



24<sup>th</sup> Summer  
School on  
**PARALLEL**  
COMPUTING

# Introduction to HPC Architectures

Andrew Emerson, Giovanni Erbacci - {a.emerson,g.erbacci}@cineca.it  
SuperComputing Applications and Innovation Department



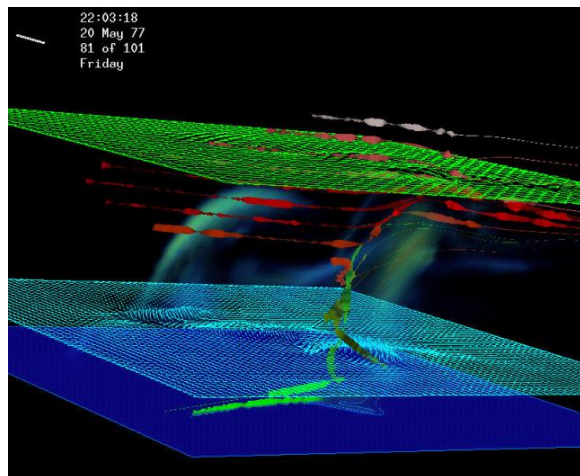


## Contents

- Computational Science
- Features of Traditional Supercomputers
  - Processors and Moore's Law
  - Memory bandwidth and Cache
- Aspects of parallelism
  - Functional Units
  - Pipelining and vectorisation
  - Flynn Taxonomy, memory distribution and networks
- HPC Trends
  - Bluegene and Accelerators
  - HPC Systems Evolution, Top500 and Cineca experience
  - PRACE and other European projects
- Summary



# Computational Sciences

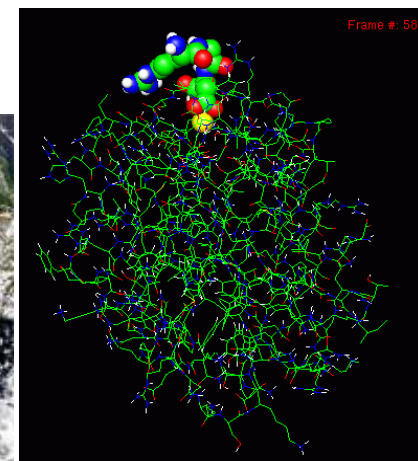


Identify the scientific disciplines that use mathematical models and computing systems to analyze and solve scientific problems.

Computational methods allow us to study complex phenomena, giving a powerful impetus to scientific research.

The use of computers to study physical systems allows to manage phenomena

- **very large**  
*(meteo-climatology, cosmology, data mining, oil reservoir)*
- **very small**  
*(drug design, silicon chip design, structural biology)*
- **very complex**  
*(fundamental physics, fluid dynamics, turbulence)*
- **too dangerous or expensive**  
*(fault simulation, **nuclear** tests, crash analysis)*





# Computational Sciences / 1

**Computational science** (with theory and experimentation), is the “third pillar” of scientific inquiry, enabling researchers to build and test models of complex phenomena



## The Nobel Prize in Chemistry 1998

"for his development of the density-functional theory"



**Walter Kohn**

"for his development of computational methods in quantum chemistry"



**John A. Pople**

## Owen Willans Richardson, Years '20

Nobel Prize in Physics 1928

*for his work on the thermionic phenomenon and especially for the discovery of the law named after him"*



## John Von Neumann, Years '40



## Kenneth. Wilson, Years '80

Nobel Prize in Physics 1982

*"for his theory for critical phenomena in connection with phase transitions"*





# Size of computational applications

## Computational Dimension:

number of operations needed to solve the problem, in general is a function of the size of the involved data structures ( $n$ ,  $n^2$ ,  $n^3$ ,  $n \log n$ , etc.)

## flop - Floating point operations

indicates an arithmetic floating point operation.

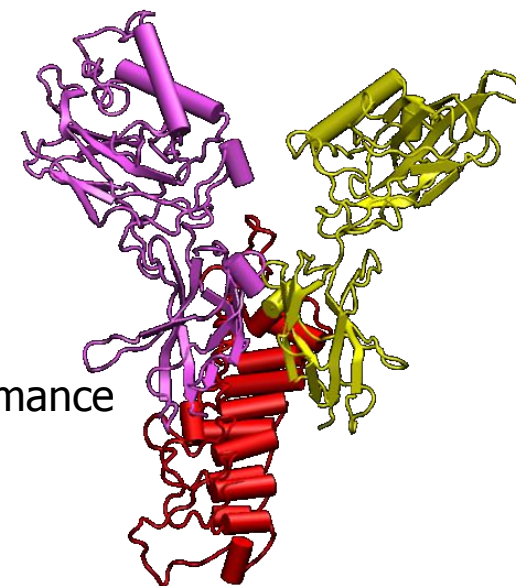
## flop/s - Floating points operations per second

is a unit to measure the speed of a computer.

computational problems today:  $10^{15} - 10^{22}$  flop

One year has about  $3 \times 10^7$  seconds!

Most powerful computers today have reach a sustained performance is of the order of Tflop/s - Pflop/s ( $10^{12} - 10^{15}$  flop/s).





## Example: Weather Prediction

Forecasts on a global scale (*.....too accurate and inefficient!!*)

### - 3D Grid to represent the Earth

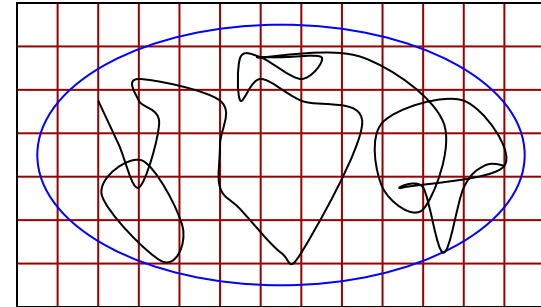
- Earth's circumference:  $\cong 40000$  km
- radius:  $\cong 6370$  km
- Earth's surface:  $\cong 4\pi r^2 \cong 5 \cdot 10^8$  km<sup>2</sup>

### - 6 variables:

- temperature
- pressure
- humidity
- wind speed in the 3 Cartesian directions

### - cells of 1 km on each side

- 100 slices to see how the variables evolve on the different levels of the atmosphere
- a 30 seconds time step is required for the simulation with such resolution
- Each cell requires about 1000 operations per time step (Navier-Stokes turbulence and various phenomena)



On a **global scale** this is currently a precision quite impossible. unimaginable!

On a **local scale** normally the cells are 10-15 km on each side



## Example: Weather Prediction / 1

**Grid:  $5 \cdot 10^8 \cdot 100 = 5 \cdot 10^{10}$  cells**

- each cell is represented with 8 Byte
- Memory space:

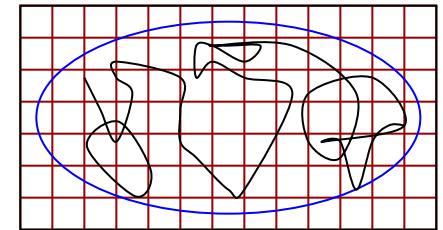
$$(6 \text{ var}) \cdot (8 \text{ Byte}) \cdot (5 \cdot 10^{10} \text{ cells}) \cong 2 \cdot 10^{12} \text{ Byte} = 2\text{TB}$$

A 24 hours forecast needs:

- $24 \cdot 60 \cdot 2 \cong 3 \cdot 10^3$  time-step
- $(5 \cdot 10^{10} \text{ cells}) \cdot (10^3 \text{ oper.}) \cdot (3 \cdot 10^3 \text{ time-steps}) = 1.5 \cdot 10^{17} \text{ operations !}$

A computer with a power of 1Tflop/s will take  $1.5 \cdot 10^5 \text{ sec.}$

- 24 hours forecast will need 2days to run
- ..... but we shall obtain a very accurate forecast

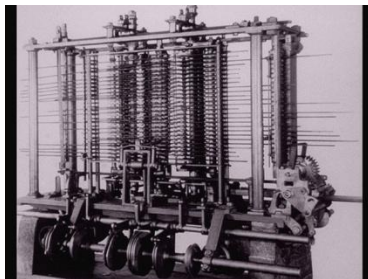




# Supercomputers

supercomputers are defined as the more powerful computers available in a given period of time.

**Powerful** is meant in terms of execution speed, memory capacity and accuracy of the machine.



**Supercomputer:** "*new statistical machines with the mental power of 100 skilled mathematicians in solving even highly complex algebraic problems*".

**New York World,** march 1920

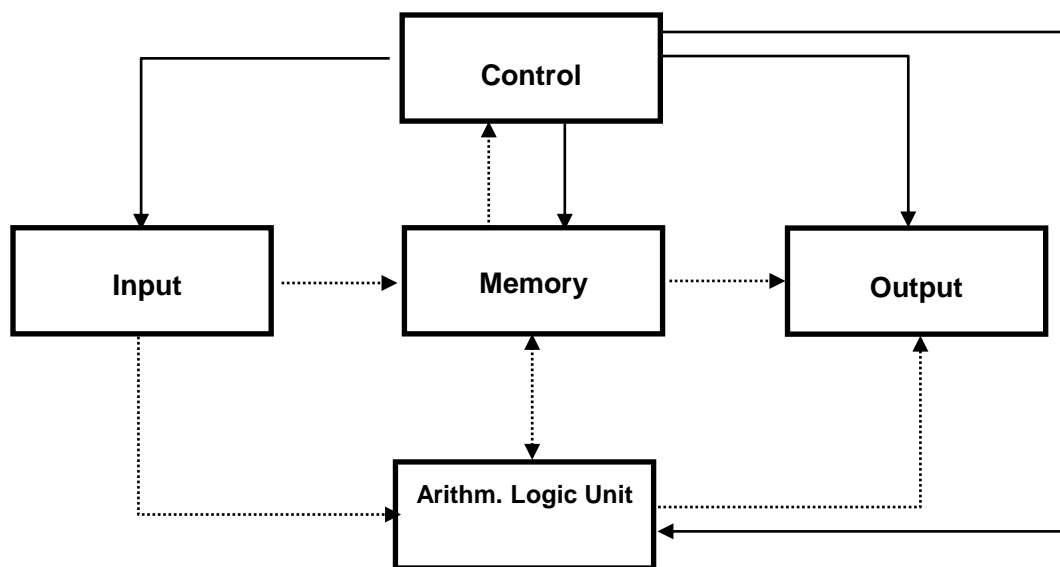
to describe the machines invented by Mendenhall and Warren, used at Columbia University's Statistical Bureau.



