C-DAC Four Days Technology Workshop

ON

Hybrid Computing – Coprocessors/Accelerators
Power-Aware Computing – Performance of Applications Kernels

hyPACK-2013
(Mode-1:Multi-Core)

Lecture Topic:

Multi-Core Processors : Introduction

Venue : CMSD, UoHYD ; Date : October 15-18, 2013

Multi Core Arch System Overview : Agenda

Quick overview of what this Lecture is all about

- Introduction
- Multi Cores : Development Motivation
- Multi Cores : Software and Hardware Trends
- Multi Cores : Issues and Challenges
- Multi Cores : Programming Paradigms

Source : Reference : [4], [6], [14],[17],]22], [28]

Multi Core Arch System Overview : Questions

Questions to be Addressed

- Is it difficult to port my applications on Multi Cores?
- Do I need to change my Style of Prog for Multi core ?
- Is there any Multi Core Programming Paradigms ?
- How do Multi Core Compiler can help me ?

Multi Core Arch System Overview : Questions

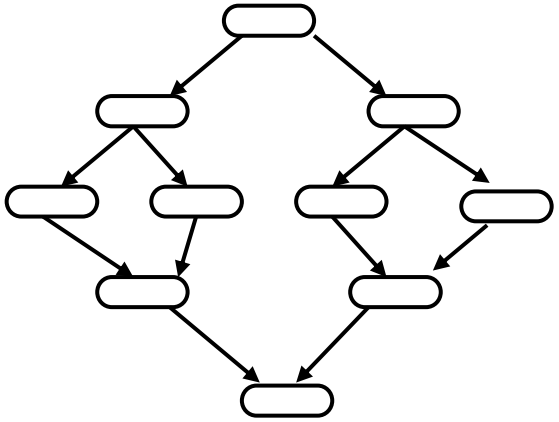
Questions to be Addressed

- Building of *hand-coded* Multi Core applications using the current low-level programming tools (e.g.,C, C++, Fortran, Java, sockets, Threads, OpenMP ,PVM, MPI etc…)

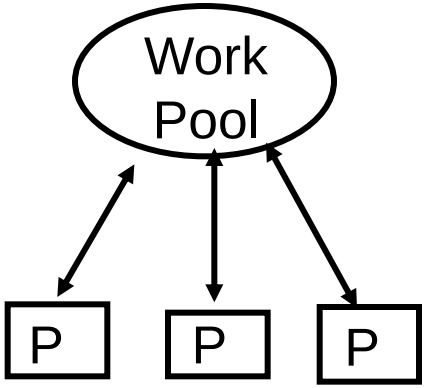
  - Not Very Hard (Not Easy) ?
  - Error prone – A **real nightmare** for programmers
  - The way of programming Multi Core is heroic.

Distributed Computing : Algorithmic Paradigms

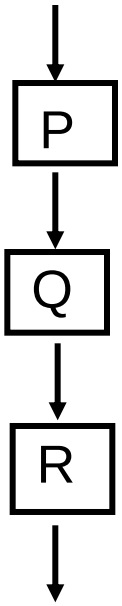
Divide and conquer



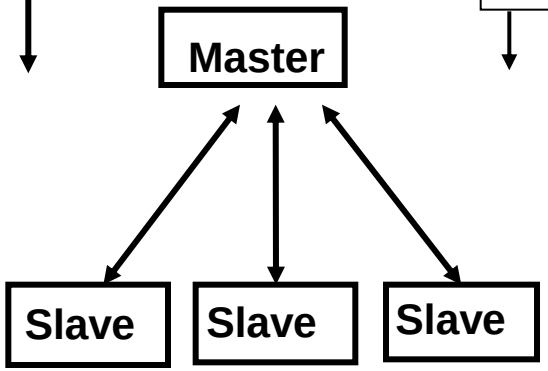
Work pool



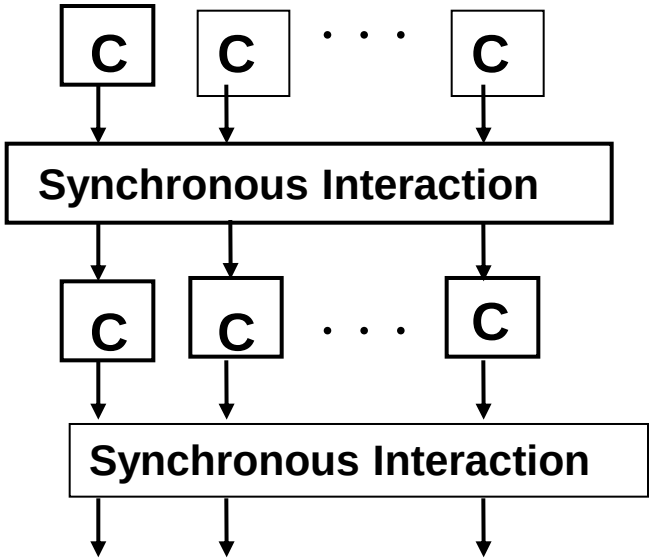
Data stream



Process farm



Phase Parallel Model



Parallel Programming Models

Implicit parallel programming models

- Automatic Parallelization of sequential programs using compiler technology.

