BIG DATA MANAGEMENT

CZ4123

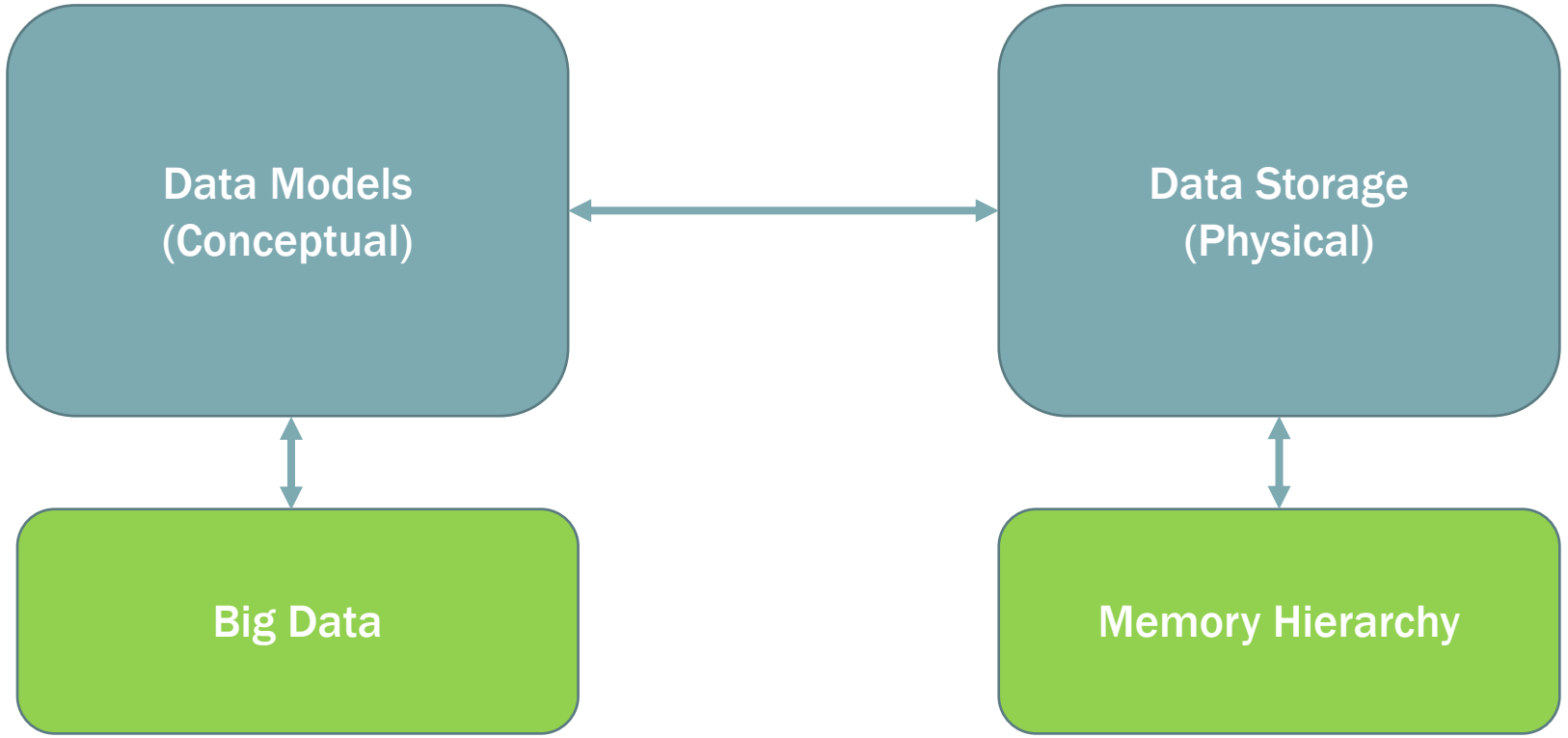
MEMORY HIERARCHY (PART I)