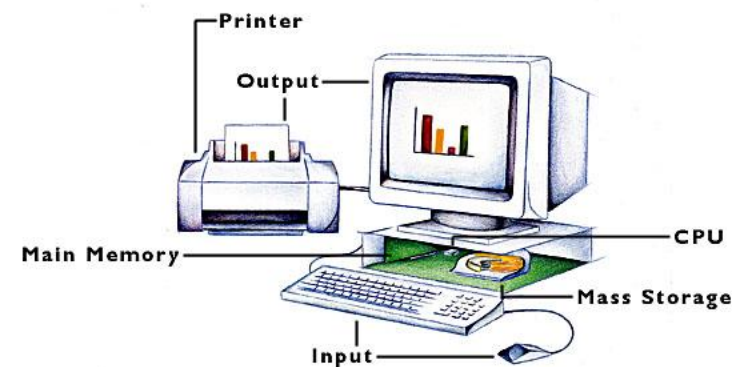


# von Neumann Model

## Conventional Computer



*Von Neumann Model of Computer Architecture*



..... Data  
\_\_\_\_\_ Control

### Instructions are processed sequentially

- 1 A single instruction is loaded from memory (**fetch**) and decoded
- 2 Compute the addresses of operands
- 3 Fetch the operands from memory;
- 4 Execute the instruction ;
- 5 Write the result in memory (**store**).



## Speed of Processors: Clock Cycle and Frequency

The **clock cycle**  $\tau$  is defined as the time between two adjacent pulses of oscillator that sets the time of the processor.

The number of these pulses per second is known as **clock speed** or **clock frequency**, generally measured in GHz (gigahertz, or billions of pulses per second).

The clock cycle controls the synchronization of operations in a computer: All the operations inside the processor last a multiple of  $\tau$ .

Processor	$\tau$ (ns)	freq (MHz)
CDC 6600	100	10
Cyber 76	27.5	36,3
IBM ES 9000	9	111
Cray Y-MP C90	4.1	244
Intel i860	20	50
PC Pentium	< 0.5	> 2 GHz
Power PC	1.17	850
IBM Power 5	0.52	1.9 GHz
IBM Power 6	0.21	4.7 GHz

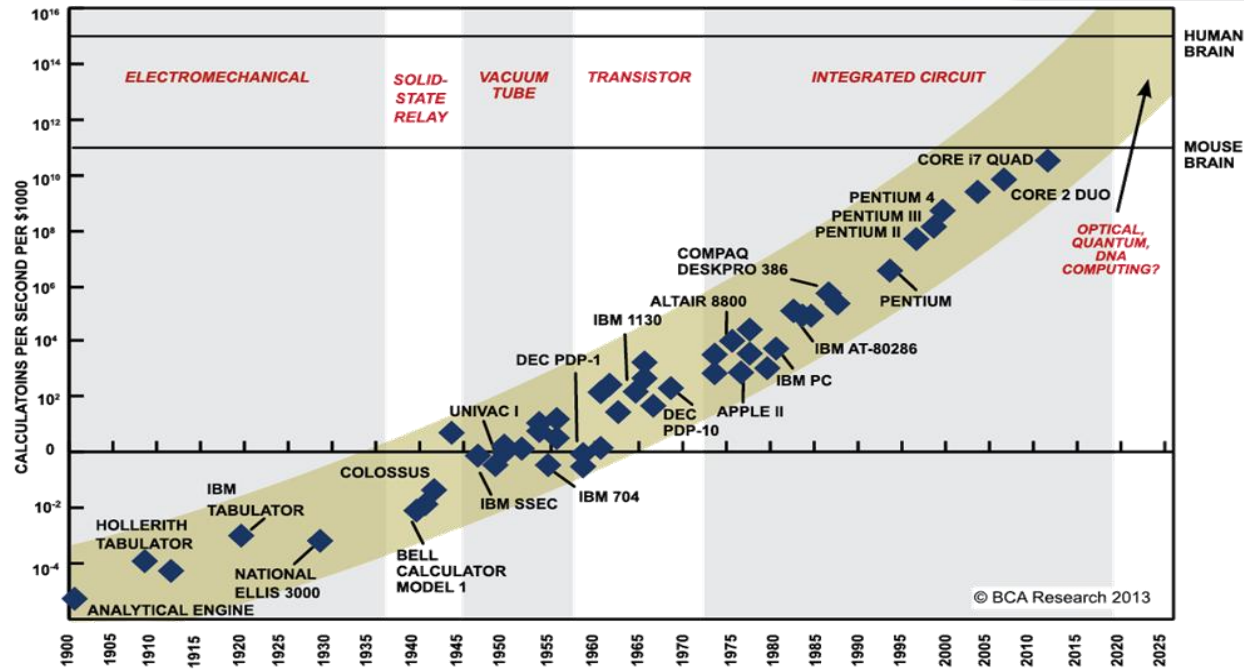
### Increasing the clock frequency:

The **speed of light** sets an upper limit to the speed with which electronic components can operate .

Propagation velocity of a signal in a vacuum: **300.000 Km/s = 30 cm/ns**

**Heat dissipation** problems inside the processor. Also Quantum tunnelling expected to become important\*

# Moore's Law



SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, THE VIKING PRESS, 2006. DATAPPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

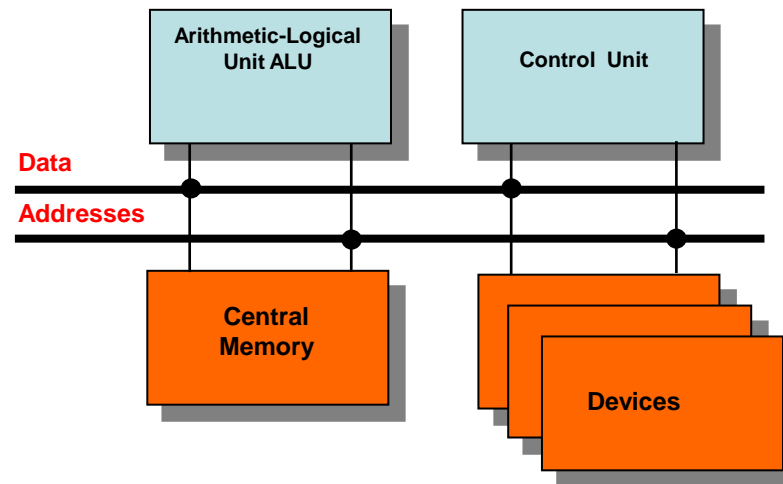
Empirical law which states that the complexity of devices (number of transistors per square inch in microprocessors) doubles every 18 months..

**Gordon Moore**, INTEL co-founder, 1965

It is estimated that Moore's Law still applies in the near future but applied to the number of cores per processor



# Other factors that affect Performance



In addition to processor power, other factors affect the performance of computers:

- Size of memory
- Bandwidth between processor and memory
- Bandwidth toward the I/O system
- Size and bandwidth of the cache
- Latency between processor, memory, and I/O system



# Memory hierarchies

**Time to run code = clock cycles running code + clock cycles waiting for memory**

**Memory access time:** the *time* required by the processor to *access* data or to *write* data from / to *memory*

The hierarchy exists because :

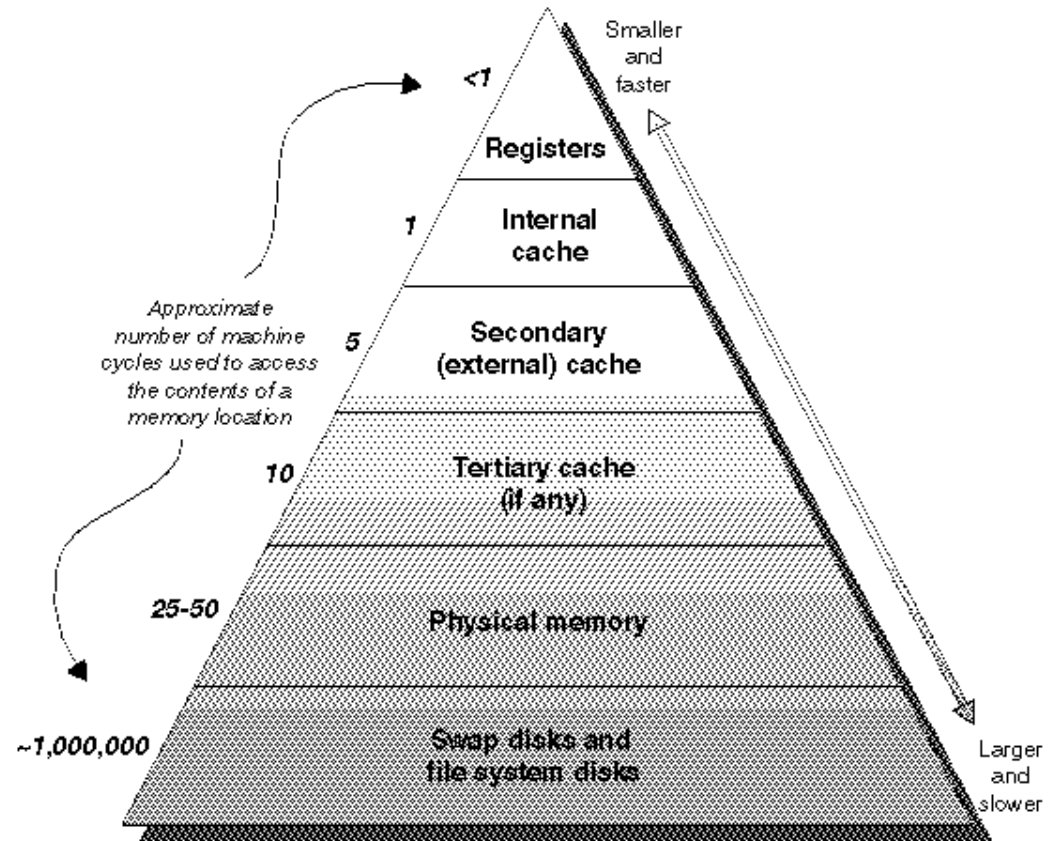
- fast memory is expensive and small
- slow memory is cheap and big

## Latency

- how long do I have to wait for the data?
- (cannot do anything while waiting)

## Throughput

- how many bytes/second. but not important if waiting.