Explicit parallel programming models

Three dominant parallel programming models are :

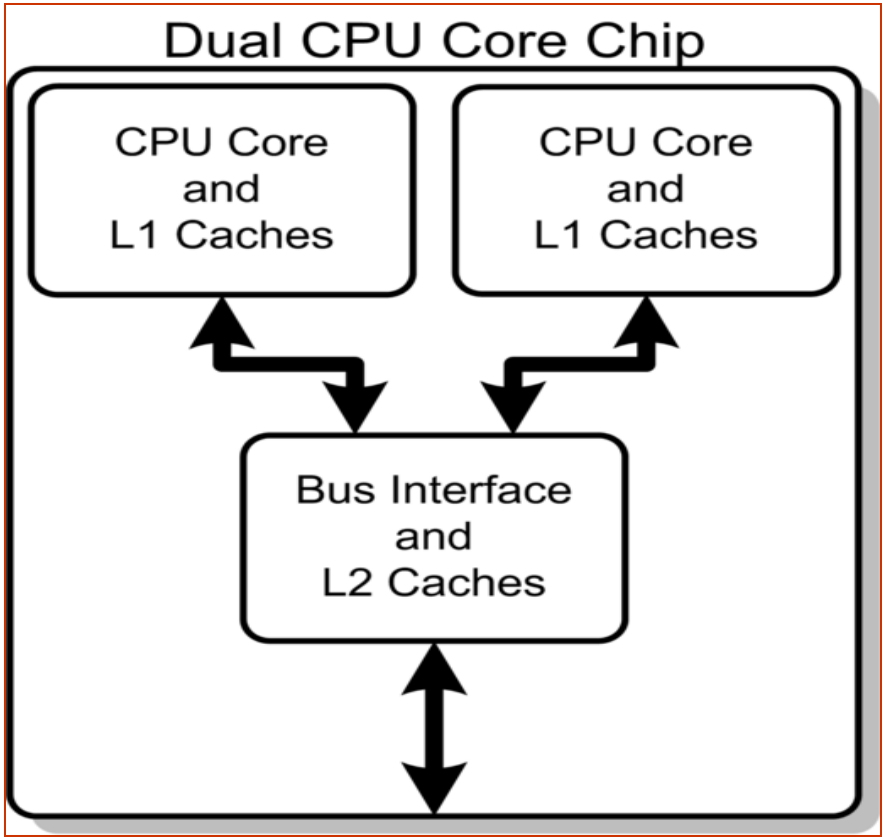
- Data-parallel model (f90/HPF)
- Message-passing model (MPI/PVM)
- Shared-variable model (OpenMP/Pthreads)

Note : All the parallel programming models share a common computational characteristics

Dual Core Processor

Conceptual diagram of a dual-core CPU, with

- CPU-local Level 1 caches, and
- Shared, on-chip Level 2 caches

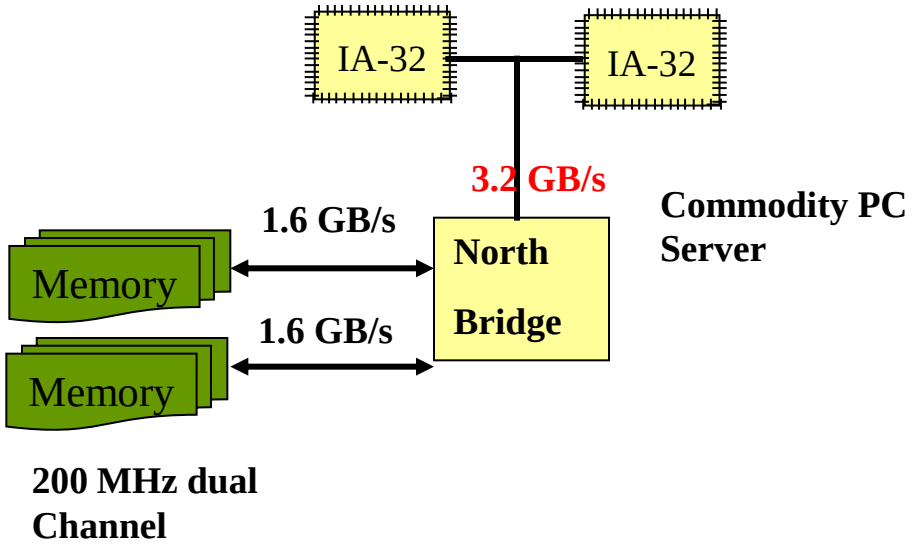


Commodity PC -Server

Memory Architecture for Dual processor system

- Shared Bus Micro Architecture
  Intel IA-32 processors – Incorporates two processors on a single motherboard that shares a common Northbridge and Memory DIMMS

