OUTLINES

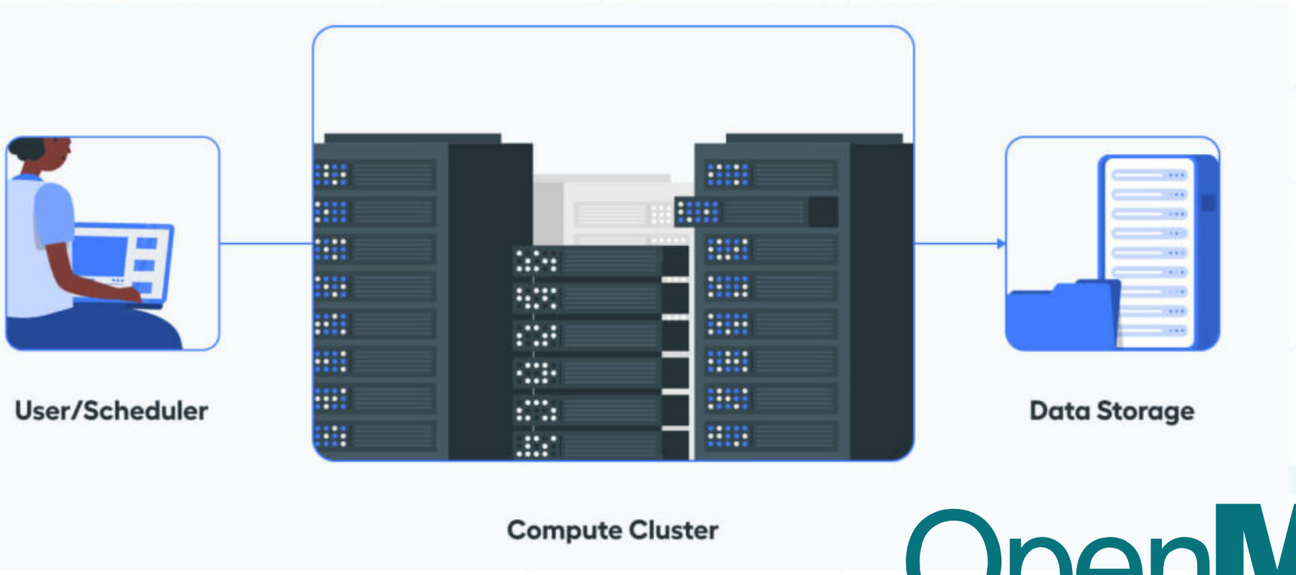
Parallel Programming(CDSC 604)

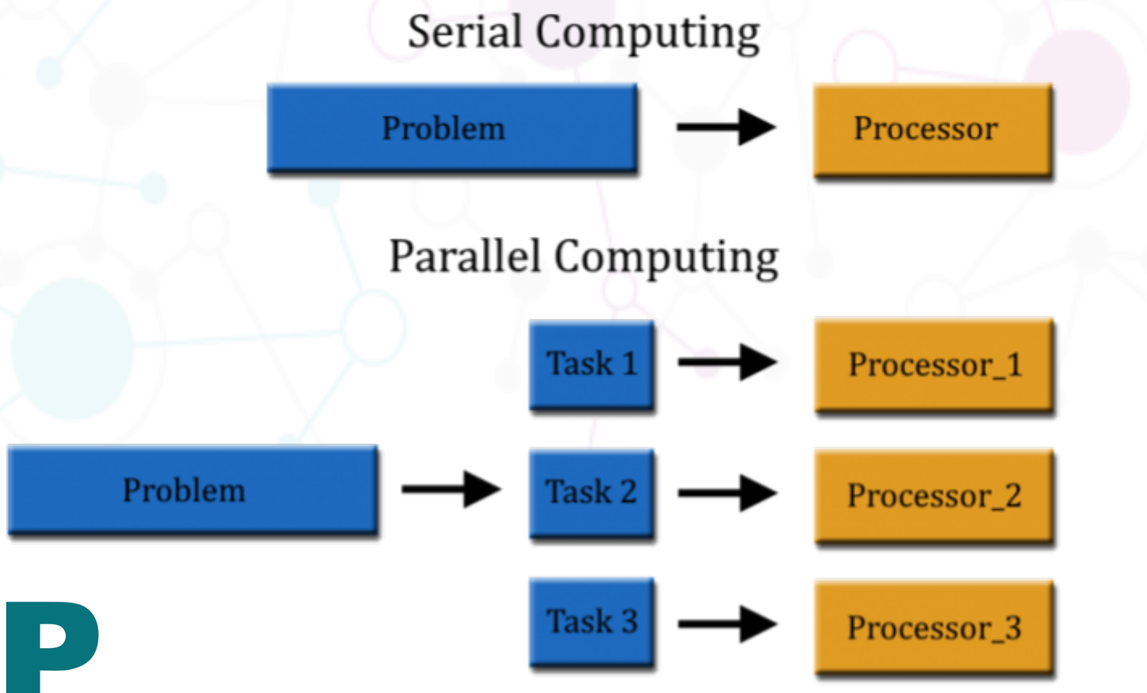
Mesfin Diro Chaka

Computational Data Science Program

Addis Ababa University

March, 2024

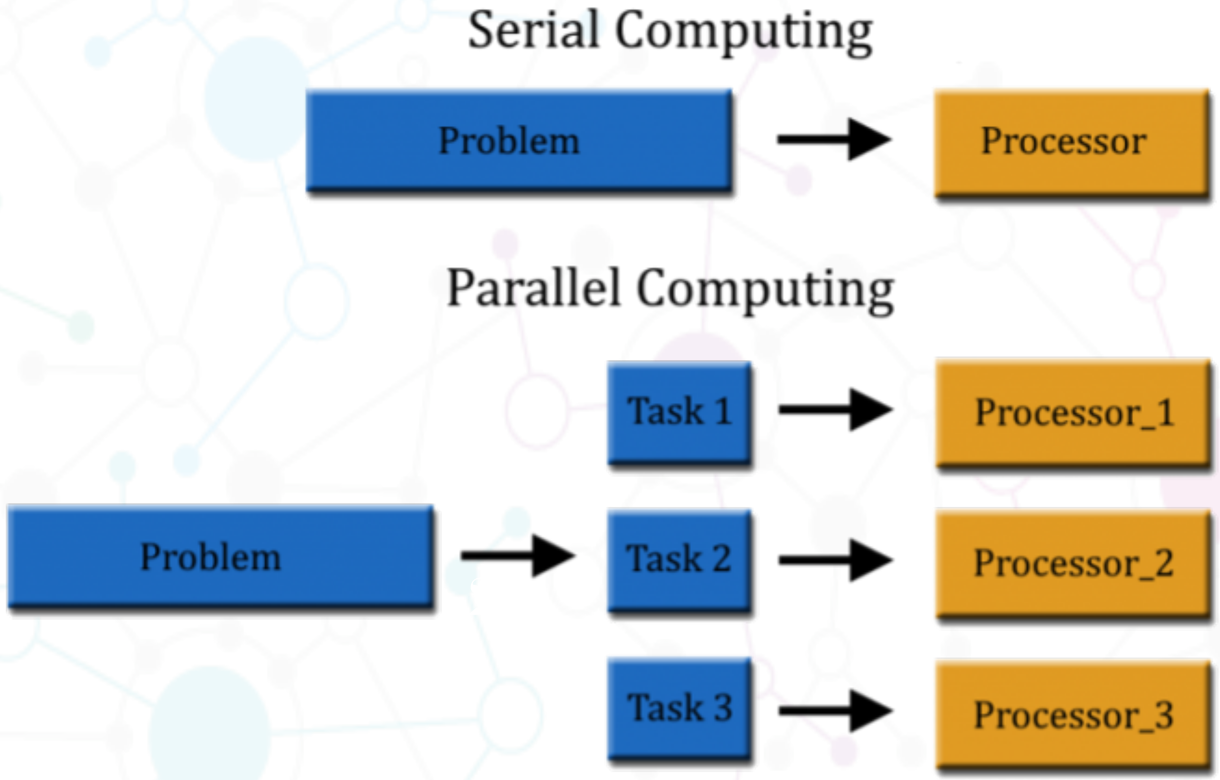




High Performance Computing(HPC)

Part 1: Lecture 1

- **Serial computation** involves executing instructions sequentially, one after another on a single processor.
- **Parallel computing** is a type of computing architecture in which many calculations or processes are carried out simultaneously.
- Parallel computing is a fundamental concept in **High Performance Computing (HPC)** that involves connecting multiple processors to memory, through high-speed networks
- HPC refers to the use of **parallel processing** with advanced computing technologies and systems to **solve complex problems** and process big data at high speeds.



High Performance Computing(HPC)

Part 1: Lecture 1

HPC involves the utilization of powerful computational resources including supercomputers, clusters, and specialized hardwares to perform computationally intensive tasks and simulations

- HPC focuses on achieving high processing speeds, large-scale parallel processing, and efficient utilization of computing resources
- Key characteristics of HPC include:
  - Parallel processing
  - High speed interconnects
  - Scalability
  - Specialized Hardwares like GPU
- Application of HPC are diverse and span across various fields including scientific research, engineering, healthcare, finance, climate modeling, oil, and gas exploration and many others.

Introduction to High Performance Computing

Part 1: Lecture 1

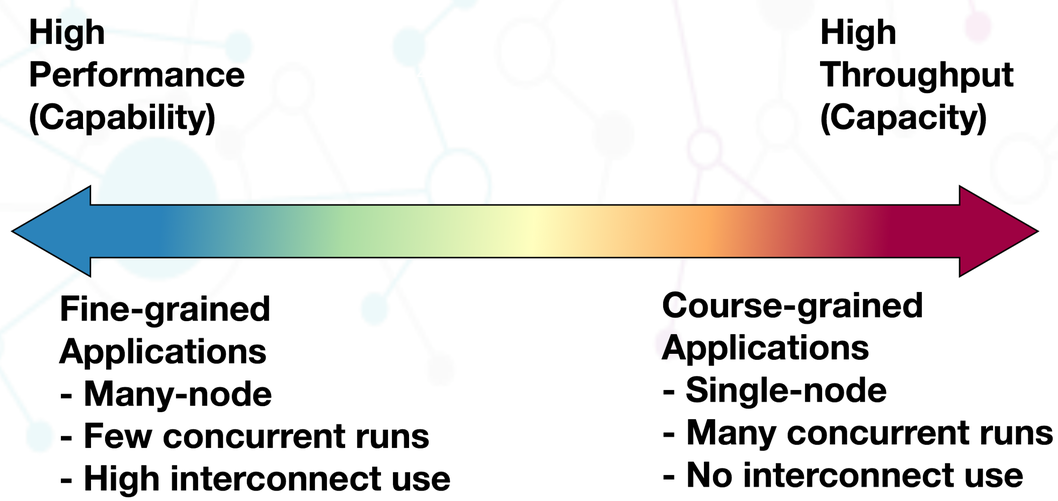
Cluster, Grid and cloud computing

- Cluster computing (HPC): several similar small to large computers with similar OS are connected locally to operate as a single system to aggregate the resources using job scheduling

- Grid computing (HTC): several large and dissimilar computers are connected in different physical location and they can operate independently as well to optimize resources using scale execution

- Cloud computing: the cloud provider has high-end small to large servers in different physical location to provide services to heterogenous client computers to economization of resources using self-managed job execution

