Lecture 1

Introduction to Parallel and Distributed Computing

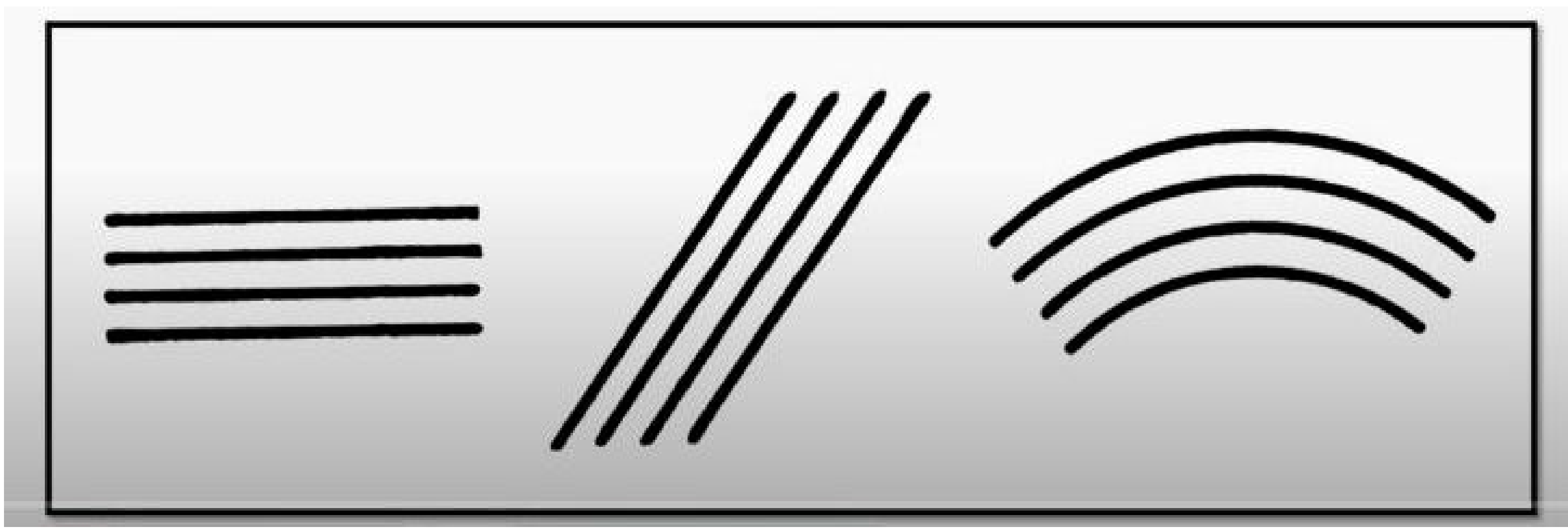
Institute of Computer Science & Information Technology,
Faculty of Management & Computer Sciences,
The University of Agriculture, Peshawar, Pakistan.

