UC San Diego

DSC 102
Systems for Scalable Analytics

Rod Albuyeh

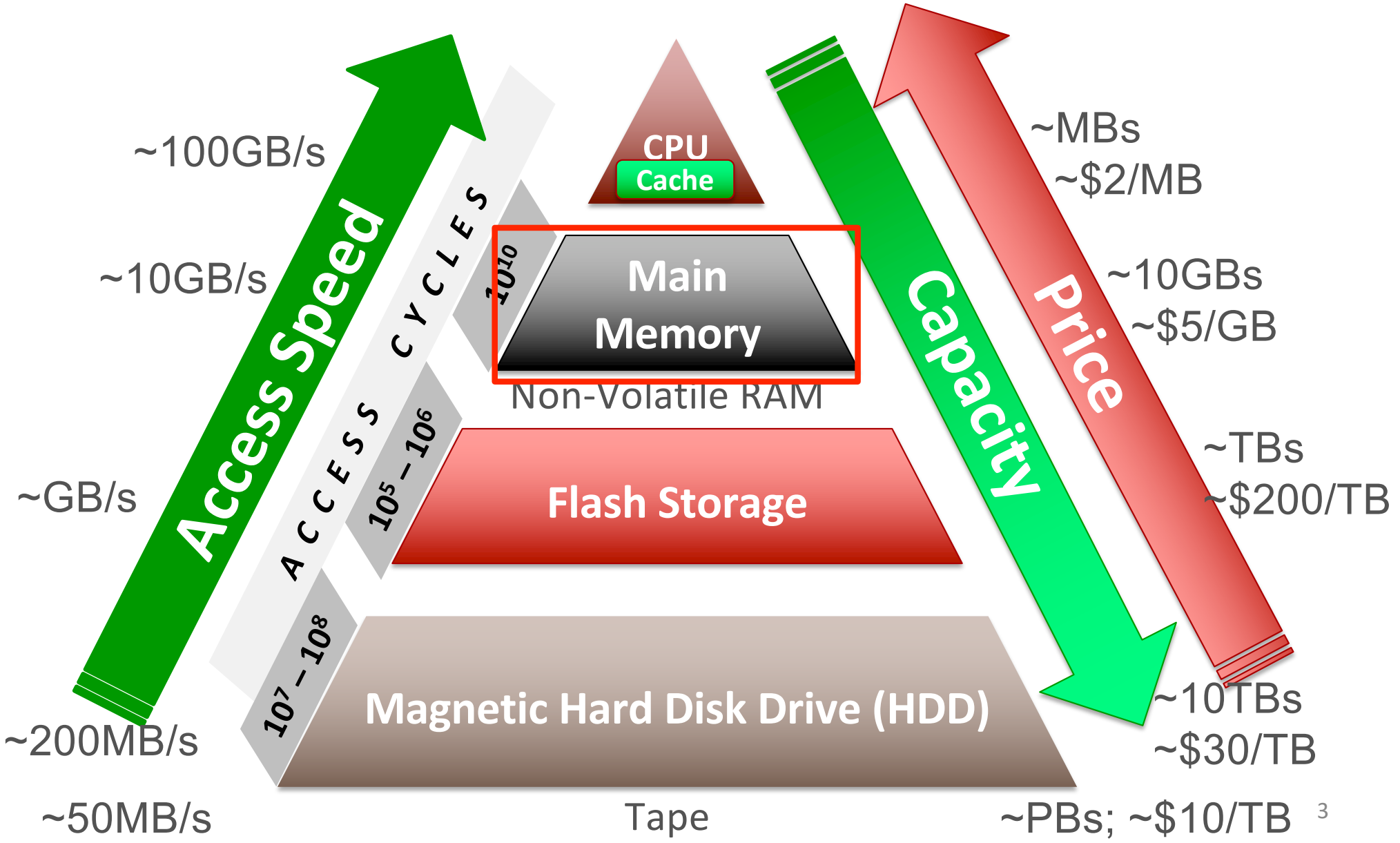
Topic 1: Basics of Machine Resources
Part 2: Operating Systems

Ch.18.1, 21, 22, 26, 36, 37, 39, and 40.1-40.2 of Comet Book

Outline

- Basics of Computer Organization
  - Digital Representation of Data
  - Processors and Memory Hierarchy
- Basics of Operating Systems (OS)
  - Process Management: Virtualization; Concurrency
  - Filesystem and Data Files
  - Main Memory Management
- Persistent Data Storage

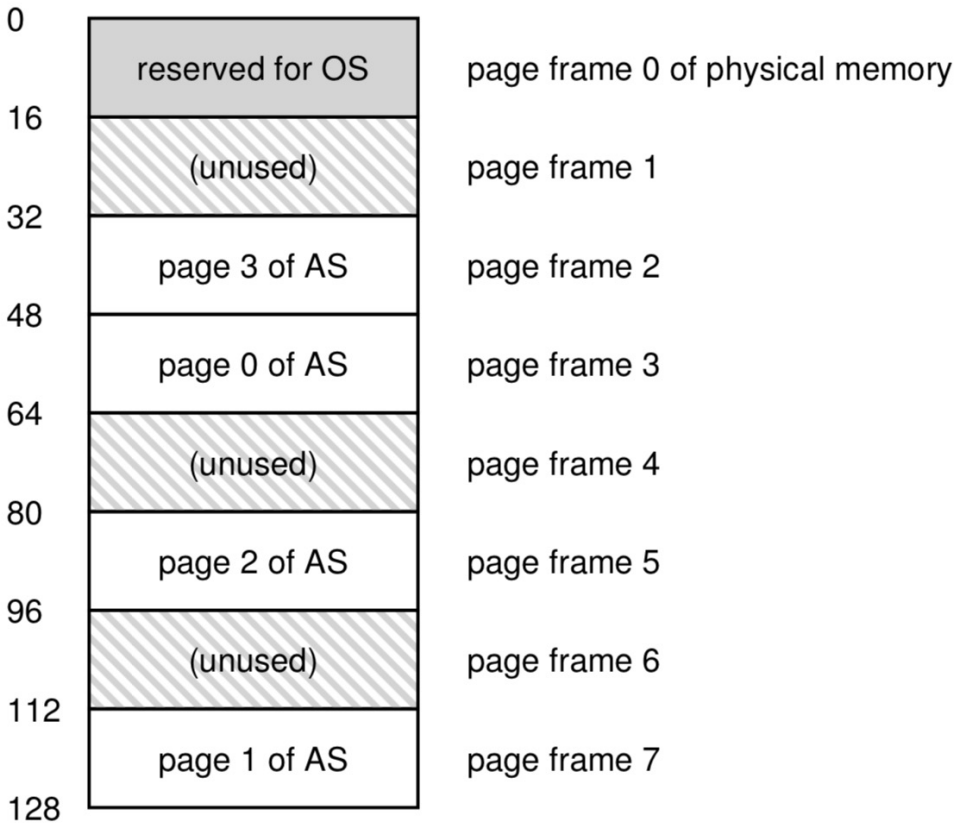
Memory/Storage Hierarchy



Virtualization of DRAM with Pages

- **Page**: An abstraction of *fixed* size chunks of memory/storage
  - Makes it easier to virtualize and manage DRAM
- **Page Frame**: Virtual slot in DRAM to hold a page's content

