The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

1967-8

Advanced School in High Performance and GRID Computing

3 - 14 November 2008

Modern architectures for HPC computation.

COZZINI Stefano

CNR-INFM Democritos
c/o SISSA
via Beirut 2-4
34014 Trieste
ITALY

Advanced School in
High Performance
and GRID Computing

eLab
Joint DEMOCRITOS/SISSA
Laboratory for e-Science

MODERN ARCHITECTURES FOR HPC COMPUTATION

Stefano Cozzini

CNR-INFM DEMOCRITOS, Trieste

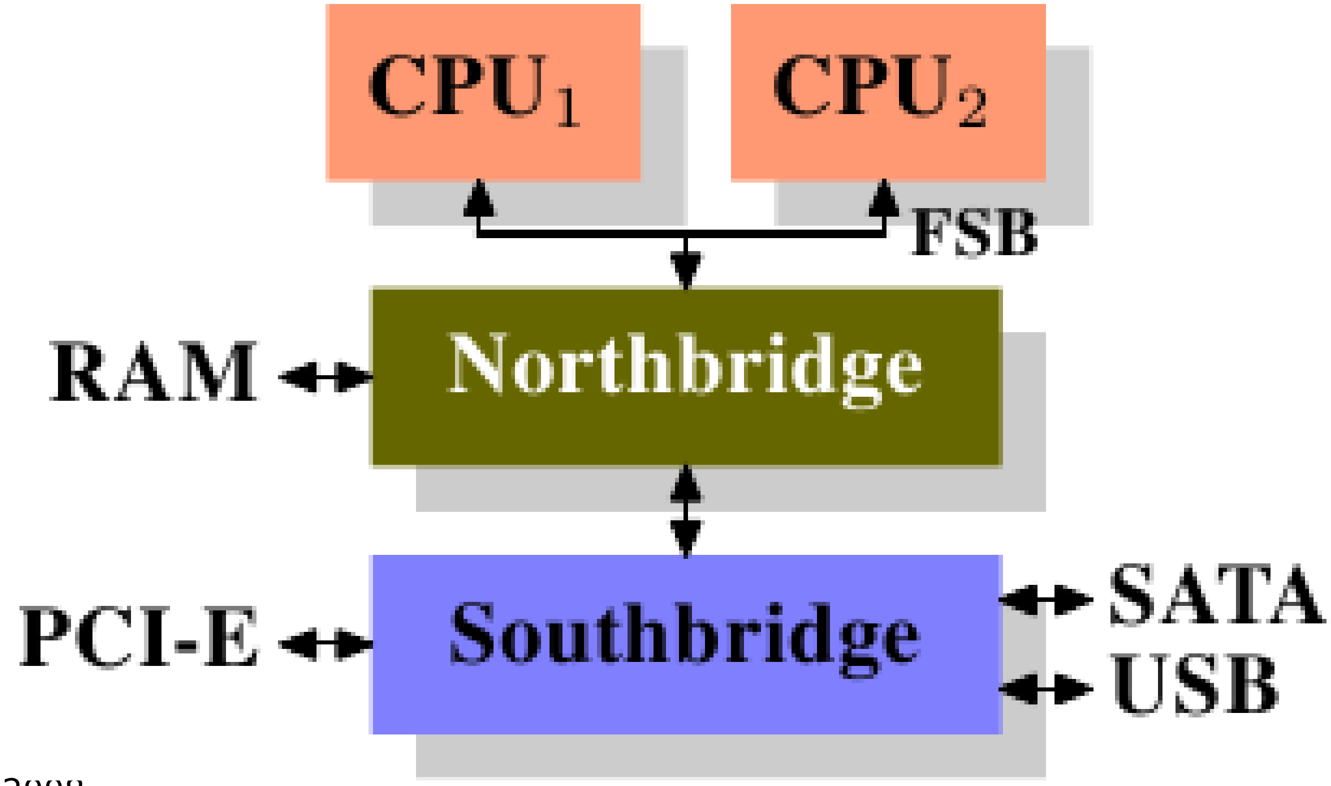
ICTP HPC School 2008 – Trieste, Italy - November 03-14, 2008

Introduction

- The goal of this lecture is to introduce some basic understanding of how the CPU and memory work together to perform a calculation.
- Classify the various parts of the computer into a hierarchy of performance based upon device response time.
- Relate the physical limitations of the hardware to the various performances of a type of computational operation.
- Point out possible bottle necks in calculations can occur and how this may be avoided.

