AMD Opteron™ 6000 Series Platform architecture: Red Hat Enterprise Linux performance

Bhavna Sarathy
Tech. Engineering Lead, Red Hat Alliance
bhavna.sarathy@amd.com

June 23, 2010

Disclaimer and Attribution

DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

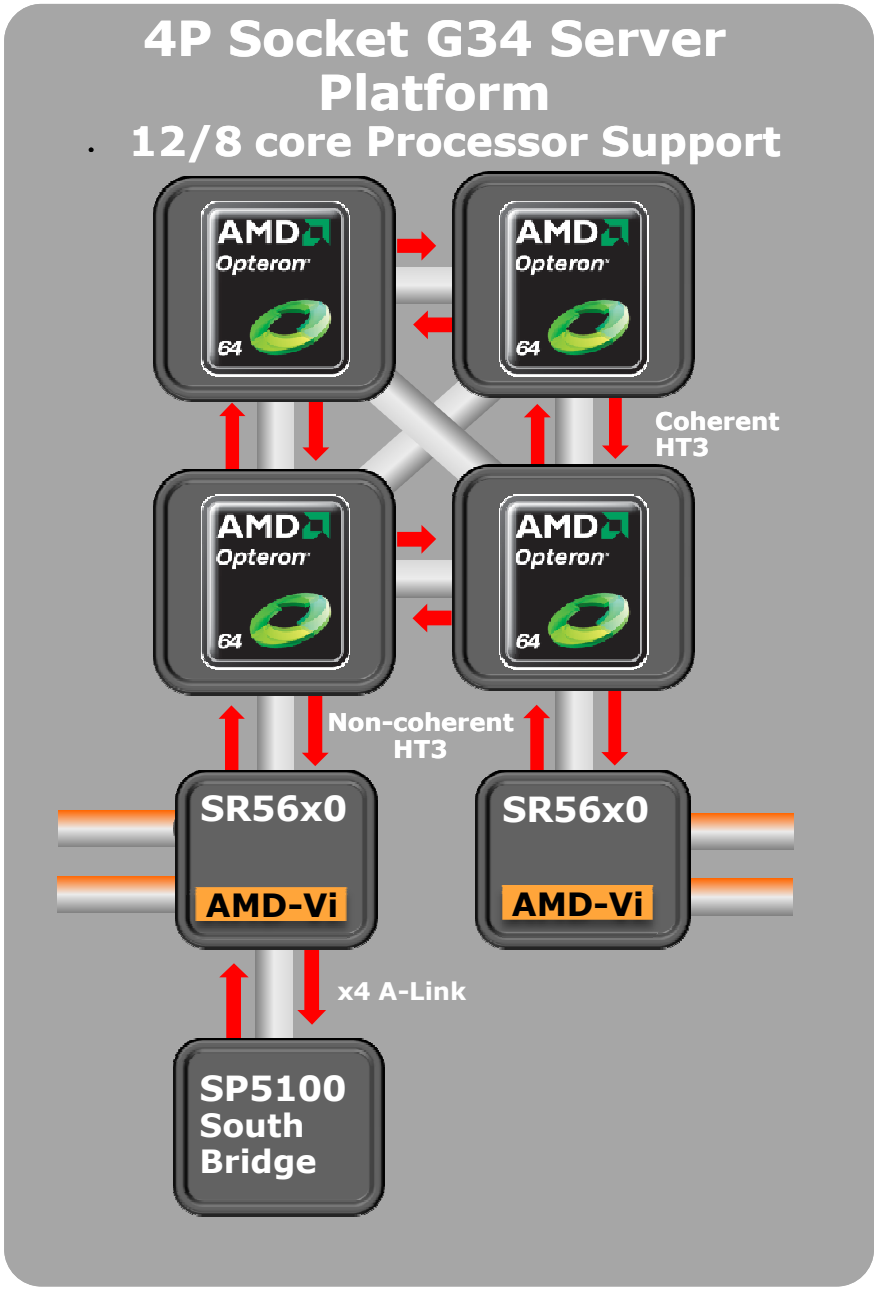
ATTRIBUTION

Outline

- AMD Opteron™ 6000 Series Platform
  - Magny-Cours Processor architecture
  - AMD IOMMU - chipset features
- RHEL feature enablement
- Performance benchmarking results
- AMD Virtualization (AMD-V ™) futures
- Summary

AMD Opteron™ 6000 Series Platform

- 12-core and 8-core 12M L3 Cache,
- Cool Core™ Technology, Enhanced AMD PowerNow!, C1E, CoolSpeed Technology, APML
- Up to 3 DIMMs/channel, 12 per CPU
- Platforms 2P/2U, 2P Tower, 4P rack, 4P Blade
- Performance-optimized Power/thermals
- Quad 16-bit HT3 links, up to 6.4 GT/s per link
- AMD SR56x0 chipset with AMD-Vi and PCIe Gen2



Outline

- AMD Opteron™ 6000 Series Platform
  - "Magny-Cours" Processor architecture
  - AMD IOMMU - chipset features
- RHEL feature enablement
- Performance benchmarking results
- AMD Virtualization (AMD-V ™) futures
- Summary

What is Direct Connect Architecture 2.0?

Direct Connect Architecture 2.0

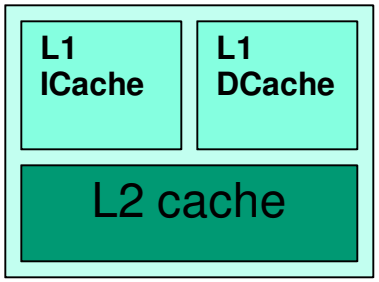
refers to set of two nodes in one socket, connected via HT links. Each node has two memory controllers, 6M L3 cache, and a northbridge. A fourth HT 3.0 link results in fully connected topology.

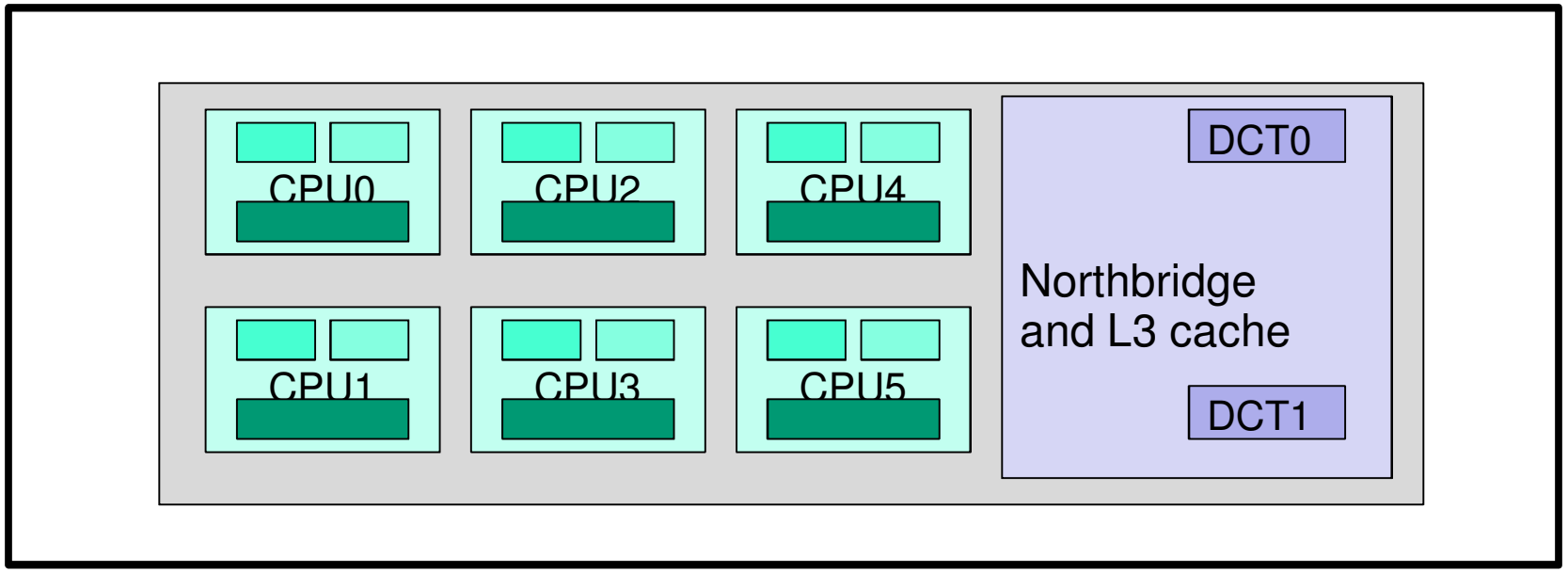
AMD Magny-Cours processor introduces it's first multi-node processor architecture. For multi-node processors:

Processor $\not==$ Node

Magny-Cours Direct Connect Architecture 2.0

CPU Core

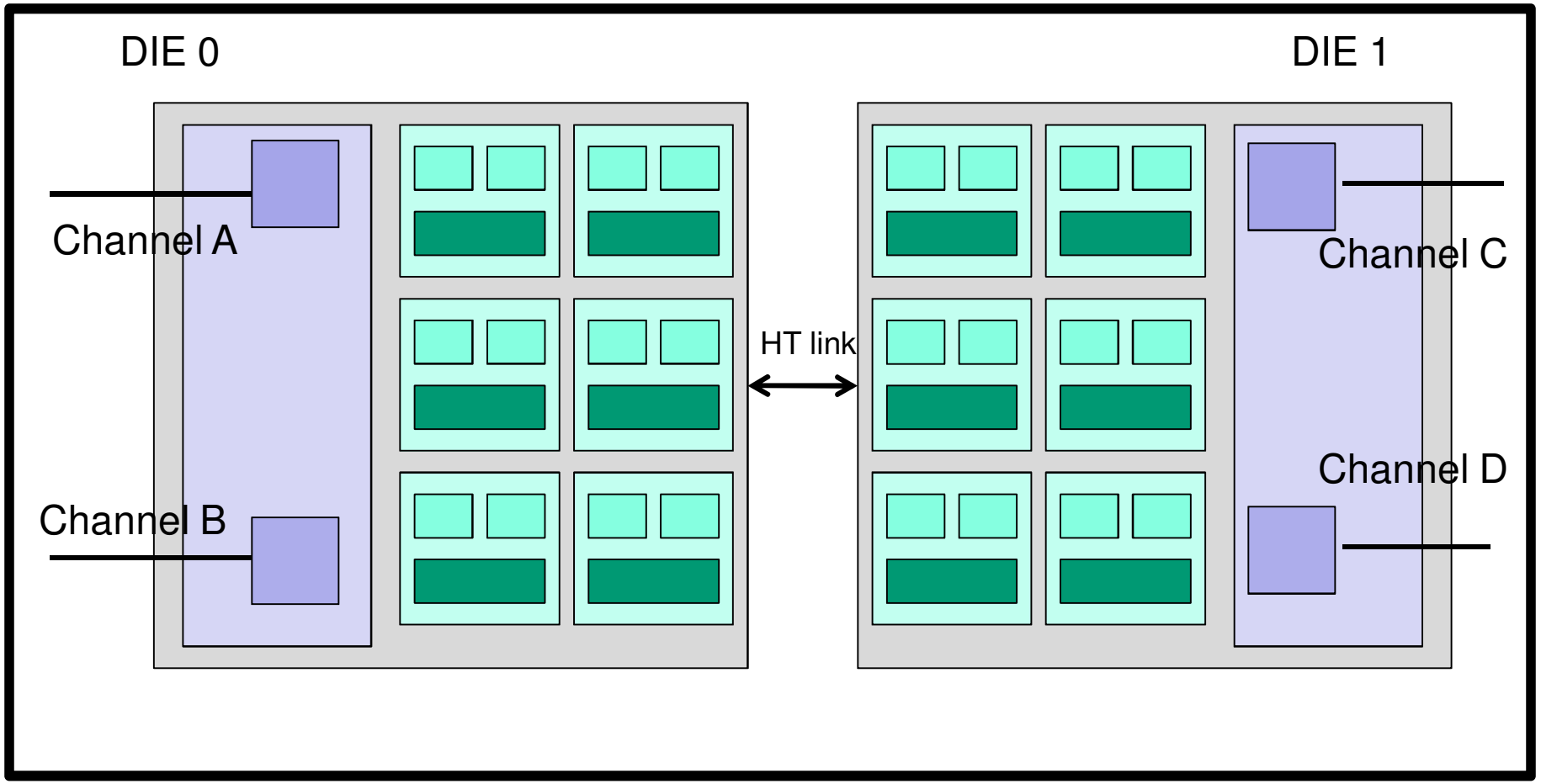




Single Chip Module (SCM)

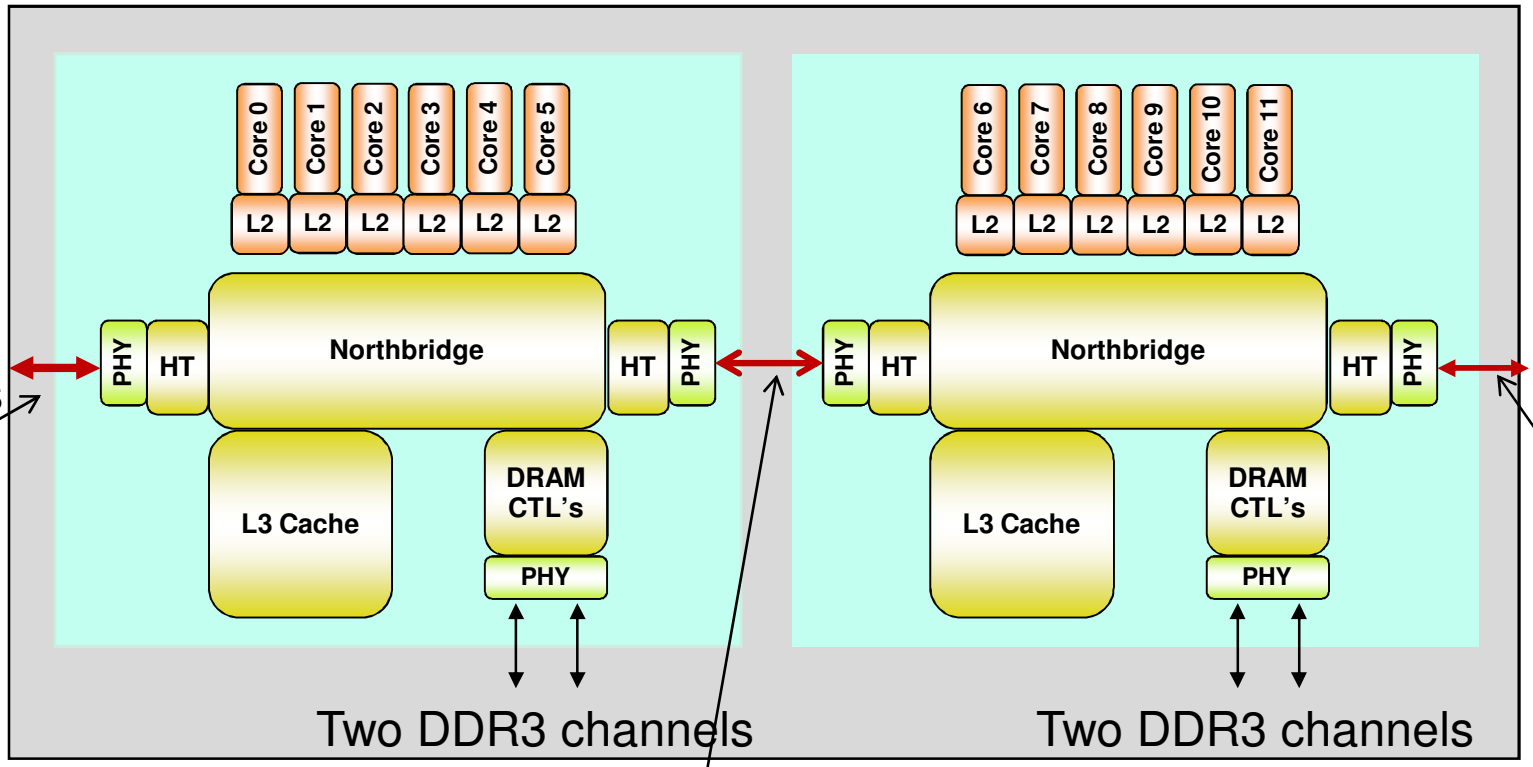
DCT = DRAM Controller

Magny-Cours Direct Connect Architecture 2.0 (contd)



Direct Connect Module (DCM)

Socket G34 Server Products (Magny-Cours)



2 ½ HT links

Two 16-bit HT links, and one 8-bit HT link, available for external use.

16-bit HT link that connects the two nodes together. Not available for external use.

1 ½ HT links

One 16-bit HT link, and one 8-bit HT link, available for external use.