- Page size is usually an OS configuration parameter
  - E.g., 4KB to 16KB
- OS **Memory Management** has mechanisms to:
  - Identify pages uniquely
  - Read/write page from/to disk when requested by a process



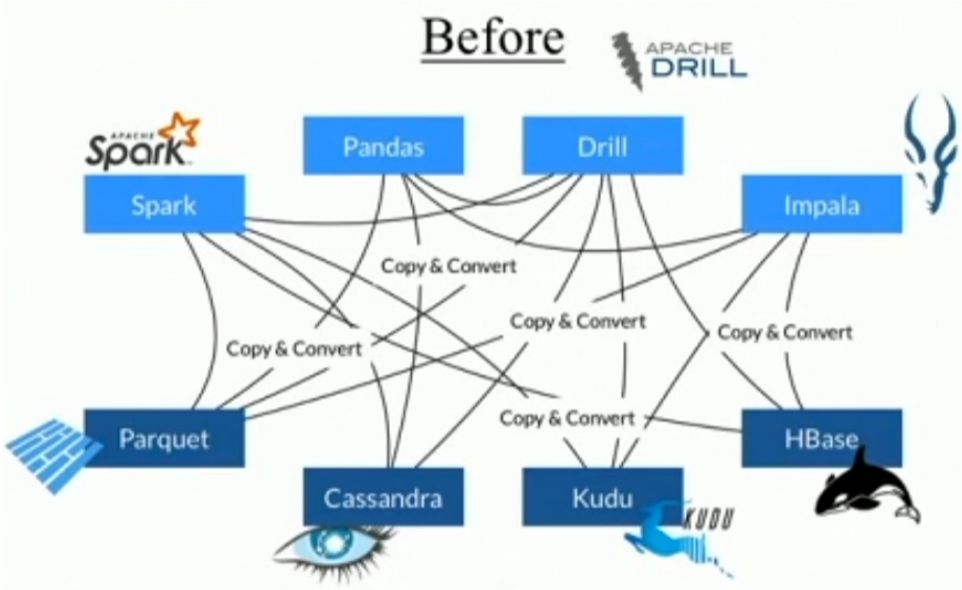
Apportioning of DRAM: Elements

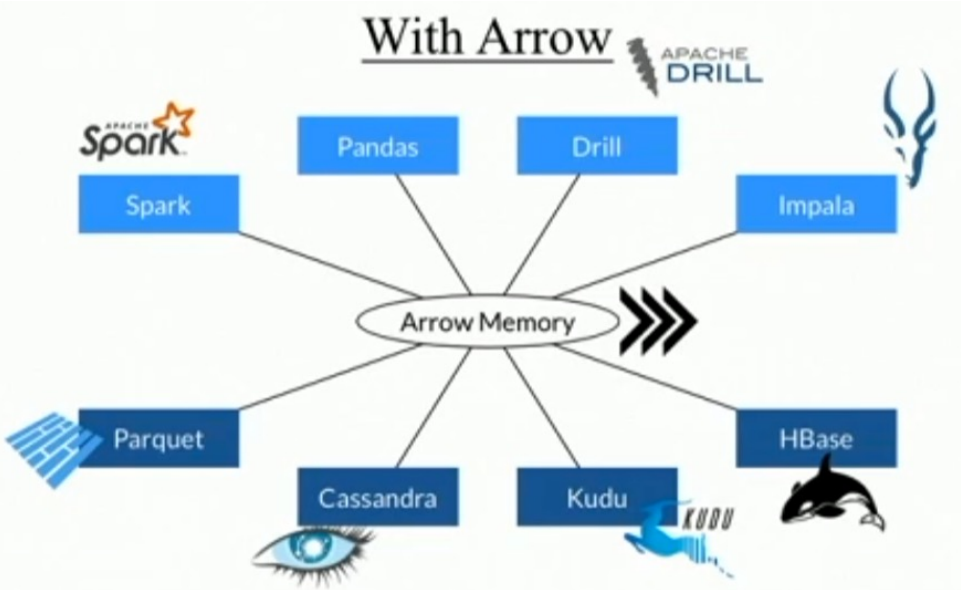
- A process's **Address Space:**
  - Slice of virtualize DRAM assigned to it alone!
  - OS "translates" DRAM vs disk address
- **Page Replacement Policy:**
  - When DRAM fills up, which cached page to evict?
  - Many policies in OS literature
- **Memory Leaks:**
  - Process forgot to "free" pages used a while ago
  - Wastes DRAM and slows down system
- **Garbage Collection:**
  - Some PL implementations can auto-reclaim some wasted memory

Take CSE 120 or 132C for more on memory management

Storing Data In Memory

- Any data structure in memory is overlaid on pages
- Process can ask OS for more memory in System Call API
  - If OS denies, process may crash
- **Apache Arrow:**
  - Emerging standard for columnar in-memory data layout
  - Compatible with Pandas, (Py)Spark, Parquet, etc.