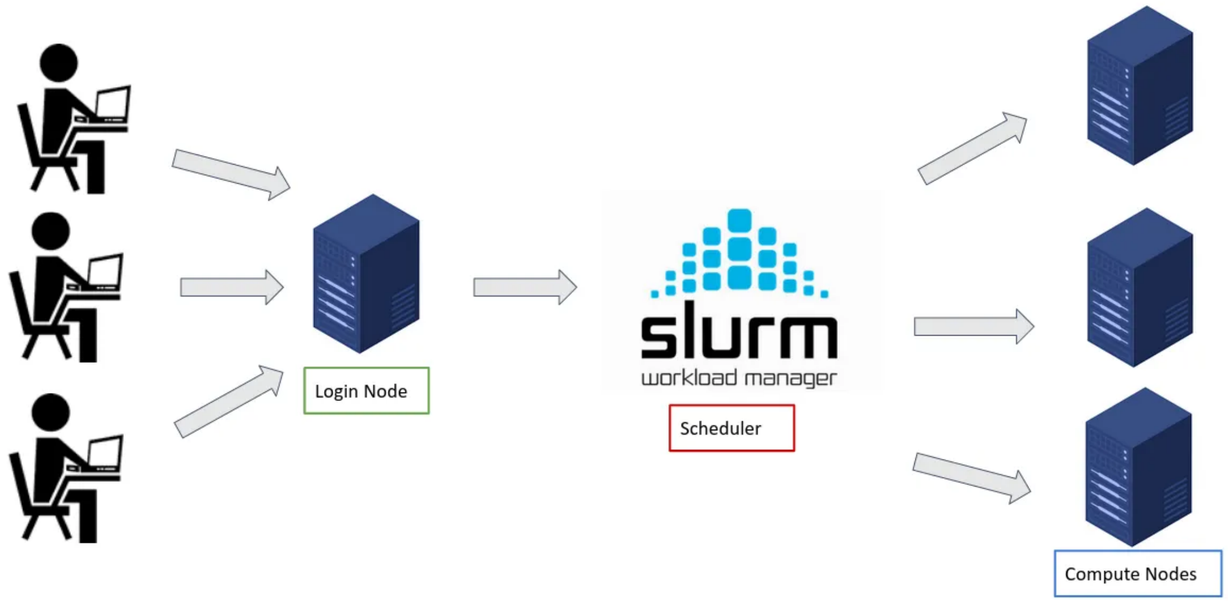
- High-Performance Computing (HPC) can occur on various platforms, including:
  - Workstations,
  - desktop, laptop, smart-phone
  - Supercomputer,
  - Cluster
  - Grid or cloud

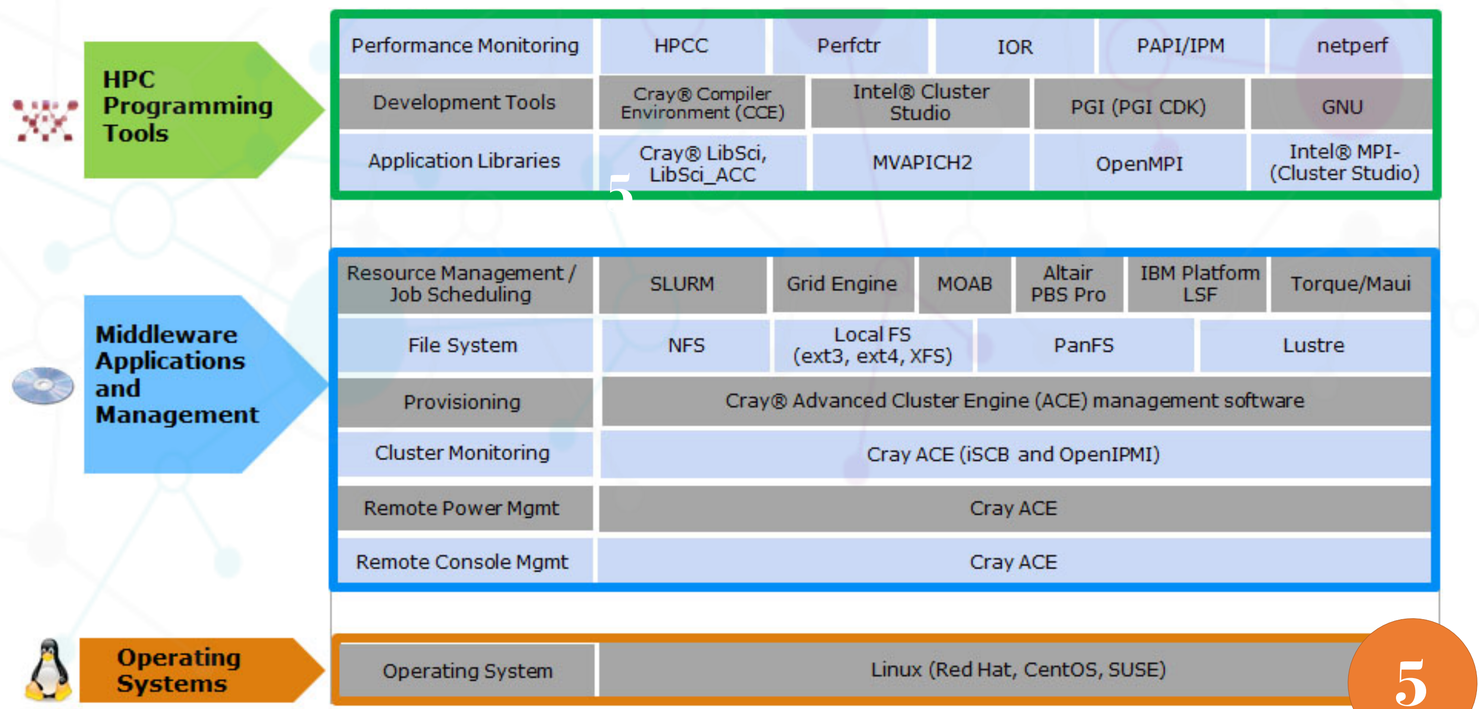


Introduction to High Performance Computing

Part 1: Lecture 1

- **HPC include work on "Four basic building blocks" in this course:**
  - **Theory (numerical laws, physical models, speed-up performance, etc)**
  - **Technology (multi-core, supercomputer, networks, storage , etc,)**
  - **Architecture (shared-memory, distributed-memory, interconnects, etc)**
  - **Software (libraries, schedulers, monitoring, applications, etc)**





Introduction to High Performance Computing

Part 1: Lecture 1

HPC is designed to deliver speed and performance, helping businesses save time and resources

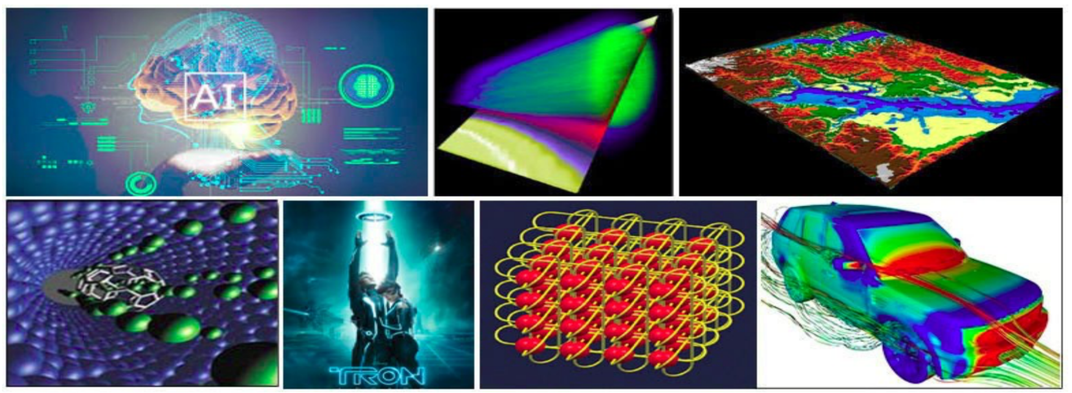
MANY FIELDS CAN USE HPC TECHNOLOGIES, INCLUDING:

01 — Engineering - designing and optimising new physical products, including automobiles, planes, structures, and consumer appliances.

02 — Scientific Research - basic and applied research to model climate change, more accurate weather forecasts, galaxy and star formation, and weapons modernisation.

03 — Finance - fast decisions based on very low latency calculations to make trading decisions.

04 — Healthcare - design new drugs, create personalised care, and recognise disease causes and possible contributors to these diseases.

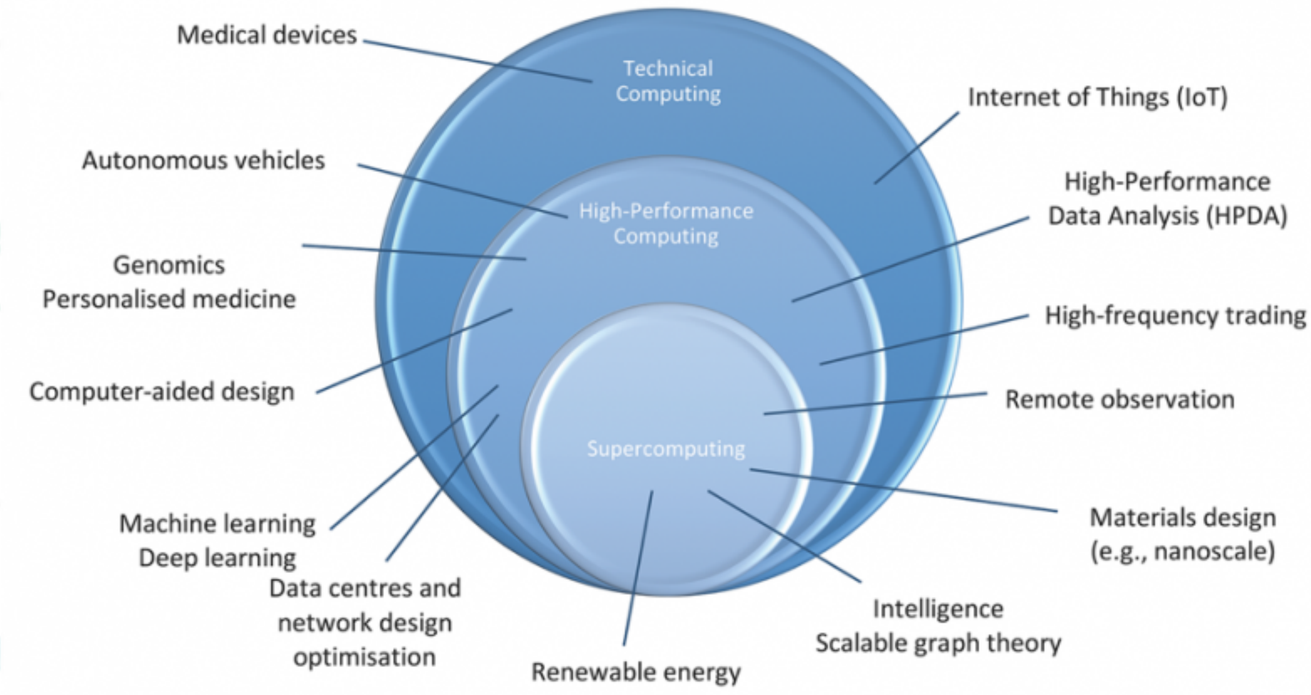


Introduction to High Performance Computing

Part 1: Lecture 1

Why would we care about HPC?

