QUMRANET™

Virtualisation: The KVM Way

Amit Shah
amit.shah@qumranet.com

foss.in/2007

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

Copyright © 2007 Qumranet, Inc. All rights reserved.

Uses

- Server consolidation
- Testing, R&D
- Virtual Desktops

Copyright © 2007 Qumranet, Inc. All rights reserved.

QUMRANET™

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

Copyright © 2007 Qumranet, Inc. All rights reserved.

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

Copyright © 2007 Qumranet, Inc. All rights reserved.

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

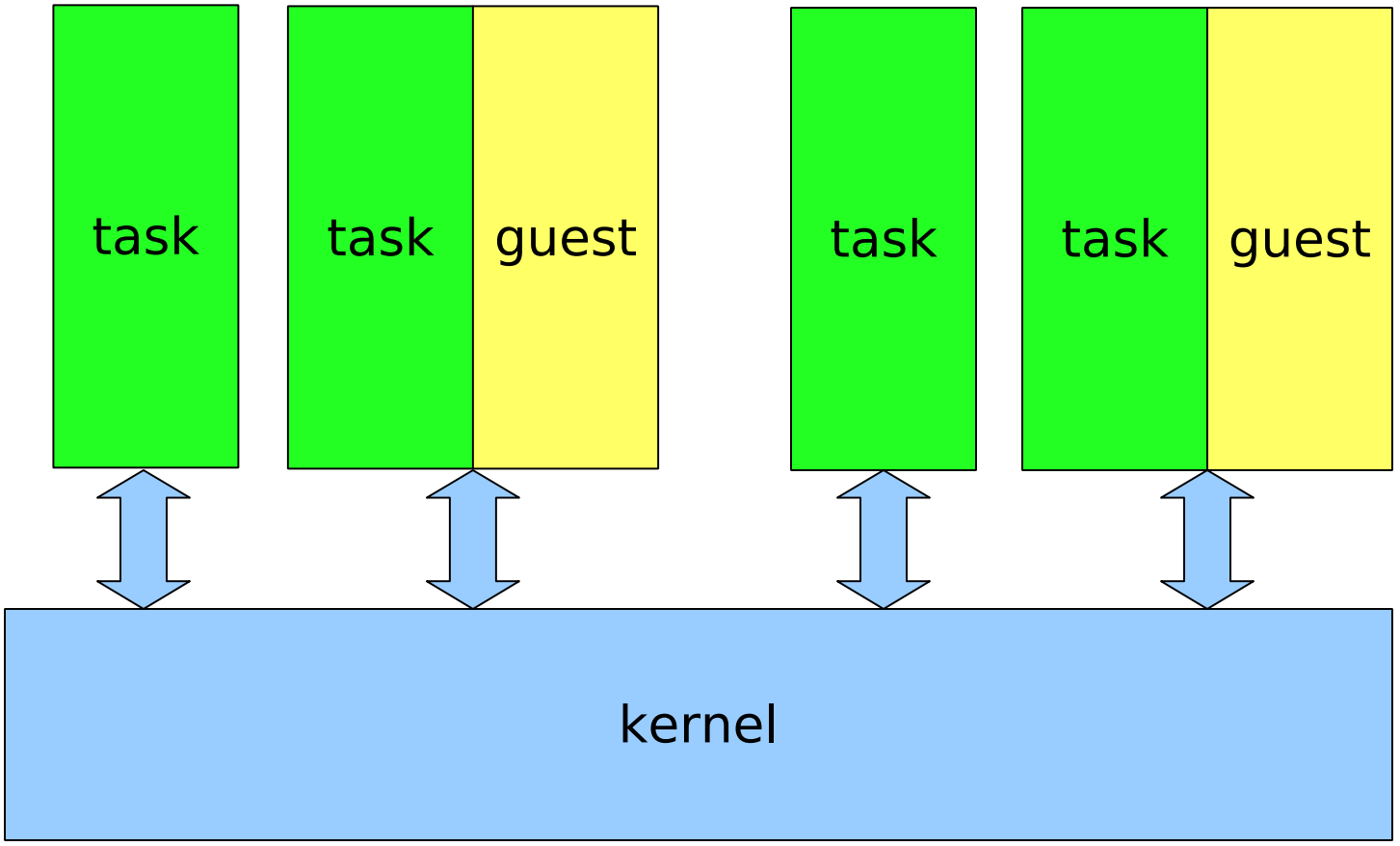
Copyright © 2007 Qumranet, Inc. All rights reserved.

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

Copyright © 2007 Qumranet, Inc. All rights reserved.

