

COMP2400

Relational Databases

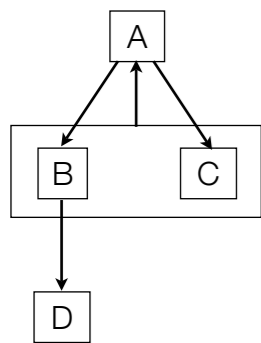
Lecture 29: Database Hardware

Ben Lippmeier
Australian National University
Semester 2
2008

EN3NF, an equivalent definition

- In “A New Normal Form for the Design of Relational Database Schemata” - Carlo Zaniolo, 1982
- A relation R is (EN)3NF iif for every non-trivial FD of R, $X \rightarrow A$
 - a) X is a super key for R or,
 - b) A is a key attribute for R.
- A relation R is BCNF iif for every non-trivial FD of R, $X \rightarrow A$, X is a super key for R.
- A trivial functional dependency $X \rightarrow A$ is one where X contains A.
- “key attribute” == “prime attribute”

Assignment 2 / Normal Forms



fun deps = { $A \rightarrow B$, $A \rightarrow C$, $BC \rightarrow A$, $B \rightarrow D$ }
candidate keys = { A, BC }
prime attrs = { A, B, C }
non-prime attrs = { D }
choose primary key = { A }
highest normal form = 2NF, but not EN2NF!

1NF: Attributes are atomic values, not more tuples or sets.

2NF: 1NF + Every non-prime attribute is fully functionally dependent on the primary key.
(fully functionally dependent = not dependent on a subset of the key)

3NF: 2NF + Every non-prime attribute is directly dependent on the primary key.
(directly dependent = non-transitively dependent)

For EN2NF and EN3NF consider all candidate keys, not just the primary key.

DBMS needs hardware to work.

- Our database needs to run on a physical machine. (shock!)
- What hardware should we buy for our application?
- Machine needs to be fast enough to handle the expected load.
- Must have enough storage space.
- Must be reliable, backed up, physically secure.
- Must be cost effective.

What's the setting?

- **Small Business**
 - Employee / Product / Stock / Order / Contact etc data.
 - Ubiquitous.
 - Often maintained ad-hoc in Excel / MYOB / Flat files etc.
 - Few 10's - 100's of MB, self maintained.
- **Medium / Large Business**
 - Fully fledged DBMS on dedicated hardware.
 - Maintained by dedicated staff.
 - Cost of downtime > \$100k / day
- **Critical Business / High Reliability**
 - Banking, stock market, airline, shipping etc
 - Replicated in multiple physical locations.
 - Cost of downtime > \$10Mil / day

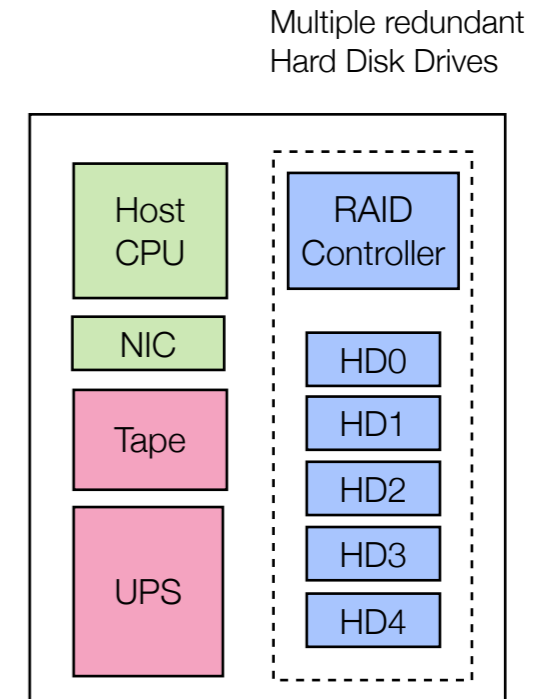
Typical dedicated DB server

Off-the-shelf host processor
DB usually not too CPU intensive.

Network interface.

Tape or other removable
media for offsite backup.
Often shared.

Uninterruptible Power Supply
for clean shut down in the
event of power failure.
Often shared.



What's the setting?

- **Industrial / Real-time**
 - Networking, telephony, industrial process.
 - Queries must complete within hard time limits, else mission fails.
 - Aggressive caching, perhaps store all data in RAM.
- **Scientific**
 - Experimental results, sensor readings, satellite imagery etc.
 - 10's of MB to many TB.
 - Multiple offsite backups.
 - Experimental results must be archived indefinitely.
- **Analytic**
 - De-normalized versions of existing databases.
 - Business modeling, historical records.