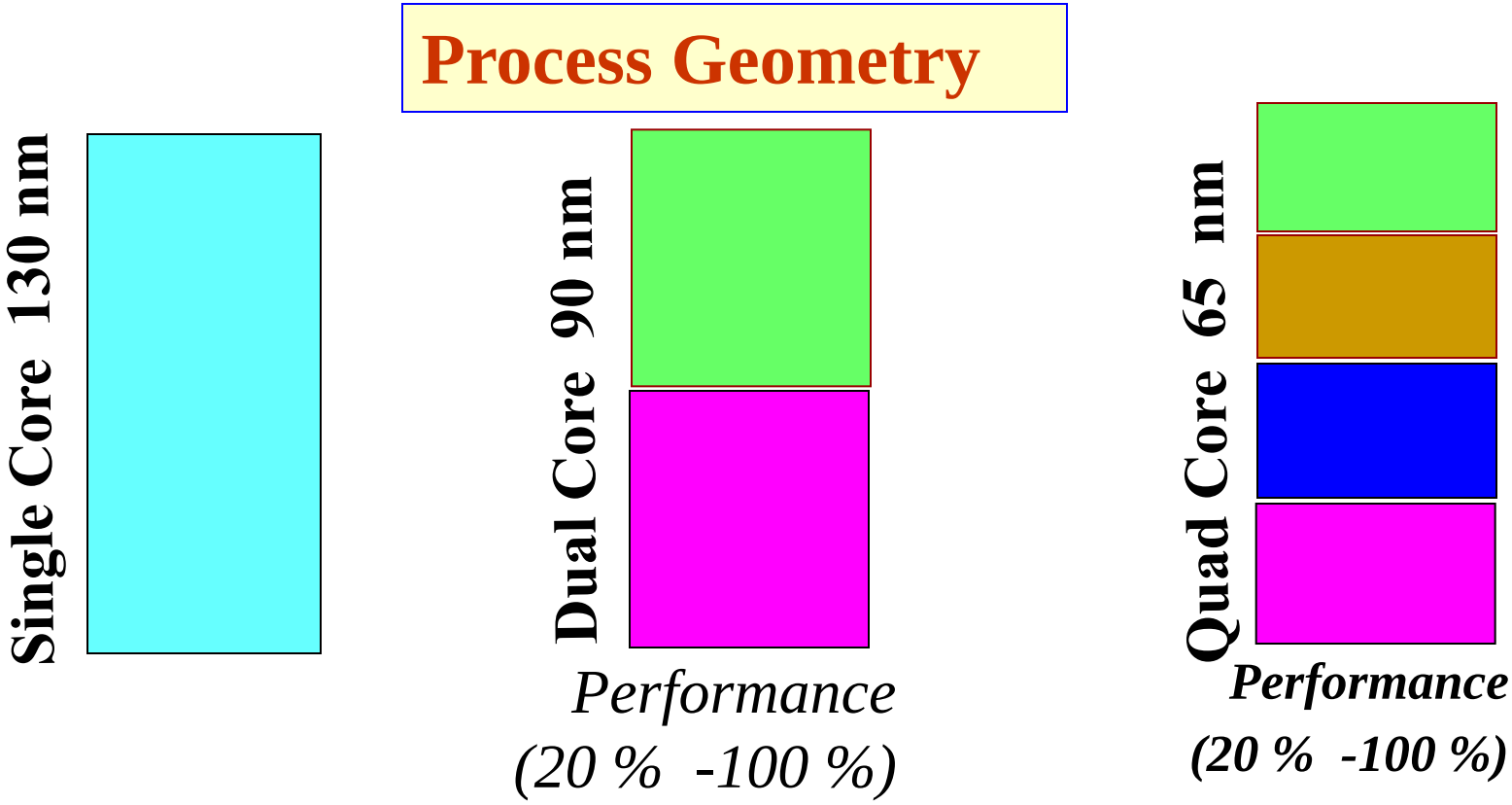
- Share the 400 MHz front-side bus (FSB)
  3.2 GB /s Bandwidth



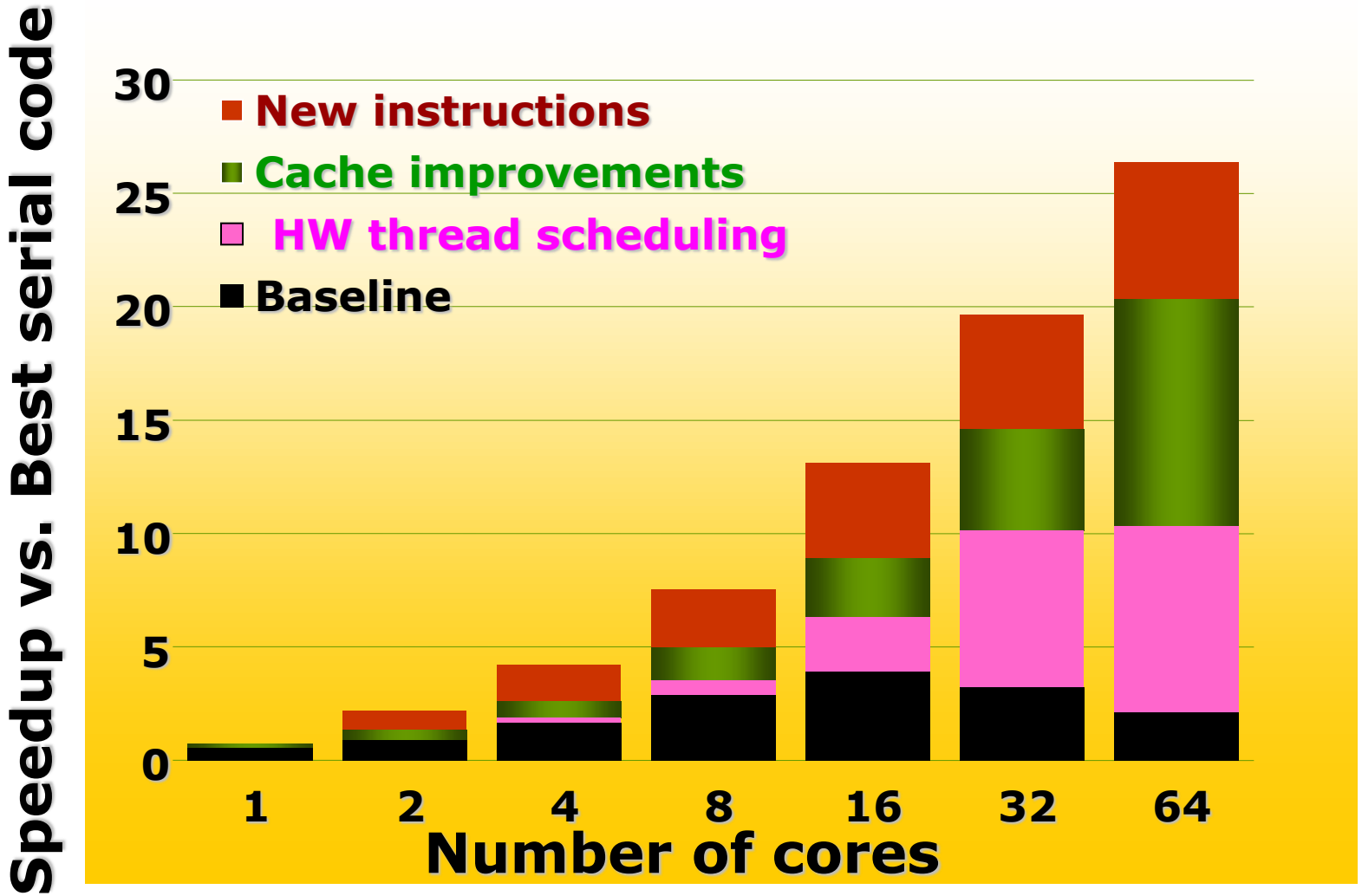
Source : http://www.intel.com; Reference : [6]

