

# OPEN SOURCE CLOUD COMPUTING FORUM

February 10, 2010

## KVM

Mike Day  
Chief Virtualization Architect,  
IBM Open Systems Development  
IBM Distinguished Engineer

# KVM: Overview

- Integrated Hypervisor for Linux
- Upstream since Linux 2.6.20 (2007)
- Elegant, simple design reuses Linux and builds upon CPU virtualization assistance
- Control over future evolution is held by linux development community
- Supported in RHEL since v5.4 (Sept. 2009)

# The Linux Kernel should also be a Hypervisor...

As a Linux developer, it's hard for me to be that interested in Xen... When you think about it, it is really quite silly. We advocate Linux for everything from embedded systems to systems requiring real-time performance, to high-end mainframes. I trust Linux to run on my dvd player, my laptop, and to run on the servers that manage my 401k. Is virtualization so much harder than every other problem in the industry that Linux is somehow incompatible of doing it well on its own? Of course not.

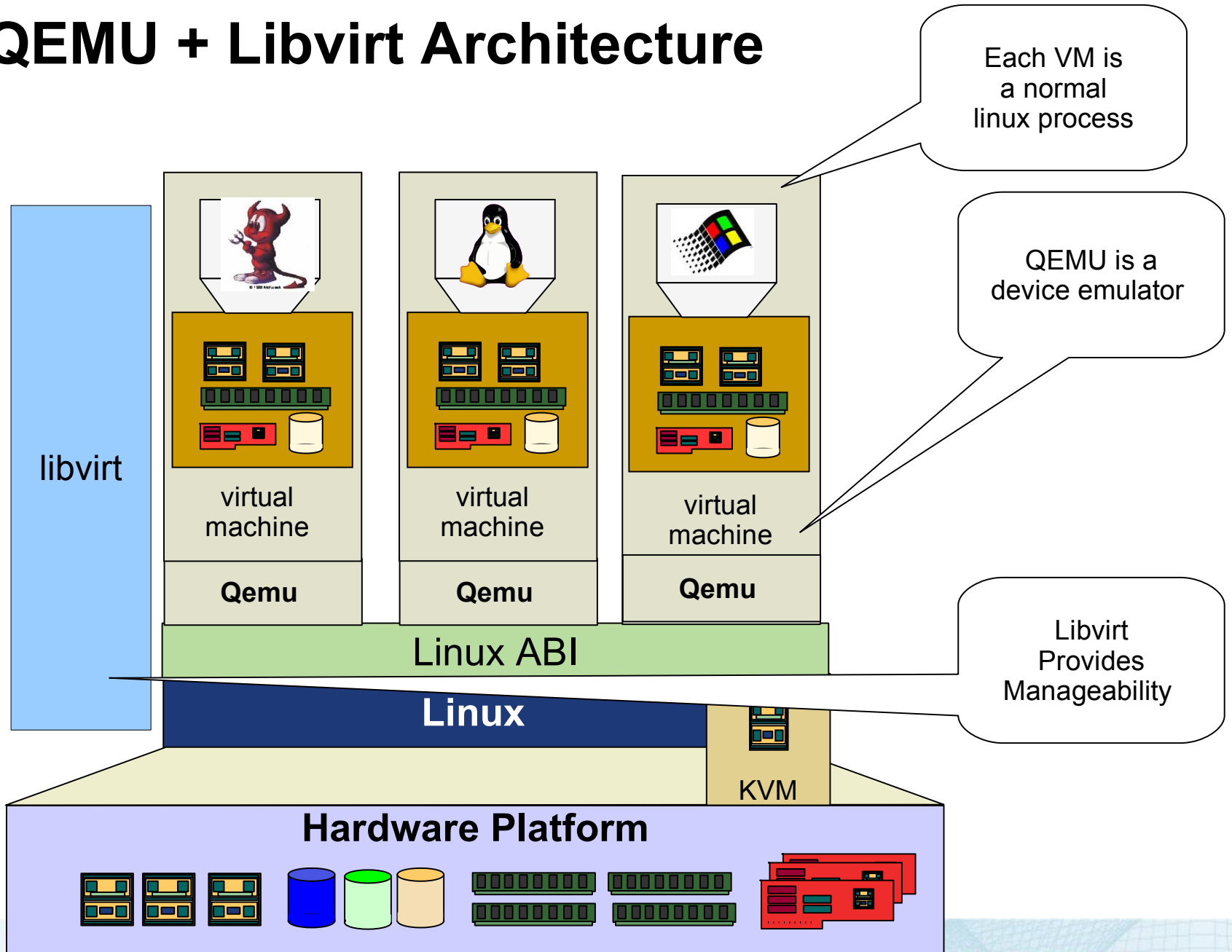
-- Anthony Liguori, Qemu maintainer

# ... So KVM Developers can focus on Virtualization

- Linux provides essential services
  - Hardware support
  - Bootstrap
  - Memory Management
  - Process Management and Scheduling
  - Access control
  - IPC and Sharing infrastructure
  - Scaling
  - RAS
  - Power Management

# KVM Development

# KVM + QEMU + Libvirt Architecture



Each VM is a normal linux process

QEMU is a device emulator

Libvirt Provides Manageability

# KVM Development Communities - 2009

- KVM-devel

- 18,303 messages
- 884 unique participants
- 382 unique address domains

9471 redhat.com  
1382 ibm.com  
929 intel.com  
949 novell.com

- Qemu

- 23,562 messages
- 757 unique participants
- 349 unique address domains

8751 redhat.com  
2643 ibm.com  
819 aurel32.net  
712 codesourcery.com

- Libvirt

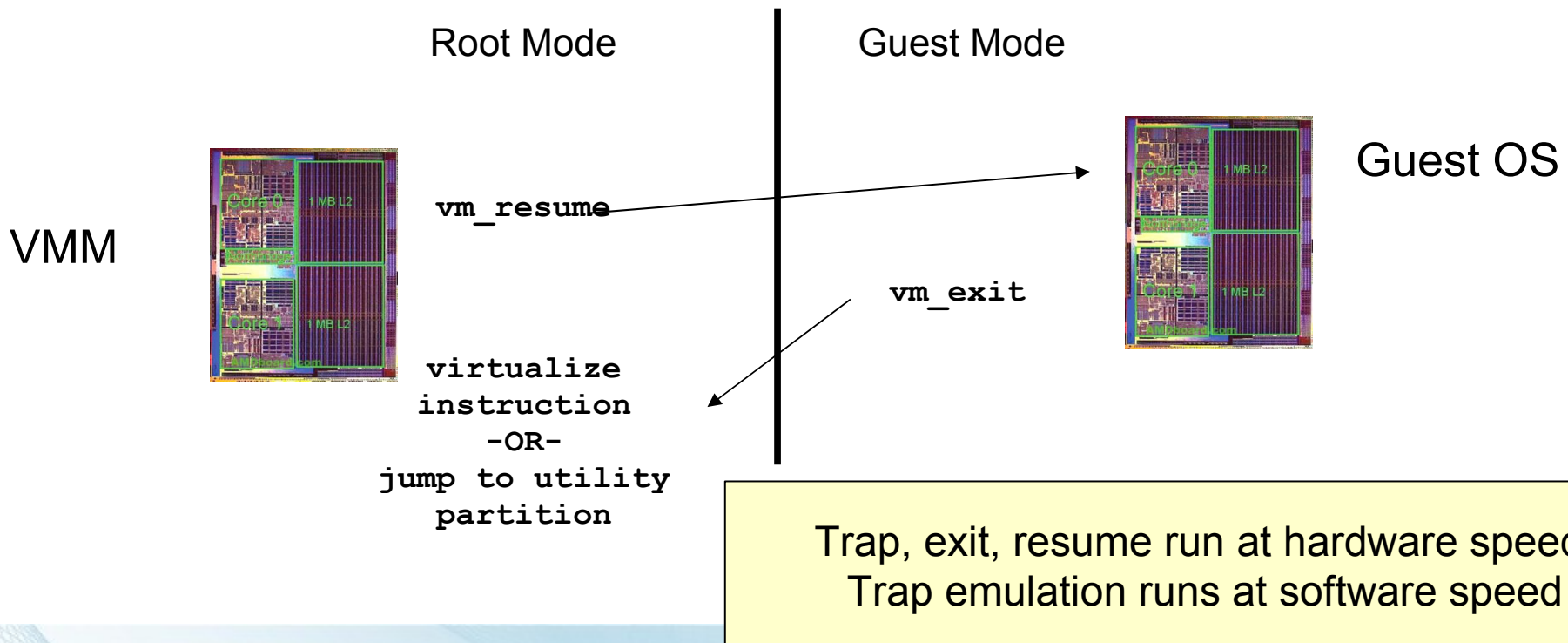
- 8,835 messages
- 370 unique participants
- 194 unique address domains

