Cisco Unified Computing System (UCS)

A Complete Reference Guide to the Cisco Data Center Virtualization Server Architecture

Silvano Gai
Tommi Salli
Roger Andersson

ciscopress.com

Cisco Unified Computing System (UCS)
A Complete Reference Guide to the Data Center Virtualization Server Architecture

Silvano Gai
Tommi Salli
Roger Andersson

Published by:

Cisco Press

201 West 103rd Street

Indianapolis, IN 46290 USA

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing May 2010

Library of Congress Cataloging-in-Publication Data

Gai, Silvano.

Unified computing system (UCS) : a data center virtualization server /

Silvano Gai, Tommi Salli, Roger Andersson.

p. cm.

Includes bibliographical references.

ISBN 978-1-58714-193-5 (pbk.)

1. Virtual computer systems. 2. Parallel processing (Electronic

computers) I. Salli, Tommi, 1977- II. Andersson, Roger, 1970- III.

Title.

QA76.9.V5G35 2010

004'.35--dc22

2010016455

ISBN-10: 1-58714-193-0

ISBN-13: 978-1-58714-193-5

Warning and Disclaimer

This book is designed to provide information about the Cisco Unified Computing System (UCS). Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc., shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Preface

This book is the result of the work done by the authors initially at Nuova Systems and subsequently at Cisco Systems on Project California, officially known as Cisco Unified Computing Systems (UCS).

UCS is a innovative technology platform that consolidates many traditional data center technical skill sets into one system. For this reason, the authors, from three very different backgrounds (and even different countries), have decided to combine their knowledge to publish this book together.

The book describes UCS from an educational view: We have tried to provide updated material about all server components and new data center technologies, as well as how these components and technologies are used to build a state of the art data center server.

We wish to express our thanks to the many engineering and marketing people from both Nuova Systems and Cisco Systems with whom we have collaborated with during the recent years.

UCS would not have been possible without the determination of the Cisco management team. They understood the opportunity for Cisco in entering the server market and decided to pursue it. Our gratitude goes out to them.

Nomenclature

Engineering projects are identified from inception to announcement by fantasy names that are often names of geographical places to avoid any trademark issue. Project California is not an exception. Project California or simply “California” is the name of the overall systems and city names, such as Palo and Menlo, are used to identify specific components.

Before the launch, Cisco decided to assign project California the official name of “Unified Computing System (UCS)”, but the term California will continue to be used informally.

Cisco UCS Manager

The Cisco UCS Manager software integrates the components of the Cisco Unified Computing System into a single, seamless entity. The UCS Manager is described in Chapter 7.

Cisco UCS 6100 Series Fabric Interconnects

The Cisco UCS 6100 Series Fabric Interconnects are a family of line-rate, low-latency, lossless 10 Gigabit Ethernet, Cisco Data Center Bridging, and Fiber Channel over Ethernet (FCoE) switches, designed to consolidate the I/O at the system level. The Fabric Interconnects are described in Chapter 5.

Cisco UCS 2100 Series Fabric Extenders

The Cisco UCS 2100 Series Fabric Extender provides an extension of the I/O fabric into the blade server enclosure providing a direct 10 Gigabit Cisco Data Center Bridging connection between the blade servers and the Fabric Interconnects, simplifying diagnostics, cabling, and management. The Fabric Extenders are described in Chapter 5.

Cisco UCS 5100 Series Blade Server Enclosures

The Cisco UCS 5100 blade server Enclosures physically house blade servers and up to two fabric extenders. The Blade Server Enclosures are described in Chapter 5.

Cisco UCS B-Series Blade Servers

The Cisco UCS B-Series blade servers are designed for compatibility, performance, energy efficiency, large memory footprints, manageability, and unified I/O connectivity. They are based on Intel® Xeon® 5500 (Nehalem-EP), 5600 (Westmere-EP), and 7500 (Nehalem-EX) series processors (described in Chapter 2). The blade servers are described in Chapter 5.

Cisco UCS C-Series Rack Servers

The Cisco UCS C-Series rack servers are designed for compatibility, performance, energy efficiency, large memory footprints, manageability, expandability, and unified I/O connectivity. They are based on Intel® Xeon® 5500 (Nehalem-EP), 5600 (Westmere-EP), and 7500 (Nehalem-EX) series processors (described in Chapter 2). The rack servers are described in Chapter 6.

Cisco UCS Adapters

The Cisco UCS Adapters are installed on the UCS B-Series blade servers in order to provide I/O connectivity through the UCS 2100 Series Fabric Extenders. The different adapters are described in Chapter 4.

Chapter 2

Server Architectures

Processor, memory, and I/O are the three most important subsystems in a server from a performance perspective. At any given point in time, one of them tends to become a bottleneck from the performance perspective. We often hear of applications that are CPU-bound, memory-bound, or I/O-bound.

In this chapter, we will examine these three subsystems with particular reference to servers built according to the IA-32 (Intel® Architecture, 32-bit), often generically called x86 architecture. In particular, we will describe the most recent generation of Intel® processors compatible with the IA-32 architecture; i.e., the Intel® microarchitecture (formerly known by the code-name Nehalem).[1]

The Nehalem microarchitecture (see "Intel Microarchitectures" in Chapter 2, page 45) and its variation, the Westmere microarchitecture, include three families of processors that are used on the Cisco UCS: the Nehalem-EP, the Nehalem-EX, and the Westmere-EP. Table 2-1 summarizes the main characteristics of these processors.

Table 2-1 UCS Processors

	Nehalem-EP	Westmere-EP	Nehalem-EX	Nehalem-EX
Commercial Name	Xeon® 5500	Xeon® 5600	Xeon® 6500	Xeon® 7500
Max Sockets Supported	2	2	2	8
Max Cores per Socket	4	6	8	8
Max Threads per Socket	8	12	16	16
MB Cache (Level 3)	8	12	18	24
Max # of Memory DIMMs	18	18	32	128

1 The authors are thankful to Intel Corporation for the information and the material provided to edit this chapter. Most of the pictures are courtesy of Intel Corporation.

Figure 2-1 An Intel® Xeon® 5500 Processor

The Processor Evolution

Modern processors or CPUs (Central Processing Units) are built using the latest silicon technology and pack millions of transistors and megabytes of memory on a single die (blocks of semiconducting material that contains a processor).

Multiple dies are fabricated together in a silicon wafer; each die is cut out individually, tested, and assembled in a ceramic package. This involves mounting the die, connecting the die pads to the pins on the package, and sealing the die.

At this point, the processor in its package is ready to be sold and mounted on servers. Figure 2-1 shows a packaged Intel® Xeon® 5500.

Sockets

Processors are installed on the motherboard using a mounting/interconnection structure known as a "socket." Figure 2-2 shows a socket used for an Intel® Processor. This allows the customers to personalize a server motherboard by installing processors with different clock speeds and power consumption,

The number of sockets present on a server motherboard determines how many processors can be installed. Originally, servers had a single socket, but more recently, to increase server performance, 2-, 4-, and 8-socket servers have appeared on the market.

In the evolution of processor architecture, for a long period, performance improvements were strictly related to clock frequency increases. The higher the clock frequency, the shorter the time it takes to make a computation, and therefore the higher the performance.

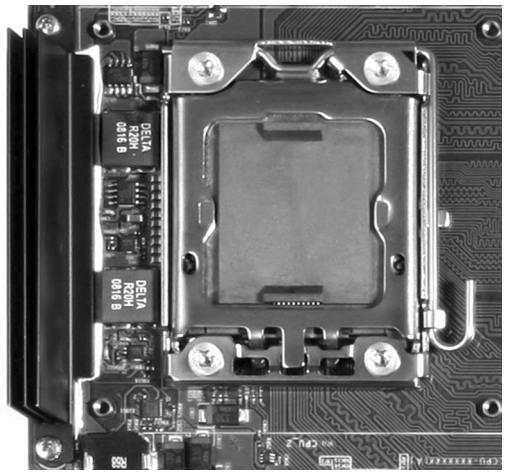


Figure 2-2 An Intel® processor socket

As clock frequencies approached a few GigaHertz, it became apparent the physics involved would limit further improvement in this area. Therefore, alternative ways to increase performance had to be identified.

Cores

The constant shrinking of the transistor size (Nehalem uses a 45-nanometer technology; Westmere uses a 32-nanometer technology) has allowed the integration of millions of transistors on a single die. One way to utilize this abundance of transistors is to replicate the basic CPU (the "core") multiple times on the same die.

Multi-core processors (see Figure 2-3) are now common in the market. Each processor (aka socket) contains multiple CPU cores (2, 4, 6, and 8 are typical numbers). Each core is associated with a level 1 (L1) cache. Caches are small fast memories used to reduce the average time to access the main memory. The cores generally share a larger level 2 (L2) or level 3 (L3) cache, the bus interface, and the external die connections.

In modern servers, the number of cores is the product of the number of sockets times the number of cores per socket. For example, servers based on Intel® Xeon® Processor 5500 Series (Nehalem-EP) typically use two sockets and four cores per sockets for a total of eight cores. With the Intel® Xeon® 7500 (Nehalem-EX), 8 sockets each with 8 cores are supported for a total of 64 cores.

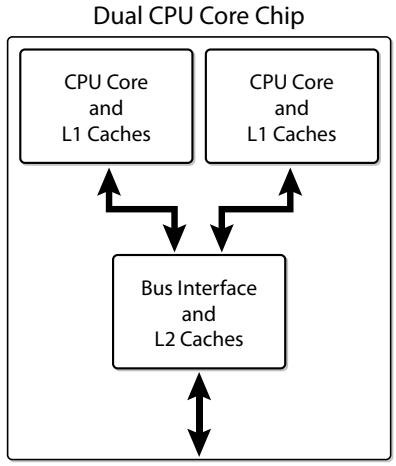


Figure 2-3 Two CPU cores in a socket

Figure 2-4 shows a more detailed view of a dual-core processor. The CPU's main components (instruction fetching, decoding, and execution) are duplicated, but the access to the system buses is common.

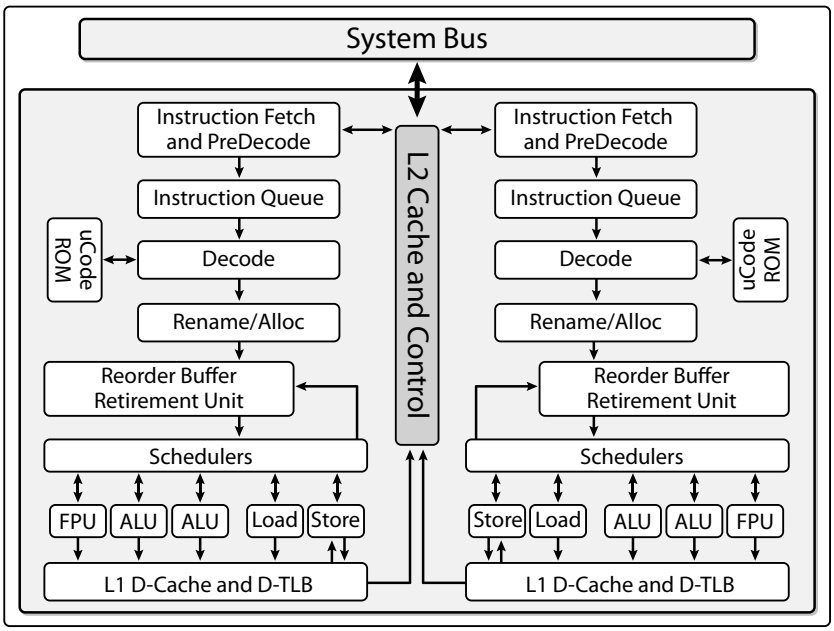


Figure 2-4 Architecture of a dual-core processor

Threads

To better understand the implication of multi-core architecture, let us consider how programs are executed. A server will run a kernel (e.g., Linux®, Windows®) and multiple processes. Each process can be further subdivided into "threads". Threads are the minimum unit of work allocation to cores. A thread needs to execute on a single core, and it cannot be further partitioned among multiple cores (see Figure 2-5).

Processes can be single-threaded or multi-threaded. A process that is single thread process can execute in only one core and is limited by the performance of that core. A multi-threaded process can execute on multiple cores at the same time, and therefore its performance can exceed the performance of a single core.

Since many applications are single-threaded, a multi-socket, multi-core architecture is typically convenient in an environment where multiple processes are present. This is always true in a virtualized environment, where a hypervisor allows consolidating multiple logical servers into a single physical server creating an environment with multiple processes and multiple threads.

Intel® Hyper-Threading Technology

While a single thread cannot be split between two cores, some modern processors allow running two threads on the same core at the same time. Each core has multiple execution units capable of working in parallel, and it is rare that a single thread will keep all the resources busy.

