



# vPars Presentation



**Asghar Riahi**  
**Senior Technical Consultant**  
**Hewlett-Packard**

© 2004 Hewlett-Packard Development Company, L.P.  
The information contained herein is subject to change without notice



# Agenda

- **What is Virtual Partitioning**
- **HP Partitioning Continuum for Always On**
- **Where HP vPars provides the most value**
- **Definition of Partitioning**
- **vPars w/ WLM**
- **What is vPars**
- **vPars Components Overview**
- **How vPars works**
- **Partitioning Plan**
- **vPars commands**
- **vPars Installation**
- **MC/SG and vPars**
- **vPars GUI vparmgr**
- **Summary**

# What is Virtual Partitioning

Partitions are physical or logical mechanisms for isolating operational environments within single or multiple servers to offer the flexibility of dynamic resizing while ensuring that applications can enjoy protection from unrelated events that could otherwise cause disruption, interruption or performance degradation

- In traditional server environments, all CPUs within a server run a single OS instance and one or more applications. Application and OS failures may affect the entire system.
- With HP-UX Virtual Partitions, however, the software stack contains the vPar Monitor between the hardware/firmware level and OS, allowing each partition to support a different instance of HP-UX 11i (or later) with each instance capable of being at a different version and patch level.

# HP Partitioning Continuum for Always On

**HyperPlex**  
*hard partitions with multiple nodes*  
 (City w/ houses)



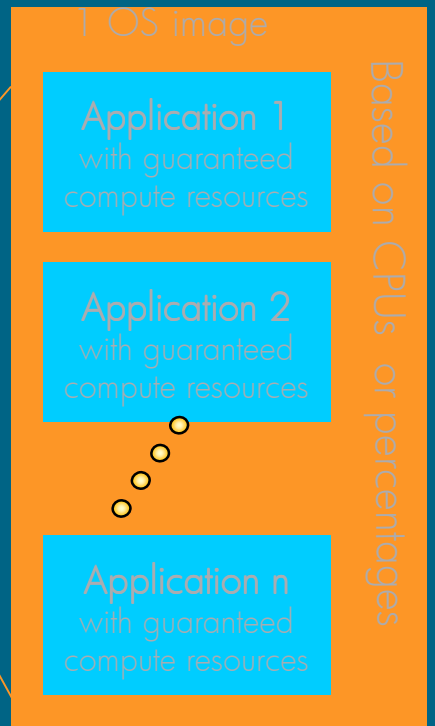
**nPartitions**  
*hard partitions within a node*  
 (House w/ apartments)



**Virtual Partitions** *within a hard partition*  
 (Apartments with rooms) **New!!**



**Resource partitions**  
*within a single OS image*  
**PRM with psets** **New!!**  
 (Rooms with closets)



hp-ux wlm (workload manager) -automatic goal-based resource allocation via set SLOs

**Isolation** highest degree of separation **Flexibility** highest degree of dynamic capabilities

# HP-UX 11i

Internet critical. E-services ready.



Hard Partitions with multiple nodes

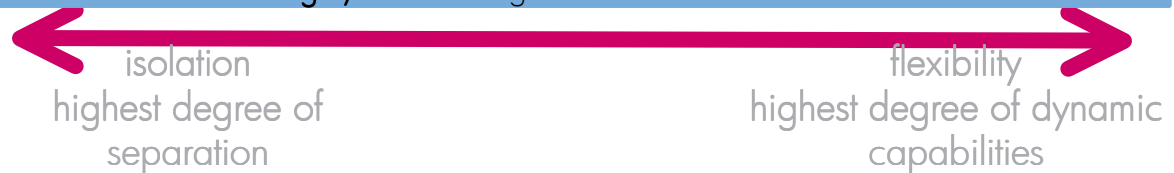
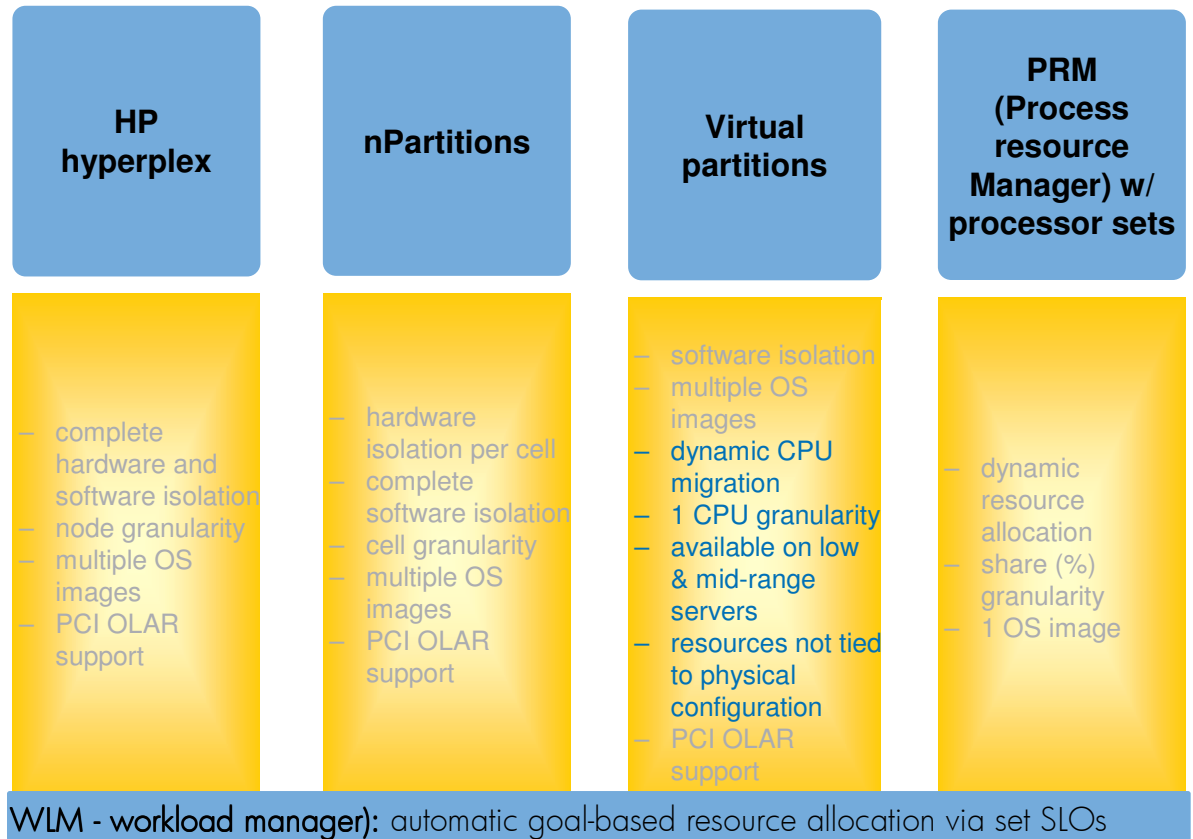
Hard Partitions within a node

Virtual Partitions within a hard partition

PRM with psets resource partitions within an OS image

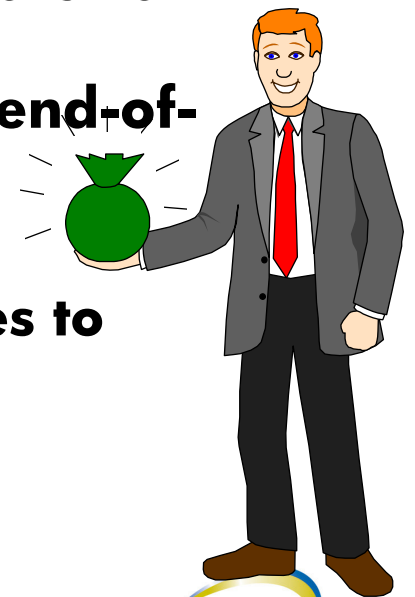
*Partitioning continuum*  
HP-UX partitioning offerings are all integrated.

They have been designed to complement each other to provide greater flexibility.