5791 redhat.com  
415 meyering.net  
260 ibm.com  
230 sun.com

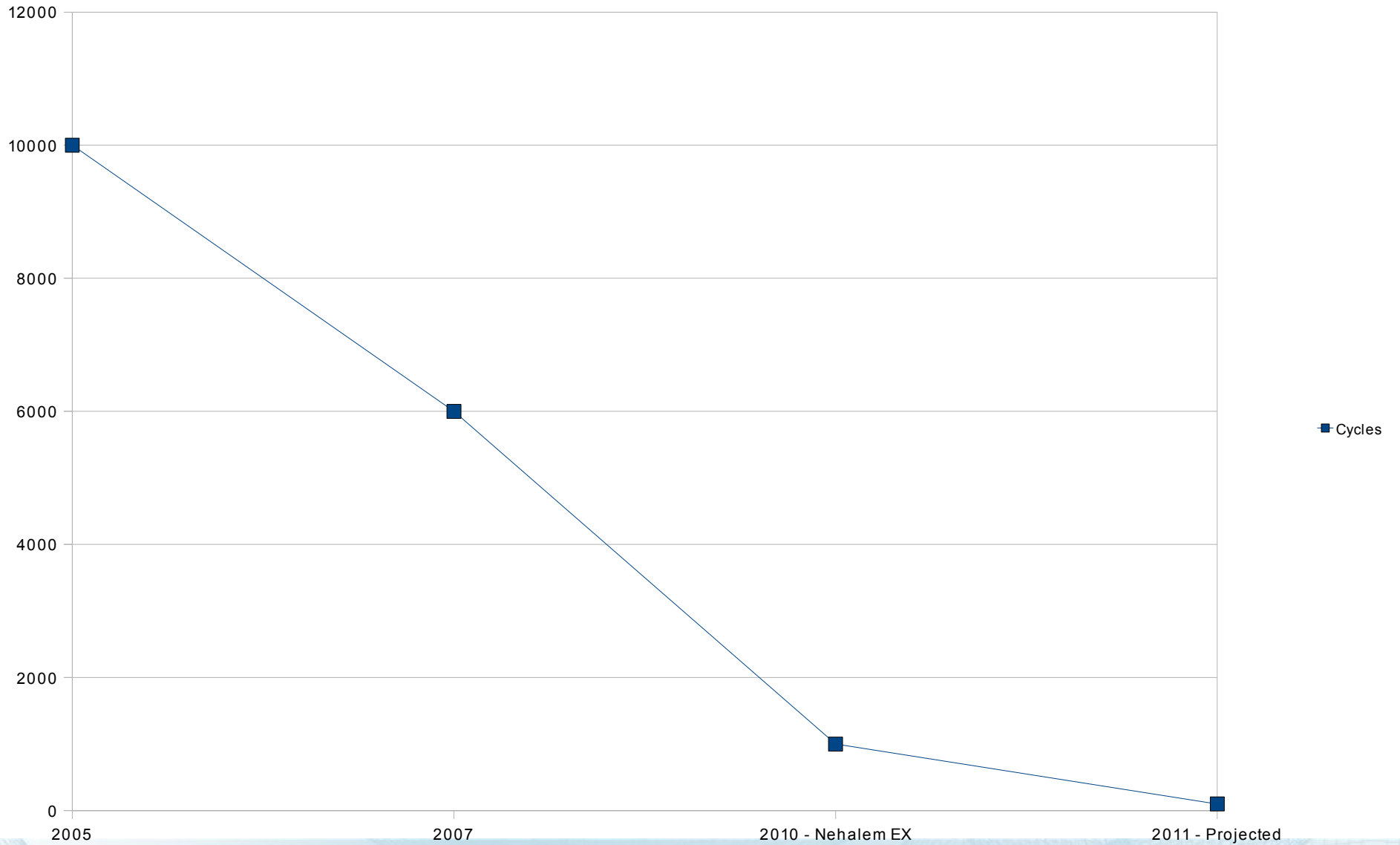
# KVM and Hardware Virtualization Assistance

