Introduction to High Performance Computing

Andrew Emerson

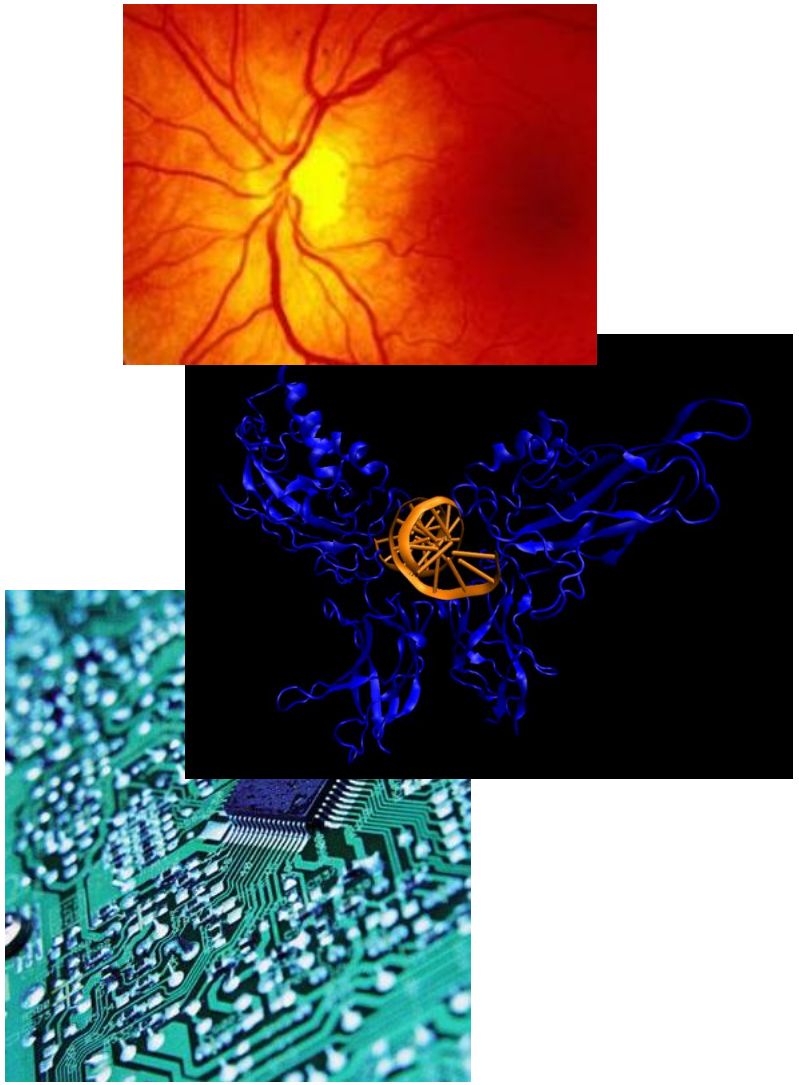
Cineca |www.hpc.cineca.it

PART 1. Supercomputers and how they work.

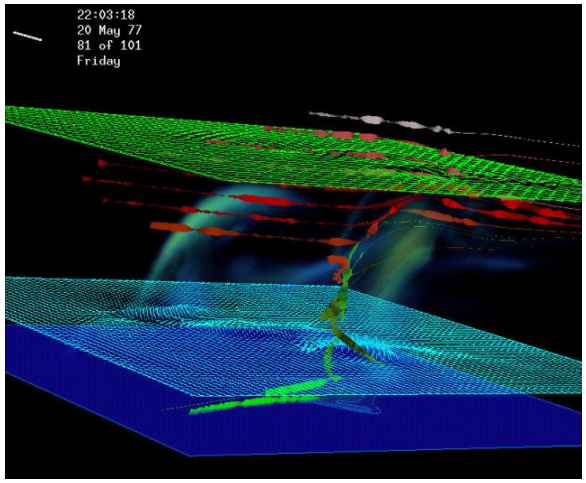
Computational Science

"Computational science is concerned with constructing mathematical models and quantitative analysis techniques and using computers to analyze and solve scientific problems. In practical use, it is typically the application of computer simulation and other forms of computation from numerical analysis and theoretical computer science to problems in various scientific disciplines." (Wikipedia)

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



Computational Sciences

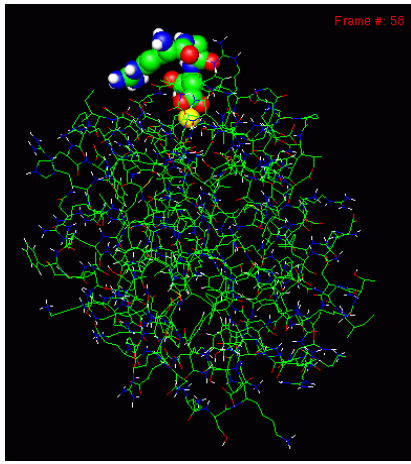


Identify the scientific disciplines that use mathematical models and computing systems to analyze and solve scientific problems.

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 which are:

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



Computational Sciences - pioneers

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

Ada Lovelace (1815-1852), *regarded as the first computer programmer.*

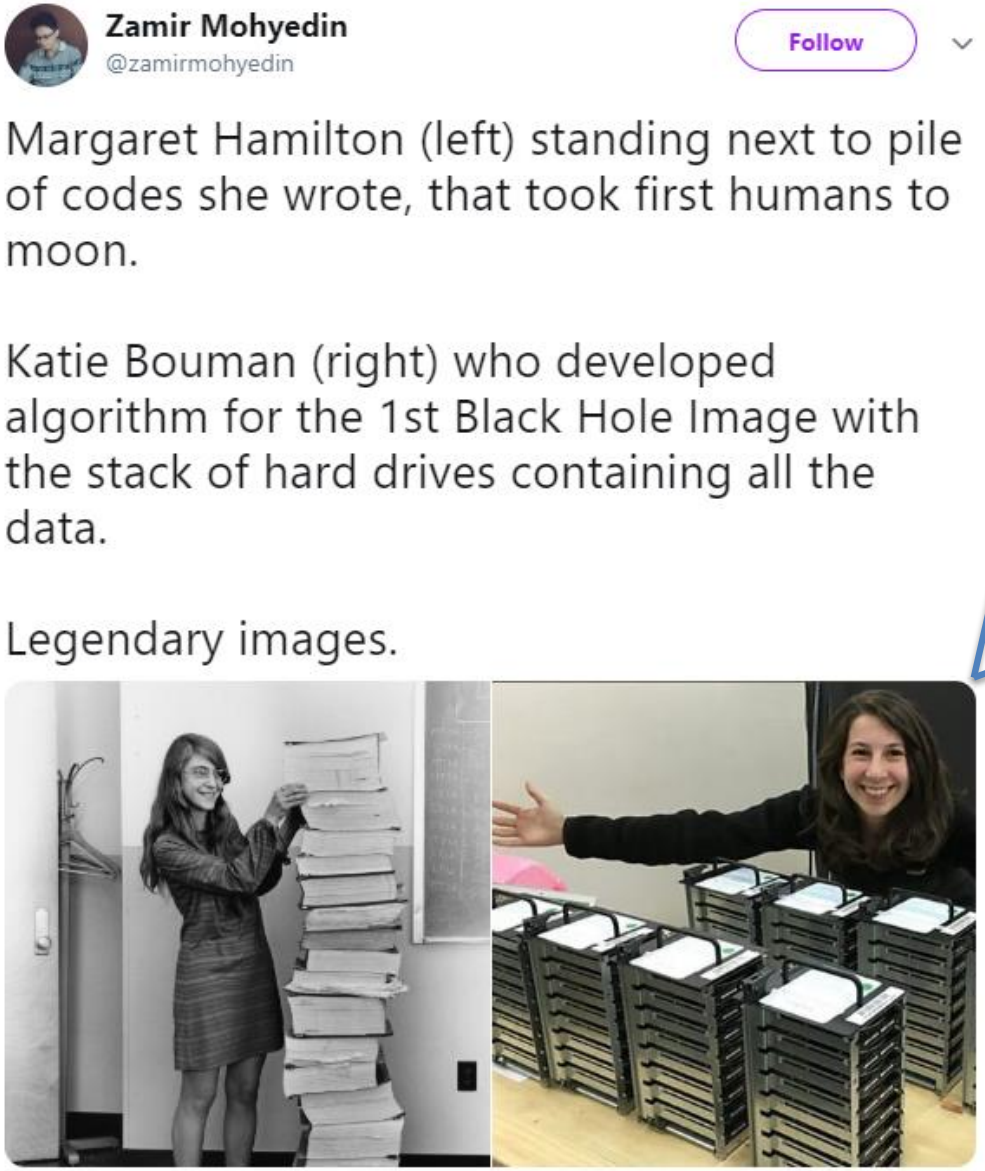
John Von Neumann, *1920s*

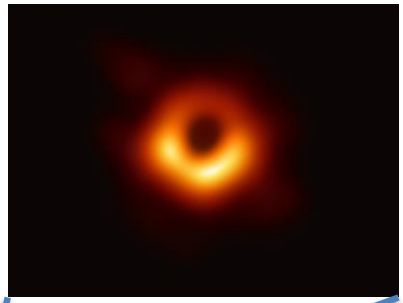
polymath and computer pioneer

Alan Turing (1912-1954)

The father of theoretical computer science

Lesser known pioneers





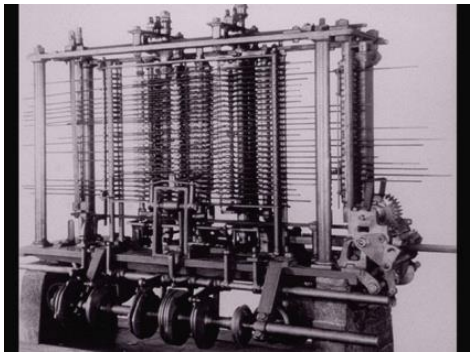
Katherine Johnson

African-American mathematician whose calculations of orbital mechanics as a NASA employee were critical to the success of the first and subsequent U.S. manned spaceflight

Supercomputers

Supercomputers are defined as the most powerful computers available in a given period of time.

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

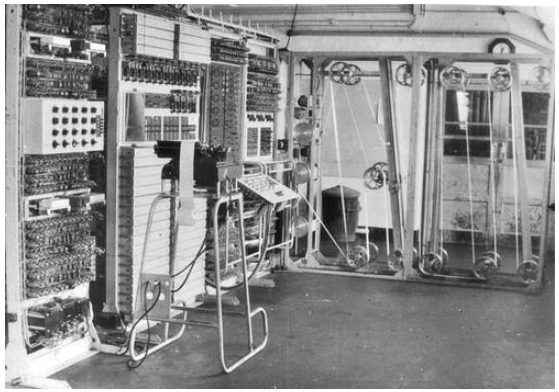


Supercomputer:"*new statistical machines with the mental power of 100 skilled mathematicians in solving even highly complex algebraic problems*"..

