

How I created few vPars in a legacy Superdome SD32

Author : Siddhartha Sankar Sinha

Website : www.sinhass.com

Rev : 1.0

Date : 12/01/2011

Created by Siddhartha Sankar Sinha

Hardware : Legacy Superdome SD32

OS : HPUX11i-VSE-OE,B.11.31.1109

Packages : T1335DC, T2782AC, VseAssist, PHKL_41922 (All Inbuilt with VSE OE)

As I and using VSE OE everything I need is part of the OS. You need to install those

Packages and may be few extra packages to build vPars. Check HP Documents.

Created by Siddhartha Sankar Sinha

Here is my *parstatus* output I will work both from console and from a ssh session.

```

root@R01010101:~# parstatus
Note: No action specified. Default behavior is display all.
[Complex]
Complex Name :
Complex Capacity
  Compute Cabinet (8 cell capable) : 1
Active MP Location : cabinet 0
Model : 8000/800/SI32000
Original Serial Number : 80004010101
Current Product Order Number : A8201A
Original Manufacturer : HP
Complex Profile Revision : 1.0
The total number of partitions present : 1

[Cabinet]
          Cabinet I/O      Bulk Power  Backplane
          Blowers Fans      Supplies   Power Boards
          OK/     OK/     OK/        OK/
Cab      Failed/ Failed/ Failed/    Failed/
Num Cabinet Type N Status N Status N Status N Status      MP
-----
0 8 cell slot 4/0/N+ 5/0/NA 5/0/N+ 3/0/N+      Active

Notes: N+ = There are one or more spare items (fans/power supplies).
       N  = The number of items meets but does not exceed the need.
       N- = There are insufficient items to meet the need.
       ?  = The adequacy of the cooling system/power supplies is unknown.
       HD = Housekeeping only; The power is in a standby state.
       NA = Not Applicable.

[Cell]
Hardware Location  Actual Usage  CPU OK/ Deconf/ Max  Memory (GB) OK/ Deconf  Connected To  Core Cell Capable  Use On Next Boot  Par Num
-----
cab0,cell0 Active Core 4/0/4 18.0/0.0 cab0,bay1,chassis3 yes yes 0
cab0,cell1 Active Base 4/0/4 12.0/0.0 cab0,bay0,chassis3 yes yes 0
cab0,cell2 Active Base 4/0/4 8.0/0.0 - no yes 0
cab0,cell3 Active Base 4/0/4 12.0/0.0 - no yes 0
cab0,cell4 Active Base 4/0/4 12.0/0.0 cab0,bay0,chassis1 yes yes 0
cab0,cell5 Active Base 4/0/4 12.0/0.0 - no yes 0
cab0,cell6 Active Base 4/0/4 12.0/0.0 cab0,bay1,chassis1 yes yes 0
cab0,cell7 Active Base 4/0/4 12.0/0.0 - no yes 0

Notes: * = Cell has no interleaved memory.

[Chassis]
Hardware Location  Usage  Core ID  Connected To  Par Num
-----
cab0,bay0,chassis0 Absent - - -
cab0,bay0,chassis1 Active yes cab0,cell4 0
cab0,bay0,chassis2 Absent - - -
cab0,bay0,chassis3 Active yes cab0,cell1 0
cab0,bay1,chassis0 Absent - - -
cab0,bay1,chassis1 Active yes cab0,cell6 0
cab0,bay1,chassis2 Absent - - -
cab0,bay1,chassis3 Active yes cab0,cell0 0

[Partition]
Par Num Status # of Cells # of Chassis I/O Core cell Partition Name (first 30 chars)
-----
0 Active 8 4 cab0,cell0 Partition 0

