Operating Systems Design
23. Virtualization

Paul Krzyzanowski
pxk@cs.rutgers.edu

© 2012 Paul Krzyzanowski

Virtualization

- Memory virtualization
  - Process feels like it has its own address space
  - Created by MMU, configured by OS
- Storage virtualization
  - Logical view of disks “connected” to a machine
  - External pool of storage
- CPU/Machine virtualization
  - Each process feels like it has its own CPU
  - Created by OS preemption and scheduler

© 2012 Paul Krzyzanowski

Storage Virtualization

© 2012 Paul Krzyzanowski

Logical Volume Management

- "Logical partitions"
  - Span multiple physical disks
  - Can be resized
- Physical disk
  - Divided into one or more Physical Volumes
- Logical Volume Groups
  - Created by combining Physical Volumes.

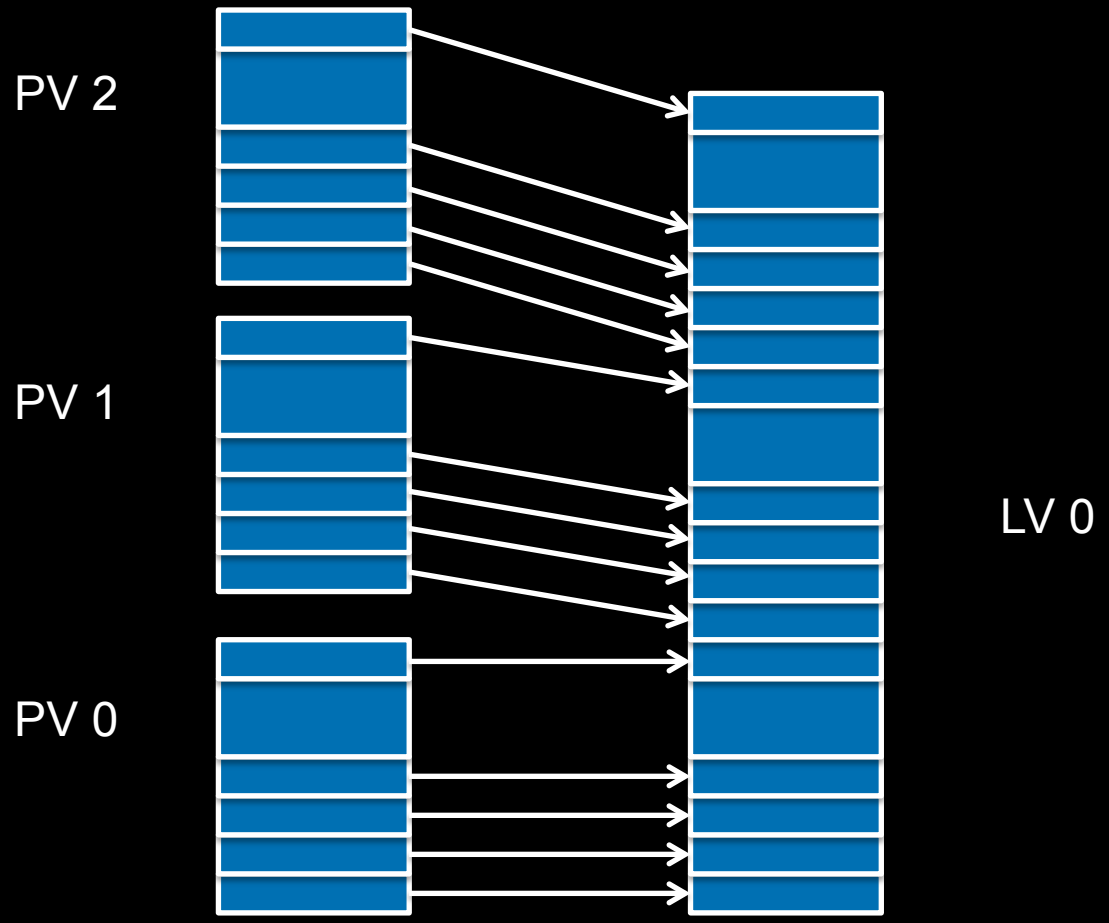
Mapping Logical to Physical data

- Storage on physical volumes is divided into clusters (misnamed *extents*)