Siqiang Luo

Assistant Professor

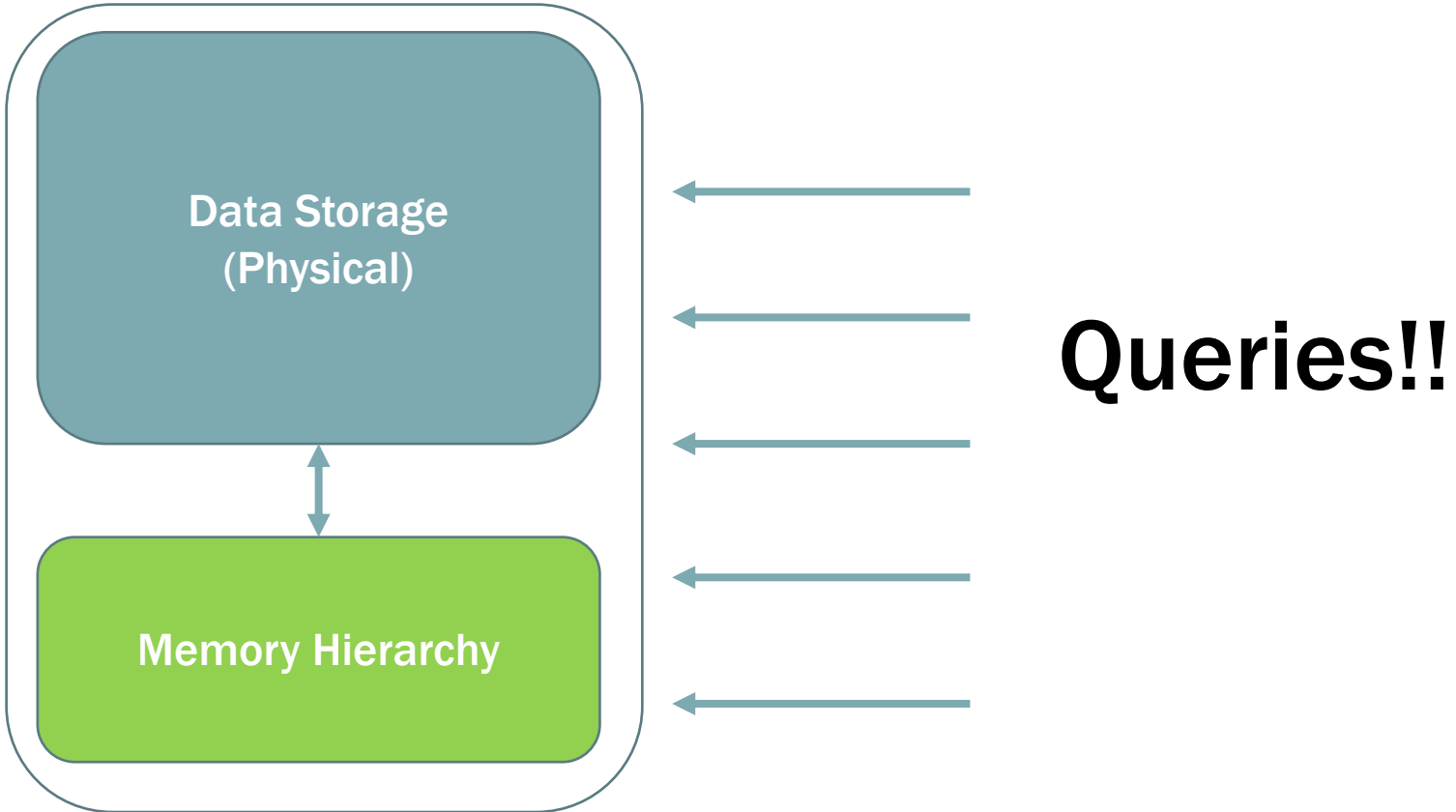
PREPARATION

- In previous lectures, we have learnt (conceptual) data models. These data models need to be (physically) stored in the storage medium (e.g., disks).



