



# Virtualisation The KVM Way

**Amit Shah**  
[amit.shah@qumranet.com](mailto:amit.shah@qumranet.com)

**convergence/2008**



# Advances with Virtualisation (?)

“Operating systems are becoming less important with introduction of virtualization.”

-- More about Virtualization,

<http://www.cse.iitb.ac.in/convergence/?file=quotes.php>



# Virtualisation

- ◆ Simulation of computer system in software
- ◆ Components
  - ◆ Processor Management: register state, instructions, exceptions
  - ◆ Memory Management: paging, protection, TLB
  - ◆ IO Management: storage, human interface
- ◆ Essentials:
  - ◆ Performance
  - ◆ Fidelity



# Uses

- ◆ Server consolidation
  - ◆ Increase server utilization
  - ◆ Reduce mgmt cost/overhead
- ◆ Testing, R&D
  - ◆ Rapid deployment
  - ◆ Save redundant hw
  - ◆ Legacy OS/HW support
- ◆ Virtual Desktops
  - ◆ Centralized create/config/deploy/backup/restore/control
  - ◆ Reduced TCO



# Virtualisation Basics

- ◆ Trap changes to privileged state
  - ◆ Guest cannot access hardware
- ◆ Hide privileged state
  - ◆ Guest cannot detect that the host is changing things behind its back
- ◆ Example: interrupt enable flag



# A Look Back

- ◆ “Native” Hypervisors
  - ◆ Have a runtime
  - ◆ Need a “primary” guest OS
  - ◆ Examples: Xen, VMWare ESX Server, IBM mainframes
- ◆ Containers
  - ◆ Different namespaces for different guests
  - ◆ Run on host kernel
  - ◆ Userland can be different from host
  - ◆ Examples: OpenVZ, FreeVPS, Linux-Vserver
- ◆ Paravirtualisation
- ◆ Emulation
  - ◆ Examples: QEMU, PearPC



