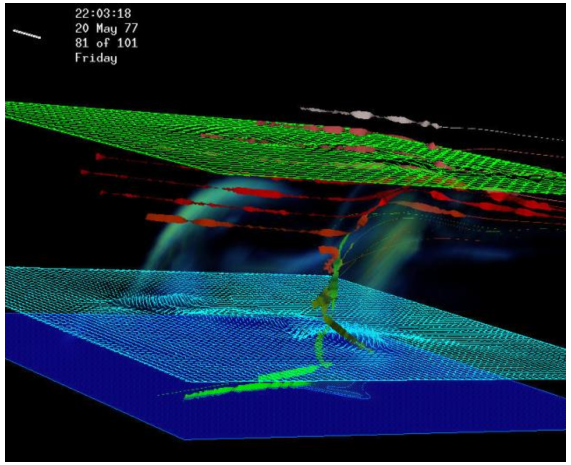
Introduction to High Performance Computing

A. Emerson, and many others

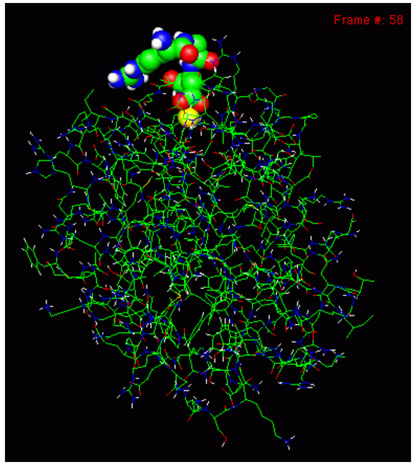
Computers and Computational Science

Past, Present and Future

HPC allows us to do computational science



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



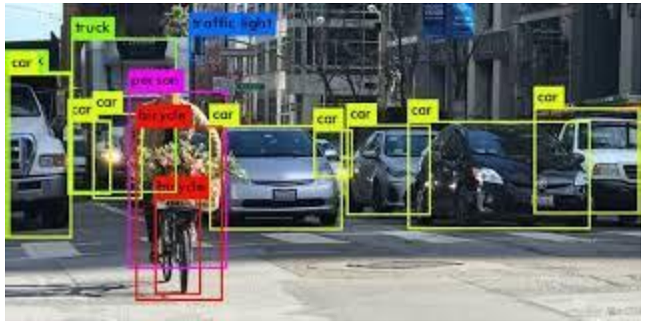
The use of computers to study physical systems allows us explore phenomena at all scales:

- **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)*

In more recent years

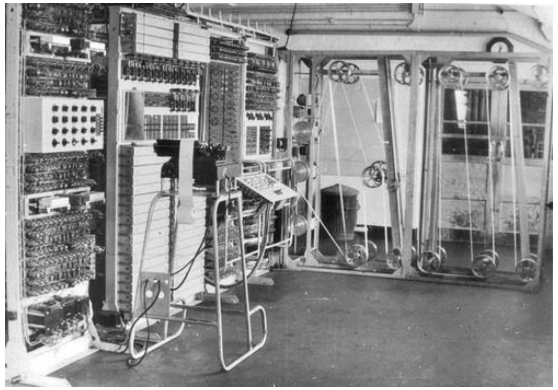
Digital Twins

Movies and Games

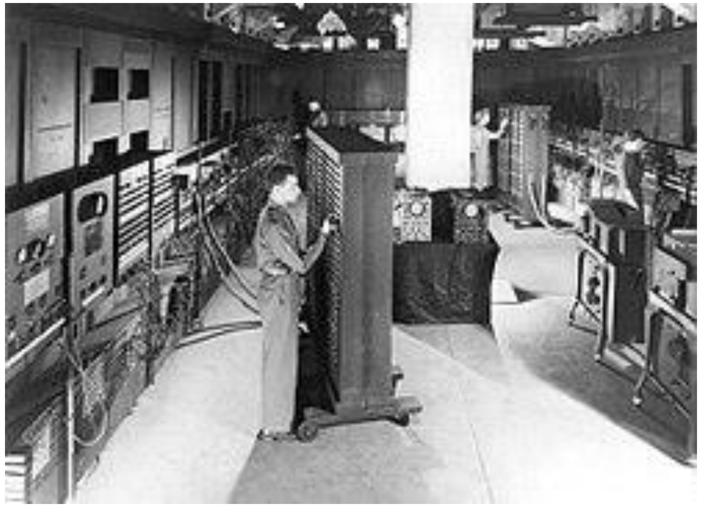


Machine learning

The first computers



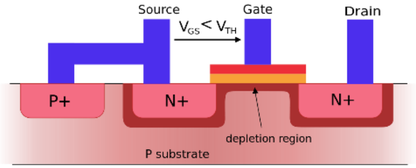
COLUSSUS, Bletchley Park, UK first programmable computer(1943-1945)



ENIAC, U. Penn. (USA) - first electronic computer (1945)

The first computers – transistors, integrated circuits and parallelism

Seymour Cray, founder of Control Data Corporation (CDC), designed one of the first computers to use transistors (1964, CDC 6600 - the first *supercomputer*)



transistors (esp. MOSFETs) and integrated circuits have revolutionized computers

cray -2
Seymour Cray pioneered *parallelism* to push the boundaries of computing

For at least 15 ***years core memories****, made of ferrite cores linked together, were the most used form of memory. We still use in UNIX a* ***core dump****.*

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.*

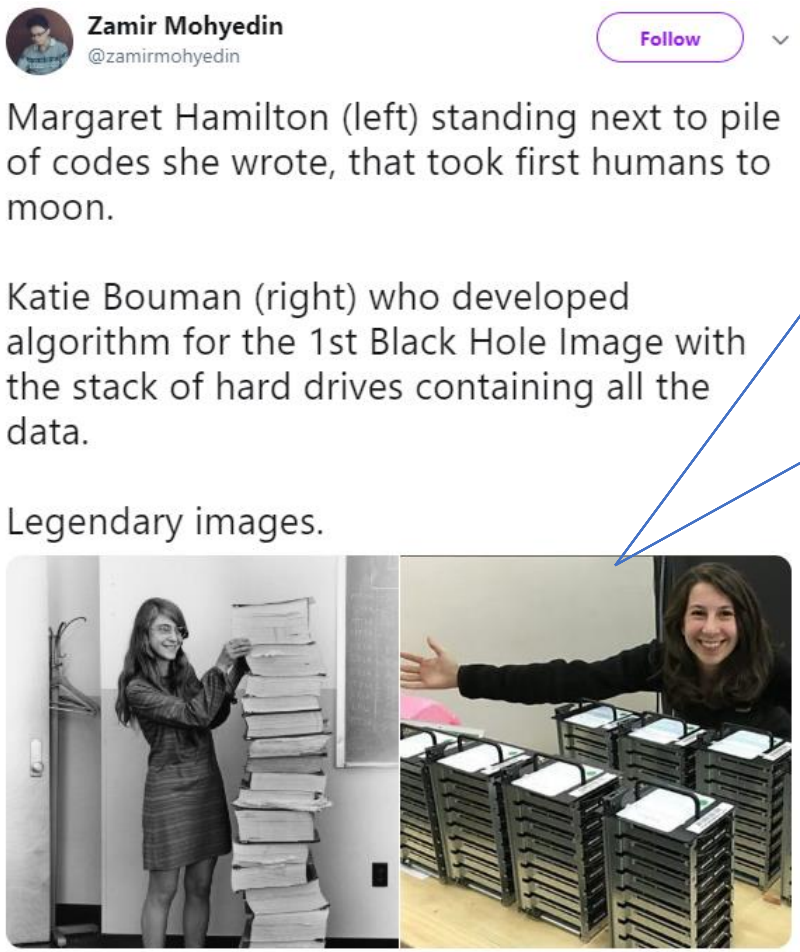
John Von Neumann, *1920s*

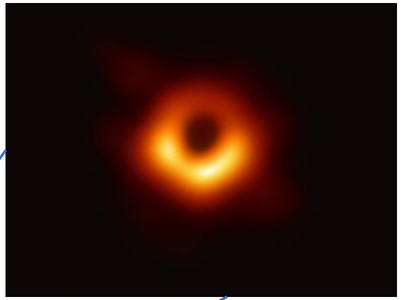
polymath and computer pioneer

Alan Turing (1912-1954)