Software and Hardware Trends

- **Multi core to many core :**
  - **Dual core → Quad core → Eight Core**



Architecture-Algorithm Co-Design



Source : http://www.intel.com

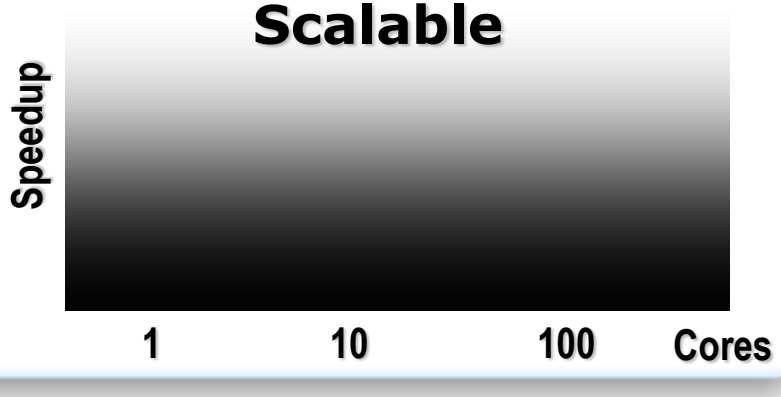
Creating Multi-Core Benchmarks

No. of Processors can be easily packed into a single rack

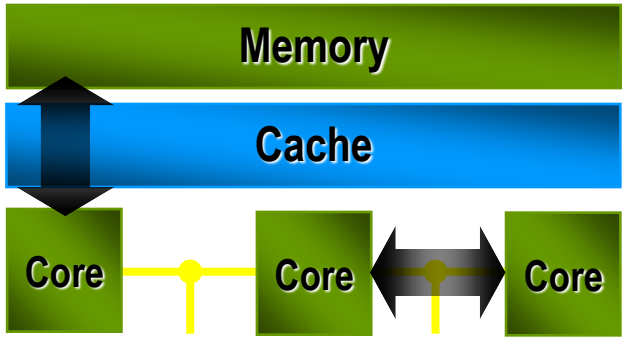
- Few Kilowatts
- CPU frequencies
- DRAM Content per system

Performance & Energy Aware

Highly Threaded & Scalable



Differentiated and Stressful



Source : http://www.intel.com

Multi Cores : Development Motivation

- CMOS manufacturing tech has physical limits of semiconductor based microelectronics become major design concern.
- Effect of these physical limitations can cause heat dissipation.
- It can also cause data synchronization problems.
- A combination of increased available space due to refined manufacturing processes and the demand for increased TLP led to creation of multi-core CPUs.