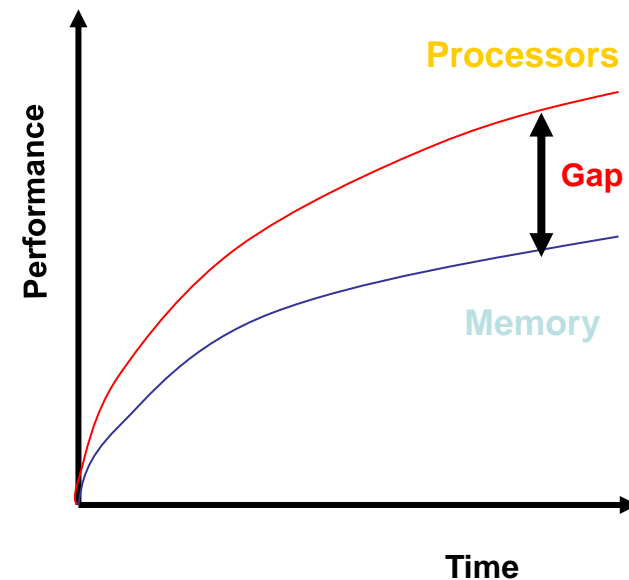
ZK-1083U-AI

**Total time = latency + (amount of data / throughput)**



## Memory access

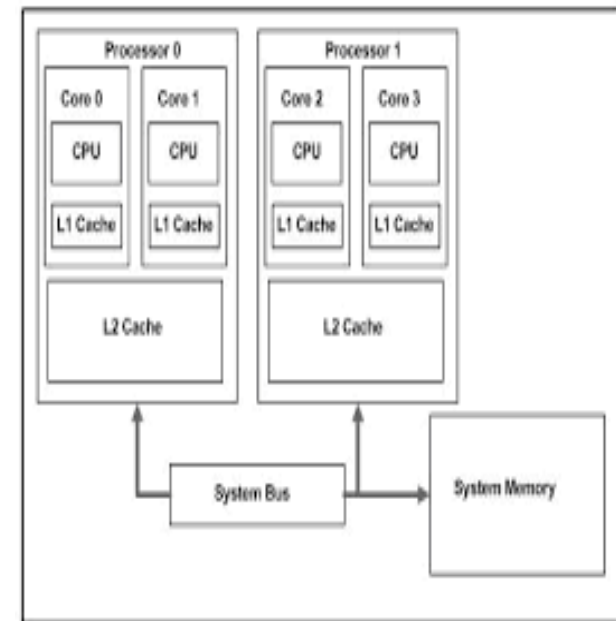
- Important problem for the performance of any computer is access to main memory. Fast processors are useless if memory access is slow!
- Over the years the difference in speed between processors and main memory has been growing.





## Cache Memory

- High speed, small size memory used as a buffer between the main memory and the processor. When used correctly, reduces the time spent waiting for data from main memory.
- Present as various “levels” (e.g. L1, L2, L3, etc) according to proximity to the functional units of the processor.
- Cache efficiency depends on the locality of the data references:
  - **Temporal locality** refers to the re-use of data within relatively small time frame.
  - **Spatial locality** refers to the use of data within close storage locations (e.g. one dimensional array).
- *Cache can contain Data, Instructions or both.*





## Cache Memory / 1

The code performance improves when the instructions that compose a heavy computational kernel (eg. a loop) fit into the cache

The same applies to the data, but in this case the work of optimization involves also the programmer and not just the system software.

### ***DEC Alpha 21164 (500 MHz):***

#### ***Memory access time***

Registers	2 ns
L1 On-chip	4 ns
L2 On-Chip	5 ns
L3 Off-Chip	30 ns
Memory	220 ns

### ***IBM SP Power 6 (4.7 GHz):***

#### ***Memory access time (in clock cycles)***

Registers	
L1: 2 x 64KB	< 5
L2: 2 x 4MB	22 cc
L3: 32 MB	160 cc
Memory 128 GB	400 cc



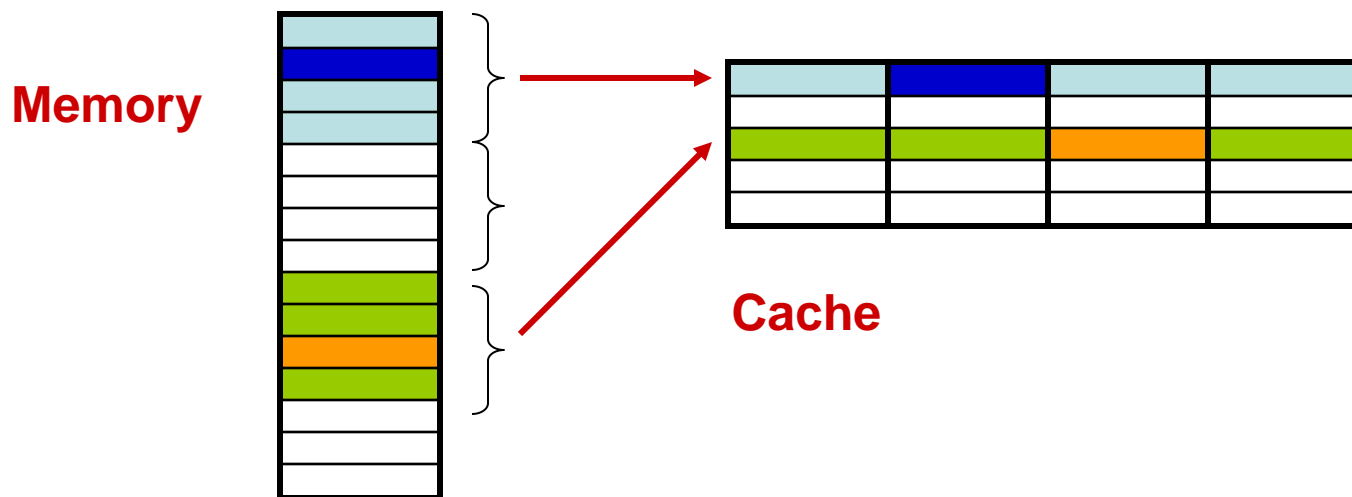


## Cache organisation

The cache is divided into slots of the same size (**lines**)

Each line contains k consecutive memory locations (ie 4 words).

When a data is required from memory, (if not already in the cache) the system loads from memory, the entire cache line that contains the data, overwriting the previous contents of the line.





## Aspects of parallelism

- It has been recognised for a long time that constant performance improvements cannot be obtained just by increasing factors such as processor clock speed - parallelism is needed.
- In HPC parallelism can be present at many levels:
  - Functional parallelism within the CPU.
  - Pipelining and vectorisation
  - Multi-processor and multi-core
  - Accelerators
  - Parallel I/O



## Multiple Functional Units

Arithmetic logic unit (**ALU**) executes the operations.

ALU is designed as a set of **independent functional units**, each in charge of executing a different arithmetic or logical operation,

- Add
- Multiply
- Divide
- Integer Add
- Integer Multiply
- Branch ....

The functional units can operate in parallel. This aspect represents the **first level of parallelism**. It is a parallelism **internal to the single CPU**.

The **compiler** analyses the different instructions and determine which operations can be done in parallel, without changing **the semantics of the program**.



## Pipelining

Is a technique where more instructions, belonging to a stream of sequential execution, overlap their execution

This technique improves the performance of the processor

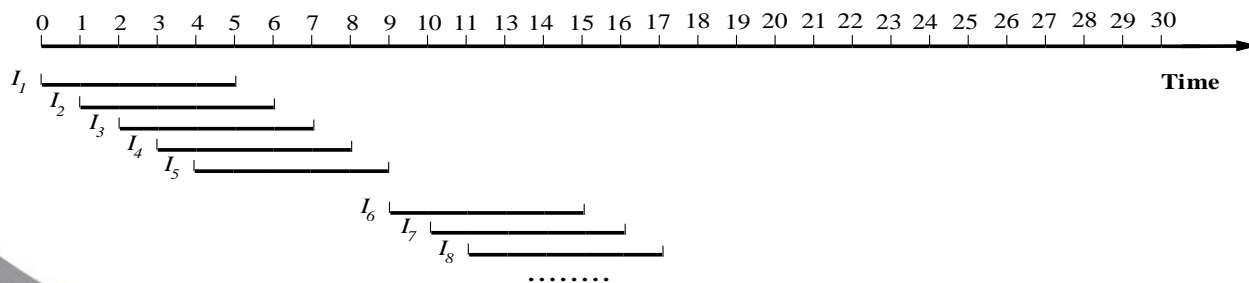
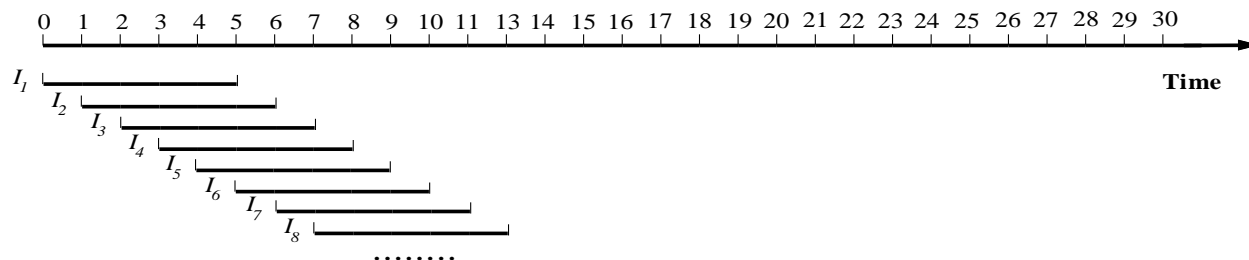
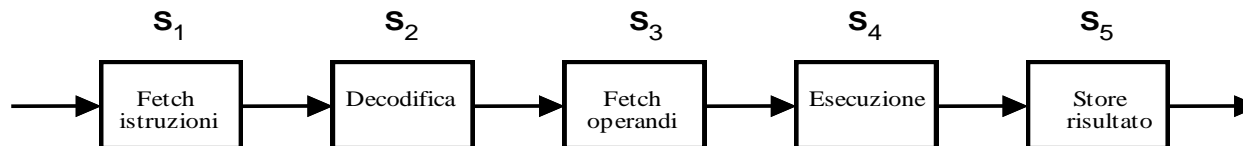
The concept of pipelining is similar to that of **assembly line** in a factory where in a flow line (**pipe**) of assembly stations the elements are assembled in a continuous flow.

