CONSTRUCTOR UNIVERSITY

Storing Data: Disks and Files

Garcia Molina, Ullman, Widom

Ramakrishnan/Gehrke Ch. 9

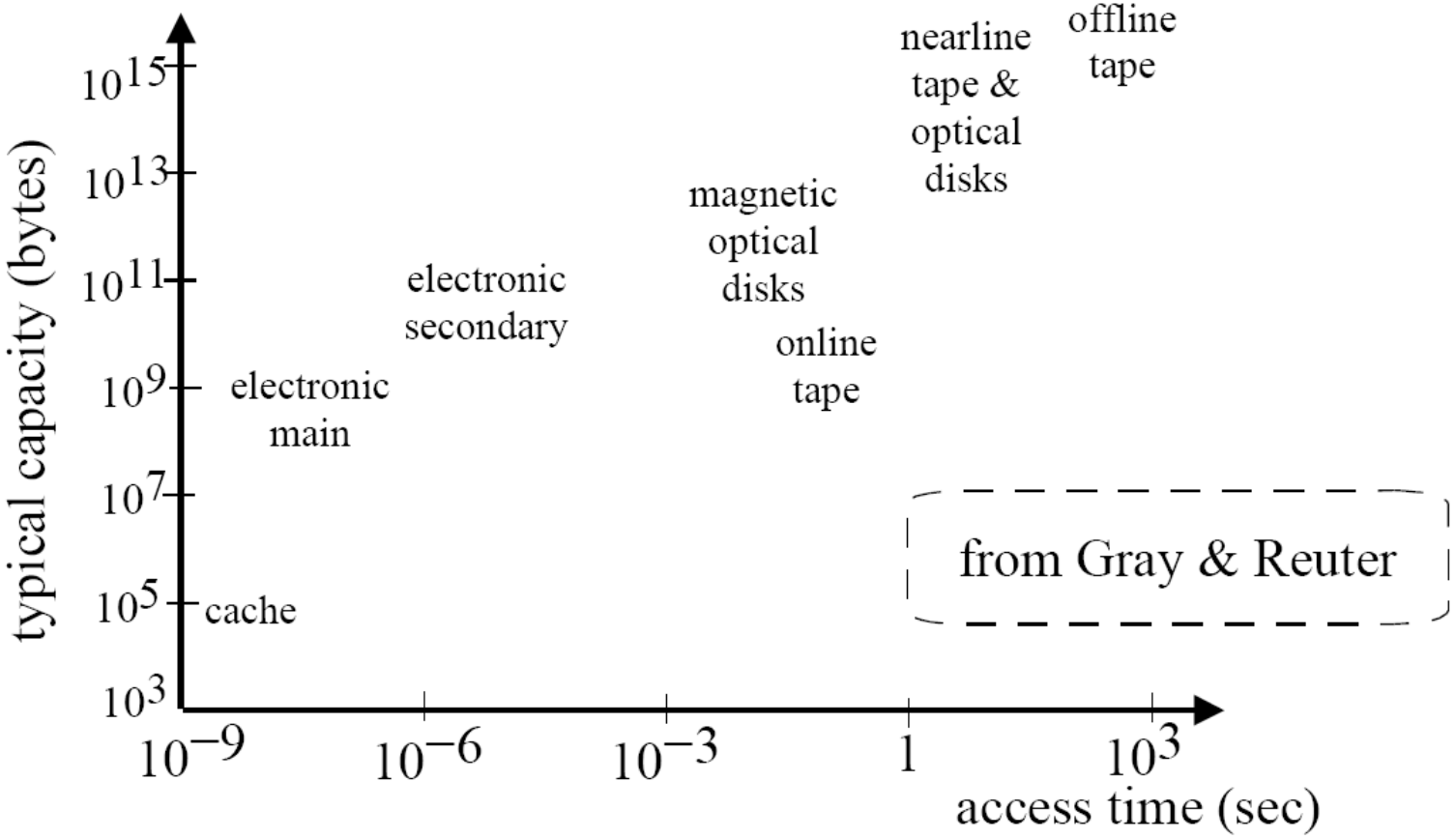
"Digital information lasts forever
- or five years, whichever comes first."
-- Jeff Rothenberg, RAND Corp., 1997

Why Not Everything in Main Memory?

