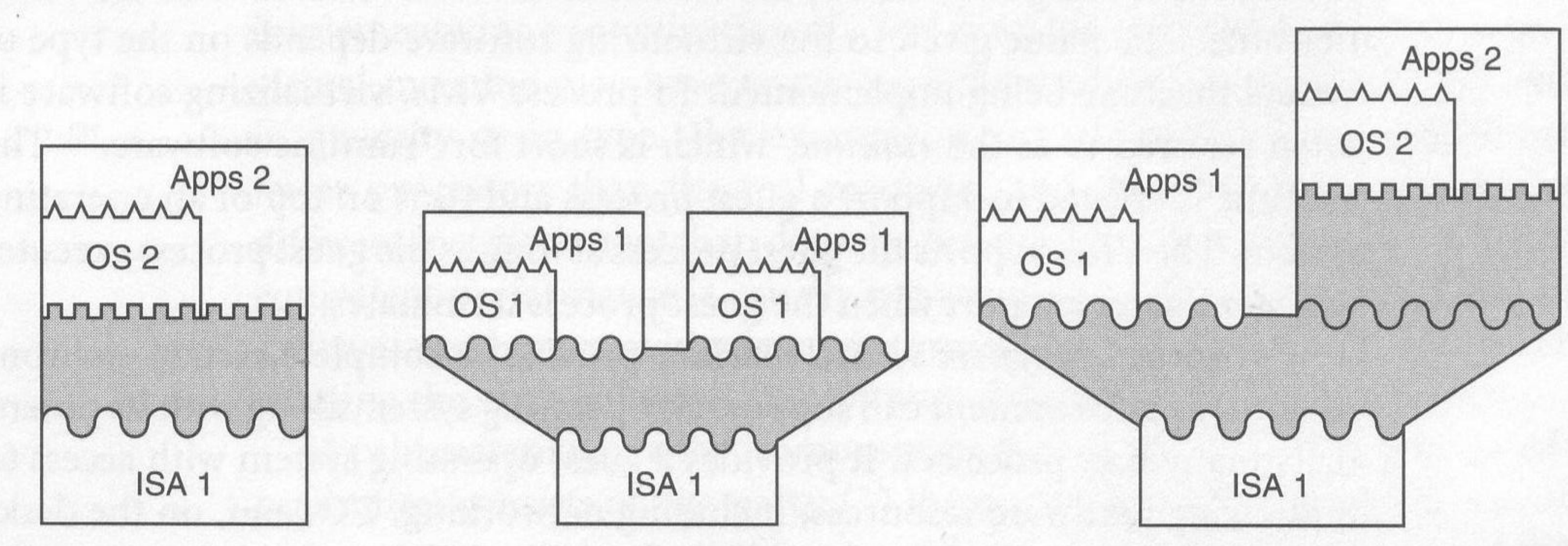
Server Virtualization Approaches

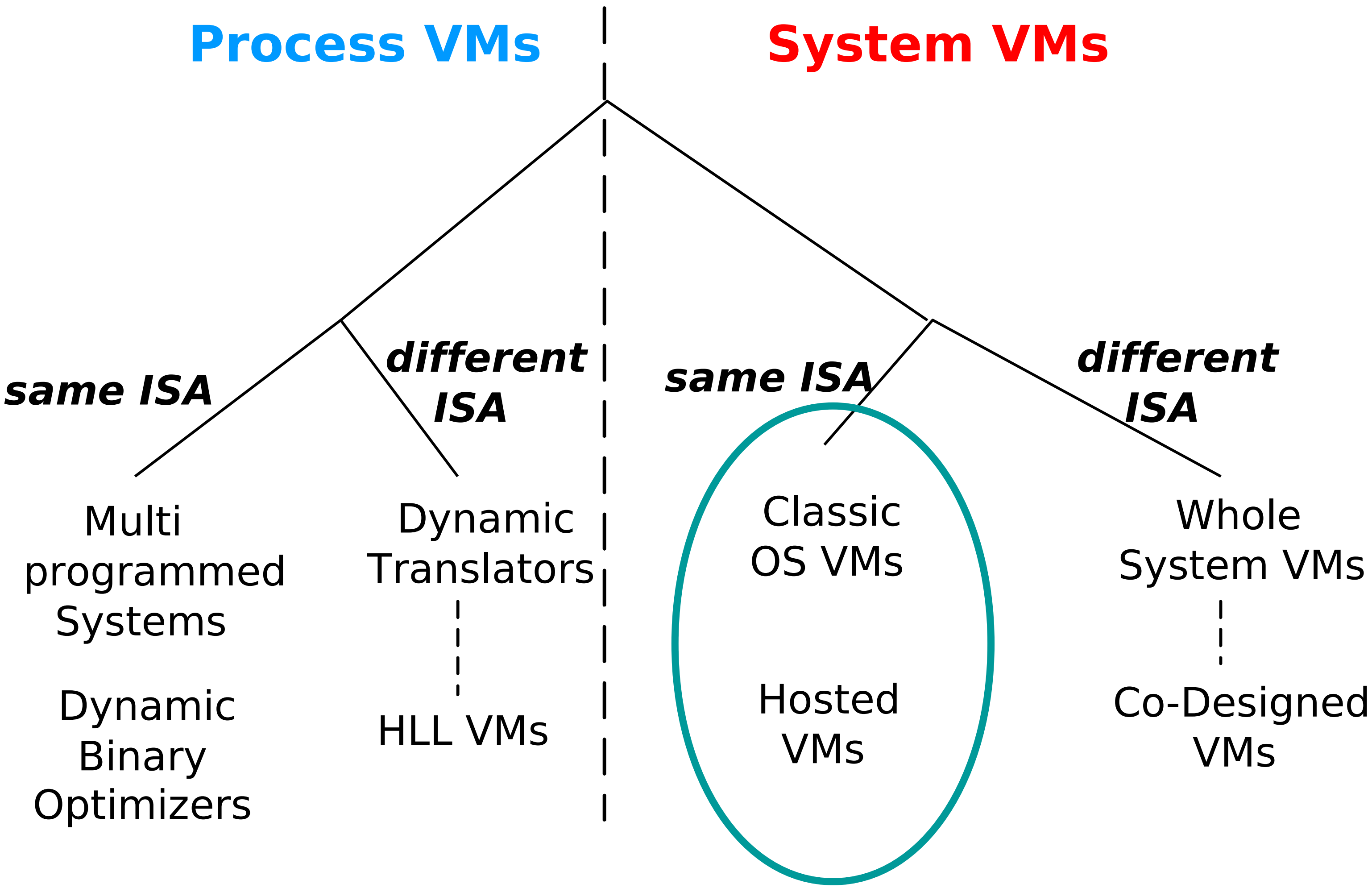
Virtual Machine Applications



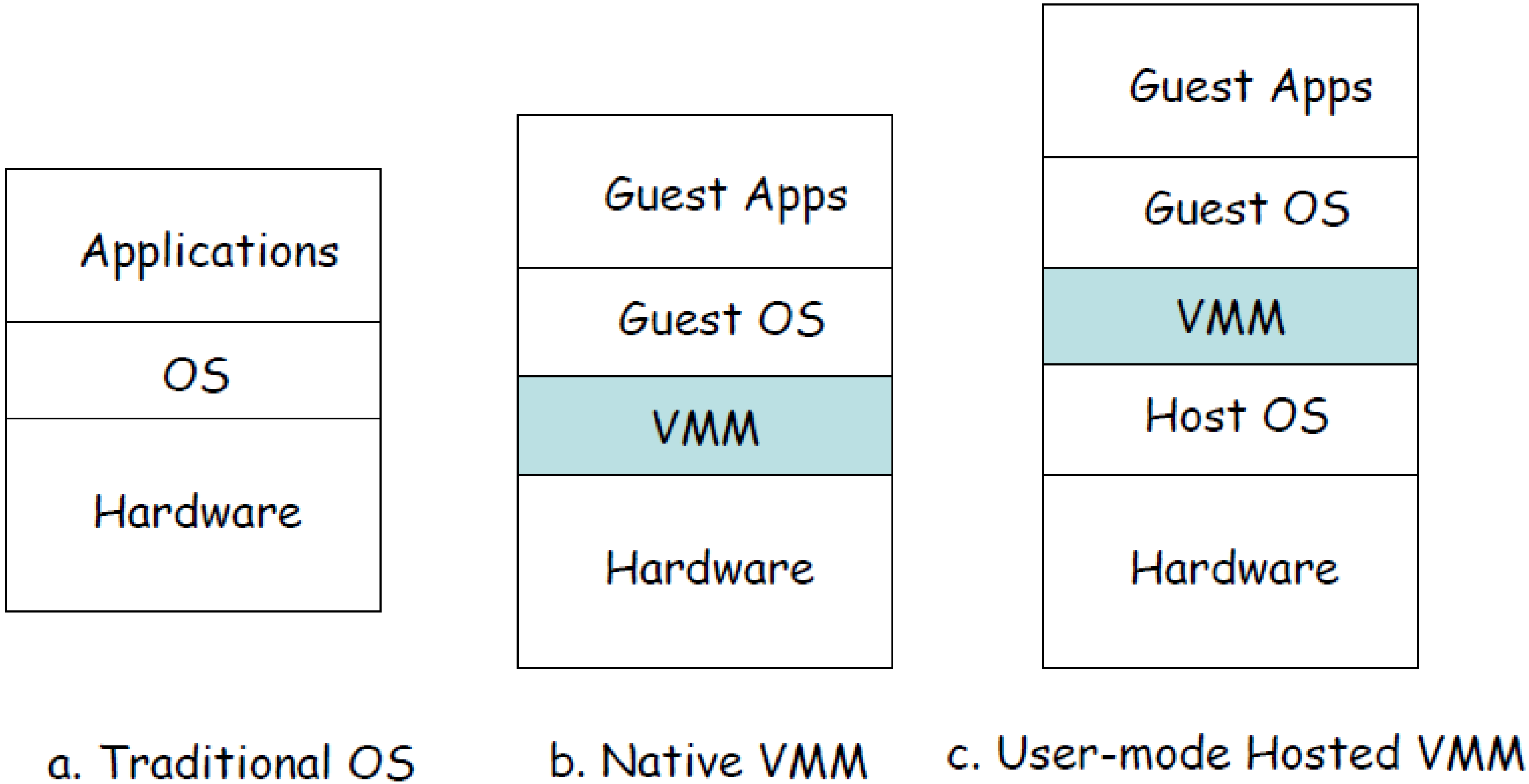
Emulation Replication Composition

- Emulation: Mix-and-match cross-platform portability
- Replication: Multiple VMs on single platform
- Composition: Form more complex flexible systems

Our focus: Same ISA System VMs

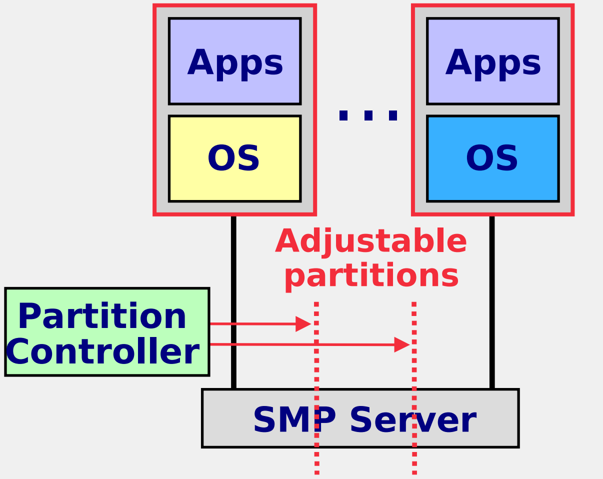


System VMs


Applications
OS
Hardware
Guest Apps
Guest OS
VMM
Hardware
Guest Apps
Guest OS
VMM
Host OS
Hardware
a. Traditional OS
b. Native VMM
c. User-mode Hosted VMM

Server Virtualization Approaches

Hardware Partitioning



Server is subdivided into fractions each of which can run an OS

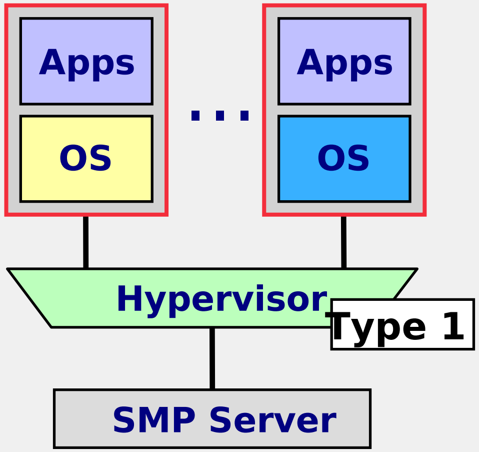
Physical partitioning

S/370 SI-to-PP and PP-to-SI, Sun Domains, HP nPartitions

Logical partitioning

pSeries LPAR, HP (PA) vPartitions

Bare Metal Hypervisor

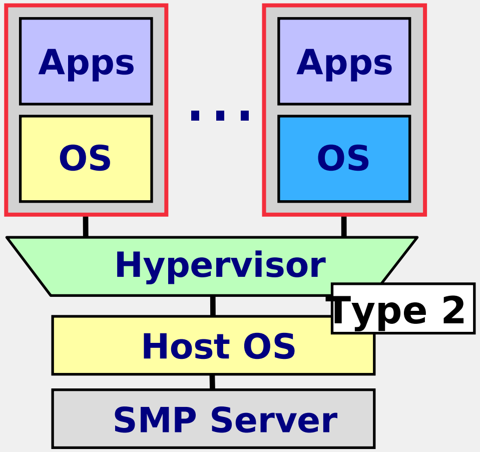