# Virtualisation

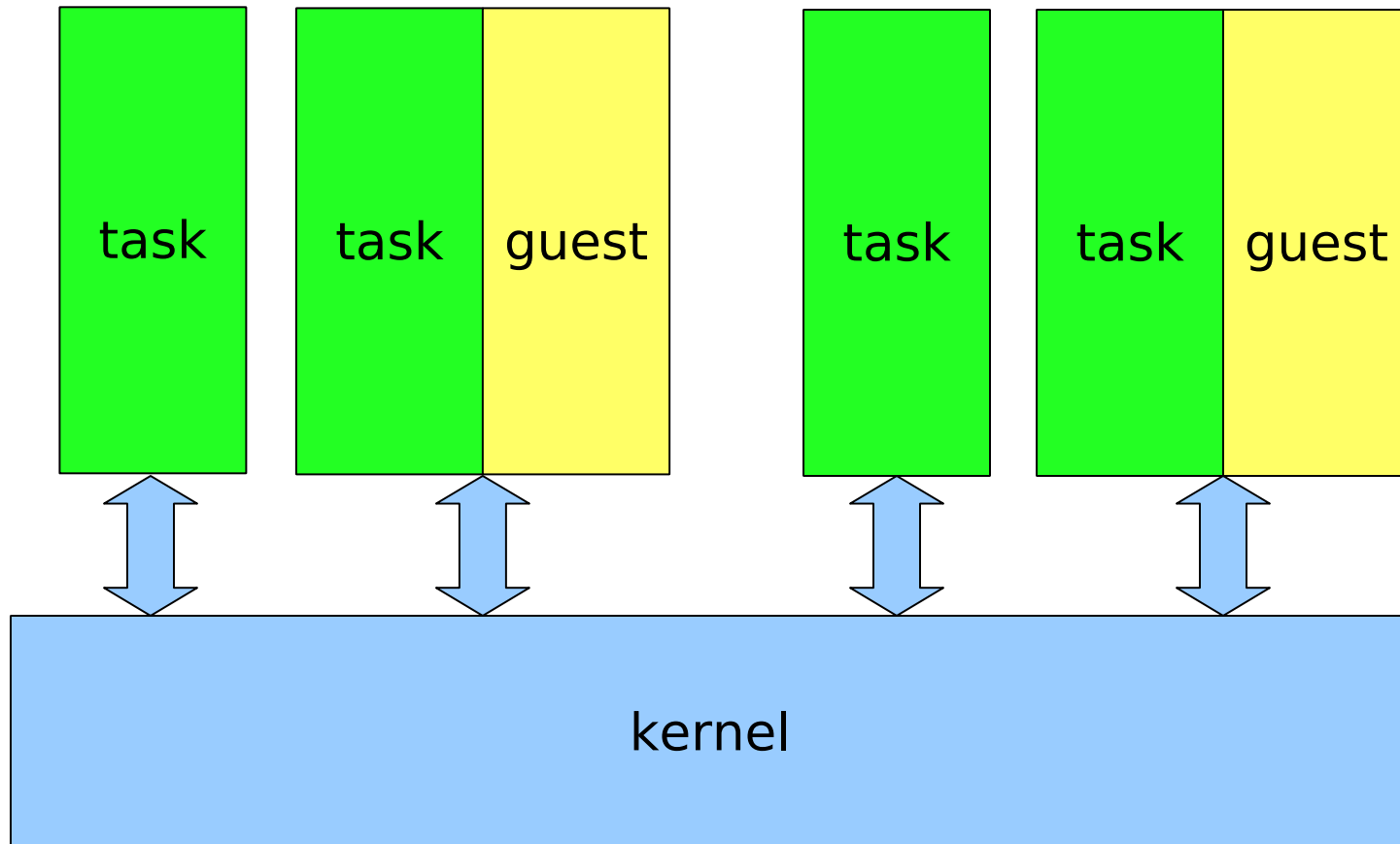
- ◆ Simulation of computer system in software
- ◆ Components
  - ◆ Processor Management: register state, instructions, exceptions
  - ◆ Memory Management: paging, protection, TLB
  - ◆ IO Management: storage, human interface
- ◆ Essentials:
  - ◆ Performance
  - ◆ Fidelity



# The KVM Approach

- ◆ We had most of the hypervisor ready: Linux
- ◆ Reuse code as much as possible
- ◆ Focus on virtualisation, leave other things to respective developers
- ◆ Integrate well into existing infrastructure, codebase and mindset
- ◆ Linux
  - ◆ Add capability to run a guest
- ◆ QEMU
  - ◆ IO virtualisation
- ◆ Difference from *emulation* is emphasis on near-native performance