Source : http://www.intel.com ; http://www.amd.com

Multi Core : Unique Challenges

Do We Target…

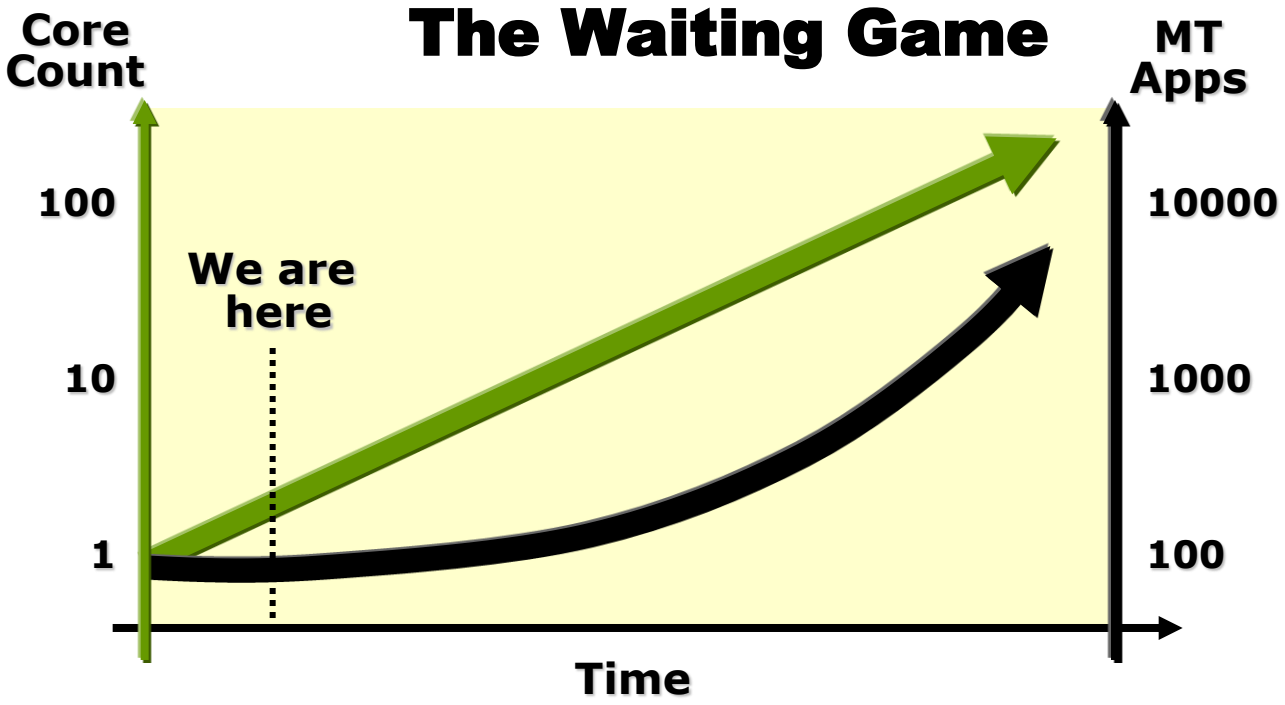
Bigger *OR* Smaller Cores

Performance *OR* Scalability

Compute *OR* I/O Intensive

Cache Friendly OR Memory Intensive

Multi Core Unique Challenges

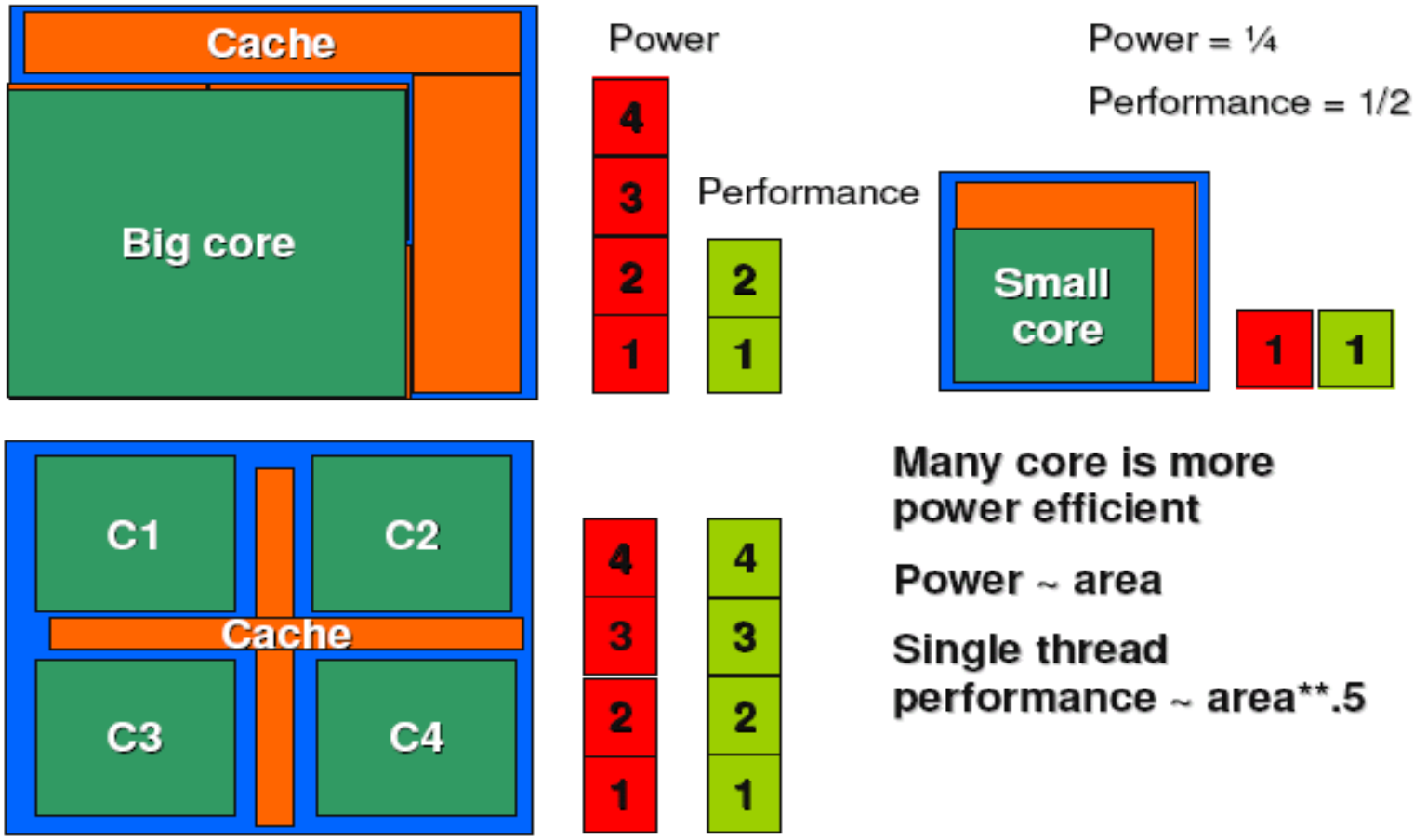


Designing **2010** Processors Today

Must **Anticipate** Future **Applications**

Source : http://www.intel.com

Multi Cores : Deliver more Performance per Watt



Why Multi-Core Matters ?

All about Processor Performance

- The recent shift from 130 nm to 90 nm process geometries doubled the transistor budgets available to chip designers. (*Manufactured in Cost effective manner can be achieved*)

- Dual Core Processor ability to increase the performance of many applications

Why Multi-Core Matters ?