Hypervisor provides fine-grained timesharing of all resources

Hypervisor software/firmware runs directly on server

System z PR/SM and z/VM
POWER Hypervisor
VMware ESX Server
Xen Hypervisor

Hosted Hypervisor

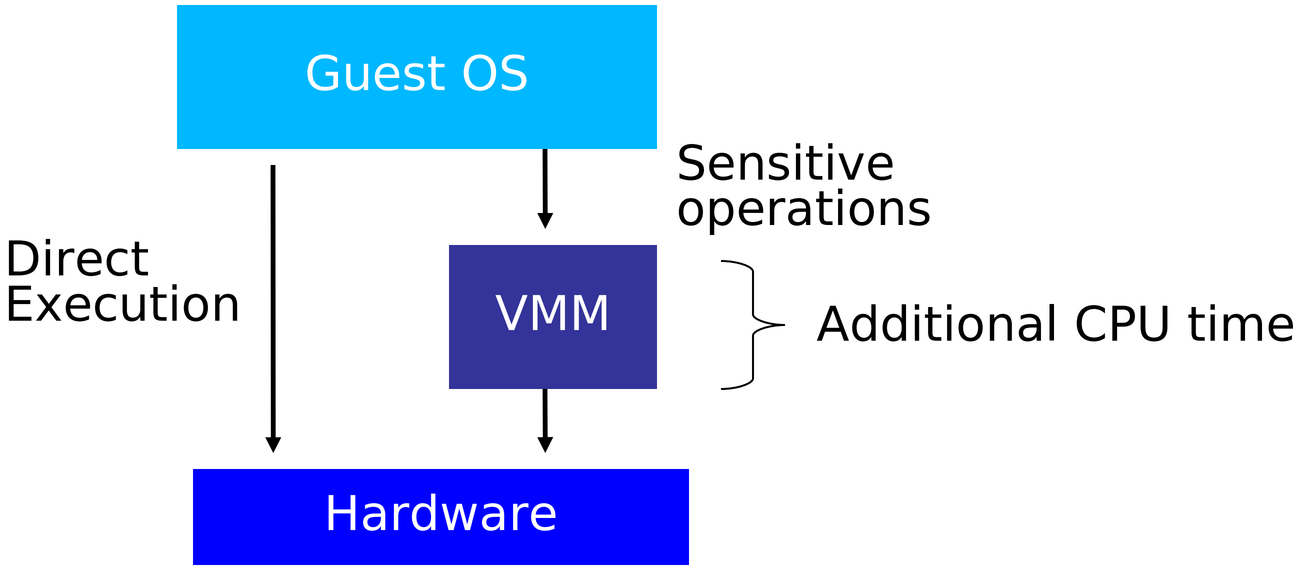


Hypervisor uses OS services to do timesharing of all resources

Hypervisor software runs on a host operating system

VMware Server
Microsoft Virtual Server
HP Integrity VM
User Mode Linux

Virtualization Overhead



- Time spent by the VMM
  - Increases host CPU utilization
  - Increases latency
  - However, throughput can be acceptable if there is enough CPU power

Virtualization Overhead

- Overhead is not fixed
  - Varies with workload
  - Varies with Hardware – newer hardware has lower instruction latency
  - Depends on how the application is tuned
  - Depends on the hypervisor
- Examples
  - SPEC CPU2006 - CPU intensive workload runs at near native speed.
  - Real life workloads have varying mix