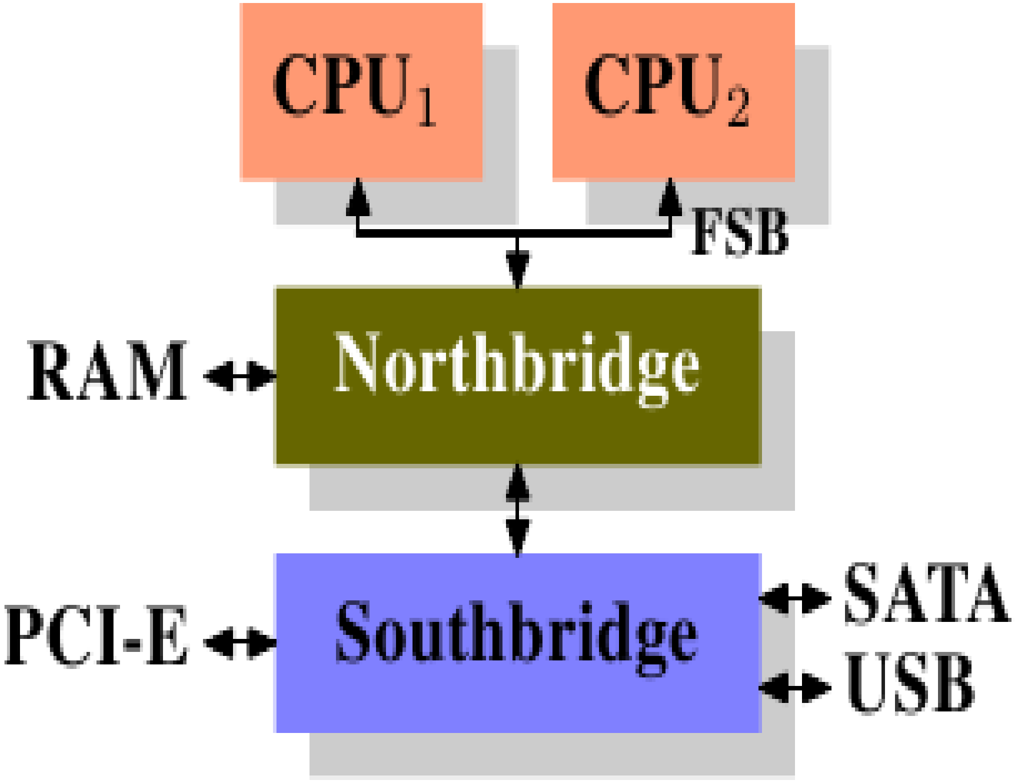
standard architecture

- Characteristics:
  - more than one CPU !
  - 64 bit adress space
  -



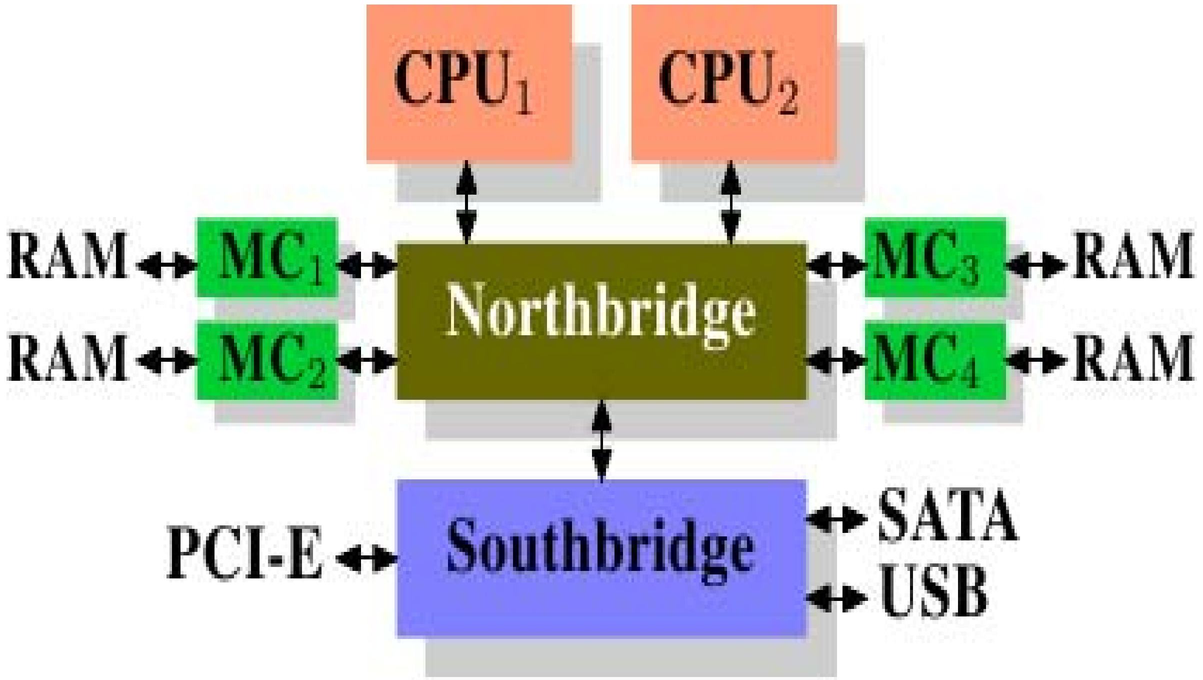
standard modern architecture

- All data communication from one CPU to another must travel over the same bus used to communicate with the Northbridge.
- All communication with RAM must pass through the Northbridge.
- Communication between a CPU and a device attached to the Southbridge is routed through the Northbridge.



more expensive architecture

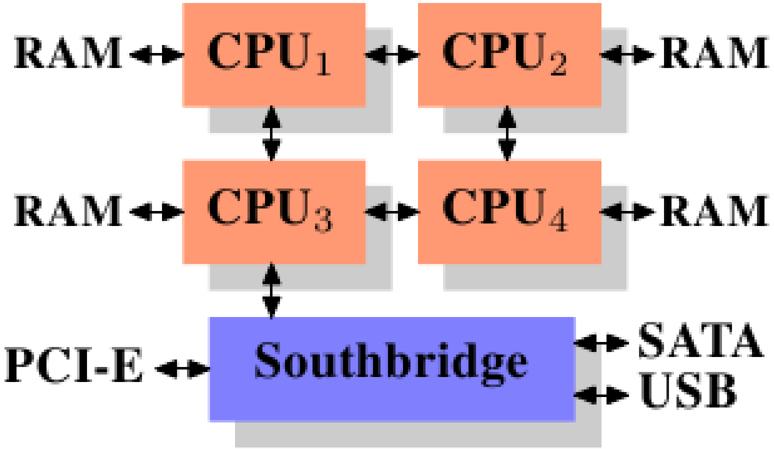
- Northbridge can be connected to a number of external memory controllers (in the following example, four of them).



INCREASE IN BANDWIDTH TOWARD MEMORY

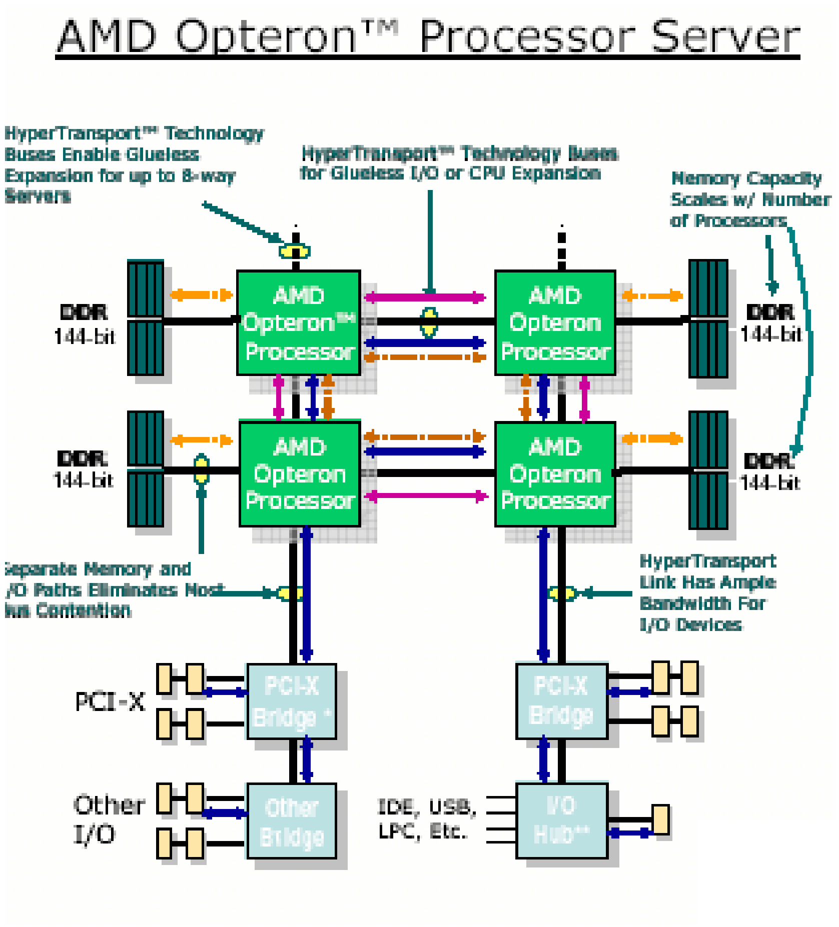
Another kind of architecture..