All about Processor Performance : Several factors account for the decreased utility of clock frequency as a source of enhanced performance

- The mismatch between processor cycle time and DRAM cycle time (Memory gap)
- Increase in frequency force a chip to use more power, which in turn makes it harder (more expensive) to cool.

Harder to Power and harder to cool

Why Multi-Core Matters ?

Why do new systems Use so Much Power ?

- A few years ago, users were ecstatic, if they could fit 36IU 2-way (64 processors) servers in a single rack
- The equipment in a typical 19-inch 42U data centre rack rarely consumed more than a few kilowatts
- It is possible to fit could fit more than 64 or 128 processors in the same space
- New Technology improves the efficiency of power consumption (Intel /AMD)

Source : http://www.intel.com ; http://www.amd.com

Why Multi-Core Matters ?

Why do new systems Use so Much Power ?

- Harder to Power and harder to cool
- The system power consumption has grown, as CPU frequencies and DRAM content per system have increased in Multi-Core Systems
- If we let the power density on the chip continue [to increase],, it's hot plate……

Multi Cores : Advantage

- In today's digital world the demands of complex 3D simulations, streaming media files ,larger databases exceed single processor capability
- Multi-core enable true multitasking.
- Multi-core technology improve system efficiency and application performance
- Cache coherency circuitry can operate at a much higher clock rate than is possible if the signals have to travel off-chip
- Physically multi-core CPU designs require much less Printed Circuit Board space
- Less power requirement ?
- Less space

Source : http://www.intel.com ; http://www.amd.com

Multi Cores : Disadvantage

- Need efficient OS support
- Difficult to manage thermally
- Ultimately single CPU designs may make better use of the silicon surface area
- Multi (Two) processing cores sharing the same system Bus and memory bandwidth limits the real- world performance advantage.

Multi Core : Commercial Incentive

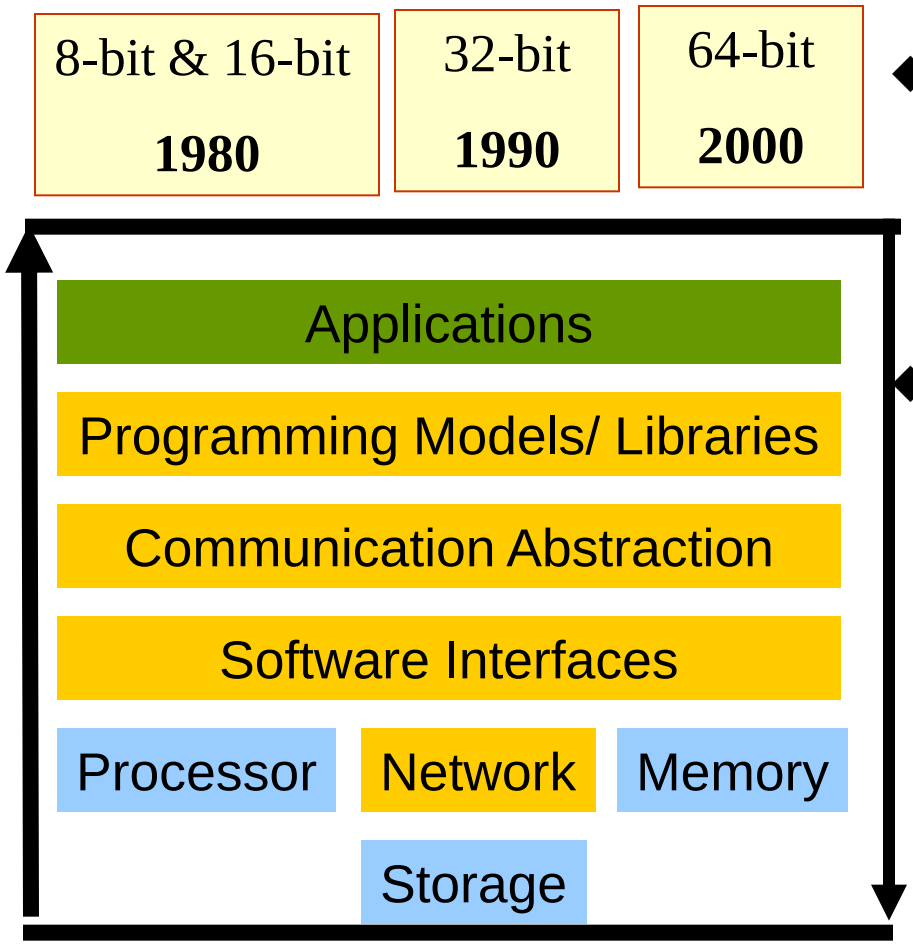
- SMP designs have been long implemented
- Supporting software's are well known
- Utilizing a proven processing core design without architectural changes reduces design risk to move to dual core technology
- There was increase difficulty of improving processor performance by only increasing frequency

Multi Core : Commercial Example

- International Business Machines (IBM)
  POWER4/POWER 5, Dual – Core Module processor
- IBM Cell Processors 2006
- Sun Microsystems
  UltraSPARC IV; UltraSPARC IV+
  UltraSPARC T1 8 core, 32 threads
- INTEL - **DUAL/QUAD** core processors (2007)
- AMD- **DUAL/QUAD** core processors (2007)

Source : http://www.intel.com ; http://www.amd.com
http:/www.ibm.com; http://www.sun.com

Multi Cores :32/64 bit Computing : Challenges



- Developmental opportunities exist at very level
  - Yesterday's HPC is today's commodity
- Performance in all layers not kept-up with the advances in processor technology
  - (Broadly) Engineering, Technology and Commercial issues in hardware layers
  - Schema and Abstraction issues in software layers

