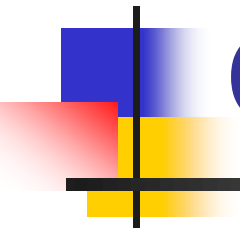
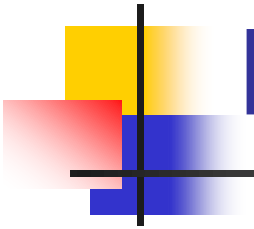


Introduction to Parallel Computing

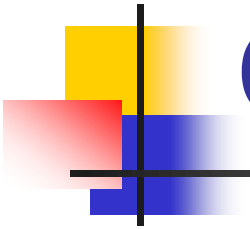
Dra. Mireya Paredes López

mireya.paredes@udlap.mx



References

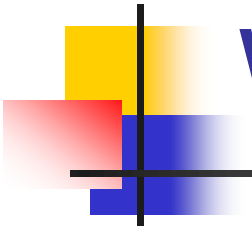
- Notes from Introduction to Parallel Computing, Blaise Barney, Lawrence Livermore National Laboratory
- Notes from Introducción al Cómputo Paralelo, Arturo Díaz Pérez, CINVESTAV



Overview

- What is parallel computing?
- Why use parallel computing?
- Who is using parallel computing?

What is Parallel Computing?

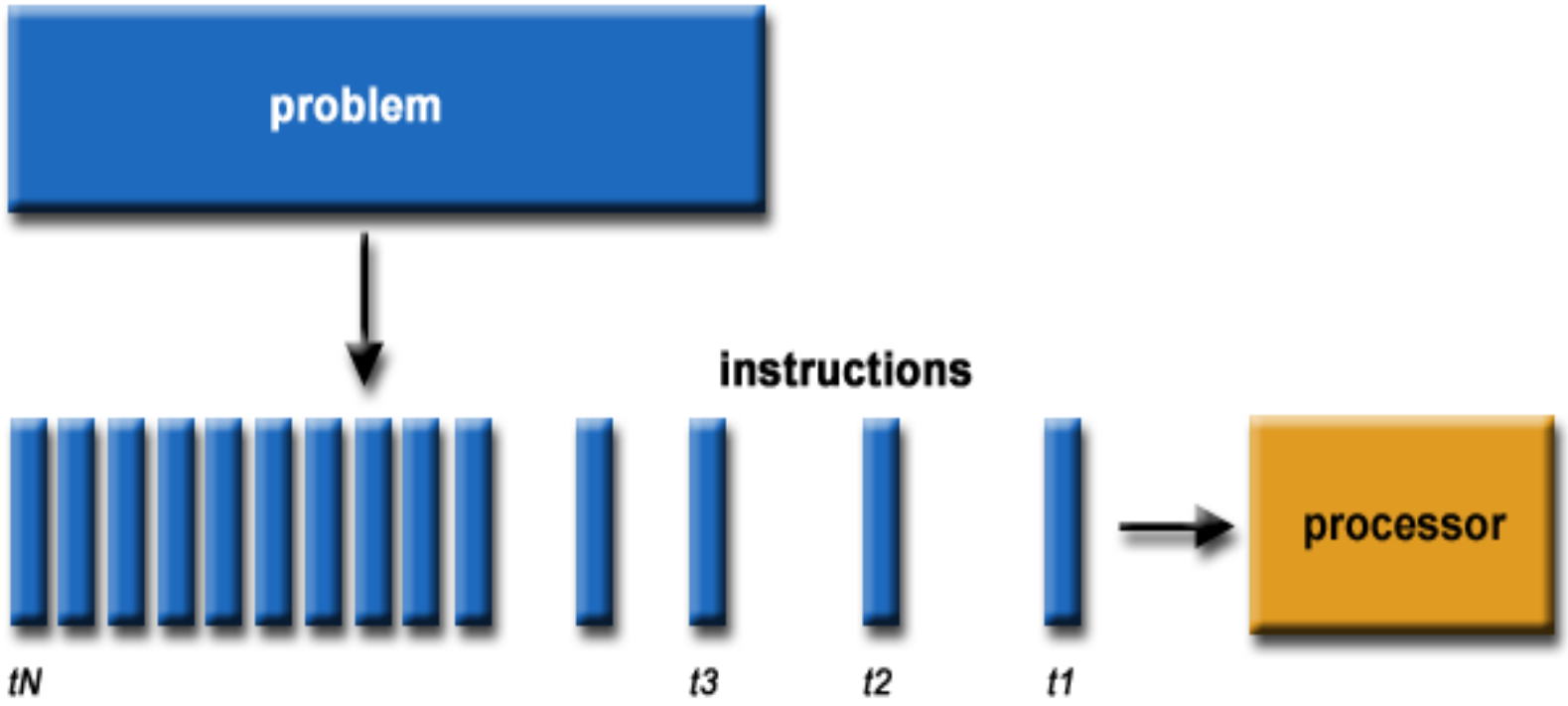


Serial Computing:

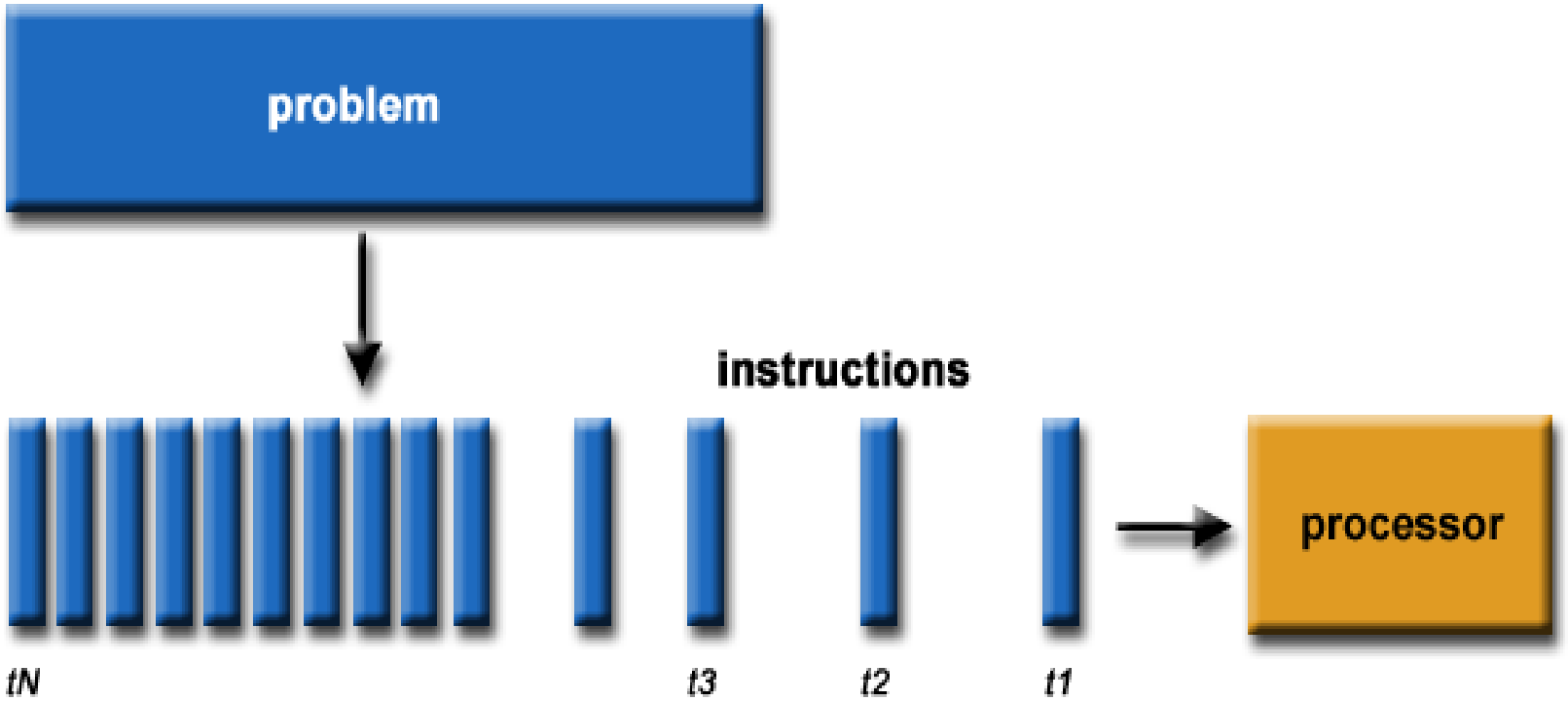
Traditionally, software has been written for serial computation:

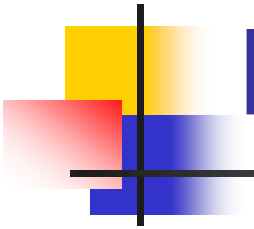
- A problem is broken into a discrete series of instructions.
- Instructions are executed sequentially one after another.
- Executed on a single processor.
- Only one instruction may execute at any moment in time.