```

We have Total 32 Processor, 96GB Memory, 4 Core Capable Cell and 4 Non Core

Cell. Now I will make a list of all the I/Os I will use for my test. List of all CPUs available in this Server

```
root@sd32a0-~/root>ioscan -fnkCprocessor
Class      I  H/W Path  Driver      S/W State  H/W Type  Description
=====
processor  0  0/10      processor   CLAIMED    PROCESSOR  Processor
processor  1  0/11      processor   CLAIMED    PROCESSOR  Processor
processor  2  0/12      processor   CLAIMED    PROCESSOR  Processor
processor  3  0/13      processor   CLAIMED    PROCESSOR  Processor
processor  4  1/10      processor   CLAIMED    PROCESSOR  Processor
processor  5  1/11      processor   CLAIMED    PROCESSOR  Processor
processor  6  1/12      processor   CLAIMED    PROCESSOR  Processor
processor  7  1/13      processor   CLAIMED    PROCESSOR  Processor
processor  8  2/10      processor   CLAIMED    PROCESSOR  Processor
processor  9  2/11      processor   CLAIMED    PROCESSOR  Processor
processor 10  2/12      processor   CLAIMED    PROCESSOR  Processor
processor 11  2/13      processor   CLAIMED    PROCESSOR  Processor
processor 12  3/10      processor   CLAIMED    PROCESSOR  Processor
processor 13  3/11      processor   CLAIMED    PROCESSOR  Processor
processor 14  3/12      processor   CLAIMED    PROCESSOR  Processor
processor 15  3/13      processor   CLAIMED    PROCESSOR  Processor
processor 16  4/10      processor   CLAIMED    PROCESSOR  Processor
processor 17  4/11      processor   CLAIMED    PROCESSOR  Processor
processor 18  4/12      processor   CLAIMED    PROCESSOR  Processor
processor 19  4/13      processor   CLAIMED    PROCESSOR  Processor
processor 20  5/10      processor   CLAIMED    PROCESSOR  Processor
processor 21  5/11      processor   CLAIMED    PROCESSOR  Processor
processor 22  5/12      processor   CLAIMED    PROCESSOR  Processor
processor 23  5/13      processor   CLAIMED    PROCESSOR  Processor
processor 24  6/10      processor   CLAIMED    PROCESSOR  Processor
processor 25  6/11      processor   CLAIMED    PROCESSOR  Processor
processor 26  6/12      processor   CLAIMED    PROCESSOR  Processor
processor 27  6/13      processor   CLAIMED    PROCESSOR  Processor
processor 28  7/10      processor   CLAIMED    PROCESSOR  Processor
processor 29  7/11      processor   CLAIMED    PROCESSOR  Processor
processor 30  7/12      processor   CLAIMED    PROCESSOR  Processor
processor 31  7/13      processor   CLAIMED    PROCESSOR  Processor
```

List of all adapters with Disk Drives. I don't need a DVD Drive as I will install them from Ignite Server.

```
# ioscan -funCdisk|grep -v DVD|awk '{print $3}'|grep -v H \
|sed '/^$/d'|cut -c 1-5|uniq
```

```

root@sd32a0-</root>ioscan -funCdisk|grep -v DVD|awk '{print $3}'|grep -v H \
> |sed '/^$/d'|cut -c 1-5|uniq
0/0/1 ← vPar0 i/o for local disks
1/0/1 ← vPar1 i/o for local disks
4/0/1 ← vPar2 i/o for local disks
4/0/2 ← vPar3 i/o for local disks
6/0/1 ← vPar4 i/o for local disks

```

I am planning only for 2 vPars for now. But I want to save this list in case I need to create more vPars later. The list will save some time.

List of all Network Adapters I am going to use for vPar0 and vPar1. Now in HP-UX 11iv3 *nwmgr* command which will tell me the connection status of all the network cards. So I don't have to run *lanadmin* command on each individual adapter or I don't have to write any shell script to find it. Using *nwmgr* command I found that the following Adapters are actually cabled. In real world all of the Network Interfaces should be cabled. But I can still verify that with this command.

```

root@sd32a0-</root>nwmgr|grep -v DOWN

```

Name/ ClassInstance	Interface State	Station Address	Sub- system	Interface Type	Related Interface
lan0	UP	0x00306EE96E8C	btlan	100Base-TX	
lan6	UP	0x0014C2543978	igelan	1000Base-T	
lan8	UP	0x00306EE95EC9	btlan	100Base-TX	
lan10	UP	0x00110A43D347	igelan	1000Base-T	
lan11	UP	0x00306E4C9ED8	igelan	1000Base-SX	
lan12	UP	0x00306E4C8ED8	igelan	1000Base-SX	
lan13	UP	0x00306EE96E44	btlan	100Base-TX	
lan14	UP	0x00163573C38A	igelan	1000Base-T	
lan21	UP	0x00110A4353F4	igelan	1000Base-T	
lan23	UP	0x00306EF4A336	igelan	1000Base-T	
lan4	UP	0x00306EF4B349	igelan	1000Base-T	

