Virtual Partitions Hands–on Lab

Bev (Woody) Woods
HP Services Americas WD&OE

Alan Hymes

HP Competitive Sales and Presales

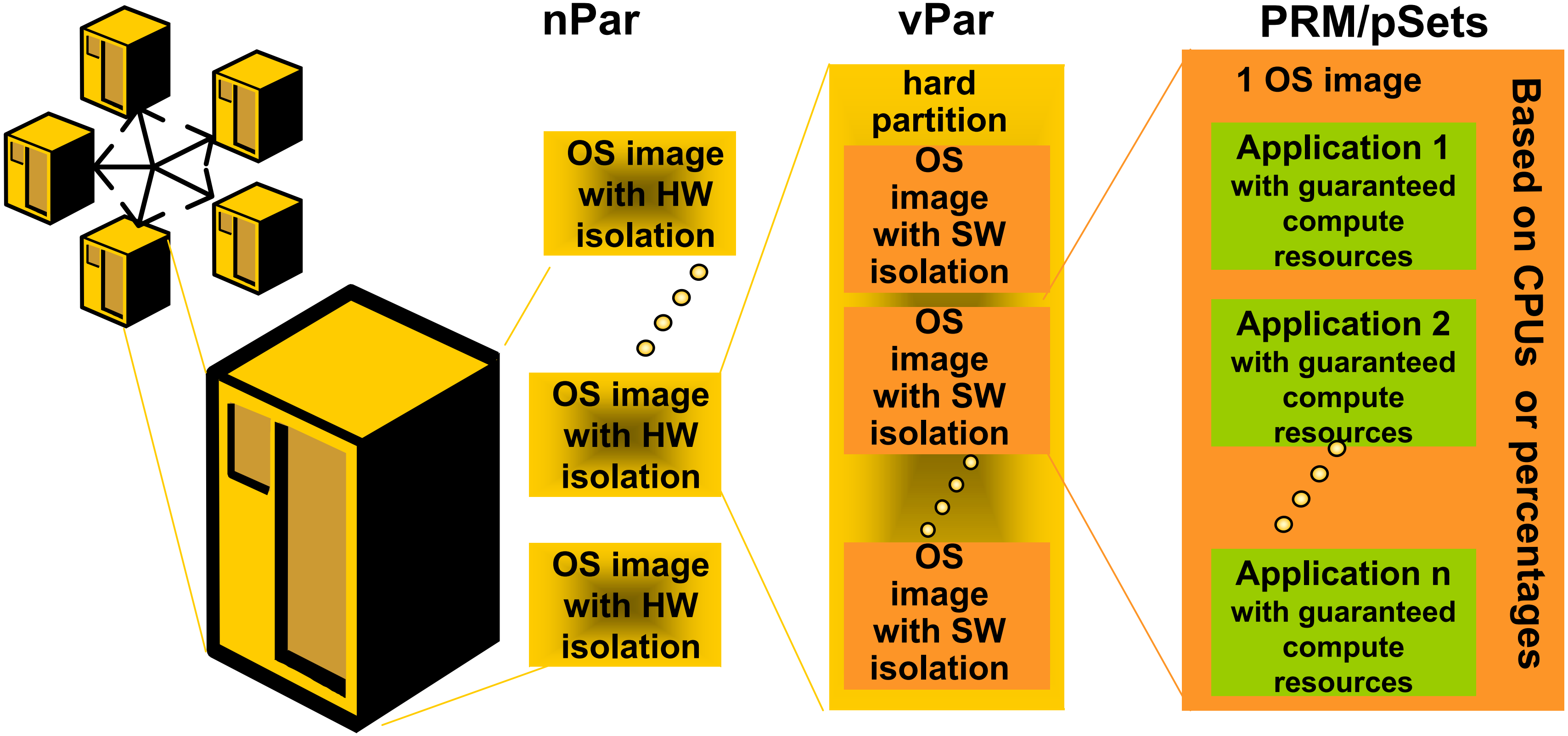
Johnny Earnest

HP Competitive Sales and Presales

HP WORLD 2003
Solutions and Technology Conference & Expo

Partitioning Continuum Overview

HP Partitioning Continuum for HP-UX



Clustered nodes

Hard partitions
nPar

Virtual partitions
vPar

Resources partitions
PRM/pSets

OS image with HW isolation

OS image with HW isolation

OS image with HW isolation

hard partition

OS image with SW isolation

OS image with SW isolation

OS image with SW isolation

1 OS image

Application 1 with guaranteed compute resources

Application 2 with guaranteed compute resources

Application n with guaranteed compute resources

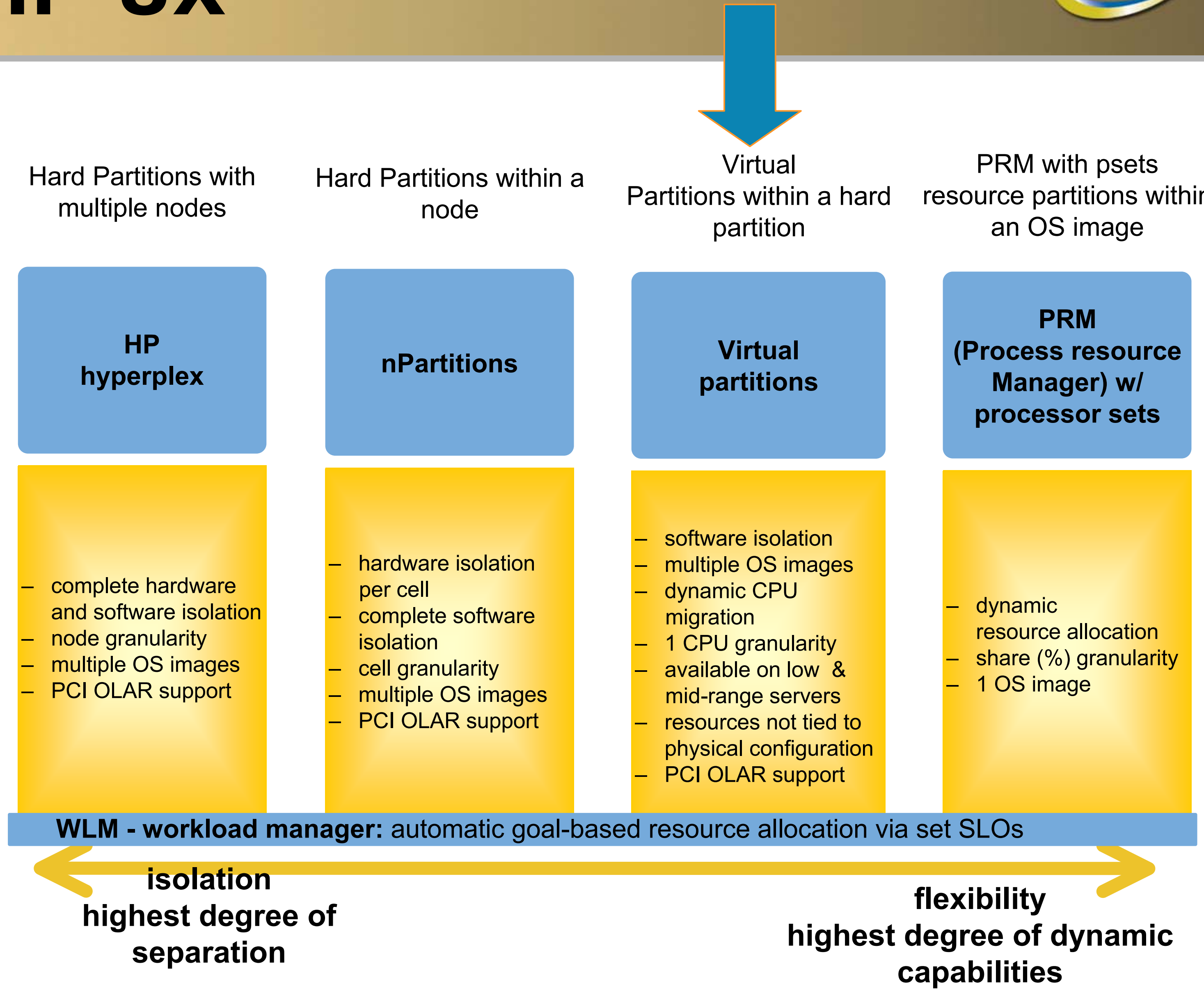
Based on CPUs or percentages

HP-UX Workload Manager

Isolation
Highest degree of separation

Flexibility
Highest degree of dynamic capabilities

HP Partitioning Continuum for HP-UX



Hard Partitions with multiple nodes

HP hyperplex

- complete hardware and software isolation
- node granularity
- multiple OS images
- PCI OLAR support

Hard Partitions within a node

nPartitions

- hardware isolation per cell
- complete software isolation
- cell granularity
- multiple OS images
- PCI OLAR support

Virtual Partitions within a hard partition

Virtual partitions

- software isolation
- multiple OS images
- dynamic CPU migration
- 1 CPU granularity
- available on low & mid-range servers
- resources not tied to physical configuration
- PCI OLAR support