# KVM Process Model

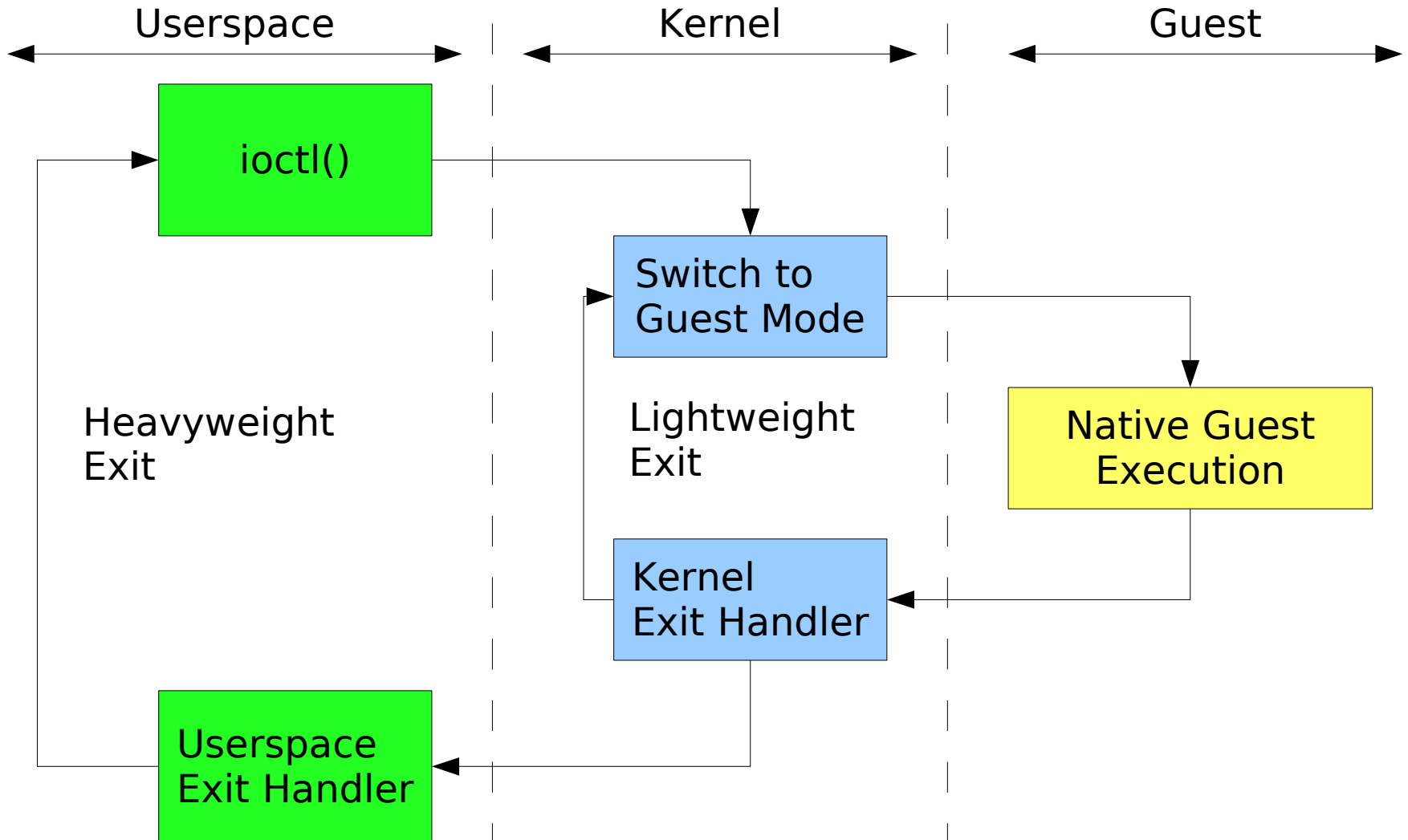


# KVM Process Model (cont'd)

- ◆ Guests are scheduled as regular processes
- ◆ kill(1), top(1) work as expected
- ◆ Guest physical memory is mapped into the task's virtual memory space
- ◆ Virtual processors in one VM are threads



# KVM Execution Model



# X86 Hardware Extensions

- ◆ 'guest mode' in addition to user and kernel modes
- ◆ Raise a trap for all privileged instructions
- ◆ Virtualised registers
- ◆ Processor
  - ◆ Intel-VT (VMX)
  - ◆ AMD-V (SVM)
- ◆ MM
  - ◆ EPT (Intel)
  - ◆ NPT (AMD)
- ◆ IO
  - ◆ VT-d (Intel)
  - ◆ IOMMU (AMD)



# What's handled in the kernel?

- ◆ CPU virtualisation (special instructions)
- ◆ MMU virtualisation
- ◆ Local APIC, PIC and IOAPIC
- ◆ (guest) paravirtualised network and block device
  - ◆ virtio-net, virtio-block
- ◆ (guest) paravirtualised kernel support code
  - ◆ paravirt\_ops



# Flow Example: Memory Access

- ◆ Guest accesses an unmapped memory location
- ◆ Hardware traps into kernel mode
- ◆ kvm walks the guest page table, determines guest physical address
- ◆ kvm performs guest physical -> host physical translation
- ◆ kvm installs shadow page table entry containing guest virtual -> host physical translation
- ◆ Processor restarts execution of faulting instruction