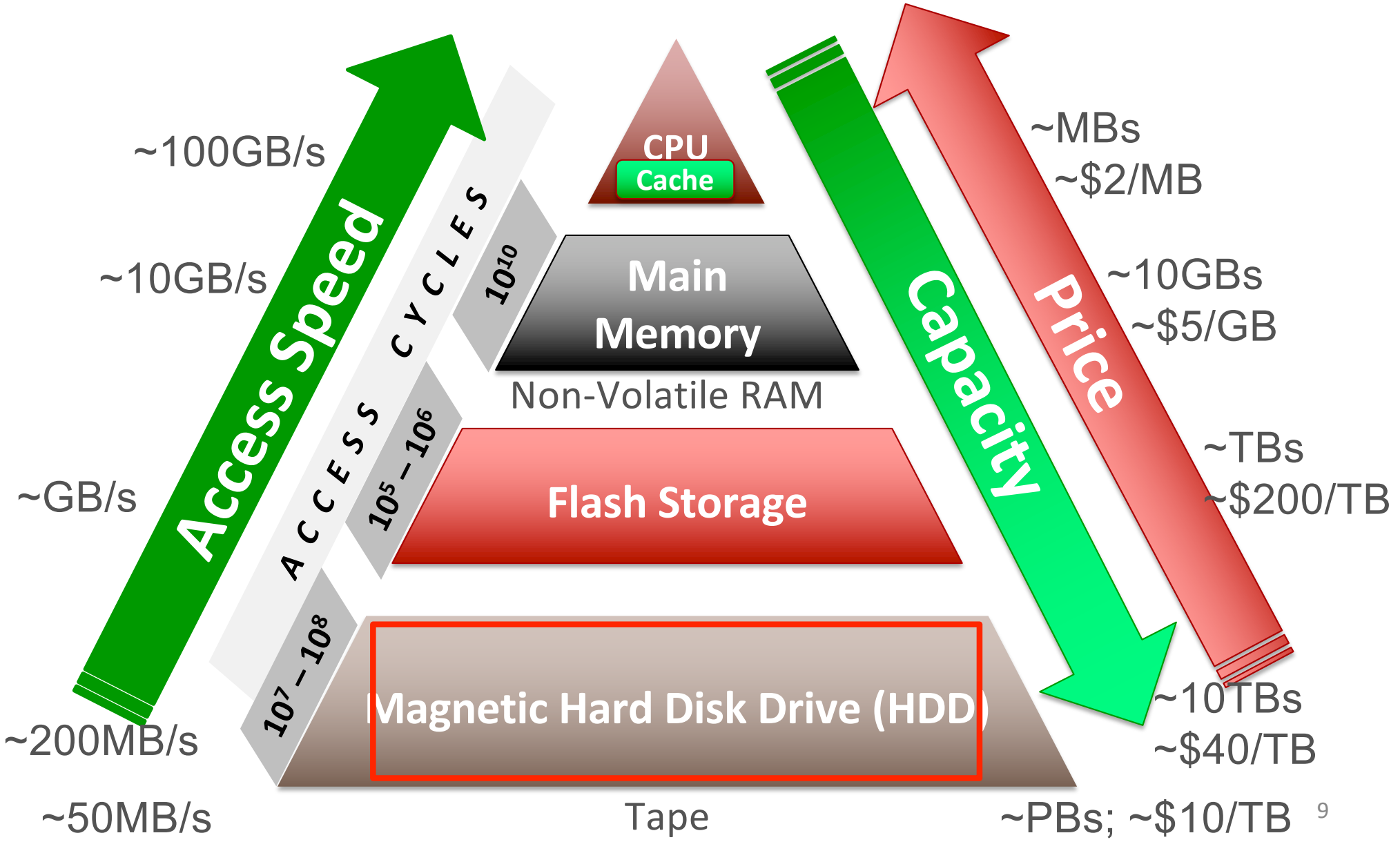
Outline

- Basics of Computer Organization
  - Digital Representation of Data
  - Processors and Memory Hierarchy
- Basics of Operating Systems (OS)
  - Process Management: Virtualization; Concurrency
  - Filesystem and Data Files
  - Main Memory Management
- Persistent Data Storage

Persistent Data Storage

- **Volatile Memory**: A data storage device that needs power/electricity to store bits; e.g., DRAM, CPU caches (SRAM)
- **Persistence**: Program state/data is available intact even after process finishes
- **Non-Volatile** or **Persistent** memory/storage: A data storage device that retains bits intact after power cycling
  - E.g., all levels below DRAM in memory hierarchy
  - "**Persistent Memory** (PMEM)": Marketing term for large DRAM that is backed up by battery power!
  - **Non-Volatile RAM (NVRAM)**: Popular term for DRAM-like device that is genuinely non-volatile (no battery)
  - Note: PMEM and NVRAM are typically used in high-performance servers and storage systems where fast, reliable access to data is critical.

Memory/Storage Hierarchy



Disks

- Aka secondary storage; likely holds the vast majority of the world's day-to-day business-critical data!
- Data storage/retrieval units: **disk blocks** or **pages**
- Unlike RAM, different disk pages have different retrieval times based on location:
  - Need to *optimize* layout of data on disk pages
  - Orders of magnitude performance gaps possible

