CSE 444: Database Internals

Lecture 3
DBMS architecture and deployments

Late Days Policy

- Reminder: Lab 1 Part 1 is due on Monday
  - Go through logistics of getting started
  - Start to make some small changes to the code
  - PLEASE ASK QUESTIONS!
- Late days:
  - 3 late days total
  - Can use in 24 hour chunks at any time
  - NO OTHER EXTENSIONS!
- No late days on web quizzes!

What we already know…

- A DBMS helps companies, organizations, and individuals to manage their data
- By providing capabilities to easily
  - Describe the data (database schema)
  - Load the data
  - Query the data
  - Update the data
  - Etc.
- For same reason, DBMS simplifies development of applications that need to operate on data

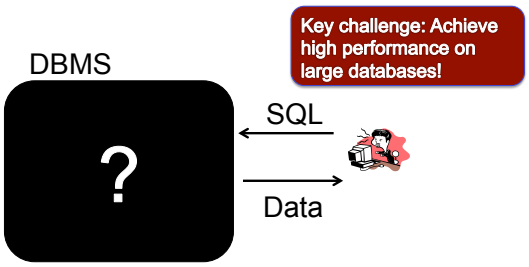
What we already know…

- In 344, we learned about different data models: relational and semi-structured (XML)
- Relational model was proposed in 1970
- Most commonly used model today
- We reviewed the relational model in lecture 2

Benefits of relational model

- Helps provide logical data independence
  - Because data represented with simple structures
  - Can change schema without affecting applications
  - Thanks to views and simple data structure
- Helps provide physical data independence
  - Can change data organization on disk for performance without affecting applications
  - Thanks to set-at-a-time query language
    - Relational algebra

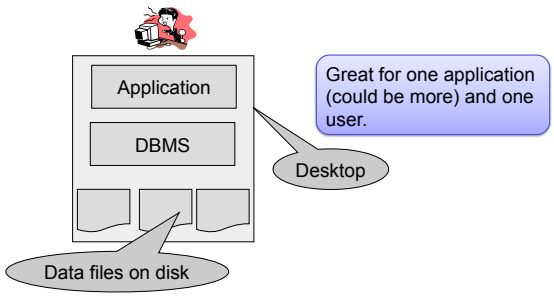
How to Implement a Relational DBMS?