- Logical volume defined and managed by mapping of logical extents to physical extents

LVM Linear Mapping

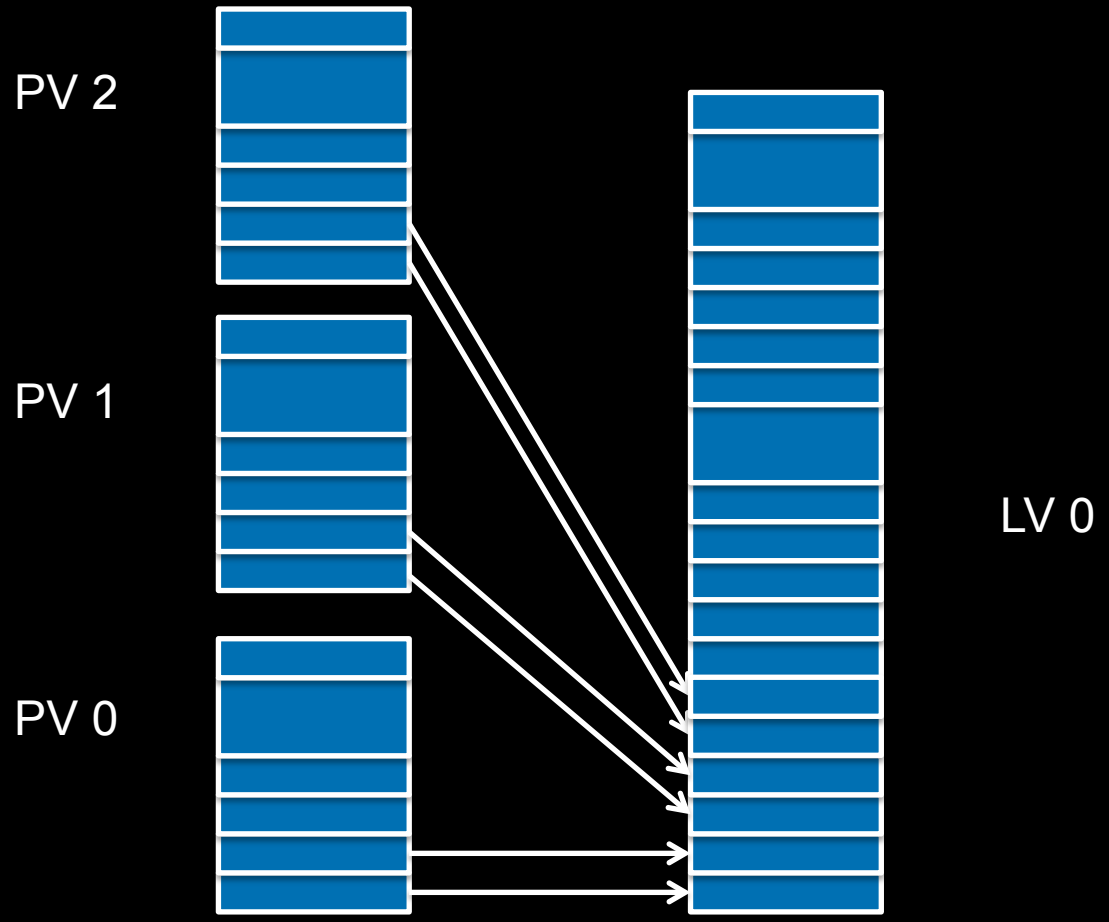
Concatenate multiple physical disks to create a larger disk



© 2012 Paul Krzyzanowski

LVM Striped Mapping

Groups from alternate physical volumes mapped to a logical volume.
N physical extents per stripe. Improve bandwidth of file transfers