PREPARATION

- "How to store the data" must be related to "how to retrieve/query them efficiently".



PREPARATION

What is an ideal case for storing big data?

PREPARATION

What is an ideal case for storing big data?

- Ideally, we should have infinite size of fast accessing storage, and they are persistent.
  - Infinite size to store big data.
  - Fast accessing guarantees fast read/write of the data.
  - Persistency ensures keeping the data when we power off the system.

PREPARATION

However, the ideal case is not easy to realize

PREPARATION

The dilemma of storage design

- Fast storage with a large size is expensive; we may afford
  - Faster storage with a smaller size
  - Slower storage with a larger size

PREPARATION

The dilemma of storage design

- Fast storage with large size is expensive; we may afford
  - Faster storage with smaller size
  - Slower storage with larger size

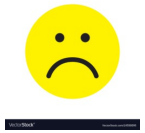
- Suppose you are given 800SGD, how do you allocate your budget to buy different types of storage?
  - Cache: $20/MB
  - Main Memory: $20/GB
  - Disk: $20/TB

PREPARATION

If all the budget is for disk

large storage (40TB)

Slow access

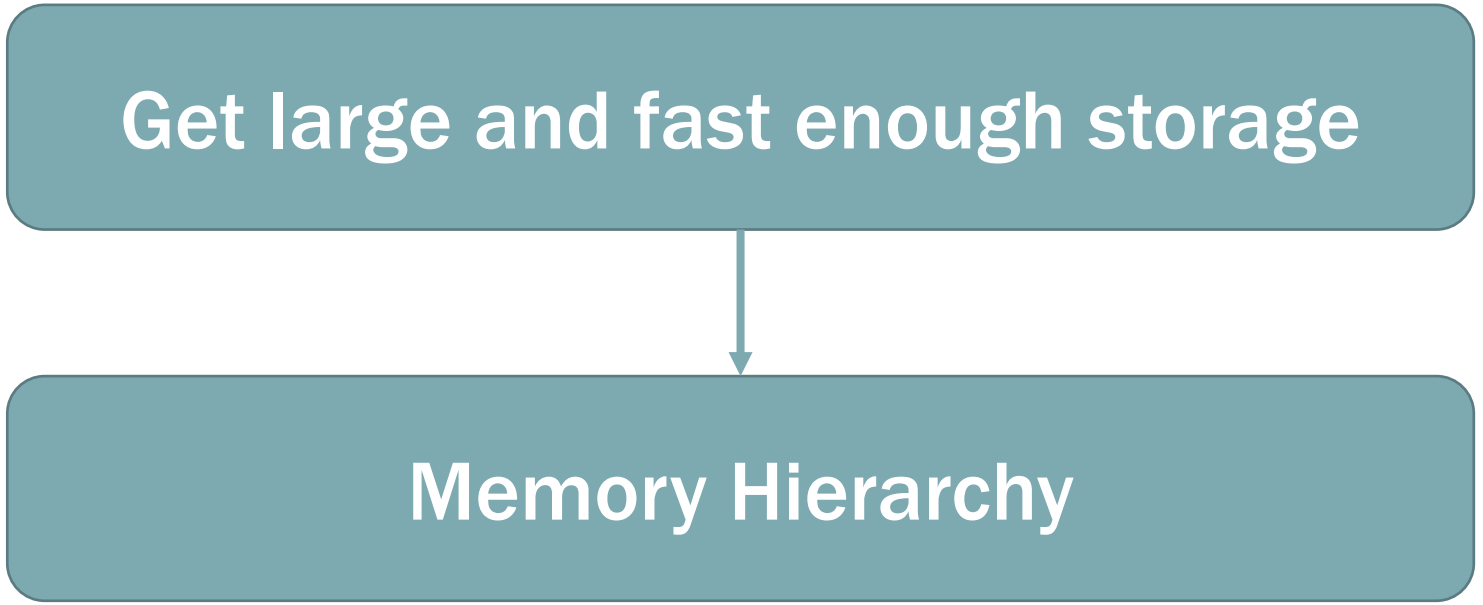


If all the budget is for cache

small storage (40MB)

Fast access

PREPARATION



We will introduce the concept more formally

OUTLINE

- ❑ What is memory hierarchy?
- ❑ Why we need memory hierarchy?