- **High-Performance Computing (HPC) is crucial for various reasons:**
  - **Improved Efficiency: HPC systems can process large amounts of data quickly, enhancing data processing capabilities**
  - **Cost Reduction: In industry, HPC is used to improve products, reduce production costs, and accelerate the development of new products, leading to cost savings.**
  - **Speed and Performance: HPC delivers high speed and performance, making it essential for handling complex computational tasks efficiently**
  - **Infrastructure Support: HPC has been a fundamental component of infrastructure and hardware engineering for many years, highlighting its importance in technological advancements**



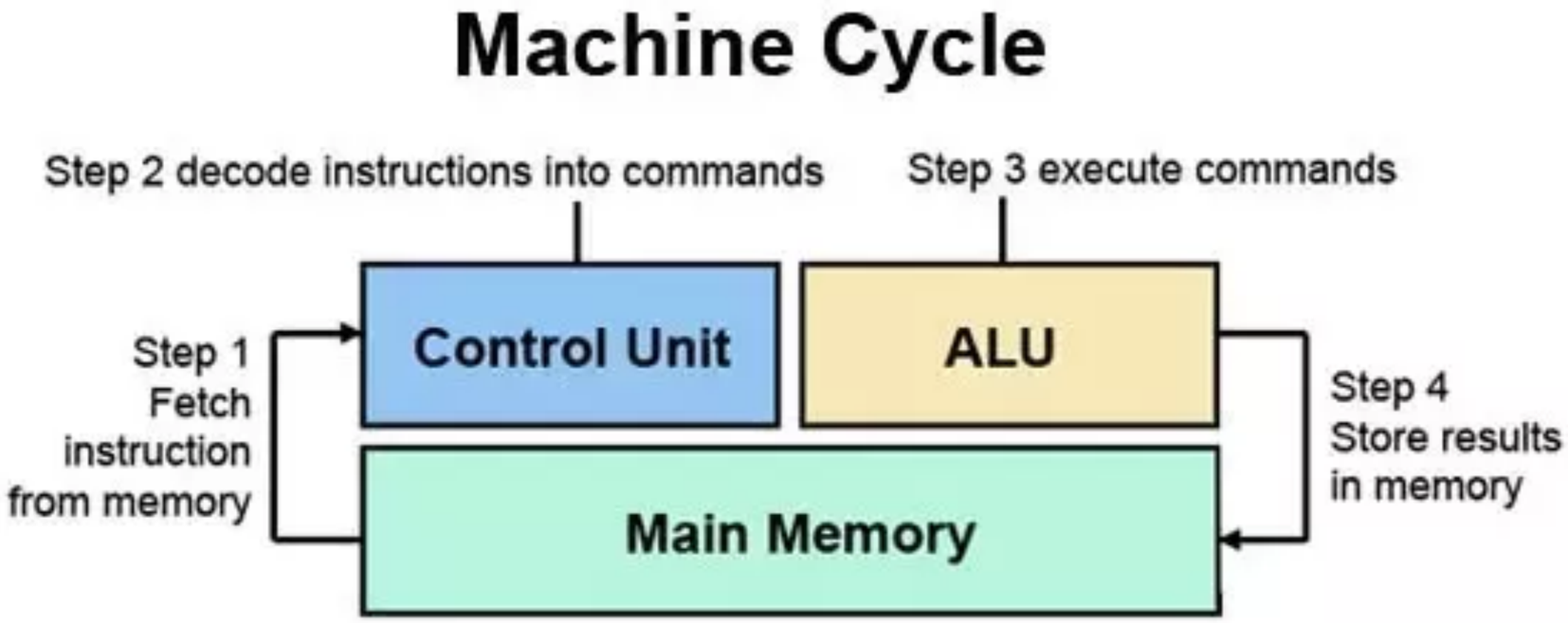
Galaxy Formation | Planetary Movments | Climate Change

Introduction to High Performance Computing

Part 1: Lecture 1

What determines HPC performance?

- How fast is my CPU?
- How fast can I move data around? How well
- can I split work into pieces?



Introduction to High Performance Computing

Part 1: Lecture 1

HPC Performance Meterics

- The peak performance of a computer system is the theoretical maximum computational power when all CPU components work at maximum speed.
- Tech specifications describe the theoretical peak, while benchmarks measure the real peak performance.
- Applications demonstrate the actual performance value in real-world scenarios.
- CPU performance is often measured in Floating Point Operations Per Second (GFLOP/s).
- Real performance is closely related to memory bandwidth, measured in Gigabytes per second (GBytes/s), as it affects data access and processing speed.

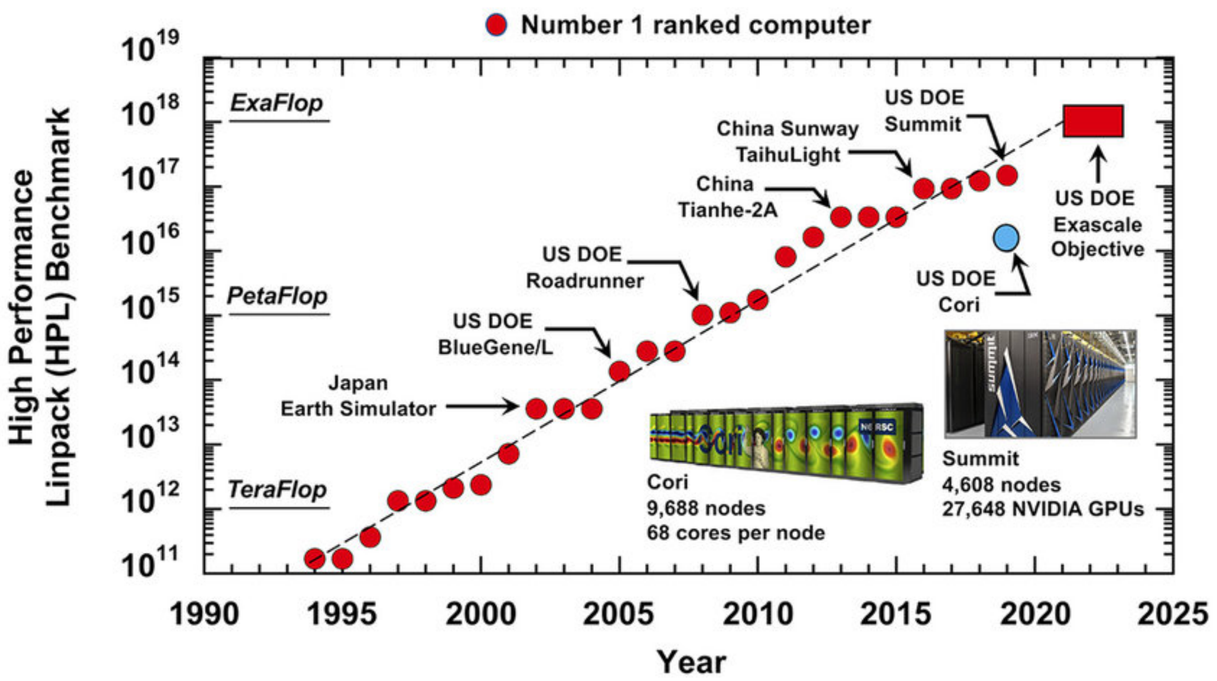
Introduction to High Performance Computing

Part 1: Lecture 1

How fast is my CPU?

- CPU power is measured in FLOPS
  - number of floating point operations per second
  - FLOPS = number of cores x clock x FLOP/cycle
- FLOP/cycle is number of double precision addition and/or multiplications completed per clock
  - architectural limit (data load/store)
  - depend also by the instruction set supported

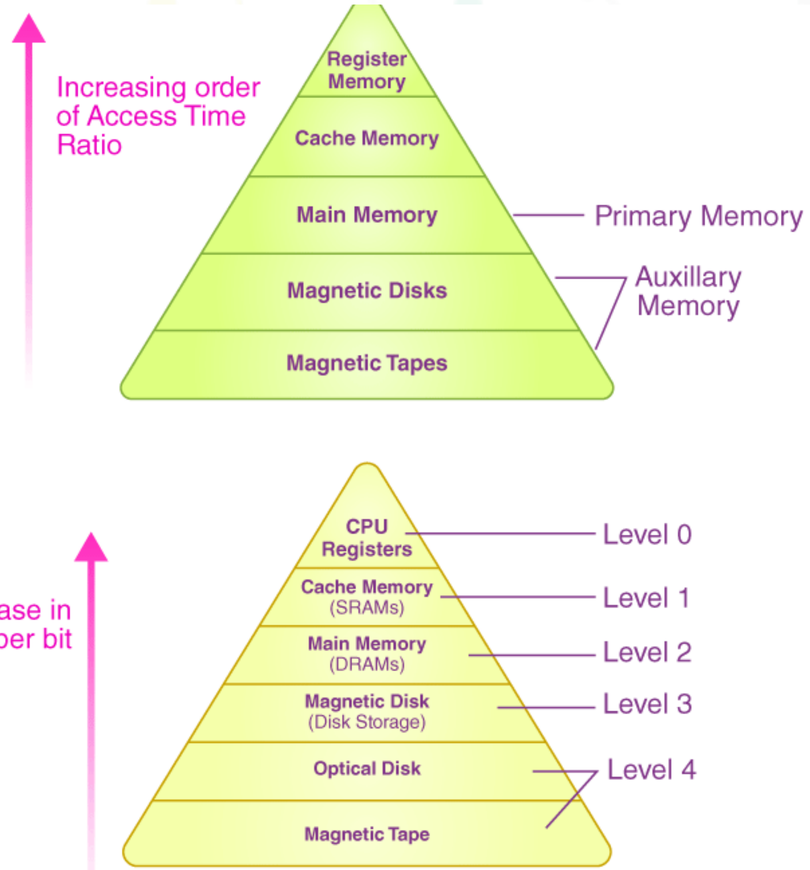
For example, if a CPU runs at 2.5 GHz with 8 FLOPs per clock cycle, the calculation would be: 2.5 GHz $\times 8 = 20$ GigaFLOP s

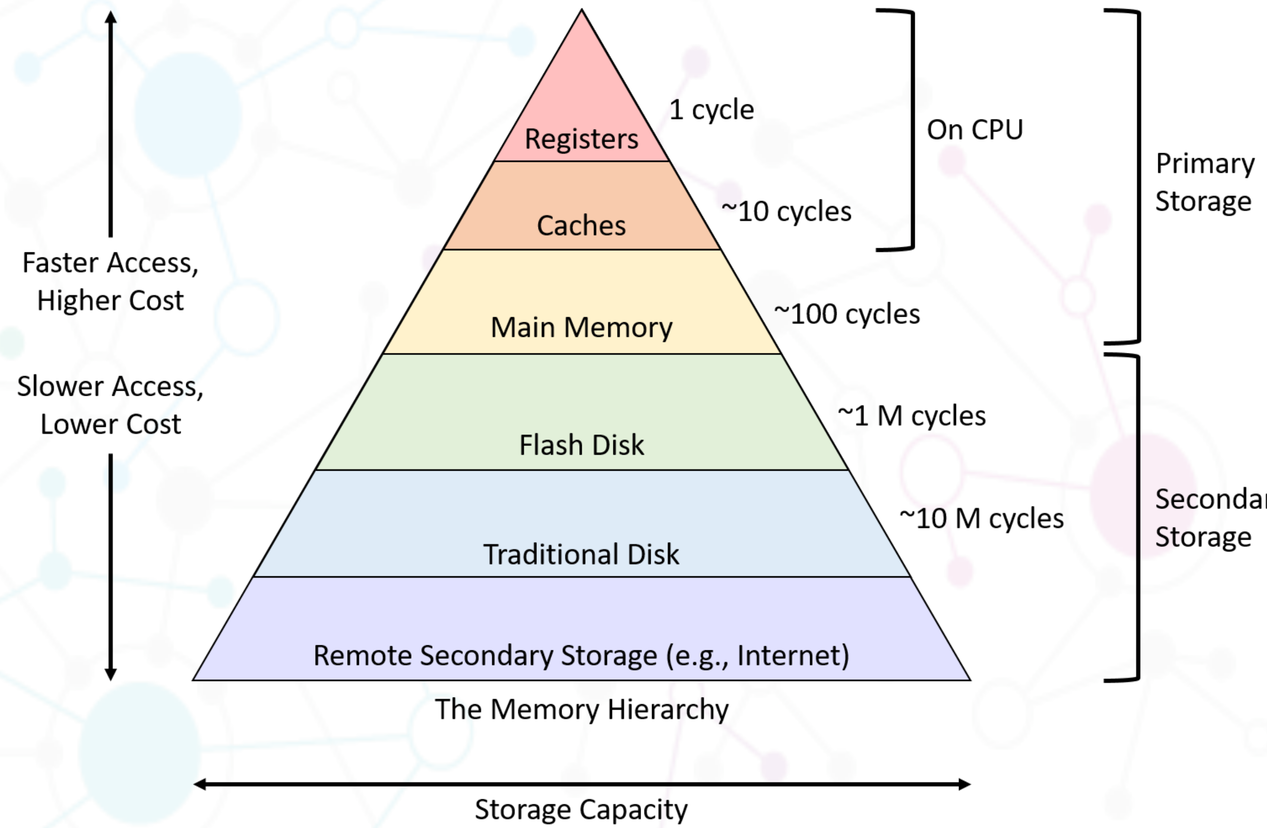


Introduction to High Performance Computing

Part 1: Lecture 1

How fast is my CPU?