All the assembly stations must operate at the **same processing speed**, otherwise the station slower becomes the **bottleneck** of the entire pipe.





# Instruction Pipelining





# Vector Computers

Vector computer architectures adopt a set of **vector instructions**, In conjunction with the scalar instruction set. The vector instructions operates on a set of **vector registers** each of which is able to contain more than one data element.

- **Cray vector systems** of the 80s and 90s
- **Cray C90**: 8 vector registers each with 128 elements at 64-bits
- Also the **current microprocessors** have a set of vector registers and a set of vector instructions

The vector instructions implement a particular operation to be performed on a given set of operands called **vector**.

Functional units when executing vector instructions exploit pipelining to perform the same operation on all data operands stored on vector registers.

Data transfer to and from the memory is done through **load** and **store** operations operating on vector registers.





## CPU Vector units

- Vectorisation performed by dedicated hardware on chip.
- Compiler generates vector instructions, when it can, from programmer's code.
- Important optimisation which can lead to 4x, 8x speedups according "size" of vector unit (e.g. 256 bit).

### Scalar Processing

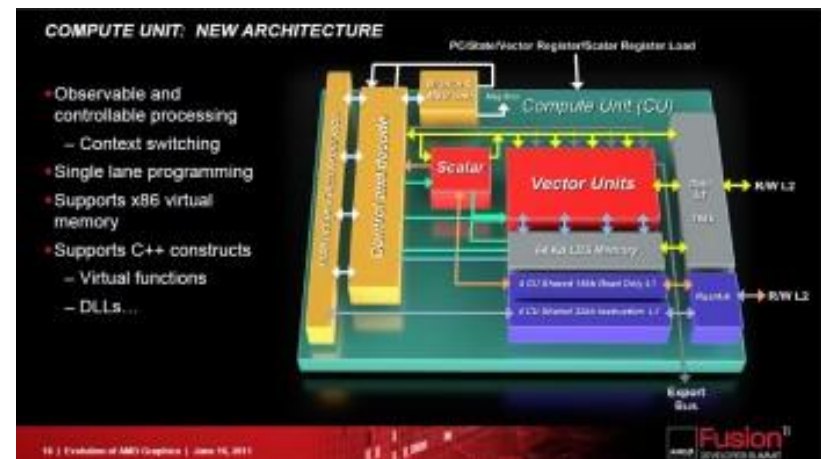
$$\begin{aligned} a_1 + b_1 &= c_1 \\ a_2 + b_2 &= c_2 \\ a_3 + b_3 &= c_3 \\ &\vdots \\ a_n + b_n &= c_n \end{aligned}$$

```
for i = 1 to n
  c[i] = a[i] + b[i]
end
```

### Vector Processing

$$\begin{aligned} a_1 & \quad b_1 & \quad c_1 \\ a_2 & \quad b_2 & \quad c_2 \\ a_3 & \quad b_3 & \quad c_3 \\ \vdots & \quad \vdots & \quad \vdots \\ a_n & \quad b_n & \quad c_n \end{aligned} \quad + \quad =$$

```
c[1:n] = a[1:n] + b[1:n]
```





# Flynn Taxonomy

M. J. Flynn

*Very high speed computing systems*, proceedings of the IEEE (1966).

*Some computer organizations and their effectiveness*, IEEE Transaction on Computers.(1972).

*"The multiplicity is taken as the maximum possible number of simultaneous operations (instructions) or operands (data) being in the same phase of execution at the most constrained component of the organization"*

A computer architecture is categorized by the multiplicity of hardware used to manipulate **streams of instructions** (sequence of instructions executed by the computer) and **streams of data** (sequence of data used to execute a stream of instructions).

**SI** Single Instruction stream

**SD** Single Data stream

**MI** Multiple Instruction stream

**MD** Multiple Data stream

4 possible combinations : **SISD, SIMD, MISD, MIMD**





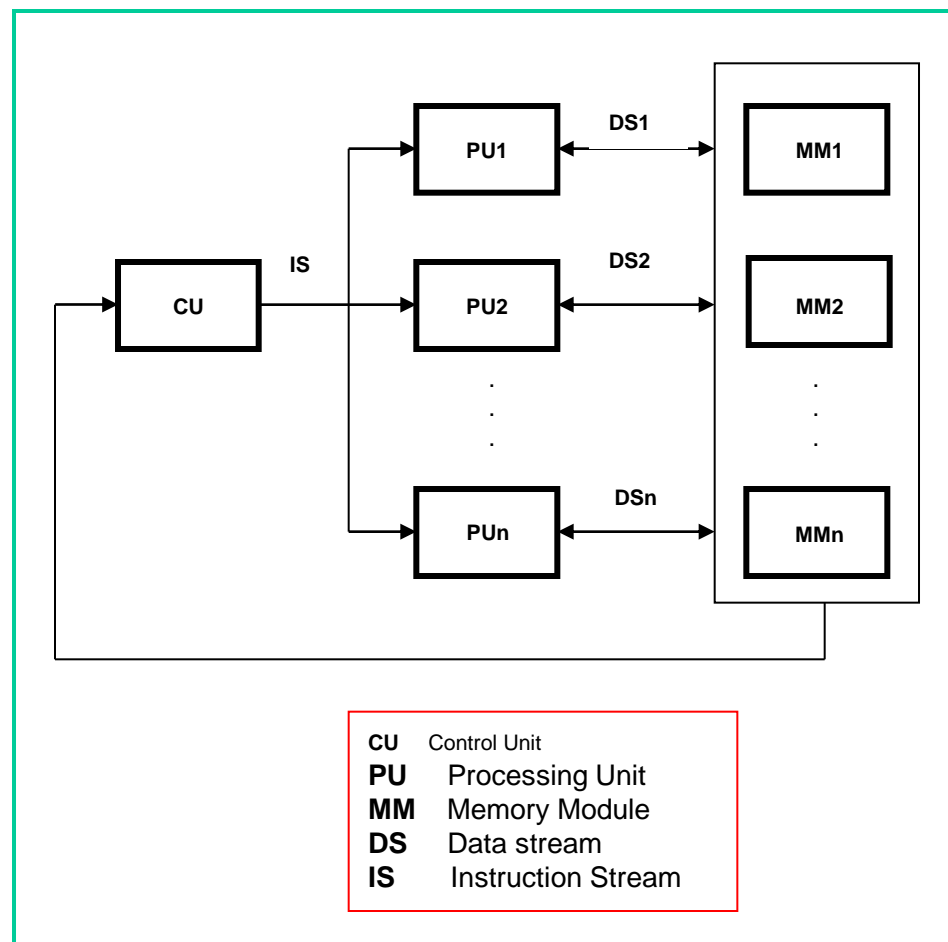
# SIMD Systems

## Synchronous parallelism

SIMD systems presents a single control unit

A single instruction operates simultaneously on multiple data.