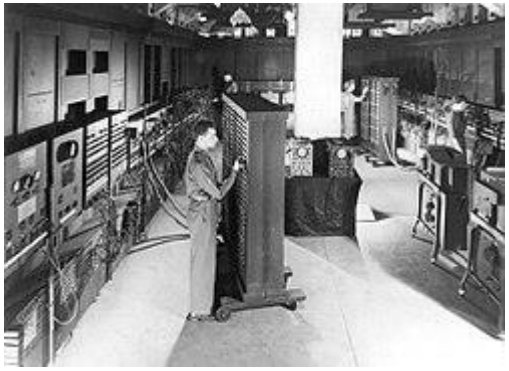
New York World, march 1920

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

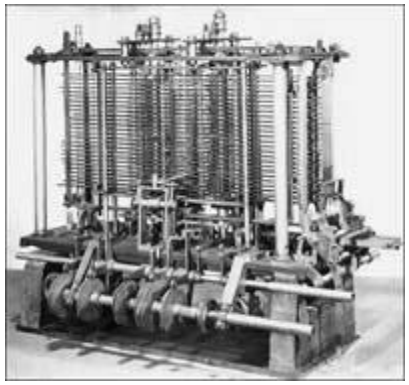
The first computers



COLUSSUS, Bletchley Park, UK (first programmable computer)



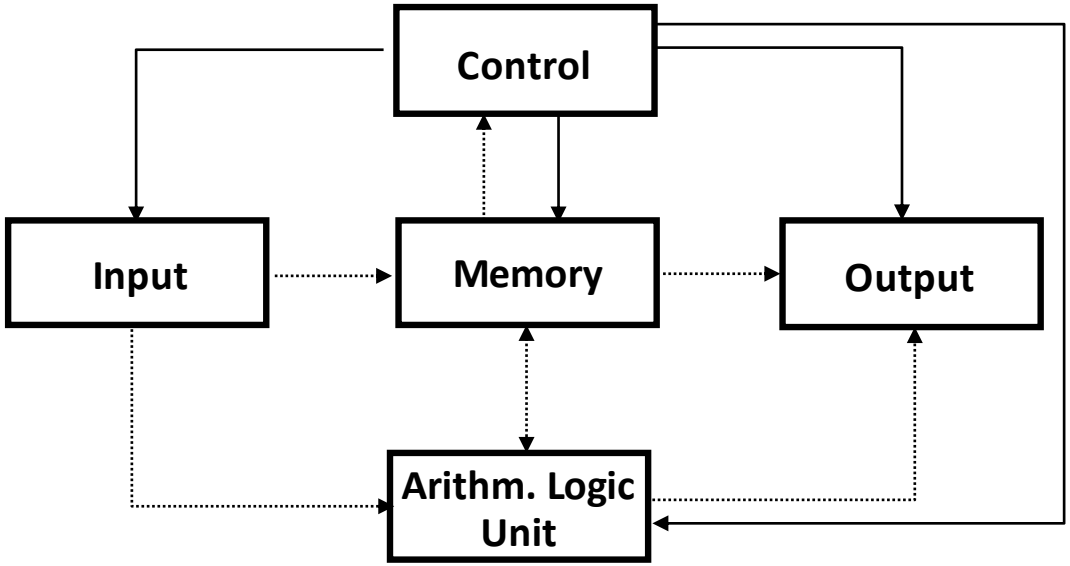
ENIAC - first electronic computer

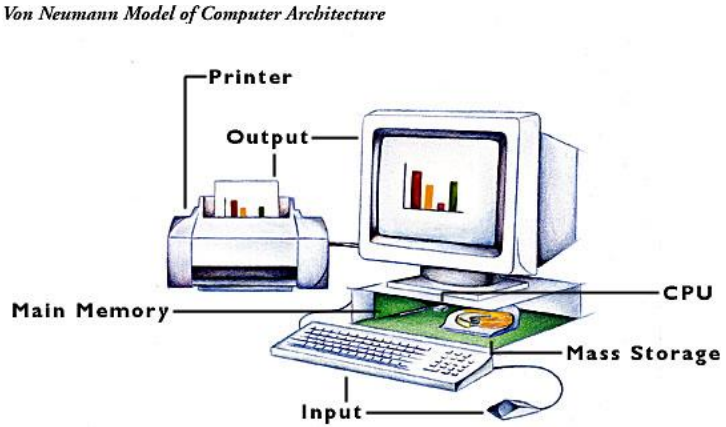


Analytical Engine (Babbage)

How do they work? - it starts from the von Neumann Model

Conventional Computer





..... 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**).

Processor speed: Clock Cycle and Frequency

The instructions of all modern processors need to be *synchronised* with a timer or *clock*.

The *clock cycle* τ 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 τ.

Processor	τ (ns)	freq (MHz)
CDC 6600	100	10
Cyber 76	27.5	36
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

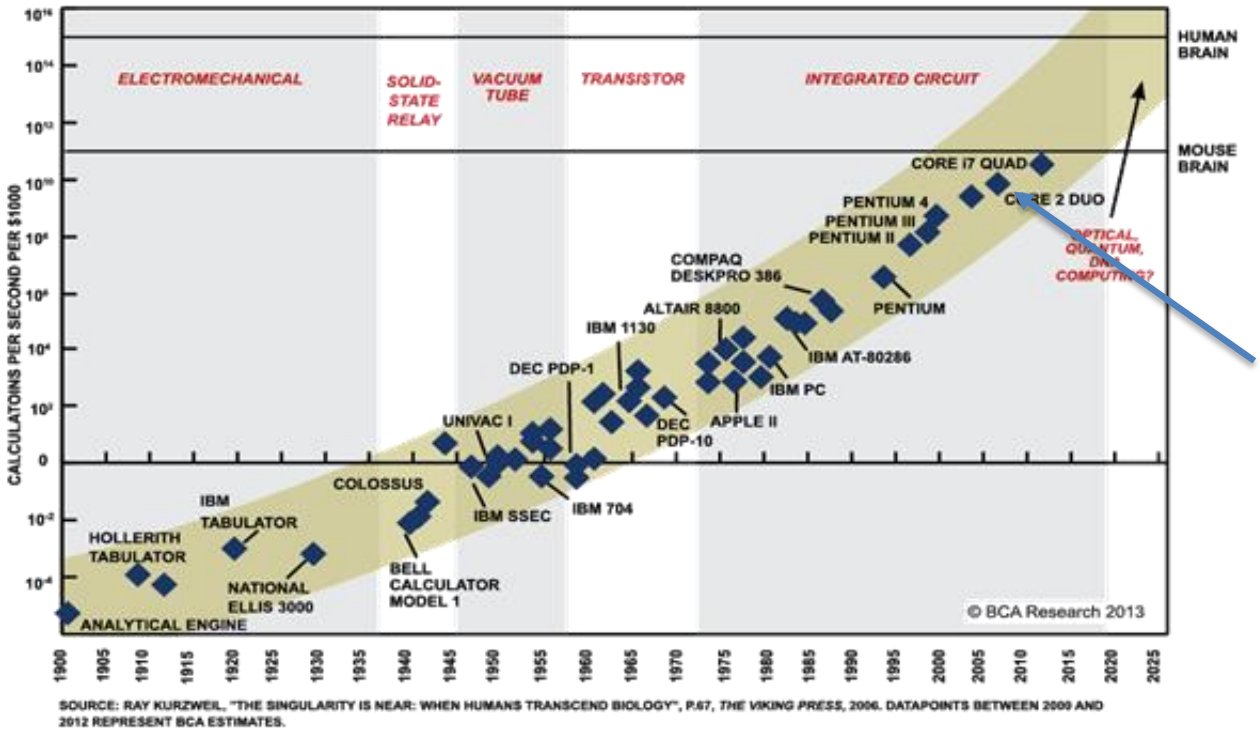
Increasing the clock frequency:

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. Power consumption varies as the square or cube of the clock frequency.

Moore's Law



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

Tri-Gate 3D transistor (2011, Intel, e.g.22nm Ivy Bridge)

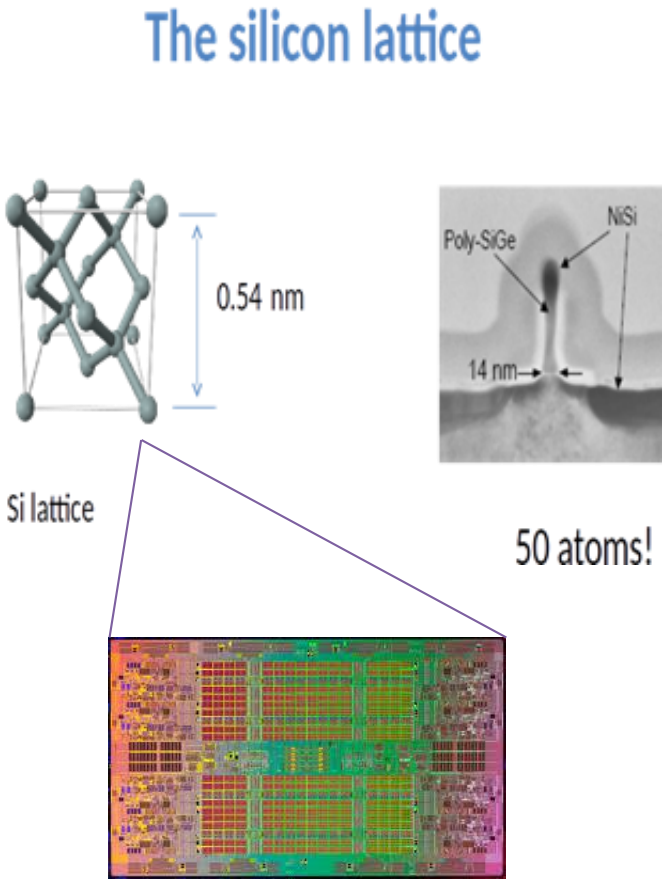
Empirical law which states that the complexity of devices (number of transistors per square inch in microprocessors) doubles every 18 months..

Gordon Moore, INTEL co-founder, 1965

The end of Moore's Law?

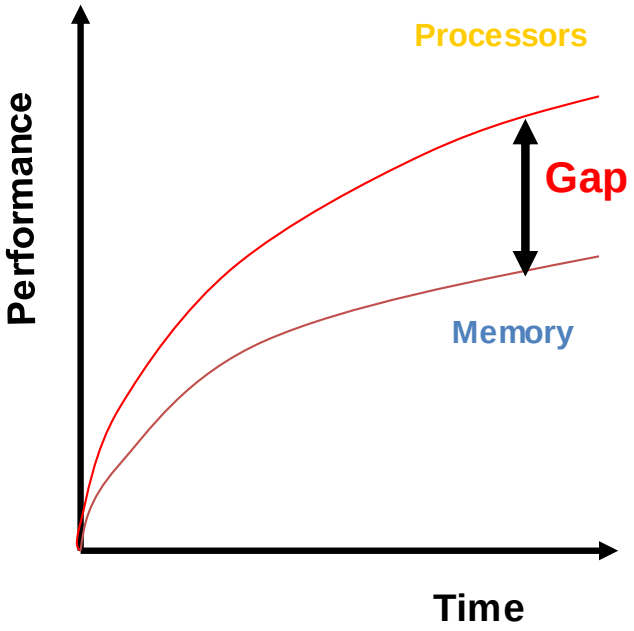
There is some debate as to whether Moore's Law still holds (probably not) but will undeniably fail for the following reasons:

- Minimum transistor size
  - Transistors cannot be smaller than single atoms. Most chips today use 14nm fabrication technology, although IBM in 2015 demonstrated a 7nm chip.
- Quantum tunnelling
  - As transistors get smaller quantum effects such as tunnelling get more important and can cause current leakage.
- Heat dissipation and power consumption
  - Increasingly difficult to remove heat and keep power levels within reasonable limits. Partially offset by multi-core chips.



Increase in transistor numbers does not necessarily mean more CPU power - software usually struggles to make use of the available hardware threads.

