MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

IN COLLABORATION WITH MIT MANAGEMENT SLOAN SCHOOL | UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

MASTER MEIM 2022-2023

DIGITAL TECH

High Performance Computing

Lesson 2

Prof. Livia Marcellino

Prof. of High Performance Computing, Università degli Studi di Napoli Parthenope

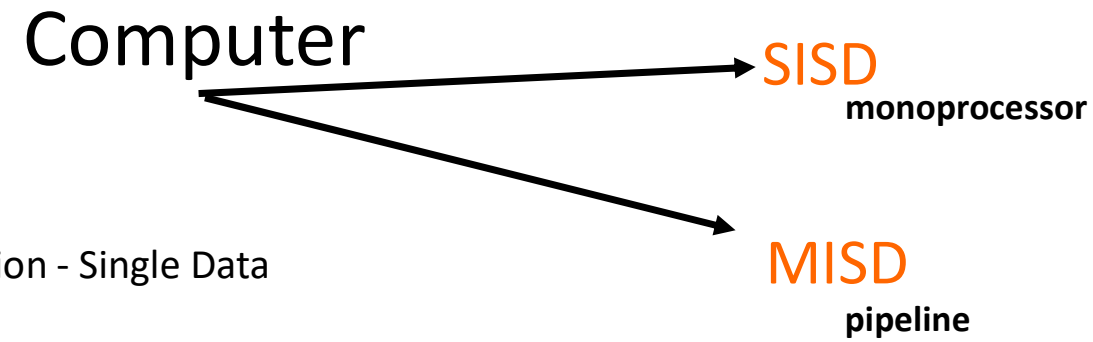
MEIM MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

The parallel paradigm:
Flynn's taxonomy

Flynn's taxonomy (since 1966)

Michael J. Flynn begins to classify computers...



SISD: Single Instruction - Single Data

MISD: Multiple Instruction - Single Data

Pipeline: critical issues

- The first problem arises from the **parallel work** of the units:

suppose the pipelined CPU needs to execute the following code

C=A+B

D=C-1

The first statement takes the values stored in variables A and B, then adds them together and finally stores the result in variable C.

The second statement takes the value stored in variable C, subtracts it by one and saves the result in D.

But the second instruction cannot be processed until the data of the first operation is available in memory and then the second operation is stopped and awaits the end of the first one!

Pipeline: critical issues

- The second problem consists in the **conditioned jumps**.

If the code contains conditional instructions (**logical condition**) and this condition is not verified, the serial flux of code is **interrupted** and it shifts towards a new part of the code.

Whenever this happens the **microprocessor** must perform different operations and then **it must empty the pipeline** of data in order to **upload the new data**.

Obviously these operations delay the execution.

New forms of parallelism:

To overcome the intrinsic limits of temporal parallelism, starting from the 1970s, it was introduced a **higher level** of parallelism involving **several processors**

synchronously or **asynchronously**.

SPATIAL PARALLELISM

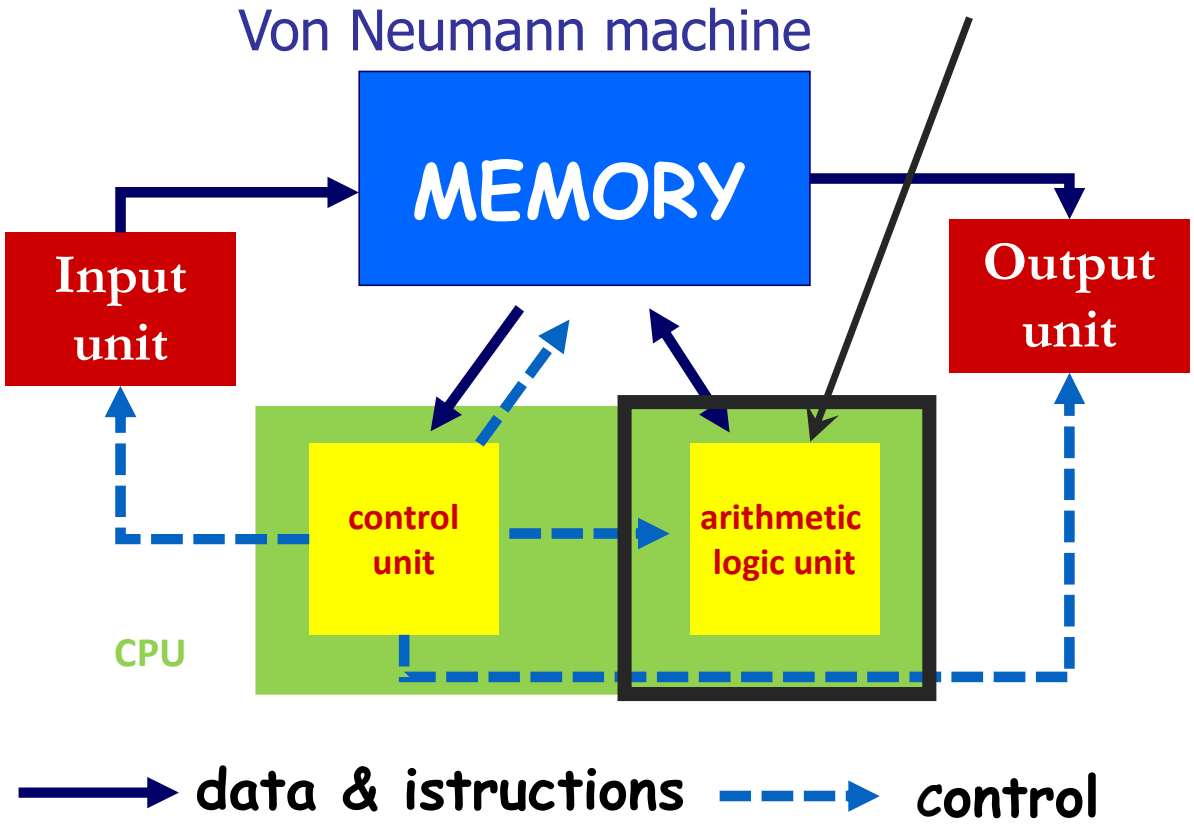
Second type of parallelism: synchronous

How the synchronous parallelism

is implemented in a computer?

Parallelism involving several arithmetic-logical units

Second type of parallelism (multiple ALU)

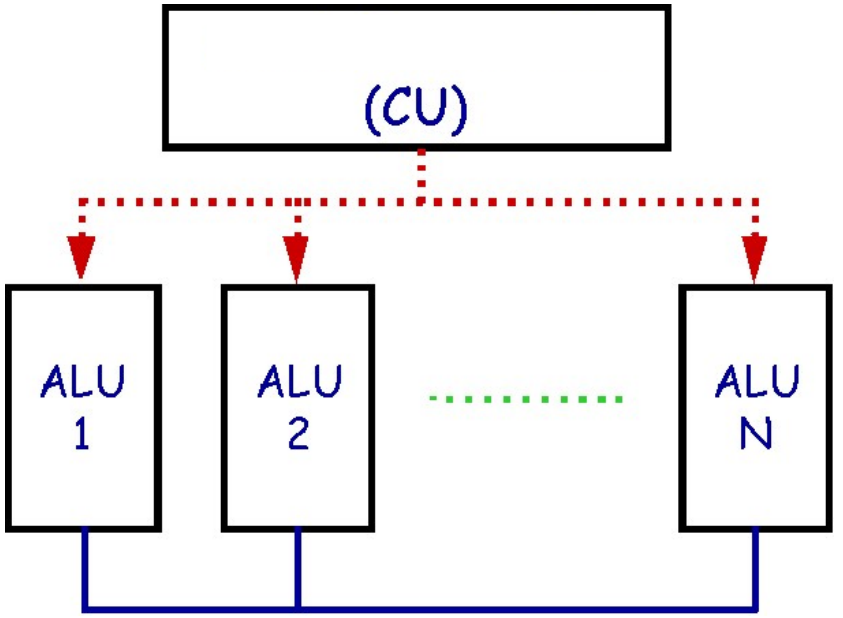


Spatial parallelism synchronous

Several arithmetic-logic units (ALU)

operate following a single control (CU) running in parallel

the ***same instruction*** on ***different data***



SIMD computer

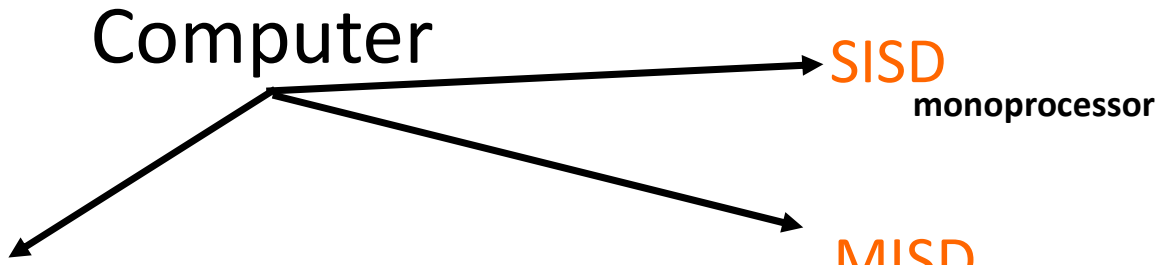
(Single Instruction Multiple Data)

Spatial parallelism synchronous

The **Cray-1** (**1976**) destined to become one of the most famous and most successful supercomputers in history.

A single control unit transmits instructions, which are executed by all processors.
The array processor is especially suitable for computation with matrices and vectors.