# VMLAUNCH → VMEXIT → VMRESUME

- VMLAUNCH to start executing a guest
- When Guest OS traps, hardware executes a VMEXIT and returns control to the Virtual Machine Monitor
- The VMM emulates the trap and returns control to the Guest by executing VMRESUME

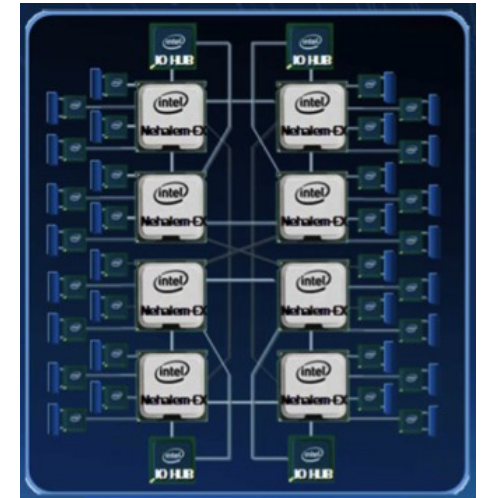


# VMExit Latency and Performance



# Nehalem EX “Westmere”

- 8 sockets, 8 cores per socket, 2 threads per core
  - 128-way SMP
- 16 slots DDR3 memory per socket, 16 GB per slot
  - 8 sockets X 256 GB = 2 TB RAM
  - On-socket buffering of DDR3 accesses
- Machine Check Architecture
  - Implementation of Itanium RAS features
  - Allows hypervisor to take a greater role in mitigating machine checks
  - Roadmap shows increase in types of machine checks that may be corrected or mitigated by the hypervisor
- Interrupt remapping
  - Lays groundwork for improvements in Sandy Bridge interrupt handling by VM guests
- VMExit cycle time
  - Order of magnitude improvement (10,000 -> 1,000)



# KVM as Cloud Infrastructure

# Cloud Computing and Hypervisors

- Cloud Computing is primary about Economics
  - Driving down the cost of all aspects of Data Center Operations
  - Sharing Data Center Resources for increased Flexibility
- For KVM, this translates to:
  - Upward pressure on VM Density
  - KVM must get more out of less hardware
  - Downward pressure on Energy Consumption
  - Increased Security and Auditing needs
  - Creative use of storage resources

# Cloud is Driving KVM Development...

- Physical Resource Over-provisioning
  - As long as guests don't experience peak load concurrently, we can “borrow” compute, I/O, and memory resources from one guest and “loan” them to another guest
  - Transparent memory sharing
  - Memory “Ballooning” (memory borrowing)
  - Host memory swapping
  - VCPU over-provisioning
    - Virtual CPUs > physical CPUs
- In best cases, resources can be highly leveraged

