iNurture Education Solutions
TOMORROW'S HERE

Subject Name: HIGH PERFORMANCE COMPUTING

Module Number: 01

Module Name: Introduction to Parallel and Distributed Computing

High Performance Computing

Syllabus

Introduction to Parallel and Distributed Computing

Introduction to high performance computing, Computational Demands, Parallel Processing Terminology - Data, Temporal and Control Parallelisms ,Flynn’s Taxonomy , Processor arrays, Multiprocessors, Multicomputer- Fundamental Algorithms . Message Passing Interface: MPI Introduction , Collective Communication , Data Grouping Communication

High Performance Computing

Aim

It is a technique for processing enormous amounts of data quickly using a network of computers and storage devices. HPC makes it feasible to investigate and discover solutions to some of the biggest issues in business, engineering, and research.

Objectives

The Objectives of this module are

- Identify different machine architecture
- Learn about message passing interface

High Performance Computing

Outcome

At the end of this module, you are expected to:

- Recognize important parallel processing concepts.
- Understand multiprocessor and multicomputer.
- Understand about Message passing interface

High Performance Computing

Contents

What is Introduction to Parallel and Distributed Computing

Parallel computing and distributed computing are the two primary methods of computation. A computer system may carry out duties as directed by a human. In a computer system, a single processor can only carry out one task at a time, which is inefficient. This issue is resolved by parallel computing, which enables multiple processors to complete work simultaneously.

Parallel Computing:

It also goes by the name of parallel processing. It makes use of many processors. The tasks that have been assigned to each processor are finished by them. In other words, parallel computing entails carrying out several tasks at once. To support parallel computing, a shared memory or distributed memory system can be employed. In systems with shared memory, all CPUs can access the memory. In distributed memory systems, the processors share memory.

Introduction to Parallel and Distributed Computing

What is Introduction to Parallel and Distributed Computing

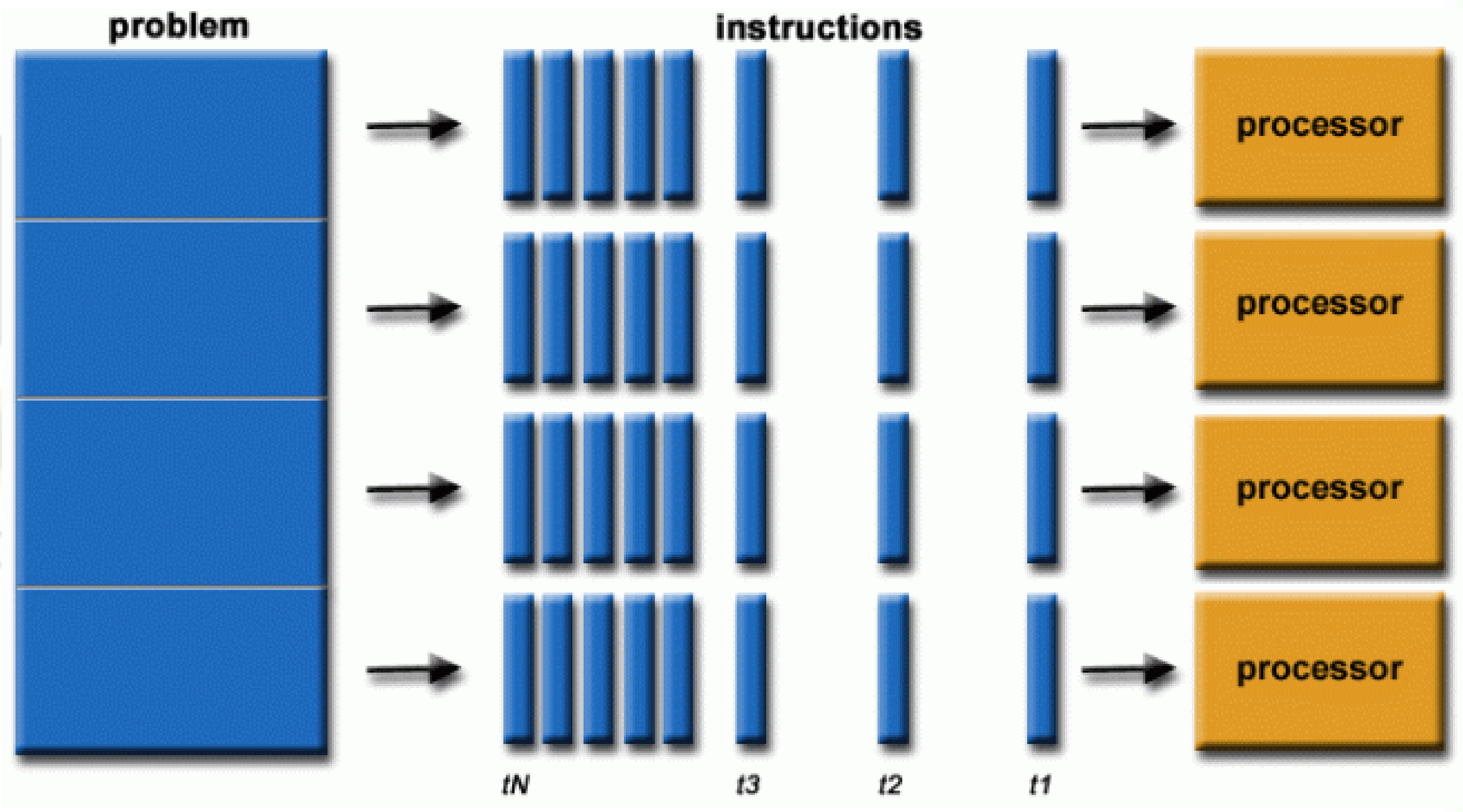
The simplest definition of parallel computing is the concurrent utilization of several computer resources to address a computational issue:

A problem is divided into distinct components that can be resolved simultaneously.

Each component is then divided into a set of instructions.

Each component's instructions run concurrently on different processors.

There is a systemic control and coordination mechanism.



What is Introduction to Parallel and Distributed Computing

The benefits and drawbacks of parallel computing

Both benefits and drawbacks of parallel computing exist. The following are a few benefits and drawbacks:

Advantages:

Because multiple resources work together to reduce time and expenses, it saves time and money. Larger issues may be challenging to address using serial computing. You can use a lot of computing resources at once to do several tasks. For modeling, simulating, and interpreting complex real-world events, parallel computing is significantly superior to serial computing.

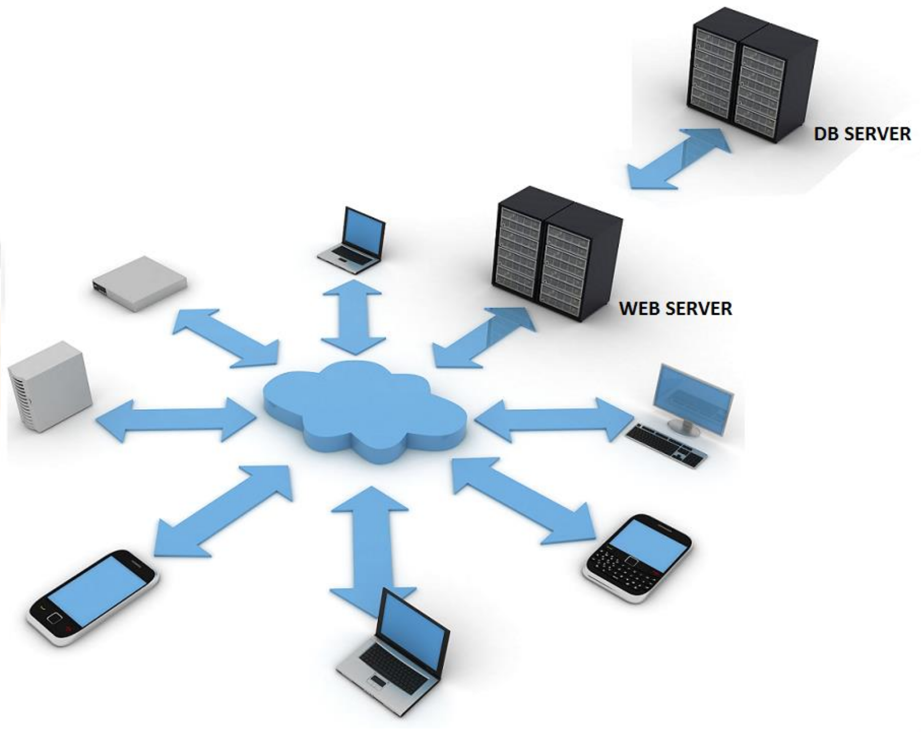
Disadvantages

Power usage for multi-core designs is high. Due to the complexity of communication and coordination, parallel solutions are more challenging to implement, debug, and prove correct, and they frequently perform worse than their serial counterparts.

What is Introduction to Parallel and Distributed Computing

Distributing Computing

It consists of a number of software parts that are installed on many computers yet work together as a single system. The computers that make up a distributed system can be geographically separated and connected by a wide area network or physically close to one another and connected by a local network (WAN). Mainframes, PCs, workstations, and minicomputers are just a few examples of the many distinct configurations that might make up a distributed system. Making a network function as a single computer is the primary goal of distributed computing.



What is Introduction to Parallel and Distributed Computing

Advantages and Disadvantages of Distributed Computing

The benefits and drawbacks of distributed computing are numerous. The following are a few benefits and drawbacks:

Advantages

Because of its adaptability, installing, utilizing, and debugging new services is a breeze.If more machines are needed, you can add them in distributed computing. Other servers are unaffected if the system crashes on one server. A distributed computer system can speed up traditional systems by combining the computational power of numerous machines.