OUTLINE

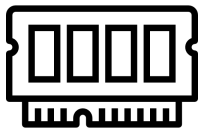
- What is memory hierarchy?
- Why we need memory hierarchy?

CONCEPTS

- The storage space in the computer is used to store data and instructions
  - Instructions are the “codes” that tell the processor what to do
- The storage space is divided into multiple cells, each having an address
  - Most modern computers are byte-addressable.
  - Each address identifies a single byte of storage.
  - For example, 256GB memory has 256*1024*1024*1024 addresses

THREE MAIN CATEGORY OF STORAGE

Cache Memory

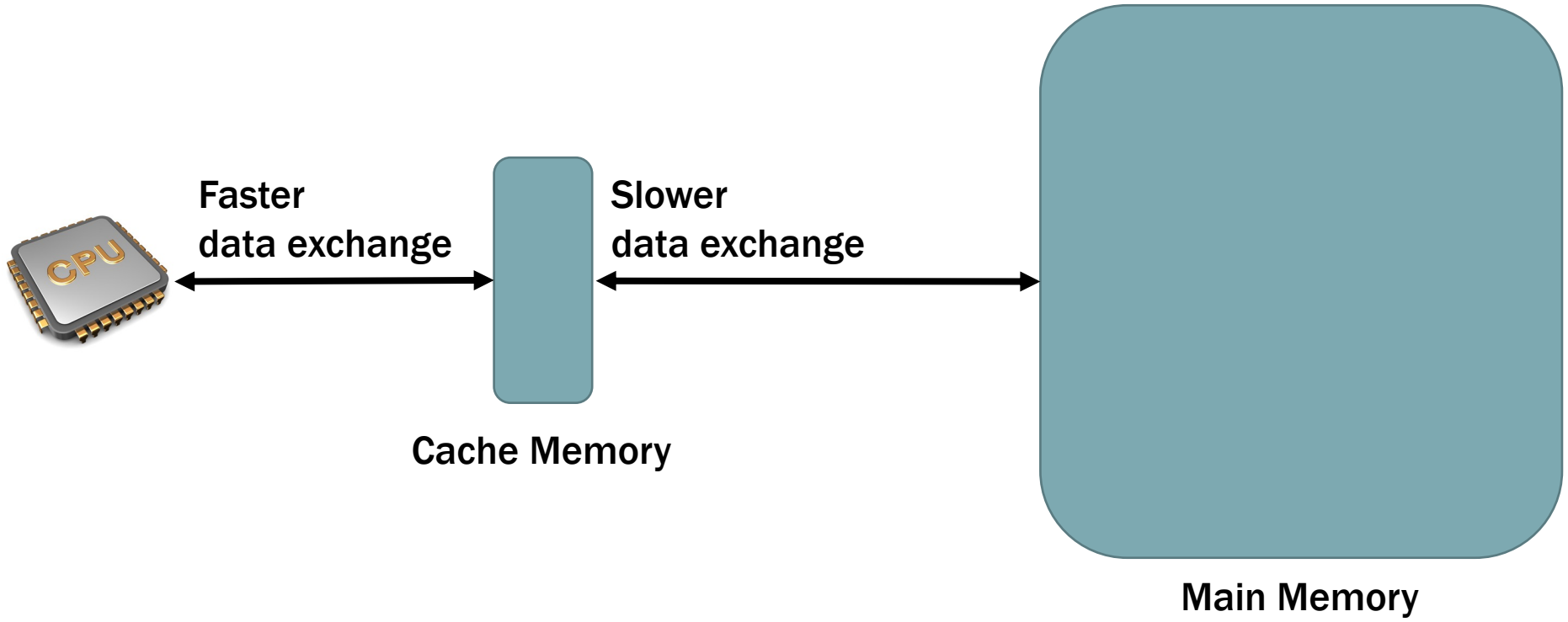


Main Memory

Secondary Memory

CACHE-MEMORY

- High speed
- Small capacity (often between 8 KB and 64 KB)
- Expensive
- Regarded as a buffer between the CPU and the slower main memory
- Hold data and program instructions which are most frequently used by the CPU


CPU
Faster
data exchange
Slower
data exchange
Cache Memory
Main Memory

MAIN-MEMORY

- ❑ Holds data and instructions on which the computer is currently working
- ❑ Relatively high speed but slower than cache-memory
- ❑ Data is lost if power-offed
- ❑ Much larger capacity than cache memory (often between 2 GB and 32 GB)
- ❑ Less expensive than cache memory

