



AMD Xen™ Update

Xen Summit
November 2007

Tom Woller
thomas.woller@amd.com

Talk Outline

1. AMD Virtualization [™] Completed Features

- Nested Page Tables (Rapid Virtualization Indexing/RVI)**
- ASIDs (tagged TLB)**
- CR8/TPR Reduce Intercepts**
- BSD Xen Ongoing**

2. AMD Virtualization [™] Future Features

Completed Features



Nested Paging (Rapid Virtualization Indexing/RVI)

AMD: Wei Huang (wei.huang2@amd.com)

- Feature in Barcelona (Fam10h)
- Full NP Support in Xen 3.1 (use boolean "hap" option)
- Live Migration fixes in for Xen 3.1.1 (thanks Tim Deegan)
- "hap" on is now default for 64b/32b (unstable/3.2 testing)
- Heterogeneous AMD migration supported (Shadow <-> NestedPT)
- **Status:** Completed

ASID (Address Space Identifier)

- Tagged TLB (ID 0 == Host/hv, 1-63 for guests)
- Support is in 3.1.1 release
- Feature is in Barcelona Fam10h, Fam0Fh RevG (Desktop/Mobile)
- **Status:** Completed

Completed Features



CR8/TPR Intercept Reduction

AMD: Travis Betak (travis.betak@amd.com)

- Reduce CR8/TPR intercepts/vmexits for 64b HVM guests
- Patch is upstream in unstable/3.2
- Reduce intercepts from 10,000,000 to 6,000 on windows HVM guest boot
- CR8 read intercept disabled, write enabled as needed

Status: Completed

Completed Features



OpenBSD/NetBSD Xen

AMD: Christoph Egger (christoph.egger@amd.com)

- Assist with maintaining and porting of OpenBSD on Xen
 - 64b xen domain0/U kernel buildable on NetBSD/amd64 (need to test on OpenBSD/FreeBSD amd64 but should work)
 - 32bit PV DomU, and 32bit HVM guests boot using 32bit NetBSD/Xen as Domain0
- Future Work
 - Bllktap, xenmon and xentrace need some larger rework to be able to build on *BSD
 - On BSD-side 32bit binaries need additional porting of include files and libraries, to allow firmware/hvmloder to be built and used with a NetBSD/Xen as Dom0
 - NetBSD started its port to NetBSD/Xen to run as 64bit PV DomU. The 64b Dom0 is on the todo list.

Status: Ongoing support

Talk Outline

1. AMD Virtualization TM Completed Features

2. AMD Virtualization TM Future Features

- **AMD IOV (IOMMU)**
- **MCA/MCE**
- **NUMA**
- **2Meg Page Support**
- **PowerNOW!**
- **Extended Migration**

Future Features

AMD IO Virtualization (IOMMU)

AMD: Wei Wang (wei.wang2@amd.com), Joerg Roedel (joerg.roedel@amd.com)

- Feature will be part of future generation designs
- Initial AMD IOMMU patches in Xen unstable/3.2
- Use "enable_amd_iommu" as hypervisor boot parameter.
- Completed Items
 - Initial Direct IO support (hv patches in unstable)
 - Non-iommu aware dom0 boots, all devices share entire guest memory under 1:1 mapping
 - DMA layer support (not posted)
 - Implement IOMMU-aware DMA layer in dom0
 - Dom0 w/ modified dma layer boots, PCI devices isolated in different protection IO domains (not submitted upstream yet)
 - PV Support Hypercalls invoked by generic DMA layer to achieve device isolation within guest domain (needs posting and discussion)

Future Features

AMD IO Virtualization (IOMMU)

- Work In Progress Items
 - PV domain support
 - direct assignment of pci dev to PV guest other than dom0
 - HVM domain support
 - direct assignment of pci device
 - SATA controller working
 - USB/Network/IDE work in progress

Status: In progress

Future Features

MCA/MCE Enhancements

AMD: Christoph Egger (christoph.egger@amd.com)

