Porting bhyve to SmartOS

Patrick Mooney
Software Engineer @ Joyent
bhyvecon Ottawa 2019

Background on SmartOS

- Downstream from illumos (OpenSolaris lineage)
- Designed for production workloads in the datacenter
- Tenants: native zones, LX zones, hardware virtualization
- Basis for Joyent Triton (and Joyent Manta)

History of KVM in SmartOS

- Ported Linux KVM to SmartOS beginning in 2010/2011
- Both KVM and qemu are kept out-of-tree
- Effectively forked as of 2012
  - Limited platform (Intel + EPT only)
  - Limited features (no pci-passthru, slow virtio-net)
  - Minor bug fixes as required

Rationale for porting bhyve

- More focused scope
- Tighter integration with OS (being in-tree)
- Community participation
- Features (pci-passthru, CPU capabilities, etc)

Code from Pluribus

- Received in late Sept. 2017
- Consisted of bhyve circa 2013 (with a mix of patches)
  - Built-in EPT (prior to vmspace-driven approach)
  - SMP support, but never started additional CPUs
  - Accelerated virtio-net (viona), non-functional
  - Not targeted for recent illumos
- Booting Linux guest by mid Oct. 2017

Update to “modern” bhyve

- Targeted FreeBSD 11.1 bhyve as upstream
- Wholesale copy-over or walking commits forward to sync
- Needed new EPT implementation conforming to vmspace interface
- Semi-functional by late Nov. 2017
- Relying on uefi ROM for boot

Stabilization

- Zones integration
- Coexistence (read: exclusion) with KVM
- Merged into illumos-joyent#master late Feb. 2018
- Testing pci-passthru with GPGPU
- Update guest images for bhyve/KVM
- Help guest memory allocation WRT pressure from system and ARC

Stabilization (ongoing)

- Real coexistence between KVM and bhyve
- Bug fixes for various guests (virtio-block and UART for Windows)
- Performance issues (ZFS cache flush updates)
- Wiring up viona for hardware checksum offload and LSO
- RVI implementation to plumb SVM support
- mdb support for bhyve

Porting Challenges: Kernel Preemption

- No perfect analog to `critical_enter()` in illumos
- Some unexpected operations can yield CPU
- Thread `ctxops` from illumos help solve this

Porting Challenges: Timers and Spinlocks

- Callouts for bhyve implemented with illumos cyclics
- Stays on a CPU once programmed there
- Migration/localization required for performance
- Took some iteration to get right (see: OS-7012)

Divergence

- Handling TSC offsets between CPUs
- Saving additional MSRs which are not global like FreeBSD
- Kernel driver interface into bhyve instances
  - IO port hooks, interrupt delivery, access to guest memory
- Certain functions controlled by Hypervisor Multiplexing API
  - VMX init, VPID allocation, FPU saving
  - Allows real coexistence with KVM and VirtualBox (kind of)

Futures

- AMD AVIC (accelerated virtual APIC)
- Dynamic per-CPU resource allocation (under way through Rod's work)
- Richer driver API
- Bootrom options (seabios, as a CSM?)
- Adopting KVM/qemu-isms for wider support (kvm-clock, fw interface)
- Plumbing virtio-vsock
- Configuration enhancements (key-value args)

Thanks

@pfmooney on twitter

pmooney on Freenode
(in #bhyve and #smartos)