Xen/ HVM SMP Status

Xin Li
Eddie Dong
Extend Xen to support SMP HVM guest

Domain 0
- XenLinux
- Control Panel (xm/xend)
- Native Device Drivers

Guest VM (VMX)
- Unmodified OS
- Guest BIOS
- Virtual Platform
- vCPU
- vCPU
- VMExit
- VMExit
- ctxt/VMCS
- vLAPIC
- vLAPIC
- vIOAPIC
- ctxt/VMCS

Event channel
- Callback
- Hypercall

Control Interface
- Scheduler
- Event Channel
- Hypercalls
- Processor
- Memory
- I/O: PIT, APIC, PIC, IOAPIC

Xen Hypervisor

Intel
Open Source Technology Center
Extend Xen to support HVM SMP guest

• SMP configuration / Detection
  ▪ ACPI MADT configuration / MP table

• vLAPI C / vI OAPI C

• Per vCPU IO event channel
  ▪ Each vCPU has an IO request slot in the IO shared page

• Guest AP Startup
  ▪ In INIT-SIPI-SIPI sequence:
    ▪ hvm_bringup_ap ➔ hvm_init_ap_context
    ▪ VMX side : dx ← lapic id / bx ← trampoline vector

• Enhance shadow page table
Status

• **SMP guests supported**
  - Linux
    - RHEL4/FC4/FC3, SLES9/10
  - Windows XP
    - 32bit SMP windows can install and boot.

• **Workload tested**
  - Kernel Build, CPU2000, SysBench, Crashme

• **With old shadow code**
  - All guest/host paging levels combinations are working
  - 4 vCPUs 64 bit SMP guest run crashme, LTP and kernel build simultaneously for 72 hours.

• **Instability with new shadow code**
  - Kernel Build on 32 bit guest/32 bit Xen fails sometimes
Issues & To-Do

• Improve device model parallelism
  ▪ Create thread(s) to handle asynchronous and time-consuming jobs.
    ▫ Like IDE DM today.
  ▪ Each vCPU should have its own polling loop on IO events.
    ▫ Currently only one polling loop for a HVM domain.

• Current XenTrace doesn’t support HVM SMP guest
  ▪ We have a patch that works

• With more vCPUs, host LAPIC timer interrupts are increasing too fast.

• Guest time keeping issue
Guest time keeping - Challenges

- **Guest time**
  - Periodic timer is maintained by Interrupts from PIT/RTC
  - Monotonically increasing time is represented by a timer like TSC/HPET.

- **Time from different sources on real platforms are synchronized.**
  - Linux assume PIT time and TSC time are synchronized.
    - PIT interrupt handler adjust lost ticks by comparing with TSC time back & forth
    - If guest PIT time lose synchronization with TSC time for long time, clock fall back.
      - “Losing too many ticks”
      - “TSC cannot be used as a timesource” …
Guest time keeping - Current Status

• **Synchronizing guest TSC with guest PIT**
  - Guest time jumps only at PIT interrupt injection time.
  - Guest time frozen when the domain is de-scheduled.

• **But synchronizing guest time among VPs is almost impossible.**
  - VPs are scheduled individually
  - A periodic time interrupt delivered to a non-active VP or IRQ disabled VP may block the guest time forward.
    - Before the IRQ is injected, guest time is represented by last jiffies.

• **Current approach for SMP**
  - Synchronizing guest time within VP
    - Guest LAPIC time is synchronized with TSC.
  - Binding platform timer (PIT/RTC) interrupt to BSP
    - PIT or RTC time is synchronized with BSP TSC time too.
Issues & To-Do

• **How to synchronize guest time across VPs?**
  - Guest time jump (PIT/RTC interrupt injection) on one VP needs to IPI others for synchronization.
  - When guest time is frozen on one VP, all VP’s TSC_OFFSET need to be disabled.

• **Should the accumulated PIT/RTC interrupts be injected contiguously?**
  - Threads on other VPs may be hungry
  - A device waiting for event may be timeout due to big guest time jump

• **API C time, RTC and ACPI time are not synchronized with guest TSC yet.**