Storage

Anne Denton

Department of Computer Science
North Dakota State University

Outline

Table of Contents

Types of storage

- Technologies
  - Primary storage (electronic): Registers, cache memory, static RAM, dynamic RAM
  - Secondary storage (magnetic, optical): Magnetic hard drives (HDD), solid state drives (SSD, CD, DVD
  - Tertiary storage: Tapes, tape juke boxes
- Distinctions
  - Online vs. offline
  - Volatile vs. nonvolatile
  - Random access vs. sequential access

Capacity and size

- Capacity

kilobyte	megabyte	gigabyte	terabyte	petabyte	exabyte
KB	MB	GB	TB	PB	EB
10^3, strictly 2^{10}	10^6	10^9	10^{12}	10^{15}	10^{18}

- Size

millisecond	microsecond	nanosecond	picosecond	femptosecond
ms	μs	ns	ps	fs
10^{-3}	10^{-6}	10^{-9}	10^{-12}	10^{-15}

Storage technology from fastest to slowest

Registers	Used directly for computations; electronic; part of processor
Cache	Faster than memory; short distance from processor
Memory	Primary storage; access nanoseconds
SSD	Secondary storage; access microseconds; may be used for caching between magnetic disk and memory
Magnetic:	Conventional hard drive (HDD); access times of the order of milliseconds
Optical:	CD or DVD, using audio / video technology; access time around 1s
Tape:	Rarely used now and, if so, usually only for backup; access time of the order of minutes

Buffering

- Mismatch in speeds typically bridged using buffering techniques
- Double buffering: One buffer used for I/O, the other for processing
- Buffering used between any level in the storage hierarchy and the level above/below
- SSDs can be used for buffering between RAM and HDD

Table of Contents

Magnetic drives

Platter: Individual disk of a disk pack ($1 - 12$)

Track: Circle of bits on platter

Cylinder: Collection of corresponding tracks on all platters: All data on one cylinder can be reached without head movement (> 10000)

Tracks: As many tracks per cylinder as surfaces

Sector: Section of track; smallest unit that can be physically addressed (512 bytes or 4KB)

Block: Smallest addressable unit, depends on formatting (usually 4KB, but can be between 1 and 8 sectors)

Clusters: Larger units can be used additionally

Properties of magnetic drives

- Latency
  - Seek time: Time for read-write head to be positioned on the correct cylinder
  - Rotational latency: Time until platter has rotated to starting point of sector
    - 4.17ms for a spindle speed of 7200 RPM
    - 2 ms for a spindle speed of 15000 RPM
  - Make magnetic hard drives slower to respond than RAM by a factor of $10^6 - 10^7$
- Bulk transfer rate: Transfer rate after first block is located
  - Slower than SSD but not by as much

Question 1

The latency of magnetic hard drives is higher than of RAM by a factor of about

1. 10
2. 100
3. 1000
4. 1000000

Algorithms for external / secondary storage

- Algorithms to work with secondary / external storage have to address
  - Much larger latency
  - A block (typically 4 KB) is read at once
- Data structures organized around block size
  - B+ trees have block-sized nodes
  - External hash files have block-sized bucket
  - External sorting is organized around blocks

Fragmentation

- Fragmentation
  - Storage of files broken up across multiple non-contiguous blocks
  - Defragmentation software can restore performance
- Internal fragmentation
  - Space between end of file and end of block is wasted
  - Can increase storage substantially when storing a large number of tiny files
  - The Unix tar command combines directories into files, thereby recovering the lost space
  - The Windows zip command includes tar and compression

Question 2 (Multiple answers can be correct)

Defragmentation

1. Can reduce internal fragmentation
2. Can reduce delays due to rotational latency of an HDD
3. Can reduce delays due to a low bulk transfer rate of an HDD
4. Can reduce storage requirements of files

Question 3 (Multiple answers can be correct)