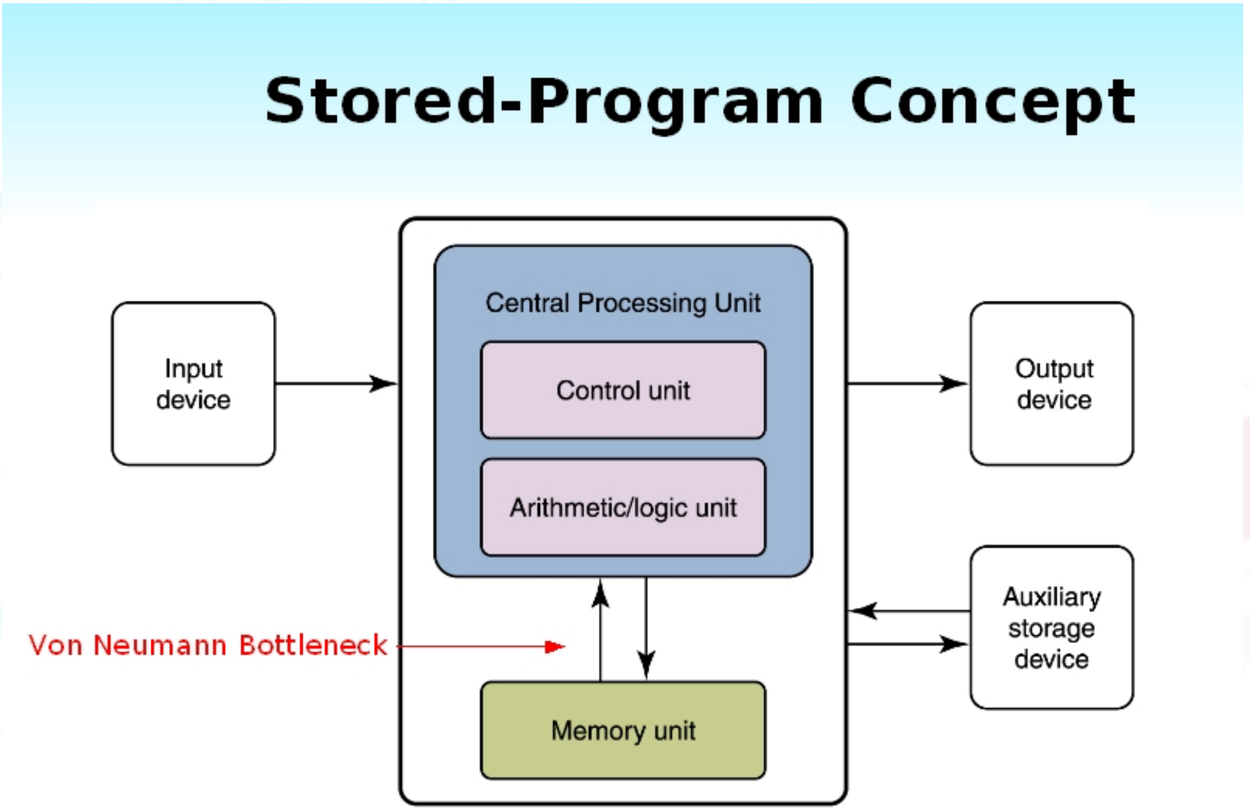
CPU memory Hierarchy

Introduction to High Performance Computing

Part 1: Lecture 1

Stored Program concept in von Neumann architecture

- Control Unit and arithmetic units together with appropriate interfaces to memory and I/O are called the Central Processing Unit(CPU)
- In a stored-program computer(in binary), the processor fetches instructions from memory, decodes them, and executes them
- The “Program which feeds the control unit is stored in memory together with any data the arithmetic unit requires
- This concept was developed by von Neumann
- von Neumann Bottleneck: A limitation where instruction fetch and data operation cannot occur at the same time due to shared common bus, often limiting performance.



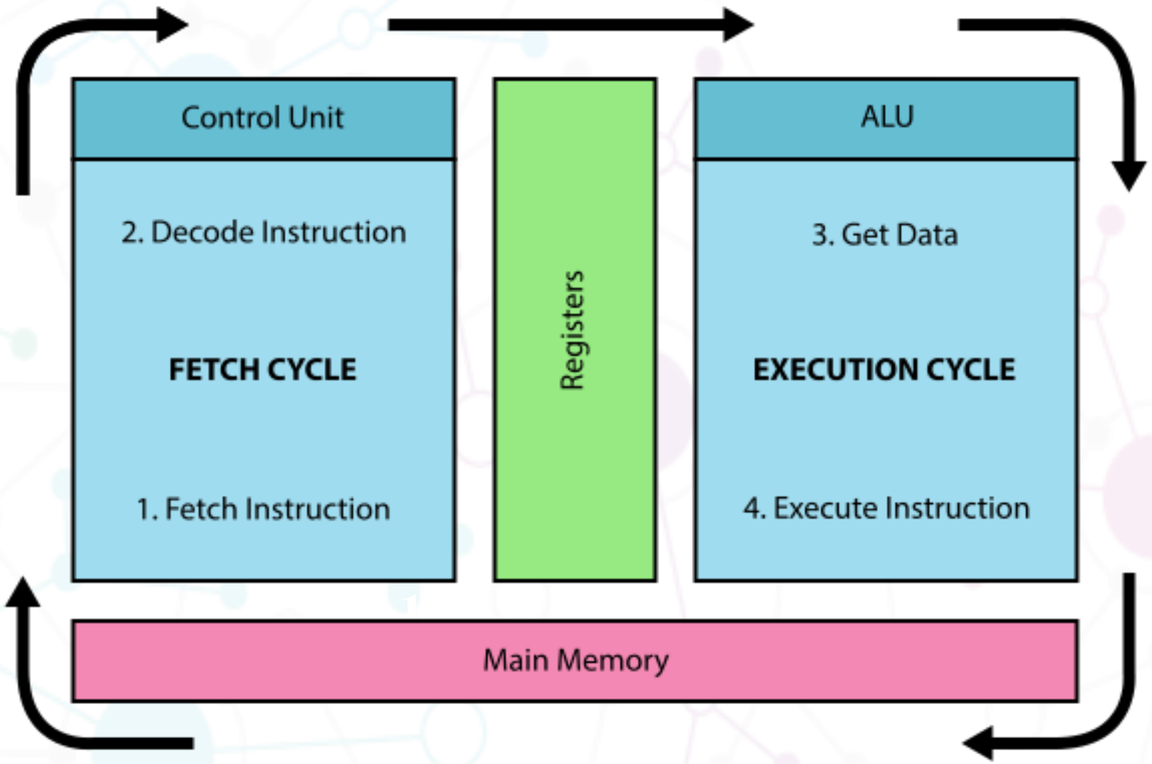
von Neumann architecture

Introduction to High Performance Computing

Part 1: Lecture 1

Instruction cycles

- The instruction cycle, also known as the fetch-execute cycle, is the sequence of steps a CPU follows to execute a program. It consists of the following stages:
  - Fetch: The CPU retrieves the next instruction from memory.
  - Decode: The CPU deciphers the instruction and determines the operation to be performed.
  - Execute: The CPU performs the operation specified in the instruction.
  - Store: The CPU stores the results of the operation in memory or registers.

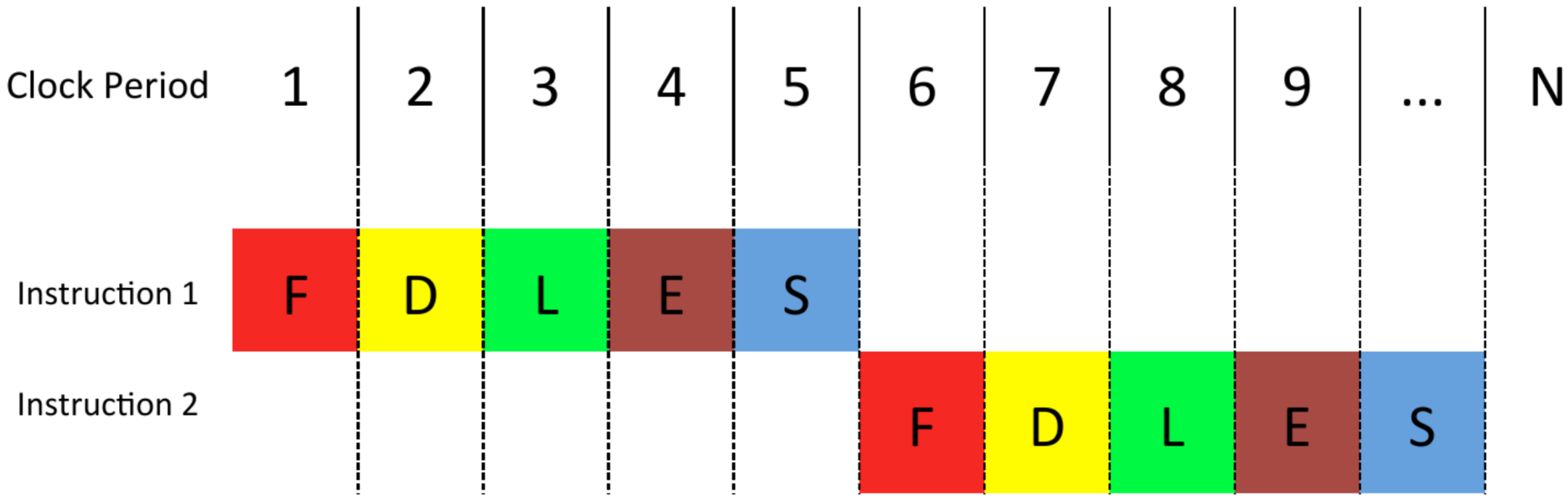


Introduction to High Performance Computing

Part 1: Lecture 1

Sequential Processing

- **Sequential processing involves executing tasks in a linear sequence, while pipelining involves dividing tasks into smaller units and executing them simultaneously.**

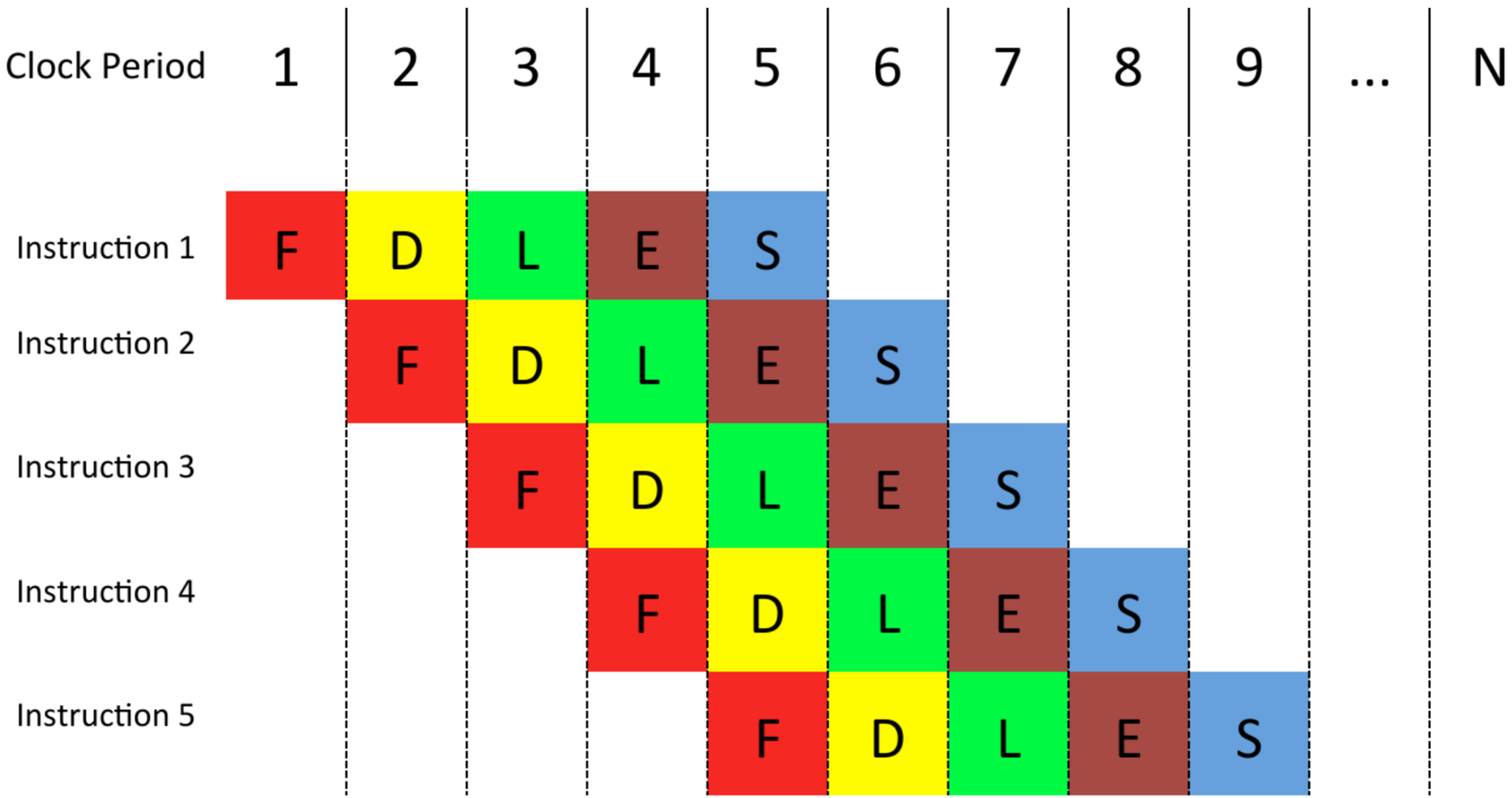


Introduction to High Performance Computing

Part 1: Lecture 1

Pipelining

- **Pipelining can lead to faster execution times and more efficient processing, but it requires careful scheduling and coordination of tasks to ensure correct execution**