Data Organization on Disk

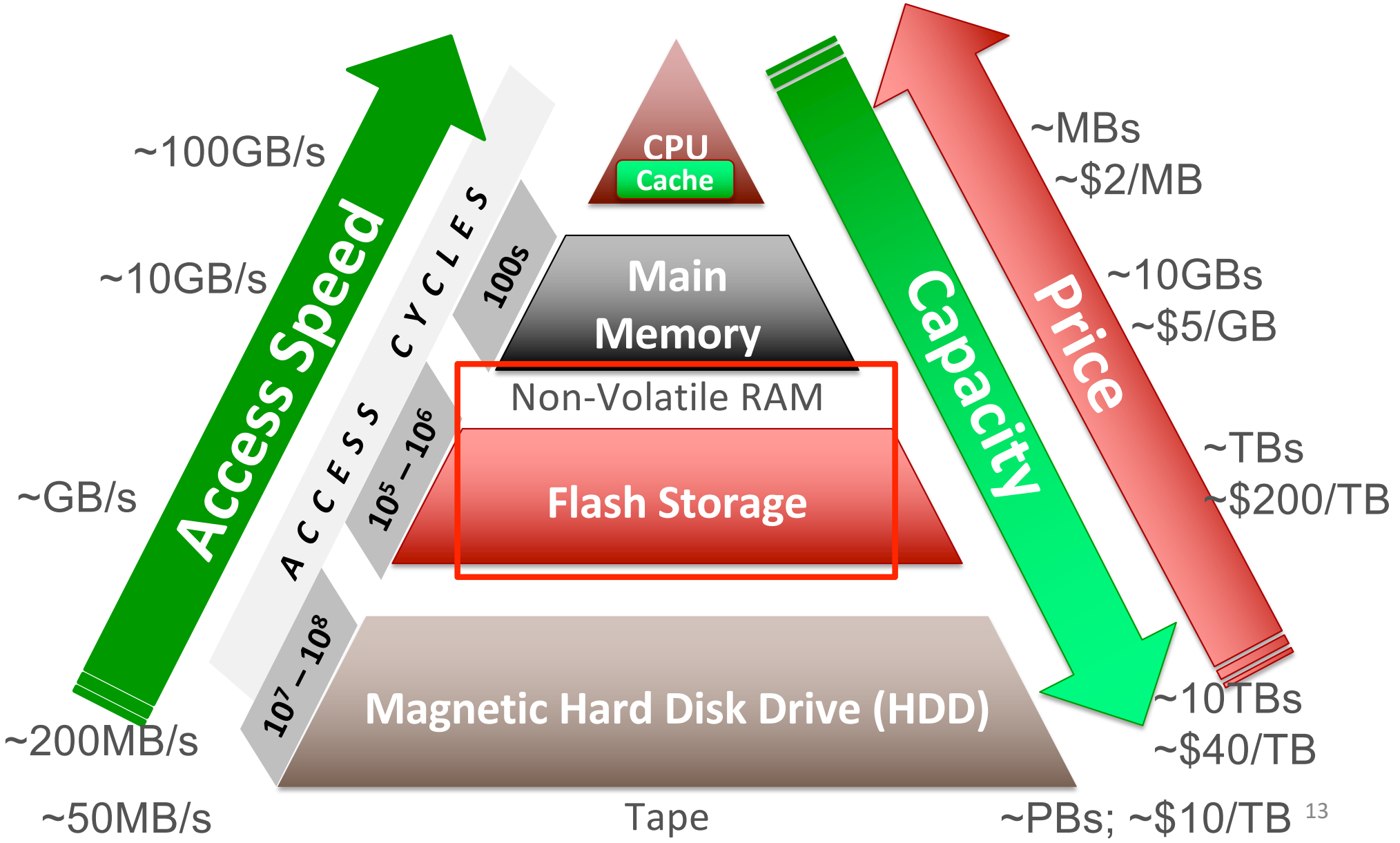
- Disk space is organized into **files**
- Files are made up of disk **pages** aka **blocks**
- Typical disk block/page size: 4KB or 8KB
  - Basic unit of reads/writes for a disk
  - OS/RAM page is *not* the same as disk page!
  - Typically, [OS/RAM page size] = [Disk page size] but not always; disk page can be a multiple, e.g., 1MB
- File data (de-)allocated in increments of disk pages

Magnetic Disk Quirks

- **Key Principle**: Sequential vs. Random Access Dichotomy
- Accessing disk pages in sequential order gives *higher throughput*
  - Random reads/writes are OOM slower!
- Need to carefully lay out data pages on disk, not the case for DRAM.
- Abstracted away by data systems: Dask, Spark, RDBMSs, etc.

Take CSE 132C for more on quirks of magnetic disks

Memory/Storage Hierarchy



Flash SSD vs Magnetic Hard Disks

Roughly speaking, flash combines the speed benefits of DRAM with persistence of disks

- Random reads/writes are not much worse
  - Different locality of reference for data/file layout
  - But still block-addressable like HDDs
- Data access latency: 100x faster! (Note: Access ~ Lookup)
- Data transfer throughput: Also 10-100x higher (Note: Access ~ Read/Write once lookup complete)
- Parallel read/writes more feasible
- Cost per GB is 5-15x higher!
- Read-write impact asymmetry; much lower lifetimes

NVRAM vs. Magnetic Hard Disks

Roughly speaking, NVRAM is like a non-volatile form of DRAM, but with similar capacity as SSDs

- Random R/W with less to no SSD-style wear and tear
  - Byte-addressability (not blocks like SSDs/HDDs)
  - Spatial locality of reference like DRAM; radical change!
- Latency, throughput, parallelism, etc. similar to DRAM
- Alas, limited to HPC and enterprise environments

Review Questions

- What are the 2 levels of a filesystem? Why the dichotomy?
- How is a database different from a file?
- What are the 2 levels of a database? Why the dichotomy?
- Name 3 forms of structured, 2 forms of semistructured, and 2 forms of unstructured data models.
- Describe 2 differences between a relation and a DataFrame.
- Can you store a relation as a DataFrame? Vice versa?
- Can you store a tensor as a relation? Vice versa?
- What is the address space of a process? What is a memory leak?
- What is Parquet? Explain 3 pros of Parquet over CSVs.
- What is Arrow? How is it different from Parquet? Why are both gaining popularity in practice?

  (See: https://stackoverflow.com/questions/56472727/difference-between-apache-parquet-and-arrow#56481636)
- Name 3 forms of persistent storage devices and describe some tradeoffs.

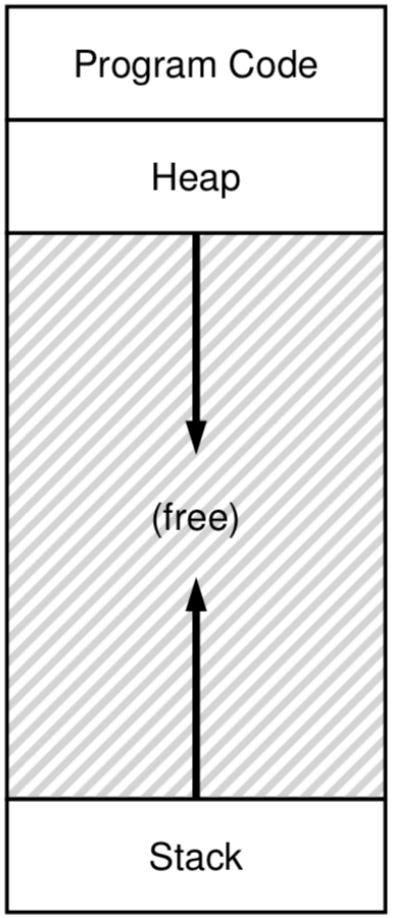
Outline

- Basics of Computer Organization
  - Digital Representation of Data
  - Processors and Memory Hierarchy
- Basics of Operating Systems (OS)
  - Process Management: Virtualization; Concurrency
  - Filesystem and Main Memory Management
- Persistent Data Storage
- Bonus non-syllabus material
  - More on Memory Management and Hard Disks

Optional: More on Memory Management
Not included in syllabus

Address Space

- Chunk(s) of memory assigned by OS to a process
  - Helps virtualizes and apportion physical memory
- Split into 3 **segments**: Code, Stack, and Heap
  - Stack stores mostly statically known data (function arguments, return values, etc.)
  - Heap is for dynamically created data structures (**malloc()** system call)
  - Stack/Heap can grow/shrink on the fly when a process is running
  - **Segmentation fault**: illegal address access
  - **Memory leak**: program failed to **free()** dynamic space



