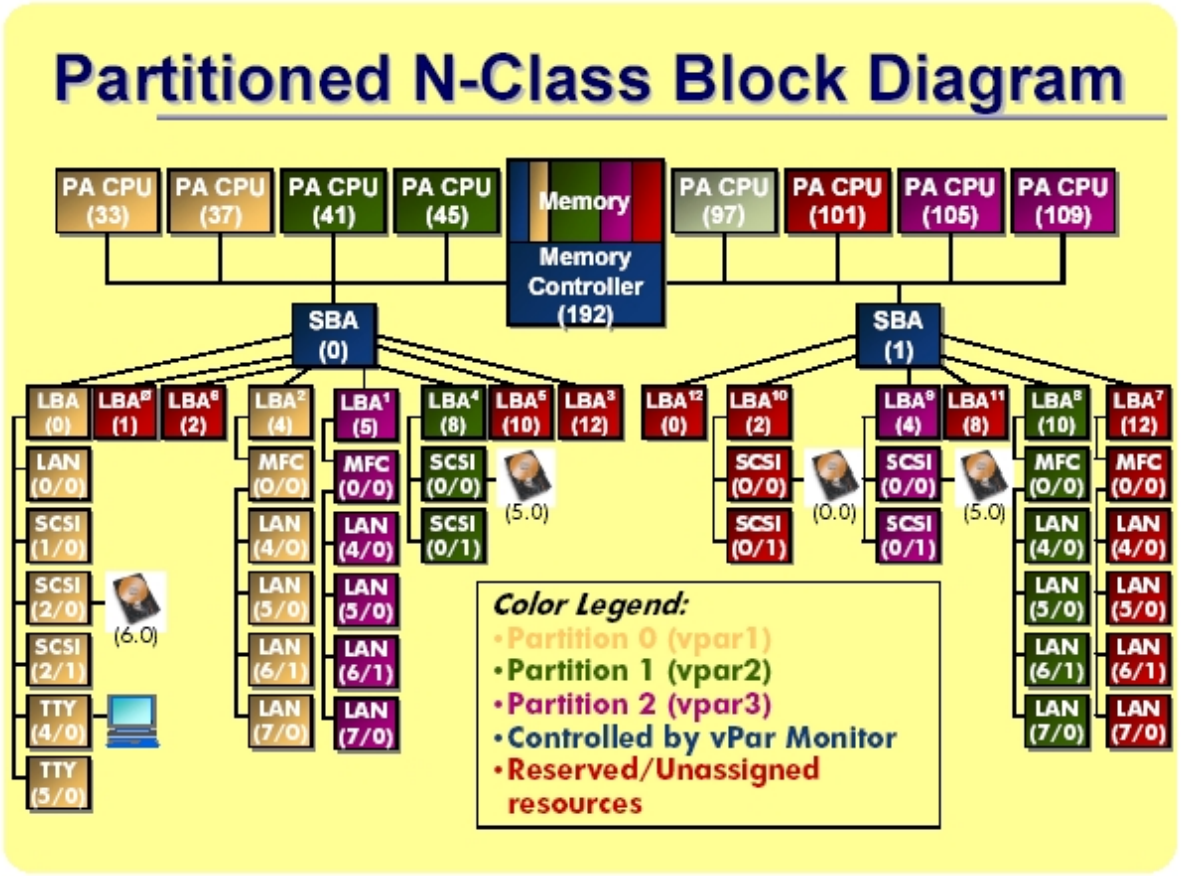
To create these Partitions we need `vparcreate(1m)`.

```
# vparcreate -p vpar1 -a cpu::2
                      -a mem::1024
                      -a io:0/0 -a io:0/4 -a io:0/0/2/0.6.0:boot
                      -B auto

# vparcreate -p vpar2 -a cpu::2
                      -a mem::4096
                      -a io:0/8 -a io:1/1 -a io:0/8/0/0.5.0:boot
                      -B auto
```

```
# vparcreate -p vpar3 -a cpu::2
                      -a mem::2048
                      -a io:0/5 -a io:1/4 -a io:1/4/0/0.5.0:boot
                      -B auto
```

Now we see the three partitions in our block diagram:



Installing vPars

The vPars product needs to be installed on a system which is already running HP-UX 11.11.

Installation options:
- you can install each vPar separately from CD **while the vPars Monitor is down**, or
- you can use Ignite-UX “golden image” (created using make_net_recovery) to clone a vPar from another vPar or install from Ignite-UX depots.

Ignite-UX is the preferred way, especially if you want to be able to perform task (installing and configuring applications,…) on the other running vPars while the new vPars is installing.

