

Foundations of High Performance Computing

Lecture 2: HPC hardware&software

“High Performance Computing”



2022-2023 Stefano Cozzini

Agenda

HPC is parallel

Serial Computers

Moore law/Dennard Scaling

Parallel computers

HPC infrastructure

ORFEO HPC infrastructure

HPC

=

PARALLEL
COMPUTING

HPC

=

PARALLEL

COMPUTERS

Agenda

Why HPC is parallel ?



Serial Computers

Moore law/Dennard Scaling

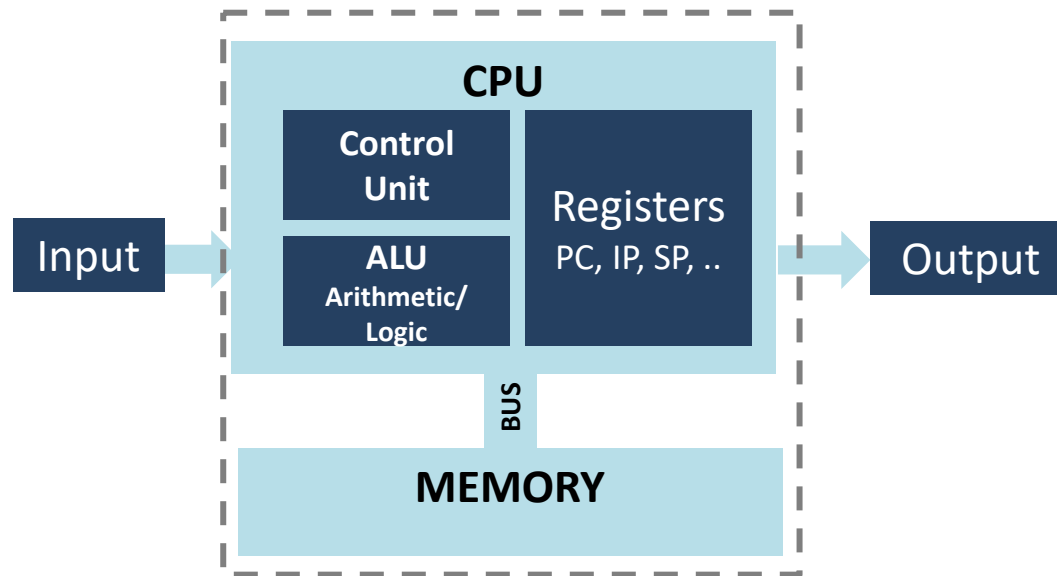
Parallel computers

HPC infrastructure

ORFEO HPC infrastructure

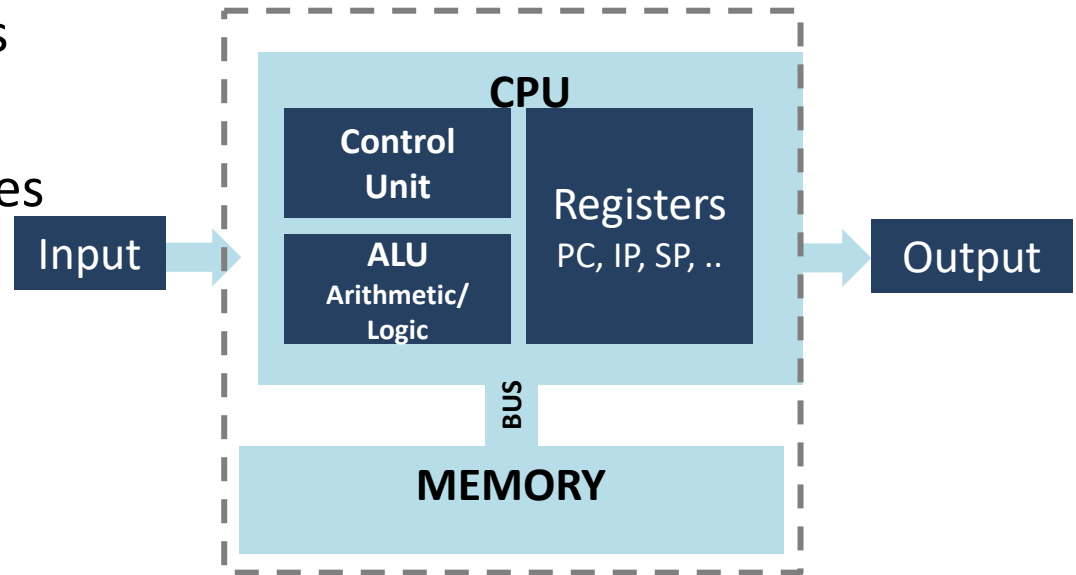
What is a serial computer ?

- Von Neumann architecture (the fundamental model)



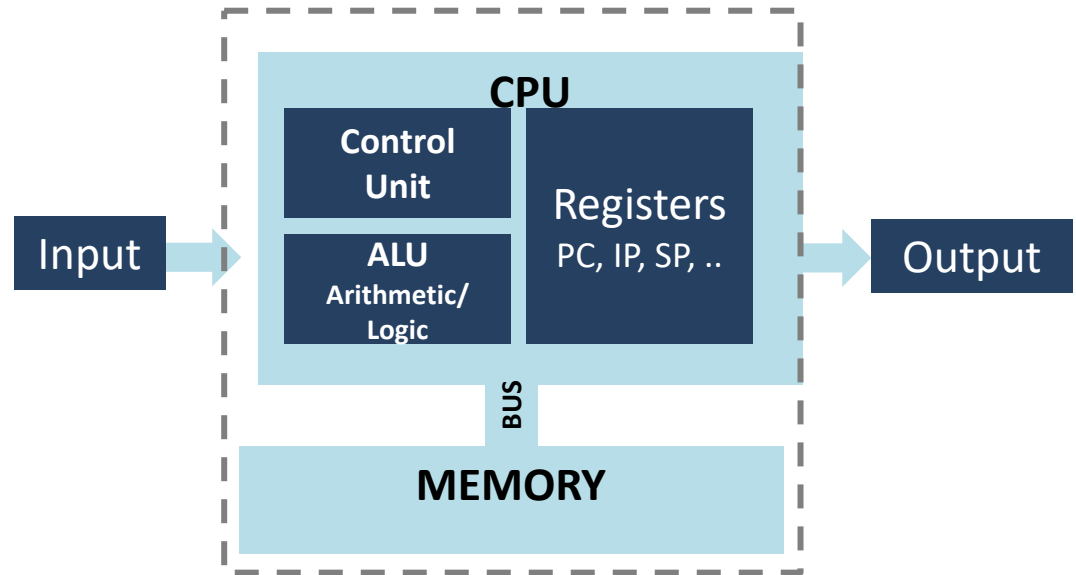
Von Neumann architecture:

- There is only one process unit (CPU)
 - Control Unit: processes instructions
 - ALU: math and logic operations
 - Register: store data



Von Neumann architecture:

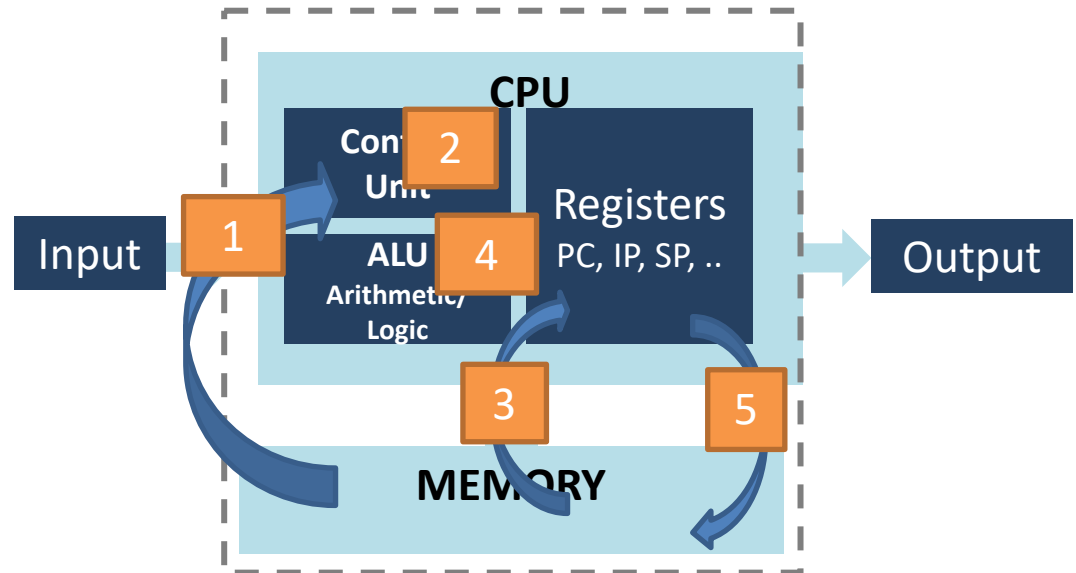
- 1 instructions is executed at a time
- memory is “flat”:
 - access on any location has always the same cost
 - access to memory has the same cost than op execution



Von Neumann architecture:

5 step WORKFLOW:

1. instruction fetch
 2. Instruction decode:
determine operation and
operands
 3. Memory fetch: Get
operands from memory
 4. Perform operation
 5. Write results back
- Continue with next instruction



Instruction set architecture (ISA)

- The deeper level accessible to the programmers
- It is the boundary between SW and HW
- The interface between the programmer and the microarchitecture
- Different microarchitectures can have the same ISA (binary Compatible)
- Different generation of microarchitectures can be backward compatible
- For us: **x86 instruction set**

A very simple operation..

```
void store(double *a, double *b, double *c) {  
    *c = *a + *b;  
}
```

```
[exact@master ~]$ gcc -O2 -S -o - frammento.c  
.file "frammento.c"
```

```
.text
```

```
.p2align 4,,15
```

```
.globl store
```

```
.type      store, @function
```

```
store:
```

```
.LFB0:
```

```
.cfi_startproc
```

```
movsd     (%rdi), %xmm0  #load *a to mmx0
```

```
addsd     (%rsi), %xmm0  # load b and add to *a
```

```
movsd     %xmm0, (%rdx)  # store to C
```

```
ret
```

```
.cfi_endproc
```

```
.LFE0:
```

```
.size store, .-store
```

```
.ident     "GCC: (GNU) 4.4.7 20120313 (Red Hat 4.4.7-4)"
```

```
.section   .note.GNU-stack,"",@progbits
```

Agenda

Why HPC is parallel ?



