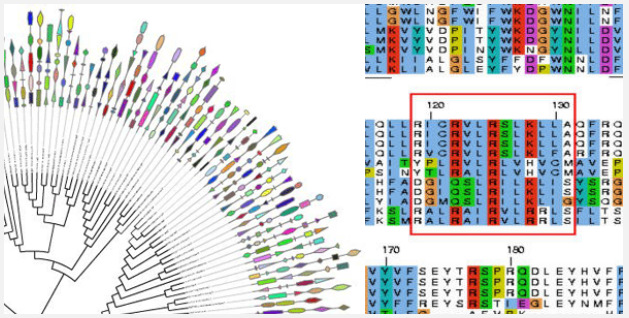
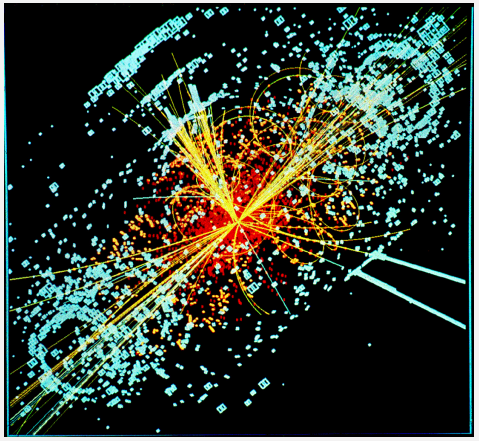
CS 564: DATABASE MANAGEMENT SYSTEMS

Fall 2015

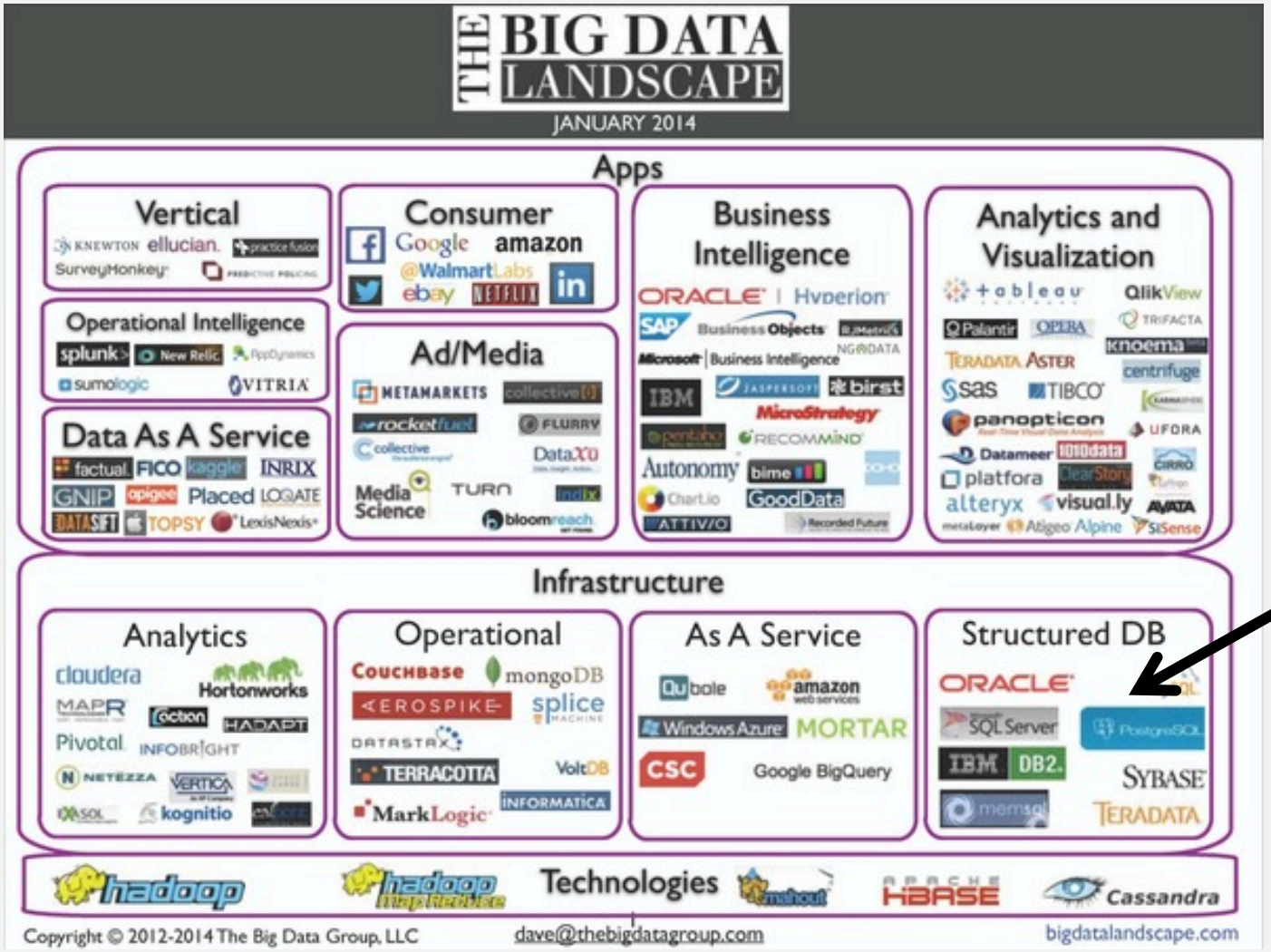
What is This Class about?