- General enhancement of MCA/MCE in Xen base
- Overview
 - Domain0 contains support for MCA/MCE
 - Domain0 registers w/ HV MCE evnt hldr (CE), NMI trap hldr (UE)
 - Domain0/Guests use hypercall to fetch MCE telemetry
- Potential Uses
 - Self-Healing possible
 - Memory Page Retirement (MPR)
 - Utilizing online spare RAM
 - Guest or Process specific retirement/migration instead of system crash

Status: Patches submitted, target post 3.2

Future Features

MCA/MCE Enhancements Logic Details

case I) - Xen receives a MCE from the CPU
case II) - Xen receives Dom0 instructions via Hypercall

Machine check data is stored in an internal producer/consumer array. Guests uses a fetch machine check hypercall to receive the machine check error telemetry.

case I) - Xen receives a MCE from the CPU

1) Xen MCE handler figures out if error is an correctable (CE) or uncorrectable error (UE)

2a) error == CE:

Xen notifies Dom0 via machine check event if Dom0 installed an MCA event handler for statistical purpose

2b) error == UE and UE impacts Xen or Dom0:

Xen tries some self-healing

and notifies Dom0 via machine check trap on success if Dom0 installed MCE trap handler or Xen panics on failure. Xen also panics if Dom0 installed no MCE trap handler.

2c) error == UE and UE impacts DomU: In case of Dom0 installed MCE trap handler:

-> Xen notifies Dom0 via machine check trap and Dom0 tells Xen whether to also notify DomU via machine check trap and/or does some operations on the DomU (case II)

In case Dom0 did not install MCE trap handler

-> Xen notifies DomU via machine check trap directly

3a) DomU is a PV guest:

- if DomU installed MCE trap handler, it gets notified to perform self-healing

- if DomU did not install MCA trap handler, Dom0 does some operations on DomU (case II)

- if DomU nor Dom0 did not install MCE trap handlers, then Xen kills DomU

3b) DomU is a HVM guest:

- if DomU features a PV MCA driver, then behave as in 3a)

- if DomU has no such PV MCA driver, notify Dom0 to do some operations on DomU (case II)

- if neither DomU nor Dom0 did not install MCA trap handler, then Xen kills DomU

case II) - Xen receives Dom0 instructions via Hypercall

When Dom0 got enough CEs so that UEs are very likely to happen in order to "circumvent" UEs, then it will perform one or more operations on a DomU.

Future Features

NUMA

AMD: Andre Przywara (andre.przywara@amd.com)

- Added ontop of Ryan Harper's NUMA patches
- Current NUMA limitations in unstable
 - Memory is allocated from NUMA node where first VCPU is scheduled, so for performance reasons Domains are (practically) limited to one NUMA node
 - Dom0 ballooning has some issues with numa=on
 - Migration functionality not NUMA aware (memory/core destination not specified on move)

Future Features

NUMA

- Possible solutions
 - Spread guest domains over NUMA nodes and propagate ACPI SRAT/SLIT info into each guest:
 - HVM guests via hvm_info_table (patches available)
 - PV guest support (in development)
 - Introduce load balancing of guest between domains, including using live migration (no code yet)
 - Intra-machine node Live migration w/ NUMA awareness
 - “xm migrate --live 5 127.0.0.1”, pin and migrate memory also
- NUMA HOWTO for Xen 3.2 in backup slides