PRM with psets resource partitions within an OS image

PRM (Process resource Manager) w/ processor sets

- dynamic resource allocation
- share (%) granularity
- 1 OS image

WLM - workload manager: automatic goal-based resource allocation via set SLOs

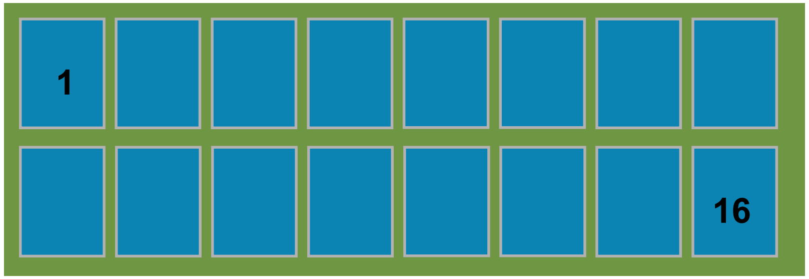
isolation
highest degree of separation

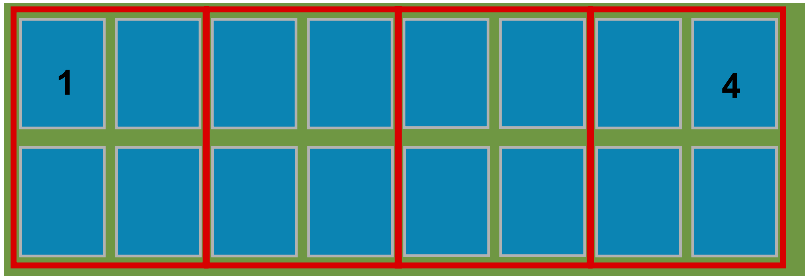
flexibility
highest degree of dynamic capabilities

Hard Partition (nPar) Hardware Overview

nPartitions (hard partitions)

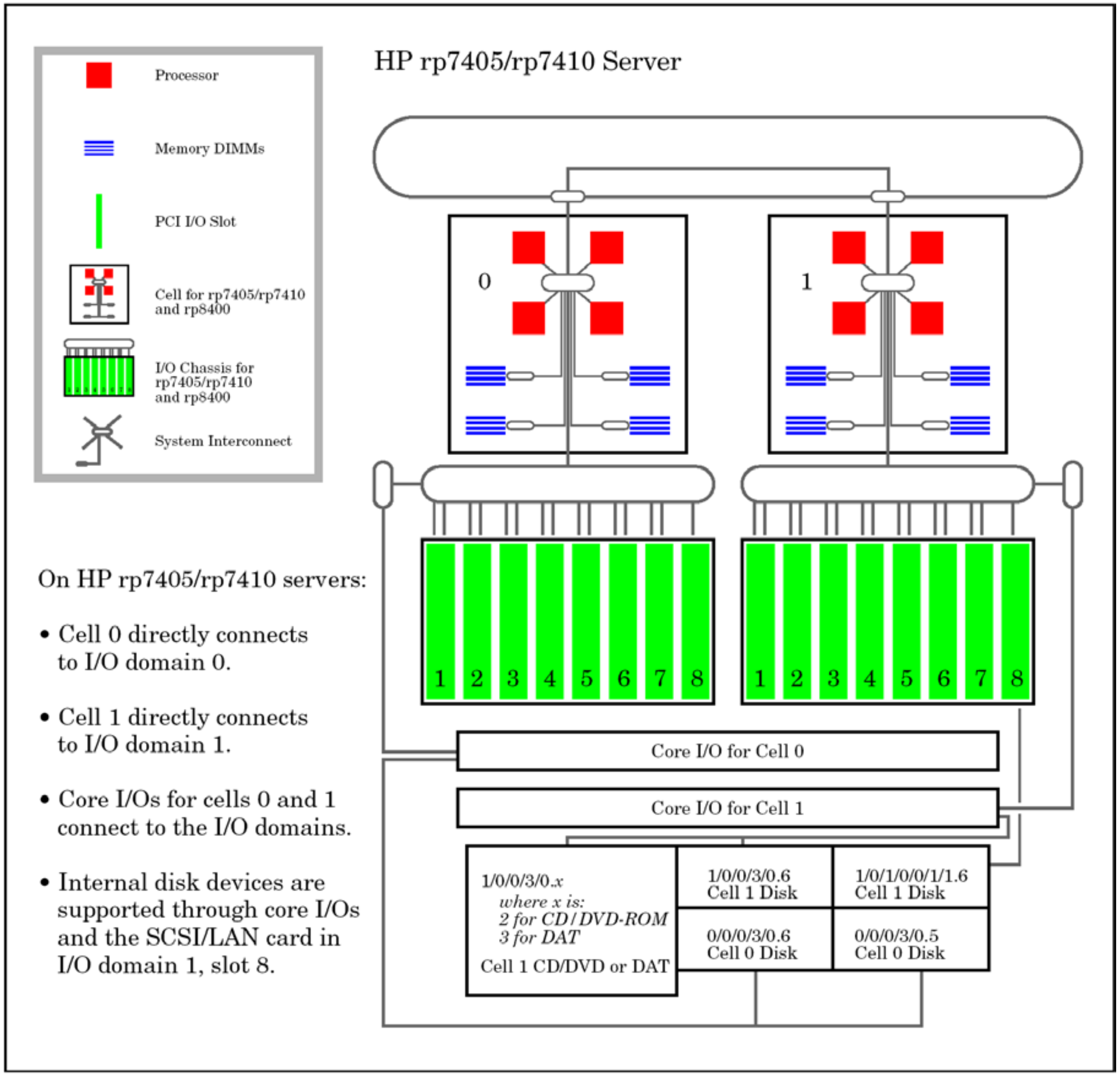
Multiple O/S instances per node with hardware isolation





- Increased system utilization
  - Partitioning a server increases the utilization level. A Superdome can have up to 16 nPartitions
- Increased Flexibility:
  Multi OS
  - Multi OS support: HP-UX, Linux (*), Windows (*)
  - Multi OS version and patch level support
- Increased Uptime
  - Hardware (electrical) and software isolation across nPartitions
  - Serviceguard support (within a Server or to another HP 9000 server)
- Available on Superdome, rp8400 and rp7410

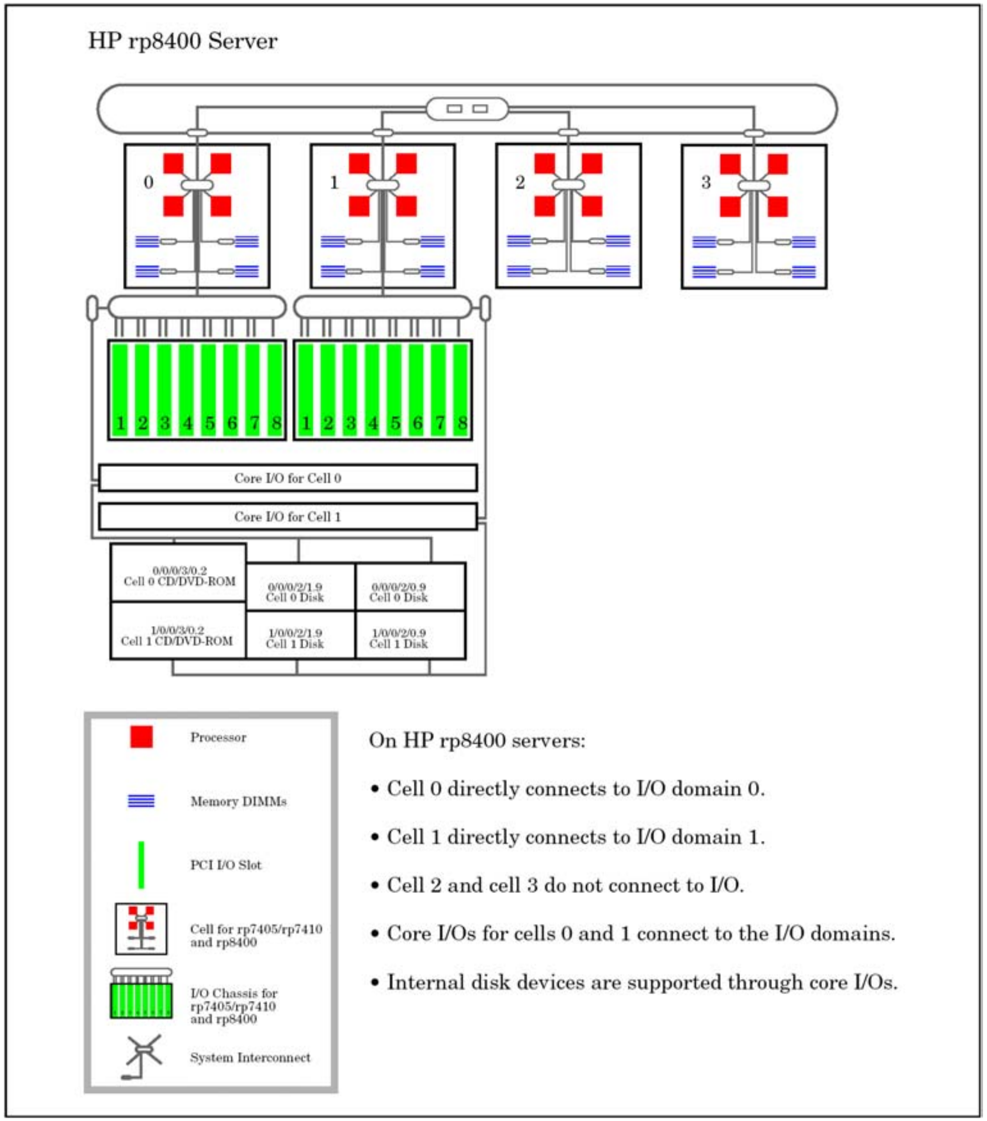
rp7410 nPartitions architecture



On HP rp7405/rp7410 servers:

- Cell 0 directly connects to I/O domain 0.
- Cell 1 directly connects to I/O domain 1.
- Core I/Os for cells 0 and 1 connect to the I/O domains.
- Internal disk devices are supported through core I/Os and the SCSI/LAN card in I/O domain 1, slot 8.

rp8400 nPartitions architecture



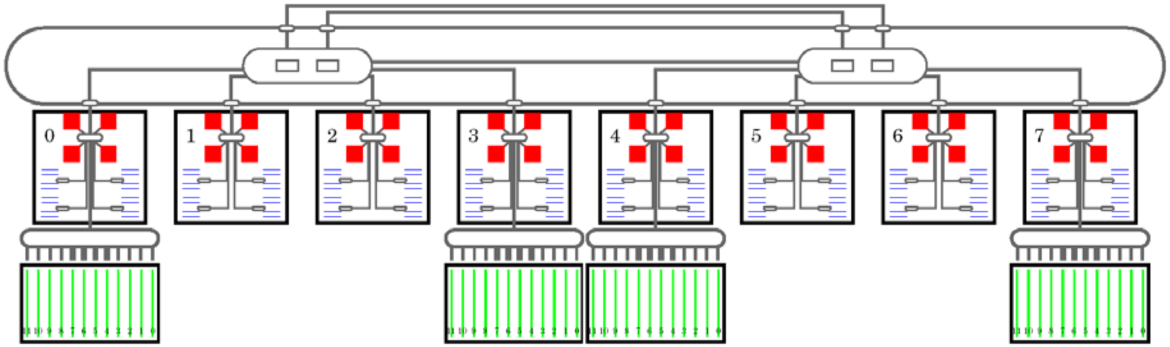
HP rp8400 Server

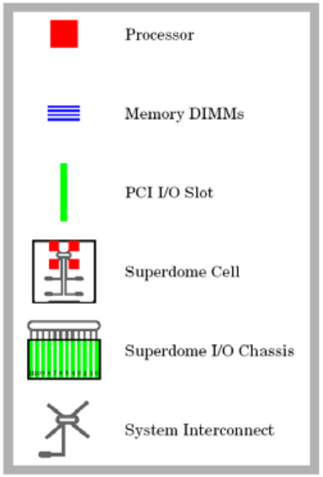
On HP rp8400 servers:

- Cell 0 directly connects to I/O domain 0.
- Cell 1 directly connects to I/O domain 1.
- Cell 2 and cell 3 do not connect to I/O.
- Core I/Os for cells 0 and 1 connect to the I/O domains.
- Internal disk devices are supported through core I/Os.

Superdome 32 way nPartitions architecture

HP Superdome 32-Way Server (SD32000)





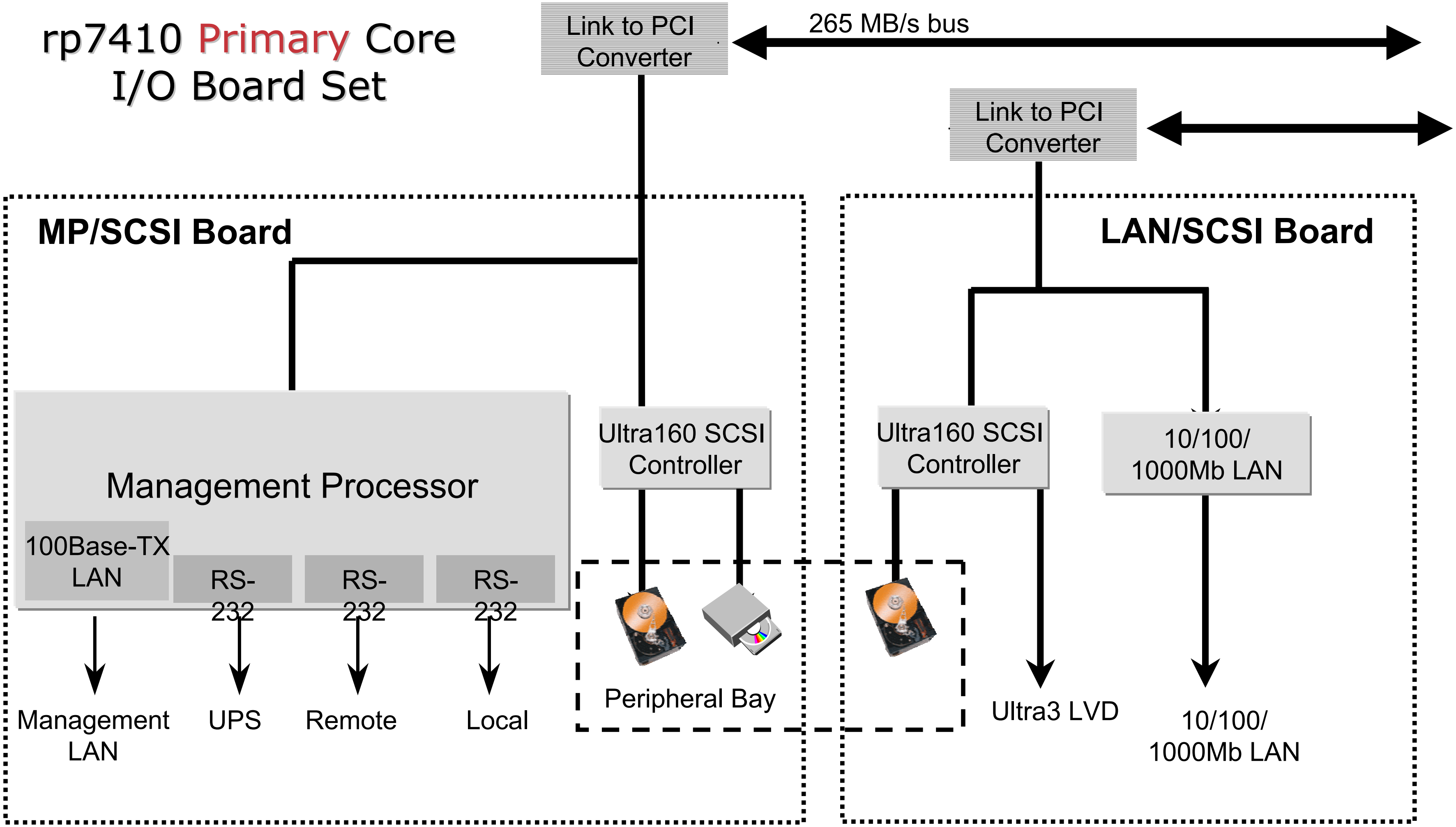
On HP Superdome 32-Way servers:

- Each cell (0–7) can connect to any one of the available I/O chassis.
- Additional I/O chassis can be provided in a connected I/O expansion cabinet.
- PCI card slot 0 in each I/O chassis is for use by a Superdome core I/O card.

