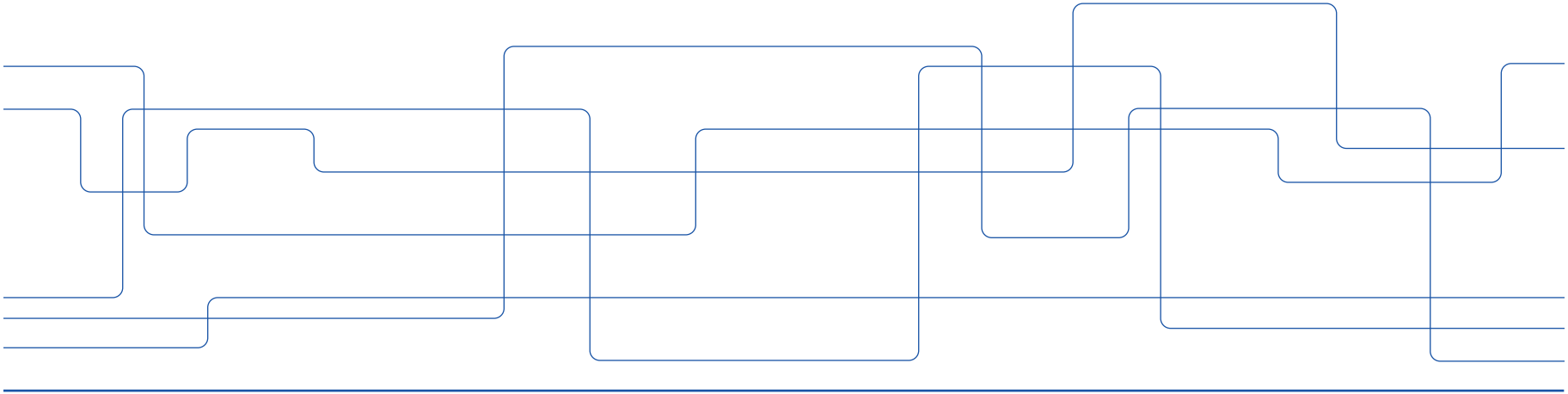
DD2358 – Fundamentals of Computer Systems – Memory Units

Stefano Markidis

KTH Royal Institute of Technology

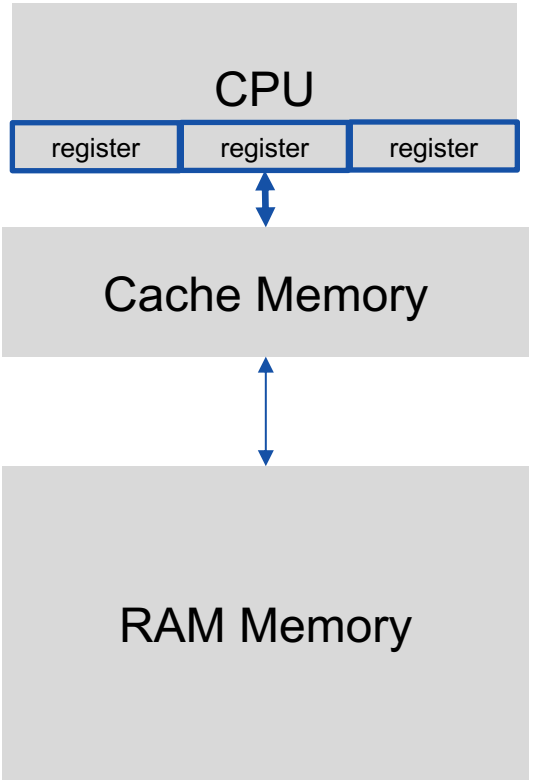


Intended Learning Outcomes

- Describe the parameters characterizing the performance of memories
- List and describe different kinds of memories and their technologies
- Quantify the capacity and speed of different kind of memories
- List different optimization techniques for effective memory usage

Memory Units

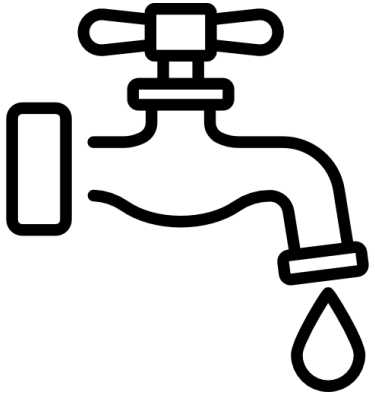
- *Memory units* in computers are used to store bits.
  - These could be bits representing **variables** in your program or bits representing the **pixels of an image**.
- Thus, the abstraction of a memory unit applies to the **registers** in the processor as well as **caches**, **RAM** and **hard drive**.