Flynn's taxonomy (since 1966)



Multiple arithmetic-logic units, each with its own operand register, the single flow of instructions acts simultaneously on all units.
In this way, **the latency due to the upload of operands is not eliminated, but the number of processed instructions (per second) is increased.**

Remember this observation when we will talk about GPUs!

Third type of parallelism: asynchronous

How the *asynchronous parallelism*

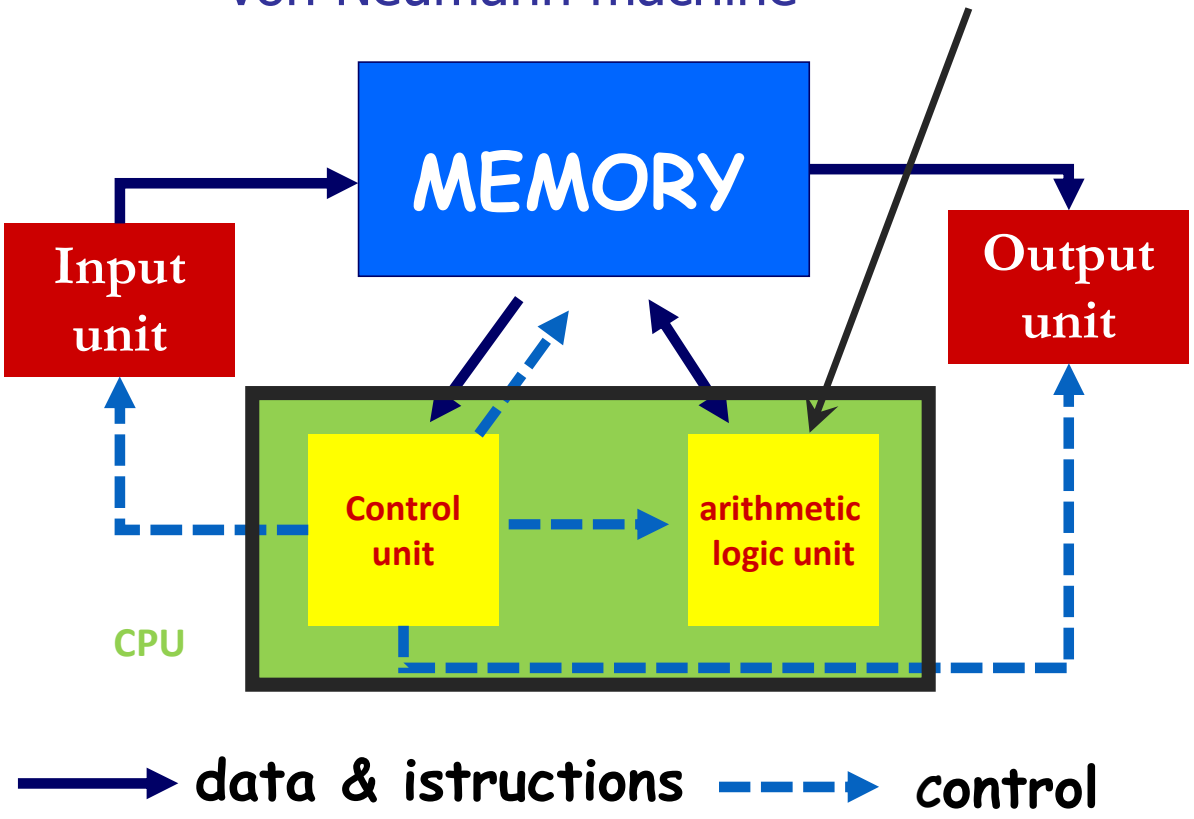
is implemented in a computer?

Different processors cooperate by executing different instructions on different data

Parallelism involving several CPU=ALU+CU

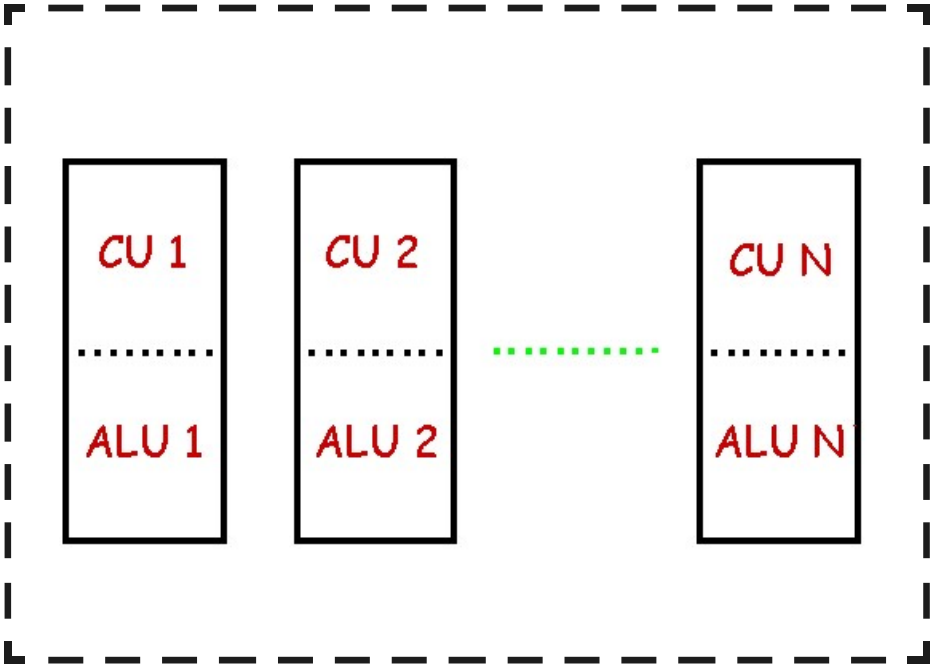
Third type of parallelism *(multiple CPU)*

Von Neumann machine



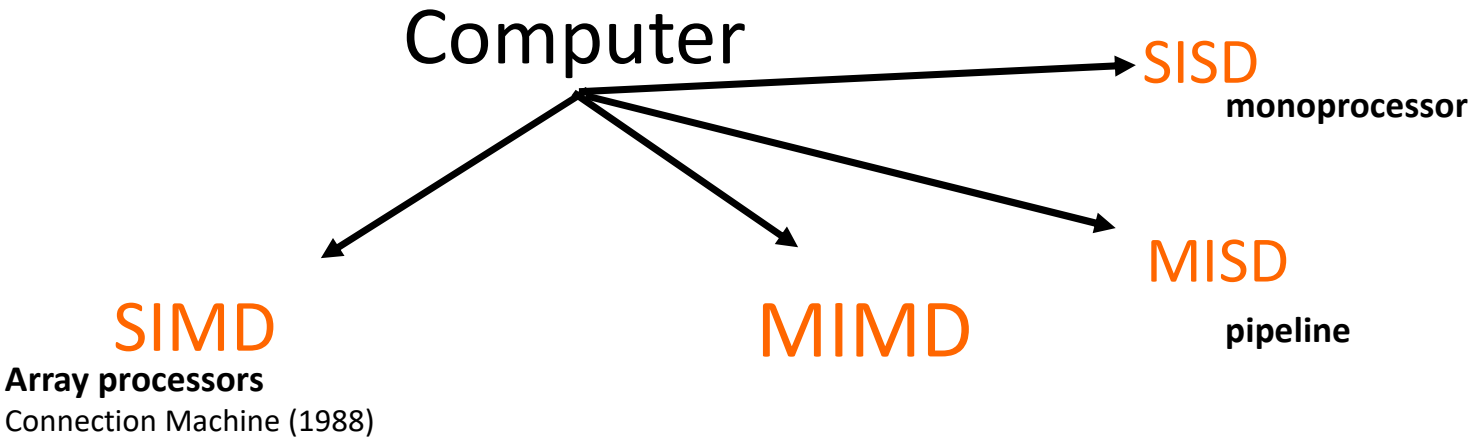
Asynchronous parallelism

More CPUs (ALUs + CUs)
execute in parallel
the ***different instructions***
on ***different data***



Computers MIMD
(Multiple Instruction Multiple Data)

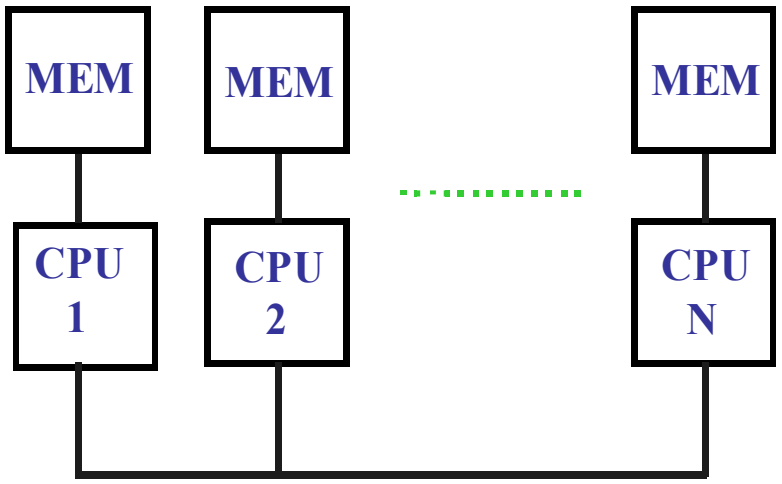
Flynn's taxonomy (since 1966)



