Large Server and Resource Management Comparison: HP-UX and Tru64 UNIX

Session 3182

Sally Parker

UNIX Technology Consultant

Hewlett-Packard

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

Agenda

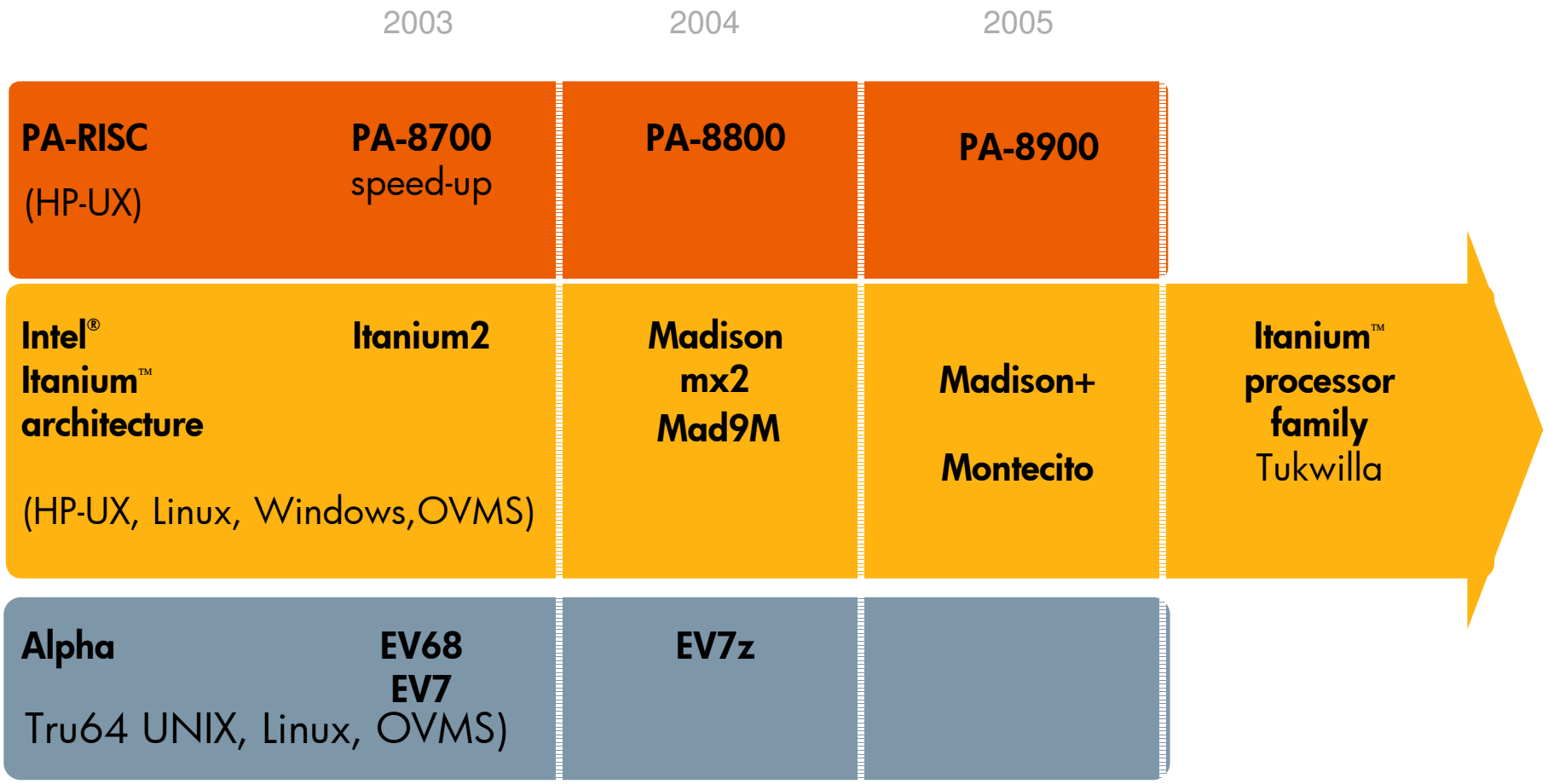
- Why this talk?
- Server overview
  - Alpha, PA-RISC, IA64
- Server management considerations
- Resource management capabilities
  - basic tools
  - additional tools to optimize server usage
- Resources and Futures

‘IT’ is evolving…

- Tru64, Alpha and PA-RISC in the future
- HP-UX the Enterprise UNIX of choice
- Itanium as the effective building block
- Much closer alignment of IT and the business
  - demise of geek heaven
  - IT must be more responsive, more flexible and less expensive
- Adaptive Enterprise

Server Overview..

Server Architecture: the processor, the most basic building block

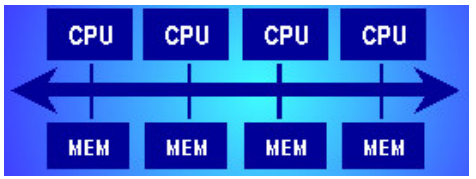


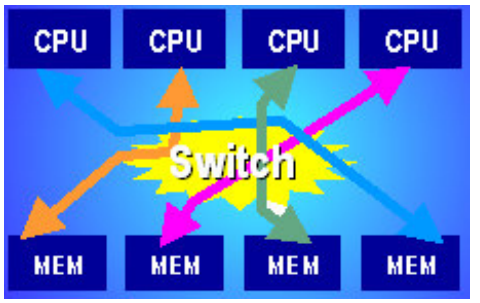
HP Server Portfolio

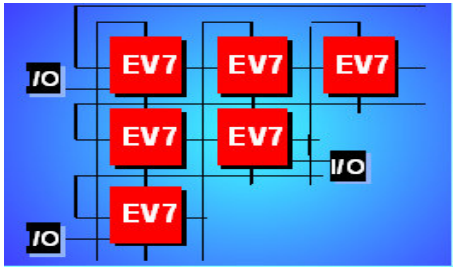
	Itanium	PA-RISC	Alpha
high-end	 ProLiant DL 700 series	Integrity Superdome	HP 9000 Superdome	AlphaServer SC series AlphaServer GS series	 NonStop S76000/ S86000
mid-range	 ProLiant DL/ML 500 series  ProLiant BL p-Class	Integrity rx7620 Integrity rx8620	rp8400 rp8420 rp7410 rp7420	AlphaServer ES series	 NonStop S76 series
entry level	 ProLiant DL/ML 300 series  ProLiant BL e-Class	 Integrity rx4640 Integrity cx2640   Integrity rx2600 Integrity rx5670 	 rp5430/70  rp2430/70	 AlphaServer DS series

Basic System Architectures

- Bus-based topology
  - Bus bandwidth shared by all processors and I/O
  - Latency varies significantly depending upon number of outstanding transactions
- Cross-bar topology
  - Simultaneous connection to all CPUs
  - Well-defined latency, limited request queuing
- Mesh architecture
  - Direct processor-to-processor and I/O interconnects
  - Extremely high bandwidth and very low latency







Tru64 UNIX servers

Physical and Logical Diagrams…

Tru64 UNIX Servers

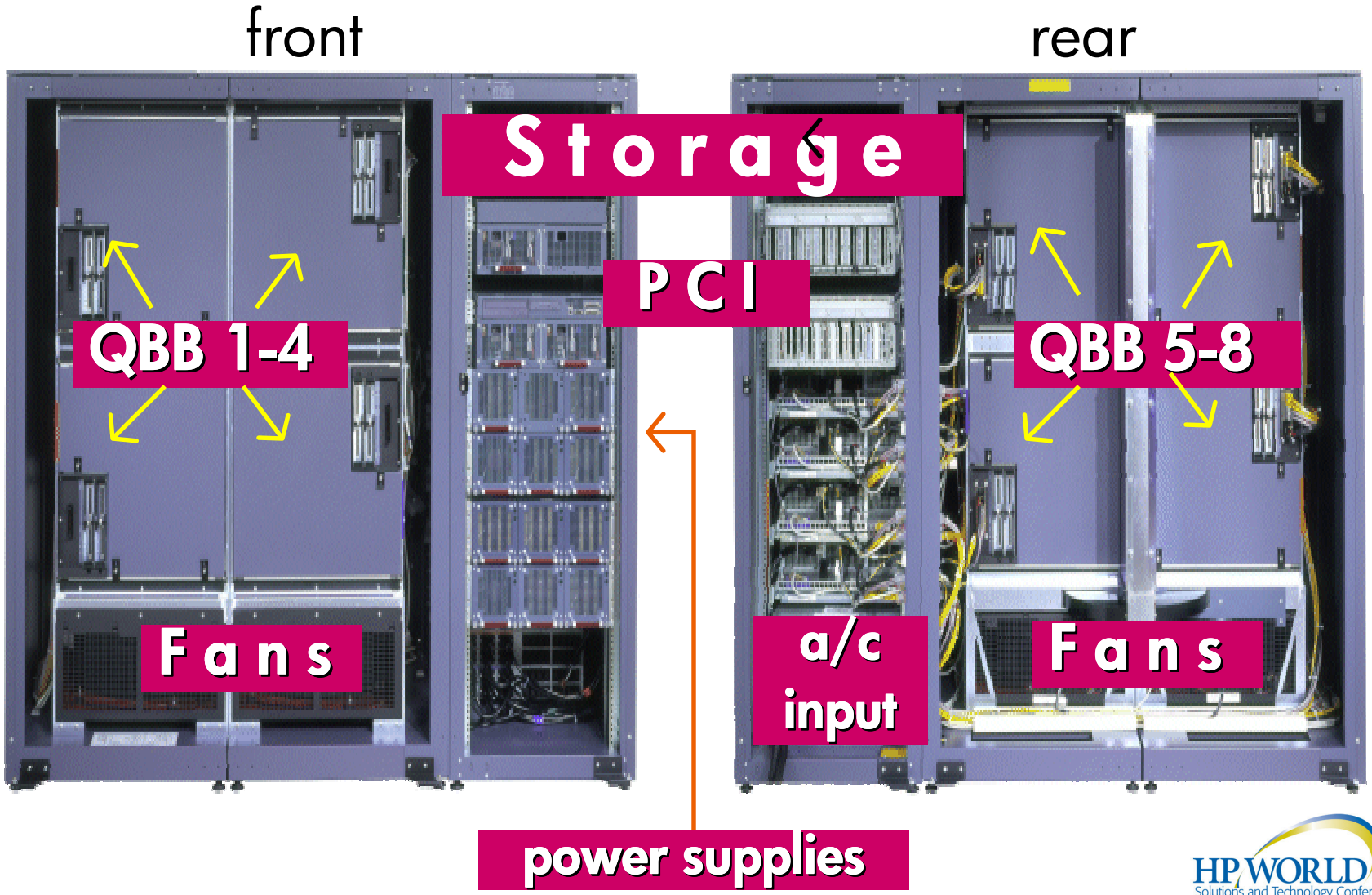
GS320 (GS80,GS160)

- aka *WILDFIRE*
- 32 x 1.22 GHz EV68s
- 57 GB/s System BW
- 256 GB memory
- SPECint2000 833
- SPECfp2000 1014
- Tru64 UNIX V4.0G/V5

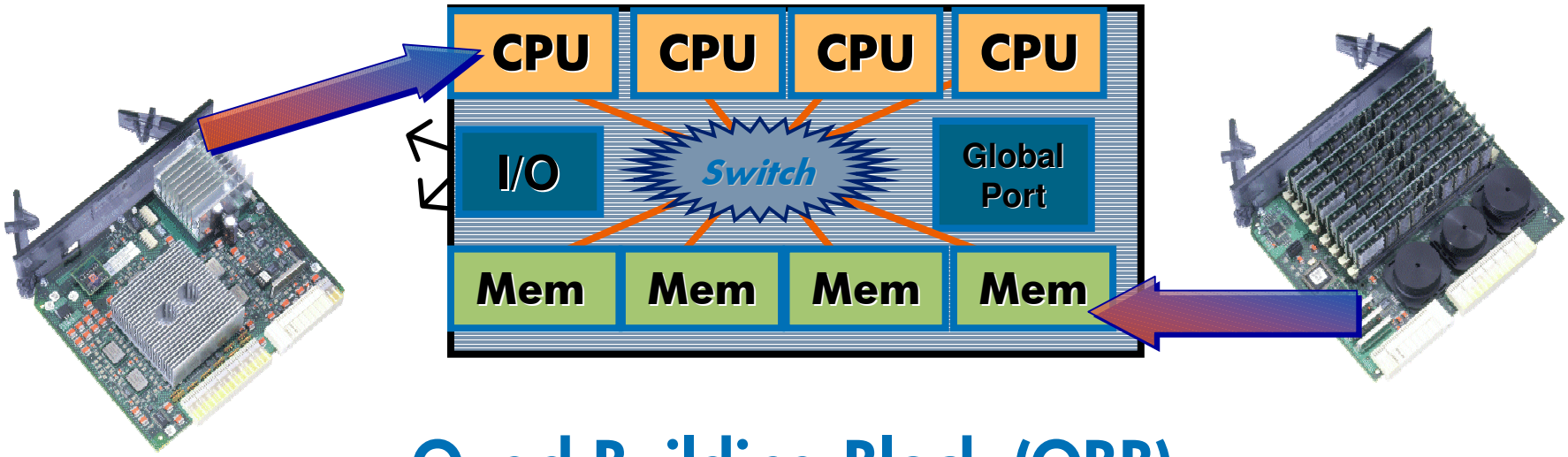
GS1280 (ES80,ES47)

- aka *MARVEL*
- 64 x 1.15GHz EV7 (1.3GHz EV7z)
- 800GB/s system BW
- 1/2 TB memory
- SPECint2000 877 (EV7z 994)
- SPECfp2000 1482 (EV7z 1684)
- Tru64 UNIX V5.1b

GS320 Physical Architecture



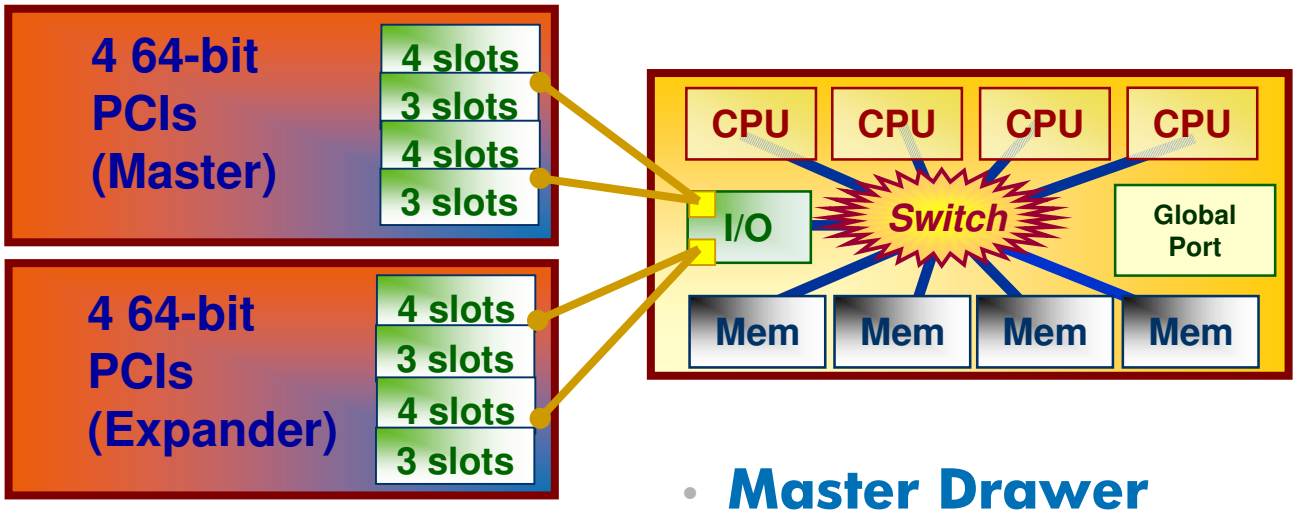
GS320 Basic Building Block



Quad Building Block (QBB)

- Up to 4 1224 MHz EV68 CPUs
  - 16 MBs of on-board Cache per CPU
- Up to 4 Memory Modules
  - Up to 8 GB per Memory Module
- GB/s Memory System Datarate
  - 1.6 GB/s per CPU/MEM link

GS320 I/O



2 IO 'risers per QBB

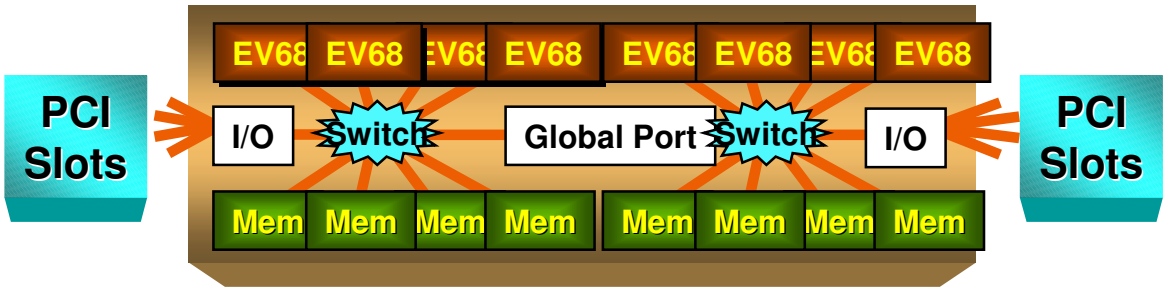
2 'hoses' per riser

2 PCI segments per hose

Up to 2 PCI drawers

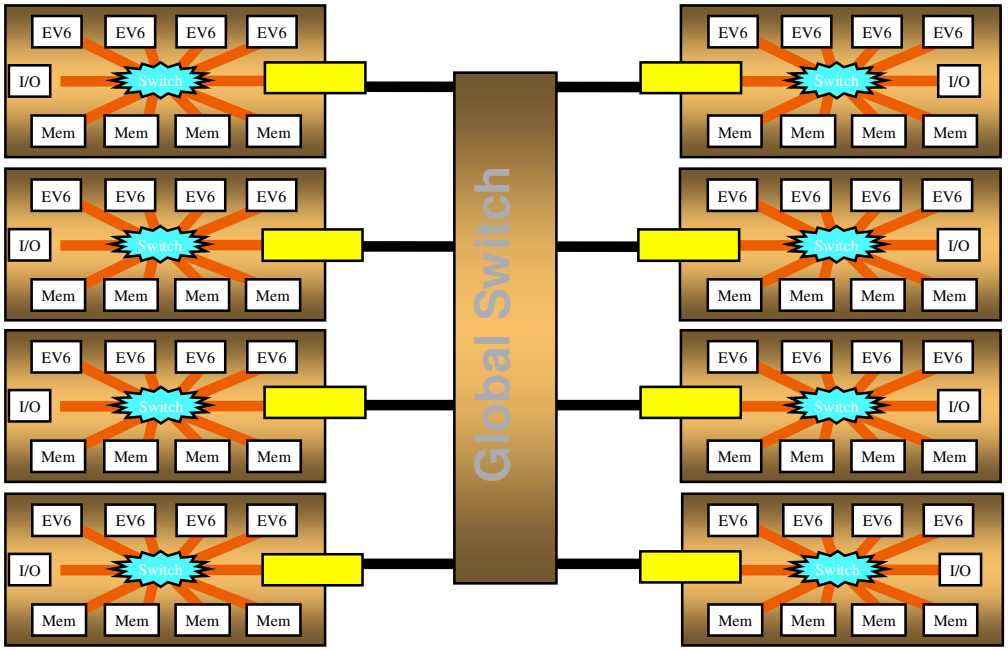
- **Master Drawer**
  - Standard I/O Module in 1 PCI Slot
  - Disk, CD, Floppy, Network, Console
  - 13 extra PCI Slots across 4 PCI Segments
- **Expander Drawer**
  - 14 PCI Slots
- **Global Port**
  - Connects QBBs to build larger systems