# Where HP vPars provides the most value

- ① **System/data center consolidation**
- ① **development/test environments**
- ① **increased system utilization**
- ① **varying workload requirements:**
  - **time of day: order entry during day, batch at night**
  - **time of month (payroll, end-of-month/end-of-year financials)**
  - **as particular needs require**
- ① **service provider (providing system resources to different users/applications)**
- ① **unique application tuning of O/S**
- ① **time zoning**



# Definition of Partitioning

Partitions are physical or logical mechanisms which:

- Isolate operational environments within single or multiple servers
- Offer flexibility of dynamic resizing
- Protect applications from unrelated events that could cause disruption, interruption, or performance degradation

## Hard partitions

Electrical isolation

Can perform hardware maintenance on one without affecting the others

Suitable for mixing production and R&D environments

Suitable for hosting multiple customers or different financial entities

## Soft/virtual partitions

Software isolation

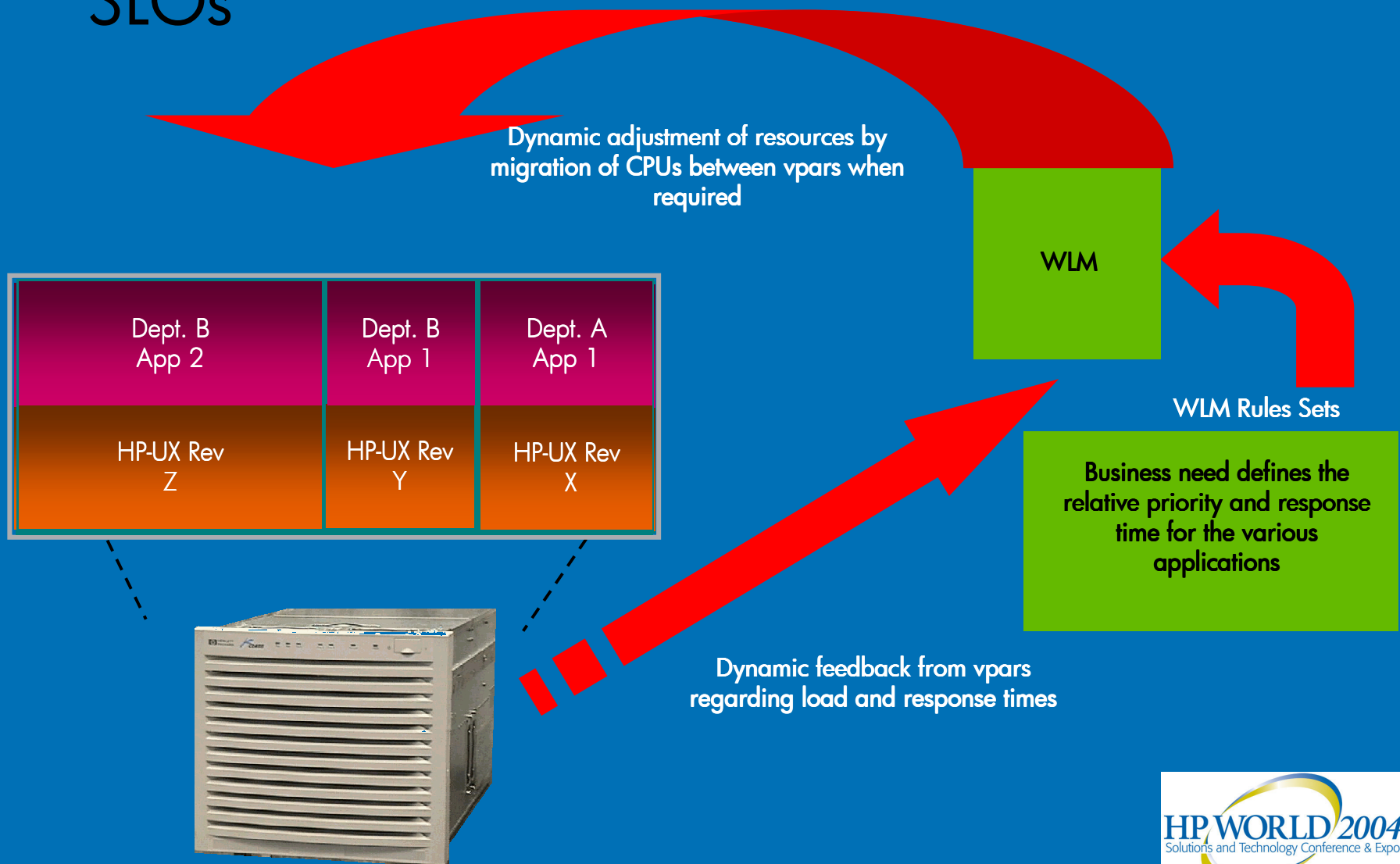
Cannot perform hardware maintenance on one without affecting the others

Not suitable for mixing R&D and production environments

Not suitable for mixing customers or different financial entities



# vPars w/ WLM: automatic goal-based resource allocation between vPars based on SLOs



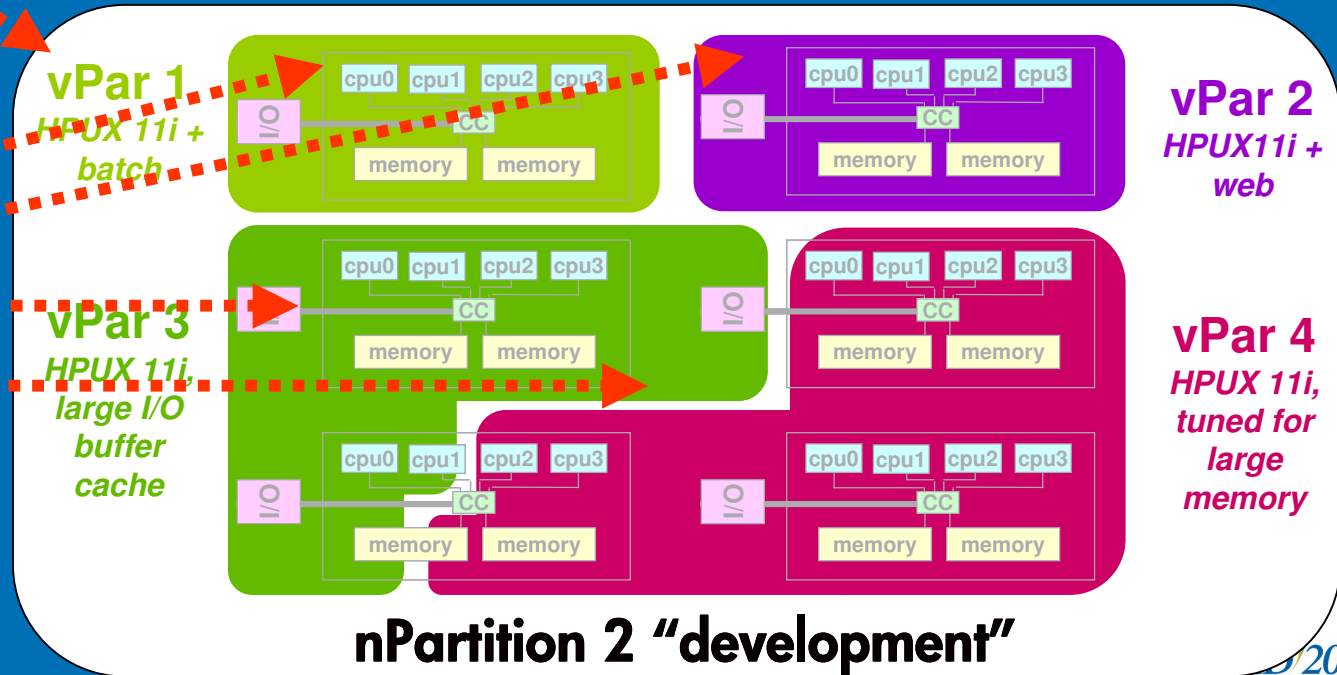
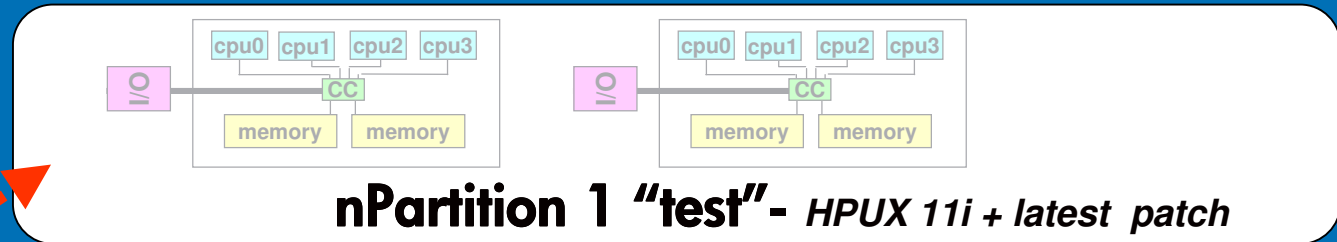
# HP Partitioning- vPars within nPars

- nPartitions provide electrical hardware isolation - separate systems, I/O, boot, CPU, memory etc.
- vPars are software isolated (OS, middleware, apps) for different customers, different OS tuning - each tailored by number of CPUs, amount of RAM, amount of I/O per customer and application mix



• **electrical h/w isolation**

• **s/w isolation**  
 • **dynamic reconfiguration**  
 • **single CPU granularity**  
 • **low to high servers**  
 • **resources not tied to physical configuration**





## INCREASED FLEXIBILITY:

- ✓ Integration with nPartitions
- ✓ Integration with iCOD

## SIMPLIFIED MANAGEMENT:

- ✓ Automatic, SLO-based workload management ACROSS vPars (WLM cross-vPar integration)
- ✓ Availability of new vparmgr GUI

## EASIER DEPLOYMENT:

- ✓ Ignite/UX is now vPar-aware (special WINSTALL file not needed)