Status: NUMA guest patches readying to push post 3.2

Future Features

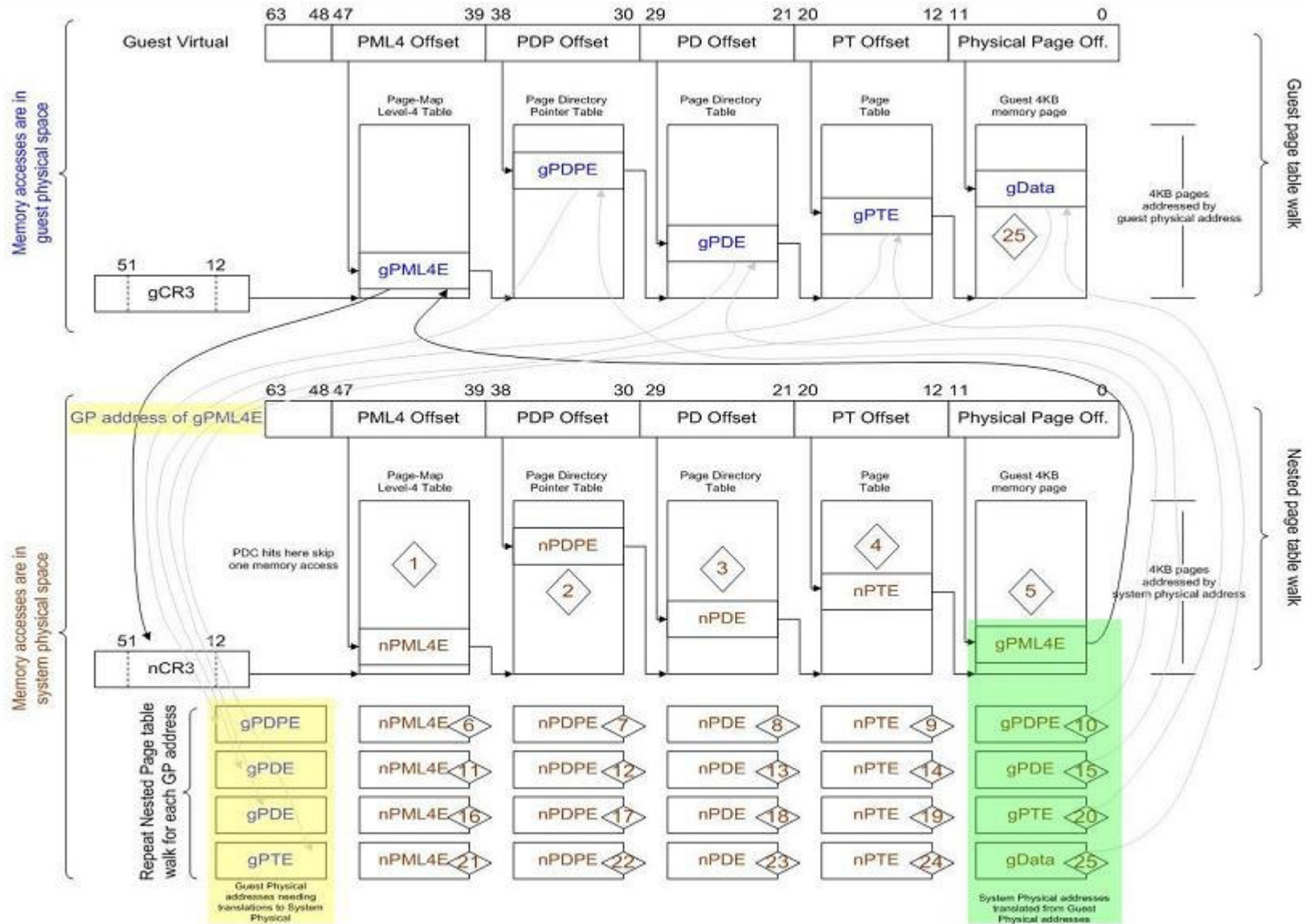
2 Meg Page Table Support

AMD: Wei Huang (wei.huang2@amd.com)

- 2 Meg page tables initially backing HVM guest for NP (guest has 4k pages)
- Benefits
 - Barcelona has additional 2Meg TLB entries
 - TLB translation hit
 - One 2MB TLB entry can cover 512 4k TLB entries, which implies less TLB misses for apps with large memory footprint
 - TLB translation miss
 - Reducing Worst case cost of a Nested Page Walk with 3 level backing pages instead of 4 level
 - $3 \times 4 = 20$ memory accesses vs $25 = 20\%$ reduction
 - 2Meg Guest pages backed by 2Meg hv pages
 - $3 \times 3 = 16$ memory accesses vs $25 = 36\%$ reduction

Status: in progress

Nested Paging Page Entry Accesses



Future Features

PowerNOW!