© 2012 Paul Krzyzanowski

Advantages

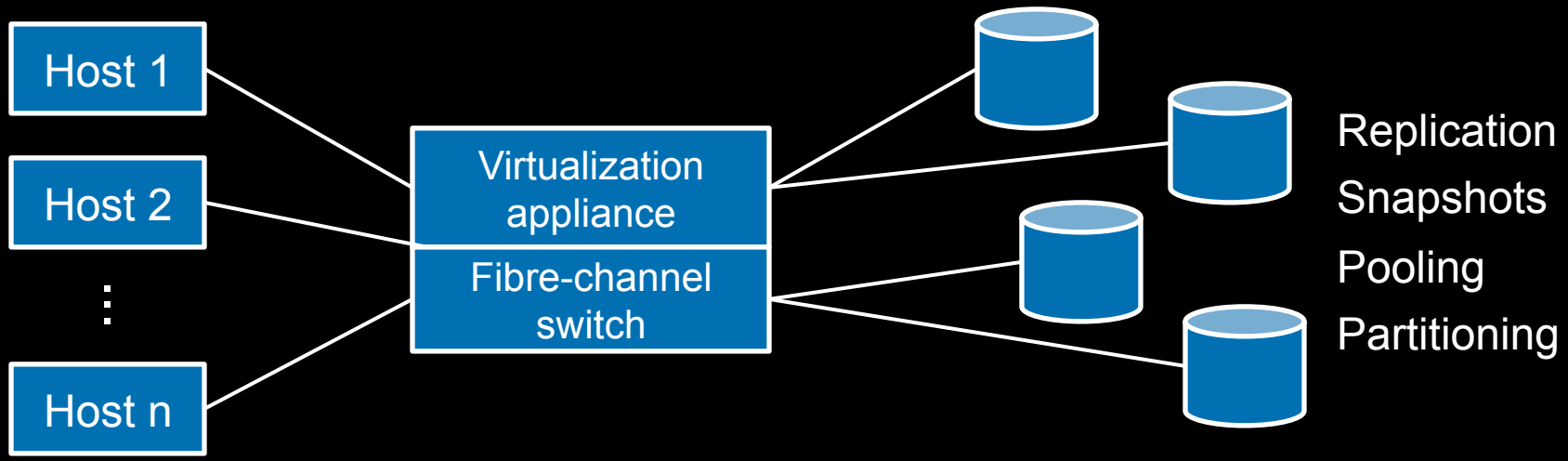
- Logical disks can be resized while mounted
  - Some file systems (e.g., ext3 on Linux or NTFS) support dynamic resizing
- Data can be relocated from one disk to another
- Improved performance (through disk striping)
- Improved redundancy (disk mirroring)
- Snapshots
  - Save the state of the volume at some point in time.
  - Allow backups to proceed while the file system is being modified

Storage Virtualization

- Dissociate knowledge of physical disks
  - The computer system does not manage physical disks
- Software between the computer and the disks manages the view of storage
- Examples:
  - Make eight 500 GB disks appear as one 4 TB disk
  - Make one 1 TB disk appear as two 400 GB disks and one 200 GB disk, with each of the 400 GB virtual disks available to different servers while the 200 GB disk can be shared by all.
  - Have all writes get mirrored to a backup disk
- Virtualization software translates read-block/write-block requests for logical devices to read-block/write-block requests for physical devices

Storage Virtualization

- Logical view of disks “connected” to a machine
- Separate logical view from physical storage
- External pool of storage



© 2012 Paul Krzyzanowski

Processor Virtualization

© 2012 Paul Krzyzanowski

Virtual CPUs (sort of)

- *What time-sharing operating systems give us*

- Each process feels like it has its own CPU
  - But cannot execute privileged instructions
    (e.g., modify the MMU or the interval timer, halt the processor, access I/O)
- Created by OS preemption and scheduler

Process Virtual Machines

- CPU interpreter running as a process
- Pseudo-machine with interpreted instructions
  - 1966: O-code for BCPL
  - 1973: P-code for Pascal
  - 1995: Java Virtual Machine (JIT compilation added)
  - 2002: Microsoft .NET CLR (pre-compilation)
  - 2008: Dalvik VM for Android
- Advantage: run anywhere, sandboxing capability
- No ability to even pretend to access the system hardware
  - Just function calls to access system functions

Machine Virtualization

Machine Virtualization

- Machine virtualization
  - Partition a physical computer to act like several real machines
    - Manipulate memory mappings
    - Set system timers
    - Access devices
  - Migrate an entire OS & its applications from one machine to another

- 1972: IBM System 370

Machine Virtualization

- An OS is just a bunch of code!