# Cloud is Driving KVM Development...(cont'd.)

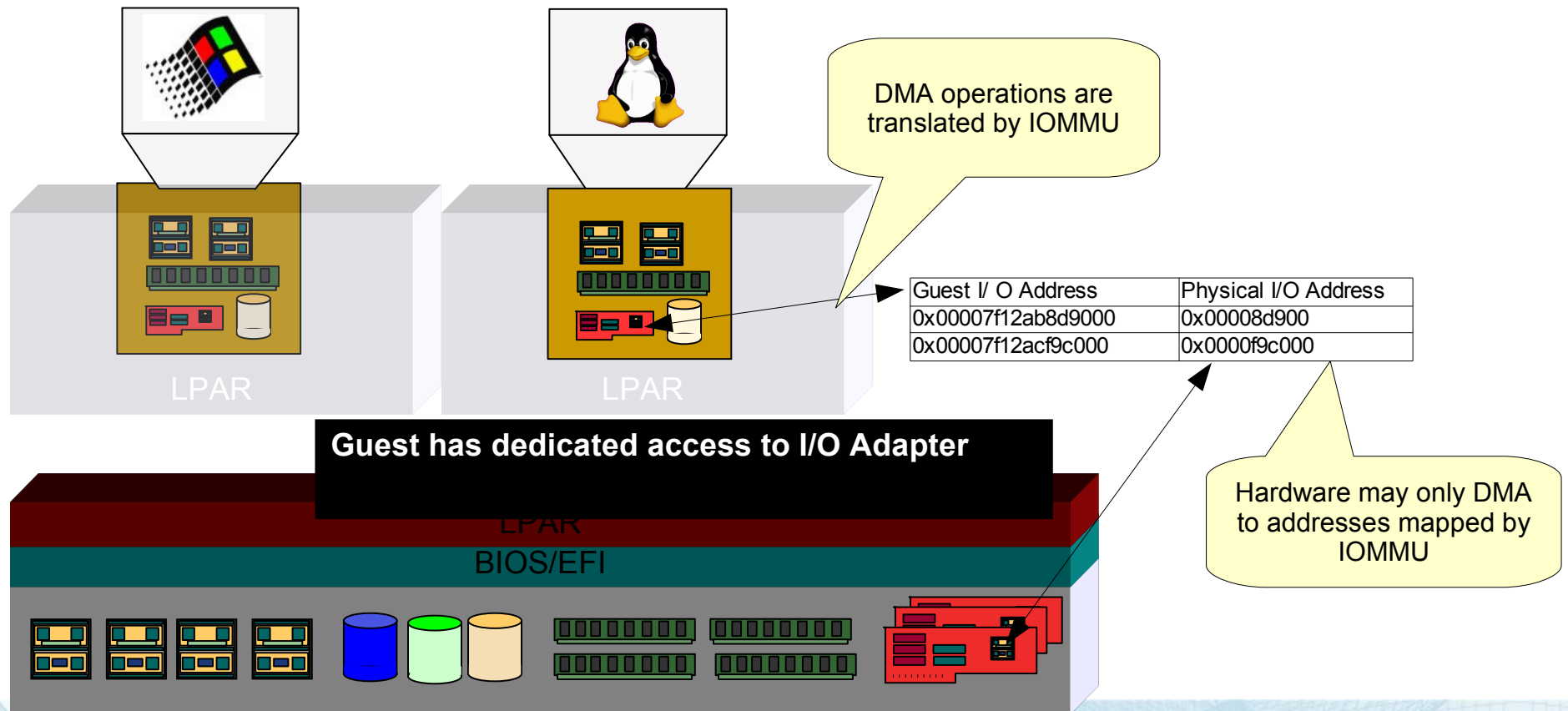
- Increased Density requires greater scalability
  - Recent enhancements such as read-copy-update for shared data structures
    - Avi Kivity recently increased the vcpu max to 64
  - Testing and validation with large systems
    - > 512 Gb RAM
    - > 16-32 cores (32-64 threads)
  - I/O latency and throughput enhancements
    - In-kernel paravirtual I/O server (vhost-net)
    - PCI pass-through and SR-IOV
- Multi-tenancy requires Security validations and resource isolation guarantees