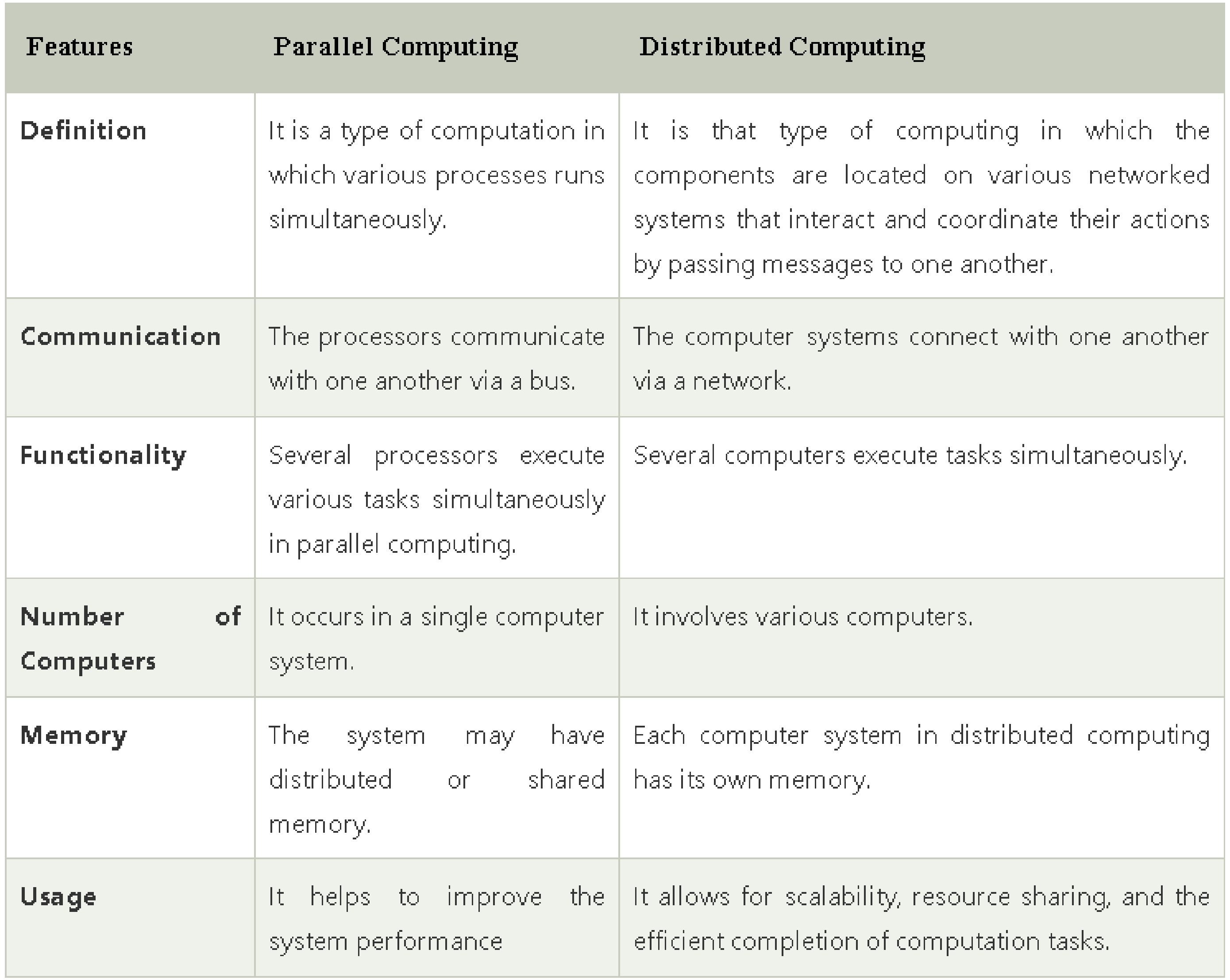
•**Disadvantages**

Due to the characteristics of open systems, data security and sharing are the key problems in distributed systems. The dispersion over numerous servers makes troubleshooting and debugging more difficult. The absence of software support is distributed computer systems' fundamental drawback.

Introduction to Parallel and Distributed Computing

What is Introduction to Parallel

Key differences between the Parallel Computing and Distributed Computing



Features	Parallel Computing	Distributed Computing
Definition	It is a type of computation in which various processes runs simultaneously.	It is that type of computing in which the components are located on various networked systems that interact and coordinate their actions by passing messages to one another.
Communication	The processors communicate with one another via a bus.	The computer systems connect with one another via a network.
Functionality	Several processors execute various tasks simultaneously in parallel computing.	Several computers execute tasks simultaneously.
Number of Computers	It occurs in a single computer system.	It involves various computers.
Memory	The system may have distributed or shared memory.	Each computer system in distributed computing has its own memory.
Usage	It helps to improve the system performance	It allows for scalability, resource sharing, and the efficient completion of computation tasks.

Introduction to Parallel and Distributed Computing

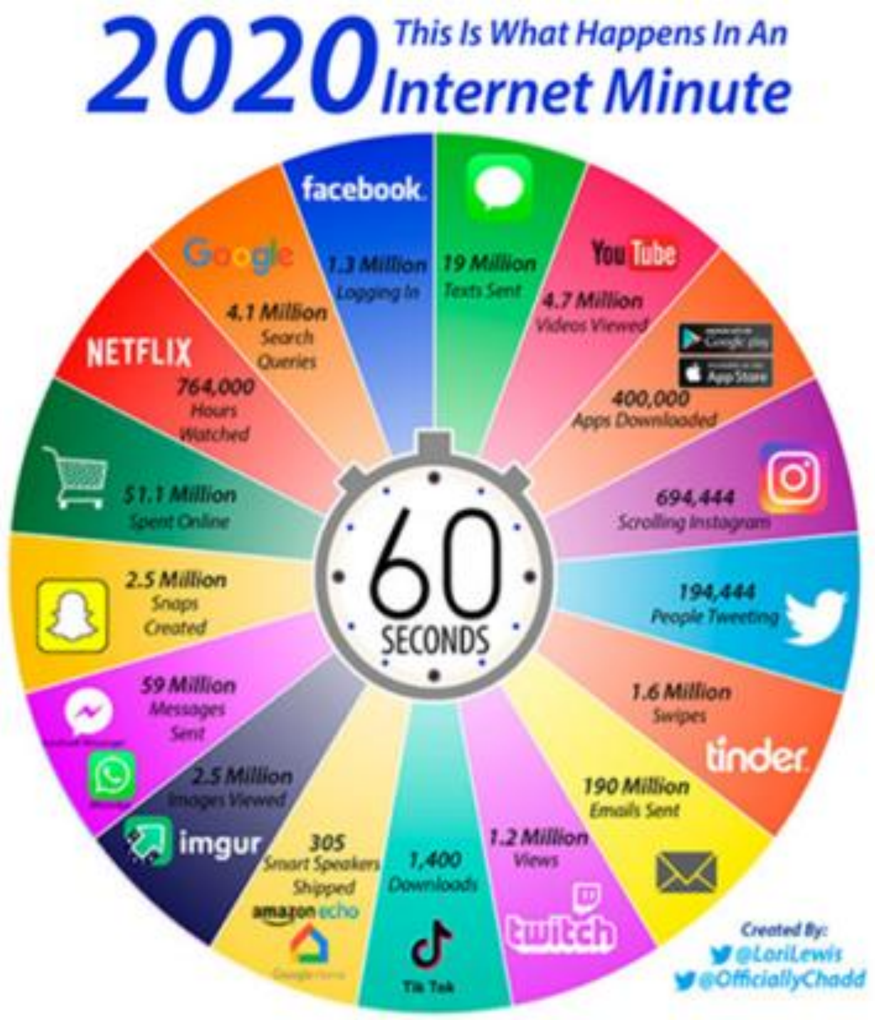
Introduction to high performance computing

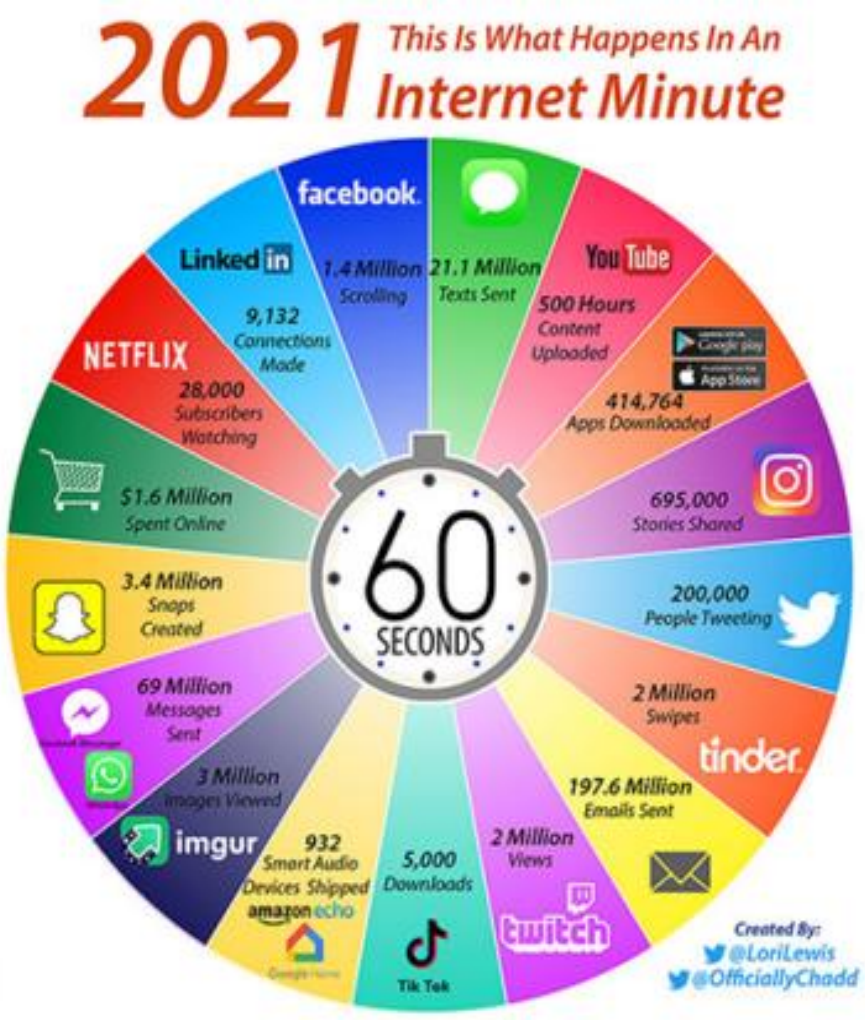
High-speed data processing and intricate calculations are capabilities of high-performance computing (HPC). High performance computing (HPC) enables the quick resolution of challenging computing issues. High-performance computers (shared-memory mainframes, computing clusters) comprised of numerous parallel processors are used for computation.It is used for many different things, from big data analysis and machine learning to solving engineering challenges in the MATLAB environment.The supercomputer is one of the most well-known forms of HPC solutions. Thousands of compute nodes are found in a supercomputer, and they collaborate to finish one or more tasks. We refer to this as parallel processing. It is comparable to having thousands of PCs networked together to combine computational power and speed up task completion.