- Privileged vs. unprivileged instructions
- Regular applications use unprivileged instructions
  - Easy to virtualize

- If regular applications execute privileged instructions, they trap
- VM catches the trap and emulates the instruction

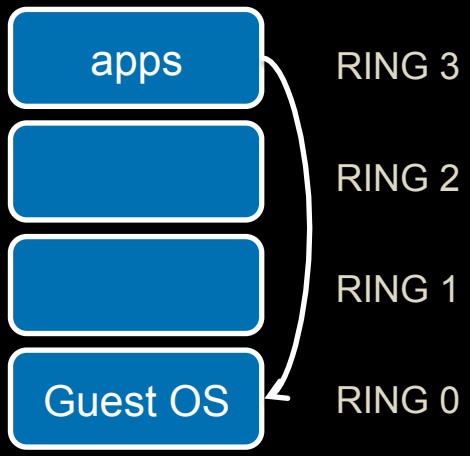
Hypervisor

- Program in charge of virtualization
  - Aka Virtual Machine Monitor
  - Provides the illusion that the OS has full access to the hardware
  - Arbitrates access to physical resources
  - Presents a set of virtual device interfaces to each host

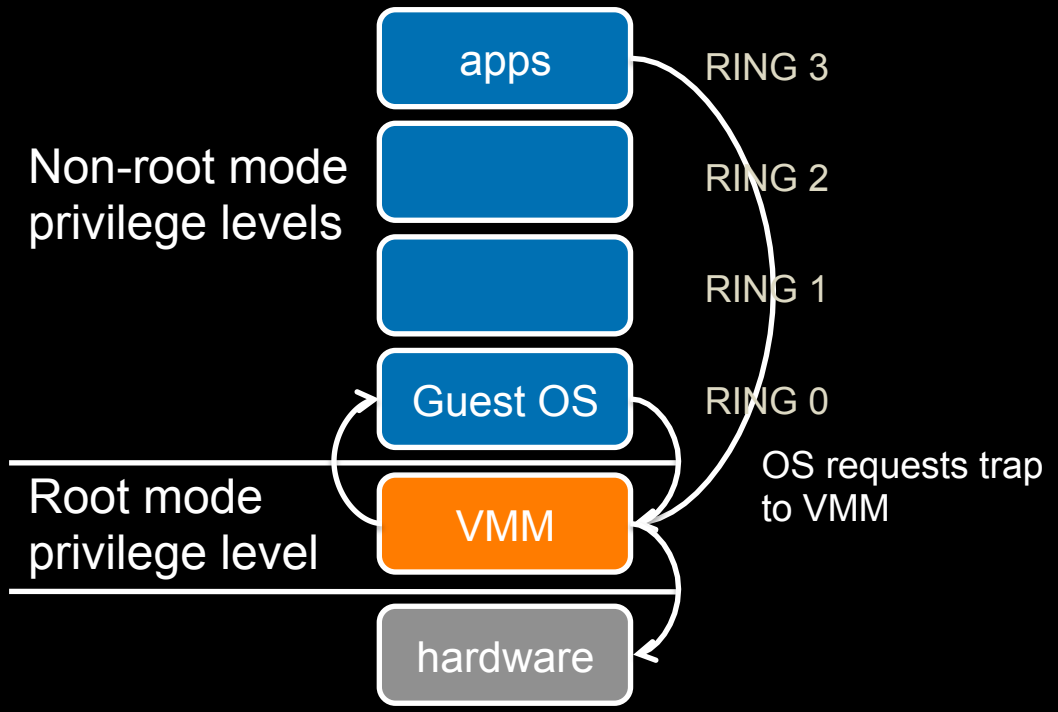
- Guest OS runs until:
  - Privileged instruction traps
  - System interrupts
  - Exceptions (page faults)
  - Explicit call: VMCALL (intel) or VMMCALL (AMD)