## A.02.02 PLATFORM SUPPORT:

- ✓ rp5405, rp5470 (L3000), rp7400/ N400, rp7405, rp7410, rp8400, Superdome

**new!** HP Virtual Partitions:

Multiple applications on the same server with software partition


**1st Unix software partitioning**

**core functionality for free!**

**New w/ HP-UX 11i**

Dept. A App 1	Dept. A App 1'	Dept. B App 2	Dept. B App 3
HP-UX Revision A.1	HP-UX Revision A.2	HP-UX Revision B.3	HP-UX Revision B.3

vPar Monitor



**Built on HP resiliency**

**with performance & availability**

## increased system utilization

- up to 80-90%, since additional partitions can use unused parts of system

## greater flexibility

- multiple independent OSs
- ❖ 1 CPU granularity per vPar
- ❖ dynamic movement of CPU power between vPars
- ❖ resources not tied to physical config. (like hard partitions)
- ❖ vPars on low to high-end

## isolation

- of OS, applications, resources
- individual reconfig & reboot



# Hard Partitions

- Designed to isolate applications from single point of failures
- Full hardware isolation
  - consists of multiple nodes
  - single node
- Can be divided into virtual partitions

# HP-UX Virtual Partitions

## HP-UX Virtual Partitions (vPars)

Partitioning Technology that provides applications and operating system isolations

Virtual Partitions can be created to run multiple instances of HP-UX 11i simultaneously on:

- A Single Server Node
- nPartitions

Each partition is assigned its own resources and hosts its own set of applications

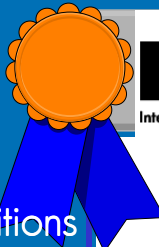
- Multiple applications or multiple instances of same application may coexist on single system

# vPars Benefits

## Benefits of using vPars

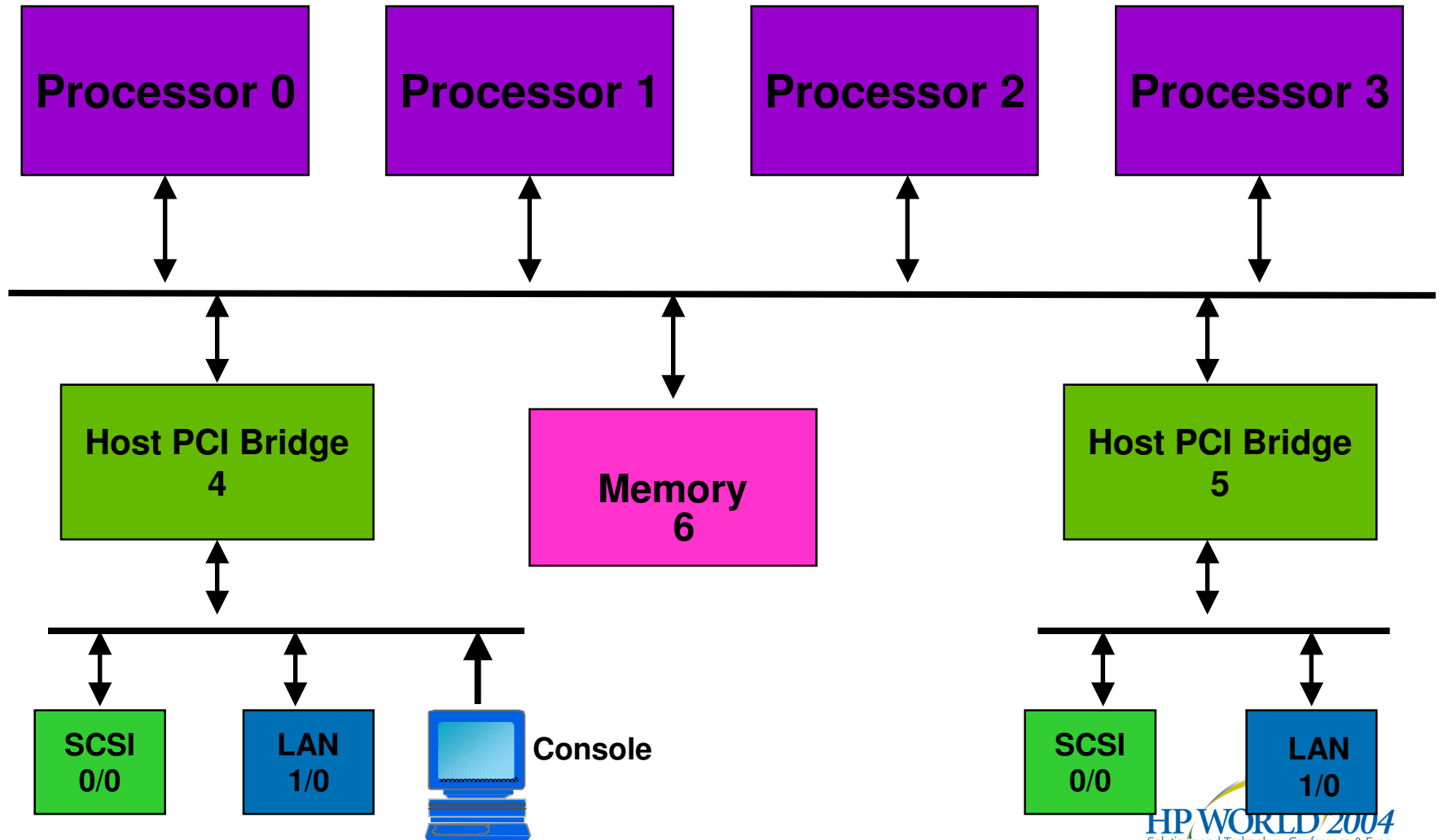
- Increased System Utilization
- Isolation of OS/OE & Applications faults
- Improved flexibility in configurations
- Improved system availability and scalability

# The most comprehensive partitioning offering for the UNIX operating system

	 <b>HP-ux 11i</b> <small>Internet critical. E-services ready.</small>	Solaris™ 8	AIX L™ v5.1
Hard partitions	<b>Yes</b> nPars	<b>Yes</b>	<b>No</b>
Soft partitions	<b>Yes</b> vPars	<b>No</b>	<b>Yes</b>
Integration with goal-based workload mgmt	<b>Yes</b>	<b>No</b>	<b>No</b>
Dynamic reconfig. of hard partitions	<b>Yes</b>	<b>Yes</b>	<b>No</b>
Dynamic reconfig. of soft partitions	<b>Yes</b>	<b>No</b>	<b>No</b>