Now list of all the Network Interfaces available in this Server.

```

root@sd32a0-</root>lanscan|grep -v DOWN|awk '{print $1"\t\t"$5}'
Hardware          Net-Interface
Path              NamePPA
0/0/0/1/0        lan0
0/0/4/0/0/6/0   lan1
0/0/4/0/0/6/1   lan2
0/0/6/0/0/6/0   lan3
0/0/12/0/0/6/0  lan5
0/0/14/0/0/6/0  lan6
0/0/14/0/0/6/1  lan7
1/0/0/1/0       lan8
1/0/4/0/0/6/0   lan9
1/0/4/0/0/6/1   lan10
1/0/12/0/0/6/0  lan11
1/0/14/0/0/6/0  lan12
4/0/0/1/0       lan13
4/0/4/0/0/6/0   lan14
4/0/4/0/0/6/1   lan15
4/0/12/0/0/6/0  lan16
4/0/12/0/0/6/1  lan17
4/0/14/0/0/6/0  lan18
4/0/14/0/0/6/1  lan19
6/0/0/1/0       lan20
6/0/4/0/0/6/0   lan21
6/0/4/0/0/6/1   lan22
6/0/9/0/0       lan23
6/0/12/0/0/6/0  lan24
0/0/9/0/0       lan4

```

And now the FC HBAs

```

root@sd32a0-</root>ioscan -funCfc| awk '{print $3}'|grep -v H \
> | sed '/^$/d'|cut -c 1-5|uniq
0/0/4
0/0/6
0/0/8
0/0/1
1/0/2
1/0/4
1/0/8
1/0/1
4/0/4
4/0/1
6/0/2
6/0/3
6/0/4
6/0/8
6/0/1

```

Please be noted that few FC HBA and Network Adapter slot no. will be same as they are Combo Adapters. You select I/O for Network may be you are selecting

the FC HBA too. So plan accordingly. Now we are done with 50% of our planning. We have collected all the information we need. Now we will complete the rest 50% of the planning. Here is the resource list for vPar0 I am going to use. I will use the exactly same network cards configured in nPar0 so that I can avoid double work for vPar0

vPar0 name =vPar0
CPU = 4 Min=2 Max=4, 4 Bound CPU Slot = 0/10, 0/11, 1/10, 2/10
Memory = 8192 MB
Boot Device = 0/0/1/0/0.4.0 same as current nPar boot device
I/O SLOT = 0/0/1 = SCSI (Hard Disks)
I/O SLOT DVD = 0/0/11/0/0.1.0 (It will select 0/0/11)
I/O SLOT LAN = 0/0/0 (CIO lan 100FD), 0/0/4 (2 Port GB, 2Port 2GB FC Combo Adapter) 6/0/9 (1*1GB LAN 1GB FC Combo Adapter)
I/O SLOT FC ADAPTER = Same as last two adapter

vPar1 name =vPar1
CPU = 6 , Min=4 Max=8, 4 Bound CPU Slot = 0/12, 1/11, 2/11, 3/11
Memory = 8192 MB
Boot Device = 1/0/1/0/0.0.0
I/O SLOT = 1/0/1 for Local Hard Disks
I/O SLOT LAN = 0/0/14 Dual Lan & Dual FC Combo Adapter
I/O SLOT FC ADAPTER = Same as Network Adapter

Note: for vPars it is not necessary to boot from Core IO LAN card.