Computing

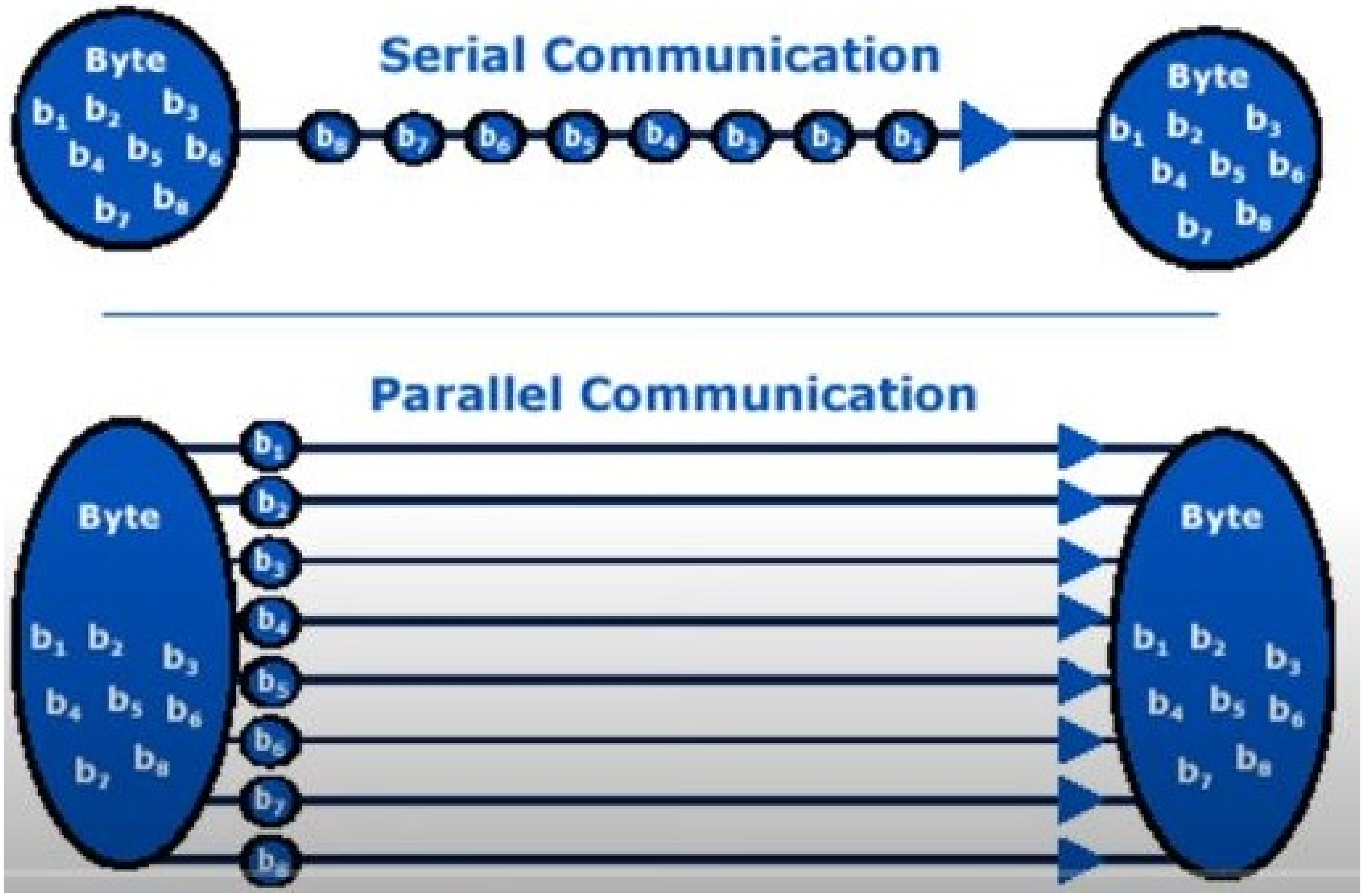
- Computing is the process to complete a given goal-oriented task by using computer technology
- Computing may include the design and development of software and hardware systems for a broad range of applications, often consists of structuring, processing and managing any kind of information

Parallel

- Parallel (in mathematics) means two lines that never intersect.