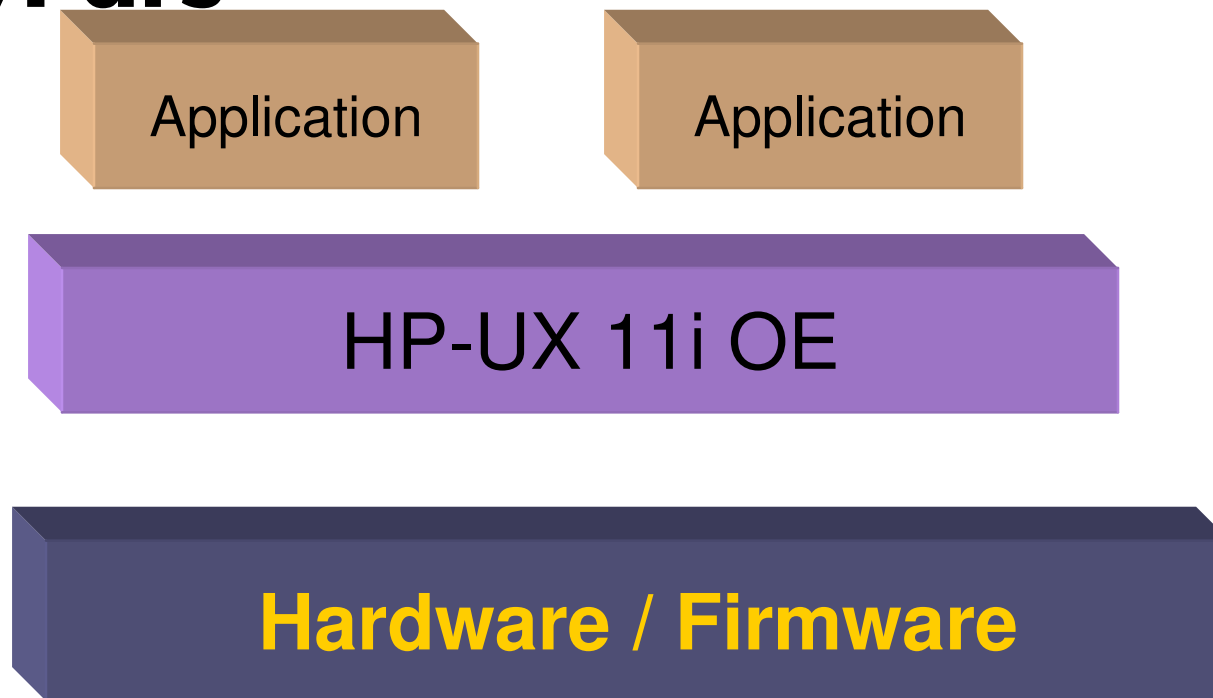


# System without vPars

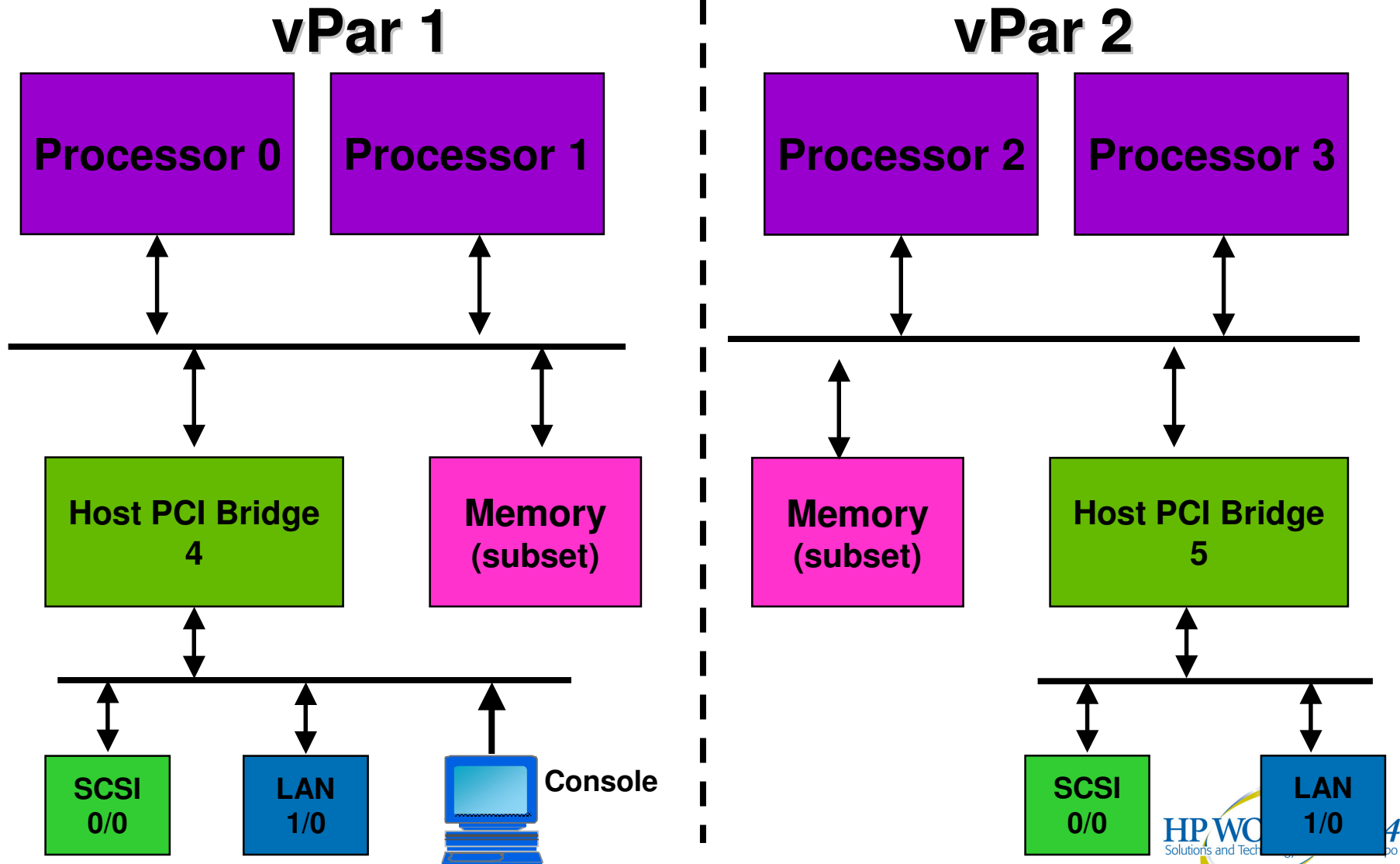


# System without vPars

## – Software stack on system without vPars



# System with vPars



# vPars Components Overview

- **vPars Monitor**
- **vPars Partition Database**

## vPars Monitor

- **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

- **Contains partition configuration information**
- **Is located in vPars Monitor and a local copy is also stored on each virtual partition**
- **vPars Monitor uses the database to:**
  - Track which virtual partition exist
  - Identify hardware resources and partition attributes that are associated with each partition

# vparstatus

- **SYNOPSIS**

```
vparstatus [-v | -M] [-p vp_name].  
[-D db_file]
```

```
vparstatus -A [-M]
```

```
vparstatus -w
```

```
vparstatus -e
```

```
vparstatus -R [-p vp_name]
```

## Example: vparstatus

- **To show summary of all attributes and resources of all virtual partition:**

```
# vparstatus
```

- **To display detailed attributes and resources of partition vpar1:**

```
# vparstatus -p vpar1 -v
```

- **To display vpar2's resources in machine readable format:**

```
# vparstatus -p vpar2 -M
```

- **To see available resources that have not been assigned to any partitions:**

```
# vparstatus -A
```



# Boot Sequence

- With vPars, the monitor layer is added to the boot sequence
  - **ISL**                    **Initial System Loader**
  - **hpux**                    **Secondary System Loader**
  - **/stand/vpmon (vPars monitor and partition database)**
  - **/stand/vmunix**                    **(kernel of the virtual partitions)**

## Example: Boot Sequence

- To load hpux with no partitions

```
ISL> hpux /stand/vmunix
```

- To boot vPars monitor

```
ISL> hpux /stand/vpmon <options>
```

NOTE: if no options are given, monitor will be in interactive mode

- To boot vPar from the monitor

```
MON> vparload -p <vparName>
```

- To boot additional vPars from a running vPar (vpar1)

```
vpar1# vparboot -p <vparName>
```

- This can be executed from any running vPar

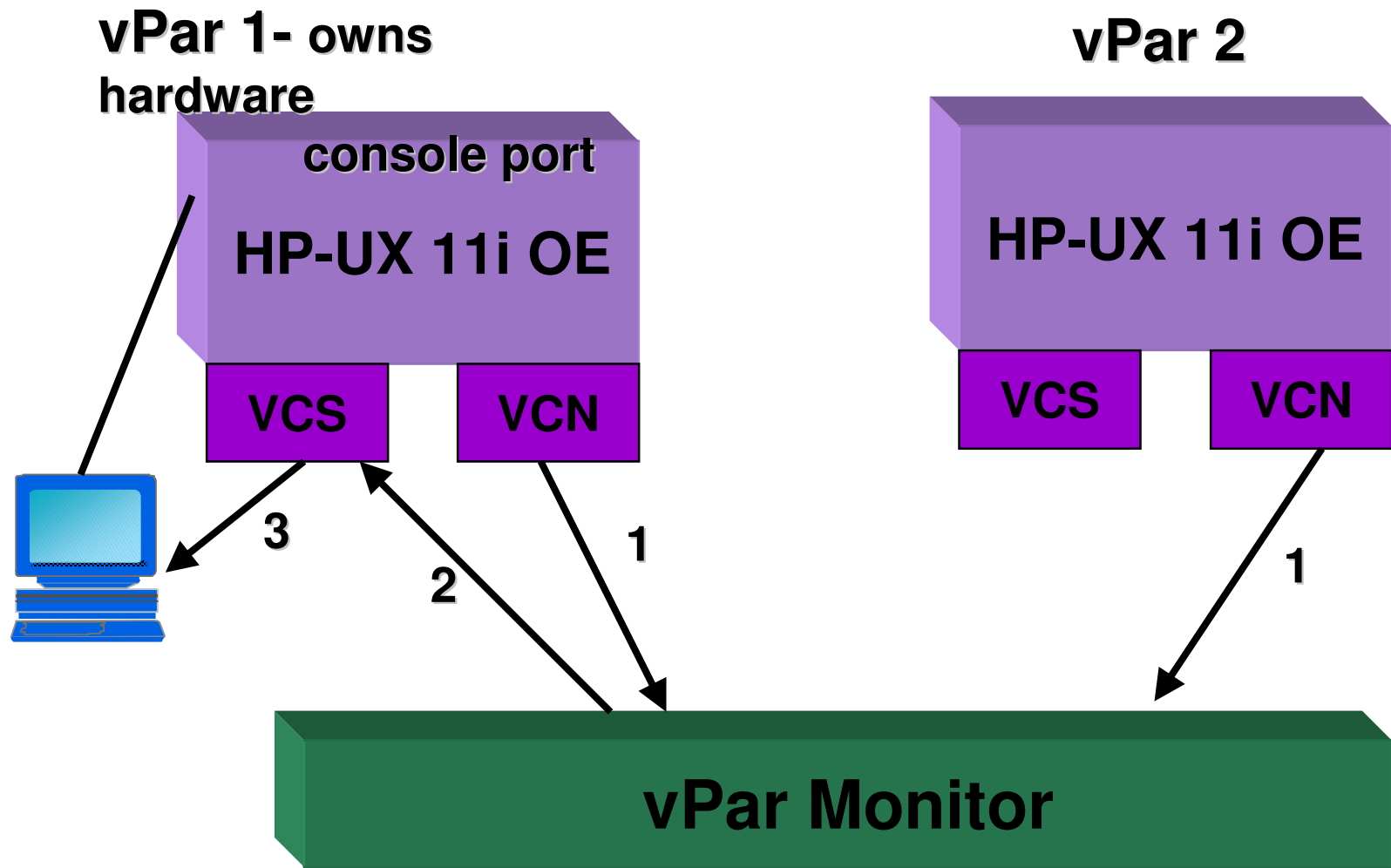
# Virtual Consoles

- **With limited number of PCI slots, you may not want to allocate one serial port for use as a console port for each partition you create.**
  - **With vPars, each virtual partition has its own virtual console.**
  - **First vPar created must own the Local Bus Adapter (LBA) that contains the physical hardware console port.**

# Virtual Consoles

- **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



# CPU Resources

## Two types of CPUs:

- **Bound** : CPUs that are assigned to a partition when partition is active
  - Interrupts are only handled by the bound CPUs
- **Unbound**: CPUs that can be dynamically reassigned among partitions
  - *Provide greater flexibility of movement between virtual partitions*
    - Can be added and removed without needing to bring down the affected partitions.

