Trends in Computer Architecture and their Impact on Computing

Anupama Potluri

School of Computer and Information Sciences
University of Hyderabad

Outline

Outline

Multiprocessor Architectures to Multi-core Architectures

Factors driving the development of multi-core architectures: [1]

- High power consumption and limitations in cooling beyond a certain clock speed.
- Many applications are better suited to **thread level parallelism** than **instruction level parallelism**.
- Increased available space on the chip due to miniaturization.
- To keep up with Moore's law of doubling CPU capability every 18 months, multi-core was the only way forward.

Advantages and Disadvantages of Multi-core over Multi-processor Architectues

Advantages

- Improves **cache coherency**.
- Less PCB space, decreased power.
- Higher performance for power consumed.

Disadvantages

- Maximizing usage requires changes to OS and applications.
- Sharing the same bus and memory bandwidth limits the performance.

Outline

Graphical Processing Units I

A **GPU** is a processor that is built specially for graphics operations such as translation, rotation and other operations in different coordinate systems and rendering polygons. However, their highly parallel nature has made them highly effective for problems with high **data parallelism**. Research has also shown that the GPUs outperform the general purpose CPUs from 2 to 5 times for data crunching [9, 10].

Advantages of GPUs over CPUs:

- Very fast.
- Cheap.
- Use a lot less power than CPUs.

Limitations of GPUs over CPUs:

- Good for parallel data crunching but ineffective for branching and other complex logic that drives typical applications.

Graphical Processing Units II

Today, Nvidia and ATI, the two companies that are the leaders in GPU technology provide programming interfaces for development of non-graphics applications using GPUs. Nvidia's Compute Unified Device Architecture (CUDA) and the open standard OpenCL are used for development of applications using GPUs.

OpenCL standard

OpenCL is defined to allow development using both GPUs and CPUs and is supported by Intel, AMD, Nvidia and ARM.

Outline

Peripheral Component Interconnect Express (PCIe)

PCI Architecture

PCI uses a shared parallel bus architecture where the clocking is limited to the lowest clocking device on the bus and there is contention for the shared bus requiring arbitration [4].

PCIe Architecture

PCIe, on the other hand, uses point-to-point topology that supports full-duplex communication between any pair of devices. There is also no inherent limitation on concurrent accesses [4].

Advantages of PCIe [4]:

- Higher maximum bus throughput.
- Smaller physical footprint.
- Better performance-scaling for bus devices.
- Hot-plug functionality.

Outline

Challenges of Software Development in Multi-core Architectures

Some of the challenges of software development in multi-core architectures [3]:

- Data races – timing dependent mutual exclusion errors.
- Deadlocks.
- Understanding the inherent parallelism in the algorithm and developing multi-threaded applications, which is non-trivial.

Transactional Memory Support I

Definition

Transactional memory is a technique to simplify the writing of parallel programs that require mutual exclusion based on the concept of transaction in databases. Introduced for the first time by **Maurice Herlihy and Eliot Moss** in 1993.

Typically, the code that is written for mutual exclusion locks the code using mutexes or semaphores as the code may be written to by another thread simultaneously. However, if no other thread accesses the data being manipulated, the locking is keeping other threads locked out and slows down the execution.

Transactional Memory Support II

Intel’s Haswell Processor

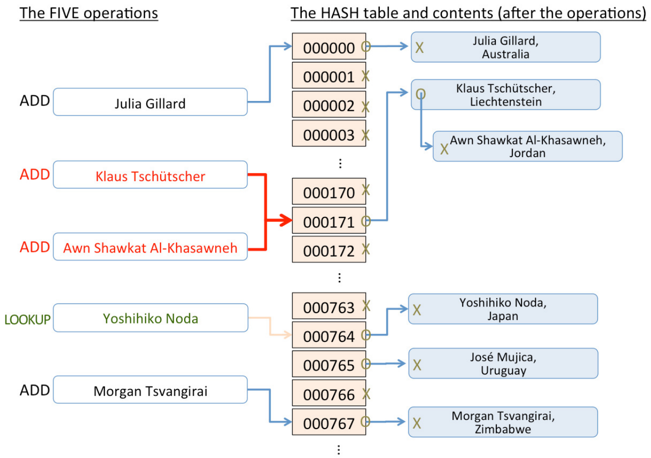
Transactional memory Extensions (TSX) in Intel’s Haswell processor that is coming up allow programmers to specify these pieces of code as transactions [6].