Serial Computers



Moore law/Dennard Scaling

Parallel computers

HPC infrastructure

ORFEO HPC infrastructure

Moore Law

- Typically stated as: “Performance doubles every X months”
- Actually, closer to: “Number of transistors per unit cost doubles every X months”

The original Moore Law

The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. [...]

Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.

-- Gordon Moore,
Electronics, 1965

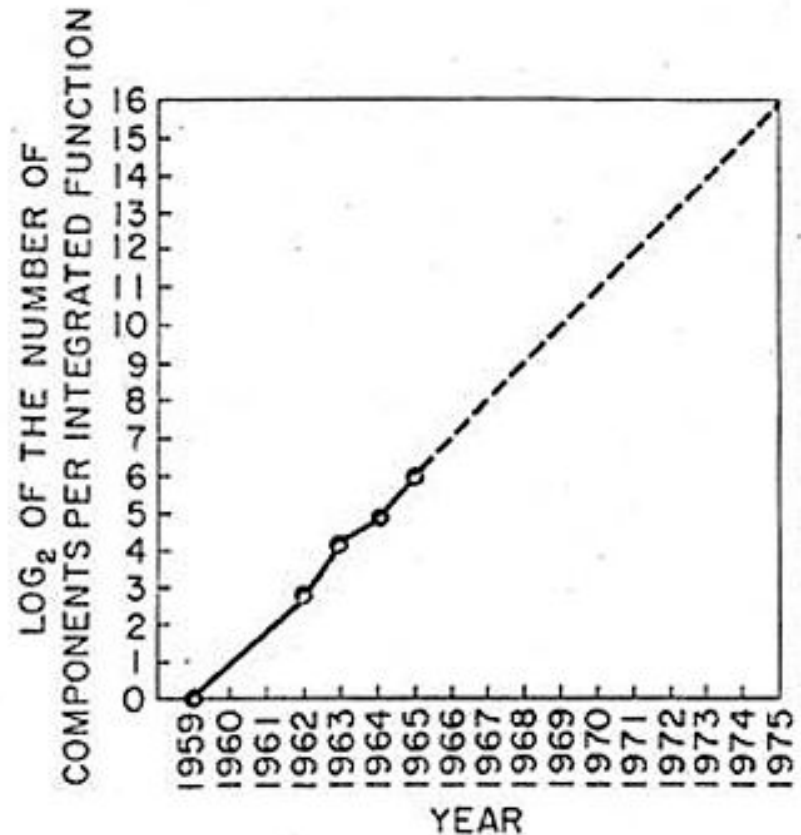


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Why is Moore's Law connected with processor performance?

Dennard Scaling: From Moore's Law to performance

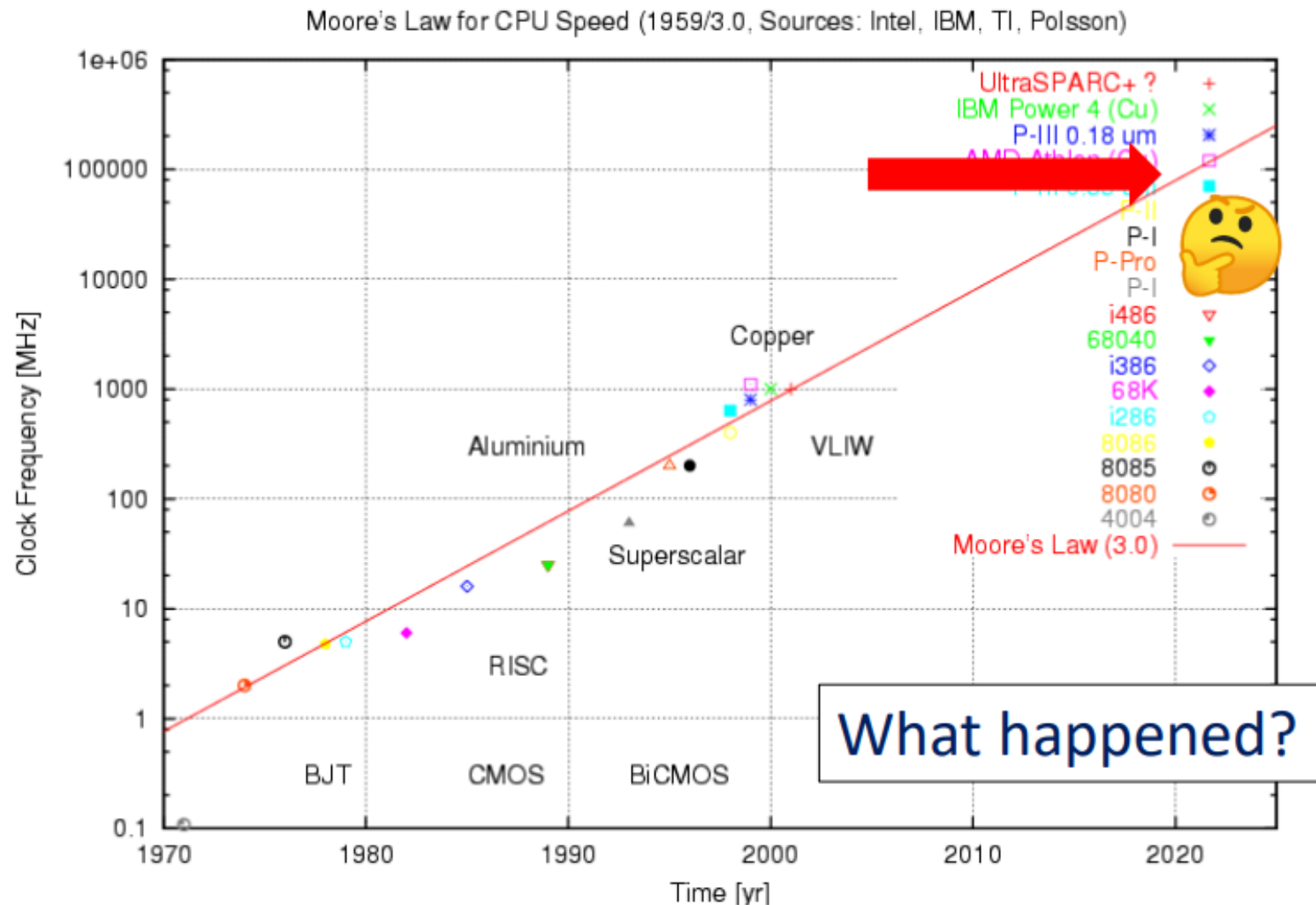
- *“Power density stays constant as transistors get smaller”*

Robert H. Dennard, 1974

- Intuitively:
Smaller transistors → shorter propagation delay → faster frequency
Smaller transistors → smaller capacitance → lower voltage
 $Power \propto Capacitance \times Voltage^2 \times Frequency$

Moore's law → Faster performance @ Constant power!