The father of theoretical computer science

Pioneers-2



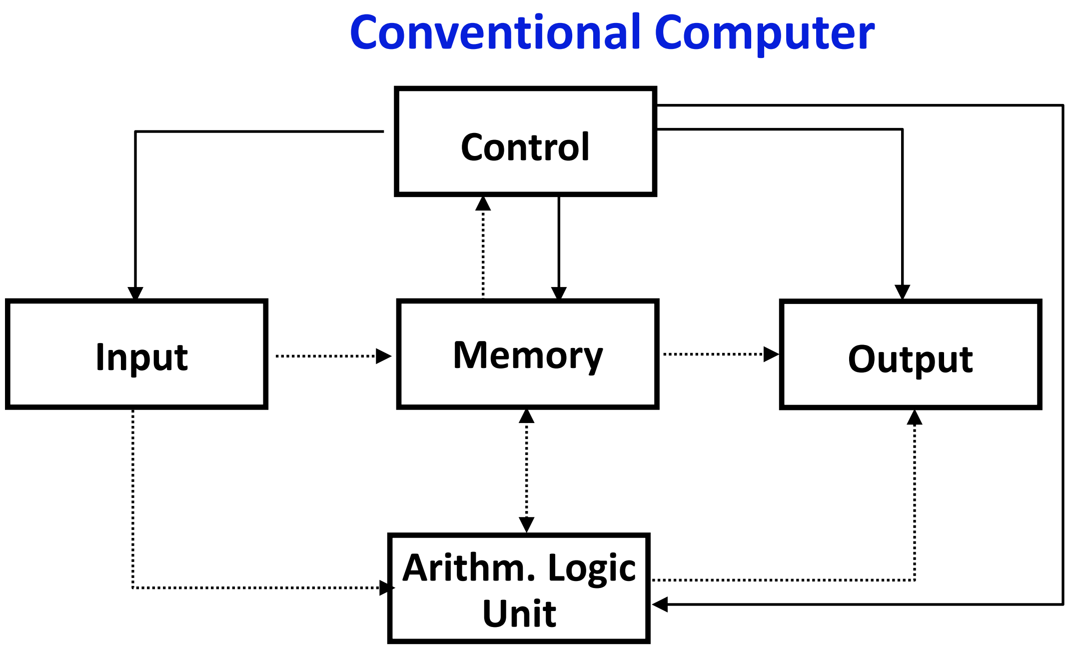


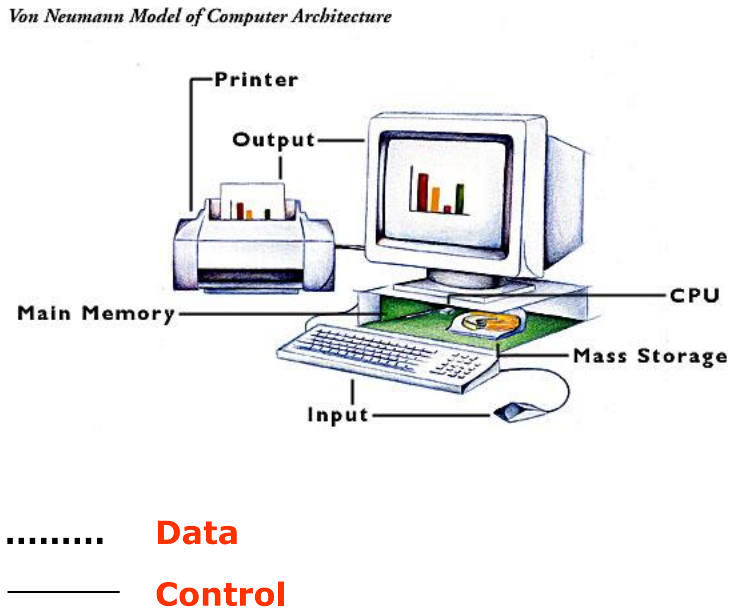
Margaret Hamilton was the first person to use the term *software engineering*

Grace Hopper

Her work led to COBOL, one of the first computer languages

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





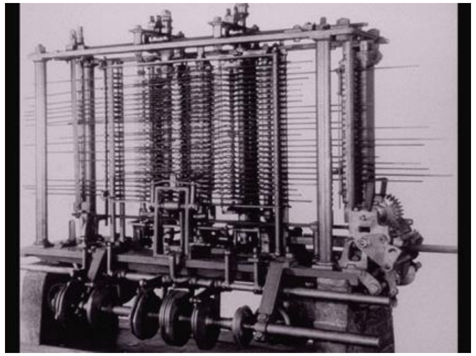
Instructions are processed sequentially

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**).

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*"..

NewYork World,March 1920

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

Processor speed, clock cycle and frequency

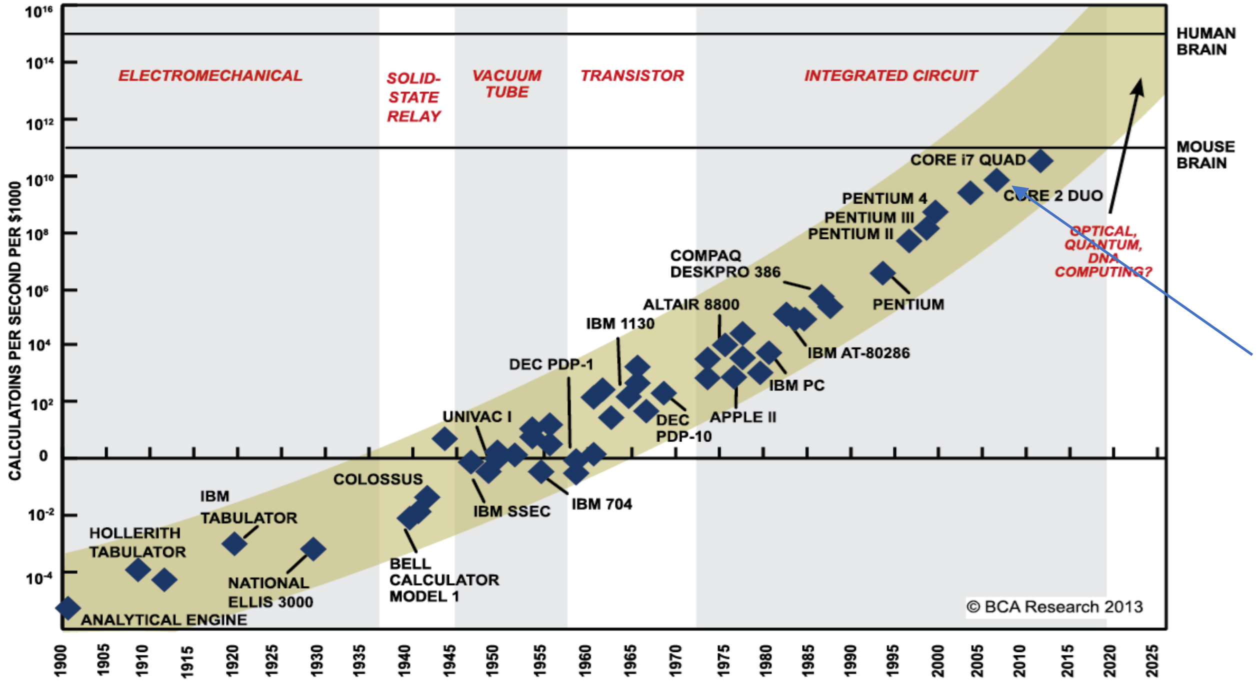
- The instructions of all modern processors need to be *synchronised* with a timer or *clock*.
- The *clock cycle* τ is defined as the time between two adjacent pulses of oscillator that sets the time of the processor.
- The number of these pulses per second is known as clock speed or clock frequency.

Processor (bits)	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 (32)	< 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
Intel Skylake (64)	0.47	2.1 GHz

Highest frequency chip used at CINECA

Limits of clock frequency:
- Power consumption
- Heat dissipation
- Speed of light
- Cost

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?

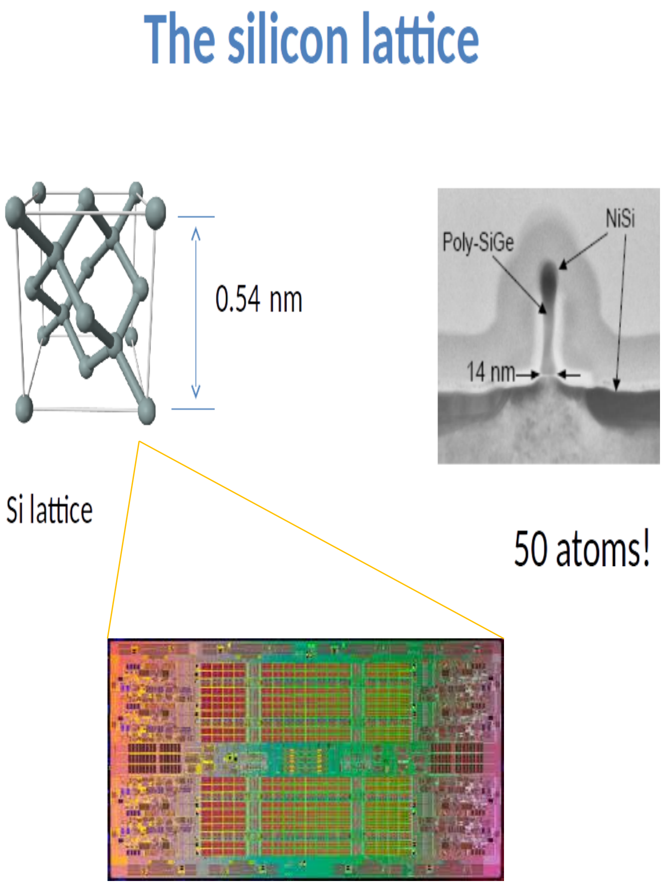
Some debate as to whether Moore's Law still holds but will undeniably fail for the following reasons:

- Minimum transistor size
    - Transistors cannot be smaller than single atoms.