The Unix tar command

1. Can reduce fragmentation
2. Can reduce internal fragmentation
3. Can reduce storage requirements of files

SSDs

- Storage electronic; no moving parts
- Typically similar technology to flash drives
- Concept of blocks as addressable units like HDD
- Speed intermediate to HDDs and SRAM: Electronic like SRAM, but greater distance to CPU
- No benefit of storing files in contiguous blocks
- Can be used to bridge performance gap with regard to HDDs

Question 4 (Multiple answers can be correct)

SSDs differ from magnetic hard drives in

1. That they have no moving parts
2. That they have lower latency
3. That they typically have more storage capacity

Table of Contents

Beyond a single machine / single disk

- Combining disks into RAID arrays
  - Improves throughput
  - Can introduce redundancy
- Virtualization of systems and storage
  - Many common systems run as virtual machines on virtualized hosts
  - Typically have internal ephemeral storage
  - External persistent block storage is typical for database storage
  - Object storage for large objects

RAID Arrays

- Redundant array of inexpensive disks
- Appear as one logical disk
- Can be internal to a machine or network-attached
- Increases capacity and can increase access speed (depends somewhat on RAID level)
- Problem: Mean Time To Failure (MTTF) of n disk is 1/n that of each individual disk
- Simplest solution: Mirroring (RAID 1), i.e. 2 identical copies of the same data
- Higher RAID levels: Error correction
- Block-level striping is typical but other strategies exist
- Do not use more than about 10 disks per array, due to risk of additional failure during recovery

Question 5 (Multiple answers can be correct)

RAID 1, i.e. mirroring, in comparison with a single disk typically

1. reduces the bulk transfer rate for read operations
2. does not increase bulk transfer rate for write operations
3. reduces the probability of losing data due to disk failure

RAID Levels

- **RAID 0** No redundancy; usually higher performance than individual disk
- **RAID 1** Mirroring, i.e. exact copies; helps read performance but not write performance; storage efficiency poorer than some other levels
- **RAID 5** Block-level striping; parity information distributed among drives; can handle failure of single disk; helps read performance but not write performance
- **RAID 6** Block-level striping; Reed-Solomon codes extend concept of parity bits, such that array can handle failure of two disks; read performance like RAID 5 but write performance poorer
- **RAID 10** 1+0, i.e., combination of mirroring and striping

Question 6 (Multiple answers can be correct)

Using RAID 10 rather than RAID 6

1. Allows faster reads
2. Means that fewer disk are needed to guarantee that 2 disks can fail without data loss
3. Decreases the probability of failures during recovery

Network attached storage (NAS)

NFS (Network File System): Non-distributed file server, multiple hosts access it, e.g., for home directories in lab

SAN (Storage Area Network): Typically using block storage model

Distributed storage: Many connected disks, often use object storage model

Block storage

- Conventional RDBMSs require block storage
- Architecture of file system maintained, even when storage remote
- For production systems, storage typically on separate storage servers
- Storage servers typically house many RAID arrays
- Block storage is also available in public clouds
  - Example Amazon EBS (Elastic Block Store)
  - Fast RAIDed storage given out to virtual machines in virtualized environments
- Parallelization through clustered file system possible but not always available

Distributed storage

- Many applications have been developed that work with different architectures
  - Object storage
  - Hadoop and other distributed file systems
- Does not use RAID arrays but rather multiple replicates of larger granularity
- Example: Amazon S3
- Not suitable for conventional RDBMSs
- NoSQL databases have been built on Hadoop among others
  - HBase is a key/value store
  - Hive allows SQL-like queries

The cloudiness of cloud terminology

- What makes a cloud be a cloud?
  - Maybe virtual machines with block storage?
  - Maybe distributed applications for which processing and storage are distributed in a way that is hidden from the user?
- "Cloud" is a political term
- Use more precise terms, for example
  - Infrastructure as a Service (IaaS): Virtual machines
  - Platform as a Service (PaaS): Services to build and deploy apps
  - Software as a Service (SaaS): Applications that run in web server and backend is distributed

CS Department "Cloud"

