KVM Status Report 2019

Christian Borntraeger
IBM
IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.
If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.

The following are trademarks or registered trademarks of other companies.
• Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
• Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.
State of KVM

- 6 “old” architectures
  - All in-tree, all actively maintained
  - ARM, ARM64, MIPS, PPC, x86, s390
- 1 “new” architecture on the horizon
  - RISC-V
  - Code looks pretty good, waiting for architecture stabilization
  - See yesterday's talk “The Hype Around the RISC-V Hypervisor”
- 2 core maintainers: Paolo Bonzini, Radim Krcmar
- Arch maintainers for all architectures
  - Loss of Christopher Dall as ARM co-maintainer
Usage of KVM

- QEMU/libvirt is the main choice for data center virtualization, IaaS, KubeVirt
  - Work in progress to cover new usecases in QEMU
- A zoo of custom userspaces for special purposes
  - lkvm (native Linux kvm tool)
  - Crosvm
  - Amazon Firecracker (see today's talk Firecracker: Lessons from the Trenches)
  - RustVMM (see Friday's talk Playing Lego with Virtualization Components)
  - ....
- KVMs usage by cloud providers is not tied to a single userspace stack (e.g. Amazon Nitro)
KVM as a building block

- KVM is used as a building block
- for container isolation
  - Kata Containers (via QEMU)
  - runq (via QEMU)
  - gVisor can optionally use KVM as backend
- KVM/QEMU/libvirt is used to provide classic virtual machines in k8s via kubevirt
  - (see fridays KubeVirt Community Update)
Commits in each release (long term)

KVM is an active and growing project
Commits 4.20-rc1..5.4-rc1

- 1116 non-merge commits
- 117 merge commits for kvm related files
- 319 commits have “Reviewed-by:”
- 87 commits have “Acked-by:”
- 222 commits have “Fixes:”
- 113 commits have “Cc: stable..”

Top authors
227 Sean Christopherson
78 Paolo Bonzini
48 Vitaly Kuznetsov
45 Marc Zyngier
41 Dave Martin

Top reviewers
42 Jim Mattson
33 Sean Christopherson
27 Cornelia Huck
27 Andrew Jones
26 David Hildenbrand

Top repairmen
62 Sean Christopherson
15 Vitaly Kuznetsov
12 Paolo Bonzini
12 Marc Zyngier
11 Paul Mackerras
Companies

- Commits from >20 companies
  - Redhat: Overall maintainer
  - IBM: Power, IBM Z
  - ARM
  - Oracle
  - Google
  - Tencent Cloud
  - Amazon
  - Microsoft
  - SUSE, Oracle, Huawei, Virtuozzo, Cavium, Samsung, MIPS, Canonical, etc.
Highlights overall

- memcg accounting (core code and x86)
- Dirty pages tracking improvements
- Documentation conversion txt → rST
Highlights Testing

- In-kernel selftests for KVM now for arm64, s390, and x86
- Kvm-unit-tests for arm, power, s390 and x86 (using QEMU)
Highlights x86

• Nested virtualization enabled by default!
  - Lots of accuracy improvements and new tests
  - Hyper-V enlightened VMCS
  - Still some work to do on AMD
• More PV
  - yield to IPI target
  - guest-side interrupt polling
  - Hyper-V enlightenments

• "Thin" virtualization
  - C-state MSRs
  - guest memory not backed by struct page
  - PMU event filtering
  - Exitless timers
• HW feature enablement
• Many optimizations and cleanups
Highlights ARM/ARM64

• Security
  - Guest entry hardening
  - cache sanitization for 32-bit guests
  - allow side-channel mitigation status to be migrated

• Guest features
  - Improved guest IPA space support (32 to 52 bits)
  - direct physical timer assignment
  - support for SVE and Pointer Authentication in guests
  - support for chained PMU counters in guests

• Scalability
  - support for 512 vCPUs
  - large PUD support for HugeTLB
  - ITS translation cache

• Hygiene
  - PMU fixes and improvements
  - RAS event delivery for 32bit
  - improved SError handling
  - standardise most AArch64 system register accesses to msr_s/mrs_s
Highlights s390

• I/O
  - vfio-ap crypto HSM virtualization

• Hardware
  - Z15 cpu features for guests
  - Processor subfunctions for cpu models
  - host program identifier

• Hygiene
  - Optimize page table locking
  - Interrupt cleanup
  - Selftest fallout ioctl hardening, bugfixes
Highlights POWER

• Nested HV KVM support for radix guests on POWER9
  − Migration, arbitrary levels of nesting and PCI pass-through are all supported.

• Support for guests accessing the XIVE interrupt controller directly
  − This reduces interrupt latency and overhead
  − The XIVE interrupt controller was new on POWER9

• Preliminary support for Protected Execution facility running...
  − as a secure guest under the Protected Execution Facility (PEF, also known as Ultravisor)
  − as a host in a PEF-enabled system.

• Optimizations and performance improvements in PCI pass-through and memory management.
Highlights MIPS

• Cleanups
• Optimize tlbwr
Upcoming RISC-V support

- Patch set on the list by Western Digital
  - Support for both RV32 (32bit) and RV64 (64bit) Host
  - No RISC-V specific KVM IOCTL
  - Minimal possible KVM world-switch
  - Full save-restore via KVM vcpu_load()/vcpu_put()
  - FP lazy save/restore
  - KVM ONE_REG interface for user-space
  - Timer and IPI emulation in kernel-space
  - PLIC emulation is done in user-space
  - Hugepage support
  - SBI v0.1 interface for Guest
  - Unhandled SBI calls forwarded to KVM userspace
• Enjoy the rest of KVM forum!
THANKS

Christian Borntraeger
IBM  borntraeger@de.ibm.com