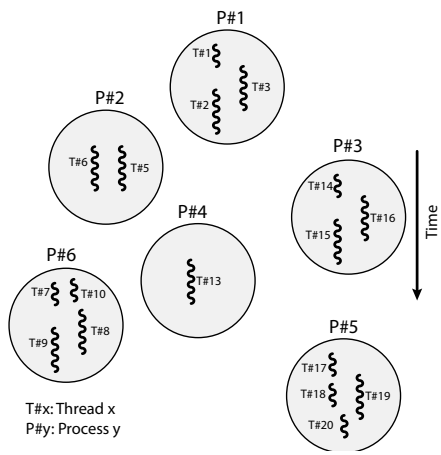


Figure 2-5 Processes and threads

Figure 2-6 shows how the Intel® Hyper-Threading Technology works. Two threads execute at the same time on the same core, and they use different resources, thus increasing the throughput.

Front-Side Bus

In the presence of multi-sockets and multi-cores, it is important to understand how the memory is accessed and how communication between two different cores work.

Figure 2-7 shows the architecture used in the past by many Intel® processors, known as the Front-Side Bus (FSB). In the FSB architecture, all traffic is sent across a single, shared bidirectional bus. In modern processors, this is a 64-bit wide bus that is operated at 4X the bus clock speed. In certain products, the FSB is operated at an information transfer rate of up to 1.6 GT/s (Giga Transactions per second, i.e., 12.8 GB/s).

The FSB is connected to all the processors and to the chipset called the Northbridge (aka MCH: Memory Controller Hub). The Northbridge connects the memory that is shared across all the processors.

One of the advantages of this architecture is that each processor has knowledge of all the memory accesses of all the other processors in the system. Each processor can implement a cache coherency algorithm to keep its internal caches in synch with the external memory and with the caches of all other processors.

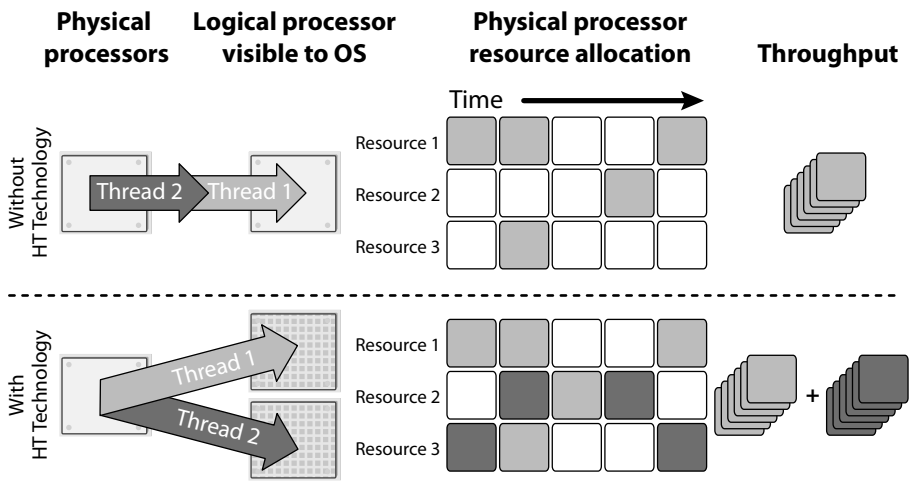


Figure 2-6 Intel® Hyper-Threading Technology

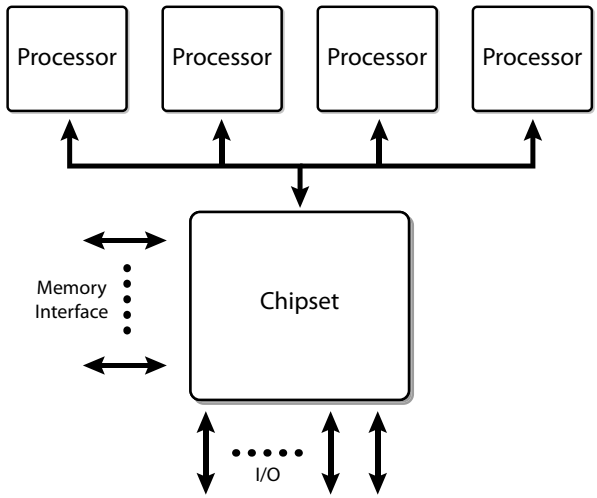


Figure 2-7 A server platform based on a front-side bus

Platforms designed in this manner have to contend with the shared nature of the bus. As signaling speeds on the bus increase, it becomes more difficult to implement and connect the desired number of devices. In addition, as processor and chipset performance increases, the traffic flowing on the FSB also increases. This results in increased congestion on the FSB, since the bus is a shared resource.

Dual Independent Buses

To further increase the bandwidth, the single shared bus evolved into the Dual Independent Buses (DIB) architecture depicted in Figure 2-8, which essentially doubles the available bandwidth.

However, with two buses, all the cache consistency traffic has to be broadcasted on both buses, thus reducing the overall effective bandwidth. To minimize this problem, "snoop filters" are employed in the chipset to reduce the bandwidth loading.

When a cache miss occurs, a snoop is put on the FSB of the originating processor. The snoop filter intercepts the snoop, and determines if it needs to pass along the snoop to the other FSB. If the read request is satisfied with the other processor on the same FSB, the snoop filter access is cancelled. If the other processor on the same FSB does not satisfy the read request, the snoop filter determines the next course of action. If the read request misses the snoop filter, data is returned directly from memory. If the snoop filter indicates that the target cache line of the

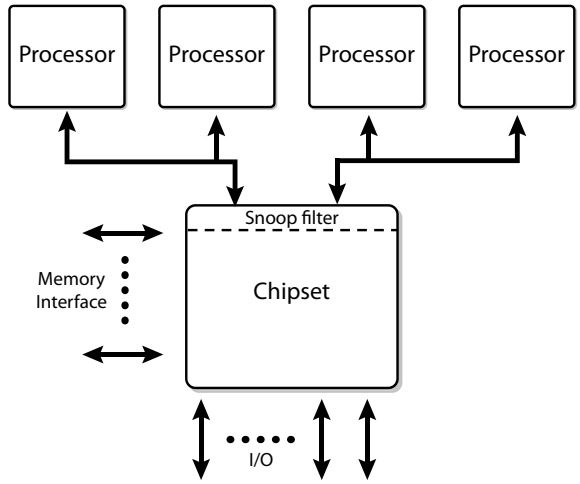


Figure 2-8 A server platform based on Dual Independent Buses

request could exist on the other FSB, the snoop filter will reflect the snoop across to the other segment. If the other segment still has the cache line, it is routed to the requesting FSB. If the other segment no longer owns the target cache line, data is returned from memory. Because the protocol is write-invalidate, write requests must always be propagated to any FSB that has a copy of the cache line in question.

Dedicated High-Speed Interconnect

The next step of the evolution is the Dedicated High-Speed Interconnect (DHSI), as shown in Figure 2-9.

DHSI-based platforms use four independent FSBs, one for each processor in the platform. Snoop filters are employed to achieve good bandwidth scaling.

The FSBs remains electrically the same, but are now used in a point-to-point configuration.

Platforms designed using this approach still must deal with the electrical signaling challenges of the fast FSB. DHSI drives up also the pin count on the chipset and require extensive PCB routing to establish all these connections using the wide FSBs.

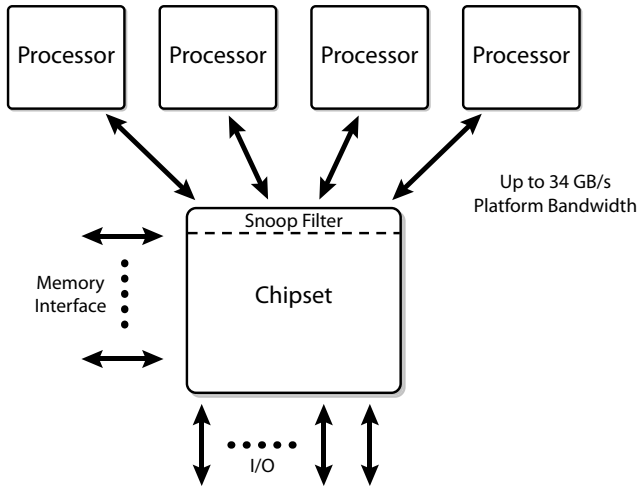


Figure 2-9 A server platform based on DHSI

Intel® QuickPath Interconnect

With the introduction of the Intel Core i7 processor, a new system architecture has been adopted for many Intel products. This is known as the Intel® QuickPath Interconnect (Intel® QPI). This architecture utilizes multiple high-speed uni-directional links interconnecting the processors and the chipset. With this architecture, there is also the realization that:

- A common memory controller for multiple sockets and multiple cores is a bottleneck.
- Introducing multiple distributed memory controllers would best match the memory needs of multi-core processors.
- In most cases, having a memory controller integrated into the processor package would boost performance.
- Providing effective methods to deal with the coherency issues of multi-socket systems is vital to enabling larger-scale systems.

Figure 2-10 gives an example functional diagram of a processor with multiple cores, an integrated memory controller, and multiple Intel® QPI links to other system resources.

In this architecture, all cores inside a socket share IMCs (Integrated Memory Controllers) that may have multiple memory interfaces (i.e., memory buses).

IMCs may have different external connections:

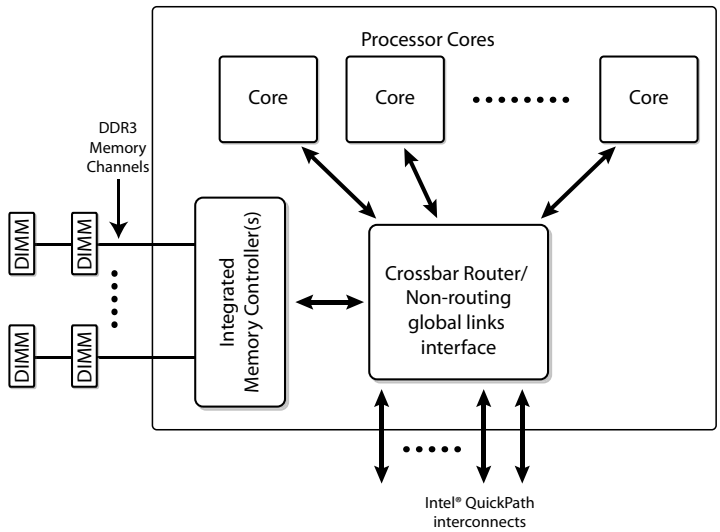


Figure 2-10 Processor with Intel® QPI and DDR3 memory channels

- DDR3 memory channels: In this case, the DDR3 DIMMs (see the next section) are directly connected to the sockets, as shown in Figure 2-12. This architecture is used in Nehalem-EP (Xeon 5500) and Westmere-EP (Xeon 5600).
- High-speed serial memory channels, as shown in Figure 2-11. In this case, an external chip (SMB: Scalable Memory Buffer) creates DDR3 memory channels where the DDR3 DIMMs are connected. This architecture is used in Nehalem-EX (Xeon 7500).

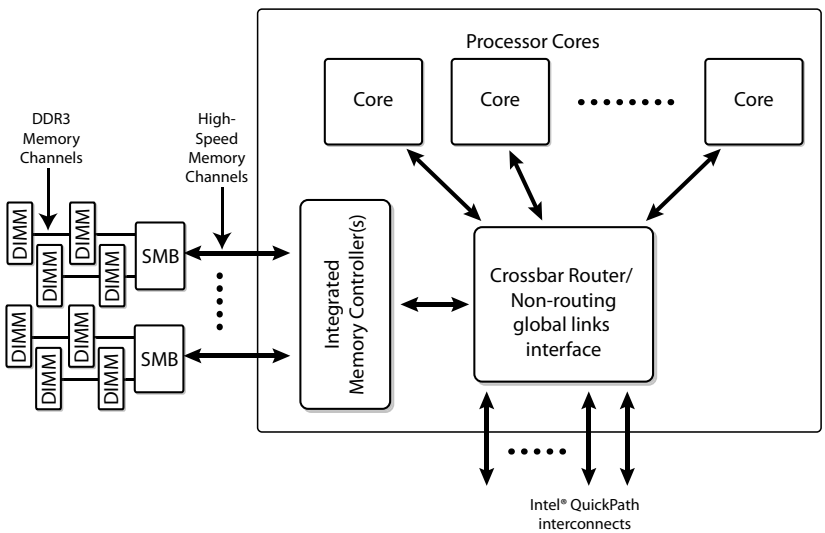


Figure 2-11 Processors with high-speed memory channels

IMCs and cores in different sockets talk to each other using Intel® QPI.