Console Access and rp7410 Core I/O

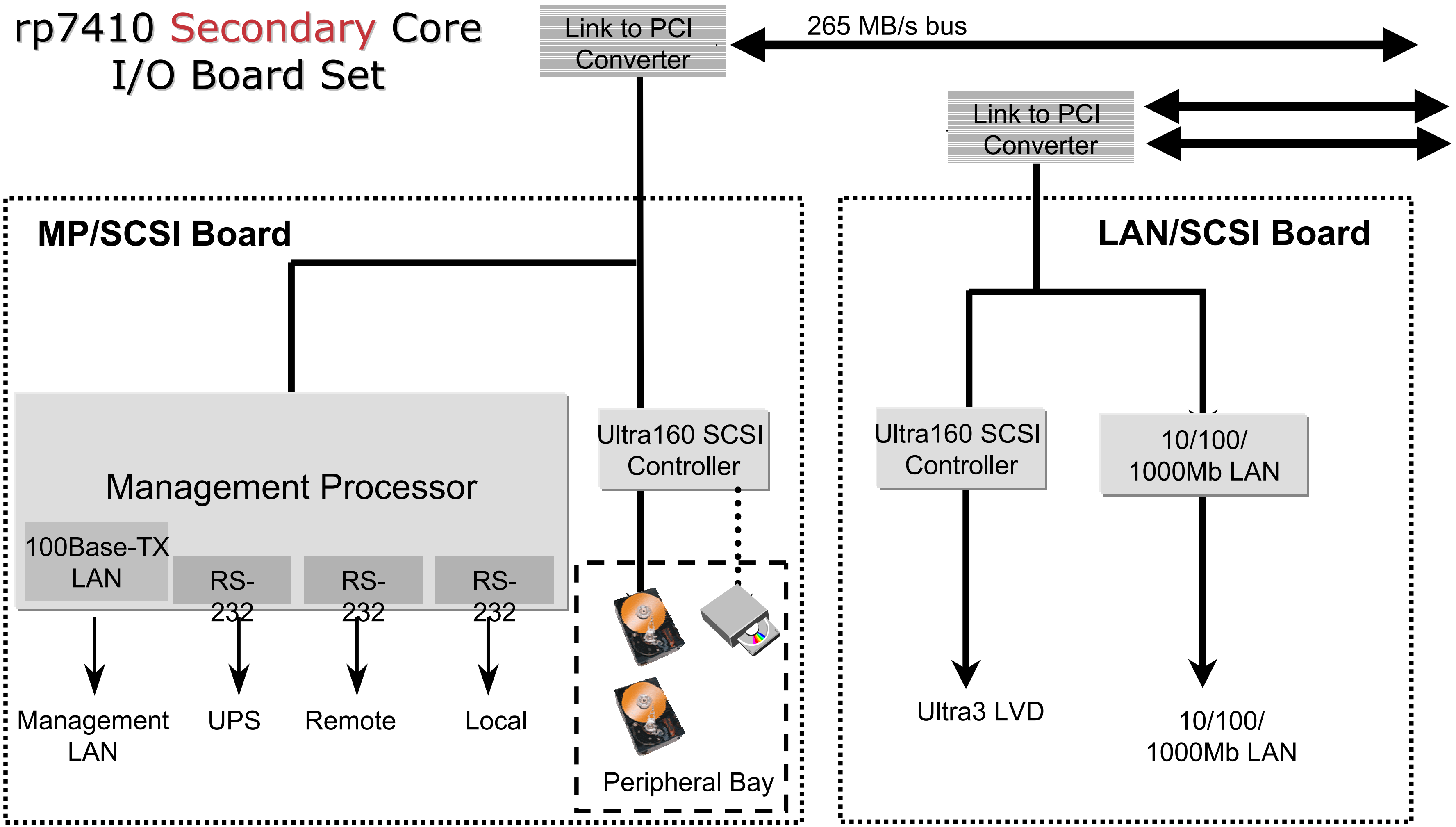
Management Processor Console Access - Primary

rp7410 Primary Core
I/O Board Set

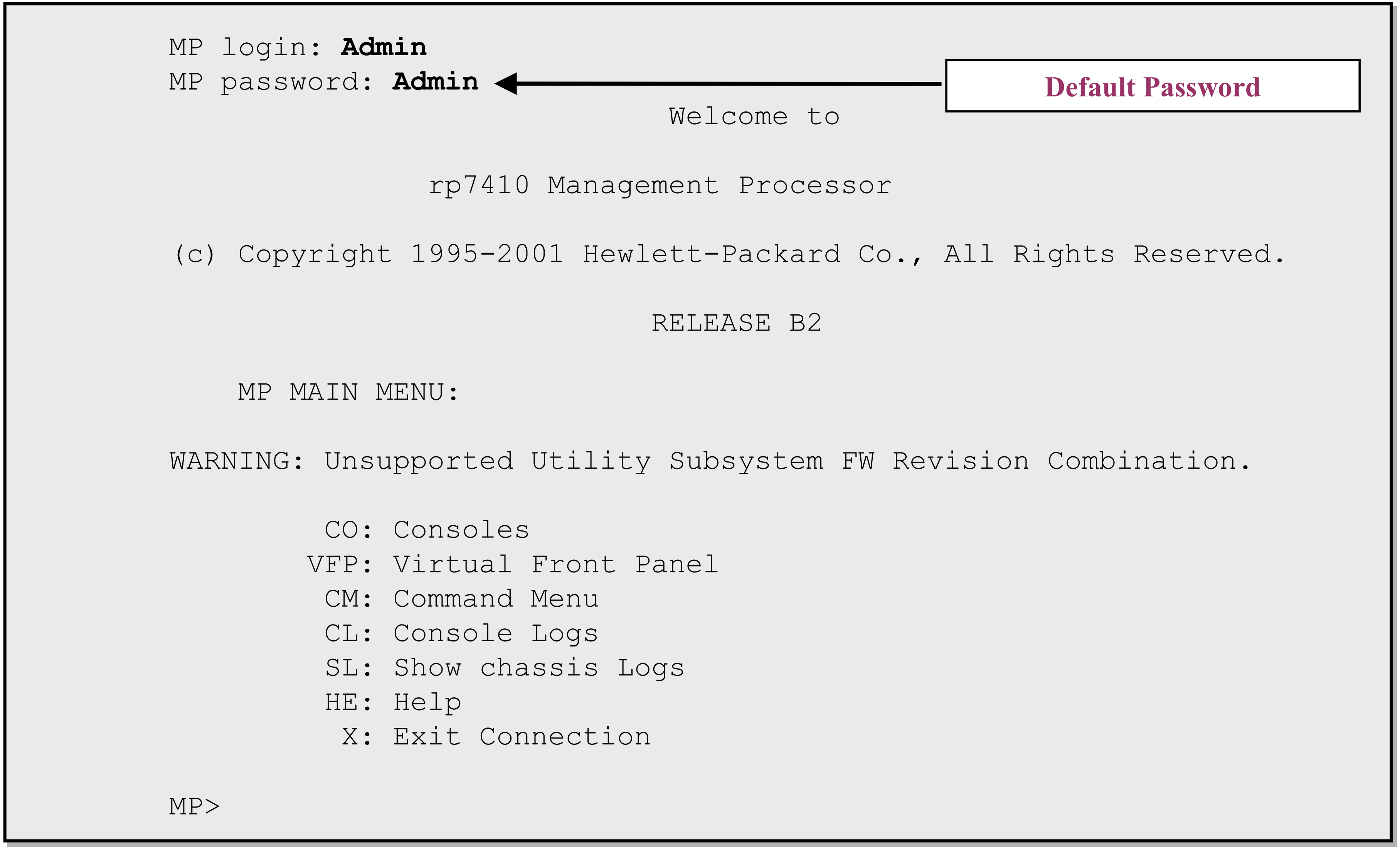


Management Processor Console Access - Secondary

rp7410 Secondary Core I/O Board Set



Management Access: *Logging in to the MP*



```
MP login: Admin
MP password: Admin
                             Welcome to

               rp7410 Management Processor

(c) Copyright 1995-2001 Hewlett-Packard Co., All Rights Reserved.

                            RELEASE B2

    MP MAIN MENU:

WARNING: Unsupported Utility Subsystem FW Revision Combination.

         CO: Consoles
        VFP: Virtual Front Panel
         CM: Command Menu
         CL: Console Logs
         SL: Show chassis Logs
         HE: Help
          X: Exit Connection

MP>
```

Default Password

Virtual Partitions (vPars) Components Overview

vPars Components Overview

- vPars Monitor
- vPars Partition Database
- vPars Daemons
  - vpard
  - vphbd
- Virtual Console Components
  - vcn - virtual console
  - vcs - virtual console slave
  - vpmon console module

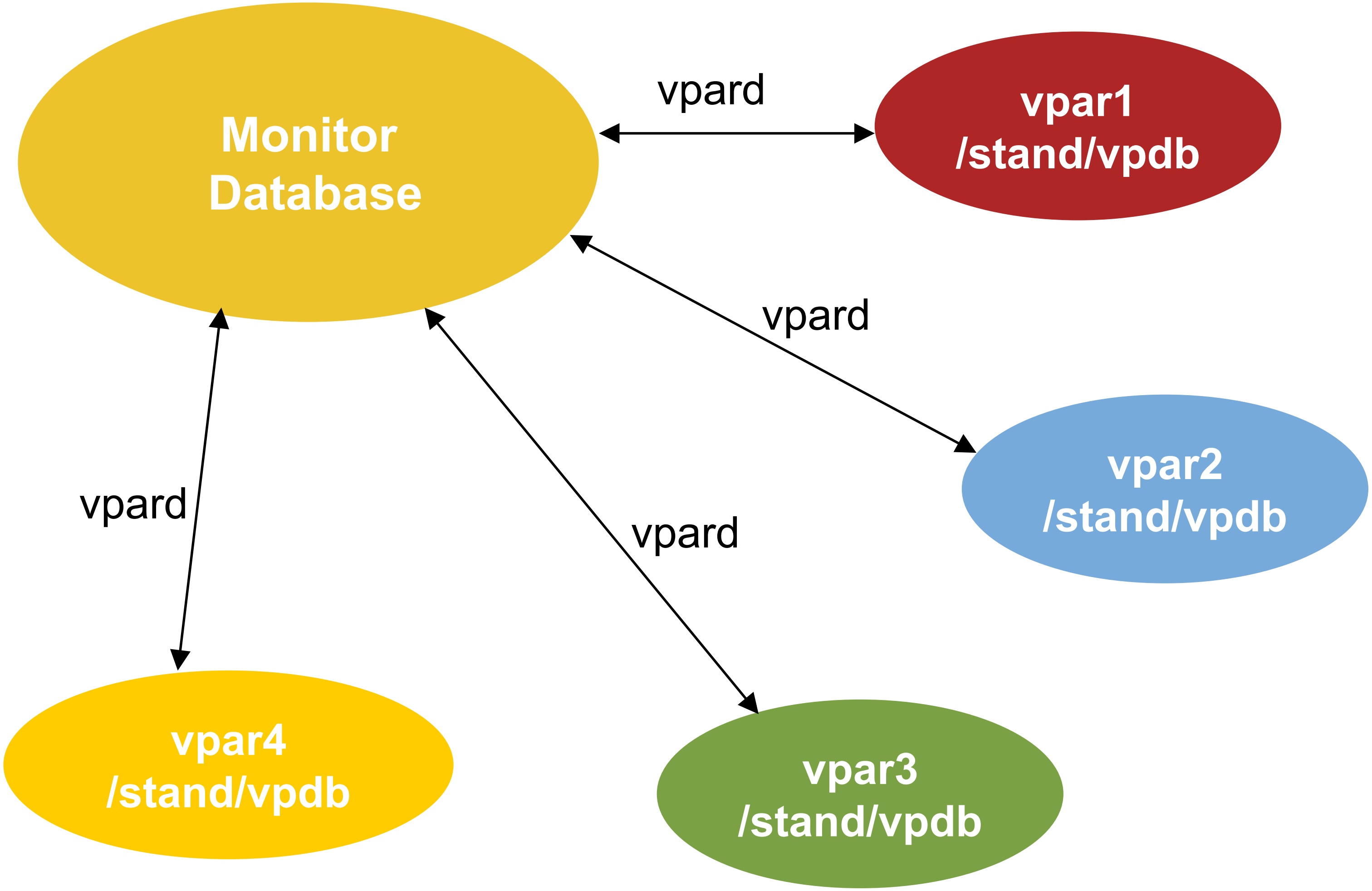
vPars Components Overview vPar Monitor and Database