Array processor and vector systems fall in this class





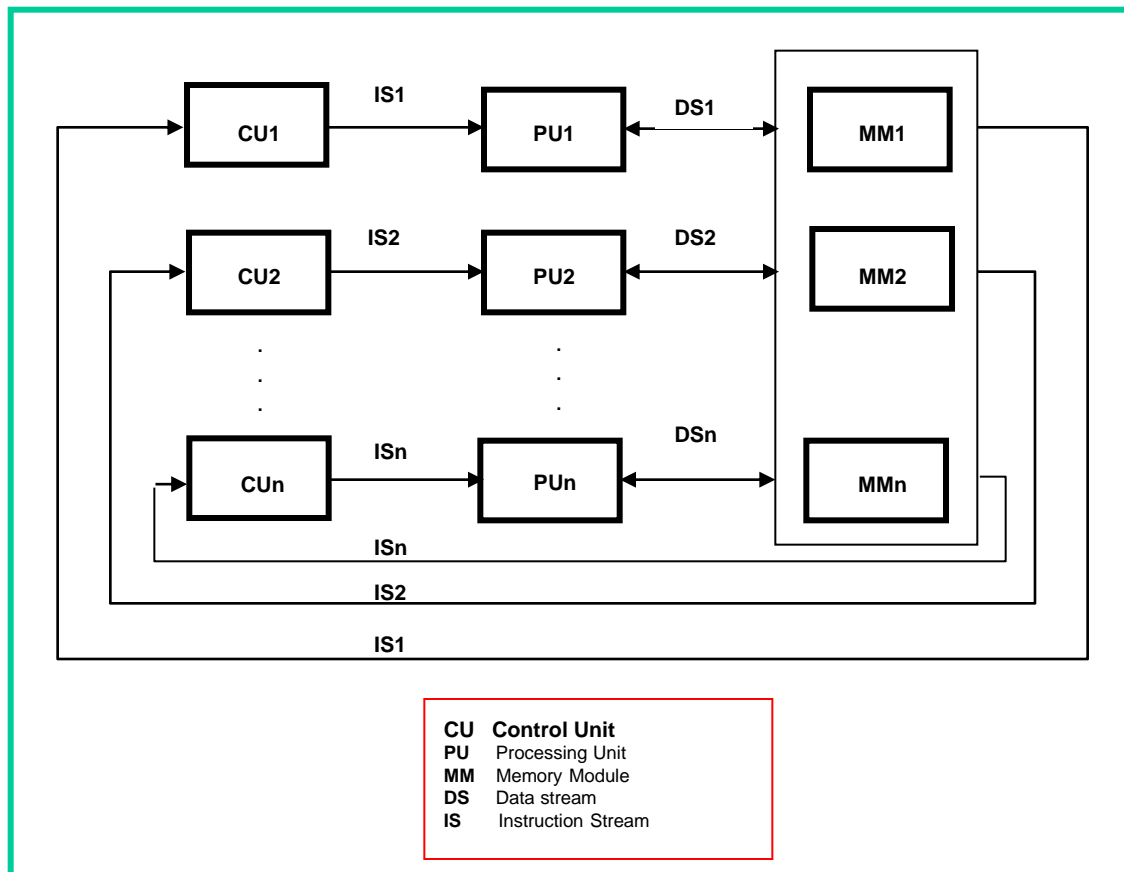
# MIMD Systems

## Asynchronous parallelism

Multiple processors  
execute different  
instructions operating on  
different data.

Represents the  
multiprocessor version of  
the SIMD class.

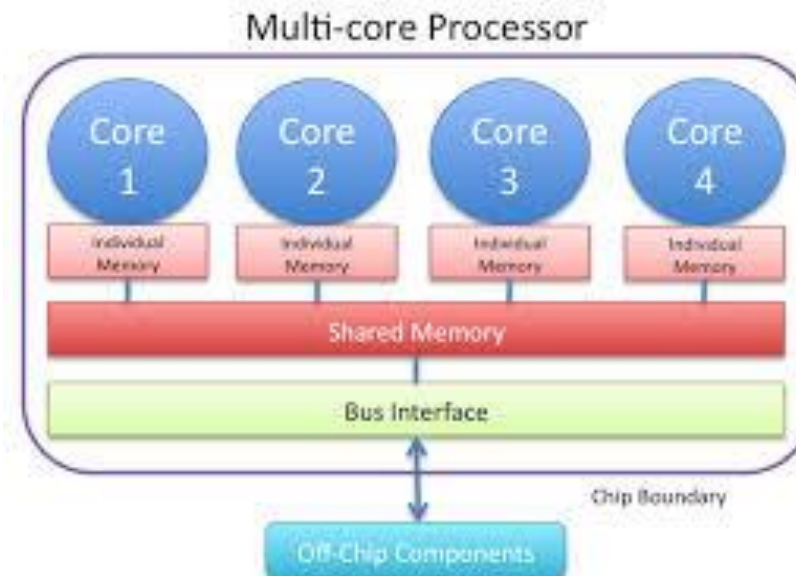
Wide class ranging from  
multi-core systems to  
large MPP systems.





## Multi-core processors

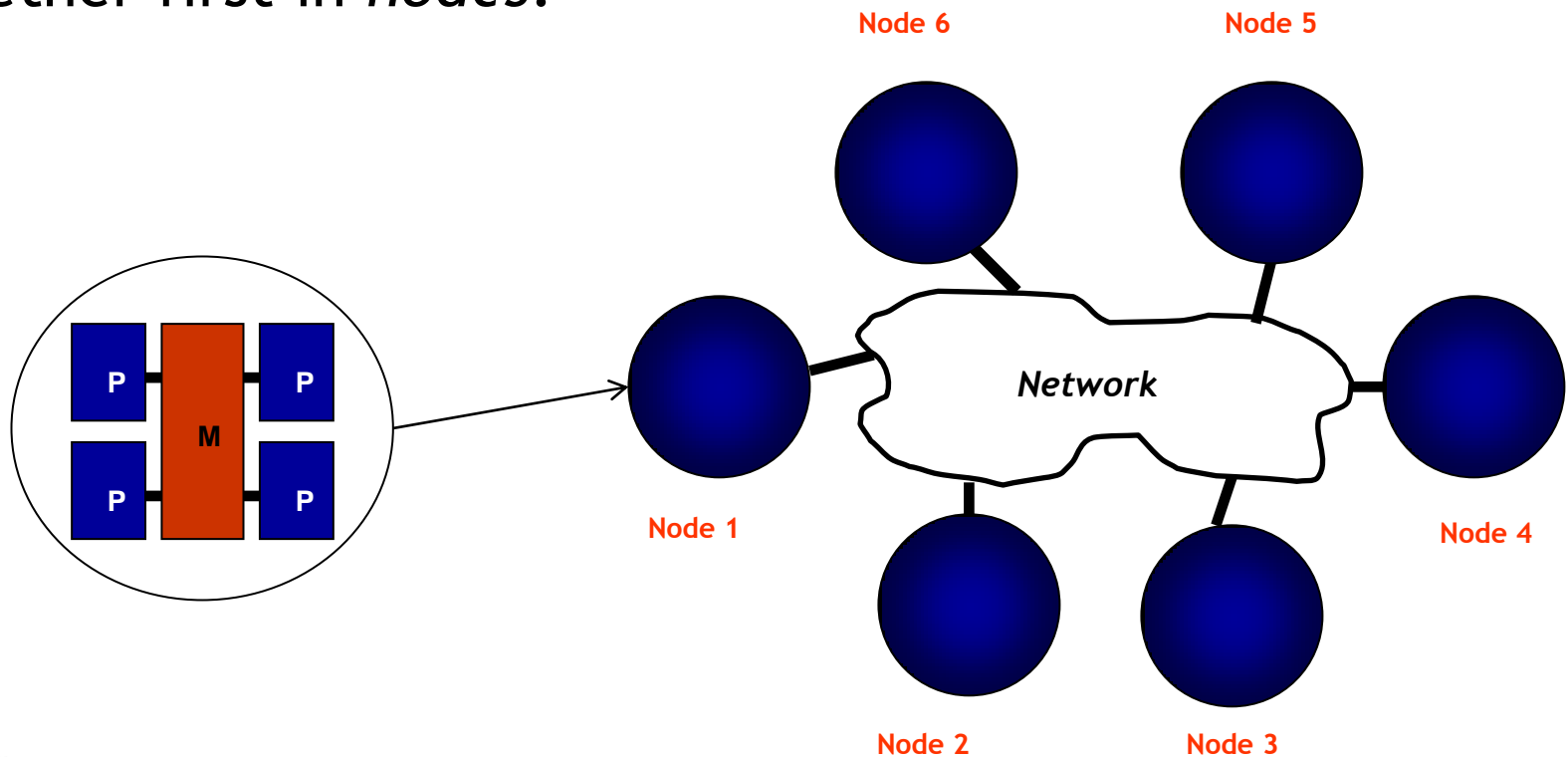
- Because of power, heat dissipation, etc increasing tendency to actually *lower* clock frequency but pack more computing cores onto a chip.
- These cores will share some resources, e.g. memory, network, disk, etc but are still capable of independent calculations





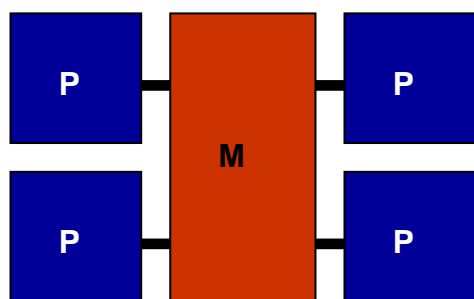
## Multi-processor systems

- One way to increase performance is to link (multi-core) processors together in *clusters*, perhaps grouped together first in *nodes*.

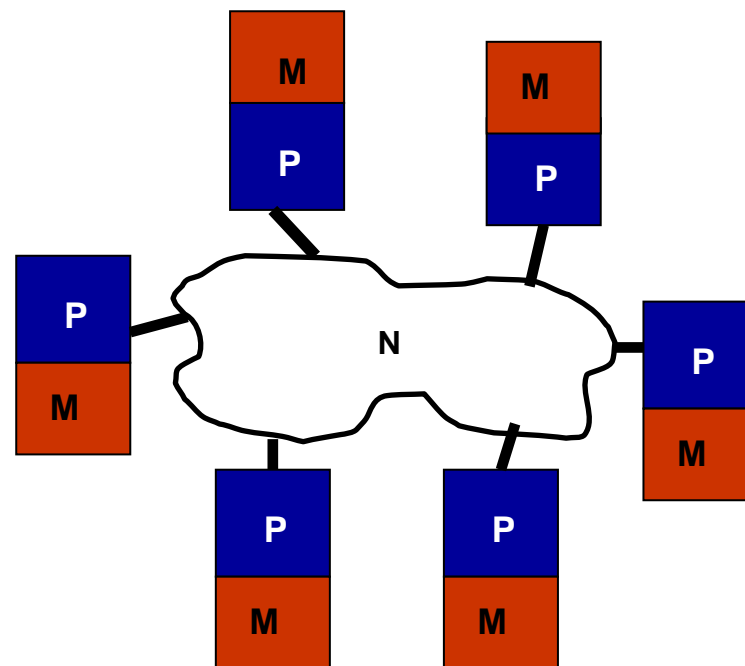




## Classification based on the Memory



Shared Memory System



Distributed Memory System



## Shared memory systems

All the processors (cores) share the main memory.

