



ACPI support for HVM Guest

Winston Wang

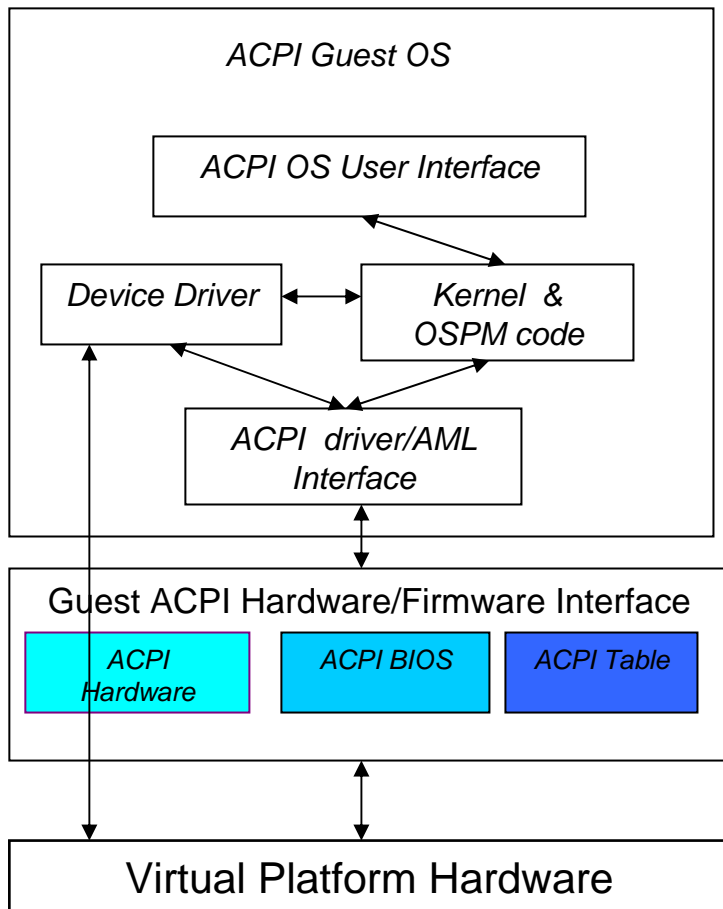
Xen Summit 2006



Goals

- **Support 32 bit and 64 bit UP and SMP Guest OS in ACPI mode**
 - Installation
 - Boot
 - Shutdown
 - Power Management
- **Focus on configuration now; will add Power Management support later on**

HVM Guest ACPI Architecture



HVM Guest environment

Guest ACPI Hardware/Firmware Interface implementation overview

ACPI Hardware: emulated by device model for piix4 ACPI bridge/ACPI registers

ACPI BIOS: E820 table ...

ACPI Table:
FADT: Fixed ACPI Description Table
DSDT: Differentiated System Description Table
MADT: Multiple APIC Description Table
...

Need PIIX4 ACPI Bridge Model

- **Why**

- PIIX4 ACPI bridge supports all ACPI registers required by the ACPI spec
- It is the most natural extension to the PIIX3 model in QEMU

- **We implemented it by**

- Populate the 82371AB PIIX4 PCI ACPI configuration space
- Register PIIX4 ACPI bridge
- Setup ACPI bridge basic PCI configuration:
 - Device ID, class ID
- Defined and register ACPI pm space IO register
- Accessed run time “reader” and “writer”

ACPI Registers We Implemented

- **PM1a Event Register Block**
 - 32 bit registers:16 bit PM1 Status register and counter Status register; base port pointed by FADT
 - Required during OS installation, boot, shutdown and power Management.
 - We simplified the implementation by adding support for the ACPI only programming model, i.e. only take care of PM Timer, Power button, RTC
- **PM1a Control Register Block**
 - 16 bit PM1a Control register, base port pointed by FADT
 - Required during OS installation, boot, shutdown and power Management.
 - We implemented SCI and System power state controls to take care of ACPI shut down

ACPI Timer Implementation

- **OS need it for profiling and ACPI SCI event**
 - 24-bit free running timer at 3.57954 MHz
 - Base port pointed by FADT
 - Required during OS installation, boot and power Management
- **Zero performance loss when emulating the free running timer**
 - An independent timer will cause severe performance loss due to timer frequency
 - We used vm-clock delta to calculate ACPI timer to avoid possible performance loss
 - Will add SCI event generation when we add power management support

HVM ACPI ACPI Table – FADT

Fixed ACPI Description Table (FADT)

- Report ACPI Hardware Register Blocks base address emulate in device model
- SCI interrupt: IRQ 9
- Ownership of ACPI hardware is OS
- Support Processor C state, WBINVD and etc
- Physical address of FACS and DSDT
- Report no support of the following
 - SMI support
 - Legacy S4 support
 - Power Management Event (PME) blocks and General Purpose Event (GPE) blocks
 - Power Button, Sleep button, RTC

HVM ACPI ACPI Table – DSDT

Differentiated System Description Table (DSDT)

- **Point to Differentiated Definition Block**
- **HVM platform configuration information details in the form of AML code implementation and configuration**
 - Report current resource setting and platform current reserved
 - Define logical processors
 - Power off support by adding `_S5` control method
- **Provide PIC and APIC mode `_PRT` table**
 - Check for `_PIC` method input to know the OS operating mode
 - Return PCI Interrupt Routing Table for PIC and APIC mode
 - Based on virtual platform and chipset interrupt routing.
 - Use the device mode IOAPIC emulation information to form the actual interrupt number and interrupt pin
 - Global Interrupt Base of each IOAPIC defines the starting interrupt number

Current Status and Next Step

- **Current tested ACPI OS**

- Linux OS when ACPI=on
- Windows
 - 32 bit and pae mode XP/SP2 UP and SMP
 - Windows 64 bit
 - MS Vista

- **Next step**

- Windows installation ACPI mode checking
 - Turn off non ACPI guest firmware information: \$PIR, MPS table if guest boot in ACPI mode
- Cover ACPI HCT test support
 - Clean up wrong ACPI data and NVS reserve range
- Power Management support
 - S3 and S4
 - Add missing features required for power management