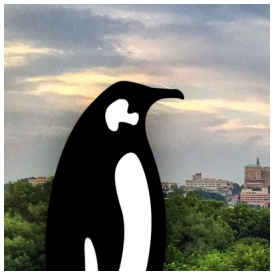

Contribution ID: **253** Type: **not specified**

Multi-KVM Abstract

Tuesday, 14 November 2023 10:15 (25 minutes)

Problem Statement

Rolling out KVM bug fixes and feature upgrades requires unloading KVM modules, which disrupts guests.

Multi-KVM is a proposal to allow multiple, independent KVM modules to be loaded, unloaded, and run concurrently on the same Linux host to:

- Upgrade and rollback KVM without disrupting running VMs and other KVMs on the host.
- Enable running KVM modules with different parameters on the same host.
- Facilitate easier A/B testing for KVM.

Proposal & Objectives

The proposed solution for multi-KVM requires refactoring and "privatizing"KVM code, mainly for conflict resolution between the various KVM modules running on the host.

The approach we plan to take is:

1. Make all KVM-owned data structures and values KVM-only on x86 (and to a lesser extent on other architectures) so that the structures/values don't need to be fixed across KVM modules.
2. Fold KVM vendor modules (kvm_intel.ko and kvm_amd.ko) into KVM, removing any exports along the way to avoid DLL hell with vendor modules on x86.
3. Extract shared system resources out of KVM, e.g. ASID management, into a new base module (proposed name VAC - Virtualization Acceleration Code). This module will run as a prerequisite to KVM, and will manage the shared state of all the KVM modules on the system.
4. Allow the builder to assign a unique name to KVM modules and devices at compile time.
5. Make multi-KVM opt-in, and have KVM be fully backward compatible when multi-KVM isn't utilized.

Since these are fundamental changes in the way the KVM code is laid out, we would like to elicit feedback from the broader KVM community on our proposal.

Primary author: GHULATI, Anish

Co-author: CHRISTOPHERSON, Sean (Google)

Presenters: GHULATI, Anish; CHRISTOPHERSON, Sean (Google)

Session Classification: KVM MC

Track Classification: LPC Microconference: KVM MC