The memory can be addressed globally by all the processors of the system

*Uniform Memory Access (UMA)* model  $\Leftrightarrow$  **SMP: Symmetric Multi Processors**

The memory access is **uniform**: the processors present the same access time to reference any of the memory locations.

Processor-Memory interconnection via **common bus**, **crossbar switch**, or **multistage networks**.

Each processor can provide local caches,

Shared memory systems can not support a high number of processors

Currently in Europe, there are very few entire systems which are classified as shared memory, but sub-components (e.g nodes) may be.



## Distributed memory systems

The memory is physically distributed among the processors (**local memory**).  
Each processor can access directly only to its own local memory

- ***NO-Remote Memory Access (NORMA) model***

Communication among different processors occurs via a specific communication protocol (**message passing**).

The messages are routed on the interconnection network

In general distributed memory systems can scale-up from a small number of processors  $O(10^2)$  to huge numbers of processors  $O(10^6)$  but the power of the single cores is not too high, to reduce global costs and power consumption but power is not too high, called processing nodes.

The performance of the system are influenced by:

- **Power of the node**
- **Topology of the interconnection network**



## NUMA systems

### *Non Uniform Memory Access (NUMA) model*

Memory is **physically distributed among all the processors** (each processor has its own local memory) but the collection of the different local memories forms a global address space accessible by all the processors

Hw support to ensure that each processor can access directly the memory of all the processors

The **time** each processor needs to access the memory **is not uniform**:

- access time is faster if the processor accesses its own local memory;
- when accessing the memory of the remote processors **delay** occurs, due to the interconnection network crossing.





## Interconnection network

It is the set of links (**cables**) that define how the different processors of a parallel computer are connected between themselves and with the memory unit. The time required to transfer the data depends on the type of interconnection. The transfer time is called the **communication time**.

### Features of an interconnection network:

- **Bandwidth**: identifies the amount of data that can be sent per unit time on the network. **Bandwidth must be maximized**.
- **Latency**: identifies the time required to route a message between two processors. Latency is defined also as the time needed to transfer a message of length zero. **Latency must be minimized**.

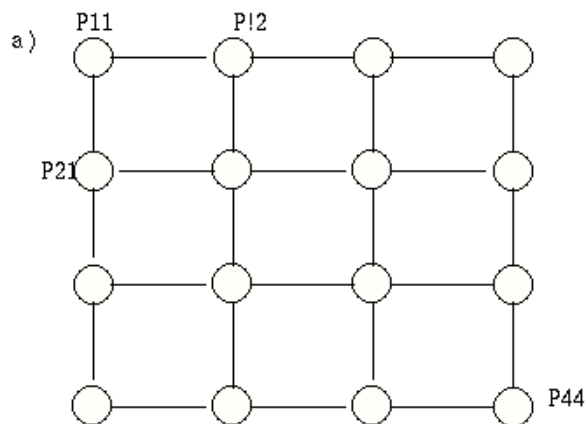
Other points to consider:

- **Cost**
- **Scalability**
- **Reliability**
- **Diameter**
- **Degree**



## Example networks

EXAMPLE

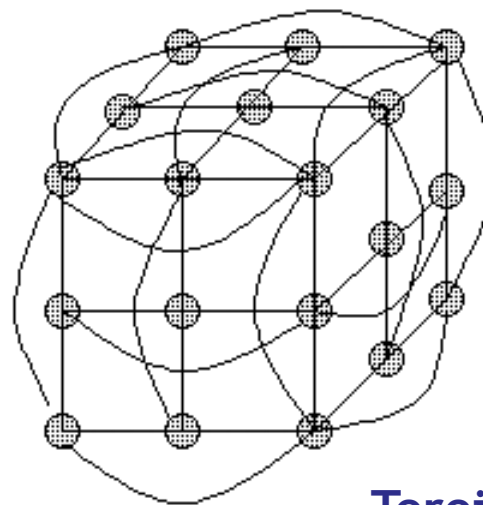


2D mesh of width 4 with  
no wraparound connections  
on edge or corner nodes

corner nodes have degree 2  
edge nodes have degree 3

### MESH Topology

c)



$k = 3$   $w = 3$

i.e.  $3^3 = 27$  nodes

with wraparound connections

all nodes have degree 6 ( $2k$ )

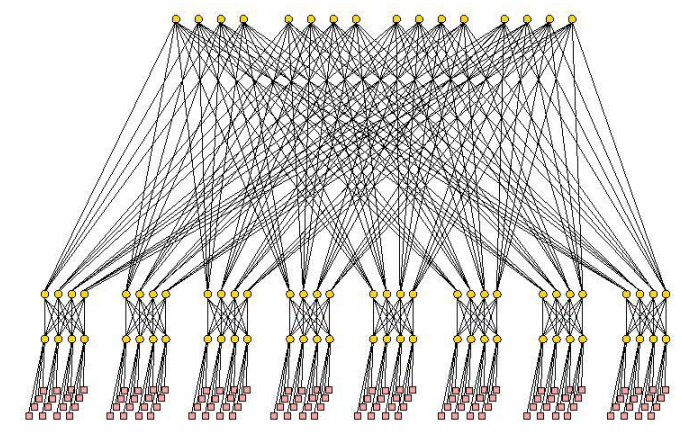
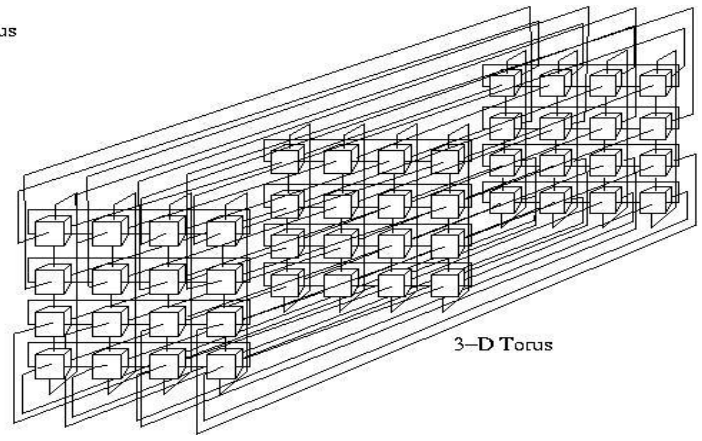
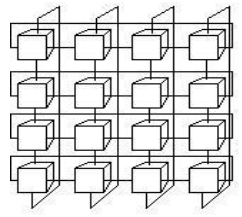
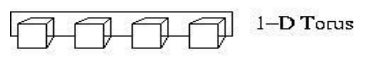
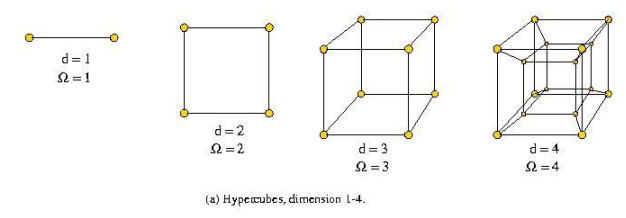
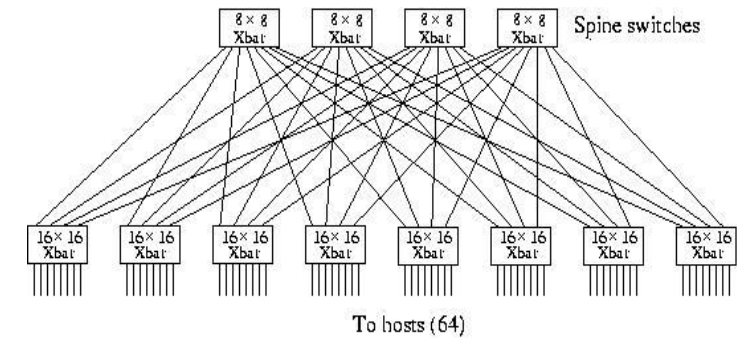
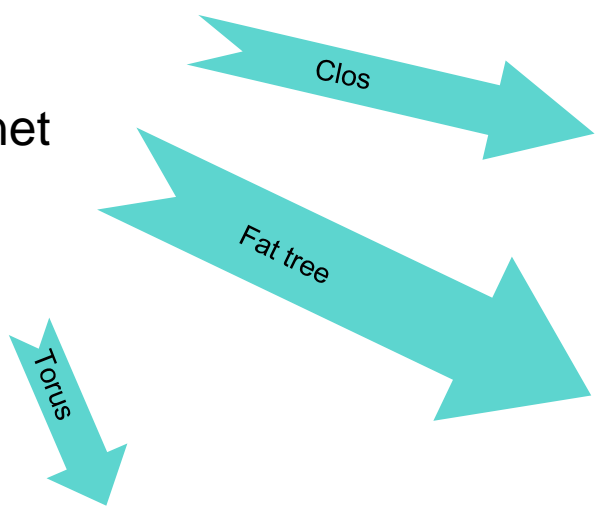
### Toroidal Topology

Some variations of the mesh model have **wrap-around type** connections between the nodes to the edges of the mesh (**torus topology**). The **Cray T3E** adopts a 3D torus topology **IBM BG/Q** adopts a 5D torus topology



# Commodity Interconnects

Gig Ethernet  
Myrinet  
Infiniband  
QsNet  
SCI



(b) A 128-way fat tree.



## Recent HPC Trends - IBM BlueGene

- Multi-core, multi-processor clusters limited by factors such as physical space and particularly energy consumption and cooling.
- One approach is to lower even further single processor power but increase massively the number of cores.
- In the IBM Bluegene range hundreds of thousands of lower power PowerPC cores are connected by a fast network.