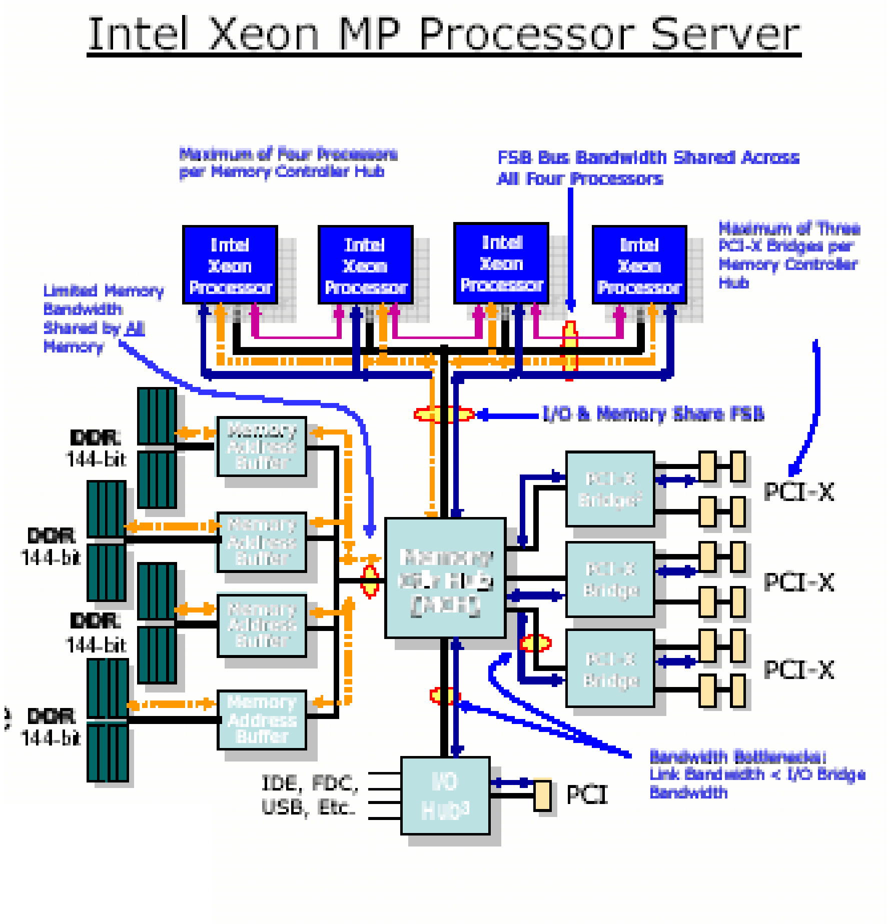
- Integrated memory controllers (AMD style)



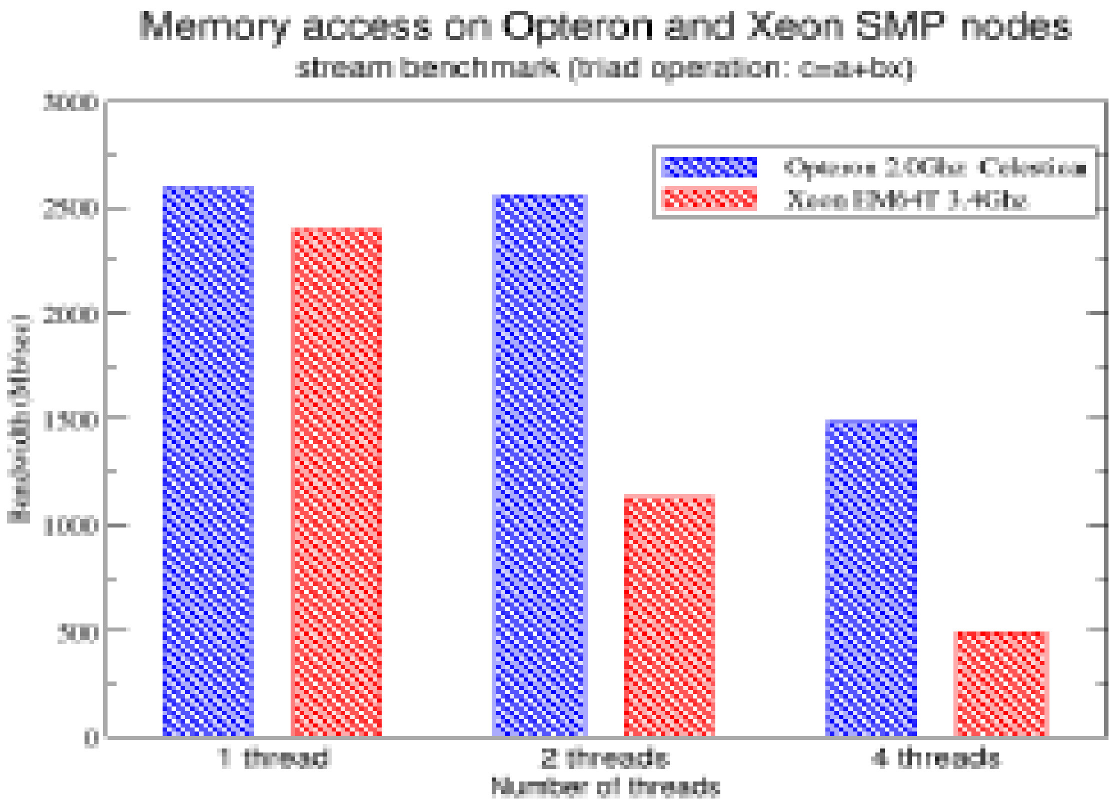
NUMA ARCHITECTURE !

AMD/Intel XEON comparison


AMD Opteron™ Processor Server
HyperTransport™ Technology Buses Enable Glueless Expansion for up to 8-way Servers
HyperTransport™ Technology Buses for Glueless I/O or CPU Expansion
Memory Capacity Scales w/ Number of Processors
DDR 144-bit
AMD Opteron™ Processor
AMD Opteron Processor
DDR 144-bit
DDR 144-bit
AMD Opteron Processor
AMD Opteron Processor
DDR 144-bit
eparate Memory and /O Paths Eliminates Most us Contention
HyperTransport Link Has Ample Bandwidth For I/O Devices
PCI-X
PCI-X Bridge *
PCI-X Bridge
Other I/O
Other Bridge
IDE, USB, LPC, Etc.
I/O Hub**



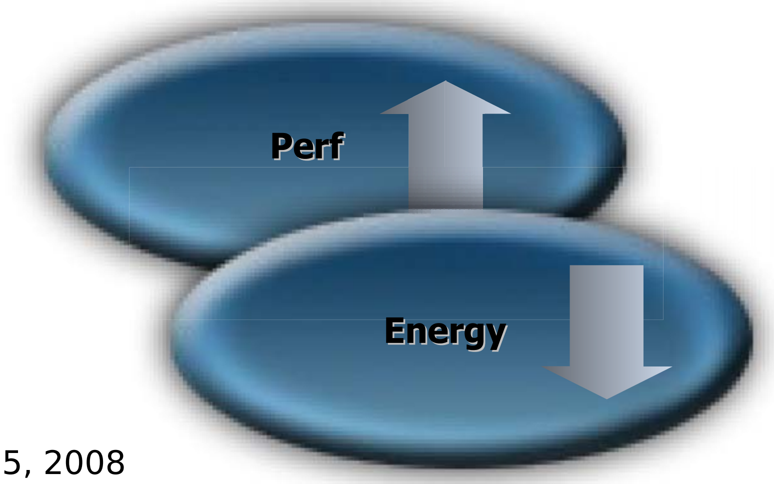
AMD/Intel XEON comparison



which kind of CPUS ?