**Lowest numbered bound CPU is designated as the monarch CPU**

# When to use Bound or Unbound CPUs

## Use bound CPU:

- **When the application in the virtual partition are I/O intensive**
- **Only bound CPUs can process I/O interrupts**
  - With I/O intensive applications, there should be more bound CPUs than unbound CPU's

## Use unbound CPU:

- **When the applications are CPU intensive**
  - Can easily adjust number of CPUs via dynamic CPU migration

# CPU Migration

- **CPU Migration is adding and deleting CPUs from a virtual partition.**
- **Dynamic CPU Migration is migrating CPUs while the target virtual partition is running.**
  - **Both bound and unbound CPUs can be migrated**
    - **Only unbound CPUs can be dynamically migrated**
    - **HP-UX cannot dynamically migrate I/O interrupts. (bound CPUs)**
- **The upper limit on the number of CPUs that can be added when partition is running is the smaller of either:**
  - **maximum count set for partition minus the bound count**
  - **available number of floaters**
- **User has to balance the ratio of bound and unbound CPUs in a partition based on required I/O throughput**



# Dual-core Issues: Identifying Cell Boards

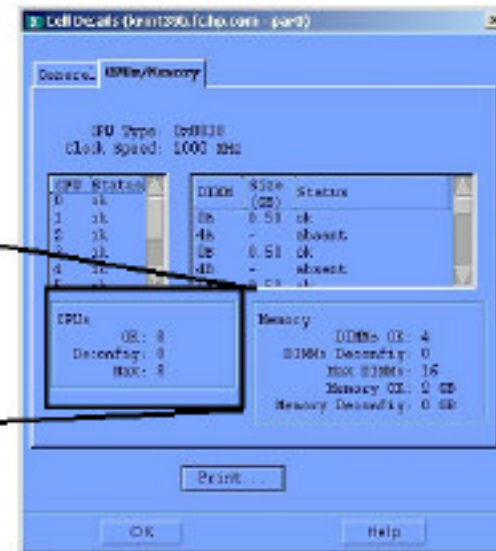
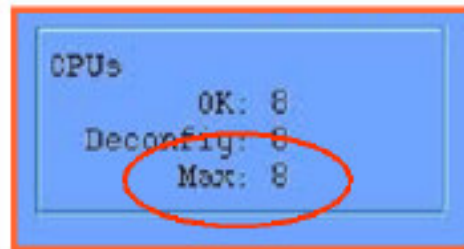


## Check maximum number of CPUs in the cell

```
# parstatus -c 0  
[Cell]
```

Hardware Location	Actual Usage	CPU Deconf/Max	Memory (GB) OK/Deconf	Connected To	Use Core Cell Capable	Use On Next Boot	Par Num
cab0,cell0	active core	8/0/8	2.0/ 0.0	cab0,bay0,chassis0	yes	yes	0

Or use parmgr(1M):



If you're not comfortable running diagnostics, you can use either the GUI parmgr(1M) or the command-line interface parstatus(1M). The -C option to parstatus(1M) includes the maximum number of processors that are supported on each cell. If that number is 8, you've got dual-core processors. Here's an example:

```
# parstatus -C
```

```
[Cell]
```

Hardware Location	Actual Usage	Core	CPU OK/Deconf/Max	Memory (GB) OK/Deconf	Connected To	Core Cell Capable	Use On Next Boot	Par Num
cab0,cell10	Active	Core	4/0/8	32.0/0.0	cab0,bay0,chassis0	yes	yes	0
cab0,cell11	Active	Core	4/0/8	32.0/0.0	cab0,bay0,chassis1	yes	yes	1
cab0,cell12	Active	Base	4/0/8	32.0/0.0	-	no	yes	0
cab0,cell13	Active	Base	4/0/8	32.0/0.0	-	no	yes	1

All of the cell boards in this system have dual-core processors.

You can get the same information from the "CPUs/Memory" tab in parmgr(1M).

## How to recognize dual-core “siblings”

- Use Management Processor’s DF command
- Use monitor “scan” command:

```
This system has Sockets with Multiple CPU cores
```

```
The following CPUs are siblings:
```

```
CPU 0/10, owned by vPar "vparmh2"
```

```
CPU 0/11, owned by vPar "vparmh3"
```

- Use ioscan(1M)
  - Use parmgr(1M) or parstatus(1M) to confirm dual-core status
  - Identify siblings by adjacent hardware paths

0/10	processor	Processor
0/11	processor	Processor
0/12	processor	Processor
0/13	processor	Processor
0/14	processor	Processor
0/15	processor	Processor
0/16	processor	Processor
0/17	processor	Processor

## List of Supported Systems



Machine	Minimum Firmware	First support
rp5405/rp5470 (L3000), rp7400 (N4000)	41.02	A.01
Superdome	PDC 35.3, SMS 1.2	A.02.01
Superdome (PA-8800)	MFW 14.22, System 20.7, Scan 3.0.13**	A.03.01
rp7405/rp7410	6.0*	A.02.02
rp7420	1.01 (RM_4.7)	A.03.01
rp8400	6.0*	A.02.01
rp8420	1.01 (RM_4.7)	A.03.01

\*Only releases A.02.03 and later are compatible with version 6.0.

\*\*MFW is "Utility Firmware". System is "PA Firmware"

# Memory Resources

- **Three ways to specify memory:**
  - **Size in MB (recommended method)**
    - **Starting address and size should be multiple of 64MB**
  - **One or more ranges with each range containing starting byte address and size in MB (not recommended)**
  - **Combination of both**
    - **A partition requires at-least 64MB below 2GB for kernel**
- **Memory cannot be added or removed when partition is active**

# I/O Resources

- **System Bus Adapters (SBAs) and memory controllers are owned by monitor**
- **Local Bus Adapter (LBA) is the minimum granularity of assignment**
- **Combo SCSI/LAN card is not supported as a boot disk**
- **At the minimum a partition should own an LBA with boot disk and LAN port**
- **LBA cannot be removed or added to an active partition**
- **I/O path can have following attributes: console, boot, altboot, dump, and swap**
- **Full I/O path has to be specified when the path has attributes**
- **Monitor multiplexes console among partitions even if a partition owns the LBA that contains console**

# Security Concerns

## **vPars commands restricted to root access**

- **Works on any partitions, regardless of which partition the commands are executed from.**
- **A user on one partition can affect another virtual partition by targeting the virtual partition in a vPars command.**
  - **For example, a root user running on the partition vpar2 can reset the partition vpar3 using the vparreset command.**

## **With console access a user can gain access to the file systems on any of the virtual partitions in the same computer or hardware partition.**

- **This is done by performing a hardware reset of the computer (Ctrl-B), interrupting the boot sequence, and at the ISL prompt manually booting any partition into single-user mode.**

# Partition Plan

## **In the Partition Plan, the following has been identified:**

- **Number of virtual partitions to be created**
- **Virtual Partitions Names**
  - Each vPar requires a unique name
  - Maximum length of alphanumeric name is 239 characters
- **Hardware resource allocation**
- **Amount of memory to be allocated to each virtual partition**
- **Boot and LAN Paths has been determined**
- **Ensure hardware console port is owned by the first virtual partition**



# Partition Plan

<b>vPar Name</b>	<b>vpar1</b>	<b>vpar2</b>	<b>vpar3</b>
<b>CPUs</b>	num = 3, min = 2	num = 2, min = 2	num = 2, min = 2
<b>Memory</b>	1GB	4GB	2GB
<b>I/O Paths (LBAs)</b>	0/0/* 0/4/*	0/8/* 1/10/*	0/5/* 1/4/*
<b>Boot Path</b>	0/0/2/0.6.0	0/8/0/0.5.0	1/4/0/0.5.0
<b>Console</b>	Virtual	Virtual	Virtual
<b>Kernel Image</b>	/stand/vmunix	/stand/vmunix	/stand/vmunix
<b>Autoboot</b>	On	On	On

# Create a Partition Worksheet

<b>Resource or Attribute</b>	<b>vparcreate Option</b>
Partition name is vpar4	-p vpar4
Four total CPUs	-a cpu::4
Three bound CPUs and maximum of five CPU's.	-a cpu:::3:5
At Hardware Paths 101, 105, 107	-a cpu:101 -a cpu:105 -a cpu: 107
2048 MB Memory	-a mem::2048
All Hardware where path begins with 0/5	-a io:0/5
All Hardware where path begins with 1/4	-a io:1/4
Hardware at 1/4/0/0.5.0 as the boot disk	-a io:1/4/0/0.5.0:boot