- Costs too much
  - [Rama/Gehrke] $1000 will buy you either 128MB of RAM or 7.5GB of disk
  - Today: 80 EUR will buy you either 4 GB of RAM or 1 TB of disk
  - …but today we have multi-Terabyte databases!
- Main memory is volatile
  - want data to be saved between runs (obviously!)
- Typical storage hierarchy:
  - Main memory (RAM) for currently used data
  - Disk for main database (secondary storage)
  - Tapes for archiving older versions of data (tertiary storage)

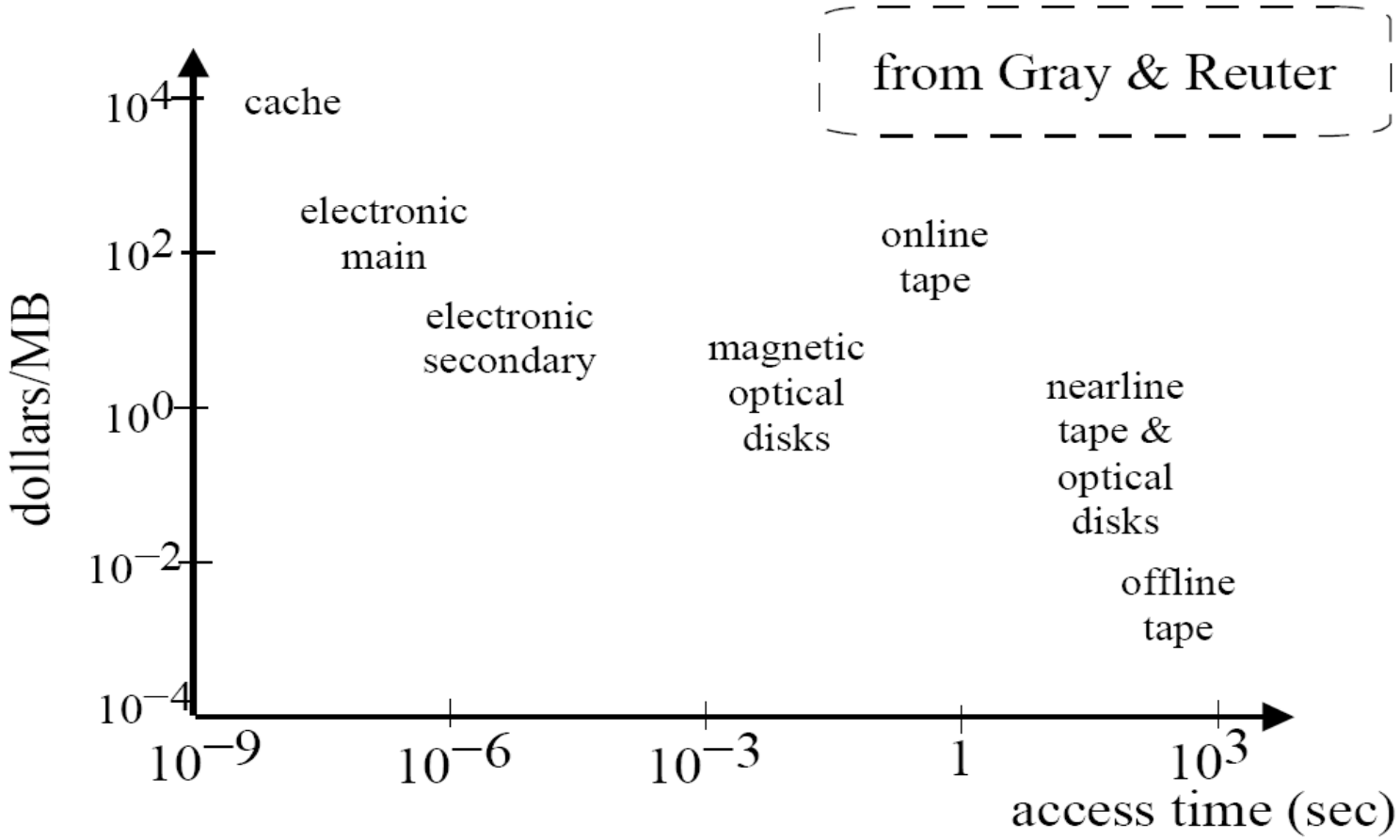
Storage Capacity