Serial computing



Serial computing



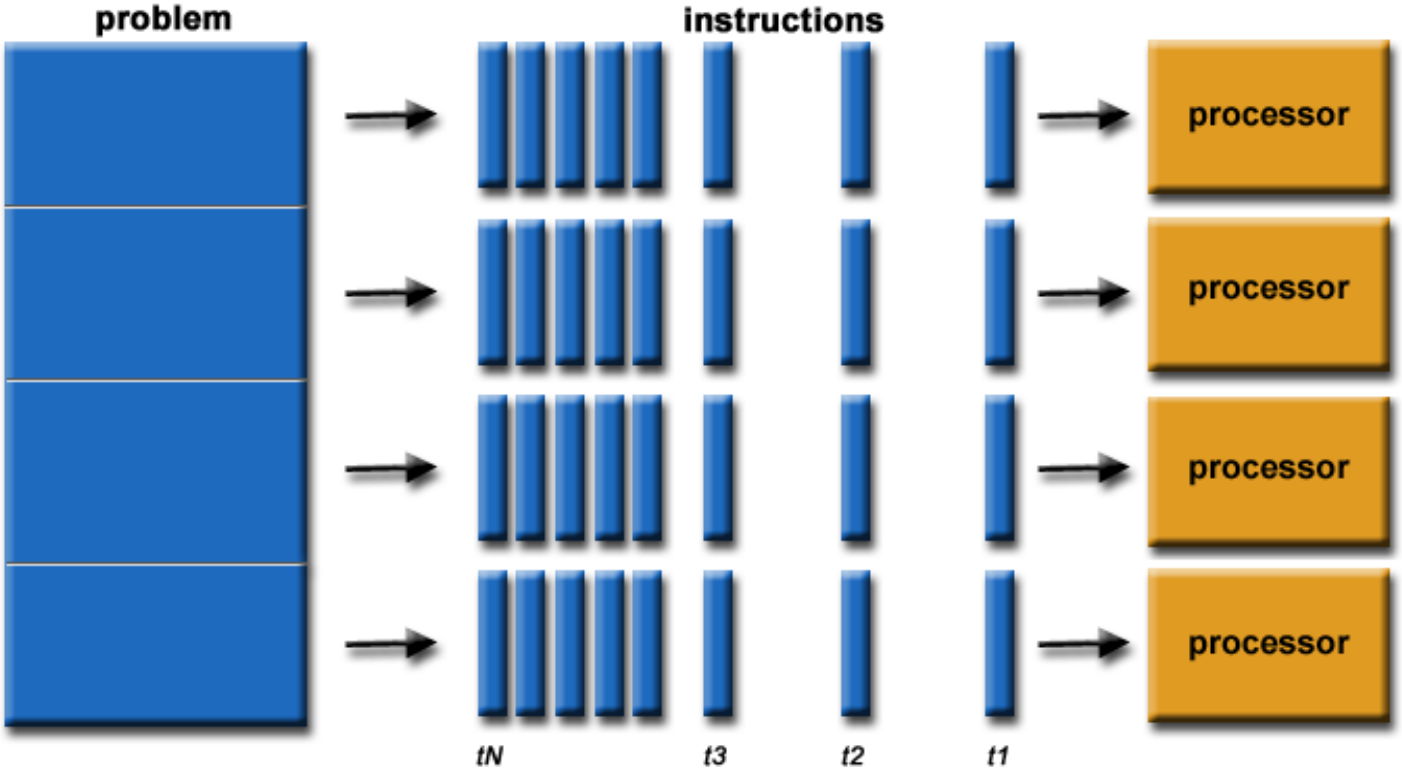


Parallel Computing

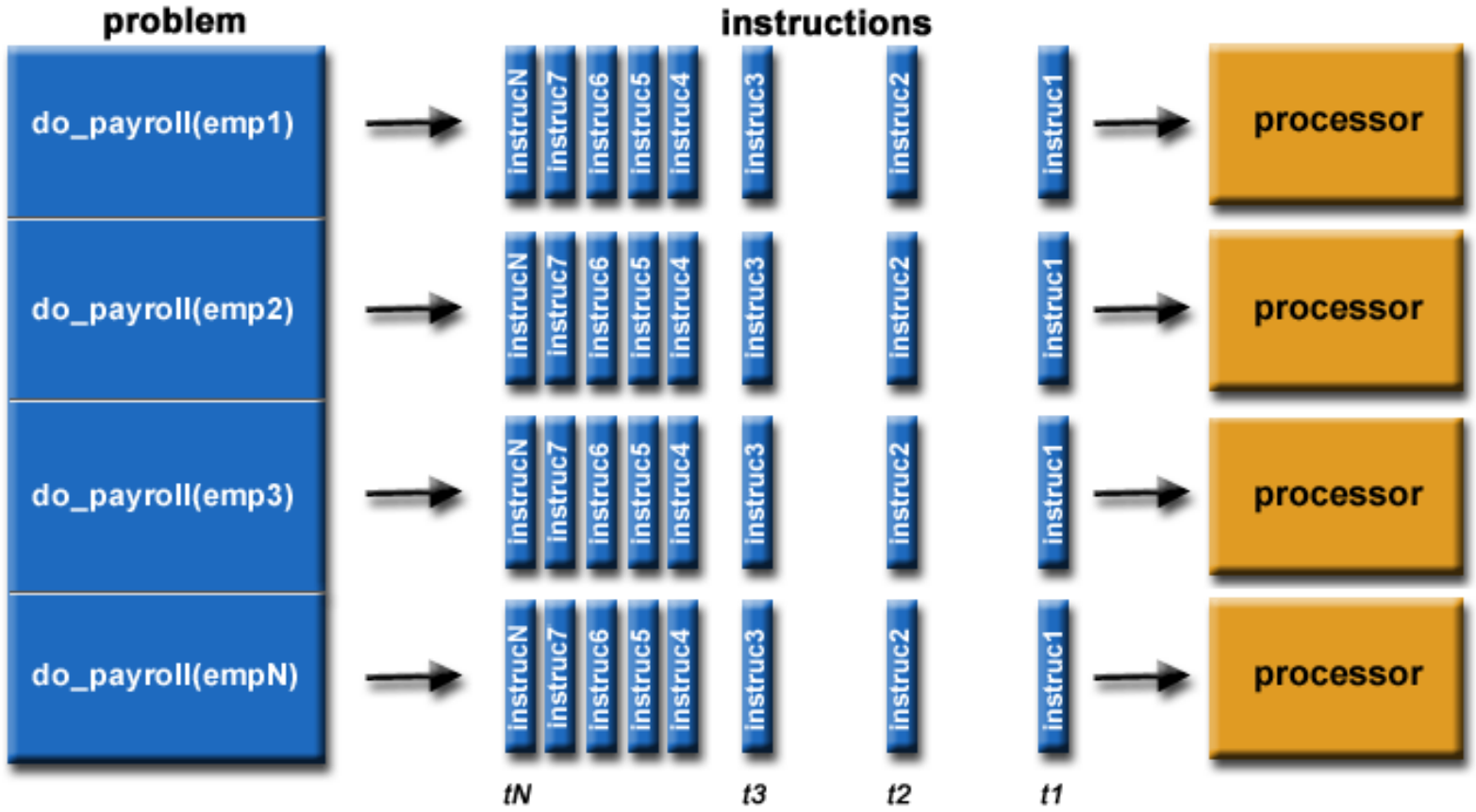
Parallel computing: It is the simultaneous use of multiple compute resources to solve a computational problem:

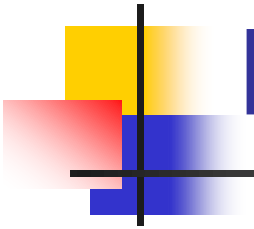
- A problem is broken into discrete parts that can be solved concurrently.
- Each part is further broken down to a series of instructions.
- Instructions from each part execute simultaneously on different processors.
- An overall control/coordination mechanism is employed.

Parallel computing



Parallel computing

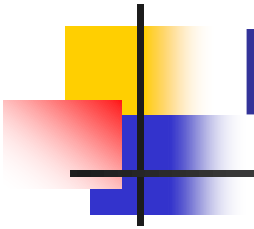




Parallel Computing

The computational problem should be able to:

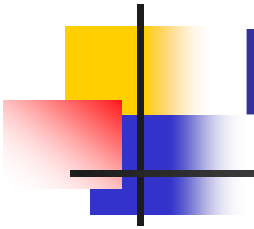
- Be broken apart into discrete pieces of work that can be solved simultaneously.
- Execute multiple program instructions at any moment in time.
- Be solved in less time with multiple compute resources than with a single compute resource.



Parallel Computing

The compute resources are typically:

- A single computer with multiple processors/cores.
- An arbitrary number of such computers connected by a network.