The processor - memory bottleneck and cache

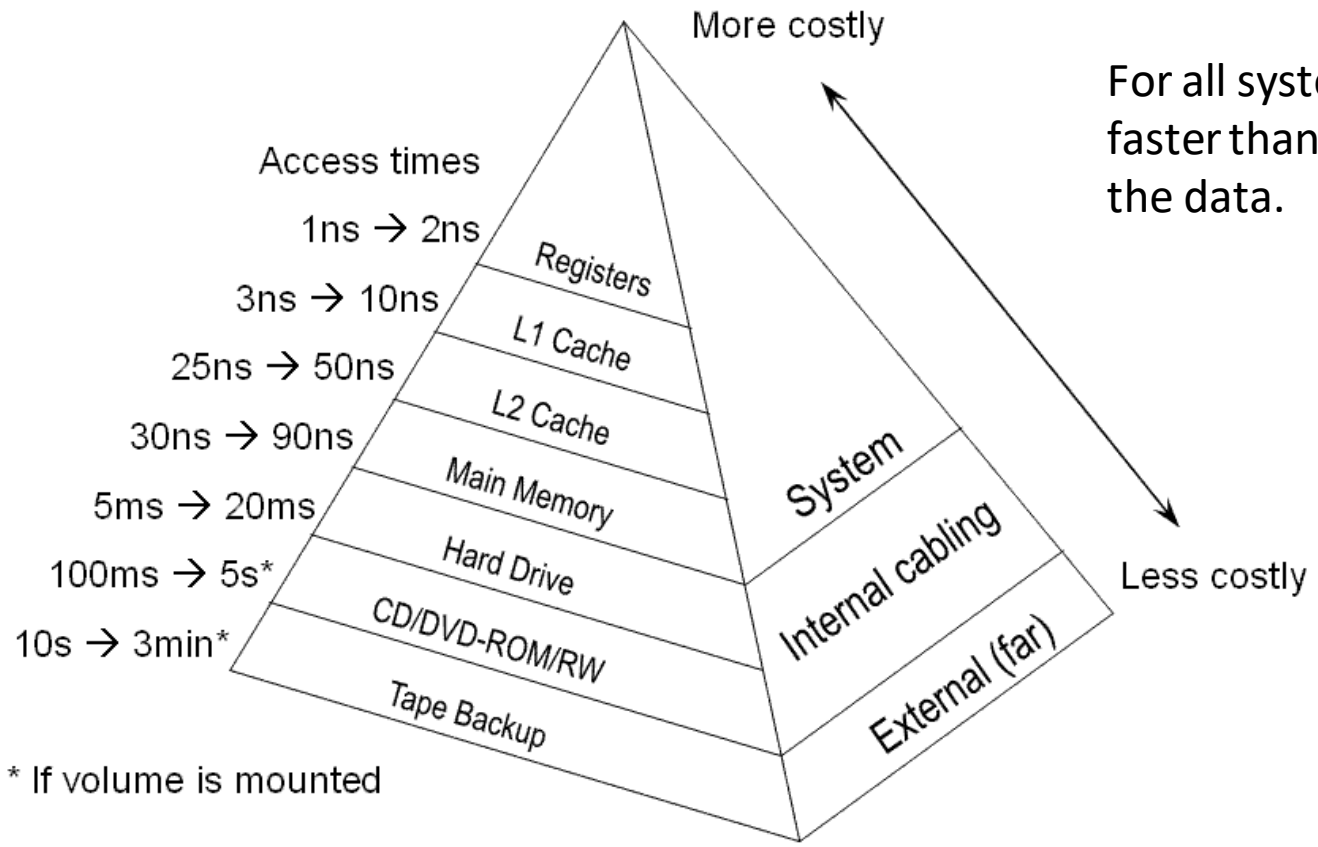


Regardless of the power of the processor, the real limitation in HPC is the performance difference between processors and getting data to/from memory which has been increasing in time.
Very important in both software and hardware design to minimise the time it takes to transfer data to and from the CPU.

Two important concepts regarding memory/data transfer:

1. *Bandwidth* - how much data can be transferred in a data channel.
2. *Latency* - the minimum time needed to transfer data.

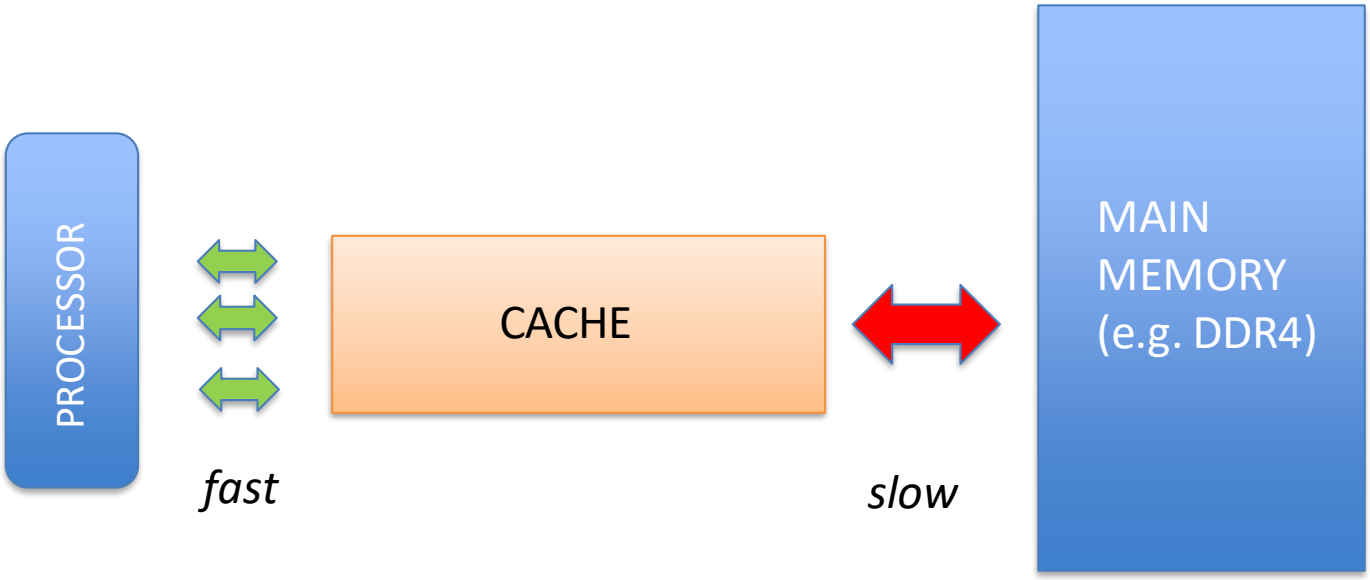
Memory Hierarchy



For all systems, CPUs are much faster than the devices providing the data.

Cache Memory

Cache memory is small but very fast memory which sits between the processor and the main memory.



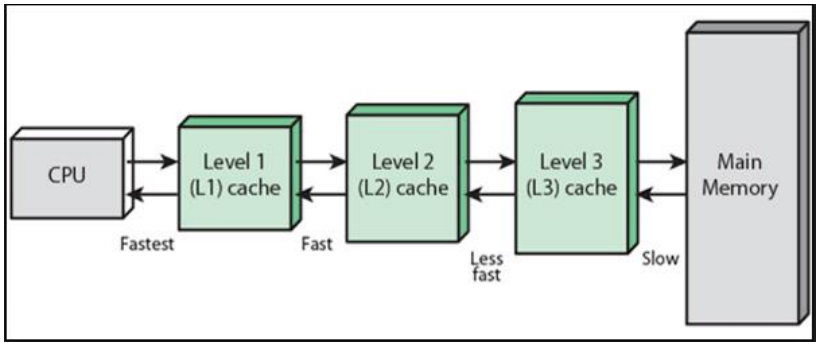
There are various strategies on how to use cache, but the idea is that when data are requested first the cache is checked, before main memory.
If the main memory needs to be accessed then similar data from main memory are loaded into the cache in the hope that the next data access will be from the cache (*cache hit*) and not from main memory (*cache miss*).

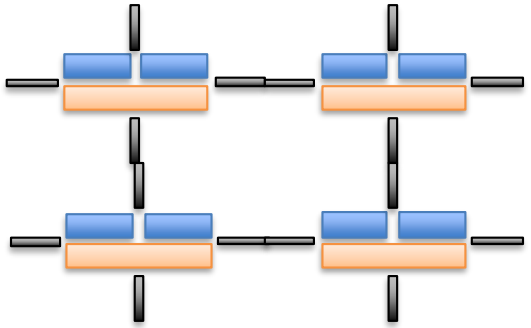
Cache levels

Cache memory is classified in terms of *levels* which describe its closeness to the microprocessor.

- Level 1 (L1) - extremely fast but small (e.g 32K), usually embedded in the CPU.
- Level 2 (L2) - bigger (e.g. 2Mb) but slower, maybe on a separate chip. Each core may have its own dedicated L1 and L2 cache.
- Level 3 (L3) - often shared amongst cores.

Caches are searched in order when seeking data (L1 -> L2 -> L3 -> main memory).





Intel KNL cache hierarchy

In HPC exploiting the cache is crucial for performance.
In particular, loops accessing arrays must be written with special care.

Programs constantly waiting for data from memory are memory bound.

Concepts of Parallelism

It has been recognised for some time that *serial computing* cannot bring the increases in performance required in HPC applications.

The key is to introduce *parallelism* which can be present at many levels:

- Instruction level (e.g. fma = fused multiply and add).
- Vector processing (e.g. data parallelism)
- Hyperthreading (e.g. 4 hardware threads/core for Intel KNL, 8 for PowerPC).
- Cores / processor (e.g. 18 for Intel Broadwell)
- Processors (or sockets) / node - often 2 but can be 1 (KNL) or >2/
- Processors + accelerators (e.g. CPU+GPU)
- Nodes in a system

To reach the maximum (*peak*) performance of a parallel computer, all levels of parallelism need to be exploited.

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).

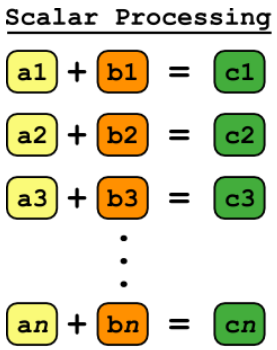
SI Single Instruction stream **SD** Single Data stream

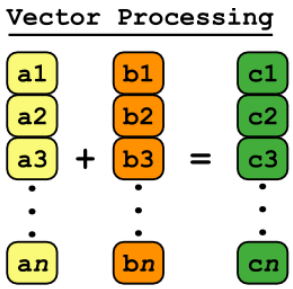
MI Multilpe Instruction stream **MD** Multilpe Data stream

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

Single Instruction Multiple Data (SIMD)

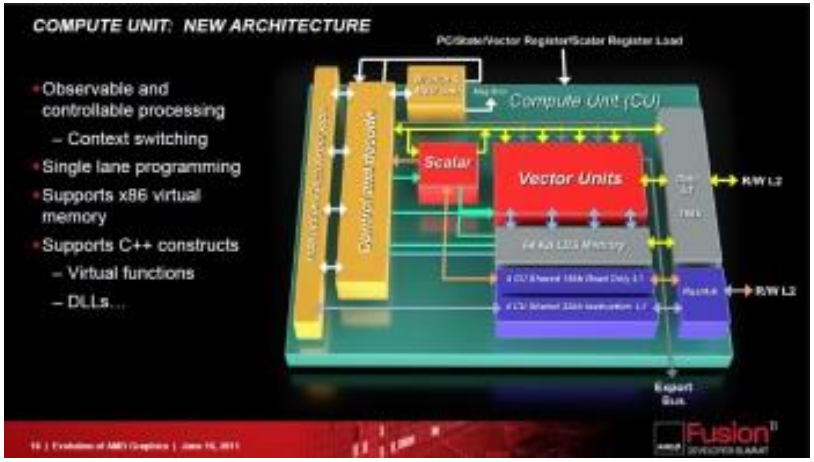
- In SIMD a single instruction is performed simultaneously on multiple data.
- In modern systems the most common use of SIMD is in *vectorisation*, performed by dedicated hardware on the chip (*vector units*).
- Compiler generates vector instructions, when it can, from programmer's code.
- Important optimisation which can lead to 4x, 8x, or more speedups according to the "size" of the vector unit (e.g. 256 bit).






