Systems Primer

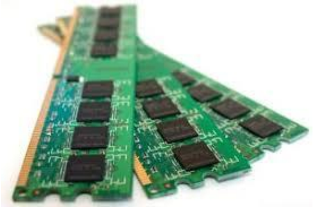
Bits to disks to clouds

In this Section

- Basics of Systems
    - GBs/TBs of data, speeds on RAM, disks, clouds
- Examples analyzing system design choices

RAM, Disks, Clouds

(Example [datacenter] from 2:50 min)



Fast: Random access vs disks, byte addressable

- ~10x faster for sequential access
- ~100,000x faster for random access!

Volatile: Lose Data, if e.g. crash occurs, power goes out

Expensive: For $100, 16GB of RAM vs. 2TB of disk!

disk rotation [video]

Slow: Sequential *block* access

- Disk read / writes are slow/expensive!

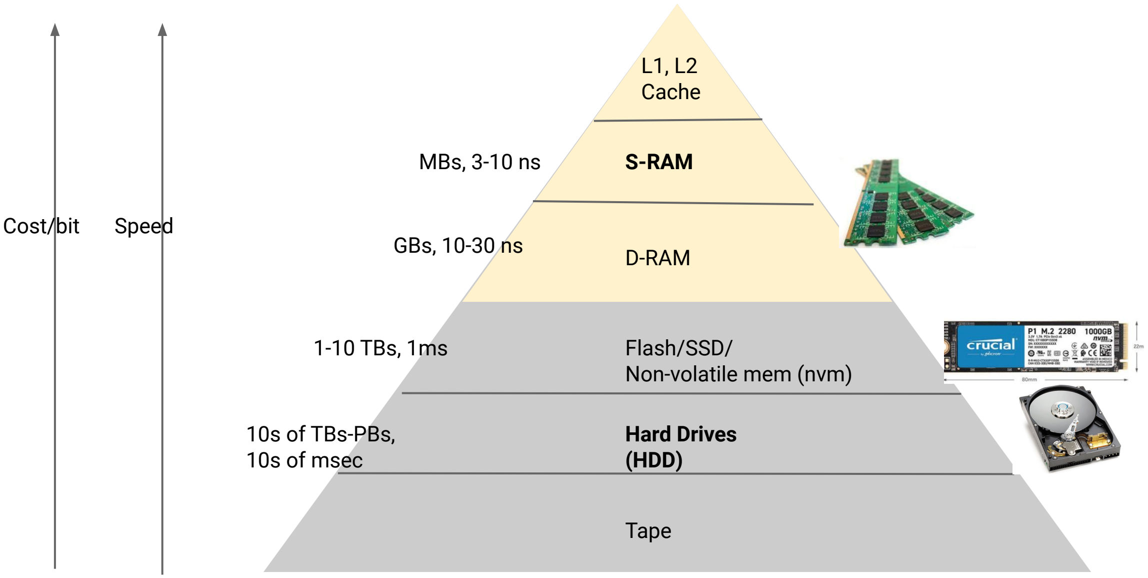
Durable: Data is safe* (assume for this class)!

Cheap

IO Hierarchy

Volatile -- data lost when you turn off power

Non-Volatile



Rough rule of thumb

<1-10 GBs
Usual CS algorithms
Pandas + SQL

> 10 GBs - 10 TBs
SQL on a cluster
Store on SSDs

> 10 TBs
SQL + Cs145 algorithms
Store on HDs

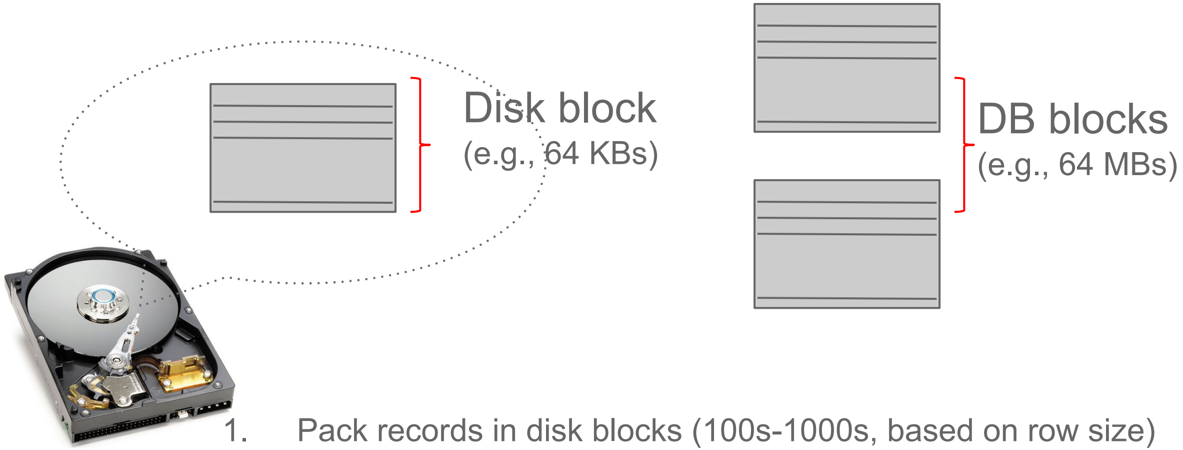
⇒ Rest of cs145: Focus on simplified RAM + HDD model
(learn tools for other IO models)

Disk blocks

& DB blocks

After all the hard work to seek, get a big Block?