# I/O Virtualization – The Current Bottleneck

# I/O Memory Translation with VT-D

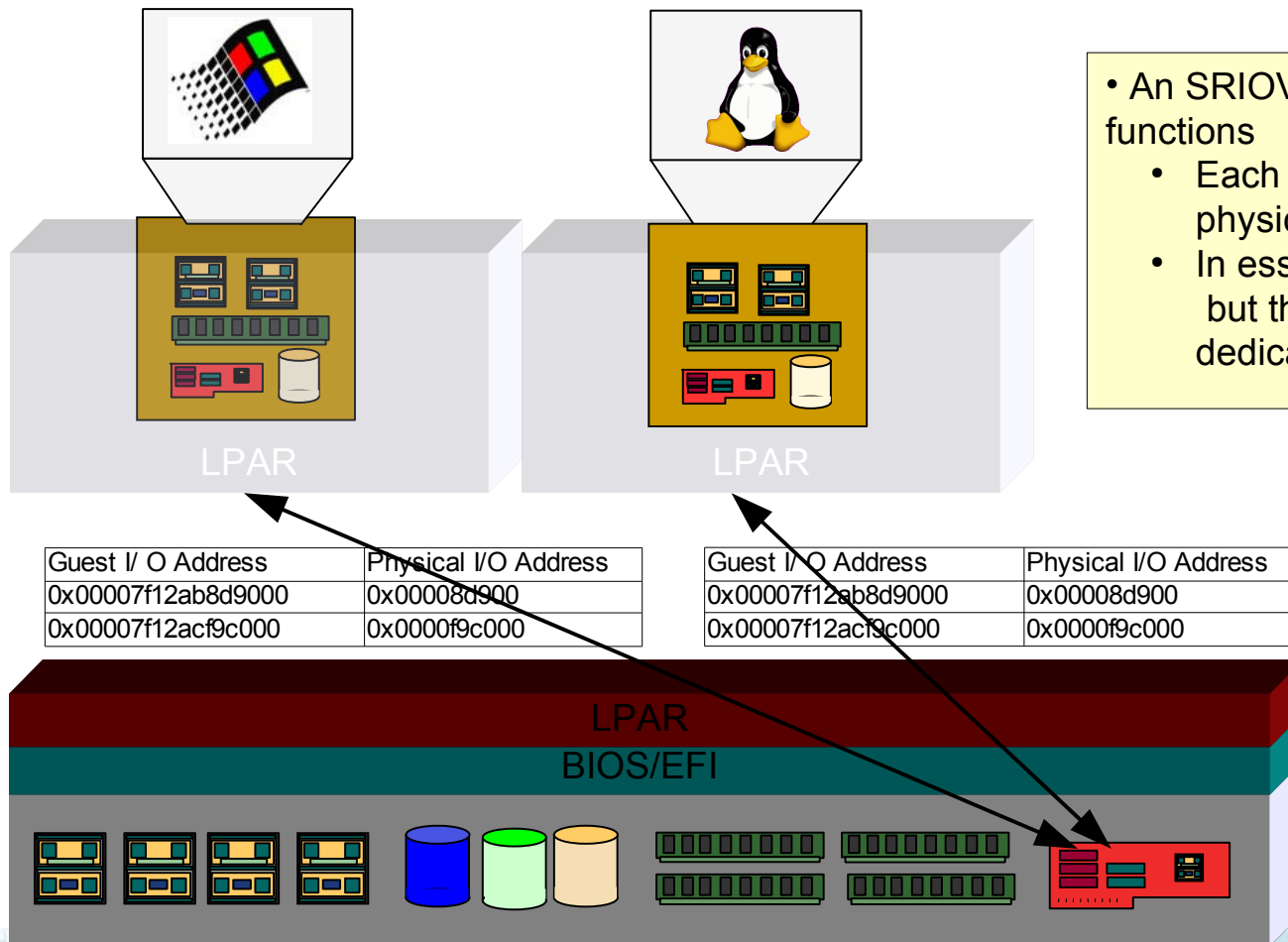
Each guest may have its own I/O translation table