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

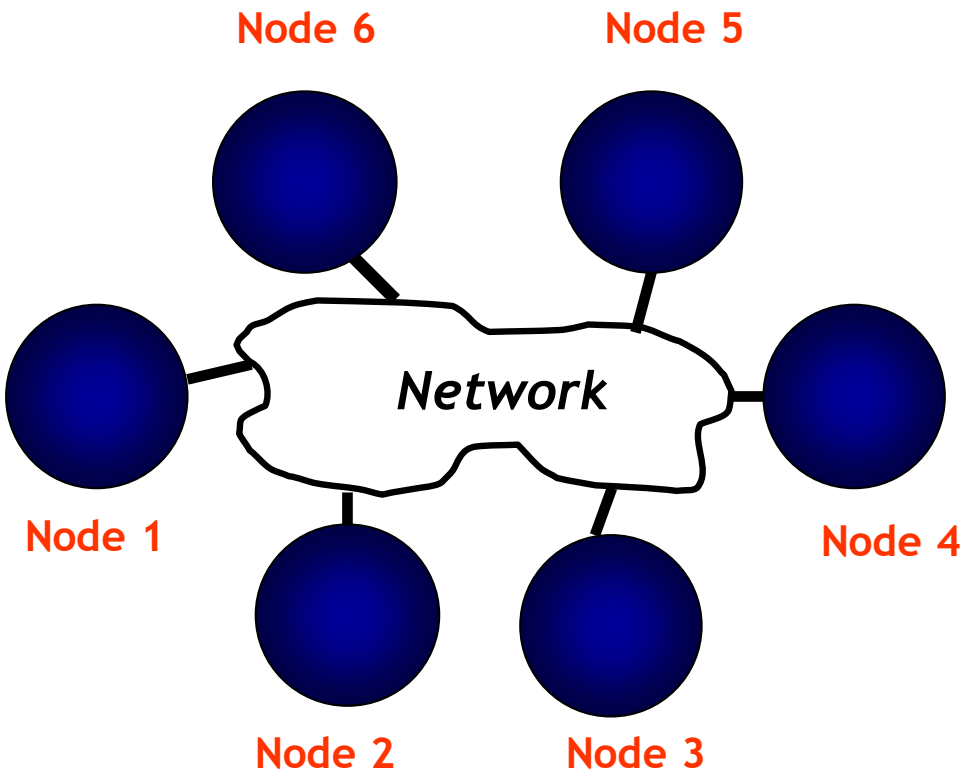
```
c[1:n] = a[1:n] + b[1:n]
```



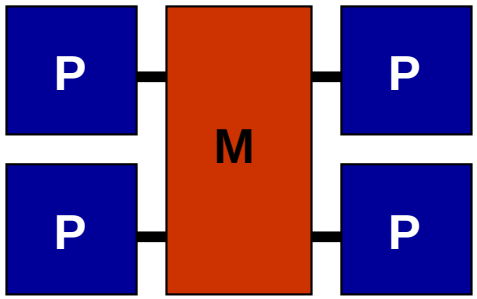
Multiple Instruction Multiple Data (MIMD)

Multiple processors execute different instructions operating on different data.

MIMD applies to most other parallel situations in HPC, e.g. supercomputing clusters.

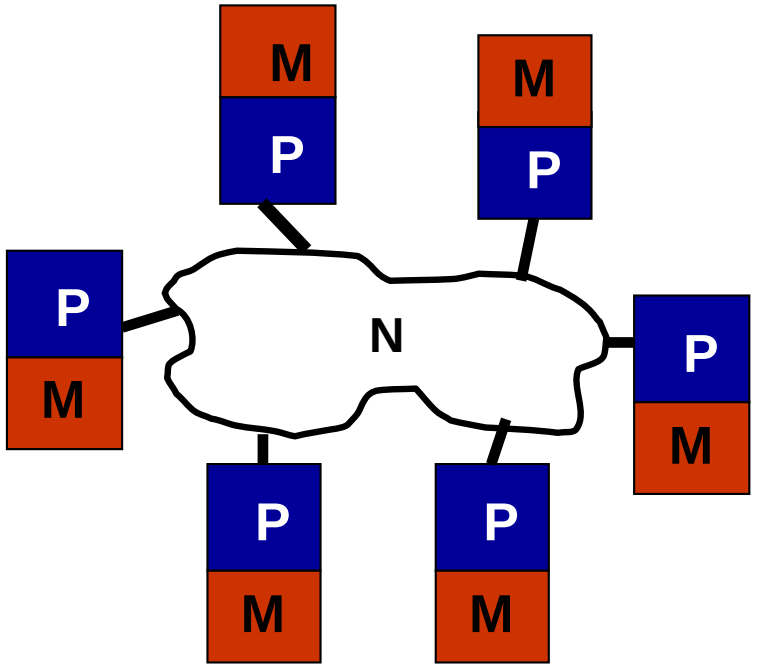


Cluster Classification based on Memory



Shared Memory System

It is rare nowadays to find entire clusters where all processors in the system see the same memory (poor performance). Instead we have clusters where memory is shared only within a node. For processors in different nodes to share memory the network must be used.

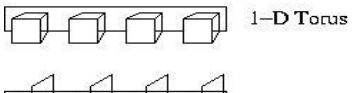


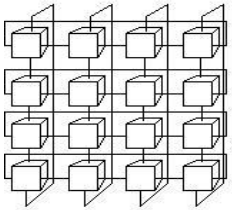
Distributed Memory System

Non Uniform Memory Access (NUMA)
The time each processor needs to access the memory is not uniform - faster if a processor accesses local memory, rather than use the network.

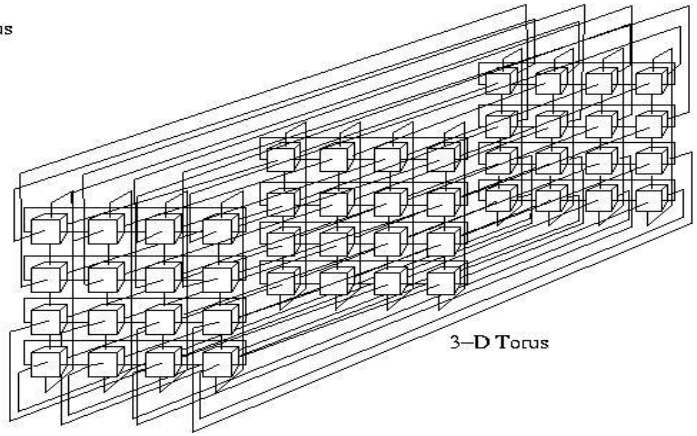
Network technologies and topologies

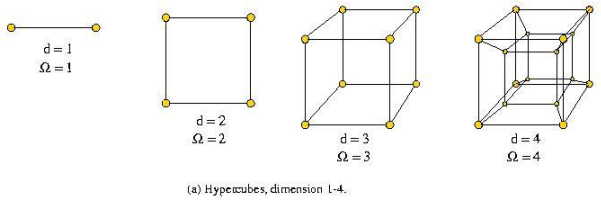
Networks linking the nodes in a distributed system are available according to price, performance and hardware vendor.
Examples include: Ethernet, gigabit, Infiniband, Omnipath (Intel), etc.
If switches are used then the network is often called the *fabric*.
In addition, networks can be configured in a particular *topology*.





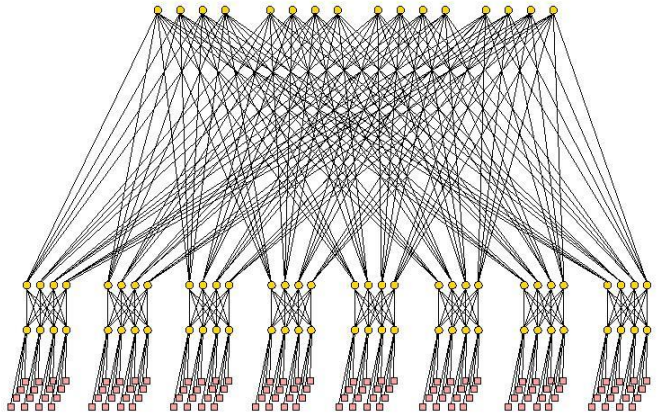
TORUS





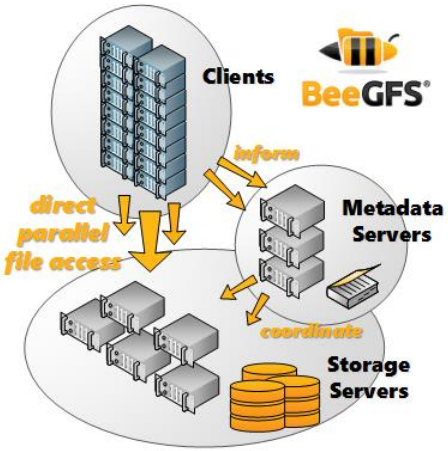
(a) Hypercubes, dimension 1-4.

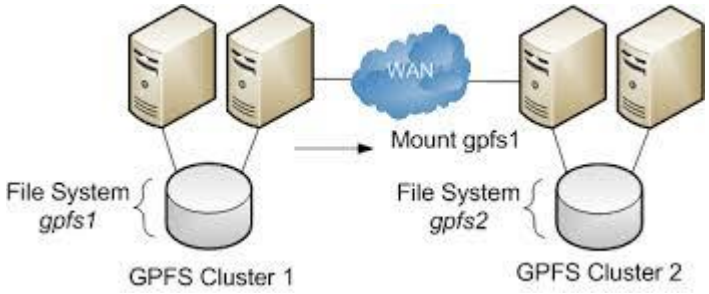
FAT TREE


(b) A 128-way fat tree.

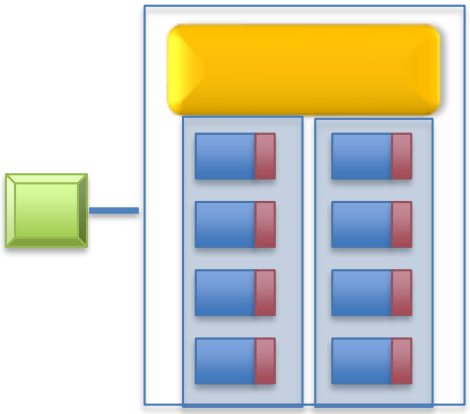
Parallel Filesystems

The filesystem manages how files are stored on disks and how they can be retrieved or written.
In a parallel architecture, with many simultaneous accesses to the disks, important to use a *parallel filesystem* technology such as GPFS, LUSTRE, BeeGFS etc.