Techniques

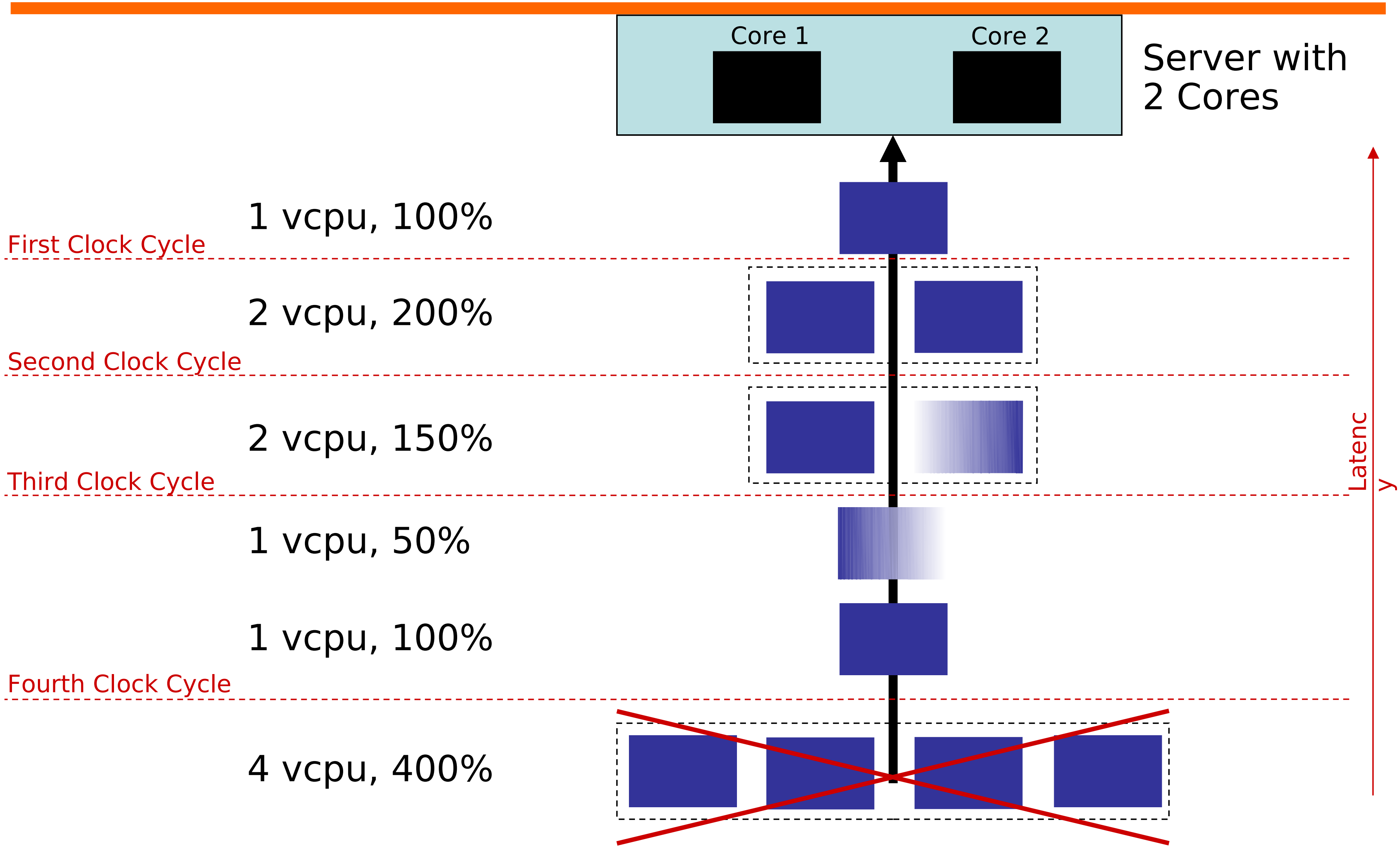
- Full virtualization:
  - Transparent Virtualization:
  - “Classical” x86 is ***not fully virtualizable*** – it has sensitive instructions that don't trap in user mode.
  - Some of VMM's overhead comes from trying to detect these instructions !

    - Advantages:
      - » Easy migration from physical to virtual systems (P2V).
    - Disadvantages:
      - » Performance penalty

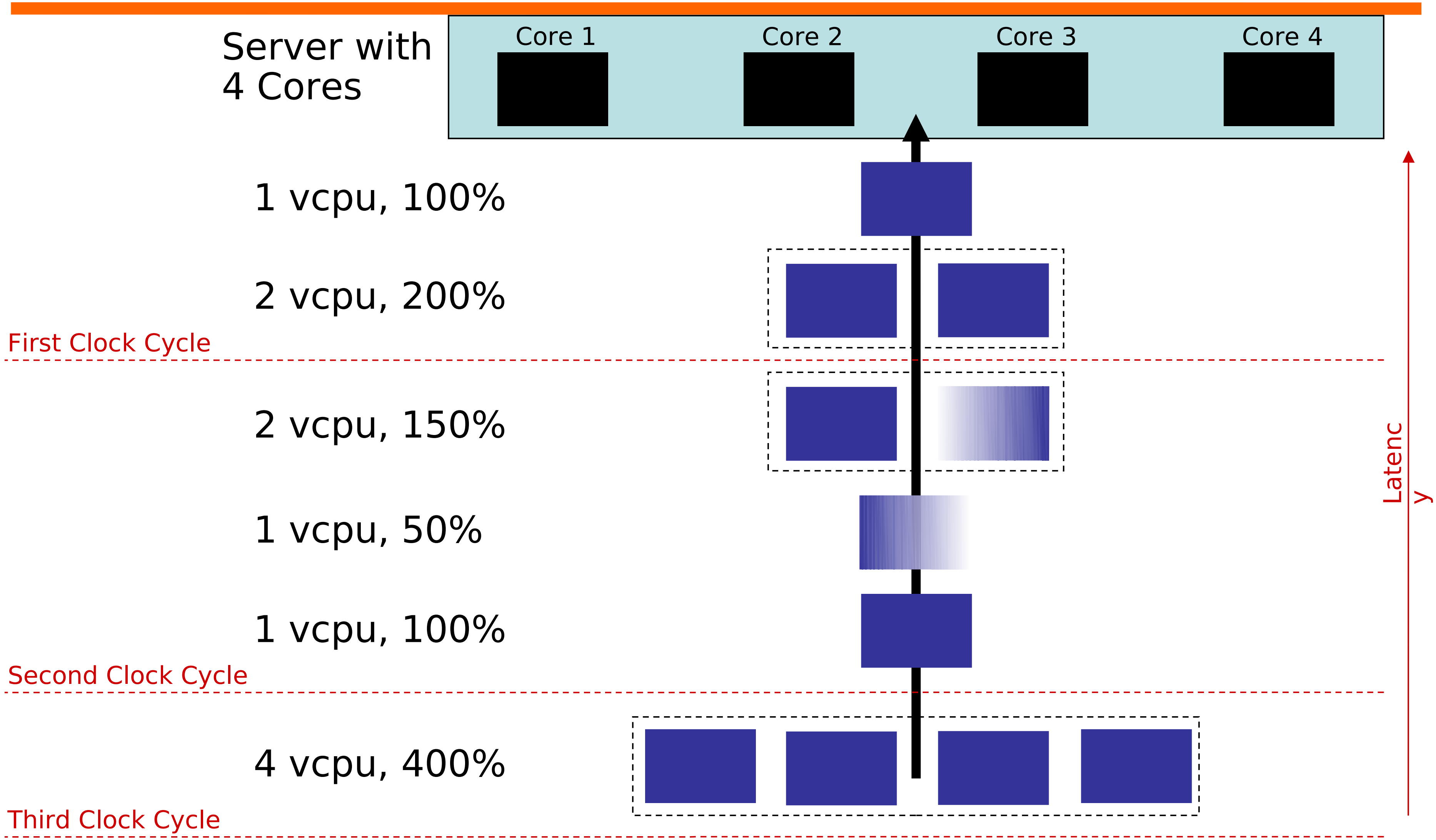
- Paravirtualization
  - Technique that presents a software interface to virtual machines that is similar but not identical to that of the underlying hardware.
  - The guest OSs are aware that they are executing on a VM.

    - Advantages:
      - » Lightweight and fast
    - Disadvantages:
      - » Requires porting Guest OS, to use ***hypercalls*** instead of ***sensitive instructions***.

