

Storing Data: Disks and Files

(From Chapter 9 of textbook)

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Storing and Retrieving Data

- Database Management Systems need to:
 - Store large volumes of data
 - Store data reliably (so that data is not lost!)
 - Retrieve data efficiently
- Alternatives for storage
 - Main memory
 - Disks
 - Tape

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Why Not Store Everything in Main Memory?

- *Costs too much.*
- *Main memory is volatile.*

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Why Not Store Everything in Tapes?

- *No random access.*
- *Slow!*

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Disks

- Secondary storage device of choice



- Main problem?

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Solution 1: Techniques for making disks faster

- Intelligent data layout on disk
- Redundant Array of Inexpensive Disks (RAID)

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Solution 2: Buffer Management

- Keep “currently used” data in main memory
- Typical (simplified) storage hierarchy:

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Outline

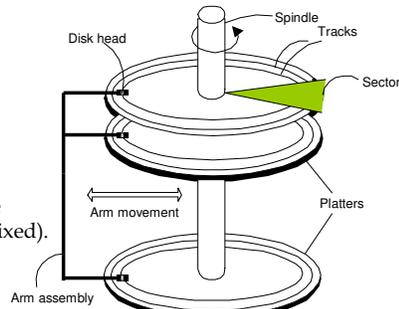
- Disk technology and how to make disk read/writes faster
- Buffer management
- Storing “database files” on disk

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Components of a Disk

v Only one head reads/writes at any one time.

v *Block size* is a multiple of *sector size* (which is fixed).



Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Accessing a Disk Page

- Time to access (read/write) a disk block:
 - Seek time:** 1 to 20msec
 - Rotational delay:** 0 to 10msec
 - Transfer rate:** ~ 1msec per 4KB page
- Key to lower I/O cost: reduce seek/rotation delays!

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Arranging Pages on Disk

- `Next'* block concept:
 - Blocks in a file should be arranged sequentially on disk (by *`next'*), to minimize seek and rotational delay.

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

In-Class Exercise

- Consider a disk with:
 - average seek time of 15 milliseconds
 - average rotational delay of 6 milliseconds
 - transfer time of 0.5 milliseconds/page
 - Page size = 1024 bytes
- Table: 200,000 rows of 100 bytes each, no row spans 2 pages
- Find:**
 - Number of pages needed to store the table
 - Time to read all rows sequentially
 - Time to read all rows in some random order

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

In-Class Exercise Solution

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

RAID (Redundant Array of Independent Disks)

- Disk Array: Arrangement of several disks that gives abstraction of a single, large disk.
- Goals: Increase **performance** and **reliability**.
- Two main techniques:
 - Data striping
 - Redundancy

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

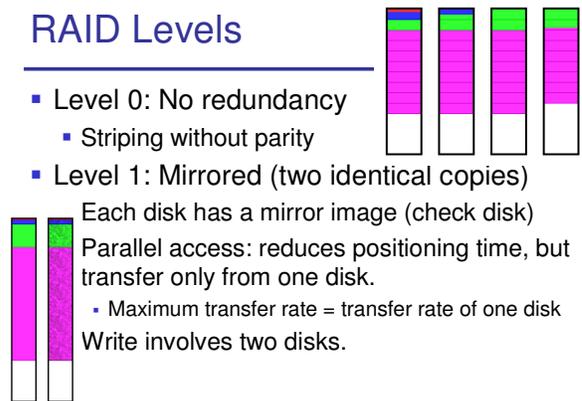
Parity

- Add 1 redundant block for every n blocks of data
 - XOR of the n blocks
- Example: D1, D2, D3, D4 are data blocks
 - Compute DP as $D1 \text{ XOR } D2 \text{ XOR } D3 \text{ XOR } D4$
 - Store D1, D2, D3, D4, DP on different disks
 - Can recover any *one* of them from the other four by XORing them

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

RAID Levels

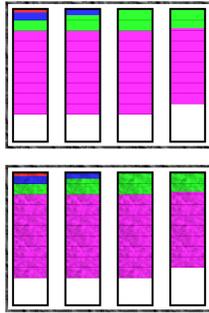
- Level 0: No redundancy
 - Striping without parity
- Level 1: Mirrored (two identical copies)
 - Each disk has a mirror image (check disk)
 - Parallel access: reduces positioning time, but transfer only from one disk.
 - Maximum transfer rate = transfer rate of one disk
 - Write involves two disks.



