Presentation of parallel distributing computing

Topics Name:

Memory Consistency Models And Memory Hierarchy

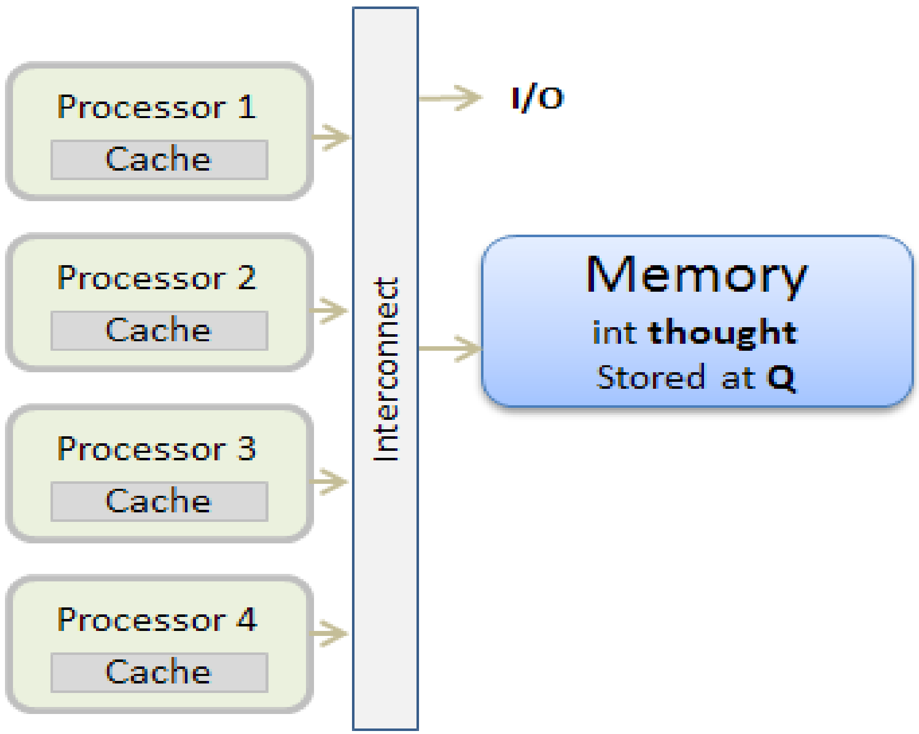
Groups Members Names And Roll Numbers

Names	Roll numbers
Eman Fatima	2123110084
Aqsa noor	2123110064
Meerab shahbaz	2123110101
Fatima sarfraz	2123110076

Topics names

- Consistency vs. Coherence
- Memory Operation Ordering
- Memory Consistency Models
- Memory Hierarchy
- Characteristics of Memory Hierarchy Design

Consistency vs. Coherence



Memory Consistency

- Defines the order in which memory operations (from any process) appear to execute with respect to one another.
- Describes the behavior of reads and writes in relation to other locations.
- Defining guarantees on when loads and stores will happen and when they will be seen by the different cores
- It handles the ordering of reads and writes to all memory locations
- Concerns with **When** a written value will be returned by a read.

Memory Coherence

- Refers to the consistency of shared resource data that is stored in multiple local caches.
- Describes the behavior of reads and writes to the same memory location.
- Coherence is the guarantee that caches will never affect the observable functionality of a program
- It is concerned with the ordering of writes to a single memory location.
- Concerns with **What** values can be returned by a read?

Memory Operation Ordering

- A program defines a sequence of loads and stores

Four types of memory operation orderings:

- **W→R:** write to X must commit before subsequent read from Y
- **R→R:** read from X must commit before subsequent read from Y
- **R→W:** read to X must commit before subsequent write to Y
- **W→W:** write to X must commit before subsequent write to Y

W→R: “write must commit before subsequent read” means: When a write comes before a read in program order, the write must commit (its results are visible) by the time the read occurs.

Memory Consistency Models

Definition of memory consistency model :

A consistency model defines how consistent the shared memory data should be in a system. It acts as an agreement between a distributed data store and the processes using it. The processes follow certain rules, and in return, the system ensures that data behaves correctly.

Types of Memory Consistency Models:

- **Data-Centric Consistency Model** : Focuses on how data is stored and accessed across multiple nodes in a system.
- **Client-Centric Consistency Model** :Focuses on the experience of individual clients accessing data, ensuring consistency from their perspective.

Data-Centric Consistency Model

- **Data-centric consistency models:**

Focus on ensuring that data remains consistent across multiple processors in a distributed or parallel system.

1. Strict Consistency Model : The strictest model is the Strict Consistency Model, which ensures that any read operation always returns the most recent write. This means all processors see memory updates simultaneously, maintaining a global order of execution. However, implementing strict consistency in distributed systems is challenging due to network delays and performance limitations.

2. Sequential Consistency Model

2. **Sequential Consistency Model :**The Sequential Consistency Model relaxes strict consistency by allowing memory operations to appear in a fixed order for all processors. While the order of execution is maintained, it does not necessarily follow real-time constraints.

3. Causal Consistency Model : The Causal Consistency Model maintains the order of related memory operations, ensuring that if one operation influences another, their sequence remains unchanged. However, unrelated operations may be observed in different orders by different processors, making this model more flexible while still preserving logical relationships.