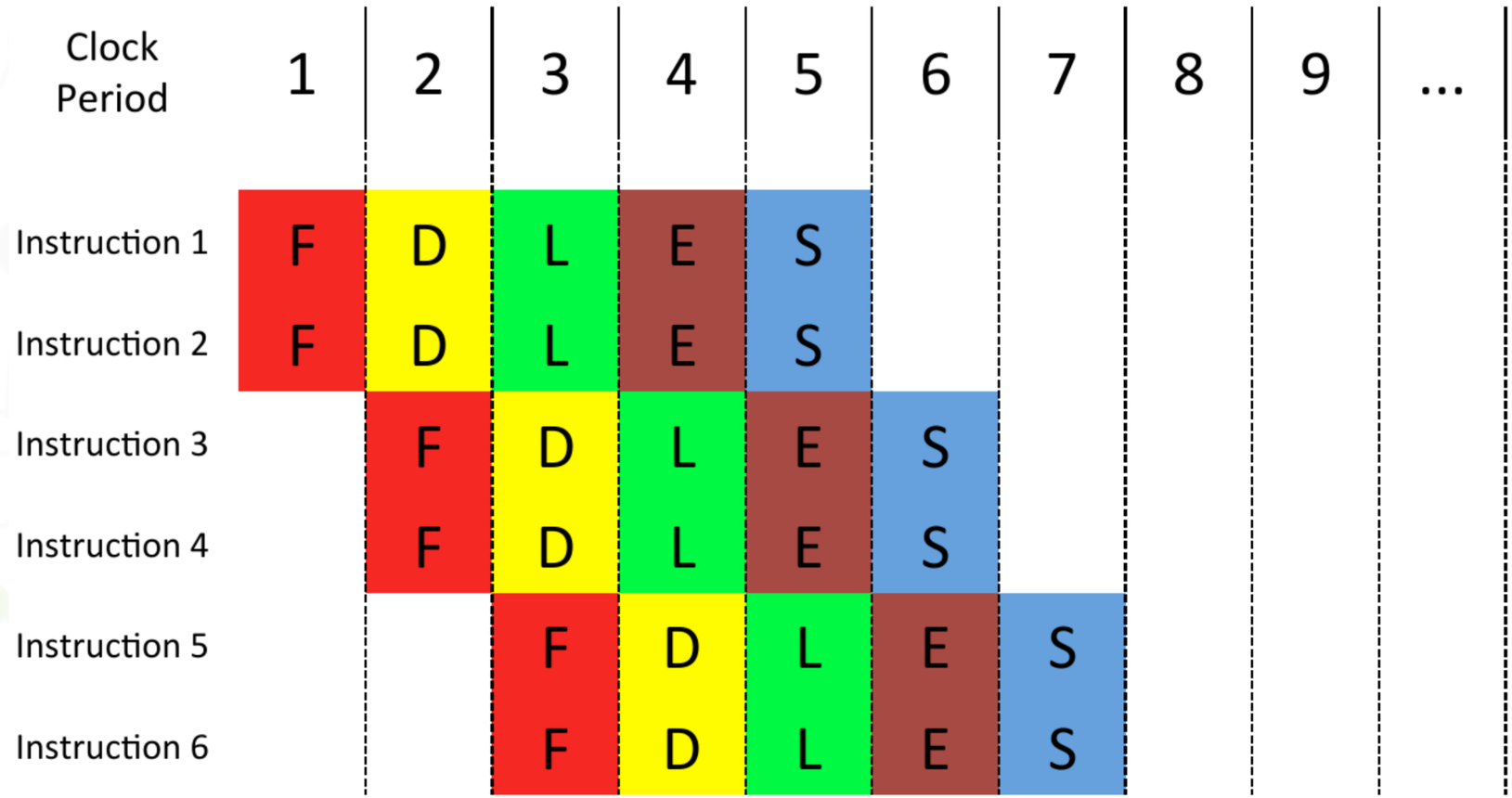


Introduction to High Performance Computing

Part 1: Lecture 1

Superscalaring

- Superscaling is a technique used in computer processors to increase the number of instructions that can be executed in a single clock cycle.
- It involves the processor being able to issue multiple instructions in a single clock cycle, with redundant facilities to execute them

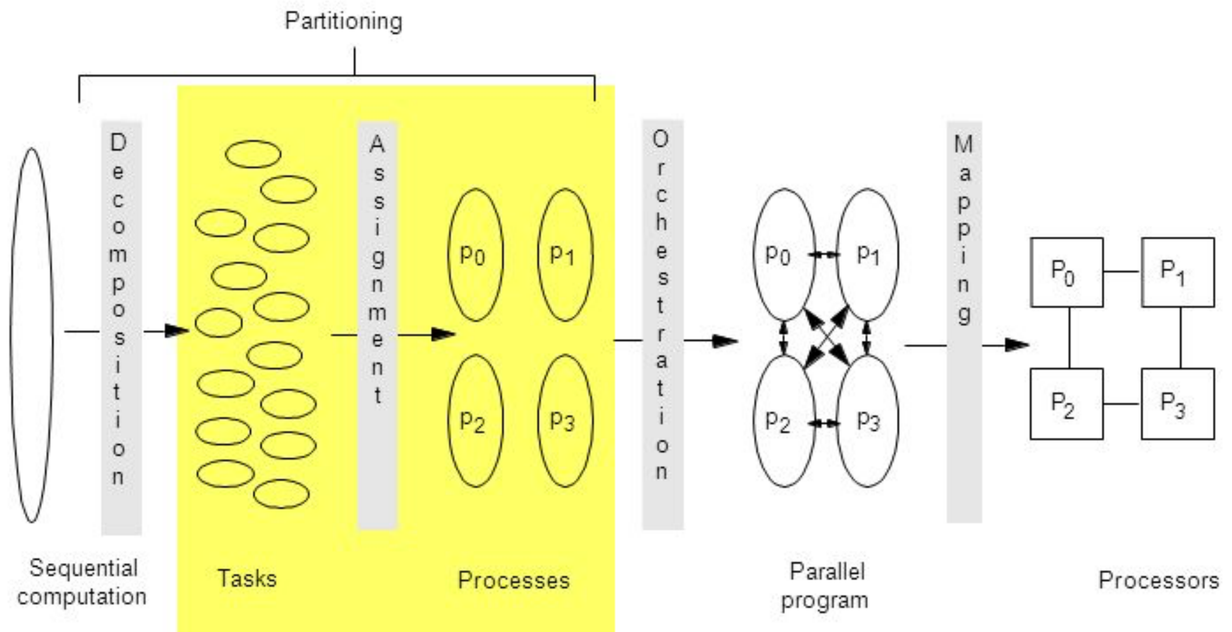


Introduction to High Performance Computing

Part 1: Lecture 1

Steps in Creating a Parallel Program

- Creating a parallel program involves several key steps, including decomposition, assignment, orchestration, and mapping
- Decomposition: Break down the task into smaller subtasks for concurrent execution.
- Assignment: Allocate tasks to processing units to optimize workload distribution.
- Orchestration: Coordinate task execution, manage synchronization, and ensure proper sequencing.
- Mapping: Assign tasks and data to physical processing units for efficient parallel processing.



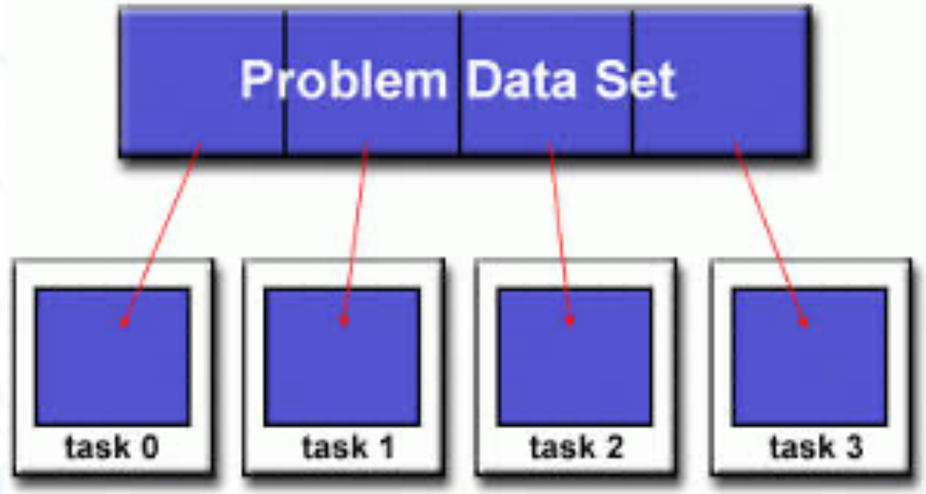
4 steps: Decomposition, Assignment, Orchestration, Mappin

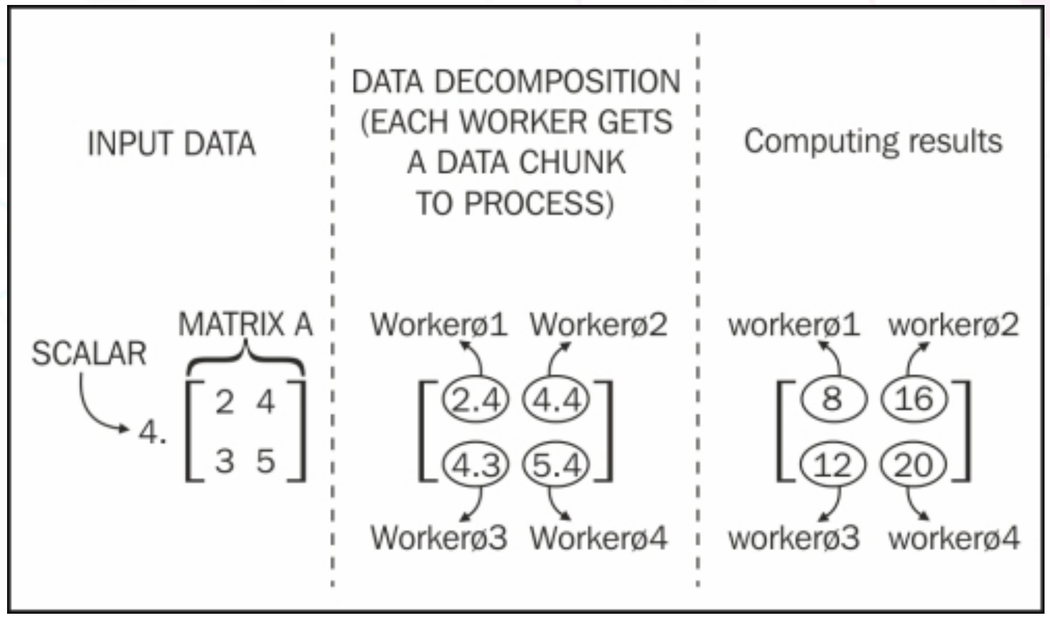
Introduction to High Performance Computing