Guests and hardware may safely DMA without VMM interposition



# Single-root I/O Virtualization (SRIOV)

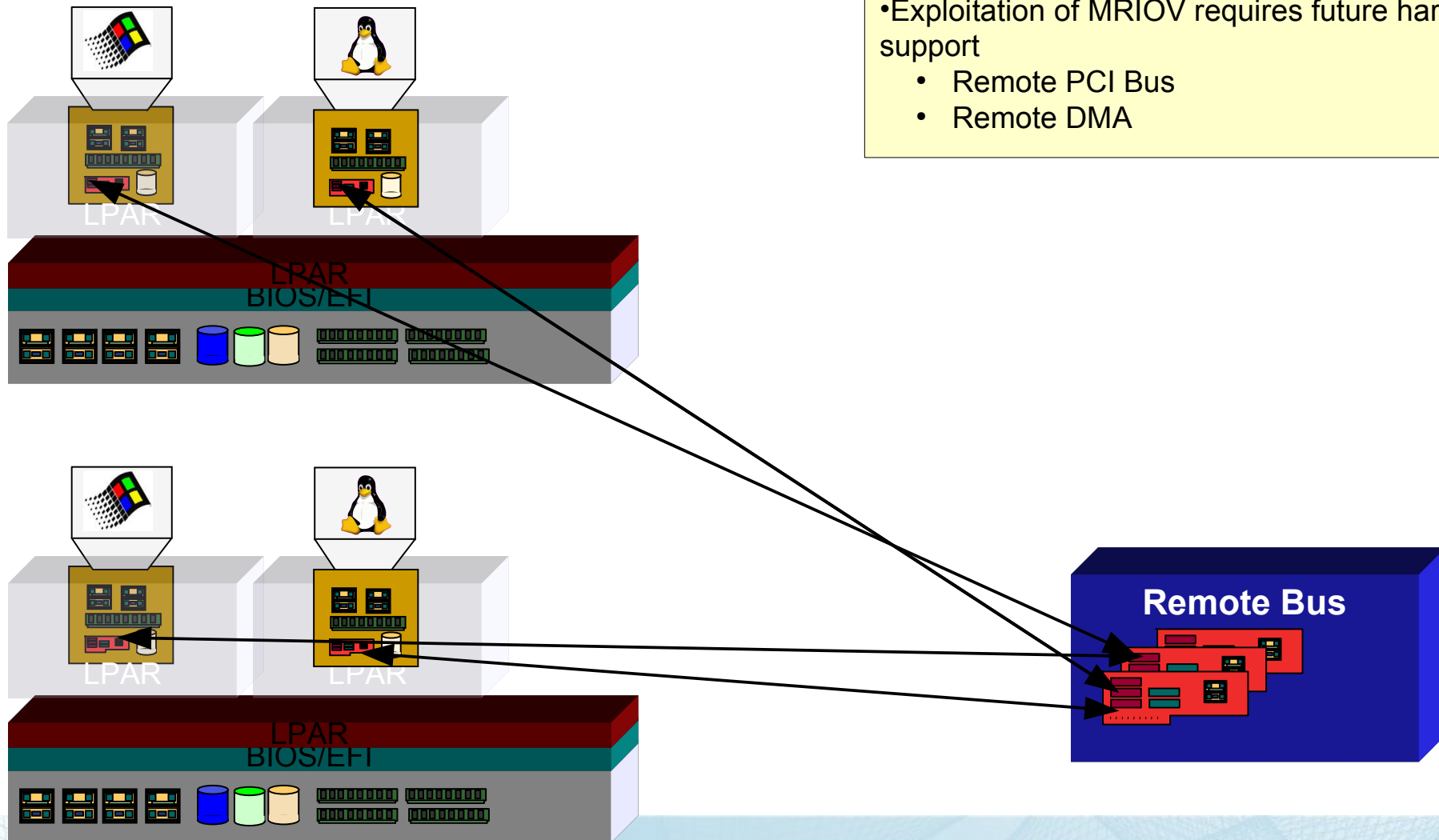
“Root” in “Single-root” refers to PCI bus and device tree



