Beyond the Hypervisor
A Technical Roadmap for Open Virtualization, Linux, KVM

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

Saturday, February 22, 2014

mdday@us.ibm.com
IBM and Virtualization

- IBM leads introduction of Xen to Linux
- Virtualization on IBM mainframes
- Virtualization on POWER
- Intel adds x86 hardware virtualization
- KVM goes upstream
- Red Hat & IBM start KVM investment
- KVM becomes multi-platform
- OVA/oVirt
- PureSystems
- System x & Blade Center
- Power Systems

IBM and Virtualization

A brief history of virtualization

1964

1997

2004

-2006

2007

2008

2009

2010

2011

Red Hat & IBM start KVM investment

Intel adds x86 hardware virtualization

Virtualization on POWER

Virtualization on IBM mainframes

Power Systems

System x &
Blade Center

KVM goes upstream

OVA/oVirt

PureSystems

KVM becomes multi-platform

IBM leads introduction of Xen to Linux

1970
KVM’s Unique Relationship with Linux
KVM’s Unique Relationship with Linux (cont’d.)

- KVM Kernel Modules unlock Hardware Virtual Machine Monitor (VMM)
  - On Power (PPC64) this is a firmware VMM
- Transforms Linux into a Hypervisor
  - Kernel Engages in a Co-processing Relationship with KVM
- KVM Uses Linux Resources
  - Scheduler, Drivers, File System, Memory Management
- KVM Benefits from Linux Scalability, Quality, Maturity
Beyond The Hypervisor - Into the Kernel

- Functions Spanning Kernel, Qemu, and Userspace
  - Virtual Function Input/Output (VFIO)
    - Brings order and relative simplicity to pci pass-through and SR-IOV
    - Enables GPU pass-through
  - NUMA Mirroring, Migration, and Control
    - Guest NUMA zones mirror the host’s physical NUMA zones
    - Pin guest virtual CPUs to physical CPUs, memory nodes to physical memory nodes.
    - Automatic NUMA balancing and migration
  - IOeventfd
    - fast event signaling from the kernel to Qemu - for virtual interrupts and other events
Beyond the Hypervisor - to the Data Center

- OpenStack is Driving the KVM Roadmap
  - Clustered File System Integration
    - Distributed Block Service
  - Hot-plug Support for Most Resources
    - OpenStack convention of booting a guest with temporary resources, then hot-plugging permanent resources once the guest is on-line
  - Improvements in Migration Speed
  - Support for Network Function Virtualization
KVM and OpenStack

- KVM is the Choice of Over 95% of OpenStack Clouds*
  - KVM provides the Default development environment for OpenStack
  - Easy to get - several reliable Linux/KVM/OpenStack distributions
  - Scalable, Efficient, Economical

*Source: IDC White Paper, sponsored by IBM and Red Hat, KVM: Open Virtualization Becomes Enterprise Grade, February 2013
Beyond the Hypervisor - KVM and OpenStack

- OpenStack Obscures the Hypervisor - Which is Seen as a Benefit
- OpenStack Requires a Hypervisor to be fully functional
- This is a Great Scenario for KVM, which Fulfills the Hypervisor Role and Doesn’t Get in the Way
  - Breaks the “Vendor Lock-in”
- KVM’s Fast Development Cycle Keeps it Up-to-Date with New OpenStack Features

*Source: IDC White Paper, sponsored by IBM and Red Hat, KVM: Open Virtualization Becomes Enterprise Grade, February 2013
Beyond the Hypervisor - Into the Processor

CPU Cycles To Execute a Bare VMExit

Saturday, February 22, 14
Beyond the Hypervisor - Into the Processor

- Virtualization Functions Pulled Into Hardware Over the Years:
  - Shadow VMCS - tracking of VMCS state in hardware
  - Shadow Page Tables - Overtaken by Extended Page Tables (except during Migration). This reduced 90% of the VMExits in some workloads.
    - Management of Guest Virtual to Host Physical page mapping
    - Timer interrupt - APIC Virtualization
      - “Tickless Mode” which reduces timer interrupts benefits virtualization performance at least as much as it reduces power consumption
Beyond the Hypervisor - Into the Processor

- VMCS Shadowing
  - “Can you run VMware as a guest?”
  - Trap expansion on vmread/vmwrite

- The Turtles Project
  - [https://www.usenix.org/legacy/event/osdi10/tech/full_papers/Ben-Yehuda.pdf](https://www.usenix.org/legacy/event/osdi10/tech/full_papers/Ben-Yehuda.pdf)

- Feature Validation

- Utility Processors - On 390

- Utility Processors - On x86?

- Today: VMCS Shadowing, APIC virtualization, Nested Virtualization
Pushing KVM Into Advanced Workloads - High-Performance Computing

- Advancements in Hardware Support for KVM
- $2^{10}$ Increase in VMExit Performance since 2005
- 160 Cores, 4 TB Per Host Supported
- Highest Ever SPECVirt Score - 432 VMs
- 1.5 Million IOPs within a single Guest
• Read-Copy-Update
  

• Helped Kernel Obtain Impressive Scalability

  https://github.com/bonzini/qemu/tree/rcu

• Qemu In the Process of Becoming Asynchronous and More Scalable

  • Also Sequence Locks, Finer-grained locks
Beyond the Hypervisor - Block I/O

- **VFIO - Virtual Function Input/Output**
  - VFIO now supports setting CPU affinity on MSI interrupts.
  - SCSI Devices as well as Ethernet/RDMA and Fiberchannel

- **virtio-blk-dataplane**
  - Converts Block I/O to an Asynchronous Thread

- **ivshmem - Nahanni shared memory transport**
  - Win for HPC but also Applicable to Workloads that Move Bulk Data into and out of Main Memory

- **RDMA - Remote Direct Memory Access**

- **Gluster FS - Integration, new translators**
Block I/O Performance

Single Virtual Machine
Direct Random I/Os at 4KB Block Size
Host Server = Intel E7-8870@2.4GHz, 40 Cores, 256GB

- RHEL 6.4 KVM w/ virtio-blk-data-plane: 1,577,684.0 I/Os Per Second (IOPS)
- RHEL 6.4 KVM w/ PCI Pass-through: 939,199.0 I/Os Per Second (IOPS)
- Existing KVM w/ virtio-blk: 147,365 I/Os Per Second (IOPS)
- VMware vSphere 5.1: 1,059,304 I/Os Per Second (IOPS)
KVM - The First Multi-Platform Hypervisor?

- x86_64
- s390
- PPC
- ARM
- Others
Beyond the Hypervisor - Kimchi

https://github.com/kimchi-project/kimchi
Beyond the Hypervisor - Virt-Manager

http://virt-manager.org
Beyond the Hypervisor - oVirt

http://www.ovirt.org