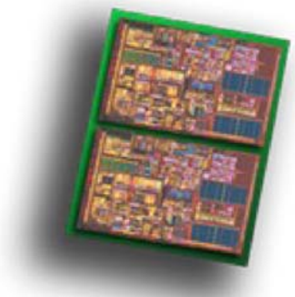
- MULTICORE !!

Multiple, externally visible processors on a single die where the processors have independent control-flow, separate internal state and no critical resource sharing



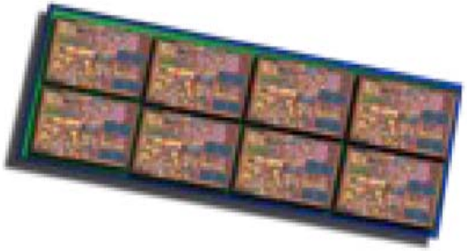
Evolutionary configurable architecture

Large, Scalar cores for high single-thread performance



Dual core

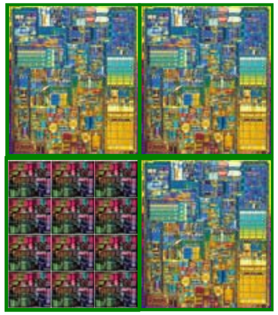
- Symmetric multithreading

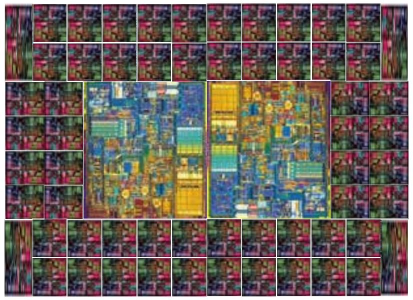


Multi-core array

- CMP with ~10 cores

Scalar plus many core for highly threaded workloads





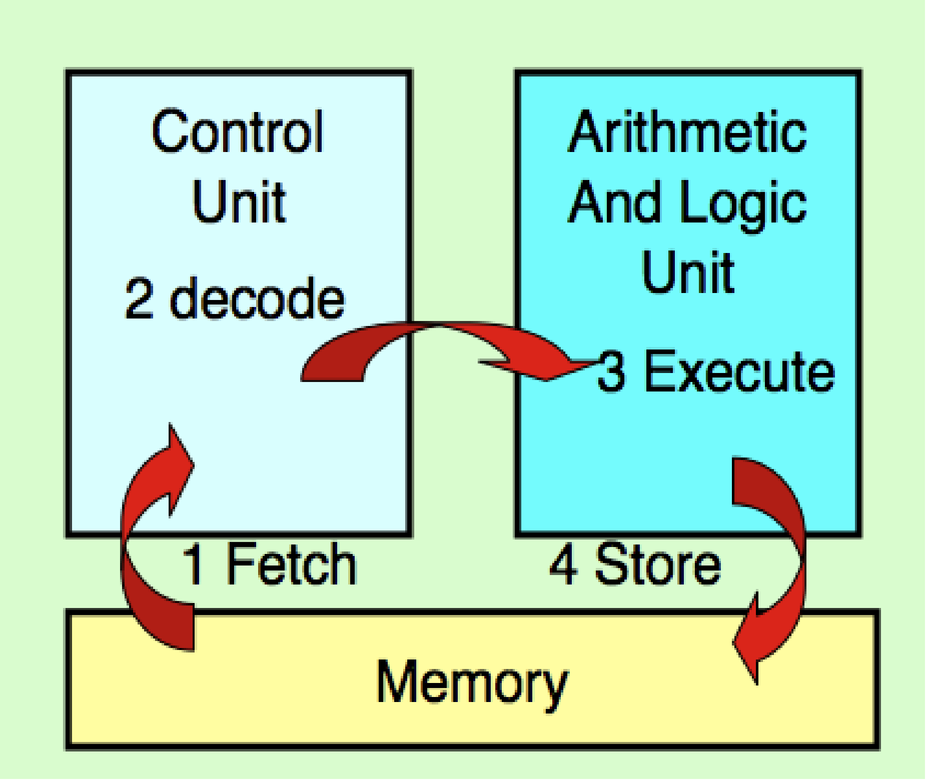
Many-core array

- CMP with 10s-100s low power cores
- Scalar cores
- Capable of TFLOPS+
- Full System-on-Chip
- Servers, workstations, embedded…

Micro2015: Evolving Processor Architecture, Intel® Developer Forum, Mar 2005

What within a core ?

- CPU contains Control Unit: processes instructions and ALU: math and logic operations
- At each cycle the CPU fetches both data and a description of what operations need to be performed and stores them in registers.
- On modern CPU there are many other stuff:
  - Pipelined functional units
  - Superscalar execution
  - Floating point instruction set extensions



Pipelined Functional Units

- For the processors in most modern parallel machines, the circuitry on the chip which performs a given type of operation on operands in registers is known as a *functional unit*.
- Most integer and floating point functional units are *pipelined*, meaning that they can have multiple independent executions of the same instruction placed in a queue. The idea is that after an initial startup latency, the functional unit should be able to generate one result every clock period (CP).
- Each stage of a pipelined operation can be working simultaneously on different sets of operands.

modern processors are superscalar !