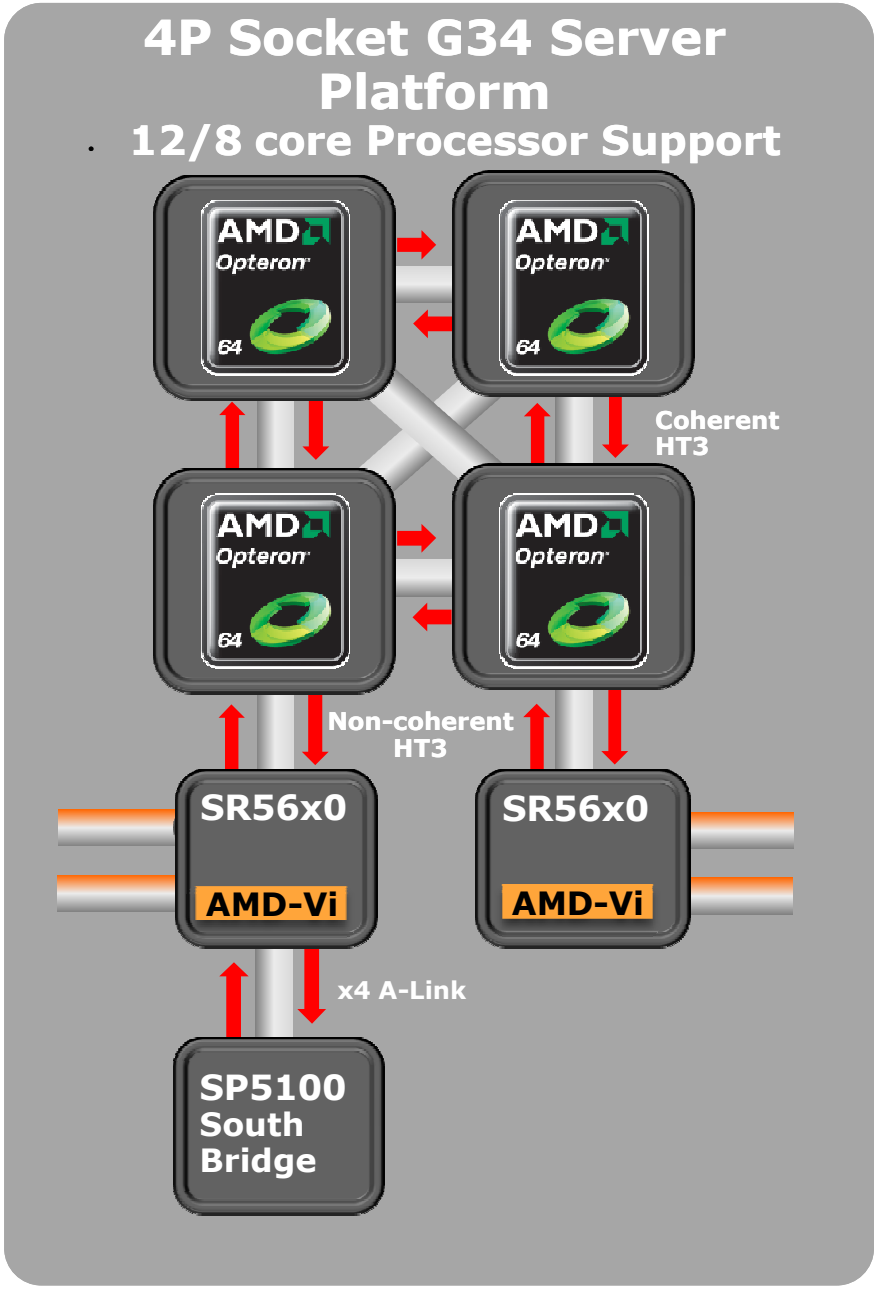
Outline

- AMD Opteron™ 6000 Series Platform
  - “Magny-Cours” Processor architecture
  - AMD IOMMU chipset features
- RHEL feature enablement
- Performance benchmarking results
- AMD Virtualization (AMD-V ™) futures
- Summary

AMD I/O virtualization in Maranello platform

AMD I/O virtualization:

- Introduced in Maranello Magny-Cours platforms
- SR 56x0 chipsets (SR5690, SR5670, SR5650)
- AMD IOMMU driver implemented in RHEL (Linux, KVM and Xen drivers)



What is AMD I/O Virtualization?

AMD I/O Virtualization (IOMMU)

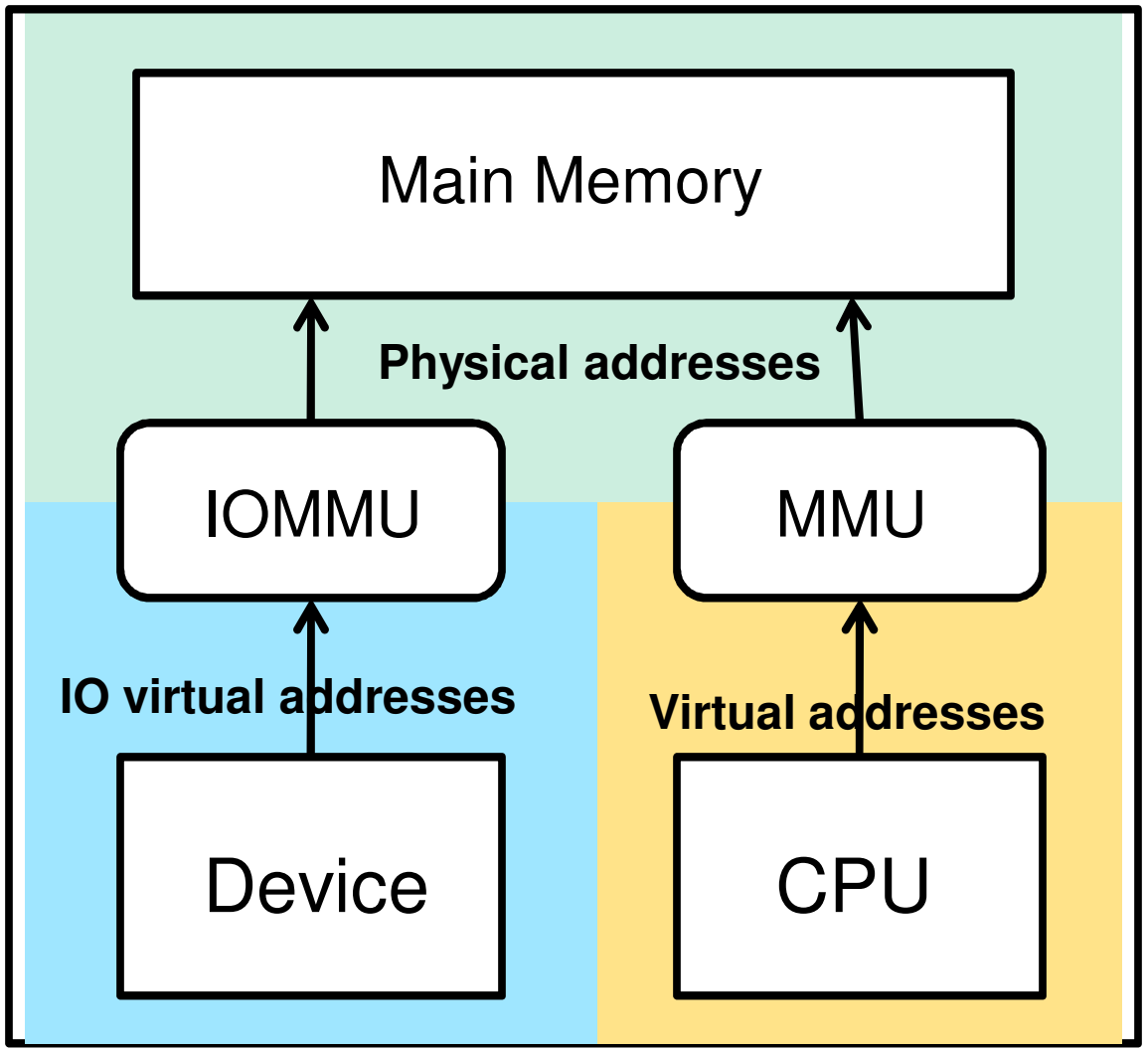
manages device access to system memory, translating device requests into system memory addresses, while ensuring the accesses are permitted.

Benefits:

- Improves performance with reduced overheads in virtualized systems.
- Provides security by isolation

AMD IOMMU

H/W help for I/O Virtualization is already here…

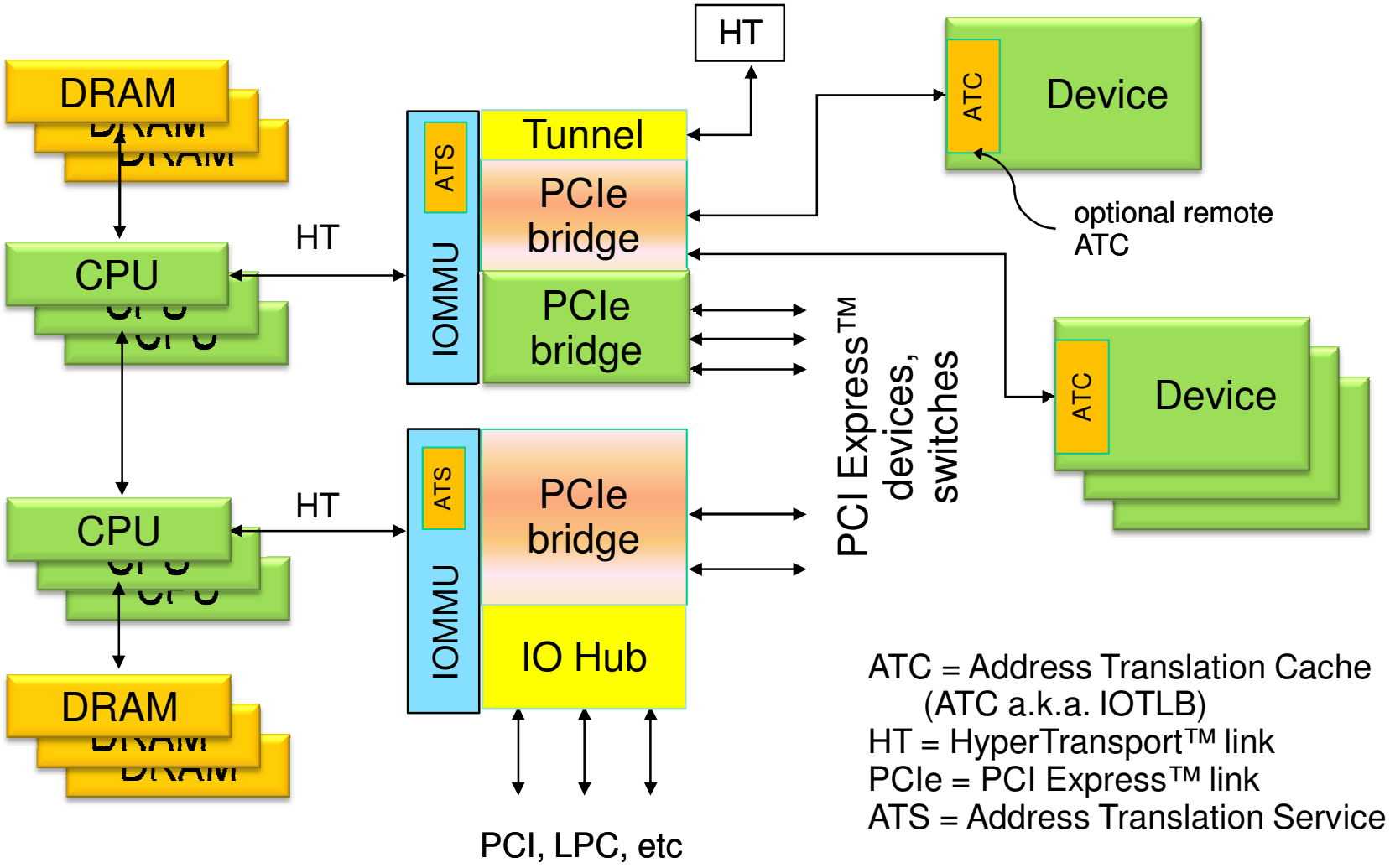


IOMMU is to Devices

as

MMU is to CPUs

Virtualizing The Platform IOMMU Version 1



ATC = Address Translation Cache
(ATC a.k.a. IOTLB)
HT = HyperTransport™ link
PCIe = PCI Express™ link
ATS = Address Translation Service