Processors implementing Intel® QPI also have full access to the memory of every other processor, while maintaining cache coherency. This architecture is also called "cache-coherent NUMA (Non-Uniform Memory Architecture)"—i.e., the memory interconnection system guarantees that the memory and all the potentially cached copies are always coherent.

Intel® QPI is a point-to-point interconnection and messaging scheme that uses a point-to-point differential current-mode signaling. On current implementation, each link is composed of 20 lanes per direction capable of up to 25.6 GB/s or 6.4 GT/s (Giga Transactions/second); (see "Platform Architecture" in Chapter 2, page 46).

Intel® QPI uses point-to-point links and therefore requires an internal crossbar router in the socket (see Figure 2-10) to provide global memory reachability. This route-through capability allows one to build systems without requiring a fully connected topology.

Figure 2-12 shows a configuration of four Intel® Nehalem EX, each processor has four QPI and interconnects with the three other processors and with the Boxboro-EX chipsets (SMB components are present, but not shown).

The Memory Subsystem

The electronic industry has put a significant effort into manufacturing memory subsystems capable of keeping up with the low access time required by modern processors and the high capacity required by today's applications.

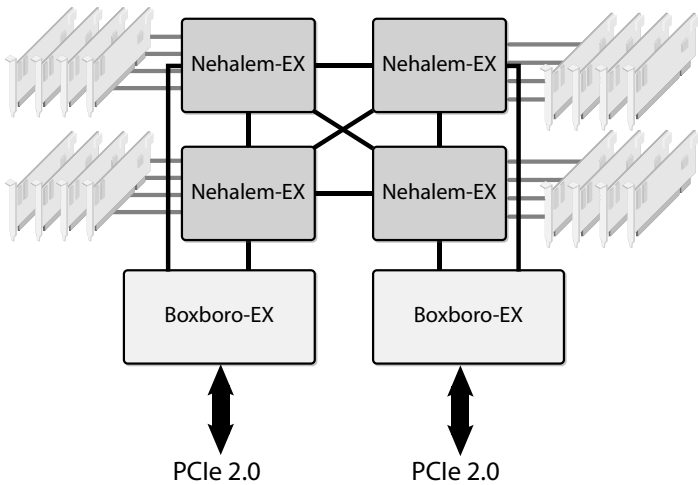


Figure 2-12 Four-socket Nehalem EX

Before proceeding with the explanation of current memory subsystems, it is important to introduce a glossary of the most commonly used terms:

- RAM (Random Access Memory)
- SRAM (Static RAM)
- DRAM (Dynamic RAM)
- SDRAM (Synchronous DRAM)
- SIMM (Single Inline Memory Module)
- DIMM (Dual Inline Memory Module)
- UDIMM (Unbuffered DIMM)
- RDIMM (Registered DIMM)
- DDR (Double Data Rate SDRAM)
- DDR2 (Second Generation DDR)
- DDR3 (Third Generation DDR)

In particular, the Joint Electron Device Engineering Council (JEDEC) is the semiconductor engineering standardization body that has been active in this field. JEDEC Standard 21 [21], [22] specifies semiconductor memories from the 256 bits SRAM to the latest DDR3 modules.

The memory subsystem of modern servers is composed of RAMs (Random Access Memories), i.e., integrated circuits (aka ICs or chips) that allow the data to be accessed in any order, in a constant time, regardless of its physical location. RAMs can be static or dynamic [27], [28], [29], [30].

SRAMs

SRAMs (Static RAMs) are generally very fast, but smaller capacity (few megabytes) than DRAM (see next section), and they have a chip structure that maintains the information as long as power is maintained. They are not large enough to be used for the main memory of a server.

DRAMs

DRAMs (Dynamic RAMs) are the only choice for servers. The term "dynamic" indicates that the information is stored on capacitors within an integrated circuit. Since capacitors discharge over time, due to leakage currents, the capacitors need to be recharged ("refreshed") periodically to avoid data loss. The memory controller is normally in charge of the refresh operations.

SDRAMs

SDRAMs (Synchronous DRAMs) are the most commonly used DRAM. SDRAMs have a synchronous interface, meaning that their operation is synchronized with a clock signal. The clock is used to drive an internal finite state machine that pipelines memory accesses. Pipelining means that the chip can accept a new memory access before it has finished processing the previous one. This greatly improves the performance of SDRAMs compared to classical DRAMs.

DDR2 and DDR3 are the two most commonly used SDRAMs (see "DDR2 and DDR3" in Chapter 2, page 41 [23]).

Figure 2-13 shows the internal architecture of a DRAM chip.

The memory array is composed of memory cells organized in a matrix. Each cell has a row and a column address. Each bit is stored in a capacitor (i.e., storage element).

To improve performance and to reduce power consumption, the memory array is split into multiple "banks." Figure 2-14 shows a 4-bank and an 8-bank organization.

DDR2 chips have four internal memory banks and DDR3 chips have eight internal memory banks.

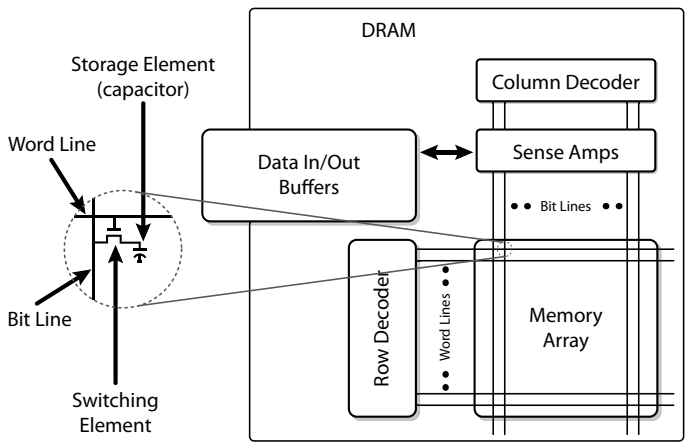


Figure 2-13 Internal architecture of a DRAM chip

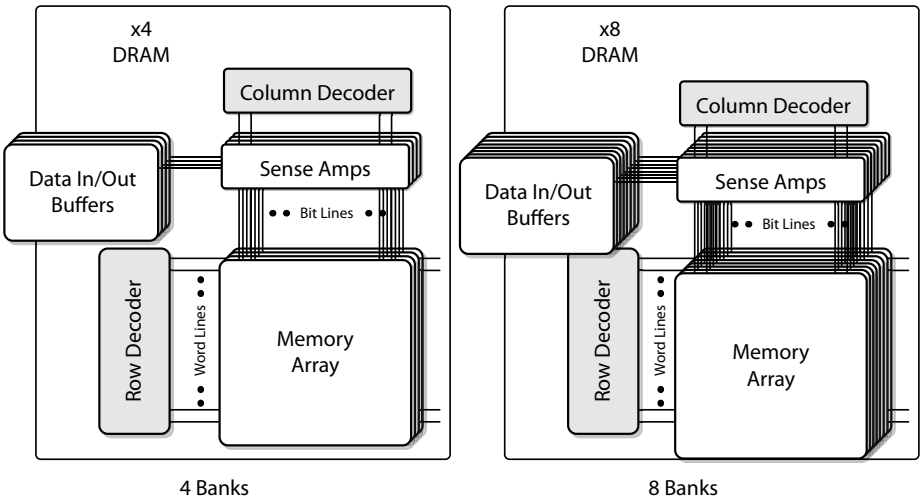


Figure 2-14 Memory banks

DIMMs

Multiple memory chips need to be assembled together to build a memory subsystem. They are organized in small boards known as DIMMs (Dual Inline Memory Modules).

Figure 2-15 shows the classical organization of a memory subsystem [24]. For example, a memory controller connects four DIMMs each composed of multiple DRAM chips. The memory controller (that may also integrate the clock driver) has an address bus, a data bus, and a command (aka control) bus. It is in charge of reading, writing, and refreshing the information stored in the DIMMs.

Figure 2-16 is an example of the connection between a memory controller and a DDR3 DIMM. The DIMM is composed of eight DRAM chips, each capable of storing eight bits of data for a total of 64 bits per memory word (width of the memory data bus). The address bus has 15 bits and it carries, at different times, the "row address" or the "column address" for a total of 30 address bits. In addition, three bits of bank address allow accessing the eight banks inside each DDR3 chip. They can be considered equivalent to address bits raising the total addressing capability of the controller to eight Giga words (i.e., 512 Gbits, or 64 GB). Even if the memory controller has this addressing capability, the DDR3 chips available on the market are significantly smaller. Finally, RAS (Row Address Selection), CAS (Column Address Selection), WE (Write Enabled), etc. are the command bus wires.

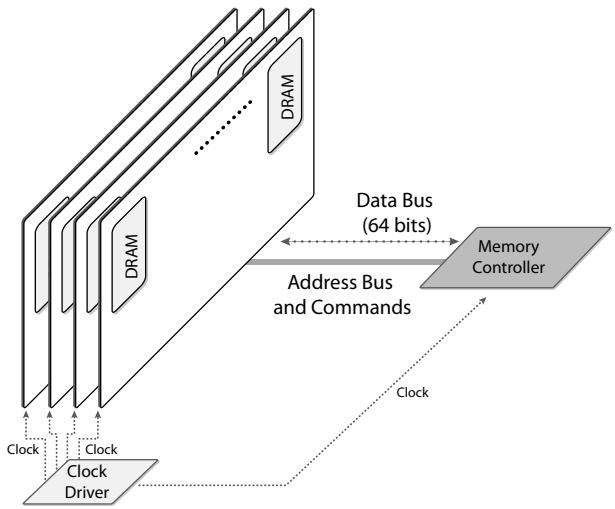


Figure 2-15 Example of a memory subsystem

Figure 2-17 shows a schematic depiction of a DIMM.

The front view shows the eight DDR3 chips each providing eight bits of information (normally indicated by "x8"). The side view shows that the chips are on one side of the board for a total of eight chips (i.e., 64 bits).

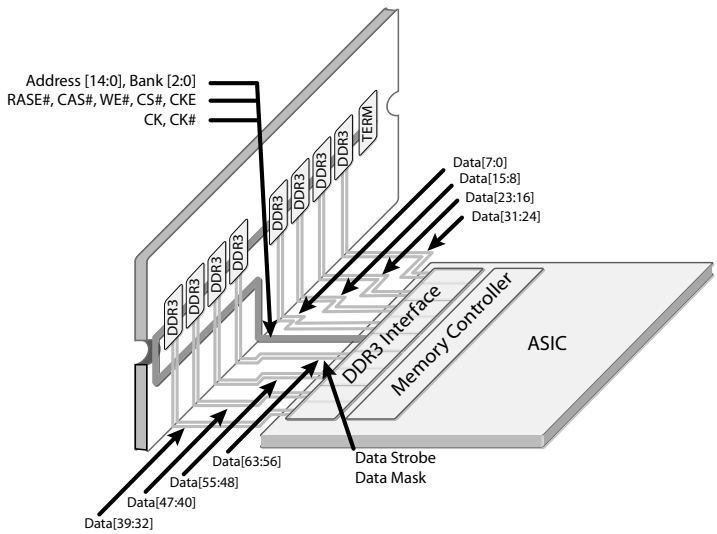


Figure 2-16 Example of a DDR3 memory controller

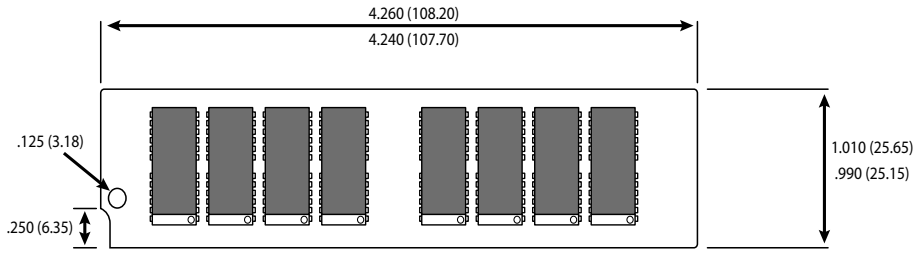


Figure 2-17 A DIMM

ECC and Chipkill®

Data integrity is a major concern in server architecture. Very often extra memory chips are installed on the DIMM to detect and recover memory errors. The most common arrangement is to add 8 bits of ECC (Error Correcting Code) to expand the memory word from 64 to 72 bits. This allows the implementation of codes like the Hamming code that allows a single-bit error to be corrected and double-bit errors to be detected. These codes are also known as SEC/DED (Single Error Correction / Double Error Detection).