Part 1: Lecture 1

Decomposition

- Breaking up computation into tasks to be divided among processes
- Identify concurrency and decide level at which to explicit it
- There are two main parallel decomposition:
  - Data-based decomposition
  - Instruction-Based (Model) Decomposition



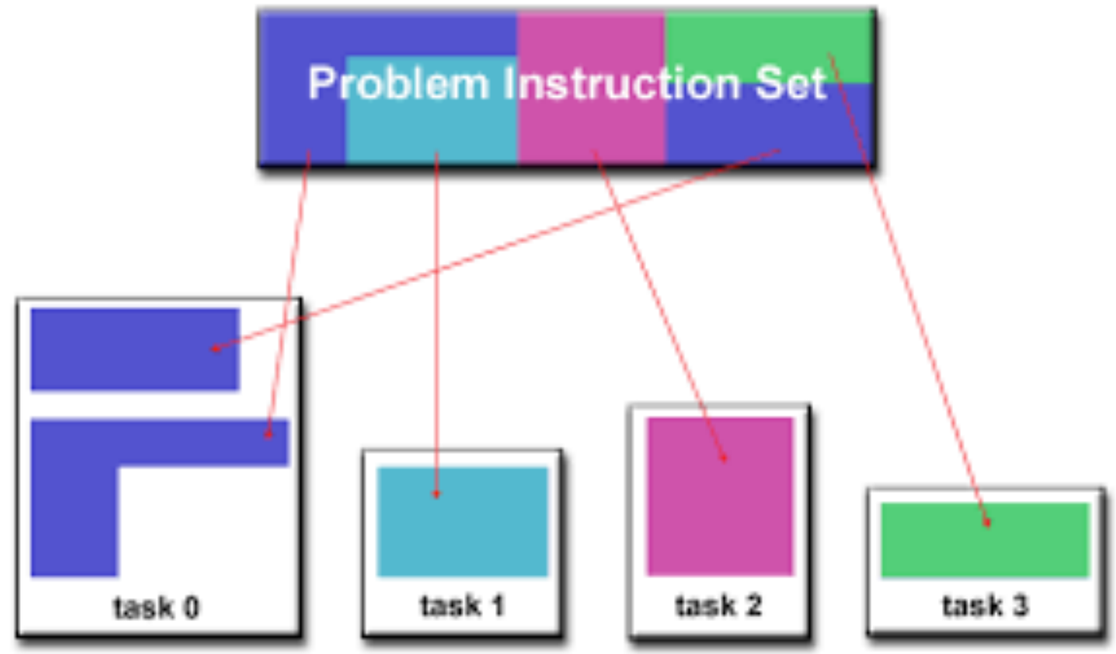


Introduction to High Performance Computing

Part 1: Lecture 1

Functional Decomposition

- The focus is on the computation that is to be performed rather than on the data
- Problem is decomposed according to the work that must be done
- Each task then performs a portion of the overall work



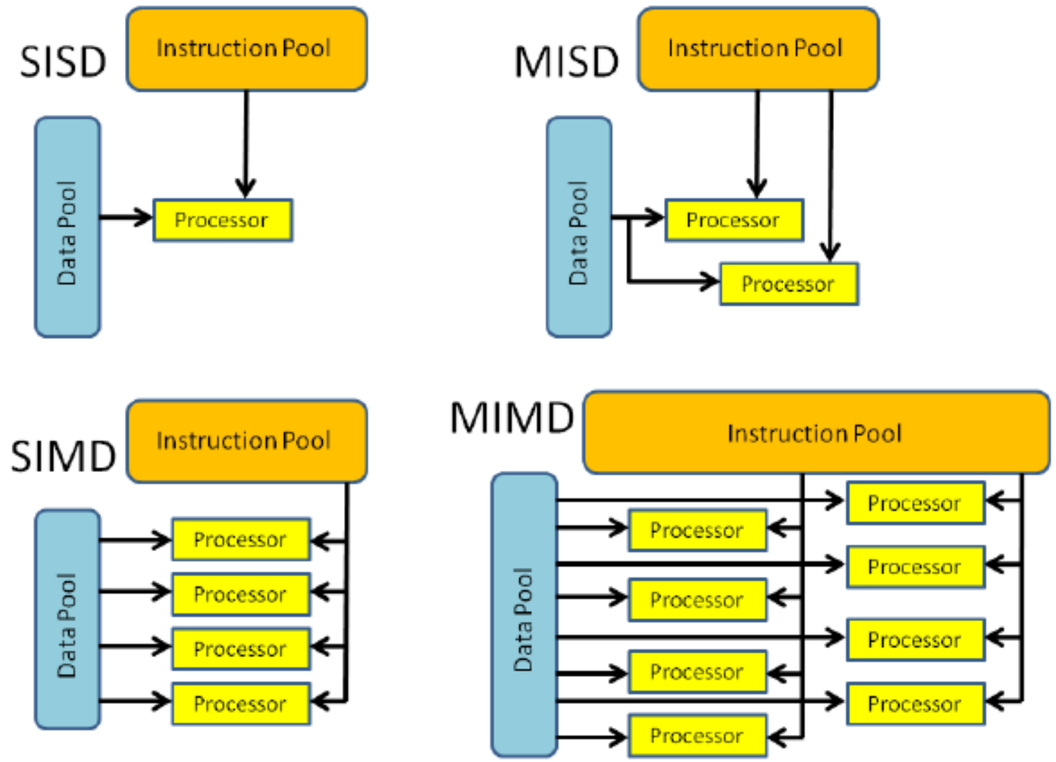
Introduction to High Performance Computing

Part 1: Lecture 1

Taxonomy of parallel computing paradigms

- **The Flynn taxonomy of parallel architecture describe the amount of concurrent control and data streams present**
- **The dominating concepts today are the SIMD and MIMD variants**
- **The shared- memory and distributed-memory parallel computers are typical examples for the MIMD paradigm.**

S I S D Single Instruction stream Single Data stream	**S I M D** Single Instruction stream Multiple Data stream
M I S D Multiple Instruction stream Single Data stream	**M I M D** Multiple Instruction stream Multiple Data stream



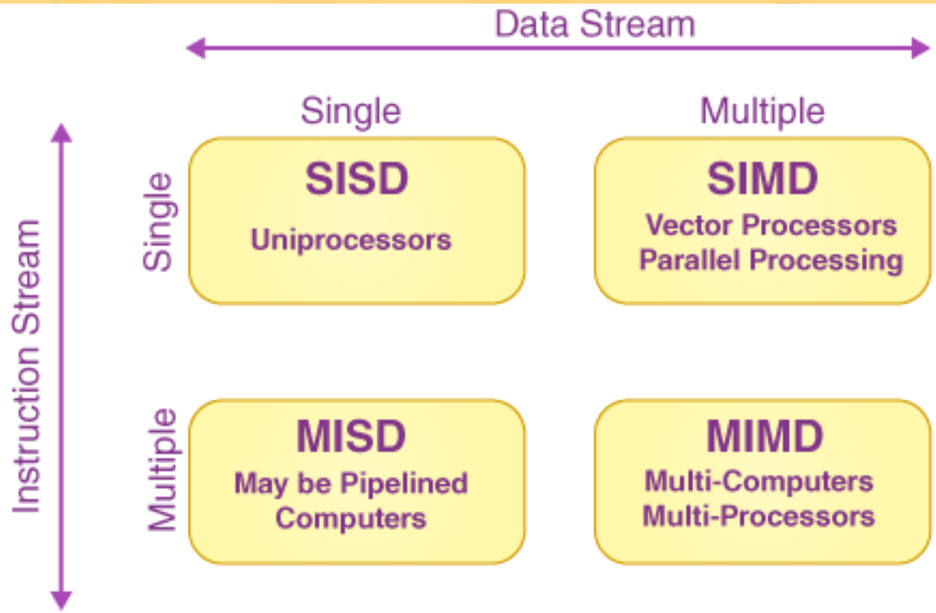
Introduction to High Performance Computing

Part 1: Lecture 1

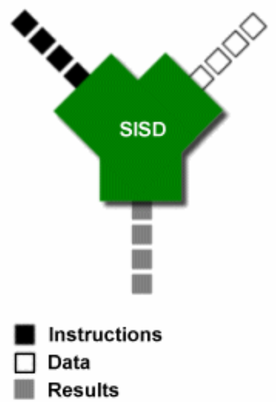
Single Instruction Single Data (SISD)

- **Single Instruction, Single Data (SISD) is a computer architecture where a central processing unit (CPU) executes one instruction at a time on a single data piece.**
- **Deterministic execution**
- **In SISD architecture, instructions are decoded by the Control Unit and then sent to the processing units for execution.**
- **Data streams flow bidirectionally between processors and memory.**

Examples of SISD systems include older generation computers, microcontrollers, and workstations



Flynn's Classification of Computers

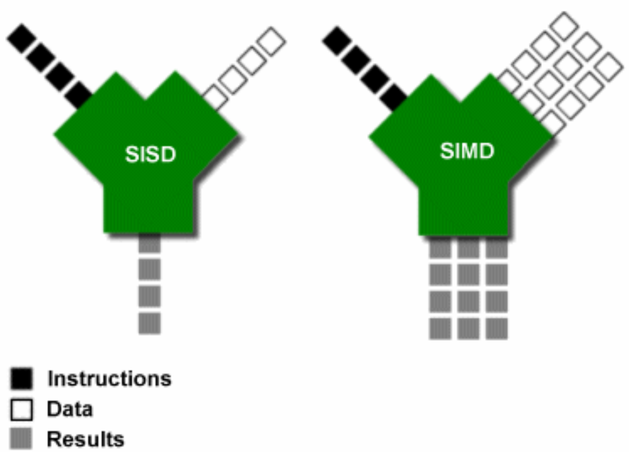


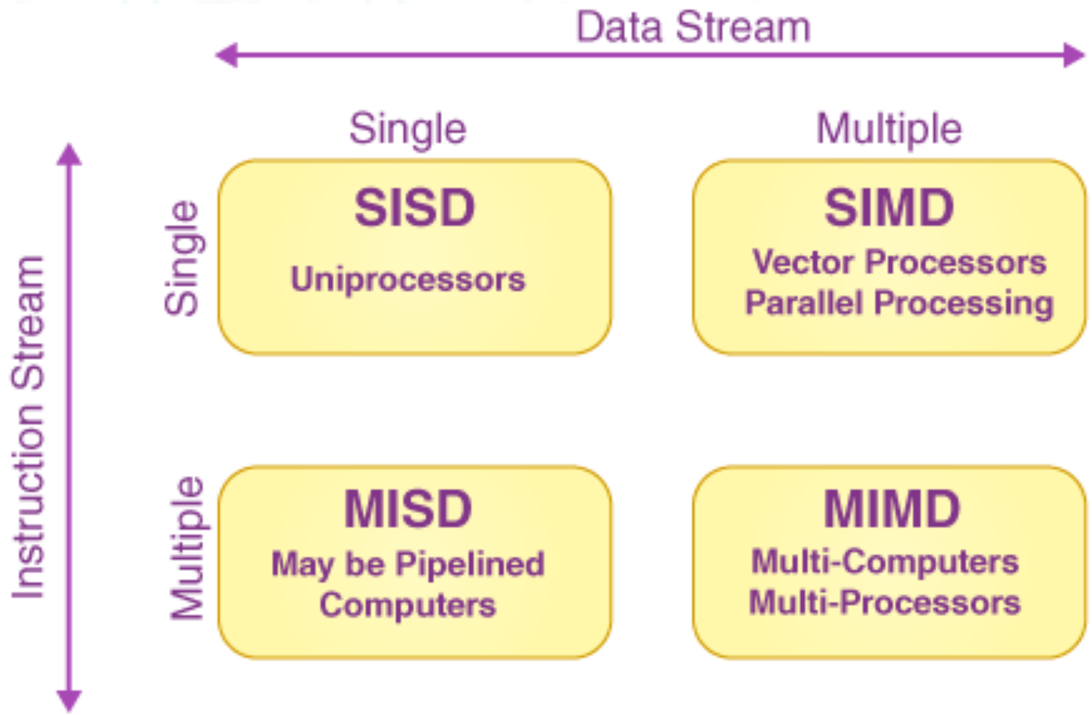
Introduction to High Performance Computing