- Data is everywhere!
- Managing data is critical:
  - scientific discoveries
  - online services (social networks, online retailers)
  - decision making
- **Databases** are the core technology
- In this class:
  - How do we use a database?
  - How to we build a database?



DATA LANDSCAPE



COURSE LOGISTICS

TEACHING STAFF

- **Instructor**: Paris Koutris
  - paris@cs.wisc.edu
  - Office hours: Monday 1-2pm, Thursday 2-3pm

- **TA**: Kavin Mani (Section 1)
  - Office hours: Monday 11:15-12:15
- **TA**: Apul Jain (Section 2)
  - Office hours: Tuesday 1-2pm

About Me

- undergrad in Athens, Greece
- Ph.D. in University of Washington (the other UW)
- at UW-Madison since 2015!

Research Interests

- parallel processing of big data
- data pricing
- uncertainty in data management

COURSE FORMAT

- Lectures **M+W** 2:30-3:45 pm
- Discussions **F** 2:30-3:45 pm

- 4 Projects (in groups of 2)
- 2 Homework assignments (individual)

- Midterm Exam
- Final Exam

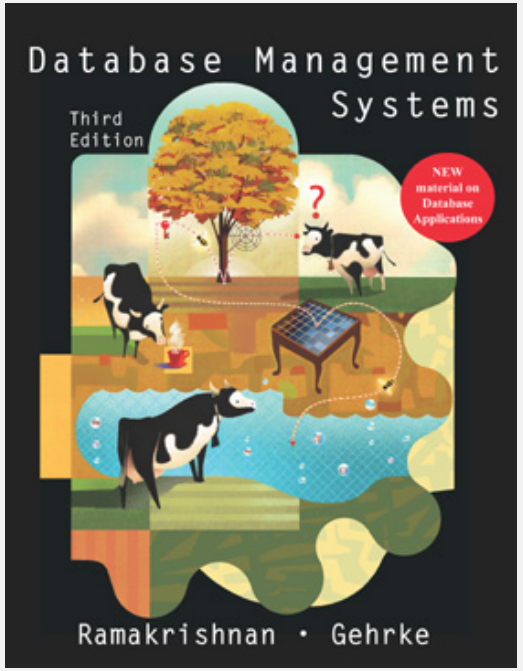
COMMUNICATION

- **Webpage**: http://pages.cs.wisc.edu/~paris/cs564-f15
  - Announcements
  - Lectures
  - Assignments
- **Mailing List**: compsci564-1-f15@lists.wisc.edu
- **Piazza**: https://piazza.com/wisc/fall2015/compsci564_fa15/home
  - Questions
  - Discussions

TEXTBOOK

- Database Management Systems (3d edition)

  Raghu Ramakrishnan and Johannes Gehrke



Come to class!!