Building A GS320 Server

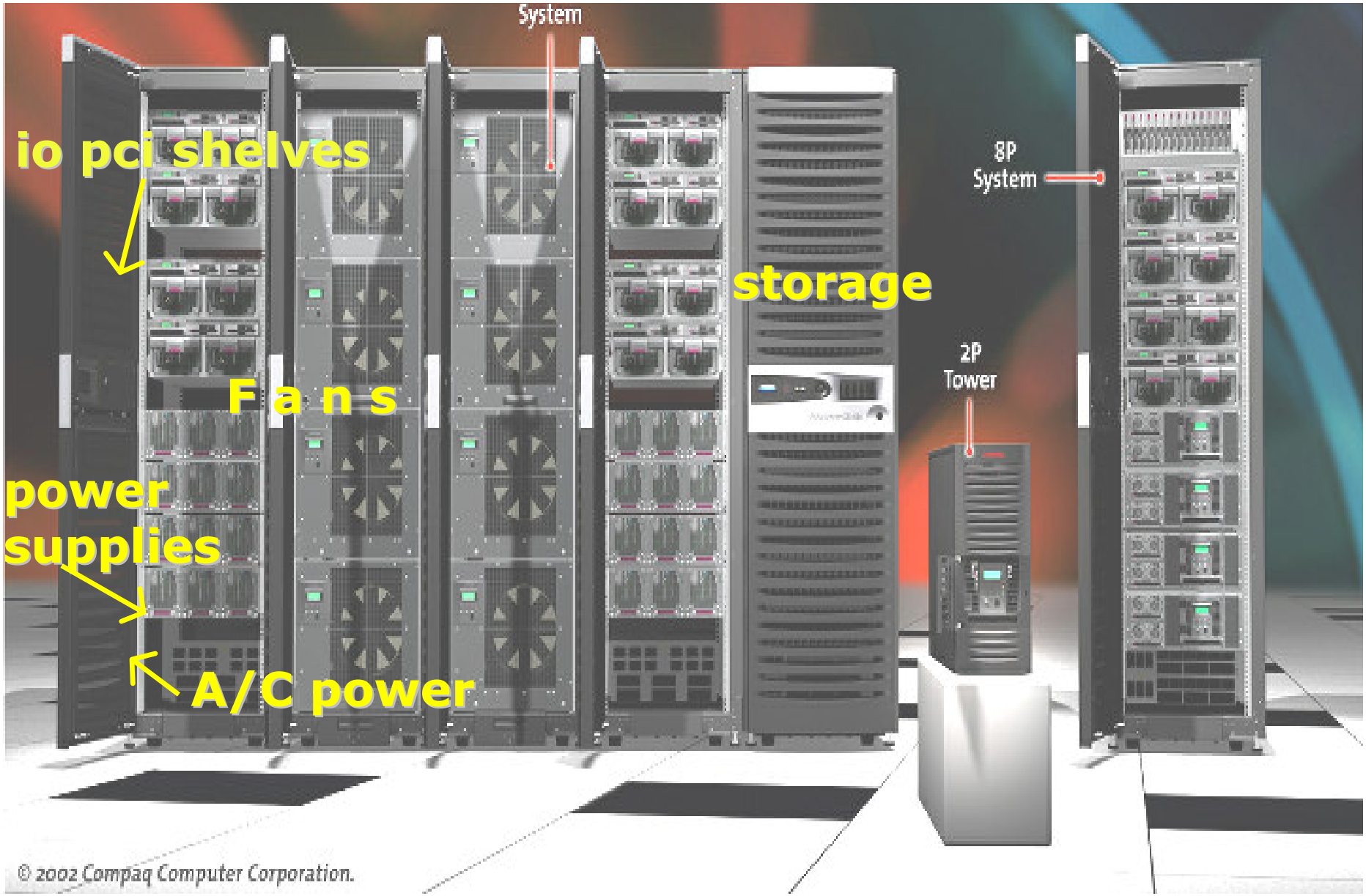


GS80

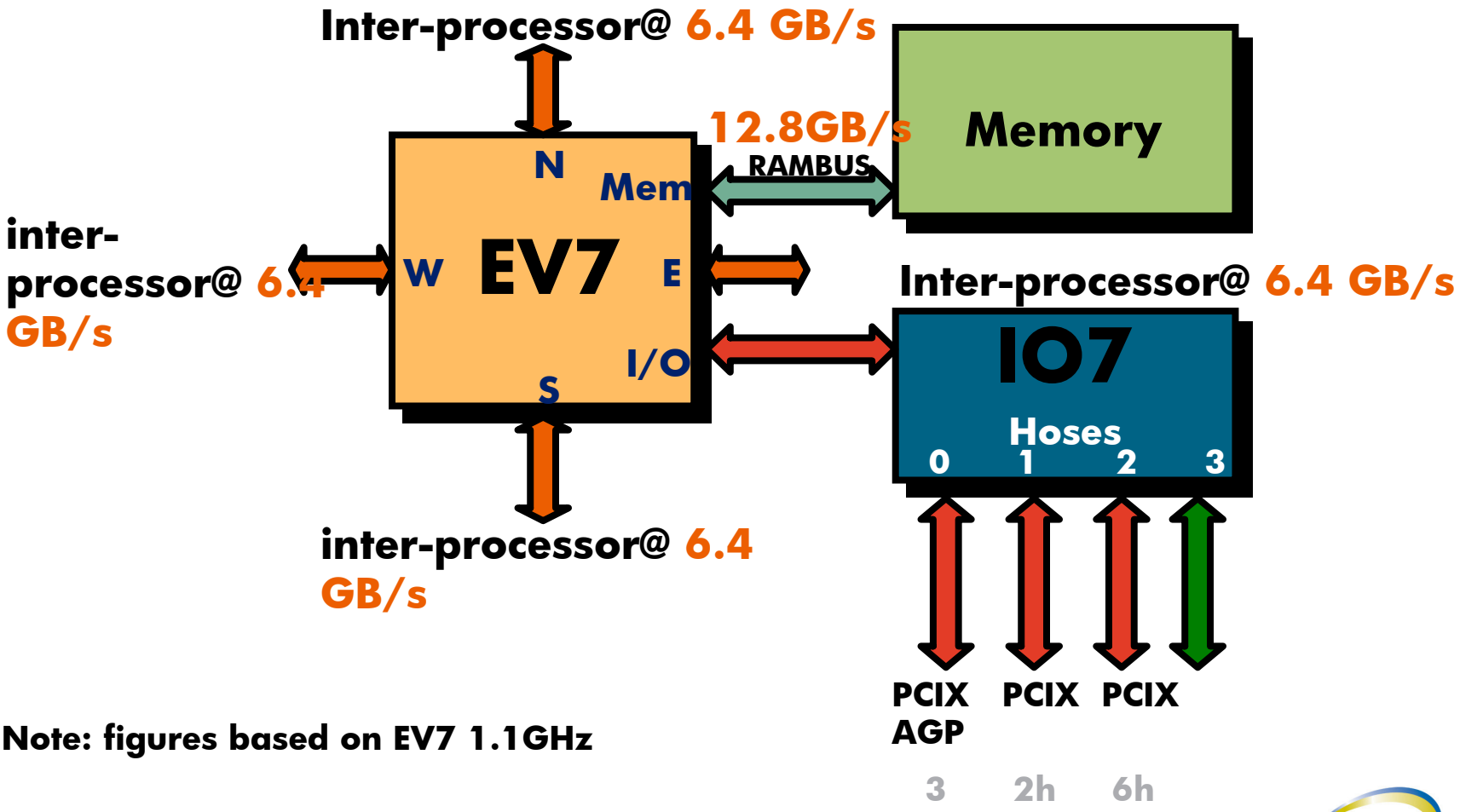
GS320



GS1280 Physical View

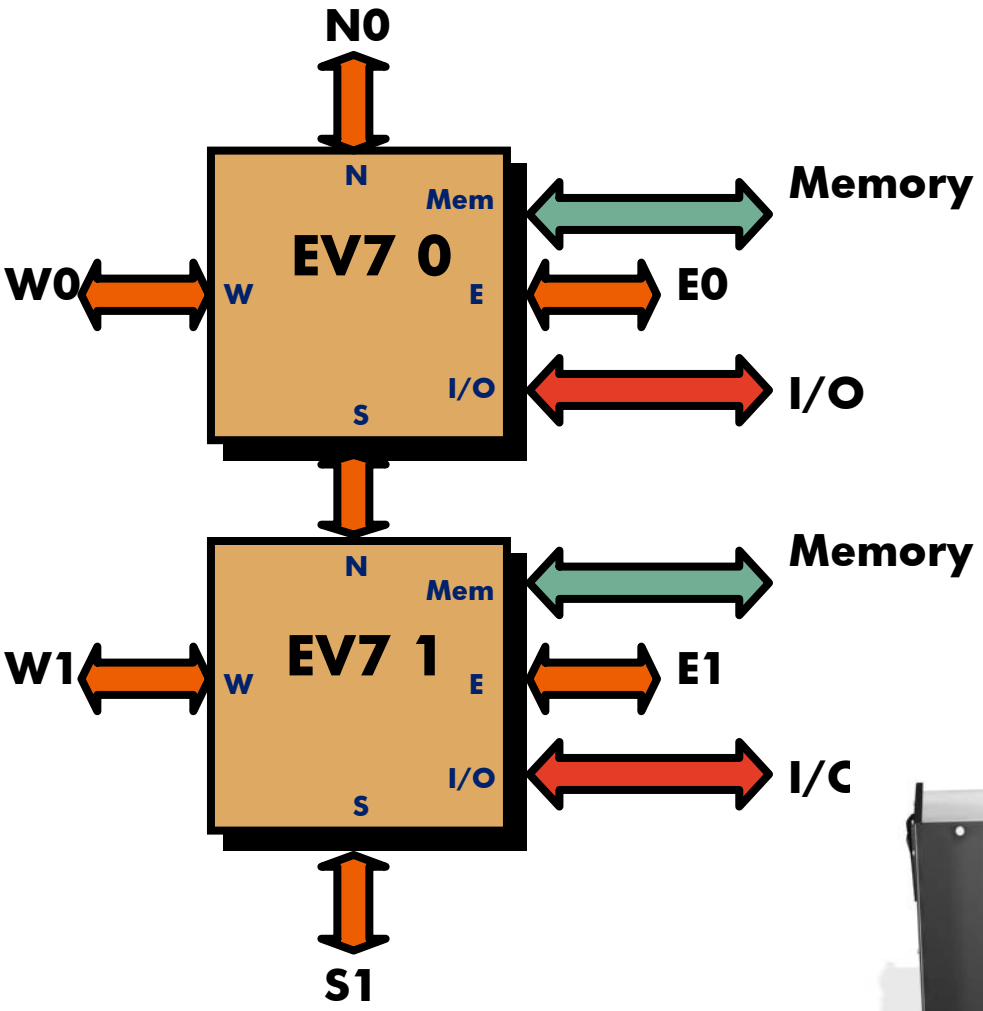


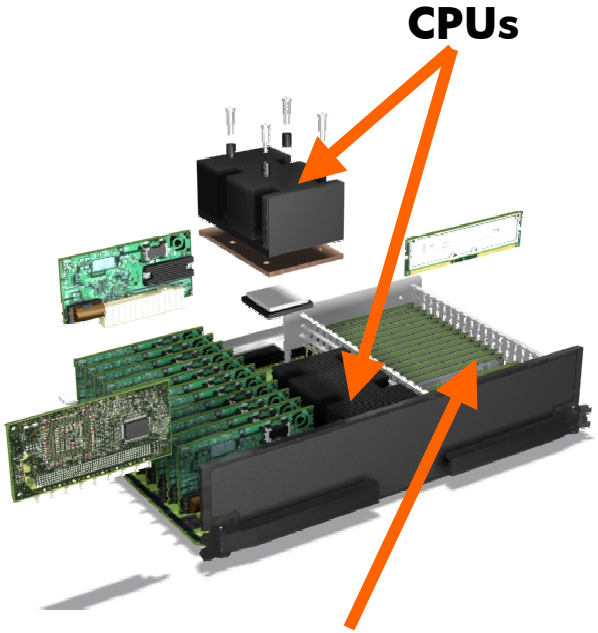
GS1280 Basic Building Block - EV7 Processor



Note: figures based on EV7 1.1GHz

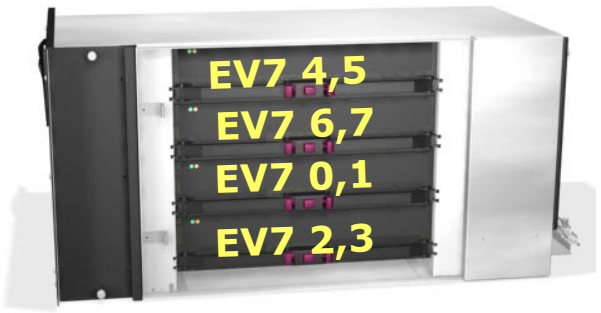
GS1280 Building A Glue-less SMP Server





RAMBus Memory Modules (RiMMs)

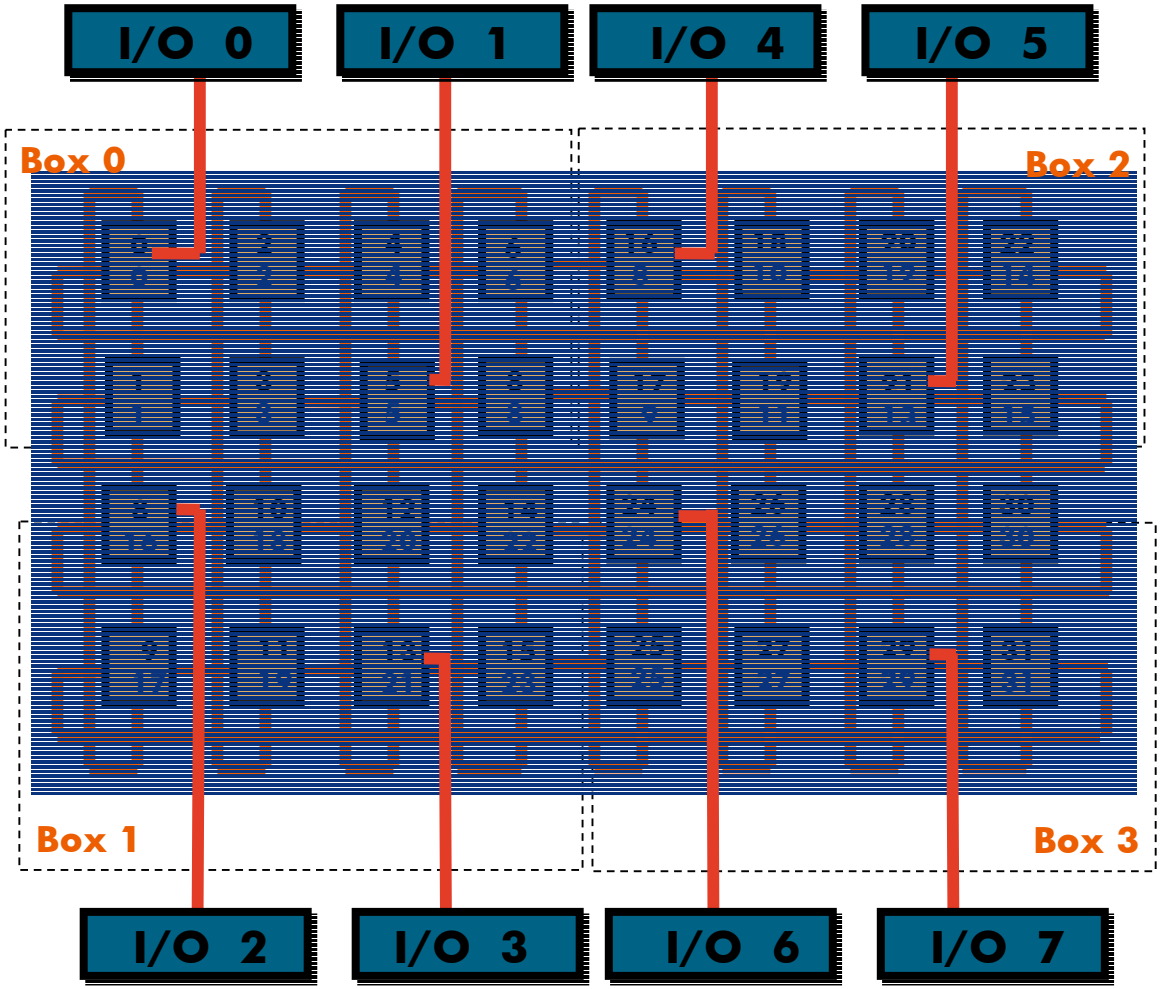
20 RiMMs (16 data, 4 redundancy)



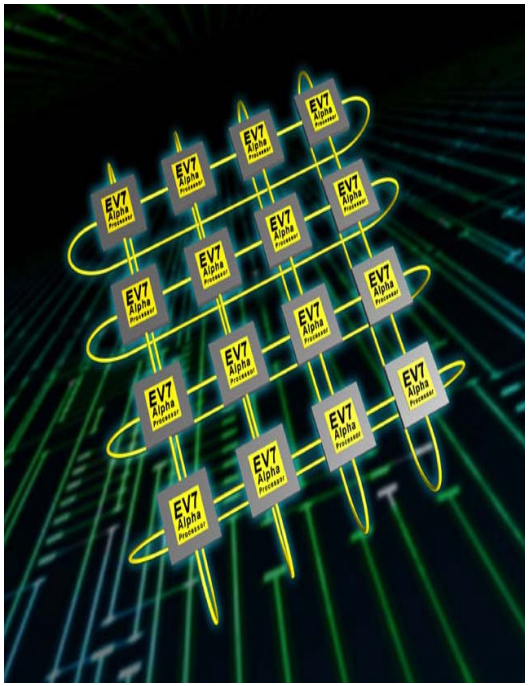
Packaging

GS1280 Torus and I/O

32p (standard shelf)