Memory Performance: Bandwidth and Latency

- The one major difference between all of these types of memory units is the speed at which they can read/write and access data.
- Two important parameters characterize memory performance
  - ***Latency*** is the time it takes the device to find the data that is being used
  - ***Bandwidth*** is amount of data that is read or written per second

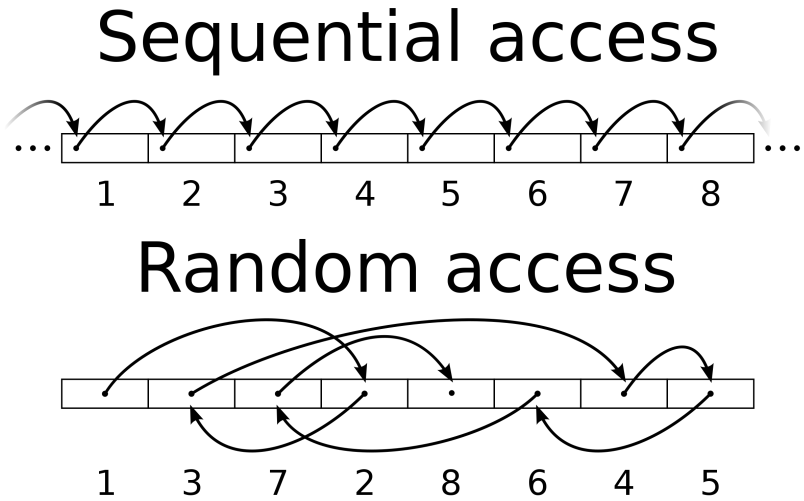
Faucet analogy



Latency: how much time it takes for water to exit
Bandwidth: flow

Memory Performance: Access Patterns

- To make things more complicated, the read/write speed is **heavily dependent on the way that data is being read**.
- Most memory units perform much better when they **read one large chunk of data** as opposed to many small chunks
  - This is referred to as ***sequential read*** versus ***random data***

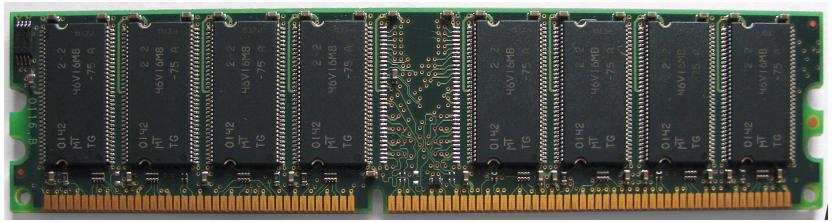


Sequential vs Random Access: the Pages Analogies

- If the data in these memory units is thought of as **pages** in a large book
- Most memory units have better read/write speeds when going through the book page by page rather than constantly flipping from one random page to another.

Kind of Memory I: RAM

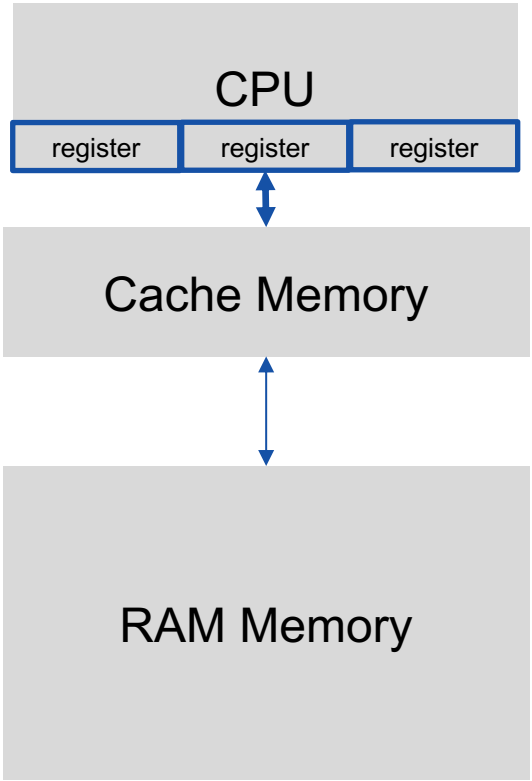
- Used to store application code and data (such as any variables being used).
- Has fast read/write characteristics and performs well with random access patterns
  - It is generally limited in capacity (64 GB range).
- Different technologies: DRAM & HBM