Single-core performance scaling



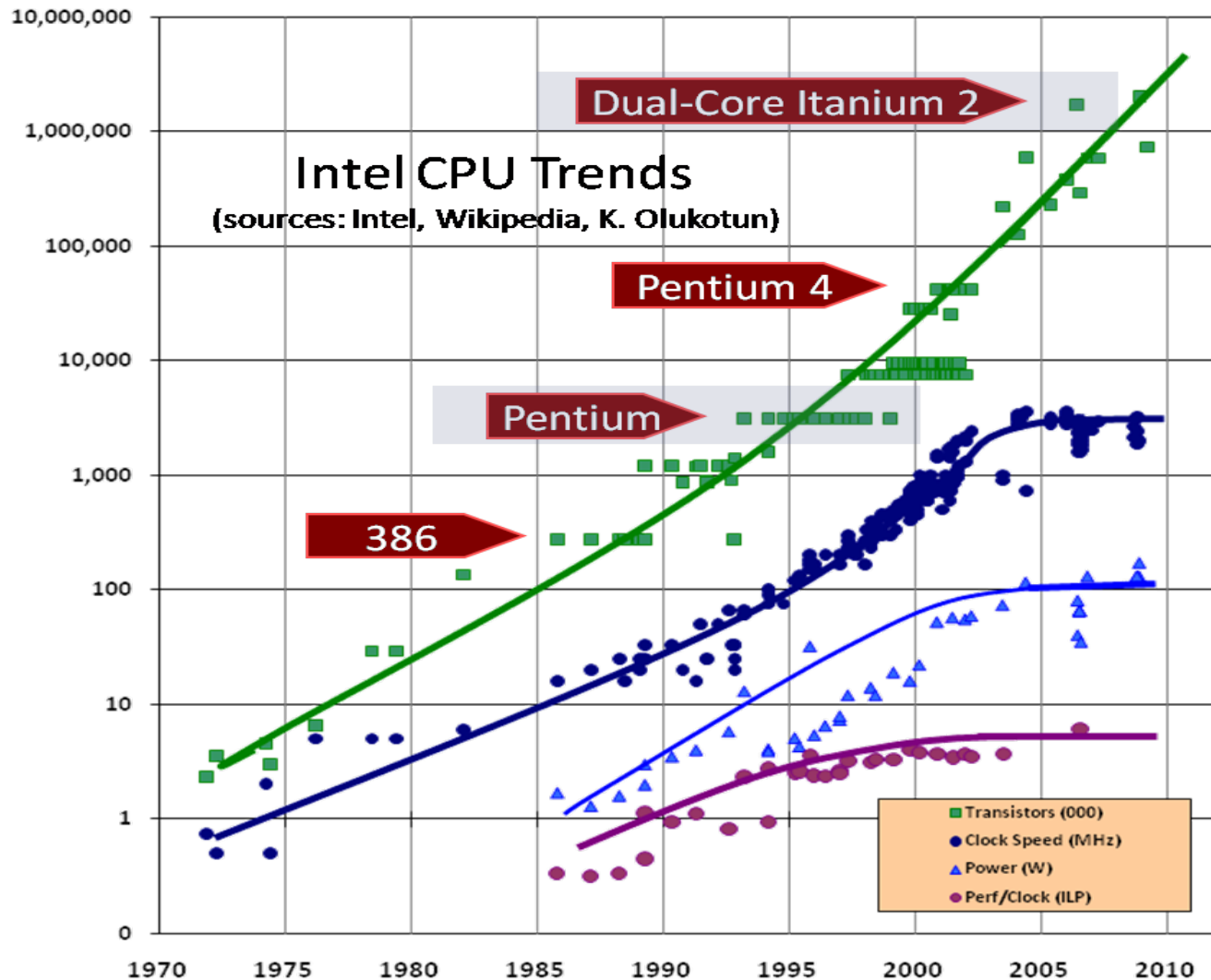
End of Dennard Scaling

- Even with smaller transistors, we cannot continue reducing power..

And now ?

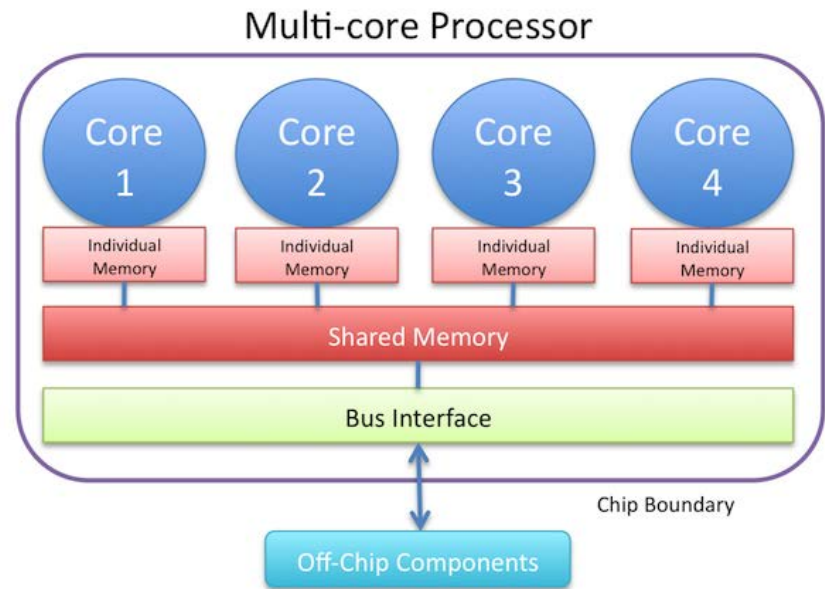
- 2 options:
 - Increase power (when increase frequency)
 - Stop frequency scaling...

(original) Moore law still valid...



CPU are multicore processor

- Because of power, heat dissipation, creating 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



Agenda

Why HPC is parallel ?



Serial Computers



Moore law/Dennard Scaling



Parallel computers

HPC infrastructure

ORFEO HPC infrastructure

PARALLELISM IS
EVERYWHERE
even in your
laptop..

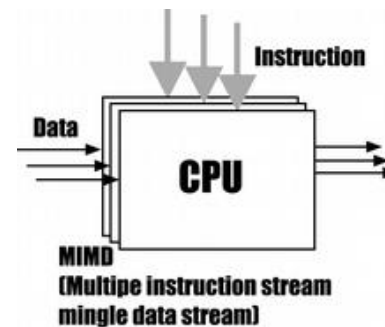
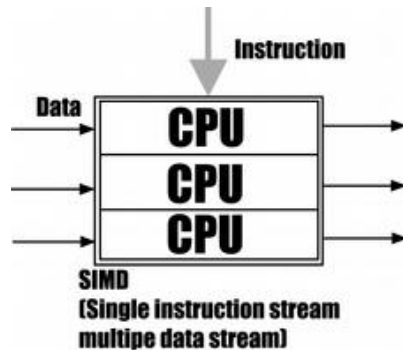
Parallel Computers

- Flynn Taxonomy (1966): may help us in classifying them:
 - Data Stream
 - Instruction Stream

		Instruction stream	
		Single	Multiple
Data stream	Single	SISD	MISD
	Multiple	SIMD	MIMD

Comments

- Flynn taxonomy does not help too much nowadays with modern HPC infrastructure
 - CPU and computers are changed too much in the last 50 years
- However, SIMD and MIMD concepts are still used HPC hardware



Comments (2)

	HW level	SW level
SISD	A Von Neumann CPU	no parallelism at all
MISD	On a superscalar CPU, different ports executing different <i>read</i> on the same data	<ul style="list-style-type: none">• ILP on same data;• Multiple tasks or threads operating on the same data
SIMD	Any vector-capable hardware, the vector registers on a core, a GPU, a vector processor, an FPGA, ...	data parallelism through vector instructions and operations
MIMD	Every multi-core/processor system; on a superscalar CPUs, different ports executing different ops on different data	<ul style="list-style-type: none">• ILP on different data;• Multiple tasks or threads executing different code on different data.

Essential component of a HPC cluster

- Several computers (nodes)
 - often in special cases (1U) for easy mounting in a rack
- One or more networks (interconnects) to hook the nodes together
- Some kind of storage
- A login/access node..