Multi Core : 32 bit /64 bit Computing

- 64 bit : refers to the size of the addresses the processor uses to organise the system main memory banks
- 32-bit processor can directly address as many as 4 Gigabytes (Billion Bytes) in the main memory
- 64 bit system can address **16 Exabytes** (that is **16 million Gigabytes**)
- Run database applications, Allow more concurrent users and applications to access data, more memory a processor can access at a time, Compilation & Execution, Accuracy of the precision
- Inexpensive 64-bit processors

Multi Core Programming Tools

- Out of Order Execution
- Multitasking
- Pre-emptive and Co-operative Multitasking
- SMP to the rescue
- Super threading with Multi threaded Processor
- Hyper threading the next step (Implementation)
- Caching and SMT

Source : [6], http://www.intel.com

Programming Multicore Processors

- Explicit Parallel Programming
  - Thread-based Programming Models.
  - Data Parallel Programming Models
  - Stream Programming Models
- Automatic Parallelization
  - Features of Most compliers for SMP systems, but currently see very little practical use
  - Polyhedral framework for dependencies and loop transformations – enabling composition of complex transformations over multiple statements.

Multi Core : Performance oriented Prog.

❖**Two issues to be addressed**

- How well does the single-threaded version run ?
- How well can the work be divided up among multiple processors with the least amount of overhead ?
- Are we implemented well-designed algorithm ?
- Are we implemented well-tuned application ?

Multi Core : Performance oriented Prog.

- The Underlying performance of the single-threaded code
- The percentage of the program that is run in parallel and its scalability
- CPU utilization, effective data sharing, data locality and load balancing
- The amount of synchronization and communication among the threads
- Memory Conflicts caused by shared memory or falsely shared memory.

Memory Performance of Dual Core Systems

- The Latency incurred in accessing different levels of memory is crucial in cache un-friendly applications involving gather/scatter operations – Sparse Matrix Computations
- Hardware support for data pre-fetching is available in most platforms as means of hiding memory latency
  - The detection and implementation of data pre-fetch streams varies with platform and compiler.
  - Advantage of potential increase in memory bandwidth, and offset latency.
- Write code so that a compiler find it easy to locate optimizations
- Reduce the Overheads due to Multi-Threaded Programming.

Chip Multiprocessors

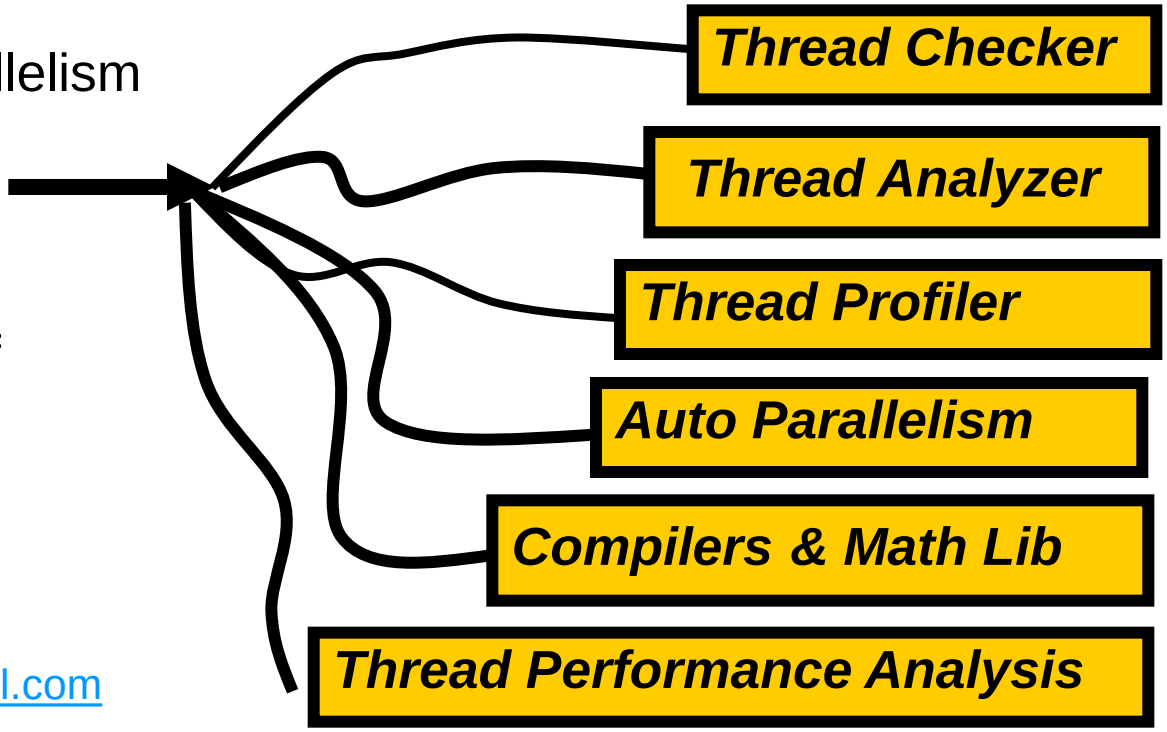
- Several CPU Cores
  - Independent execution
  - Symmetric (for now)
- Share Memory Hierarchy
  - Private L1 Caches
  - Shared L2 Cache (Intel Core)
  - Private L2 Caches (AMD)

  (kept coherent via crossbar)
  - Shared Memory Interface
  - Shared System Interface
- Lower clock speed

Multi Core Programming Tools

Intel Programming Tools : Intel Thread Building Blocks

- Performance
- Tools to Discover Parallelism
- Use of Math Libraries
- Measure Overheads of Threads
- Hardware Counters