Introduction to Parallel and Distributed Computing

Introduction to high performance computing

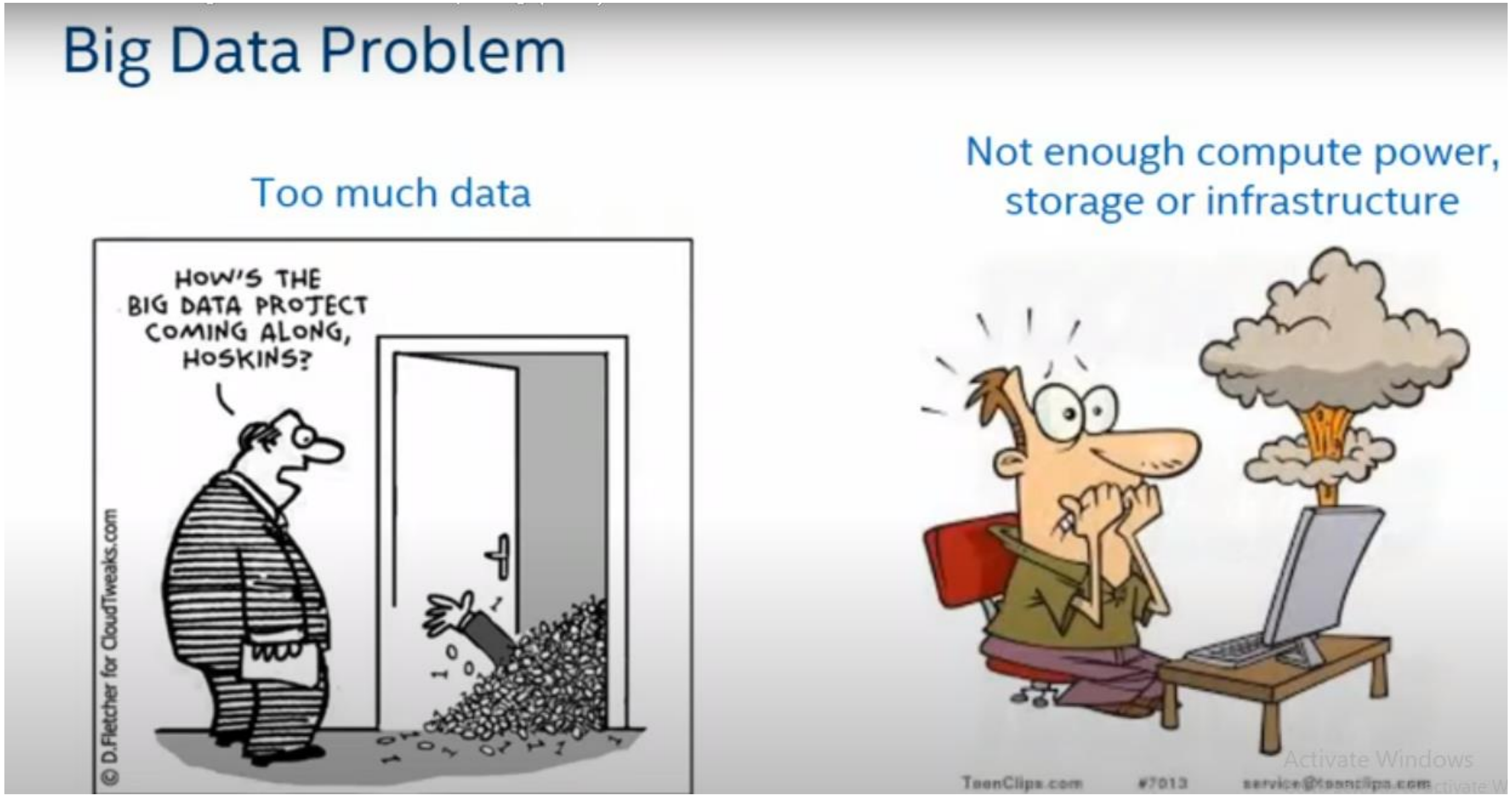
Why HPC:





Introduction to Parallel and Distributed Computing

Introduction to high performance computing



Big Data Problem

Too much data

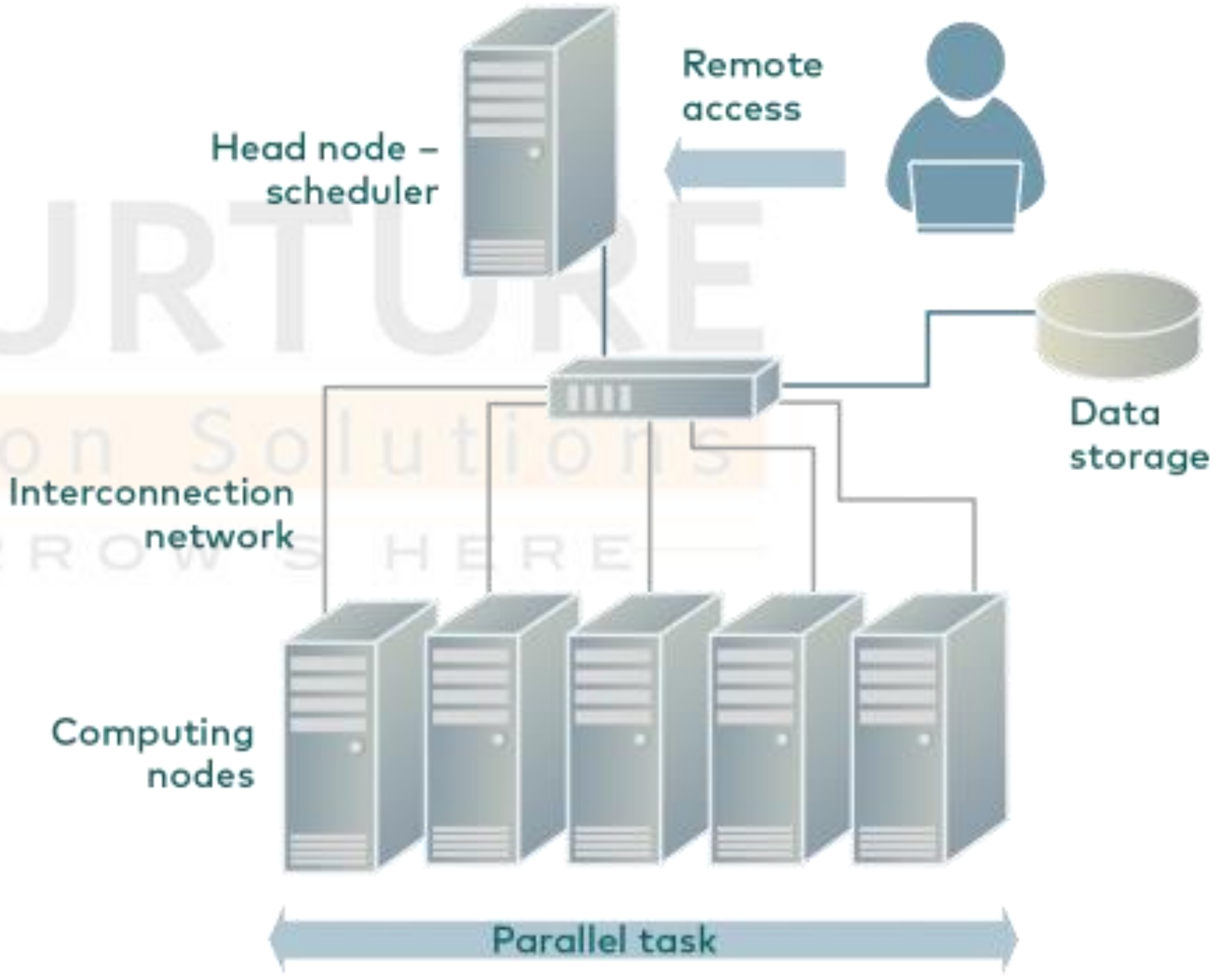
Not enough compute power, storage or infrastructure

HOW'S THE BIG DATA PROJECT COMING ALONG, HOSKINS?
© D.Fletcher for CloudTweaks.com
ToonClips.com
#7013
service@toonclips.com

Introduction to Parallel and Distributed Computing

Introduction to high performance computing

High performance computing (HPC) enables the quick resolution of challenging computing issues. High-performance computers (shared-memory mainframes, computing clusters) comprised of numerous parallel processors are used for computation. A modern HPC is a crucial component of any modern institution and is utilized for a variety of tasks, from big data analysis and machine learning to addressing engineering problems in the MATLAB environment.



Introduction to Parallel and Distributed Computing

Introduction to high performance computing

How does HPC work

The main method used by a normal computing system to handle issues is serial computing, which involves breaking the workload up into a series of tasks and executing them one at a time on the same processor.

Introduction to high performance computing

Computer clusters (also called HPC clusters): An HPC cluster is made up of several networked high-speed computers, and the workload for parallel computing is managed by a central scheduler. High-performance multi-core CPUs or, more often today, GPUs (graphics processing units) are used by the nodes, which are computers. These processors are well suited for complex mathematical calculations, machine learning models, and graphics-intensive applications. One HPC cluster may contain 100,000 nodes or more.

High-performance components: Networking, memory, storage, and file systems are all high-speed, high-throughput, and low-latency computing resources in an HPC cluster that can keep up with the nodes and maximise the computing power and performance of the cluster.

Introduction to high performance computing

WHY SHOULD YOU USE HPC?

High-performance processors and graphical accelerators are included in the HPC cluster.

•Simulation can be expedited by dividing the workload among numerous processors, leading to a more thorough investigation. By moving tasks to HPC, one can use the computer for other tasks.

•There are less disruptions to the simulation because the HPC cluster is housed in a data centre with uninterruptible power supply, cooling, and administrator supervision.

•On the HPC cluster, some scientific software is already installed.

•Many programmers come with an HPC support feature that hides the user from the HPC cluster environment.

Introduction to Parallel and Distributed Computing

Introduction to high performance computing

Computational Demands

Nowadays, the majority of HPC systems are built with these workloads in mind. AI applications in general and machine learning and deep learning applications in particular have become synonymous with HPC applications. Continuous innovation is being fueled by these HPC applications in:

Healthcare, genomics and life sciences. Today's HPC systems can sequence a human genome in less than a day, compared to the first attempt, which took 13 years. Rapid cancer diagnosis, molecular modelling, and drug discovery and design are other HPC applications in healthcare and life sciences.