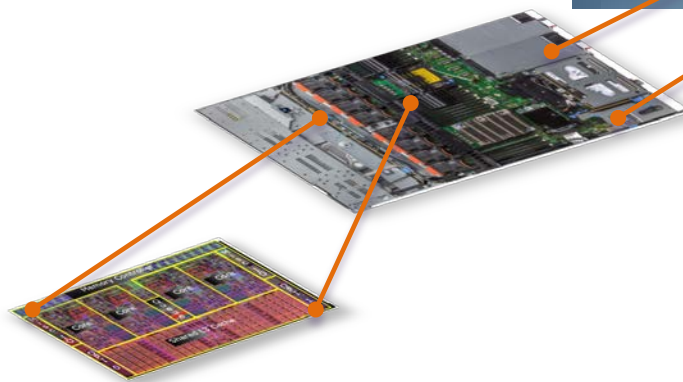


The hardware behind HPC

interconnected
racks of connected
nodes



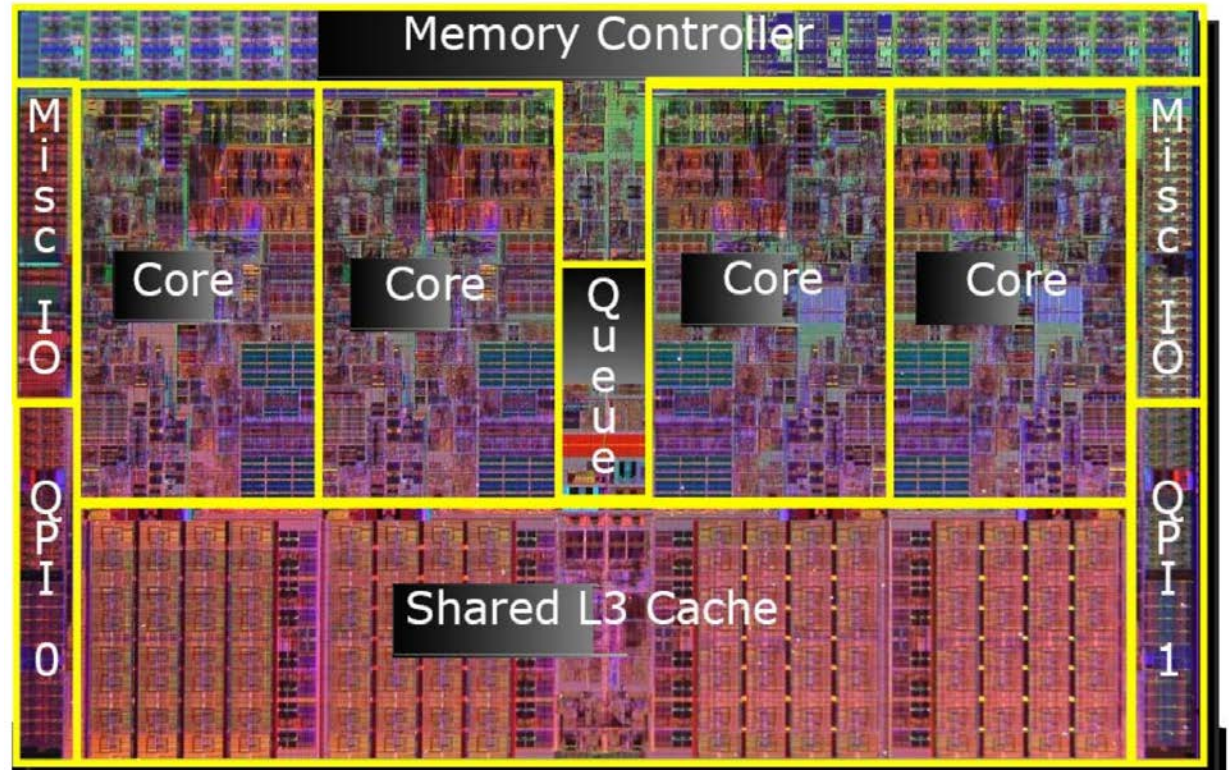
single nodes



single cpu

Single CPU topology

**Modern
CPUS**
are multi- (or
many-) cores

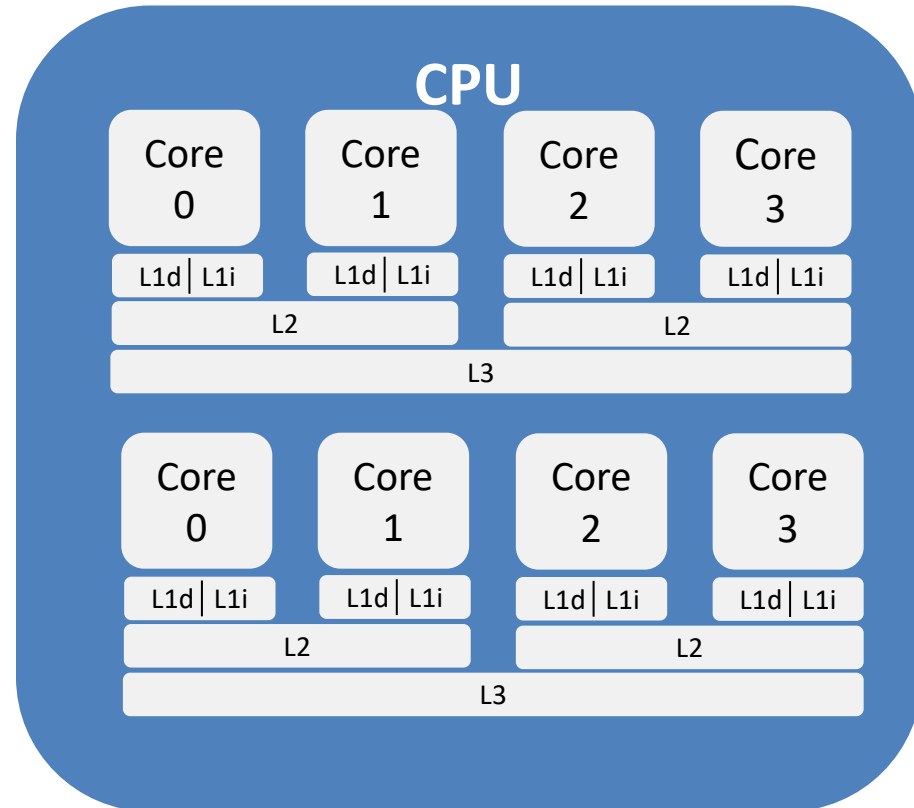
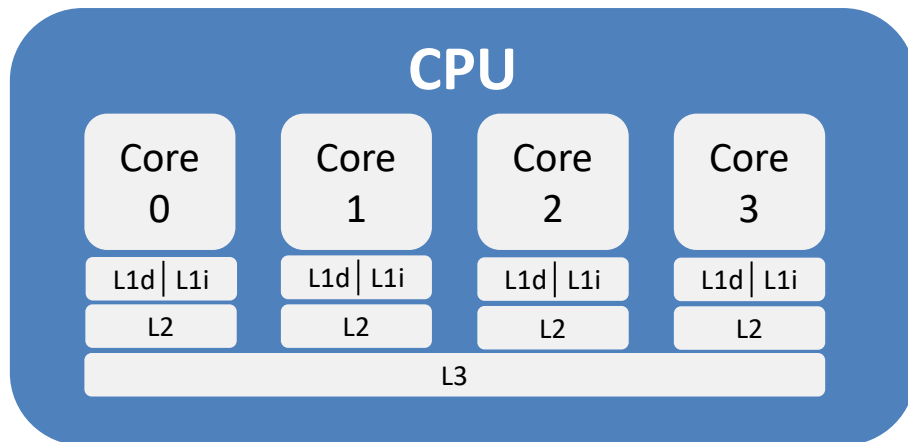


Core : definition

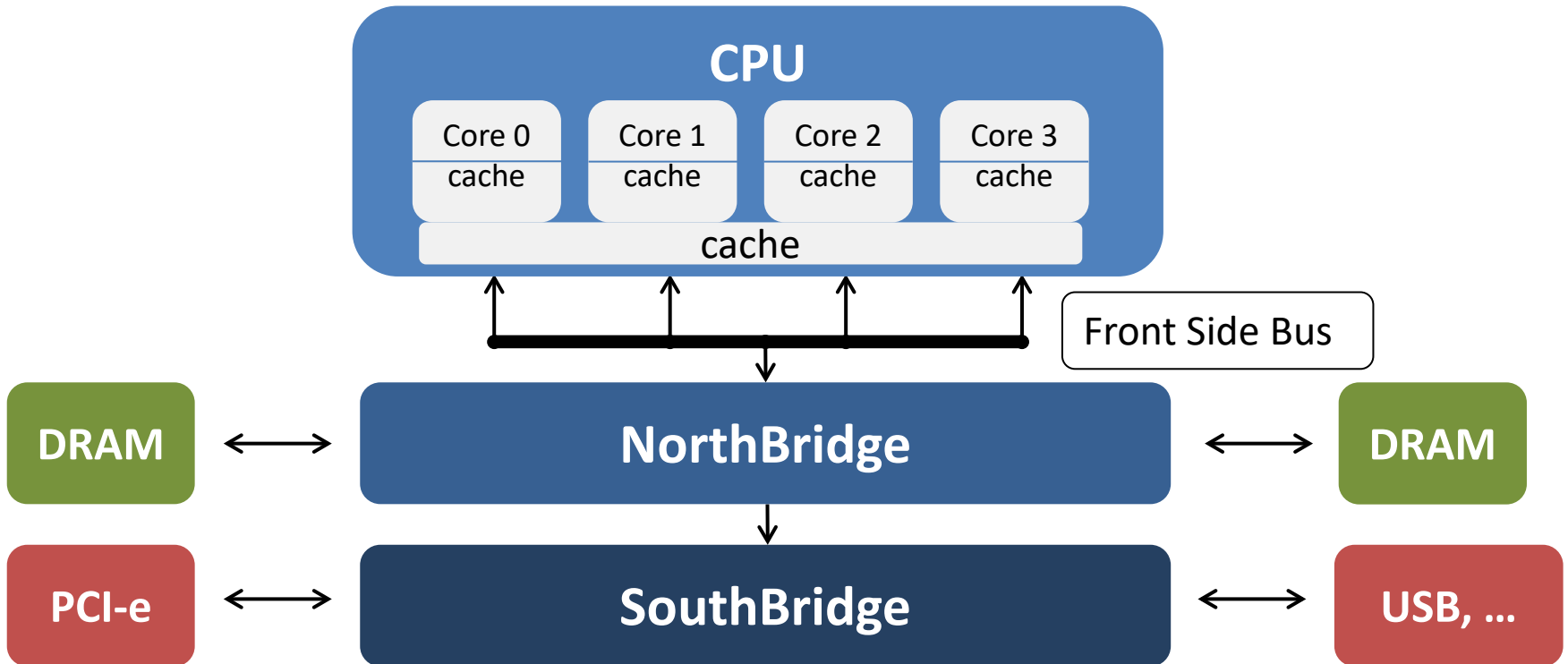
- A core is the smallest unit of computing, having one or more (hardware/software) threads and is responsible for executing instructions.

Single CPU topology

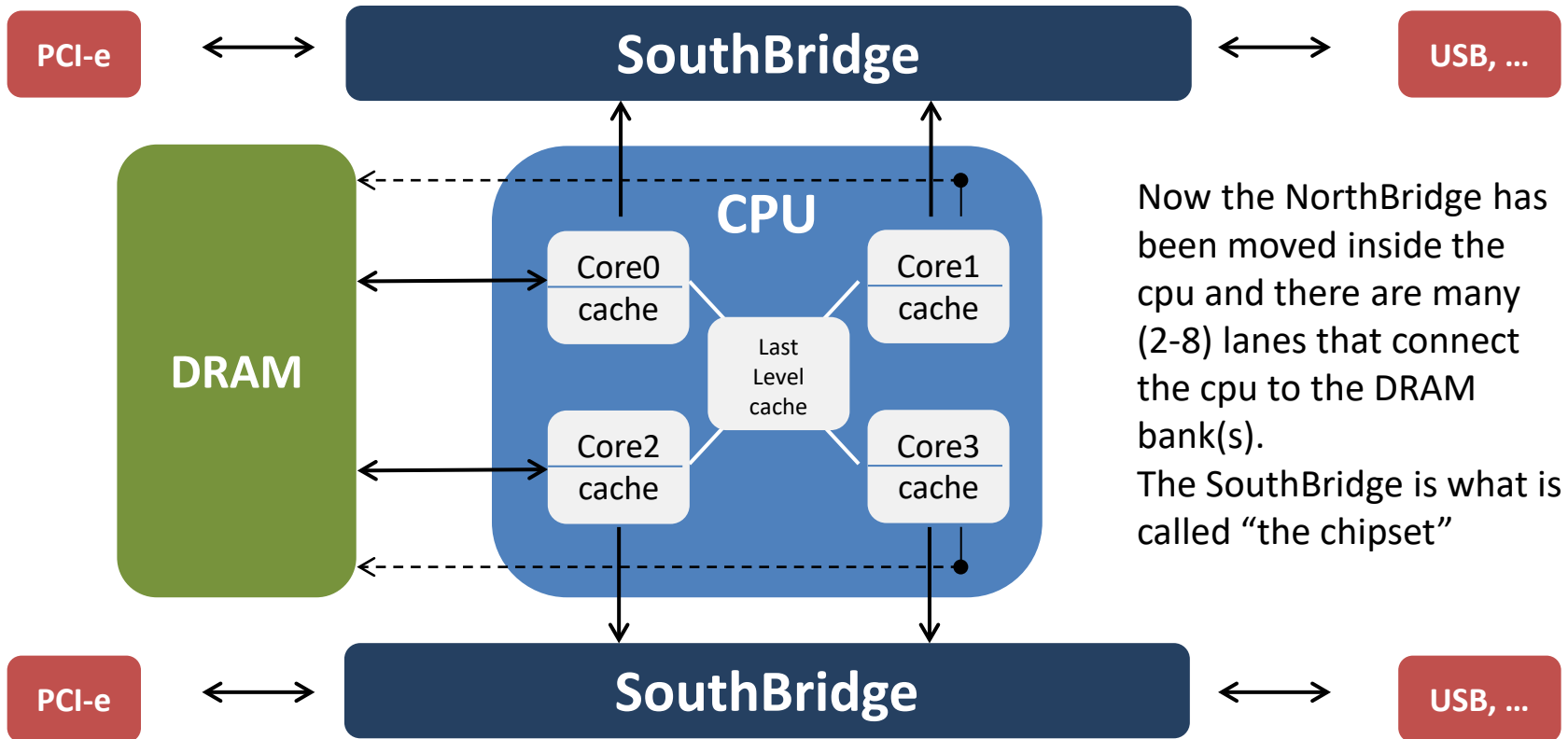
- Cache hierarch can have different topologies