PREREQUISITES

- Data structures and algorithm background necessary!
  - **CS 367** is a must

- For the projects
  - programming-heavy
  - C++ will be used for the database internals

GRADING

- Projects (4): 30%
- Homework (2): 10%
- Midterm: 25%
- Final: 35%

PROJECTS

- **Project #1**
  - Designing a database (ER-model, schema)
- **Project #2**
  - Querying a database (SQL)
- **Project #3**
  - Implementing a database (Buffer management)
- **Project #4**
  - Implementing a database (B+ trees)

EXAMS

- **Midterm Exam**
  - *when*: October 23 (2:30-3:45pm)
  - *where*: in class

- **Final Exam**
  - *when*: December 19 (7:25-9:25pm)
  - *where*: TBD

DATABASES: A SHORT INTRO

DATABASE

What is a database?

- A collection of files storing related data

What are some examples of databases?

- payroll database
- Amazon's product information
- bank account database

DBMS

*What is a Database Management System (**DBMS**)?*

- A program written by someone else that allows us to manage efficiently a large database and allows data to persist over long periods of time

What are examples of DBMSs?

- SQL Server, Microsoft Access (Microsoft)
- DB2 (IBM)
- Oracle
- MySQL, PostgreSQL, SQLite

EXAMPLE: ONLINE BOOKSTORE

- What data do we need to store?
- How will we use the data stored?

EXAMPLE: ONLINE BOOKSTORE

- What functionality do we want to support?
  - efficient querying
  - multiple users
  - recovery after crashes
  - security, user authorization

DATA STORAGE

- Data stored for a long period of time (persistent data): *the data outlives the application*
- Large amounts of data (100s of GB)
- User authorization on which data to access
- Protection from system crashes

QUERIES & UPDATES

- Store and retrieve data in an efficient way
  - Organize data on disk
  - Index data for faster access
- Make efficient use of memory hierarchy
- Safely allow concurrent access to the data
- Allow the data to be updated safely

CONCURRENCY CONTROL

- Alice and Bob have the same number for a gift certificate of $100 at the online bookstore
  - Alice @ her office orders ”Book A” for $30
  - Bob @ his office orders ”Book B” for $60

- Questions:
  - What is the ending credit?
  - What if second book costs $80?
  - What if system crashes?

CRASH RECOVERY

- How do we make sure no data is lost after the system has crashed?

Schema Change

- Say that we need to add a new field to books
  - entails changing file formats
  - need to rewrite virtually all applications

What Can a DBMS Do?

- All the above!!
- Automate a lot of boring operations on data
  - don't have to program over and over
  - can write complex data manipulations in just a few lines
- Make execution very fast
  - scales up to very large data sets
- Make concurrent access/modification possible
  - many users can use the data at the same time

KEY CONCEPTS

- **Data model:** abstraction that describes the data
- **Schema:** describes a specific database using the "language" of the data model
- **Query Language:** high-level language to allow a user to pose queries easily
  - Declarative languages (SQL)
- **Query optimizer/compiler:** code that evaluates the query efficiently

DATA INDEPENDENCE

The application does not change when the underlying data structure or storage changes

- Physical independence: can change how data is stored on disk without maintenance to applications
- Logical independence: can change schema without affecting applications

Relational Model

- The data is stored in tables (relations in the mathematical sense)
- A database is a set of tables

schema

bookID	title	author	hardcover
007456	The Da Vinci Code	Dan Brown	yes
909405	Ender's Game	Orson Scott Card	no
…	…	…	…

record/tuple

QUERYING THE DATA

- SQL or other declarative languages
- Example: *find all books written by Dan Brown*

```
SELECT *
FROM books B
WHERE B.author = “Dan Brown”
```

QUERY PROCESSOR

- **Optimizer**: what is the best imperative execution plan for the given query?
- **Evaluation**: execute the plan as efficiently as possible

PEOPLE

- **DB application developer**: writes programs that query and modify data
- **DB designer**: establishes schema
- **DB administrator**: loads data, tunes system, keeps whole thing running
- **Data analyst**: data mining, data integration
- **DBMS implementor**: builds the DBMS

COURSE CONTENT

Design & Modeling Data:

- Entity-Relationship model
- Relational model, schema normalization

Querying the Data:

- Relational algebra
- SQL

Database Internals:

- Data storage, file organization, buffer management
- Indexes
- Relational operators, query optimization

Transactions, Big Data

Interested in More?

CS 764

- gory details on how a DBMS works
- transactions/concurrency/internals

CS 784

- newer types of data and how to manage them
- data integration