Parallel computers - putting it all together



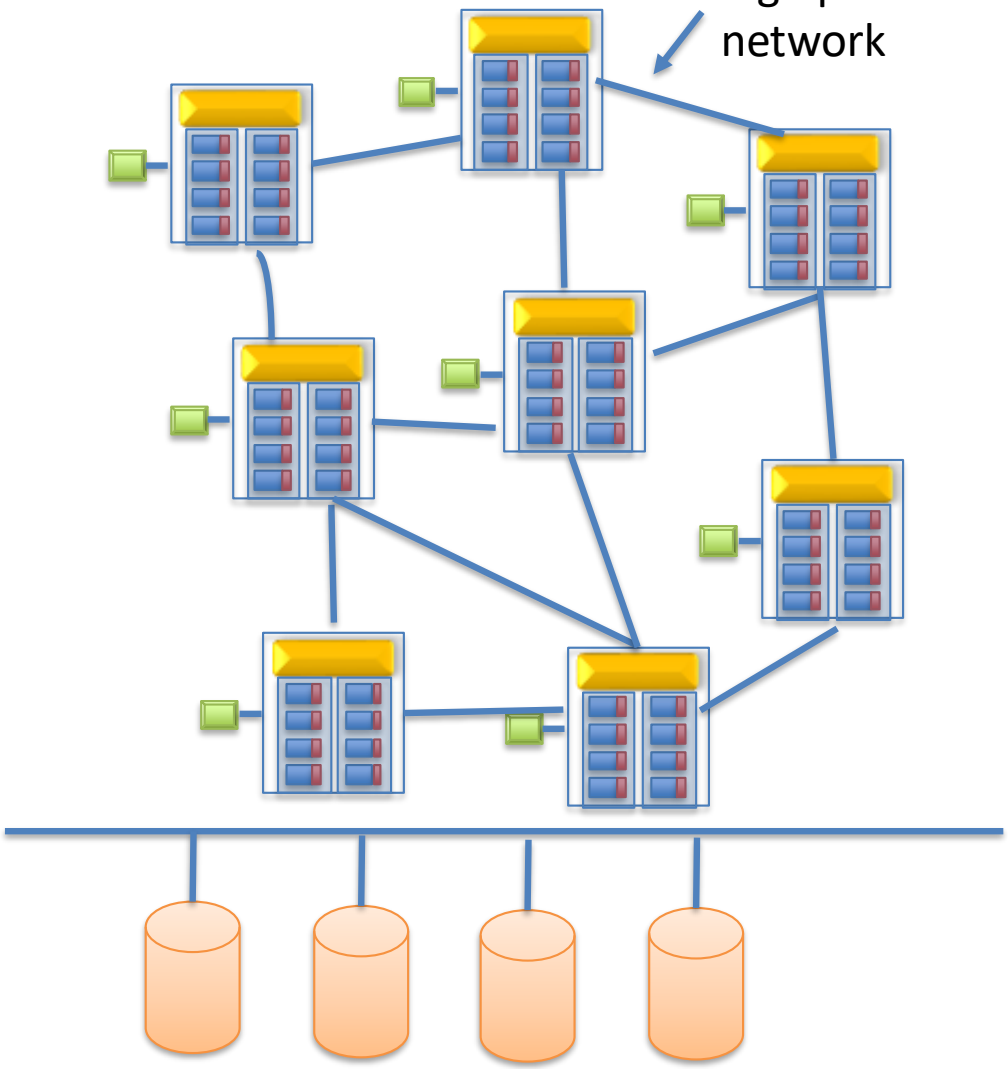
memory

accelerator (e.g. GPU)

processor

core

vector unit



Programming parallel computers

An HPC application needs to be programmed to use efficiently the hardware at the following levels:

Intra-node (within one node)

- Processor-level
  - Core-level
    - Cache memories (particularly L2 and possibly L3)
    - Vector units
    - Hyper-threading (esp. Intel KNL, IBM Power8)
- Effective use of all cores within node using shared memory and threads (e.g. careful *pinning* to ensure processes are allocated sensibly among processors in the node).
- Accelerator such as GPU or FPGA.
- Local disk storage

Inter-node

- Minimise communications between nodes
- Careful use of parallel filesystem.

Programming parallel computers

We therefore need libraries, tools, language extensions, algorithms and paradigms which allow us to:

- within a node exploit vector and cache units, hardware threads on cores, use shared memory within nodes;
- manage inter-node connections such that a process can exchange data with processes on other nodes
- debug and profile programs to check correctness of results and performance

As of 2018 the software languages most used for parallel programming have been FORTRAN and C/C++ - languages not originally designed for parallelism.

Also used include Python MPI, CUDA (GPUs) and Intel CILK++.

Some examples of parallel programming libraries and languages

Message Passing Interface (MPI).

- Allows parallel processes to communicate via sending “messages” (i.e. data). Most standard way of communication between nodes, but can also be used within a node.

OpenMP

- Allows parallel processes to communicate via shared memory in a node. Cannot be used between shared memory nodes.

Hybrid MPI+OpenMP

- Combines both MPI+OpenMP. A situation could be to use OpenMP within a shared memory node and MPI between nodes.

OpenAcc

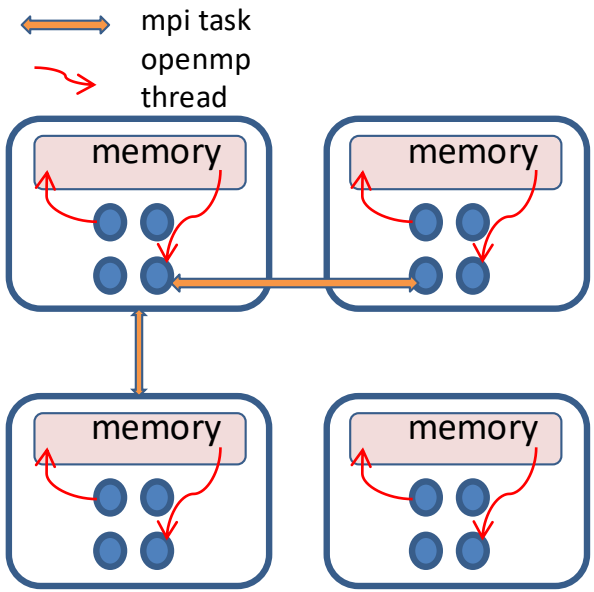
- Similar to OpenMP but used to program devices such as GPUs.

CUDA

- Nvidia extension to C/C++ for GPU programming.

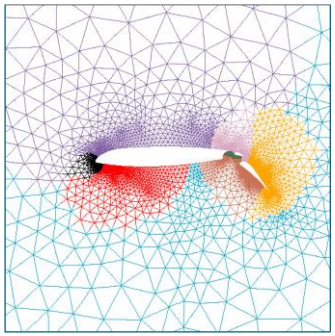
OpenCL

- Non-Nvidia alternative to programming GPUs.

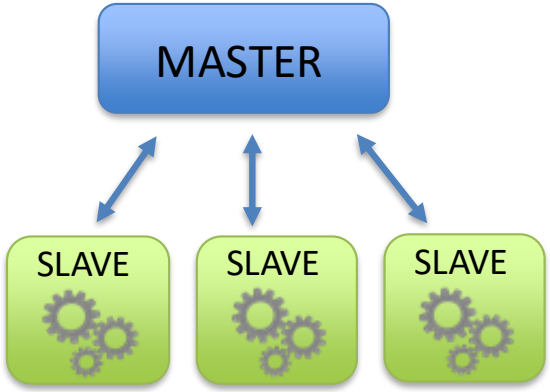


Parallel algorithms

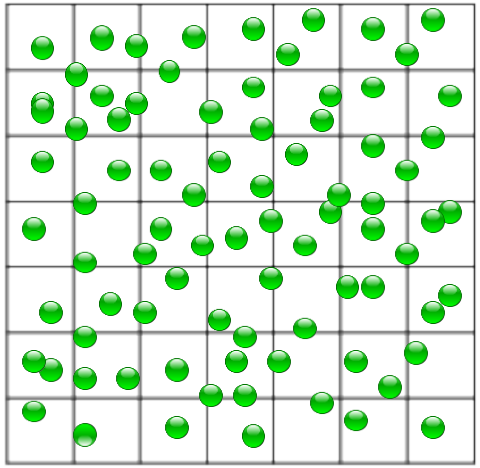
We have the parallel hardware and software libraries but we also need to map our problem into a parallel algorithm. Types of parallelism include: data parallelism, task parallelism and pipelining. From an application viewpoint parallel models include:



structured and un-structured meshes (e.g. CFD)



master-slave (e.g. data analysis).



domain decomposition (e.g. FFT, molecular dynamics)

PART 2. Trends and advances

HPC evolution

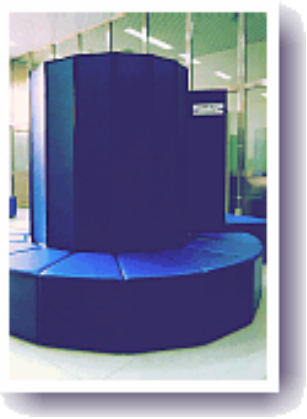
Which factors drive the evolution in HPC architecture?

- The first (super) computers were mainly used by defence organisations and the US Govt (esp. Department of Energy) still makes significant investments. Later they were used for scientific research in a small number of centres.
- But the high cost of the dedicated components, and the fact that HPC is not a strong market, has caused a shift into using commodity or off-the-shelf devices such as processors, disks, memories, networks, etc.
- This shift has had a number of consequences:
  - Some manufactures have changed business or no longer make supercomputers (e.g. SUN microsystems).
  - Other supercomputer vendors (e.g. CRAY and SGI) no longer make microprocessors so the market is dominated by one or two brands, i.e. Intel and, to a lesser extent, IBM PowerPC.
  - Since microprocessors were not designed for HPC, the programmer must work harder to get maximum performance.
- But porting has become easier as only a few processor types are available and Linux has replaced all the other operating systems. It is now also possible for smaller organisations such as university departments to run small clusters.

Supercomputer evolution in Cineca

1969: CDC 6600 1st system for scientific computing
1975: CDC 7600 1st supercomputer
1985: Cray X-MP / 4 8 1st vector supercomputer
1989: Cray Y-MP / 4 64
1993: Cray C-90 / 2 128
1994: Cray T3D 64 1st parallel supercomputer
1995: Cray T3D 128
1998: Cray T3E 256 1st MPP supercomputer
2002: IBM SP4 512 1 Teraflops
2005: IBM SP5 512
2006: IBM BCX 10 Teraflops
2009: IBM SP6 100 Teraflops
2012: IBM BG/Q 2 Petaflops
2016: Lennovo (Marconi) 13 Pflops
2018: Lennovo (Marconi) 20 Pflops








TOP500 list November 2017

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - 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
3	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 Cray Inc.	361,760	19,590.0	25,326.3	2,272
4	Japan Agency for Marine-Earth Science and Technology Japan	**Gyoukou** - ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz ExaScaler	19,860,000	19,135.8	28,192.0	1,350
5	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
6	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
7	DOE/NNSA/LANL/SNL United States	**Trinity** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect Cray Inc.	979,968	14,137.3	43,902.6	3,844
8	DOE/SC/LBNL/NERSC United States	**Cori** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect Cray Inc.	622,336	14		
9	Joint Center for Advanced High Performance Computing Japan	**Oakforest-PACS** - PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path Fujitsu	556,104	13		
10	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

List published twice a year (June and November) listing the top 500 most powerful computer systems.
Dominated in recent years by Japan and China, although US expected to advance with the CORAL procurement.
CINECA currently in 14th place but may rise due to upgrade.
Not included yet 18.6 Pflop GPU cluster owned by ENI.