VMM Scheduler



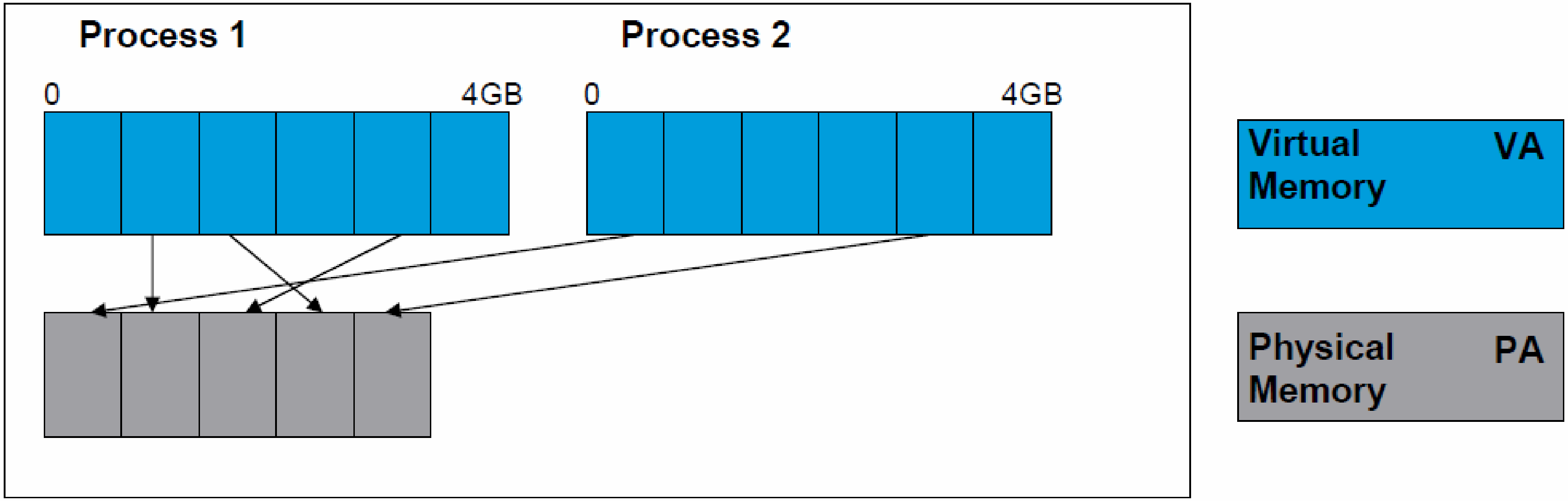
VMM Scheduler



Memory Virtualization

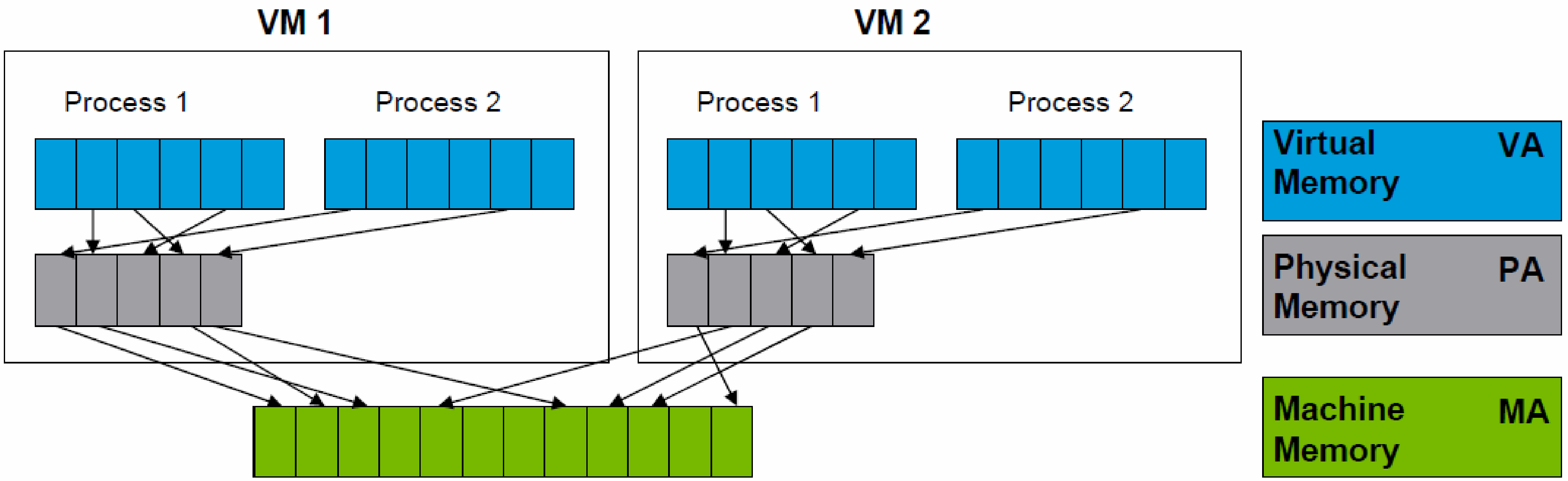
- Virtual Memory in a VM
  - Each guest OS maintains own set of page tables.
  - Guest OS translates virtual memory locations to real memory locations (“physical memory” of VM.)
  - Guest OS has swap space on virtual disk.
- VMM
  - Translates real memory to physical memory using MMU.
  - VMM *may have* a swap space on physical disk.

Memory management without Virtualization



- **Applications see contiguous virtual address space, not physical memory**
- **OS defines VA -> PA mapping**
  - Usually at 4 KB granularity: a *page* at a time
  - Mappings are stored in page tables

Memory management with Virtualization



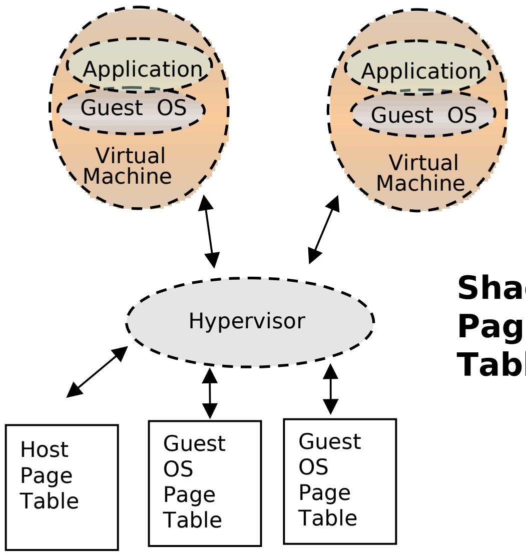
- **To run multiple VMs on a single system, another level of memory virtualization must be done**
  - Guest OS still controls virtual to physical mapping: VA -> PA
  - Guest OS has no direct access to machine memory (to enforce isolation)
- **VMM maps guest physical memory to actual machine memory: PA -> MA**

Nested Page Tables

Issue:
- VM page table updates must be handled by the hypervisor.
- If done in software, VM page table updates can be very expensive.

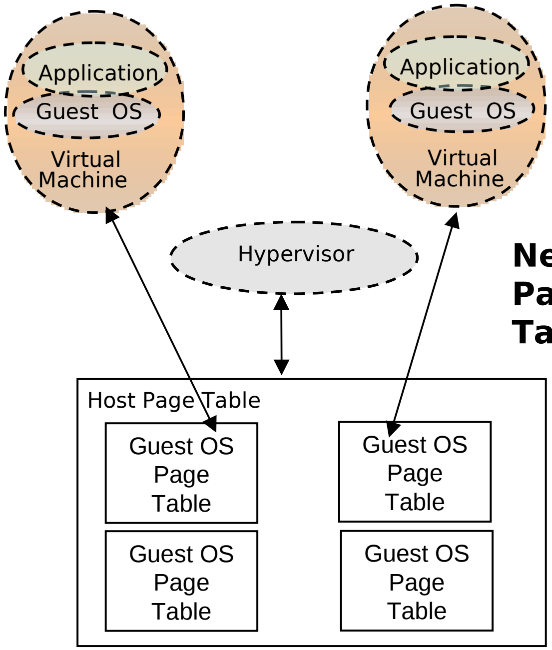
Solution:
- Nested Page Tables (NPT) provides hardware support for translating the VM's virtual page tables to physical page tables.
- The hypervisor configures the hardware so that it intercepts any attempt of a VM to update its own virtual page tables.
- When the hardware catches a write to the VM's page tables, the hypervisor determines the physical pages that will map the VM's virtual pages and sets the real hardware page tables to the physical mapping.
- NPT provides a significant performance boost for workloads that do a lot of paging, such as creating and destroying lots of processes.

Nested Paging



Shadow Page Tables

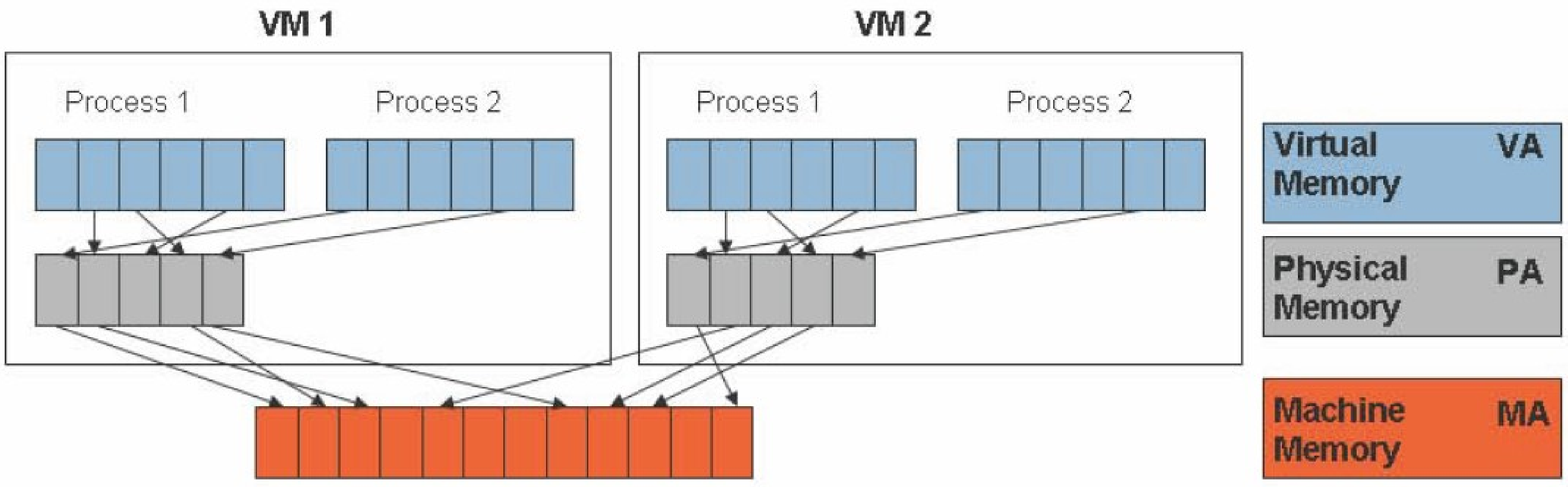
- Provides the guest OS with the illusion that it is managing memory
- Page tables are actually kept up by the hypervisor in software
- Requires more software intervention from the hypervisor