- Absolute times outdated, but ratios still ~ same

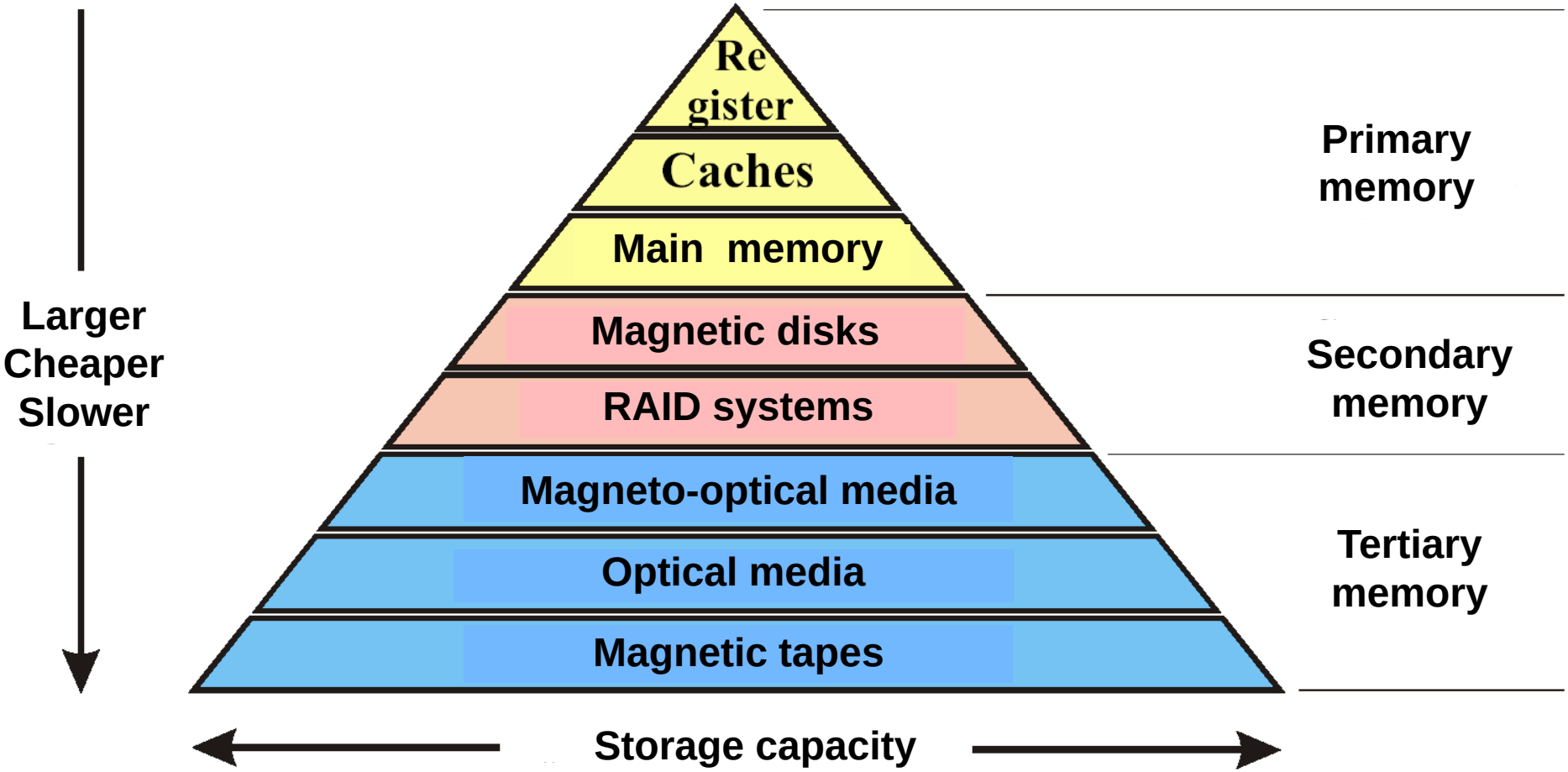


Storage Cost

- Again, absolute values outdated, but ratios still ~ same



Storage Hierarchies



Numbers

CPU Register	1 ns
Main Memory	100 ns
Flash Drive	100,000 ns
Hard Drive	10,000,000 ns

[http://carlos.bueno.org/2014/11/cache.html]

Nearline (Tertiary) Storage

- Usually tape
  - Reel, today: cartridge
  - Capacity 10 GB → ~6 TB per tape
- Tape robots
  - HSM =
    Hierarchical storage management
  - multi-Petabytes

Caching & Virtual Memory