- Processors which have multiple functional units which can operate concurrently are said to be superscalar.
- Examples:
  - AMD Opteron
    - 3 Floating point/MMX/SSE units
    - 3 Integer units
    - 3 Load/store units
  - Intel Xeon
    - 2 Floating point units
    - 2 Integer units
    - 2 Load/store units

Floating Point Instruction Set Extensions

- additional floating point instructions beyond the usual floating point add and multiply instructions:
  - Square root instruction --usually not pipelined!
    - AMD Opteron / Intel Xeon
  - SIMD (a.k.a. vector) floating point instructions
    - AMD Opteron/ Intel Xeon
    - IBM Cell –designed around the concept!
- Combined floating point multiply/add (MADD) instruction
  - AMD Opteron ("Barcelona" and after, using SIMD)
  - Intel Xeon ("Woodcrest" and after, using SIMD)

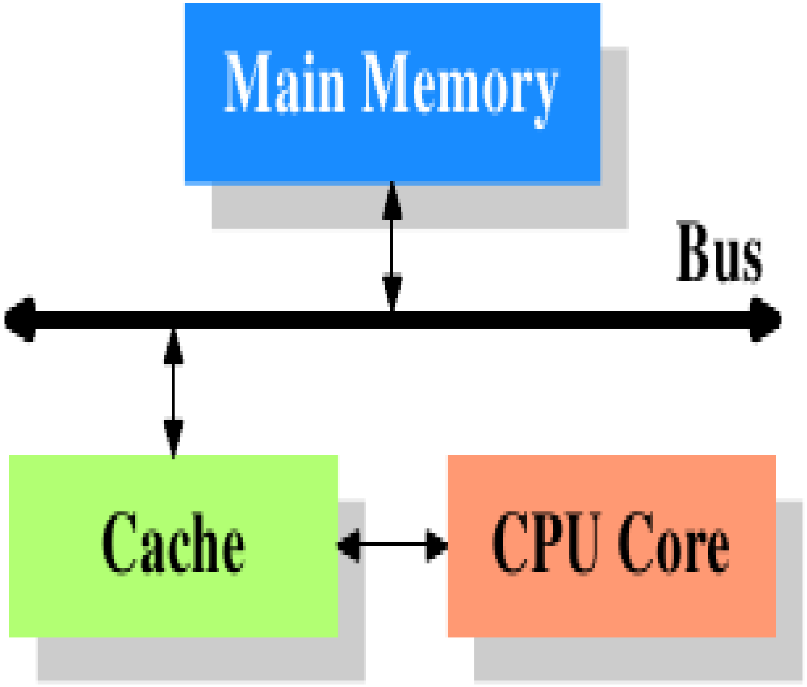
Instruction Set Extensions

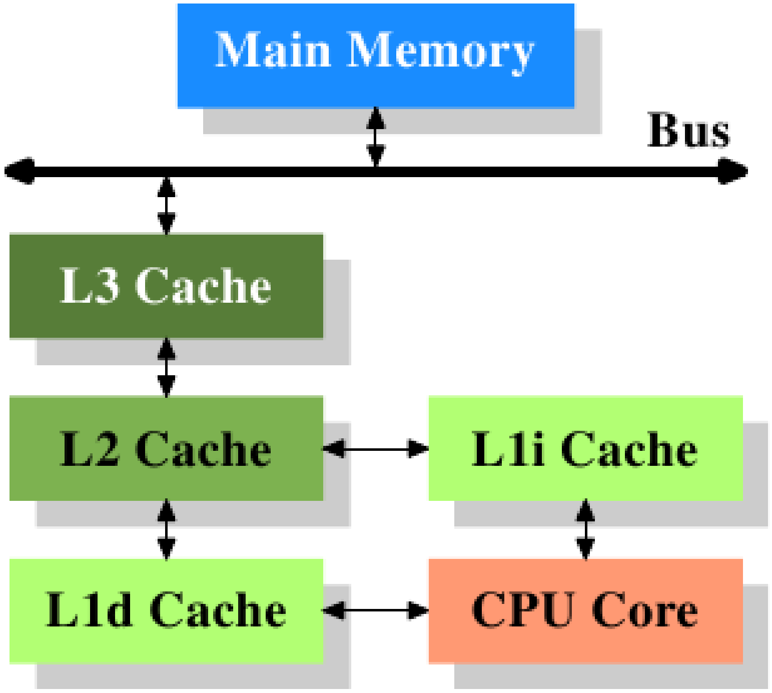
- Intel MMX (Matrix Math eXtensions)/ SSE (Streaming SIMD Extensions) / SSE2 (Streaming SIMD Extensions 2)
- AMD 3DNow! / AMD 3DNow!+ (or 3DNow! Professional, or 3DNow! Athlon) ...
-
- To check what you have on your machine:
  - cat /proc/cpuinfo

CACHE and MEMORY

- CACHE:A store of things that will be required in future, and can be retrieved rapidly. A cache may, or may not, be hidden.

[wikidictionary]

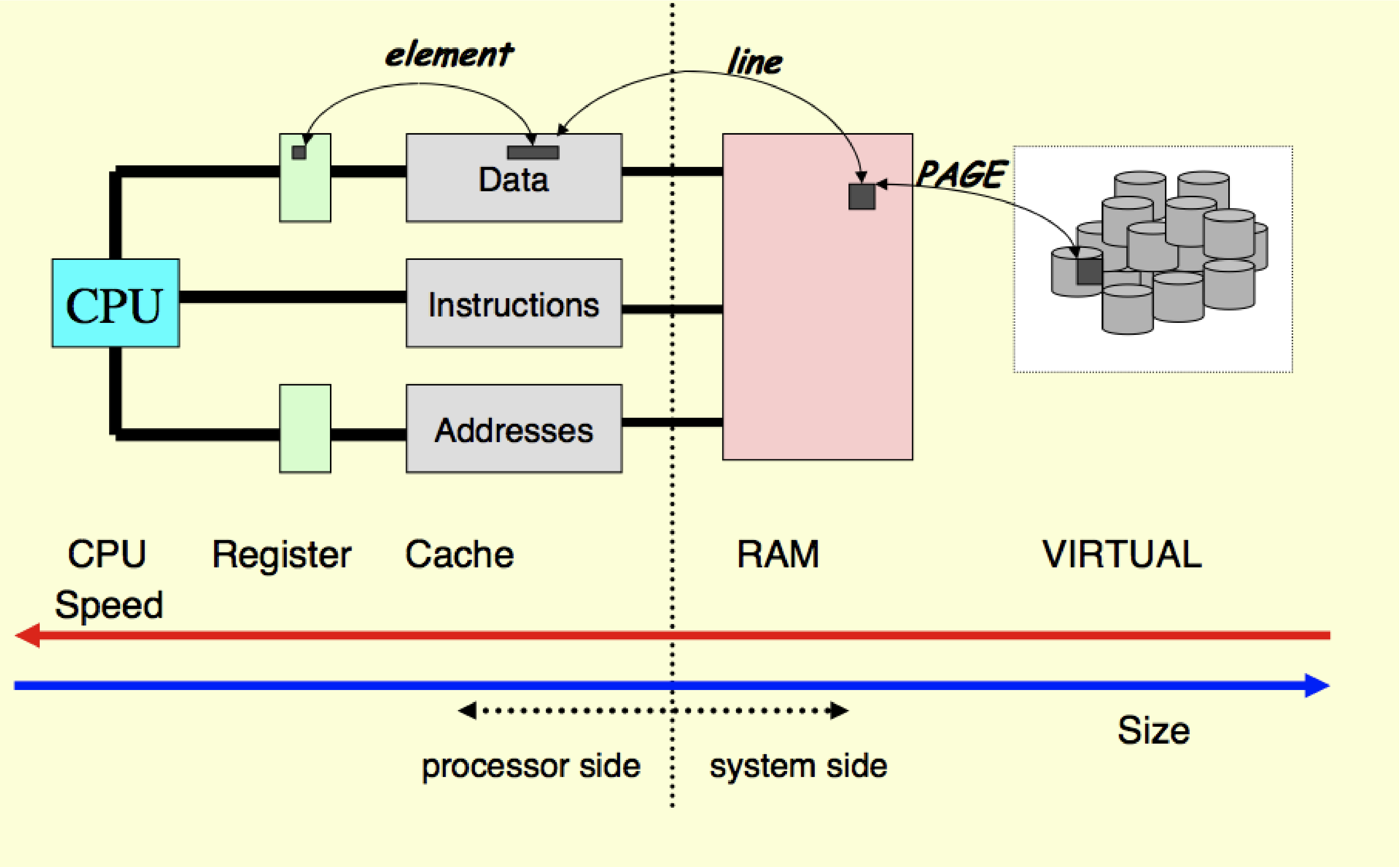




Hierarchy of memory..