With a careful organization of how the memory words are written in the memory chips, ECC can be used to protect from any single memory chip that fails and any number of multi-bit errors from any portion of a single memory chip. This feature has several different names [24], [25], [26]:

- Chipkill® is the IBM® trademark.
- Oracle® calls it Extended ECC.
- HP® calls it Chipspare®.
- A similar feature from Intel is called Intel® x4 Single Device Data Correction (Intel® x4 SDDC).

Chipkill® performs this function by bit-scattering the bits of an ECC word across multiple memory chips, such that the failure of any single memory chip will affect only one ECC bit. This allows memory contents to be reconstructed despite the complete failure of one chip.

While a complete discussion of this technology is beyond the scope of this book, an example can give an idea of how it works. Figure 2-18 shows a memory controller that reads and writes 128 bits of useful data at each memory access, and 144 bits when ECC is added. The 144 bits can be divided in 4 memory words of 36 bits. Each memory word will be SEC/DED. By using two DIMMs, each with 18 4-bit chips, it is possible to reshuffle the bits as shown in Figure 2-18. If a chip fails, there will be one error in each of the four words, but since the words are SEC-DEC, each of the four words can correct an error and therefore all the four errors will be corrected.

Memory Ranks

Going back to how the DIMMs are organized, an arrangement of chips that produce 64 bits of useful data (not counting the ECC) is called a "rank". To store more data on a DIMM, multiple ranks can be installed. There are single, dual, and quad-ranks DIMMs. Figure 2-19 shows three possible organizations.

In the first drawing, a rank of ECC RAM is built using nine eight-bit chips, a configuration that is also indicated 1Rx8. The second drawing shows a 1Rx4 arrangement in which 18 four-bit chips are used to build one rank. Finally, the third drawing shows a 2Rx8 in which 18 eight-bit chips are used to build two ranks.

Memory ranks are not selected using address bits, but "chip selects". Modern memory controllers have up to eight separate chip selects and therefore are capable of supporting up to eight ranks.

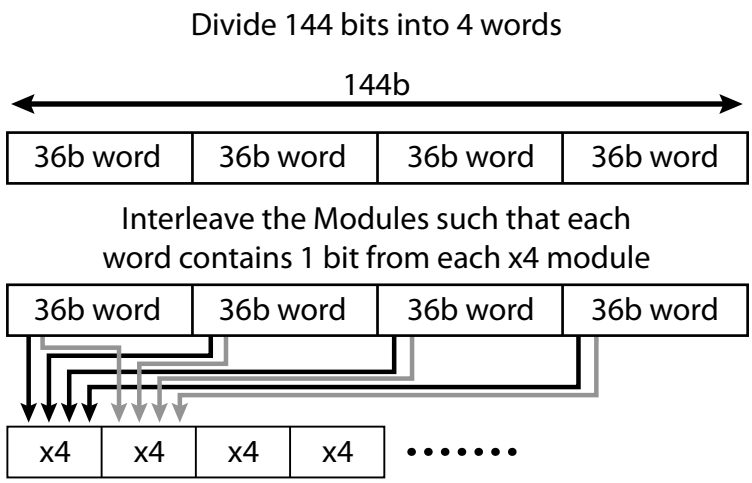


Figure 2-18 A Chipkill® example

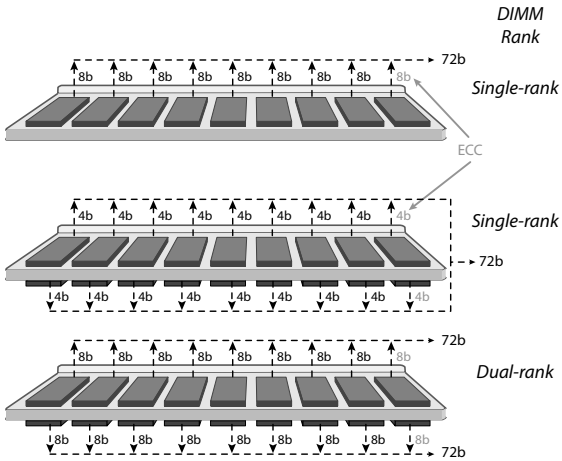


Figure 2-19 DIMMs and memory ranks

UDIMMs and RDIMMs

SDRAM DIMMs are further subdivided into UDIMMs (Unbuffered DIMMs) and RDIMM (Registered DIMMs). In UDIMMs, the memory chips are directly connected to the address and control buses, without any intermediate component.

RDIMM have additional components (registers) placed between the incoming address and control buses and the SDRAM components. These registers add one clock cycle of delay but they reduce the electrical load on the memory controller and allow more DIMM to be installed per memory controller.

RDIMM are typically more expensive because of the additional components, and they are usually found in servers where the need for scalability and stability outweighs the need for a low price.

Although any combination of Registered/Unbuffered and ECC/non-ECC is theoretically possible, most server-grade memory modules are both ECC and registered.

Figure 2-20 shows an ECC RDIMM. The registers are the chips indicated by the arrows; the nine memory chips indicate the presence of ECC.

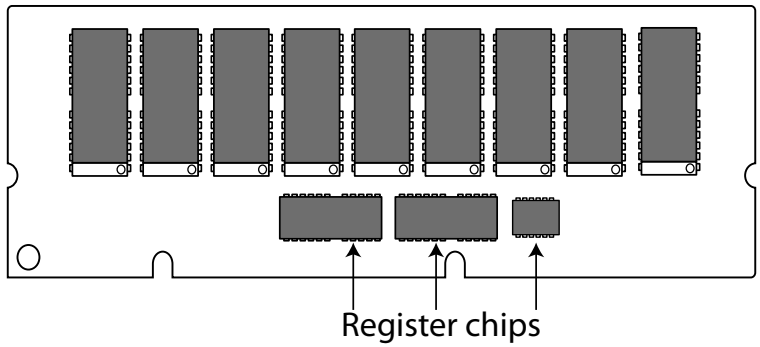


Figure 2-20 ECC RDIMM

DDR2 and DDR3

The first SDRAM technology was called SDR (Single Data Rate) to indicate that a single unit of data is transferred per each clock cycle. It was followed by the DDR (Double Data Rate) standard that achieves nearly twice the bandwidth of SDR by transferring data on the rising and falling edges of the clock signal, without increasing the clock frequency. DDR evolved into the two currently used standards: DDR2 and DDR3.

DDR2 SDRAMs (double-data-rate two synchronous dynamic random access memories) operate at 1.8 Volts and are packaged in 240 pins DIMM modules. They are capable of operating the external data bus at twice the data rate of DDR by improved bus signaling.

The rules are:

- Two data transfers per DRAM clock
- Eight bytes (64 bits) per data transfer

Table 2-2 shows the DDR2 standards.[2]

Table 2-2 DDR2 DIMMs

Standard name	DRAM clock	Million data transfers per second	Module name	Peak transfer rate GB/s
DDR2-400	200 MHz	400	PC2-3200	3.200
DDR2-533	266 MHz	533	PC2-4200	4.266

continues

2 Some of the DDR2 modules have two names, depending on the manufacturer.

Standard name	DRAM clock	Million data transfers per second	Module name	Peak transfer rate GB/s
DDR2-667	333 MHz	667	PC2-5300 PC2-5400	5.333
DDR2-800	400 MHz	800	PC2-6400	6.400
DDR2-1066	533 MHz	1,066	PC2-8500 PC2-8600	8.533

DDR3 SDRAMs (double-data-rate three synchronous dynamic random access memories) improve over DDR2 in the following areas:

- Reduced power consumption obtained by reducing the operating voltage to 1.5 volts.
- Increased memory density by introducing support for chips from 0.5 to 8 Gigabits; i.e., rank capacity up to 16 GB.
- Increased memory bandwidth by supporting a burst length = 8 words, compared to the burst length = 4 words of DDR2. The reason for the increase in burst length is to better match the increased external data transfer rate with the relatively constant internal access time. As the transfer rate increases, the burst length (the size of the transfer) must increase to not exceed the access rate of the DRAM core.

DDR3 DIMMs have 240 pins, the same number as DDR2, and are the same size, but they are electrically incompatible and have a different key notch location. In the future, DDR3 will also operate at faster clock rate. At the time of publishing, only DDR3-800, 1066, and 1333 are in production.

Table 2-3 summarizes the different DDR3 DIMM modules.

Table 2-3 DDR3 DIMMs

Standard name	RAM clock	Million data transfers per second	Module name	Peak transfer rate GB/s
DDR3-800	400 MHz	800	PC3-6400	6.400
DDR3-1066	533 MHz	1,066	PC3-8500	8.533
DDR3-1333	667 MHz	1,333	PC3-10600	10.667
DDR3-1600	800 MHz	1,600	PC3-12800	12.800
DDR3-1866	933 MHz	1,866	PC3-14900	14.900

The I/O Subsystem

The I/O subsystem is in charge of moving data from the server memory to the external world and vice versa. Historically this has been accomplished by providing in the server motherboards I/O buses compatible with the PCI (Peripheral Component Interconnect) standard. PCI was developed to interconnect peripheral devices to a computer system, it has been around for many years [1] and its current incarnation is called PCI-Express.

The Peripheral Component Interconnect Special Interest Group (PCI-SIG) is in charge of the development and enhancement of the PCI standard.

PCI Express®

PCI Express (PCIe®) [2] is a computer expansion card interface format designed to replace PCI, PCI-X, and AGP.

It removes one of the limitations that have plagued all the I/O consolidation attempts—i.e., the lack of I/O bandwidth in the server buses. It is supported by all current operating systems.

The previous bus-based topology of PCI and PCI-X is replaced by point-to-point connectivity. The resultant topology is a tree structure with a single root complex. The root complex is responsible for system configuration, enumeration of PCIe resources, and manages interrupts and errors for the PCIe tree. A root complex and its endpoints share a single address space and communicate through memory reads and writes, and interrupts.

PCIe connects two components with a point-to-point link. Links are composed of N lanes (a by-N link is composed of N lanes). Each lane contains two pairs of wires: one pair for transmission and one pair for reception.

Multiple PCIe lanes are normally provided by the SouthBridge (aka ICH: I/O Controller Hub) that implements the functionality of “root complex”.

Each lane connects to a PCI Express endpoint, to a PCI Express Switch or to a PCIe to PCI Bridge, as in Figure 2-21.

Different connectors are used according to the number of lanes. Figure 2-22 shows four different connectors and indicates the speeds achievable with PCIe 1.1.

In PCIe 1.1, the lanes run at 2.5 Gbps (2 Gbps at the datalink) and 16 lanes can be deployed in parallel (see Figure 2-23). This supports speeds from 2 Gbps (1x) to 32 Gbps (16x). Due to protocol overhead, 8x is required to support a 10 GE interface.

PCIe 2.0 (aka PCIe Gen 2) doubles the bandwidth per lane from 2 Gbit/s to 4 Gbit/s and extends the maximum number of lanes to 32x. It is shipping at the time of writing. A PCIe 4x is sufficient to support 10 GE.

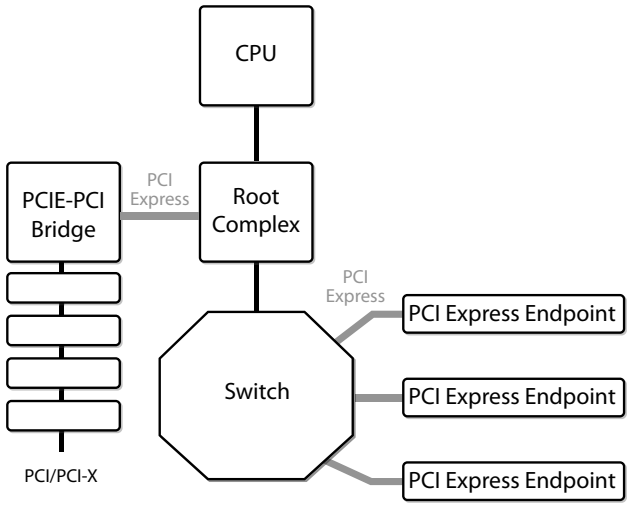


Figure 2-21 PCI-Express root complex

PCIe 3.0 will approximately double the bandwidth again. The final PCIe 3.0 specifications, including form factor specification updates, may be available by mid 2010, and could be seen in products starting in 2011 and beyond [3]. PCIe 3.0 will be required to effectively support 40 GE (Gigabit Ethernet), the next step in the evolution of Ethernet.

All the current deployments of PCI Express are Single Root (SR), i.e., a single I/O Controller Hub (ICH) controlling multiple endpoints.