# vPars Commands

## *vPar commands:*

- # **vparboot(1 m)**  
*Boot a virtual partition*
- # **vparcreate(1 m)**  
*Create a vPar, with or without resources*
- # **vparmodify(1 m)**  
*Modify the attributes of a virtual partition*
- # **vparremove(1 m)**  
*Remove a virtual partition*
- # **vparreset(1 m)**  
*Reset a virtual partition*
- # **vparstatus(1 m)**  
*Display information about one or more vPars*

## *vPar resources*

- # **vparresources(5)**
- # **vpartition(5)**



# Example: vparcreate

- **To create a virtual partition called vpar1 with no resources assigned:**

```
# vparcreate -p vpar1
```

- **To create a virtual partition called vpar2, setting the autoboot attribute and specifying five processors, two of which are bound, 2 Gb of physical memory, and a boot disk:**

```
# vparcreate -p vpar2 -a cpu::5 -a cpu:::2:6 \ -a  
mem::2048 -a io:1/0 -a io:1/0/2/0.6.0:BOOT\ -B  
AUTO
```

- **To set the primary boot path and the secondary boot path**

```
# vparcreate -p vpar1 -a io:0/8/0/0.5.0:BOOT \  
-a io:0/8 -a io:0/8/0/0.2.0:ALTBOOT
```

# Example: vparboot

- **To boot the virtual partition vpar1:**

```
# vparboot -p vpar1
```

- **To boot the virtual partition vpar2 with an alternate kernel:**

```
# vparboot -p vpar2 -b  
/stand/vmunix_debug
```

- **To cold-install the virtual partition vpar3 from a file at the default path /opt/ignite/boot on the current virtual partition**

```
# vparboot -p vpar3 -I WINSTALL
```

# vparstatus Output

```
# vparstatus
```

```
[Virtual Partition]
```

Virtual Partition Name	State	Attributes	Kernel Path	Boot Opts
vpar1	Up	Stat,Auto	/stand/vmunix	
vpar2	Up	Dyn,Manl	/stand/vmunix	

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

Virtual Partition Name	CPU		Num IO	Memory (MB)	
	Min/Max	Bound/ Unbound		# Ranges/ Total MB	Total MB
vpar1	2/ 4	2 1	3	2/640	704
vpar2	2/ 4	2 2	2	2/704	768

## Installation Methods

### **There are several ways to install vPars:**

- As a cold install, along with an HP-UX 11i Server OE**
- As an application on an HP-UX 11i Server OE**
- From a running vPar, using the vparboot install process**
- Using make\_net\_recovery (Ignite-UX)**

# Initial vPars Install Process

- The process to install vPars on your HP-UX server is:
  - Choose a disk path to be the 'primary' boot disk for vPars
  - Install vPars software (T1335AC) on the vPars boot disk [use `swinstall(1m)`]
  - Verify vPars software installation
    - Review `swinstall` log for errors or warnings



# Initial vPar Creation

## To create the initial vPar

– Use the vparcreate(1M) command:

```
# vparcreate -p vpar1 -a cpu::2 -a cpu::2 -a  
mem::640 \  
-a io:0/0 -a io:0/4 -a io:0/0/2/0.6.0:BOOT
```

– Do not assign the Monarch processor, if possible

- The Monarch is used by the monitor (/stand/vpmon)
- The Monarch is usually the lowest numbered processor in the box or complex
  - Use ioscan to verify

# Additional vPar Creation

## Creating additional vPars:

### ◆ Examples:

```
# vparcreate -p vpar2 -a cpu::1 -a cpu:::1:2 -  
a mem::1024 \  
-a io:1/2 -a io:1/4 -a io:1/2/0/0.6.0:BOOT \  
-a io:1/2/0/0.5.0:ALTBOOT
```

```
# vparcreate -p vpar3 -a cpu::1 -a cpu:::1:3 -  
a mem::8192 \  
-a io:X/Y -a io:X/Z -a io:X/Y/0/0.A.0:BOOT \  
-a io:X/Y/0/0.B.0:ALTBOOT
```

◆ It is not necessary to create additional vPars right now, but doing so may be more efficient.

# vPar Installation Verification

- To verify and test vPar Installation:
  - Shutdown HP-UX

```
# shutdown -ry 0
```

- Place the server in Boot Console Handler (BCH) mode
- Boot from primary disk, interact with ISL

```
Main Menu: Enter command or menu > bo pri  
Interact with IPL (Y, N, or Cancel)?> y  
ISL>
```

# vPar Installation Verification

- To verify and test vPar Installation:
  - Boot the vPars Monitor “/stand/vpmon”

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

- From the Monitor, boot the first vPar

```
MON> vparload -p vpar1
```

- Access the vPar console
- Login, and verify status

```
# vparstatus -v -p vpar1
```

# vPars licensing & delivery

## **Base functionality**

- Capability to create a maximum of 2 vpars per hard partition (or non-partitioned machine), where 1 vPar has only 1 CPU *{good for trial, upgrades, small machines}*
- Available at \$0 cost with HP-UX 11i (upon support for particular server)
- Available through HP Software Depot: [www.software.hp.com](http://www.software.hp.com)

## **Full functionality (purchase T1335AC per CPU)**

- Full functionality is licensed per installed CPU, in hard partitions (or non-partitioned machines) running vPars
- Available through separate vPars CD, upon order of product

## **Licensing impact on other HP OE or ISU software**

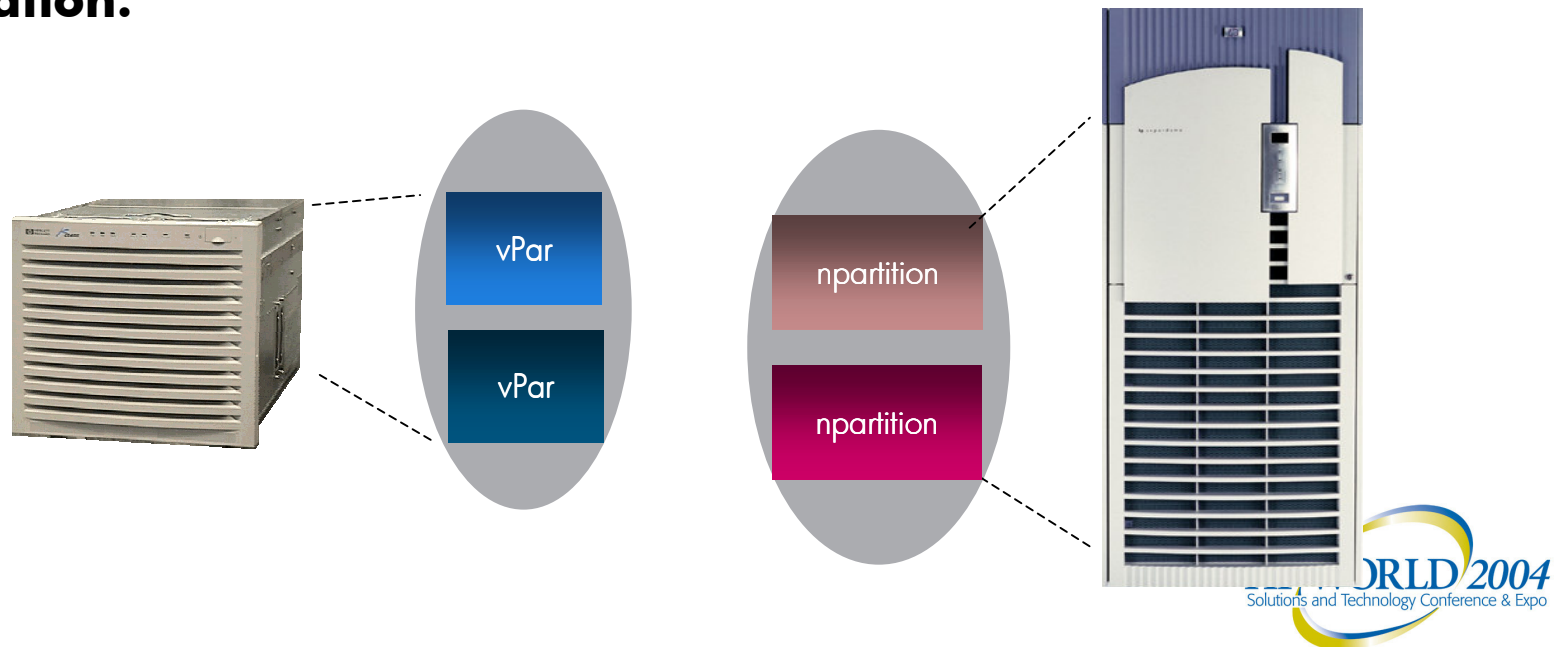
- None; HP OE or ISU software licensing per box, hard partition or tier is not impacted if that server becomes virtual-partitioned
- OE and application software continues to be purchased on a hard partition basis *(since vPars can be so easily reconfigured)*



