Hardware Based Virtualization Technologies

Elsie Wahlig
elsie.wahlig@amd.com
Platform Software Architect
Outline

- What is Virtualization?
- Evolution of Virtualization
- AMD Virtualization
- AMD’s IO Virtualization
- Xen
What is Virtualization?

Virtualization is the pooling and abstraction of resources in a way that masks the physical nature and boundaries of those resources from the resource users.
What Problem is Virtualization Solving?

● Problem 1: You have ancient x86 operating systems and legacy applications running on even older hardware
  - Ancient hardware is distributed across enterprise and ready to die
  - No modern replacement for these legacy OS and Applications
  - Need to move this critical software to new hardware

● Problem 2: Your high-performance processors are under utilized
  - You want to run more applications on this hardware
  - But each application may need different Operating System
  - Or each application needs to be fully isolated from each other

● Need a solution that runs multiple, incompatible, x86 OS & Apps side-by-side on the same processor system

That solution is called “virtualization”
Why ...

- Operating Systems are selfish
  - They expect to own all resources of a machine
  - They aren’t designed to share a machine with another OS

- In order to “fool” these OSes into running side-by-side, Virtualization technology is designed to make each OS think it is running in its own little machine
  - We virtualize the resources, state, and execution of a physical system and assign each OS its own private set

- The hardware based virtualization gives each OS a chance to run on real hardware
  - During that time, the OS’s private (“virtual”) set of resources are restored/saved to the hardware’s real set
  - Hardware based virtualization ensures that an OS’s private set is unaffected by the operation of another OS
Virtual Machine Approaches
Carve a System into Many Virtual Machines

Hosted Virtualization

<table>
<thead>
<tr>
<th>App</th>
<th>App</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guest OS</td>
<td>Guest OS</td>
</tr>
</tbody>
</table>

Virtualization Software

Host Operating System

X86 Hardware

- Virtualization software manages resources between Host and Guest OS’s
- Application can suffer decreased performance due to added overhead

Hypervisor-based Virtualization

<table>
<thead>
<tr>
<th>App</th>
<th>App</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guest OS</td>
<td>Guest OS</td>
</tr>
</tbody>
</table>

| Service Guest |

Xen Hypervisor

AMD64 with AMD Virtualization

- Virtualization Software / Hypervisor is the host environment
- Enables better SW performance by eliminating some of associated overhead
- If Hardware is available, the Hypervisor can be designed to take advantage of it
Challenges Of Virtualizing x86

- Overhead of software techniques
  - Operating systems want zero-based, contiguous physical memory
  - Shadow page table management, adds extra memory requirements
- Requires complex techniques to wrap privileged instructions
  - Para-virtualization (requires modified guest OSes)
  - Ring compression
  - Binary translation
  - IO device emulation
- Guest may not see hardware
- No Hardware enforced memory protection
- IO device drivers forced to primary domain
- DMA capable devices that corrupts memory
**Virtual Machine Approaches**

**Divide a computer into many virtual machines**

- **Problem:** applications need less than a full processor, computers are underutilized, applications can interfere with each other
- **Solution:** partition a computer into several independent machines that can support different OS’s and applications concurrently
- **Benefit:** more efficient use of hardware

**Unite many computers into a virtual machine**

- **Problem:** computers are configured into cluster or grid architecture, workloads are peaky, applications occasionally needing larger capacity
- **Solution:** combine several computers into a large machine than can be reconfigured as needed to run required applications
- **Benefit:** can resize hardware to fit use demands
Outline

- What is Virtualization?
- Evolution of Virtualization
- AMD Virtualization
- AMD’s IO Virtualization
- Xen
Evolution Of Virtualization

- 1970’s: Virtualization in IBM VM/370
- 2000’s: x86 and AMD64 enter the Datacenter
Eliminating Architectural Bottlenecks

Legacy x86 Architecture
- 20-year old front-side bus architecture
- CPUs, memory, I/O all share a bus
- Major bottleneck to performance
- Faster CPUs or more cores ≠ performance

AMD64 technology with Direct Connect Architecture
- Industry-standard AMD64 technology
- Direct connect architecture reduces FSB bottleneck
- HyperTransport™ interconnect for high bandwidth and low latency
x86 Virtualization Enabling Technology

- Reduce and remove overhead in virtualizing x86
- Intercept based Virtualization
  - Selectively intercept both exceptions and instructions
- Processor Guest Mode
- Control Data Structure (VMCB)
- Paged Real Mode
- Secure Kernel Support (skinit)
- External Access Protection (DEV)
- Nested Page Tables
- Customizable Interrupts support
Outline

- What is Virtualization?
- Evolution of Virtualization
- AMD Virtualization
- AMD’s IO Virtualization
- Xen
How does a world switch work?