- Virtualized with with storage servers that offer block storage
- Hadoop runs distributed on the lab machines

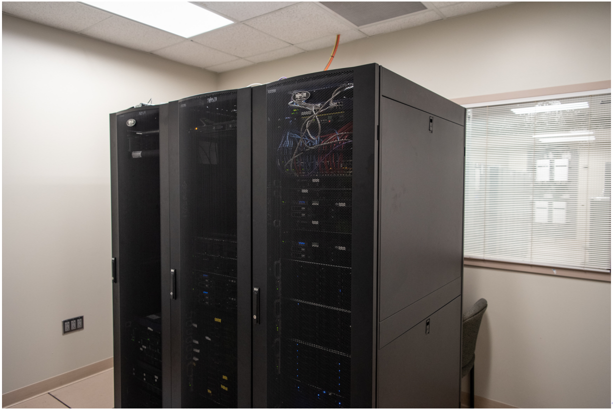

Table of Contents

Text vs. binary

- Text files
  - Humanly readable
  - Text format not storage efficient, especially for numerical data
  - ASCII uses one byte per character, Unicode two
  - Common for transfer between databases
- Binary files
  - Not humanly readable
  - Format differs by data type
  - Typical for database backend

Delimited text

- Most basic text format
  - Delimiter may be comma (.csv)
  - Tab as delimiter allow commas in fields without requiring quotes or escape characters; easier to process from a program
  - From a standardization perspective, XML or JSON preferable
  - If a single table is being represented, delimited text has benefit of simplicity
  - Many programs can read it, e.g., Python pandas library, and spreadsheet applications
- Identified fields, that have field label ahead of each data element, allow more flexibility
  - Standard formats like XML or JSON preferable when flexibility needed

XML, JSON

- For semi-structured text data
- Followed from the goal of representing data in a form similar to HTML
- Multiple alternate ways of defining a schema
  - Document Type Definition (DTD)
  - XML Schema
- Inherently hierarchical rather than relational
- Can be queried but not very efficiently

Standardization

- In programming, try to use standard formats because of availability of libraries!
  - For record storage, XML and JSON common current formats
  - Delimited text not as standardized but has less overhead
  - Many legacy formats that are less than ideal but sometimes unavoidable (e.g., in bioinformatics)
- Databases have SQL standardized interface
  - Allows DBMS to choose non-standard format at backend while still maintaining interoperability
  - Some databases allow extracting database state as SQL statements (sqldump)
  - Some databases allow conversion to XML

Question 7 (Multiple answers can be correct)

Which of the following are standardized (allowing for multiple standards)

1. The format of backend tables in RDBMSs
2. The language for querying RDBMS tables
3. The format of XML storage

Object streams for persistent storage from programs

- Create some list of all objects that should persist, e.g., HashSet
- Very easy to implement
- Use writeObject and readObject
- Objects must implement "Serializable"
- Object definition must not change between program executions
- All objects from a stream must be read back in
- No random access object streams
- Inefficient when only some of the objects are required

Question 8 (Multiple answers can be correct)

Which of the following are humanly readable?

1. XML
2. Delimited text
3. RDBMS table
4. Output of Java Object stream

Table of Contents

Streams

- Application programs commonly view files as streams
- Analogy to telephone switching system
- Streams are logical files
- Standard I/O and standard error are treated as logical files
- Operating system links logical file with physical file
- In Java: java.io

Sequential vs. random access

- Do not confuse with sequential vs. random access storage media! (Sequential storage media largely irrelevant now)
- Reading XML or JSON is normally sequential
- Random access easier to implement for fixed-length records
    - Use “seek" function in programming language
    - Use case for random access would be data-intensive single-user application, e.g. science or engineering
    - Increasingly many use cases solved with database technology

Question 9 (Multiple answers can be correct)

Which of the following can be opened for sequential access?

1. File
2. Keyboard input
3. Terminal output (stdout)
4. Output of application errors (stderr)

Question 10 (Multiple answers can be correct)

Which of the following can be opened for random access?