Hardware support for virtualization

- Root mode
  - Layer of execution more privileged than the kernel



Without virtualization



Intel Ugliness

- Intel/AMD systems prior to Core 2 Duo (2006) do not support trapping privileged instructions
- Two approaches
  - Binary translation (BT)
    - Scan instruction stream on the fly (when page is loaded) and replace privileged instructions with instructions that work with the virtual hardware (VMware approach)
  - Paravirtualization
    - Don't use non-virtualizable instructions (Xen approach)

Architectural Support

- Intel Virtual Technology
- AMD Opteron

- Certain privileged instructions are intercepted as VM exits to the VMM
- Exceptions, faults, and external interrupts are intercepted as VM exits
- Virtualized exceptions/faults are injected as VM entries

- BUT: not all x86 processors still support virtualization

Two Approaches

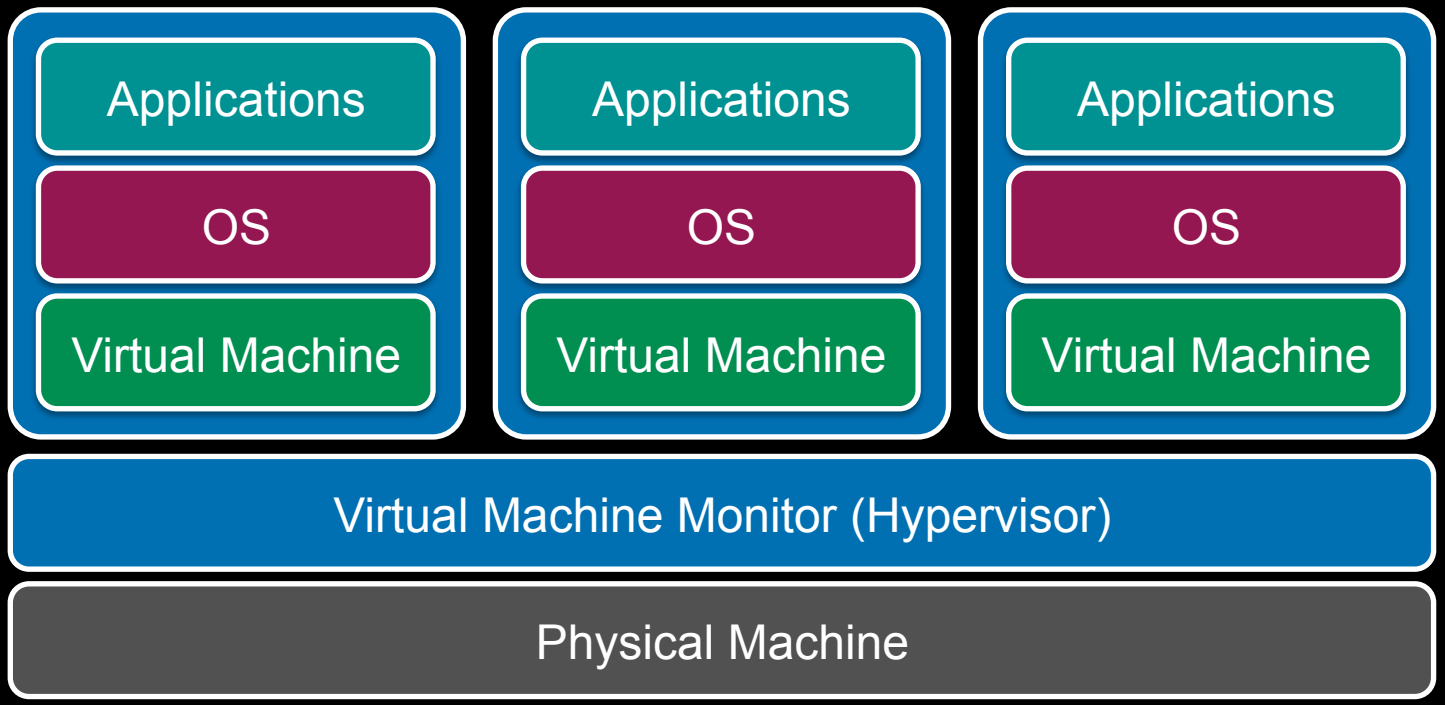
- Native VM (hypervisor model)
- Hosted VM