Now I have one nPar already loaded with HP UX 11iv3 VSE OE and I am going to create the 1st vPar right now with the above list. See the command in the box

```
root@sd32a0-~/root>vparcreate -p vPar0 -a cpu::4 -a cpu:::2:8 \
> -a cpu:0/10 -a cpu:1/10 -a cpu:2/10 -a mem::8192 \
> -a io:0/0/1 -a io:0/0/11 -a io:0/0/0 -a io:0/0/4 -a io:6/0/9 \
> -a io:0/0/1/0/0.4.0:BOOT
vparcreate: Note: Database is created with the following information:
      CLM granularity is 128 MBytes
      ILM granularity is 128 MBytes
All the subsequent virtual partition's CLM or ILM memory specifications must be
an integral multiple of the corresponding granularity.
root@sd32a0-~/root>
```

We will check with *vparstatus*

```

root@sd32a0-~/root>vparstatus
vparstatus: Warning: Not accessing live monitor database, Requested resources sh
own.
[Virtual Partition]

Virtual Partition Name      State Attributes  Kernel Path      Boot
=====
vPar0                      N/A   Dyn,Auto,Nsr  /stand/vmunix

[Virtual Partition Resource Summary]

Virtual Partition Name      CPU      Num   Num   Memory Granularity
Min/Max  CPUs   IO    ILM    CLM
=====
vPar0                      2/ 8    4     6     128     128

                                Memory (MB)
                                ILM          CLM

Virtual Partition Name      # User
Ranges/MB          Total MB    # User
Ranges/MB          Total MB
=====
vPar0                      0/ 0          8192    0/ 0          0
root@sd32a0-~/root>

```

Again `vparstatus -v -p vPar0` to make sure I selected the correct device for boot.

It must be the same Drive as the Current boot device.

(For Itanium you must change boot mode setting to vPar using this command

`vparenv -m vPar`. This is unsupported on HPPA. You may have to run

`vparenv -g ilm:128 -g clm:128` and then Verify them with `vparenv` command.

Output will be similar to this output

```

# vparenv
vparenv: The next boot mode setting is "vPars"..
vparenv: The CLM granule size setting is 128.
vparenv: The ILM granule size setting is 128.
vparenv: Note: Any changes in the above settings will become effective
only after the next system reboot.
vparenv: Note: The maximum possible CLM granules per cell is 256.
vparenv: Note: The maximum possible ILM granules for this system is 1024

```

)

Once everything verified, then I will reboot the server


```
root@sd32a0-~/root>vparstatus -v -p vPar0
vparstatus: Warning: Not accessing live monitor database, Requested resources sh
own.
[Virtual Partition Details]
Name:          vPar0
State:         N/A
Attributes:    Dynamic,Autoboot,Nosearch
Kernel Path:  /stand/vmunix
Boot Opts:

[CPU Details]
Min/Max:      2/8
User assigned [Path]:  0.10
                          1.10
                          2.10

Boot processor [Path]:
Monitor assigned [Path]: <no path>

Non-cell-specific:
  User assigned [Count]:    3
  Monitor assigned [Count]: 1
Cell-specific [Count]:    Cell ID/Count
                          <none>

[IO Details]
  0.0.1
  0.0.11
  0.0.0
  0.0.4
  6.0.9
  0.0.1.0.0.4.0.0.0.0.0 BOOT

[Memory Details]
ILM, user-assigned [Base /Range]:
                          (bytes) (MB)
ILM, monitor-assigned [Base /Range]:
                          (bytes) (MB)
ILM Total (MB):  8192 (Floating 0)

ILM Granularity (MB):  128

CLM, user-assigned [CellID Base /Range]:
                          (bytes) (MB)
CLM, monitor-assigned [CellID Base /Range]:
                          (bytes) (MB)
CLM (CellID MB):

CLM Granularity (MB):  128

[OL* Details]
Sequence ID: N/A
Operation: N/A
Status: N/A

root@sd32a0-~/root>
```

I verified with `lvlnboot -v / ioscan -m dsf` and `ioscan -funCdisk` that it is the correct disk so I will continue to next step. Now please remember few more steps required for Itanium Servers and I will mention them in proper place And time. We will run `vparstatus` and check the vPar is correctly created or not. Now we will make sure `/stand/vpmon` is there (just in case).

```

root@sd32a0-</>ll /stand/vp*
-rw-r--r-- 1 root      root      16064 Dec  1 16:03 /stand/vpdb
-rwxr-xr-x 1 bin       bin      171084 Dec  6 2010 /stand/vpmon
-rw-r--r-- 1 root      root     5888840 Nov 23 12:12 /stand/vpmon.dsp
root@sd32a0-</>