Installation from CD:

You´ll find the following packages:

- vPars bundle (T1335AC)
- Partition Manager (B6826AA)
- ICOD 5.0 (B9073AA)
- VPars Manager (VPARMGR)
- Online Diagnostics (OnlineDiags)

Installation from Ignite-UX:

Assumptions

- A stable HP-UX system on which Ignite-UX B.3.7.X may be (or has been) installed. This Ignite-UX server could be an external server (preferred) or one the vPars. We´ll see in detail later.
- The Ignite-UX server and the target system are on the same subnet, or there is a Gateway Helper configured to allow the target system to boot from the Ignite-UX server.
- The user has access to the Ignite-UX, HP-UX, and vPars product software, including any necessary patches.

At this point we have two choices:

Create depots

Step 1.
Create HP-UX 11i OE depots from appropriate media. In this example we're using a CDROM device (c0t3d0) that is assumed to contain the HP-UX 11i media: (/var/opt/ignite/depots/Rel_B.11.11/core)

```
# make_depots -r B.11.11 -s /dev/dsk/c0t3d0
```

Add the vPars application software to the depot. Here's an example using a tar-format depot file: (/var/opt/ignite/depots/Rel_B.11.11/apps)

```
# make_depots -r B.11.11 -s
/tmp/B6826AA,r=B.11.11.01.05,a=HPUX_B.11.11_32_64,v=HP.tar
# make_depots -r B.11.11 -s
/tmp/T1335AC,r=A.02.01.00,a=HPUX_B.11.11_64,v=HP.tar
# make_depots -r B.11.11 -s
/tmp/VPARMGR,r=B.11.11.01.01,a=HPUX_B.11.11_32_64,v=HP.tar
```

Create an Ignite-UX config file for the above depots (11i OEs and vPars):

```
# make_config -r B.11.11
```

The above command creates the "**/var/opt/ignite/data/Rel_B.11.11/apps_cfg** " and "**/var/opt/ignite/data/Rel_B.11.11/core_cfg**" configuration files.
Manage the Ignite-UX index file for applications:

```
# manage_index -a -f /var/opt/ignite/data/Rel_B.11.11/apps_cfg
```

Step 2.
Start the Ignite-UX server manager:

```
# /opt/ignite/bin/ignite
```

Step 3.
Once the server is running, complete the Ignite-UX server configuration.
Select: Options → Server Configuration …

- Select the Server Options tab.
- Default Configuration: HP-UX B.11.11 Default
- Clients Timeouts: 40
- Run client installation UI on: server
- Select: Options → Server Configuration … → Server Options
  - Verify that only these options are set:
  - Confirm new clients -
  - Show the welcome screen for the install server
- Ensure the option "**Halt the client after installation**" is not selected.

Note that the Ignite-UX server configuration created by the above procedure includes, BUT DOES NOT AUTOMATICALLY INSTALL, the required vPars 2.0 software components. If you use Ignite-UX server installed in this way, YOU MUST manually select it for installation during the software selection phase of Ignite-UX client installation process.

Cloning a System Using make_net_recovery

The recovery configurations and archives created by make_net_recovery are stored in a separate directory on the Ignite-UX server for each client. Using the configuration and archive created by make_net_recovery on one system to install a different system involves manually copying some configuration files, and allowing NFS access to the source system's archive.

Step 1.
Use make_net_recovery or ignite-UX to create a system recovery archive of the source system.

Step 2.
Login to the Ignite-UX server.

Step 3.
If the target system to be installed does not currently have a directory in /var/opt/ignite/clients but is up and running, then use the ignite to create that directory using **Actions Add New Client for Recovery**. If the system is not running, you will either need to boot the client from the Ignite-UX server (or from a tape made with make_boot_tape in order

for this directory to be created.

Step 4.
Copy the CINDEX and recovery directory from the source client to the target client directory. If the target client has previously used make_net_recovery then it will already have a INDEX file. If the CINDEX file for the target system exists already, you may want to save a copy, and/or hand edit the file to add the desired entries from the source client. The commands below copy the required files. You may specify src_client and target_client using either the LAN addresses (such as 0x0060B04AAB30), or by using the client's hostname (which is a symlink to the LAN address):

```
# cd /var/opt/ignite/clients/src_client \
  find CINDEX recovery | cpio -pdvma ../target_client
```

Step 5.
Give the *target_client* NFS access to the archive of the source system. To do this, login to the server that holds the archive (normally the Ignite-UX server).
Typically each client has its own directory for storing the archives, and the directory is exported only to the individual client. In this case, you will need to edit the /etc/exports file to allow access to both the source and target clients:

- Enter: vi /etc/exports
- Append :target-client to the end of the source-client's line.
- Enter: exportfs -av

Step 6.
Boot the target-client from the Ignite-UX server (using any method you wish). Then when you install the system, you can select from the recovery configurations of the source system.