More On Segmentation Faults

- Can happen for several reasons
  - Null pointer – program tries to access memory location that has not been allocated
  - Out of bounds access – attempt to access memory that was not allocated by OS
  - Writing to read-only memory
  - Stack overflow – program uses too much stack space (creating too many nested function calls or allocating large arrays on stack

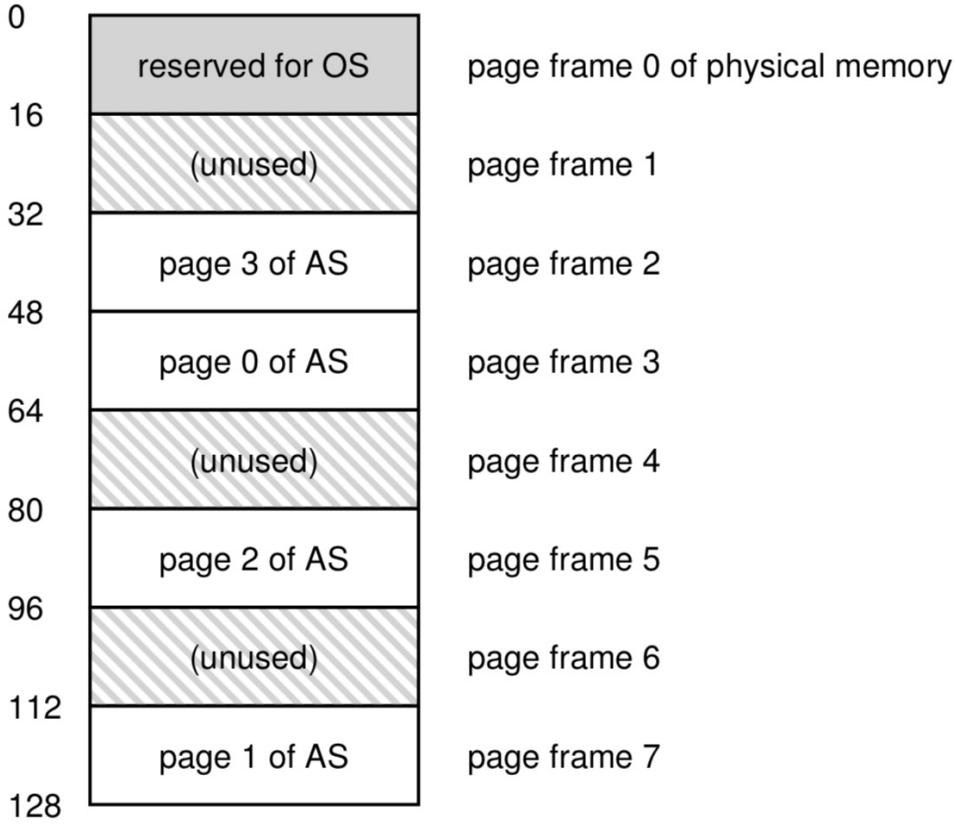
Virtual Memory

- **Virtual** Address vs **Physical** Address:
  - Physical is tricky and not flexible for programs
  - Virtual gives “isolation” illusion when using DRAM
  - OS and hardware work together to quickly perform **address translation**
- OS maintains **free space list** to tell which chunks of DRAM are available for new processes, avoid conflicts, etc.
  - Variable-sized
  - **Fragmentation** possible; algorithms exist to tackle it
- If DRAM space not enough, OS can map virtual address to disk (lower level in memory hierarchy)

Abstraction of Page in Memory

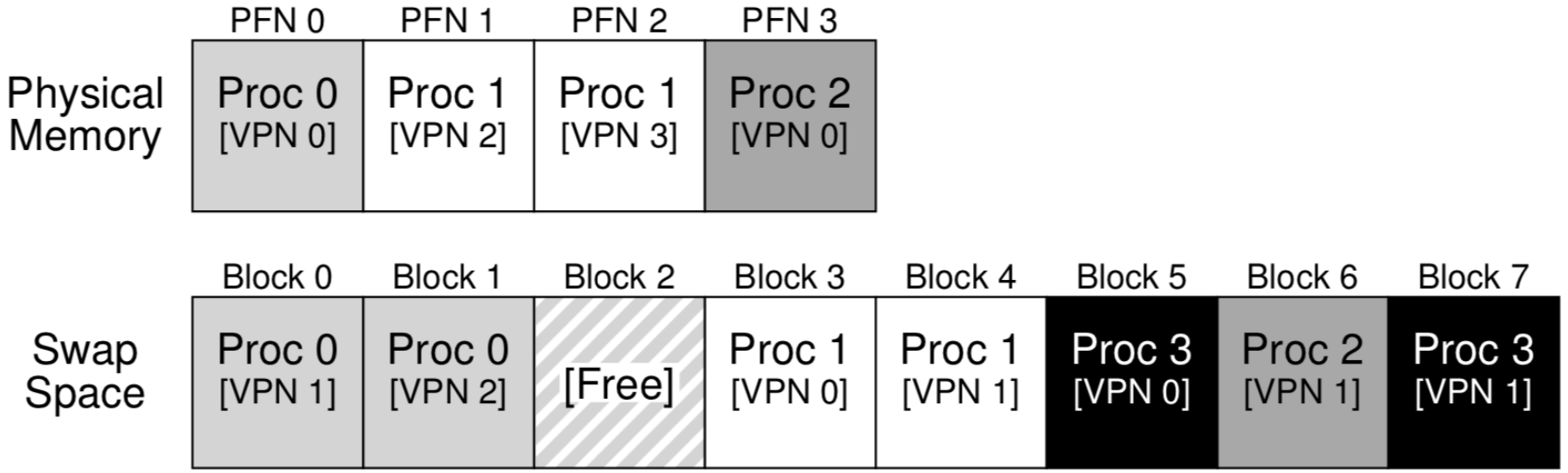
- **Page**: An abstraction of *fixed* size chunks of storage
  - Makes it easier to manage memory virtualization
- **Page Frame**: A virtual "slot" in DRAM to hold a page

- Frame numbers; virtual vs physical page numbers
- OS has **page table** data structure per process to map virtual to physical
- Overall, DRAM chopped up by OS neatly into frames