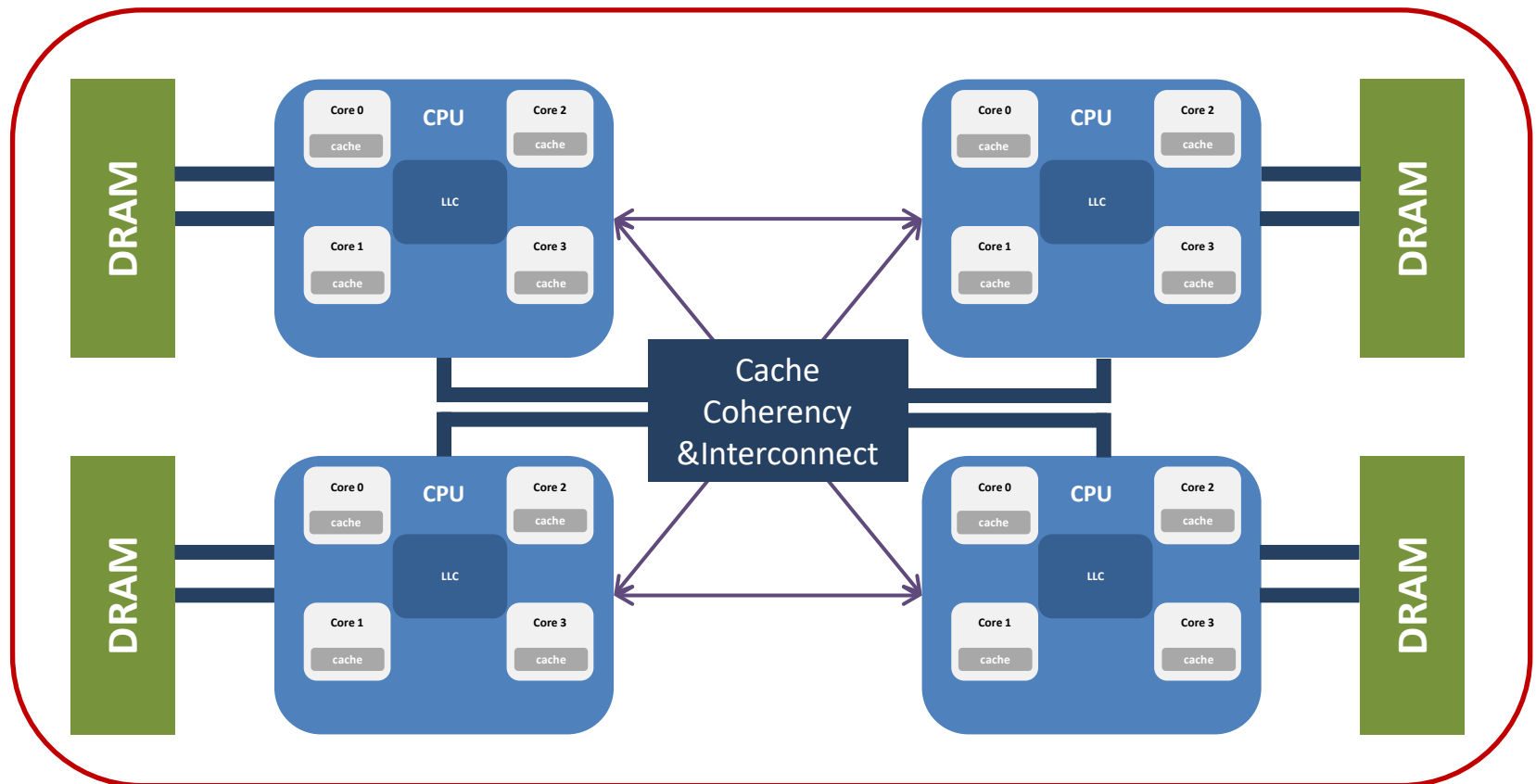
Old CPU internal layout



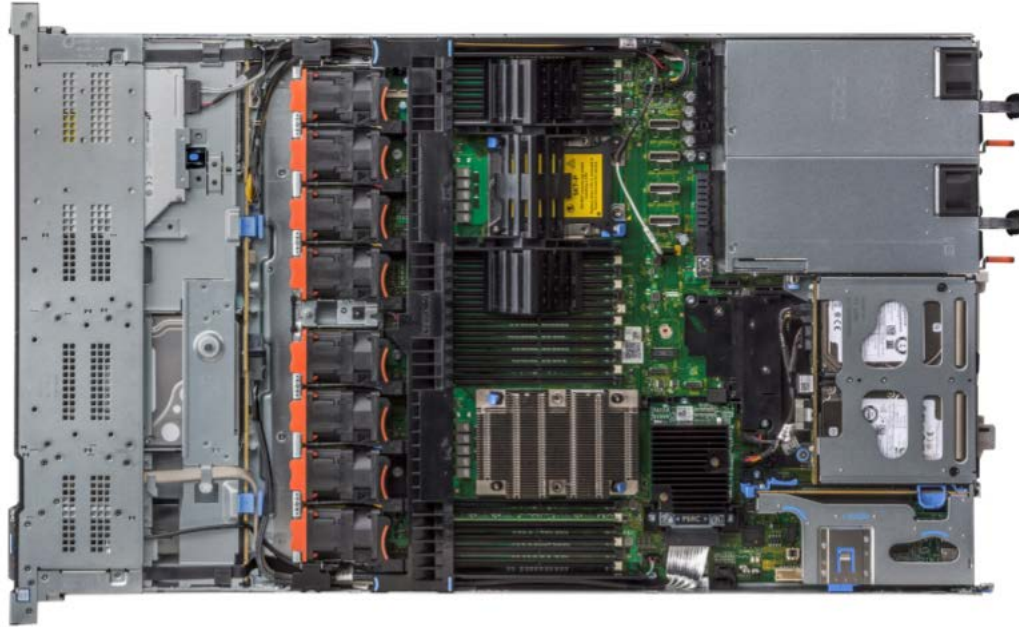
Modern CPU layout



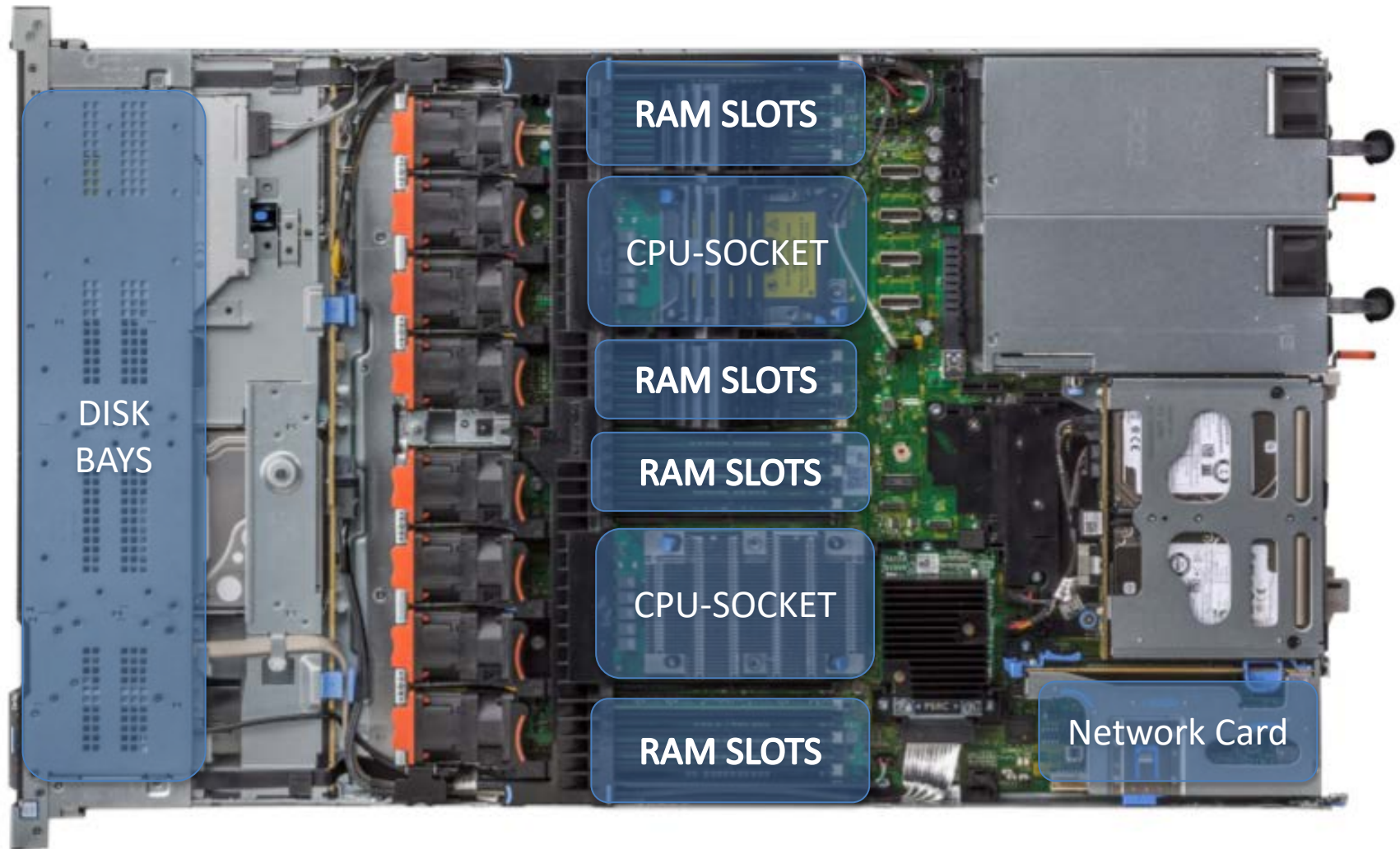
Node topology



Modern 1U computing nodes