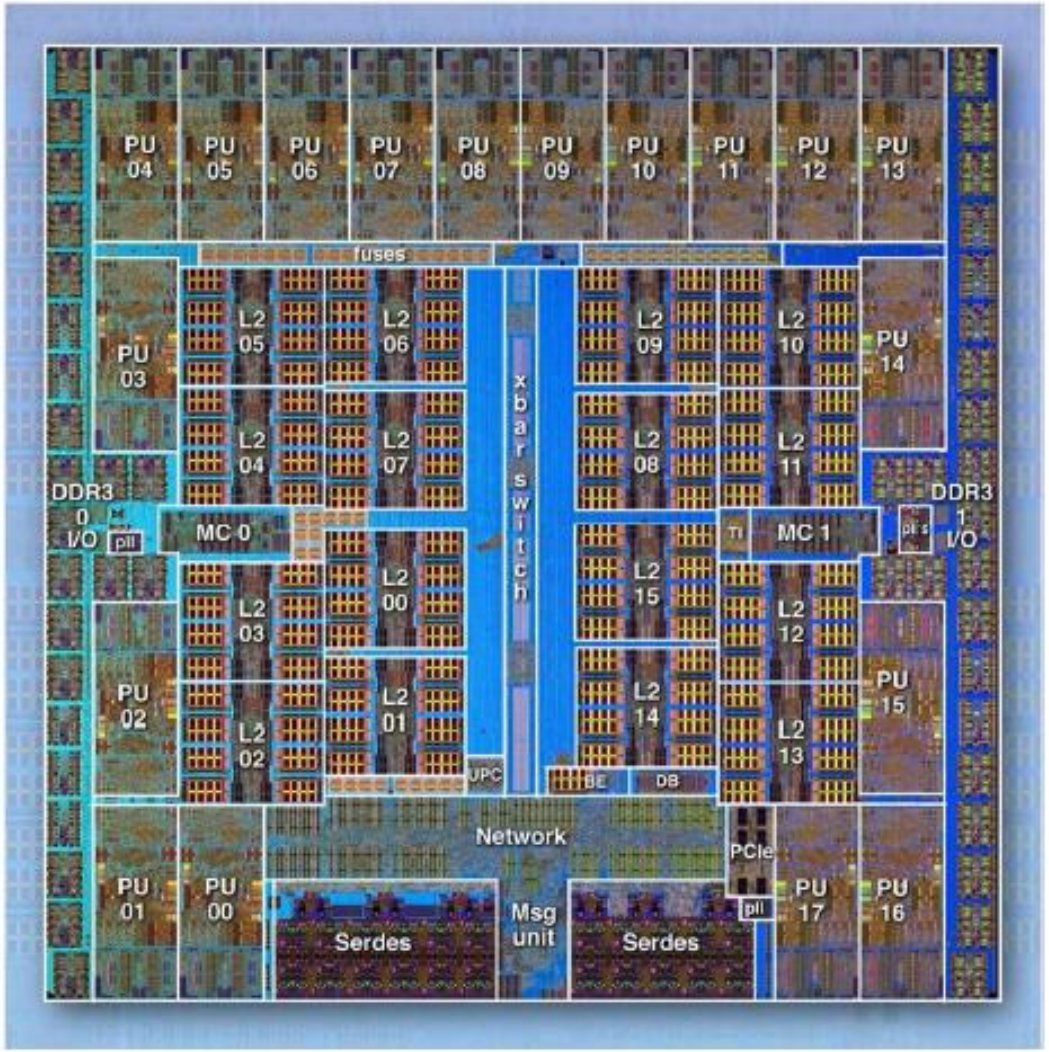
Parallel Computers

Virtually all stand-alone computers today are parallel from a hardware perspective:

- Multiple functions units (L1 cache, L2 cache, prefetch, decode, floating-point, graphics processing (GPU), etc.)
- Multiple executions units/cores.
- Multiple hardware threads.

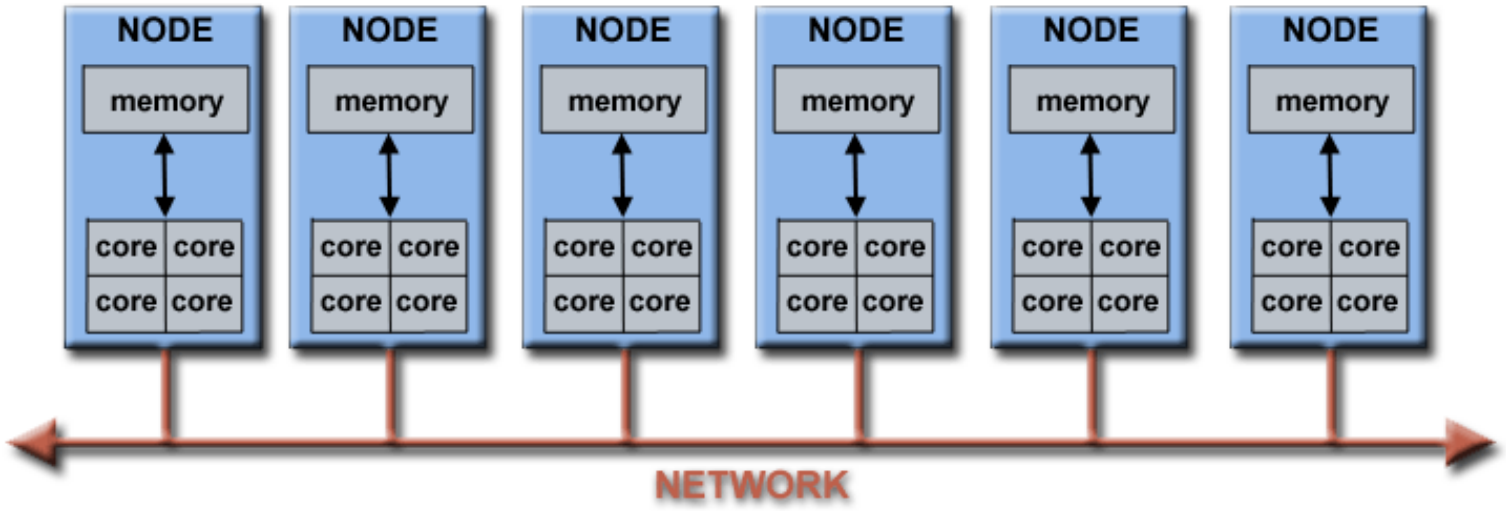
BlueGene/Q Compute chip

System-on-a-Chip design : integrates processors, memory and networking logic into a single chip



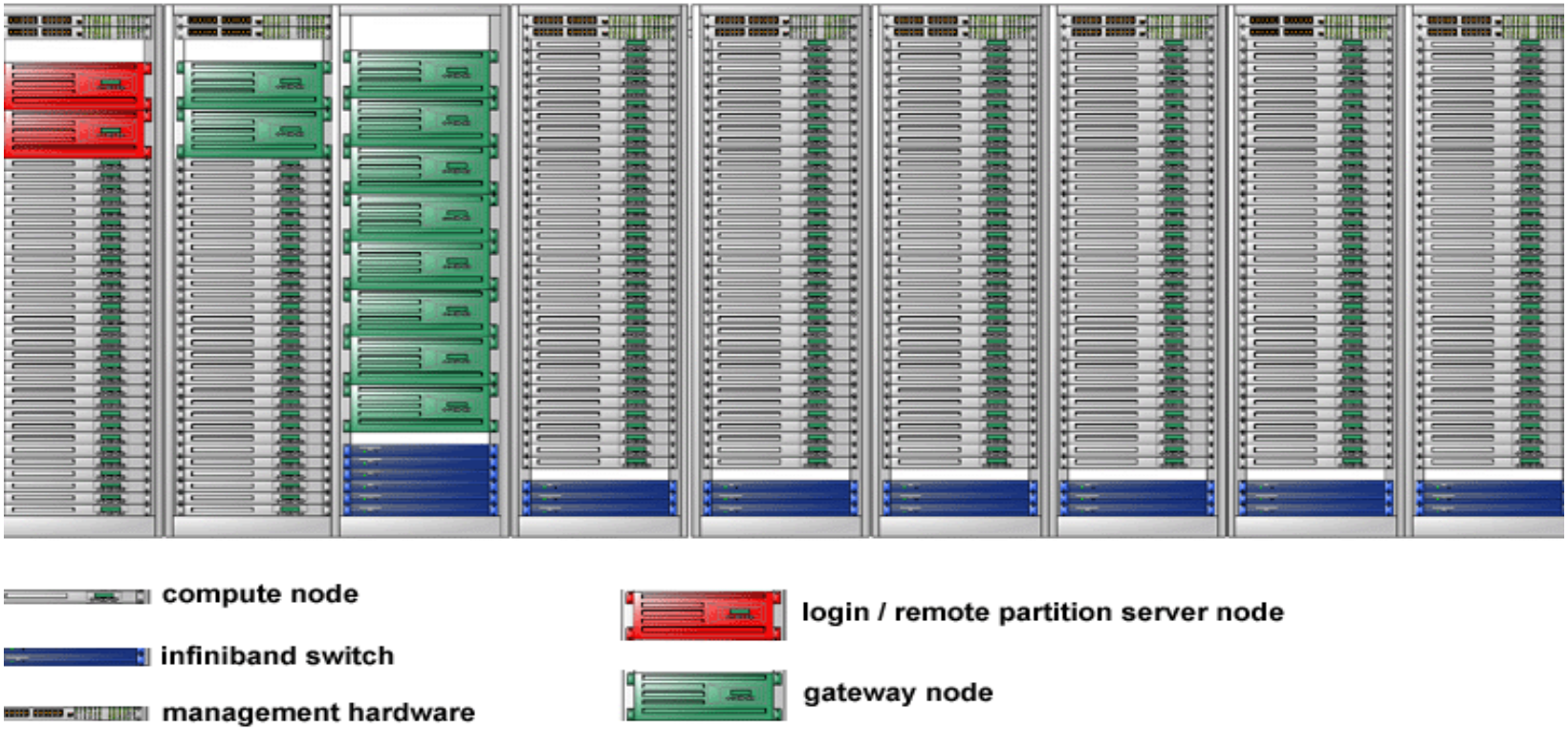
Parallel computer cluster

Multiple stand-alone computers (nodes) to make larger parallel Computer clusters.