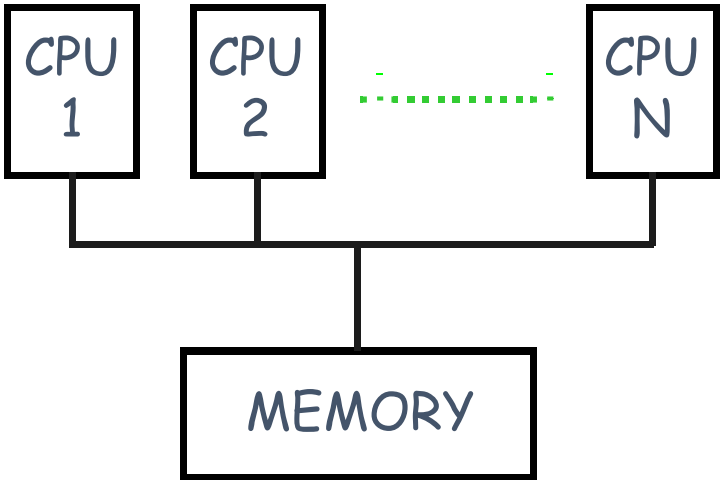
Array processors
Connection Machine (1988)

MIMD

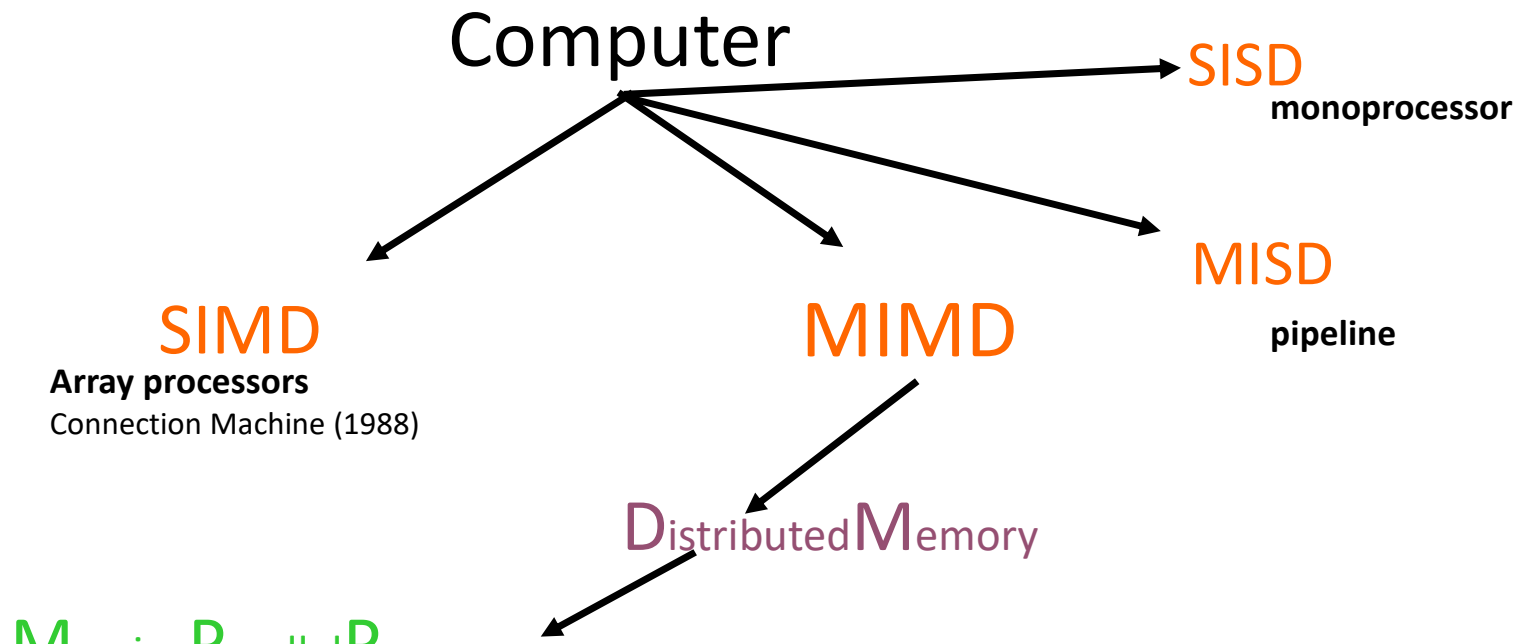
Computer MIMD with
Distributed Memory
(DM)



Computer MIMD with
Shared Memory
(SM)



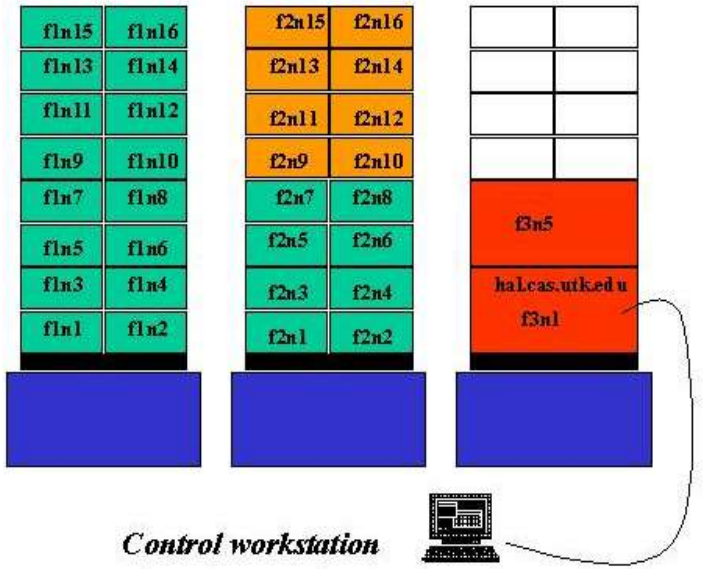
Flynn's taxonomy (since 1966)



used to indicate an architecture with independent ALUs or microprocessors working in parallel (**IBM RS/6000 SP, Earth Simulator, Blue Gene, ASCI White, ASCI Red, Roadrunner**)

Examples of computers MIMD-DM-MPP

IBM RS/6000 SP



Scalable system up to 512 nodes
Each node is composed of:
4-16 64-bit POWER 3 (RISC) processors
Connection between nodes via high-performance switches (star topology)

Examples of computers MIMD-DM-MPP

ASCI Red

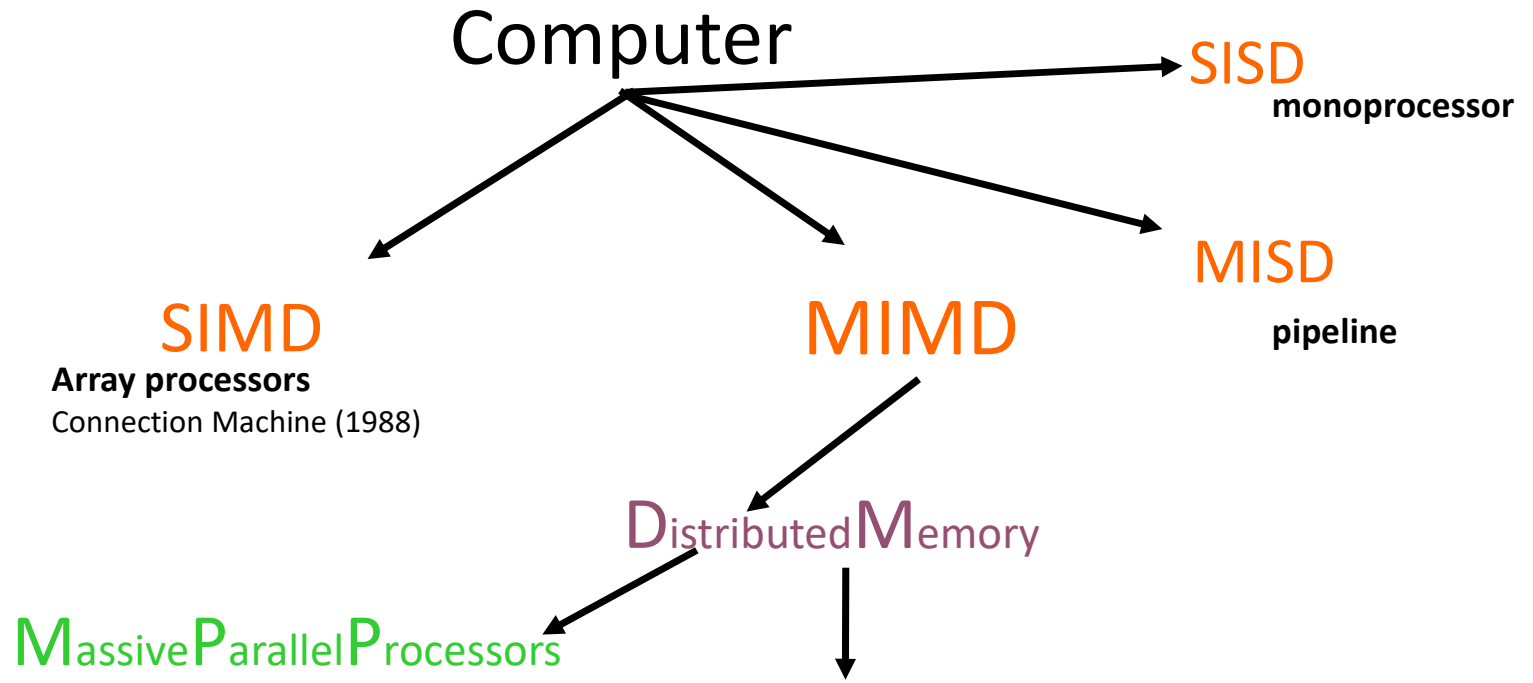
Intel TeraFLOPS MPP architecture
9632 processors
Grid topology
Overall performance: 2.3 Tflop
Peak Performance: 3.2 Tflop

MIMD: distributed memory (DM)

To create a distributed memory MIMD machine that has undemanding production costs, **clusters** are generally used, i.e. sets of autonomous computers connected to each other through the I/O interconnections, and therefore with connectors and cables typical of a standard network.