Intel’s TSX comes in two flavors:

- **Hardware Lock Elision (HLE):** Hardware removes locks around software and optimistically schedules the threads until a conflict occurs. At this stage, the threads that are affected are rolled back and re-scheduled sequentially.
- **Restricted Transactional Memory (RTM):** This is a new instruction set interface that provides a way for programmers to provide **transactional processing** with more flexibility than that provided by HLE.

Transactional Memory Support III

The TSX HLE can be explained with the following example [1]:



[1] Coarse-grained locks and TSX explained [7]

Parallelizing Compilers

Parallelizing Compilers that intelligently and automatically take care of parallelizing the sequential code written by the regular programmers is a definite requirement.

The communication latency between different hardware threads to synchronize is the limitation that stymied the growth of parallelizing compilers.

A recent approach is to use history of execution of a program to re-compile and re-optimize a program to take advantage of the cores. An example of this is **Helix** [5].

Outline

Data Centers

Increasing Network Bandwidth and widespread use of the Internet lead to a lot of applications such as the web-based email and content hosting on the web spurring the growth of search engines.

Further growth has been seen in social networking and Web 2.0 etc. applications that require storing and mining huge amounts of data. All of this data is stored in large server farms, the size of huge football fields or even large cities.

Power and cooling requirements reduction has become an important part of computing in an attempt to achieve **Green Computing** – i.e., minimization of resource consumption.

Outline

Virtualization and its benefits

Virtualization Definition

Virtualization is the technique of separation of the physical reality from the *user perceived reality*.

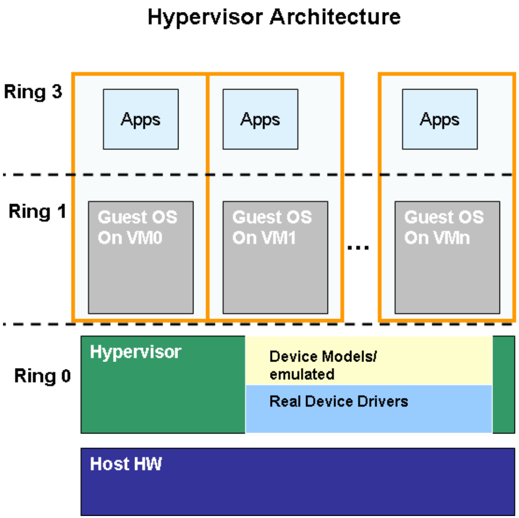
Virtualization is a reality today in all areas of computing including **CPU**, **I/O devices**, **Network** and **Storage**.

Advantages

- Better utilization of the huge computing power of the multi-core CPUs.
- Better utilization of the network bandwidth.
- Capability to migrate from one physical server to another without disruption to the user, i.e., **High Availability (HA).**
- Redundancy leading to **high fault-tolerance.**

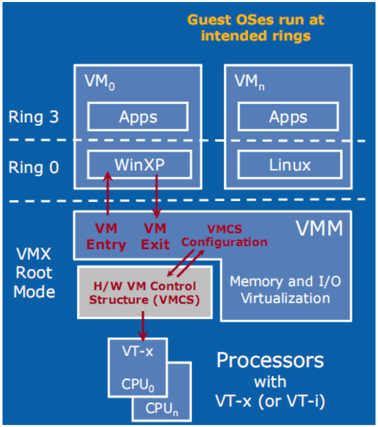
OS Virtualization

Operating System Virtualization is illustrated in the figure below [11]:



Intel's VT-X Technology I

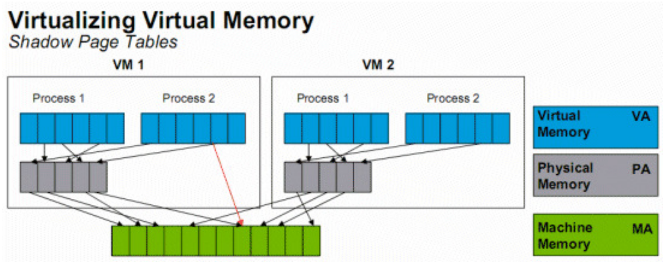
The first extension in Intel's VT-X technology allows Guest OSes to run at the Ring Level 0 and adds a new higher privilege ring called **Root Ring** for the Hypervisor to run in as shown in the figure below [11]:



Guest OSes run at intended rings
VM$_0$
VM$_n$
Ring 3
Apps
Apps
Ring 0
WinXP
Linux
VM Entry
VM Exit
VMCS Configuration
VMM
VMX Root Mode
H/W VM Control Structure (VMCS)
Memory and I/O Virtualization
VT-x
CPU$_0$
CPU$_n$
Processors with VT-x (or VT-i)

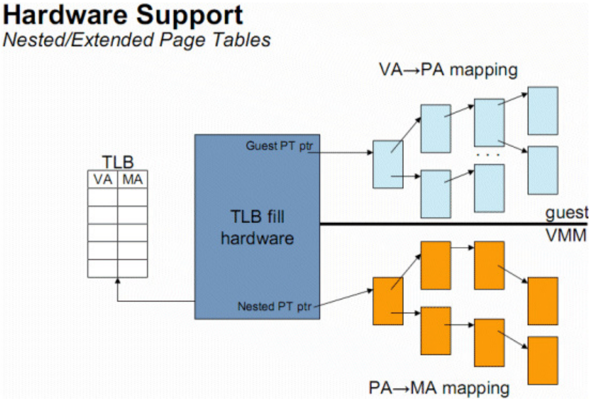
Intel’s VT-X Technology II

Memory Management is another challenge in virtualization where, there is a virtual memory per VM which is translated into physical memory per VM. This has to be translated into actual physical memory. This is shown in the figure below [11].



Intel's VT-X Technology III

The second extension is the **Extended Page Tables**. In this the page table translation entries for both the virtual memory to physical memory of a VM and the translation to actual physical memory are stored in a TLB. This is illustrated in the figure below [11]:



Server Virtualization

Data Centers are made up of server farms most of which are heavily underutilized if each of the powerful physical servers is used to host a different client system.

Server Virtualization is simply the technology of running multiple VMs on a single system – sometimes as many as 40 VMs on a single physical server – and keeping them separate at the same time to satisfy the security requirements of the clients.

The advantages are

- Migration from one physical server to another.
- Dynamic provisioning of VMs based on demand.
- Licensing costs are reduced as software like SQL and Oracle licenses are based on the number of cores used.

I/O Virtualization I

Storage Area Networks (SANs)

SANs are used to have centralized storage consisting of disk arrays and other storage devices attached on the network through Fibre Channel cables so that these devices appear as though they are **local devices** to the CPU [12].

I/O Virtualization

I/O Virtualization extends the virtualization to the I/O handled by the data center servers [8].

Motivation: Each of the data center servers has

- Network Interface Card (NIC)
- Fibre Channel Adapter (FCA) for accessing storage
- Cabling to the switches

I/O Virtualization II

Benefits

I/O Virtualization helps in **reducing costs** by reducing the requirement of these per server.

I/O Virtualization allows the top-of-the-rack switch to consist of few high capacity adapters/cards that are connected via a PCIe bus to all the servers in that rack. Each of these high capacity cards are virtualized as small capacity individual server virtual devices. The I/O virtualization box takes care of handling the actual traffic on the network to the data Internet, on the SAN network to access storage and so on.