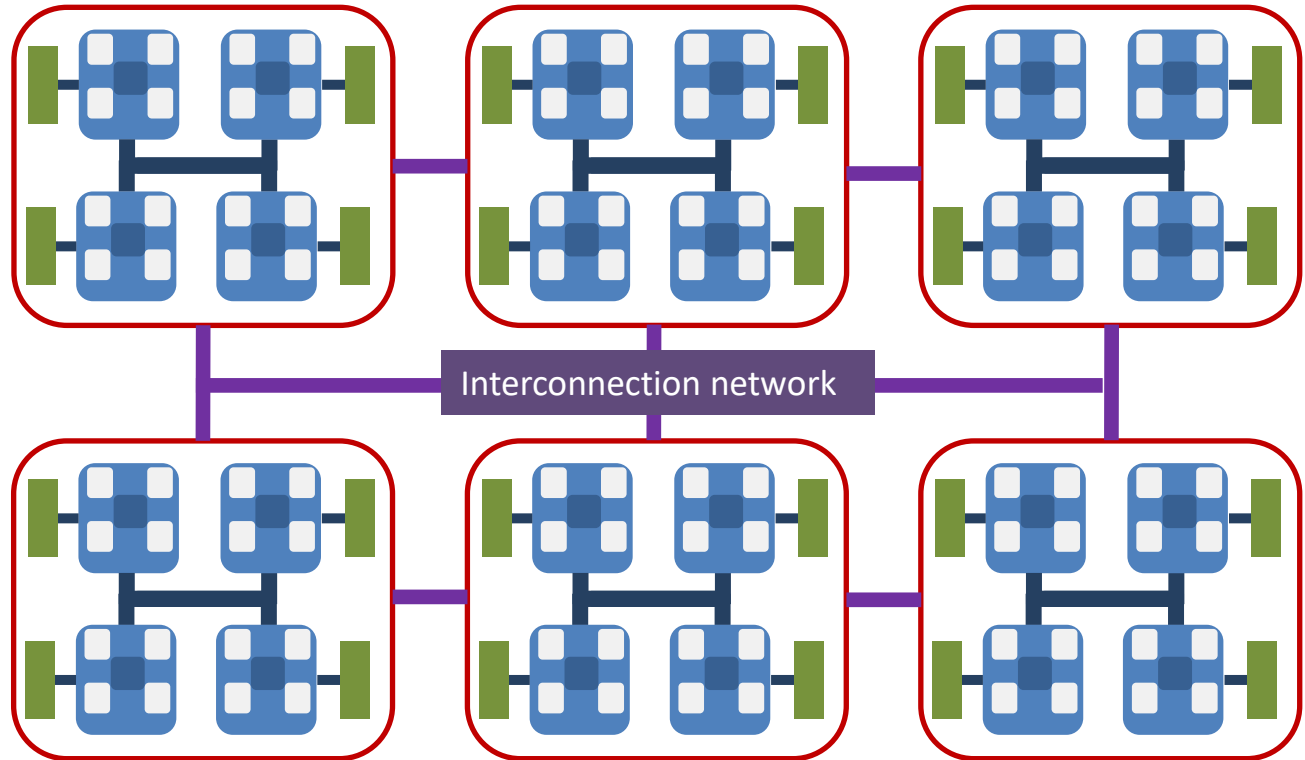
What does one node contain exactly ?



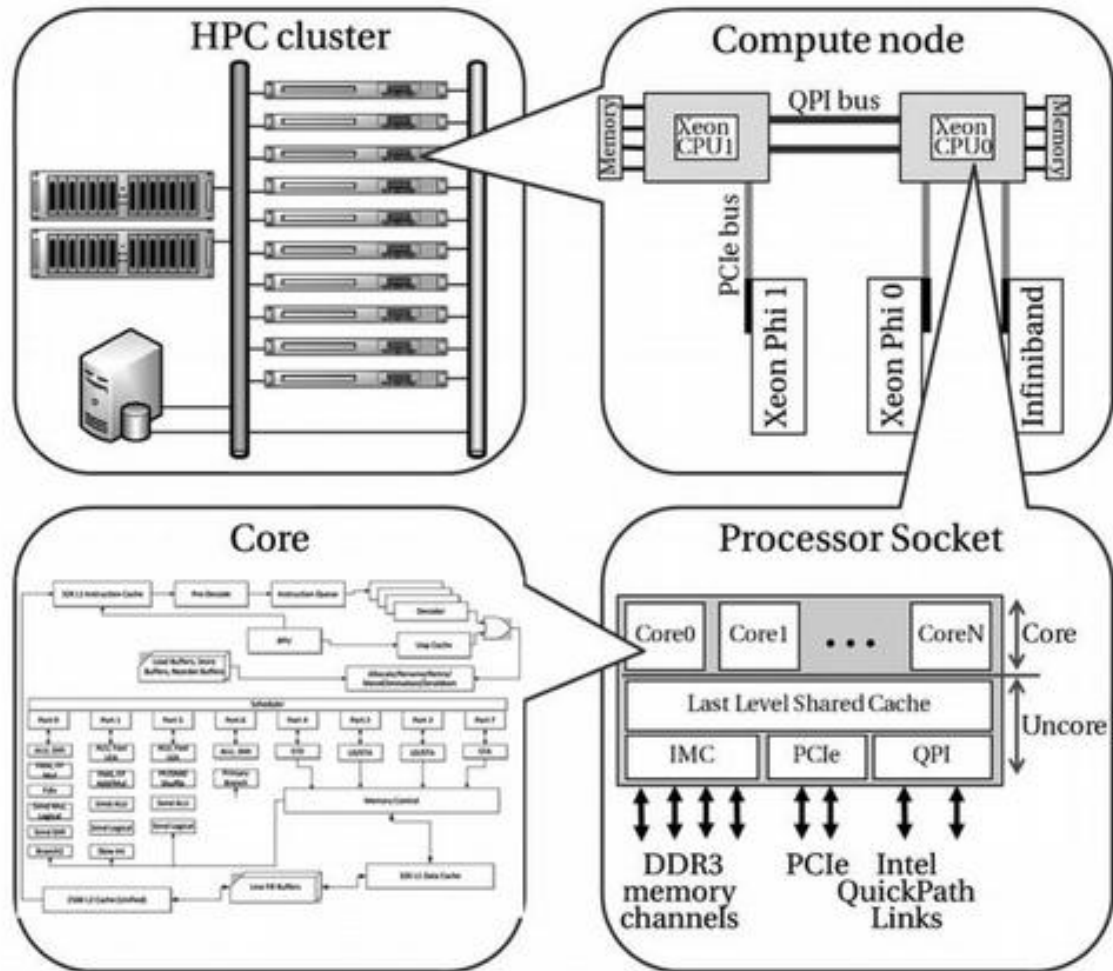
The overall topology

CLUSTER OF COMPUTING NODES

Note: there are many
different topologies for the
interconnection network.