AMD IOMMU V1 - Uses

- I/O Virtualization
  - Direct device assignment for more efficient I/O
  - I/O interrupt steering helps prevent HV interaction
  - Legacy devices – helps prevent “bounce buffers”
  - PCI-SIG
    - PCIe IOV – using SR-IOV
    - PCIe ATS 1.0 - Address Translation Services
- RAS
  - Device DMA containment
  - Denial-of-service protection -- interrupt flood or MSI spoofing

Outline

- AMD Opteron™ 6000 Series Platform
  - “Magny-Cours” Processor architecture
  - AMD IOMMU chipset features
- RHEL feature enablement
- Performance benchmarking results
- AMD Virtualization (AMD-V™) futures
- Summary

Red Hat Enterprise Linux 5.5 support

- Support Magny-Cours topology
- AMD IOMMU driver support in KVM
- IOMMU interrupt remapping in Xen
- Memory placement and NUMA fixes in Xen
- RAS features
  - Support Family10h EDAC (amd64_edac) driver
  - L3 cache index disable

Magny-Cours topology changes in RHEL5.5

- Core topology changes in Linux kernel
- New function amd_fixup_dcm()
- New feature flags:
  - X86_FEATURE_NODEID_MSR (6*32+19)
  - X86_FEATURE_AMD_DCM (3*32+27)
- If processor is Magny-Cours:
  - Set cpu capability feature flag
  - Store nodeID, and store sibling data in llc_shared_map
  - Fix-up core ID to fall within the cores/per node range

Magny-Cours topology code snippets

```
/* read NodeID MSR */
rdmsrl(MSR_FAM10H_NODE_ID, value);
/* set cpu capability to the DCM flag */
set_cpu_cap(c, X86_FEATURE_AMD_DCM);
cores_per_node = c->x86_max_cores / nodes;
/* store nodeID, use llc_shared_map to store sibling info */
per_cpu(cpu_llc_id, cpu) = value & 7;
/* fixup core id to fall within the cores per node range */
c->cpu_core_id = c->cpu_core_id % cores_per_node;
```

arch/x86_64/kernel/setup.c; arch/i386/kernel/cpu/amd.c

AMD IOMMU features in RHEL5

The AMD IOMMU driver implementation provides:

- *Device Isolation* – mapping a device to a guest, while ensuring guest stays in its address space
- *Interrupt remapping* - remaps a shared interrupt to an exclusive vector, to ensure accurate guest delivery
- *Direct Device Assignment* - ability to directly assign a physical device to a guest

  arch/x86_64/kernel/amd_iommu.c;
  arch/x86_64/kernel/amd_iommu_init.c

Outline

- AMD Opteron™ 6000 Series Platform
  - “Magny-Cours” Processor architecture
  - AMD IOMMU chipset features
- RHEL feature enablement
- Performance benchmarking results
- AMD Virtualization (AMD-V™) futures
- Summary

RHEL5.5 Performance Testing on Twelve-Core AMD Opteron™ "Magny-Cours"

Presenting engineering testing data to show architectural features and tuning with RHEL5.5 – tests done at Red Hat performance labs

- Bare Metal Scalability Testing with Oracle OLTP workload
- KVM multiguest testing with OLTP workload
- Taking advantage of NUMA
- RHEL5 and huge page support
- Adjacent versus remote NUMA node

RHEL5.5 Performance Testing on Twelve-Core AMD Opteron™ "Magny-Cours"

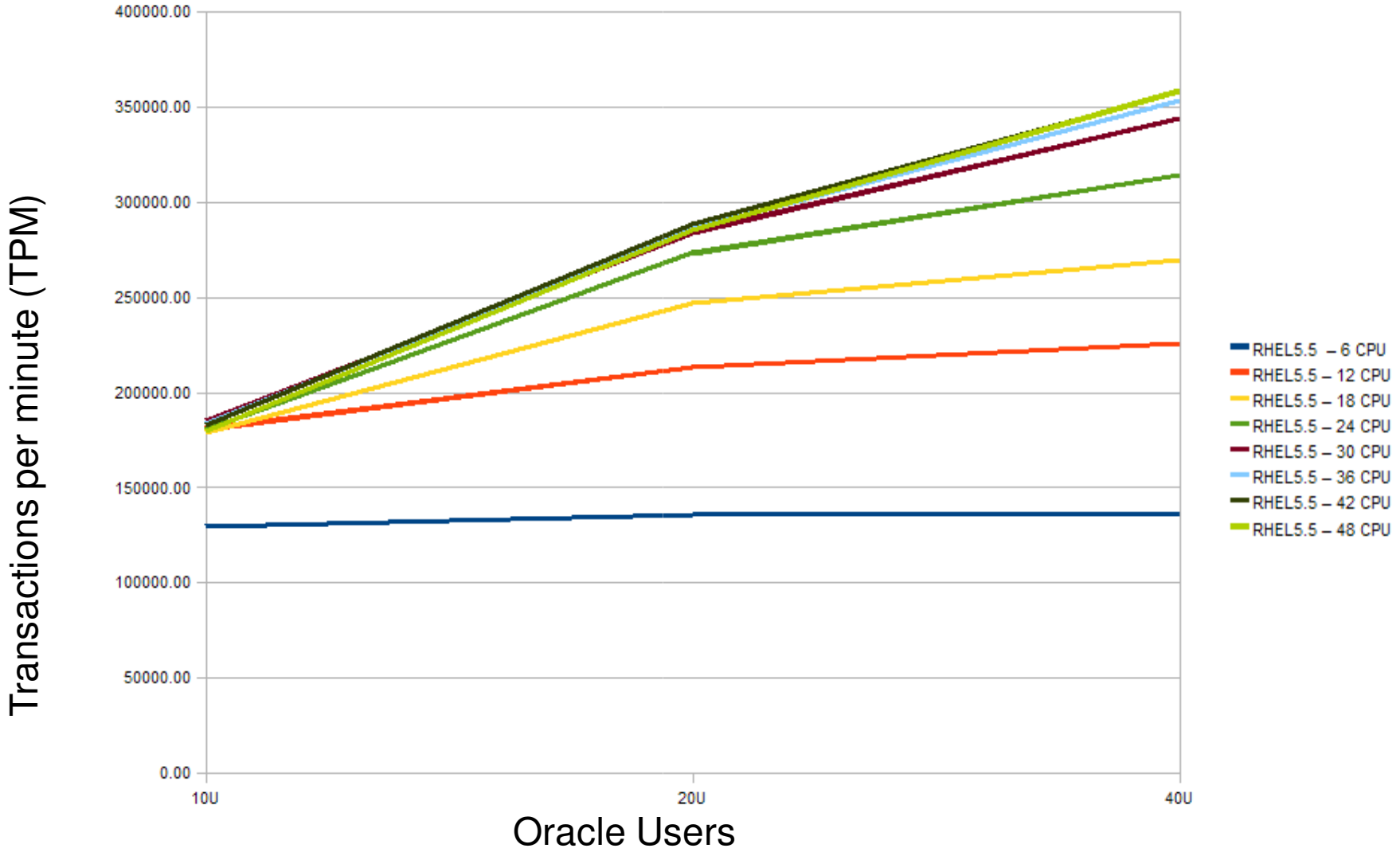
Hardware Configuration:

- 4-socket - AMD Opteron™ Processor "Magny-Cours"– engineering reference platform
- 64GB memory
- HP Modular Smart Array (MSA) Fiber channel storage

Disclaimer:

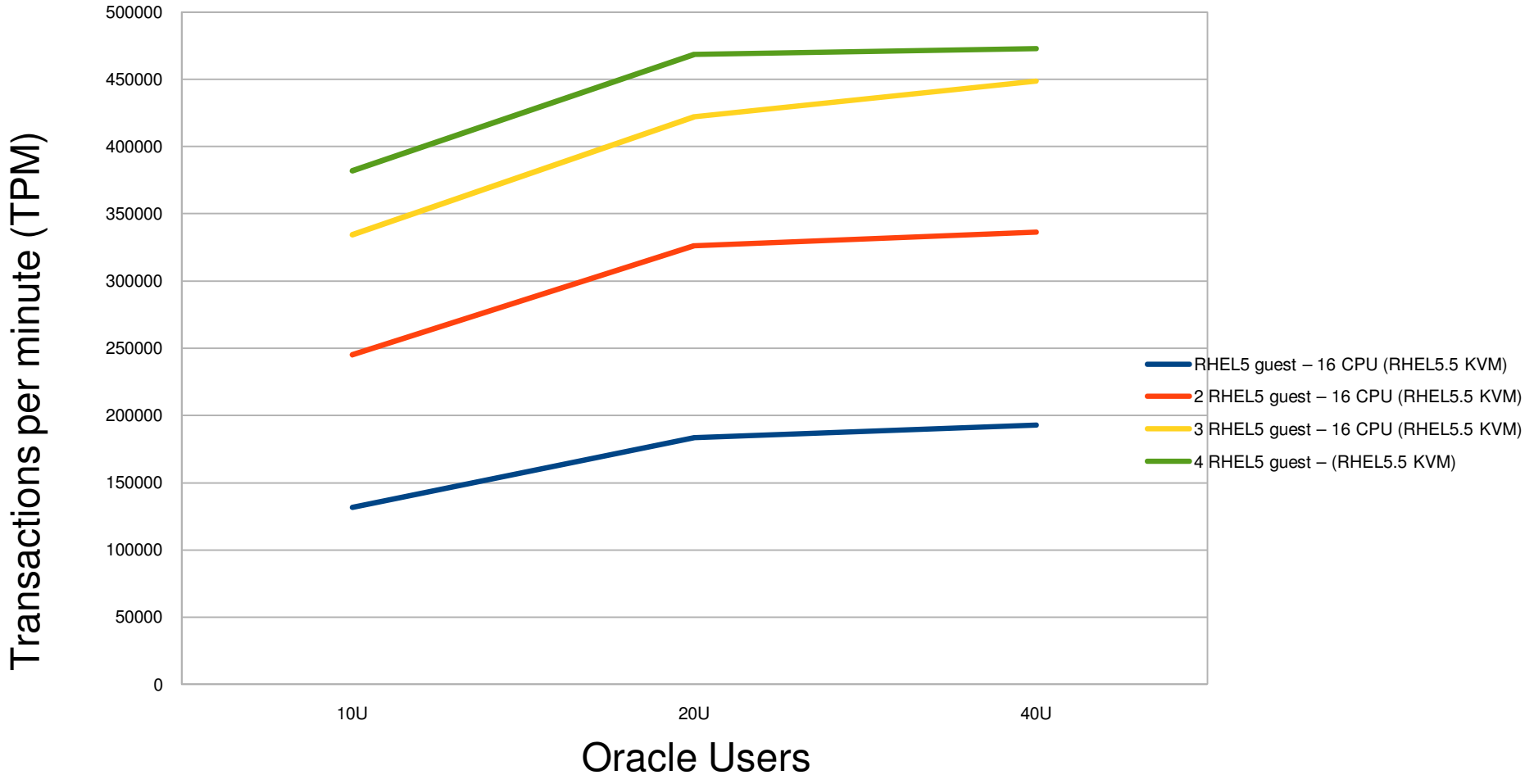
- Testing done on AMD engineering reference platform
- Insufficient storage to drive the 48 core Magny-Cours system

Magny-Cours scaling data- Oracle OLTP workload



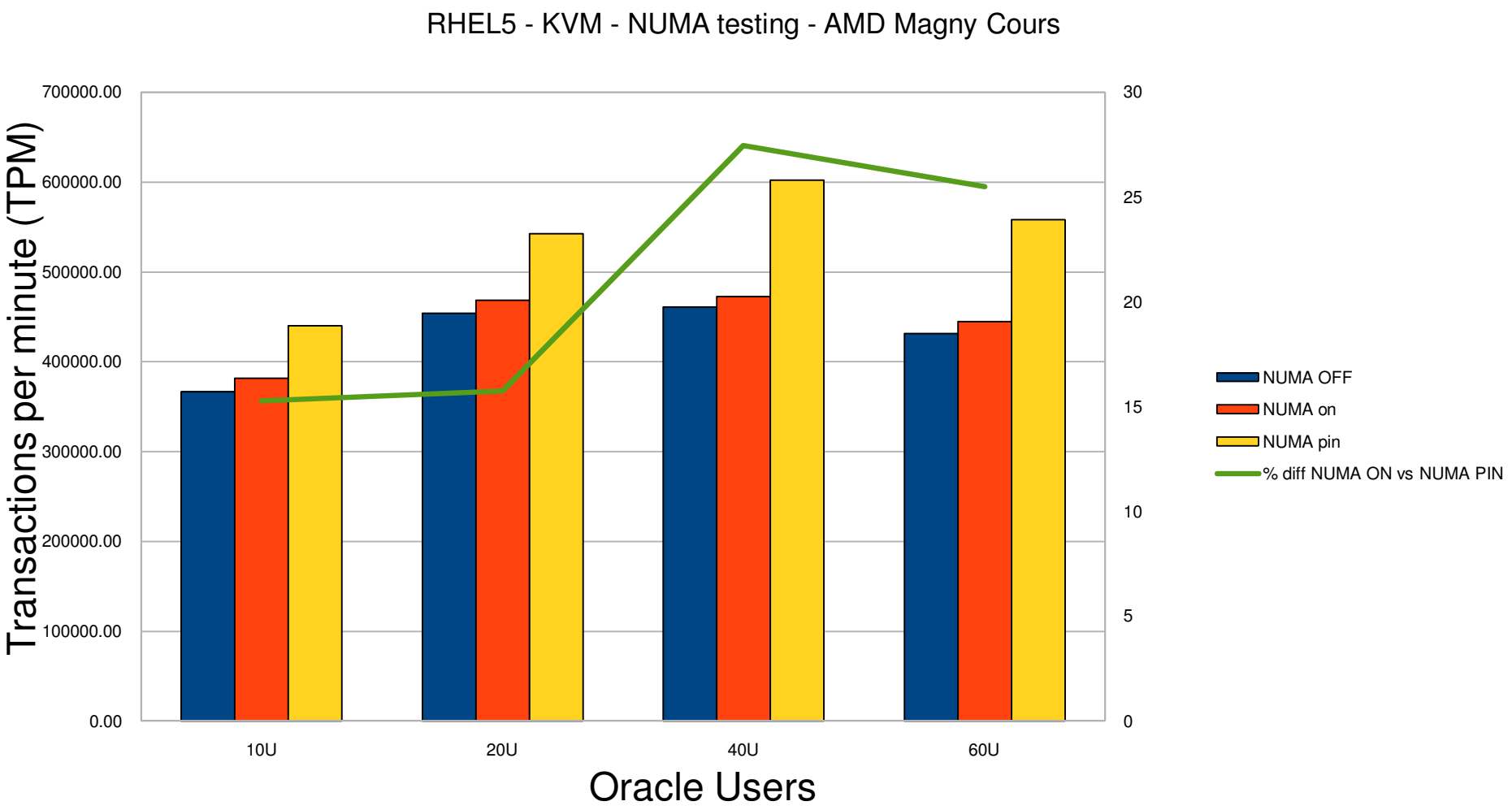
Graph shows scaling with Oracle OLTP workload, system scales with increase in user count

Magny-Cours: KVM multi-guest CPU scaling - Oracle OLTP workload



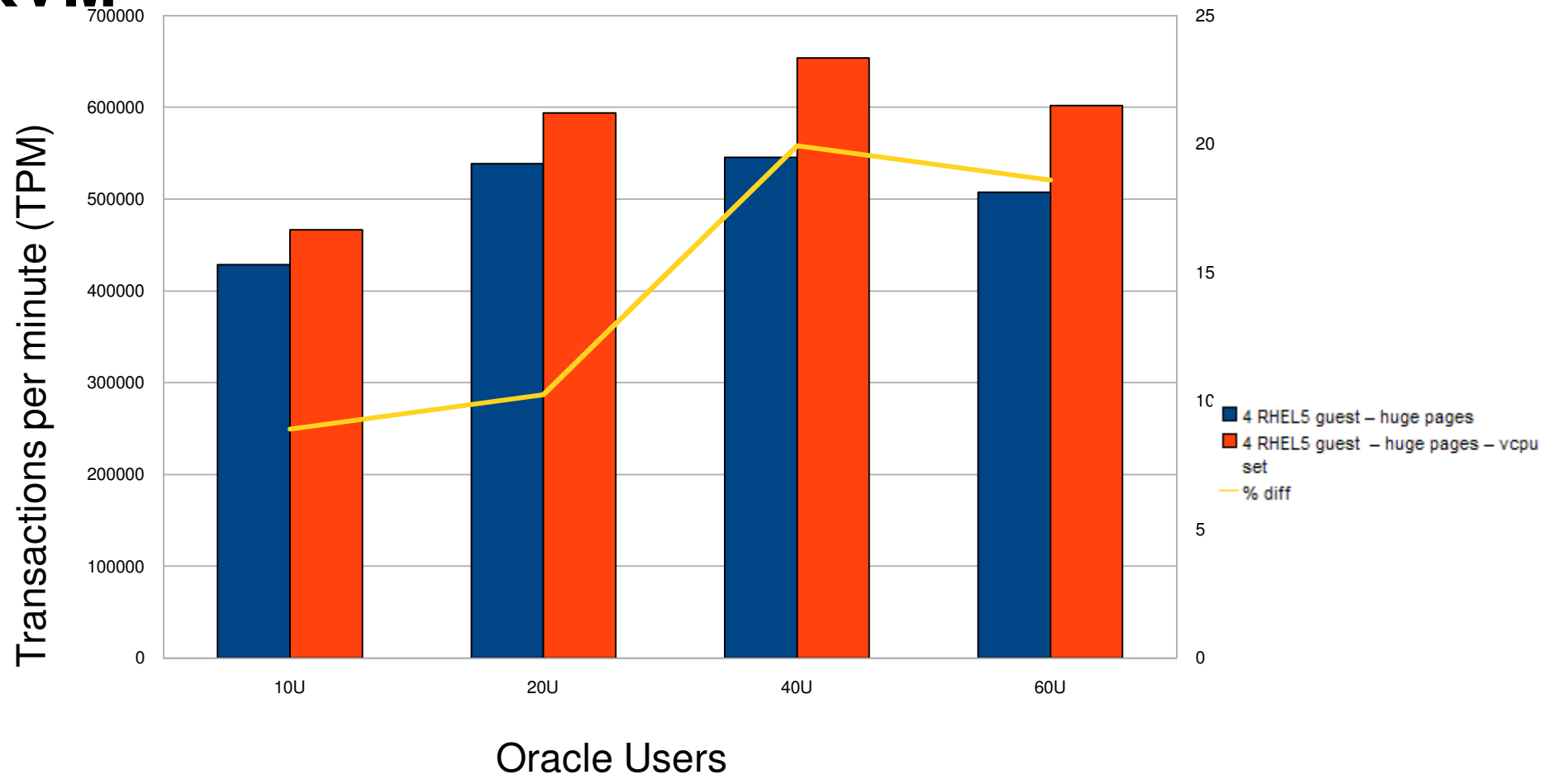
Graph shows scaling with Oracle OLTP workload – multiguest KVM – over subscribed with no significant penalty