Example Hardware



IBM x3650 Server



Sun StorageTek 2500
Disk Array

Hard Disk Drives provide nonvolatile storage

- Many databases are too large to fit in main memory.
- When the power goes off, the data remains.
- Cheap, reliable, ubiquitous.
- Block-at-a-time access.
- Typical “server level” drive, in Q4 2008:

Seagate Barracuda ES.2 ST31000340NS
 1TB capacity, 7200 RPM, 116MB/s throughput,
 9ms access time, 11W power, AUD\$362 cost.



Access time is limited by platter rotation speed.

$$\text{AccessTime} = \text{SeekTime} + \text{RotationalDelay} + \text{BlockTransferTime}$$

AccessTime total time to load a data block into HDD cache.

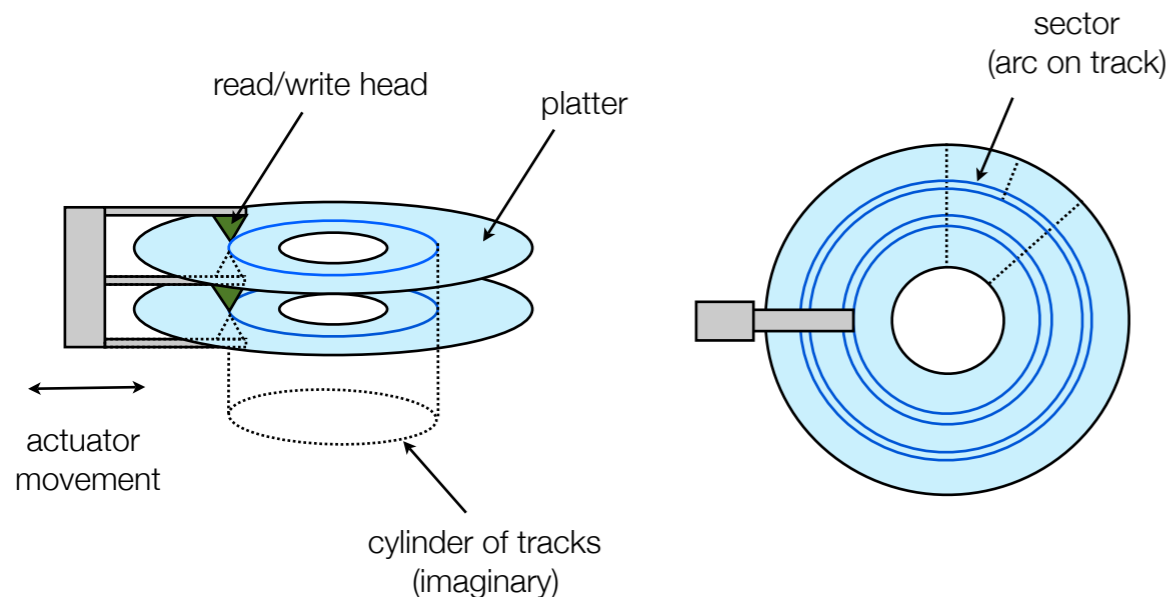
SeekTime time to move head to target track.

RotationalDelay time to wait for block rotate under head.

BlockTransferTime time to read target block.

- Increasing density of data increases storage capacity and throughput, but not access times.

HDD cross section, adapted from E&N 13.2



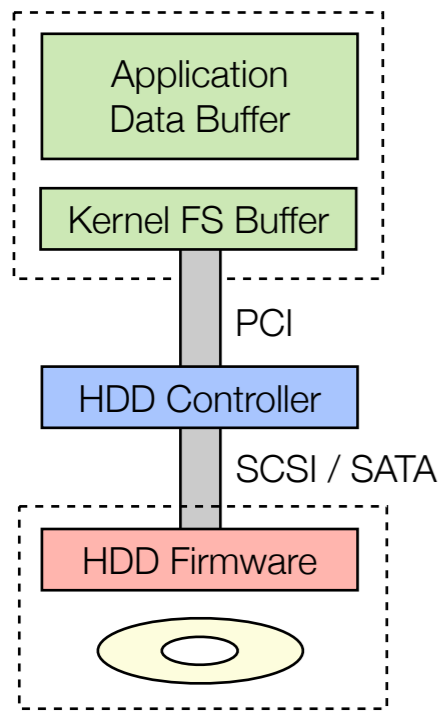
Access times have not kept up.