Step 7.
Change the system networking parameters for the target system during the installation.

NOTES: This document assumes that you are using the September,2002 release of Ignite-UX, version B.3.7.X.
More information about Ignite-UX can be found in the Ignite-UX Administration Guide (B2355-90750)

Using Virtual Partitions

This chapter describes the usage of Virtual Partition Software and the commands. In this chapter there are also information how to boot the Monitor and the Partition.

Booting the Monitor

The vPars Monitor is a file and called /stand/vpmon. This file will be loaded before the HP-UX Kernel is loaded see also chapter Introduction.

The Monitor will be loaded at the ISL prompt.

```
ISL> hpux /stand/vpmon
ISL> hpux /stand/vpmon vparload -p vpar1  (loads the vPars directly)
```

With no arguments to `/stand/vpmon`, the Monitor will load and go into interactive mode with the following prompt:

```
MON>
```

The following options are available when booting the Monitor:

`-a`	boots all virtual partitions that have the autoboot attribute set. For more information, see `vparmodify(1M)`.
`-D database_filename`	boots the virtual partitions using an alternate partition database file. For more information, see "Using an Alternate Partition Database File" at http://docs.hp.com. The default partition database file is `/stand/vpdb`.

If the Monitor is loaded you can use the Monitor commands from next section. The most useable command is to boot a partition with `vparload -p <partition name>`.

Monitor Commands

The following table gives an overview over the Monitor commands at the `MON>` prompt. The most useable command is `vparload`. The `vparload` command does the same one as `vparboot` command from a running partition.

readdb	reads an alternative database.

vparload	boots or loads a vPar from the Monitor. Examples:
	`MON> vparload -all`
	`MON> vparload -auto`
	`MON> vparload -p vpar1`
	`MON> vparload -p vpar1 –o "-is" –b /stand/vmunix.prev`
	This command performs the same function as the `vparboot` command from the HP-UX shell prompt.

bootpath	Shows the bootpath from which `vpmon` was loaded
reboot	reboots the Monitor and the whole server like CTRL-B, RS !!!
cat	can be used for listing a file, similar to the **cat(1)** command within HP-UX. The default directory is **"/stand**".
cbuf	Display consolen buffer
help or **?**	displays the list of Monitor commands.
lifls	lists the contents of the **LIF** (Logical Interchange Format) volume.
getauto	returns the contents of the **AUTO** file within the **LIF** volume.
log	displays the contents of the Monitor log.
ls	will list the files in a directory. The default directory is **"/stand**".
scan	displays the hardware found by the Monitor.
toddriftreset	will reset the **Time Of Day** (TOD) real-time clock drift.
vparinfo	displays information about a vPar, or, about unassigned hardware.

readdb

```
MON> readdb <filename>
```

Reads an alternate partition database filename for partition configuration information filename must be an absolute path and reside on a HFS file system.

Example:
If you have a backup copy of the partition database in the file `/stand/vpdb.backup`, you can read the database configuration information using:

```
MON> readdb /stand/vpdb.backup
```

vparload

```
MON> vparload -all
MON> vparload -auto
MON> vparload -p partition_name
             [-b kernelpath] [-o boot_options] [-B hardware_path]
```

Boots the virtual partition partition_name; this command is similar to the vPars Unix shell command vparboot.

- `-all` boots all virtual partitions, regardless of the autoboot attribute. For more information on the autoboot attribute, see the vparcreate (1M) or vparmodify (1M) manpages.
- `-auto` boots all virtual partitions that have their autoboot attribute flag set to AUTO.
- `-b` kernelpath boots the virtual partition using the kernel kernelpath, such as /stand/vmunix.prev, instead of the default kernel /stand/vmunix.
- `-o` boot_options boots the virtual partition using the options boot_options, such as -is for single-user mode or -lm for LVM maintenance mode.
- `-B` hardware_path boots the virtual partition using the disk device at the hardware_path

Examples:
To boot the partition vpar2 into single-user mode:

```
MON> vparload -p vpar2 -o "-is"
```

To boot the partition vpar2 using the kernel /stand/vmunix.other:

```
MON> vparload -p vpar2 -b /stand/vmunix.prev
```

To boot the partition vpar2 using the disk device at 0/8/0/0.2.0:

```
MON> vparload -p vpar2 -B 0/8/0/0.2.0
```

NOTE:
The option `-b kernelpath` allows you to change the target kernel for only the next boot of `partition_name`.If you wish to make a permanent change to the partition database, use the `vparmodify` command. For example, to change the partition database information so that vpar2 always boots using `/stand/vmunix.other`:

```
# vparmodify -p vpar2 -b /stand/vmunix.other
```

See the `vparmodify(1M)` manpage for more information on modifying the partition database.

bootpath

Displays the device from which the vPars Monitor (`/stand/vpmon`) was booted.