Swap Space and Paging

- Sometimes, DRAM may not be enough for process(es)
  - OS expands virtual memory idea to disk-resident data
- **Swap Space**: OS reserved space on disk to *swap* pages in and out of DRAM (physical memory)
  - OS should know **disk address** of pages and translate
  - Later: how data is laid out on disks

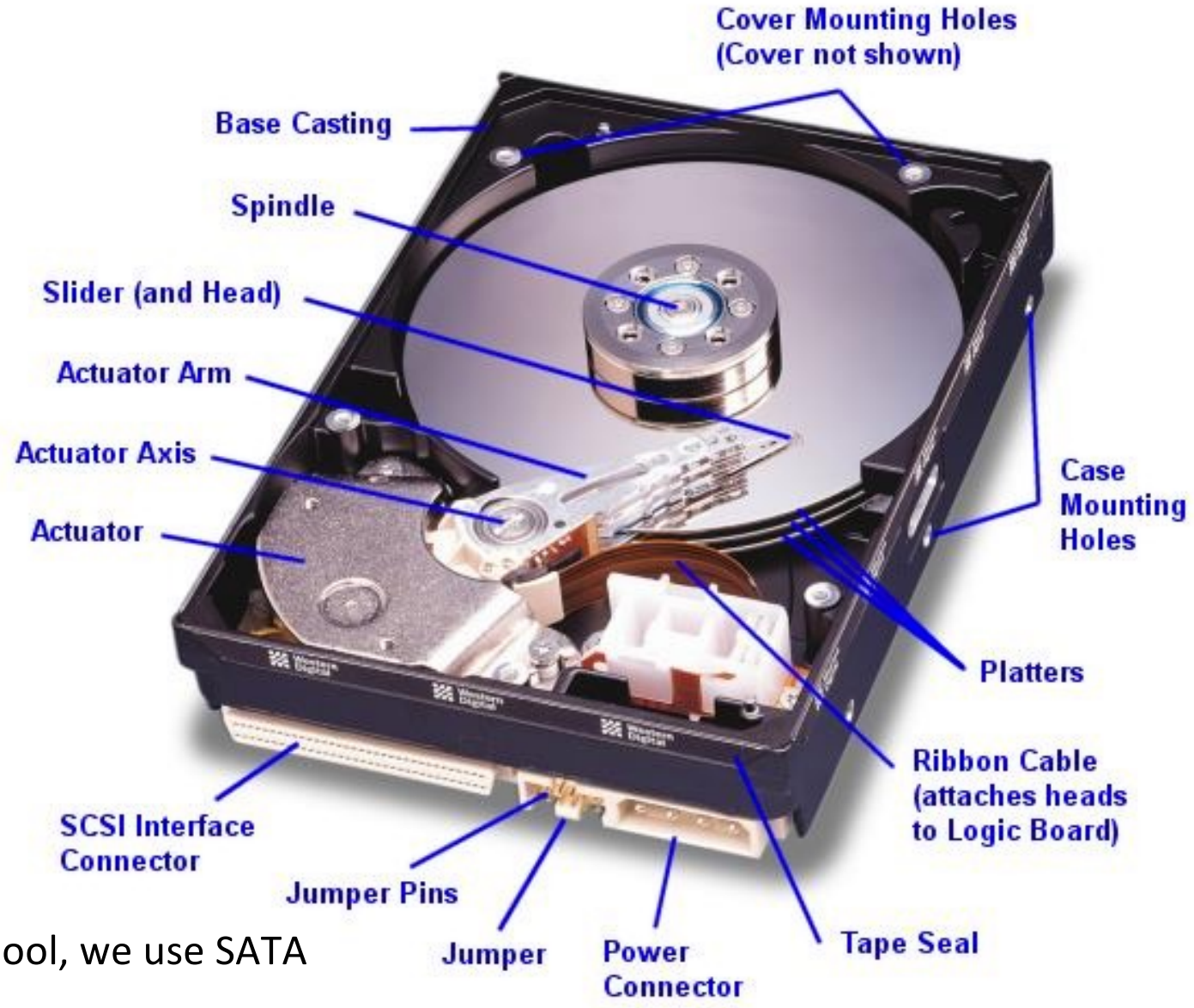


Page Replacement

- Recall DRAM has page frames to hold page content; a process's address space may only have so many frames
- **Page Fault**: A page required by process is not in DRAM
  - OS intervenes to read page from disk to DRAM
  - If free page frame available in DRAM, all good
- **Page Replacement**: If no frame is free when page fault happens, OS must evict some occupied frame's page!
  - **Page Replacement Policy** (aka cache repl. policy): Algorithm that OS uses to tell what page to evict
  - Various policies exist with different *performance* and *complexity* tadeoffs: FIFO, MRU, LRU, etc. (later topic)