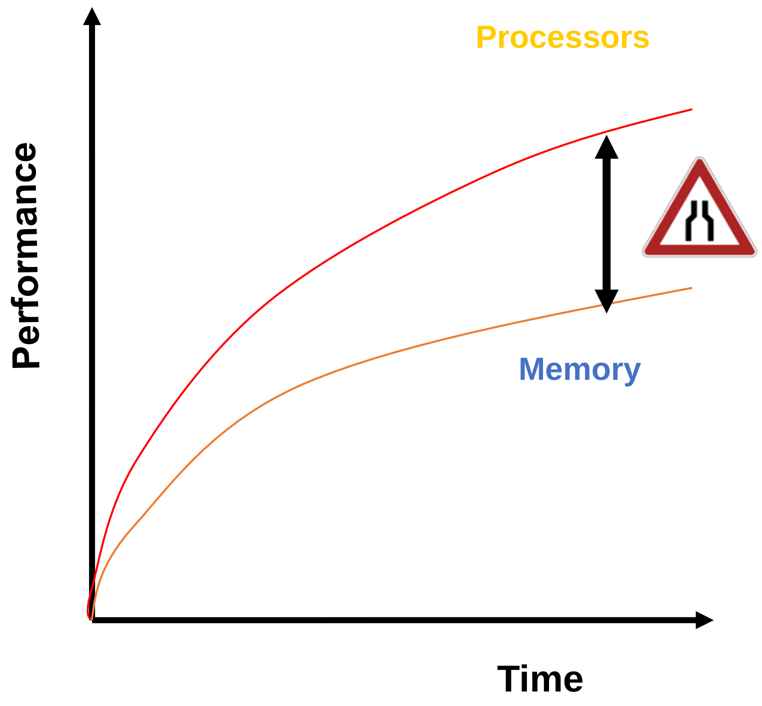
    (10-14nm feature sizes are common)
- Quantum tunnelling
    - Quantum effects (e.g. tunnelling) can cause current leakage.
- Heat dissipation and power consumption
    - Difficult to remove heat.

Increase in transistor numbers ≠ faster programs !

Software usually struggles to make use of the available hardware threads.



The processor - memory bottleneck and cache

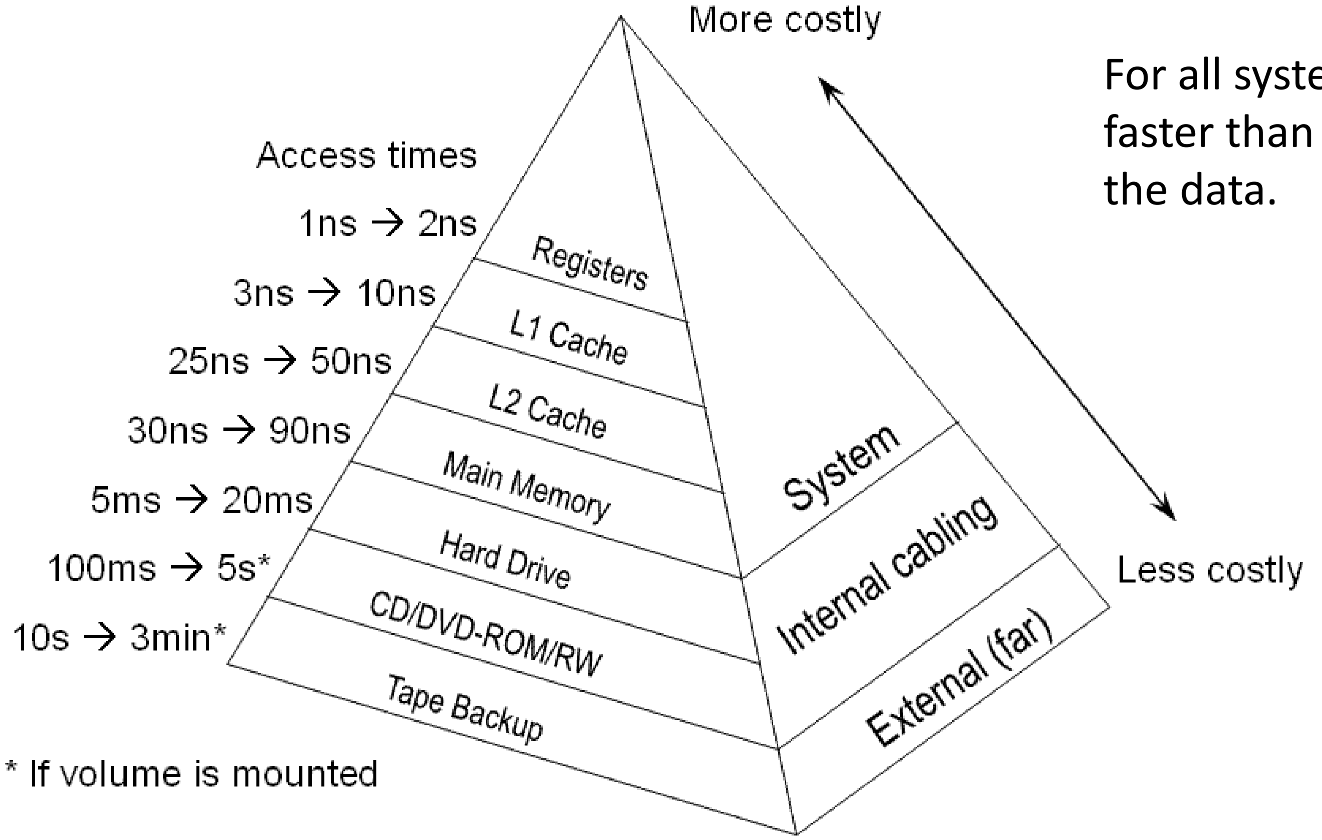


- The real limitation in HPC is the performance difference between processors and getting data to/from memory which has been increasing in time.
- Very important to minimise the time it takes to transfer data to and from the CPU.

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.

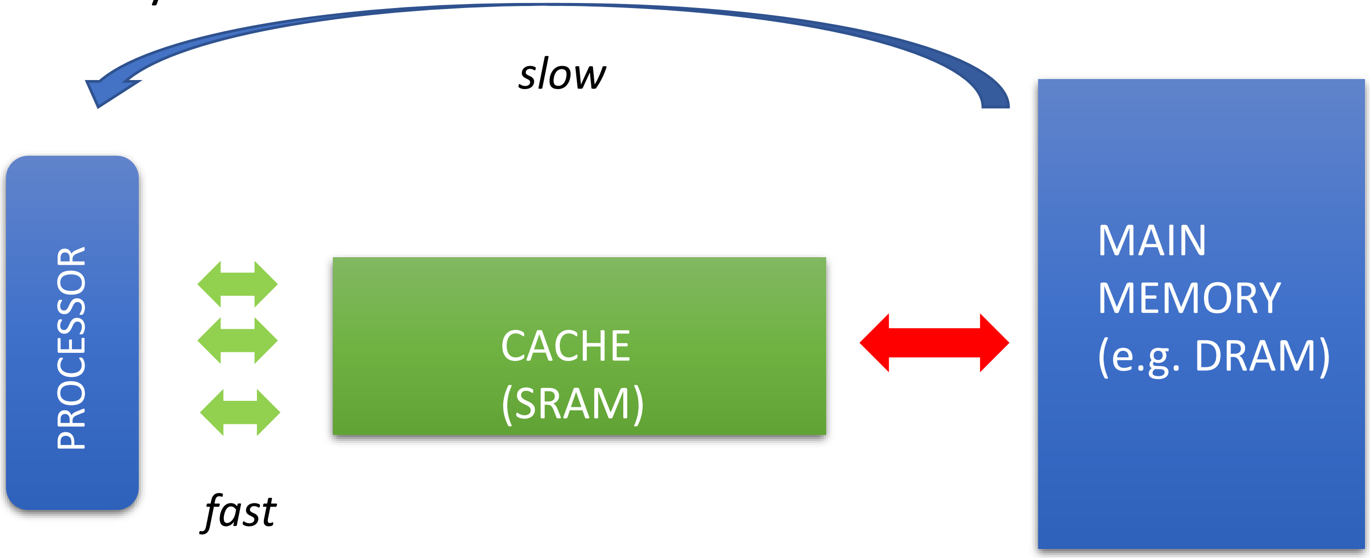
Memory Hierarchy



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.



General strategy:

- check cache before main memory
- if not in cache then similar data from main memory are loaded in the hope that the next data access will be from the cache (*cache hit*) and not from main memory (*cache miss*).

Parallel Computing

Concepts of Parallelism

serial computing is too slow for HPC

Must introduce *parallelism* :

- Instruction level (e.g. fma = fused multiply and add).
- SIMD or 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.

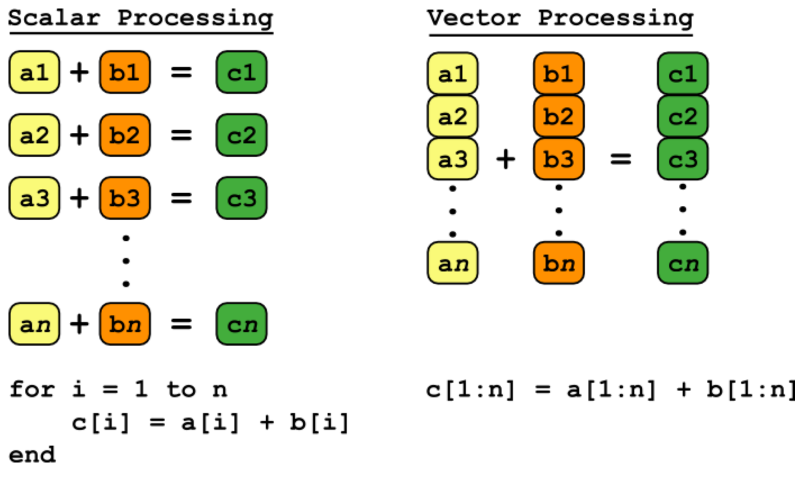
Vectorisation

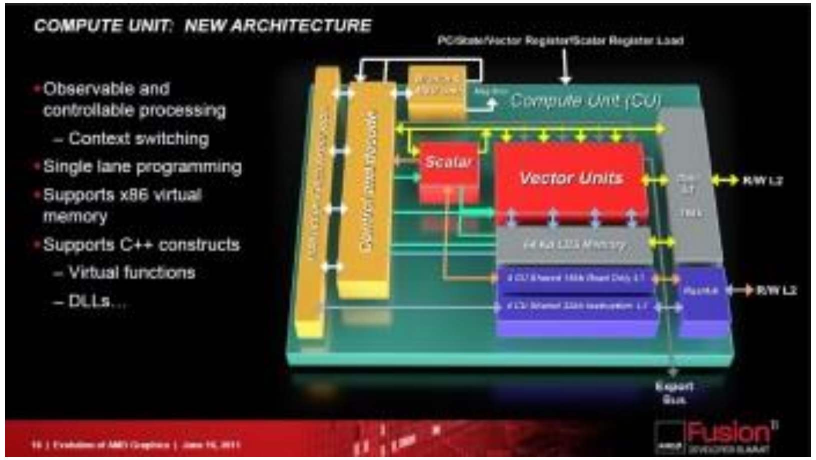
Definition

- Single instruction performed on multiple data elements

How it works

- Compiler looks for loops in code
- If possible, generates vector instructions
- Vector instructions sent to SIMD unit at run-time.
- Well-vectorised loops may become 2,4,8X faster (depends on vector bit width)