11 PCI slots per IO7



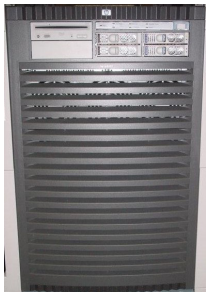
HP-UX Servers

Physical and Logical Diagrams

HP-UX PA-RISC 'HP9000' Cellular Servers

Superdome

- 16 cells
- 64 x PA8700
- 128 x PA8800
- 512 DIMM slots
- 192 PCI slots



rp84xx

- 4 cells
- 16 x PA8700
- 32 x PA8800
- 64 DIMM slots
- 32 PCI slots,

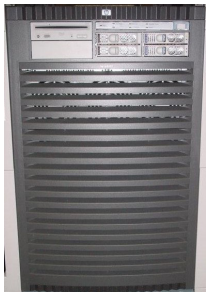
rp74xx

- 2 cells
- 8 x PA8700
- 16 x PA8800
- 32 DIMM slots
- 16 PCI slots,

HP-UX Itanium 'Integrity' Cellular Servers

Superdome

- 16 cells
- 64 Itanium cpus
- 512 DIMM slots
- 192 PCI slots
- HP-UX V2



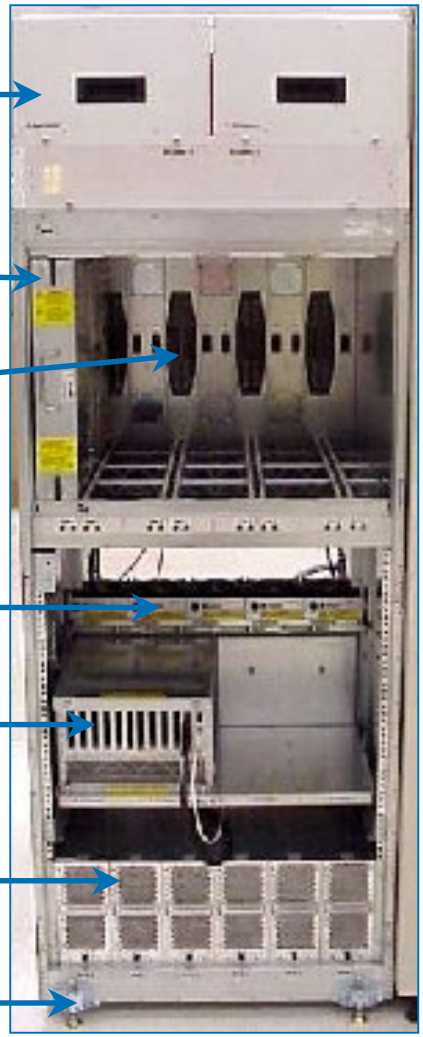
rx8620

- 4 cells
- 16 Itanium cpus
- 64 DIMM slots
- 32 PCI slots
- HP-UX V2

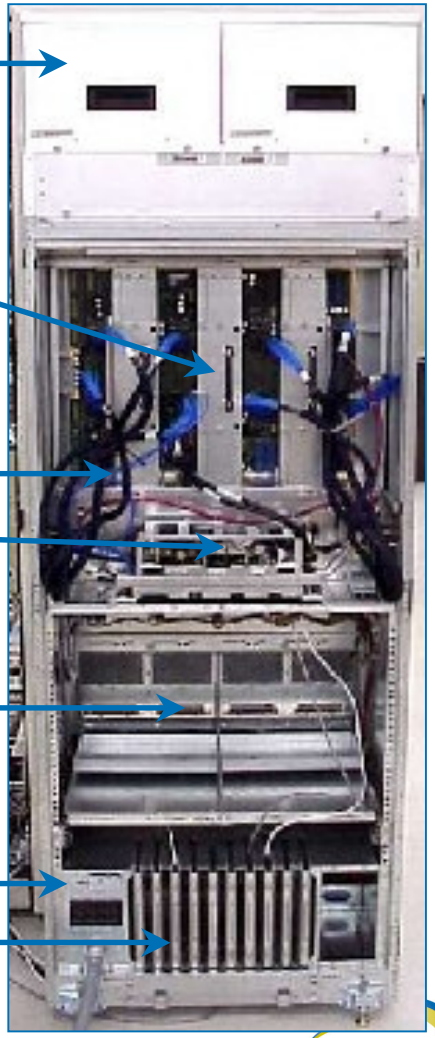
rx7620

- 2 cells
- 8 Itanium cpus
- 32 DIMM slots
- 16 PCI slots
- HP-UX V2

Superdome Physical View

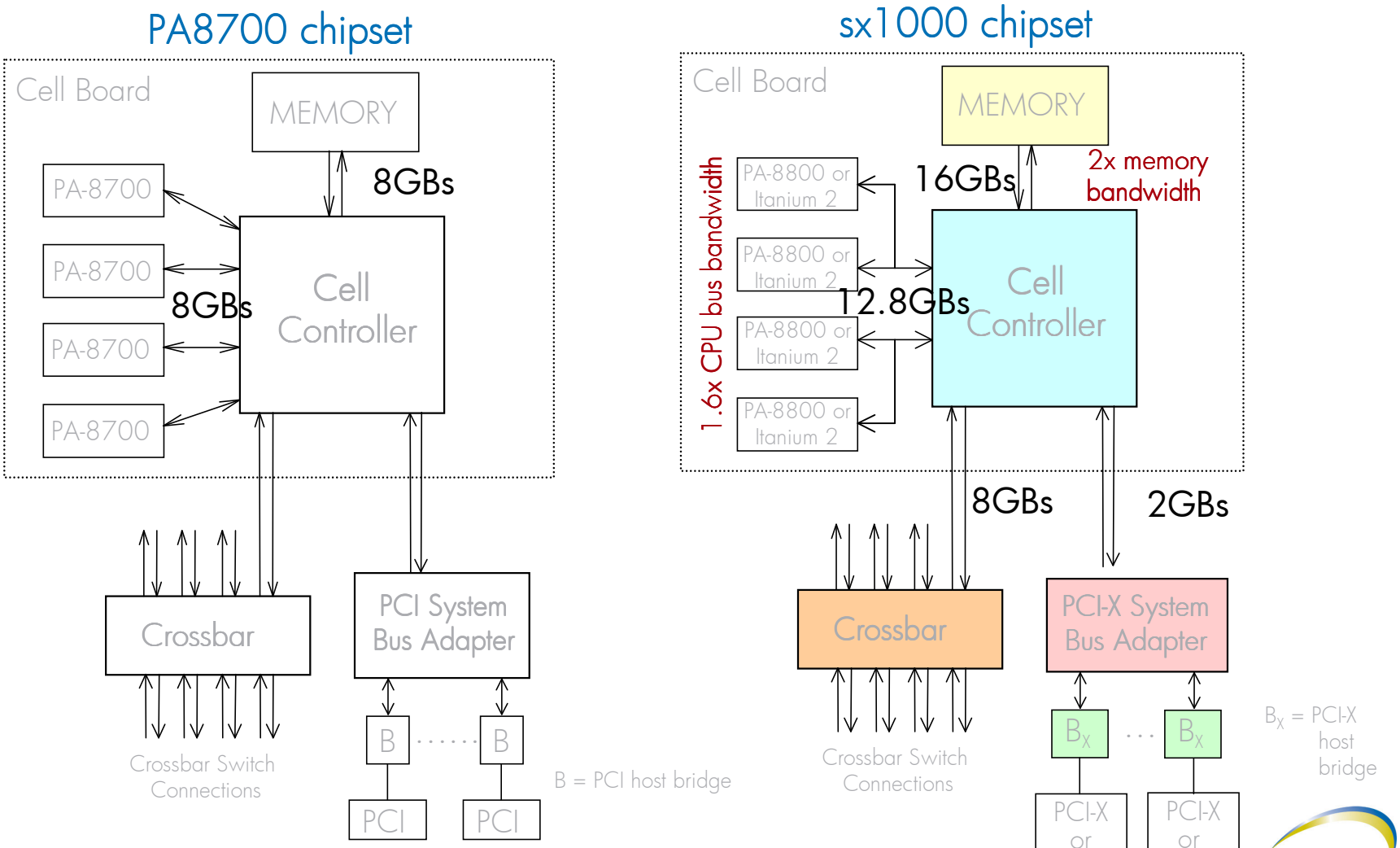


Front



Back

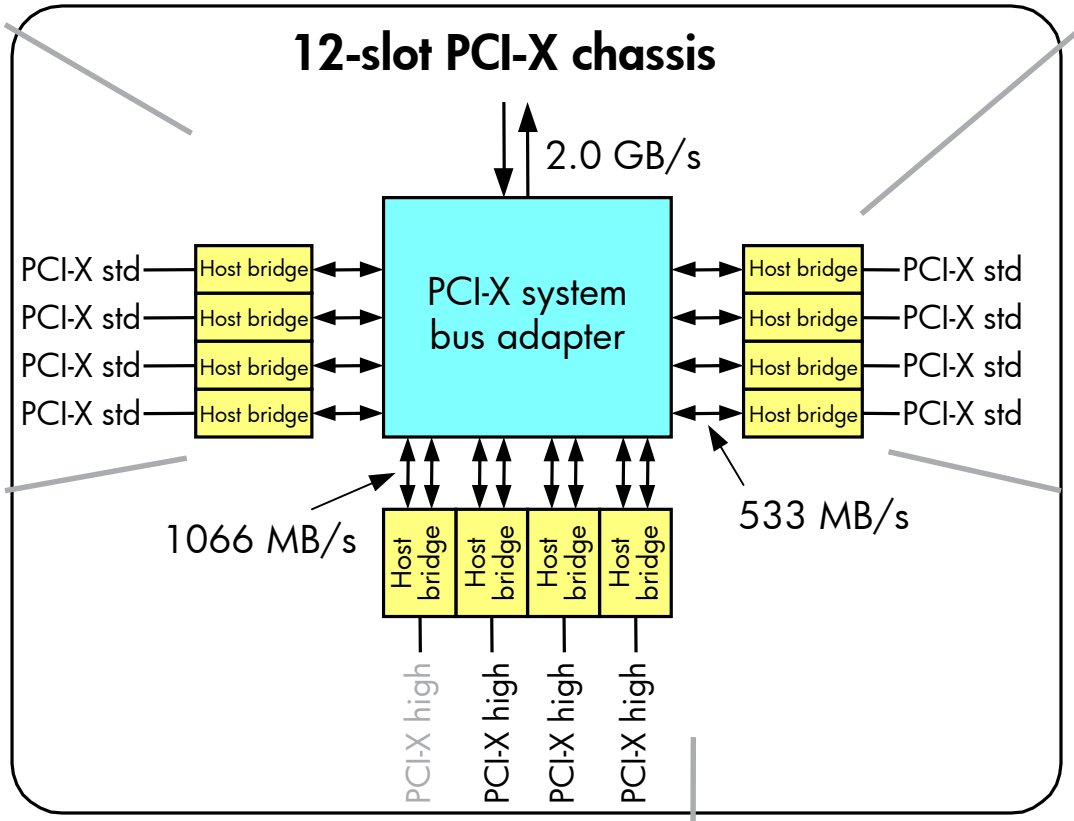
Superdome Building Block – The Cell



Superdome is a cell-based hierarchical cross-bar system

Superdome12-slot PCI-X Chassis

Contains
8 – standard PCI-X
4 – high-bandwidth PCI-X



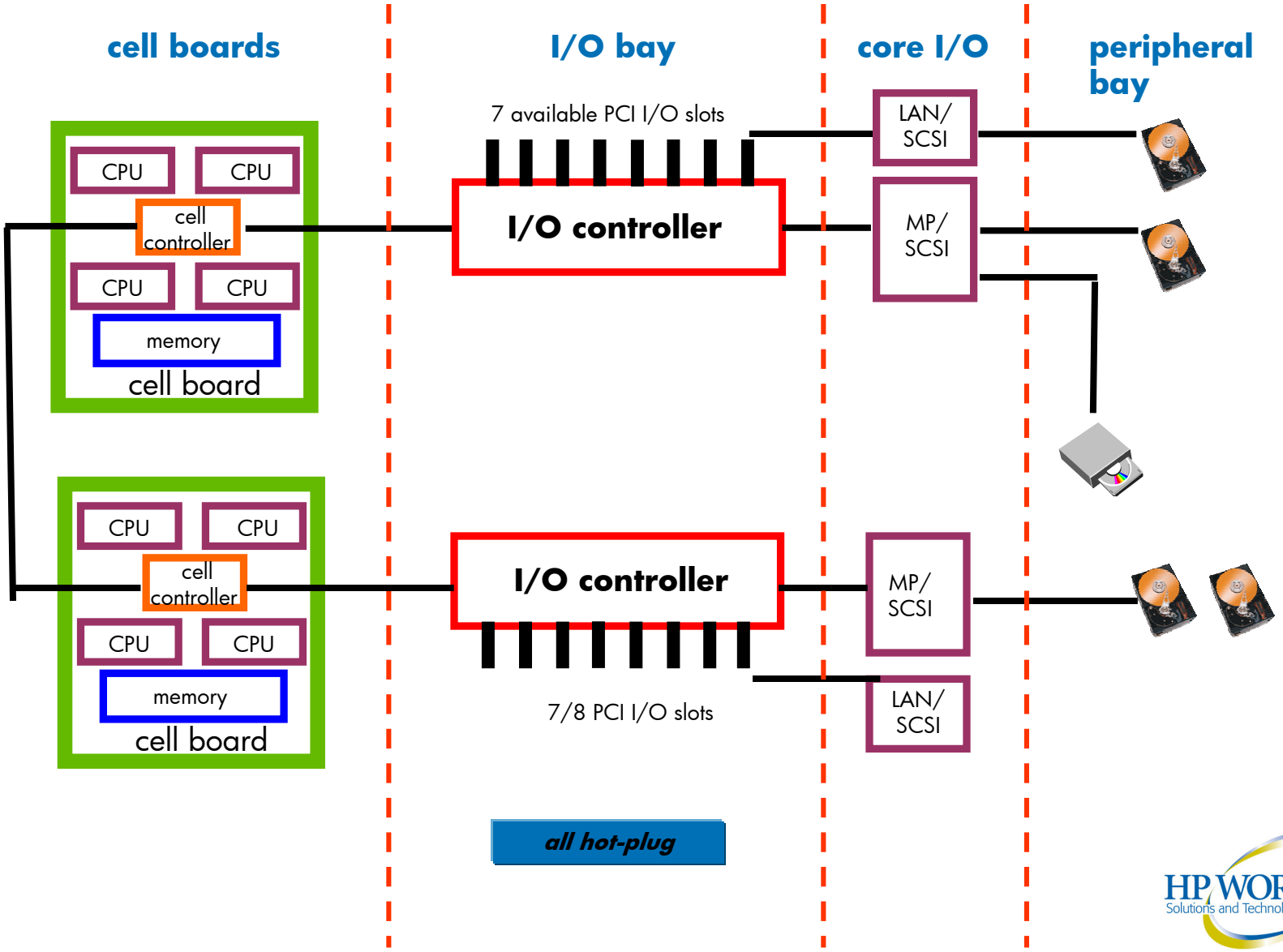
Online PCI addition/deletion
OS dependent

Complete card-to-card isolation

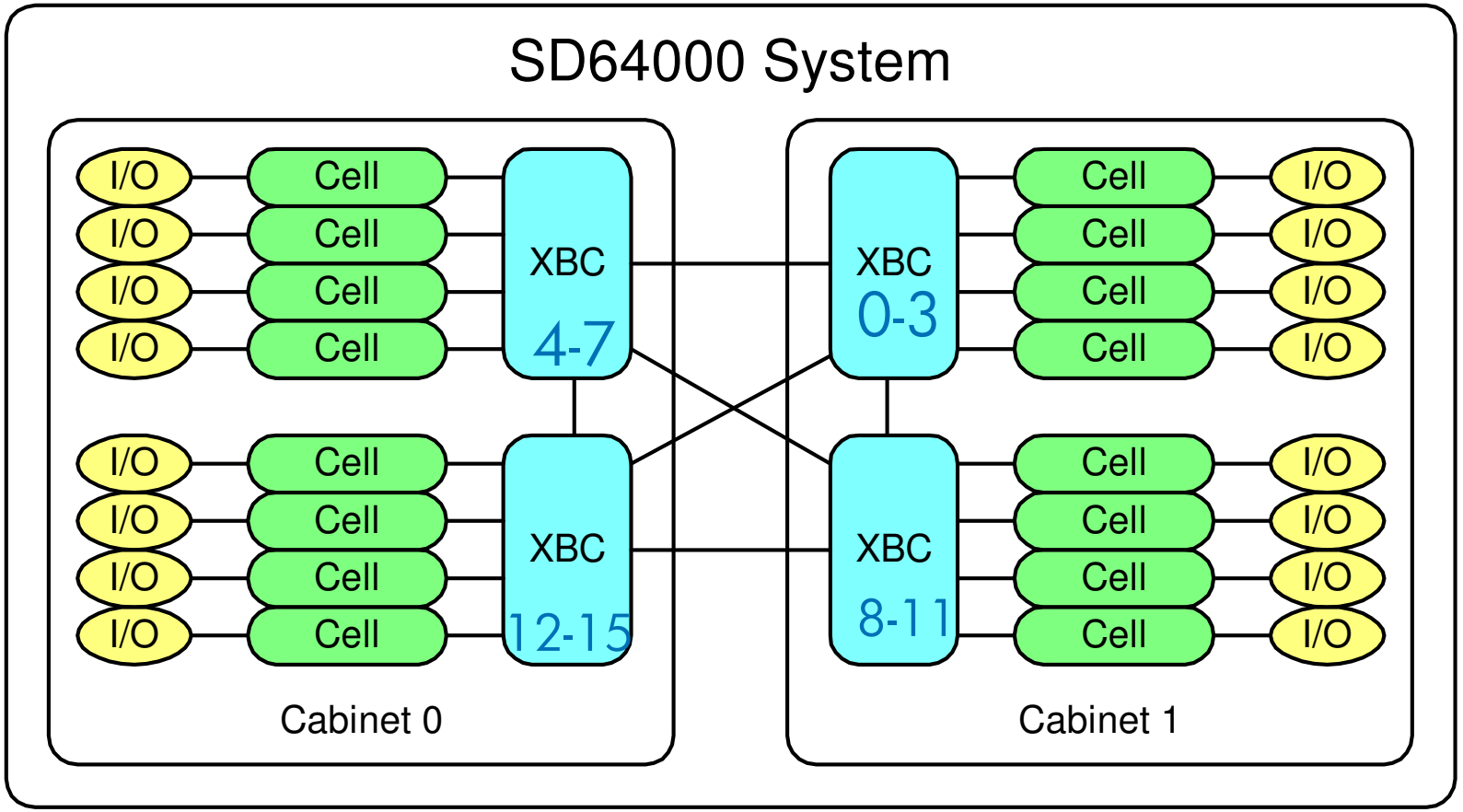
Individual power controls

Two caches, strong ordered access to all memory

Building an 8p Server



Building a Superdome



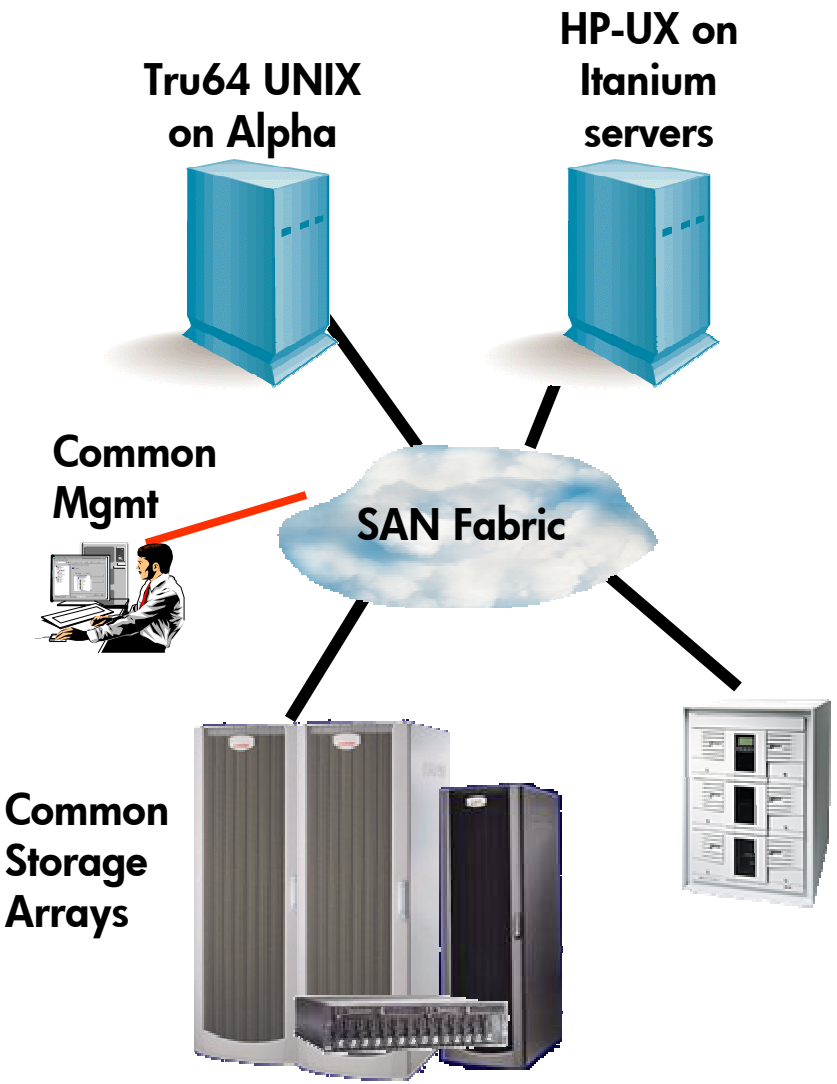
Server Management Considerations

- Firmware
- Storage
- Console
- RAS features
- Topology and performance
- Capacity on demand
- Partitioning
- Management tools

Firmware

- Alpha
  - SRM
  - Power ON -> SRM -> Tru64 UNIX boot loader (osf_boot) -> Kernel /sbin/init
- PA-RISC
  - ISL/IPL
  - Power On → PDC → ISL → hpux → Kernel → /sbin/init
- IA64
  - EFI
  - Power On → EFI → EFI Boot Manager → HP-UX Boot Loader → Kernel→ /sbin/init

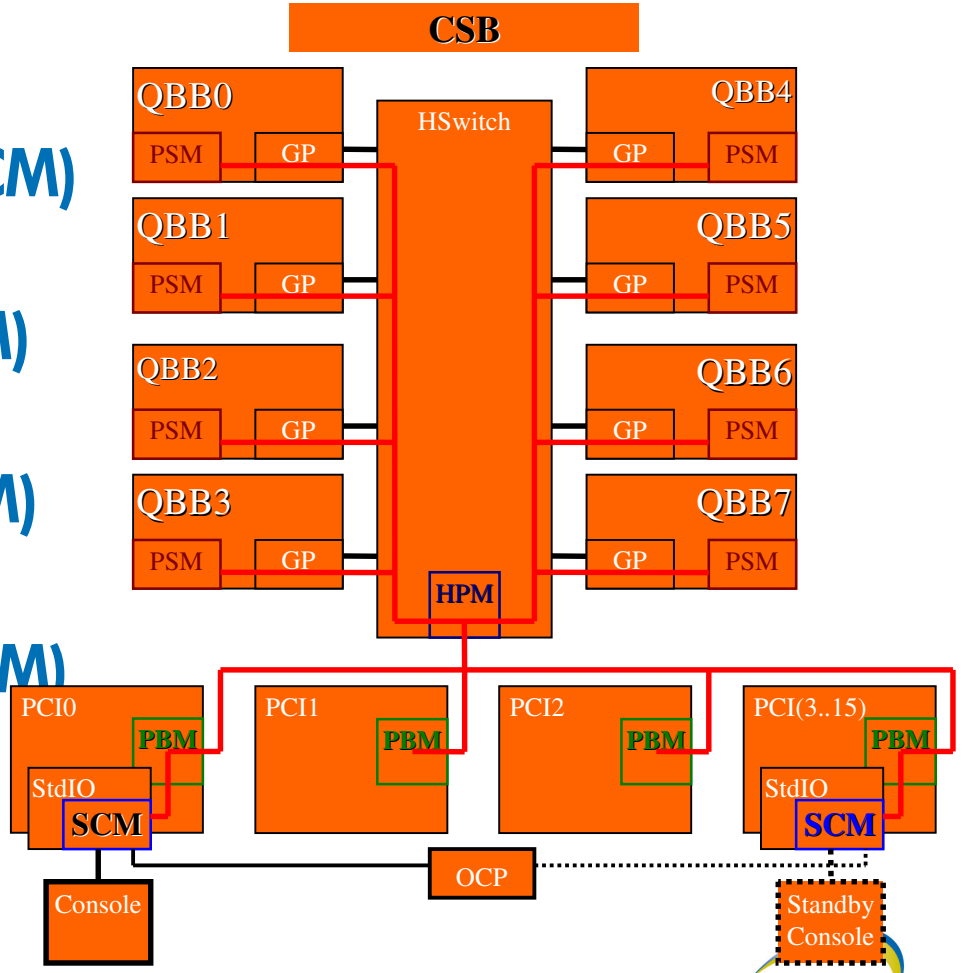
Storage
Tru64/HP-UX co-existence



- Expect Tru64 and HP-UX 11i to co-exist on common SAN infrastructure
  - common SAN fabric
  - common data management software
  - common tape libraries
- Support for current StorageWorks technologies on both Tru64 and HP-UX
  - StorageWorks EVA
  - StorageWorks XP
  - StorageWorks MSA1000 (2H 2004)
- Support for HSG80 arrays on HP-UX on the Itanium processor family
- Expect that parallel SCSI-based storage technologies will not directly migrate
  - HSZ40/50/70/80
  - JBOD SCSI storage
  - Backplane RAID

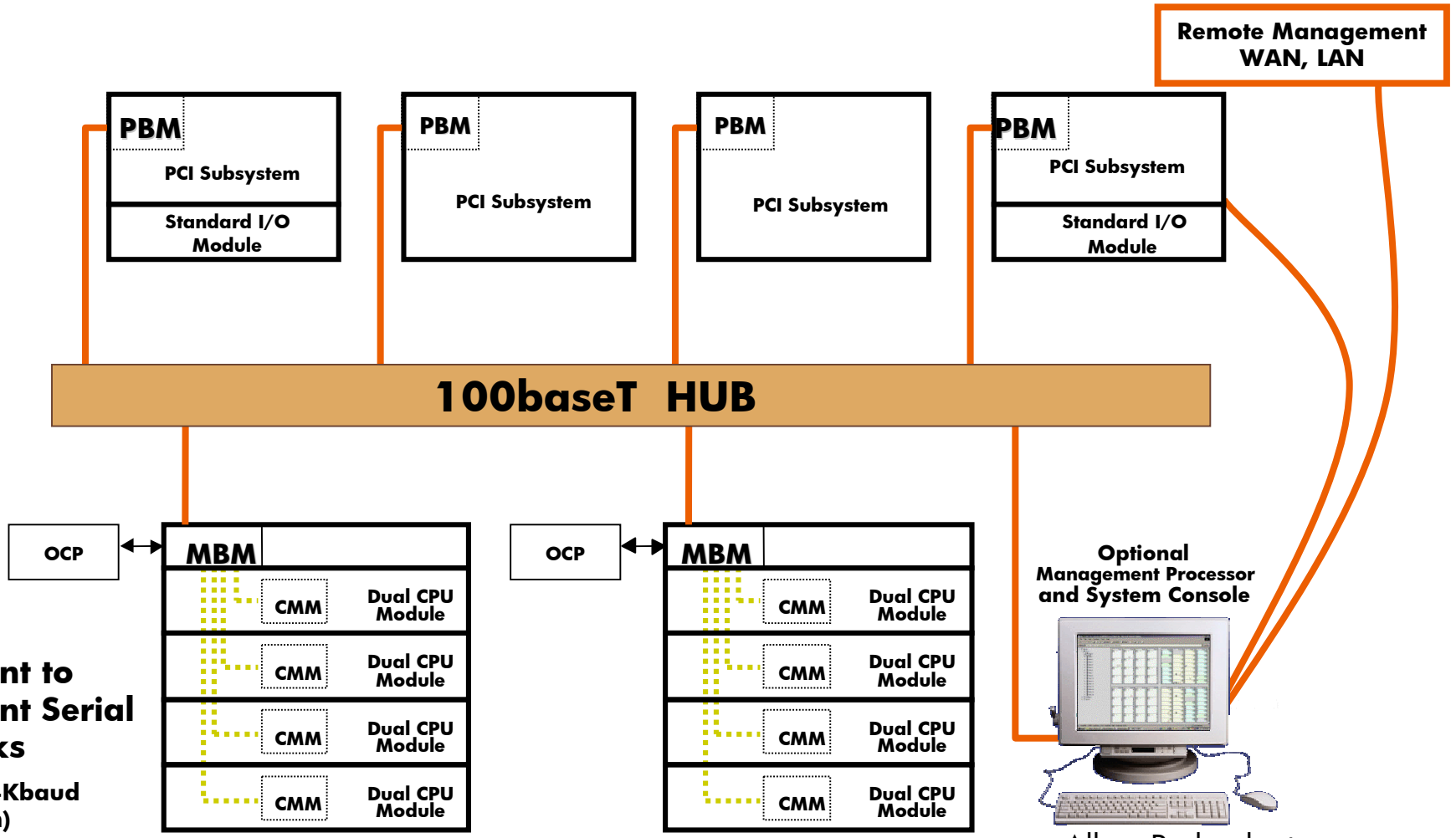
GS320 Console Logical Diagram

- **Console Serial Bus (CSB)**
- **System Console Manager (SCM)**
  - Master of the CSB
- **Power System Manager (PSM)**
  - controls QBB
- **PCI Backplane Manager (PBM)**
  - controls PCI box
- **HSwitch Power Manager (HPM)**
  - controls HSwitch



Master PCI Drawer required per hard partition

GS1280 Console Logical Diagram



GS1280 Console example
Connect to the MBM

 # telnet sally.mycorp.net

Could use serial connection or AMS

Port 323 for SRM

MBM>

MBM> help

The following help topics are available:

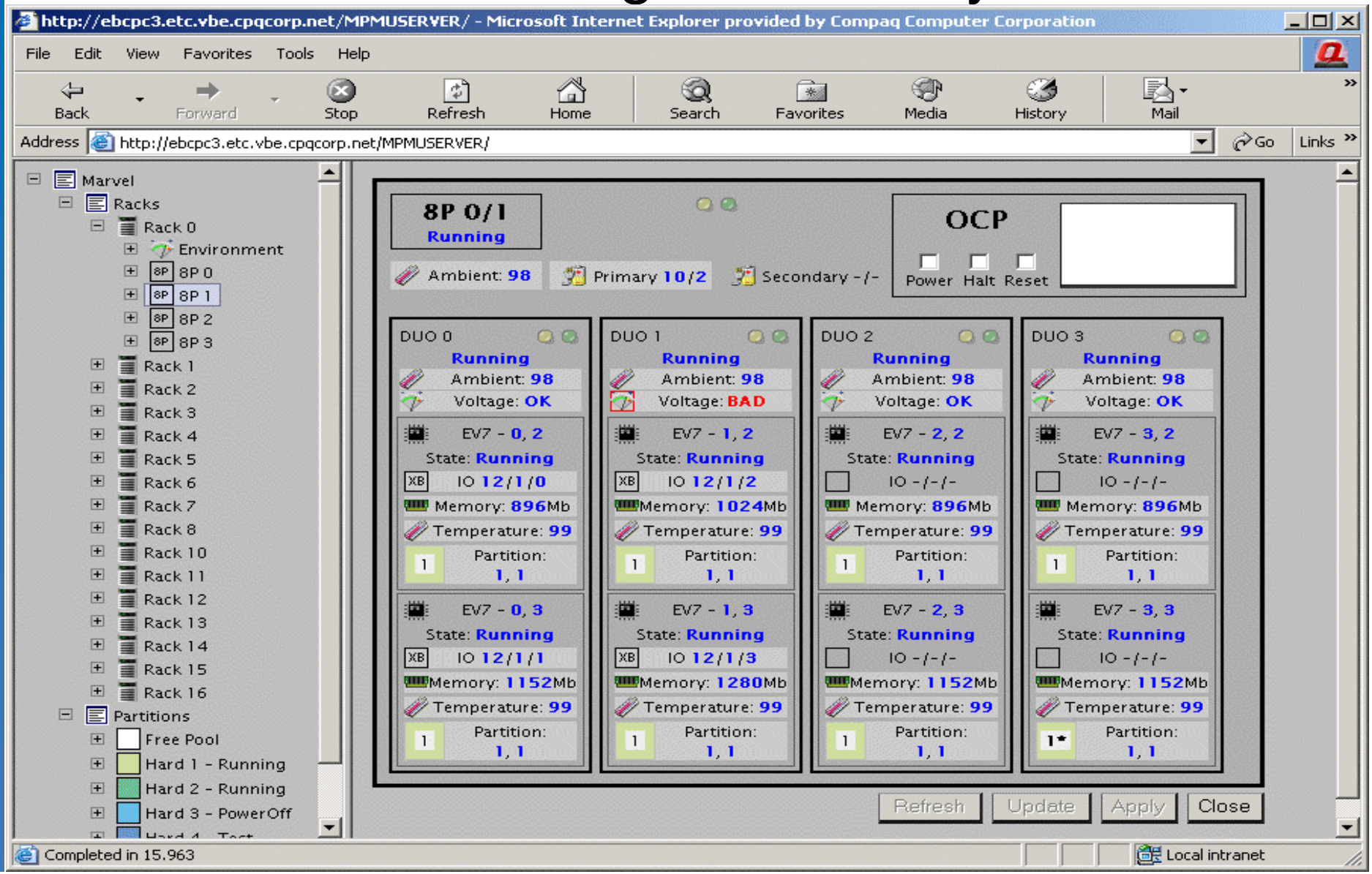
assign component	assign memory	build fru	clear
clear alert	clear display	clear error	clear port
config cable	connect	create partition	d
delete partition	deposit	disable alert	disable remote
disable test	e	el	enable alert
enable remote	enable test	examine	halt in
halt out	hangup	help	init modem
Logout	power off	power on	reset
rlogin	save partition	set	set alert
set baud	set dial	set escape	set flow
set init	set password	set time	shell
show	show cable	show error	show fru
show modem	show network	show partition	show power
show system	show time	show version	test
test alert	test led	update	

MBM> connect –hp <hard_part> -sp <soft_part>

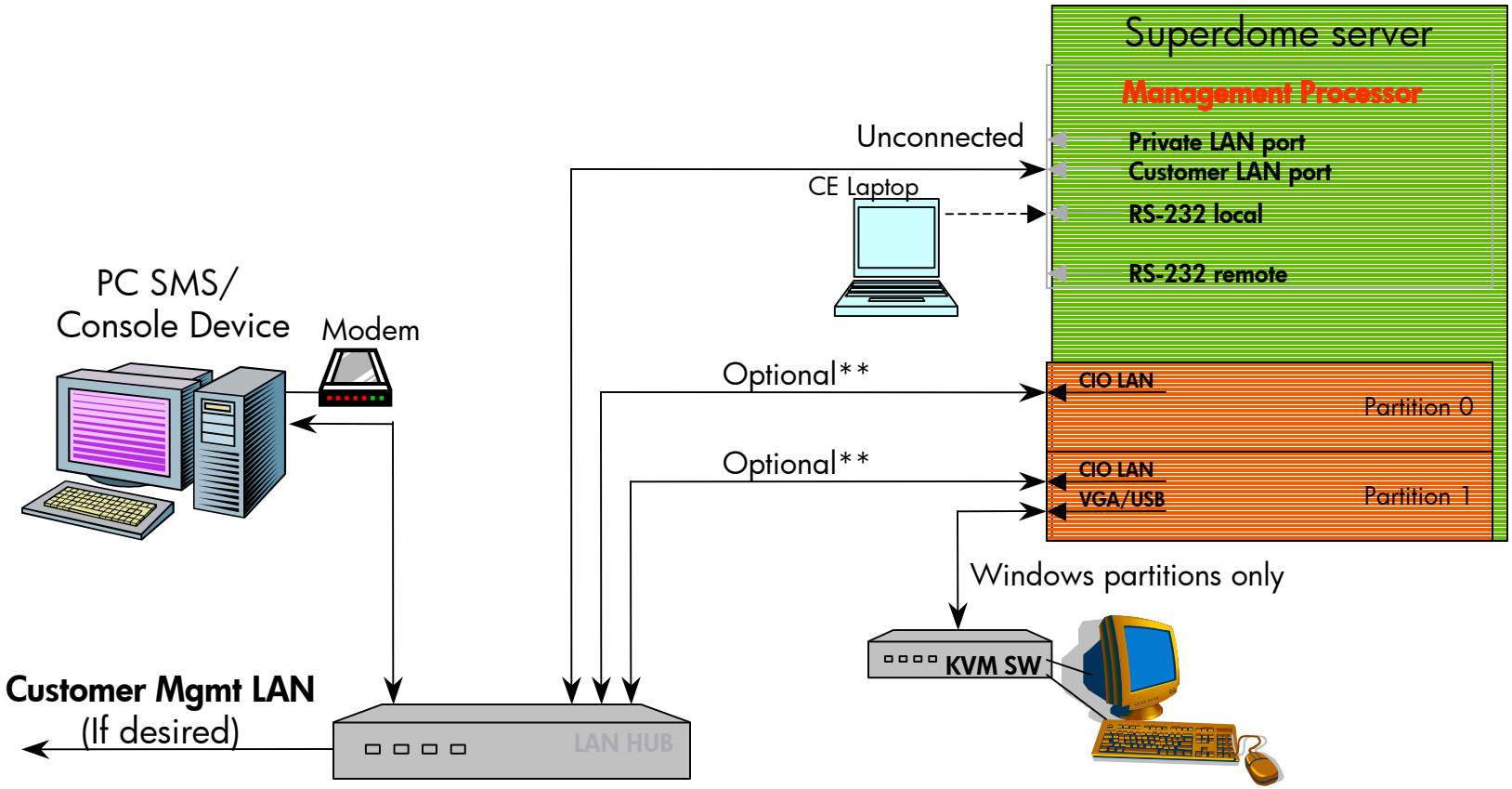
P00>> <Esc><Esc>MBM

To toggle between MBM and SRM

GS1280 Console
Marvel Platform Management Utility



Superdome Console and SMS Network Configuration



Suoerdome Console Connect to MP example

```
# telnet sally.mycorp.net
MP login: Admin
MP password:
[Read-only - use ^Ecf to attach to console.]
[Bumped user - Admin.]
                        Welcome to the
                rp8400 Management Processor
(c) Copyright 1995-2002 Hewlett-Packard Co., All Rights Reserved.
                        Version 3.05

                MP MAIN MENU:
                CO: Consoles
                VFP: Virtual Front Panel (partition status)
                CM: Command Menu
                CL: Console Logs
                SL: Show chassis Logs
                HE: Help
                X: Exit Connection
```

Admin user already logged in will go into 'spy' mode

Superdome Console Connect example

```
MP > co
  Partitions available:
  Part #   Name
   -----        ----
     0)  Partition 0
     1)  Partition 1
     Q)  Quit
     Please select partition number: 0
     Connecting to Console: Partition 0
     (Use ^B to return to main menu.)

     [A few lines of context from the console log:]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
HP-UX rp8400-0 B.11.11 U 9000/800
Console Login: root
```

RAS Tru64 example Online add of a cpu

GS320 example

- add physical cpu respecting configuration rules
- power on cpu using sysman gui or hwmgr cli

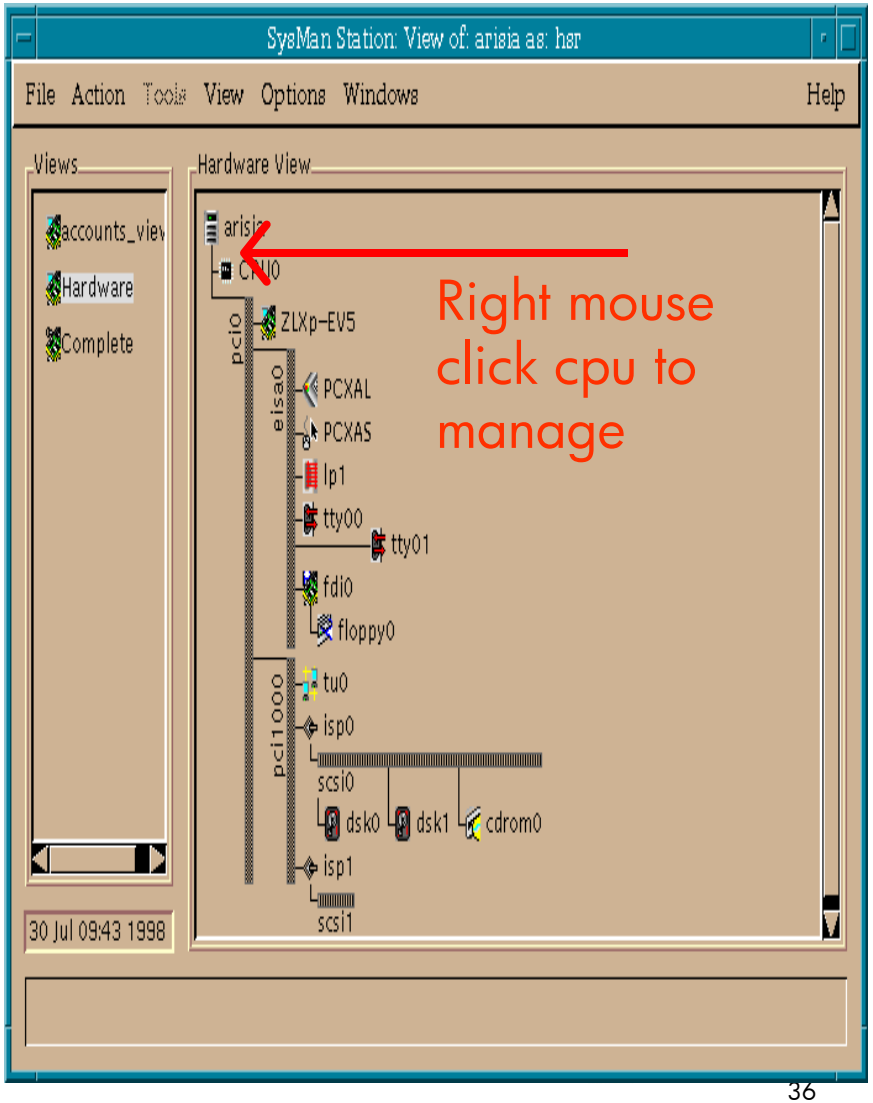
  # hwmgr – power on –name CPU2

  …self tests performed

  …LED goes green (power on)
- online cpu for operating system

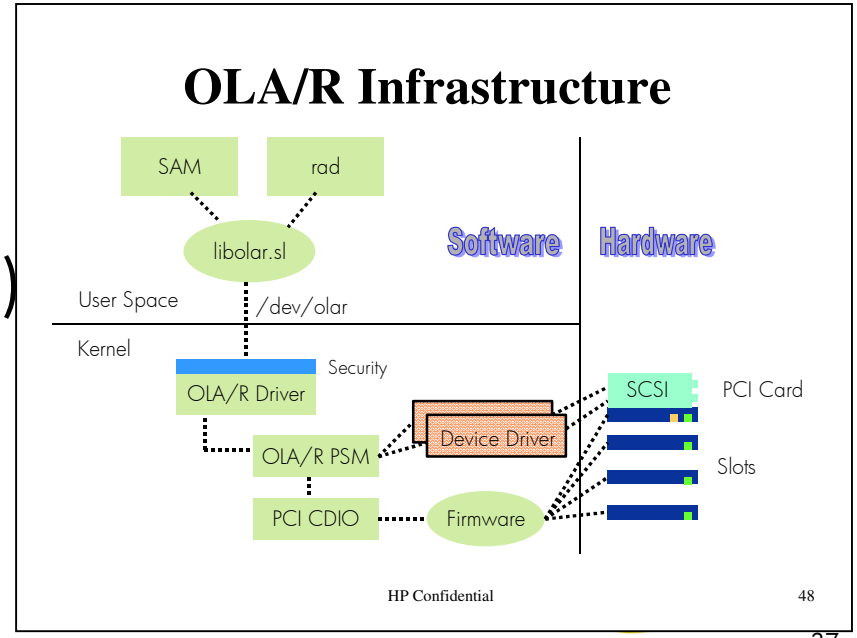
  # hwmgr –online –name CPU2
- Verify status

  # hwmgr –status component

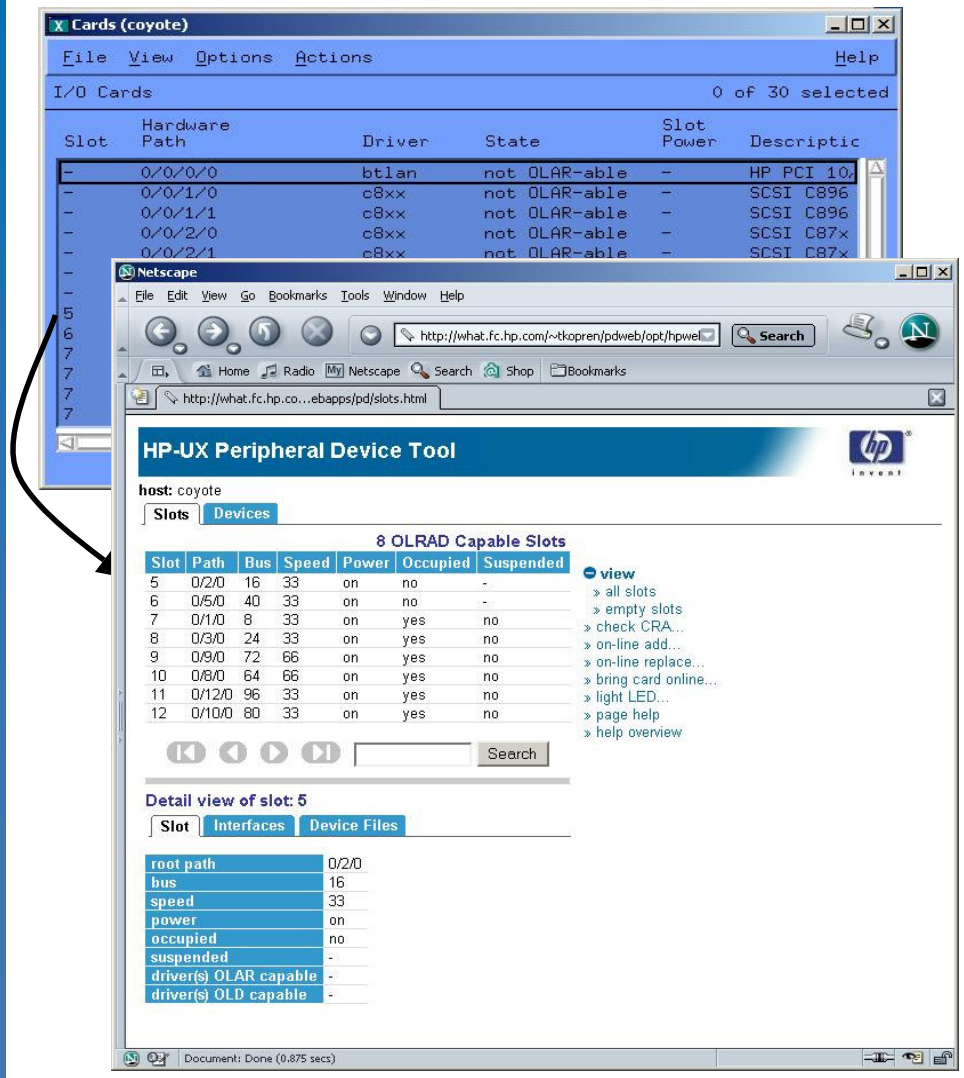


RAS HP-UX Superdome example Pci OLAR

- OLAR capable hardware only
- consider physical impacts such as enough power, same frequency
- manage using SAM or rad
- SAM is the primary interface to OLAR
  - correct tasks & sequences
  - informative & Logging
- rad is a CLI for use when SAM not available (single user mode)
  - no checking
  - requires experienced user



RAS HP-UX V2
New Peripheral Devices (pdweb)

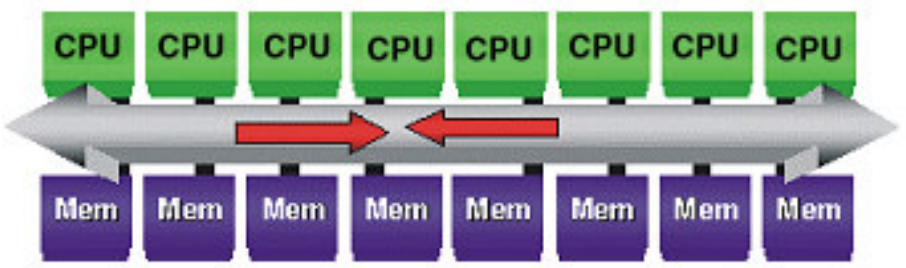


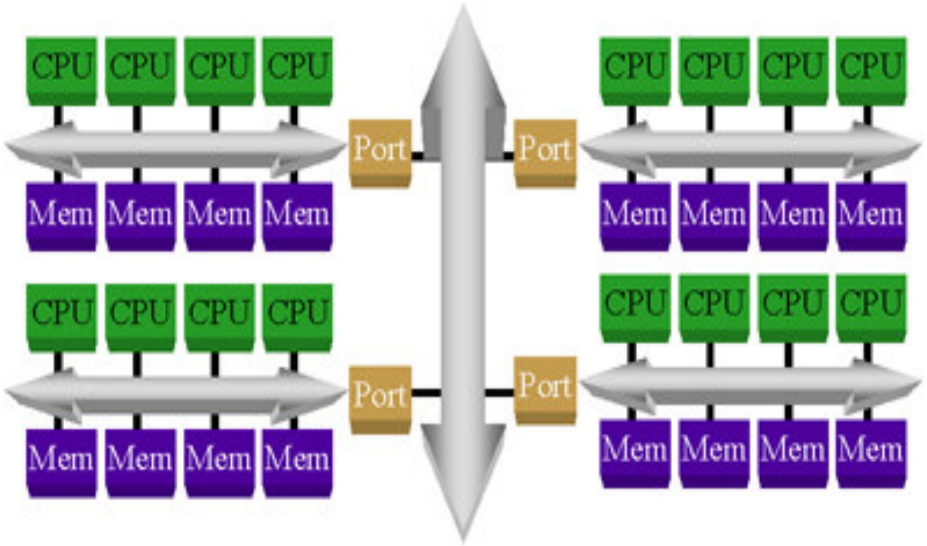
Coming for HP-UX 11i v2

- Same functionality as in SAM PD (replacement); SAM and SCM 3.0 launch
- PCI OLAR support
- Displays slots and devices
- Check CRA (critical resource analysis)
- Improved functionality in CLI
- Command preview
- Creates device files
- Web-based interface
- GUI, messages, help localized

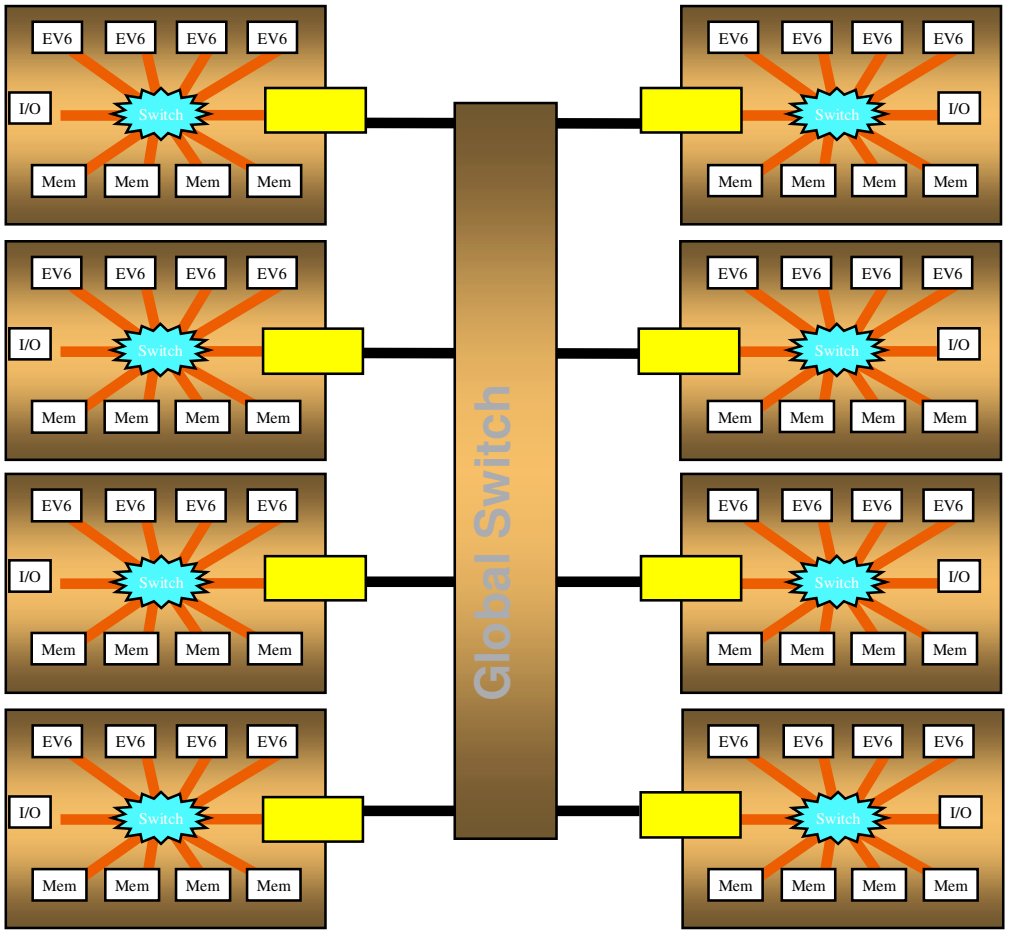
Basic Server Topologies

- UMA/SMP
  - Uniform Memory Access with equal access to all resources
  - Simple Scheduling
  - Scaling Limitations
  - Alphaserver 8400
- NUMA
  - Non Uniform Memory Access
  - Asymetrical Memory & I/O latencies
  - Scheduling may be tailored for latency
  - GS320 family
  - Superdome (PA and IA) cell based servers
  - GS1280 family
    - Introduces mesh