Magny-Cours – RHEL5 KVM NUMA testing



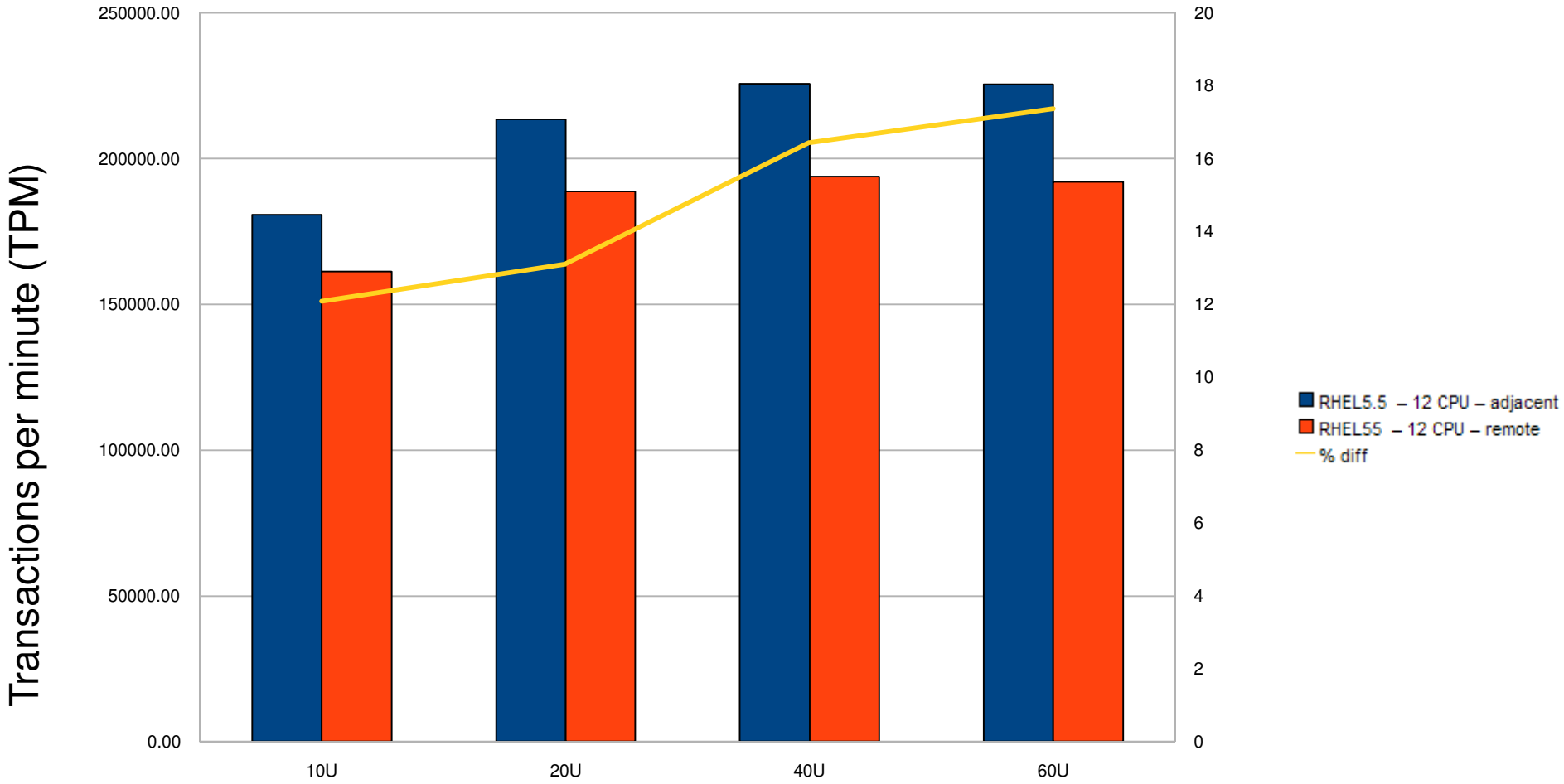
Comparison between NUMA off, NUMA on and NUMA pinning (using numactl)

Magny-Cours: RHEL5 with huge pages – RHEL5.5 KVM



Comparison between multi-guest – using huge pages vs huge pages + NUMA CPU pinning

Magny-Cours: adjacent and remote node data



Shows effects of tuning the system and comparison of using adjacent nodes/localized memory and remote node/memory

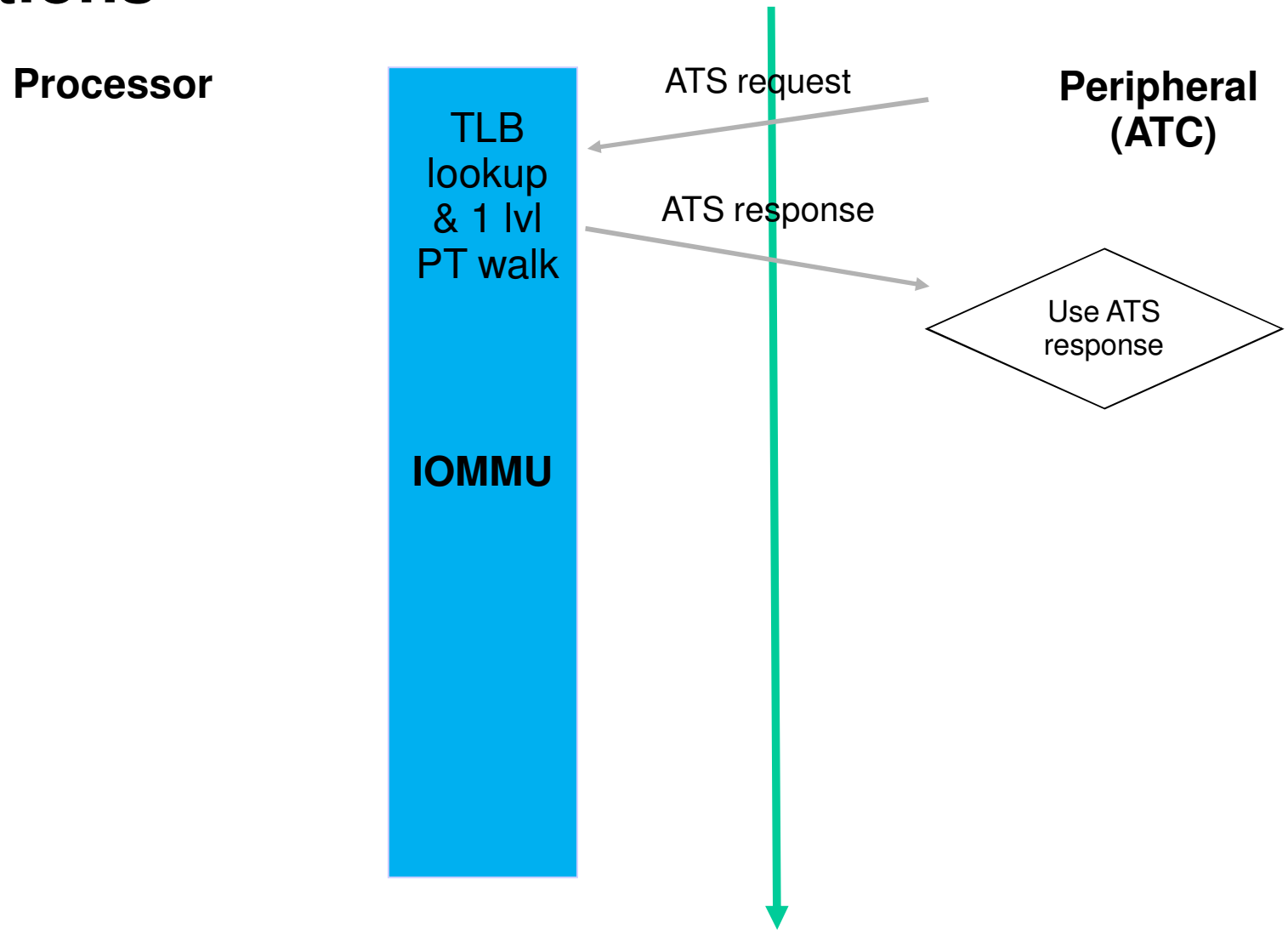
Outline

- AMD Opteron™ 6000 Series Platform
  - “Magny-Cours” Processor architecture
  - AMD IOMMU chipset features
- RHEL feature enablement
- Performance benchmarking results
- AMD Virtualization (AMD-V ™) futures
- Summary

Introducing AMD IOMMU Version 2

- IOMMU version 1 compatibility
- ATS 1.1 PRI support (Page Request Index)
  - Supports “Page Faults” for devices
  - Allows Hypervisor memory overcommit for guests (Demand paging)
  - RDMA usage without pinning memory
- Nested Page Tables
  - 2nd levels of page table walking supported
    - L1: Guest virtual to Guest Physical (AMD64 compatible)
    - L2: Guest Physical to System Physical (v1 compatibility)
  - 100% AMD64 compatible level
  - Allows direct device assignment in virtualized systems to use guest virtual address
  - Share OS PTs in assigning User Level I/O to devices in native environments