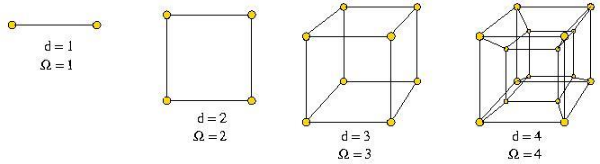




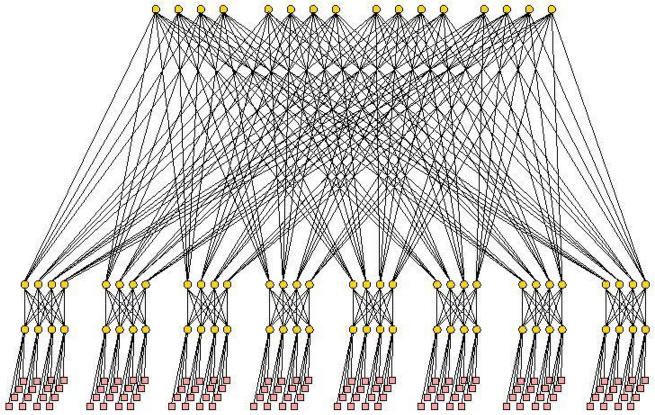
SIMD = Single Instruction Multiple Data

Network technologies and topologies

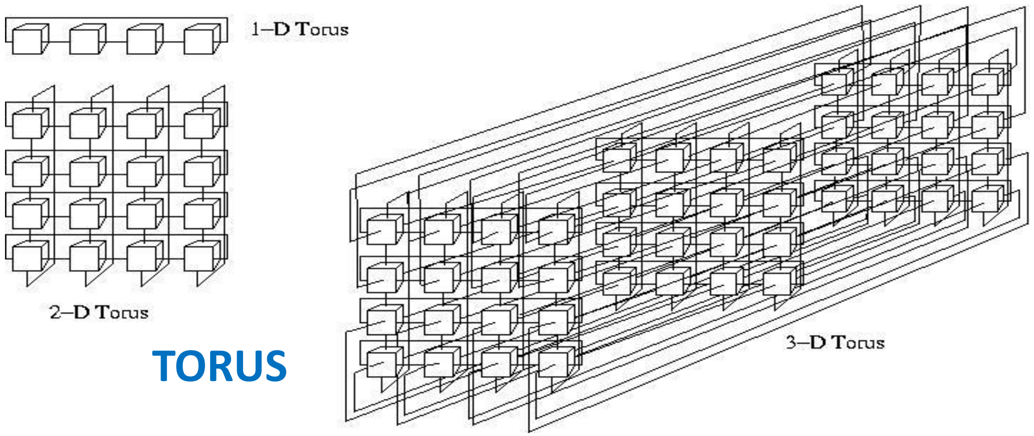
- Networks linking the nodes in a distributed system are available according to price, performance and hardware vendor.
- Examples include: Ethernet, gigabit, Infiniband, Omnipath (Intel), etc.
- If switches are used then the network is often called the *fabric*.
- In addition, networks can be configured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



(a) Hypercubes, dimension 1-4.



(b) A 128-way fat tree.

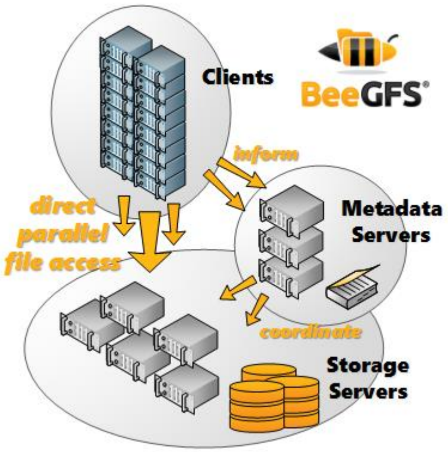


TORUS

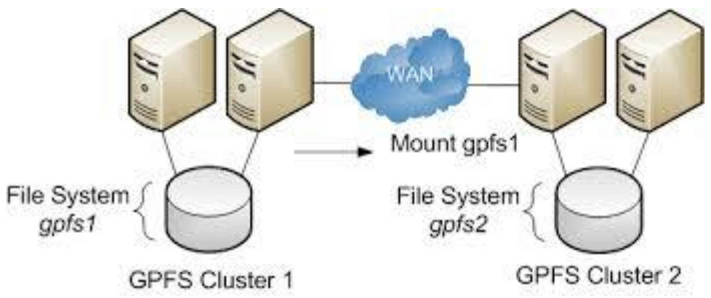
FAT TREE

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.