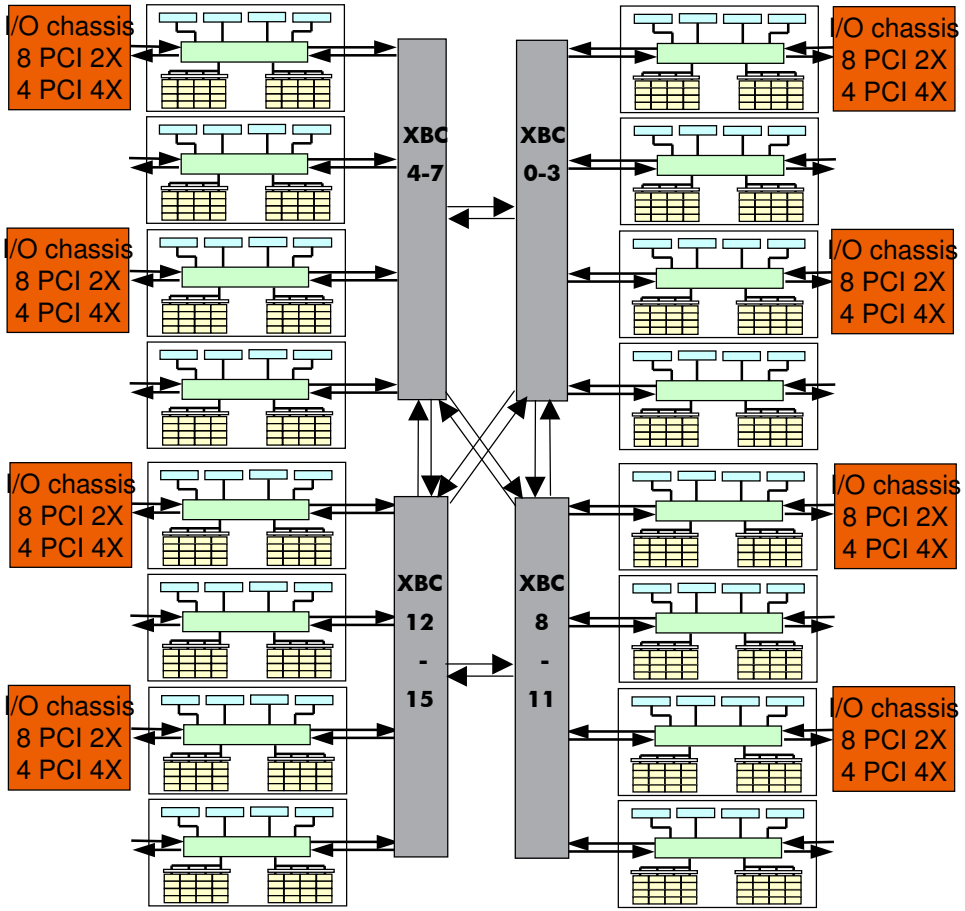


Performance Considerations GS320 family



- Latency
  - Least with QBB, access is local or remote
- Memory
  - populate evenly across QBBs
  - same size boards across QBBs for best interleaving
- CPU
  - generally better to fully populate QBB with cpus before starting another
- IO
  - configure for each QBB to avoid bottlenecks
- Tru64 UNIX
  - V5 is numa aware, V4 is not

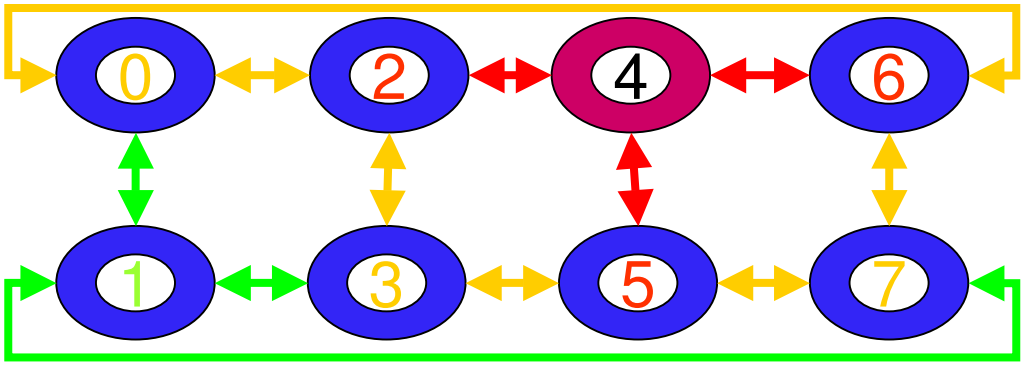
Performance Considerations Superdome family

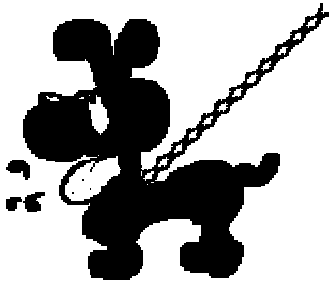


- Memory
  - 2GB min per cell
  - Like sized within partition
- Partitions
  - comprised of a power of 2 number of cells (2,4,8,16) best
- Lowest latency is within XBC
  - cell slots 0-3 on one XBC
  - cell slots 4-7 on another
- HP-UX
  - V2 cell local memory numa aware

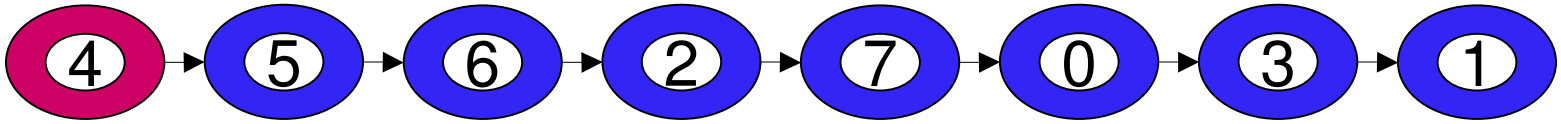
Performance Considerations GS1280 Scheduling and Hops

CPU Numbering





Search Order from CPU 4



Distance	1	1	1	2	2	2	3

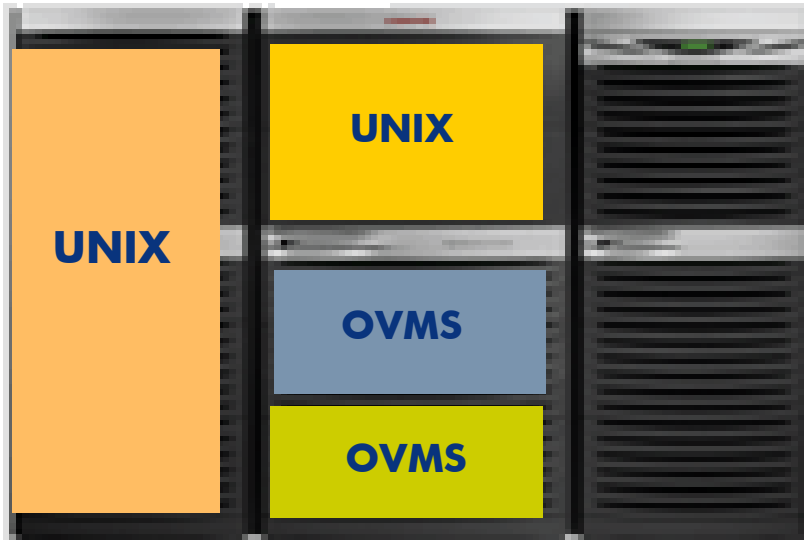
Capacity on Demand Technologies

- Instant Capacity on Demand (iCOD)
  - Activate new permanent capacity when needed
  - Tru64 & HP-UX
- Instant Capacity on Demand – Temporary Capacity (TiCOD)
  - Activate/deactivate new temporary CPU capacity when needed
  - HP-UX
- Pay-per-Use Utility Computing – (PPU)
  - Lease systems based on CPU utilization
  - HP-UX

Partitioning
What and Why?

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.

Hard Partitions on AlphaServer



- balanced QBBs within a partition (GS320)
- configure via console or SCM
- reboot to reconfigure
- dynamic re-assignment of cpus not supported
- each partition requires dedicated i/o

Partition granularity:

- QBB on GS320 (RAD)
- CPU on GS1280 (RAD)

Hard partitions
GS320 Command Line Configuration

- Power up to primary QBB SCM prompt
- After the power up self-test is completed

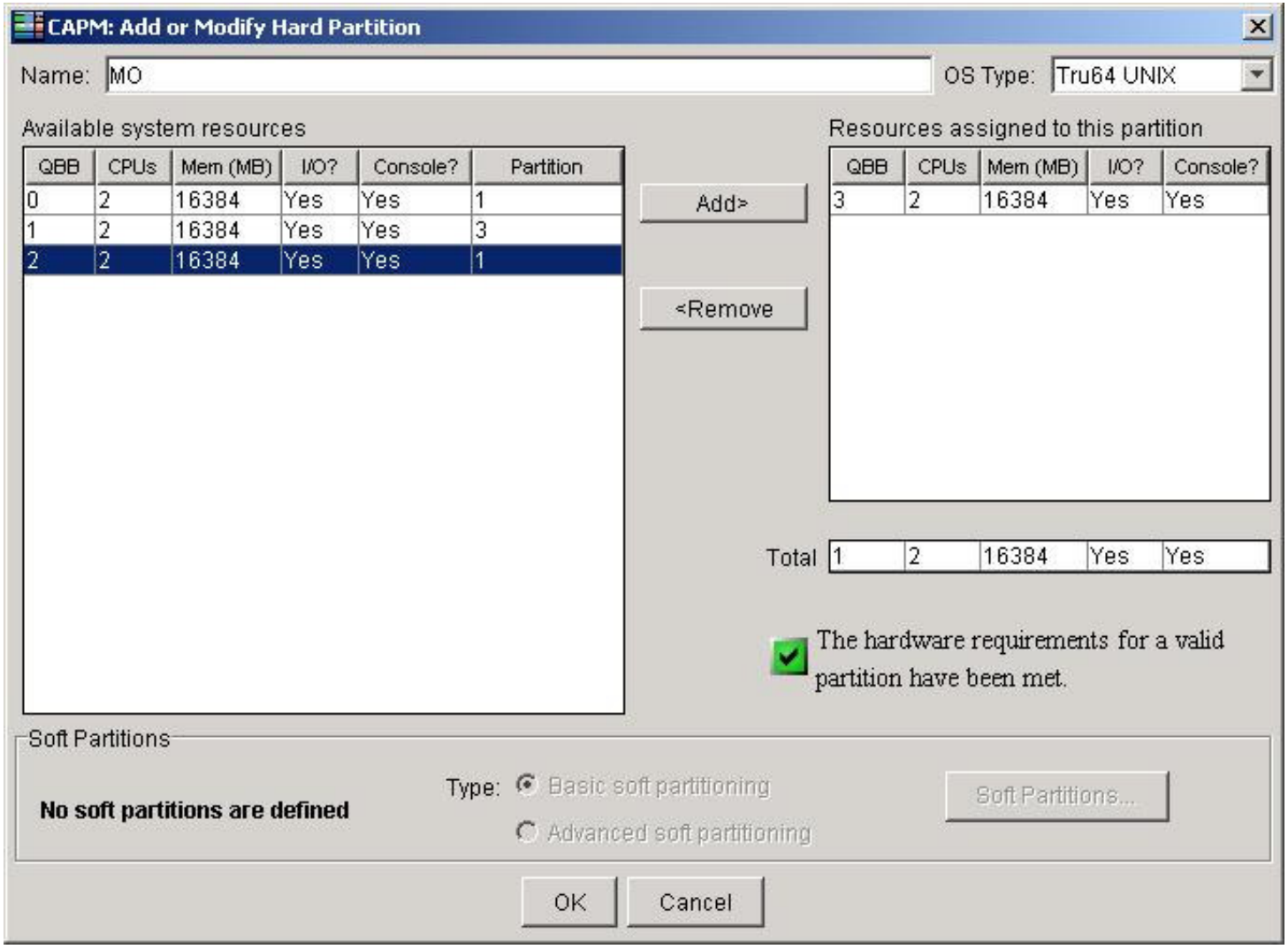
  SCM_E0> set hp_qbb_mask0 3
  SCM_E0> set hp_qbb_mask1 c

- Creates two partitions each having two QBBs
  - Par 0, QBB 0 & 1 (00000011 = 1+2 = 3)
  - Par 1, QBB 2 & 3 (00001100 = 4 + 8 = 12 = c)
- Each part can now start the SRM console

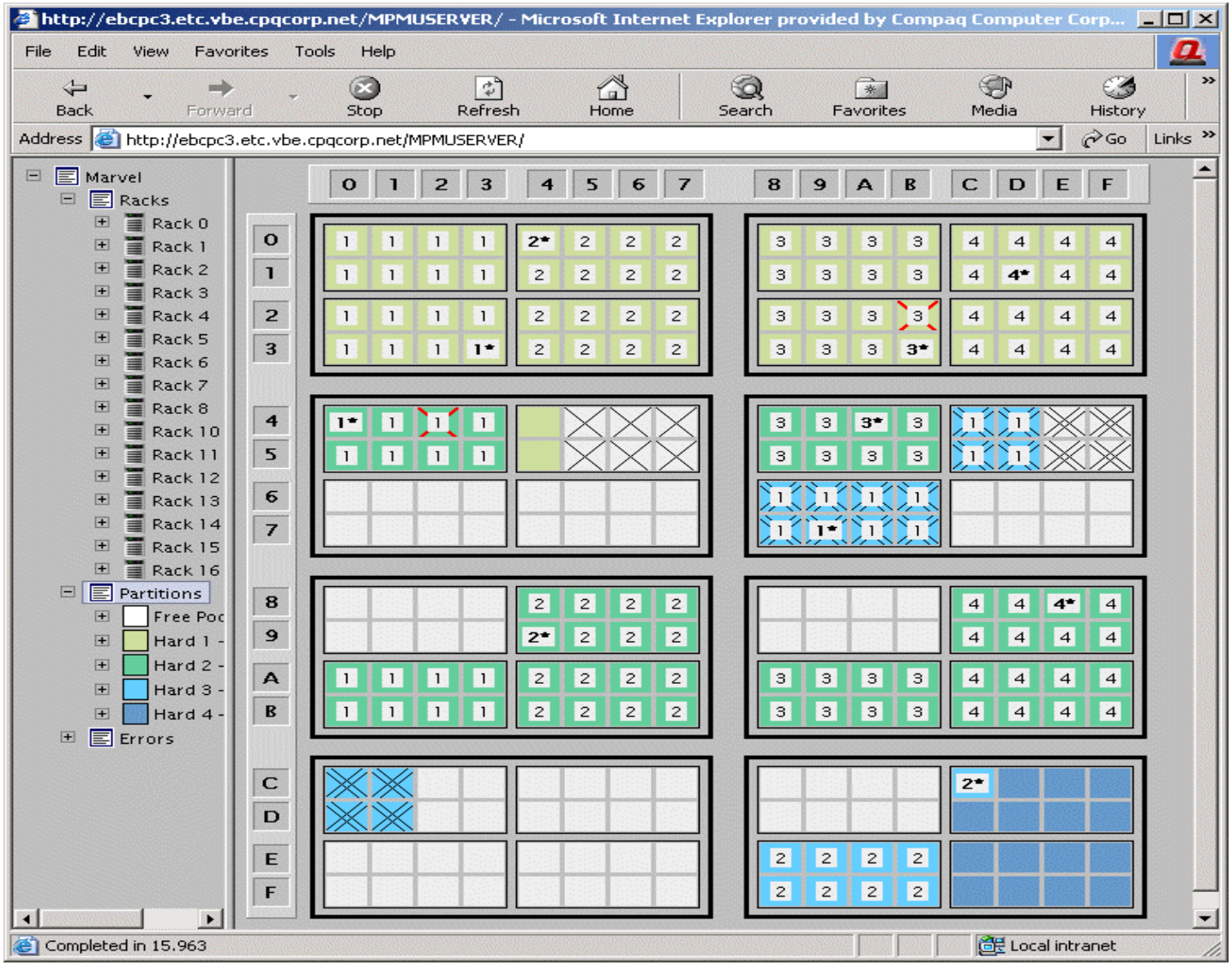
  SCM_E0> power on -par 0
  SCM_E0> power on –par 1

- Install & Configure Operating Systems

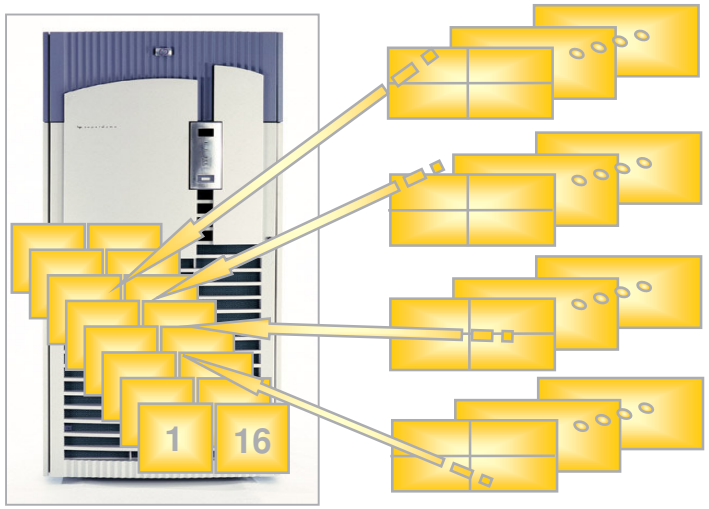
Hard Partition Configuration: GS1280
Marvel Platform Management Utility (MPMU)



Hard Partition Configuration: GS1280 - MPMU 2/2



Hard Partitions on Superdome

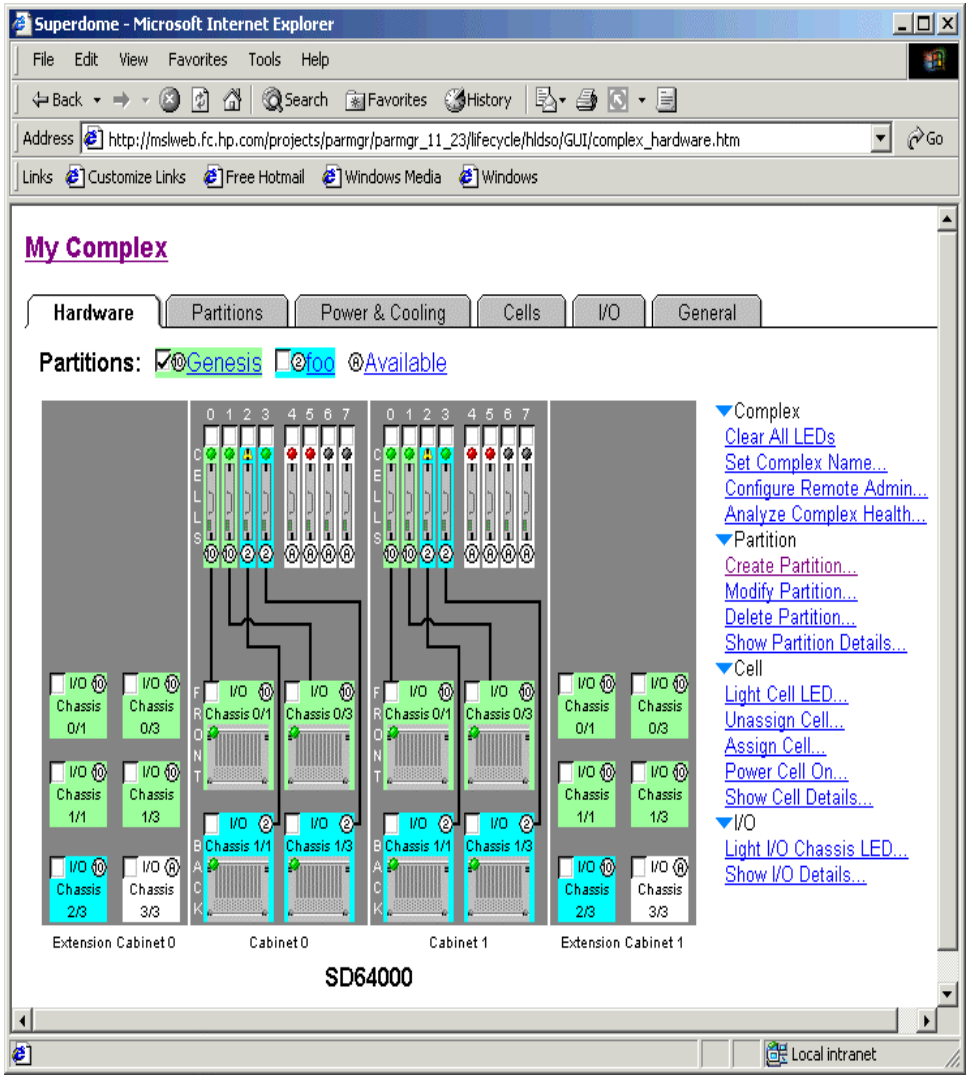


- Hard partitions
  - have a cell boundary
  - Require a core i/o module
- Configure
  - evenly populated cells
  - using MP, SDCLI or parmgr

Superdome Command Line

- Parcreate to create a new partition
- Parmodifyto modify an existing partition.
- Parstatus to display partition & available resource information
- Parremove to remove an existing partition.
- Frupower /Fruled to power on/off cells, cabinets & I/O chasis
- reboot & shutdown

Hard Partitions on Superdome Partition Manager (HP-UX 11i v2)