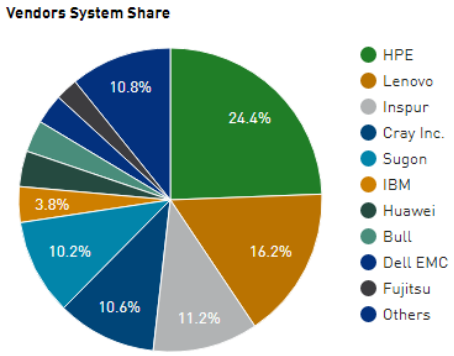


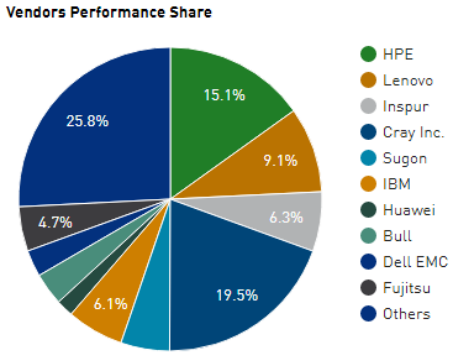
Typical parallel computer cluster

Each computer node is a multi-processor parallel computer in itself.
Multiple compute nodes are networked together with an Infiniband network.
Special purpose nodes, alto multi-processor, are used for other purposes.



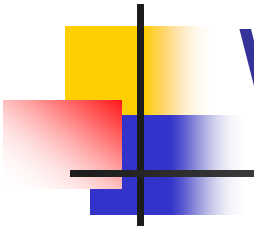
Well known vendors





Vendors	Count	System Share (%)
HPE	122	24.4
Lenovo	81	16.2
Inspur	56	11.2
Cray Inc.	53	10.6
Sugon	51	10.2
IBM	19	3.8
Huawei	19	3.8
Bull	17	3.4
Dell EMC	16	3.2
Fujitsu	12	2.4
Penguin Computing	10	2
NUDT	4	0.8
PEZY Computing / Exascaler Inc.	4	0.8
NEC	4	0.8
Atipa	3	0.6
Lenovo/IBM	3	0.6
Nvidia	2	0.4
Dell EMC / IBM-GBS	2	0.4
T-Platforms	2	0.4
IBM/Lenovo	2	0.4
Self-made	1	0.2
T-Platforms, Intel, Dell	1	0.2
ClusterVision	1	0.2
Supermicro	1	0.2
ExaScaler	1	0.2
E4 Computer Engineering S.p.A.	1	0.2
RSC Group	1	0.2

Taken from Top500.org



Why Use Parallel Computing?

The Real World is Massively Parallel:

- In the natural world, many complex, interrelated events are happening at the same time, yet within a temporal sequence.
- Compared to serial computing, parallel computing is much better suited for modeling, simulating and understanding complex, real world phenomena.

The World is Massively Parallel

Galaxy Formation

Planetary Movments

Climate Change

Rush Hour Traffic

Plate Tectonics

Weather

The World is Massively Parallel

Auto Assembly

Jet Construction

Drive-thru Lunch

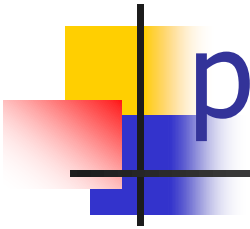
Save time and/or money

- In theory, throwing more resources at a task will shorten its time to completion, with potential cost savings.
- Parallel computers can be built from cheap, commodity components.

Solve larger/more complex problems

- Many problems are so large and/or complex that it is impractical or impossible to solve them on a single computer, especially given limited

Solve larger/more complex problems



- Example "Grand Challenge Problems" n.wikipedia.org/wiki/Grand_Challenge)

- Example: Web search engines/databases processing millions of tansactions every second.

Provide concurrency:

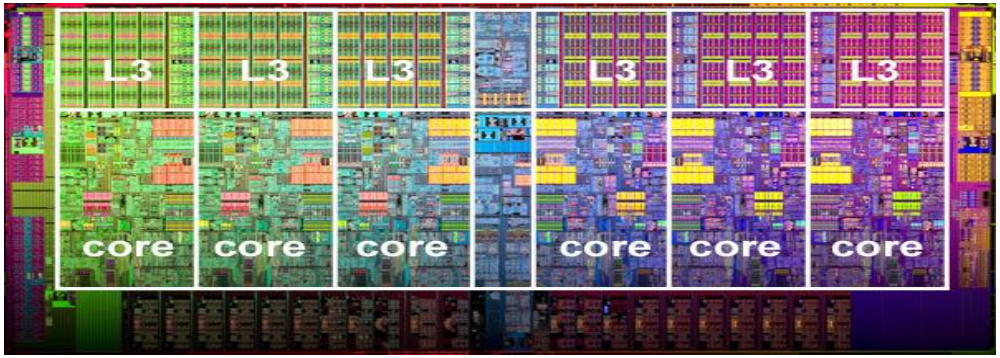
- A single computer resource can only do one thing at a time. Multiple compute resources can do many things simultaneously.
- Example: Collaborative Networks provide a global venue where people from around the world can meet and conduct work “virtually”.

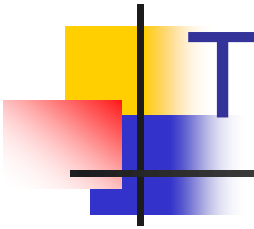
Take advantage of non-local resources:

- Using compute resources on a wide area network, or even the Internet when local compute resources are scarce or insufficient.
- Example: SETI@home (setiathome.berkeley.edu)
- Example: Folding@home (folding.standford.edus)

Make better use of underlying parallel hardware:

- Modern computers, even laptops, are parallel architecture with multiple processors/cores.
- Parallel software is specially intended for parallel hardware with multiple cores, threads, etc.
- In most cases, serial programs run on modern computers "waste" potential computing power.





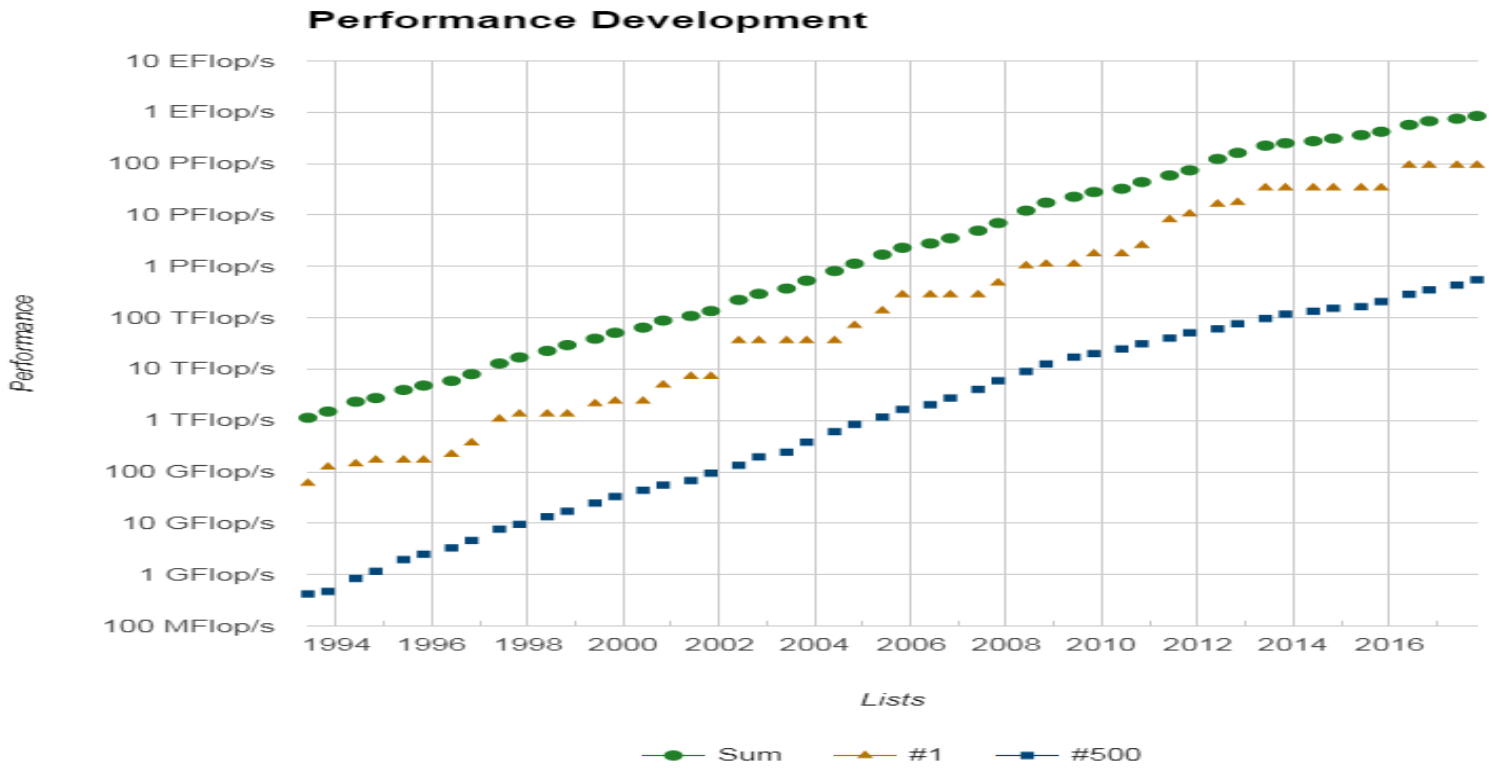
The Future:

- During the past 20+ years, the trends indicated by ever faster networks, distributed systems, and multi-processor computer architectures (even at the desktop level) clearly show that ***parallelism is the future of computing.***
- In this same time period, there has been a greater than **500,000x** increase in supercomputer performance, with no end currently in sight.