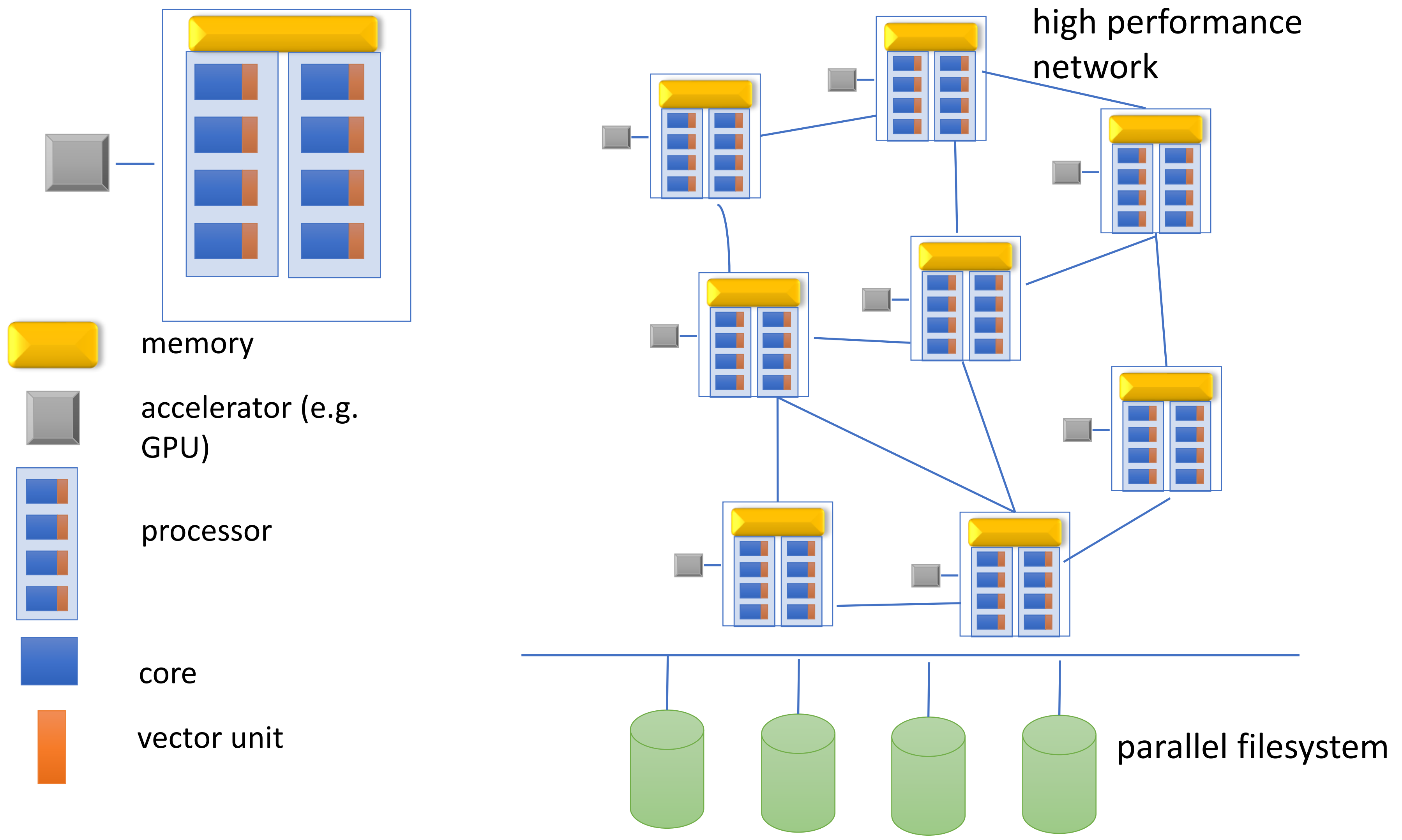
BeeGFS



GPFS

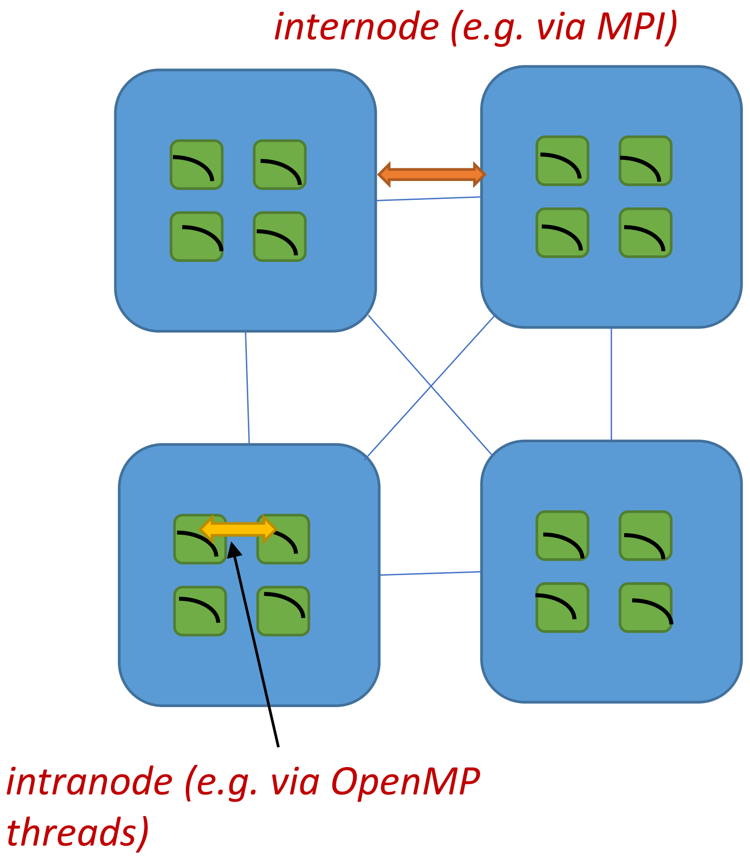
Parallel filesystems like GPFS and LUSTRE perform best with few, large files rather than many small files

Parallel computers - putting it all together



Programming parallel computers

- We need libraries, tools, language extensions, algorithms and paradigms which allow us to:
  - exploit within a node vector and cache units, hardware, shared memory;
  - manage inter-node connections to exchange data with processes on other nodes
  - debug and profile programs to check correctness of results and performance
  - use appropriately the disk space.
- The languages most used for parallel programming have been FORTRAN and C/C++ -- not originally designed for parallelism.
- Also commonly used include Python MPI, CUDA (GPUs).



Parallel programming

More automatic parallelisation → avoid code rewrites and access heterogenous devices (CPUs, GPUs, FPGAs,)

MPI

```
void initialise(double**, double**, int, int, int);
double* allocate_matrix_as_array(int nrows, int ncols);
double** allocate_matrix(int nrows, int ncols, double* arr_

int main(int argc, char * argv[]) {
        int size, myrank;

        MPI_Init(&argc, &argv);
        MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
        MPI_Comm_size(MPI_COMM_WORLD, &size);

        if (argc != 3) {
                if (myrank==0) fprintf(stderr, "You must pr
);
                return -1;
        }
```

Most difficult - parallelism must be explicitly programmed

SYCL, oneAPI, etc

```
// Kernel
parallel_for(count, kernel_functor([ = ](id<> item) {
  int i = item.get_global(0);
  r[i] = a[i] + b[i] + c[i];
}));
});
```

heterogeneous programming

CUDA

```
__global__
void saxpy(int n, float a, float *x, float *y)
{
  int i = blockIdx.x*blockDim.x + threadIdx.x;
  if (i < n) y[i] = a*x[i] + y[i];
}
```

Implicit parallelism but low-level and NVIDIA gpu only

OpenMP

```
#pragma omp parallel for shared(m, n, Anew, A)
for( int j = 1; j < n-1; j++) {
    for( int i = 1; i < m-1; i++ ) {
        Anew[j][i] = 0.25f * ( A[j][i+1] + A[j][i-1]
                             + A[j-1][i] + A[j+1][i]);
        error = fmaxf( error, fabsf(Anew[j][i]-A[j][i]));
    }
}
```

Annotate sections of code to parallelise

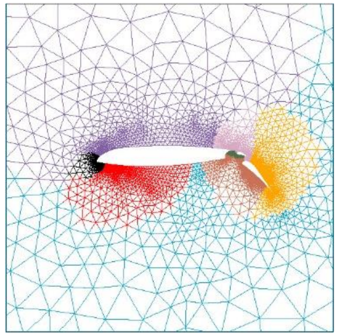
OpenAcc

```
#pragma acc kernels
  for( int j = 1; j < n-1; j++)
  {
  for( int i = 1; i < m-1; i++ )
  {
A[j][i] = Anew[j][i];
  }
  }
```

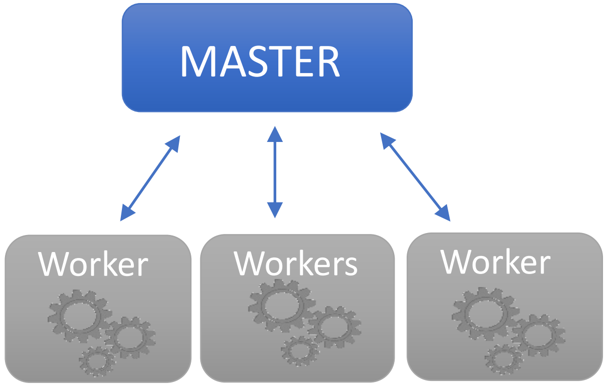
Similar to OpenMP but more automatic parallelism

Parallel algorithms

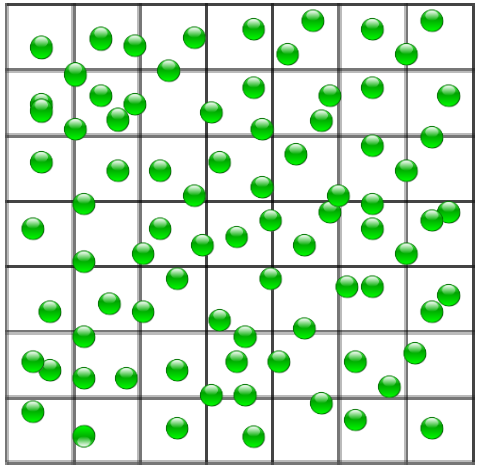
We have the parallel hardware and software libraries but we also need to map our problem into a parallel algorithm.



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

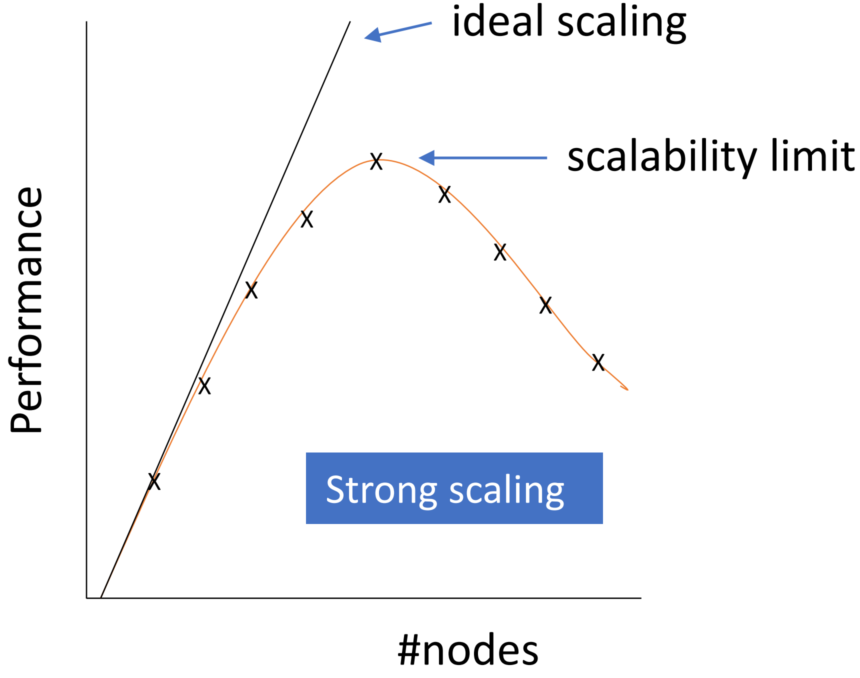


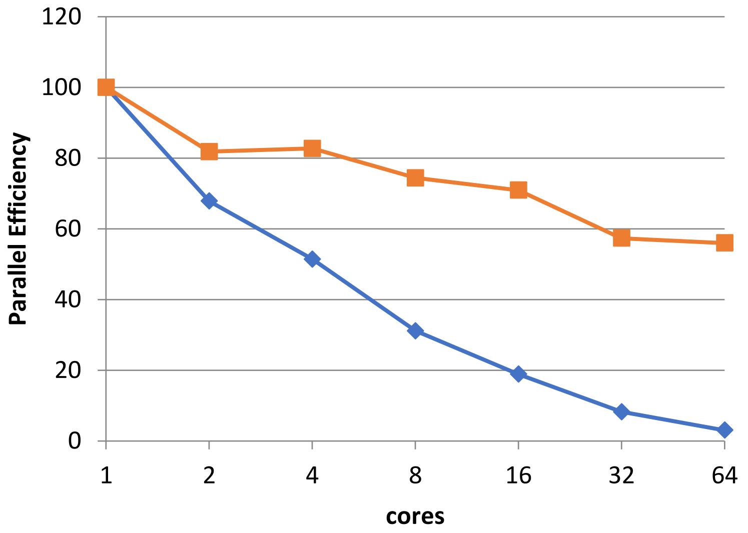
master-slave (e.g. data analysis).