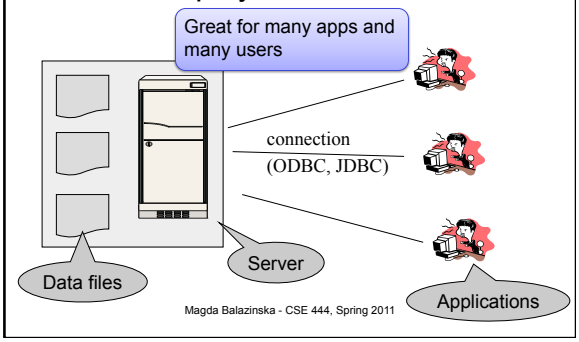
Outline for Today

- **How DBMSs are deployed**
- Overview of DBMS architecture

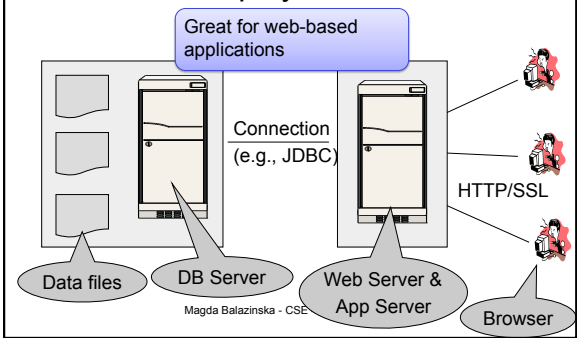
DBMS Deployment: Local



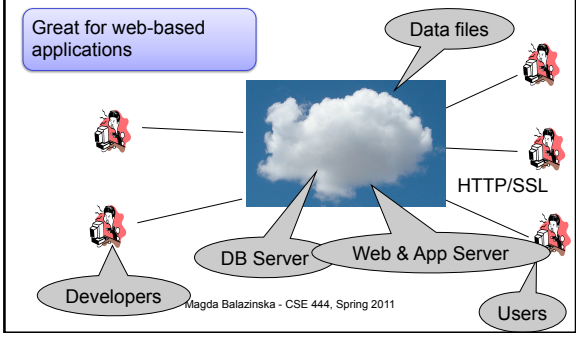
DBMS Deployment: Client/Server



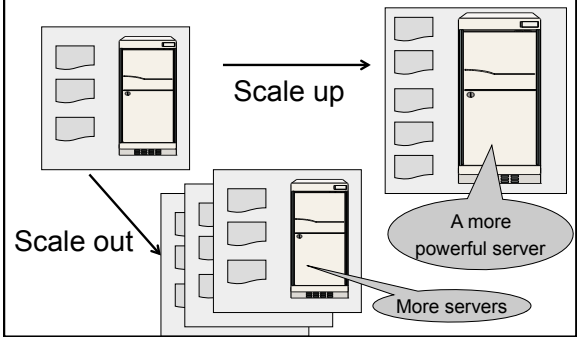
DBMS Deployment: 3 Tiers



DBMS Deployment: Cloud



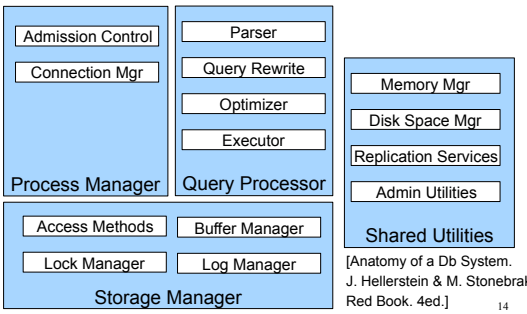
How to Scale a DBMS?



Outline for Today

- How DBMSs are deployed
- **Overview of DBMS architecture**

DBMS Architecture



[Anatomy of a Db System. J. Hellerstein & M. Stonebraker. Red Book. 4ed.]

Query Processor

- **Parser**
  - Parses query into an internal format
  - Performs various checks using catalog
    - Correctness, authorization, integrity constraints
    - Typically, catalog is stored in the form of set of relations
- **Query rewrite**
  - View rewriting, flattening, etc.

Query Processor

- **Optimizer**
  - Find an efficient query plan for executing the query
  - A **query plan** is
    - **Logical**: An extended relational algebra tree
    - **Physical**: With additional annotations at each node
      - Access method to use for each relation
      - Implementation to use for each relational operator
- **Executor**
  - Actually executes the physical plan

Storage Manager

- **Buffer Manager**
  - Caches data in memory
  - Reduces the number of disk IO operations
  - Care is needed to support ACID transactions!
- **Access Methods**
  - Files ("heap files") and indexes
- **Log and Lock Managers**
  - Necessary to support transactions

Process Manager

- **Connection Manager**
  - Process per user or thread per user?
  - Various variants exist, partly for historical reasons
- **Admission Control**
  - To avoid thrashing
  - And provide "graceful degradation" under load
  - Second level of admission control: before running a query

Shared Utilities

- **Memory Manager**
  - Manages memory used by various components: internal operator state, query optimizer, etc.
  - Note: Buffer manager holds only *data*
- **Disk Space Manager**
  - Two basic deployment alternatives:
    - Use "raw" disk device interface directly
    - Use OS files
  - DB file abstraction on top of disk or OS file abstraction

Shared Utilities

- **Replication Services**
  - For increased fault-tolerance
  - Or for increased performance
- **Admin Utilities**
  - Collecting statistics about data for optimizer
  - Re-organize data on disk, build indexes, etc.
  - Backup or export database