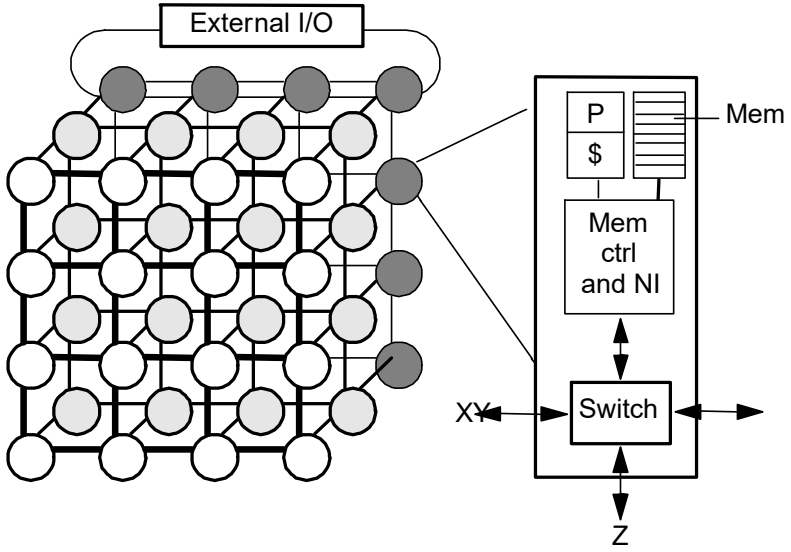
Each computer has its own **separate copy of the operating system**, which **increases the administration costs**, but this drawback can be easily **overcome by using virtual shared memory machines.**

Flynn's taxonomy (since 1966)



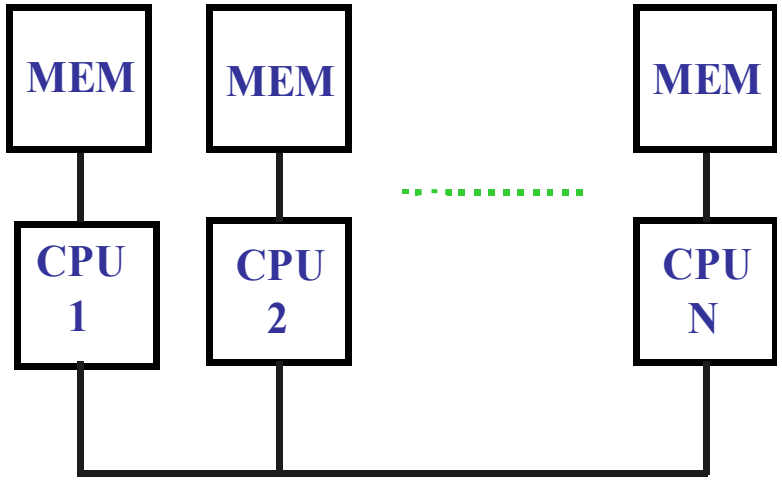
Exemples of computers MIMD-DM-DSM

Cray T3E

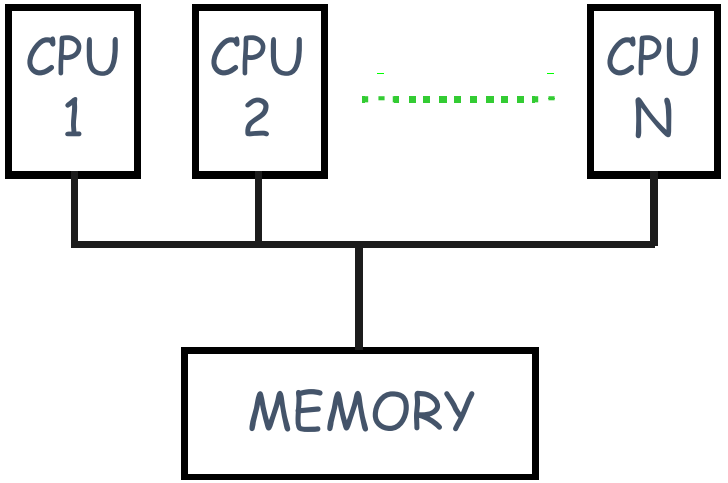


MIMD

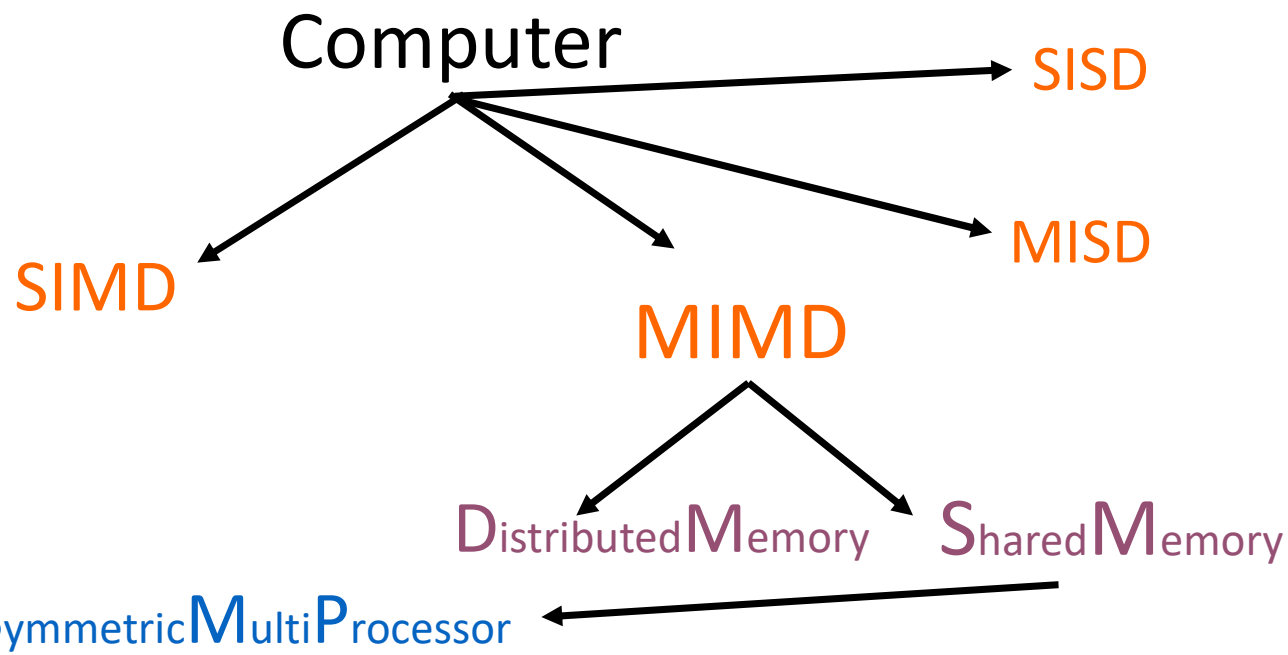
Computer MIMD with distribuited memory (DM)



Computer MIMD with shared memory (SM)



Flynn's taxonomy (since 1966)



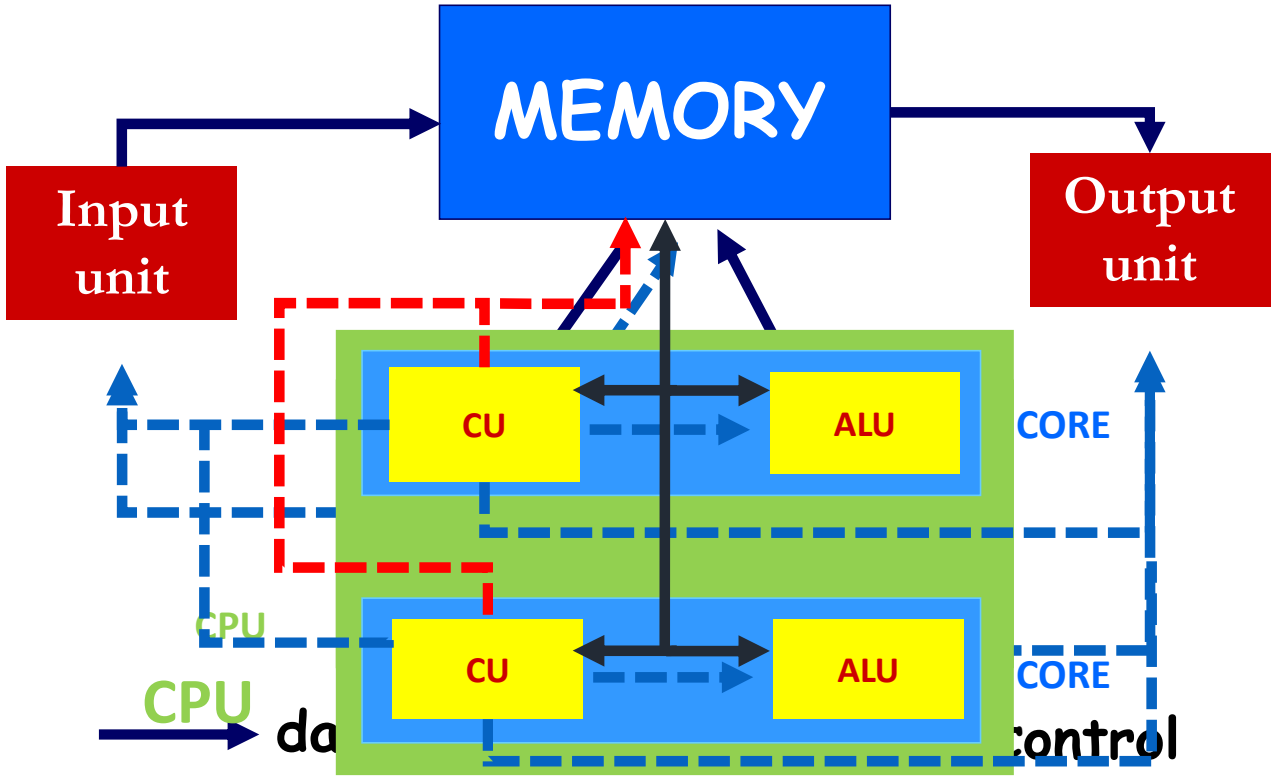
Used to indicate a multiprocessor architecture **in which** there are two or more **identical** processor connected to a single shared memory.
(**MULTICORE INTEL**)