# KVM on other architectures

- ◆ s390
  - ◆ IBM mainframes: a hypervisor is a must
  - ◆ WIP
- ◆ IA-64
  - ◆ Should be merged in 2.6.26
- ◆ x86
  - ◆ kvm-lite
- ◆ Embedded PowerPC
  - ◆ Architecture support for hypervisor
  - ◆ WIP

# Paravirtualisation

- ◆ Modifying guest OS for performance
- ◆ Virtio
  - ◆ Common drivers for all hypervisors
  - ◆ Hypervisor-specific backend
  - ◆ KVM backend in qemu
  - ◆ Faster performance
  - ◆ Efficient block, net drivers
  - ◆ Balloon
- ◆ PV DMA
  - ◆ Pass through Ethernet devices
- ◆ paravirt\_ops

# KVM Pros

- ◆ Leverages Linux scheduler, memory management, I/O
- ◆ No scheduler involvement for I/O
- ◆ Full virtualisation: No changes to the guest necessary
  - ◆ Paravirt drivers available for better performance
- ◆ Uses existing Linux security model
  - ◆ can run VM as ordinary user
- ◆ Uses existing management tools
- ◆ Power management
- ◆ Guest memory swapping
- ◆ Real-time support
- ◆ NUMA support
- ◆ Leverages Linux development momentum
- ◆ Live Migration
- ◆ Snapshotting of disks and VMs

# Distro / Industry interest

- ◆ libvirt
  - ◆ Managing various guests under a hypervisor
  - ◆ Support for Xen, KVM
  - ◆ APIs between UI, middle layer and virtualisation backend
- ◆ Distributions
  - ◆ Debian
  - ◆ Ubuntu
  - ◆ RedHat EL
  - ◆ SLES
- ◆ Qumranet

# Release Philosophy

- ◆ Development snapshots every 1-2 weeks
  - ◆ Release early and often
  - ◆ Features introduced quickly
  - ◆ Bugs fixed quickly
  - ◆ Bugs added quickly
  - ◆ Allows developers and users to track and test the latest and greatest
- ◆ Stable releases part of Linux 2.6.x
  - ◆ With bugfixes going into Linux 2.6.x.y



# Journey

- ◆ Linux 2.6.20 (4 Feb 2007): Initial release
- ◆ Linux 2.6.21 (25 Apr 2007): Stability, suspend/resume
- ◆ Linux 2.6.22 (8 Jul 2007): Stable ABI
  - ◆ Old userspace, new kernel
  - ◆ New userspace, old kernel
- ◆ Linux 2.6.23 (9 Oct 2007): SMP, performance
- ◆ Linux 2.6.24 (24 Jan 2008): In-kernel APIC, preemptibility, virtio
- ◆ Linux 2.6.25 (soon): Swapping, paravirt\_ops, balloon drv
- ◆ Linux 2.6.26 (future): NPT, more paravirt, mmu notifiers, ...



# KVM is Developer-friendly

- ◆ No need to reboot (usually)
- ◆ Netconsole, oprofile, all the tools work
- ◆ Small codebase
- ◆ Friendly community



# Future

- ◆ Consolidate various virtualisation solutions existing in the kernel
  - ◆ Started with move to virt/ from drivers/kvm/
- ◆ More architecture support
- ◆ More hardware features support
- ◆ More paravirtualisation support
- ◆ Improve guest scaling
- ◆ Support for management layers like libvirt
- ◆ Intel Real Mode



# Do Read

- ◆ virt/\*, arch/x86/kvm/\*
- ◆ [KvmForum2007](http://kvm.qumranet.com) wiki page on <http://kvm.qumranet.com>
- ◆ [kvm-devel@lists.sourceforge.net](mailto:kvm-devel@lists.sourceforge.net)
- ◆ [virtualization@lists.osdl.org](mailto:virtualization@lists.osdl.org)



# Thank You

---