DDR Memory Module

Kind of Memory II: Caches and Registers

- Main memory contains the program data
  - **Cache memory contains a *copy* of the main memory data**
    - > ***Cache is faster*** *but consumes more space and power*
- **Registers contain working data only**
  - Modern CPUs perform most or all operations only on data in register



Kind of Memory (Storage) III: Spinning Hard Drive

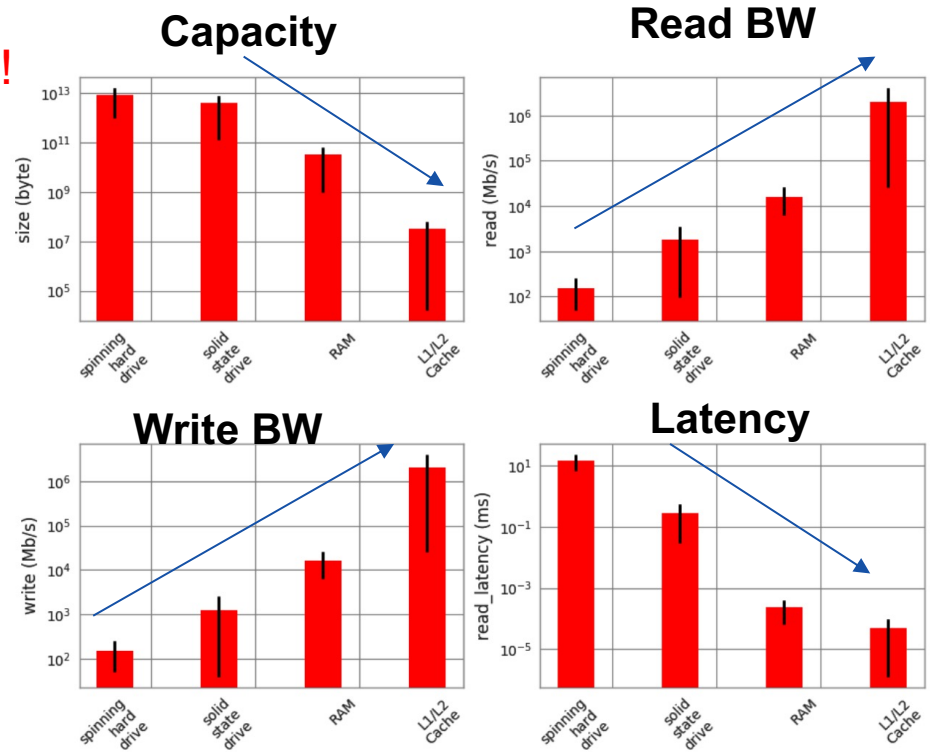
- Long-term storage that persists even when the computer is shut down.
- Generally, it has slow read/write speeds because the disk must be physically spun and moved.
- Degraded performance with random access patterns but very large capacity (10 TB range).
- Mechanical parts involved → failures

Kind of Memory (Storage) IV: Solid State Disks (SSD)

- Similar to a spinning hard drive
- Faster read/write speeds
- Lower life expectancy than HDD
- Smaller capacity (1 Terabyte range).