CINECA Marconi

HPC Evolution - The Energy cost

The main driver in the last 10 years has been the need to reduce the power consumption.

This has already led to multi-core processors which are now multi-core (4,6,8, 10 and increasing) but of low frequency (rarely above 2.5 GHz) as power consumption varies exponentially with GHz.

To increase overall performance, but keep energy costs down, overall parallelism must be increased.

Two possible solutions have been proposed for low energy clusters:

1. Large homogenous clusters
2. Hybrid clusters with different processors and accelerators

The Homogenous solution - IBM Bluegene

Homogeneous cluster containing very large number of low-power cores (tens or hundreds of thousands).

Common HPC solution for a number of years but unpopular with users since applications needed to be very highly parallel (at least use upto 1024 cores) and had poor I/O performance.

Also non-Linux OS on compute nodes.

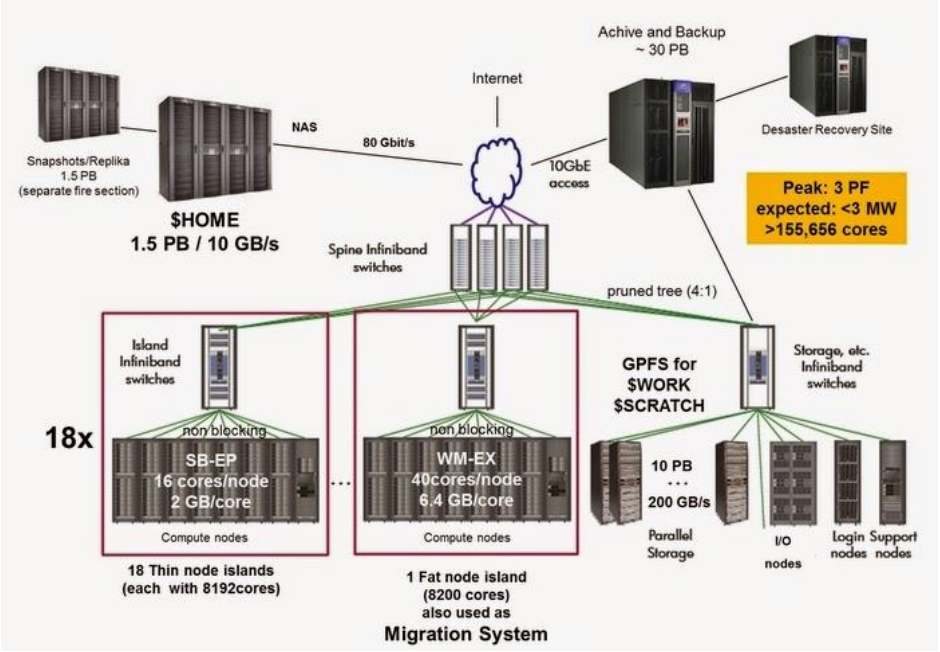
Discontinued by IBM but still in TOP500.

IBM BG/Q system (Fermi)

- 163840 cores at 1.6GHz
- 10240 nodes (16 cores/node)
- 16 GB/node
- 5D Torus network
- 2 Pflops performance

Heterogenous, hybrid clusters

- The most common solution is a hybrid architecture consisting of different types of processor, accelerator or other devices in “islands” or partitions.
- Might be difficult to manage but more flexible since different application-types are likely to fit.
- Cineca Marconi is a hybrid system with 3 types of Intel processors (Broadwell, KNL, Skylake) in 3 partitions (A1, A2 and A3).
- Other clusters may have “fat” or “thin” nodes depending on memory available (e.g. LRZ in Munich).



Intel Xeon PHI

Low power device range based on Intel's Many Integrated Core (MIC) technology.

Large number of low frequency Pentium cores (e.g. 1.0 GHz) loosely connected on a chip with onboard memory.

The first commercially available Xeon Phi device Knight's Corner (KNC) could be used only as an accelerator.

Although runs standard FORTRAN and C/C++, difficult to obtain good performance.

Many application developers did not optimise codes for KNC.

Intel Knight's Corner

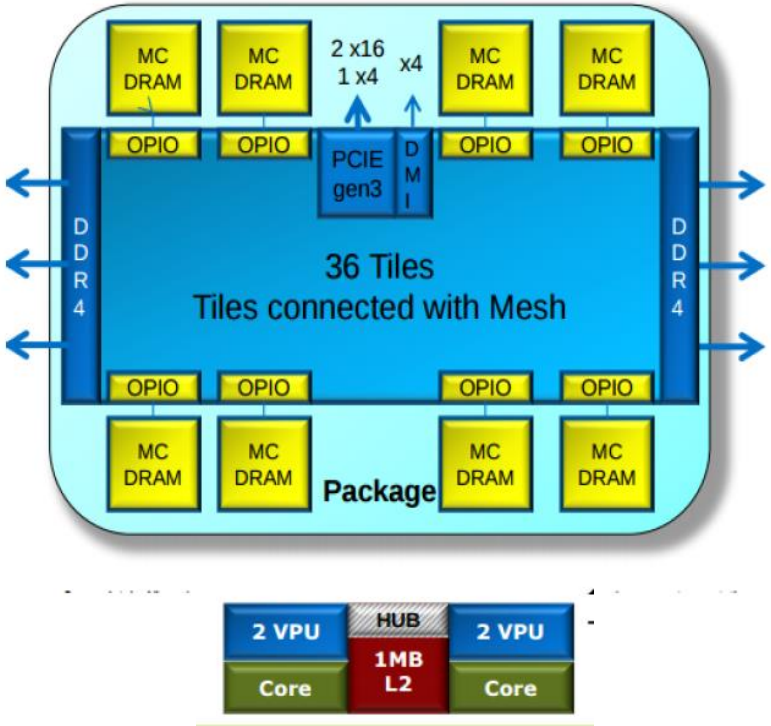
- 61 cores, 1.0-1.2 GHz
- 8-16 Gb RAM
- 512 bit vector unit
- 1-2 Tflops
- ring topology of cores
- With compiler option, runs standard FORTRAN or C (i.e. no CUDA or OpenCL necessary) and MPI.

Intel Xeon PHI - Knight's Landing (KNL)

Major upgrade to KNC:

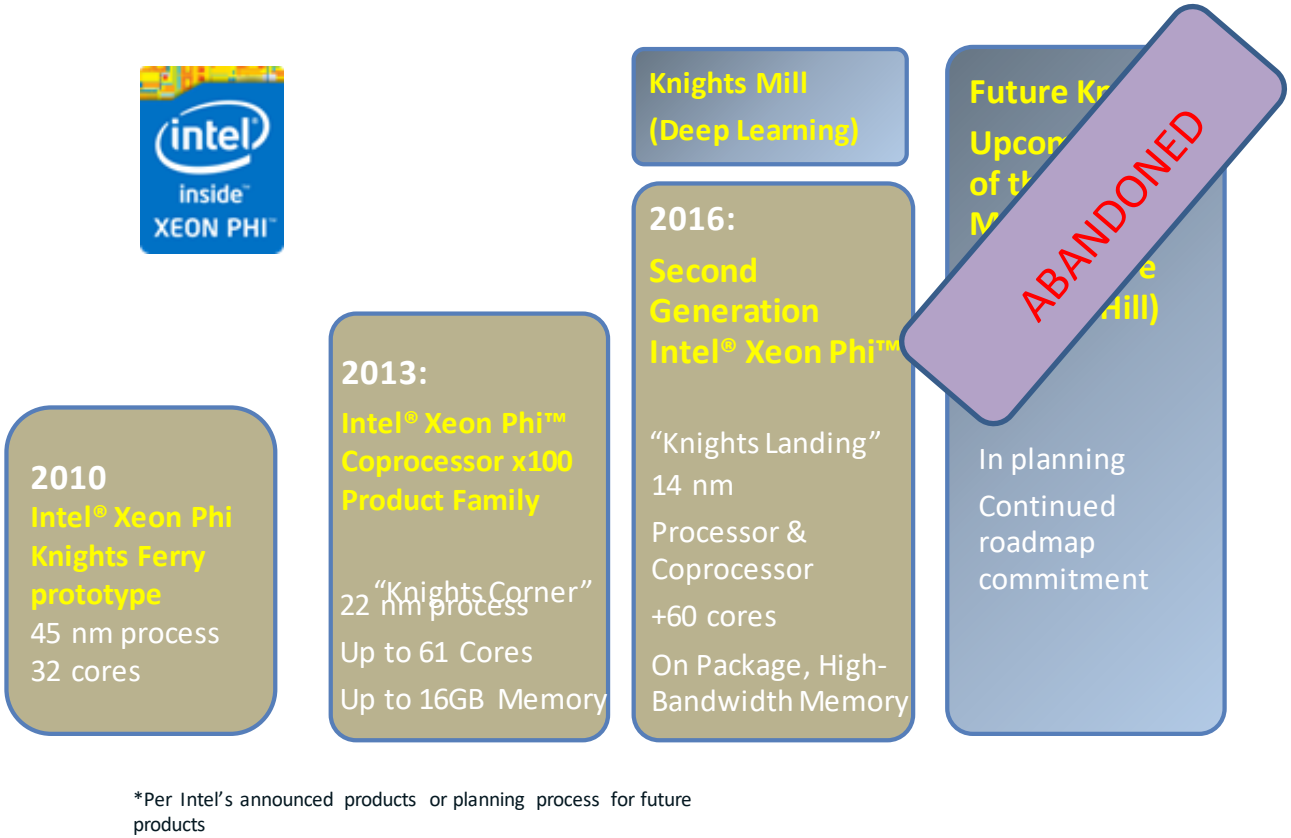
- Standalone, self-boot CPU.
- Upto 72 Silvermont-based cores (1.4 GHz)
- 4 threads, 2 AVX512 vector units/core (i.e. 272 threads in total).
- 2D Mesh interconnect
- 16 GB MCDRAM (High Bandwidth Memory) 400 Gb/s.
- Intel OmniPath on chip.
- 3 Tflops (DP) peak per package.

Binary compatible with other Intel processors but recommended to recompile to allow use of extended vector units.



Marconi A2 partition consists of 3600 nodes with a total performance of 13 Pflops.

Intel Xeon PHI Roadmap



*Per Intel's announced products or planning process for future products

In November 2017 Intel announced that the Xeon Phi line would be abandoned due to "market and customer needs" (Intel).
Probably means that KNL had no market outside HPC.

Intel Xeon Phi lessons

Although no future for Intel Xeon Phi, the experience has been useful:

- with 68 cores/node you need only a few KNL nodes in order to test parallel scaling (i.e. speedup as no. of cores increase)
- emphasised the need for multi-threaded applications, good for testing programming paradigms which use this model (e.g mixed MPI+OpenMP).
- KNLs have been good for testing new high bandwidth memories (i.e. MCDRAM).