PCI Express

Example Connectors

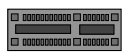
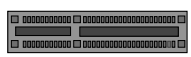
	Bandwidth	
x1	**Single Direction:** 2.5 Gbps/200 MBps **Dual Directions:** 5 Gbps/400 MBps	
x4	**Single Direction:** 10 Gbps/800 MBps **Dual Directions:** 20 Gbps/1.6 GBps	
x8	**Single Direction:** 20 Gbps/1.6 GBps **Dual Directions:** 40 Gbps/3.2 GBps	
x16	**Single Direction:** 40 Gbps/3.2 GBps **Dual Directions:** 80 Gbps/6.4 GBps	

Figure 2-22 PCI Express connectors

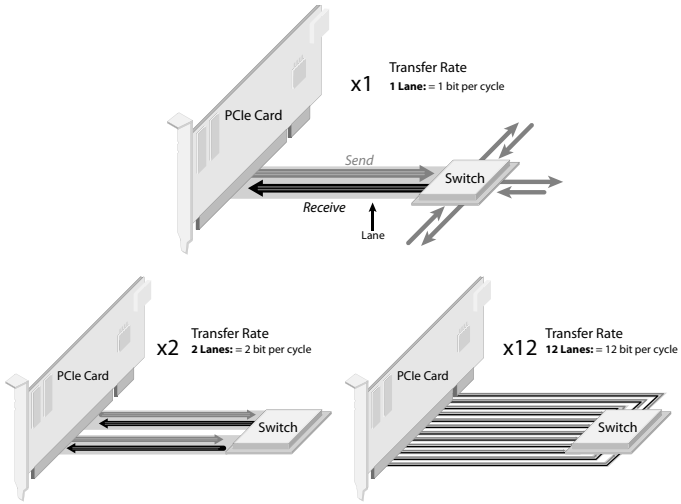


Figure 2-23 PCI Express lanes

Multi Root (MR) has been under development for a while, but it has not seen the light yet, and many question if it ever will, due to the lack of components and interest.

SR-IOV (Single Root I/O Virtualization) is another extremely relevant standard developed by PCI-SIG to be used in conjunction with Virtual Machines and Hypervisors. It is discussed in "DCBX: Data Center Bridging eXchange" in Chapter 3, page 73.

Intel Microarchitectures

The Cisco UCS uses the Intel® Processor belonging to the Nehalem and Westmere microarchitectures (more generally, the 32nm and 45nm Hi-k Intel® Core™ microarchitectures).

The Nehalem microarchitectures was introduced in servers in early 2009 and was one of the first architectures to use the new 45 nm (nanometer) silicon technology developed by Intel [32], [33], [34]. Nehalem processors span the range from high-end desktop applications, up through very large-scale server platforms. The codename is derived from the Nehalem River on the Pacific coast of northwest Oregon in the United States.

In Intel® parlance, processor developments are divided into "tick" and "tock" intervals, as in Figure 2-24. Tick is a technology that shrinks an existing processor, while tock is a new architecture done in the previous technology. Nehalem is the 45 nm tock. Westmere is the 32 nm tick following Nehalem.

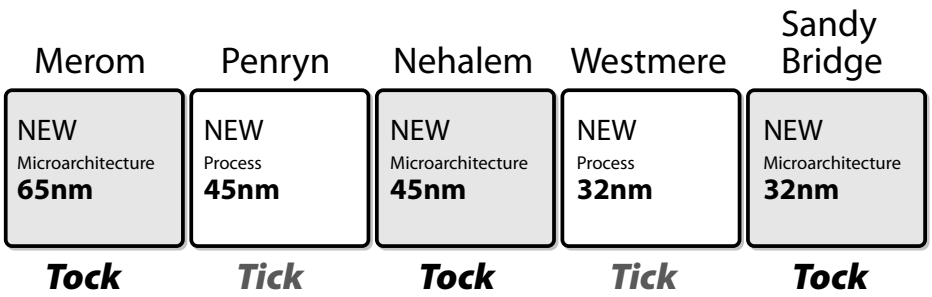


Figure 2-24 Intel® "Tick/Tock" processor development model

Nehalem and Westmere are a balance between different requirements:

- Performance of existing applications compared to emerging application (e.g., multimedia)
- Equally good support for applications that are lightly or heavily threaded
- Implementations that range from laptops to servers

They try to optimize for performance and at the same time reduce power consumption. This discussion is based on a nice Intel® Development Forum Tutorial [32]. In the rest of this chapter, the innovations are described in relation to the Nehalem microarchitecture, but they are also inherited in the Westmere architecture. There are few places where the architectures of Nehalem and Westmere differ and these will be highlighted.

Platform Architecture

It is the biggest platform architecture shift in about 10 years for Intel. The inclusion of multiple high-speed point-to-point connections, i.e., Intel® QuickPath Interconnect (see "Dedicated High-Speed Interconnect" in Chapter 2, page 30), along with the use of Integrated Memory Controllers (IMCs), is a fundamental departure from the FSB-based approach.

An example of a dual-socket Intel® Xeon® 5500 (Nehalem-EP) systems is shown in Figure 2-25. Please note the QPI links between the CPU sockets and from the CPU sockets to the I/O controller, and the memory DIMMs directly attached to the CPU sockets.

Integrated Memory Controller (IMC)

In Nehalem-EP and Westmere-EP, each socket has Integrated Memory Controller (IMC) that supports three DDR3 memory channels (see "DDR2 and DDR3" in Chapter 2, page 41). DDR3 memories run at higher frequency when compared with DDR2, thus higher memory bandwidth. In addition, for dual-socket architecture, there are two sets of memory controllers instead of one. All these improvements lead to a 3.4x bandwidth increase compared to the previous Intel® platform (see Figure 2-26).

This will continue to increase over time, as faster DDR3 becomes available. An integrated memory controller also makes a positive impact by reducing latency.

Power consumption is also reduced, since DDR3 is 1.5 Volt technology compared to 1.8 Volts of DDR2. Power consumption tends to go with the square of the voltage and therefore a 20% reduction in voltage causes approximately a 40% reduction in power.

Finally, the IMC supports both RDIMM and UDIMM with single, dual, or quad ranks (quad ranks is only supported on RDIMM; see "Memory Ranks" on page 39 and "UDIMMs and RDIMMs" on page 40, both from Chapter 2).

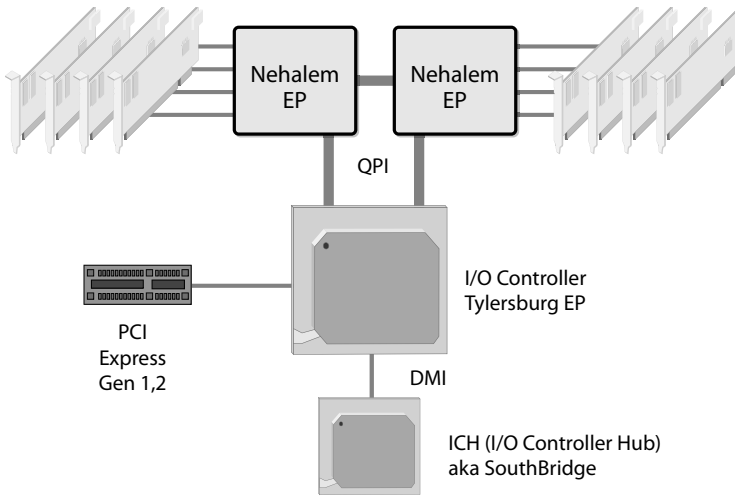


Figure 2-25 Two-socket Intel Xeon 5500 (Nehalem-EP)

Nehalem-EX has a similar, but not identical, architecture. In Nehalem-EX, there are two IMCs per socket. Each IMC supports two Intel® Scalable Memory Interconnects (SMIs) connected to two Scalable Memory Buffers (SMBs) for a total of four SMBs per socket (see Figure 2-27). Each SMB has two DDR3 buses, each connecting two RDIMMs. The total number of RDIMMs per socket is therefore sixteen.

The overall memory capacity of a Nehalem-EX system as a function of the number of sockets and of the RDIMM capacity is summarized in Table 2-4.

Table 2-4 Nehalem-EX Memory Capacity

	4GB RDIMM	8GB RDIMM	16GB RDIMM
2 sockets	128 GB	256 GB	512 GB
4 sockets	256 GB	512 GB	1 TB
8 sockets	512 GB	1 TB	2 TB

Intel® QuickPath Interconnect (QPI)

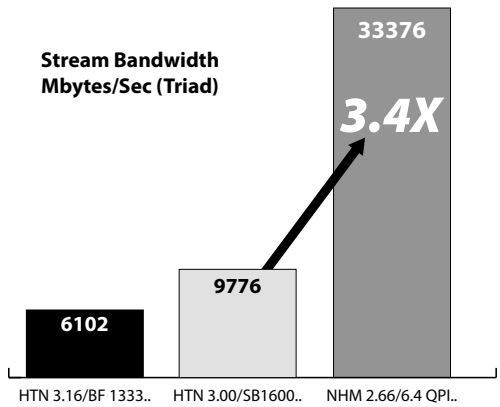


Figure 2-26 RAM bandwidth

Source: Intel Internal measurements—August 2008

HTN: Intel® Xeon® processor 5400 Series (Harpertown)

NHM: Intel® Core™ microarchitecture (Nehalem)

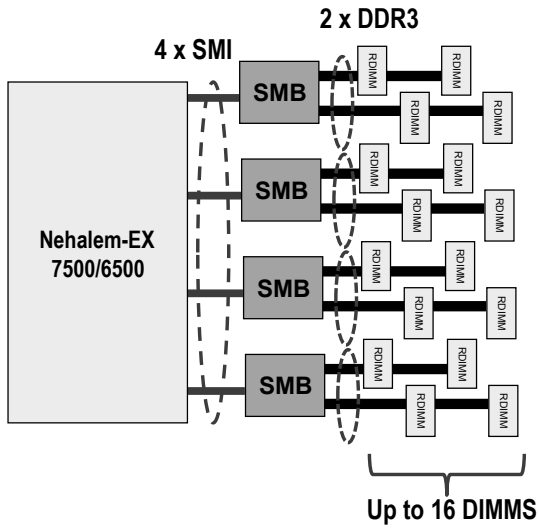


Figure 2-27 SMIs/SMBs

All the communication architectures have evolved over time from buses to point-to-point links that are much faster and more scalable. In Nehalem, Intel® QuickPath Interconnect has replaced the front-side bus (see Figure 2-28).

Intel® QuickPath Interconnect is a coherent point-to-point protocol introduced by Intel®, not limited to any specific processor, to provide communication between processors, I/O devices, and potentially other devices such as accelerators.

The number of QPIs available depends on the type of processor. In Nehalem-EP and in Westmere-EP, each socket has two QPIs allowing the topology shown in Figure 2-25. Nehalem-EX supports four QPIs allowing many other glueless topologies, as shown in Figure 2-29.

The Intel® Xeon® processor 7500 is also compatible with third-party node controllers that are capable of interconnecting more than eight sockets for even greater system scalability.

CPU Architecture

Nehalem increases the instructions per second of each CPU by a number of innovations depicted in Figure 2-30.

Some of these innovations are self-explanatory; we will focus here on the most important one that deals with performance vs. power.

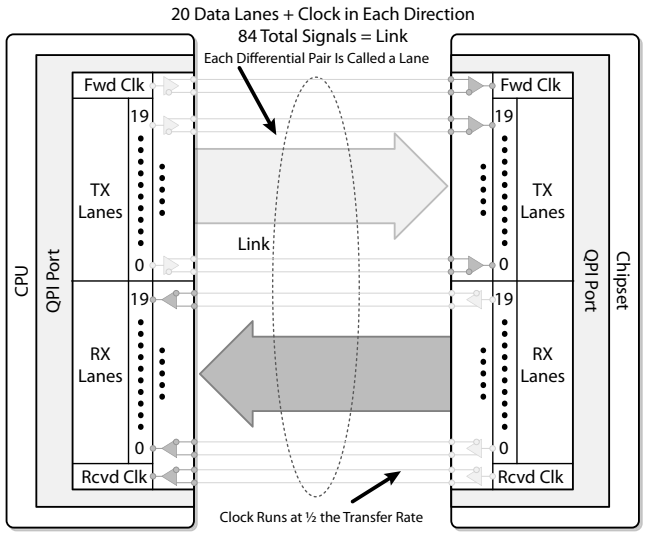


Figure 2-28 Intel® QPI

In comparing performance and power, it is normally assumed that a 1% performance increase with a 3% power increase is break-even. The reason is that it is always possible to reduce the voltage by 1% and reduce the power by 3% (see "Chip Design" in Chapter 2, page 61).