Performance – Top500

The race is already for **Exascale Computing!**

Exaflop = 10^18 calculations per second

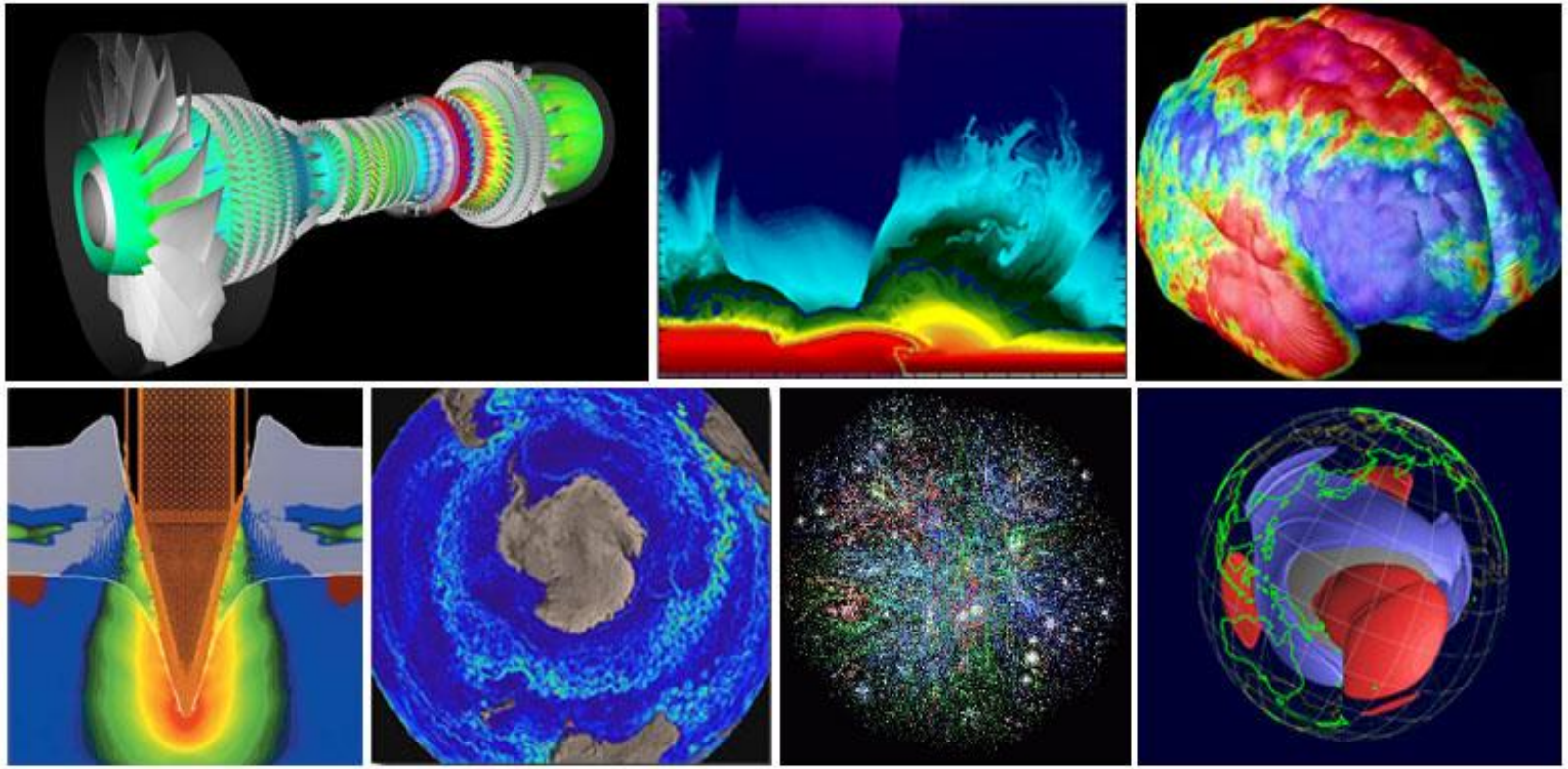


Who is using parallel computing?

"The high end of computing" → model difficult problems:

- Atmosphere, Earth, Environment
- Physics – applied, nuclear, particle, condensed matter,
- High pressure, fusion, photonics
- Bioscience, biotechnology, geneticsw
- Chemistry, molecular sciences
- Geology, seismology
- Mechanical Engineering – spacecraft
- Computer Science, Mathematics
- Defense, Weapons