```

Next thing we will not change the boot path to use `/stand/vpmon` instead of `/stand/vmunix` and verify that.

```

root@sd32a0-</stand>mkboot -a "hpux /stand/vpmon -a" /dev/rdisk/disk18
root@sd32a0-</stand>lifcp /dev/rdisk/disk18:AUTO -
hpux /stand/vpmon -a
root@sd32a0-</stand>

```

Now I will reboot nPar0

```
root@sd32a0</>shutdown -r -y now
```

Once the server is booted it booted to vpar0 now not npar0.

```

10 seconds expired.
Proceeding...

Initializing boot Device.

Boot IO Dependent Code (IODC) Revision 4

Boot Path Initialized.

HARD Booted.

ISL Revision A.00.44 Mar 12, 2003

ISL booting hpux /stand/vpmon -a

Boot
: disk(0/0/1/0/0.4.0.0.0.0.0;0)/stand/vpmon
974848 + 201528 + 17604040 start 0x23000
[MON] Booting vPar0...

```

Now the Server is back online and I logged in now. I can see that hostname, Network settings, bootdisk didn't change. But iocan outputs are showing less I/Os

```

root@sd32a0-</root>iocan -fnkCprocessor
Class      I  H/W Path  Driver      S/W State  H/W Type  Description
=====
processor  0  0/10     processor   CLAIMED   PROCESSOR  Processor
processor  1  0/11     processor   CLAIMED   PROCESSOR  Processor
processor  2  1/10     processor   CLAIMED   PROCESSOR  Processor
processor  3  2/10     processor   CLAIMED   PROCESSOR  Processor
root@sd32a0-</root>iocan -funCdisk
Class      I  H/W Path  Driver      S/W State  H/W Type  Description
=====
disk       1  0/0/1/0/0.0  sdisk   CLAIMED   DEVICE    HP 73.4GMAP3735NC
           /dev/dsk/c0t0d0 /dev/rdisk/c0t0d0
disk       2  0/0/1/0/0.2.0 sdisk   CLAIMED   DEVICE    HP 73.4GST373307LC
           /dev/dsk/c0t2d0 /dev/rdisk/c0t2d0
disk       0  0/0/1/0/0.4.0 sdisk   CLAIMED   DEVICE    HP 73.4GST373307LC
           /dev/dsk/c0t4d0 /dev/rdisk/c0t4d0
disk       3  0/0/11/0/0.1.0 sdisk   CLAIMED   DEVICE    HP DVD-ROM 30
5
           /dev/dsk/c5t1d0 /dev/rdisk/c5t1d0
root@sd32a0-</root>iocan -funCfc
Class      I  H/W Path  Driver      S/W State  H/W Type  Description
=====
fc         0  0/0/4/0/0/4/0 fcd    CLAIMED   INTERFACE HP AB465-60001 PCI/PC
I-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 1)
           /dev/fcd0
fc         1  0/0/4/0/0/4/1 fcd    CLAIMED   INTERFACE HP AB465-60001 PCI/PC
I-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 2)
           /dev/fcd1
root@sd32a0-</root>iocan -funClan
Class      I  H/W Path  Driver      S/W State  H/W Type  Description
=====
lan        0  0/0/0/1/0     btlan   CLAIMED   INTERFACE HP PCI 10/100Base-
TX Core
lan        1  0/0/4/0/0/6/0 igelan  CLAIMED   INTERFACE HP AB465-60001 PCI
/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter
lan        2  0/0/4/0/0/6/1 igelan  CLAIMED   INTERFACE HP AB465-60001 PCI
/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter
lan        23 6/0/9/0/0     igelan  CLAIMED   INTERFACE HP A6825-60101 PCI

```

Now our first vPar ready and I am ready to create the second vPar.

```
root@sd32a0-~/root>vparstatus
[Virtual Partition]
```

Virtual Partition Name	State	Attributes	Kernel Path	Boot Opts
vPar0	Up	Dyn,Auto,Nsr	/stand/vmunix	

```
[Virtual Partition Resource Summary]
```