Financial services. HPC is used to power applications in Monte Carlo simulation and other risk analysis techniques, in addition to automated trading and fraud detection (mentioned above).

Introduction to high performance computing

Computational Demands

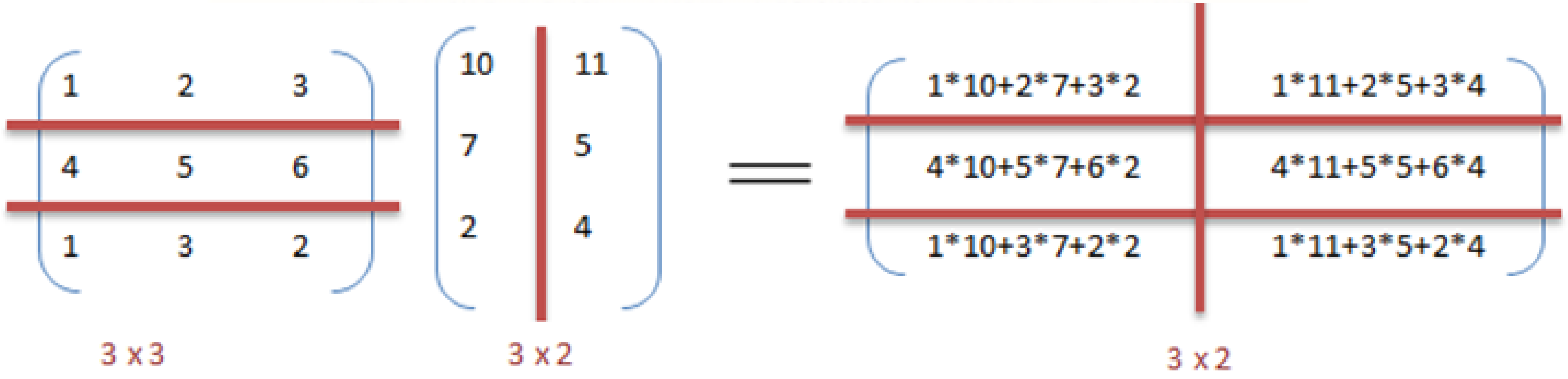
Government and defense Weather forecasting and climate modelling, which both involve processing enormous volumes of historical meteorological data and millions of daily changes in climate-related data points, are two expanding HPC use cases in this field. Research on energy and intelligence work are two other government and defence uses.

Energy. Energy-related HPC applications include seismic data processing, reservoir simulation and modelling, geospatial analytics, wind simulation, and terrain mapping, some of which intersect with government and defence.

Introduction to high performance computing

Data Parallelism

Concurrent execution of the same task on several processing cores is known as data parallelism. Take the sum of the elements in an array of size N as an example. One thread would simply add the elements [0] through [N 1] for a single-core system. In contrast, thread B, running on core 1, could total the elements [N/2]... [N 1] for a dual-core system, while thread A, running on core 0, could sum the elements [0]... [N/2 1]. Therefore, the two threads would be running concurrently on different processing cc



Introduction to Parallel and Distributed Computing

Introduction to high performance computing

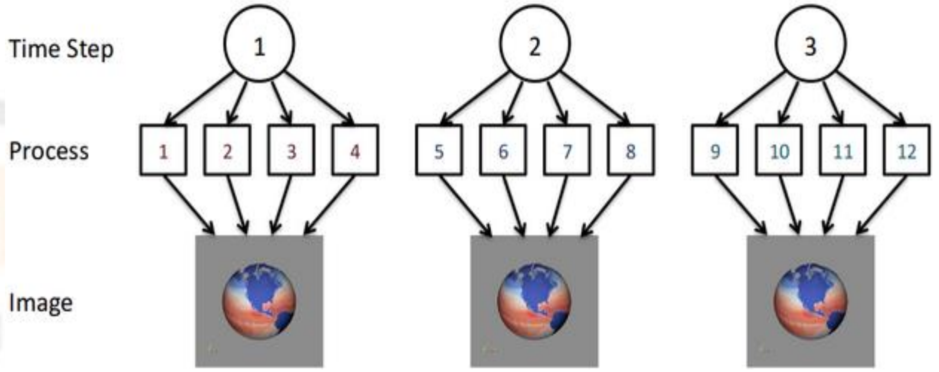
Temporal Parallelism

By dividing the processing task into a number of steps that are then progressively applied to each unit of information, temporal parallelism can achieve the same outcomes as the original task. In other words, the task is divided into time-based steps, with each step being applied to a different informational unit. Manufacturing on a "assembly line" is a classic example. Pipelined structures are created when computing uses temporal parallelism. In fact, it takes significantly longer to create a single result with a pipeline than it would without one, but the rate at which results are produced increases according to how many steps are added to the original operation. Many applications place a high value on total processing time, and in these sectors parallelism can be achieved by breaking the problem down into its component time-domain components. where each sub-task of the main system function is assigned to a processing unit, and the primary system function can be thought of as the basic unit of work

Introduction to Parallel and Distributed Computing

Introduction to high performance computing

The hardware description language (HDL) has a drawback represented in low runtime performance, and to harness the temporal parallelism in this side, the entire simulation can be divided into slices and each of these slices can be executed by independent simulator. Figure shows that the temporal parallelism can also reduce communication and even synchronization in parallel simulation environments using the HDL. On the other hand, since each process will load a whole functionality unit, temporal parallelism may require a larger quantity of memory.



Flynn’s Taxonomy

Flynn’s Classification

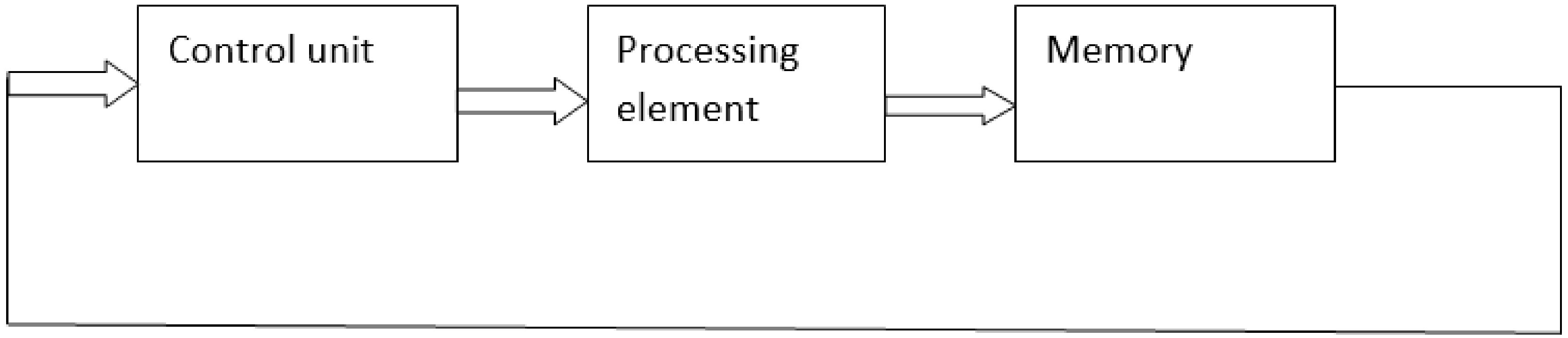
- was put forth in 1966 by researcher Michael J. Flynn.
- It is the taxonomy of computer organization that is most often used.
- Computers are categorized in this categorization based on how many instructions they process simultaneously and whether they work with a single data set or several.

Flynn’s Taxonomy

1) SISD (Single Instruction Single Data Stream)

Single instruction: During each clock cycle, the CPU acts or executes only one instruction stream.

Single data stream: One clock cycle only uses one data stream as input.

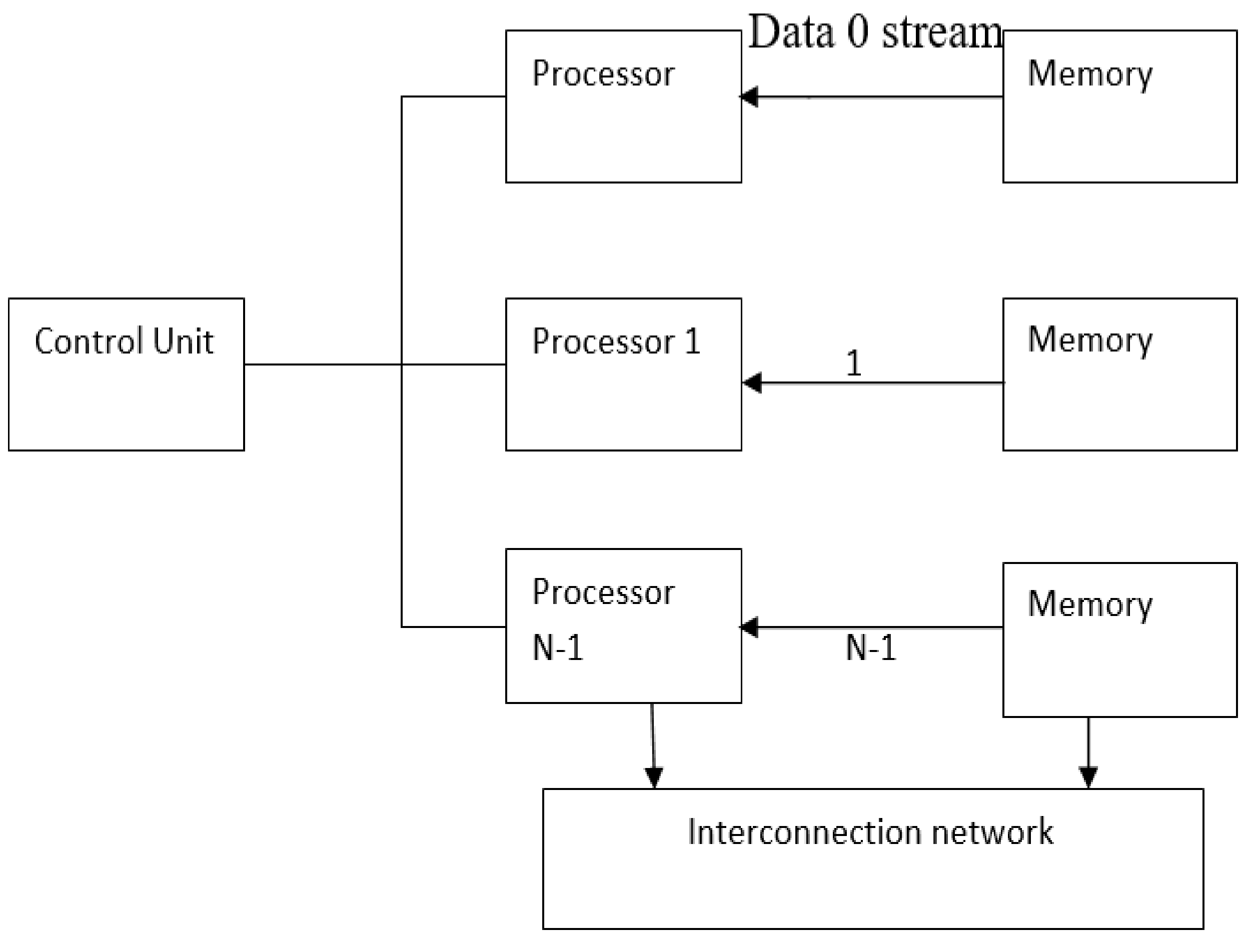


A SISD computing system is a uniprocessor computer that can carry out just one instruction while processing just one data stream. In the SISD architecture found in the majority of conventional computers, all of the information that needs to be processed must first be stored in primary memory.