## Recent HPC trends - IBM Bluegene

- In the IBM BG/Q (Cineca), for example, there are 168K cores in total, where 1node =16 cores+16Gb.
- Suitable for very highly parallel applications (>2048cores) but many codes don't well.
- The Bluegene range is not being continued by IBM





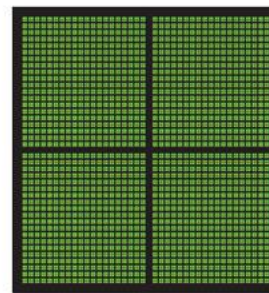
## Recent HPC Trends - accelerators/GPUs

- Co-processors or accelerators have been around for a while but it was only when Nvidia released CUDA did GPUs become interesting for HPC (2006).
- GPGPUs or simply GPUs work in a different way to conventional CPUs. Emphasis on *stream processing*.
- Acceleration can be significant but depends on application.

Features	Tesla K80 <sup>1</sup>
GPU	2x Kepler GK210
Peak double precision floating point performance	<b>2.91 Tflops</b> (GPU Boost Clocks) 1.87 Tflops (Base Clocks)
Peak single precision floating point performance	8.74 Tflops (GPU Boost Clocks) 5.6 Tflops (Base Clocks)
Memory bandwidth (ECC off) <sup>2</sup>	480 GB/sec (240 GB/sec per GPU)
Memory size (GDDR5)	24 GB (12GB per GPU)
<u>CUDA</u> cores	4992 ( 2496 per GPU)



CPU  
MULTIPLE CORES

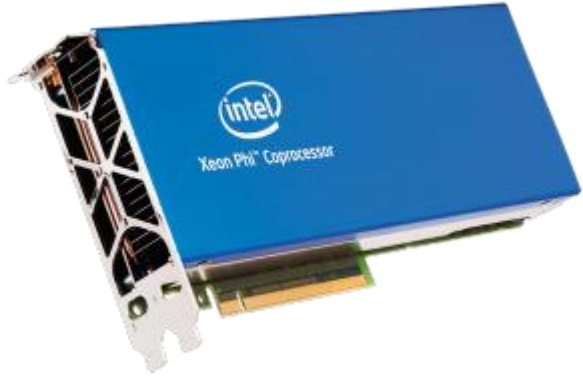


GPU  
THOUSANDS OF CORES





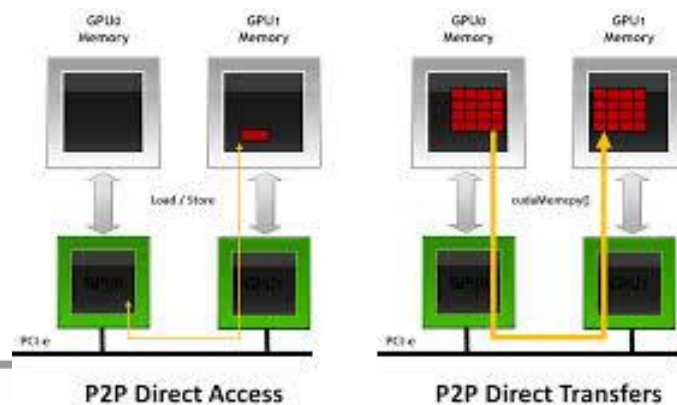
## Recent HPC Trends- Accelerators/Intel Xeon Phi (MIC)

- Also an accelerator but more similar to a conventional multicore CPU.
  - Current version, Knight's Corner (KNC) has 57-61 1.0-1.2 GHz cores, 8-16GB RAM. 512 bit vector unit.
  - Cores connected in a ring topology and MPI possible.
- 
- An Intel Xeon Phi Coprocessor (MIC) is shown, a blue printed circuit board with a silver metal bracket on the left side. The Intel logo and the text "Xeon Phi Coprocessor" are visible on the blue surface.
- No need to write CUDA or OpenCL as Intel compilers will compile Fortran or C code for the MIC.
  - 1-2 Tflops, according to model.



## Recent HPC Trends - Accelerators

- GPUs and MICs are attracting interest in HPC because of high performance and efficiency (i.e. Flops/watt).
- Currently, they need to be attached to host CPUs via the PCIe bus (a standard PC-like connection).
- Both device families have limitations:
  - low device memory
  - slow transfer rate via PCIe link
  - difficulty in programming (particularly CUDA).
  - speedup is highly application and data dependent.
- But future models are likely to be standalone models (e.g Knight's Landing) and with faster connections (Nvlink).







# HPC Trends - TOP500

## TOP 10 Sites for November 2014

For more information about the sites and systems in the list, click on the links or view the [complete list](#).

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	<b>Tianhe-2 (MilkyWay-2)</b> - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	<b>Titan</b> - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	<b>Sequoia</b> - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	<b>Mira</b> - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945
6	Swiss National Supercomputing Centre (CSCS) Switzerland	<b>Piz Daint</b> - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115,984	6,271.0	7,788.9	2,325
7	Texas Advanced Computing Center/Univ. of Texas United States	<b>Stampede</b> - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462,462	5,168.1	8,520.1	4,510
8	Forschungszentrum Juelich (FZJ) Germany	<b>JUQUEEN</b> - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	458,752	5,008.9	5,872.0	2,301

- List of the most powerful supercomputers in the world, published twice a year.
- Performance measured by the Linpack benchmark.



# Top 500: some facts

- 1976** Cray 1 installed at Los Alamos: peak performance 160 MegaFlop/s ( $10^6$  flop/s)
- 1993** (1° Edition Top 500) N. 1 59.7 GFlop/s ( $10^{12}$  flop/s)
- 1997** Teraflop/s barrier ( $10^{12}$  flop/s)
- 2008** Petaflop/s ( $10^{15}$  flop/s): **Roadrunner** (LANL) Rmax 1026 Gflop/s, Rpeak 1375 Gflop/s  
hybrid system: 6562 processors dual-core AMD Opteron accelerated with 12240 IBM Cell processors  
(98 TByte di RAM)
- 2011** 11.2 Petaflop/s : K computer (SPARC64 VIIIfx 2.0GHz, Tofu interconnect) RIKEN Japan
- 2015** 34 Petaflop/s: TIANHE-2 (MILKYWAY-2), Intel Xeon /Xeon PHI, Guangzhou China.
- 62% of the systems on the top500 use processors with six or more cores
  - 39 systems use GPUs as accelerators (35 NVIDIA , 2 Cell , 2 ATI Radeon)



## Real HPC Crisis is with Software

- A supercomputer application is usually much more long-lived than hardware
  - Hardware typically 4-5 years
  - FORTRAN and C still main programming models (hasn't changed much since the 1970s)
- Porting applications to Petaflop systems is a major challenge.
  - New parallelization strategies are needed.
  - Not just program code - some datasets cannot scale to thousands of cores.
  - Also using supercomputer systems hasn't changed. Users are still expected to know UNIX and batch systems



## The European perspective - PRACE



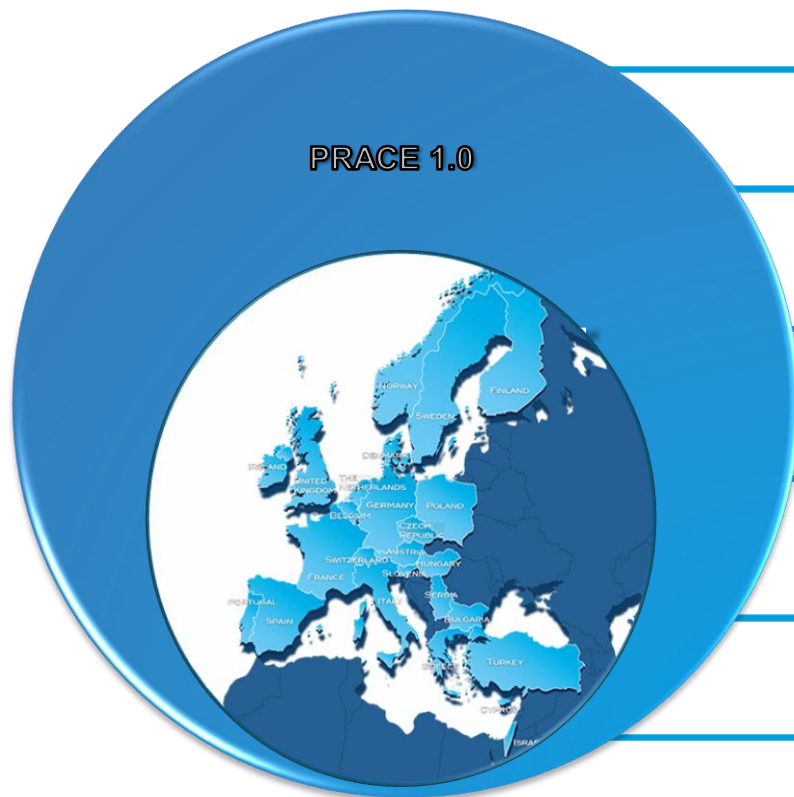
- Partnership for Advanced Computing in Europe
- PRACE is part of the ESFRI roadmap and has the aim of creating a European Research Infrastructure providing world class systems and services and coordinating their use throughout Europe.
- It covers both hardware at the multi petaflop/s level and also very demanding software (parallel applications) to exploit these systems.



<http://www.prace-project.eu/>



# The HPC European e-infrastructure (ESFRI)



25 members,  
AISBL since  
2010

530 M€ for 2010-2015 (inc 70M€  
from UE)

6 supercomputers in 4 hosting  
countries, different architectures

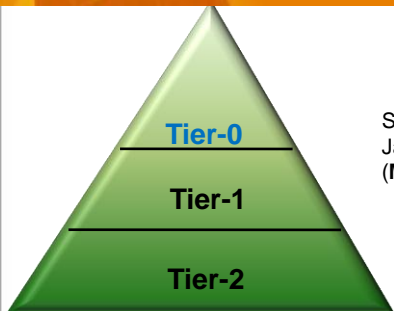
research and industrial access (open  
R&D) for all disciplines **based on  
excellence in science, free of charge**

Nearly 15 Pflop/s

5 billion hours  
granted since 2010



# PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE



Sixth production system available by January 2013: **1 Petaflop/s** IBM (MareNostrum) at BSC.