Nested Page Tables

- Each guest physically has their own world to manage
- Requires less intervention
- Memory look ups done in hardware which can be faster than software management

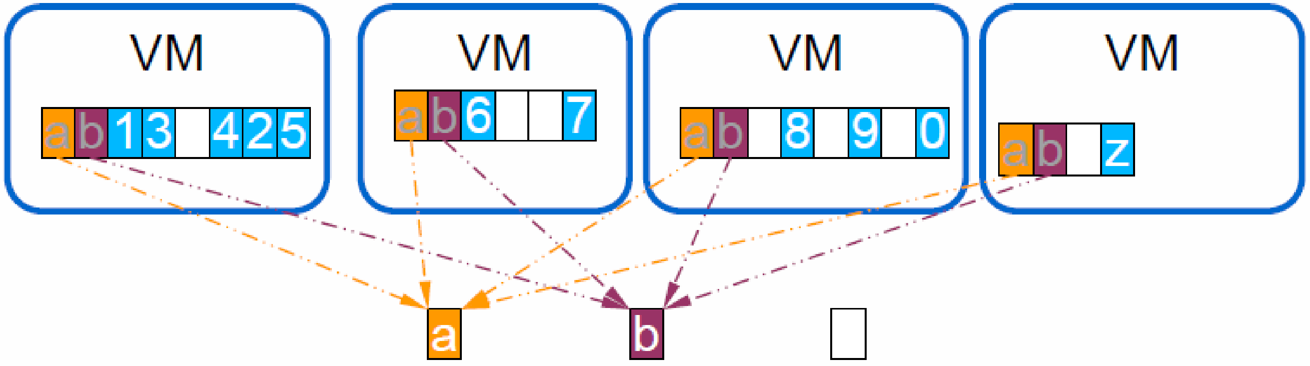
Nested Page Tables



- Hardware support for memory virtualization is on the way
  - AMD: Nested Paging / Nested Page Tables (NPT)
  - Intel: Extended Page Tables (EPT)
- Conceptually, NPT and EPT are identical
  - Two sets of page tables exist: VA -> PA and PA -> MA
  - Processor HW does page walk for both VA -> PA and PA -> MA

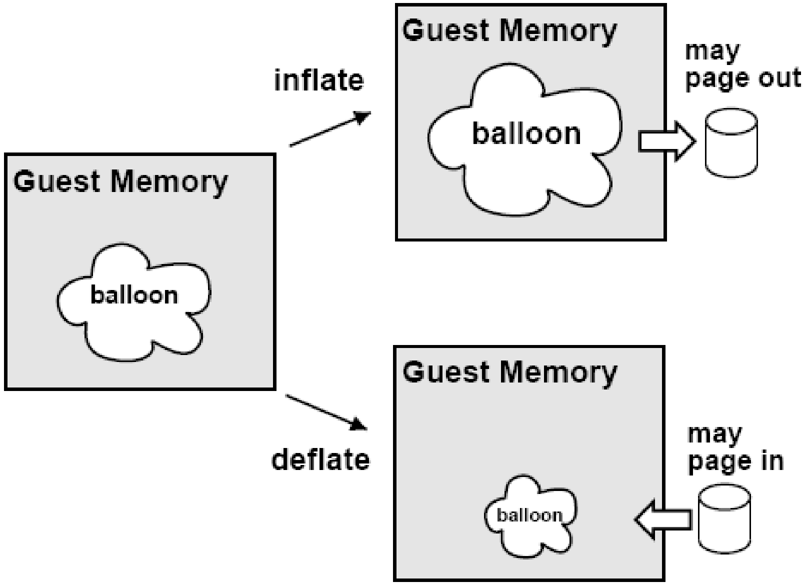
Advanced Topics: Memory Sharing

- **Motivation**
  - Multiple VMs running the same OS, applications, and libraries
  - Many redundant copies of code, data, zeros
- **Transparent page sharing**
  - Periodically scan memory
  - Map duplicate pages to a single machine page
  - Write-protect pages for safety, copy-on-write for correctness



Advanced Topics: Memory Ballooning

- **Inflating a balloon**
  - When the server wants to reclaim memory
  - Increase memory pressure in the guest OS, reclaim space to satisfy the driver allocation request
  - Driver communicates the physical page number for each allocated page to VMM
- **Deflating**
  - Frees up memory for general use within the guest OS



I/O Virtualization

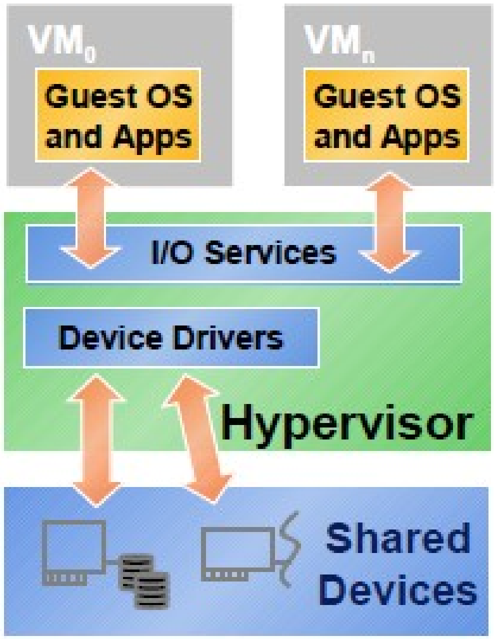
- A number of different types of I/O devices
- Construct a virtual version of the device
- I/O activity directed at the device is intercepted by VMM and converted to equivalent request for underlying physical device.

Device Types

- **Dedicated Device**
  - Display, keyboard, mouse etc.
  - VMM routes, but does not interpret the I/O instructions
- **Partitioned Devices**
  - E.g. A hard disk can host several virtual disks
- **Shared Devices**
  - E.g. network adapter
- **Nonexistent Physical Devices**
  - E.g. network adapter to communicate only among VMs

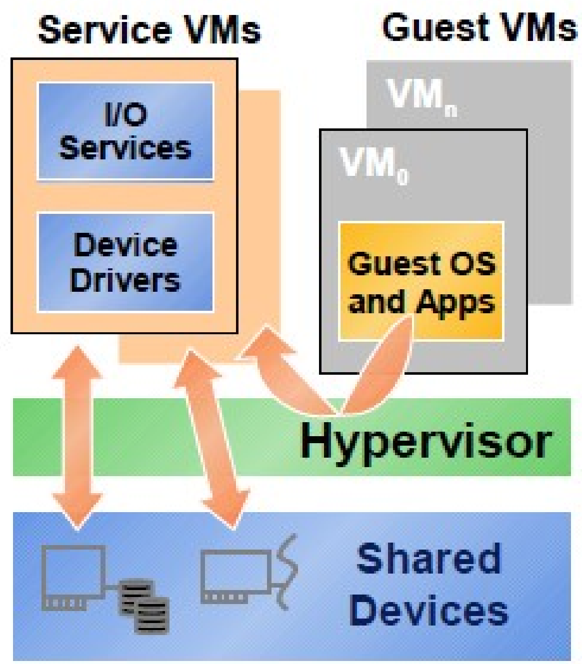
Options For I/O Virtualization

Monolithic Model



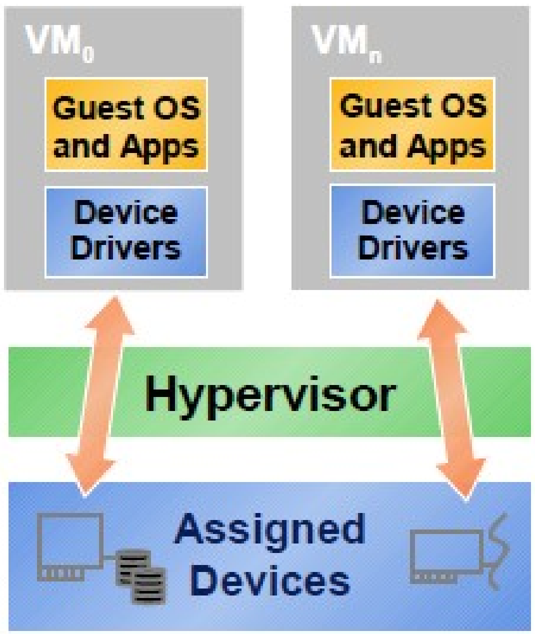
- Pro: Higher Performance
- Pro: I/O Device Sharing
- Pro: VM Migration
- Con: Larger Hypervisor