Flynn's Taxonomy

2) SIMD (Single Instruction Multiple Data Stream)

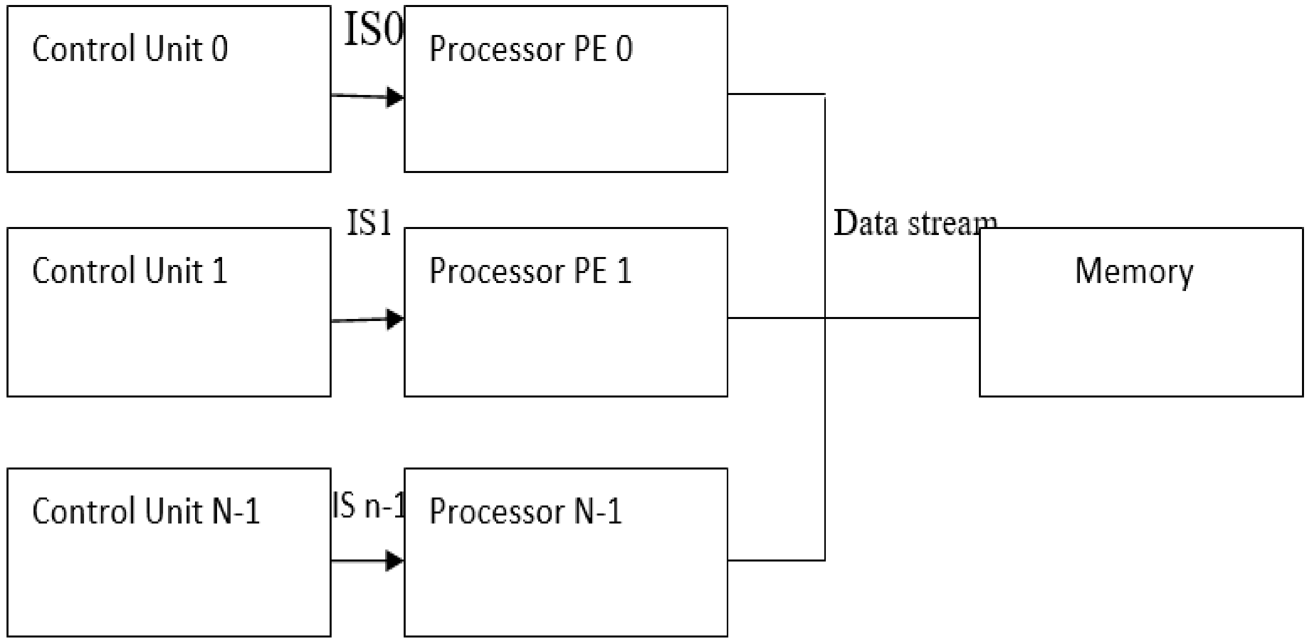
A multiprocessor machine with the ability to execute the same command on each CPU while using a distinct data stream is known as a SIMD system. The practical application of SIMD is the IBM 710..



Introduction to Parallel and Distributed Computing

Flynn’s Taxonomy

3) **MISD (Multiple Instruction Single Data stream)**
An MISD computing system is a multiprocessor device that can carry out various instructions on various processing components while still operating on the same data set. .

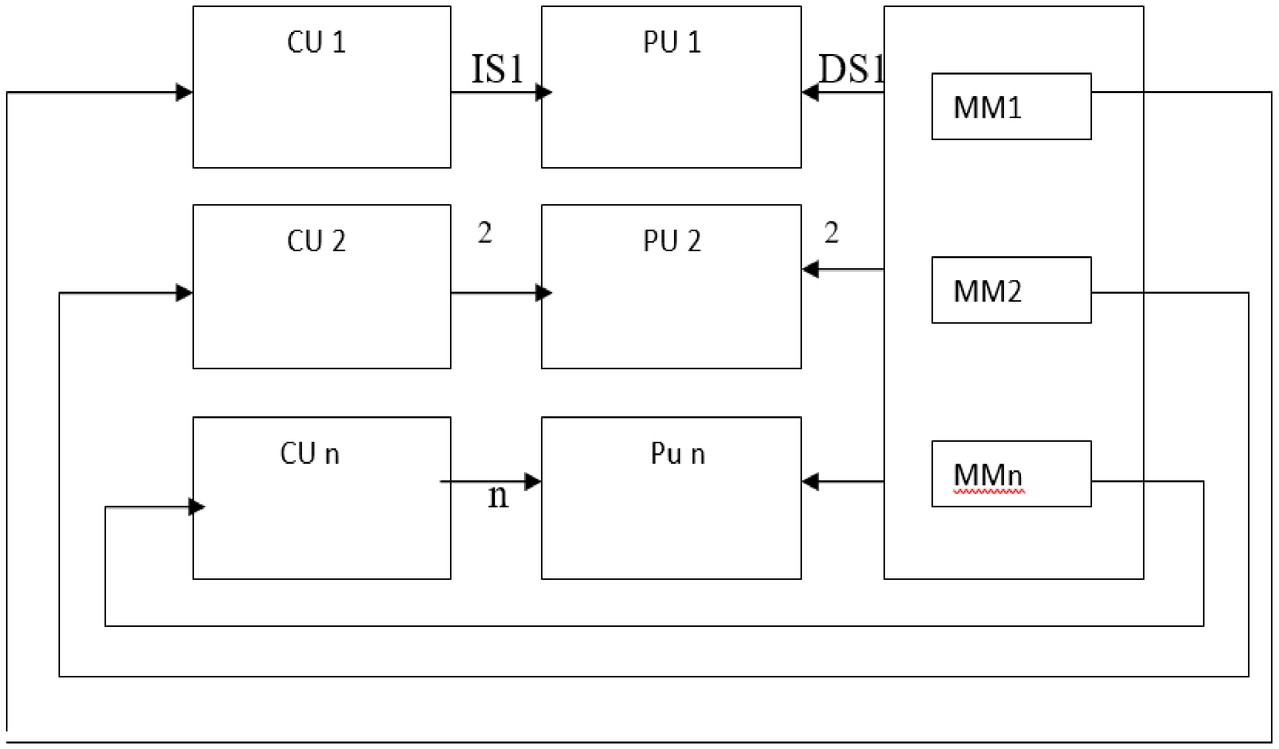


Introduction to Parallel and Distributed Computing

Flynn's Taxonomy

4) **MIMD (Multiple Instruction Multiple Data Stream)**

A multiprocessor device that can carry out numerous instructions through numerous data streams is called a MIMD system. There are distinct instruction streams and data streams for each processing element.



Introduction to Parallel and Distributed Computing

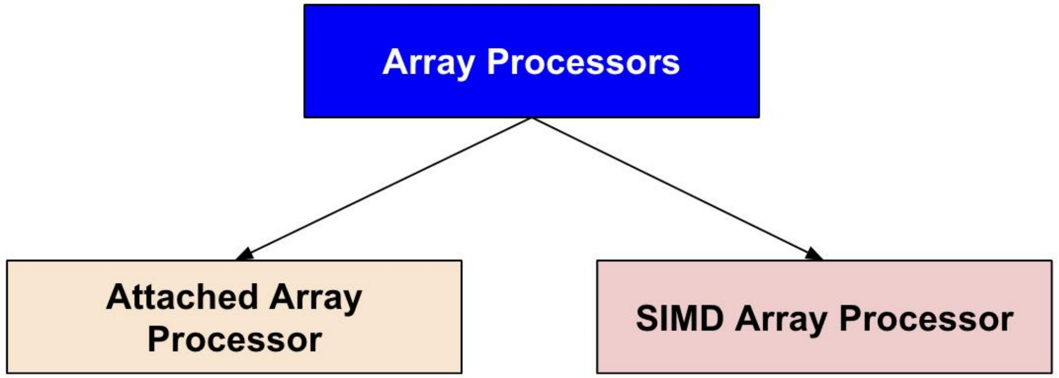
Array Processor

An array processor is a processor that runs calculations on a huge variety of data. Other names for array processors include multiprocessors and vector processors. On an array of data, only one instruction is ever executed at a time. To carry out computations, they work with enormous data sets. Consequently, they are utilised to improve the performance of computers.

Array processors can be divided into two categories:

1. Attached Array Processors
2. SIMD(Single Instruction Stream, Multiple Data Stream) Array Processors

We will go through the explained detail in the sections below.



Array Processor

Attached Array Processors

A general-purpose computer is connected to an auxiliary processor called an attached array processor in order to boost and increase the machine's performance in numerical computational tasks. Due to the extensive use of parallel processing and functional units, it offers high performance.A common processor with an input/output interface and a local memory interface is part of the associated array processor.The local memory is connected to the main memory.