Virtual Partition Name	CPU Min/Max	Num CPUs	Num IO	Memory ILM	Granularity CLM
vPar0	2/ 8	4	6	128	128

Virtual Partition Name	Memory (MB) ILM		Memory (MB) CLM	
	# User Ranges/MB	Total MB	# User Ranges/MB	Total MB
vPar0	0/ 0	8192	0/ 0	0

Creating vPar1 now.

```
root@sd32a0-~/root>vparcreate -p vPar1 -a cpu::6 -a cpu:::4:8 -a cpu:0/12 \
> -a cpu:1/11 -a cpu:2/11 -a cpu:3/11 -a mem::8192 -a io:1/0/1 \
> -a io:0/0/14 -a io:1/0/1/0/0.0.0:BOOT
```

```
root@sd32a0-~/root>vparstatus
[Virtual Partition]
```

Virtual Partition Name	State	Attributes	Kernel Path	Boot Opts
vPar0	Up	Dyn,Auto,Nsr	/stand/vmunix	
vPar1	Down	Dyn,Auto,Nsr	/stand/vmunix	

```
[Virtual Partition Resource Summary]
```

Virtual Partition Name	CPU Min/Max	Num CPUs	Num IO	Memory ILM	Granularity CLM
vPar0	2/ 8	4	6	128	128
vPar1	4/ 8	6	3	128	128

Virtual Partition Name	Memory (MB) ILM		Memory (MB) CLM	
	# User Ranges/MB	Total MB	# User Ranges/MB	Total MB
vPar0	0/ 0	8192	0/ 0	0
vPar1	0/ 0	8192	0/ 0	0

Next step is to load this vPar from Ignite Server. For Itanium Servers Syntax is little different.

```
root@sd32a0-~/root>vparboot -p vPar1 -I 192.168.192.6,\
> /opt/ignite/boot/Rel_B.11.31/WINSTALL
vparboot: Booting vPar1. Please wait...
root@sd32a0-~/root>
```

For Itanium - Syntax is like this

```
# vparboot -p testvpar4 -I -s 192.168.192.6 -c 192.168.192.54 -g 192.168.192.6 -m 255.255.255.0 -b \
/opt/ignite/boot/nbp.efi -o IINSTALL
```

Now I will go to the console of vPar1 by pressing Ctrl+A. vPar1 is booted from The Ignite Server now. And I will start loading the Server. I will skip all the details of the Ignite loading as you are already aware of it.

```
                Welcome to Ignite-UX!

Use the <tab> key to navigate between fields, and the arrow keys
within fields. Use the <return/enter> key to select an item.
Use the <return/enter> or <space-bar> to pop-up a choices list. If the
menus are not clear, select the "Help" item for more information.

Hardware Summary:                System Model: 9000/800/SD32000
+-----+-----+-----+-----+-----+-----+
| Disks: 3 ( 204.6GB) | Floppies: 0 | LAN cards: 2 |
| CD/DVDs: 0 | Tapes: 0 | Memory: 8064Mb |
| Graphics Ports: 0 | IO Buses: 3 | CPUs: 6 |
+-----+-----+-----+-----+-----+-----+
[ Scan Again ]

[ Install HP-UX ]

[ Run an Expert Recovery Shell ]

[ Advanced Options ]

[ Reboot ] [ Help ]
```

It will take 45 minutes to an hour from now.

```

insf: Installing special files for esdisk instance 5 address 64000/0xfa00/0x2
insf: Installing special files for pseudo driver dlpi
insf: Installing special files for pseudo driver keptd
insf: Installing special files for pseudo driver framebuf
insf: Installing special files for pseudo driver sad
    * Constructing the bootconf file.
    * Setting primary boot path to "1/0/1/0/0.0x0.0x0".
Primary boot path set to 1/0/1/0/0.0x0.0x0 (/dev/rdisk/disk5)
    * Starting install of the source (HP-UX Core Software).
    * Running command: "/opt/swm/bin/swm install
-s192.168.192.6:/ignite_depot/depots/Rel_B.11.31/11iv3vse
-xsuspend_after_load=true -xallow_oe_selection=true
-xdefer_kernel_rebuild=true -xautoreboot=true
-f/tmp/ign_configure/software_file -xpatch_save_files=true "

===== 12/02/11 20:08:41 UTC BEGIN Software Install (user=root)
(jobid=000001)

    * Reading Target System's Installed Product Database (IPD)
    * Target IPD: sd32a1
    * Reading Source Depot
    * Source depot: 192.168.192.6:/ignite_depot/depots/Rel_B.11.31/11iv3vse
    * Analyzing Software To Be Installed