- In modern computer system same data is stored in several storage devices during processing
- The storage devices can be described & ranked by their speed and “distance” from the CPU
- There is thus a hierarchy of memory objects
- Programming for a machine with memory hierarchy requires optimization for that memory structure.

Memory hierarchy



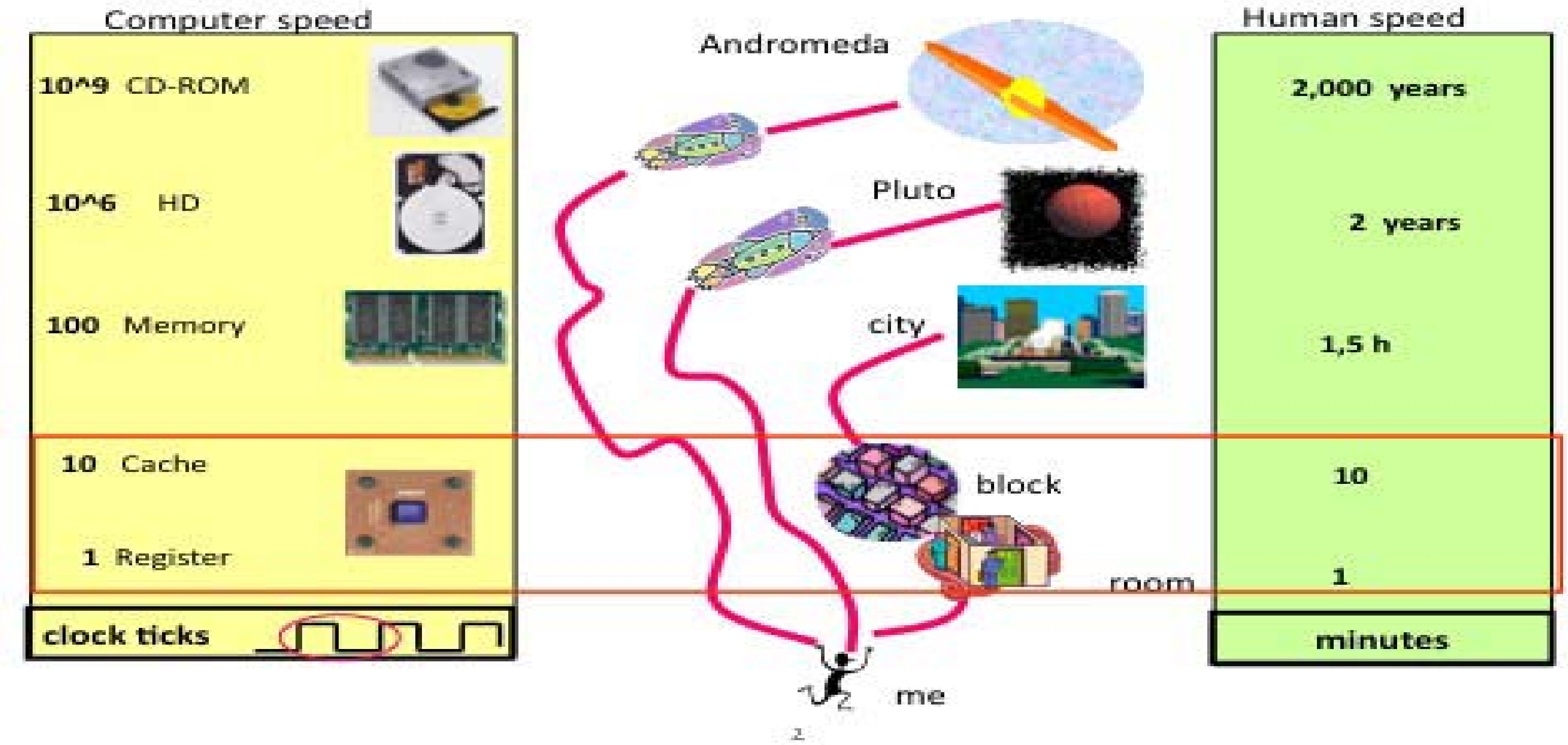
Components:

- **Registers:** On-chip circuitry used to hold operands and results of functional unit calculations.
- **L1 (Primary) Data Cache**: Small (on-chip) cache used to hold data about to operated on by processor.
- **L2 (Secondary) Cache**: Larger (on-or off-chip) cache used to hold data and instructions retrieved from local memory. Some systems also have L3 and even L4 caches.
- **Local Memory**: Memory on the same node as the processor.
- **Remote Memory**: Memory on another node but accessible to all processors in the network.
- **Disks**: Storage space where to save read large amount of data
- **Tapes/SAN:** space where to store data rarely needed.