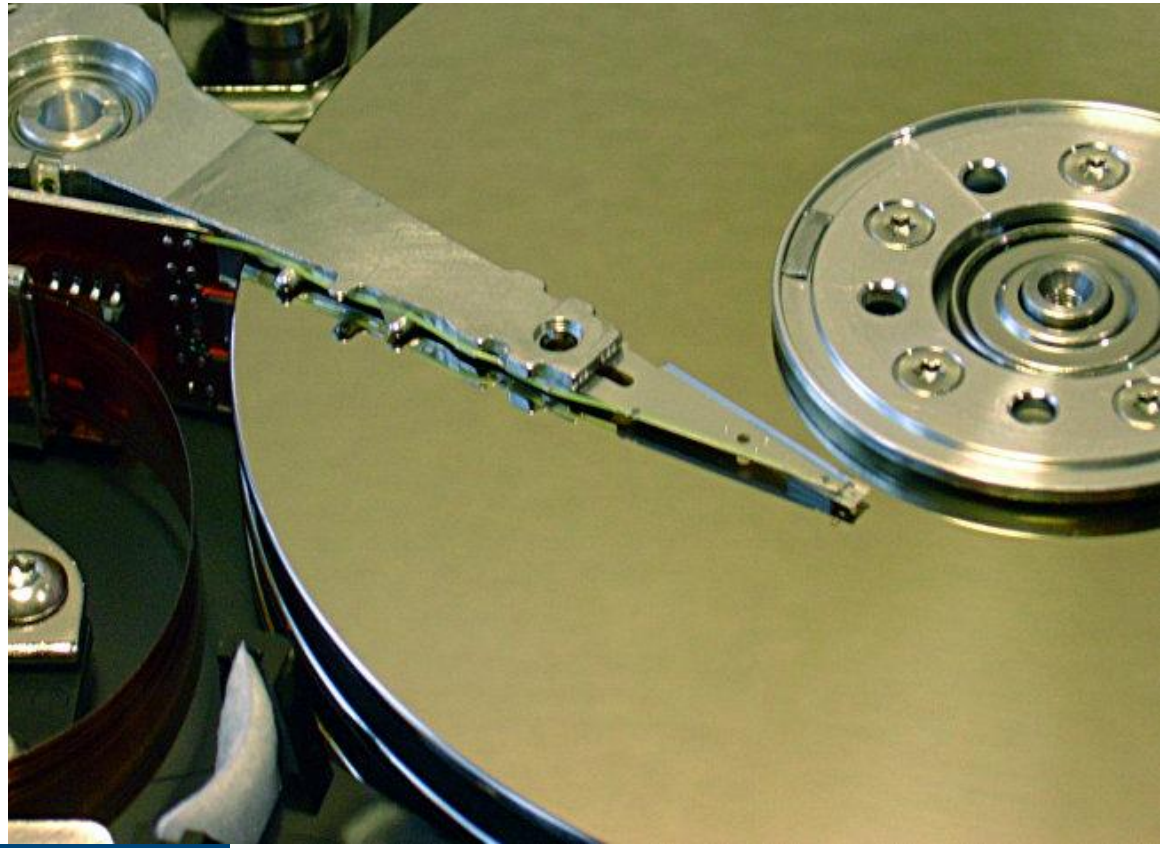
- Cache: Fast memory, holding frequently used parts of a slower, larger memory
  - small (L1) cache holds a few kilobytes of the memory "most recently used" by the processor
  - Most operating systems keep most recently used "pages" of memory in main memory, put the rest on disk
- Virtual memory
  - programs don't know whether accessing main memory or a page on secondary memory page (most operating systems)
- Database systems usually take explicit control over 2ndary memory access