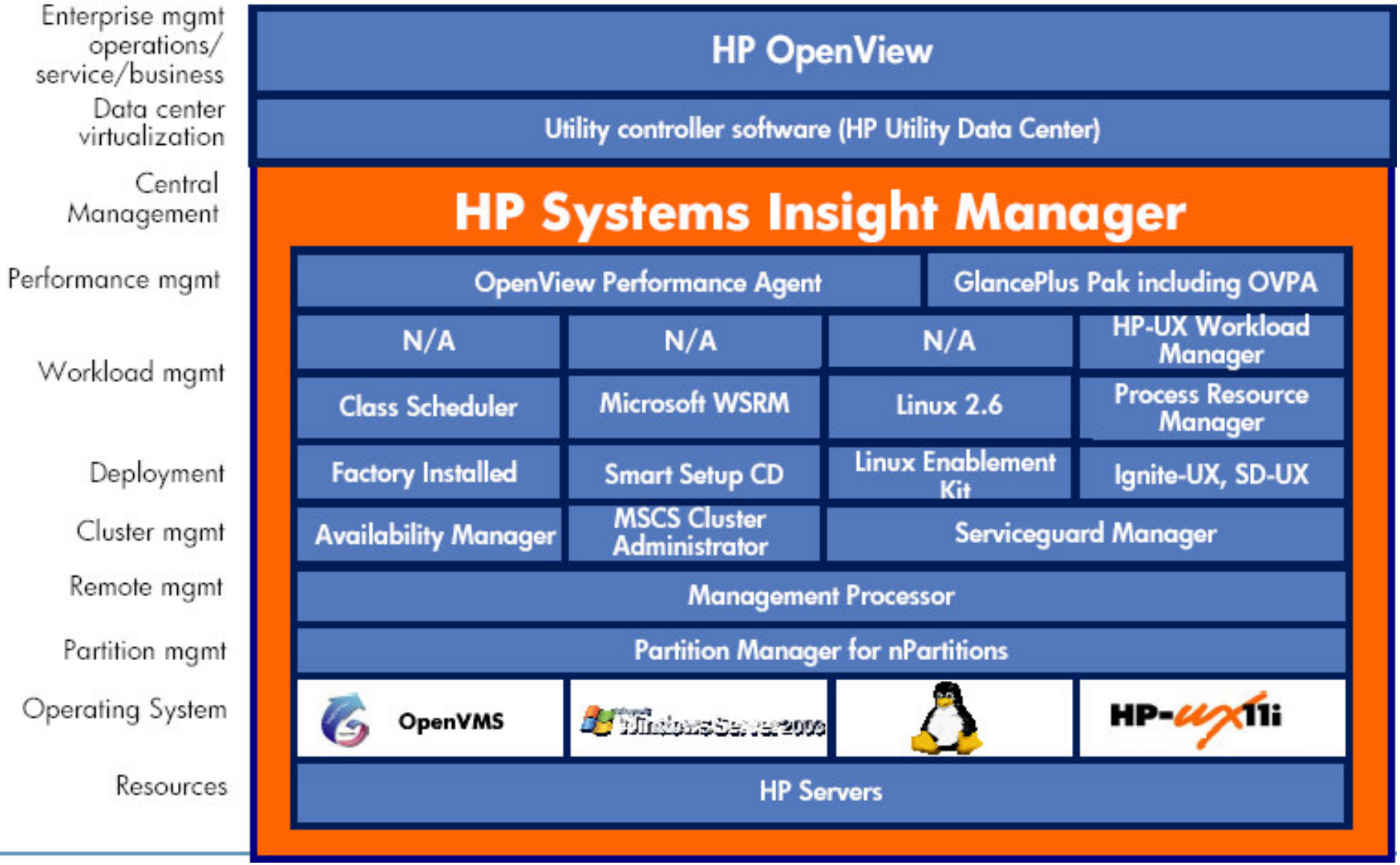


- Hardware-centric management of hard-partitioned servers
- Builds on existing parmgr features
- new web interface
- graphical “big picture” views of
  - nPars, io, cells, power, cooling
- Status lights
- Intelligent menus
- Remote admin
- Compatible with iCOD/PPU and vPars
- Access via
  - ServiceControl Mgr
  - Systems Insight Manager

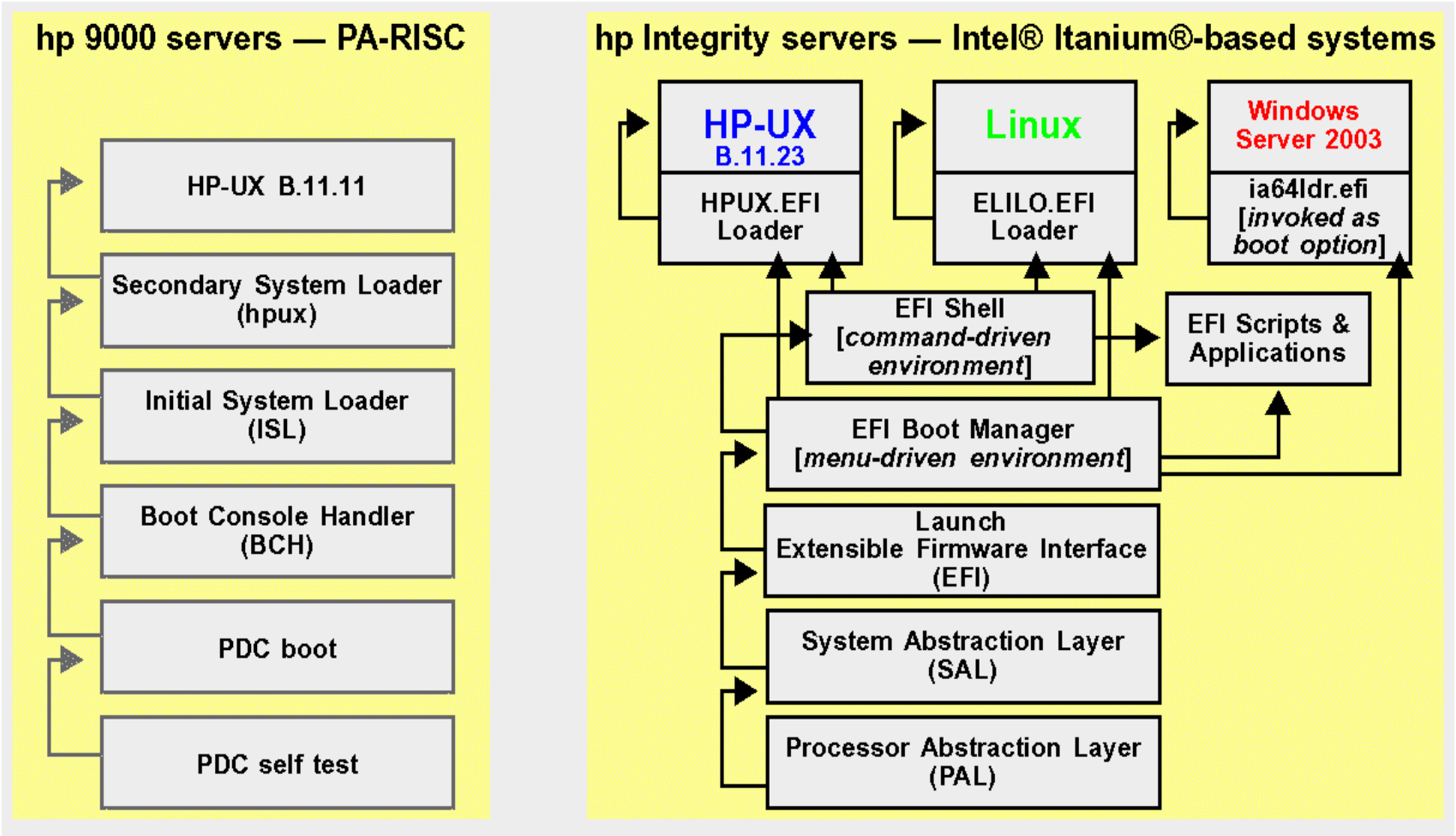
Hard Partitions and Integrity: Supported Operating Systems

- Operating systems supported on nPartition-capable hp Integrity servers:
  - **HP-UX V2**
    - Up to 16 cells or 64 processors per nPartition running HP-UX B.11.23.
  - **Windows Server 2003**.
    - *Datacenter Edition* only for Superdome.
    - *Enterprise Edition* also for rx8620 and 7620 servers.
  - **Red Hat Linux Advanced Server 3.0**.
- OVMS and SUSE support coming.

Hard Partitions and Integrity Multi-OS Capability

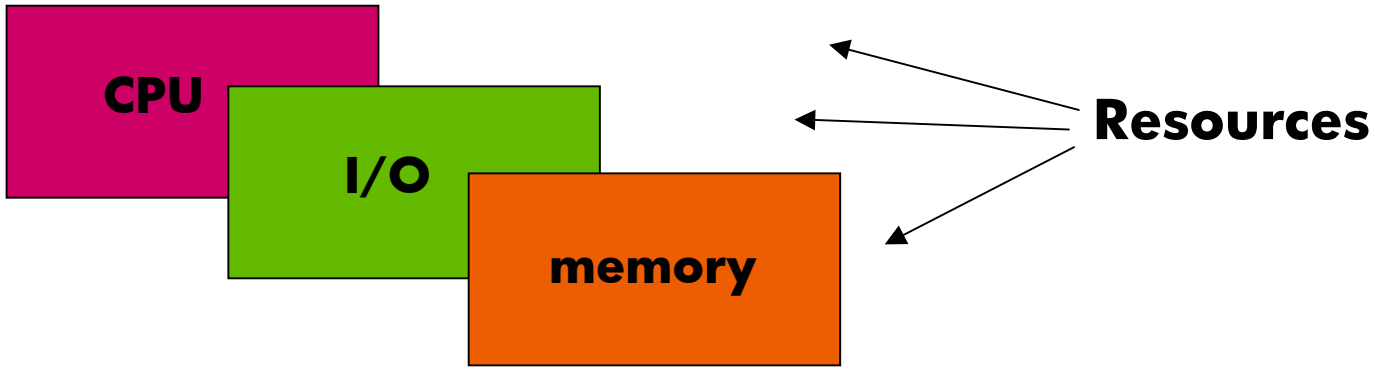


HP Integrity server Boot Overview



Resource Management Capabilities…

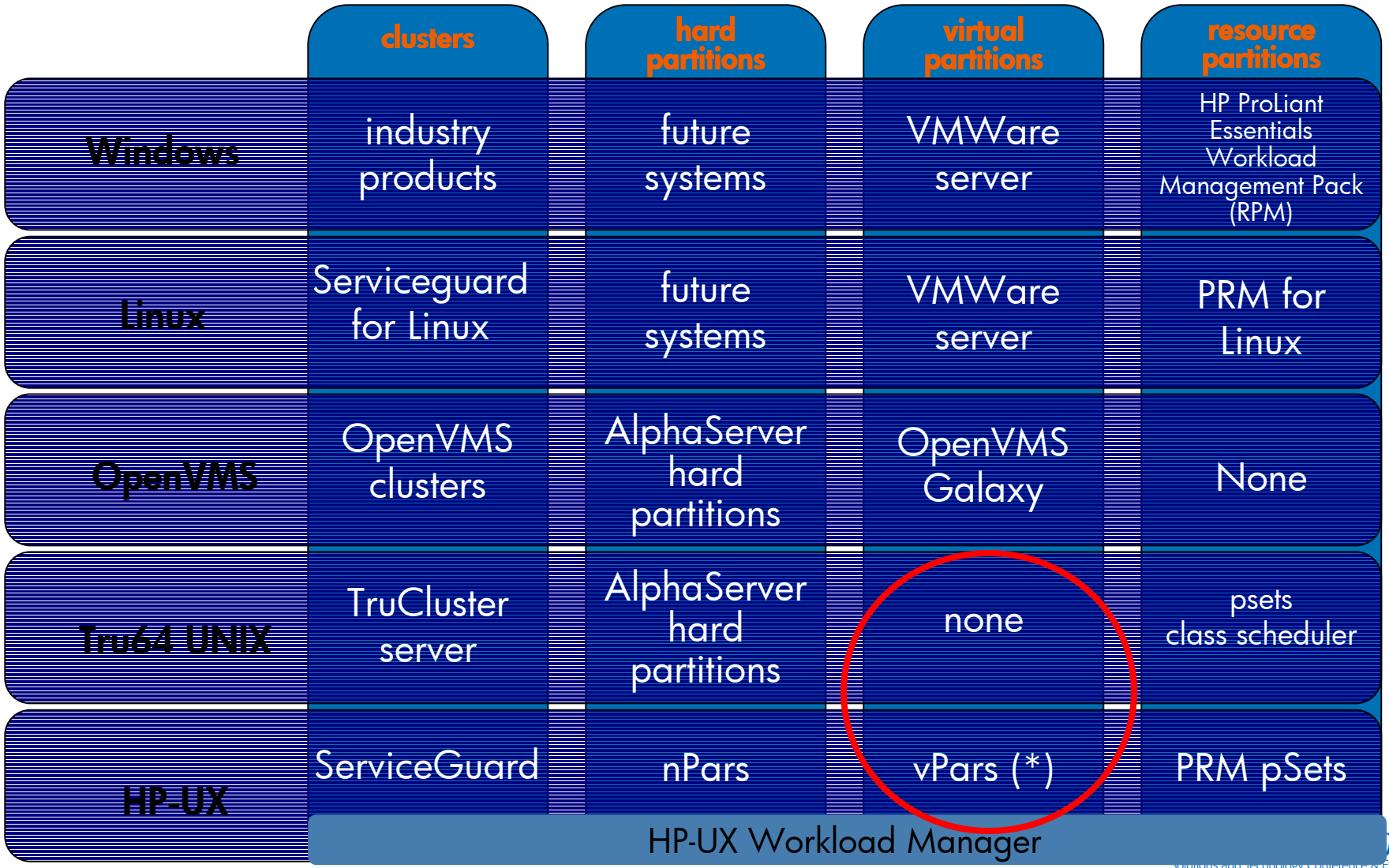
Resource Management Basics



In addition to hardware provided resource management capabilities, software provides another layer of functionality

- Basic UNIX management tools such as cron, at , nice…
- Additional tools
  - Virtual partitions (soft)
  - processor sets
  - class scheduler
  - process resource management
  - cluster features
  - workload management

HP Partitioning Continuum



Virtual Partitions Superdome and HP-UX

Features:

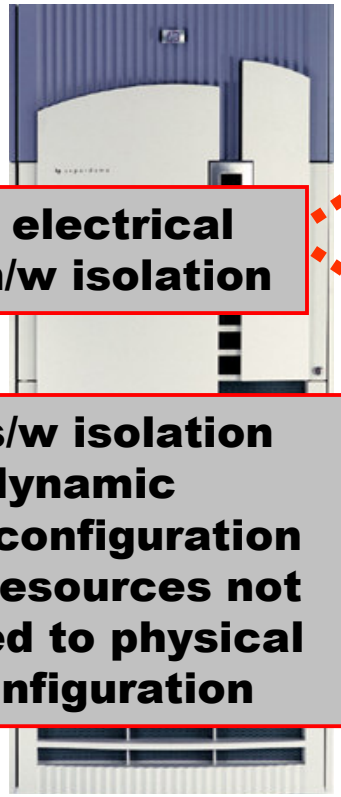
- Provides software isolation
- Ability to run different versions of HP-UX
- Dynamic reassignment of CPUs
  - Floaters only
- CPU granulartity
- Management via CLI or vparmgr GUI
- Single toggle console
- Integration with npar and iCOD
- Vpar Monitor
  - creates illusion of many separate hardware platforms
  - manages shared physical resources
  - monitors health of vpars

Requirements:

- 1 bound CPU minimum
- Memory specified on creation
  - 64MB chunks or range
- I/O connection to a boot disk and LAN

HP-UX Virtual Partitions within Hard Partitions

'floater' cpus move with workload requirements



- **electrical h/w isolation**
- **s/w isolation**
- **dynamic reconfiguration**
- **resources not tied to physical configuration**

nPartition 1 "test" - *HPUX 11i + latest patch*

cpu0 cpu1 cpu2 cpu3 | I/O | CC | memory memory

nPartition 2 "development"

vPar 1 *HPUX 11i + batch*

vPar 2 *HPUX11i + web*

vPar 3 *HPUX 11i large I/O buffer cache*

vPar 4 *HPUX 11i, tuned for large memory*

HP-UX Virtual Partitions
Vpar command line management

- *vparcreate* to create a new virtual partition
- *vparremove* to destroy an existing partition
- *vparmodify* to add/remove resources for an existing partition and modify the attributes (boot path…) of a partition
- *vparboot* to load and launch a virtual partition
- *vparreset* to stop/reset a partition
- *vparstatus* to display Vpar & available resource info

Examples:

- delete CPU from Vpar test

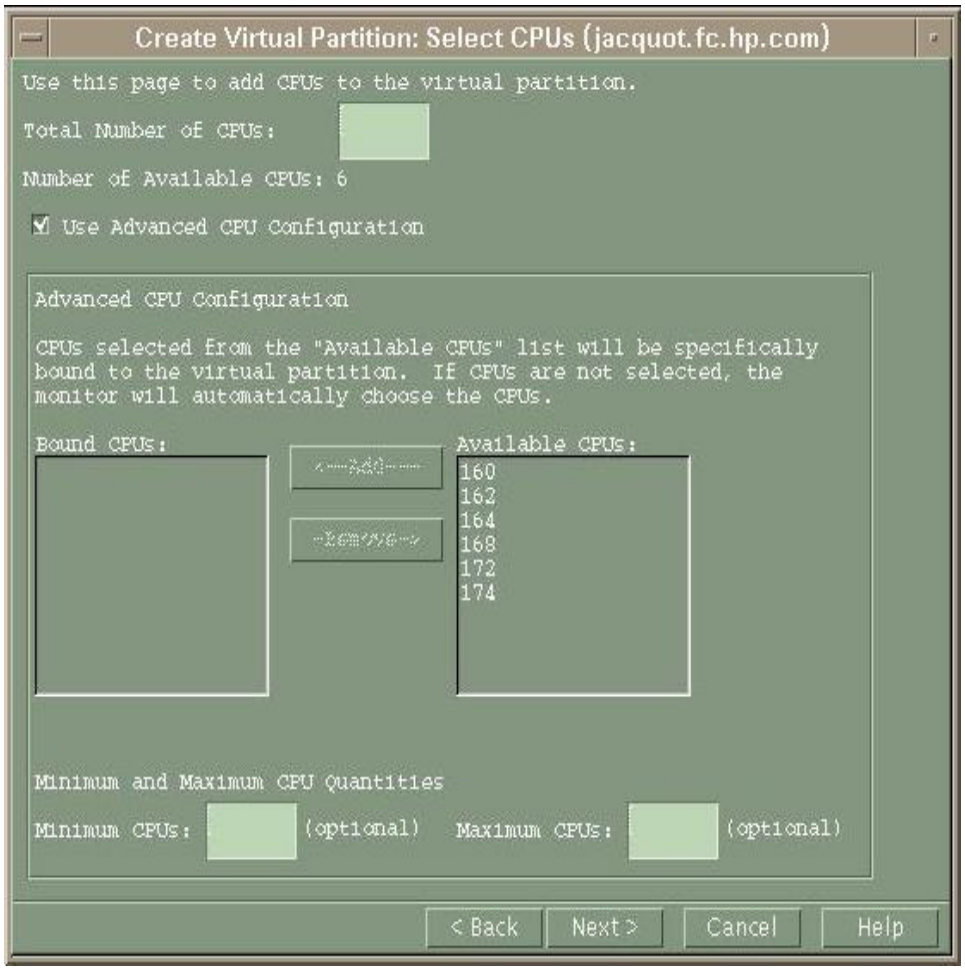
```
# vparmodify –p test –d cpu::1
```

- add a CPU to Vpar dev

```
# vparmodify –p dev –a cpu::1
```

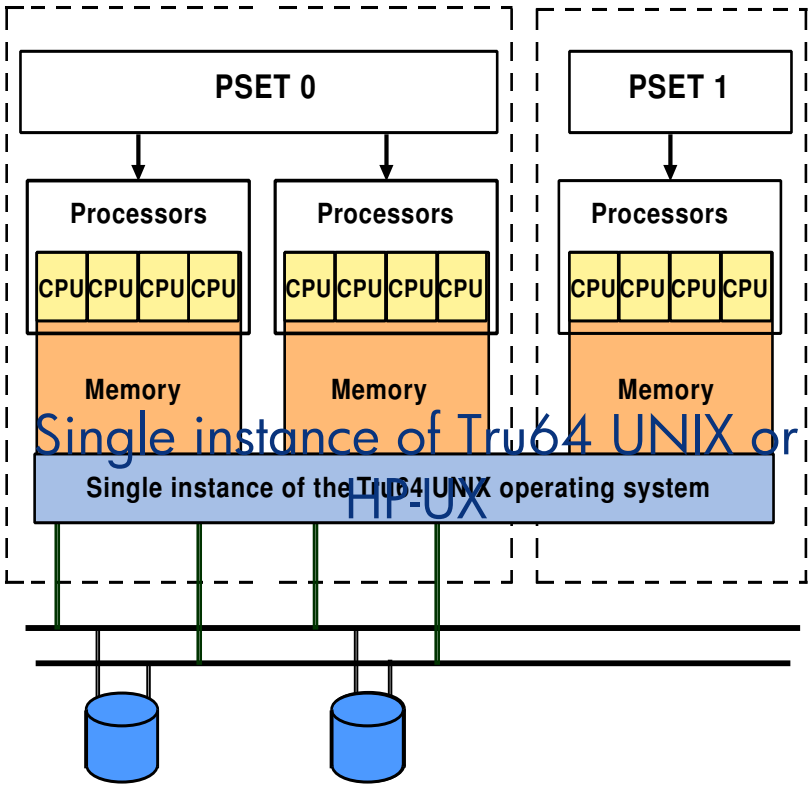

© Copyright Hewlett-Packard Co. 2001

HP-UX Virtual Partitions 'vparmgr' Configuration gui



- 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
- Preview create/modify vpar command lines prior to execution

Processor Sets (psets)



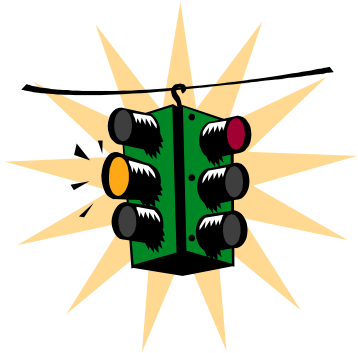
Purpose

- Provides flexibility in multiprocessing (SMP) environments

Description

- A processor set represents a group (subset) of CPUs in the system
- A scheduling allocation domain

Processor Sets (psets)



For both HP-UX and Tru64 UNIX

- Default: a single pset containing all 'active' processors
- A CPU belongs to only one pset at a time
- Processes and threads have binding to one pset at any time
  - child process inheritance
  - dynamic migration supported
- Provides CPU resource isolation for applications and users
- No hardware fault isolation
- Dynamic creation, deletion, and reconfiguration
- CPU 0 always belongs to default pset
- A process will not run on a pset with no CPUs assigned to it
- pset IDs are incremental and not reused

Processor Sets (psets)

- Additional psets can be created and managed by root
  - or delegate via *SAM* or SysMan
- Manage via CLI or system calls
  - Tru64
    - pset _create, pset_assign_cpu, pset_assign, pset_destroy,runon
    - create_pset(), assign_pid_to_pset() etc…
  - HP-UX
    - psrset (single command for all pset operations)
    - pset_create, pset_destroy, pset_assign, pset_bind, pset_getattr etc…
- System activity
  - Tru64
    - scheduled in default pset
  - HP-UX
    - system threads will ignore psets
    - HP-UX utilities (top, glance) are 'pset' aware
- Performance
  - Consider logical boundaries on numa systems

Processor Set Configuration

HP-UX

- **create pset**

  # psrset -c 1
  successfully created pset 3
  successfully assigned processor 1 to pset 3

- **assign cpu to pset**

  # psrset -a 3 2
  successfully assigned processor 2 to pset

- **assign pid to pset**

  # psrset -b 3 2523
  successfully bound pid 2523 to pset 3

- **destroy a pset**

  # psrset -d all
  psrset: pset 0 cannot be destroyed
  successfully destroyed pset 3

Tru64

- **create pset**

  # pset_create
  pset_id = 3

- **assign cpu to pset**

  # pset_assign_cpu 3 2

- **assign pid to pset**

  # pset_assign_pid 3 2523

- **destroy a pset**

  # pset_destroy 3

Tru64 Processor Set Example

Create pset

```
# /usr/sbin/pset_create
pset_id = 5
```

Identify required CPUs

```
# /usr/sbin/psrinfo
0     on-line since     03/19/2002     12:00:00
1     on-line since     03/19/2002     12:00:00
```