- vPars Monitor (vpmon)
  - Manages the assignment of hardware resources to virtual partitions
  - Boots virtual partitions and their kernels
  - Emulates certain firmware calls
  - Creates the illusion that each vPar is a complete isolated HP-UX system
- vPars Partition Database (vpdb)
  - Contains partition configuration information
  - Is located in vPars Monitor and a local copy is also stored on each virtual partition
  - An alternate partition database file can be used to create an entirely different virtual partition configuration without affecting the live partition database
- vPars Monitor uses the database to:
  - Track which virtual partition exist
  - Identify hardware resources and partition attributes that are associated with each partition

vPars Components Overview
vPar Monitor and Database

- The vPar daemon “vpard”
  - Is started with “/sbin/init.d/vpard”
  - Synchronizes the databases between memory and all vPars (with a status of “up”)
  - Default synchronization interval is 5 seconds
  - Manages communications with the virtual console
- vphbd (the vPar heartbeat daemon)
  - Is started with “/sbin/init.d/vparhb”
  - Provides a consistent heartbeat status
  - Default 360 second “sleep” between heartbeats
  - A vPar is considered “hung” if 10 heartbeats are “missed”

vPars Components Overview
vPar Monitor and Database

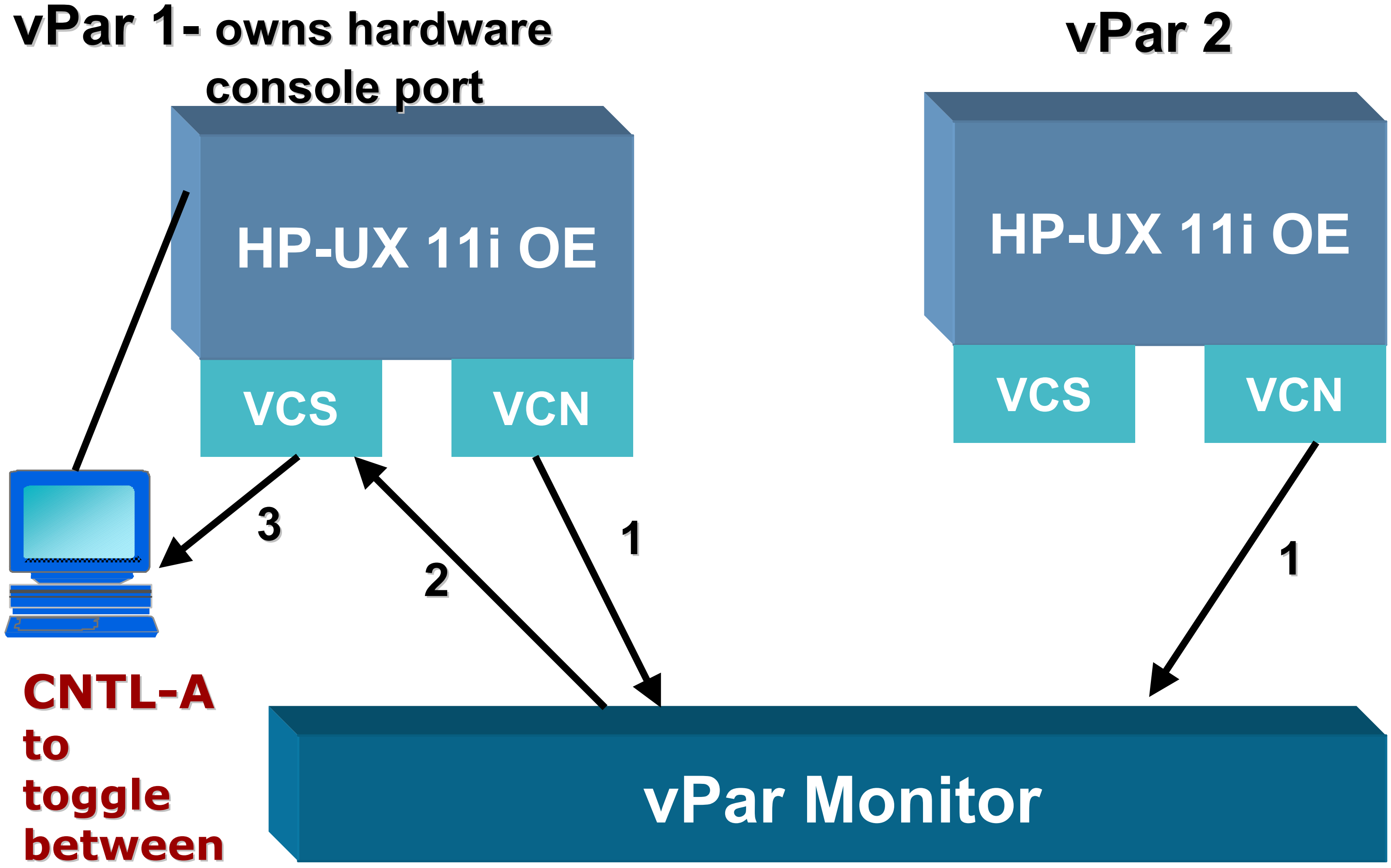


vPars Components Overview Virtual Console

- Virtual Consoles
  - First vPar created must own the Local Bus Adapter (LBA) that contains the physical hardware console port.
  - With vPars, each virtual partition has its own virtual console.
- Virtual Console Access
  - Recommended way to access console
    - Default configuration: each vPar uses the virtual console

      use Ctrl-A to cycle between all live vPars
    - Access h/w console through terminal, lan console, or lantronix
    - Alternative is to give one vPar the hardware console (no other vPar will have access to the console)
- Virtual Console Components
  - The components of the Virtual Console
    - vcn - virtual console
    - vcs - virtual console slave
    - vpmon console module
- Virtual Console works even if vPar that owns the console is down