First type of parallelism
on-chip
multiple functional units
within a single ALU

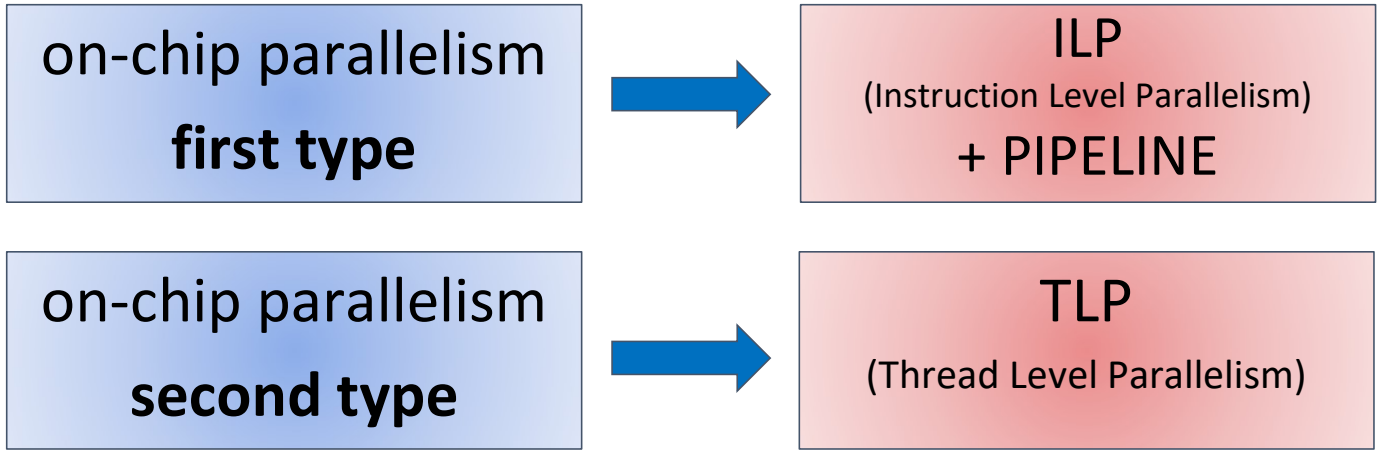
pipeline evolution (temporal parallelism)

Second type of parallelism on-chip
design of microprocessors with more CORE (cpu)
on the same chip

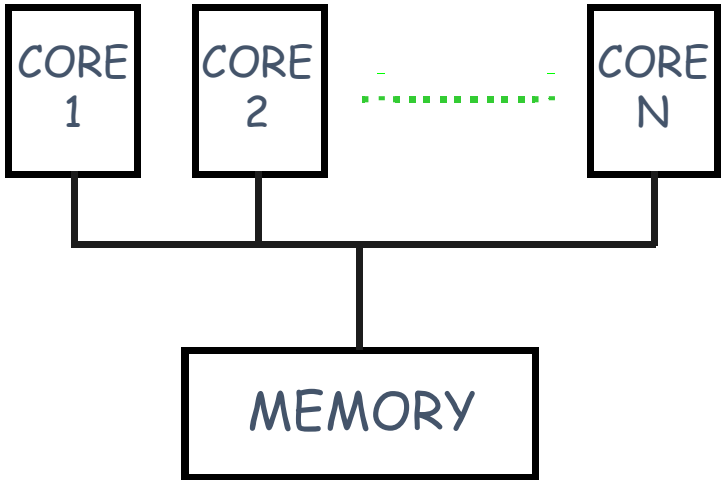
Multicore


MEMORY
Input unit
Output unit
CU
ALU
CORE
CPU
CU
ALU
CORE
CPU
control

Goal: Increase performance

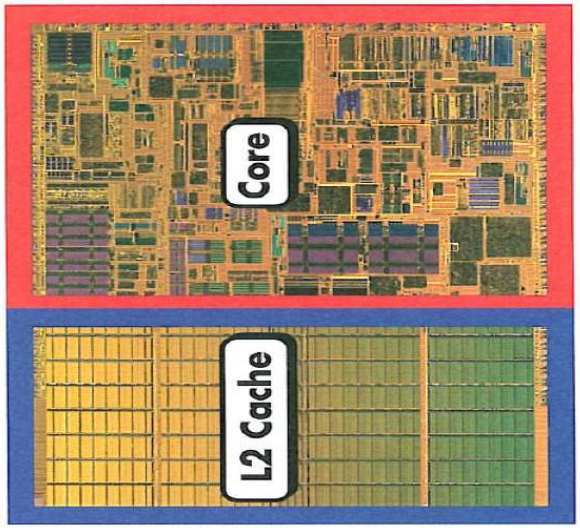


MIMD:
Shared memory (SM) vs distributed memory (DM)

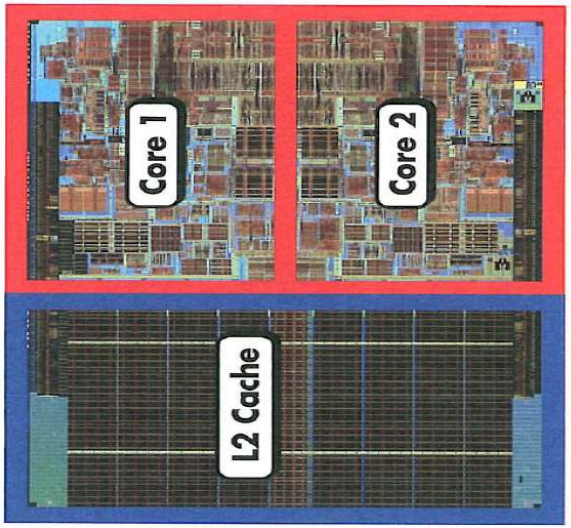


All cores share the memory → To distribute subproblems across threads is more efficient than assigning them to different processes

Multicore: examples



Processore Intel Pentium M (single core)



Processore Intel core duo

Does Moore's law still hold true?

It is clear that to ensure that the prediction to still be valid

(and the the systems growth and potential do not stop)

is undoubtedly the parallelism

MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

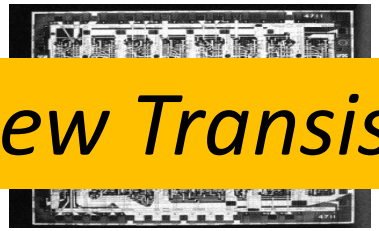
Do you remember the Moore's Law?

Currently
all processors
use different forms of parallelism,
and no system can be defined purely sequential anymore.

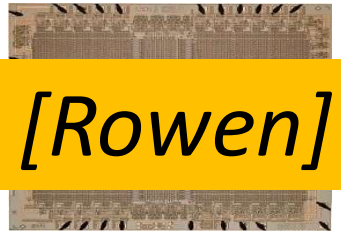
Integrated circuits

"The Processor is the new Transistor" [Rowen]

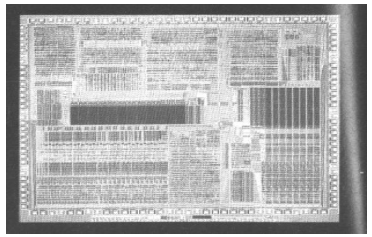
Small Scale Integration
5 transistor
(1964)



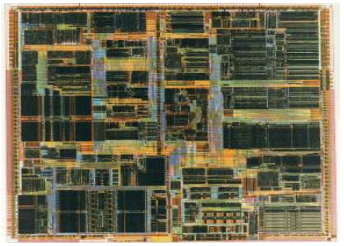
Medium Scale Integration
180 transistor
(1968)



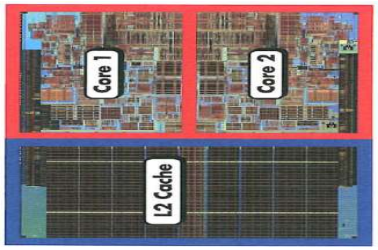
Large Scale Integration
10000 transistor
(1976)



Very Large Scale Integration
132000 transistor
(~1983)



Ultra Large Scale Integration
7500000 transistor
(1997)

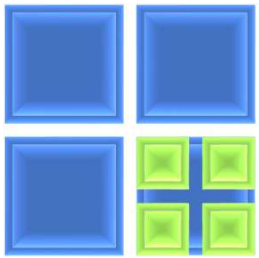


Processore Intel core duo
(2006)

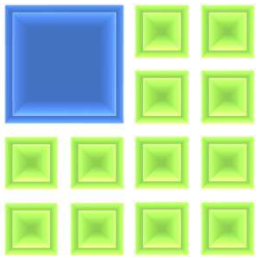
Revisiting of Moore's Law

The **number of cores per chip** will double every two years (instead of the clock rate)