Example:
```
MON>bootpath
disk(0/0/2/0.6.0)
```

Reboot

Reboots the entire hard partition. Other hard partitions are not affected.

cat

```
cat <filename> [openonly]
```

Displays the contents of filename. When openonly is specified, this command only prints "open succeeded" if the Monitor was able to open the filename. This command is similar to the Unix cat command.filename must be a text file on an HFS file system. `/stand` is the default directory.

Example:
To display the file `/stand/notes.txt`
```
MON> cat notes.txt
10/13/2001: built new kernel today. if problems arise, revert to saved
kernel vmunix.original
```

cbuf

```
cbuf partition_name
```

Displays the contents of the console buffer of `partition_name`

help

`help` or `?` lists all Monitor commands.

lifls

Lists the files in the LIF area

getauto

Displays the contents of the AUTO file in the LIF area

Example:
```
MON> getauto
hpux /stand/vpmon
```

log

Displays the contents, including warning and error messages, of the Monitor log. The Monitor log holds up to 16KB of information in a circular log buffer. The information is displayed in

chronological order.

ls

`ls [-alniFH ][directory]`
Lists the contents of directory. This command is similar to the Unix ls command. Directory must be on a HFS file system. `/stand` is the default directory. The `ls` command-line options are the same as the Unix shell ls options. For detailed explanations, see the `ls(1M)` manpage. In brief:

Example to view the listing of files in vpar2's `/stand` directory:

```
MON> ls /stand
lost+found          ioconfig     bootconf           system
system.d            vmunix       dlkm.vmunix.prev   build
kernrel             rootconf     vpdb               vpmon.dmp
vmunix.backup       system.prev  vmunix.prev        dlkm
vpdb.backup         vpmon
```

scan

Lists all hardware discovered by the Monitor and indicates which virtual partition owns each device.

toddriftreset

Resets the drifts of the real-time clock. Use this command if you reset the real-time clock of the hard partition at the BCH prompt.

vparinfo

`vparinfo [partition_name]`

This command is for HP internal use only.

- `-a` all entries
- `-l` long listing
- `-n` numerical UIDs and GIDs
- `-i` inode
- `-F` appends a character after the entry, depending on the file type, such as a / (slash) for a directory when no partition_name is given, vparinfo displays all unassigned resources and the names of all existing virtual partitions; when partition_name is given, vparinfo displays the resources assigned to `partition_name`.

vPars Commands

vparcreate	Create a new virtual partition
vparboot	Boot a virtual partition
vparmodify	Modify the attributes of a virtual partition
vpardump	Manages Monitor dump files
vparextract	Extracts memory images from a running virtual partition system
vparreloc	Relocates the load address of a vmunix file, determines if a vmunix file is relocatable and promotes the scope of symbols in a relocatable vmunix file
vparremove	Remove a virtual partition
vparreset	Reset a virtual partition
vparstatus	Display information about one or more vPars
vparutil	Gets and sets SCSI parameters for SCSI controllers from a virtual partition

vparcreate

vparcreate is used to create a new virtual partition. Let's have a look at a short example to see how to use vparcreate.

- To create a virtual partition named vpar1 with the following resources:
- Three total CPUs (two bound CPUs at hardware paths 41 and 45 and one unbound CPU) with a maximum of four (bound plus unbound) CPUs
- 1280 MB of memory
- all hardware where the path begins with `0/8` or `1/10`
- a boot disk at `0/8/0/0.5.0`

use the corresponding `vparcreate` command line options:

ressource or attribute	vparcreate option
virtual partition name is vpar1	`-p vpar1`
three total CPUs	`-a cpu::3`
of which two are bound CPUs and a maximum of four CPUs	`-a cpu:::2:4`
at hardware paths 41 and 45	`-a cpu:41 -a cpu:45`
1280 MB of memory	`-a mem::1280`
all hardware where the path begins with 0/8	`-a io:0/8`
all hardware where the path begins with 1/10	`-a io:1/10`
hardware at 0/8/0/0.5.0 as the boot disk	`-a io:0/8/0/0.5.0:boot`

The resulting `vparcreate` command line is:

```
vptest# vparcreate -p vpar1
                   -a cpu::3 -a cpu:::2:4 -a cpu:41 -a cpu:45
                   –a mem::1280
```

```
–a io:0/8 -a io:1/10 -a io:0/8/0/0.5.0:boot
```

vparboot

The vparboot command is used to boot a second vPar from a running vPar. The vparboot command has several options as the vparload command at the Monitor prompt.