Part 1: Lecture 1

Single Instruction Multiple Data (SIMD)

- A type of parallel computer that exploits multiple data sets against a single instruction
- 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





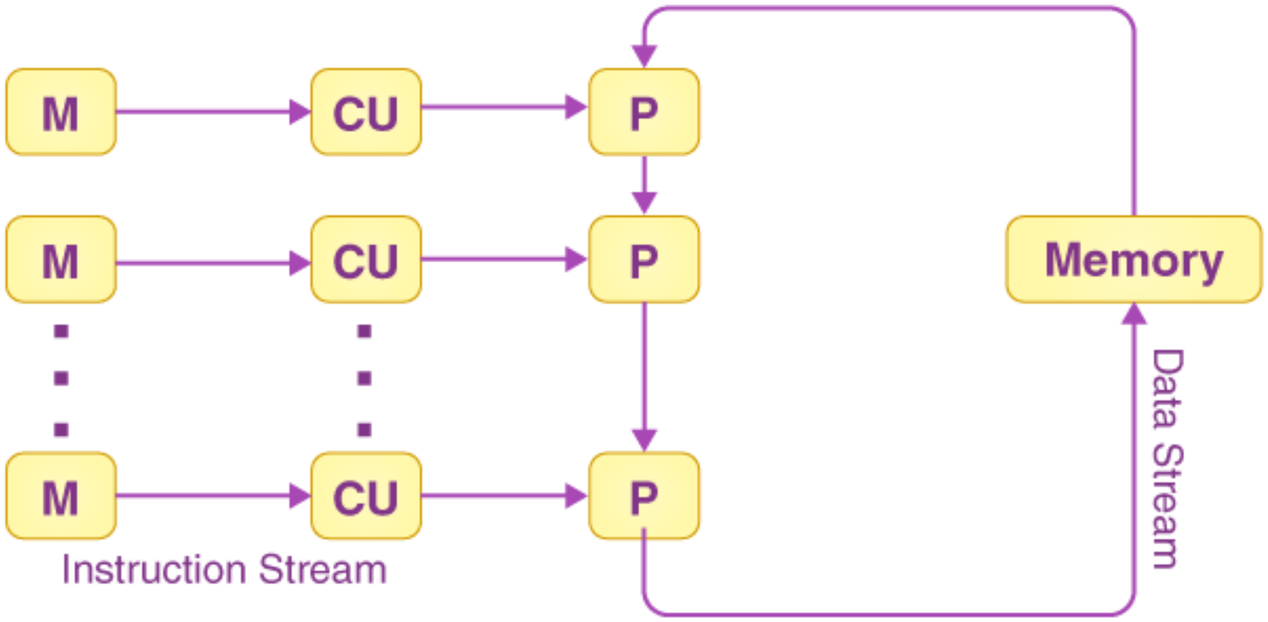
Flynn's Classification of Computers

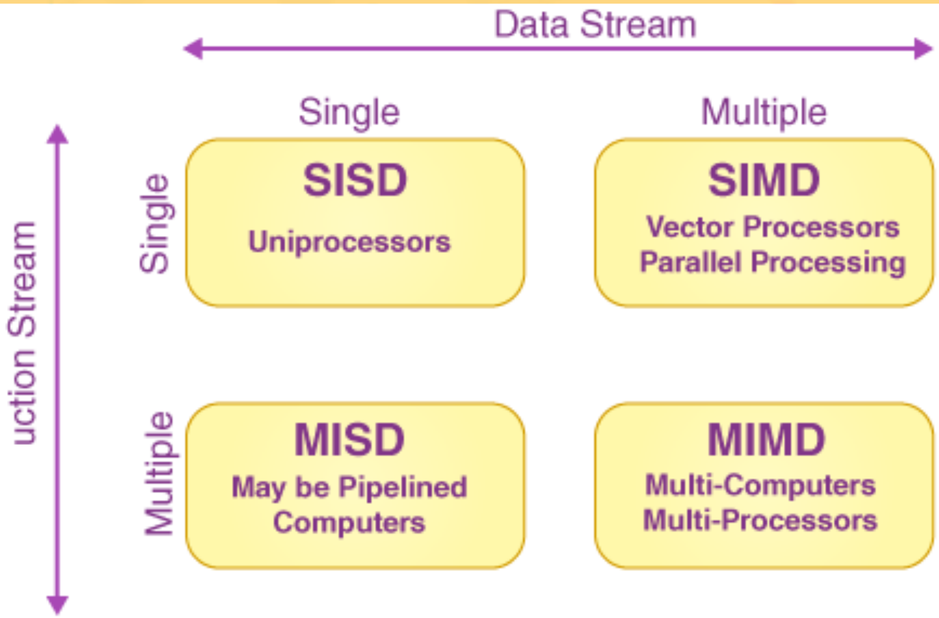
Introduction to High Performance Computing

Part 1: Lecture 1

Multiple Instruction Single Data (MISD)

- **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.**





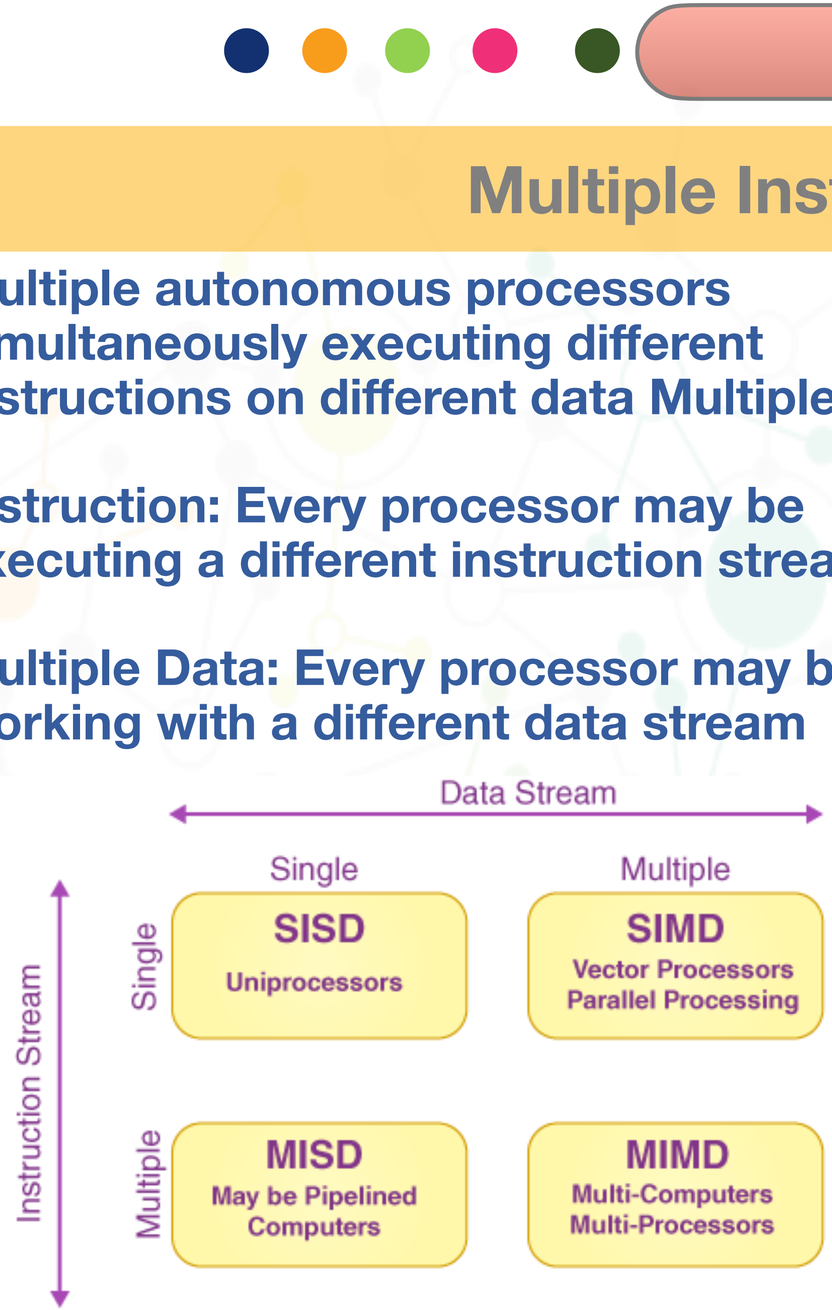
Flynn's Classification of Computers

Introduction to High Performance Computing

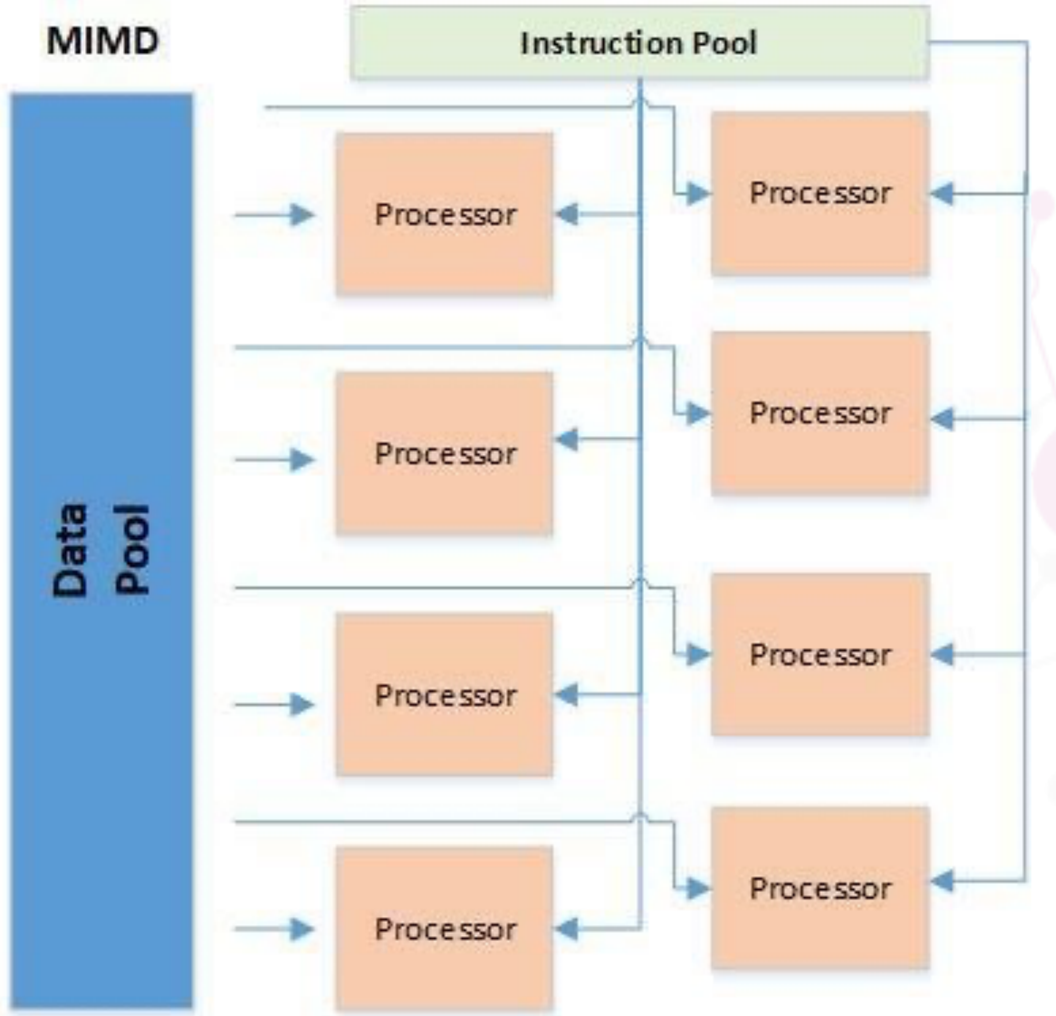
Part 1: Lecture 1

Multiple Instruction Multiple Data (MIMD)