# MC/ServiceGuard Cluster Configuration Considerations (a)

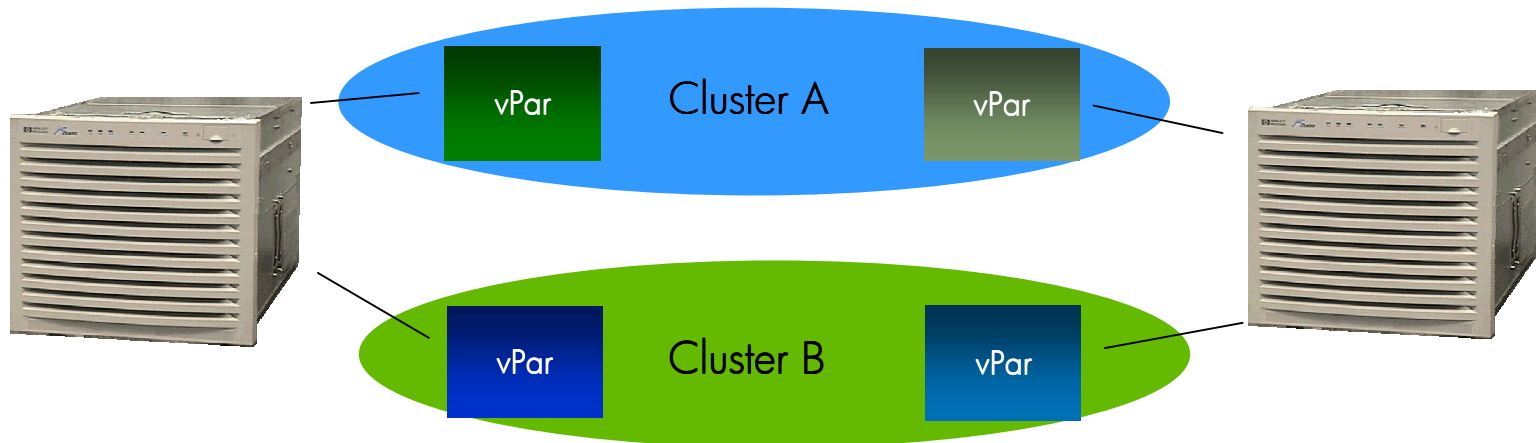
- ServiceGuard “cluster-in-a-box” is supported. Partitions of a cluster are wholly contained within the same hardware cabinet. While this configuration is subject to single points of failure, it may provide adequate availability characteristics for some applications.

**Note: “cluster-in-a-box” consisting exclusively of vPars is more likely to suffer a complete cluster failure than one made up exclusively of nPartitions. nPartitions provides hardware and software isolation whereas vPars provides software isolation.**



# MC/ServiceGuard Cluster Configuration Considerations (b)

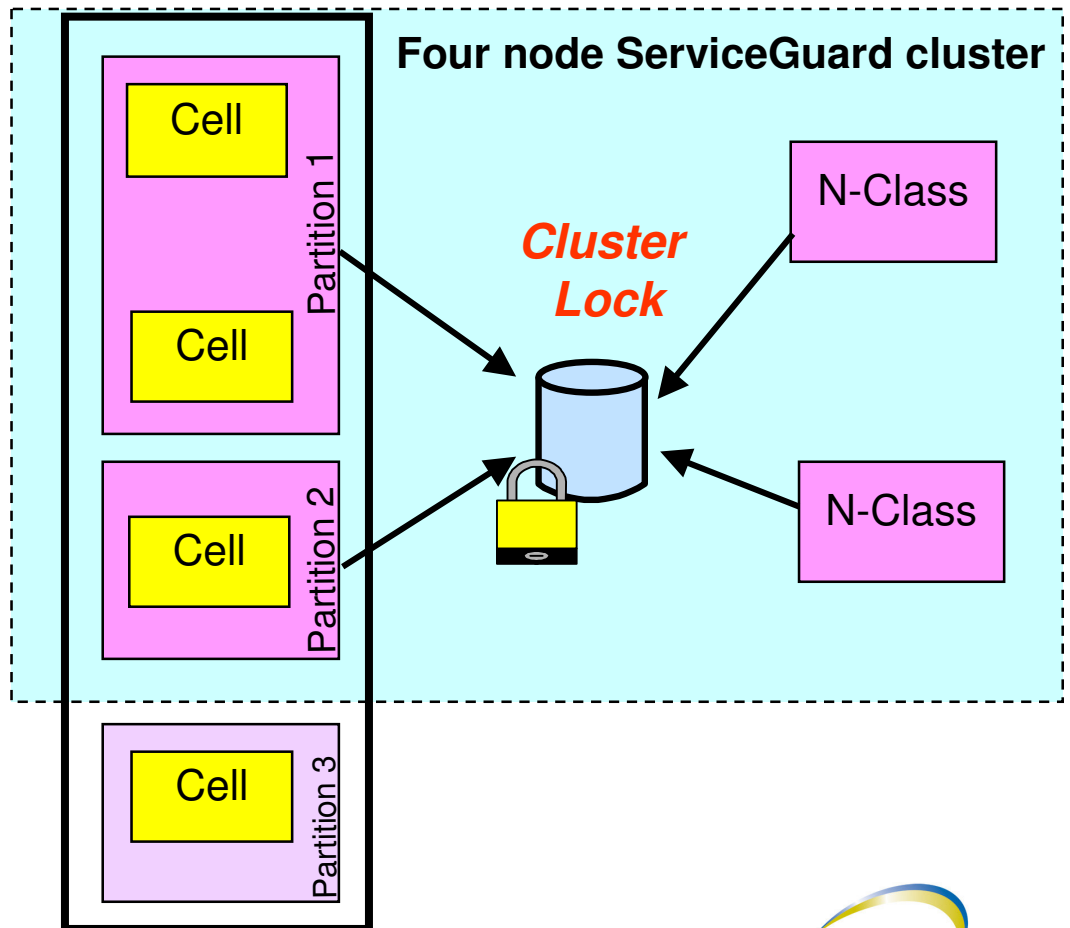
- **To configure a cluster with no SPOF, the membership must extend beyond a single hardware cabinet and comply to the standard ServiceGuard configuration rules. The cluster must be configured such that a failure of a single hardware cabinet does not result in the failure of a majority of the nodes in the cluster.**
- **Separate partitions within a physical server can be configured as part of different ServiceGuard clusters**



# Mixed Clusters with 11.0 and 11i Supported

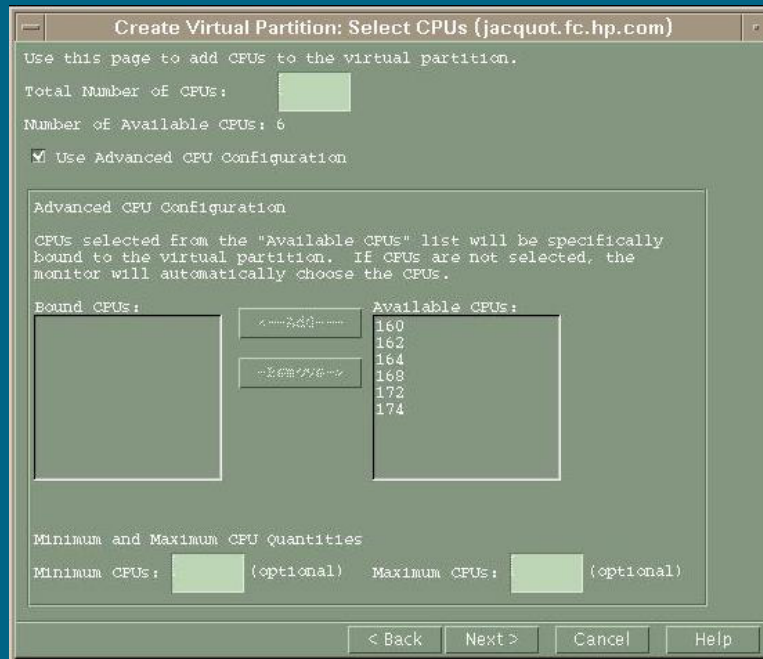
- **Mixed clusters where some nodes are running HP-UX 11.0 and some nodes running HP-UX 11i is supported – requires all nodes running the same SG revision (e.g. 11.09 or 11.13)**
- **EXAMPLE where two nodes in a SuperDome are running HP-UX 11i and two N-class nodes are running 11.0**

## SuperDome System





# virtual partition manager (vparmgr): GUI for managing virtual partitions



✓ *parmgr is vPar aware!*

*(it doesn't do vPars configuration at this point, but the 2 are planned to be integrated in the future)*

- Create, modify and delete virtual partitions (vpars)\*
- Display assigned resources, attributes, and status of vpar
- Display vpar event log and samlog
- Boot and reset a vpar
- Direct invocation of task screens
- Preview create/modify vpar command lines prior to execution