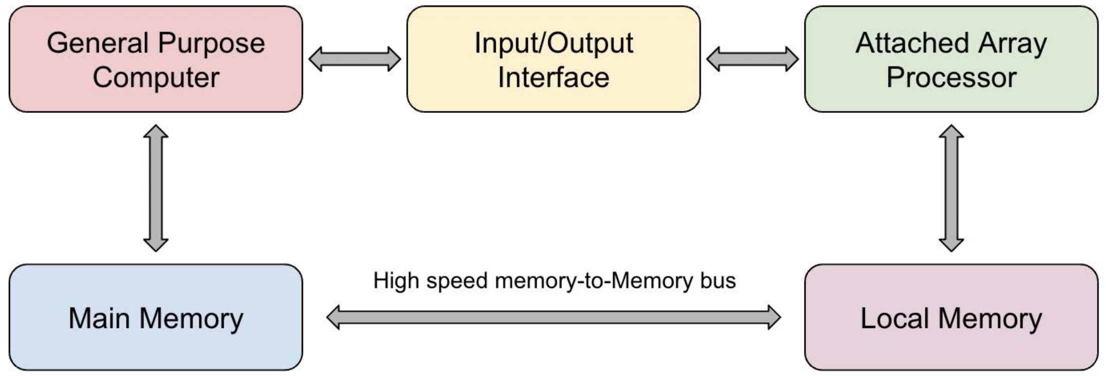


Fig- The interconnection of Attached Array Processor to Host computer

Array Processor

SIMD Array Processor

SIMD describes how a single computer is set up with numerous parallel processors. One instruction stream and numerous data streams are produced by the processing units working together under the direction of a single control unit. The general block diagram of an array processor is shown below. It is made up of several processing elements (PEs) that are all identical and have local memories (M) of their own. Each processing element has an ALU and registers. The actions of the processing elements are managed by the master control unit. Additionally, it decodes instructions to identify how they should be implemented. The primary memory houses the programme. TREhe command unit gets the directions. All PEs receive vector instructions at once, and the results are stored in memory.

Array Processor

SIMD Array Processor

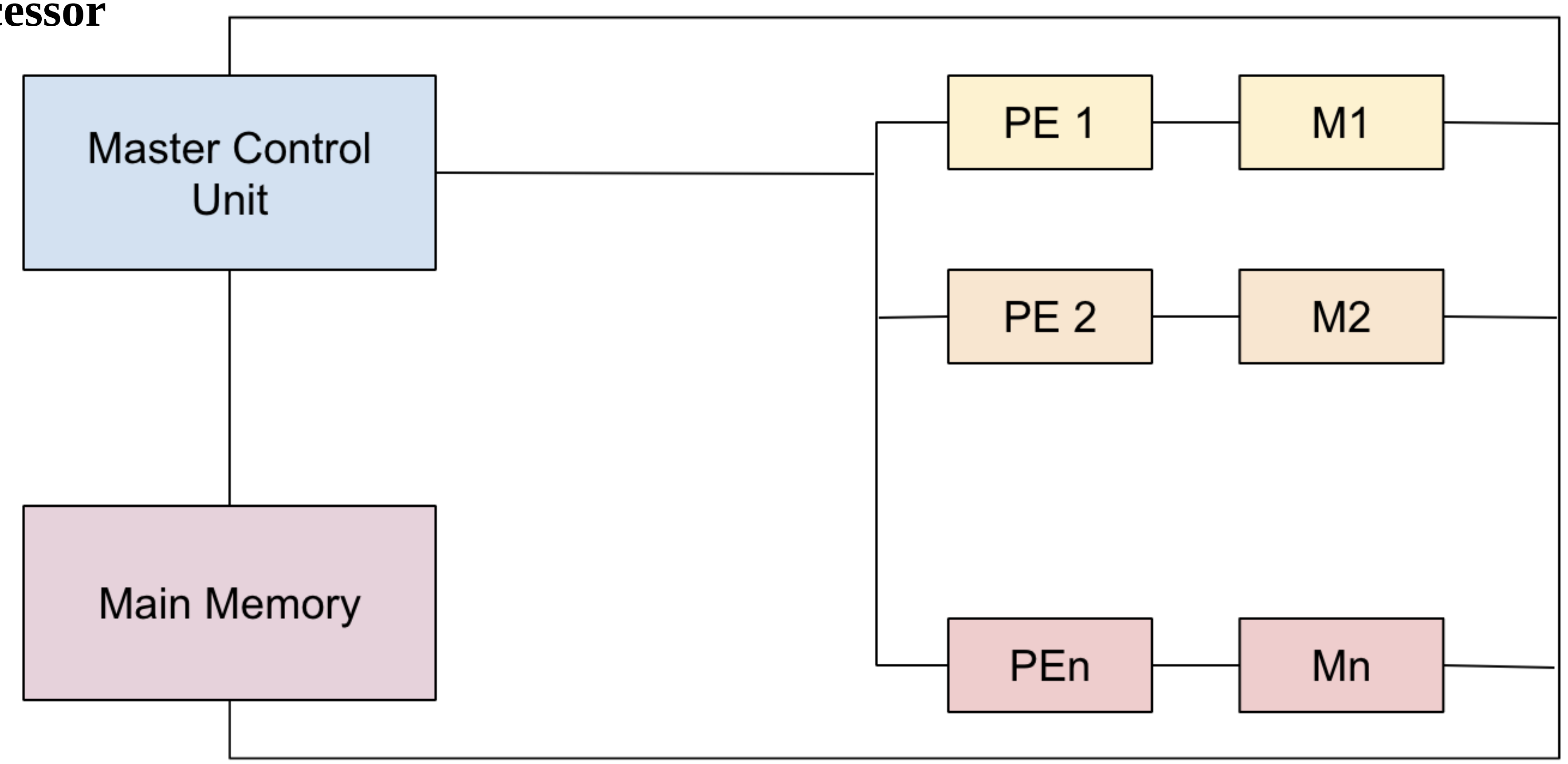


Fig-Array Processor Organisation in SIMD

Introduction to Parallel and Distributed Computing

Multiprocessor

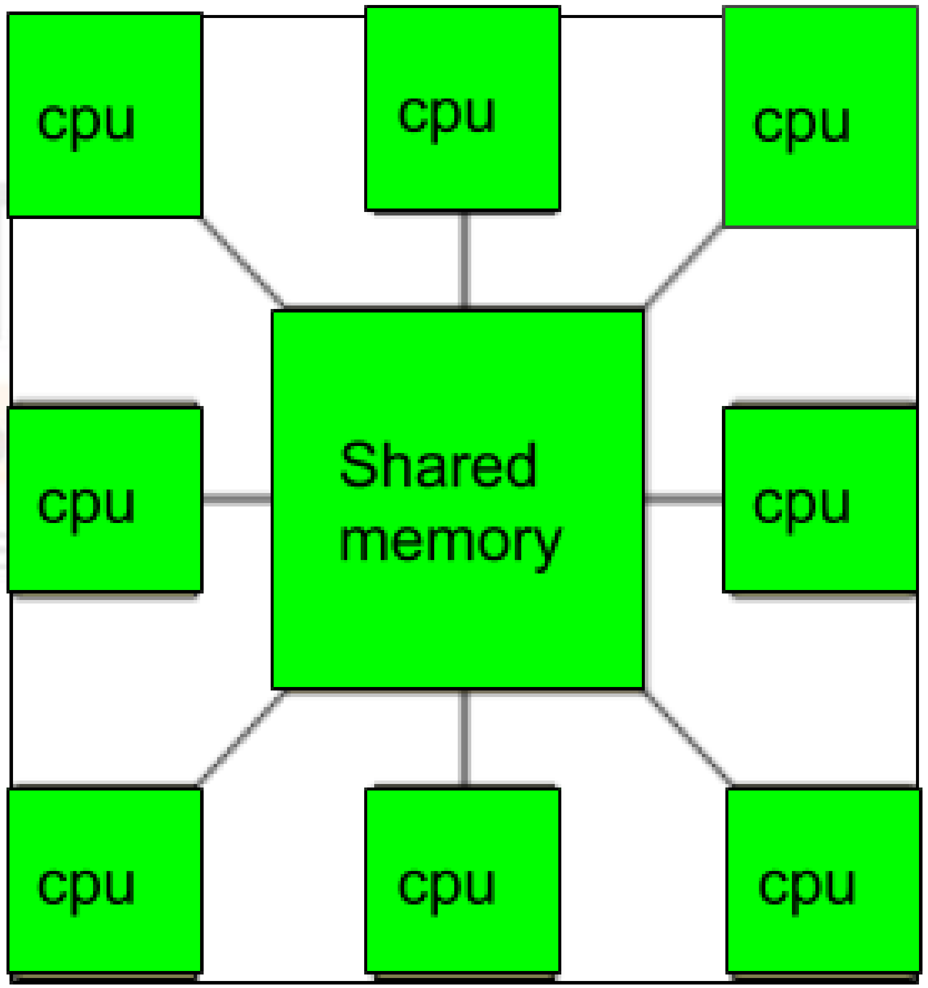
Introduction

A computer system that has two or more CPUs and full access to a single RAM is referred to as a multiprocessor. The primary goal of using a multiprocessor is to increase the system's execution speed, with fault tolerance and application matching as secondary goals. There are two different kinds of multiprocessors:

•Distributed memory multiprocessors and

• Shared memory multiprocessors..

In distributed memory multiprocessors, each CPU has its own private memory as opposed to shared memory multiprocessors, where all of the CPUs share the common memory.



Introduction to Parallel and Distributed Computing

Multiprocessor

Applications of Multiprocessor

•As a single instruction and one data stream uniprocessor (SISD).

•using a multiprocessor, such as a single instruction, multiple data stream (SIMD) processor, which is frequently utilised for vector processing.

•Numerous instruction, single data stream (MISD), which is used to describe hyper-threading or pipelined processors, is one example of a multiple series of instructions in a single perspective.

•To execute several, distinct series of instructions in multiple viewpoints, such as multiple instructions and multiple data streams, within a single system (MIMD).

Benefits of using a Multiprocessor

- Enhanced performance.
- Multiple applications.
- Multi-tasking inside an application.
- High throughput and responsiveness.
- Hardware sharing among CPUs.

Multiprocessor

Applications of Multiprocessor

•As a single instruction and one data stream uniprocessor (SISD).

•using a multiprocessor, such as a single instruction, multiple data stream (SIMD) processor, which is frequently utilised for vector processing.

•Numerous instruction, single data stream (MISD), which is used to describe hyper-threading or pipelined processors, is one example of a multiple series of instructions in a single perspective.

•To execute several, distinct series of instructions in multiple viewpoints, such as multiple instructions and multiple data streams, within a single system (MIMD).

Benefits of using a Multiprocessor

- Enhanced performance.
- Multiple applications.