© 2012 Paul Krzyzanowski

Native Virtual Machine

Native VM (or *Type 1* or *Bare Metal*)

- No primary OS
- Hypervisor is in charge of access to the devices and scheduling

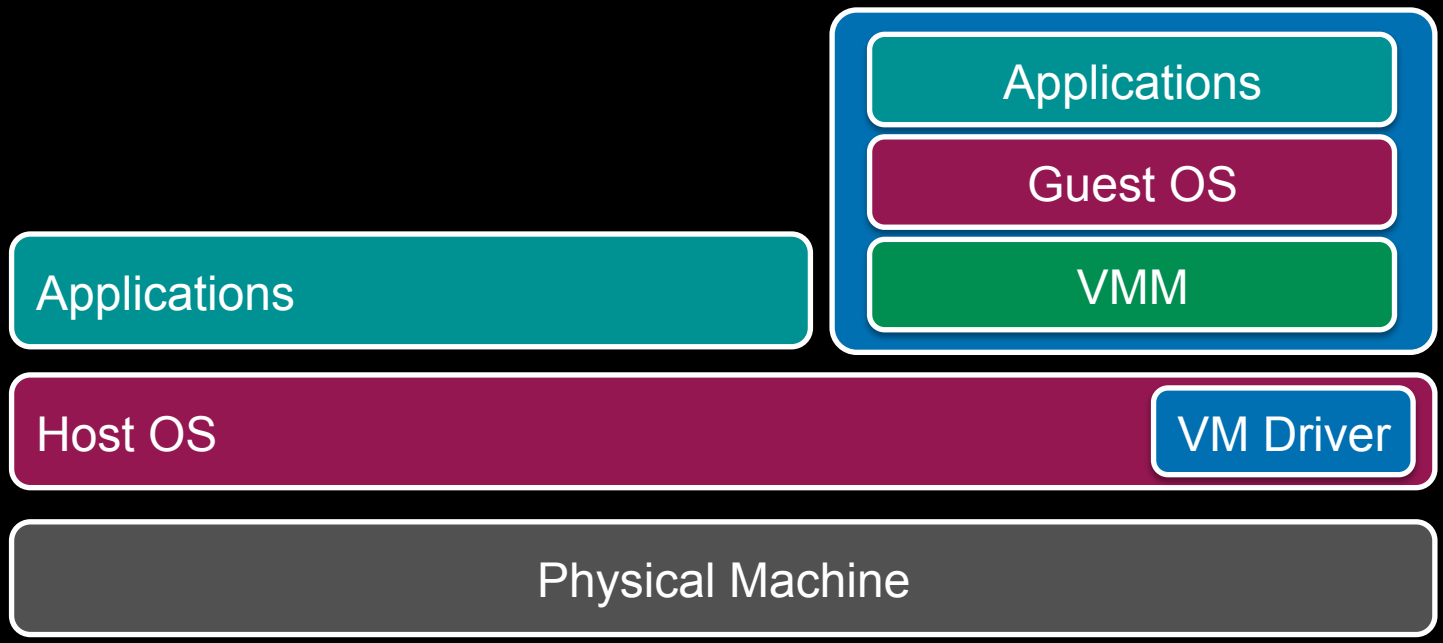


© 2012 Paul Krzyzanowski

Hosted Virtual Machine

Hosted VM

- Primary OS responsible for access to the raw machine
  - Lets you use all the drivers available for that primary OS
- Guest operating systems run under a VMM
- VMM invoked by host OS
  - Serves as a proxy to the host OS for access to devices



