KVM
Scaling to Infinity .... and beyond

Jes Sorensen
<Jes.Sorensen@gmail.com>

Japan Linux Symposium
October 22, 2009
Tokyo, Japan
Scaling agenda

- Definitions
- Background
- BIOS, what BIOS?
- KVM Scalability
- Achievements
- What is next?
Definitions

Infinity

- ACPI 1.0/2.0 – 8 bit indices, limited to 255 CPUs etc.

... and beyond

- ACPI 3.0/4.0
  - 32 bit indices
  - Proximity domains
  - SRAT (NUMA)
Why scale the guest?

- 1:1 Guest:Host mapping for new systems
  - x86 systems are finally growing!
  - Run legacy OS on modern hardware, compatibility, certification
- Simulate larger system on small system
- Testing, testing, testing
Devel & testing issues

- Lack of access to large x86 systems
  - multi-core, HT with 16/32 threads now available
- Started out using 96 CPU ia64 system as development vehicle
- One cannot emulate lack of CPU power!
  - Over-committing is a real problem
Virtual machine components

- A KVM virtual machine consists of 3 main components:
  - KVM Kernel Module
  - QEMU
  - A guest BIOS / firmware image
- All components have scalability issues:
  - Fixed sized arrays, buffers and tables
  - Pre-compiled BIOS components (ACPI)
KVM changes summary

- Change KVM_MAX_VCPUS
- Remove some static arrays
- Use nr_online_vcpus for walking arrays
- ia64 specific changes
  - ITC drifts, emulate using slower stable clock source
  - KVM module data block dynamic sizing. Module uses separate mapping from kernel
QEMU changes summary

- Remove static vCPU arrays
- Introduce `-smp, max_cpus=<X>` sub-argument to specify upper vCPU limit
  - Used by BIOS to generate / select ACPI tables
  - Required for future 'KVM_CREATE_CONTEXT' ioctl
BOCHS BIOS

- QEMU-KVM and QEMU+BOCHS versions
- Presents system configuration to guest:
  - CPUs, I/O, memory
- Must preserve compatibility with certain legacy OSes (no names needed!)
- Older Windows only handles max 15 CPUs
- Only table space for 128 CPUs!
- Requires BCC + AS86
BOCHS comparison

QEMU-KVM
- Static SSDT+DSDT
- Picks DSDT based on maxcpus: 15 vs large Processor(#) #
- CPU + PCI hotplug
- Branched far from upstream

QEMU-BOCHS
- Dynamic SSDT
- Processor() declarations based on -smp
- No hotplug
- Complicated maintenance
SeaBIOS

- Why yet another BIOS?
  - Maintained git source tree, refactored BOCHS source, easy to modify and extend
  - Compiles with GCC
- Added maxcpus support
- Dynamic build of PCI IRQ table
- Lacked some features from QEMU-KVM
  - Hotplug
- No data structure limit on # of CPUs
- Next QEMU release will default to SeaBIOS
Results

- KVM boots 192 vCPUs on ia64
  - EFI guest firmware image has memory layout problems beyond 192
- x86 – 2 socket Nehalem (16 threads)
  - 44 vCPUs with udev
  - 248 vCPUs without udev(!)
- Over-commit causes major problems when spinlock contention
  - udev in particular a problem!
Current status

- Support for ACPI 2.0
  - Processor declarations
  - APIC support
- Basic SRAT (NUMA) support
  - memory, but not processor
- x2APIC being worked on, required for ACPI 3+ support
- 255 CPUs current theoretical limit
Todo

- x2APIC and ACPI 3+ table support
- Migration issues
  - How to handle maxcpus > KVM_MAX_VCPUS on destination host
  - NUMA affinity
- Implement KVM_CREATE_CONTEXT ioctl
- Support dynamic complex ACPI tables vs static pre-defined tables?
• Questions?