vPars Components Overview
Virtual Console

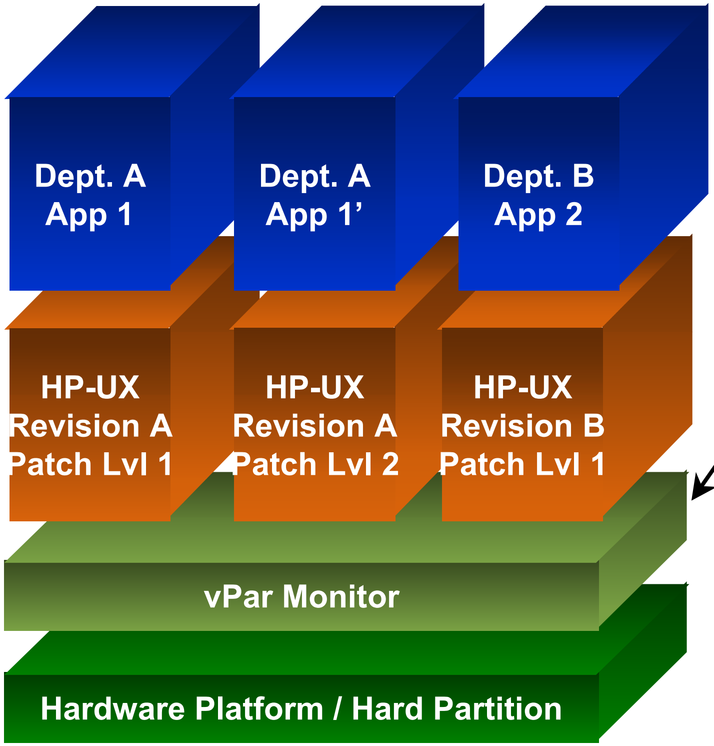


CNTL-A to toggle between vPars

vPars logical overview

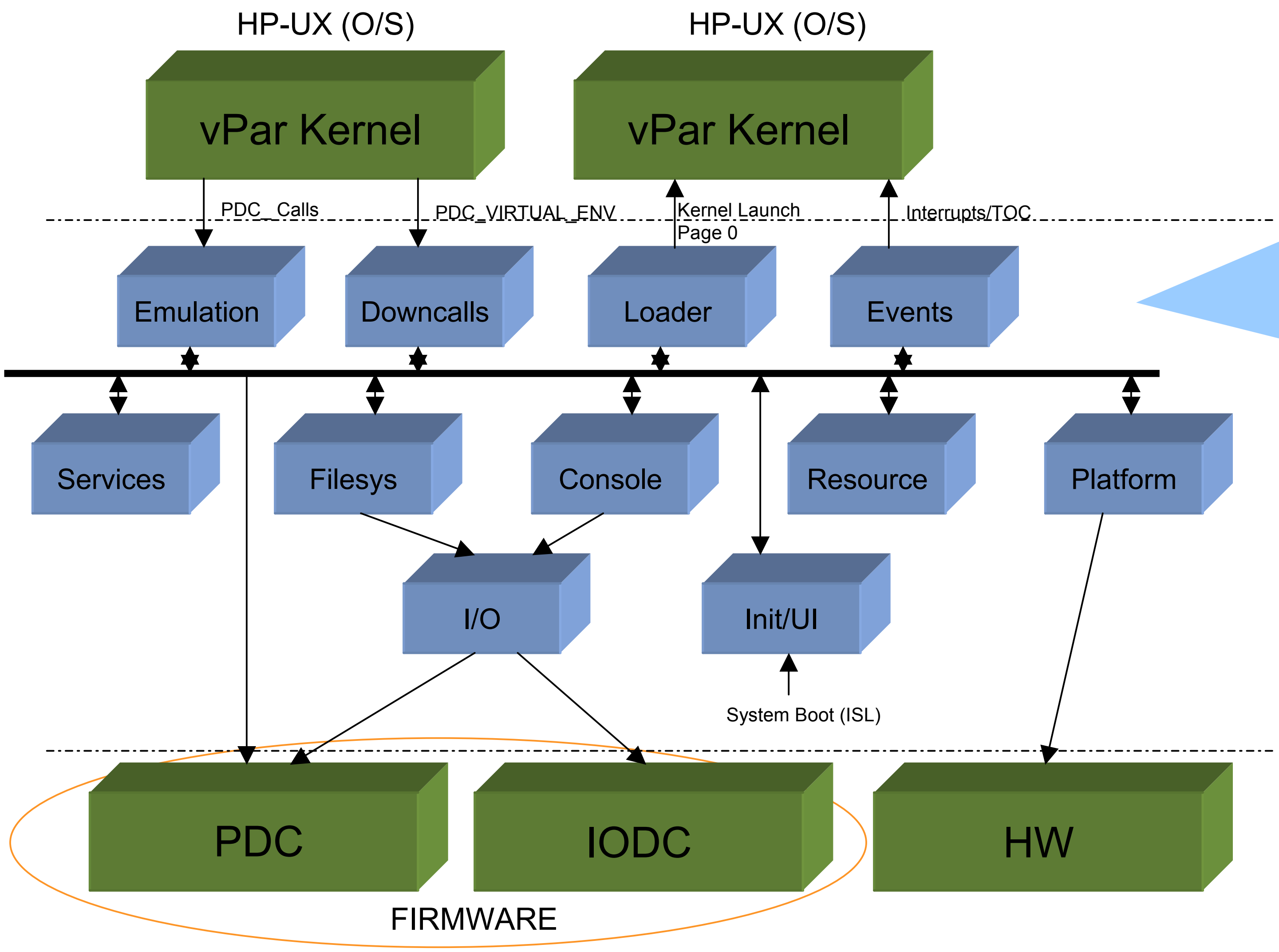
- Multiple applications or multiple instances or versions of the same application
- No name space or resource conflicts

- Creates illusion of many separate hardware platforms
- Manages shared physical resources
- Monitors health of operating system instances



- Each operating system instance tailored specifically for the application(s) it hosts
- Operating systems instances are given a user-defined portion of the physical resources
- No name space or resource conflicts

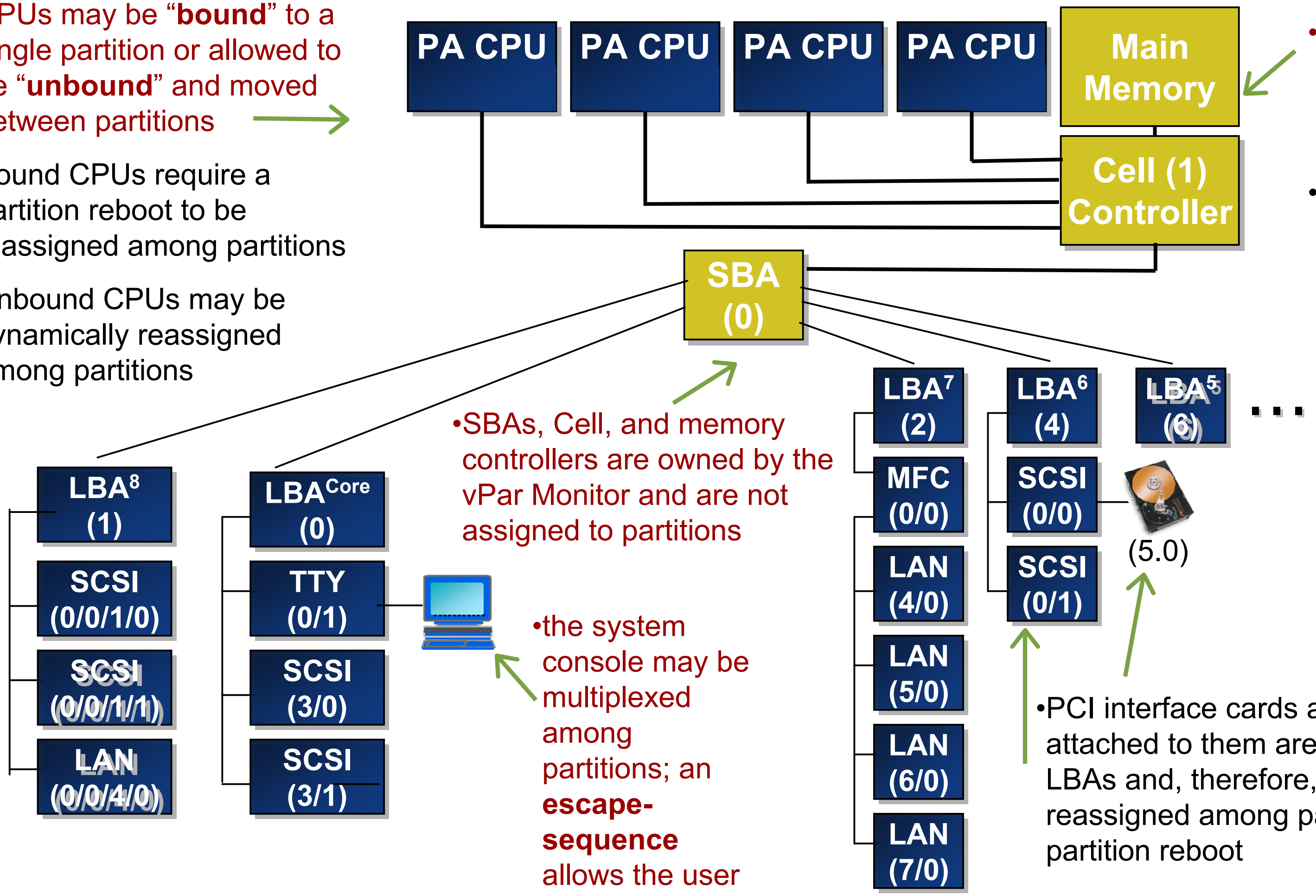
vPar monitor: Between HP-UX 11i (O/S) & firmware



- Minimal performance overhead
- Invoked (mainly) at startup/ shutdown, and during firmware or vPar commands

CPU, Memory & I/O Resources (rp7410 Cell 1)

- CPUs may be "**bound**" to a single partition or allowed to be "**unbound**" and moved between partitions
- Bound CPUs require a partition reboot to be reassigned among partitions
- Unbound CPUs may be dynamically reassigned among partitions



- main memory is allocated to partitions in multiples of **64MB** ranges
- adding or removing a memory range to or from a partition requires a partition reboot

- SBAs, Cell, and memory controllers are owned by the vPar Monitor and are not assigned to partitions

- the system console may be multiplexed among partitions; an **escape-sequence** allows the user to toggle among partitions

- **LBAs** are pinned to a single partition
- adding or removing LBAs to or from a partition requires a partition reboot

- PCI interface cards and the devices attached to them are connected through LBAs and, therefore, cannot be logically reassigned among partitions without a partition reboot