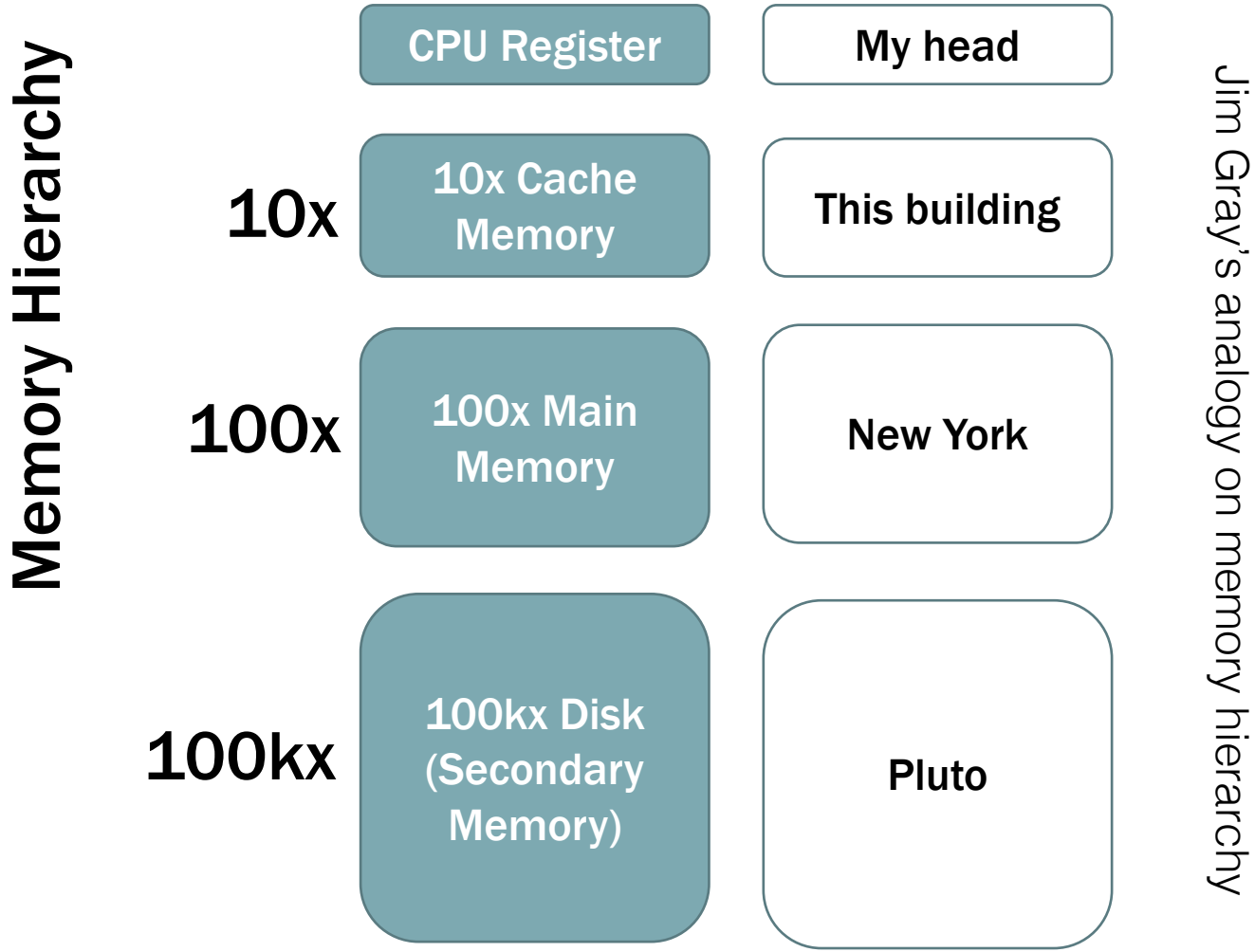
SECONDARY-MEMORY

- Also known as external memory
  - in many situations, we may simply use the most common example “disk”.
- Much slower than main memory when accessing data
  - Even with SSD (Solid State Disk), the accessing cost is still higher
- Can store data permanently
- Example: Hard Disk, SSD

MEMORY HIERARCHY

People often simplify

- ❑ Cache memory→ Cache
- ❑ Main Memory→ Memory
- ❑ Secondary Memory → Disk



Note for Register: small amounts of high-speed memory contained within the CPU

MORE COMPLICATED MEMORY HIERARCHY

- In fact, cache can be further divided into L1 cache, L2 cache, L3 cache.
  - L1 cache is the fastest but smallest
  - L3 cache is the slowest but largest
  - L2 cache is in the middle

- For simplicity, in this course we simply merge them into a simple cache layer.

OUTLINE

- ❑ What is memory hierarchy?
- ❑ Why we need memory hierarchy?

WHY WE NEED MEMORY HIERARCHY

- Suppose we need 256GB storage for a computer, can we design a computer with 256GB **cache memory** without main memory and disk/secondary memory?

WHY WE NEED MEMORY HIERARCHY

- Suppose we need 256GB storage for a computer, can we design a computer with 256GB **cache memory** without main memory and disk/secondary memory?

- -- It can be too expensive 

- -- No **permanent** storage 

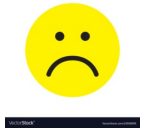
WHY WE NEED MEMORY HIERARCHY

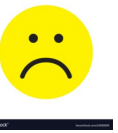
- Suppose we need 256GB storage for a computer, can we design a computer with 256GB **main memory** without cache memory and disk?

WHY WE NEED MEMORY HIERARCHY

- Suppose we need 256GB storage for a computer, can we design a computer with 256GB **main memory** without cache memory and disk?