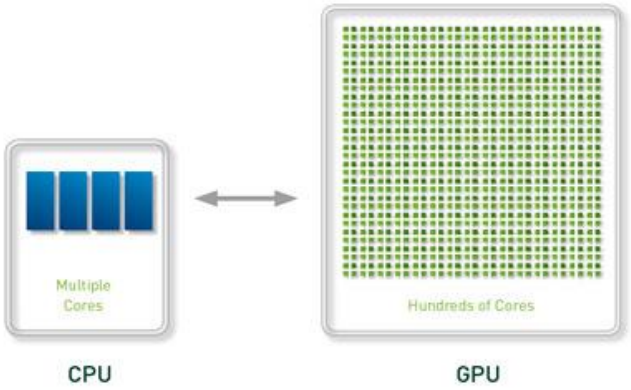
GPGPUs

General Purpose GPUs, or simply GPUs (Graphical Processing Units), are devices designed for graphics processing which are used in non-graphical applications.

Became popular in HPC in 2006-2007 when Nvidia introduced CUDA, greatly simplifying the programming of such devices.

The design is different to a standard CPU, being based on hundreds or thousands of *stream processors*.

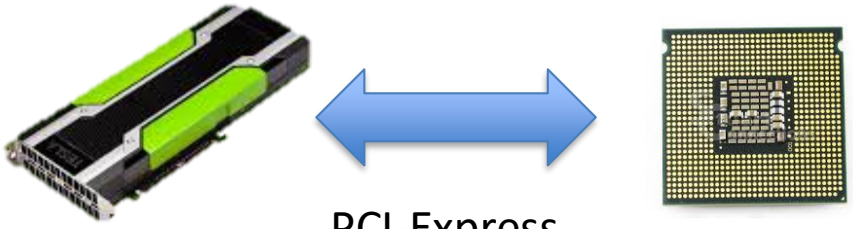
Used as an *accelerator*, attached to a conventional CPU (e.g. Intel). The idea is that parallel sections of the application are *offloaded* to the GPU, while the CPU handles the sequential sections. In certain circumstances can give large speed increases, compared to non accelerated code.



GPGPUs

Plus points:

- Can accelerate applications many times (e.g. 2x,3x even 20x or more)
- Performance in Flops/Watt and price in Flops/$ often much better than conventional CPUs.



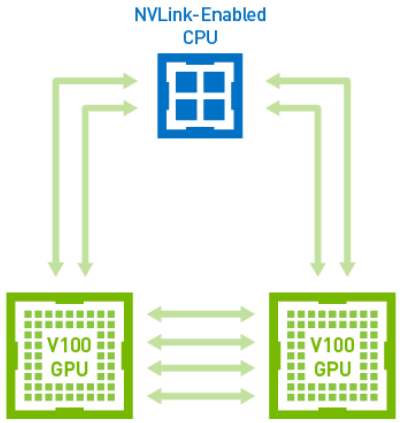
PCI-Express
(30 GB/s)

Difficulties:

- Need to use CUDA (a C dialect) to get best performance, although other methods are becoming available (OpenACC, FORTRAN etc.).
- Porting to GPUs requires effort -> some applications do not have CUDA ports.
- PCI-e bus (connection) to the CPU is quite slow and device memory is limited (e.g. 16 Gb)

GPGPU advances

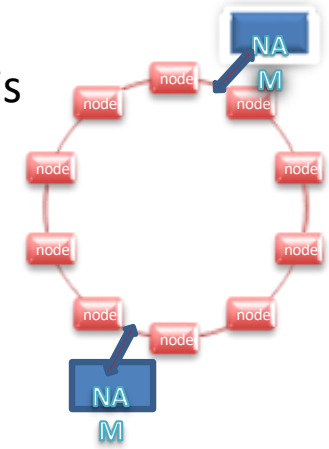
- Latest Nvidia devices (Pascal P100 and Volta V100) can use Nvlink which is upto 10X the speed of PCIe.
- Allows also fast GPU - GPU connections.
- Unified memory simplifies memory management of applications.
- Nvidia GPUs are becoming important in DEEP learning applications.



New memory technologies

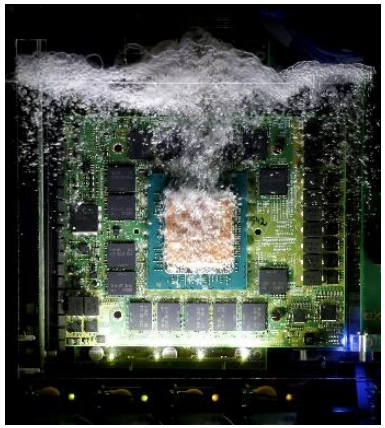
There have been rapid advances in disk and memory technologies, prompted by increasing needs in data storage:

- NVRAM (Non-Volatile Random Access Memory)
  - e.g. Flash memory. Retains information even when power switched off. Current use is for Solid State Disks (SSD) to replace "spinning disks" (i.e. conventional disks). SSD storage has particularly low latencies.
- HBM (High Bandwidth Memory)
  - e.g. MCDRAM in KNL, shared memory in Nivida P100, V100. High bandwidths (400-850 Gb/s) compared to standard DDR4 memory (e.g. 100 Gb/s). May need code changes to use.
- NAM (Network Attached Memory)
  - Memory attached directly to the network, rather than passing through the processor. Capable also of limited processing. Idea is to enable *near data computing*. Still in the research stage for HPC.



Cooling technologies

Need to reduce (and possibly re-use) waste heat has led to innovative cooling technologies such as hot water cooling instead of air and even immersing components in heat-removing solvents.



DEEP GreenICE Booster

SuperMUC uses 40 percent less energy than would be required by an equivalent air-cooled system.

The Road to Exascale

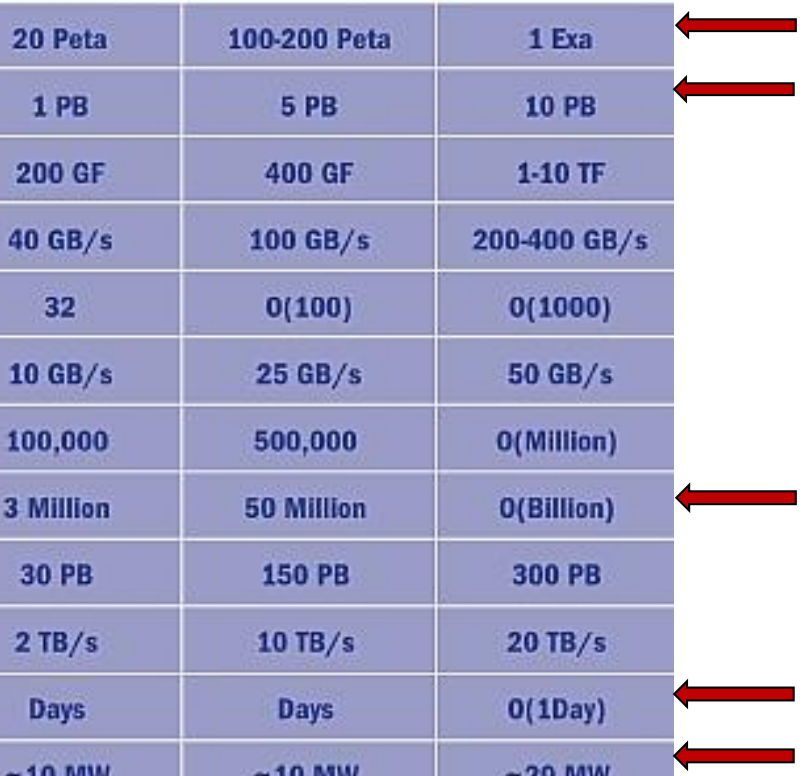
Multi-core, accelerated clusters with innovative cooling designs have brought petaflop class machines to everyday HPC users.

What do we need to go the next step, i.e. to have a computer able to reach *1 Exaflop*?

..the US DoE performed a survey

Roadmap to Exascale (architectural trends)

Systems	2009	2011	2015	2018
System Peak Flops/s	2 Peta	20 Peta	100-200 Peta	1 Exa
System Memory	0.3 PB	1 PB	5 PB	10 PB
Node Performance	125 GF	200 GF	400 GF	1-10 TF
Node Memory BW	25 GB/s	40 GB/s	100 GB/s	200-400 GB/s
Node Concurrency	12	32	O(100)	O(1000)
Interconnect BW	1.5 GB/s	10 GB/s	25 GB/s	50 GB/s
System Size (Nodes)	18,700	100,000	500,000	O(Million)
Total Concurrency	225,000	3 Million	50 Million	O(Billion)
Storage	15 PB	30 PB	150 PB	300 PB
I/O	0.2 TB/s	2 TB/s	10 TB/s	20 TB/s
MTTI	Days	Days	Days	O(1Day)
Power	6 MW	~10 MW	~10 MW	~20 MW



but widely overoptimistic – still nowhere near Exascale (2017)

Exascale

In 2018 we still do not have a computer capable of 1 Exaflop - now expected around 2022-2025.

The trends are fairly clear:

- Large number of parallel processes or threads.
- With very high parallelism, hardware failures more problematic.
- Memory and I/O bandwidth not increasing at the same rate as concurrency.
- Memory/core decreasing.

Exascale prototypes

DAVIDE

IBM Power8 +GPU (P100) cluster with innovative cooling design.

Currently in pre-production at Cineca.

Mont Blanc

Prototype hardware and clusters based on ARM chips (Barcelona Supercomputing Centre).

The Road to Exascale

But energy efficiency remains the main problem

- Energy consumption for most computers in the TOP500 is of the order of a few Gflops/W (Green Top500 Shoubu system B =17 Gflops/W for a peak of 840 Tflops).
- Scaling upto 1 Exaflop gives hundreds of MW power (~60 MW for Shoubu), i.e. impractical (and unethical) energy consumption.

In Europe 1 MWh costs between 30-50 Euros*

Also:

"This year [2018], electricity use at Bitcoin mining data centres is likely to exceed that of all Iceland's homes" (840 gigawatt hours)

(BBC, http://www.bbc.com/news/technology-43030677)

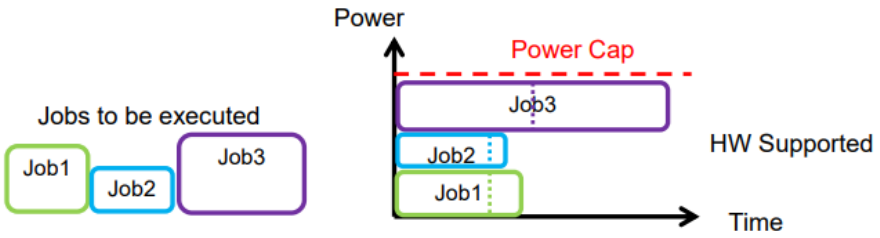
* The European Power sector in 2017, https://sandbag.org.uk/wp-content/uploads/2018/01/EU-power-sector-report-2017.pdf

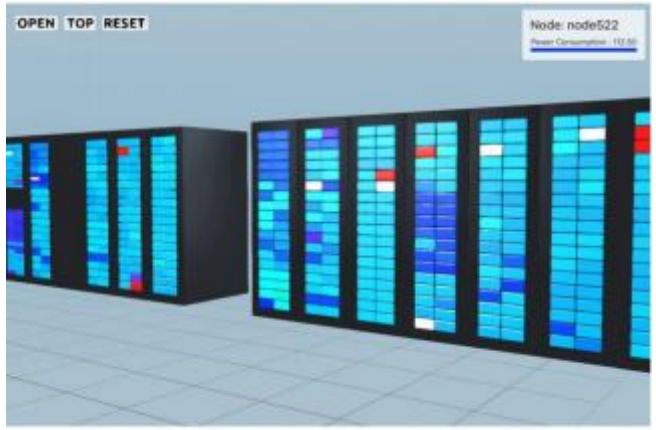
Measuring energy consumption

Considerable research efforts are being directed towards measuring the energy efficiency of applications, using sensors installed in the hardware.