## Summary

# why use vPars?

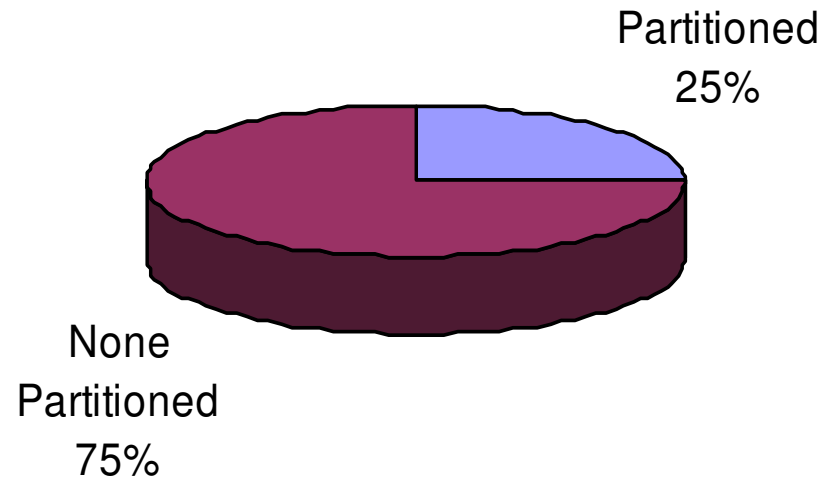


To reduce total cost of ownership (TCO) for a server environment, due to:

- ↻ increased overall system utilization*
- ↻ increased flexibility in resource configuration*
- ↻ improved system availability (due to OS and application isolation)*
- ↻ improved scalability*

## Production Environment

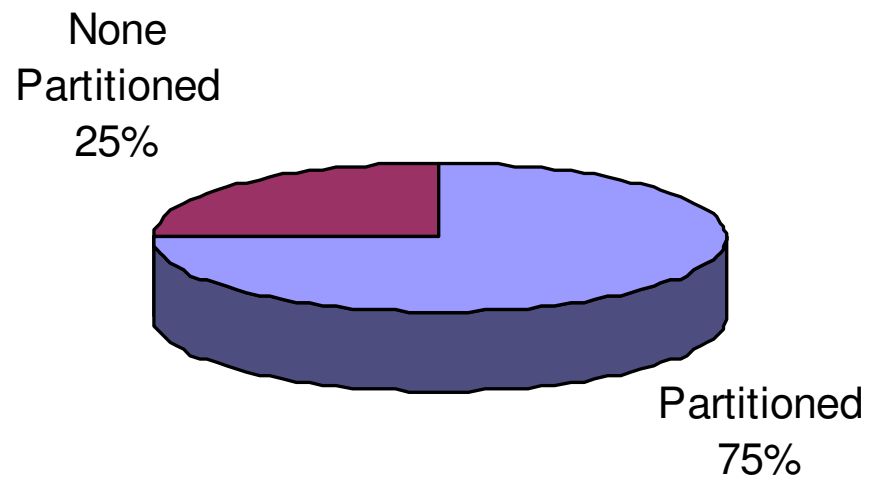
■ Partitioned ■ None Partitioned



Date: April 2003

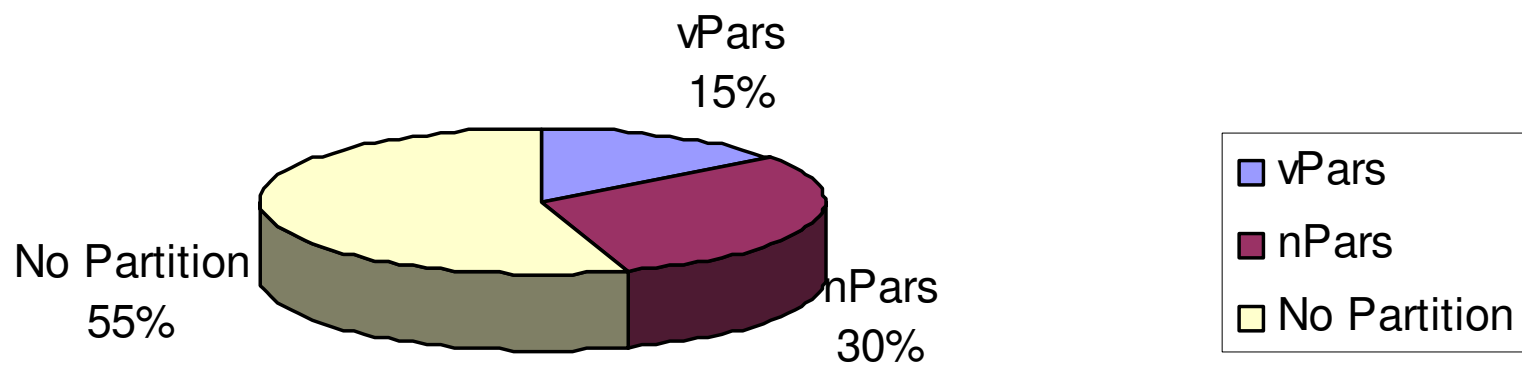
## Test and Development Environment

■ Partitioned ■ None Partitioned



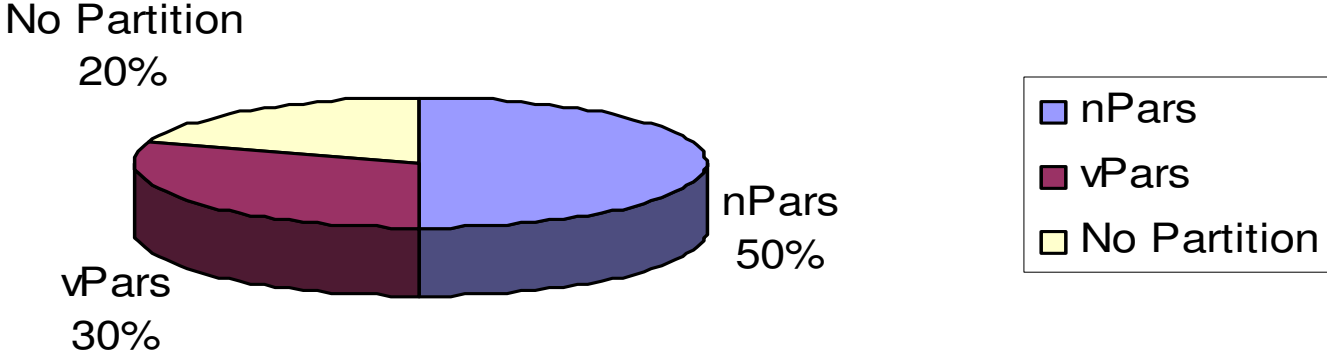
Date: April 2003

## KeyStone Partitioning



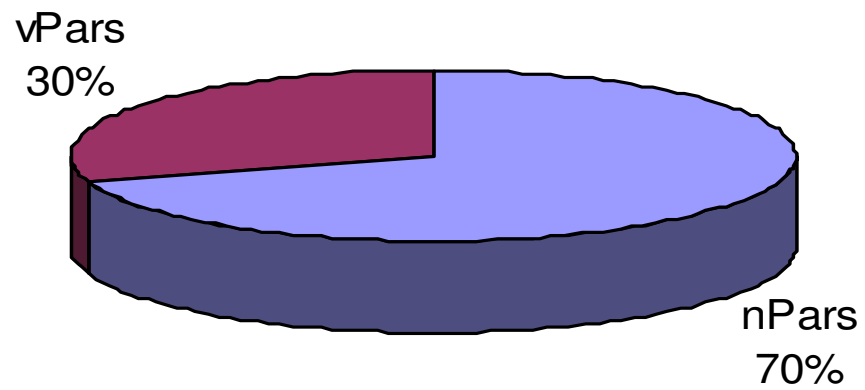
Date: April 2003

# SuperDome Partitioning



Date: April 2003

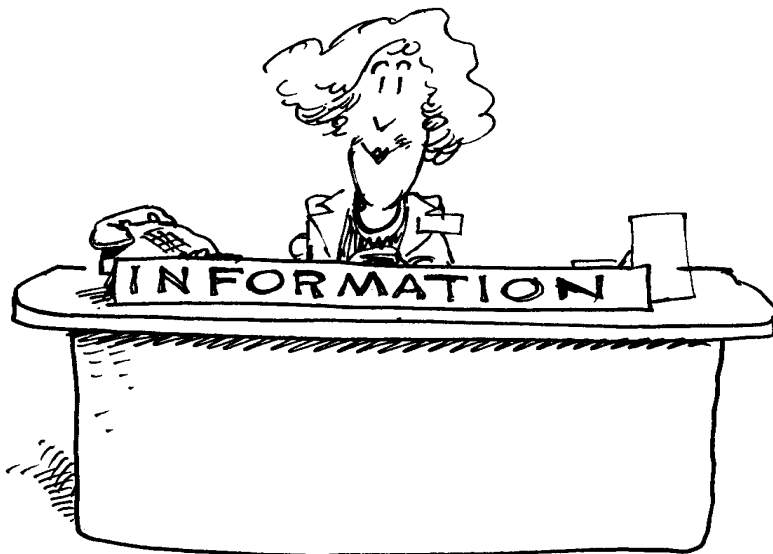
Production environment



Date: April 2003

## • External URLs

HP-UX Virtual  
Partitions



- [www.hp.com/go/partitions](http://www.hp.com/go/partitions)
- then "HP-UX Virtual Partitions"
- [www.hp.com/go/hp-ux](http://www.hp.com/go/hp-ux)
- [www.software.hp.com](http://www.software.hp.com)
- [www.docs.hp.com](http://www.docs.hp.com)







# HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:



RECOMMENDED TRAINING VENUE FOR THE  
**HP Certified Professional**