Allocate CPUs to pset

```
# /usr/sbin/pset_assign_cpu  5 1
```

Verify status

```
# /usr/sbin/pset_info
Number of processor sets on system = 2
Pset_id       # cpus    # pids    #threads load_av      created
    0            1        49        127     0.14     04/19/2002  12:00:00
    5            1         0          0     0        05/01/2002  16:50:00
```

Start a process

```
# runon -p  5 -x xclock
```

HP-UX Processor Set Example

```
CPU   Util    pset ID   LoadAvg(1/5/15 min)   CSwitch   Last Pid
-------------------------------------------------------------------
 0    100.0     0       1.8/  1.9/  1.3          734        2504
 1    100.0     0       1.7/  0.9/  0.4          151        2523
```

2 cpu hog jobs running in default pset

create pset

```
# psrset -c 1
successfully created pset 3
successfully assigned processor 1 to pset 3
CPU   Util    pset ID   LoadAvg(1/5/15 min)   CSwitch   Last Pid
-------------------------------------------------------------------
0  100.0       0       1.9/  1.8/  1.3          727        2504
1    0.4       3       1.5/  1.0/  0.5           49           4
```

New pset = 3

HP-UX Processor Set Example (2)

Bind hog process 2523 to the new pset

```
#  psrset -b 3 2523
successfully bound pid 2523 to pset 3
CPU   Util    pset ID   LoadAvg(1/5/15 min)   CSwitch   Last Pid
-------------------------------------------------------------------------------
 0   99.8       0        2.3/  2.0/  1.4          728       2504
 1   99.4       3        1.2/  1.0/  0.5          166       2523
```

Hog load moved to pset 3

Bind the other hog process to new pset

```
#  psrset -b 3 2504
successfully bound pid 2504 to pset 3
CPU   Util    pset ID   LoadAvg(1/5/15 min)   CSwitch   Last Pid
-------------------------------------------------------------------------------
 0    0.9       0        1.7/  1.9/  1.4          308       1209
 1   99.3       3        1.8/  1.2/  0.6          131       2504
```

All load in pset 3

HP-UX Processor Set Example (3)

Add a 'floating' cpu to this resource partition

 # vparmodify -p vpar2 -a cpu::1

```
CPU   Util    pset ID   LoadAvg(1/5/15 min)   CSwitch   Last Pid
-------------------------------------------------------------------------------
 0    1.9        0        0.9/  1.6/  1.4        166        3
 1  100.0        3        2.4/  1.5/  0.8        166     2523
 2    0.0        0        0.0/  0.0/  0.0        220     1209
```

New cpu belongs to default pset

Move new cpu to our busy processor set (pset 3)

 # psrset -a 3 2

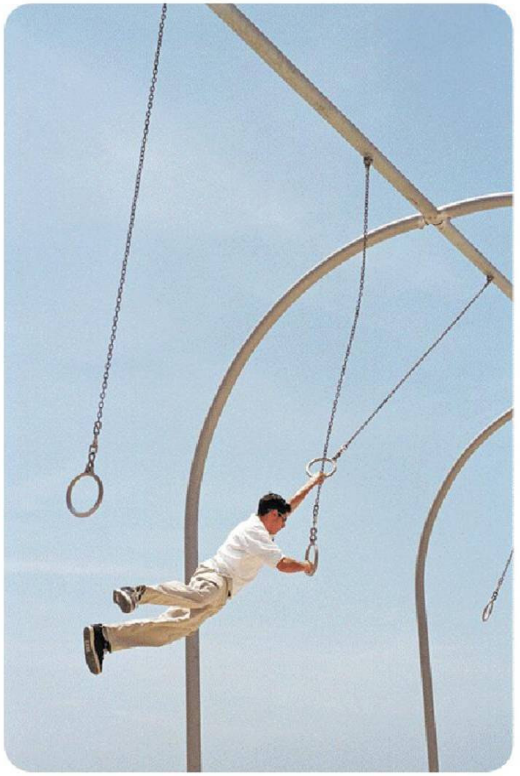
successfully assigned processor 2 to pset 3

```
CPU   Util    pset ID   LoadAvg(1/5/15 min)   CSwitch   Last Pid
-------------------------------------------------------------------------------
 0    1.3        0        0.2/  1.1/  1.2        364     1209
 1  100.0        3        2.6/  2.0/  1.1        152     2504
 2  100.0        3        0.4/  0.1/  0.0         88     2523
```

HP-UX Processor Sets and PRM

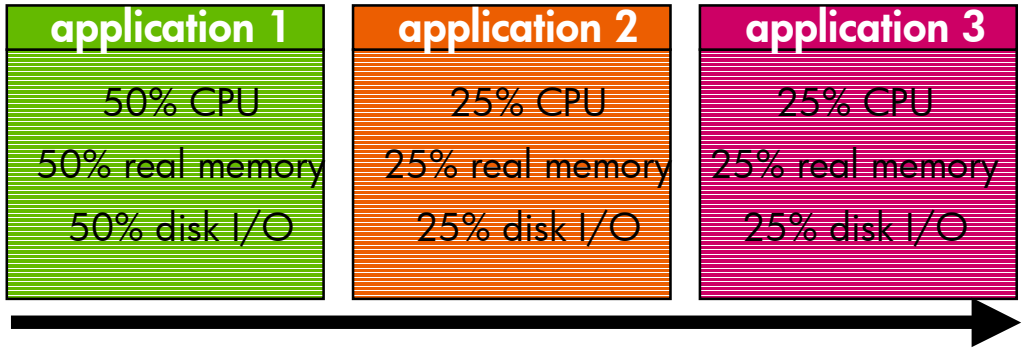
Process Resource Manager (PRM) is additional software, not required to use psets but providing some additional benefits…

- Java based GUI for configuration of psets
- Configuration maintained across reboots
- Can further partition a pset using the fair share scheduler within the default pset
- Processes and users are automatically moved into the appropriate pset or fair share group
- Provides memory management, making it possible to isolate a share of memory to a pset.
- Provides disk bandwith management



HP-UX Process Resource Manager (PRM)

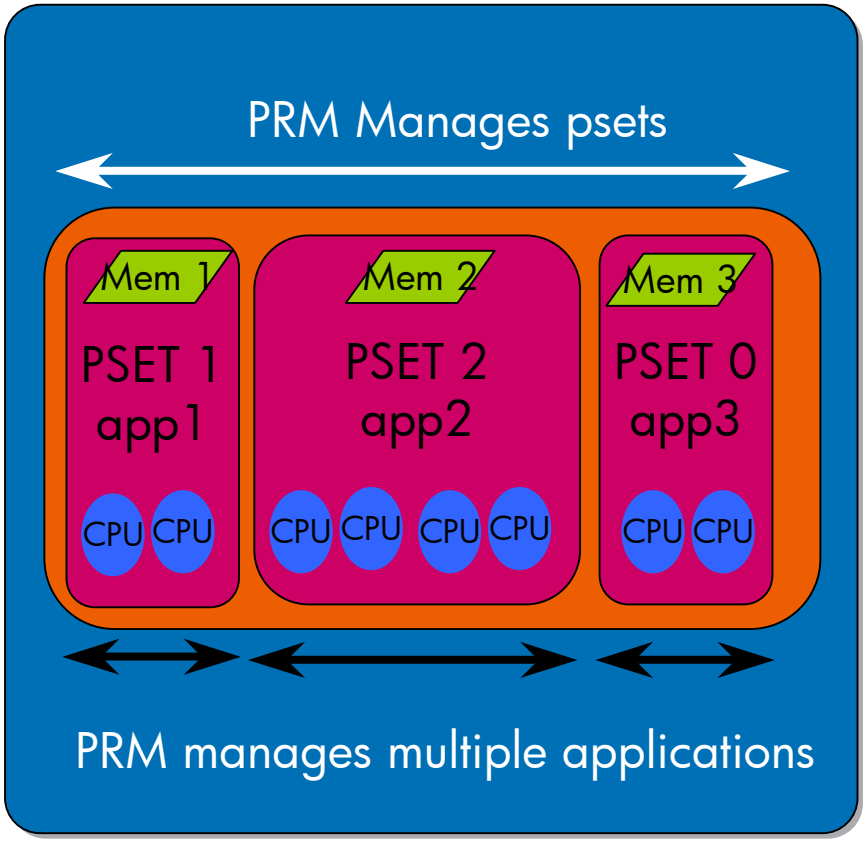
PRM is a tool used to control contention for resources between applications within a single instance of HP-UX



- Administrator
  - defines groups
  - sets resource allocation
- Groups are
  - users or groups of users
  - applications

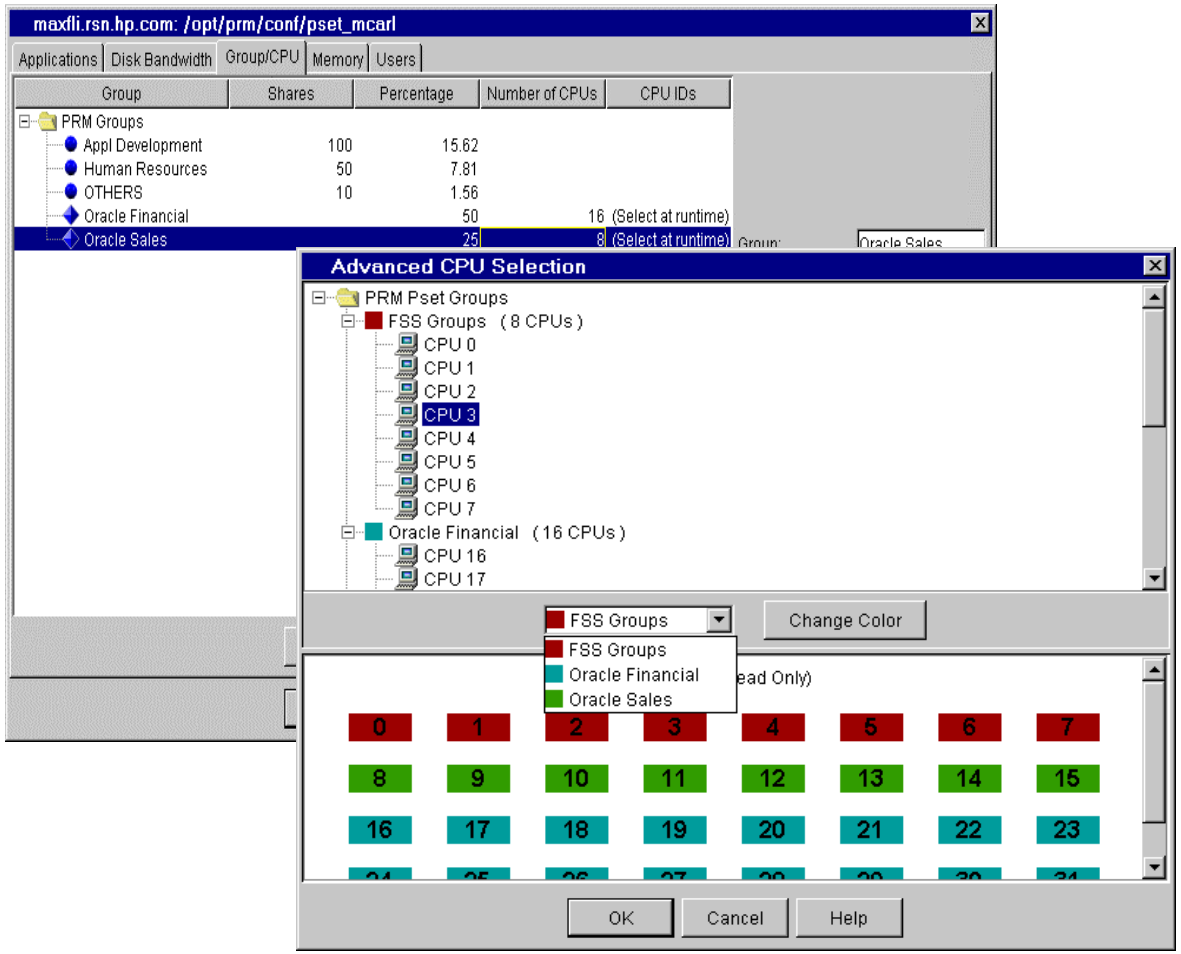
HP-UX PRM

- **CPU allocation**
  - Fair share scheduler
  - Processor sets
  - Percentage or shares
  - Optional capping
- **Real memory allocation**
  - Shares
  - Min and max
- **I/O allocation**
  - Min and max disk bandwidth
- **Entitlement based**
  - Prioritized
  - May be time dependant
- **Accounting**
  - Prmanalyze
    - summary, time-based, conflict



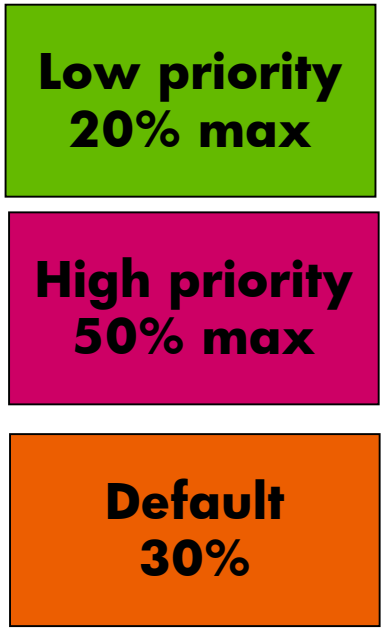
HP-UX PRM Management

- Configuration file /etc/prmconf
- prmconfig
- PRM command line
  - prmloadconf
  - prmanalyze
  - prmavail
  - prmconfig,
  - prmlist,
  - prmmonitor,
  - prmmove,
  - prmrecover,
  - prmrun



Tru64 UNIX Class Scheduler

- Provides a way of prioritizing tasks
- Allocate % CPU time to tasks to ensure
  - Menial tasks CPU limited
  - Critical tasks complete
- Percentage can be modified dynamically
- Group tasks into a class
- Works with processor sets
- Configuration DB /etc/class
- Easy management
  - SysMan (GUI)
  - class_admin (CLI)



Tru64 UNIX Class Scheduler Example

class

class> create highusers 50

highusers created at 50% cpu usage.

class> create lowusers 20

lowusers created at 20% cpu usage.

class> add highusers uid 203

uid 203 added to highusers.

runclass highusers myprog

class> enable

class scheduling enabled and daemon /usr/sbin/class_daemon started.

class> stats

current database: /etc/class/part.default

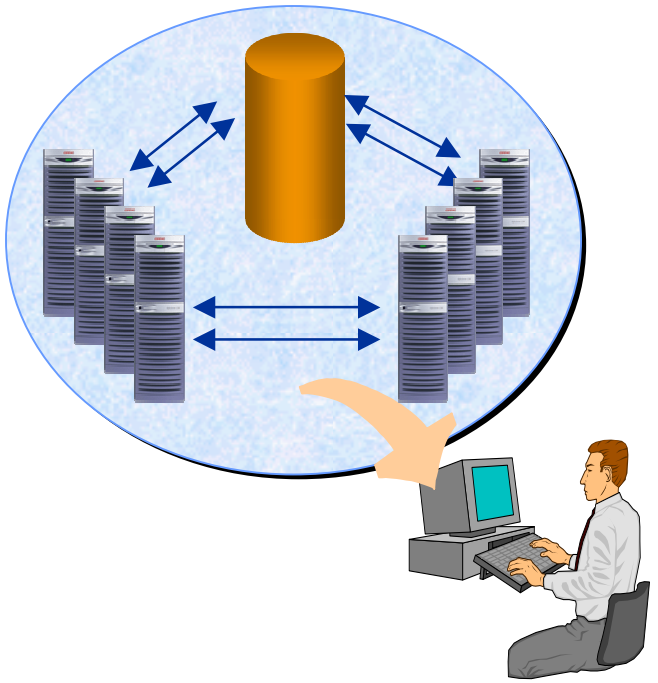
Class scheduler status: enabled

class name	target %	actual %
default	30%	0.0%
highusers	50%	0.0%
lowusers	20%	0.0%

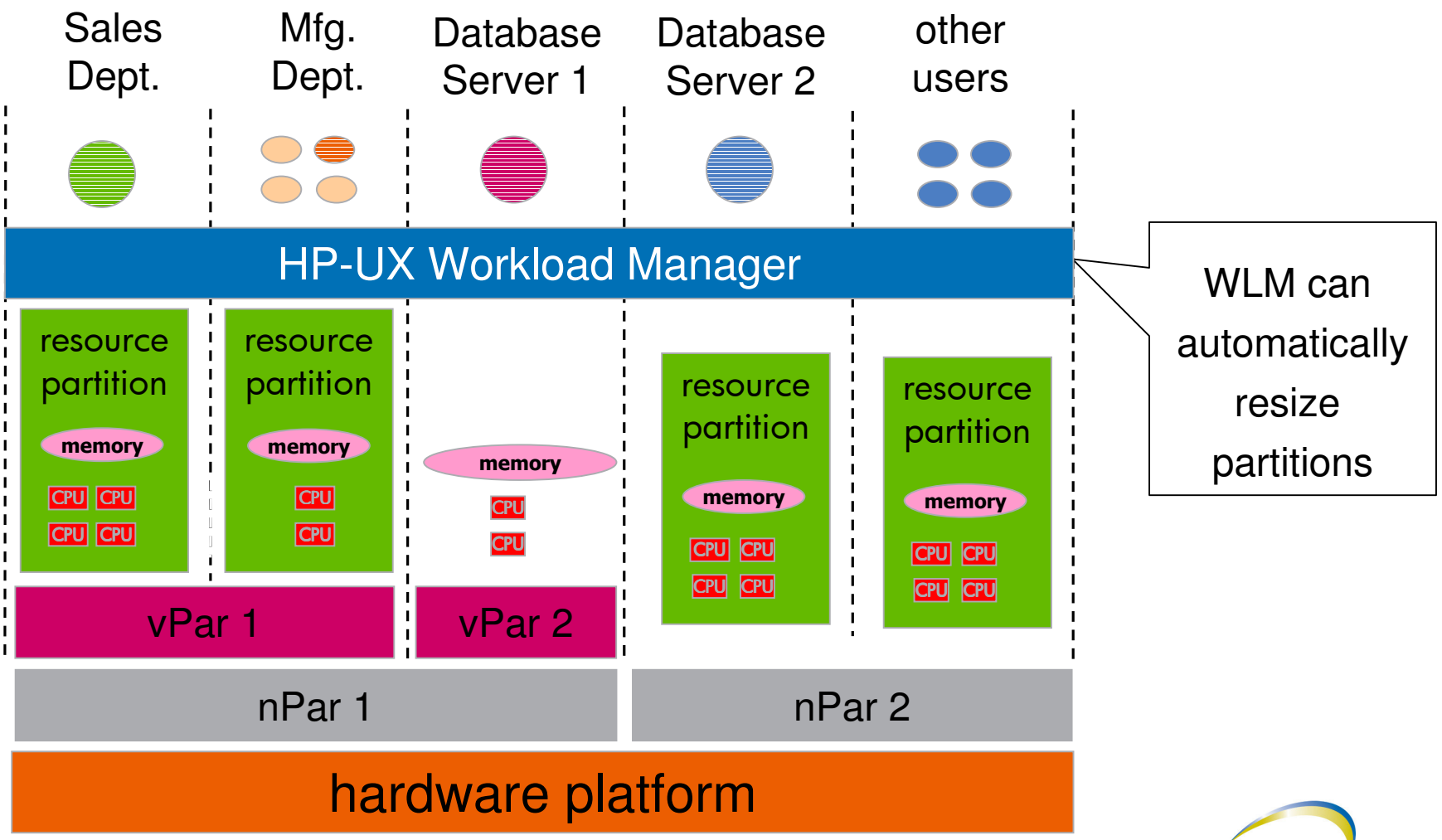
Clusters and Resource Management Tru64 TruCluster Example

Manual and scheduled load balancing enhancements

- **'Fail-back' manual override**
  - A mechanism to delay application fail-back to original member until a convenient time
- **Automated application workload balancing across the cluster**
  - Achieved by re-evaluating placement of applications according to CAA placement policy
- **Manual balance cluster workload**
- **Application uptime reporting**
  - Provides application availability history

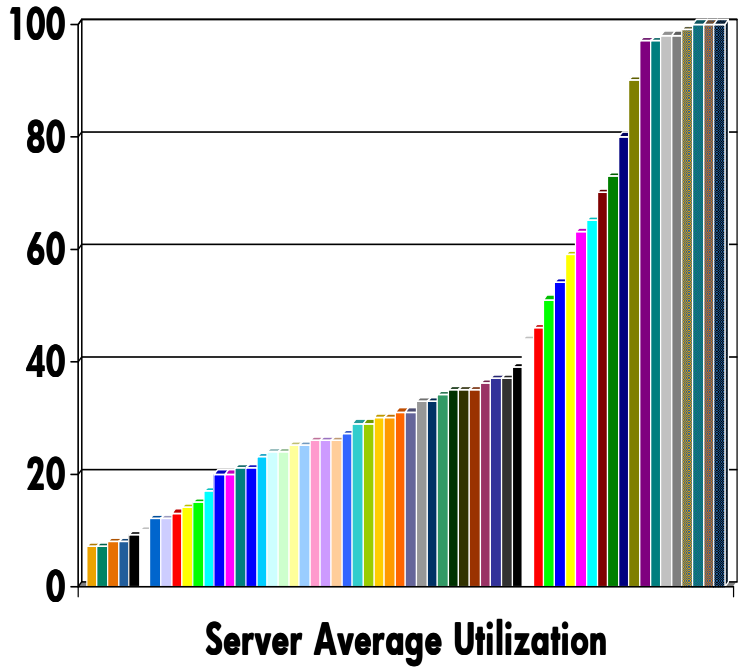


HP-UX Resource Management Tools

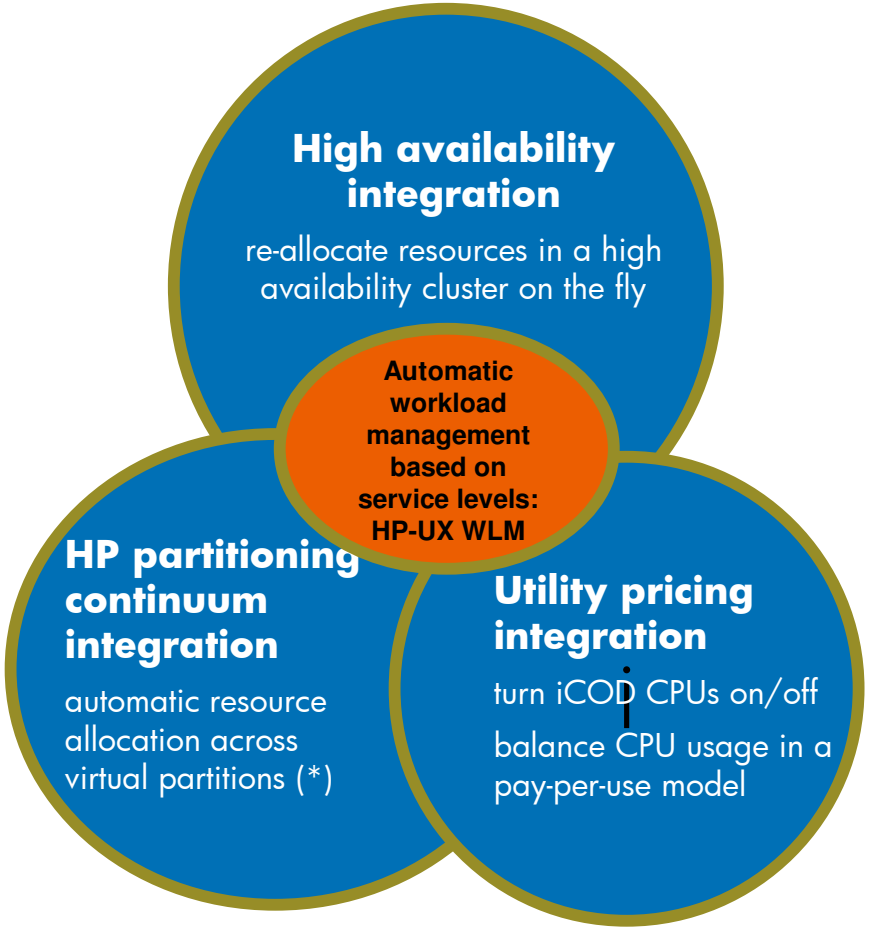


HP-UX Workload Manager (WLM)