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

Parallel scaling and efficiency





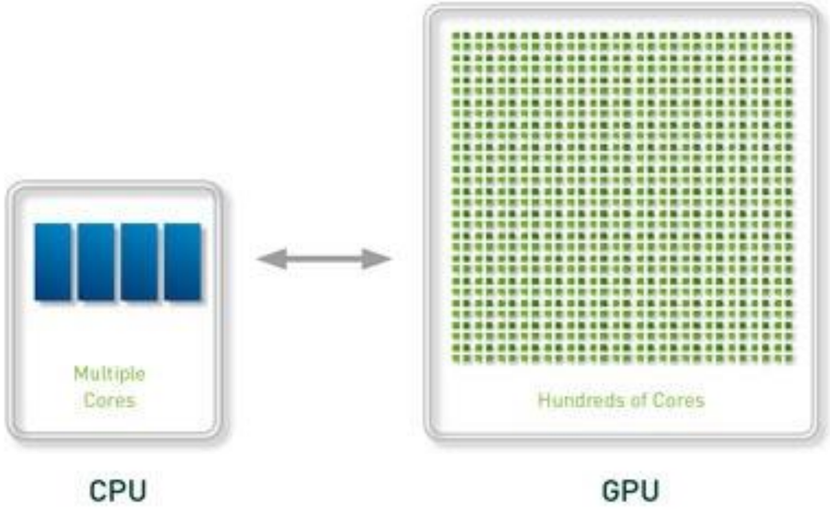
$$S = 100 \times \frac{P_N}{N \times P_1}$$

Parallel Scaling tests performance with increasing resources

Parallel Efficiency provides an alternative measure of parallelism.

Really important to do scaling tests before going into production

NVIDIA GPUs



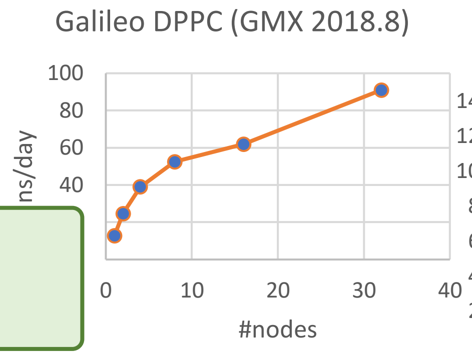
- Originally only PC video cards
- Hardware and software (CUDA) updates allows use as HPC accelerator
- GPU has many streaming SM (symmetric multiprocessor) cores for parallelism.
- Large speedups c.f. CPUs are possible
- May be limited by memory (e.g 16Gb) and bandwidth (if using PCIe express)

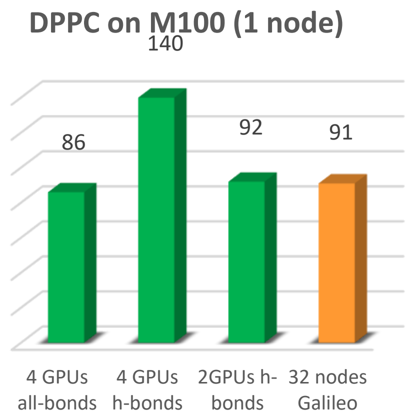
NVIDIA GPUs - the story continues

NVIDIA are market leaders but also:

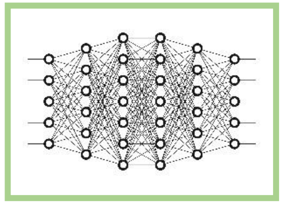
- AMD
- Intel GPU

code acceleration





Low energy per Watt, high performance/$$



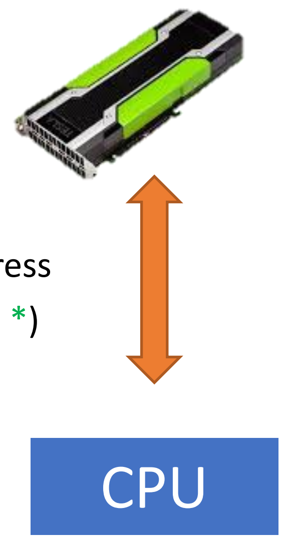
GPUs are ideal for machine learning:

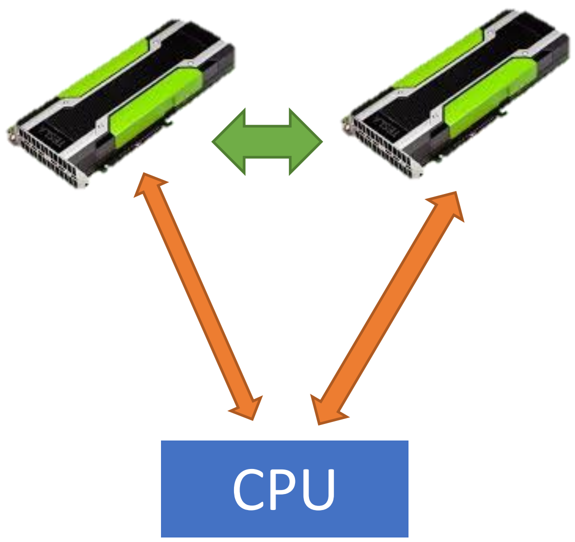
- high performance for linear algebra
- hardware support for low floating point precisions (16 and 8 bit).

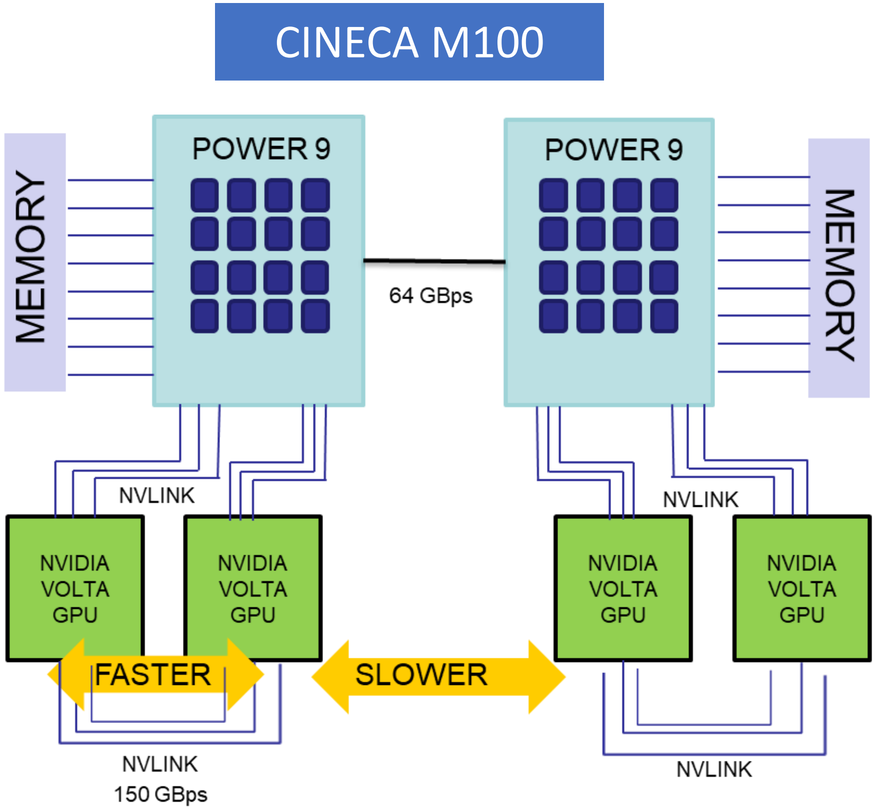
CPU-GPU connections

CPU – GPU communication is a potential bottleneck

PCI Express
(16Gb/s *)







* Assumes PCIe 3 with Nvidia V100

CPU – GPU via PCI Express ("**offloading**")

GPU Direct RDMA avoids GPU-GPU comm. via CPU memory (usable in MPI)

NVLINK 2.0 provides fast CPU-GPU and GPU-GPU communication (150 Gb/s)

State of the Art

Top500 June 2022 – First list with an Exascale machine*

1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, **HPE** DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, **Fujitsu** RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, **HPE** EuroHPC/CSC Finland	1,110,144	151.90	214.35	2,942
4	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, **IBM** DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
5	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, **IBM / NVIDIA / Mellanox** DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438

AMD GPUs !

Frontier – world's first exascale computer

ORNL – Oak Ridge National Laboratory

Linpack performance	1.1 exaflops
Mixed precision (HPL-AI)	6.88 exaflops
# nodes	9,400
Each node contains	1 EPYC proc and 4 AMD Instinct GPUs
Storage	700 Pb
Power	**40 Mw** (water cooled)

Enough to power 30,000 US homes !

Pre-exascale computers in Europe (Euro HPC)

Leonardo, 200+ Pflops (Atos/Nvidia), Bologna, Italy

Pre-Exascale resources provided by EuroHPC Joint Undertaking.