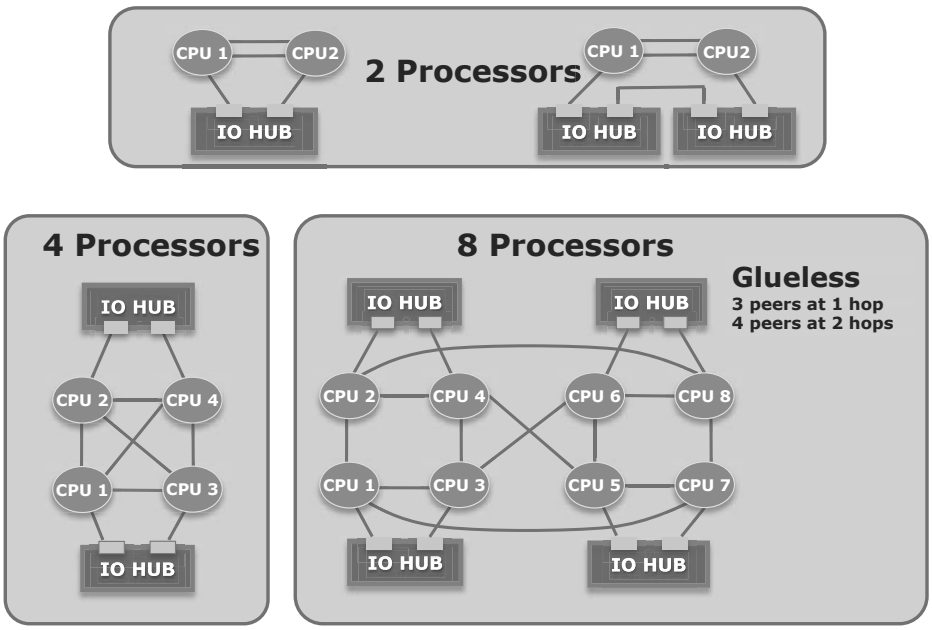


Figure 2-29 Nehalem-EX topologies

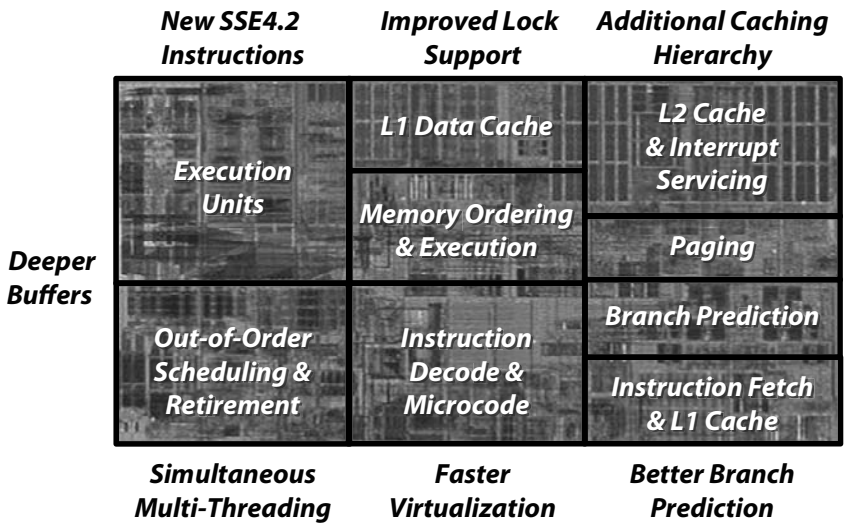


Figure 2-30 Nehalem microarchitecture innovations

The innovations that are important are those that improve the performance by 1% with only a 1% increase in power (better than break-even).

Intel® Hyper-Threading Technology (Intel® HT Technology)

Intel® Hyper-Threading Technology (Intel® HT Technology) is the capability of running simultaneously multiple threads on the same core, in the Nehalem/Westmere implementation two threads. This enhances both performance and energy efficiency (see "Intel® Hyper-Threading Technology" in Chapter 2, page 27).

The basic idea is that with the growing complexity of each execution unit, it is difficult for a single thread to keep the execution unit busy. Instead by overlaying two threads on the same core, it is more likely that all the resources can be kept busy and therefore the overall efficiency increases (see Figure 2-31). Hyper-Threading consumes a very limited amount of area (less than 5%), and it is extremely effective in increasing efficiency, in a heavily threaded environment. Hyper-Threading is not a replacement for cores; it complements cores by allowing each of them to execute two threads simultaneously.

Cache-Hierarchy

The requirement of an ideal memory system is that it should have infinite capacity, infinite bandwidth, and zero latency. Of course, nobody knows how to build such a system. The best approximation is a hierarchy of memory subsystems that go from larger and slower to smaller and faster. In Nehalem, Intel® added one level of hierarchy by increasing the cache layers from two to three (see Figure 2-32).

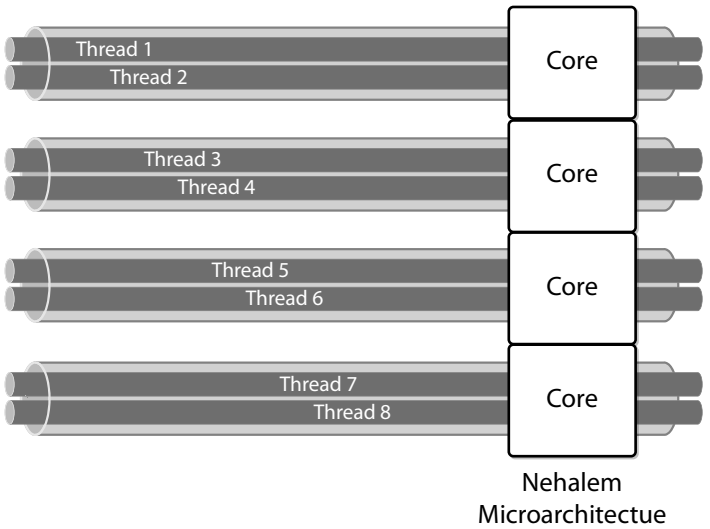


Figure 2-31 Intel® HT technology

Level one caches (L1) (Instruction and Data) are unchanged compared to previous Intel® designs. In the previous Intel® design, the level two caches (L2) were shared across the cores. This was possible since the number of cores was limited to two. Nehalem increments the number of cores to four or eight, and the L2 caches cannot be shared any longer, due to the increase in bandwidth and arbitration requests (potentially 8X). For this reason, in Nehalem Intel® added L2 caches (Instruction and Data) dedicated to each core to reduce the bandwidth toward the shared caches that is now a level three (L3) cache.

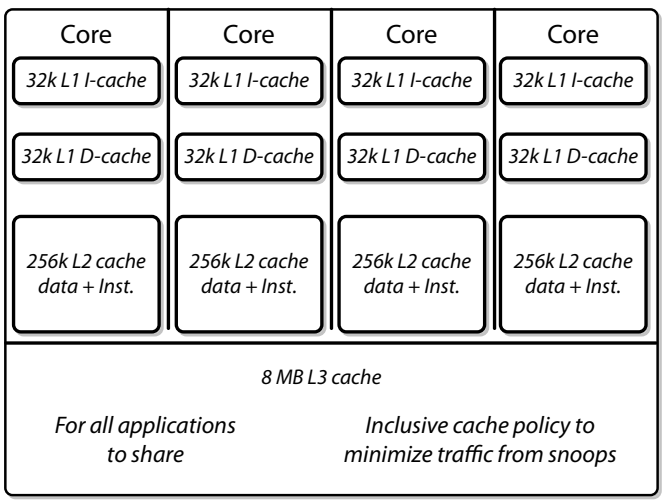


Figure 2-32 Cache hierarchy

Segmentation

Nehalem is designed for modularity. Cores, caches, IMC, and Intel® QPI are examples of modules that compose a Nehalem processor (see Figure 2-30).

These modules are designed independently and they can run at different frequencies and different voltages. The technology that glues all of them together is a novel synchronous communication protocol that provides very low latency. Previous attempts used asynchronous protocols that are less efficient.

Integrated Power Gate

This is a power management technique that is an evolution of the "Clock Gate" that exists in all modern Intel® processors. The clock gate shuts off the clock signals to idle logic, thus eliminating switching power, but leakage current remains. Leakage currents create leakage power consumption that has no useful purpose. With the reduction in channel length, starting approximately at 130 nm, leakage has become a significant part of the power, and at 45 nm, it is very important.

The power gate instead shuts off both switching and leakage power and enables an idle core to go to almost zero power (see Figure 2-33). This is completely transparent to software and applications.

The power gate is difficult to implement from a technology point of view. The classical elements of 45 nm technologies have significant leakage. It required a new transistor technology with a massive copper layer (7 mm) that was not done before (see Figure 2-34).

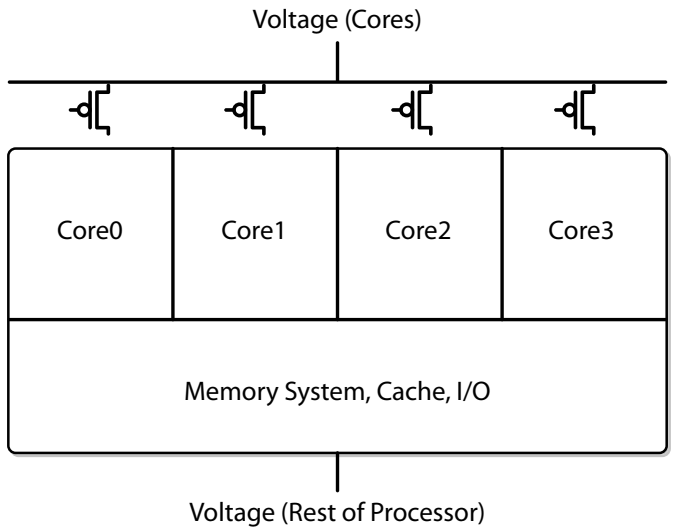


Figure 2-33 Nehalem power gate

The power gate becomes more important as the channel length continues to shrink, since the leakage currents continue to increase. At 22 nm, the power gate is essential.

Nehalem-EP and Westmere-EP have "dynamic" power-gating ability to turn core power off completely when the core is not needed for the given workload. Later, when the workload needs the core's compute capability, the core's power is reactivated.

Nehalem-EX has "static" power gating. The core power is turned off completely when the individual core is disabled in the factory, such as when an 8-core part is fused to make a 6-core part. These deactivated cores cannot be turned back on. On prior generations, such factory deactivated cores continued to consume some power. On Nehalem-EX, the power is completely shut off.

Power Management

Power sensors are key in building a power management system. Previous Intel® CPUs had thermal sensors, but they did not have power sensors. Nehalem has both thermal and power sensors that are monitored by an integrated microcontroller (PCU) in charge of power management (see Figure 2-35).

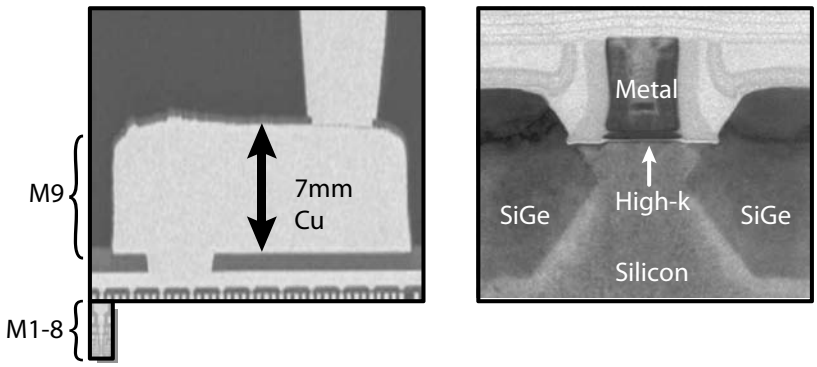


Figure 2-34 Power gate transistor

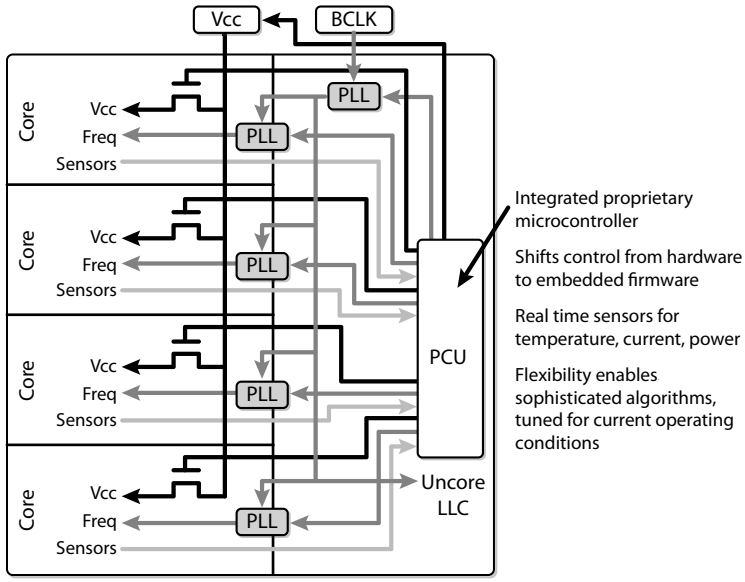


Figure 2-35 Power Control Unit

Intel® Turbo Boost Technology

Power gates and power management are the basic components of Intel® Turbo Boost Technology. Intel® Turbo Boost mode is used when the operating system requires more performance, if environmental conditions permit (sufficient cooling and power)—for example, because one or more cores are turned off. Intel® Turbo Boost increases the frequency of the active cores (and also the power consumption), thus increasing the performance of a given core (see Figure 2-36). This is not a huge improvement (from 3% to 11%), but it may be particularly valuable in lightly or non-threaded environments in which not all the cores may be used in parallel. The frequency is increased in 133MHz steps.