Seagate Barracuda Series HDD

year	model number	spindle (RPM)	capacity (GB)	sustained transfer (MByte/s)	avg access (ms)
1991	ST11950N	7200	2	5	8
1993	ST15150N	7200	5	8	8
1996	ST19171N	7200	12	13	10
1998	ST136475LW	7200	37	24	8
2001	ST1181677LW	7200	242	49	8
2004	ST3200822A	7200	200	58	9
2006	ST3500841AS	7200	500	65	8
2008	ST31000340NS	7200	1000	105	9

$$7200 \text{ rev/min} = 120 \text{ rev/s} = 8 \text{ ms/rev}$$

HDD cache hierarchy



Application data and kernel buffers share main memory, few GB.

Modern OS's speculatively cache data read from physical media.

Some DBMS bypass kernel buffering and manage this directly.

Data cached directly in HDD controller with dedicated RAM. Battery backup to protect data in write cache.

Data buffered in HDD firmware to cover latency due to bus contention.

Were you backed up?



Image: Tom's Hardware Guide

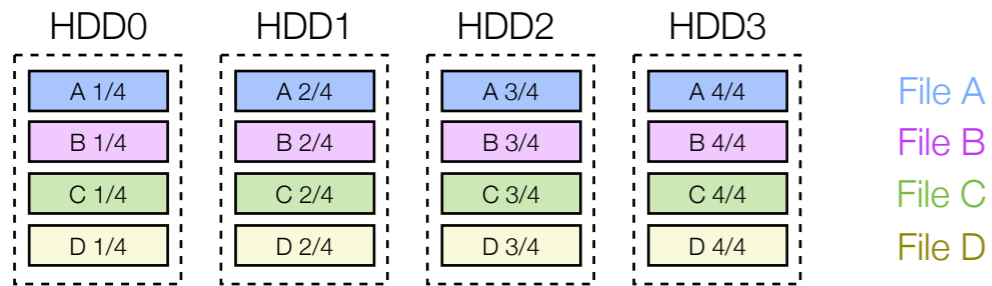
HDD failure modes

- **Media failure**
Imperfections in magnetic media result in localized data loss. HDD firmware keeps "spare" blocks to remap to once bad blocks are detected. Some bad blocks are pre-mapped during manufacturing.
- **Head crash**
Physical contact of the read/write head with the platter. Tends to gouge out a long strip of material from the platter surface.
- **Servo / controller failure**
MTTF is typically 20 - 100 years (1M hrs) for *single*, well manufactured drives. Assumes perfect environment, minimal power cycling, vibration, electrical noise, humidity, etc.

Redundant Array of Inexpensive Disks (RAID)

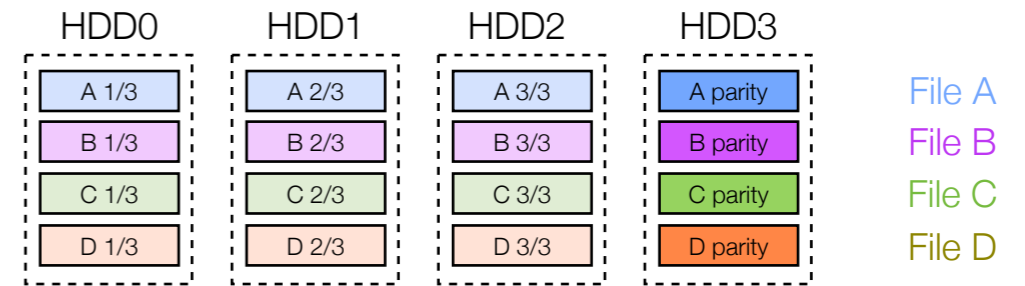
- Combine individual, small, slow, unreliable physical drives into a single large, fast, reliable virtual drive.
- Many possible configurations.
 - Some provide additional throughput but reduce reliability.
 - Some provide additional reliability, but require extra hardware.
- RAID can be implemented in hardware or in software by the OS kernel. Kernel RAID modes usually limited to levels 0 and 1.
- Combined with 'hot swapping' hardware support, drives can fail, be replaced and re-initialized without the DBMS noticing.

RAID-0, data striping



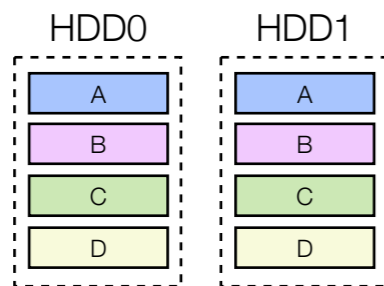
- Write single file across multiple physical disks.
- Read and write throughput increased times number of disks in array.
- Reliability *decreased*. More points of failure compared with single disk.