Mare Nostrum5, 200 Pflops, Barcelona, Spain

LUMI, 150+ Pflops (HPE Cray), Finland

TOP500 November 2022

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,463,616	174.70	255.75	5,610
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096

Europe in 3rd and 4th position

TOP500 November 2023

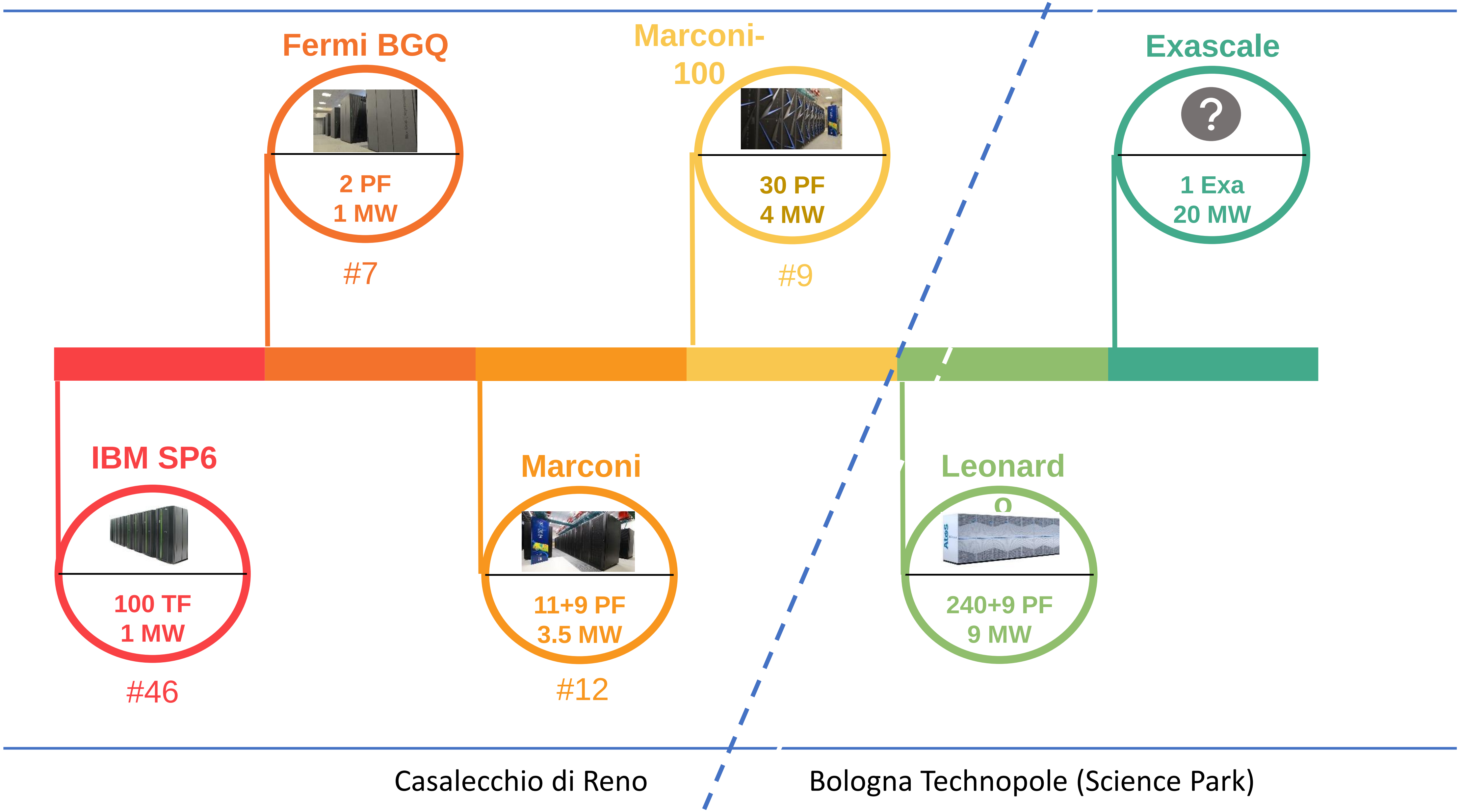
In the latest list, Lumi and Leonardo drop to 5th and 6th.

The Intel Aurora takes 2nd position with Intel Data GPU Max system

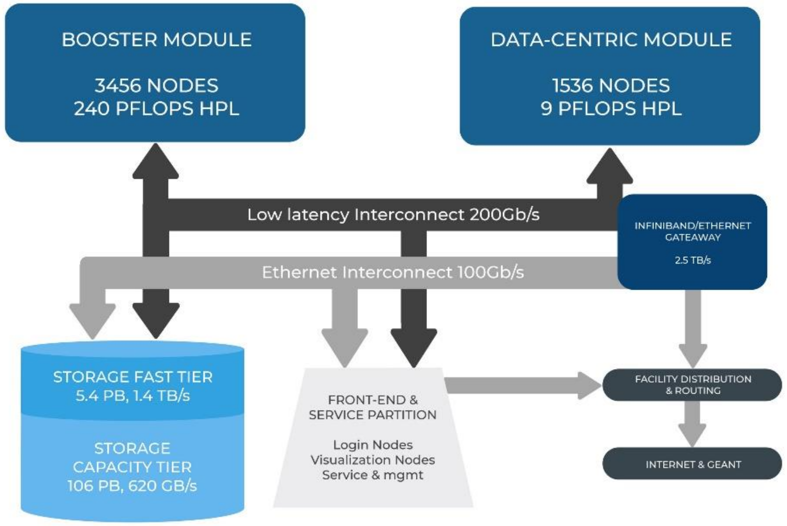
Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	4,742,808	585.34	1,059.33	24,687
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure United States	1,123,200	561.20	846.84	
4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107
6	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, EVIDEN EuroHPC/CINECA Italy	1,824,768	238.70	304.47	7,404

NVIDIA H100 “Grace Hopper” GPU

CINECA Tier-0 Roadmap



Leonardo in detail



Booster Module

- 3456 nodes consisting of 4xNVIDIA **A100** GPUs and 1 x32-core Intel Ice Lake CPU
- 512 GB RAM/node
- 89.4 TFLOPs peak perf per node

Data Centric Module

- 1536 nodes with two Intel Sapphire Rapids CPU/node (40 cores*)

Network

- Infiniband network of 200 Gb/s
- Dragonfly topology

Data Storage

- Over 200 PB total storage
- Fast Tier - 24 x 7,68 TB SSD NVMe with encryption support
- Capacity Tier -82 x 18 TB HDD

Comment on Power requirements

*«L'energia è fondamentale per l'attività di Cineca, e rappresenta un impegno notevole e costante dal punto di vista economico, tecnico ed ambientale. Attualmente Cineca assorbe circa **38 GWh**/anno di energia elettrica, che se volessimo tentare di quantificare può essere paragonato al consumo energetico medio di una cittadina di circa 40.000 abitanti»* Ufficio Tecnico del Cineca (2021)

"Energy is fundamental to Cineca's business and represents a significant and constant commitment from an economic, technical and environmental point of view. Cineca currently absorbs about **38 GWh** / year of electricity, which, if we wanted to quantify it, can be compared to the average energy consumption of a town of about 40,000 inhabitants." Cineca's Engineering department (2021).

- Energy consumption is a real challenge for supercomputing and influences strongly their design and use.
- Current solutions include:
  - Energy efficient, multi-core processors such as ARM.
  - Accelerators such as GPUs (esp. NVIDIA).
  - Novel cooling systems for computer centres (e.g. warm water instead of air cooling).
  - Job monitoring and scheduling based on predicted energy consumption.
  - Reduced precision (esp. for machine learning).
  - .. and others

Artificial Intelligence and Deep Learning

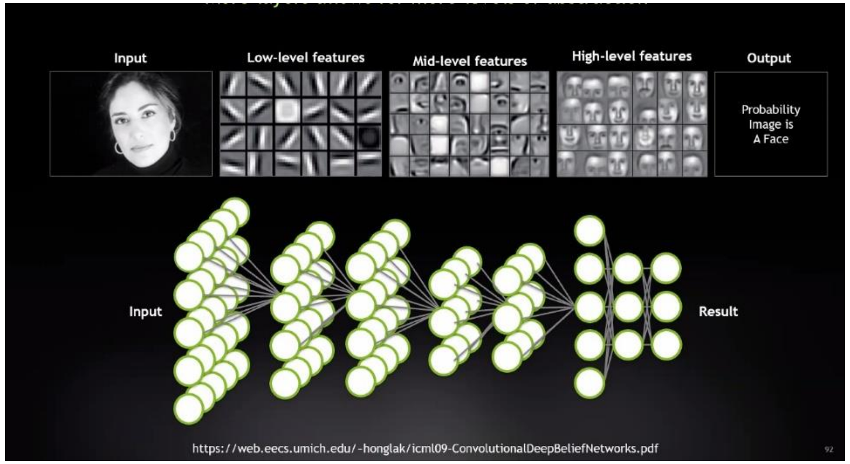
AI and Machine or Deep Learning is a major driver in modern HPC*:

- Nvidia Tensor cores. Hardware designed for multiplication of 4x4 matrices (tensors).
- Transprecision units. AI usually does not need full 32 bit floating point precision.
- Google TPU for tensor processing.
- Near-memory or in-memory computing. Processing where data resides – avoid costly data transfers.

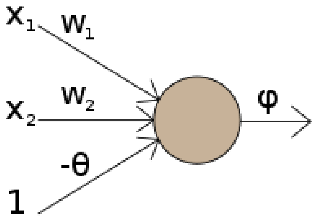
As well as libraries and software tools.

Edge computing: perform computing away from data centre to reduce data transmission. Expected to increase as a result of IoT.