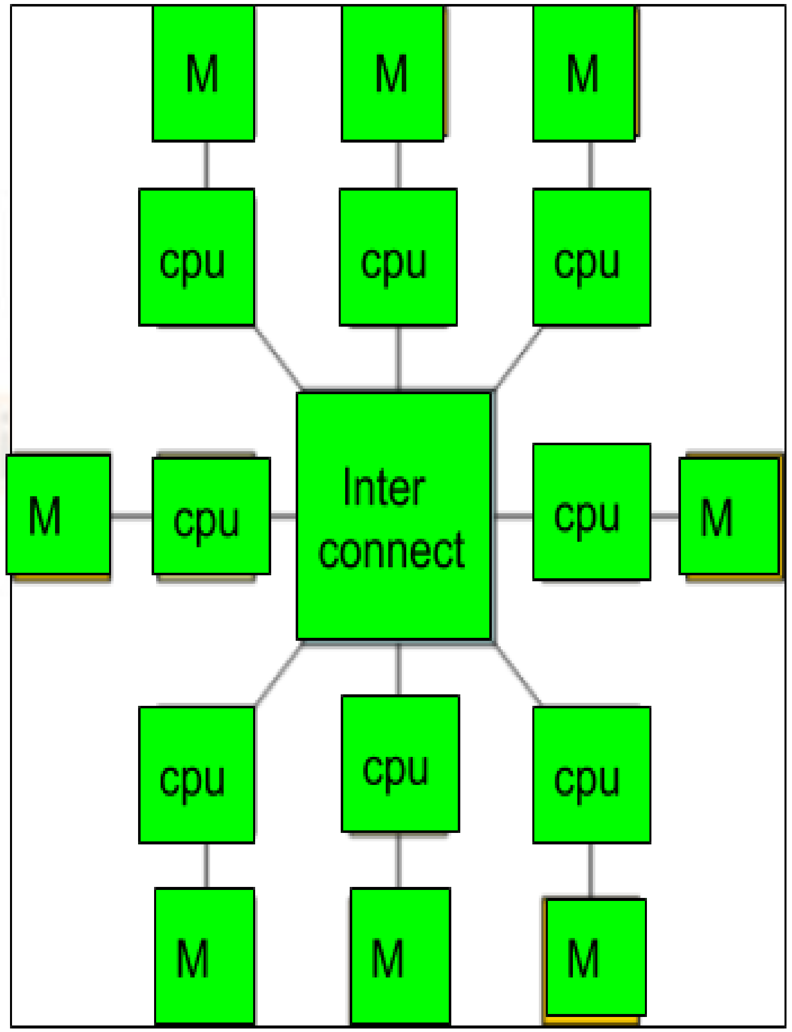
Introduction to Parallel and Distributed Computing

Multicomputer

A computer system with many processors that are linked together to solve a problem is referred to as a multicomputer system. Each CPU has a separate memory that is only accessible by that processor, and those processors are connected by interconnection network that allows them to communicate with one another.

It is possible to divide the task between the processors to complete it since the multicomputer allows messages to move between the processors. Consequently, distributed computing can be done using a multicomputer. A multicomputer is more affordable and simpler to construct than a multiprocessor.

.

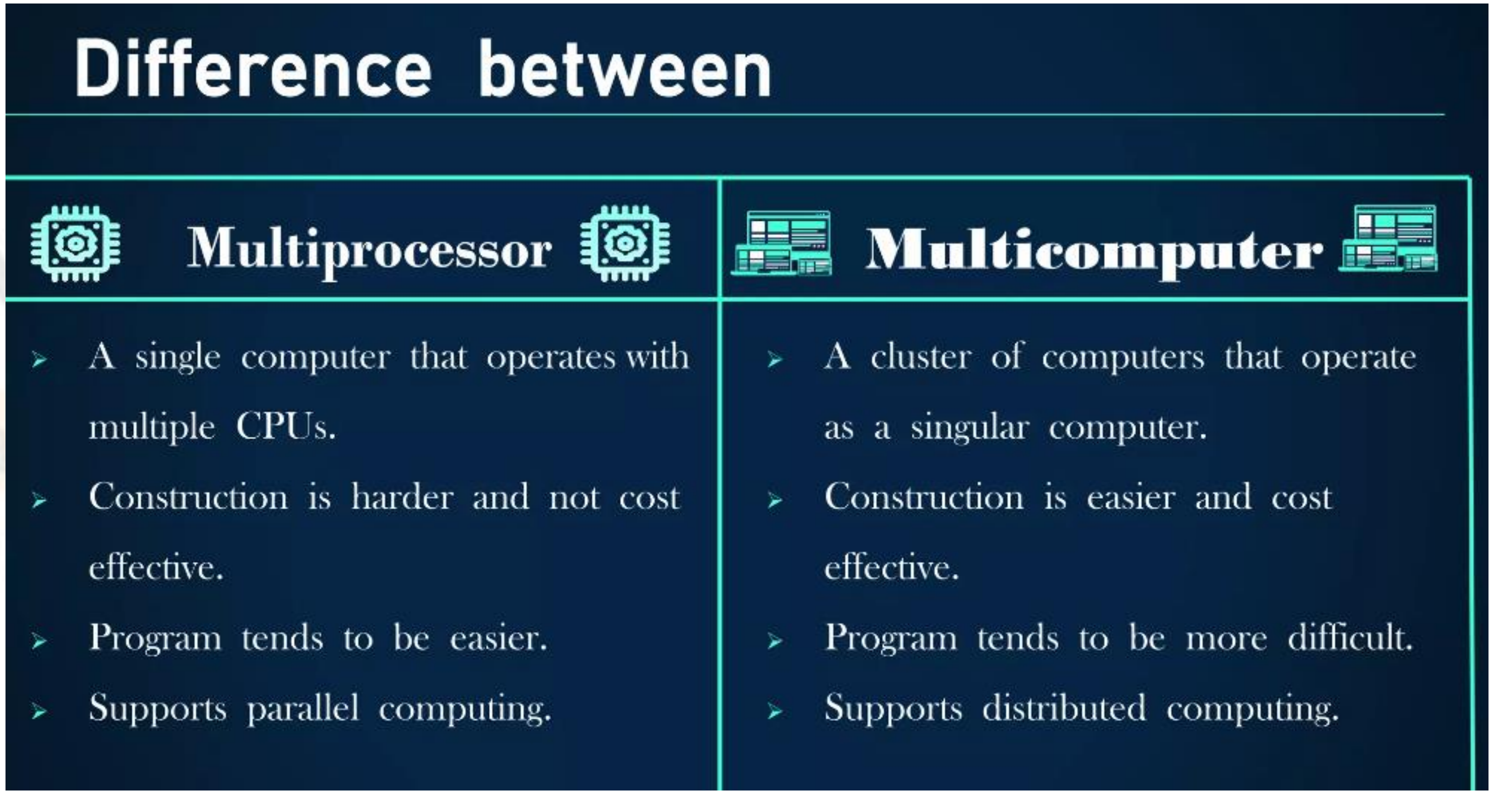


Multicomputer

Difference between multiprocessor and Multicomputer:

A multicomputer is a system with numerous processors connected over a network for the purpose of doing computations, whereas a multiprocessor is a system with two or more central processing units (CPUs) that can execute multiple jobs..

.



Difference between

Multiprocessor	Multicomputer
A single computer that operates with multiple CPUs.	A cluster of computers that operate as a singular computer.
Construction is harder and not cost effective.	Construction is easier and cost effective.
Program tends to be easier.	Program tends to be more difficult.
Supports parallel computing.	Supports distributed computing.

Message Passing Interface

What is message passing?

MPI is used to send messages from one process (computer, workstation etc.) to another. These messages can contain data ranging from primitive types (integers, strings and so forth) to actual objects.

Why send messages? – MPI is helpful whenever you need several workstations (or clusters) to work together efficiently and effectively.

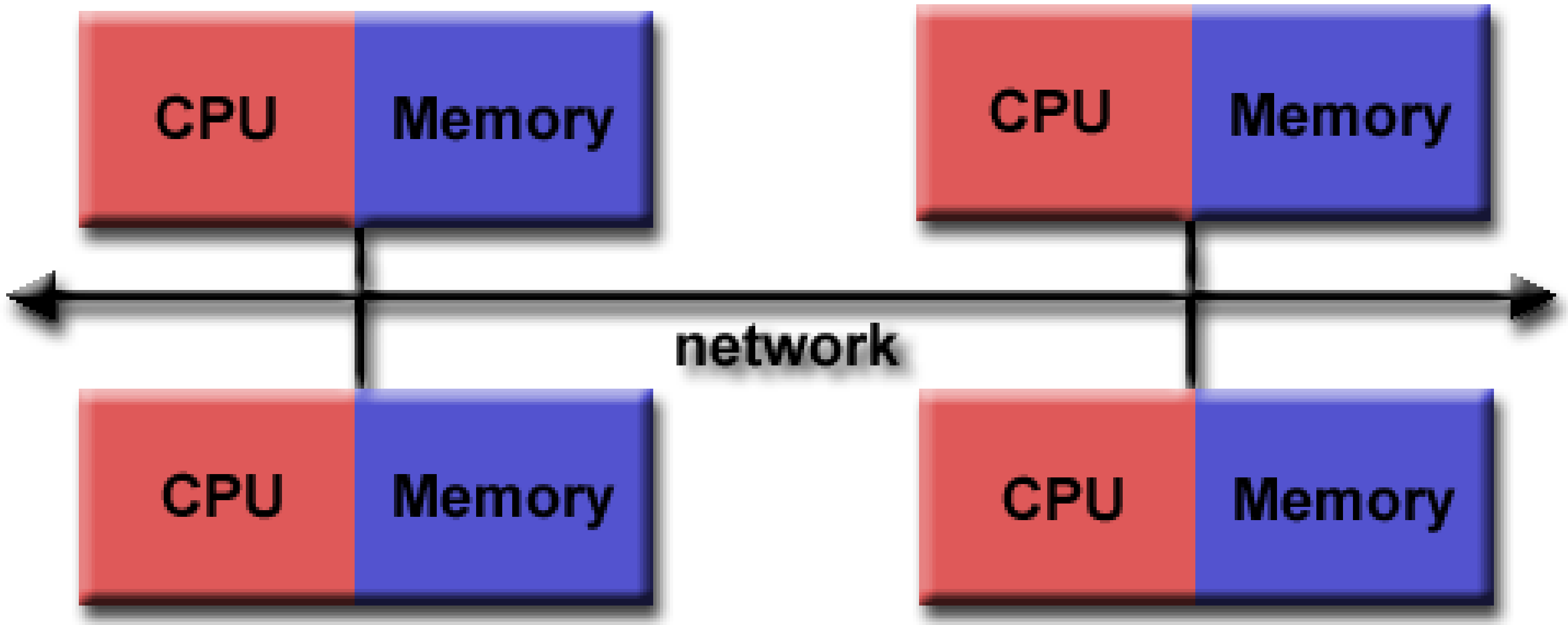
How do you send messages? – Programmer makes use of an Application Programming Interface (API) – API specifies the functionality of high-level communication routines – API's functions give access to a low-level implementation that takes care of sockets, buffering, data copying, message routing, etc

Introduction to Parallel and Distributed Computing

Message Passing Interface

Programming Model

- Originally, MPI was designed for distributed memory architectures, which were becoming increasingly popular at that time (1980s - early 1990s).