Upgrade: **5.87 Petaflop/s** IBM Blue Gene/Q (JUQUEEN)

First production system available: **1 Petaflop/s** IBM BlueGene/P (JUGENE) at GCS (Gauss Centre for Supercomputing) partner FZJ (Forschungszentrum Jülich)

Fifth production system available by August 2012: **2 Petaflop/s** IBM BG/Q (FERMI) at CINECA.



Second production system available: Bull Bull **CURIE** at GENCI partner CEA. Full capacity of **1.8 Petaflop/s** reached by late 2011.



Third production system available by the end of 2011: **1 Petaflop/s** Cray (HERMIT) at GCS partner HLRS Stuttgart.



Fourth production system available by mid 2012: **3 Petaflop/s** IBM (SuperMUC) at GCS partner LRZ (Leibniz-Rechenzentrum).



## FERMI@CINECA



- **Architecture:** 10 BGQ Frames
- **Model:** IBM-BG/Q
- **Processor type:** IBM PowerA2 @1.6 GHz
- **Computing Cores:** 163840
- **Computing Nodes:** 10240
- **RAM:** 1GByte / core (163 PByte total)
- **Internal Network:** 5D Torus
- **Disk Space:** 2PByte of scratch space
- **Peak Performance:** 2PFlop/s
- **N. 12 in Top 500 rank** (June 2013)
- National and PRACE Tier-0 calls



## Galileo@CINECA - installed Jan 2015

IBM Cluster linux

**Processor type:** 2 eight-cores Intel Xeon Haswell

X 2630 @ 2.4 GHz, 12MB Cache

**N. of nodes / cores:** 524 / 8384

**RAM:** 128 GB/Compute node

**Internal Network:** Infiniband with 4x QDR switches

**Accelerators:** 768 Intel Xeon PHI 7120p

**Peak performance:** 1 PFlop

**National and PRACE Tier-1 calls**







## PICO@Cineca

- Cluster dedicated to visualization and BigData applications.
- The **storage area** is composed by high throughput disks (based on [GSS](#) technology) for a total amount of about 4 PB, connected with a large capacity tape library for a total actual amount of 12 PByte (expandible to 16 PByte).

Model: IBM NeXtScale

	Total Nodes	CPU	Cores per Nodes	Memory (RAM)	Notes
Compute/login node	66	Intel Xeon E5 2670 v2 @2.5Ghz	20	128 GB	
Visualization node	2	Intel Xeon E5 2670 v2 @ 2.5Ghz	20	128 GB	2 GPU Nvidia K40
Big Mem node	2	Intel Xeon E5 2650 v2 @ 2.6 Ghz	16	512 GB	1 GPU Nvidia K20
BigInsight node	4	Intel Xeon E5 2650 v2 @ 2.6 Ghz	16	64 GB	32TB of local disk



## Eurora: EUROpean many integrated cORe Architecture

- Hybrid cluster based on the evolution of the AURORA architecture by Eurotech
- 64 nodes Intel Sandy Bridge dual socket, (1024 cores in total)
  - ❑ 32 nodes at 2.1 GHz and
  - ❑ 32 nodes at 3.1 GHz).
- 16 GByte DDR3 RAM, 160 GByte SSD, 1 FPGA Altera Stratix V. per Node
- Interconnection networks: Infiniband QDR and 3D Torus
- Hot water cooling
- The system is equipped with:
  - **64 MIC processors** (2 per node on 32 nodes)
  - **64 NVIDIA K20** accelerators (2 per node on 32 nodes)
- Peak performance (K20 accelerated) 175,7 Tflop/s
- **N. 467 in Top 500 rank (June 2013),**
- **N. 1 in Green500 rank (June 2013)**

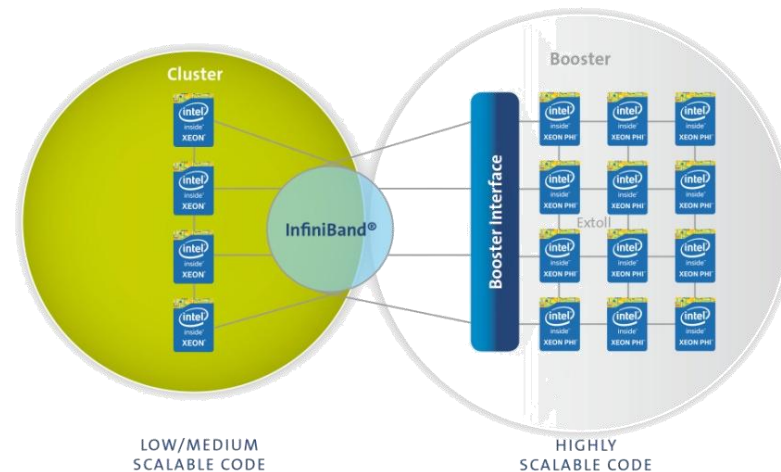


THE GREEN  
500™



## The Road to Exascale- DEEP and DEEP-ER (Dynamical Exascale Entry Platform)

- DEEP is an Exascale project funded by the EU 7th framework programme. The main goal is to develop a novel, Exascale-enabling supercomputing platform.



**DEEP-ER**

The hardware will be based on a conventional Xeon cluster linked to a so-called “Booster” consisting of Xeon Phi nodes. The idea is that highly scalable portions of the application will run on the Booster, while the remainder of the code runs on the traditional cluster. Porting scientific applications to run on the prototype is a major objective.



## The Road to Exascale- Mont Blanc

# MONT-BLANC

EUROPEAN APPROACH TOWARDS ENERGY EFFICIENT HIGH PERFORMANCE



- Emphasis on Energy Efficiency by constructing a machine from ARM chips, more usually found in mobile or embedded devices.
- This project is coordinated by the [Barcelona Supercomputing Center](#) (BSC) and has a budget of over 14 million, including over 8 million Euros funded by the European Commission.



## Summary and Trends - Hardware

Reaching physical limits of transistor densities and increasing clock frequencies further is too expensive and difficult (e.g. energy consumption, heat dissipation).

Parallelism is only solution in HPC but the BlueGene road is no longer being pursued. Hybrids with accelerators such as GPUs or Xeon PHIs becoming the norm.

Accelerator technologies advancing to remove limits associated with, for example, the PCIe bus (e.g. Nvidia NVLINK or Intel KNL).

A range of novel architectures being explored (e.g. Mont Blanc, DEEP) and technologies in many areas (NVRAM, SSD, NAM,...).

Monitoring systems for energy efficiency are becoming more sophisticated. Some schedulers now report energy consumed.



## Summary and Trends - Software

As usual software lags behind hardware but must learn to exploit accelerators.

Reluctance by some software developers to learn new languages such as CUDA or OpenCL is driving interest in compiler-directive languages such as OpenAcc or OpenMP (4.x) (despite lower efficiency.)

Continued investment in efficient filesystems, checkpointing, resilience, parallel I/O, etc.



## Bit and Byte on Information Society

### Gigabyte [ 1,000,000,000 bytes OR $10^9$ bytes ]

- 500 megabytes: A CD-ROM
- 100 megabytes: 1 meter of shelved books
- 10 megabytes: A minute of high-fidelity sound
- 5 megabytes: The complete works of Shakespeare
- 2 megabytes: A high-resolution photograph
- 1 megabyte: A small novel OR a 3.5-inch floppy disk

### Megabyte [ 1,000,000 bytes OR $10^6$ bytes ]

- 200 kilobytes: A box of punched cards
- 100 kilobytes: A low-resolution photograph
- 50 kilobytes: A compressed document image page
- 10 kilobytes: An encyclopaedia page
- 2 kilobytes: A typewritten page
- 1 kilobyte: A very short story

### Kilobyte [ 1,000 bytes OR $10^3$ bytes ]

- 100 bytes: A telegram or a punched card
- 10 bytes: A single word
- 1 byte: A single character

Byte [ 8 bits ]

Bit [ A binary digit - either 0 or 1 ]



## Bits and Bytes on Information Society/1

**Zettabyte [ 1,000,000,000,000,000,000 bytes OR  $10^{21}$  bytes ]**

5 exabytes: All words ever spoken by human beings

2 exabytes: Total volume of information generated worldwide annually

**Exabyte [ 1,000,000,000,000,000 bytes OR  $10^{18}$  bytes ]**

200 petabytes: All printed material

8 petabytes: All information available on the Web

2 petabytes: All U.S. academic research libraries

1 petabyte: 3 years of Earth Observing System (EOS) data (2001)

**Petabyte [ 1,000,000,000,000,000 bytes OR  $10^{15}$  bytes ]**

400 terabytes: National Climatic Data Center (NOAA) database

50 terabytes: The contents of a large mass storage system

10 terabytes: The printed collection of the U.S. Library of Congress

2 terabytes: An academic research library

1 terabyte: 50,000 trees made into paper and printed OR daily rate of EOS data (1998)

**Terabyte [ 1,000,000,000,000 bytes OR  $10^{12}$  bytes ]**

500 gigabytes: The biggest FTP site

100 gigabytes: A floor of academic journals

50 gigabytes: A floor of books

2 gigabytes: 1 movie on a Digital Video Disk (DVD)

1 gigabyte: A pickup truck filled with paper



# Finally just to show we have come a long way



8Mb

**STORAGE**



2015



128Gb

1975



400 Mflops

**PERFORMANCE**



2015

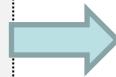


173 Gflops (GPU)

1970

```
C
PROGRAM HELLO
REAL A(10,10)
DO 50 I=1,10
  PRINT *, 'Hello'
50 CONTINUE

CALL DGEMM(N,10,I,J,A)
```



2015

```
C
PROGRAM HELLO
REAL A(10,10)
DO 50 I=1,10
  PRINT *, 'Hello'
50 CONTINUE

CALL DGEMM(N,10,I,J,A)
```

**SOFTWARE**