Serial vs. Parallel

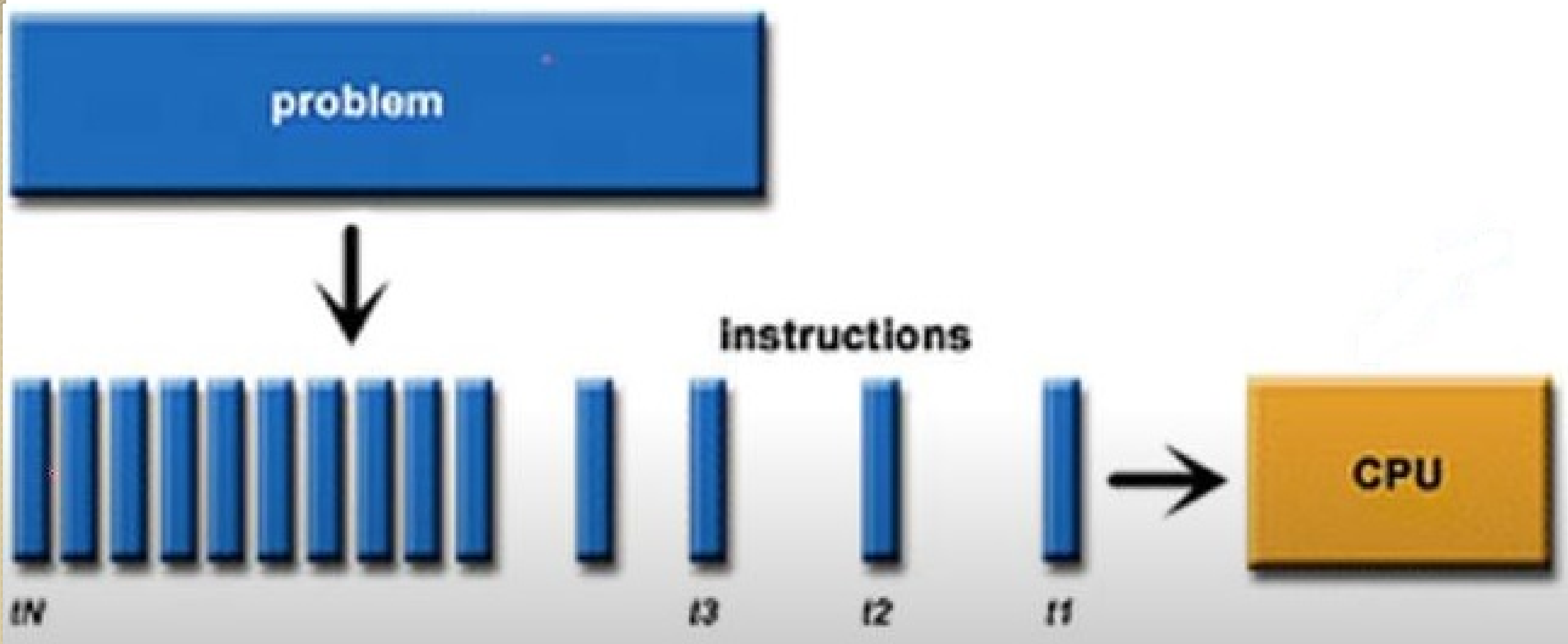


Parallel Computing Systems

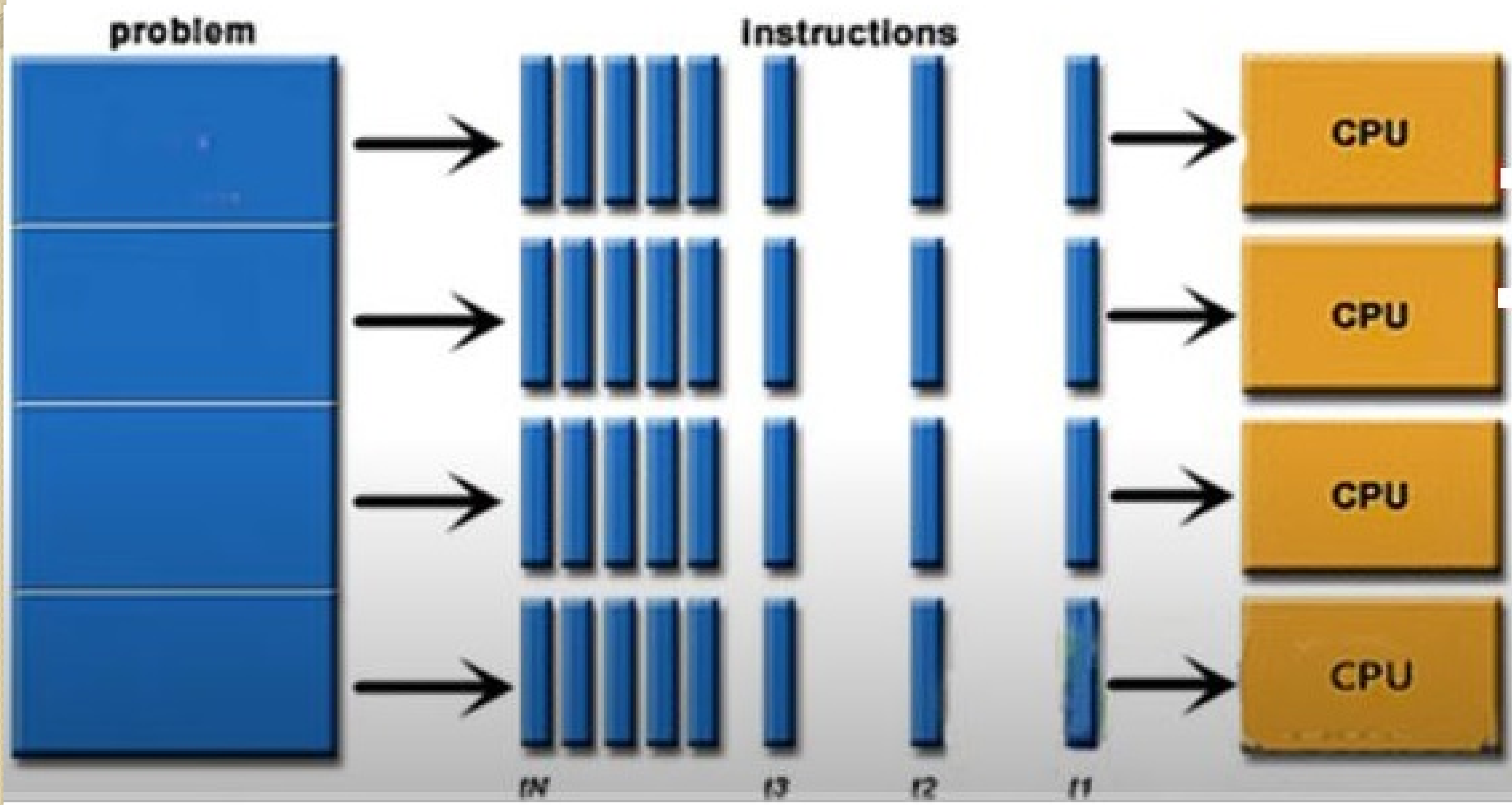
- Parallel Computing Systems are the simultaneous execution of the single task (split up and adapted) on multiple processors in order to obtain results faster.