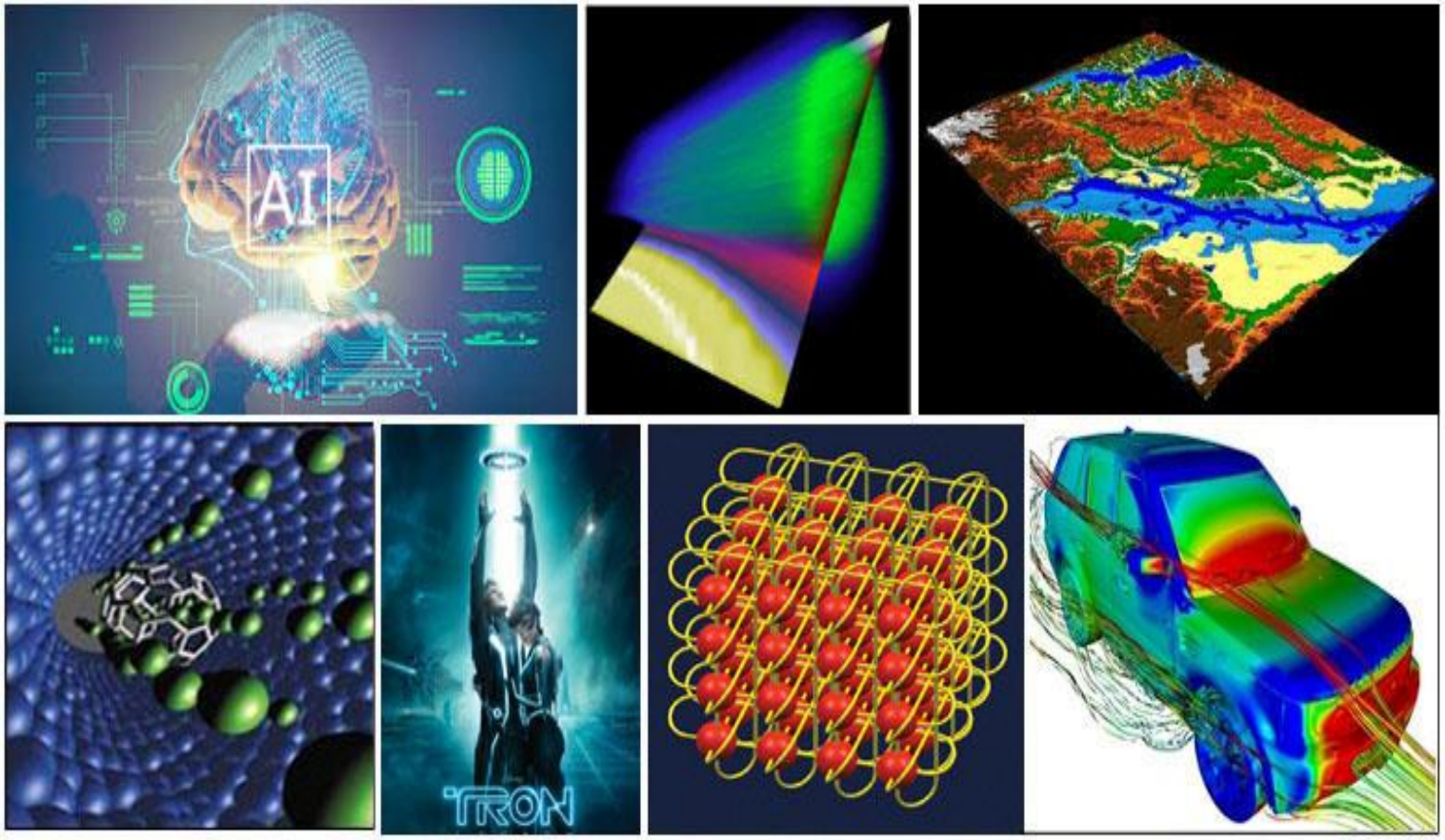
Science and Engineering



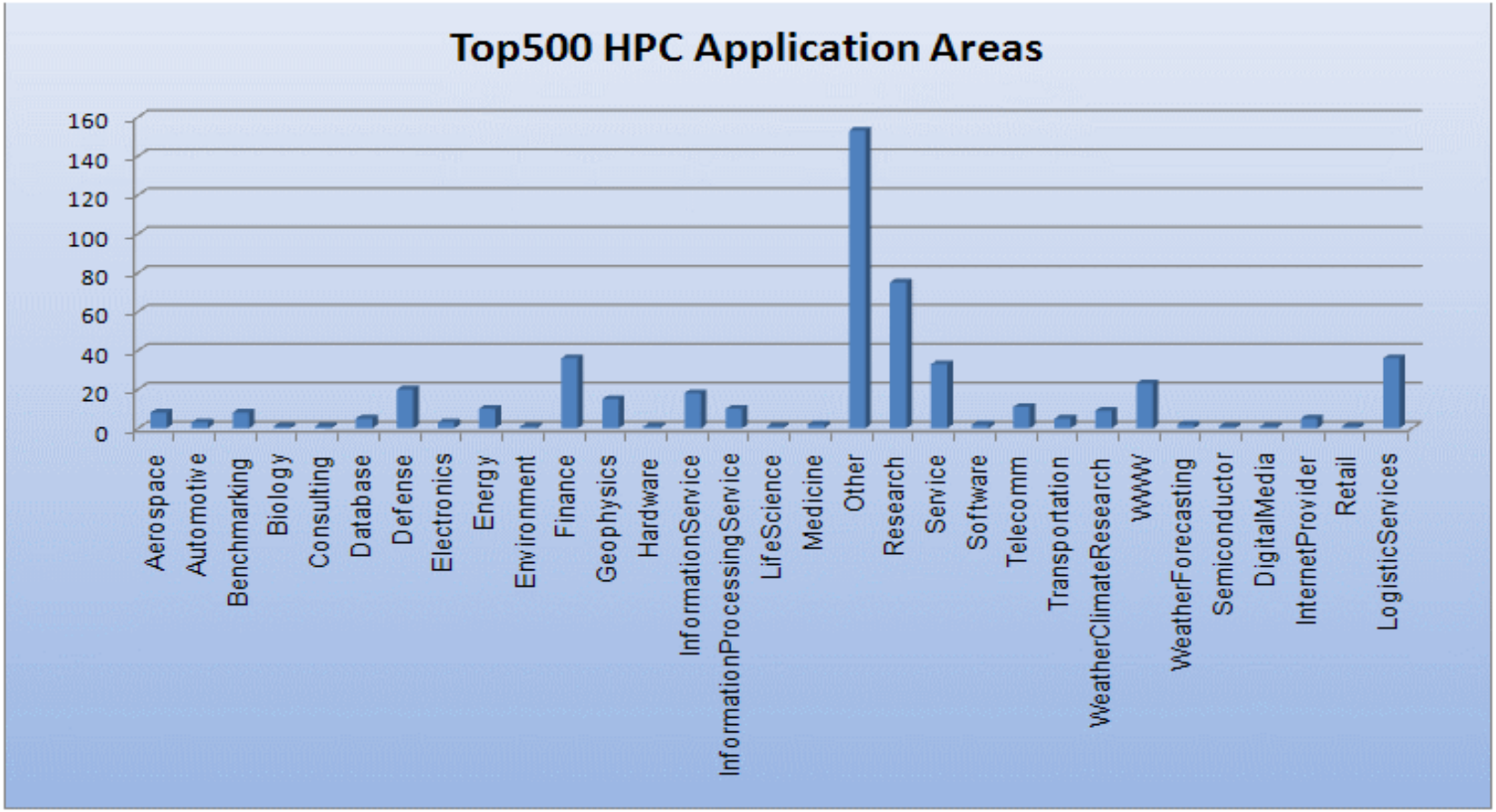
Industrial and Commercial

- “Big data”, databases, data mining
- Artificial Intelligence (AI)
- Web search engines, web based business services
- Medical imaging and diagnosis
- Financial and economic modeling
- Advanced graphics and virtual reality, particularly in the entertainment industry.
- Networked video and multi-media technologies.
- Oil exploration.

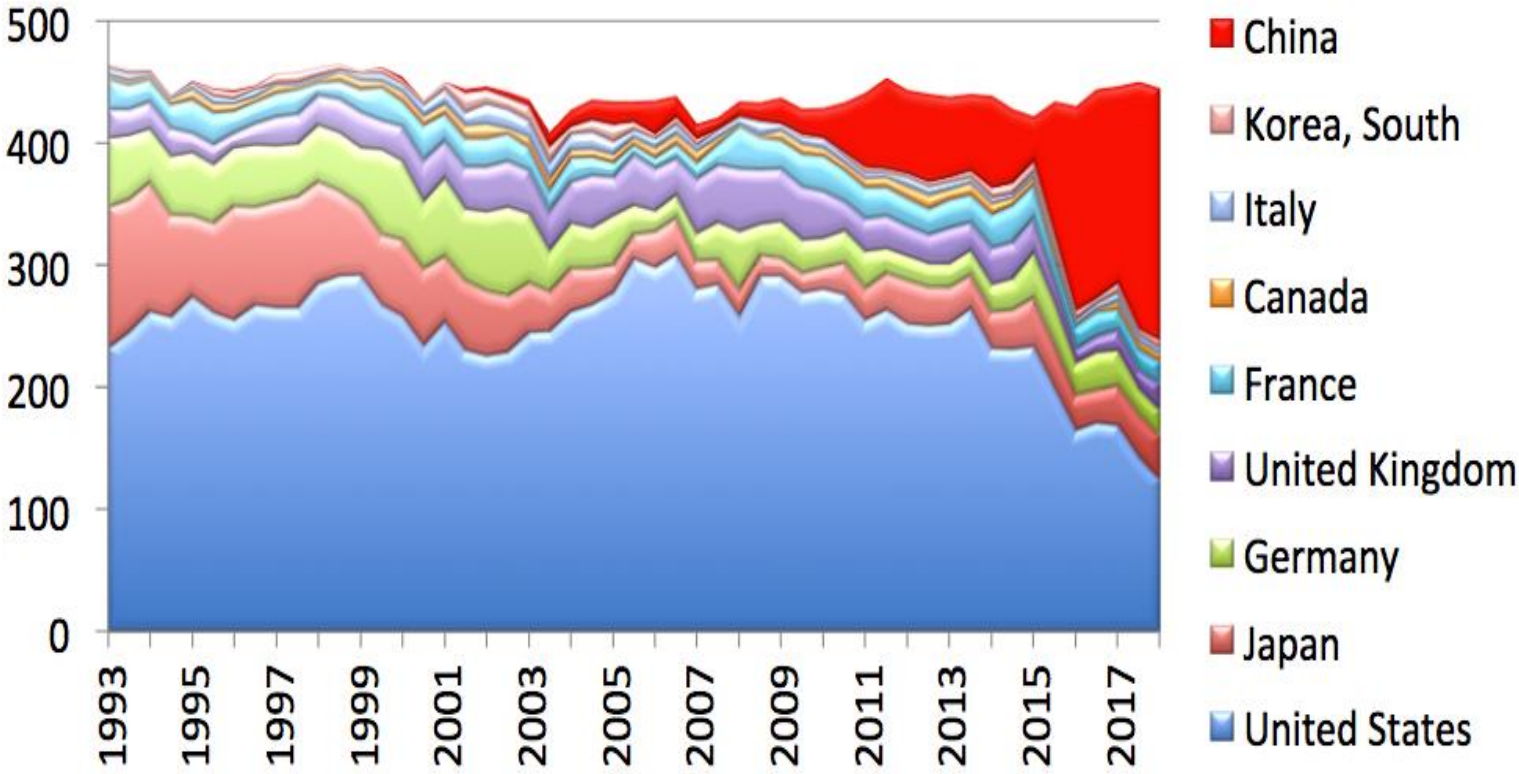
Industrial and Commercial

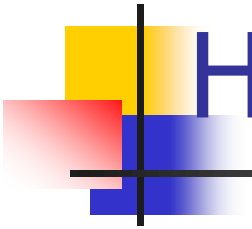


Top500 Applications



Top500 Countries



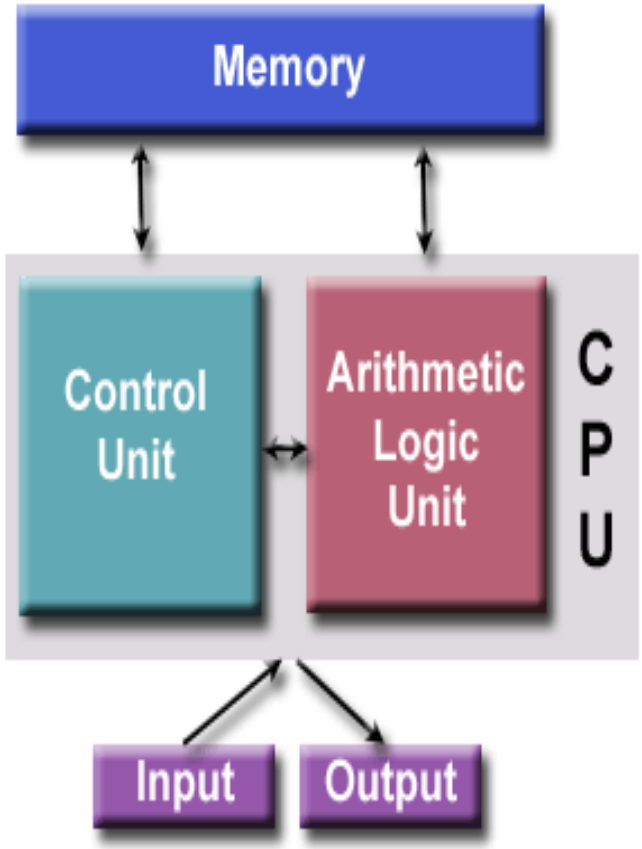


Homework:

- To investigate what is the fastest computer in the world and mention its technical specifications. Name, place, institution, number of cores, power consumption, etc.
- To investigate the technical specifications of your computers and mobiles (memory, type of memory, number of processors, number of threads).
- Take a look into the Grand Challenge problems and list three of them related to computing.