© 2012 Paul Krzyzanowski

Virtualizing Memory

- Similar to OS-based virtual memory
  - An OS sees a contiguous address space
  - Non necessarily tied to physical memory
- Need to virtualize MMU
  - Shadow page tables
    - Guest OS cannot access real page tables
    - Access attempts are trapped and emulated
  - VMM maps guest “physical memory” settings to actual memory

© 2012 Paul Krzyzanowski

Virtualizing Memory: Shadow Page Tables

- Need to virtualize MMU
  - VMM maps guest “physical memory” settings to actual memory
- Guest OS manages page tables (as always)
  - MMU hardware does not use these tables
- VMM allocates a subset of physical memory to the guest OS
- VMM manages all real page tables
  - Loaded on a VM context switch (trap a page table register write)
  - OS page tables are initially read-only
  - When the OS modifies a page table
    - VMM intercepts it (trap) and writes to a shadow table

© 2012 Paul Krzyzanowski

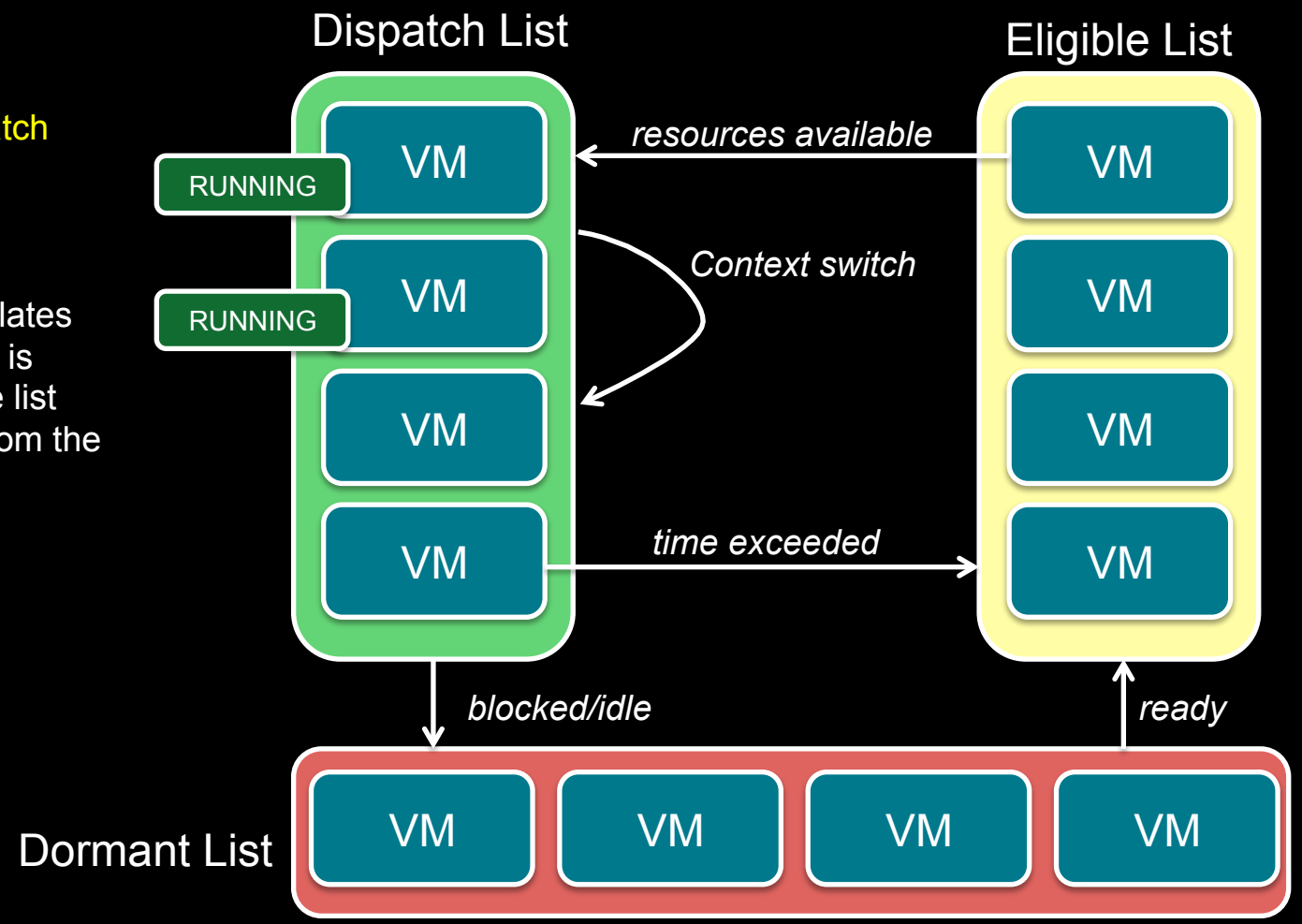
Scheduling VMs

- Each VM competes for a physical CPU
  - Typically # VMs > # CPUs
- VMs need to get scheduled
  - Each VM gets a time slice
    - Often round robin scheduler – or minor variations
  - Allocate CPU to a single-CPU VM
  - Allocate multiple CPUs to multi-CPU VMs: co-scheduling
    - Strict co-scheduler: VM with two virtual CPUs gets two real CPUs
    - Relaxed co-scheduler: if two CPUs are not available, use one
  - CPU affinity: try to run the VM on the same CPU
- VM scheduler controls the level of multiprogramming of VMs

© 2012 Paul Krzyzanowski

Scheduling VMs (IBM z/VM example)

- Run until a VM accumulates a dispatch time slice
- Readjust priority
- When a VM accumulates elapsed time slice, it is moved to the eligible list and a VM is taken from the eligible list



© 2012 Paul Krzyzanowski

Virtualizing Drivers & Events

- Device drivers
  - VMM has to multiplex physical devices & create network bridges
  - Virtualize network interfaces (e.g., MAC addresses)
  - Guest OS gets device drivers that interface to an abstract device implementation provided by the VMM