- The idea is based on the fact that the process of solving a problem usually can be divided into smaller tasks (divide and conquer), which may be carried out simultaneously with some coordination.

Parallel Computing Systems (1)



Parallel Computing System (2)

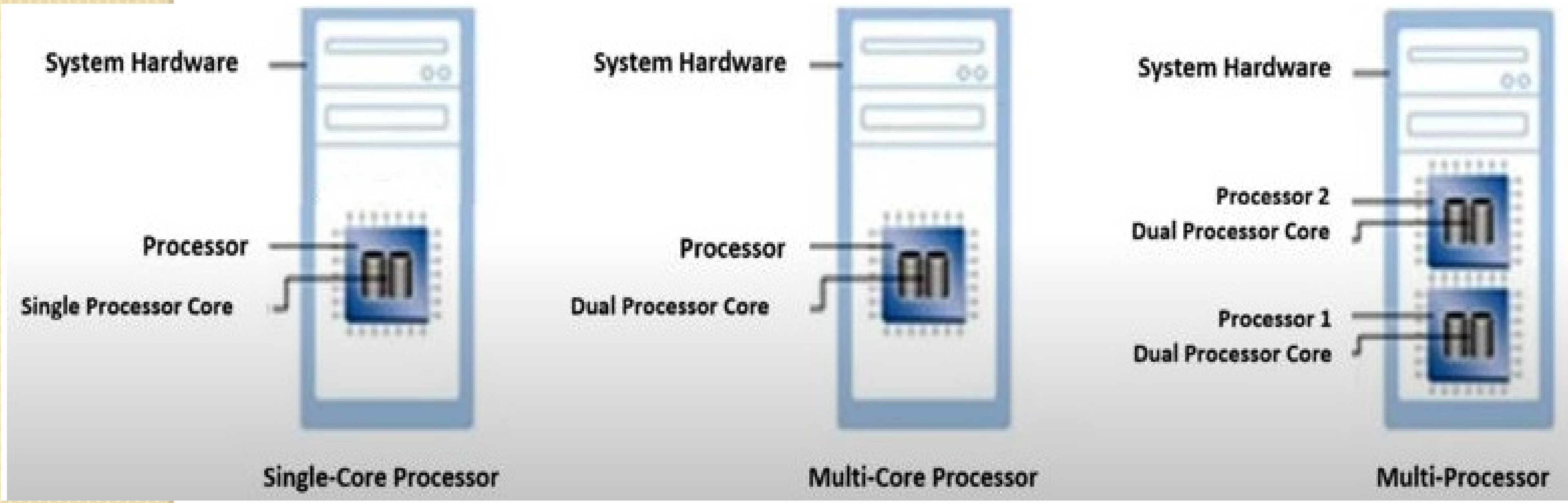


Parallel Computing (3)