Von Neumann Architecture:

- Memory
- Control Unit
- Arithmetic Logic Unit
- Input/Output

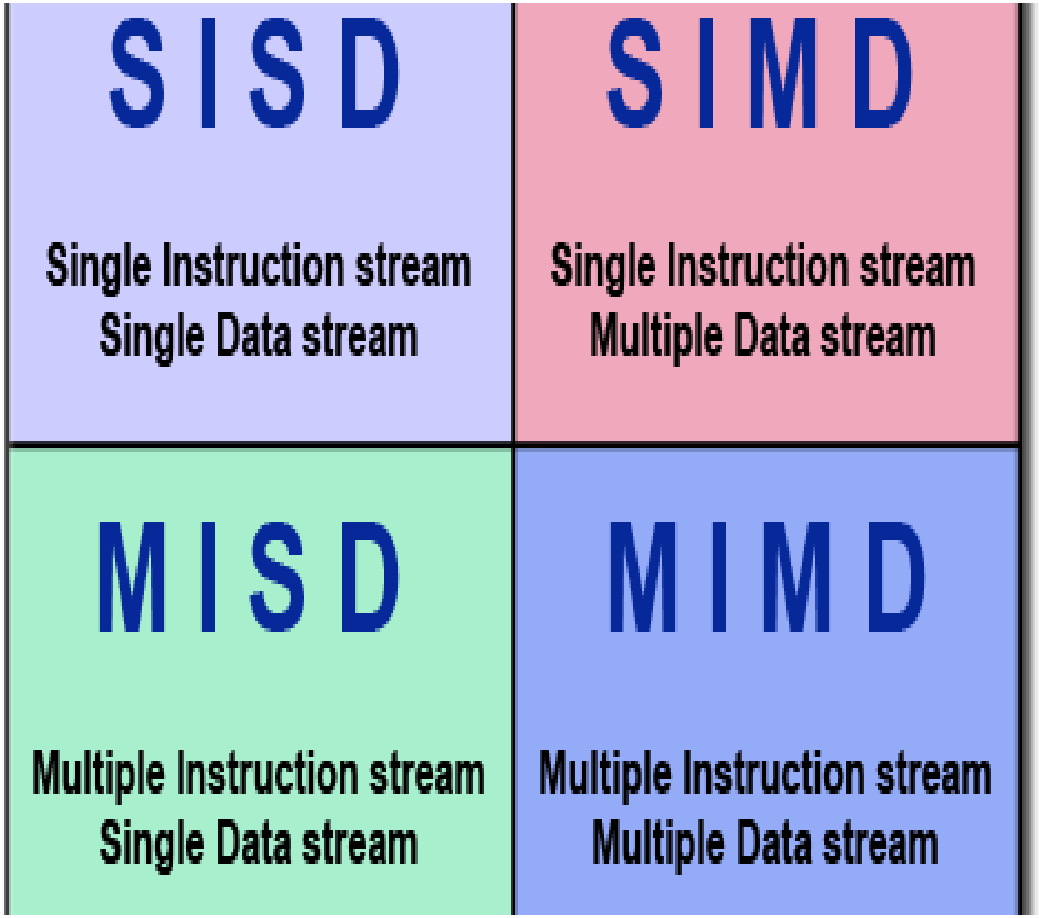


Why do we care?

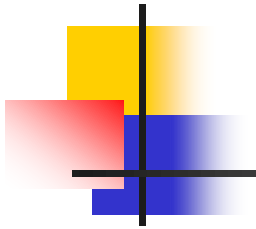
- Parallel computers still use this arch.
- Just multiplied in units.

Flynn´s Classical Taxonomy

- Two categories: **Instruction Stream** and **Data Stream**.
- These two dimensions with 2 possibilities: **Single** or **Multiple**.

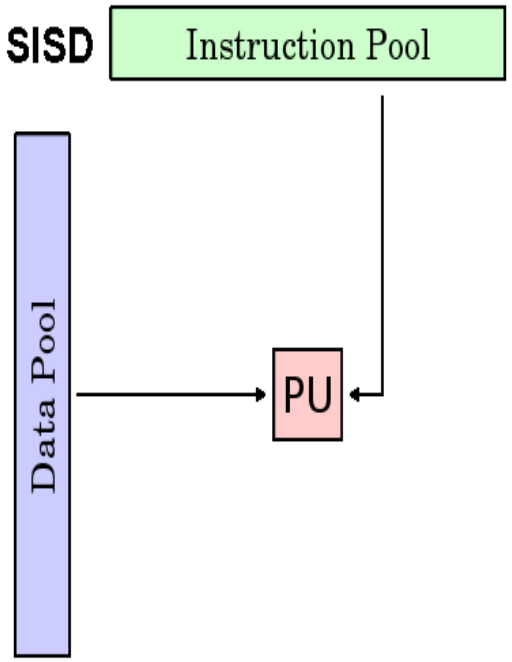


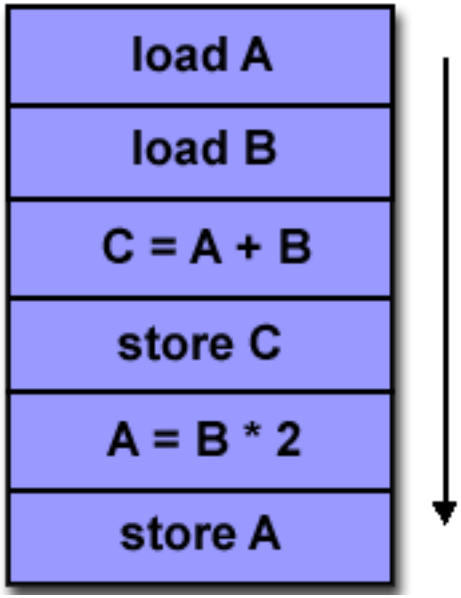
Single Instruction, Single Data (SISD)



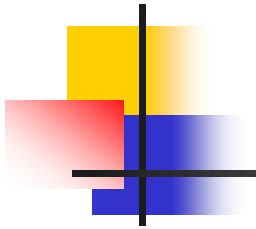
- A serial computer.
- **Single instruction:** Only one instruction is being used as input during any one clock cycle.
- **Single Data:** Only one data stream is being used as input during any one clock cycle.
- Deterministic execution
- This is the oldest type of computer.
- Mainframes, minicomputers, workstations, single processor PC.

Single Instruction, Single Data (SISD)



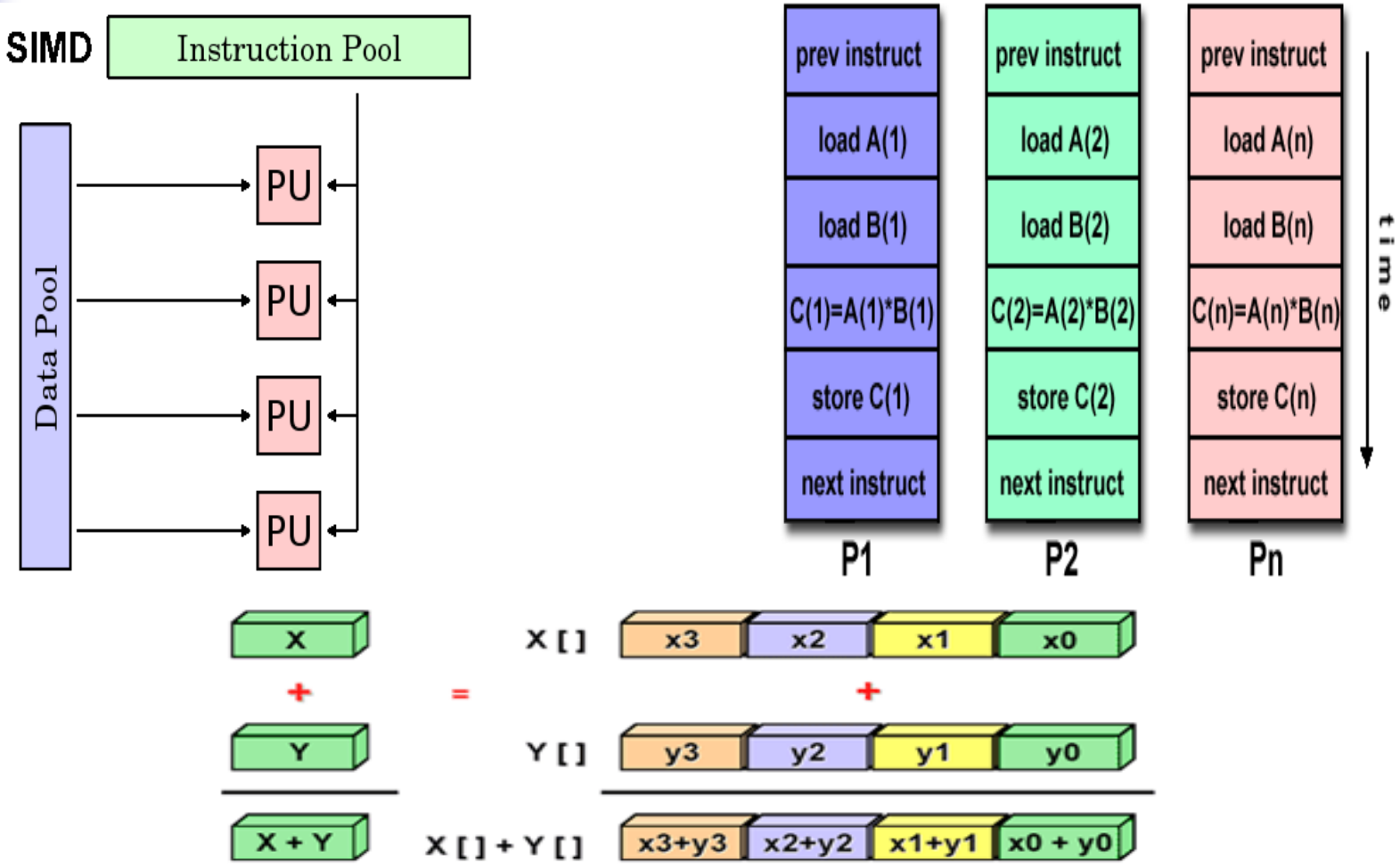


Single Instruction, Multiple Data (SIMD)