Uses of energy measurements include:

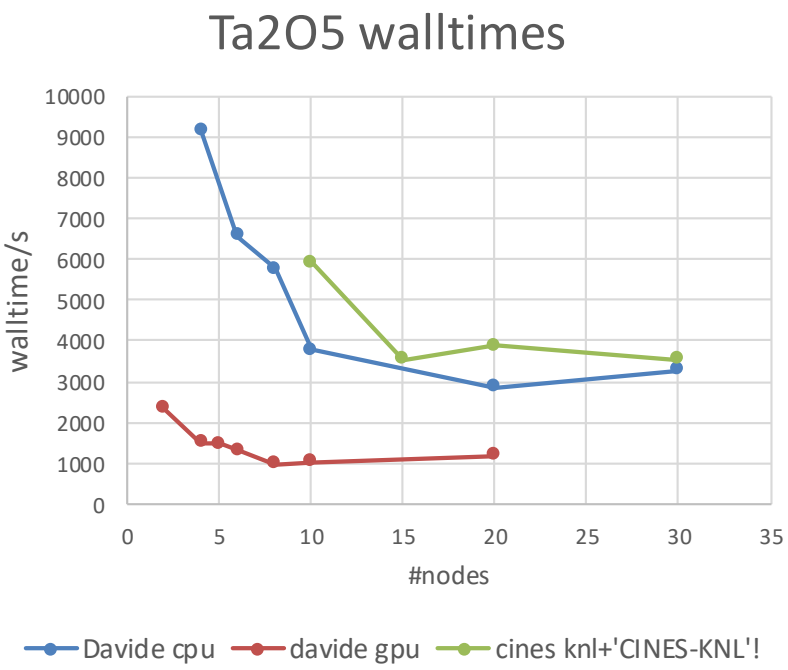
- Charging users based on energy consumption, rather than just wall time, to encourage good usage of energy resources.
- With fine-grain management possible to see which periods of a program's execution require less/more energy -> possible to cycle up/down the processor as a result.
- System managers can see which nodes are overheating or faulty or can deploy *power capping* based on estimates on what power batch jobs will use.

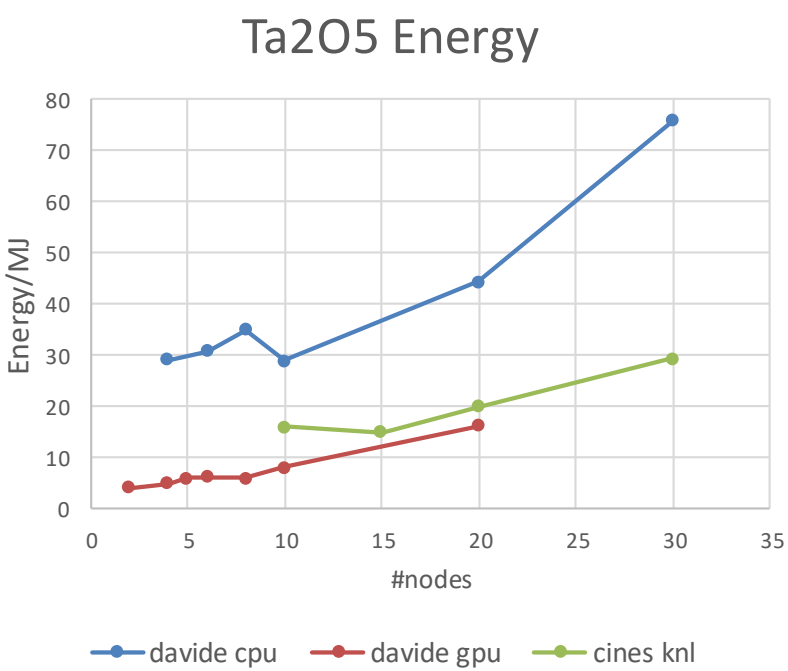




CINECA works closely with the group of Andrea Bartolini (UniBo) for energy measurements.

Energy benchmarks for Quantum Espresso





Tests in a PRACE project have shown GPUs to be more energy efficient than CPUs alone (Power8) or Intel KNLs.

Major Exascale Initiatives in US and Europe

- CORAL (United States)
  - Major US procurement program for pre-exascale computers based on different technologies.
  - Systems range from 100-200 Pflops, expandable to 1 exaflop.
  - Computer systems include:
    1. Sierra (LLNL), Power +Nvidia GPUs.
    2. Summit (Oak Ridge), Power9 +GPUs.
    3. Aurora (Argonne), originally Cray +Intel Knights Hill but now delayed to 2022-2023 (probably CPU+GPU).

CORAL
COLLABORATION
OAK RIDGE • ARGONNE • LIVERMORE

- European HPC Initiative
  - Declaration signed on March 23 2017 signed by 7 countries (France, Germany, Italy, Luxembourg, the Netherlands, Portugal and Spain) supporting exascale computing by 2022-2023.
  - Plans include to develop a European Processor.

EuroHPC
http://EuroHPC.eu

Disruptive technologies

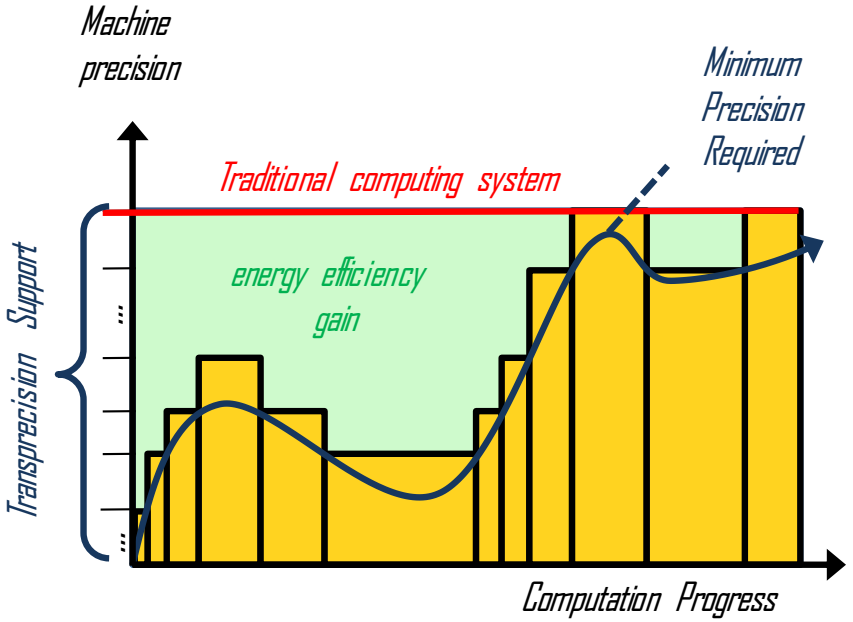
disruptive innovation - a major change in the status quo, as opposed to sustainable innovation which advances by incremental improvements.

Field Programmable Gate Arrays (FPGA)

- Devices where the hardware can be configured by the user rather than the manufacturer.
- Interest in HPC as an accelerator because of very high energy efficiency.
- Main problem is programmability.

Transprecision

- Applications are traditionally written in single or double precision but many problems do not require such accuracy (esp Deep learning but also in other fields).
- The aim of projects such as Oprecomp, for example, is to apply transprecision principles in applications, runtime systems and hardware design to reduce energy consumption.



DEEP learning and Big Data

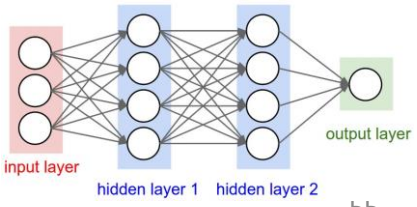
The Explosion in applications related to AI neural networks and Big Data, particularly in deep and machine learning, is having an important impact on our way of life.

The many applications include: speech and image recognition, robotics, automated driving, drug discovery, astrophysics, extreme weather events.

The huge economic importance is affecting HPC technology with the result that hardware vendors are making decisions based on the deep learning market:

- Intel is investing heavily in AI devices and software (perhaps also the reason for not continuing with KNL).
- Nvidia now produces GPUs with *tensor cores* dedicated to performing operations for deep learning (particularly matrix operations).

AI applications (e.g. TensorFlow) not traditionally highly parallel but given the long times needed for training neural networks (e.g. 3 days), HPC optimisations coming here.



Comparison of HPC centres and data centres

In common

- Need to minimise power consumption and maximise energy efficiency.
- Technology - the economic importance of deep learning is driving innovations in hardware which are being adopted by HPC.

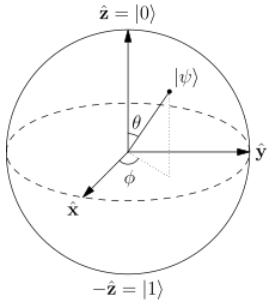
Differences

- Applications - HPC centres still concentrate on scientific research (rather than social networks ☺), although more deep learning projects being accepted.
- Data centres don't tend to use filesystems but rather virtual machines and dockers/containers which use non-volatile memories. (i.e. cloud computing).

And now for something completely different - Quantum Computing

- For many years a research area with little practical use but commercial machines are now available (D-Wave) or can be tried on request (e.g. IBM).
- Completely different principle to classical computing: you don't run a program but instead set-up a quantum system to emulate the problem you wish to study.
- Based on "Qubits", quantum entities which exist in a superposition of on (1) and off (0) states. The quantum system then evolves to give the solution to your problem.
- Problems that can be tackled by quantum computing tend to be optimisations or simulated annealing (quantum annealing), i.e. finding the "best" solution from many alternatives.
- Energy consumption is also modest, once the close to absolute zero temperatures have been attained.
- Not likely to replace classical computing, but could be used as an "accelerator" for specific problems or optimizations.

D-wave quantum computer



bloch sphere representing a qubit

IBM Q quantum computer

Summary

- HPC hardware has changed rapidly in a relative brief timeframe - from monolithic serial computers, to heterogeneous, massively parallel clusters with a wide range of different devices and memories.
- Further changes will occur, heavily influenced by the need for energy efficiency and fields such as Big Data and DEEP learning.
- The real problem though is software, which has lagged behind hardware. Application developers find it increasingly difficult to port legacy codes to modern hardware, particularly if the hardware is short-lived (e.g. bluegene or KNL) or rapidly evolving.
- New tools and paradigms are needed but the lack of standards means that only few reach critical mass to survive (e.g. MPI or OpenMP).

Last word

1965

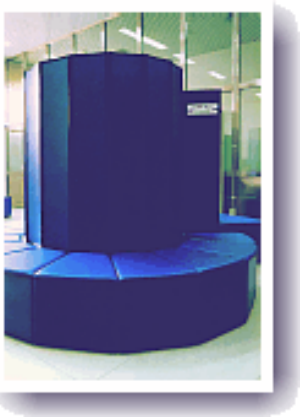
8Mb

2015

128Gb

STORAGE

1975



400 Mflops

2015

173 Gflops (GPU)

PERFORMANCE

1970

```
  PROGRAM HELLO
C
      REAL A(10,10)
      DO 50 I=1,10
        CALL DGEMM(N,10,I,J,A)
50    CONTINUE
```

2018

```
  PROGRAM HELLO
C
      REAL A(10,10)
      DO 50 I=1,10
        CALL DGEMM(N,10,I,J,A)
50    CONTINUE
```

SOFTWARE