Possible hardware configurations

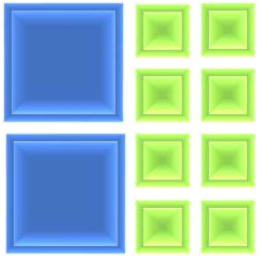


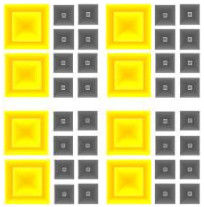
identical core (Intel Quad-Core)



Core of different size and function (IBM/Sony/Toshiba Cell)

Small and equal cores (Sun Niagara)

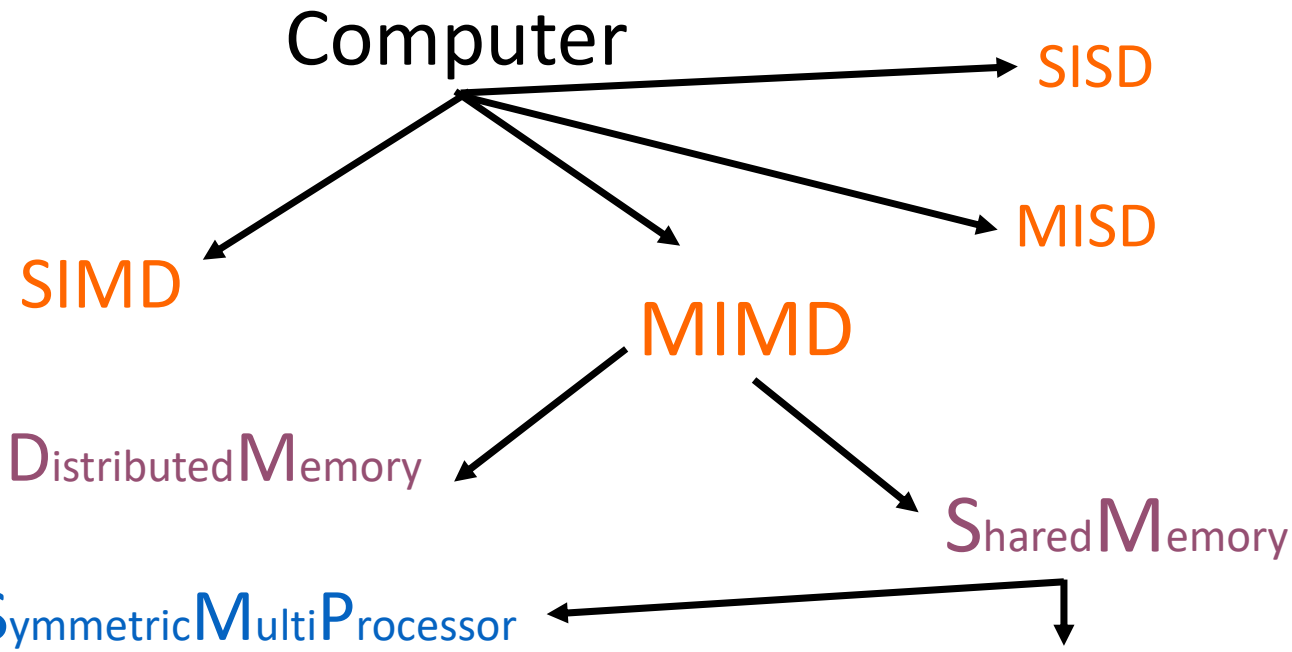




Many small core but different (Nvidia GPU)

Small cores have reduced functionality (E.g .: Single precision arithmetic) or can be grouped into SIMD units

Flynn's taxonomy (since 1966)



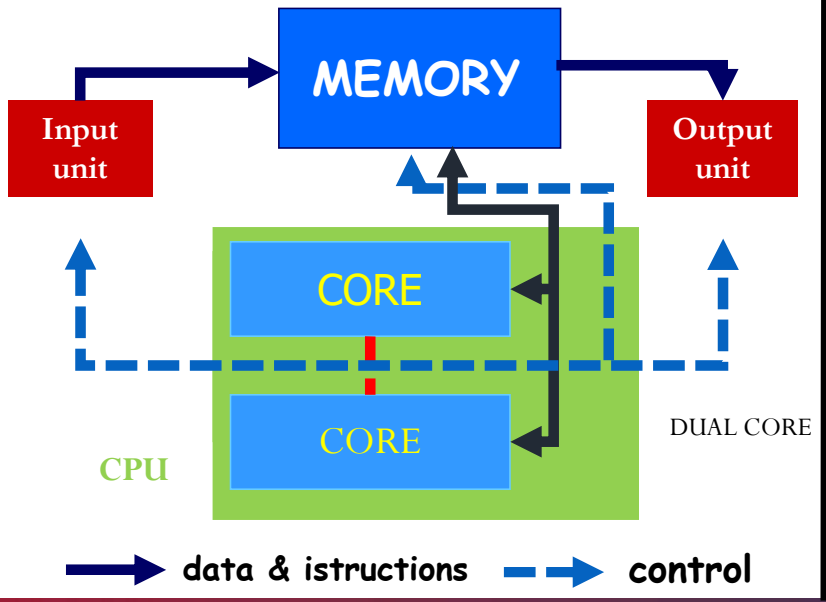
Used to indicate a multiprocessor architecture **in which** there are two or more **identical** processor connected to a single shared memory. (**MULTICORE INTEL**)

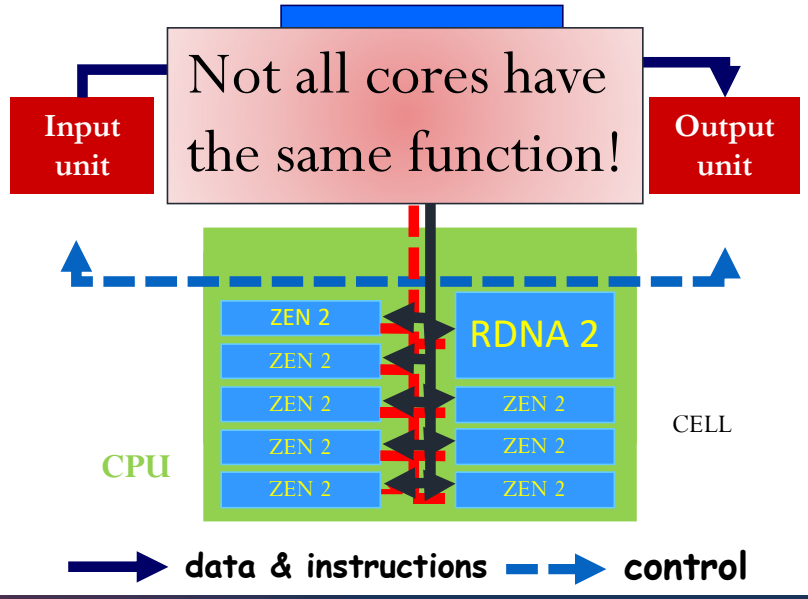
Used to indicate several CPU (or processor unit) with different roles (**cell IBM, GPU (Graphics Processing Unit), nVidia TESLA**)

Exemple of computer MIMD-SM-ASMP

Sony (PlayStation 5) november 2020

- CPU: **8** Zen 2 Cores at 3.5GHz
- GPU: **1** RDNA 2 at 10.28 TFLOPs, 36 CUs at 2.23GHz





GPU

These small computing unit are called CUDA cores (Nvidia GeForce) or Stream Processors (AMD Radeon), depending on the production house

QUADRO: GPU for computing

RDNA 2: GPU for gaming

GPUs are essentially the same. What differs is:

- the size of memory: **game cards have less memory**, since games do not require as much memory as modeling programs.
- the type of cooling: gaming cards are **quieter**, while computing cards are **noisier**.
- the drivers: the **compute cards aim to create 100% accurate images**, performing double precision calculations at full speed, the **game cards** instead **aim to increase the performance**, even applying small tricks to reduce the accuracy of the rendering, that you would only notice if you analyzed the image frame by frame and zoomed in very closely.

The Graphics Processing Units (GPUs) which today have gained so much fam, born (in **2000**) to be destined for graphic processing.

Their architecture has evolved very rapidly, and **it combines almost all the types of parallelism illustrated.**

Let's take a break

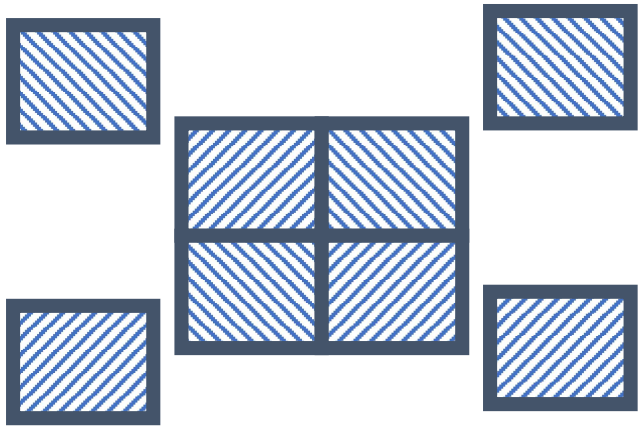
PARALLEL COMPUTING

Decompose a problem

in more subproblems

and solve them at the same time

with more processing units!



Need to create machines that can distribute the work among them
hardware development

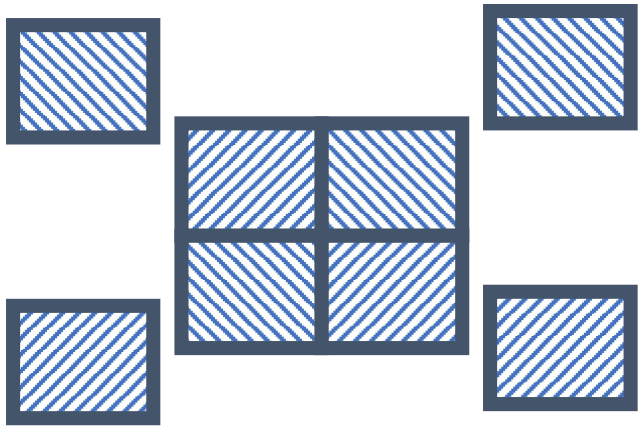
PARALLEL COMPUTING

Decompose a problem

in more subproblems

and solve them at the same time

with more processing units!