Examples:

Boot vpar2 partition from running vpar1 partition:

```
vpar1# vparboot -p vpar2
```

Boot vpar2 partition to single user mode from running vpar1 partition:

```
vpar1# vparboot -p vpar2 –o “-is”
```

vparmodify

With vparmodify we can change the resources and attributes of an existing vPar. So the options are similar to the vparcreate command. The different here is to use the option `-d` to delete a ressource or attribute. Another option is `-m` to modify the attributes or ressources in a vPar.

Examples:

Modify the primary boot path to vpar1:

```
vptest# vparmodify -p vpar1 -m io:0/8/0/0.5.0:BOOT
```

Set the alternate boot path:

```
vptest# vparmodify -p vpar1 -a io:0/8/0/0.2.0:ALTBOOT
```

Modifying a vPars can be done by deleting and then adding a resource. Otherwise we can use vparmodify with `-m` option to modify a resource. On the next table we see the different options to vparmodify for adding,removing or deleting resources. See also the man page `vparresources(5)` on a running vPars system.

Task	Syntax	allowed with vpars running

`-a (add)`	`cpu:path`
	`cpu::num`
	`cpu:::[min][:[max]]`
	`(vparcreate only)`
	`io:path[:attr1[,attr2]]`
	`mem::size`
	`mem:::base:range`
		No
		Yes
		N/A
		
		No
		No
		No

`-m (modify)`	`cpu::num`
	`cpu:::[min][:[max]]`
	`io:path[:attr1[,attr2]]`
	`mem::size`
		Yes
		No
		No
		No

`-d delete`	`cpu:path`
	`cpu::num`
	`io:path[:attr1[,attr2]]`
	`mem::size`
	`mem:::base:range`
		No
		Yes
		No
		No
		No

vpardump

The vpardump command is used to create and analyze a virtual partition Monitor dump file. `vparmon` is the image of the virtual partition Monitor and dumpfile is the crash dump file corresponding to that image. `vpardump` is normally run from the `vparinit` rc script to analyze a crash dump during HP-UX boot. If no options are specified, `vpardump` checks if dumpfile has been cleared. If not, it analyzes dumpfile and prints information about the crash to stdout. Then vpardump marks the dumpfile as clear. If dumpfile is not specified, the string `.dmp` is appended to `vparmon` as in `/stand/vparmon.dmp`. If `vparmon` is not specified, it defaults to `/stand/vpmon`.

vparextract

Extract memory images from a running virtual partition system.

vparreloc

Relocate the load address of a vmunix file, determine if a vmunix file is relocatable, or promote the scope of symbols in a relocatable vmunix file

vparremove

The vparremove command deletes a virtual partition previously created using the vparcreate command. All resources associated with the virtual partition are made available for allocation to other partitions.

Example:

```
vptest# vparremove -p vpar1
Remove virtual partition vpar1? [n]: y
```

vparreset

The vparreset command simulates, at the virtual partition level, the RS and TOC operations at a Control-B prompt on the system console. This can be done from another running vPar on the same system. The other running partitions are not affected.

Example:
To TOC the vPar vpar1

```
vptest# vparreset -p vpar1 -t
```

To shutdown vpar1

```
vptest# vparreset -p vpar1 -h
```

vparstatus

Display information about one or more virtual partitions on a system. If Monitor is not loaded the vparstatus shows this with an message. The output shows also the state of a vPar.

Examples:
Running vparstatus on system where vpar is configured, but not running at this time.

```
# vparstatus
vparstatus: Warning: Virtual partition Monitor not running, Requested
```

```
resources shown.
[Virtual Partition]
Boot
Virtual Partition Name State Attributes Kernel Path Opts
====================== ===== ========== ========================
mars                    N/A   Dyn,Auto /stand/vmunix
jupiter                 N/A   Dyn,Auto /stand/vmunix

[Virtual Partition Resource Summary]
CPU Num Memory (MB)
CPU Bound/ IO # Ranges/

Virtual Partition Name Min/Max Unbound devs Total MB Total MB
============================== ================ ==== ====================
mars                   2/4       1      2     256
jupiter                1/4 1     0      3     256
```

`vparstatus` shows the following stated of a running vPar

State	Description
`load`	A vPar is loading/booting
`boot`	A vPar is loaded into memory
`up`	A vPar is up and running.
`shut`	A vPar is shutting down.
`down`	A vPar is down.
`crash`	A vPar is crashing (panic - HPMC, TOC)
`hung`	A vPar is not responding.