RAID-3, single parity drive



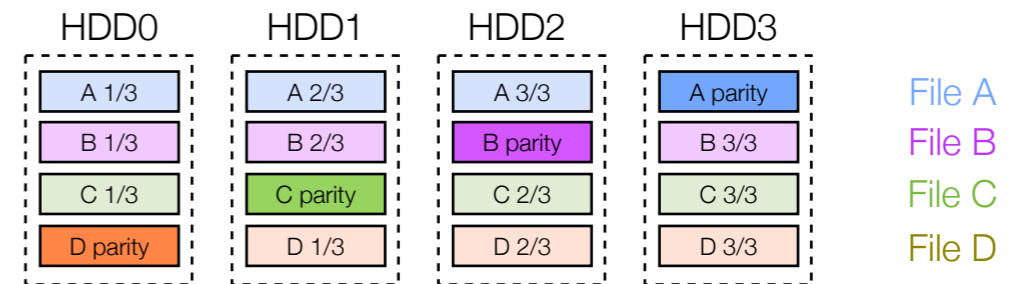
- Parity information is calculated from data and stored on additional disk.
- In the event of a single disk failure, data can be reconstructed from other disks by using parity.
- Single parity disk is a bottleneck as all writes must also update parity.

RAID-1, mirrored



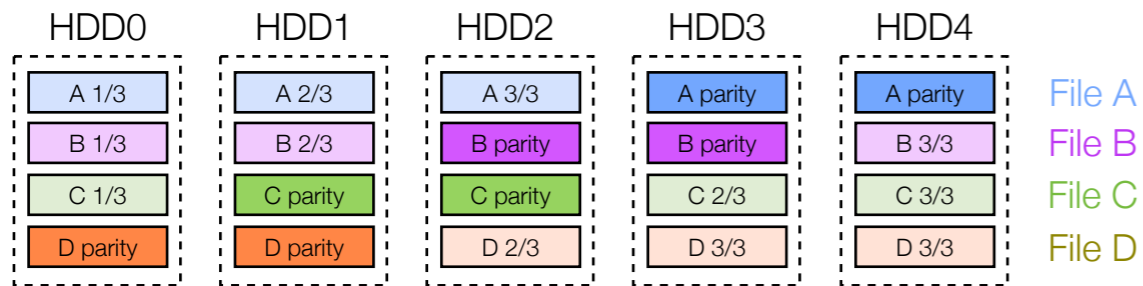
- Duplicate data on multiple physical disks
Requires twice as many disks.
- Reliability *increased* (not including possible RAID controller failure etc)
Easy to rebuild other disk after failure.
- Potential to reduce access time by locking rotation phase of second disk to plus 1/2 rotation compared with first.

RAID-5, distributed parity



- Distributing parity among all disks spreads load evenly.
- As in RAID-3, failure of one drive renders array vulnerable until it is rebuilt.

RAID-6, dual distributed parity



- Array tolerates up to two failed disks.
Protects data while first failed drive is being rebuilt.
- Similarly aged disks are likely to fail in clusters.
- Can be expanded to larger array sizes.

Tape / DVD / offsite server

- Sequential access tapes are slow but have high capacity and can be re-used many times. Writable DVD is also used for incremental backups.
- Common to use 7 day cycle of removable media to ward against media failure, loss, accidental overwrite etc.
- Can also replicate DB via network to offsite machine.
Does not protect against hackers and disenfranchised staff.



in q4 2008
Sony SDX1100VRB
400 GB/tape (native)
26 MB/s transfer
AUD\$3959,
Media is approx AUD\$50 /each

The importance of removable media

- Allows data to be physically removed from DB server and work place.
- Hail storms, snow storms, tornados, tsunamis, falling trees, errant vehicles, water leaks, lightning, power spikes, spilt coffee etc.
All can destroy physical hardware, and data along long with it.
- DBMS bugs, OS bugs, HDD firmware bugs, outdated software, mistaken commands, disenfranchised staff, hackers etc.
All can destroy data on perfectly functioning hardware.
- You will loose data due to one of the above.
- RAID will not save you.

When good times go bad.

- Backing up data is not enough.
- You must also practice recovering it, assuming total failure.
 - Can you get spare hardware in time?
 - Do you have *current* copies of DBMS software?
 - Do you remember how to configure it?
 - Do you have *current* schema for all the tables?
 - Do you have scripts to reconstruct the tables?
 - Will client applications deal with the fresh DB?
 - Will management tools deal with glitches in the logs?
 - Will you be able to pull it together when you've been awake for 72 hrs, and your job (pride?) is on the line?
- How would you cope if the data center is flooded tomorrow?
All you have left is the tape...