The building blocks of a HPC infrastructure (cluster)

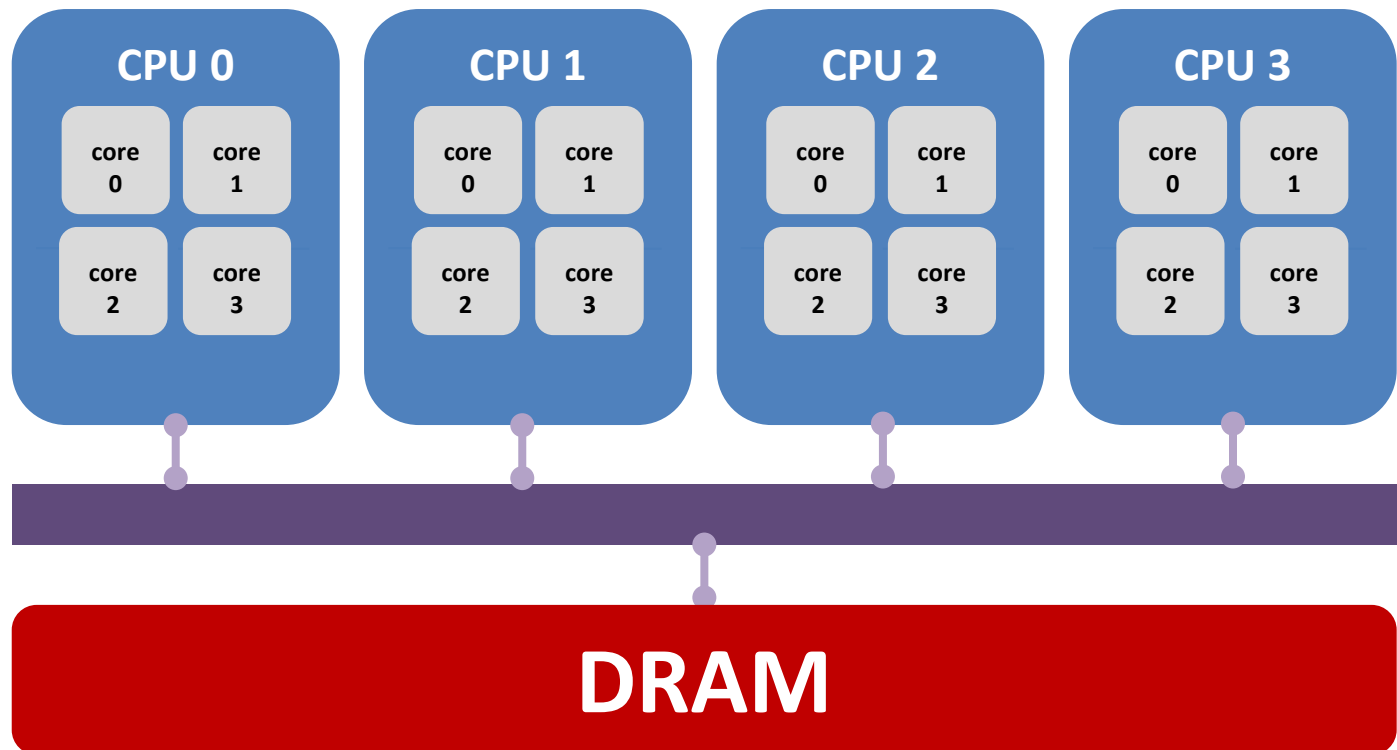


What about memory ?

- Note that on a supercomputer there is a hybrid approach as for the memory placement:
- the memory on a single nodes can be accessed directly by all the cores on that node, meaning that memory access is a “read/write” instructions irrespectively of what exact memory bank it refers to. This is called **shared-memory**.
- when you use many nodes at a time, a process can not directly access the memory on a different node. It need to issue a request for that, not a read/write instruction. That is named **distributed memory**.
- These are hardware concepts, i.e. they describe how the memory is physically accessible. However, they do also refer to programming paradigms, as we’ll see in a while.

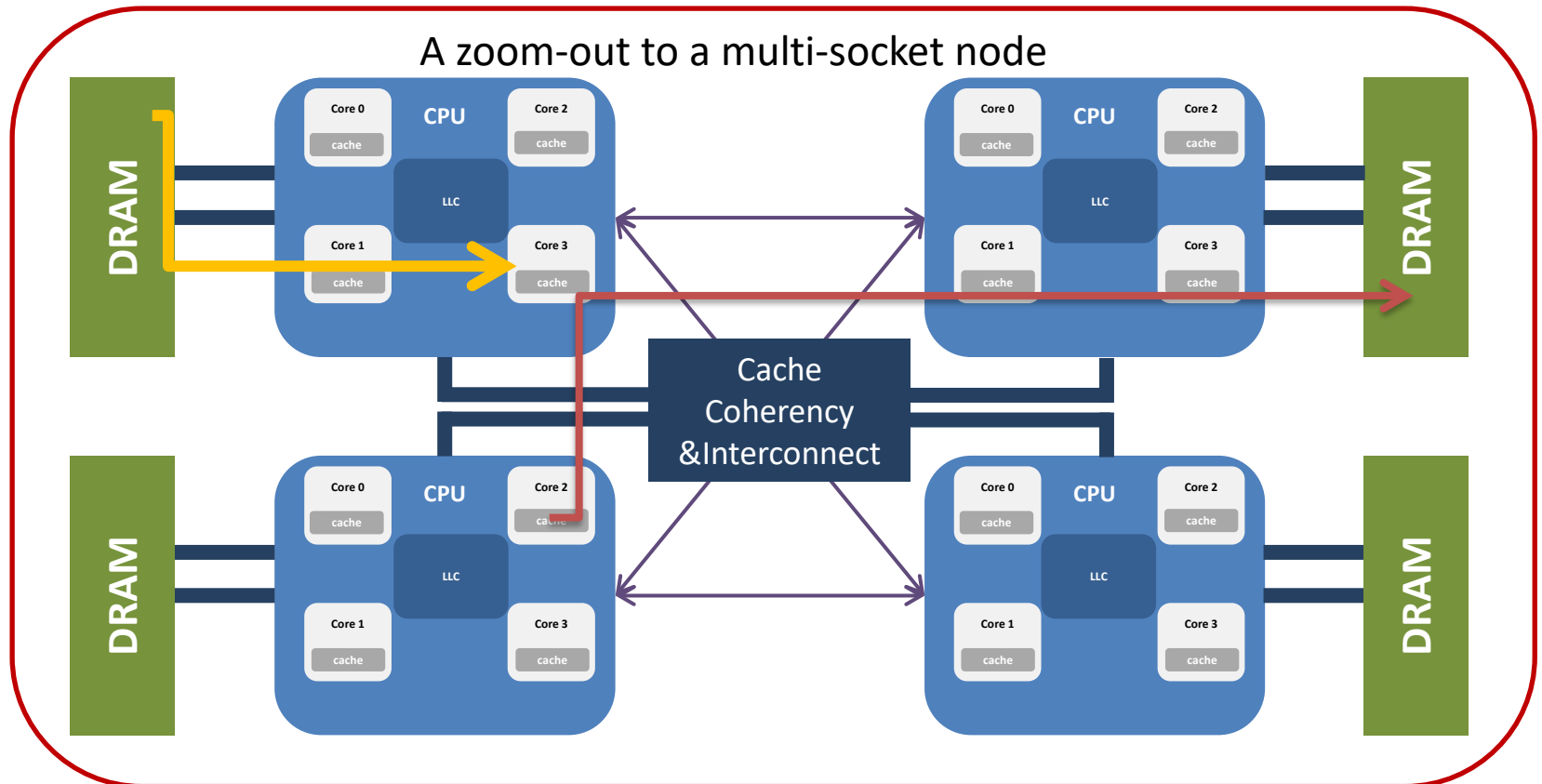
Shared memory: UMA

Uniform memory access (UMA): Each processor has uniform access to memory. Also known as symmetric multiprocessors (**SMP**)

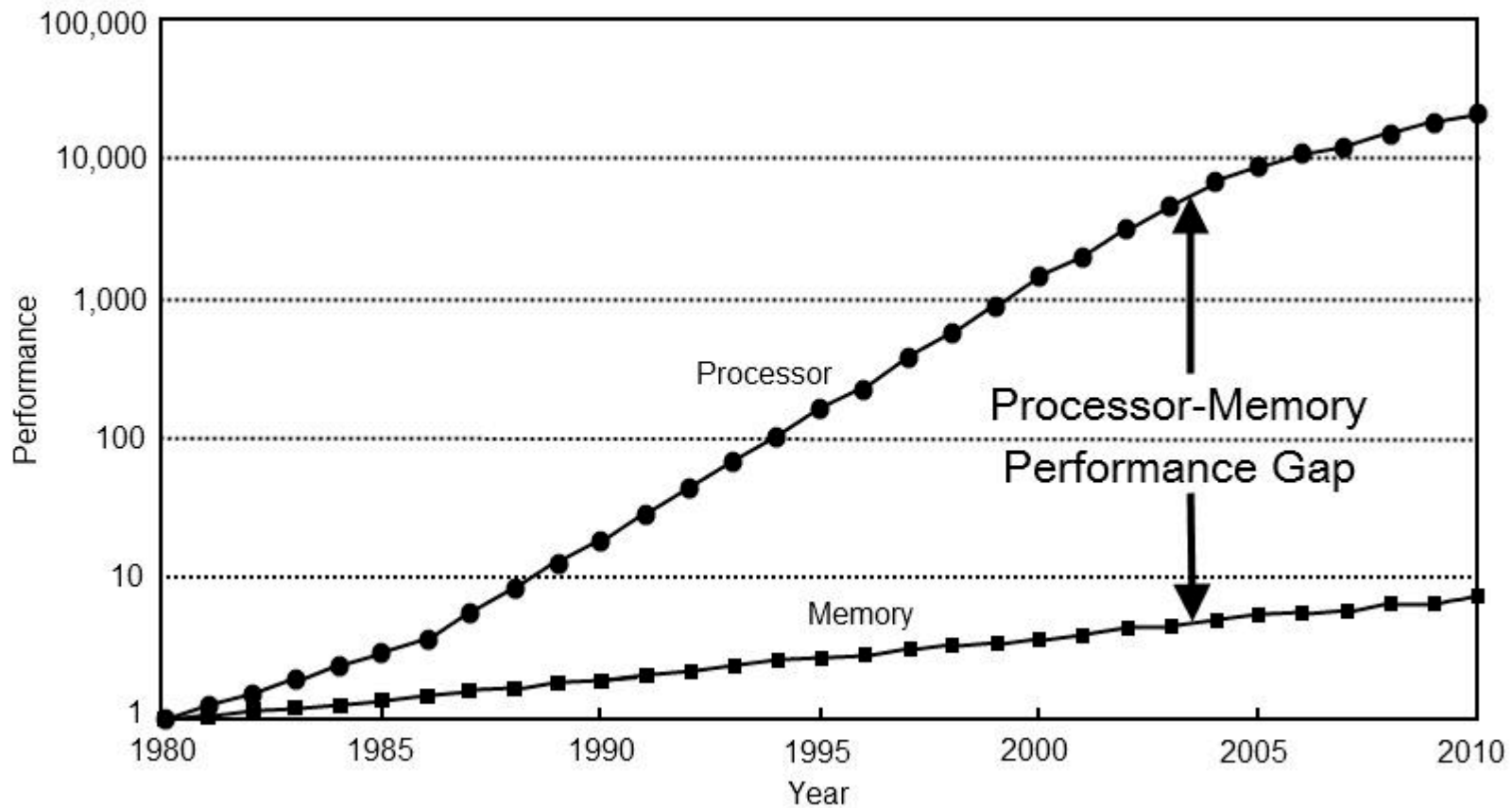


Shared memory: NUMA

Non-uniform memory access (NUMA): Time for memory access depends on location of data. Local access is faster than non-local access.