- VMM gets all system interrupts and exceptions
  - Needs to figure out which OS gets a simulated interrupt
  - Simulate those events on the guest OS

Live Migration

- Select alternate host (B)
  - Mirror block devices (for file systems)
  - Initialize VM on B
- Initialize
  - Copy dirty pages to host B iteratively
- To migrate
  - Suspend VM on A
  - Send ARP message to redirect traffic to B
  - Synchronize remaining VM state to B
  - Release state on A

Some Popular VM Platforms

- Native VMs
  - Microsoft Hyper-V
  - VMWare ESX Server
  - IBM z/VM (mainframe)
  - XenServer
    - Ran under an OS and provides virtual containers for running other operating systems. Runs a subset of x86. Routes all hardware accesses to the host OS.
    - Non-modified OS support for processors that support x86 virtualization
  - Sun xVM Server

- Hosted VMs
  - VMWare Workstation
  - VirtualBox
  - Parallels

Security Threats

- Hypervisor-based rootkits

- A system with no virtualization software installed but with hardware-assisted virtualization can have a hypervisor-based rootkit installed.

- Rootkit runs at a higher privilege level than the OS.
  - It's possible to write it in a way that the kernel will have a limited ability to detect it.

OS-Level Virtualization

- Not full machine virtualization
- Multiple instances of the same operating system
  - Each has its own environment
    - Process list, mount table, file descriptors, virtual network interface
- Advantage: low overhead: no overhead to system calls

- Examples:
  - *Linux VServer, Solaris Containers, FreeBSD Jails*
  - *Symantec Software Virtualization Solution* (originally *Altris Software Virtualization Services*)
    - Windows registry & directory tweaking
    - Allows multiple instances of applications to be installed

BSD Jails

- Jail
  - Directory subtree
    Root of namespace. Process cannot escape from this subtree
  - Hostname
    Hostname that will be used within the jail
  - IP address
    IP address used for a process within the jail
  - Command
    Command that will be run within the jail

The End

© 2012 Paul Krzyzanowski