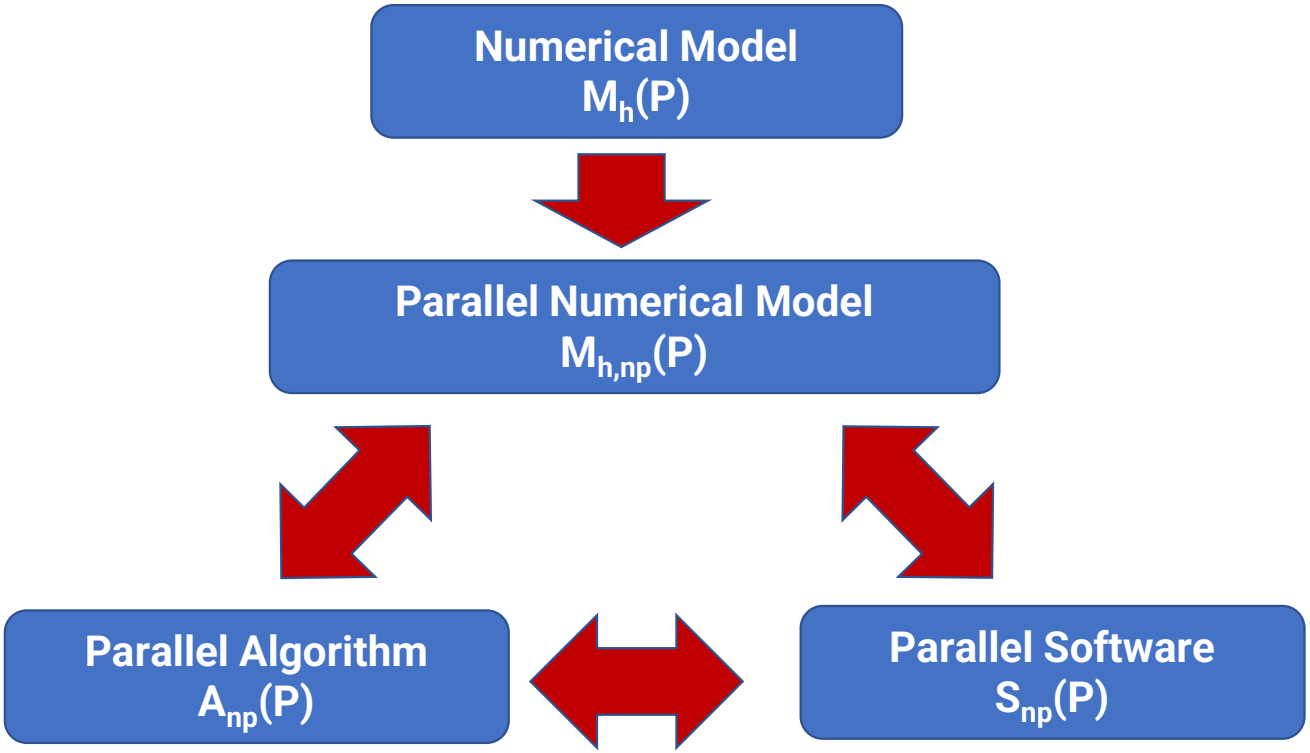
Need to create machines that can distribute the work among them
hardware development

MEIM MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Thinking parallel

What does parallel thinking mean?



What does parallel thinking mean?

Numerical Model
$M_h(P)$

Restart from numerical model (mathematical model for computer **h=1,…,N**) and analyze the **N** steps in order to distribute them, possibly, to several processing units.
More options:

- **each processing unit** performs a **different** step
  (**functional decomposition**)

What does parallel thinking mean?

Numerical Model
$M_h(P)$

Restart from numerical model (mathematical model for computer **h=1,…,N**) and to analyze in order to **find independent task that can be processed separately and simultaneously.**

Esempio:

To make a cream cake

h=1 make the cream
h=2 make the sponge cake
h=3 add the cream to the sponge cake

… what if we are two?

What does parallel thinking mean?

Numerical Model
$M_h(P)$

Restart from numerical model (mathematical model for computer **h=1,…,N**) and to analyze in order to **find independent task that can be processed separately and simultaneously.**

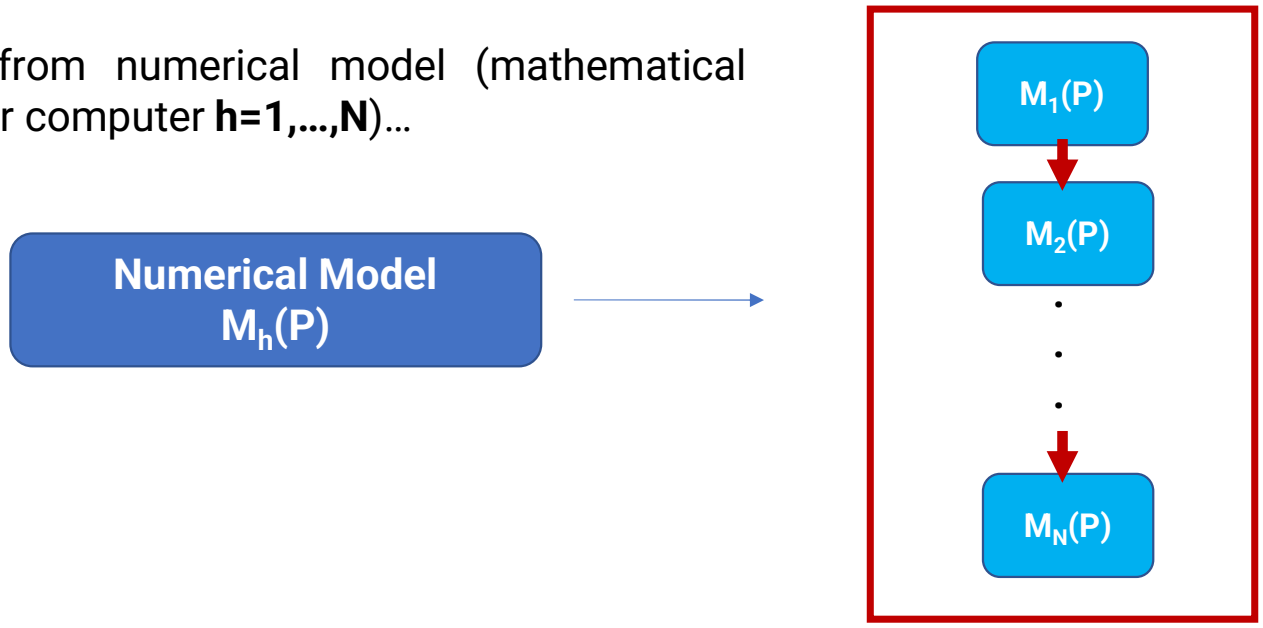
Esempio: two people (two executors)

To make a cream cake

h=1	make the cream	**h=2**	make the sponge cake
h=3	add the cream to the sponge cake		

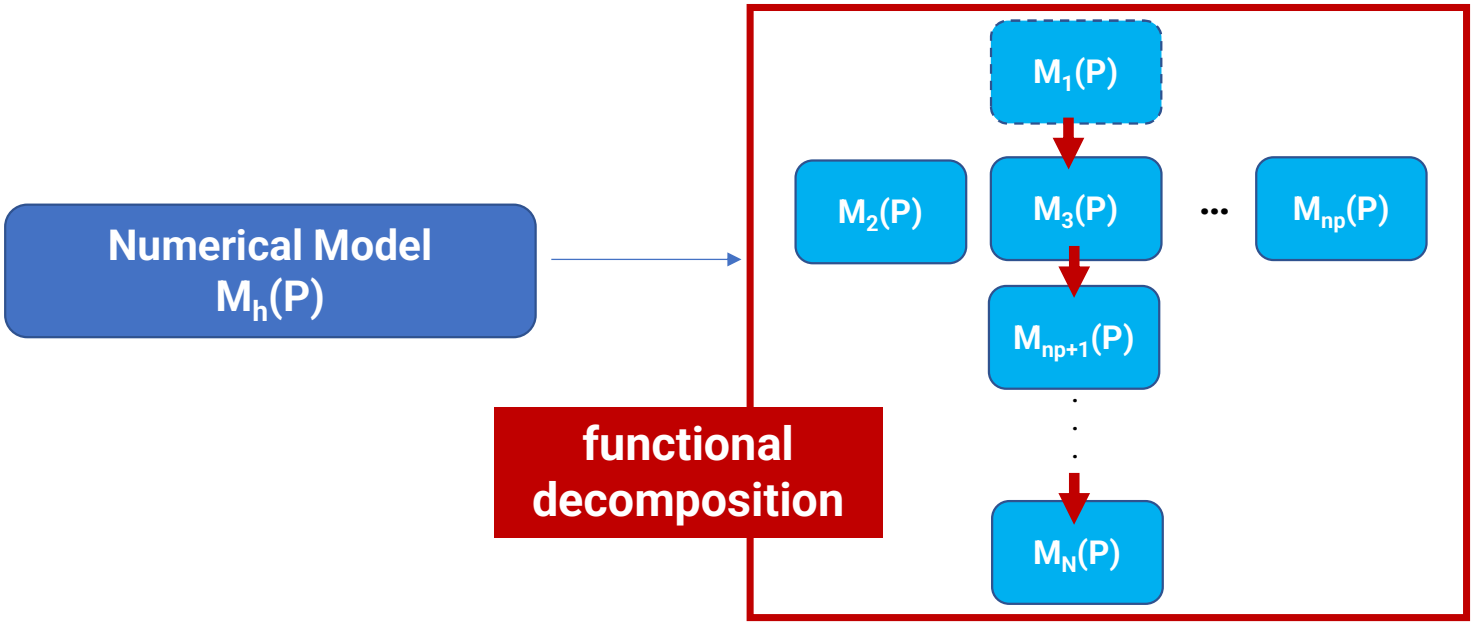
What does parallel thinking mean?