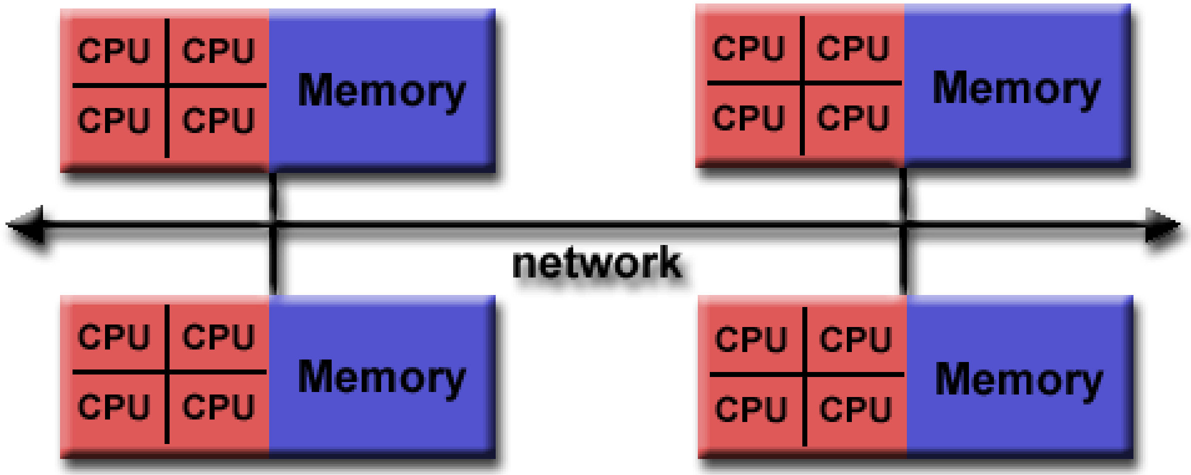


Introduction to Parallel and Distributed Computing

Message Passing Interface

Programming Model

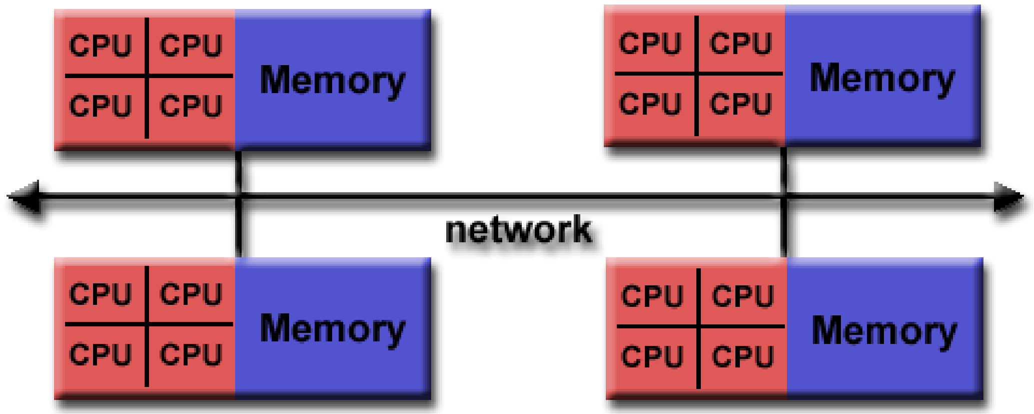
- As architecture trends changed, shared memory SMPs were combined over networks creating hybrid distributed memory / shared memory systems.
- MPI implementors adapted their libraries to handle both types of underlying memory architectures seamlessly. They also adapted/developed ways of handling different interconnects and protocols.



Message Passing Interface

Programming Model

Today, MPI runs on virtually any hardware platform:



- Distributed Memory
- Shared Memory
- Hybrid
- The programming model *clearly remains a distributed memory model* however, regardless of the underlying physical architecture of the machine.
- All parallelism is explicit: the programmer is responsible for correctly identifying parallelism and implementing parallel algorithms using MPI constructs.

Collective communication and synchronization points

Collective communication entails data communication using all processes within of a specific communicator; the default communicator is known as MPI COMM WORLD and contains all processes. A collective call must be issued by each process running inside the communicator.

Point-to-point communications will not obstruct group communication, and vice versa. Moreover, the usage of tags is not required for group communications. There is no assurance that a function will be synchronizing, and collective communication calls require matching send and receive buffers in order to function (except for barrier). Moreover, all group communication efforts are blocked. When using collective communication activities, bear the following in mind.

Collective communication and synchronization points

Types of collective communication

- Barrier Synchronization – Blocks until all processes have reached a synchronization point.
- Data Movement (or Global Communication) – Broadcast, Scatters, Gather, All to All transmission of data across the communicator.
- Collective Operations (or Global Reduction) – One process from the communicator collects data from each process and performs an operation on that data to compute a result.

Collective communication and synchronization points

Data Grouping Communication

Process identifiers (hence referred to as processes) are implementation-dependent objects, and a group is an ordered collection of those objects. A group's processes are each assigned an integer rank. Ranks begin at zero and are continuous. As groups are represented by opaque group objects, they cannot be passed directly across processes. In a communicator, a group is used to categorise and rank members in a communication "world," giving them specific names within that communication "universe.“

There is a special pre-defined group: ***MPI_GROUP_EMPTY***, which is a group with no members. The predefined constant ***MPI_GROUP_NULL*** is the value used for invalid group handles.

Collective communication and synchronization points

Data Grouping Communication

Process identifiers (hence referred to as processes) are implementation-dependent objects, and a group is an ordered collection of those objects. A group's processes are each assigned an integer rank. Ranks begin at zero and are continuous. As groups are represented by opaque group objects, they cannot be passed directly across processes. In a communicator, a group is used to categorise and rank members in a communication "world," giving them specific names within that communication "universe.“

There is a special pre-defined group: ***MPI_GROUP_EMPTY***, which is a group with no members. The predefined constant ***MPI_GROUP_NULL*** is the value used for invalid group handles.

Introduction to Parallel and Distributed Computing

Collective communication and synchronization points

Groups vs. Communicators:

A group is an organized collection of actions. Each process in a group has a certain integer rank assigned to it. When N is the number of processes in the group, rank values range from zero to N-1. In MPI, an object that represents a group exists in system memory. Only a "handle" gives the programmer access to it. An thing that communicates is almost always associated with a group.

A process that can communicate with another is referred to as a communicator. A communicator must be specified in every MPI message. The communicator can be thought of as a simple "tag" that needs be included to MPI calls. Similar to groups, communicators are represented as objects in system memory and are only reachable by "handles" for the programmer. For instance, MPI COMM WORLD is the handle for the communicator that contains all tasks.A group and a communicator are viewed by a coder as being the same thing. The main purpose of the group procedures is to describe which processes should be applied in order to build a communicator.

Introduction to Parallel and Distributed Computing

Summary

- The simplest definition of parallel computing is the concurrent utilization of several computer resources to address a computational issue.
- High performance computing (HPC) enables the quick resolution of challenging computing issues.
- Concurrent execution of the same task on several processing cores is known as data parallelism.
- Process identifiers (hence referred to as processes) are implementation-dependent objects, and a group is an ordered collection of those objects.
- A process that can communicate with another is referred to as a communicator. A communicator must be specified in every MPI message.

Self Assessment Questions

Q.1 Parallel processing may occur

a) in the instruction stream

b) in the data stream

c) both[a] and [b]

d) none of the above

Answer: both[a] and [b]

Introduction to Parallel and Distributed Computing

Self Assessment Questions

Q.2 Systems that do not have parallel processing capabilities are

a) sisd

b) simd

c) mimd

d) all of the above

Answer: sisd

Introduction to Parallel and Distributed Computing

Self Assessment Questions

Q.3 It is the simultaneous use of multiple compute resources to solve a computational problem

a) Parallel computing

b) Single processing

c) Sequential computing

d) None of these

Answer: Parallel computing

Self Assessment Questions

Q.4 Parallel computing can be used in ___

(A) Science and engineering

(B) Database and data mining

(C) Real time simulation of systems

(D) All of the above

Answer: D

Introduction to Parallel and Distributed Computing

Self Assessment Questions

Q.5 The access time of memory is _________ the time required for performing any single CPU operation.

a) longer than

b) shorter than

c) negligible than

d) same as

Answer: a longer than

Threat Management

Assignment

1. Explain Parallel and Distributed Computing with diagram.
2. Describe Advantages and Disadvantages of Distributed Computing.
3. Why should you use HPC?
4. Explain Flynn’s Taxonomy with types.
5. Differentiae Multiprocessor and multicomputer.
6. Define Message Passing Interface

Threat Management

Document Links

Topic	URL	Notes
Message passing interface	http://www.rc.usf.edu/tutorials/classes/tutorial/mpi/chapter8.html	This link explains about MPI
Flynns Classification of Parallel Processing Systems	https://www.ques10.com/p/10175/list-the-flynns-classification-of-parallel-proce-1/	This link explains about Flynns Classification
parallel processing	https://www.techtarget.com/searchdatacenter/definition/parallel-processing#:~:text=Parallel%20processing%20is%20a%20method,time%20to%20run%20a%20program.	This link explains about parallel processing

Threat Management

Video Links

Topic	URL	Notes
High performance computing	hhttps://archive.nptel.ac.in/courses/106/105/106105033/	Explains about pos tagging and stop word
What is High performance computing	https://www.youtube.com/watch?v=0biElmaDfFs	This video talks about High performance computing basic

Threat Management

E-Book Links

Topic	URL	Notes
Handbook ofHigh-Performance Computing	https://www.routledge.com/rsc/downloads/High_Performance_Computing_ChapterSampler.pdf	Ebook gives an comprehensive knowledge on High-Performance Computing