Where Databases Reside

- Hard Disk is secondary storage device of choice
  - Many flavors:
    Disk: Floppy (hard, soft); Winchester; Ram disks; Optical, CD−ROM; Arrays
- Main advantage over tapes: random access vs. sequential
- Data stored and retrieved in units called disk blocks or pages
- Unlike RAM, time to retrieve a disk page varies depending upon location on disk
  - → relative placement of pages on disk has major impact on DBMS performance!

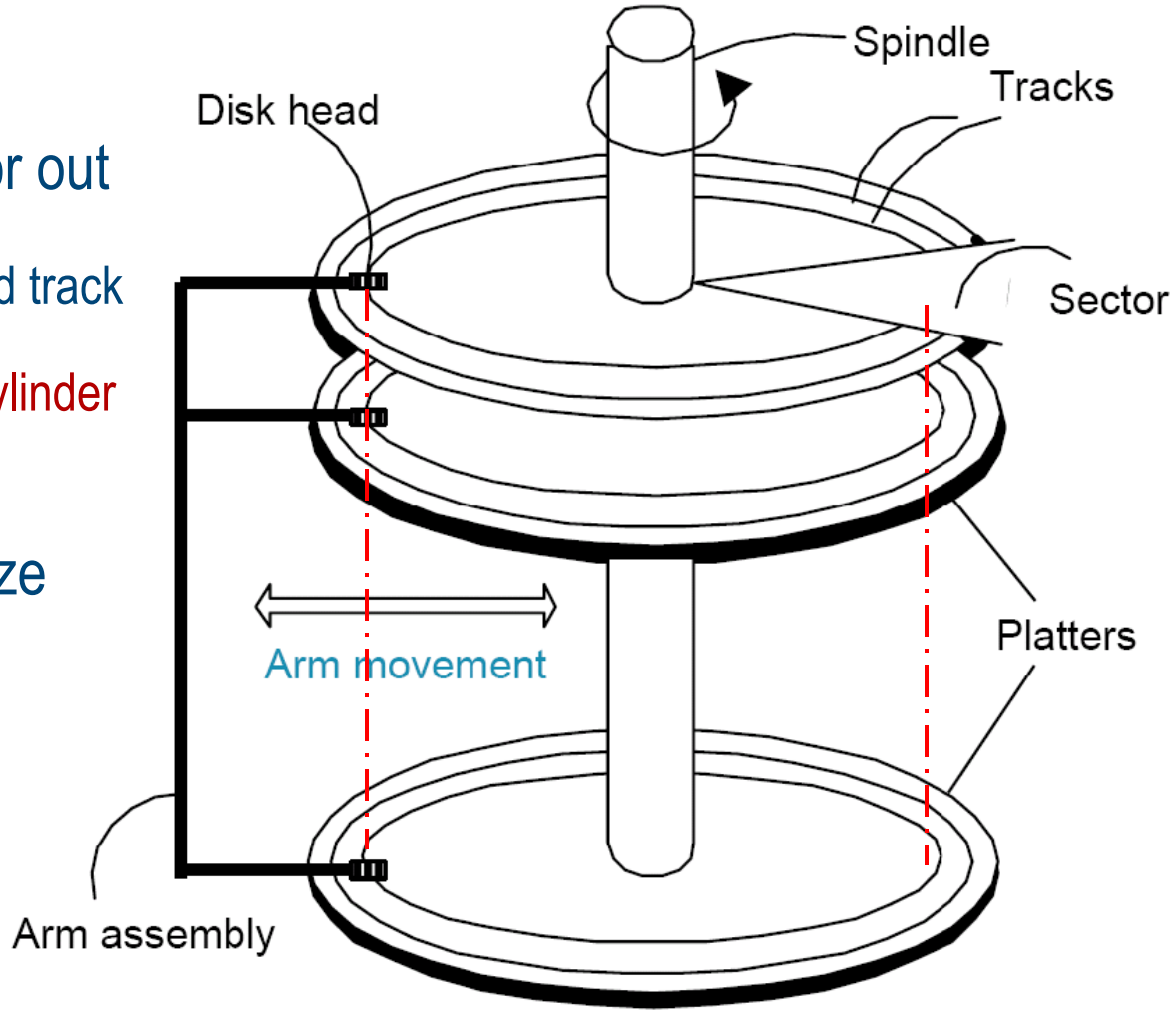
The Miracle Called "Hard Disk"