4. FIFO (First In, First Out) Consistency Model

4. FIFO (First In, First Out) Consistency Model :

The FIFO (First In, First Out) Consistency Model guarantees that writes from a single processor appear in the same order to all other processors. However, updates from different processors may be observed in varying sequences, which means consistency is maintained only for individual data streams, not across multiple sources.

5. Weak Consistency Model :

What it means: After updating data, not all users will see the change right away. It takes some time for updates to spread across the system.

Example: Imagine you post a new status on social media. Some friends see the new post instantly, while others still see the old one for a little while before the update reaches them.

6. Release Consistency Model

6. Release Consistency Model

What it means: The system uses locks to control updates. Before reading or changing shared data, a process must lock it. After making changes, it unlocks the data so others can see the update.

Example: In a document editing app, if one person is editing a paragraph, others cannot edit it at the same time. Once the editor finishes and unlocks it, others can see the changes and make their own edits.

7. Entry Consistency Model

What it means: Instead of locking the whole system, this model locks only the specific data being modified. Each piece of data has its own lock, making updates more efficient.

Example: In a shared spreadsheet, if you want to edit one cell, you lock only that cell, not the entire sheet. Other people can still work on different cells without waiting for you to finish.

Client-Centric Consistency Model

- **Client-Centric Consistency Model**

  What it means: The system makes sure that once a user sees an update, they won't see an older version later. It focuses on keeping data consistent for **each user** rather than for the entire system at once.

  Example: If you mark an email as "read" on your phone, it will still be "read" when you open it later on your laptop, even if the devices connect to different servers.

Memory Hierarchy

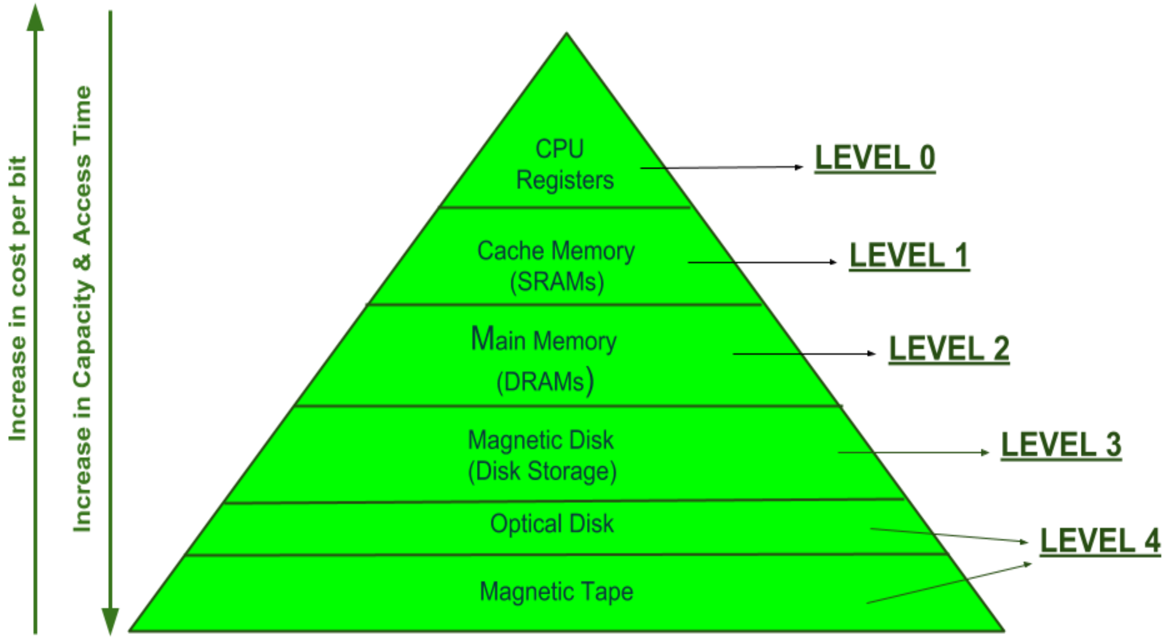
Definition of memory hierarchy :

- Memory hierarchy is the organized structure of different types of memory in a computer system, arranged by speed, cost, and capacity to optimize performance and minimize access time.

The Memory Hierarchy was developed based on a program behavior known as locality of references.

Locality of reference, also known as the principle of locality, is the tendency of a processor to access the same set of memory locations repetitively over a short period of time.

Memory Hierarchy



MEMORY HIERARCHY DESIGN

Memory Hierarchy

Memory Hierarchy Design is divided into 2 main types:

1. External Memory or Secondary Memory

- Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.

2. Internal Memory or Primary Memory

- Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

Characteristics of Memory Hierarchy Design

Capacity :It is the total amount of data a memory can store. As we go down the hierarchy, storage **increases**.

Access Time : It is the time taken to read or write data. As we go down, memory becomes slower.

Performance : Earlier, without memory hierarchy, the speed difference between CPU registers and main memory was too high, making the system slow. Memory hierarchy improves speed by keeping frequently used data in faster memory.

Cost per Bit : As we go up, memory becomes more expensive. Fast memory (like cache) costs more than slow memory (like hard disk).