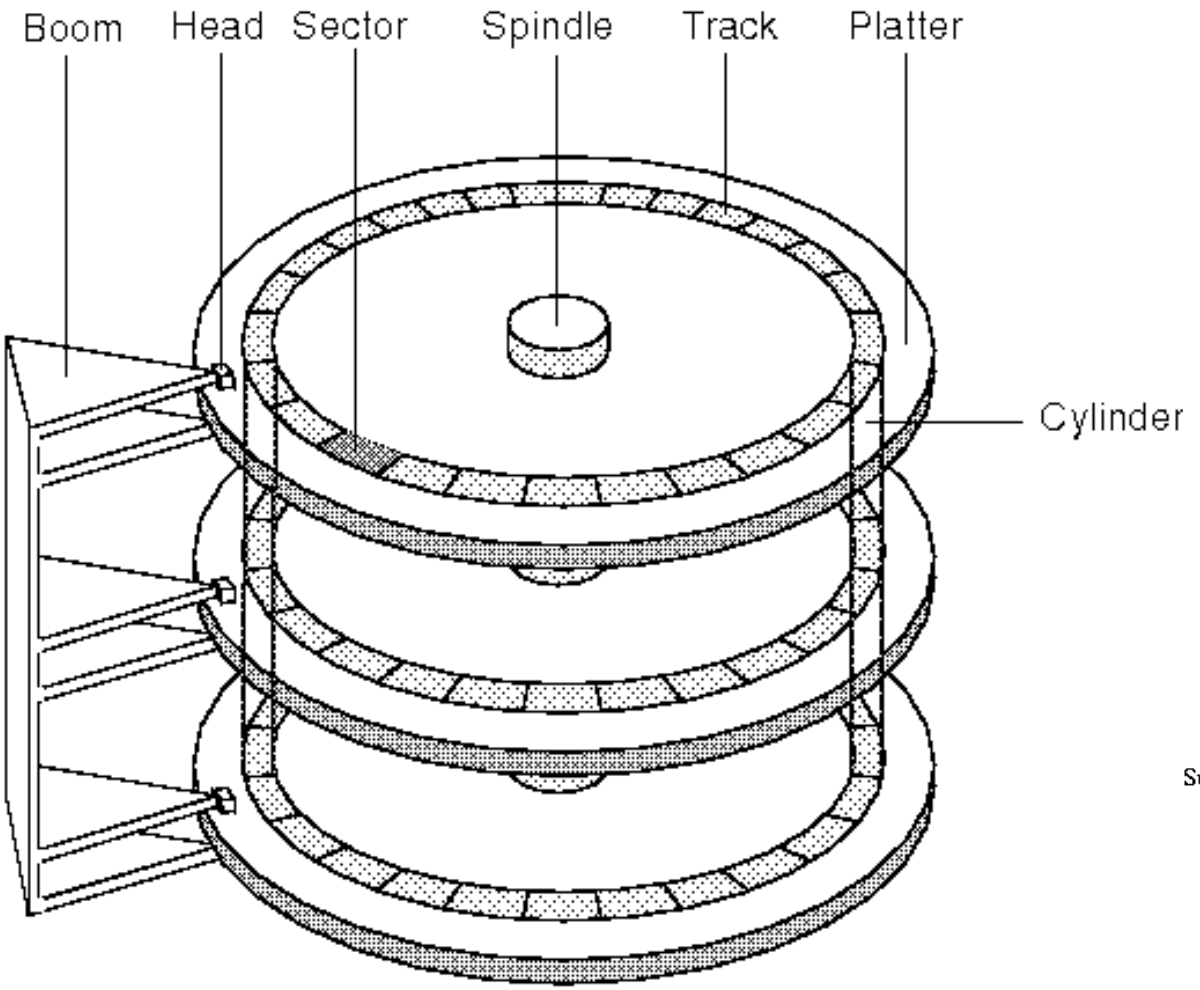
Optional: More on Magnetic Hard Disks
Not included in syllabus