Installation and Configuration Overview

Installation Overview

- Plan the vPars configuration
- Choose one of two possible methods of installation of the boot disks for the partitions
  - cold install all partitions
  - cold install first partition and use Ignite-UX for remaining partitions
- Perform the appropriate installation procedure for the method selected

Installation Method #1

- Sequentially do the following for the boot disk of **each** vPar:
  - cold install HP-UX
  - swinstall the vPars software
- From the first vPar (booted as a standalone system), create all partitions
- Shutdown the standalone system
- boot the vPars monitor
- boot each of the vPars

Installation Method #2

- Do the following for the boot disk of the first vPar:
  - cold install HP-UX
  - swinstall the vPars software
  - create all partitions
- Shutdown the first vPar
- Boot the vPars monitor
- Boot the first vPar
- Set up an Ignite-UX Server (can use first vPar)
- Use “vparboot -I” to install remaining vPars from the Ignite-UX Server

Configuring vPars

Command Overview

- **vparcreate -p** *vp_name* [**-B** *boot_attr*] [**-D** *db_file*] [**-S** *static_attr*] [**-b** *kernel_path*] [**-o** *boot_opts*] [**-a** *rsrc*]...
- **vparmodify -p** *vp_name* [**-B** *boot_attr*] [**-D** *db_file*] [**-S** *static_attr*] [**-b** *kernel_path*] [**-o** *boot_opts*] [**-P** *new_vp_name*] [**-a** *rsrc*]... [**-m** *rsrc*]... [**-d** *rsrc*]...
- **vparstatus** [**-v** | **-M**] [**-p** *vp_name*]... [**-D** *db_file*]

  vparstatus -A [**-M**]
- **vparboot -p** *vp_name* [**-b** *kernel_path*] [**-o** *boot_opts*] [**-B** *boot_addr*]

  vparboot -p *vp_name* **-I** *ignite_kernel*
- **vparremove -p** *vp_name* [**-D** *db_file*] [**-f**]
- **vparreset -p** *vp_name* [**-h**|**-t**] [**-q**] [**-f**]

Resource Specifications

memory	mem::size	note:size in MB
cpu	cpu:path:num:min:max	
io	io:path	

- other vPars attributes and boot settings

– autoboot	auto | manual
– static	dynamic | static
– boot paths	boot | alternateboot
– boot options	-is | -lq | -lm
– kernel path	

Note: See vparresources(5) for more information.

Techniques for Creating vPars

- Partitions can be fully created with one vparcreate command

```
# vparcreate –p vpar1 –a mem::512 –a cpu::2  \
-a cpu:::1:3 –a cpu:0/10 –a io:0/0/0        \
-a io:0/0/0/3/0.6.0:BOOT
```

- Partitions can also be created with a series of vparcreate and vparmodify commands

```
# vparcreate –p vpar1 –a mem::512 –a cpu::2
# vparmodify –p vpar1 –m cpu:::1:3 –a cpu:0/10
# vparmodify –a io:0/0/0 –a io:0/0/0/3/0.6.0:BOOT
```

Creating vPars

- To create two partitions without assigning any resources

```
# vparcreate –p vpar1
# vparcreate –p vpar2
```

- To make changes

  Use “vparmodify –P…” to rename partition
  Use “vparremove –p …” to remove partition

Adding Memory Resources

- Resource specification

  mem::size

- To add memory resources to each partition

```
# vparmodify  -p vpar1  -a mem::512
# vparmodify  -p vpar2  -a mem::512
```

- To make changes

  Use “vparmodify –d mem…” to delete

  Use “vparmodify –m mem…” to change

Adding CPU Resources

- Resource specification: cpu:path:num:min:max
- To add one bound and one unbound CPU to one of the partitions, specify the hardware path for the bound CPU, and set a total limit of 3 CPUs

```
# vparmodify  -p vpar1  -m cpu::2  -m cpu:::1:3 –a cpu:0/10
```

- To make changes

  Use “vparmodify –d cpu…” to delete
  Use “vparmodify –m cpu…” to change

Note: **num** and **min:max** and **path** cannot be combined into one option and must be separated as shown in the example above. See vparresources(5) for more information.

Adding IO Resources

- Resource specification

  io:path

- To add IO resources to the two partitions

```
# vparmodify  -p vpar1  -a io:0/0/0
# vparmodify  -p vpar2  -a io:1/0/0
```

- To make changes

```
# vparmodify –p vpar1 –d io:0/0/0
```

Setting Partition Attributes

- To set primary boot path

```
# vparmodify  -p vpar1  -a io:0/0/0/3/0.6.0:BOOT
```

- Other handy vparmodify options

-B	Sets autoboot attribute (auto | manual)
-S	Sets static attribute (dynamic | static)
-b	Sets absolute kernel path
-o	Sets boot options (normal | -is | -lq | -lm)
-d	Delete resource
-m	Modify resource

vparstatus Output

```
# vparstatus
 [Virtual Partition]
                                                                    Boot
 Virtual Partition Name          State Attributes Kernel Path       Opts
 ============================== ===== ========== ========================
 vpar1                            Up    Stat,Auto  /stand/vmunix      -lq
 vpar2                            Up    Dyn,Manl   /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
 ==============================   ================  ========================
 vpar1                             2/   4    2   1     3    2/640          704
 vpar2                             2/   4    2   2     2    2/704          768
```

vPars Boot Sequence

- Machine Firmware (PDC/BCH)
- ISL is loaded from LIF area of primary boot path as set in BCH
- hpux secondary loader is loaded from same LIF area
- Boot string is read from AUTO file in same LIF area. The administrator would typically set this string to

```
hpux  /stand/vpmon  vparload –auto
```

- /stand/vpmon loaded from primary boot path as set in BCH
- /stand/vmunix loaded from the primary boot path for each partition – the actual behavior of each partition is defined by attributes that are recorded in /stand/vpdb

Accessing the Monitor Prompt

- From ISL (booting the Monitor)

```
ISL> hpux /stand/vpmon
```

- From HP-UX (shell prompt) within a vPar

```
vpar1: # <cntl>-a
vpar2: # <cntl>-a
MON>
```