Hierarchical Memory and Latency

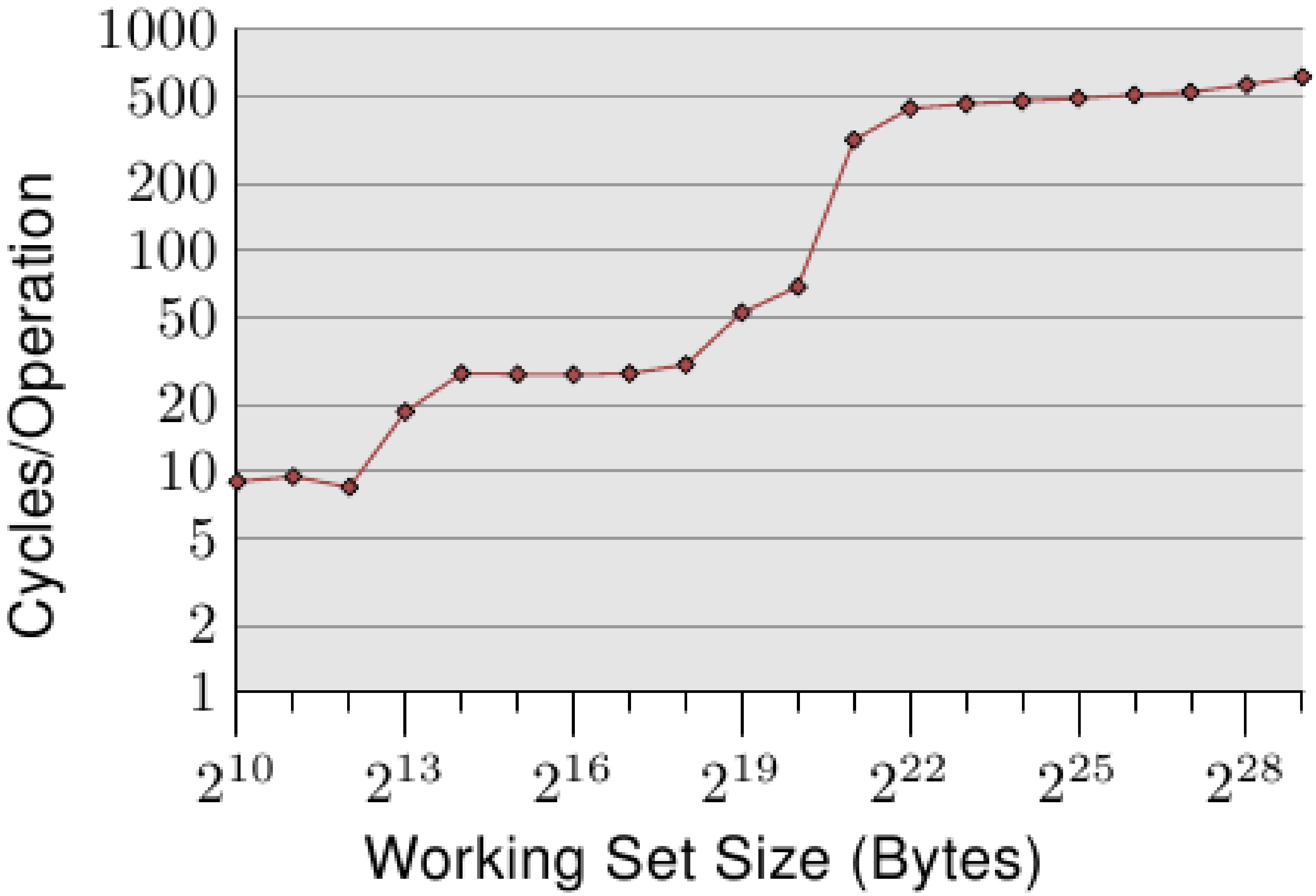
- The key to hierarchical memory is that going down each level of the hierarchy introduces approximately an order of magnitude more latency than the previous level.
-
- Actual latencies for an Opteron 8218 (2.6GHz):
  - L1 data cache: 3 CPs
  - L2 cache: 12 CPs
  - Local memory: 166 CPs

let's do some analogy...



SOURCE: JIM GRAY & GORDON BELL

how fast/large are the caches ? (afternoon' exercise)



Single core vs dual core and memory hierarchy:

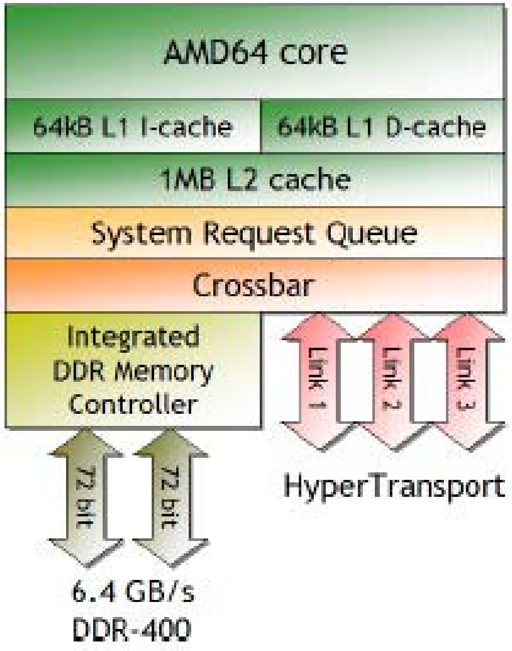


Figure 1: Single core AMD64 block diagram

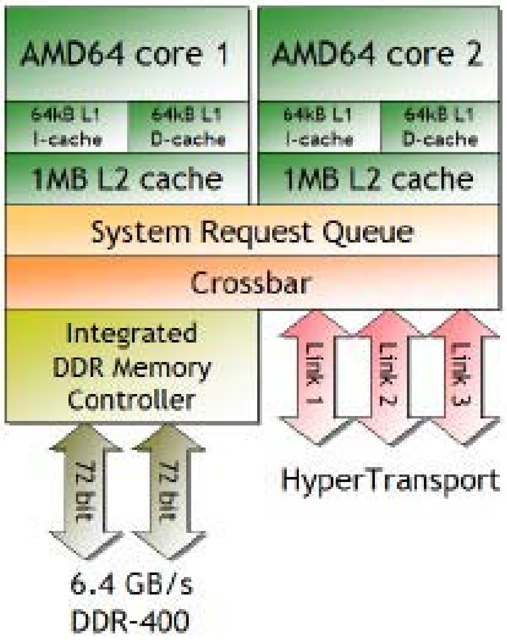
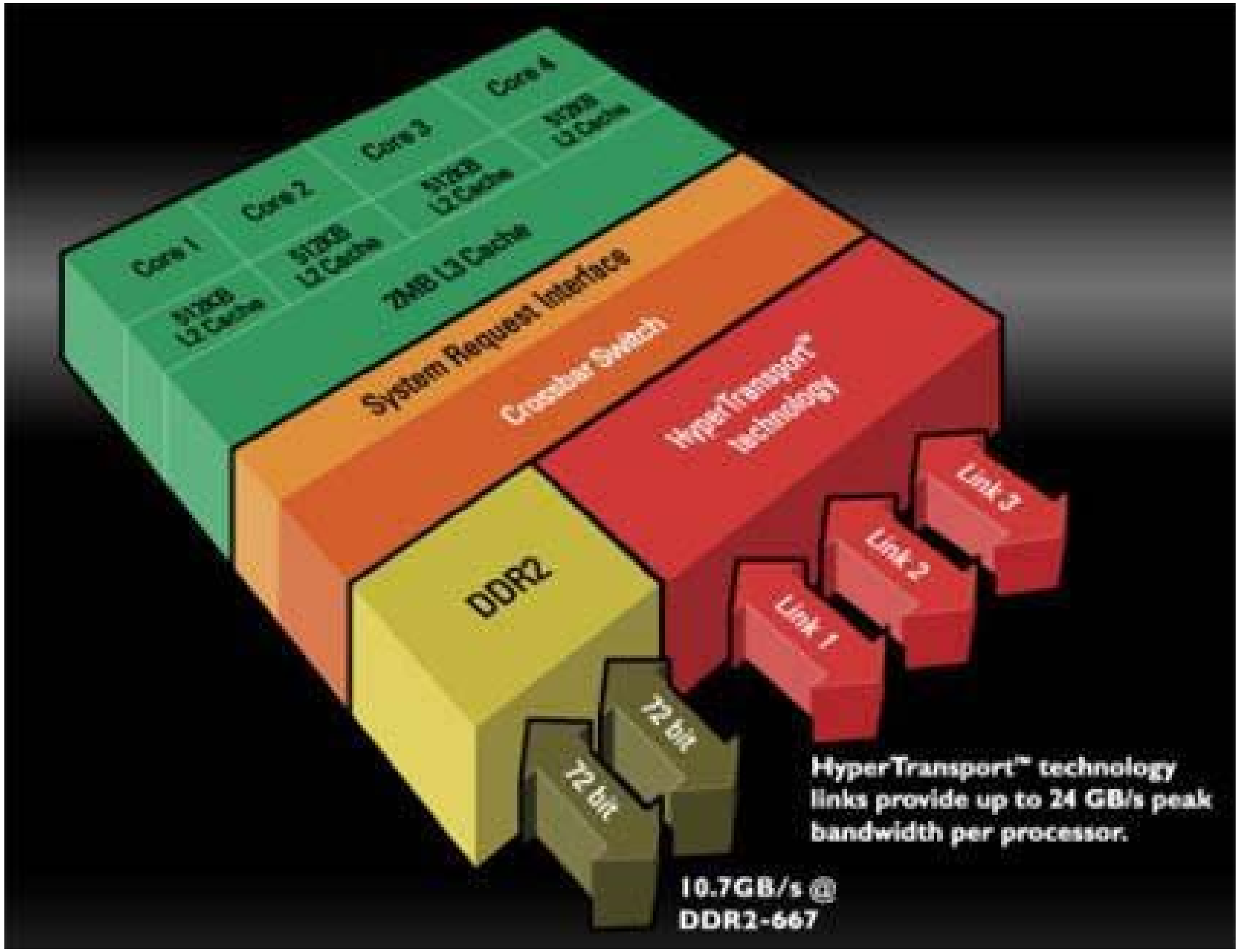