Characteristics of Different Memories Storage

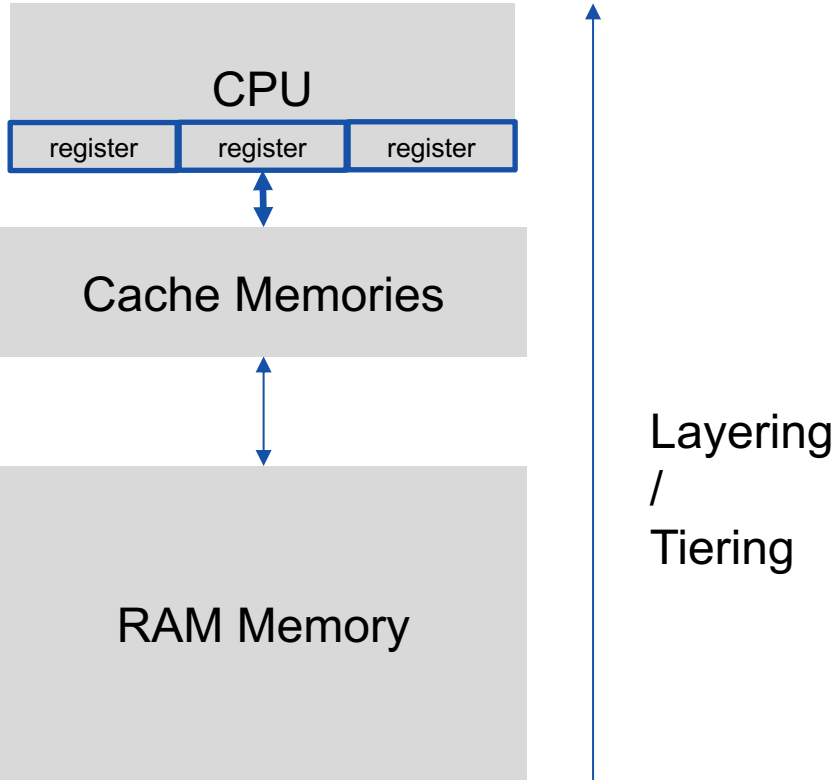
Log scale!



- A clearly visible trend is that read/write speeds and capacity are inversely proportional
  - As we try to increase speed, capacity gets reduced.

Layering Memory Systems for Performance

- All the modern systems implement a tiered approach to memory
  - Data starts in its full state in the hard drive, part of it moves to RAM, and then a much smaller subset moves to the L1/L2 cache.
- This method of tiering enables programs to keep memory in different places depending on access speed requirements.

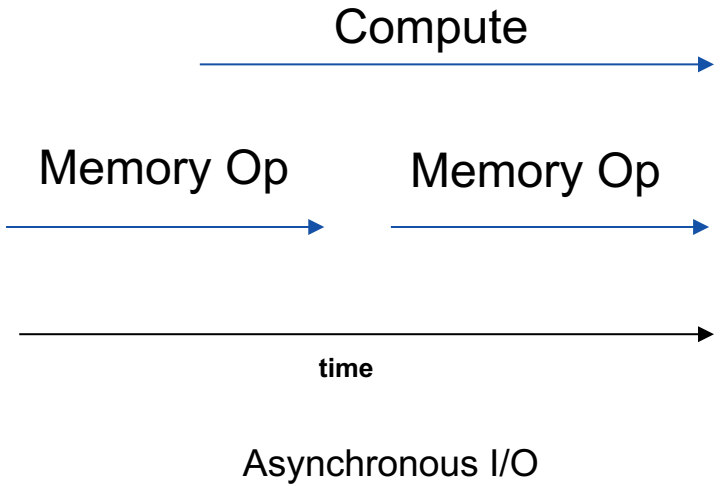


Optimization for Memories I

- When we **optimize the memory patterns of a program**, we optimize:
  - Which data is placed where (**data placement**) → allocators
  - How it is **laid out** (in order to **increase the number of sequential reads**) – blocking/ tiling → data layout
  - How many times it is moved among the various locations → rearrange code for increasing data reuse

Optimization for Memories II

- Methods such as **asynchronous I/O** and **pre-emptive caching** **(**generating cached data versions before they're requested by a user**)** provide ways to make sure that data is always where it needs to be without having to waste computing time
  - Most of these processes can happen independently, while other calculations are being performed



Asynchronous I/O

Summary

- The memory performance is characterized by two numbers: bandwidth and latency. Think about the pipe analogy.
- Memory performance depends on how we access data: read large chunks of data is faster than reading small data many times
- Memory systems include RAM, caches and registers, spinning hard drives, SSD.
- Memory capacity is inversely proportional to bandwidth: small memories are very fast and large memoris are slow
- Optimization for memory include data placement, blocking tiling techniques, rearranging the algorithm, use asynchronous I/O and pre-emptive caches.