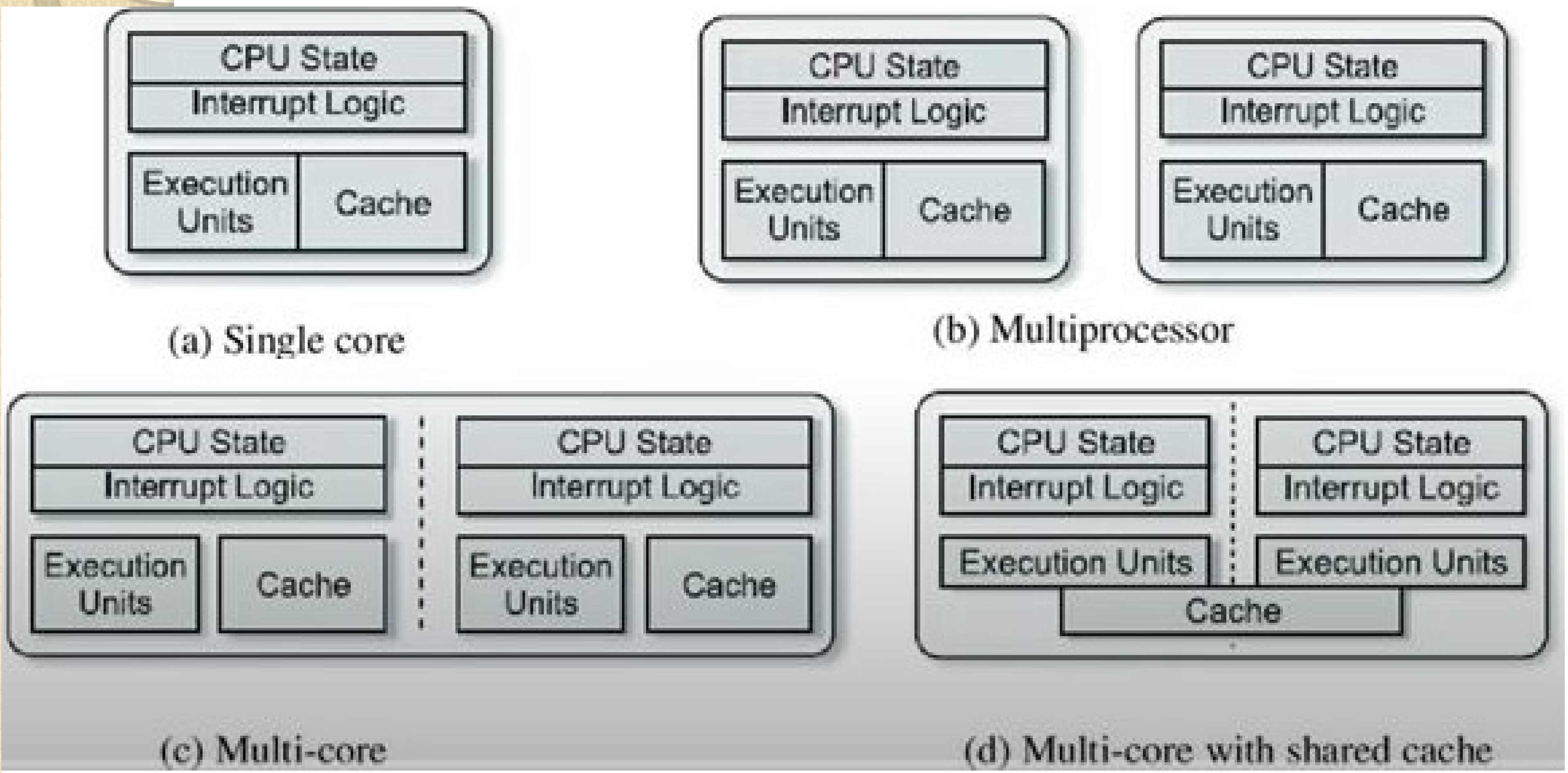
- The terms parallel computing architecture sometime used for a computer with more than one processor (few to thousands), available for processing.

- The multi-core processors (chips with more than one processor core) are some commercial examples which bring parallel computing to the desktop.