** Deep learning is basically matrix multiplication and sums.*



convolutional neural network (CNN) for face recognition



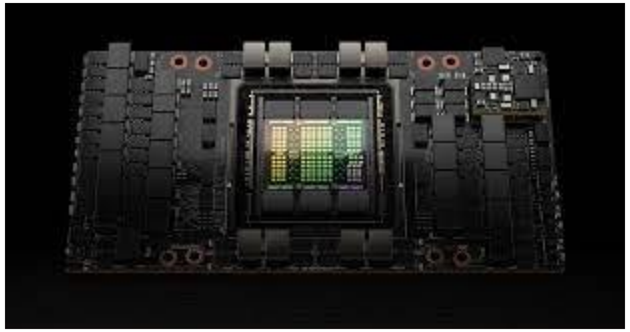
Neuron in deep learning

AI + NVIDIA: *The technology giant reported that revenues surged by 265% to $22bn (£17.4bn) in the three months to 28 January, compared to a year earlier* **(BBC News 22/02/2024)**

The memory bandwidth problem…

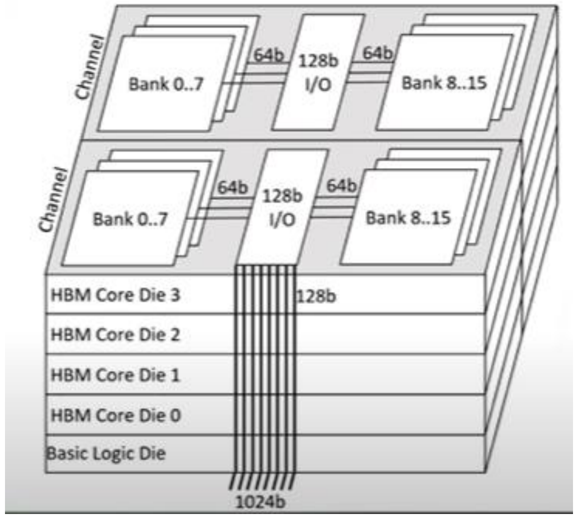
A main challenge of HPC is memory access – programs require a lot of data, but memory is relatively slow.

Good programming of cache helps, but AI has re-awakened the market for *High Bandwidth Memory* or HBM.



NVIDIA H100 has 80Gb of HBM3 at 3.3 TB/s

HBM has up to 8 layers of DRAM memory with channels linking the layers Can also include a logic layer.



3D stacked chips require less power and space, but are harder to manufacture.
Market worth up to $2bn in 2023.

Using HPC

Using HPC resources

The HPC user interface has not changed much over the years

Linux (UNIX) operating system

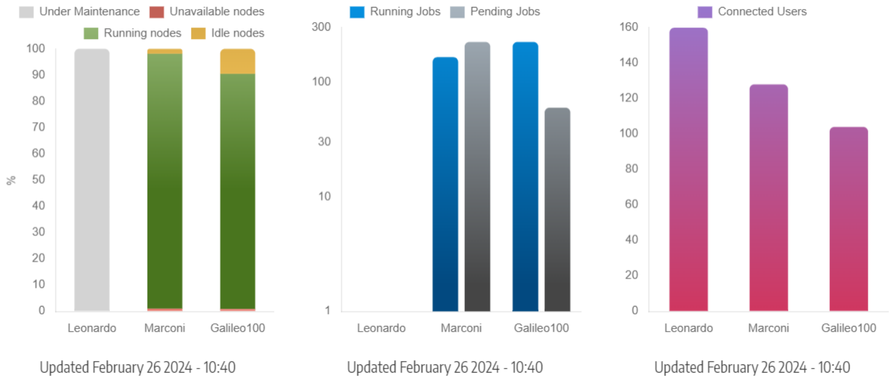
SSH protocol for access

```
(base) [aemerson@login07 aemerson]$ cd $PU
(base) [aemerson@login07 aemerson]$ ls
AmberTools23.tar.bz2  boost_1_84_0.tar.gz
amber-22              build_AmberTools.sh
amber22_src           cadd
bison                 fftw
boost-1.8.4           gmx-2023.1
boost_1_84_0          gmx-2023.2
(base) [aemerson@login07 aemerson]$
```

Command Line Interface

Batch scheduler for job management

Many centres have facilities for portals, remote visualization, workflow systems, etc, but few are standard.



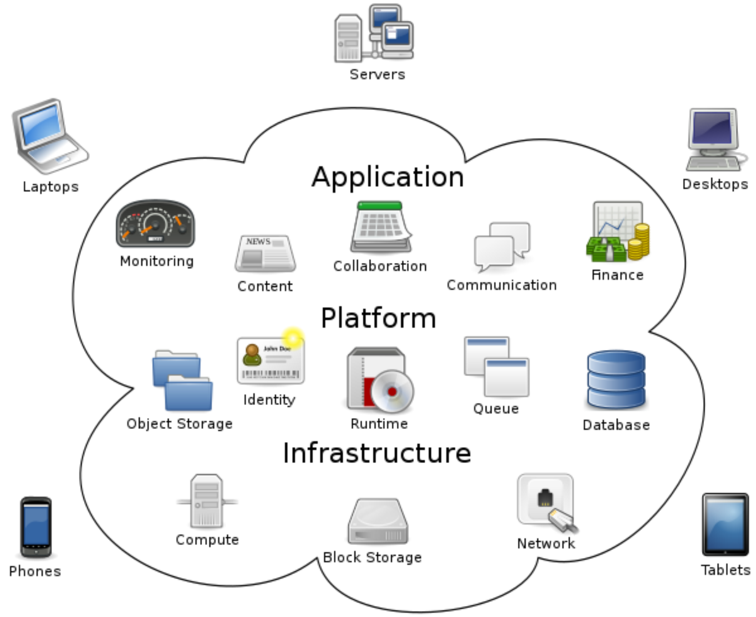
web-based summaries of system status are popular with users

Cloud Computing

"Nobody understands the cloud, it is a mystery"

(*Sex tape [2014],with Cameron Diaz and Jason Segel*)

Cloud computing – *on-demand availability of computer system resources, esp data storage and computer power, without direct active management by the user.* (Wikipedia)

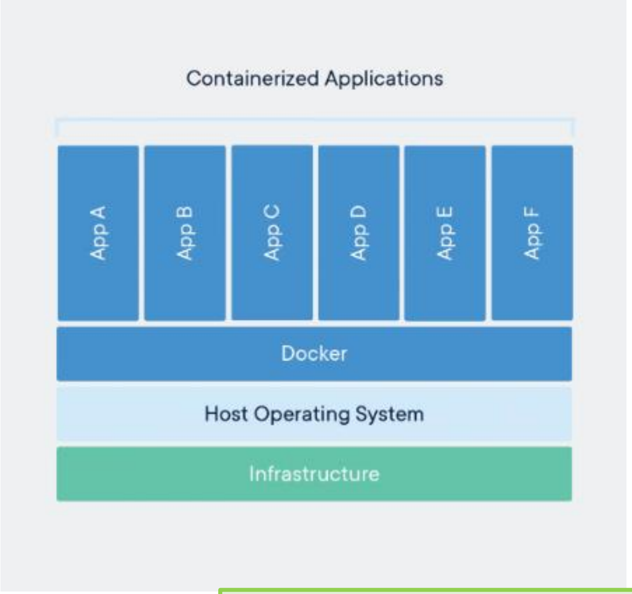


Popular with commercial companies and other users who do not wish to maintain their own computer services.

BUT

- Users must trust the provider with potentially sensitive data
- Cloud users may not have full control on how the resources are provided.

Containers




```
singularity run
docker://godlovedc/lolcow
INFO:    Converting OCI blobs to SIF
format
INFO:    Starting build...
Getting image source signature
Copying ….
…
INFO:    Creating SIF file...
INFO:    Build complete: cachest.sif
INFO:    Image cached as SIF
Output:
```

Often a big problem to maintain the dependencies (e.g. libraries) of a particular application or tool.

One solution is to prepare a ***container*** which provides a self-consistent environment containing operating system, libraries and applications.

The container can be created or downloaded, and in principle used on any hardware with compatible chip architecture (e.g. x86).

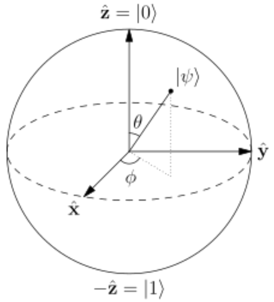
Very common type of container is ***docker.*** Since docker needs root privileges, on HPC tend to use instead **singularity** (which can manage docker containers)

Very convenient, but use over multiple nodes is complex

Quantum Computing

- For many years a research area but commercial machines are now available (D-wave, IBM Q, Pasqal, etc).
- Based on **Qubits**, quantum entities which exist in a superposition of on (1) and off (0) states.
- For *n* qubits, **2^n units** of information can be stored.
- Qubits can be **entangled** – operating on one qubit affects the other entangled qubits.
- Uses include: optimization, cryptography and machine learning.
- Low energy consumption, although requires cryogenic cooling.
- Will probably be used as an accelerator in HPC.

D-wave quantum computer



bloch sphere representing a qubit

IBM Q quantum computer

An example of non-Von Neumann computing

Summary

Rapid evolution in HPC:

- monolithic single processor → many processors → multicore → multicore + devices

Energy consumption and the commercial importance of machine learning are main drivers for hardware design.

Heterogenous architecture complicates programming:

- → increasing use of directive-based programming for offloading to GPUs, FPGA's and other devices (e.g. OpenMP, OpenAcc).
- For performance need asynchronous models.

AI and Deep Learning increasingly important

Quantum computing still needs a few years (?) → capabilities limited by #Qubits and noise but **requires low power.**

CDC 600 1964 (3 Mflops)

Frontier 2022 (1.1 Exaflop)