Service VM Model



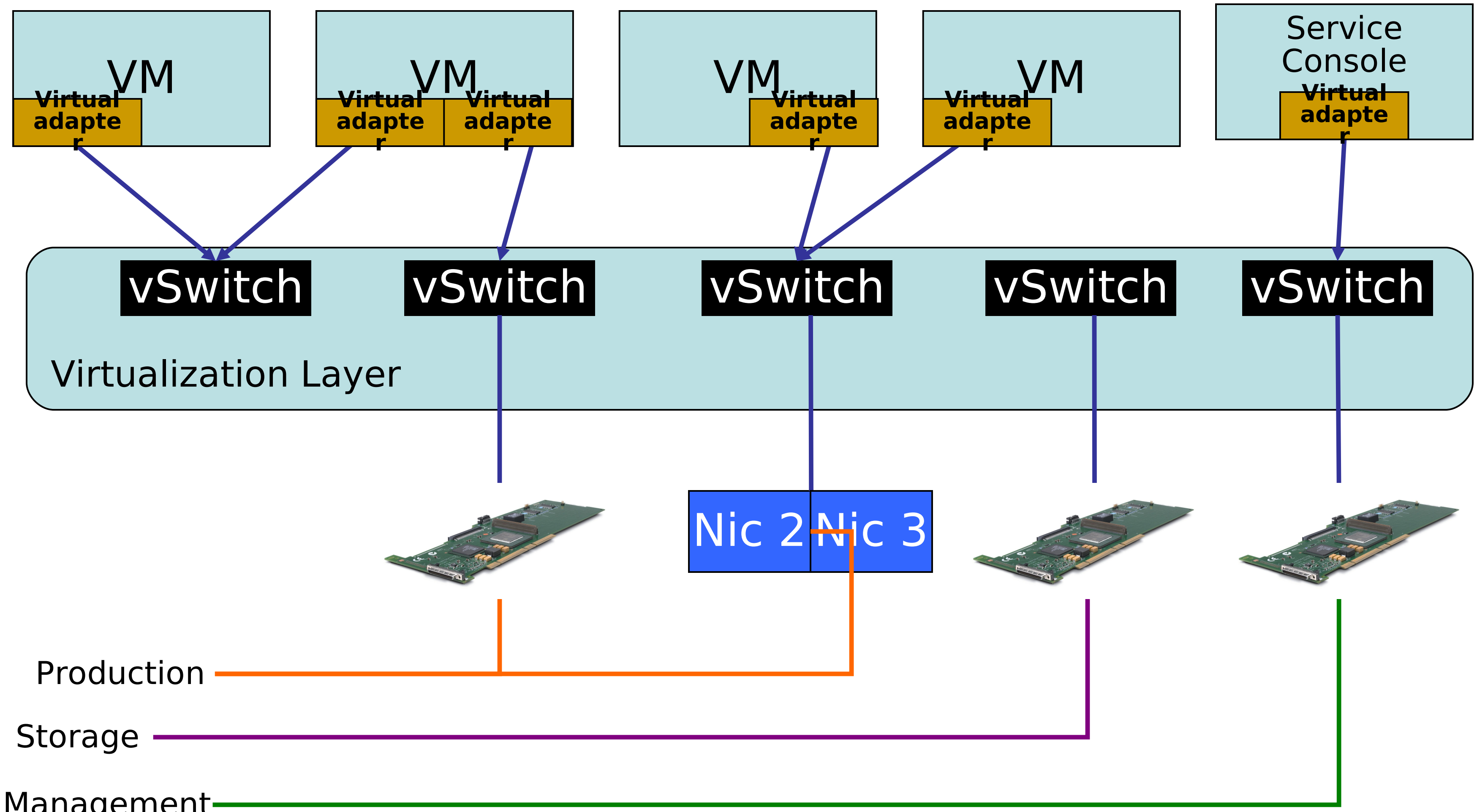
- Pro: Hypervisor independent from new I/O implementation
- Pro: High Security
- Pro: I/O Device Sharing
- Pro: VM Migration
- Con: Lower Performance

Pass-through Model

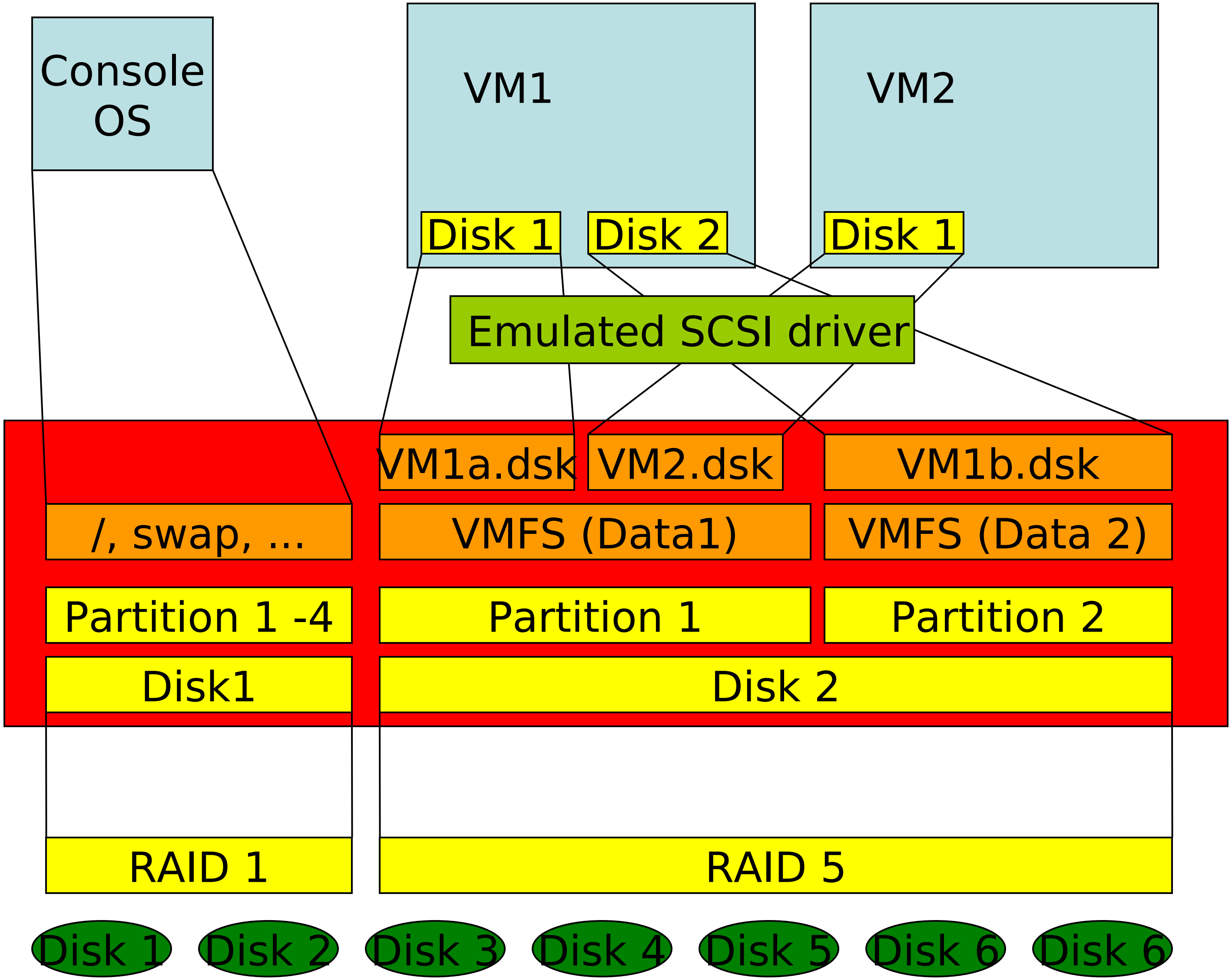


- Pro: Highest Performance
- Pro: Smaller Hypervisor
- Pro: Device assisted sharing
- Con: Migration Challenges