Figure 2-36 shows three different possibilities: In the normal case, all the cores operate at the nominal frequency (2.66 GHz); in the "4C Turbo" mode, all the cores are frequency upgraded by one step (for example, to 2.79 GHz); and in the "<4C Turbo" mode, two cores are frequency upgraded by two steps (for example, to 2.93 GHz).

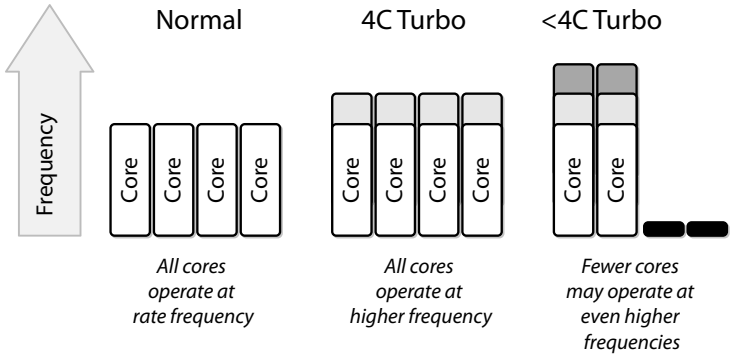


Figure 2-36 Intel® Turbo Boost Technology

Virtualization support

Intel® Virtualization Technology (VT) extends the core platform architecture to better support virtualization software—e.g., VMs (Virtual Machines) and hypervisors aka VMMs (Virtual Machine Monitors); see Figure 2-37.

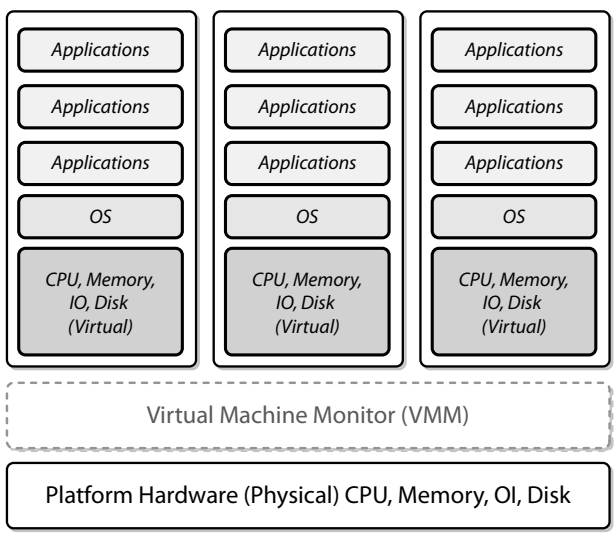


Figure 2-37 Virtualization support

VT has four major components:

- Intel® VT-x refers to all the hardware assists for virtualization in Intel® 64 and IA32 processors.
- Intel® VT-d for Directed I/O (Intel® VT-d) refers to all the hardware assists for virtualization in Intel chipset.
- Intel® VT-c for Connectivity (Intel® VT-c) refers to all the hardware assists for virtualization in Intel networking and I/O devices.
- VT Flex Migration to simplify Virtual Machine movement.

Intel® VT-x enhancements include:

- **A new, higher privilege ring for the hypervisor**—This allows guest operating systems and applications to run in the rings they were designed for, while ensuring the hypervisor has privileged control over platform resources.
- **Hardware-based transitions**—Handoff between the hypervisor and guest operating systems are supported in hardware. This reduces the need for complex, compute-intensive software transitions.
- **Hardware-based memory protection**—Processor state information is retained for the hypervisor and for each guest OS in dedicated address spaces. This helps to accelerate transitions and ensure the integrity of the process.

In addition, Nehalem adds:

- EPT (Extended Page Table)
- VPID (Virtual Processor ID)
- Guest Preemption Timer
- Descriptor Table Exiting
- Intel® Virtualization Technology FlexPriority
- Pause Loop Exiting

VT Flex Migration

FlexMigration allows migration of the VM between processors that have a different instruction set. It does that by synchronizing the minimum level of the instruction set supported by all the processors in a pool.

When a VM is first instantiated, it queries its processor to obtain the instruction set level (SSE2, SSE3, SSE4). The processor returns the agreed minimum instruction set level in the pool, not the one of the processor itself. This allows VMotion between processors with different instruction set.

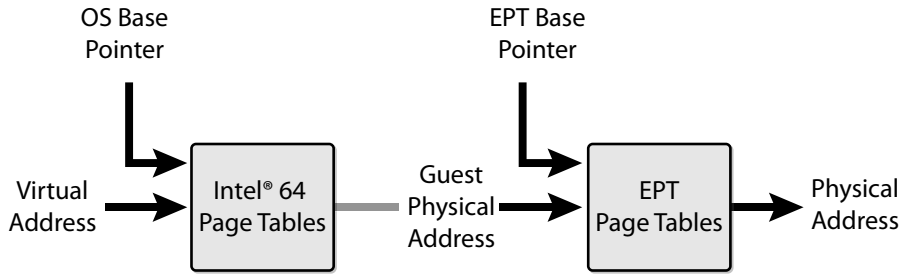


Figure 2-38 Extended page tables

Extended Page Tables (EPT)

EPT is a new page-table structure, under the control of the hypervisor (see Figure 2-38). It defines mapping between guest- and host-physical addresses.

Before virtualization, each OS was in charge of programming page tables to translate between virtual application addresses and the "physical addresses". With the advent of virtualization, these addresses are no longer physical, but instead local to the VM. The hypervisor needs to translate between the guest OS addresses and the real physical addresses. Before EPT, the hypervisors maintained the page table in software by updating them at significant boundaries (e.g., on VM entry and exit).

With EPT, there is an EPT base pointer and an EPT Page Table that allow it to go directly from the virtual address to the physical address without the hypervisor intervention, in a way similar to how an OS does it in a native environment.

Virtual Processor ID (VPID)

This is the ability to assign a VM ID to tag CPU hardware structures (e.g., TLBs: Translation Lookaside Buffers) to avoid flushes on VM transitions.

Before VPID, in a virtualized environment, the CPU flushes the TLB unconditionally for each VM transition (e.g., VM Entry/Exit). This is not efficient and adversely affects the CPU performance. With VPID, TLBs are tagged with an ID decided by the hypervisor that allows a more efficient flushing of cached information (only flush what is needed).

Guest Preemption Timer

With this feature, a hypervisor can preempt guest execution after a specified amount of time. The hypervisor sets a timer value before entering a guest and when the timer reaches zero, a VM exit occurs. The timer causes a VM exit directly with no interrupt, so this feature can be used with no impact on how the VMM virtualizes interrupts.

Descriptor Table Exiting

Allows a VMM to protect a guest OS from internal attack by preventing relocation of key system data structures.

OS operation is controlled by a set of key data structures used by the CPU: IDT, GDT, LDT, and TSS. Without this feature, there is no way for the hypervisor to prevent malicious software running inside a guest OS from modifying the guest's copies of these data structures. A hypervisor using this feature can intercept attempts to relocate these data structures and forbid malicious ones.

FlexPriority

This is a technique to improve performance on older 32-bit guest OS's. It was designed to accelerate virtualization interrupt handling thereby improving virtualization performance. FlexPriority accelerates interrupt handling by preventing unnecessary VMExits on accesses to the Advanced Programmable Interrupt Controller.

Pause Loop Exiting

This technique detects spin locks in multi-process guests to reduce "lock-holder preemption". Without this technique, a given virtual processor (vCPU) may be preempted while holding a lock. Other vCPUs that try to acquire lock will spin for entire execution quantum.

This technique is present in Nehalem-EX, but not in Nehalem-EP.

Advanced Reliability

A lot of the innovation in Nehalem-EX compared to Nehalem-EP is in the advanced reliability area or more properly RAS (Reliability, Availability, and Serviceability); see Figure 2-39.

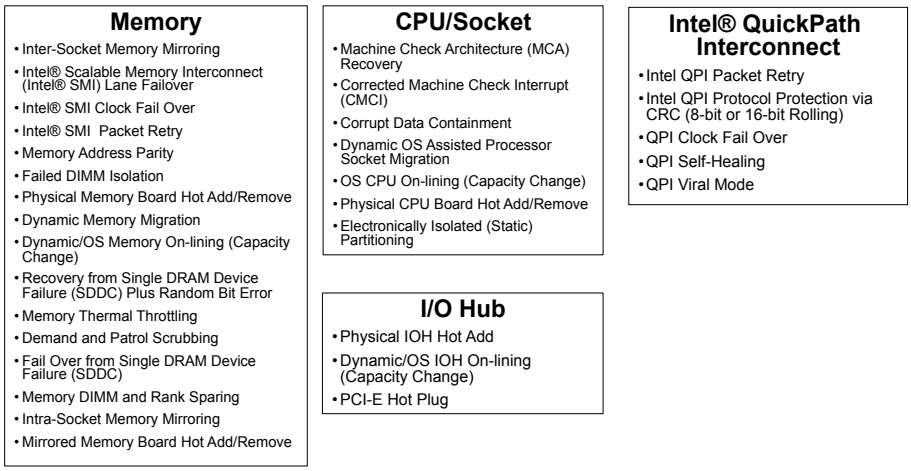


Figure 2-39 Nehalem-EX RAS

In particular, all the major processor functions are covered by RAS, including: QPI RAS, I/O Hub (IOH) RAS, Memory RAS, and Socket RAS.

Corrected Errors are now signaled using the Corrected Machine Check Interrupts (CMCI).

An additional RAS technique is Machine Check Architecture-recovery (MCAr); i.e., a mechanism in which the CPU reports hardware errors to the operating system. With MCAr, it is possible to recover from otherwise fatal system errors.

Some features require additional operating system support and/or requires hardware vendor implementation and validation.

This technology is implemented only in Nehalem-EX.

Advanced Encryption Standard

Westmere-EP adds six new instructions for accelerating encryption and decryption of popular AES (Advanced Encryption Standard) algorithms. With these instructions, all AES computations are done by hardware and they are of course not only faster, but also more secure than a software implementation.

This enables applications to use stronger keys with less overhead. Applications can encrypt more data to meet regulatory requirements, in addition to general security, with less impact to performance.

This technology is implemented only in Westmere-EP.

Trusted Execution Technology

Intel® Trusted Execution Technology (TXT) helps detect and/or prevent software-based attacks, in particular:

- Attempts to insert non-trusted VMM (rootkit hypervisor)
- Attacks designed to compromise platform secrets in memory
- BIOS and firmware update attacks

Intel® TXT uses a mix of processor, chipset, and TPM (Trusted Platform Module) technologies to measure the boot environment to detect software attacks (see Figure 2-40).

This technology is implemented only in Westmere-EP.

Chip Design

When trying to achieve high performance and limit the power consumption, several different factors need to be balanced.

With the progressive reduction of the length of the transistor channel, the range of voltages usable becomes limited (see Figure 2-41).

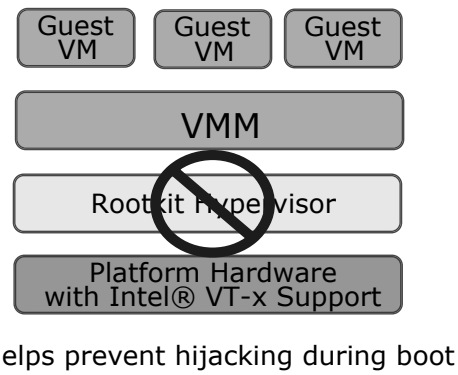


Figure 2-40 Intel® trusted execution technology

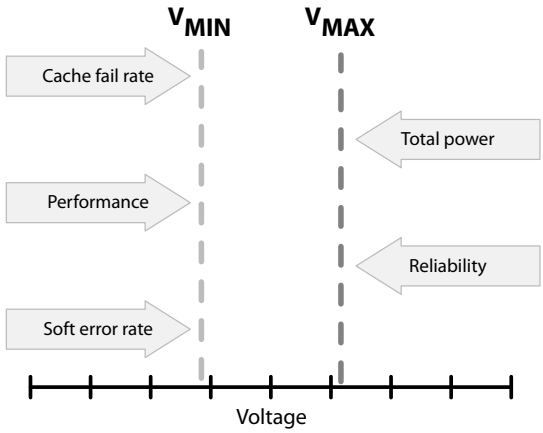


Figure 2-41 Voltage range

The maximum voltage is limited by the total power consumption and the reliability decrease associated with high power, the minimum voltage is limited mostly by soft errors especially in memory circuits.