Parallel Computing



Multi-Core & Multi-Processor

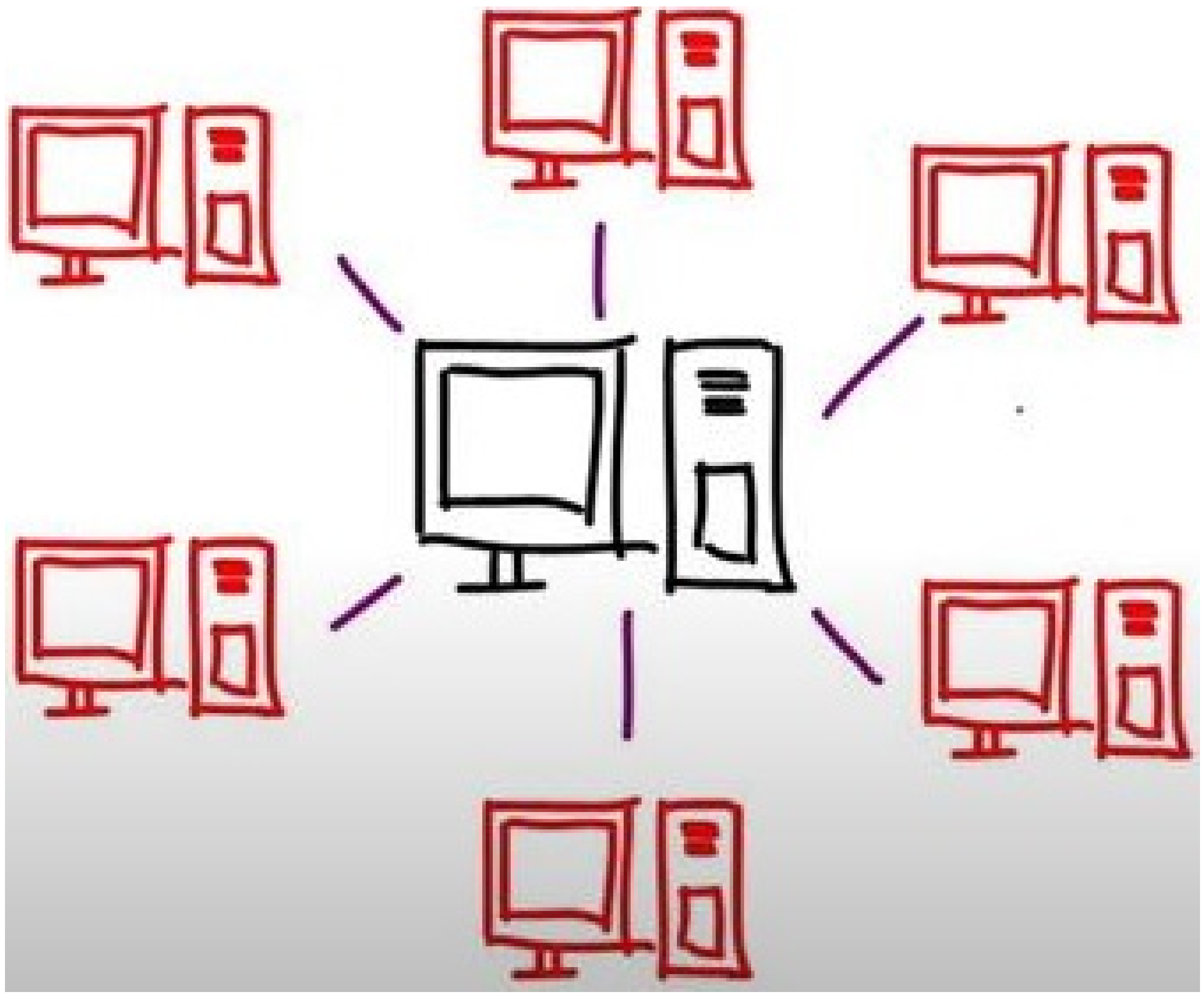


Distributed Systems

- We define a distributed system as one in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages.

- This simple definition covers the entire range of systems in which networked computers can usefully be deployed.

Distributed Systems



Why use Parallel Computing?

- 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 complex real world phenomena.

Why use Parallel Computing? (1)

Save Time & 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.

Why use Parallel Computing? (2)

Solve Large / More Complex Problems

- Many problems are so large and complex that it is impractical or impossible to solve them using a serial program, especially given limited computer memory.

- E.g., Web search engines/ databases processing millions of transaction every second.

Why use Parallel Computing? (3)

Concurrency

- A single compute resource can only do one thing at a time.
- Multiple compute resources can do many things simultaneously.

- E.g., Collaborative networks provide a global venue where people from around the world can meet and conduct work "virtually".

Why use Parallel Computing? (4)

- Take advantage of non-local resources
- Using compute resources on wide area network, or even the internet when local compute resources are scarce or insufficient.

Why use Parallel Computing? (5)

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

Why use Distributed Computing?