Figure 2: Dual core AMD64 block diagram

BANDWIDTH TOWARD LOCAL MEMORY
IS SHARED AMONG CORES !

Barcelona quad core architecture

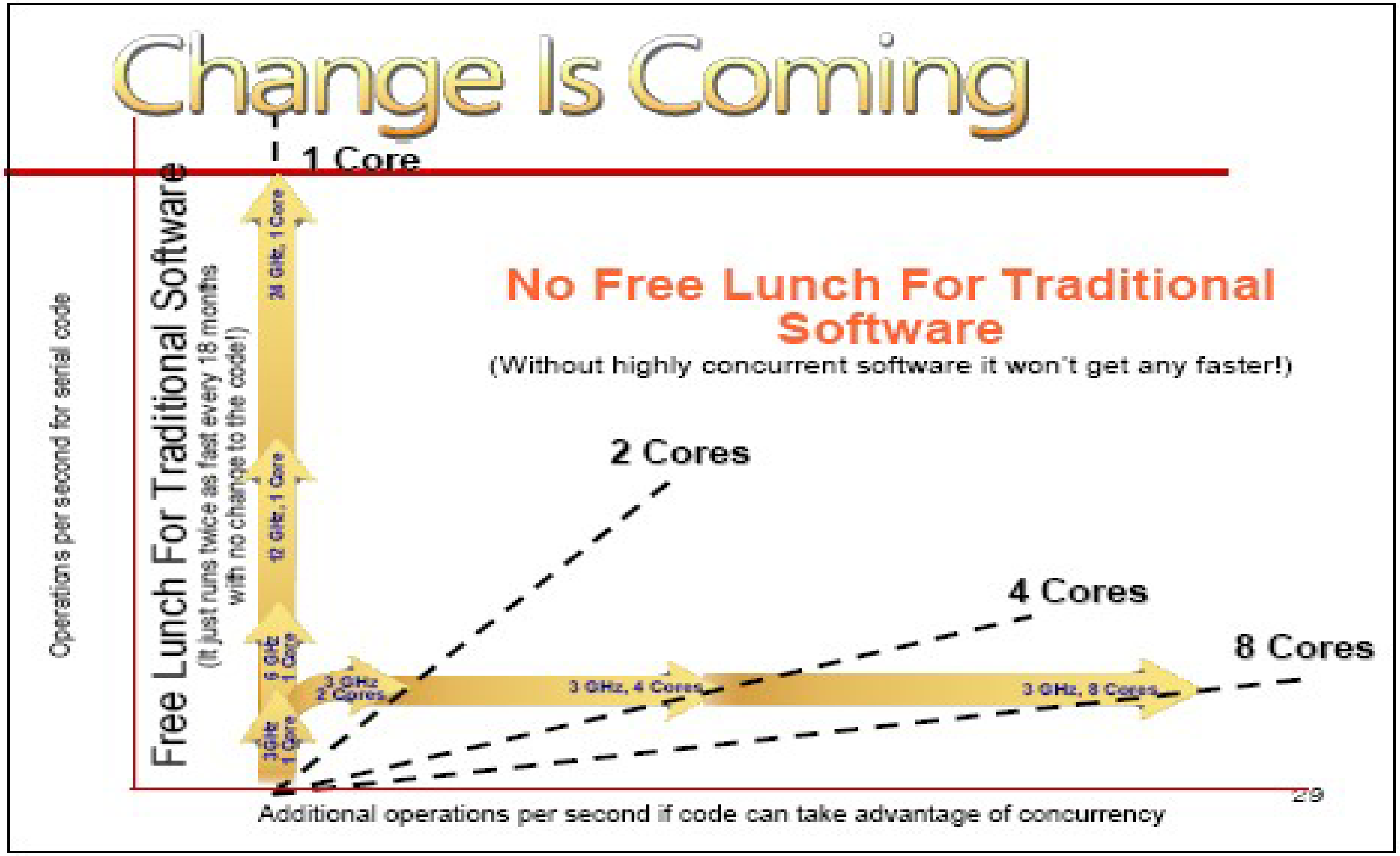


L3 CACHE IS SHARED AMONG CORES !

Few important issues

- Modern architectures have a high degree of parallelism some time hidden to the user
- In order to optimize on them you should be aware of this.
- In particolar:
  - SMP is not always valid: NUMA
  - not only RAM is shared but also L2/L3 Caches

single Core VS Multiple core (from J.Dongarra talk)


Change Is Coming
1 Core
Free Lunch For Traditional Software
(It just runs twice as fast every 18 months with no change to the code!)
Operations per second for serial code
24 GHz, 1 Core
12 GHz, 1 Core
6 GHz 1 Core
3GHz 1 Core
3 GHz 2 Cores
3 GHz, 4 Cores
3 GHz, 8 Cores
No Free Lunch For Traditional Software
(Without highly concurrent software it won't get any faster!)
2 Cores
4 Cores
8 Cores
Additional operations per second if code can take advantage of concurrency