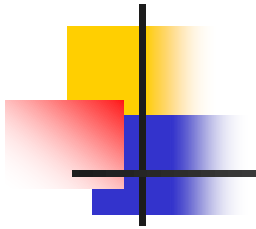


- A type of parallel computer
- Single Instruction: All processing units execute the same instruction at any given clock cycle.
- Multiple Data: Each processing unit can operate on a different data element.
- Best suited for specialized problems characterized by a high degree of regularity, graphics/image processing.
- Examples: **Processor Arrays:** Thinking Machines, ILLIAC IV. **Vector Pipelines:** IBM 9000, Cray, Fujitsu, Hitachi. **GPUs.**

Single Instruction Multiple Data (SIMD)

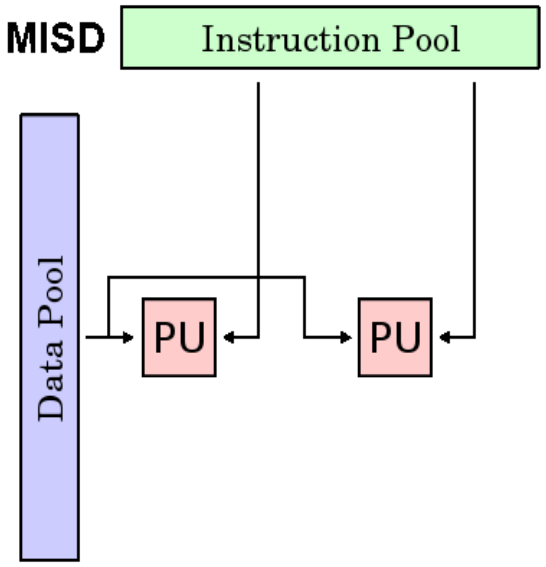


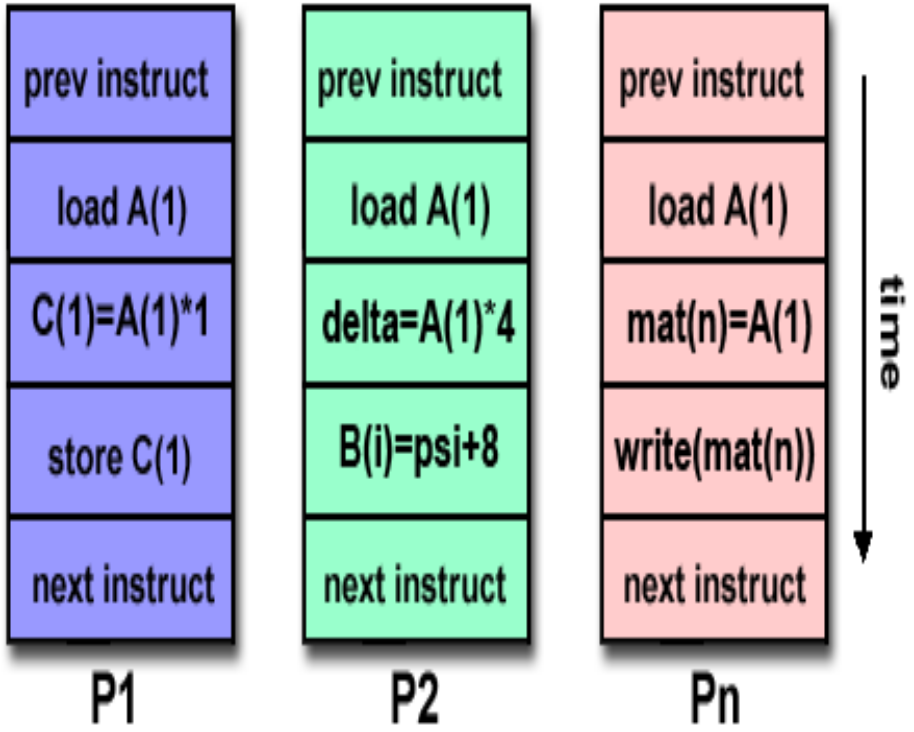
Multiple Instruction, Single Data (SISD)



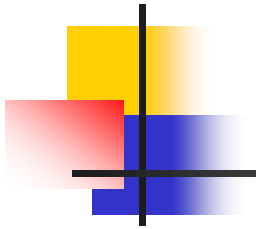
- A type of parallel computer
- **Multiple Instruction:** Each processing unit operates on the data independently via separate instruction streams.
- **Single Data:** A single data stream is fed into multiple processing units.
- Few (if any) actual examples of this class of parallel computer have ever existed.

Multiple Instruction Multiple Data (MISD)



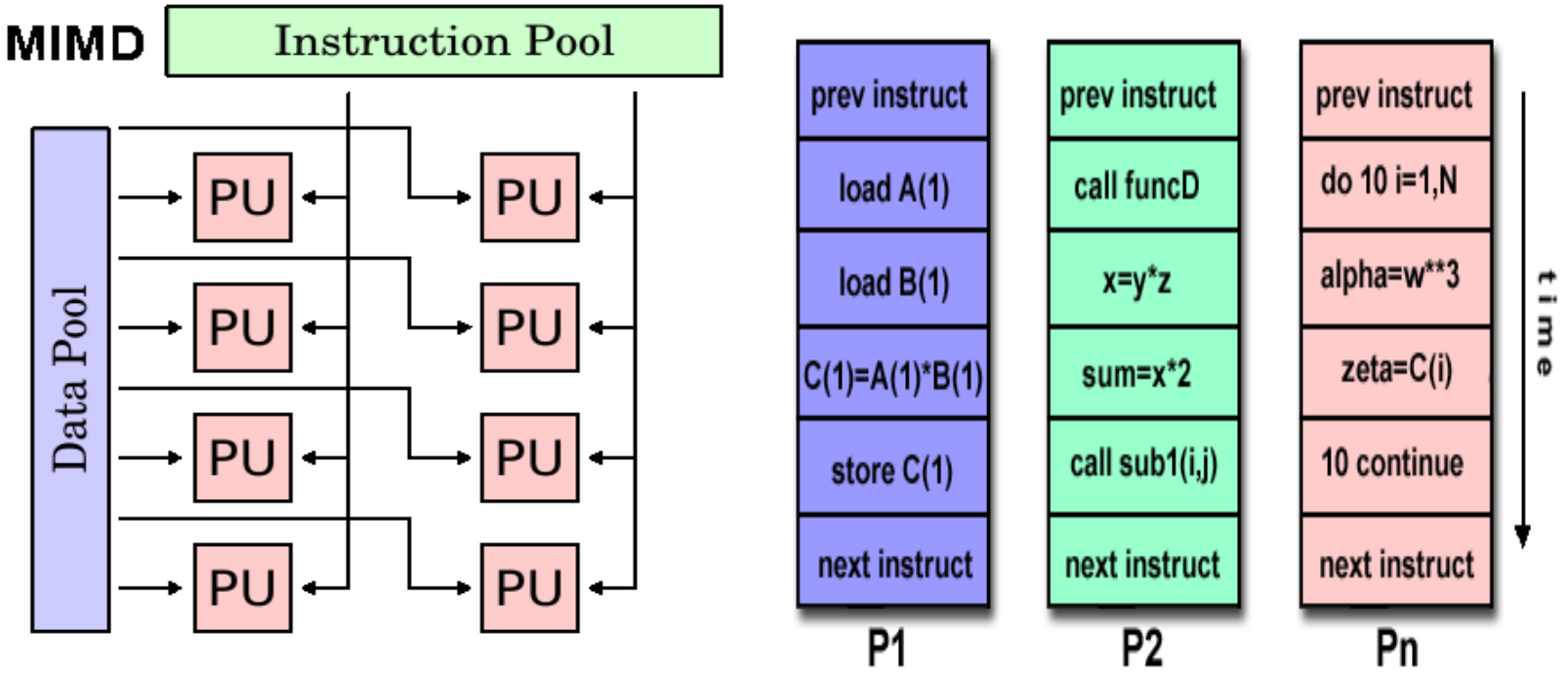


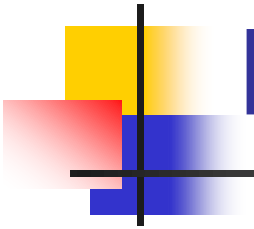
Multiple Instruction, Multiple Data (MIMD)



- A type of parallel computer
- **Multiple Instruction:** Every processor may be executing a different instruction stream.
- **Multiple Data:** Every processor may be working with a different data stream.
- Execution can be synchronous or asynchronous, deterministic or non-deterministic.
- Currently, the most common type of parallel computer - **most modern supercomputers** fall into this category.

Multiple Instruction Multiple Data (MIMD)





Projects

- Mandelbrot
- Tree search
- Random numbers generators.
- Floy’s algorithm
- Quicksort
- Dijkstra’s algorithm
- Matrix transposition