- An SRIOV PCI Device has multiple PCI functions
  - Each function behaves like a distinct physical adapter
  - In essence, the PCI device virtualizes itself, but the guest thinks it is controlling a dedicated I/O adapter

# Multi-root I/O Virtualization (MRIOV)

“Root” in “Multi-root” refers to PCI bus and device tree



•Exploitation of MRIOV requires future hardware support

- Remote PCI Bus
- Remote DMA

# I/O Paravirtualization

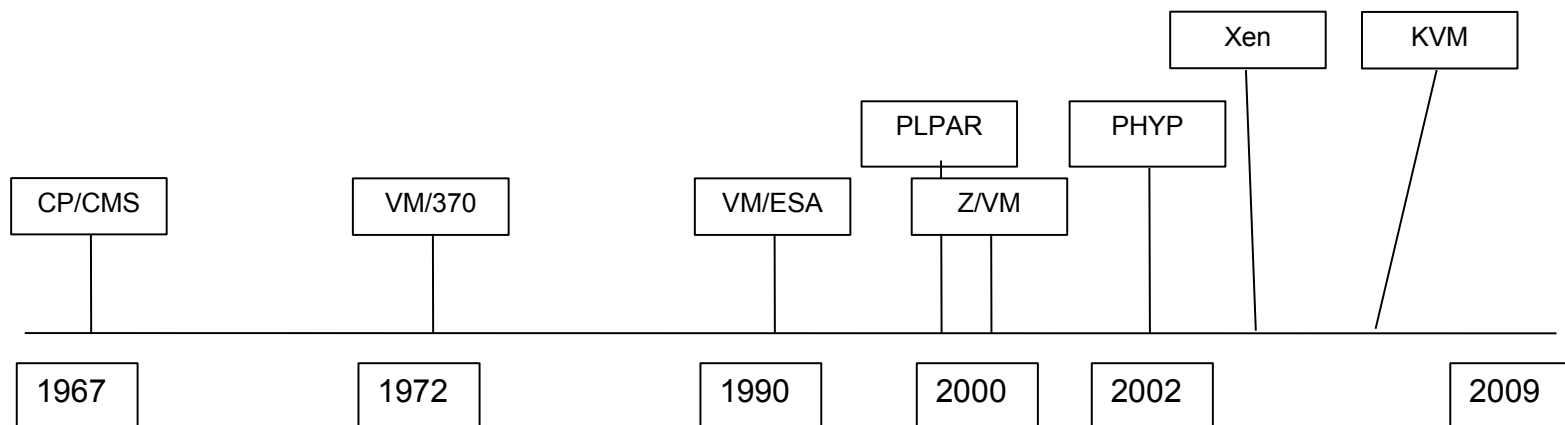
- KVM Developers prefer paravirtualized I/O
  - Performance can be comparable to direct pass-through
  - More flexible
    - Live Guest Migration
    - Integrated virtual switching
  - Hypervisor can optimize I/O scheduling to meet different performance or resource goals
  - SR and MR -IOV hardware can be paravirtualized in creative ways

# Current Upstream Development to watch

- Lockholder preemption
  - Increases performance for SMP guests
- Memory over-commitment
  - Transparent large pages
  - Guest page hinting
  - Continued Kernel Shared Memory (KSM) advances
  - Asynchronous Page Fault handling
- Virtual Switch and vhost-net Improvements
  - VEPA support
- Much more

# At IBM we like working on KVM

- 42 years of experience virtualizing our servers
  - Virtualization was originally developed to make better use of critical hardware
  - IBM runs Linux as a first-class virtualized OS across our entire hardware portfolio
  - KVM is the latest, and even better, its Linux!



# Thank You

[mdday@us.ibm.com](mailto:mdday@us.ibm.com)