Database Management Systems, R. Ramakrishnan and Johannes Gehrke

RAID Levels (Contd.)

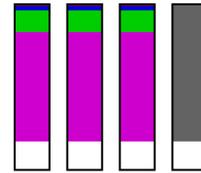
- Level 0+1: Striping and Mirroring
 - Parallel reads.
 - Write involves two disks.
 - Maximum transfer rate = aggregate bandwidth
 - Combines performance of RAID 0 with redundancy of RAID 1.
- Example: 8 disks
 - Divide into two sets of 4 disks
 - Each set is a RAID 0 array
 - One set mirrors the other



Database Management Systems, R. Ramakrishnan and Johannes Gehrke

RAID Levels (Contd.)

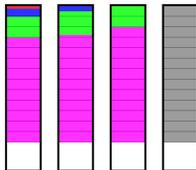
- Level 3: Bit-Interleaved Parity
 - Striping Unit: One bit. One check disk.
 - Each read and write request involves all disks; disk array can process one request at a time.



Database Management Systems, R. Ramakrishnan and Johannes Gehrke

RAID Levels (Contd.)

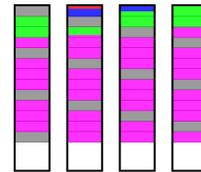
- Level 4: Block-Interleaved Parity
 - Striping Unit: One disk block. One check disk.
 - Parallel reads possible for small requests, large requests can utilize full bandwidth
 - Writes involve modified block and check disk



Database Management Systems, R. Ramakrishnan and Johannes Gehrke

RAID Levels (Contd.)

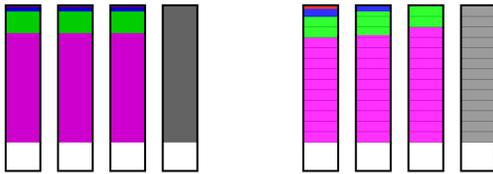
- Level 5: Block-Interleaved Distributed Parity
 - Similar to RAID Level 4, but parity blocks are distributed over all disks
 - Eliminates check disk bottleneck, one more disk for higher read parallelism



Database Management Systems, R. Ramakrishnan and Johannes Gehrke

In-Class Exercise

- How does the striping granularity (size of a stripe) affect performance, e.g., RAID 3 vs. RAID 4?



Database Management Systems, R. Ramakrishnan and Johannes Gehrke

In-Class Exercise Solution

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Which RAID to Choose?

- RAID 0: great performance at low cost, limited reliability
- RAID 0+1 (better than 1): small storage subsystems (cost of mirroring limited), or when write performance matters
- RAID 3 (better than 2): large transfer requests of contiguous blocks, bad for small requests of single blocks
- RAID 5 (better than 4): good general-purpose solution

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Which RAID to Choose? Corrected.

- RAID 0: great performance at low cost, limited reliability
- RAID 0+1 (better than 1): small storage subsystems (cost of mirroring limited), or when write performance matters
- RAID 5 (better than 3, 4): good general-purpose solution

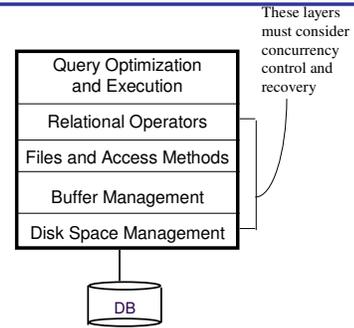
Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Disk Space Management

- Lowest layer of DBMS software manages space on disk.
- Higher levels call upon this layer to:
 - allocate/de-allocate a page
 - read/write a page
- Request for a *sequence* of pages must be satisfied by allocating the pages sequentially on disk! Higher levels don't need to know how this is done, or how free space is managed.

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Structure of a DBMS



Database Management Systems, R. Ramakrishnan and Johannes Gehrke