```

Now vPar1 OS load completed. I will go back to vPar0 and check the status

```

root@sd32a0-~/root>vparstatus
[Virtual Partition]

Virtual Partition Name      State Attributes      Kernel Path      Boot
=====
vPar0                      Up    Dyn,Auto,Nsr /stand/vmunix
vPar1                      Up    Dyn,Auto,Asr /stand/vmunix
=====

[Virtual Partition Resource Summary]

Virtual Partition Name      CPU      Num      Num      Memory Granularity
Min/Max      CPUs      IO      ILM      CLM
=====
vPar0                      2/ 8      4      6      128      128
vPar1                      4/ 8      6      3      128      128

Memory (MB)

Virtual Partition Name      ILM      CLM
# User      Total MB      # User      Total MB
=====
vPar0                      0/ 0      8192      0/ 0      0
vPar1                      0/ 0      8192      0/ 0      0
root@sd32a0-~/root>

```

Both vPars are up now. I will press CTRL+A now and go to the console of vPar1

I will check few stuffs before concluding.

```
root@sd32a0-</root>
```

```
[vPar1]
```

```
root@sd32a1-</root>ioscan -funCdisk
```

Class	I	H/W Path	Driver	S/W State	H/W Type	Description
disk	2	1/0/1/0/0.0.0	sdisk	CLAIMED	DEVICE	HP 73.4GST373307LC
			/dev/dsk/c0t0d0		/dev/rdisk/c0t0d0	
disk	1	1/0/1/0/0.2.0	sdisk	CLAIMED	DEVICE	COMPAQ BD07285A25
			/dev/dsk/c0t2d0		/dev/rdisk/c0t2d0	
disk	0	1/0/1/0/0.4.0	sdisk	CLAIMED	DEVICE	HP 73.4GST373307LC
			/dev/dsk/c0t4d0		/dev/rdisk/c0t4d0	

```
root@sd32a1-</root>lanscan
```

Hardware Station Path	Address	Crdrd	Hdw In#	State	Net-Interface Name	PPA	NM ID	MAC Type	HP-DLPI Support	DLPI Mjr#
0/0/14/0/0/6/0	0x0014C2543978	0	UP	lan0	snap0	1	ETHER	Yes	119	
0/0/14/0/0/6/1	0x0014C2543979	1	UP	lan1	snap1	2	ETHER	Yes	119	
LinkAgg0	0x000000000000	900	DOWN	lan900	snap900	4	ETHER	Yes	119	
LinkAgg1	0x000000000000	901	DOWN	lan901	snap901	5	ETHER	Yes	119	
LinkAgg2	0x000000000000	902	DOWN	lan902	snap902	6	ETHER	Yes	119	
LinkAgg3	0x000000000000	903	DOWN	lan903	snap903	7	ETHER	Yes	119	
LinkAgg4	0x000000000000	904	DOWN	lan904	snap904	8	ETHER	Yes	119	

```
root@sd32a1-</root>ioscan -funCfc
```

Class	I	H/W Path	Driver	S/W State	H/W Type	Description
fc	0	0/0/14/0/0/4/0	fcd	CLAIMED	INTERFACE	HP AB465-60001 PCI/P
			CI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 1)			
			/dev/fcd0			
fc	1	0/0/14/0/0/4/1	fcd	CLAIMED	INTERFACE	HP AB465-60001 PCI/P
			CI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 2)			
			/dev/fcd1			

```
root@sd32a1-</root>
```

Conclusion : Everything looks OK, now I am completed building 2 vPars. Using same method I can build several vPars depending of Hardware limitation.