1. File
2. Keyboard input
3. Terminal output (stdout)
4. Output of application errors (stderr)

Files vs. databases

- Whenever multiple programs or multiple users need access, consider using one of the following:
  1. Files
  2. SQL as standardized interface with RDBMS
  3. A NoSQL database
- When ability to exchange data matters, choose 1
  - Try to use standard format
- For general flexibility, choose 2
  - If application size matters pick lightweight alternative (e.g. sqlite)
- When distribution across network and millions of records, choose 3

Table of Contents

Variable length vs. fixed length

- Even one variable length field makes records variable length records
- Some DBMSs use fixed-length storage, although PostgreSQL does not
- Variable length organization common in humanly readable formats
  - Delimited text (e.g., tab-delimited or .csv)
  - XML, JSON
- Benefits of fixed length
  - Faster searching
  - More efficient insertions and deletions
- Benefits of variable length
  - More storage efficient
  - Note that having files be humanly readable may defeat that

Question 11

In each of the following alternatives, which one has the potential of being more storage efficient?

1. Text vs. binary
2. Fixed-length vs. variable length records

Organization of rows

- Heap file
  - Unsorted, new records added at end
  - Insertion and deletion fast
- Sorted file
  - File sorted according to search key
  - Good for search based on equality or inequality, especially when combined with tree index
- Hash file
  - File organized by hash key
  - Good for search based on equality

Heap file

- No specific ordering
- Records added at end
- Deletions may be filled by newly inserted records (or left open until reorganization)
- Search using linear search O(N)
- Listing in the order of search key requires external sorting O(N log(N))

Ordered file

- Sorted according to search key
- File can only be ordered according to one attribute
- Search key may be key field of database table but does not have to be
- Search could be done using binary search, but disk latency can make that slower than linear search O(log(N)) BUT prohibitive pre-factor!
- Typically used with B+ trees O(log(N))
- Secondary B+ trees can offer logical sorting (discussed later)
- Ideal for listing in order of search key value

Hash file

- Search key mapped to bins, similar to hashtable in memory
- Preferred bin size depends on block size
- File can only be hashed according to one attribute
- No advantage for sorting in order of hash key (requires external sorting just as hash file)

Internal hashing (i.e. in memory)

- Use a table with m slots (index 0 through m-1)
- Apply hash function to hash field, for example,
  - h(k) = k % m
  - Or pick digits, e.g. 3rd, 5th, and 8th digit
  - Goal: distribute values evenly
- Problem: Collisions
  - No guarantee that different values will hash to different addresses
  - Hash field space larger than address space
  - If address already contains record: collision resolution

Collision resolution

- Open addressing: check subsequent positions
- Chaining: Each record location has a pointer, and if multiple records hash to the same address: pointer points to overflow location
- Multiple hashing: The program applies a second (even third) hash function and eventually use open addressing
- Load factor: number of elements / number of slots
- Performance best for a load factor of 0.7 - 0.9
- When load factors gets too high, a larger hashtable is created and everything is rehashed

External hashing

- Bucket size: Typically the block size
- Gradual rehashing
- Extendible hashing
  - Individual buckets are split whenever full
  - Uses a directory with a depth that increases whenever a larger address space is needed
  - Distinguishes a global and local depth of the directory (not all buckets full at the same time)
- Linear hashing
  - Collision resolution applied, e.g., typically chaining
  - Number of bins increases gradually in a way that only requires splitting one bucket at a time
  - Works if h(k) = k % m is replaced with h(k) = k % (2m) for the bucket that is being split

Question 12

Which of the following returns the result fastest for a search based on equality

1. Heap file
2. Sorted file
3. Hash file

Question 13

Which of the following typically returns the result fastest for a search based on an inequality (e.g. greater than)?

1. Heap file
2. Sorted file
3. Hash file

Question 14

Which of the following allows the fastest insertion?

1. Heap file
2. Sorted file
3. Hash file