KVM Process Model



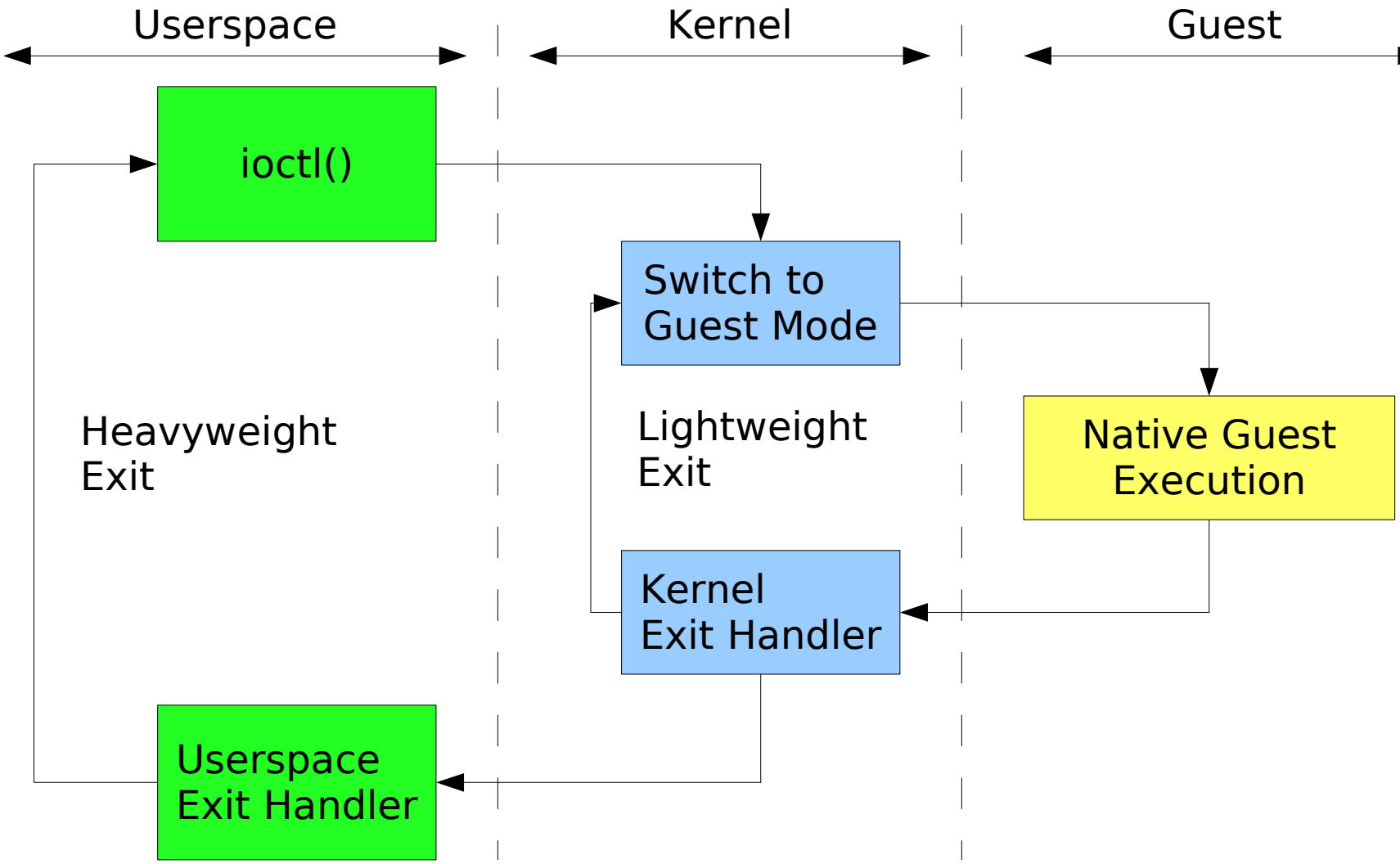
Copyright © 2007 Qumranet, Inc. All rights reserved.

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

Copyright © 2007 Qumranet, Inc. All rights reserved.

KVM Execution Model



Copyright © 2007 Qumranet, Inc. All rights reserved.

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)

Copyright © 2007 Qumranet, Inc. All rights reserved.

What's handled in the kernel?

- CPU virtualisation (special instructions)
- MMU virtualisation
- Local APIC, PIC, and IOAPIC
- (planned) paravirtualised network and block device
- (planned) paravirtualised guest kernel support code

Copyright © 2007 Qumranet, Inc. All rights reserved.

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

Copyright © 2007 Qumranet, Inc. All rights reserved.

KVM on other architectures

- s390
  - IBM mainframes: hypervisor is a must
  - WIP
- IA-64
  - Patches ready for review
- x86
  - kvm-lite
- Embedded PowerPC
  - Architecture support for hypervisor
  - WIP

Copyright © 2007 Qumranet, Inc. All rights reserved.

Paravirtualisation

- Modifying guest OS for performance
- Virtio
  - Common drivers for all hypervisors
  - Hypervisor-specific backend
  - KVM backend in progress
  - Faster performance
  - Efficient block, net drivers
- PV DMA
  - Pass through Ethernet devices

Copyright © 2007 Qumranet, Inc. All rights reserved.

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
- Also ported to FreeBSD

Copyright © 2007 Qumranet, Inc. All rights reserved.

KVM Pros

- Leverages Linux scheduler, memory management, I/O
- No scheduler involvement for I/O
- Uses existing Linux security model (can run VM as ordinary user)
- Uses existing management tools
- Power management
- Guest memory swapping
- Real-time scheduling
- Leverages Linux development momentum

Copyright © 2007 Qumranet, Inc. All rights reserved.

Release Philosophy

- Development snapshots every 1-2 weeks
  - Release early and often
  - Features introduces 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

Copyright © 2007 Qumranet, Inc. All rights reserved.

KVM is Developer-friendly

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

Copyright © 2007 Qumranet, Inc. All rights reserved.

Future

- Consolidate various virtualisation solutions existing in the kernel
- More architecture support
- More hardware features support
- More paravirtualisation support
- Improve guest scaling
- Support for management layers like libvirt

Copyright © 2007 Qumranet, Inc. All rights reserved.

Do Read

- drivers/kvm/*
- KvmForum2007 wiki page on http://kvm.qumranet.com
- kvm-devel@lists.sourceforge.net
- virtualization@lists.osdl.org

Copyright © 2007 Qumranet, Inc. All rights reserved.

Thank You