The complete status and configuration can be listed by `vparstatus` too.

```
# vparstatus -v -p vpar1

[Virtual Partition Details]
Name:         vpar1
State:        N/A
Attributes:   Dynamic,Autoboot
Kernel Path:  /stand/vmunix
Boot Opts:

[CPU Details]
Min/Max:  1/3
Bound by User [Path]:
Bound by Monitor [Path]:  <no path>
Unbound [Path]:

[IO Details]
   2.0.0
   2.0.1
   2.0.1.0.0.1.0  BOOT
   2.0.4

[Memory Details]
Specified [Base  /Range]:
          (bytes) (MB)
Total Memory (MB):  1024
```

vparutil

The vparutil command gets and sets SCSI parameters for SCSI controllers within a running virtual partition. When setting the SCSI parameters using the -s option, the virtual partition which owns the specified SCSI controller must be in the down state. If not, the command will fail with the appropriate error code.

Ignite-UX Network Recovery

For information on Ignite-UX refer to the Ignite-UX chapter.

Making an Archive of a Virtual Partition

`make_tape_recovery` is not supported for vPars hard partition. You need to use `make_net_recovery`.
`make_net_recovery` works the same for making archives of both non-vPars and vPars hard partitions.

Recovering a Virtual Partition

Ignite-UX modifies the LIF area during a recovery, yet vPars does not use the LIF area for booting a virtual partition; therefore, on a vPars hard partition you need to perform some additional steps.

NOTE: Prior to performing recovery, read the section Updating the Ignite-UX Server.

To recover a virtual partition, perform the following from a running virtual partition. (In these examples, the partition `vpar1` is running and the partition `vpar2` is the partition being recovered.)

1) Record whether the `autoboot` attribute is set. You may need to set it back to this state in Step 7.

```
vpar1# vparstatus -p vpar2
[Virtual Partition]
Virtual Partition Name     State Attributes
========================== ===== ==========
vpar2                       Down Dyn,Auto
```

2) Set the autoboot attribute to `manual` for the target virtual partition (the virtual partition you like to recover):

```
vpar1# vparmodify -p vpar2 -B manual
```

3) Set the kernel path of the target partition to use the boot kernel `/stand/WINSTALL`:

```
vpar1# vparmodify -p vpar2 -b /stand/WINSTALL
```

Ignite-UX modifies the LIF area to boot the WINSTALL kernel as part of its recovery process. However, because vPars uses the vPars database instead of the LIF area to boot a virtual partition, this change needs to be done to the vPars database.

4) Set the `TERM` environment variable to `hpterm`. For `POSIX` shell, the command is

```
vpar1# export TERM=hpterm
```

5) Boot the target partition and point the boot kernel to use your Ignite-UX server (assume the Ignite server's IP is 15.xx.yy.zz):

```
vpar1# vparboot -p vpar2 -I 15.xx.yy.zz,/opt/ignite/boot/WINSTALL
```

6) Run the Ignite-UX recovery as you would on a non-vPars hard partition, entering the data (boot disk and LAN) of the target partition.

7) After the target partition has been recovered, if the autoboot attribute was set to AUTO in Step 1, reset the autoboot attribute of the target partition back to AUTO:

```
vpar1# vparmodify -p vpar2 -B auto
```

8) Reset the kernel path of the target partition back to the normal kernel:

```
vpar1# vparmodify -p vpar2 -b /stand/vmunix
```

This changes the vPars database back to its original contents. (See Step 3 above for the manual change that was done).

Recovering all the Virtual Partitions of a Hard Partition

To recover all the virtual partitions within a hard partition, first recover the virtual partition whose boot disk is the disk set as the primary path within system-wide stable storage. Once the virtual partition is recovered, recover the other virtual partitions one by one. (There is no way to recover all partitions simultaneously.):

1) From the BCH prompt, boot the hard partition using the Ignite-UX server (assume the Ignite server's IP is 15.xx.yy.zz):

```
BCH> bo lan.15.xx.yy.zz install
interact with IPL? N
```

2) From the Ignite-UX window, select "Install HP-UX".

