CMSC424: Database Design
Introduction/Overview

Professor: Pete Keleher
keleher@cs.umd.edu

Today

- Administrivia
  - Workload etc.

- Motivation: Why study databases? What is databases?

- Current Industry Outlook

- A typical DBMS at a glance
Logistics

- Professor: Peter Keleher
  - 5146 Iribe Bldg
  - keleher@cs.umd.edu
  - Class Webpage:
    - http://sedna.cs.umd.edu/424

- Communication:
  - Piazza
  - Office hours
  - Email to me: include CMSC424 in subject as a last resort.
  - Do not message me on ELMS, I will not read it.

- Laptops and mobile devices are prohibited in class.

Logistics

- Grading
  - Whole class is curved: avg is B min, stdev up or down for A, C
  - Approximate cut-offs last year (not guaranteed)
    - 80+: A
    - 70+: B
    - 60+: C
    - 60-: D/F
  - Most had 40+ on non-exams last two times (out of 50)
    - Exams are usually somewhat harder (no curves)
    - Must average a passing grade on the total exam score
Some To-Dos

- Sign up for Piazza!

- Set up the computing environment (project0), and make sure you can run Vagrant+VirtualBox, PostgreSQL, IPython, etc.

- Upcoming: Reading Homework 1, Project 1: SQL

- Upcoming:
  - Reading Homework 1 (Due Tuesday)
  - Project 1: SQL (Feb 10)
**Motivation: Data Overload**

- Explosion of data, in pretty much every domain
  - Sensing devices and sensor networks that can monitor everything 24/7 from temperature to pollution to vital signs
  - Increasingly sophisticated smart phones
  - Internet, social networks makes it easy to publish data
  - Scientific experiments and simulations produce astronomical volumes of data
  - Internet of Things
  - **Dataification**: taking all aspects of life and turning them into data (e.g., what you like/enjoy turned into a stream of your "likes")

- How to handle that data? How to extract interesting actionable insights and scientific knowledge?

- Data volumes expected to get much worse

---

**Four V’s of Big Data**

- **Increasing data Volumes**
  - **Scientific data**: 1.5GB/genome -- can be sequenced in .5 hrs; LHC generates 100TB of data a day
  - 500M tweets per day (as of 2013)
  - As of 2012: 2.5 Exabytes of data created every day
  - EBay: Two data warehouses with 7.5PB and 40PB
  - Walmart: 583 terabytes of sales and inventory data
  - FICO monitors 2.5 billion active accounts worldwide

- **Variety**: Structured data, spreadsheets, photos, videos, natural text, ...

- **Velocity**

- **Veracity**
Four V’s of Big Data

- Increasing data Volumes
- Variety
- Velocity
  - Sensors everywhere -- can generate tremendous volumes of "data streams"
  - Real-time analytics requires data to be consumed as fast as it is generated
- Veracity
  - How do you decide what to trust? How to remove noise? How to fill in missing values?

Big Data and Data Science to the Rescue

- Terms increasingly used synonymously: also data analytics, data mining, business intelligence
  - Loosely used for any process where interesting things are inferred from data
  - Google search: “How Big Data Will Change”
- Data scientist called the sexiest job of the 21st century
  - The term has becoming very muddled at this point
- Overhyped words
  - We are headed toward the trough of Disillusionment
Is it all hype?

- No: Extracting insights and knowledge from data very important, and will continue to increase in importance
  - Big data techniques are revolutionizing things in many domains like Education, Food Supply, Disease Epidemics, ...
- But: it is not much different from what we, especially statisticians, have been doing for many years
- What is different?
  - Much more data is digitally available than was before
  - Inexpensive computing + Cloud + Easy-to-use programming frameworks = Much easier to analyze it
  - Often: large-scale data + simple algorithms > small data + complex algorithms
    - Changes how you do analysis dramatically

Motivation: Data Overload

- How do we do anything with this data?
- Where and how do we store it?
  - Disks are doubling every 18 months or so -- not enough
  - In many cases, the data is not actually recorded as it is; summarized first
- What if the disks crash?
  - Very common, especially with 10,000’s of disks
- How do we ensure “correctness”?
  - What if the system crashes in the middle of an ATM transaction?
    - Can’t have money disappearing
  - What happens when a million people try to buy tickets to <your favorite artist>’s concert at the same time?
Motivation: Data Overload

What to do with the data? How to process/analyze it?
- text search?
  - Very limited
- “find the stores with the maximum increase in sales in last month”
  - We can’t expect the users to write Java programs
- “how much time from here to Pittsburgh if I start at 2pm?”
  - Data is there; more will be soon (GPS, live traffic data)
  - Requires predictive capabilities
- Increasing need to convert “information” to “knowledge”: Data mining
  - “How many DVDs should we order?” (Netflix)
  - Find videos with this type of an event (say car break-ins)
  - Mine the “blogs” to detect “buzz”

Speed !!
- With TB’s of data, just finding something (even if you know what), is not easy
  - Reading a file with TB of data can take hours
- Imagine a bank and millions of ATMs
  - How much time does it take you to do a withdrawal?
  - The data is not local

How do we guarantee the data will be there 10 years from now?

Privacy and security !!!
- Every other day we see some database leaked on the web
  - Identity fraud, influencing elections...
- How to make sure different users’ data is protected from each other
Why not use file systems?

- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Integrity constraints (e.g., account balance > 0) become “buried” in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones
  - Atomicity of updates
    - Failures may leave database in an inconsistent state with partial updates carried out
    - Example: Transfer of funds from one account to another should either complete or not happen at all
  - Concurrent access by multiple users
    - Concurrent access needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
      - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
  - Security problems
    - Hard to provide user access to some, but not all, data