- Disk head contains magnet, hovering over spinning platter
- flight height: 10-20 nm
- (x 5,000 gives one hair!)

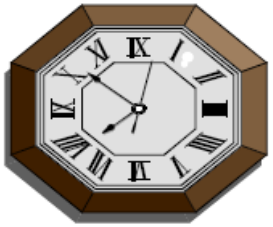


Components of a Disk

- platters spin
- arm assembly moves in or out
  - to position head on desired track
  - Tracks under heads = a cylinder (imaginary!)
- Sector size = N * block size (fixed)
- *...typical numbers?*



Disk Access Time

I want block X →  → block X in memory

?

Disk Access Time

Time = Seek Time +
Rotational Delay +
Transfer Time +
Other

Sequential Read?

- So far: Random Block Access
- What about: Reading next block?
- Disks optimized towards "consecutive" reading!
  - Blocks within track
  - Tracks within cylinder
  - Next cylinder

"Next Block" Costs

- `Next' block concept:
  - blocks on same track, followed by
  - blocks on same cylinder, followed by
  - blocks on adjacent cylinder
- If we don't need to change cylinder:

$$\text{Time to get} = \frac{\text{Block Size}}{\text{block t}} + \text{Negligible}$$

  - + switch track (ie, read next arm)
  - + once in a while, next cylinder

Random vs Sequential Read

- Rule of Thumb:
  - Random I/O: Expensive
  - Sequential I/O: Less expensive
- Ex: 1 KB Block:
  - Random I/O: ~ 20 ms
  - Sequential I/O: ~ 1 ms
- relative difference is smaller for larger blocks
- *→ Whenever possible arrange file blocks sequentially on disk (by `next') to minimize seek and rotational delay*
  - *For sequential scan, pre-fetching several pages at a time is a big win! → "burst read"*

...Writing?

- Cost for Writing ≈ cost for Reading
- ... unless we want to verify!
  - Then, need to add

$$\frac{\text{Block size}}{t} + \text{(full) rotation}$$

...To Modify a Block?

- (a) Read Block
- (b) Modify in Memory
- (c) Write Block
- [(d) Verify]

Wrap-Up

- Capacities grow, data hunger grows larger
  - Moore's Law vs Greg's Law vs disk growth
- Databases heavily i/o bound
  - Disk space management largely determines performance
- Disk access time =
  Seek Time + Rotational Delay + Transfer Time + Other
- Big win: burst read = read larger block set + cache