- **Multiple autonomous processors simultaneously executing different instructions on different data Multiple**
- **Instruction: Every processor may be executing a different instruction stream**
- **Multiple Data: Every processor may be working with a different data stream**



Flynn's Classification of Computers

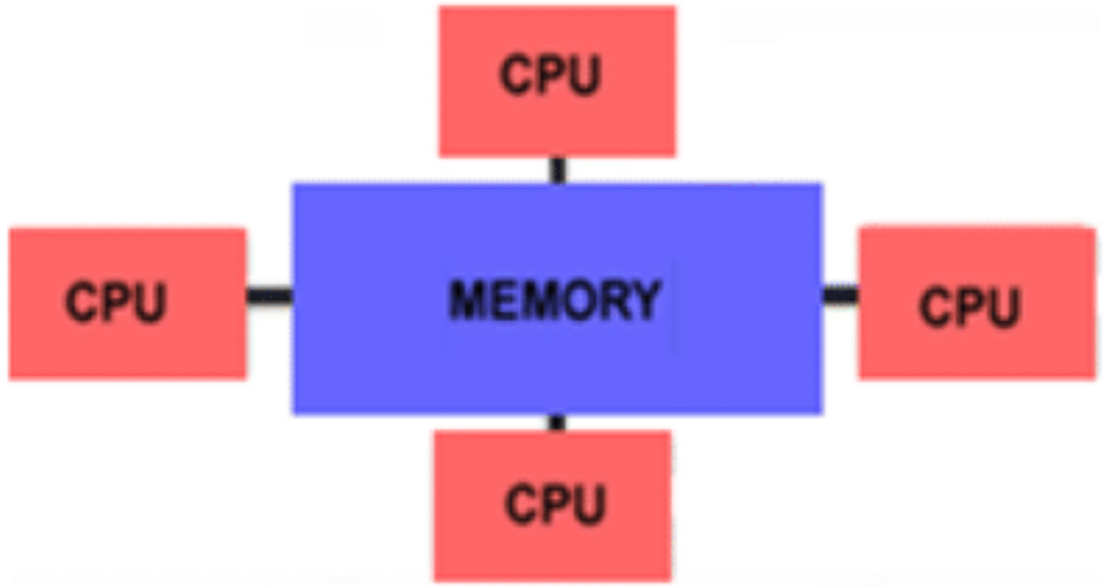


Introduction to High Performance Computing

Part 1: Lecture 1

Parallel Computer Memory Architectures

- Shared memory - all processors access all memory as global address space.
- Multiple processors can operate independently but share the same memory resources.
- Changes in a memory location effected by one processor are visible to all other processors.
- Shared memory machines can be divided into two main classes based upon memory access times: UMA and NUMA



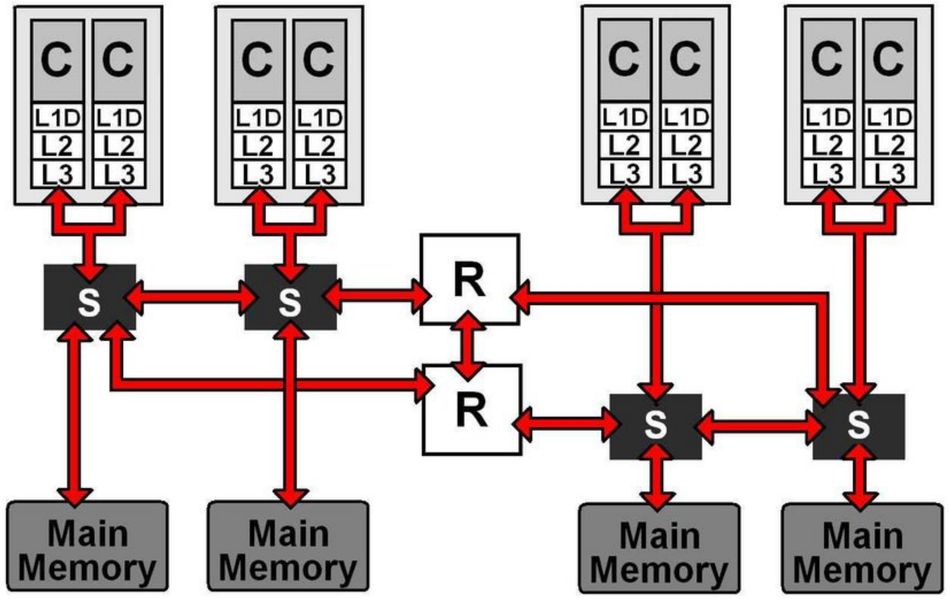
shared memory

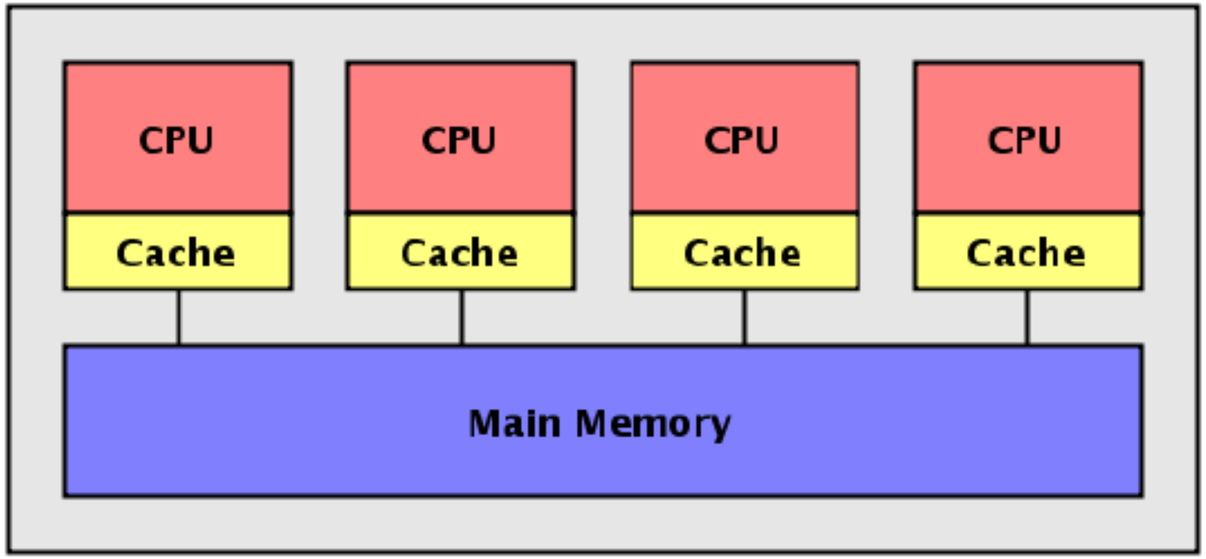
Introduction to High Performance Computing

Part 1: Lecture 1

Shared Memory Computers

- **Uniform Memory Access (UMA) : Latency and bandwidth are the same for all processors and all memory locations. This is also called *symmetric multiprocessing* (SMP).**
- **Cache Coherent Non-Uniform Memory Access(NUMA: memory is physically distributed but logically shared**

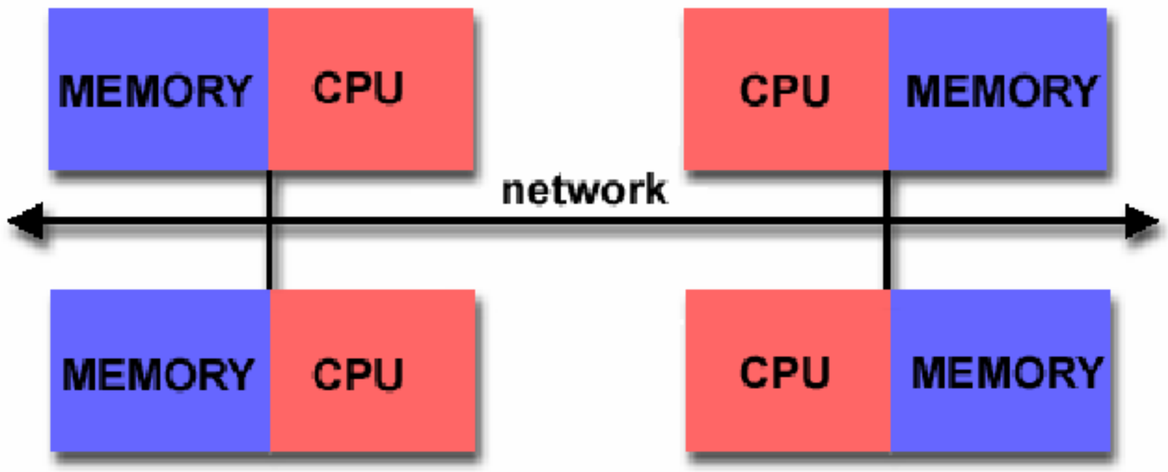


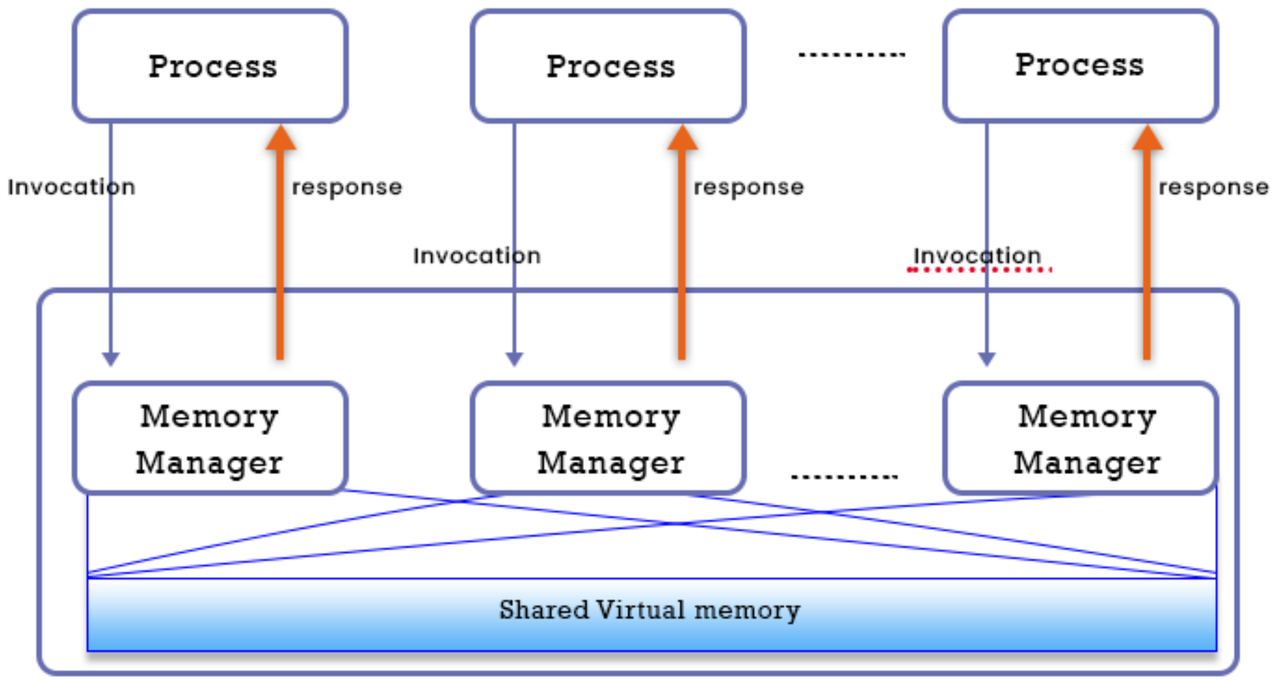


Introduction to High Performance Computing

Part 1: Lecture 1

- **In Distributed Memory High-Performance Computing (HPC), each processor has its own local memory, requiring message passing to exchange data between processors**





Distributed shared memory

THANK YOU!