- One reason is historical: Computing resources that used to operate independently now need to work together.
- E.g., consider an office that acquired personal workstations for individual use.

Why use Distributed Computing? (1)

- After a while, there were many workstations in the office building, and the users recognized that it would be desirable to share data and resources among the individual computers.
- They accomplished this by connecting the workstations over a network.

Why use Distributed Computing? (2)

- A second reason is functional: if there is special function hardware or software available over the network, then that functionality does not have to be duplicated on every computer system (or node) that needs to access the special-purpose resource.

Why use Distributed Computing? (3)

- A third reason is economical: It may be more cost-effective to have many small computer working together than one large computer of equivalent power.
- In addition, having many units connected to a network is the more flexible configuration; if more resources are needed, another unit can be added in place, rather than bringing the whole system down and replacing it with an upgraded one.

Why use Distributed Computing? (4)

- Furthermore, a distributed system can be more reliable and available than a centralized system.
- This is a result of the ability to replicate both data and functionality.
- E.g., when a given file is copied on two different machines, than even if one machine is unavailable, the file can still be accessed on the other machine.

Why use Distributed Computing? (5)

- Likewise, if several printers are attached to a network, then even if an administrator takes one printer offline for maintenance, users can still print their files using an alternate printer.

Why use Distributed Computing? (6)

- Few problem in distributed computing
- E.g., are keeping multiple copies of data consistent, and keeping the clocks on different machines in the system synchronized.
- A system that provides distributed computing support must address these new issues.

Parallel Computing vs. Distributed Computing

Sr.	Parallel Computing	Distributed Computing
1.	Many operations are performed simultaneously	System components are located at different locations
2.	Single computer is required	Uses multiple computers
3.	Multiple processors perform multiple operations	Multiple computers perform multiple operations

Parallel Computing vs Distributed Computing

Sr.	**Parallel Computing**	**Distributed Computing**
4.	It may have shared or distributed memory	It have only distributed memory
5.	Processors communicate with each other through bus	Computer communicate with each other through message passing.
6.	Improves the system performance	Improves system scalability, fault tolerance and resource sharing capabilities

Why not to use Parallel Computing

- First, we have to recall, why we used Parallel Computing

1. Solve larger/ more complex problems
2. Provide concurrency
3. Take advantage of non-local resources
4. Make better use of underlying parallel hardware

Why not to use Distributed Computing

- Now we have to understand, why we used Distributed Computing:

1. Historical: Work together from different places connected network.
2. Functional: Resource Sharing (e.g., Software / Hardware).
3. Economical: Separate collaborative working units

Speedup and Amdahl's Law

- Amdahl's law is a formula which gives the theoretical speedup in latency of the execution of a task at fixed workload that can be expected of a system whose resources are improved (scalability).

- It is named after computer scientist Gene Amdahl, and was presented at the AFIPS.

- Amdahl's law is often used in parallel computing to predict the theoretical speedup when using multiple processors.

Scalability

- It is the property of a system to handle a growing amount of work by adding resources to the system.

- A system is described as scalable if it will remain effective when there is a significant increase in the number of resources and the number of users.

Scalability (1)

- Scalability of a system can be measured along the following different dimensions:

1. Physical/Load Scalability: A system can be scalable with respect to its size, meaning that we can easily add/remove more users and resources to the system.

2. Administrative scalability: The ability for an increasing number of organizations or users to access a system.

Scalability(2)

3. Functional: The ability to enhance the system by adding new functionality without disrupting existing activities.

4. Geographic: The ability to maintain effectiveness during expansion from a local area to a larger area.

5. Generation: The ability of a system to scale by adopting new generations of components.

6. Heterogeneous: is the ability to adopt components from different vendors

Scale Out (Horizontal)

- Resources fall into two broad categories: Horizontal and Vertical

- Scaling Horizontally: (in/out) Means adding more nodes to (or removing nodes from) a system, such as adding a new computer to a distribute software applications.

- An example might involve scaling out (to increase) from one web server to three.

- Exploiting this scalability requires software for efficient resource management and maintenance.

Scale Up (Vertical)

- Scaling Vertically: (up/down) mean adding more resources to (or removing resources from) a single node, typically involving the addition of CPUs, memory or storage to a single computer.

- Larger number of elements increases management complexity, more sophisticated programming to allocate tasks among resources and handle issues such as throughput and latency across nodes.

Scalability