In general, in CMOS design the performance is proportional to the voltage, since higher voltages allow higher frequency.

$$Performance \sim Frequency \sim Voltage$$

Power consumption is proportional to the frequency and the square of voltage:

$$Power \sim Frequency \times Voltage^2$$

and, since frequency and voltage are proportional:

$$Power \sim Voltage^3$$

Energy efficiency is defined as the ratio between performance and power, and therefore:

$$Energy\ Efficiency \sim 1/Voltage^2$$

Therefore, from an energy-efficiency perspective, there is an advantage in reducing the Voltage (i.e., the power; see Figure 2-42) so big that Intel® has decided to address it.

Since the circuits that are more subject to soft error are memories, Intel® in Nehalem deploys a sophisticated error-correcting code (triple detect, double correct) to compensate for these soft errors. In addition, the voltage of the caches and the voltage of the cores are decoupled so the cache can stay at high voltage while the cores works at low voltage.

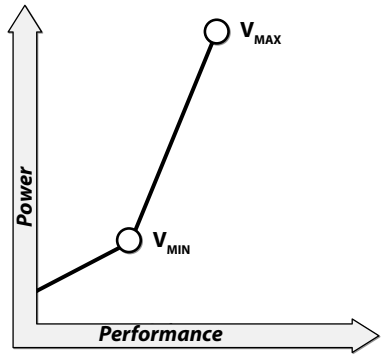


Figure 2-42 Power vs. performance

For the L1 and L2 caches, Intel® has replaced the traditional six transistors SRAM design (6-T SRAM) with a new eight transistors design (8-T SRAM) that decouples the read and write operations and allows lower voltages (see Figure 2-43).

Also, to reduce power, Intel® went back to static CMOS, which is the CMOS technology that consumes less power (see Figure 2-44).

Performance was regained by redesigning some of the key algorithm like instruction decoding.

Chipset Virtualization Support

In addition to the virtualization support provided inside Nehalem, other improvements have been implemented at the chipset/motherboard level to better support

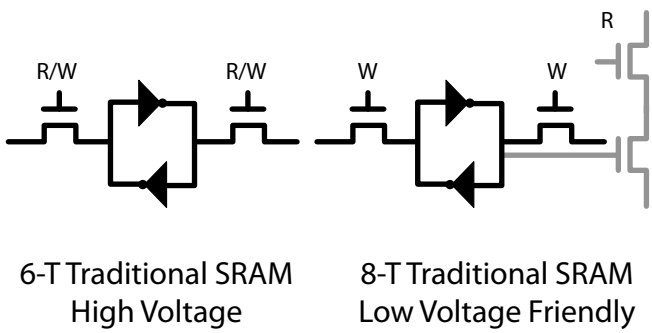


Figure 2-43 Six- vs. eight-transistor SRAM

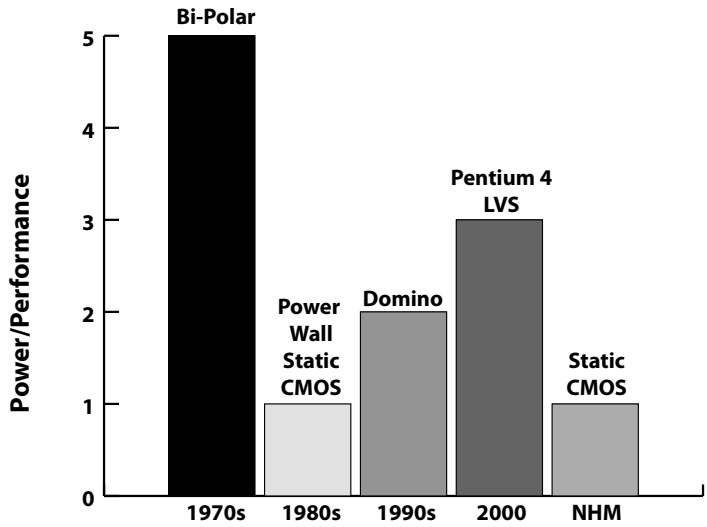


Figure 2-44 Power consumption of different technologies

virtualization. These improvements are important to increase the I/O performance in the presence of a hypervisor (in Intel parlance, the hypervisor is referred to as VMM: Virtual Machine Monitor).

Intel® VT-d for Direct I/O

Servers use an Input/Output Memory Management Unit (IOMMU) to connect a DMA-capable I/O bus (e.g., PCIe) to the main memory. Like a traditional MMU (Memory Management Unit), which translates CPU-visible virtual addresses to physical addresses, the IOMMU takes care of mapping device-visible virtual addresses to physical addresses. These units also provide memory protection from misbehaving devices.

A general requirement for I/O virtualization is the ability to isolate and restrict device accesses to the resources owned by the partition managing the device.

In 2008, Intel published a specification for IOMMU technology: Virtualization Technology for Directed I/O, abbreviated VT-d.

Intel® VT for Directed I/O provides VMM software with the following capabilities:

- **I/O device assignment:** For flexibly assigning I/O devices to VMs and extending the protection and isolation properties of VMs for I/O operations.

- **DMA remapping**: For supporting independent address translations for Direct Memory Accesses (DMA) from devices.
- **Interrupt remapping**: For supporting isolation and routing of interrupts from devices and external interrupt controllers to appropriate VMs.
- **Reliability**: For recording and reporting to system software DMA and interrupt errors that may otherwise corrupt memory or impact VM isolation.

Intel® VT-c for Connectivity

Intel® Virtualization Technology for Connectivity (Intel® VT-c) is a collection of I/O virtualization technologies that enables lower CPU utilization, reduced system latency, and improved networking and I/O throughput.

Intel® VT-c consists of platform-level technologies and initiatives that work together to deliver next-generation virtualized I/O:

- Virtual Machine Device Queues (VMDq) dramatically improves traffic management within the server, helping to enable better I/O performance from large data flows while decreasing the processing burden on the software-based Virtual Machine Monitor (VMM).
- Virtual Machine Direct Connect (VMDc) provides near native-performance by providing dedicated I/O to virtual machines, bypassing the software virtual switch in the hypervisor completely. It also improves data isolation among virtual machines, and provides flexibility and mobility by facilitating live virtual machine migration.

VMDq

In virtual environments, the hypervisor manages network I/O activities for all the VMs (Virtual Machines). With the constant increase in the number of VMs, the I/O load increases and the hypervisor requires more CPU cycles to sort data packets in network interface queues and route them to the correct VM, reducing CPU capacity available for applications.

Intel® Virtual Machine Device Queues (VMDq) reduces the burden on the hypervisor while improving network I/O by adding hardware support in the chipset. In particular, multiple network interface queues and sorting intelligence are added to the silicon, as shown in Figure 2-45.

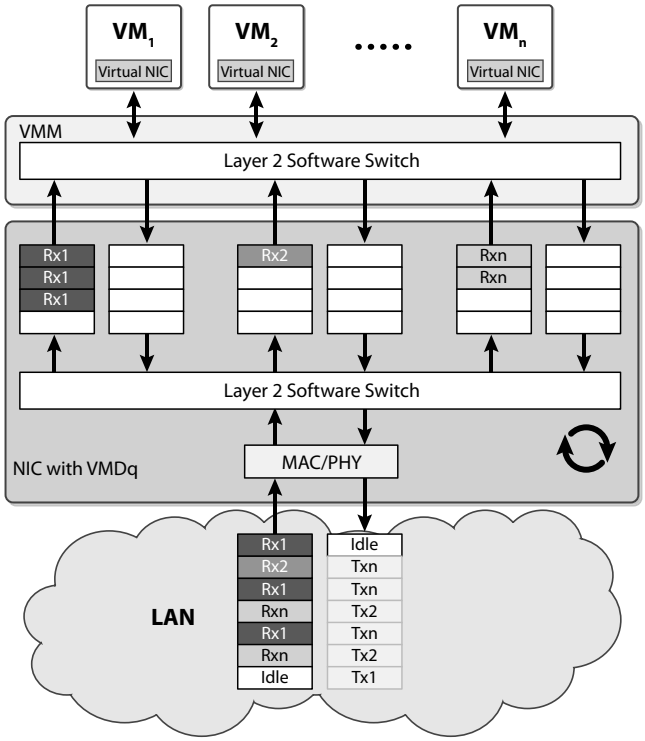


Figure 2-45 VMDq

As data packets arrive at the network adapter, a Layer 2 classifier/sorter in the network controller sorts and determines which VM each packet is destined for based on MAC addresses and VLAN tags. It then places the packet in a receive queue assigned to that VM. The hypervisor's layer 2 software switches merely routes the packets to the respective VM instead of performing the heavy lifting work of sorting data.

As packets are transmitted from the virtual machines toward the adapters, the hypervisor layer places the transmit data packets in their respective queues. To prevent head-of-line blocking and ensure each queue is fairly serviced, the network controller transmits queued packets to the wire in a round-robin fashion, thereby guaranteeing some measure of Quality of Service (QoS) to the VMs.

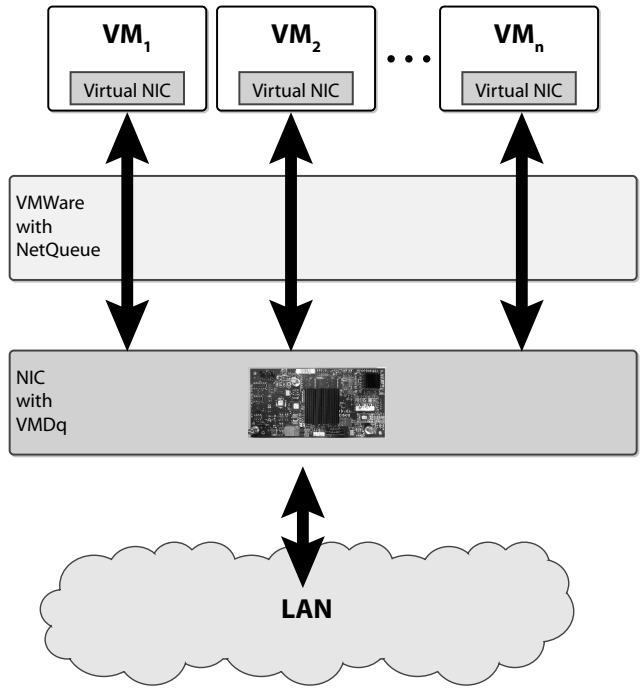


Figure 2-46 VMM NetQueue

NetQueue®

To take full advantage of VMDq, the VMMs needs to be modified to support one queue per Virtual Machine. For example, VMware® has introduced in its hypervisor a feature called NetQueue that takes advantage of the frame, sorting capability of VMDq. The combination of NetQueue and VMDq offloads the work that ESX has to do to route packets to virtual machines; therefore, it frees up CPU and reduces latency (see Figure 2-46).

VMQ®

VMQ is Microsoft's Hyper-V® queuing technology that makes use of the VMDq capabilities of the Intel Ethernet controller to enable data packets to be delivered to the VM with minimal handling in software.

VMDc®

Virtual Machine Direct Connect (VMDc) enables direct networking I/O assignment to individual virtual machines (VMs). This capability improves overall networking performance and data isolation among VMs and enables live VM migration.

VMDc complies with the Single Root I/O Virtualization (SR-IOV) standard; see "SR-IOV" in Chapter 3, page 83.

The latest Intel® Ethernet server controllers support SR-IOV to virtualize the physical I/O port into multiple virtual I/O ports called Virtual Functions (VFs).

Dividing physical devices into multiple VFs allows physical I/O devices to deliver near-native I/O performance for VMs. This capability can increase the number of VMs supported per physical host, driving up server consolidation.

VMDirectPath®

When VMDq and VMDc are combined with VMware® VMDirectPath, the ESX/vSphere hypervisor is bypassed in a way similar to a "kernel bypass", the software switch inside the hypervisor is not used, and data is directly communicated from the adapter to the vNICs (virtual NICs) and vice versa. See "VN-Link" in Chapter 3, page 90, and in particular Figure 3-20.

Index

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X