  - Note, the Monarch processor must be unassigned

- After shutting down all vPars

```
#  shutdown -hy 0
```

Monitor Commands

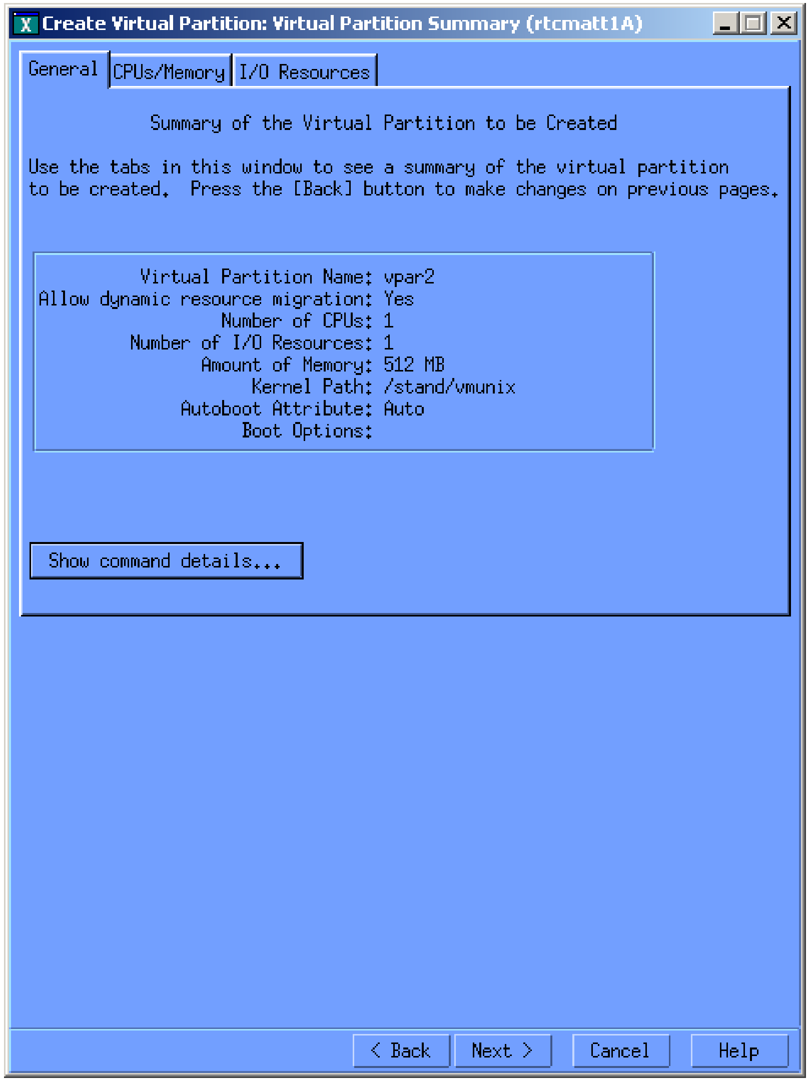
readddb
vparload
reboot

Boot Commands

cat
cbuf
help (or ?)
lifls
getauto
log
ls
scan
toddriftreset
vparinfo

Information Display Commands

vPars Manager GUI vparmgr

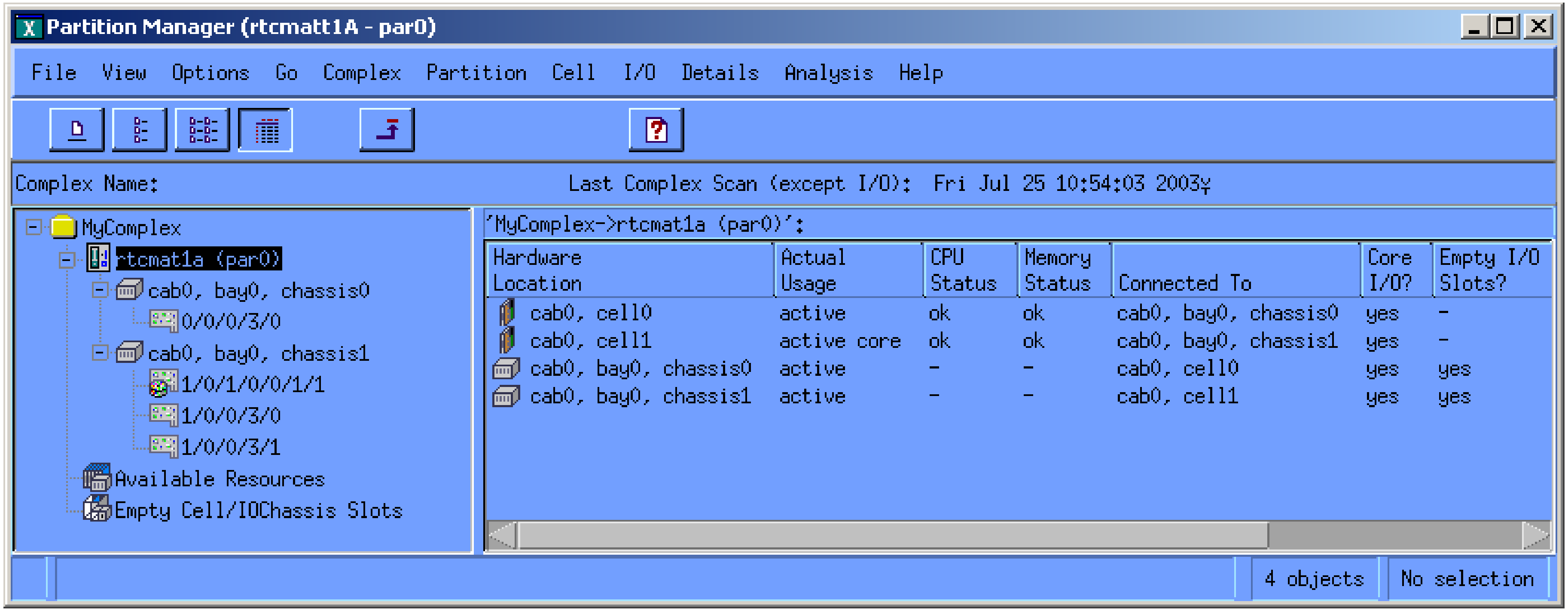

Create Virtual Partition: Virtual Partition Summary (rtcmatt1A)
General
CPUs/Memory
I/O Resources

Summary of the Virtual Partition to be Created

Use the tabs in this window to see a summary of the virtual partition to be created. Press the [Back] button to make changes on previous pages.

```
           Virtual Partition Name: vpar2
Allow dynamic resource migration: Yes
                   Number of CPUs: 1
          Number of I/O Resources: 1
                 Amount of Memory: 512 MB
                      Kernel Path: /stand/vmunix
               Autoboot Attribute: Auto
                     Boot Options:
```

nPartition Manager GUI
parmgr



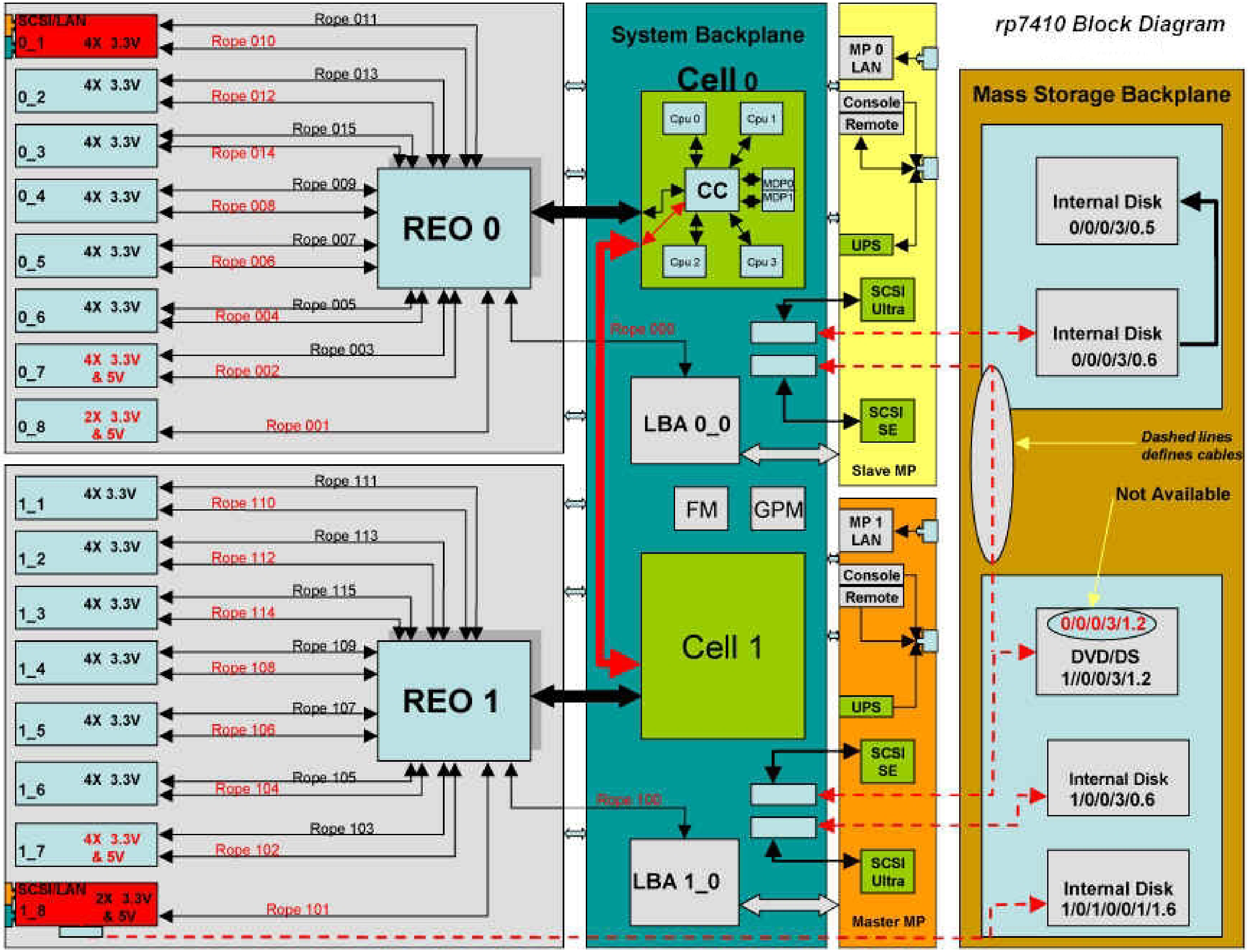
Complex Name: Last Complex Scan (except I/O): Fri Jul 25 10:54:03 2003

- MyComplex
  - rtcmat1a (par0)
    - cab0, bay0, chassis0
      - 0/0/0/3/0
    - cab0, bay0, chassis1
      - 1/0/1/0/0/1/1
      - 1/0/0/3/0
      - 1/0/0/3/1
  - Available Resources
  - Empty Cell/IOChassis Slots

'MyComplex->rtcmat1a (par0)':

Hardware Location	Actual Usage	CPU Status	Memory Status	Connected To	Core I/O?	Empty I/O Slots?
cab0, cell0	active	ok	ok	cab0, bay0, chassis0	yes	-
cab0, cell1	active core	ok	ok	cab0, bay0, chassis1	yes	-
cab0, bay0, chassis0	active	-	-	cab0, cell0	yes	yes
cab0, bay0, chassis1	active	-	-	cab0, cell1	yes	yes

vPars Labs –rp7410 Block Diagram



vPars Labs

- Partition Planning

vPars Labs

- Lab 1 – Creating and Booting the First vPar

vPars Labs

- Lab 1 – Creating and Booting the First vPar

vPars Labs

- Lab 2 – Creating and Booting a Second vPar

vPars Labs

- Lab 3 – Dynamic CPU Migration

vPars Labs

- Lab 4 – Using the vPar GUI (optional)

vPars Labs

- Lab 5 – vPars Booting (optional)

vPars Labs

- Lab 5 – vPars Booting (optional)

vPars Labs

- Lab 6 – nPartition Reconfiguration (optional)

HP WORLD 2003

Solutions and Technology Conference & Expo

Interex, Encompass and HP bring you a powerful new HP World.

interex
shared knowledge • shared power

encompass
AN HP USER GROUP

hp
invent