- -- It can still be relatively expensive 

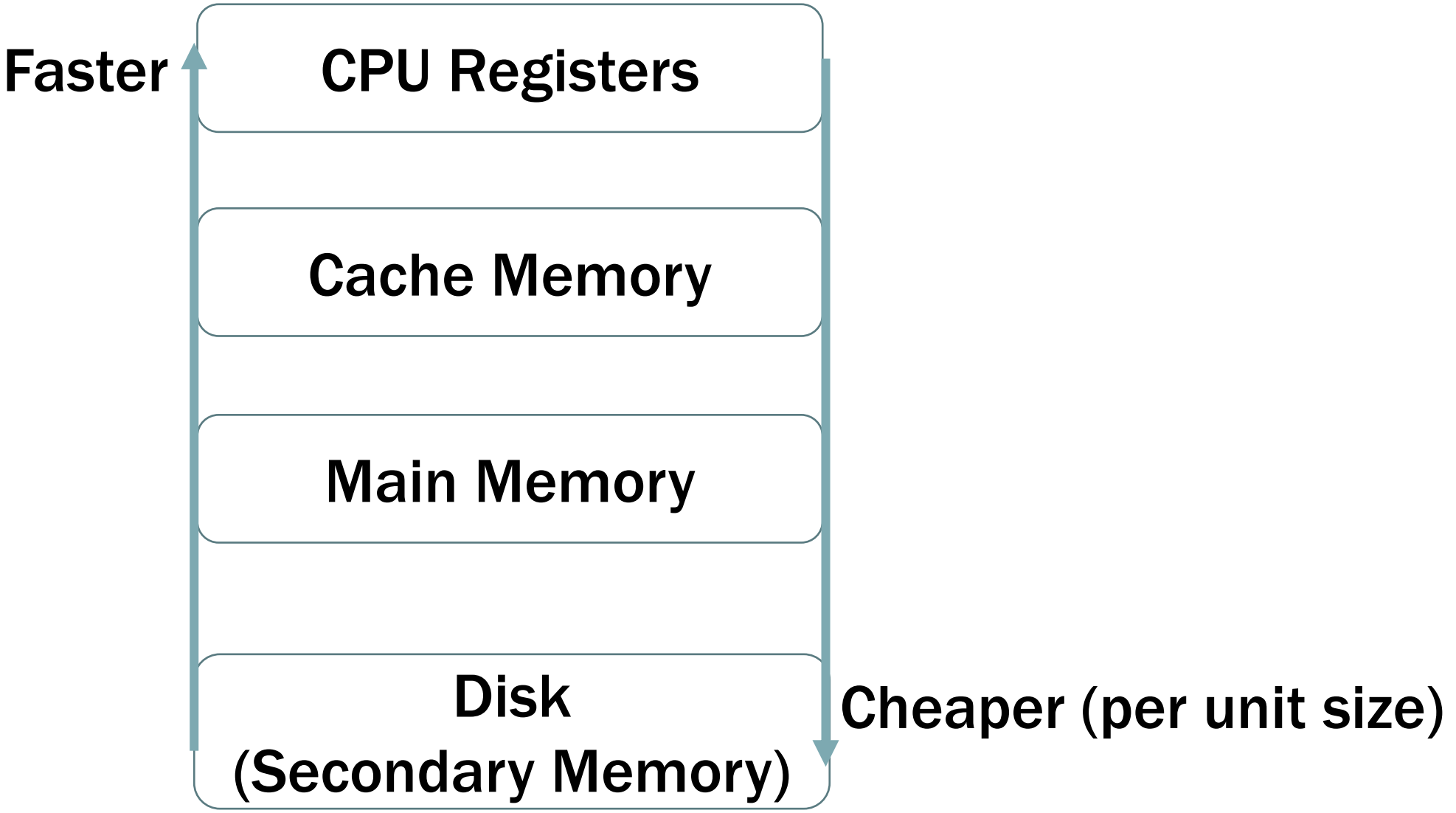
- -- No **permanent** storage 

- -- No buffer between CPU and main memory. The CPU speed can be much faster than main memory, causing latencies. 