Source : [6], http://www.intel.com

Conclusions and summary

- Why Multi Core ?
- Think of Abstract programming model
- Advantage of Multi Cores
- Multi core challenges

References

1. Andrews, Grogory R. **(2000),** Foundations of Multithreaded, Parallel, and Distributed Programming, Boston, MA : Addison-Wesley

2. Butenhof, David R **(1997),** Programming with POSIX Threads , Boston, MA : Addison Wesley Professional

3. Culler, David E., Jaswinder Pal Singh **(1999),** Parallel Computer Architecture - A Hardware/Software Approach , San Francsico, CA : Morgan Kaufmann

4. Grama Ananth, Anshul Gupts, George Karypis and Vipin Kumar **(2003),** Introduction to Parallel computing, Boston, MA : Addison-Wesley

5. Intel Corporation, **(2003),** Intel Hyper-Threading Technology, Technical User's Guide, Santa Clara CA : Intel Corporation Available at : http://www.intel.com

6. Shameem Akhter, Jason Roberts **(April 2006),** Multi-Core Programming - Increasing Performance through Software Multi-threading , Intel PRESS, Intel Corporation,

7. Bradford Nichols, Dick Buttlar and Jacqueline Proulx Farrell **(1996),** Pthread Programming O'Reilly and Associates, Newton, MA 02164,

8. James Reinders, Intel Threading Building Blocks – (**2007**) , O'REILLY series

9. Laurence T Yang & Minyi Guo (Editors), (**2006**) *High Performance Computing - Paradigm and Infrastructure* Wiley Series on Parallel and Distributed computing, Albert Y. Zomaya, Series Editor

10. Intel Threading Methodology ; Principles and Practices Version 2.0 copy right **(March 2003),** Intel Corporation

References

11. William Gropp, Ewing Lusk, Rajeev Thakur **(1999),** Using MPI-2, Advanced Features of the Message-Passing Interface, The MIT Press..

12. Pacheco S. Peter, **(1992),** Parallel Programming with MPI, , University of Sanfrancisco, Morgan Kaufman Publishers, Inc., Sanfrancisco, California

13. Kai Hwang, Zhiwei Xu, (**1998**), Scalable Parallel Computing (Technology Architecture Programming), McGraw Hill New York.

14. Michael J. Quinn (**2004**), Parallel Programming in C with MPI and OpenMP McGraw-Hill International Editions, Computer Science Series, McGraw-Hill, Inc. Newyork

15. Andrews, Grogory R. **(2000),** Foundations of Multithreaded, Parallel, and Distributed Progrmaming, Boston, MA : Addison-Wesley

16. SunSoft. Solaris multithreaded programming guide. SunSoft Press, Mountainview, CA, **(1996),** Zomaya, editor. Parallel and Distributed Computing Handbook. McGraw-Hill,

17. Chandra, Rohit, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, and Ramesh Menon, **(2001)**,Parallel Programming in OpenMP San Fracncisco Moraan Kaufmann

18. S.Kieriman, D.Shah, and B.Smaalders **(1995),** Programming with Threads, SunSoft Press, Mountainview, CA. 1995

19. Mattson Tim, **(2002),** Nuts and Bolts of multi-threaded Programming Santa Clara, CA : Intel Corporation, Available at : http://www.intel.com

20. I. Foster **(1995,** Designing and Building Parallel Programs ; Concepts and tools for Parallel Software Engineering, Addison-Wesley (1995)

21. J.Dongarra, I.S. Duff, D. Sorensen, and H.V.Vorst **(1999),** Numerical Linear Algebra for High Performance Computers (Software, Environments, Tools) SIAM, 1999

References

22. OpenMP C and C++ Application Program Interface, Version 1.0". **(1998),** OpenMP Architecture Review Board. October 1998

23. D. A. Lewine. *Posix Programmer's Guide:* **(1991),** Writing Portable Unix Programs with the Posix. 1 Standard. O'Reilly & Associates, 1991

24. Emery D. Berger, Kathryn S McKinley, Robert D Blumofe, Paul R.Wilson, *Hoard : A Scalable Memory Allocator for Multi-threaded Applications* ; The Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX). Cambridge, MA, November (**2000)**. Web site URL : http://www.hoard.org/

25. Marc Snir, Steve Otto, Steyen Huss-Lederman, David Walker and Jack Dongarra, (**1998**) *MPI-The Complete Reference: Volume 1, The MPI Core, second edition* [MCMPI-07].

26. William Gropp, Steven Huss-Lederman, Andrew Lumsdaine, Ewing Lusk, Bill Nitzberg, William Saphir, and Marc Snir (**1998**) *MPI-The Complete Reference: Volume 2, The MPI-2 Extensions*

27. A. Zomaya, editor. Parallel and Distributed Computing Handbook. McGraw-Hill, **(1996)**

28. OpenMP C and C++ Application Program Interface, Version 2.5 (**May 2005**)", From the OpenMP web site, URL **: http://www.openmp.org/**

29. Stokes, Jon 2002 Introduction to Multithreading, Super-threading and Hyper threading *Ars Technica*, October **(2002)**

30. Andrews Gregory R. 2000, Foundations of Multi-threaded, Parallel and Distributed Programming, Boston MA : Addison – Wesley (**2000)**

31. Deborah T. Marr , Frank Binns, David L. Hill, Glenn Hinton, David A Koufaty, J . Alan Miller, Michael Upton, "Hyperthreading, Technology Architecture and Microarchitecture", Intel (**2000-01)**

Thank You

Any questions ?