Virtualization Stack
Virtualization Stack

QEMU | Guest

KVM | Linux

Hardware
QEMU

- Creates the machine
- Device emulation code
  - some mimic real devices
  - some are special: paravirtualized
- Entire guest is contained within QEMU
- Uses several services from host kernel
  - KVM for guest control
  - Linux for resources
- Runs unprivileged
KVM

- Do one thing, do it right
- Linux kernel module
- Exposes hardware features for virtualization to userspace
- Emulates some devices
  - e.g. APIC
- Enables several features needed by QEMU
  - like keeping track of pages guest changes
KVM Execution Model

QEMU

ioctl()

Heavyweight Exit

Userspace Exit Handler

Switch to Guest Mode

Lightweight Exit

Kernel Exit Handler

Kernel

Native Guest Execution

Guest
Linux

- Host
  - seccomp
  - SELinux
  - Disk IO
  - Network IO
  - Transparent HugePages (THP)
  - Kernel Same-page Merging (KSM)
- Guest
  - Paravirtualized drivers
Virtualization Stack

- QEMU
  - pxe
  - seabios
  - ovmf
- KVM
  - Linux
- Hardware
Virtualization Stack
libvirt

- Management of VMs, storage, network
- Provides a stable API
- Remote management
- `virsh` – command-line interface
- `cgroups`
- `sVirt`
- Makes it possible for QEMU to run unprivileged
  - Opens files, connections and passes them on to QEMU
Virtualization Stack

- libguestfs
- oVirt
- OpenStack
- virt-manager
- GNOME Boxes
- libvirt
- QEMU
- Guest
- Guest
- QEMU
- KVM
- Linux
- Hardware
KVM Today

- 6 architectures
  - x86, ARM/ARM64, MIPS, PPC, s390
- Very good performance, scalability
  - Dominating all SPECVirt benchmarks
- Many features
  - THP; hotplug (CPU, memory, devices); CPU, memory overcommit; KSM; NUMA; device assignment; live migration; block migration; live snapshots; SPICE; sVirt; seccomp; guest agents; paravirtual devices (net, block, balloon, RNG, video, input, serial, clock); stable guest ABI; cgroups-based isolation
Community
KVM Community

- Top contributors
  - Red Hat, IBM, Intel, Linaro, ARM, AMD, Google, Huawei, Fujitsu, Siemens, Samsung, SuSE, Parallels, Xilinx, Imagination Tech, ...

- Mailing lists
  - Check linux-kvm.org, qemu-project.org, libvirt.org

- IRC
  - #kvm, #qemu, #libvirt on freenode

- Annual KVM Forum / QEMU Summit meetups

- QEMU is a member of Software Freedom Conservancy (SFC)
Features Added Last Year
KVM-RT

- Run guests that have real time workloads
- Getting deterministic behavior inside guests
- Used in:
  - telco switching (voice),
  - stock trading
  - vehicle control / avionics
- Get advantages of virtualization for RT systems
KVM-RT

- Host kernel: PREEMPT_RT
- Getting configuration right
  - involves getting BIOS config right – disable SMI/SMM
- KVM & QEMU challenges with RT
  - priority of guest tasks not available to host scheduler
  - host does not know vCPU with highest priority
  - lock holders in guest not visible to host
  - no PI possible
  - tasks on vCPU not always preemptible due to emulation in QEMU
KVM-RT

• Some solutions
  • Partition host CPUs into RT and non-RT
    • Constrain guest RT CPUs to the RT pool
    • 1:1 mapping between guest vCPU and host CPU
      • Avoids starvation and deadlocks
    • keep tasks away from RT CPUs
  • disable periodic kvmclock sync
    • guest runs ntpd and keeps clock in sync
  • disable scheduler ticks when running sched_fifo / rt task
    • if not rescheduling, no need of scheduler tick
  • timer interrupts happen slightly early to offset virt overhead
Live Migration

- Postcopy
  - Uses new \texttt{userfaultfd} functionality in Linux
  - Migrates guest first; remote-page faults memory to dest
- Enhanced autoconverge
  - Dynamically throttle vCPUs to force migration of busy guests
- x86zrle
  - Send diffs of pages instead of whole pages
- Multi-thread compression
  - Compress pages before sending
System Management Mode (SMM) Support

- Special processor mode
- Has some private memory
- Secure boot stores keys in this area
- Enables secure boot of guests
- Collaboration between Linux, QEMU, OVMF, seabios
VIRTIO 1.0

- Now an OASIS spec
- Spec reached feature parity with VIRTIO 0
- New in the spec:
  - documented assumptions
  - more robust
  - more extendable
- Implementations in Linux and QEMU
- Requirement for userspace drivers and PCI express
Userspace Accelerators

- DPDK
  - vhost-user port in DPDK
- vhost-user
  - left experimental state
  - multi-queue
  - migration
PCI Express in QEMU

- Now supported - feature parity with PCI
- Migration works
- Requirement for IOMMU/userspace drivers, AER
Guest Video

- virtio-gpu
  - 2D and 3D support merged
- vGPU
  - in progress
- Device assignment
  - Works, for some hardware
  - Mostly used for compute, not video
Device Assignment

- IOMMU unmap performance improvements
- vfio-pci device request support
  - automatic unplug
- Device quirk fixes and extensions
  - reset for select AMD GPUs
- Improved tracing support and runtime debug flags
- Power management improvements
- IRQ bypass support (Intel Posted Interrupts)
- “7 gamers 1 CPU” - [https://www.youtube.com/watch?v=LX0aCkbt4lI](https://www.youtube.com/watch?v=LX0aCkbt4lI)
  - 7 VMs, each gets 1 GPU
Block layer

- blockdev-backup
  - Backing up running guests
  - Point-in-time snapshot of a disk
  - Over the network: iSCSI, NBD, ssh
- IO throttling groups
  - All disks can be made part of a group
  - Quota restrictions can be applied to entire group
- Extended IO stats
  - Helps with understanding guest behaviour and tuning
libvirt

- virt-admin
  - Set of APIs to tune libvirtd itself
    - Gather resource usage, produce stats
    - Size up/down thread pool
  - IO thread pinning
    - like vCPU pinning
  - PPC64 became first-class citizen
    - Resulted in refactored code to support multi arch
  - Can now deal with big endian guests on LE hosts
Other bits

- virtio-keyboard, virtio-mouse, virtio-tablet
- virtio-balloon can tell guest to deflate balloon on OOM
- Memory hotplug support
- Guard pages now inserted after guest RAM
  - Guard against guest-triggered buffer overflow attacks
Architecture-specific Improvements

- s390 got PCI bus support
- ARM
  - hosts and guests support > 8 CPUs
  - GICv3 support (virtual interrupt controller)
  - dirty page tracking
- x86
  - VT-d emulation (in progress)
  - nested virt improvements
  - split irqchip
- PPC
  - CPU, memory hotplug
  - H_RANDOM hypercall
In Progress

- virtio-gpu 3D SPICE integration
- “Native” Hyper-V paravirtualization (vmbus)
- blockdev-backup
  - will gain incremental backup
  - preserve state across restart and live migration
- Many more features everywhere across the board