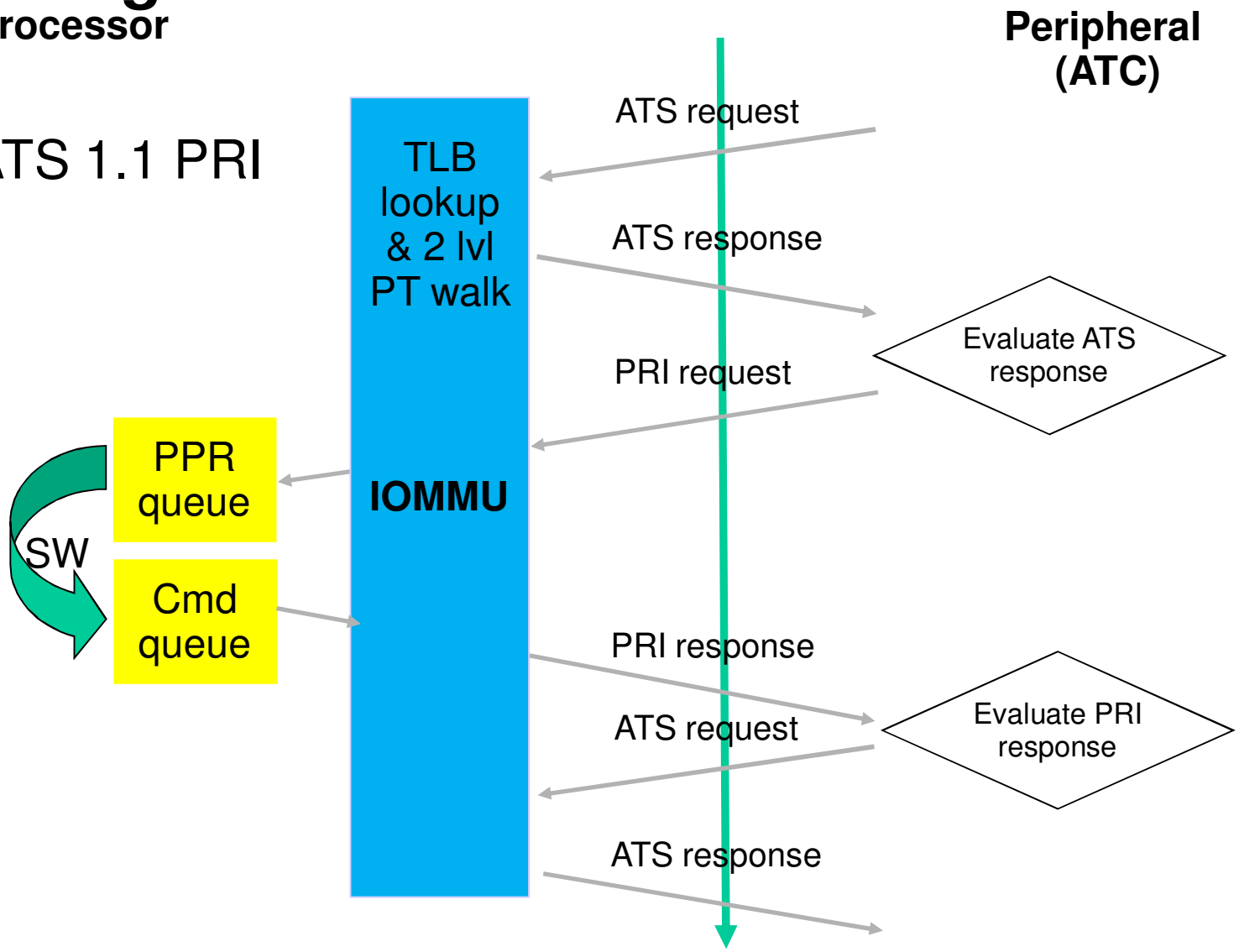
IOMMUv1 (ATS 1.0) Caching Address Translations



IOMMUv2 Page Fault & Overcommit

- PCI-SIG ATS 1.1 PRI

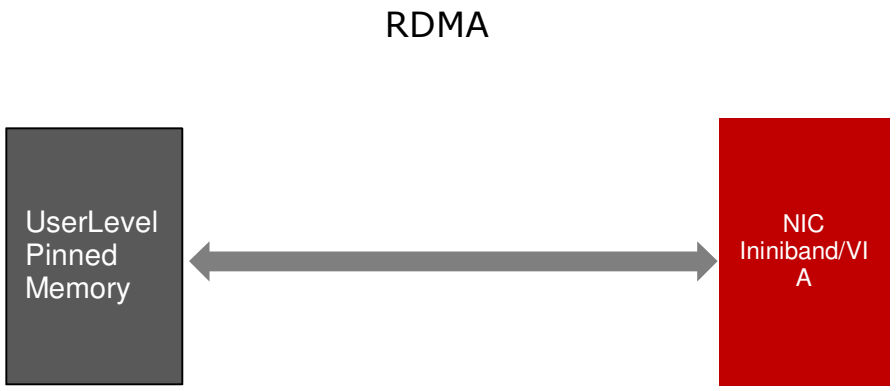
- Swap in page
- Alloc new page
- Reject request
- Upgrade privs
- Copy-on-write
- Etc.



Example: Smart NIC RDMA Use Case

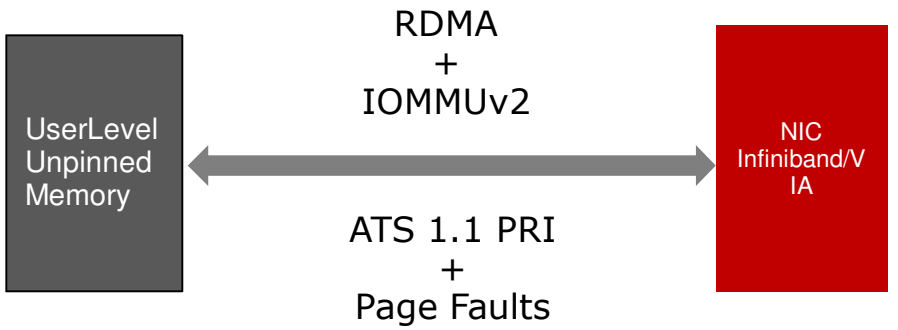
Current

- Overhead of managing pinned buffers
- Lack of demand-paging support



What do we want?

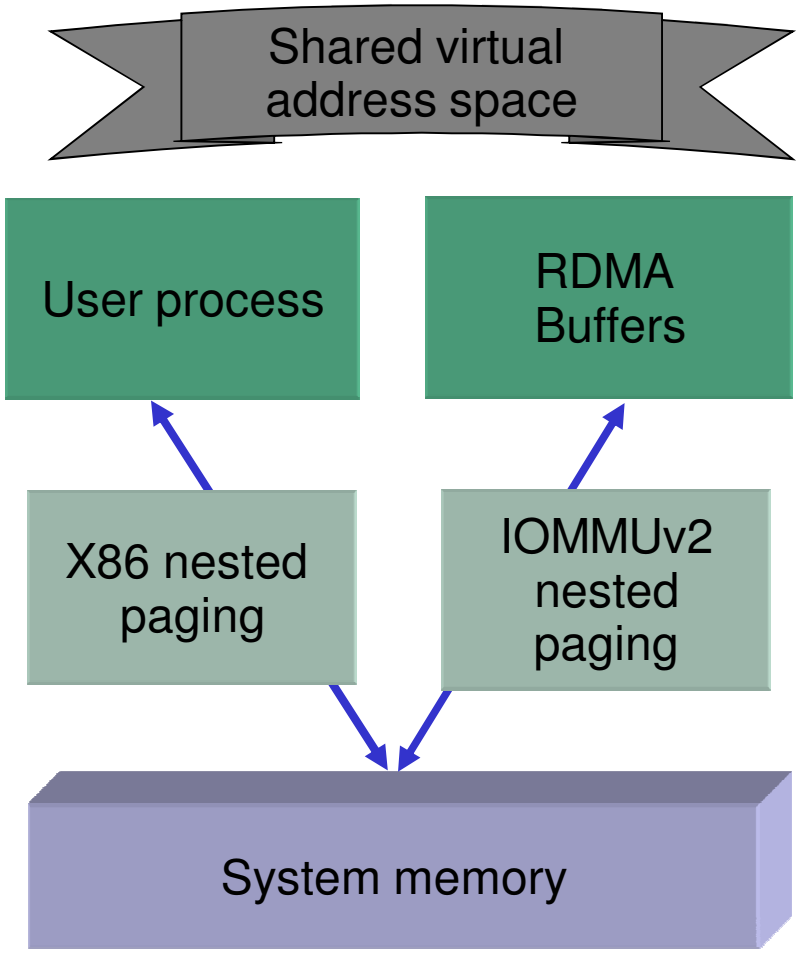
- Eliminate need for Pinned memory
- Smart NIC operates on unpinned region directly using ATS PRI and Page Faults



IOMMUv2 Direct Guest Mapping
User-level I/O

User-level I/O

- x86 PTE, IOMMU nested paging PRI+ATS
- Advanced memory model
  - Demand paging
  - Swapping
  - Copy-on-write
- Shared Virtual addresses among smart devices
- Direct access to devices at user-level reduces I/O overhead



Outline

- AMD Opteron™ 6000 Series Platform
  - “Magny-Cours” Processor architecture
  - AMD IOMMU chipset features
- RHEL feature enablement
- Performance benchmarking results
- AMD Virtualization (AMD-V ™) futures
- Summary