(not just a byte; Imagine an Amazon Prime truck delivering 1 package at a time, vs 1000 to your street)



1. Pack records in disk blocks (100s-1000s, based on row size)
2. When you seek and read, you get a full disk block (i.e., you get 64 KBs, not just a byte)
3. Even better? Create a DB block with 1000 contiguous disk blocks, I.e., get 64 MBs per seek

Example: To store a 1 TB table on a disk (with 64MB DB blocks)

⇒ We’d need 15,625 DB blocks

⇒ Each seek will get you back a full 64MB block

Basic system numbers

Srigi
@srigi

"Latency Numbers Every Programmer Should Know"

It is hard for humans to get the picture until you translate it to "human numbers":

1 CPU cycle	1 s
Level 1 cache access	3 s
Level 2 cache access	9 s
Level 3 cache access	43 s
Main memory access	6 min
Solid-state disk I/O	2-6 days
Rotational disk I/O	1-12 months
Internet: SF to NYC	4 years
Internet: SF to UK	8 years
Internet: SF to Australia	19 years
OS virtualization reboot	423 years
SCSI command time-out	3000 years
Hardware virtualization reboot	4000 years
Physical system reboot	32 millenia

nanosecond (10^-9 sec)	microsecond (10^-6 sec)	millisecond (10^-3 sec)
Cache access: 4-10 ns RAM: 20ns	Datacenter network: O(1us) High-end flash: O(10us)	Disk: O(10 ms) Low-end flash: O(1ms) Wide-area networking: O(10ms)

MB/sec	GB/sec
Disk transfer rate: 100	RAM transfer rate: 100

Typical dedicated (non-shared) machine assumptions (unless problem states otherwise):

- 64 GB RAM
- Block sizes: 64 KB (for disk block, RAM page size), 64 MB (for DB block)
- Example: AWS/GCP offer machine instances
(e.g, ec2.r5 offers 1-3GBps network bandwidth, 2CPU/16GB RAM to 96 CPU/768GB RAM for $-$$$ in Nov'18)

2^10 = 1024, 2^20 ~= 1Million, 2^30 ~= 1 Billion (10^9), 2^40 ~= 1 Trillion (10^12)

- 4 byte int32, 8 byte int64
- To store int32 records: 1 Million records = 4MB, 1 Billion records = 4 GB
- [**Often use 1000 vs 1024**, as a quick approximation]

Example

Data size

Students			
SID	**Name**	**Address**	**Bio**
40001	Mickey	43 Toontown	Mickey is a Sophomore in CS. He is...
40002	Daffy	147 Main St	Daffy is part of the Orchestra. He was...
50003	Donald	312 Escondido	Donald is a 1st year MS in EE. He was...
50004	Minnie	451 Gates	
10008	Pluto	97 Packard	

Q1: What's row **size** (i.e., size of each record)?

SID: int32 ⇒ 4 bytes
Student: char[100] ⇒ 100 bytes
Address: char[200] ⇒ 200 bytes
Bio: char[696] ⇒ 720 bytes   ## Note: Picking so row size=1024

⇒ Each row is 1024 bytes

Q2: What's table **size**

a. with 1000 students? 1000 * 1024 bytes = 1 MBytes
b. with 1M students? 1M * 1024 bytes = 1 GB
c. With 1B students? 1B * 1024 bytes = 1 TB

Example

Data speed

Q3: For 1 Billion student table of size 1TBs

a. Scan from RAM? (@100GB/sec): 1 TB/100GBps = 10 secs
b. Scan from disk? (@100 MB/sec): 10,000 secs

c. Single row fetch from RAM: 20 nsecs (i.e., 20 * 10^-9 sec)
d. Single disk block seek: 10 msecs (i.e., 10 * 10^-3 sec)

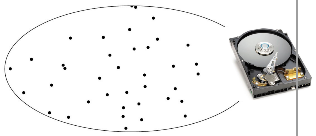
e. Read from RAM on another machine:
   i. (Network) 1 usec + (RAM) 20 nsec ~= 1.02 usec
   ii. That's 10,000x faster than reading from disk on <u>same</u> machine

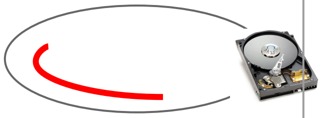
Q4: With 100 machines? (100x RAM, 100x disks)

a. Scans will be 100x faster
b. Time for first row fetch/seek? Same speed

Example: Find a student, by scanning data

Find ‘Daffy’ from a DB of billion students (1 TB)





Design Choices	Storage Cost	Time
Data in RAM (Scan sequentially & filter)	(@100$/16GB) 6000$	1000 GB / 100 GBps = 10 secs
Data in disk (in random spots) (Seek each record on disk & filter)	(@100$/TB of disk) 100$	(Seek) 10 msec * 1 Billion rows + (Scan) 1 TB /100 MBps = 10^7 secs + 10^4 secs ~= 115 days
Data organized in DB blocks (Seek to DB block, sequentially read records from disk & filter)	(@100$/TB of disk) 100$	Number of DB blocks = 1 TB/64MB = 15625 blocks (Seek) 10 msecs *15625 DB blocks + (Scan) 1 TB /100 MB-sec = 10.15^4 secs ~= 3 hrs

In 2 weeks, we’ll see how to do this a lot faster (in msecs) with good **Indexes** (e.g, hashing)

In this Section

- Basics of Systems
  - GBs/TBs of data, speeds on RAM, disks, clouds
- Examples analyzing system design choices