- Virtualization based on **VMRUN** instruction
- VMRUN executed by host causes the guest to run
- Guest runs until it exits back to the host
- World-switch: host → guest → host
- Host resumes at the instruction following VMRUN
AMD Virtualization Flow Example

World Switch

HOST

VMRUN

#VMEXIT
Intercept 7b

VMRUN

#VMEXIT
Intercept 4e

GUEST

I/O Intercept

Page Fault Intercept

Host sets up intercepts in VMCB
AMD’s Hardware Enabled Virtualization

**Guest OS**
- Guest instructions run native speed to CPU w/ no ring compression

**Hypervisor**
- Intercept PRIV instruction or register access?
- Intercept Interrupt?

**AMD Processor**
- AMD64 Core
  - Intercept PRIV instruction or register access?
- VM Tuning: Tagged TLB, etc
- Memory Access

**VMCB struct**
- Intercept Interrupt?

**VMRUN**
- Memory Controller
  - Allow access?
  - Device Exclusion Vector

**Physical Resources**
- Memory Controller
## Para-Virtualization vs. Full Virtualization

### Para-Virtualization
- Guest O/S and or drivers must be modified to run!
- Guest cooperates with host/VMM
  - e.g., non-contiguous non-zero based physical memory
  - e.g., custom devices

### Full Virtualization
- Runs unmodified off-the-shelf guests
- Export "full" x86 & platform to unmodified guest
  - guest physical space appears zero-based, contiguous
  - guest uses off-the-shelf devices (whether real or simulated)

---

<table>
<thead>
<tr>
<th>Domain 0</th>
<th>Domain 1</th>
<th>Domain 2</th>
<th>Domain 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paravirtual</td>
<td>Paravirtual</td>
<td>Unmodified</td>
<td>Unmodified</td>
</tr>
</tbody>
</table>

Hypervisor with AMDV hardware enabled
Outline

- What is Virtualization?
- Evolution of Virtualization
- AMD Virtualization
  - AMD’s IO Virtualization
- Xen
AMD’s I/O Virtualization Technology

• AMD announced broad availability of AMD I/O virtualization technology specification on February 6, 2006

• Addresses the performance bottlenecks and security issues that can be encountered when virtualizing devices in x86-based computers

• Intended to drive efficiencies into virtualizing I/O Devices

• Represents close collaboration with our ecosystem to define a solution that is right for the AMD Direct Connect Architecture

• Specification is broadly available to developers
What are the benefits of I/O Virtualization?

- Enhanced virtualization capabilities
  - Facilitates direct Guest OS use of devices
    - *with unmodified guest OS & drivers*
    - Enables (safe) direct device access by user mode applications

- Enhanced security capabilities
  - Provides a larger number of protection domains than supported directly by the processor and adds more precise control

- Support for Trusted Input and Output
  - Support for protected channel between a device and driver
More Benefits of I/O Virtualization

- Enhanced system reliability
  - Provides isolation between devices – more robust system
    - *system protected from errant device writes*

- Support legacy 32-bit devices in large-memory systems
  - May eliminate or reduce bounce buffers
AMD’s I/O Virtualization

ATC = Address Translation Cache  
HT = HyperTransport™
Outline

- What is Virtualization?
- Evolution of Virtualization
- AMD Virtualization
- AMD’s IO Virtualization
- Xen
How do I get it?

- http://fedora.redhat.com/
How do I run it?

- Install Fedora Distribution
- Select Xen package + others
- Complete installation and reboot
- From boot loader menu, select Xen
- Privileged domain will boot (Dom0)
- xend start
- Use the xm tool
- Visit the Xen readme’s at
  [http://www.fedoraproject.org/wiki/FedoraXenQuickstartFC5](http://www.fedoraproject.org/wiki/FedoraXenQuickstartFC5)
  [http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user/user.html](http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user/user.html)
Xen 3.0.2

- Minor updates in xen-unstable.hg
- 32-bit guest / 32-bit hypervisor support
  - Can boot Windows® OS as 32-bit guest
- 32-bit PAE guest / 64-bit hypervisor support
- 64-bit guests / 64-bit hypervisor support
Create your guest images

- What kind of a guest do you want?
  - Use one from Free OS Zoo
    - [http://free.oszoo.org](http://free.oszoo.org)
  - Install from ISO to QEMU image
    - `qemu-img create my6gdisk.img 6G`
    - `qemu -m 256 -hda my6gdisk.img -cdrom /dev/cdrom -boot d`
  - Use a physical disk
Xen 3.0.2 & AMD Virtualization

- VM0: Device Manager & Control s/w, GuestOS (XenLinux), Native Device Driver
- VM1: Unmodified User Software, GuestOS (XenLinux), Native Device Driver
- VM2: Unmodified User Software, GuestOS (XenLinux), Front-End Device Drivers, SMP
- VM3: Unmodified User Software, GuestOS Windows XP® 32-bit, Front-End Device Drivers

- Hardware (SMP, MMU, physical memory, Ethernet, SCSI/IDE)
- Control IF, Safe HW IF, Event Channel, Virtual CPU, Virtual MMU
- ACPI, 32/64bit, AMDV, AMDV

- Xen Virtual Machine Monitor

- Xen 3.0.2 & AMD Virtualization
Xen 3.0.2 & AMD Virtualization

VM0
- Device Manager & Control s/w
- GuestOS (XenLinux)

VM1
- Unmodified User Software
- GuestOS (XenLinux)

VM2
- Unmodified User Software
- Unmodified GuestOS Linux® 64-bit

VM3
- Unmodified User Software
- Unmodified GuestOS Windows XP® 32-bit

ACPI
- Native Device Driver
- Back-End

32/64bit
- Control IF
- Safe HW IF
- Event Channel
- Virtual CPU
- Virtual MMU

Hardware (SMP, MMU, physical memory, Ethernet, SCSI/IDE)

AMDV

Safe HW IF

Xen Virtual Machine Monitor

Front-End Device Drivers

AMD®
Links

AMD Virtualization (formerly known by the codename “Pacifica”)
Specification in AMD64 Architecture Techdocs at
http://www.amd.com/us-en/Processors/DevelopWithAMD/0,,30_2252_869_739^7044,00.html

AMD IO Virtualization
http://www.amd.com/us-en/Weblets/0,,7832_8366_7595~104860,00.html
Specification at
http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/34434.pdf
Thank You.