Network Architecture

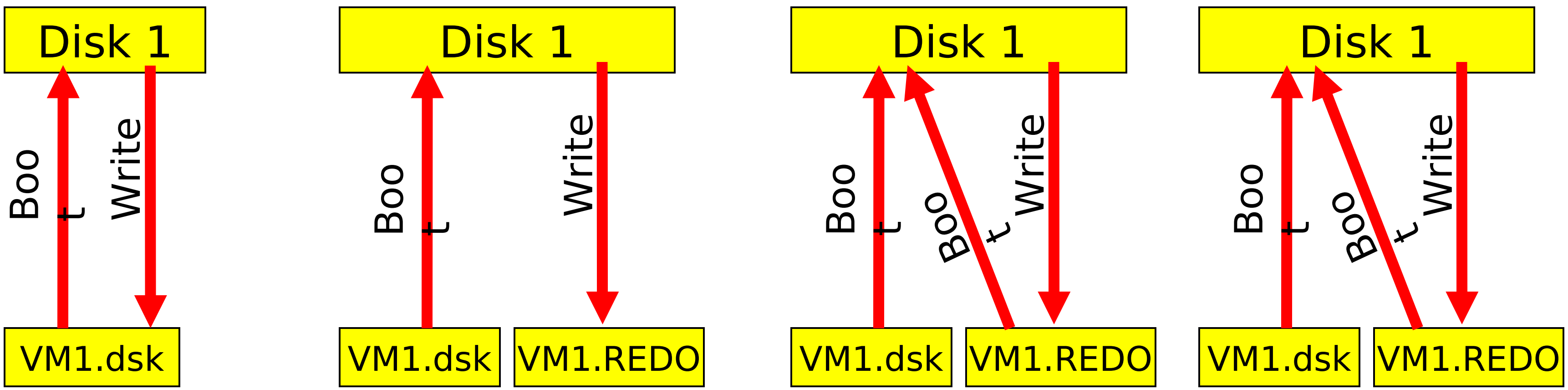


Virtualization - Storage



Disk modes

persistent | Non-persistent | undoable | append



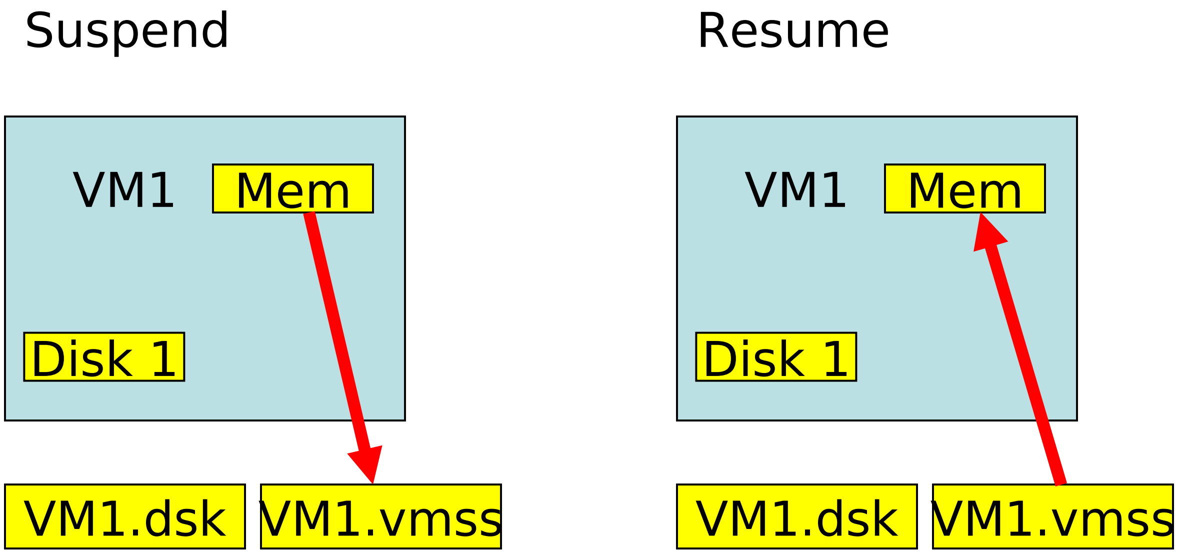
Normal

All changes are gone after reboot

It can be decided, if changes are kept or discarded

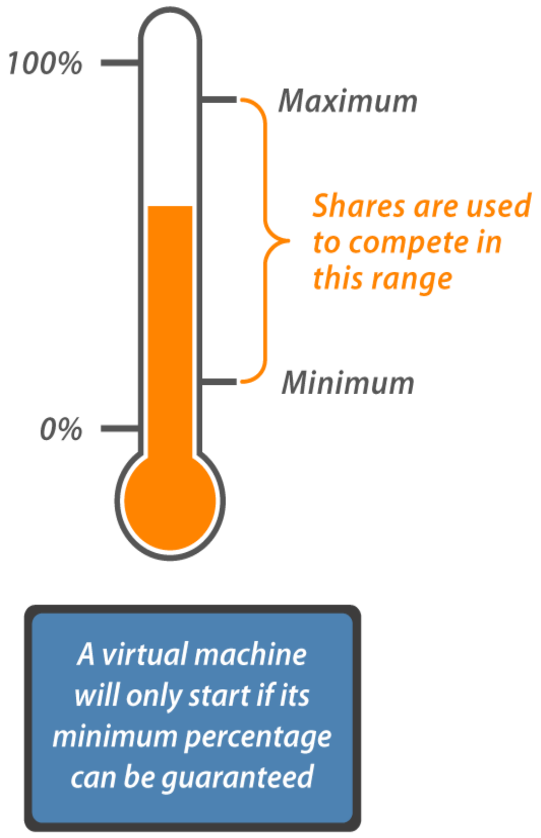
It can be decided several times, if changes are kept or discarded

Special storage states

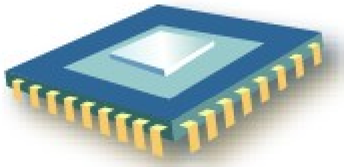


CPU Resource Settings: Percentages and Shares

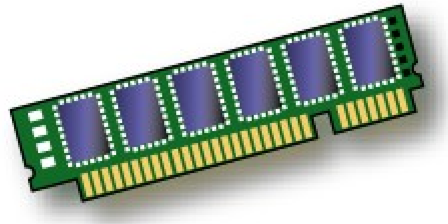
- **Minimum** absolute percentage
  - A percentage of a physical CPU reserved for this virtual machine
  - The VMM chooses which CPU
- **Maximum** absolute percentage
  - A cap on the consumption of CPU time by this virtual machine, as a proportion of a physical CPU
  - Range is 0-100% for uni virtual machines
  - Range is 0-200% for dual virtual machines
  - Because VCPUs in the same virtual machine are always co-scheduled
- **Proportional** shares (relative)
  - More shares means that this virtual machine will win competitions for CPU time



Server Performance Optimization

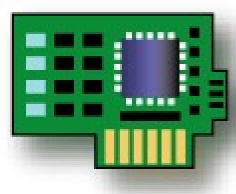


- Virtual SMP
- Minimum Rate
- Maximum Rate
- Share Allocation
- CPU Load Balancing
- Processor Affinity



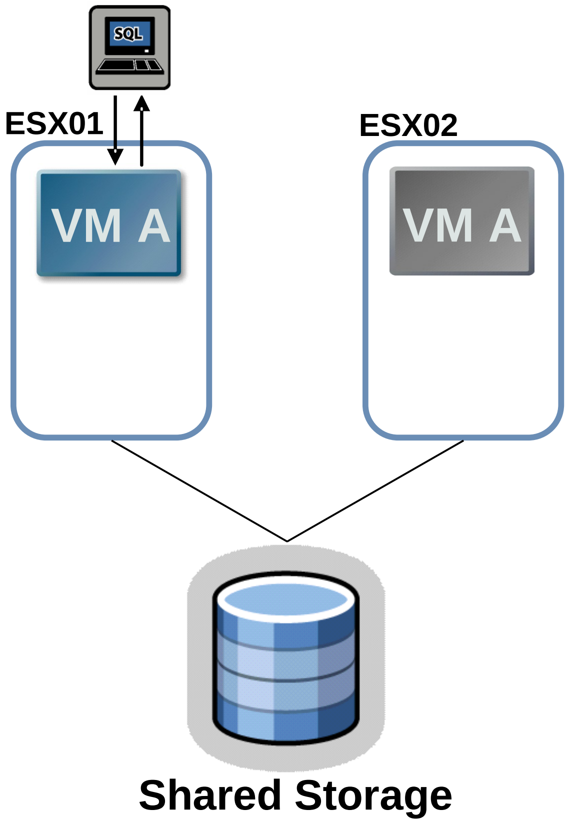
- Minimum Size
- Maximum Size
- Share Allocation
- Dynamic Allocation
- Advanced Memory Management

- Share Allocation



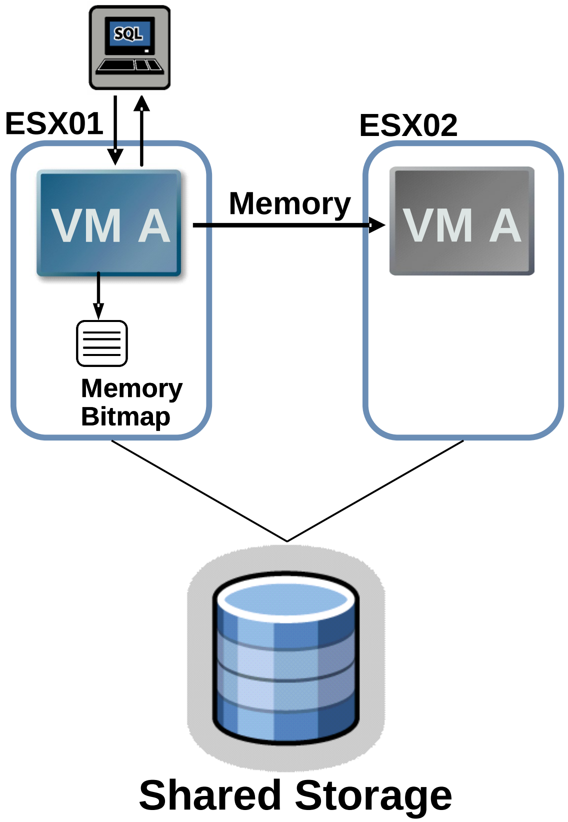
- NIC Teaming
- Traffic Shaping

VM Migration - Step 1



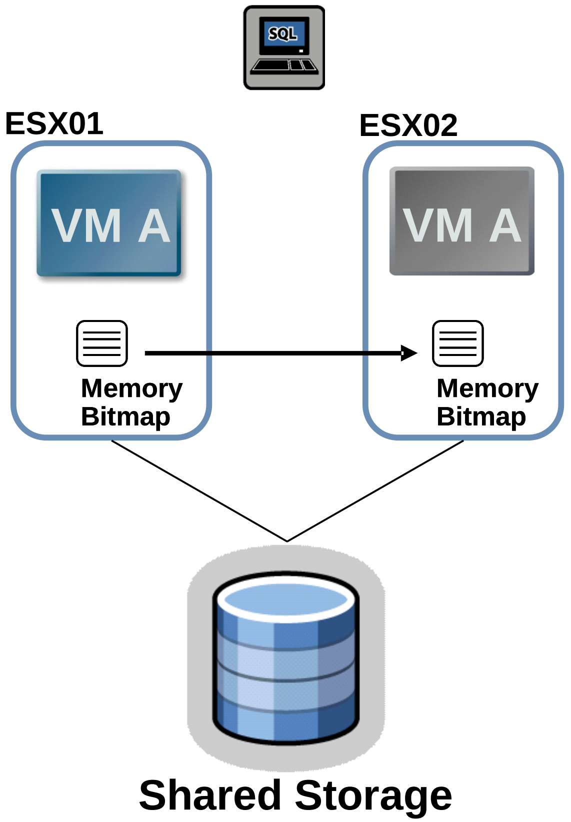
1) Provision new virtual machine on target host