Memory wall problem



Challenges for multicore

- Relies on effective exploitation of multiple-thread parallelism
 - Need for parallel computing model and parallel programming model
- Aggravates **memory wall problem**
 - Memory bandwidth
 - Way to get data out of memory banks
 - Way to get data into multi-core processor array
 - Memory latency
 - Cache sharing

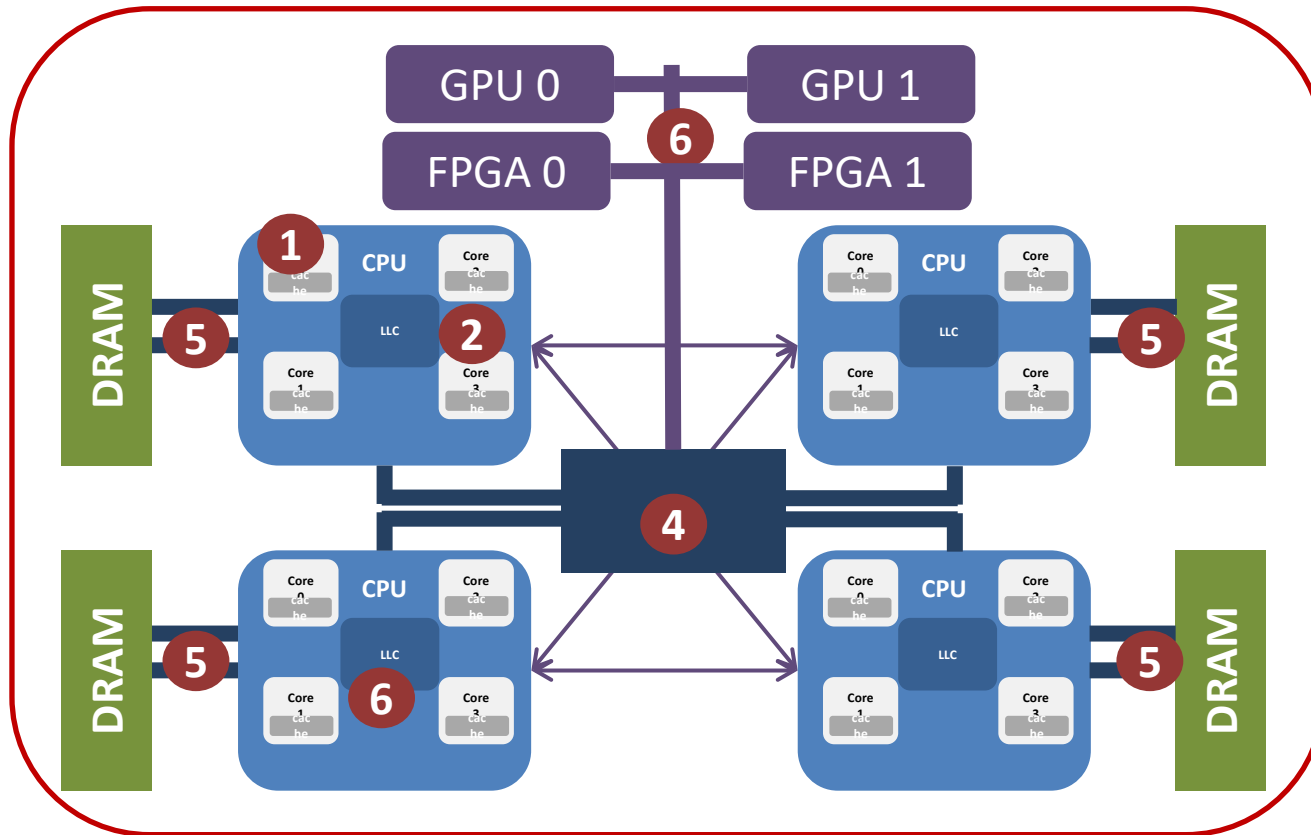
a little bit of jargon..

- Multiprocessor = server with more than 1 CPU
- Multicore = a CPU with more than 1 core
- Processor = CPU = socket

BUT SOMETIME:

- Processor = core
- a process for each processor (i.e. each core)

Parallelism within a HPC node



- Parallel resources
 - ILP/SIMD units (1)
 - Cores (2)
 - Inner cache levels (3)
 - Socket/ccNuma domains (4)
 - Multiple accelerator (5)

Discovering a Numa topology

- `numactl` tool
it also controls the Linux NUMA policy
- `/proc/cpuinfo`
- `hwloc` (by OpenMPI)

Agenda

Why HPC is parallel ?



Serial Computers



Moore law/Dennard Scaling



Parallel computers



HPC infrastructure

ORFEO HPC infrastructure



ORFEO HPC nodes..

TYPE OF NODE	RAM x nodo	CORES x nodo	GPU x nodo	Peak performance (Tflops)
10 THIN intel nodes	768 GB	24	-	1,997
2 FAT intel nodes	1536 GB	36	-	3,456
4 GPU intel nodes	256 GB	24	2 V100 (32GB)	2,073 +2x 7
8 EPYC Amd nodes (EPYC 7H12 64-Core Processor)	512 GB	128	-	?
2 DGX Nvidia Station	2048GB	128 (EPYC)	8 A100	?
TOTALE 16	~ 15 Terabyte	1688	24	~ ?

Network cluster classification

- HIGH SPEED NETWORK
 - parallel computation
 - low latency /high bandwidth
 - Usual choices: Infiniband...
- I/O NETWORK
 - I/O requests (NFS and/or parallel FS)
 - latency not fundamental/ good bandwidth
 - GIGABIT could be ok /10Gb and/or Infiniband better
- In band Management network
 - management traffic of all services (LRMS/NFS/software etc..)
- Out of band Management network:
 - Remote control of nodes and any other device

Orfeo network

- HIGH SPEED NETWORK

100

Gbit HDR Infiniband

- I/O NETWORK

- In band Management network

25Gbit Ethernet

- Out of band Management network:

1Gbit Ethernet

ORFEO storage: hardware

	FAST storage (NVMe)	FAST storage (SSD)	Standard storage (HDD)	Long term preservation
# of server	4		6	1
RAM	6 x 16GB		6 x 16GB	6 x 16GB
Disk per node	2x 1.6TB NVMe PCIe card	20 x 3.84TB	15 x 12TB	84 x 12TB + 42 x 12TB
Storage provider	CEPH parallel FS	CEPH parallel FS	CEPH parallel FS	Network FS (NFS)
RAW storage	12TB	320 TB	1080 TB	1,512 TB

I/O subsystem on ORFEO:

- Home
 - once logged in, each user will land in its home in ``/u/[name_of_group]/[name_of_user]`
 - e.g. the home of user area is in `/u/area/[name_of_users]`
 - it's physically located on ceph large FS, and exported via infiniband to all the computational nodes
 - quotas are enforced with a default limit of 2TB for each users
 - soft link are available there for the other areas

```
[cozzini@login ~]$ ls -lrt
total 548398
lrwxrwxrwx 1 cozzini area          18 Apr  7  2020 fast -> /fast/area/cozzini
lrwxrwxrwx 1 cozzini area          21 Apr  7  2020 storage -> /storage/area/cozzini
lrwxrwxrwx 1 cozzini area          21 Apr 16  2020 scratch -> /scratch/area/cozzini
```

I/O subsystem on ORFEO:

- /Scratch
 - it is large area intended to be used to store data that need to be elaborated
 - it is also physically located on ceph large FS, and exported via infiniband to all the computational nodes

```
[cozzini@login ~]$ df -h /scratch
Filesystem
Size  Used Avail Use% Mounted on
10.128.6.211:6789,10.128.6.213:6789,...:/ 598T   95T  503T  16% /large
```

- /fast
 - is a fast space available for each user, on all the computing nodes
 - is intended to be a **fast scratch area** for data intensive application

```
[cozzini@login ~] df -h /fast
Filesystem
Size  Used Avail Use% Mounted on
10.128.6.211:6789,10.128.6.212:6789,...:/  88T   4.3T   83T   5% /fast
```

I/O subsystem on ORFEO:

- Long term storage:
 - it is NFS mounted via 50bit ethernet link
 - it is intended for long-term storage of final processed dataset

```
[cozzini@login ~]$ df -h | grep 231
10.128.6.231:/illumina_run          128T   58T   70T   46% /illumina_run
10.128.2.231:/storage              37T   27T   9.9T   74% /storage
10.128.6.231:/long_term_storage    128T  112T   17T   88% /long_term_storage
10.128.6.231:/analisi_da_consegnare 100T   33T   68T   33% /analisi_da_consegnare
10.128.6.231:/onp_run_1           117T   27T   91T   23% /onp_run
10.128.2.231:/lage_archive         128T   68T   60T   54% /lage_archive
```

Hardware Completed !

Why HPC is parallel ?



Serial Computers



Moore law/Dennard Scaling



Parallel computers



HPC infrastructure



ORFEO HPC infrastructure