Restart from numerical model (mathematical model for computer **h=1,…,N**)…



What does parallel thinking mean?

Restart from numerical model (mathematical model for computer **h=1,…,N**) and to analyze in order to **find independent task that can be processed separately and simultaneously.**



Functional Decomposition

Restart from numerical model (mathematical model for computer **h=1,…,N**) and to analyze in order to **find independent task that can be processed separately and simultaneously.**

Functional Decomposition = **to perform different and independent computations simultaneously**

Characteristics:

- scalable with the number of independent processing units
- helpful only for sufficiently complex elaborations
- applicable for procedure characterized by multiple computational basic cores

What does parallel thinking mean?

Numerical Model
$M_h(P)$

Restart from numerical model (mathematical model for computer **h=1,…,N**) and analyze the **N** steps in order to distribute them, possibly, to several processing units.

More options:

- **each processing unit** performs a **different** step
  (**functional decomposition**)
- **all processing units** perform the **same** operation on a **different** subset of data
  (**domain decomposition**)

What does parallel thinking mean?

Numerical Model
$M_h(P)$

Restart from numerical model (mathematical model for computer **h=1,…,N**) and **split each task into several equal sub-tasks and process them simultaneously, but minimizing the collection of local results.**

Esempio:

To make a cream cake

h=1 make the cream
h=2 make the sponge cake
h=3 add the cream to the sponge cake

… what if we are two?

What does parallel thinking mean?

Numerical Model
$M_h(P)$

Restart from numerical model (mathematical model for computer **h=1,…,N**) and **split each task into several equal sub-tasks and process them simultaneously, but minimizing the collection of local results.**

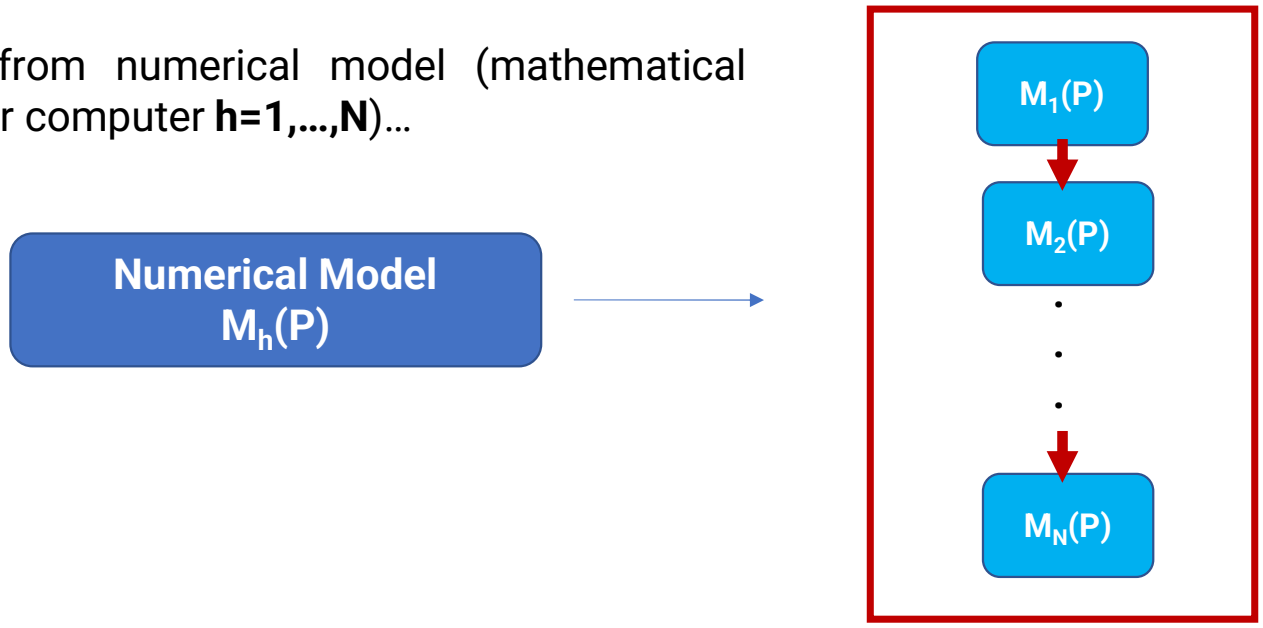
Esempio: two people (two executors)

To make a cream cake

h=1.1+1.2	**together we prepare the cream**
h=2.1+2.2	**together we prepare the sponge cake**
h=3.1+3.2	**together we add the cream to the sponge cake**

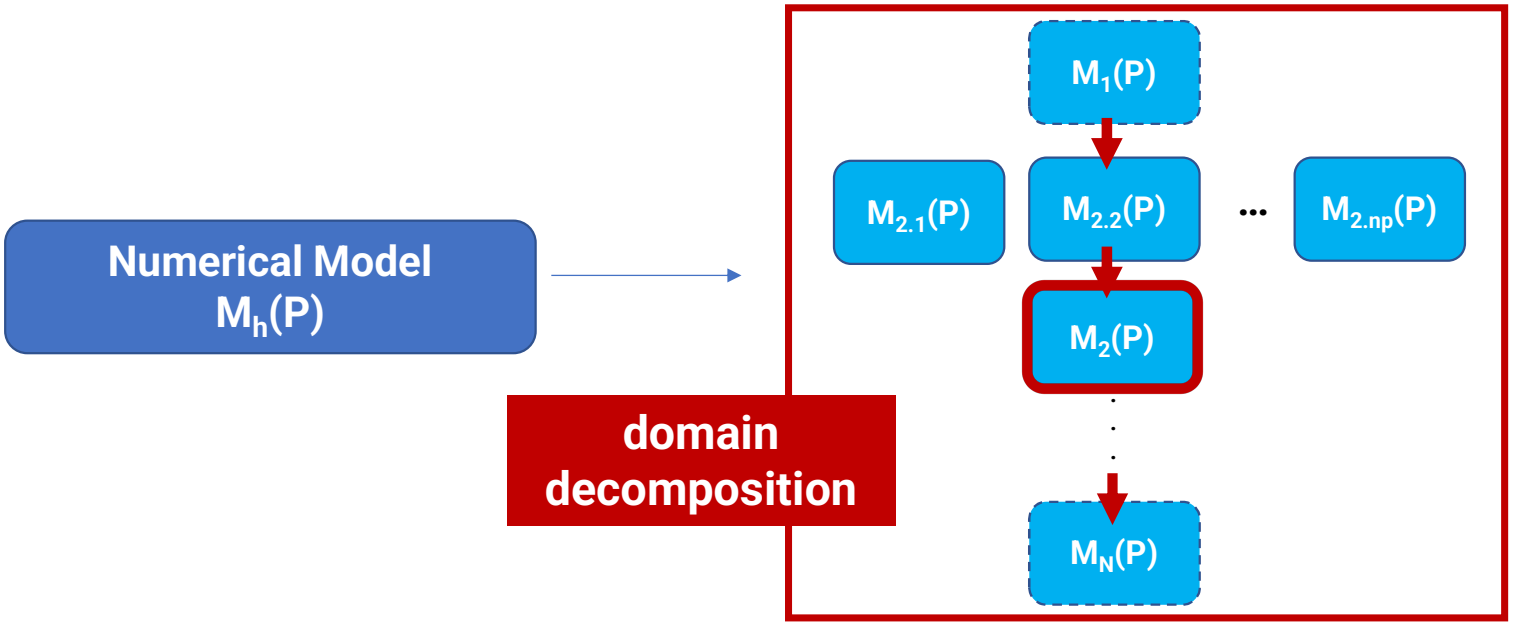
What does parallel thinking mean?

Restart from numerical model (mathematical model for computer **h=1,…,N**)…



What does parallel thinking mean?

Restart from numerical model (mathematical model for computer **h=1,…,N**) and **split each task into several equal sub-tasks and process them simultaneously, but minimizing the collection of local results.**



Domain Decomposition

Restart from numerical model (mathematical model for computer **h=1,…,N**) and **split each task into several equal sub-tasks and process them simultaneously, but minimizing the collection of local results.**

Domain Decomposition = **to split the data and process them all in the same way.**

Characteristics:

- scalable with the number of data
- helpful only when there is a simple collection of local results
- applicable for procedures with several consecutive computational cores

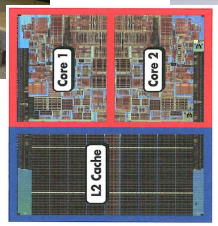
What does parallel thinking mean?

Numerical Model
$M_h(P)$

Restart from numerical model (mathematical model for computer **h=1,…,N**) and analyze the **N** steps in order to distribute them, possibly, to several processing units.

More options:

- **each processing unit** performs a **different** step
  (**functional decomposition**)
- **all processing units** perform the **same** operation on a **different** subset of data
  (**domain decomposition**)
- **combination of the two previous possibilities**



Flynn's taxonomy

That's all for today!

Thinking parallel