3) Enter the network data using the data for the virtual partition that owns the boot disk that is set as the primary path within system-wide stable storage.

4) Select Recovery Archive Configuration -> Go

5) After this virtual partition is recovered, recover the remaining partitions using the instructions in Recovering a Virtual Partition.

Updating the Ignite-UX Server

CAUTION: If you are using Ignite-UX versions B.3.4.XX (September 2001), B.3.5.XX (December 2001), or B.3.6.XX (March 2002), in addition to adding the vPars bundles to your Ignite server, you need to replace the existing file `/opt/ignite/boot/WINSTALL` with a vPars-compatible `WINSTALL` file using the script named `WINSTALL_script`. `WINSTALL` is a bootable kernel used by Ignite-UX during the installation and recovery of a virtual partition. `WINSTALL_script` copies a vPars-compatible `WINSTALL` to the correct location on the Ignite-UX server.

Determining the Version of Ignite-UX

To determine which version of Ignite-UX you are running, execute the command:

```
# swlist -l fileset -a revision Ignite-UX.FILE-SRV-11-11
```

For example, if your swlist output shows:

```
# Initializing...
# Contacting target "vpar1"...
#
# Target: vpar1:/
#
Ignite-UX.FILE-SRV-11-11 B.3.5.890
```

then your Ignite-UX version is B.3.5.89.

Installing a vPars-compatible WINSTALL file using WINSTALL_script

To replace the `WINSTALL` file on your Ignite-UX server, perform the following from your Ignite-UX server:

1) Obtain the vPars `WINSTALL` and `WINSTALL_script` files:

   From the vPars CD:
   - If you have purchased the vPars CD, the files are located in the vPars`WINSTALL` directory off the root directory of CD.

   From the vPars website:
   Go to http://www.software.hp.com
   Follow the instructions on the web page for obtaining the files `WINSTALL` and `WINSTALL_script`.

2) Run the script `WINSTALL_script` to copy the `WINSTALL` file to the correct location on your Ignite-UX server.

   NOTE: The `WINSTALL_script` saves a copy of the original `WINSTALL` file. To restore the original `WINSTALL` file, execute the `WINSTALL_script` again.

What is new in this version?

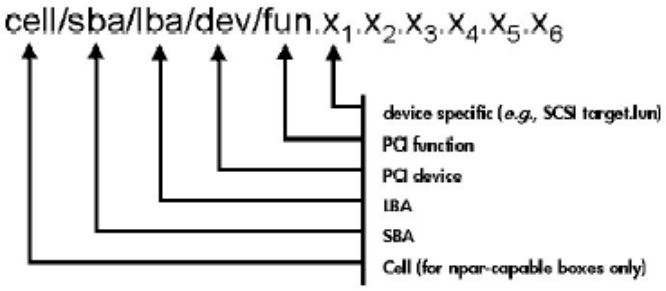
A.02.01

- Superdome support.
- WLM, iCOD support
- Graphical User Interface (GUI), vPars Manager (vparmgr).

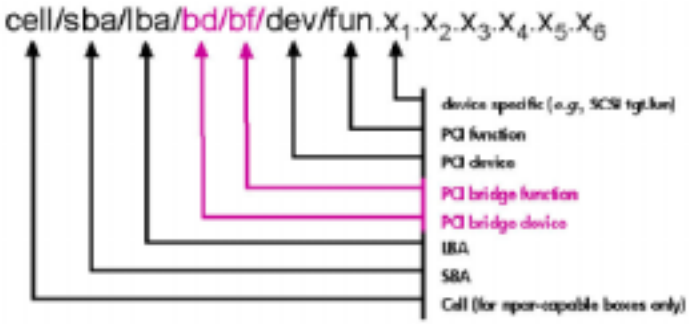
A.02.02

- New I/O hardware: Superdome IO expander, A5838A SCSI/LAN Combo card (previously not supported for boot/dump, requires new hardware path format)
- Hardware path specification format change.
  Prior to A.02.02, vPars allowed hardware path notations for boot devices to be incompletely specified in the database. There it was possible to add a boot device without the complete HW-Path from ioscan. For example, if ioscan shows as HW-Path 0/0/1/1.0.0. Then we can add this device to vPars and the Monitor expand it by

default to full device Path 0/0/1/1.0.0.0.0.0.0 if it is needed.



With A.02.02 we support the SCSI/LAN Combo card as boot device. This card has a built in PCI-Bridge below the lba. There this card has a PCI Brigde and a PCI function here which is followed by the PCI device and function. The full path length for a device is now longer than it was in A.02.01 and the databbase was modified here.