AMD: Mark Langsdorf (mark.langsdorf@amd.com)

- PowerNOW! Part of the cpufreq subsystem
- Must build into Domain0 kernel (CONFIG_CPU_FREQ)
- Use cpufreq=dom0-kernel option to enable, which causes xen to automatically pin dom0 VCPUs to respective pcores
- Domain0 performs hypercalls to obtain perf data then uses PowerNow! Driver normally to determine processor frequency changes
- No numbers yet specifically detailing power consumption improvements

Status: Initial work completed

Future Features



PowerNOW! Logic Details

- The ondemand governor schedules a function (`dbb_calc_load`) to run on each core.
- When `dbb_calc_load` runs, it makes the `get_cpuidletime` platform hypercall and passes in the `cpumask` of the processors.
- The `get_cpuidletime` hypercall fills an array with the runtimes (in nanoseconds) of the idle vcpus for each physical core. Idle vcpus are permanent vcpus that Xen schedules on unused physical cpus to account for idle time. The `get_cpuidletime` also returns the current Xen system time in nanoseconds.
- Back in `dbb_calc_load`, the `dom0` function calculates the difference between each processor's current idle time and its last idle time (`delta_idle`), and the difference between the current system time and last system time (`delta_total`). The difference between `delta_total` and `delta_idle` is then divided by `delta_total` and multiplied by 100, creating a percentage load that is passed back to the ondemand governor.
- The governor then uses its standard Linux routines to increase or decrease frequency, and schedules another call to `dbb_calc_load` in the future.

Future Features

Extended Migration

AMD: Tom Woller (thomas.woller@amd.com)

- Extended Migration is AMD's term for the ability to generalize the CPUID information for a pool of machines. AMD-V Extended Migration provides the necessary support for virtualization software to mask the differences between CPU generations, facilitating the safe live migration of virtual machines between servers running different generations of AMD processors. This includes existing single-core and dual-core processors and all future AMD processor revisions.
- 2 "masking" MSR (C001_1004/C001_1005)
Extended Migration effectively requires no additional modifications or support outside of s/w. Since K8->RevC, there have been 2 "masking" MSR (C001_1004, C001_1005) that can be used to mask bits associated with CPUID leafs 0000_0001h and 8000_0001h respectively.
- Patch contains hv boot parm to specify AMD Cpu type (amd_cpu=)
- Platform Op allows domain0 to also set amd_cpu type

Status: patch tested and functional, not submitted

Questions/Comments

Backup

Future Features

NUMA



XEN 3.1 NUMA HOWTO

Even though the current NUMA status in XEN 3.1 or unstable is not complete, you may happen to setup a system on such a machine. As a workaround follow these recommendations:

- Boot xen with **numa=on** and **dom0_mem=nnnM** on the grub XEN command line. Requires XEN 3.0.4 or higher.
- Disable ballooning in /etc/xen/xend-config.sxp: **(dom0-min-mem 0)**
- Learn about the assignment of memory and CPUs to nodes: "Ctrl-A (3 times) + u", xm info or use native Linux
- Edit the domain's config file and restrict all VCPUs to host CPUs within a single NUMA node: `cpu=0,1`
- XEN tries to allocate memory from the first VCPUs node, so avoid assigning more memory than there is in this node. Free memory information can be get by dumping heap statistics ("Ctrl-A (3 times) + H") and adding up, script available.
- For simplicity you could assume equally distributed memory linearly assigned to the CPUs. For example: 8 GB 2 socket dual-core Opteron system: two NUMA nodes, each node is assigned 4 GB of memory and 2 cores, cores 0 and 1 are in NUMA node 0 and care about the memory from 0-4.5 GB (with a 512 MB or so hole just below 4GB), while core 2 and 3 are in node 1 and care about the memory from 4.5 to 8.5 GB. There is no need for cores to be assigned in order, this depends on the BIOS.
- Balancing several domains must currently be done manually. Domains with high load should be assigned to different nodes, smaller domains can be assigned to already used nodes, too.
- Since over-committing VCPUs is questionable, restrict domain's VCPU number to the number of cores within one NUMA node.

Trademark Attribution

AMD, the AMD Arrow logo, AMD Virtualization, AMD-V, and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Linux is a registered trademark of Linus Torvalds. Windows is a registered trademark of Microsoft Corporation. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

©2007 Advanced Micro Devices, Inc. All rights reserved.