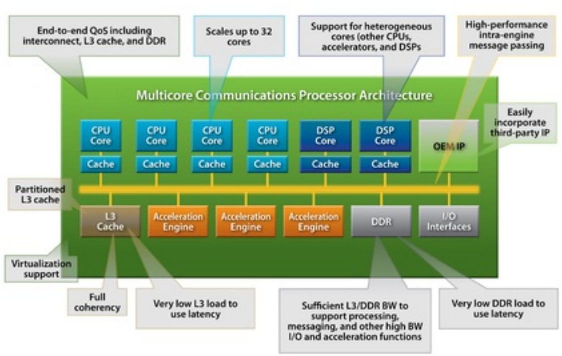
Outline

Performance gains beyond multicore I

To gain true performance from the multi-core architectures, the following are the challenges and trends in technology [2]:

- A trade-off between compilers that collect and analyze historical data to determine code that can be parallelized versus the time and resources spent on this.
- Instead of homogeneous multi-cores, some of the cores may be highly specialized and a combination of general purpose multi-cores with these leading to a System-on-Chip (SOC) architectures[]. These need to be leveraged without a burden on the regular application developers. The compiler technology has to evolve to take these into consideration too.

Performance gains beyond multicore II


End-to-end QoS including interconnect, L3 cache, and DDR
Scales up to 32 cores
Support for heterogeneous cores (other CPUs, accelerators, and DSPs
High-performance intra-engine message passing
Multicore Communications Processor Architecture
CPU Core
Cache
CPU Core
Cache
CPU Core
Cache
CPU Core
Cache
DSP Core
Cache
DSP Core
Cache
OEM IP
Easily incorporate third-party IP
Partitioned L3 cache
L3 Cache
Acceleration Engine
Acceleration Engine
Acceleration Engine
DDR
I/O Interfaces
Virtualization support
Full coherency
Very low L3 load to use latency
Sufficient L3/DDR BW to support processing, messaging, and other high BW I/O and acceleration functions
Very low DDR load to use latency

Outline

Summary I

In summary, we can say that

- Due to an **increase in the network bandwidth** and the consequent **increase in content on the web**, there was a great need for data centers that collected and/or hosted all the information on the web.
- The doubling of computing capacity every 18 months as per Moore’s law hit some physical constraints and branched off into **multi-core processors**.
- **Underutilization and lack of High Availability** due to mapping server software to a physical server lead to the growth of ***Virtualization technology***.
- Virtualization, in its turn, spurred the growth of hardware assists such as **Extended Page Tables, Transactional Memory** etc..

Summary II

- Gaming industry requirements drove research and development of **specialized GPUs** which have found tremendous use in other applications such as weather modeling, video decoding etc. The development of **CUDA and OpenCL** has lead to their use in regular applications.
- Making proper use of the multi-core architectures in regular desktops without burdening all the application developers requires great improvements in compiler technologies. **Dynamic recompilation and reoptimization** such as in Helix and handling heterogeneous System-on-Chips are the challenges in this area.

Summary III

This is perhaps one of the most exciting times for System Developers including Computer Architecture designers, OS designers, Compiler developers and Network developers. The driving force behind these changes is the **Big Data** that is radically also changing the face of Databases as we know them.

References

[1] http://en.wikipedia.org/wiki/Multi-core_processor

[2] http://www.datacenteracceleration.com/author.asp?section_id=2412&doc_id=255726

[3] http://news.cnet.com/8301-13556_3-10390800-61.html

[4] http://en.wikipedia.org/wiki/PCI_Express

[5] Simone Campanoni, Timothy M. Jones, Glenn Holloway, Gu-Yeon Wei, David Brooks: *HELIX: MAKING THE EXTRACTION OF THREAD-LEVEL PARALLELISM MAINSTREAM*, IEEE Micro, July/August 2012.

[6] http://www.realworldtech.com/haswell-tm/

[7] http://software.intel.com/en-us/blogs/2012/02/07/coarse-grained-locks-and-transactional-synchronization-explained

[8] http://searchitchannel.techtarget.com/feature/I-O-virtualization-IOV-Delivering-cost-and-power-savings-to-data-center-servers

[9] http://www.wired.com/gadgets/pcs/news/2006/11/72090

[10] http://en.wikipedia.org/wiki/Graphics_processing_unit

[11] http://www.anandtech.com/show/2480

[12] http://en.wikipedia.org/wiki/Storage_area_networks

[13] http://embedded-computing.com/articles/next-generation-architectures-tomorrows-communications-networks/