- WLM is not a partitioning technology
- WLM is an intelligent policy engine to manage peak demands for critical applications
- WLM provides automatic movement of CPU resources to workloads that need them to meet SLOs



HP-UX WorkLoad Manager (WLM)



- **Goal-based resource management**
  - automatic allocation of resources based on set service level objectives both within an OS image and across vPars
- **Provides predictable response times for mission-critical applications**
- **Integration with other tools**
  - Virtual partitions
  - Serviceguard
  - iCOD
- **Out-of-the box toolkits**
  - Oracle, Apache, SAS, WebLogic…
- **PRM included**

HP-UX WLM Goal Types

- Direct measurement of the performance of the workload
  - Response Time
  - Throughput
- Measurement of load on application
  - Number of users/processes
  - Queue length
- Resource Usage
  - CPU entitlement based on utilization of current entitlement

HP-UX WLM
Service Level Objectives

application a	application b	application c
Response time SLO	**Job duration SLO**	**CPU entitlement SLO**
- Transactions to complete in < 1 second - Performance data based on application response measurements	- Batch job to finish in < 1 hour -Performance data from existing kernel data - Condition: 1st of month	-50% of CPU allocation - No instrumentation required
priority 2	***priority 3***	***priority 1***
50% Real Memory 50% Disk I/O	25% Real Memory 25% Disk I/O	25% Real Memory 25% Disk I/O

WLM automatically reconfigures CPU entitlements to satisfy SLOs in priority order

HP-UX WLM Serviceguard and iCOD Integration

Serviceguard

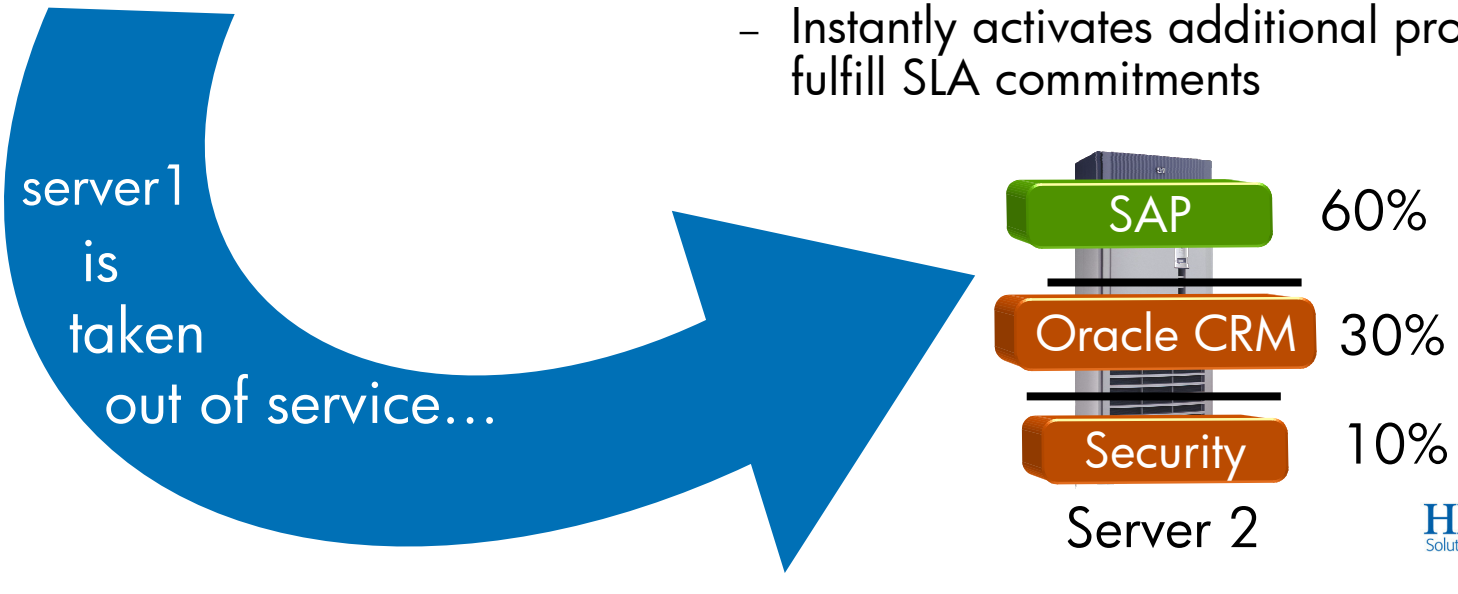
- Analyzes new infrastructure configuration
- Automatically moves package(s)

Workload Manager

- Automatically aware of new package
- Re-prioritizes packages and resources to maintain SLA goals and commitments

iCOD

- Instantly activates additional processors to fulfill SLA commitments



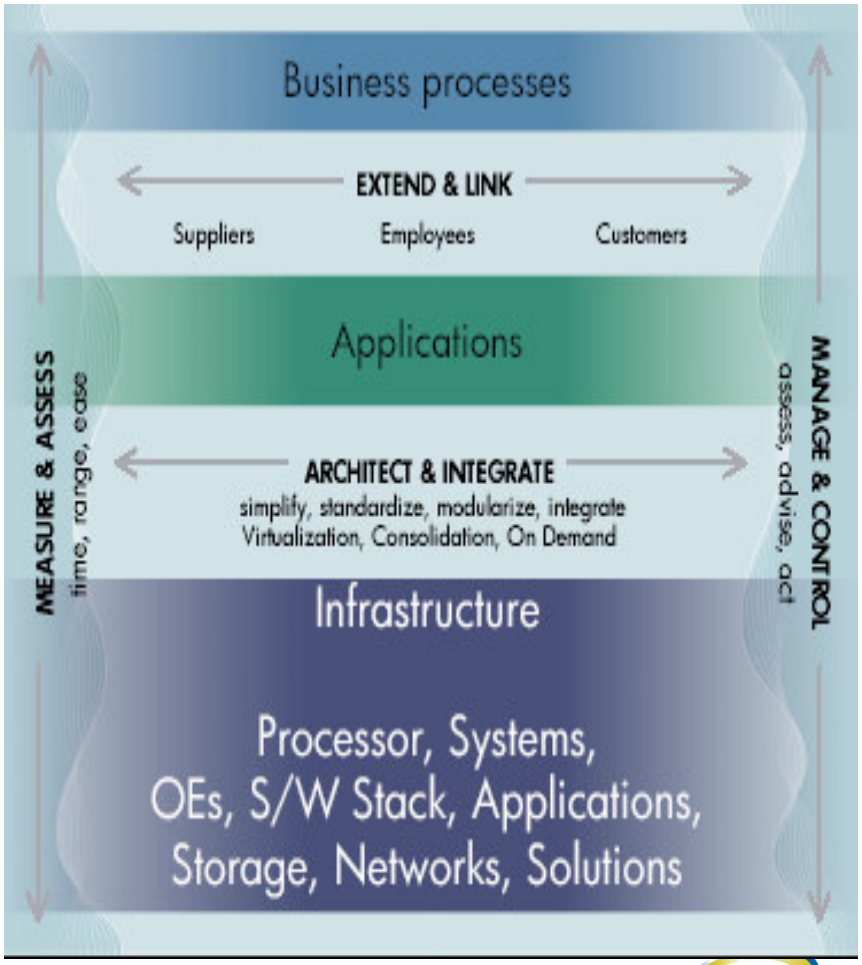
Future Direction and Resources …

The next big thing…

Adaptive Enterprise

= Business and IT Synchronized to Capitalize on Change

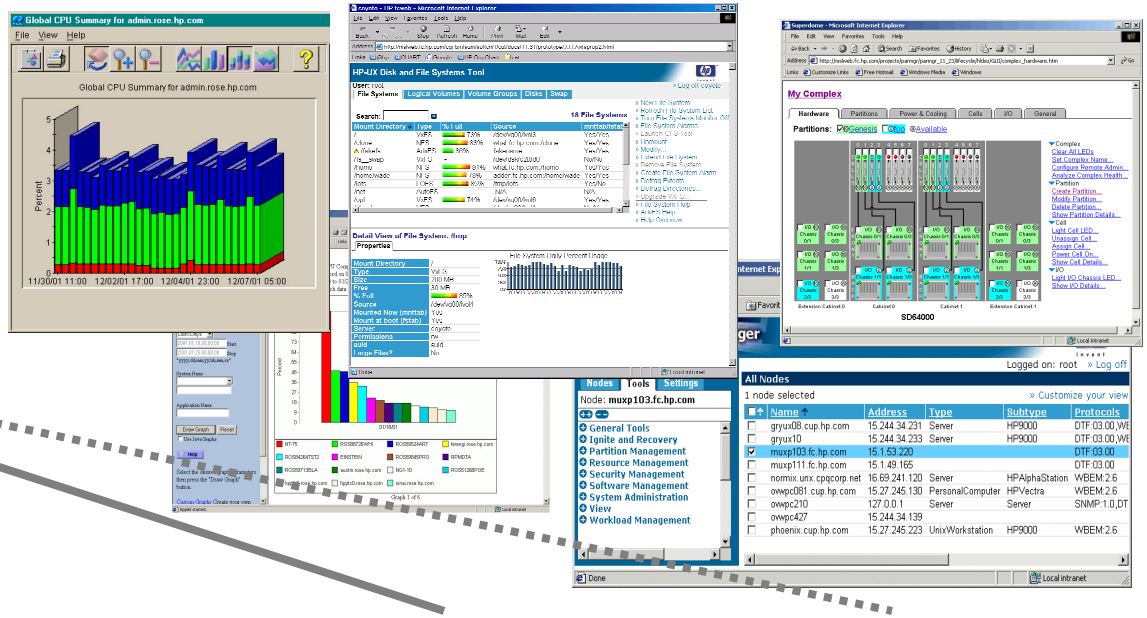
+ Standards Based Computing
+ Multi-OS
+ Flexible, Adaptable, Reliable
+ Management Simplicity
+ Dynamic Allocation of Virtualized Resources to Changing Work Loads
+ Utility Pay-per-use Business Model



Adaptive Management means for servers…

Computing resources move to where they are needed

VISUALIZE
Display system status, hardware, I/O, cluster, partition



CONFIGURE
System administrator sets-up compute environment as needed

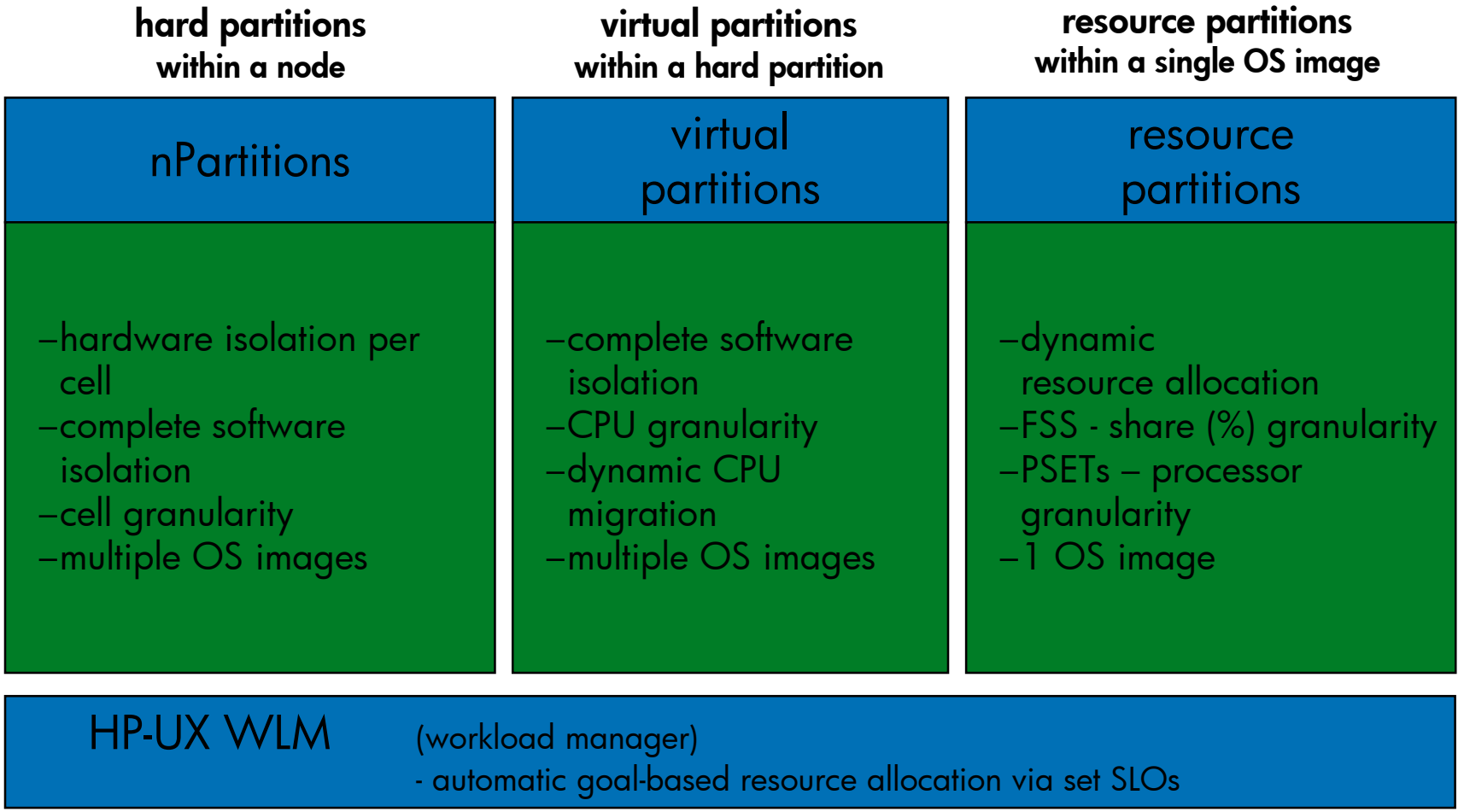
MONITOR
System warns administrator when something either goes wrong or is about to go wrong

ADVISE
System makes recommendations for fixing or optimizing compute environment

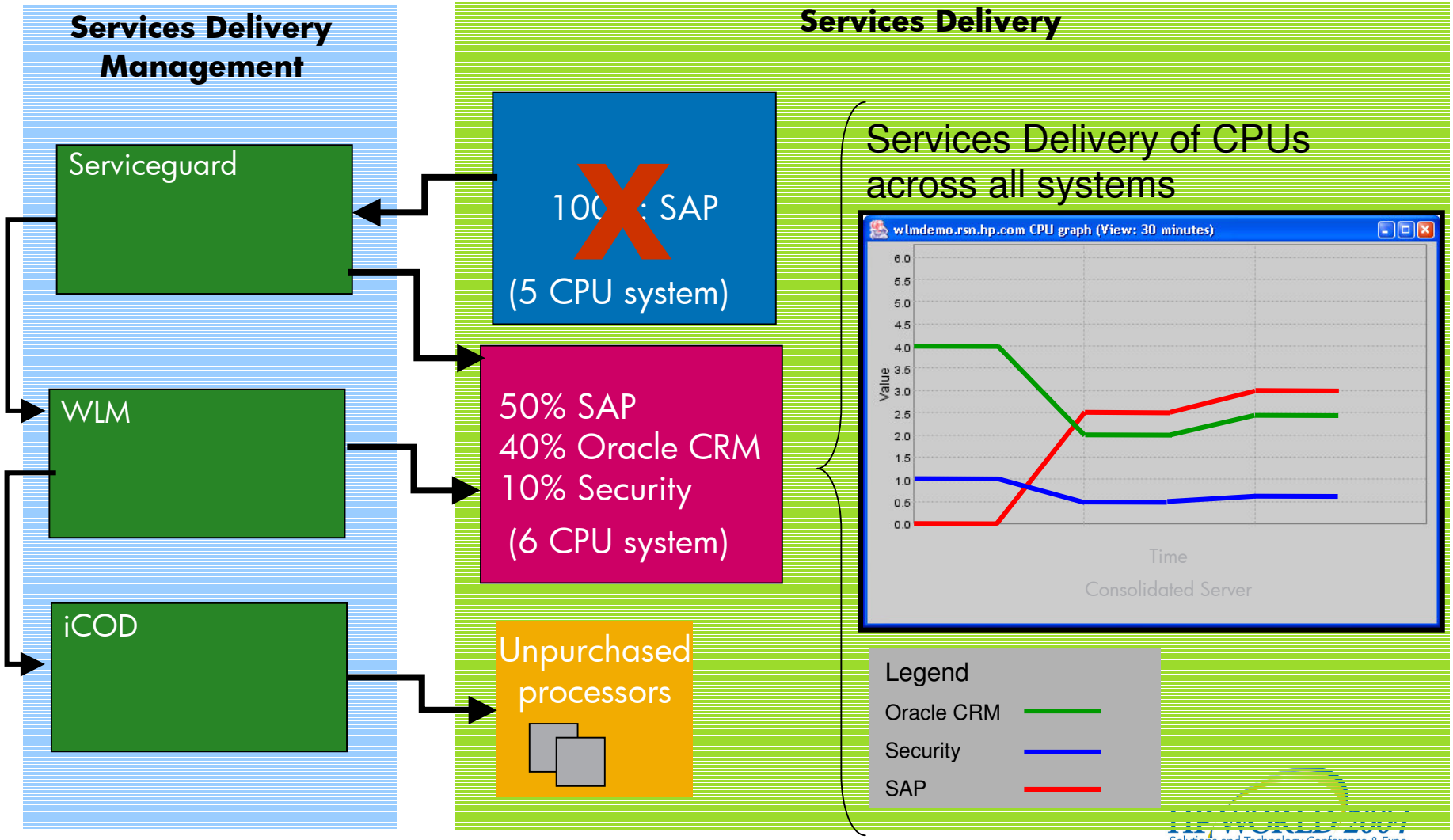
ADAPT
System automatically re-configures itself to fix or avoid problems

Increasing Automation

Adaptive Infrastructure on HP-UX
HP's Partitioning Continuum



HP-UX VSE for the Adaptive Enterprise – in production use today



Summary
Benefits/Strengths

Benefit	nPars	vPars	prm/psets	prm/fss
Maximize system utilization	Good	Better	Better	Best
Resource isolation	Best	Better	Better	Good
Os isolation	Best	Better	No	No
HP-UX (PA-Risc)	Yes	Yes	Yes	Yes
Tru64 UNIX	Yes	No	Yes	No
HP-UX (IA64)	Yes	soon	Yes	Yes
Ease of management/TCO	Good	Better	Best	Best
Ease of setup	Good	Better	Best	Best
iCOD support	Yes	Yes	Yes	Yes
WLM support	Yes	Yes	Yes	Yes

More information 1/2

- **Web sites**

HP-UX
- www.hp.com/go/superdome
- www.hp.com/go/HP-UX
- www.hp.com/go/wlm
- www.hp.com/go/prm
- www.hp.com/go/partitions
- www.hp.com/go/vse
- Norman Lindsey's white paper on ccNUMA and HP-UX www.hp.com/products1/unix/operating/infolibrary/whitepapers/WP_ccnuma_final.pdf

Itanium
- Multi OS info

www.docs.hp.com/hpux/onlinedocs/4442/Multi-OS_FAQ.pdf
www.docs.hp.com/hpux/onlinedocs/4452/Multi-OSInstallation.pdf
www.docs.hp.com/hpux/onlinedocs/4435/Multi-OS-Compatibility.pdf

- Firmware http://developer.intel.com/technology/efi/efi.htm

Tru64 UNIX
- www.hp.com/go/tru64unix
- www.tru64unix.compaq.com/workload-mgmt.html
- class scheduler, pset, best practices documentation www.tru64unix.compaq.com/docs
- application transition tools Tru64 to hp-ux www.hp.com/go/tru64appmigration/
- *Porting from Tru64 UNIX® to HP-UX* www.hp.com/transition/apps/porting_guide.html
- Alpha/Tru64 to Itanium/HP-UX whitepaper www.hp.com/go/alpha-retaintrust

More information 2/2

Related Training:

- **Partitioning/resource management classes available:**
  - HP-UX partition management with vPars and nPars @: http://www.hp.com/education/courses/u5075s.html
  - HP-UX resource management with PRM & WLM @: http://www.hp.com/education/courses/u5447s.html
- **Technical Webcasts Tru64 / HP-UX Differences**
  www.hpbroadband.com keyword tru64unix
  - Tru64 UNIX® & HP-UX: Side-by Side Comparison for System Administrators
  - Tru64 UNIX®: Side-by-Side Comparison: Clusters & Disaster Tolerance
  - Tru64 UNIX® & HP-UX: Side-by-Side Comparison: Storage Platform Configurations
  - Tru64 UNIX® & HP-UX: Side-by-Side Comparison: Resource and Workload Management
  - Transitioning your Applications from Tru64 UNIX® to hp-ux on Itanium®: Application Transition Tools
  - Tru64 UNIX® & HP-UX: Planning a Technology Transition
  - Tru64 UNIX® & HP-UX: Enterprise Server Evolution
  - Planning your Database Migrations to Integrity
- **Web based 24 x 7 (free)**
  - Tru64 UNIX to HP-UX System Administration www.hp.com/go/alpha-retaintrust

HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:

interex

shared knowledge • shared power

encompass

AN HP USER GROUP

RECOMMENDED TRAINING VENUE FOR THE

HP Certified Professional