Summary

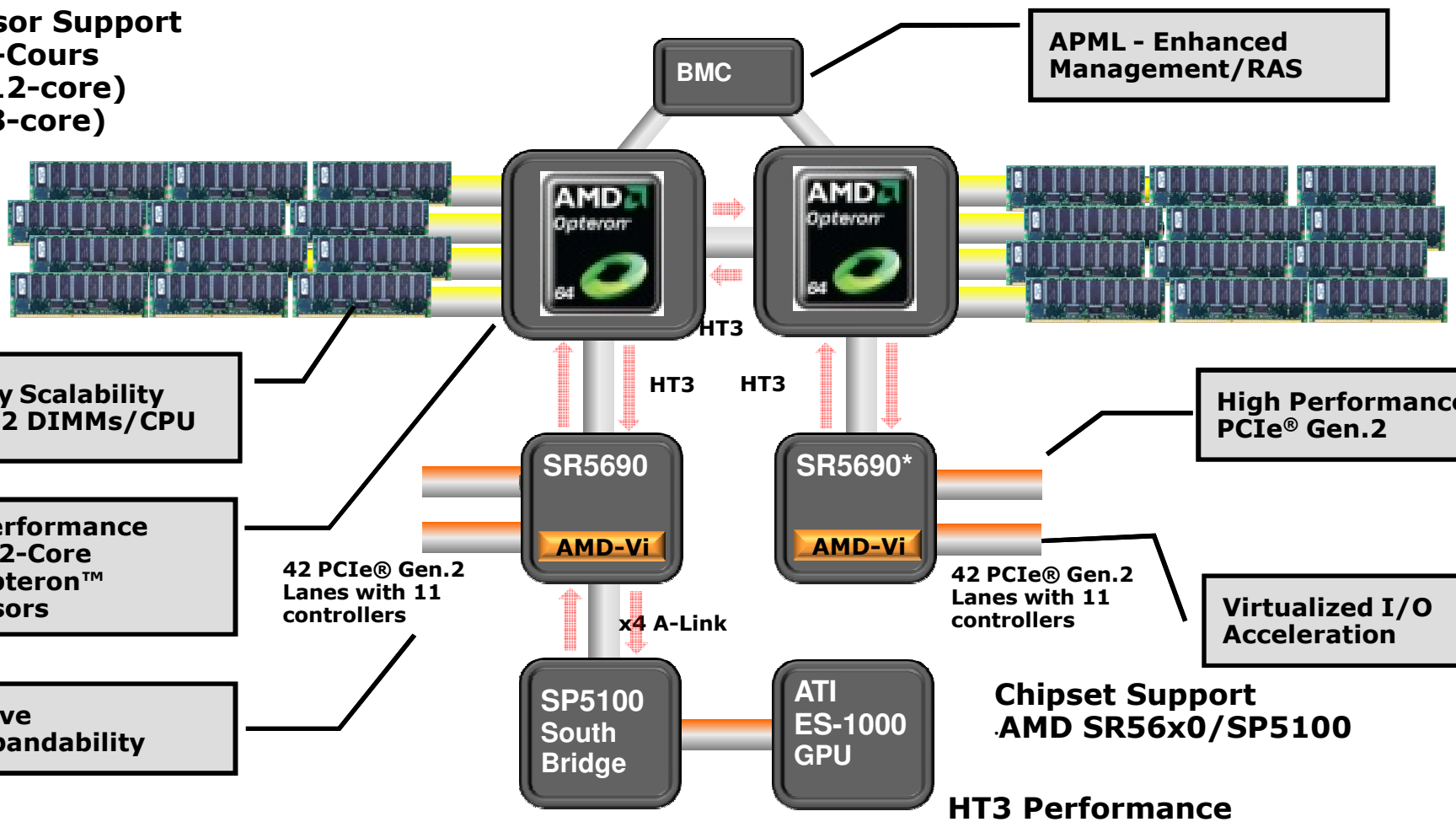
- Magny-Cours is the first 12-core processor, and is supported in RHEL5.5 and upcoming RHEL releases
- Superior performance with RHEL on Magny-Cours platform
- I/O Virtualization is an integral part of the current Magny-Cours platform
- Next generation AMD IOMMU provides another level of I/O Virtualization functionality
  - Demand Paging for smart devices (NICs, GPGPU, …)
  - Two levels of Page Table walking
    - Guest User Level I/O direct access to devices

THANK YOU

BACKUP

Socket G34 “Maranello” Platform:

Processor Support
- **Magny-Cours**
  - **(12-core)**
  - **(8-core)**



Chipset Support
- **AMD SR56x0/SP5100**

HT3 Performance

	8 Core	12 Core
HT3	25.6 GB/s (6.4 GT/s)	25.6 GB/s (6.4 GT/s)

Registered DDR3 Memory Support

	Max Frequency	Max Capacity	Max Bandwidth
Magny-Cours 12 and 8 core	1333MHz with 48GB per CPU	128GB per CPU at 1066MHz	42.7GB/s @1333MHz per CPU

*** Dual SR5690 is optional**

What is Device Isolation?

Device Isolation

refers to mapping a device to a particular guest, while ensuring the guest stays in its address space and maintains the integrity of other guests. In the bare metal scenarios the AMD IOMMU driver provides security by limiting a device's memory accesses.

What is Direct Device Assignment?

Direct Device Assignment

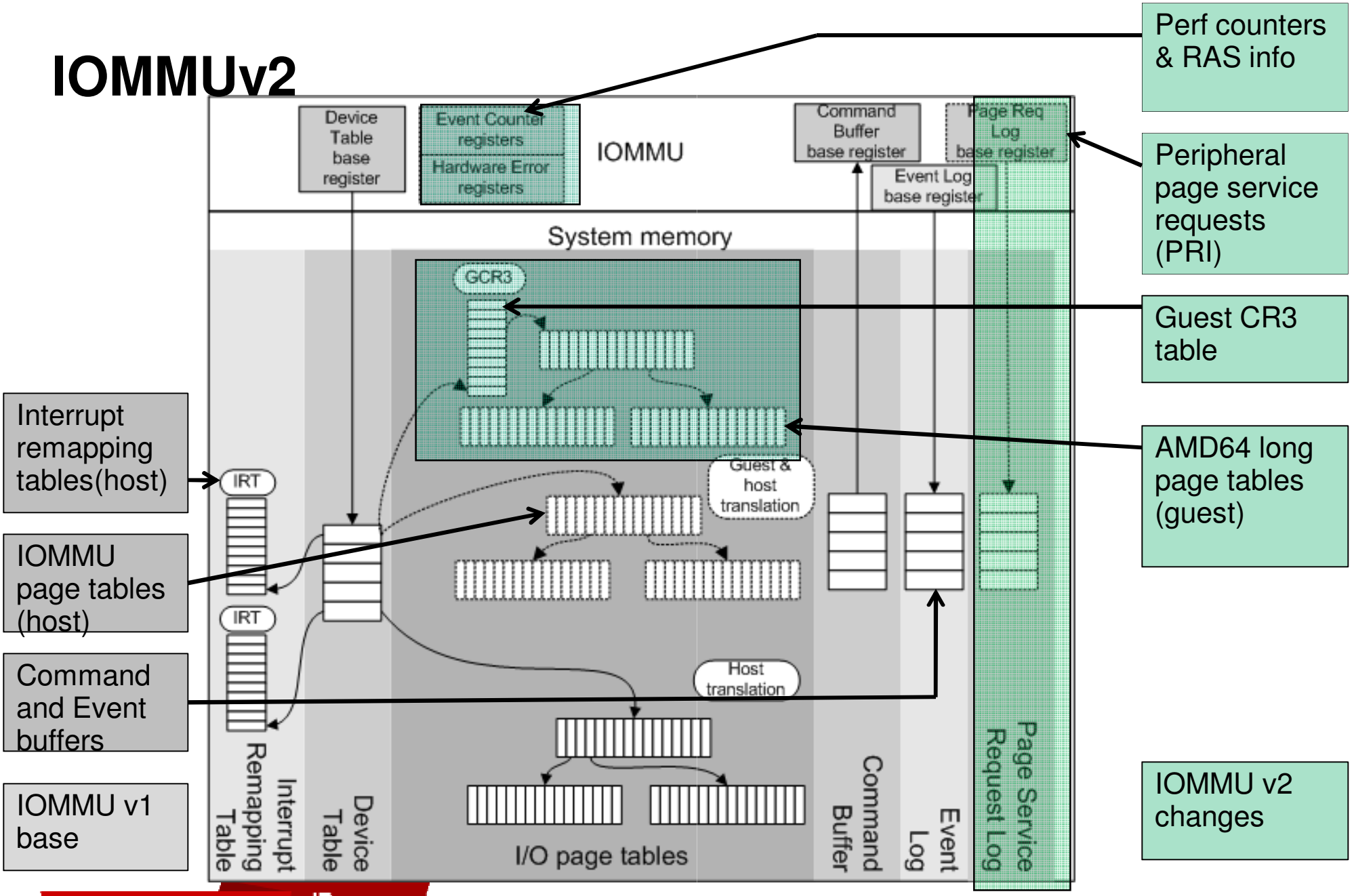
is the ability to directly assign a physical device to a guest OS. The required address space translation is handled transparently. Using IOMMU, the device address space is the same as a guest's physical address space.

What is Interrupt Remapping?

Interrupt Remapping

allows the IOMMU to separate device interrupts that are already shared by different devices. It remaps a shared interrupt to an exclusive vector to help ensure the interrupt is delivered to a particular guest OS.

IOMMUv2


Perf counters & RAS info
Peripheral page service requests (PRI)
Guest CR3 table
AMD64 long page tables (guest)
IOMMU v2 changes
Interrupt remapping tables(host)
IOMMU page tables (host)
Command and Event buffers
IOMMU v1 base
Device Table base register
Event Counter registers
Hardware Error registers
IOMMU
Command Buffer base register
Page Req Log base register
Event Log base register
System memory
GCR3
Guest & host translation
IRT
IRT
Host translation
Interrupt Remapping Table
Device Table
I/O page tables
Command Buffer
Event Log
Page Service Request Log