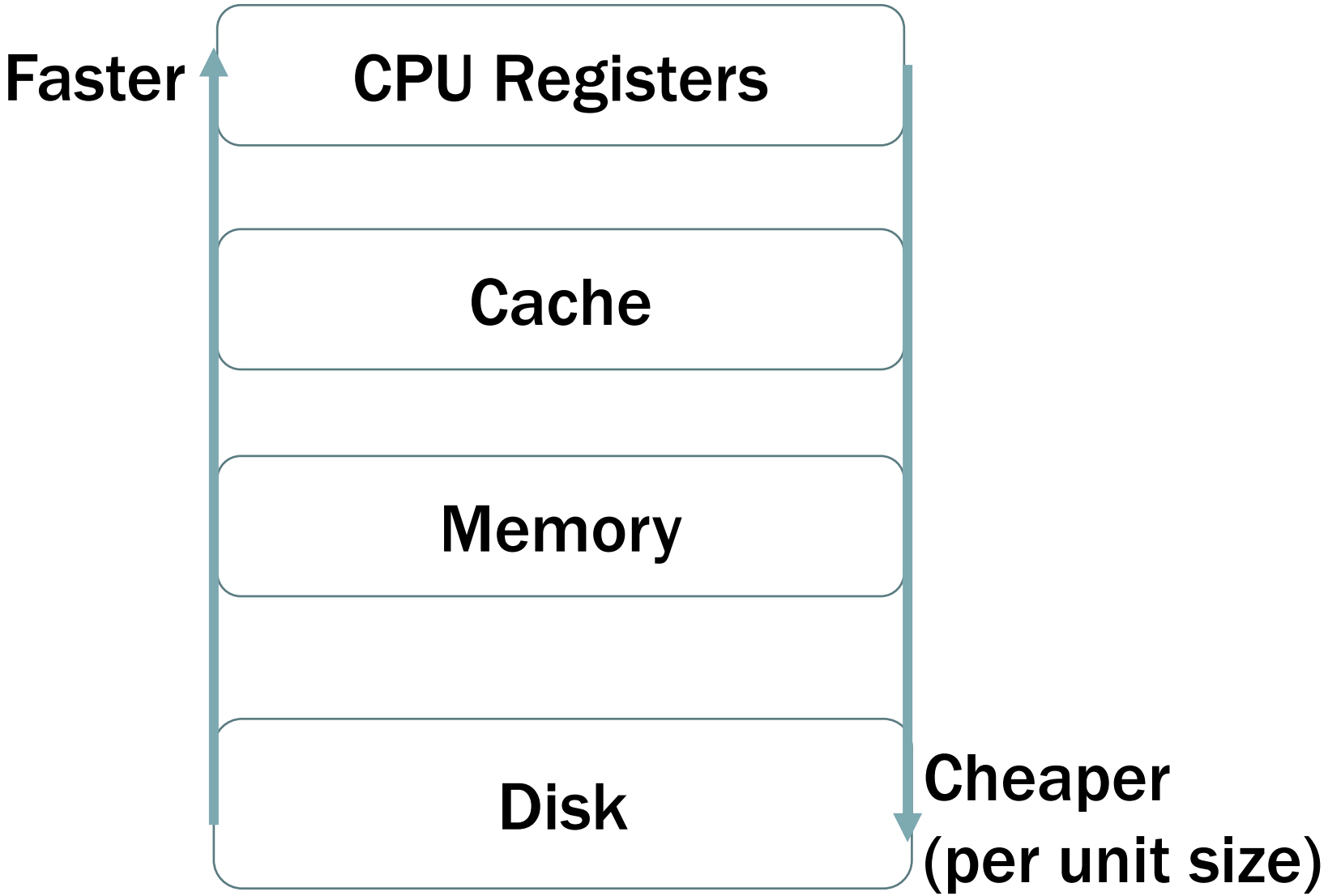
WHY WE NEED MEMORY HIERARCHY

- We need fast memory to sync up CPU speed → Need Cache Memory
  - Processor speed is much faster than main memory access time
- We need large space for permanently storing **big data** → Need Secondary Memory
- Price becomes a concern
  - Faster memory (e.g., cache memory) is much more expensive

TRADE-OFF BETWEEN SPEED AND PRICE



TRADE-OFF BETWEEN SPEED AND PRICE



SUMMARY OF MEMORY HIERARCHY

- Memory hierarchy consists of a set of memory layers, where a faster memory layer has a smaller capacity.

- Memory hierarchy is needed concerning the following factors (most important ones)
  - Reasonable price
  - Enough capacity to hold data
  - Data persistency