Components of a Disk

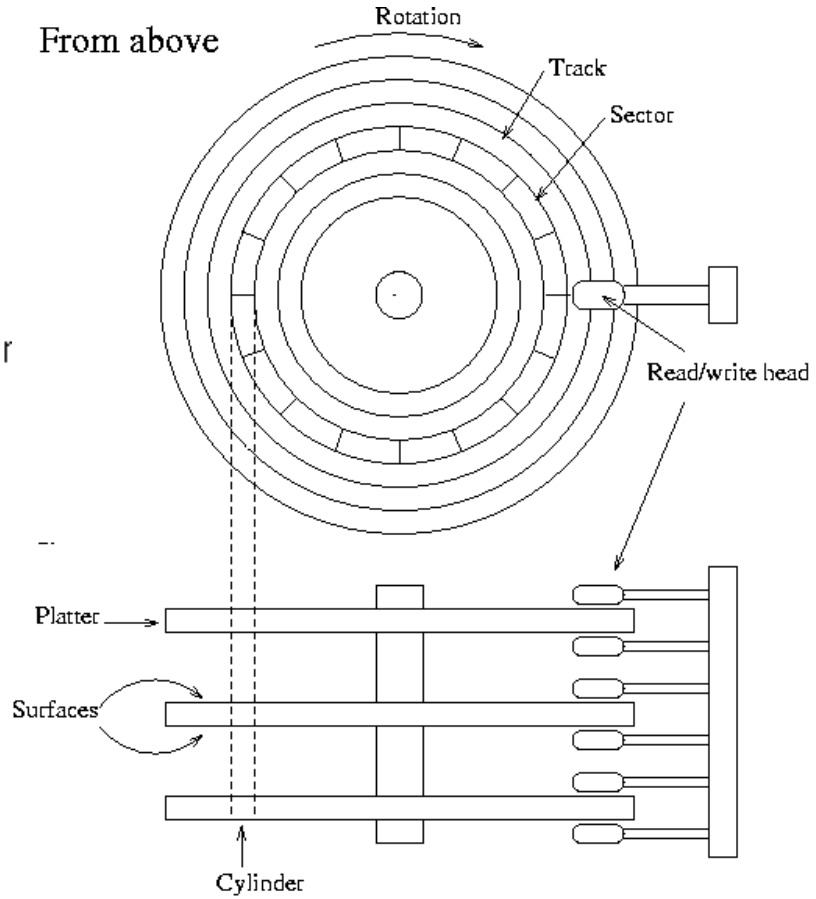


SCSI is old school, we use SATA

Components of a Disk

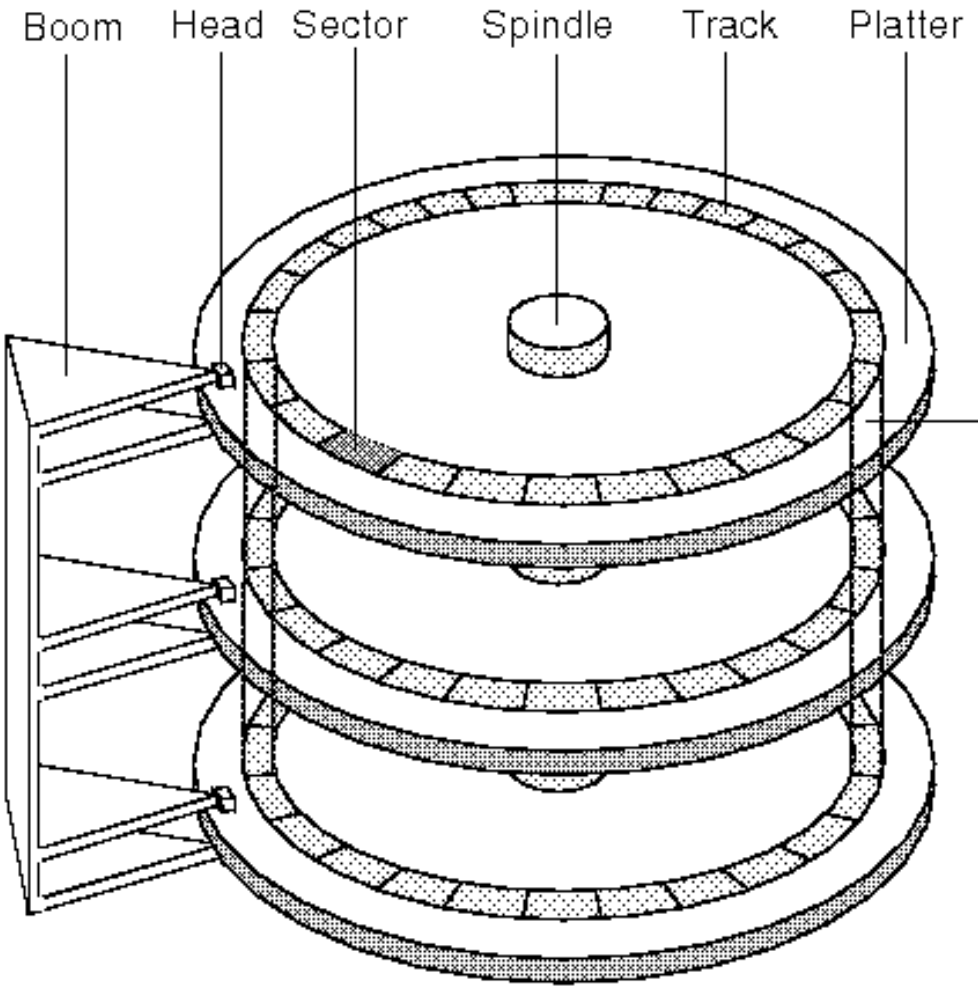


Anatomy of a regular hard disk



1 block = n contiguous sectors
(n fixed during disk configuration)

How does a Disk Work?



Anatomy of a regular hard disk

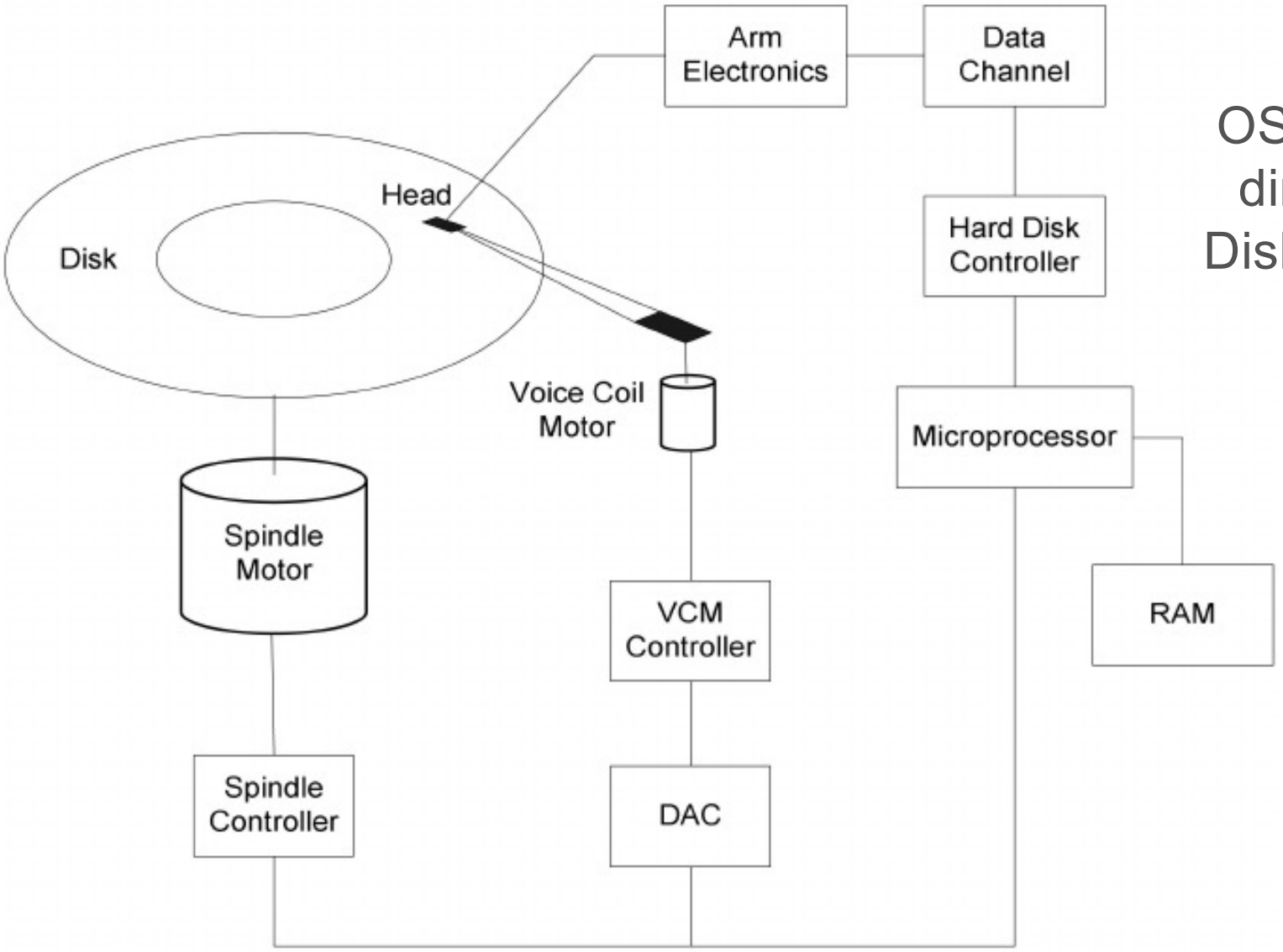
- Magnetic changes on platters to store bits
- Spindle rotates platters

7200 to 15000 RPM

(Rotations Per Minute)

- Head reads/writes track
- Exactly 1 head can read/write at a time
- Arm moves radially to position head on track

How is the Disk Integrated?



OS interfaces directly with Disk Controller

Disk Access Times

Access time = Rotational delay + Seek time + Transfer time

- Rotational delay
  - Waiting for sector to come under disk head
  - Function of RPM; typically, 0-10ms (avg v worst)
- Seek time
  - Moving disk head to correct track
  - Typically, 1-20ms (high-end disks: avg is 4ms)
- Transfer time
  - Moving data from/to disk surface
  - Typically, hundreds of MB/s!

Typical Modern Disk Spec

Western Digital Blue WD10EZEX (from Amazon)

Capacity	1TB
RPM	7200
Transfer	6 Gb/s
#Platters	Just 1!
Avg Seek	9ms
Price	$39