VM Migration - Step 2



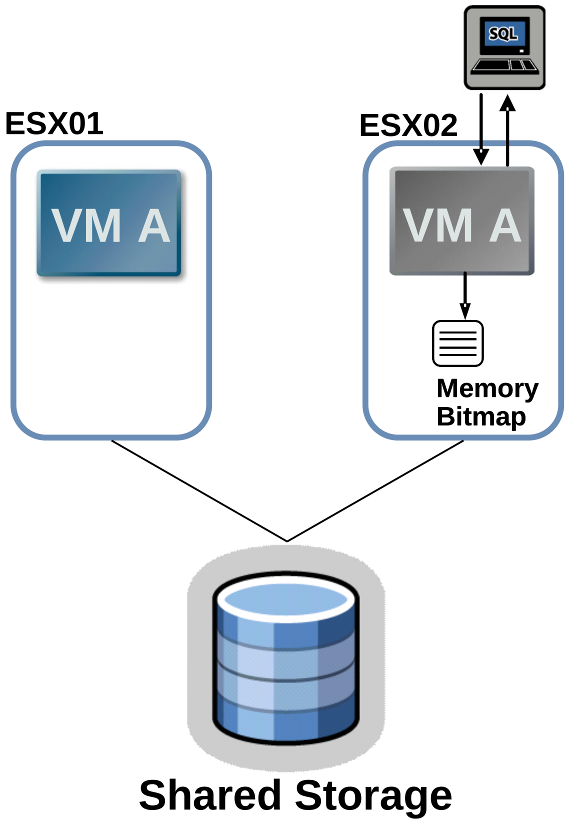
1) Provision new virtual machine on target host
2) Pre-copy memory from source to target, with ongoing changes logged to a bitmap

VM Migration - Step 3



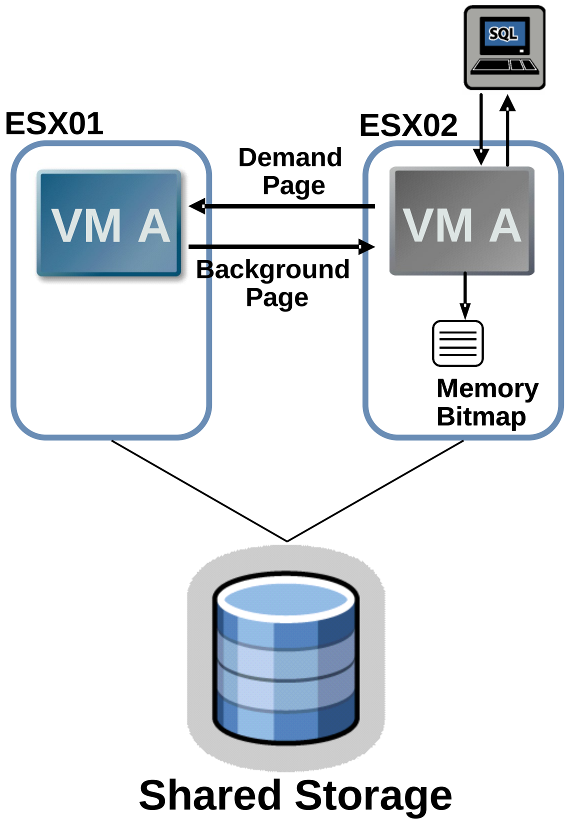
1) Provision new virtual machine on target host
2) Pre-copy memory from source to target, with ongoing changes logged to a bitmap
3) Suspend the virtual machine on the source host and copy memory bitmap to target host

VM Migration - Step 4



1) Provision new virtual machine on target host
2) Pre-copy memory from source to target, with ongoing changes logged to a bitmap
3) Suspend the virtual machine on the source host and copy memory bitmap to target host
4) Resume virtual machine on target host

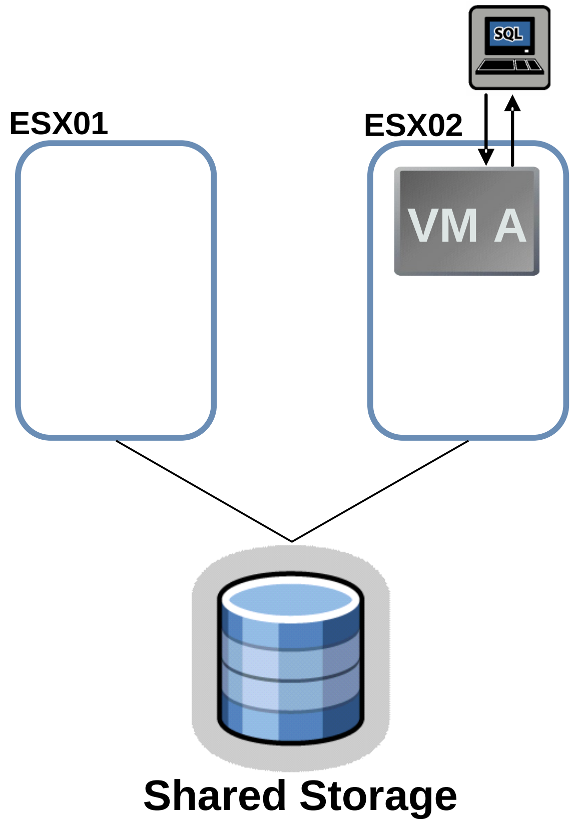
VM Migration - Step 5



1) Provision new virtual machineon target host
2) Pre-copy memory from source to target, with ongoing changes logged to a bitmap
3) Suspend the virtual machine on the source host and copy memory bitmap to target host
4) Resume virtual machine on target host
5) “Demand page” from the source virtual machine when system accesses modified memory

   “Background page” the source virtual machine until all memory has been successfully copied

VM Migration - Step 6

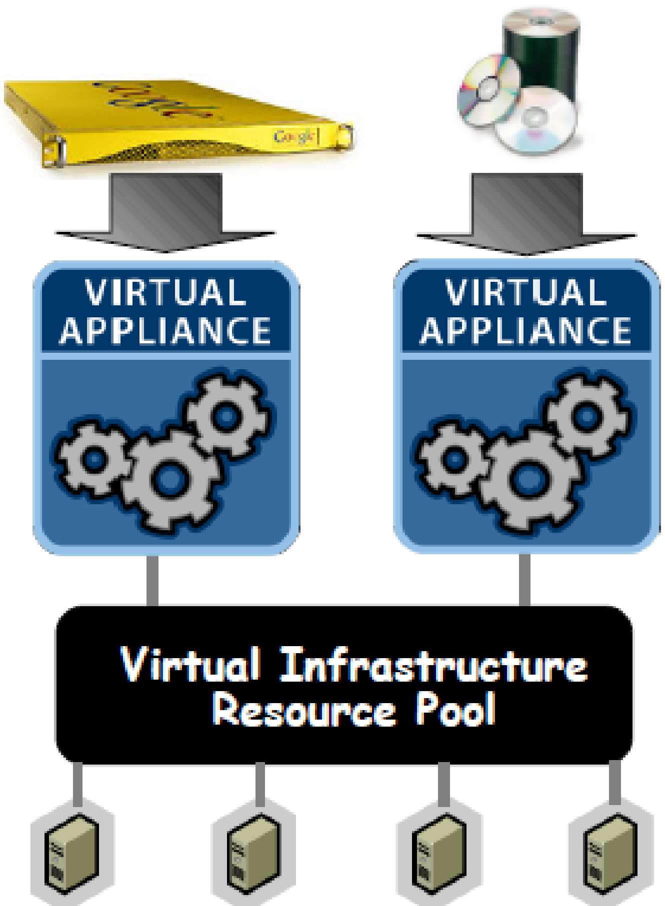


1) Provision new virtual machine on target host
2) Pre-copy memory from source to target, with ongoing changes logged to a bitmap
3) Suspend the virtual machine on the source host and copy memory bitmap to target host
4) Resume virtual machine on target host
5) “Demand page” from the source virtual machine when system accesses modified memory

   “Background page” the source virtual machine until all memory has been successfully copied

6) Delete virtual machine from source host

Virtual Appliances



- Pre-installed, Pre-configured SW stack
  - Lower "time to value" for customers
  - Reduced configuration matrix for ISV
  - Reduce support calls
- Run on Virtual Infrastructure to gain
  - HW independence
  - Better availability
  - Efficient load balancing
  - Consolidated management
  - ...