The notation to add HW-Pathes to an vPar can be done with the output from ioscan or the full IO-Path with only dots in it. VPars accept both commands and handle it different. The Database has no delimiter and cannot check what card is present here. The Monitor padded normally after the first dot six zero's with a dot to the Path. Vpars handled the slashes and dots as the same and there the vparstatus only prints dots as delimiter for the Hardware Path. Now the HW-Path can reached thirteen elements. Because we can use slashes and dot's in the HW-Path the Monitor has the following rule to handle this:

- A path will all slashes is not padded
- A path with all dots is not padded
- A hardware path that contains a dot after one or more slashes may be padded by vPars. In this case, the element after the last slash is assumed to be the PCI function, so the path will be padded with zeroes out to six elements after the first dot.

With this information it is important to specify the HW Path for a combo card or any device in A.02.02 to fully ioscan output or with correct information with all dots.

Unsupported IO hardware:

- PCI Ultra 160 SCSI car (A6828A & A6829A) configured in **multi-initiator mode**.
- A5856A RAID 4si controller (it can cause HPMC when its owning vPar is reset).
- Booting a vPar from DVD, tape, or over the network is not supported.

Upgrading to vPars A.02.02

- Follow these steps: (the same for upgrading to A.02.01)
- Boot standalone – no vPars
- Install A.02.02 products (updates Monitor on boot disk, effectively updates vPars product on the vPars that owns standalone boot disk)
- Reboot with new Monitor, boot all partitions.
- Install products on all other vPars.

Patches for vPars

The Product is bundled with a lot of Patches. All these Patches are enablements for vPars Product or needed that vPars can work. These Patches will be installed if the product is selected.
There are also serveral Patches where the name vpars is in the online description. Those patches can be installed, but didn't patch the product itself. There is only one patch for vParsA.02.02 at this time:

PHKL_28764 - vPar (A.02.02) monitor cumulative patch

This patch fixed some Problems described in the patch description. When this patch will be installed the whole nPar needs to be rebooted, because we need to load a new vPars Monitor (vpmon). The Patch must be installed on all vPars in the same nPar.
To look at the crash from a vpmon use the command:

Crash dump handling on vPars systems

If a vPar crashes and writes a dump, the crash handlig is the same as on an nPar. There we write a dump normally to `/var/adm/crash` directory. This dump can be analysed using the same tools as on nPar systems. The only difference is here when the Monitor (vpmon)

crashes.
If the Monitor crashed the whole nPar (all vPars on it) go down. The Monitor writes a dump to the file `/stand/vpmon.dmp`. This file is also present on a running vPars. The file reserved the space to save the crashdump for the Monitor.
to analyze the Monitor crash we can use the command vpardump(1m).

```
vpardump [-f |-i] [-a|-c|-q|-F] [-v] [vparmonitor [dumpimage]]
# vpardump -a /stand/vpmon /stand/vpmon.dmp
```

The output can redirected to a file and send to local Response Center to analyze it.

Interaction with other products

iCOD:

- iCOD 5.0 integrated with vPars A.02.01 & A.02.02
- iCOD 5.0 is supported with vPars but PPU IS NOT SUPPORTED.
- Uses vparmodify to activate/deactivate CPUs

WLM:

- WLM 2.0 supports Vpars
- DON´T USE WLM 2.0 and iCOD simultaneously

Glance and MeasureWare:

- Use versions C.03.50.00 or higher
- For correct reporting of interrupt-enable CPUs install kernel patch PHKL_22987 (or its successor)

MC/Service Guard:

- Supported with vPars, see Service Guard docs for more info

Additional Information

HP-UX Virtual Partitions Ordering and Configuration Guide
http://docs.hp.com/hpux/onlinedocs/1705/vPars_configguidelines_012203.pdf

Installing and Managing HP-UX Virtual Partitions
http://docs.hp.com/hpux/pdf/T1335-90018.pdf

Read Before Installing HP-UX Virtual Partitions
http://docs.hp.com/hpux/pdf/T1335-90008.pdf

Product Support Plan
http://wwwpsp.atl.hp.com/lmx_mount/supplan/psp/12/psp12064.htm (HP Internal)

vPars Training Web Page
http://cso.fc.hp.com/ssil/uxsk/hpux/products/vPars (HP Internal)

vPars technical Web
http://escatc.cup.hp.com/VirtualPartitionsHBold.htm (HP Internal)

HP Partitioning Continuum - Collective
http://escatc.cup.hp.com/vPar_files/continuum/partitioning_collective.htm (HP Internal)