Spring 2017

CS 564: DATABASE MANAGEMENT SYSTEMS

Teaching Staff

- Instructor: Jignesh Patel,
  Office Hours: Mon, Wed 9:15-10:30AM, CS 4357
- Class: MWF 8:00-9:15 AM
  [Need to attend all three sessions each week!]

- Ankur Srivastava
- Udip Pant
- Vishnu Lokhande

Instructor email list: cs564-s17@cs.wisc.edu

Web Page: http://pages.cs.wisc.edu/~jignesh/cs564/

Course Outline

- Exams: 60%
  - Mid-term on March 15 (in class): 20%
  - Final exam on May 10 (cumulative): 35%
  - Two in-class quizzes: 5% total
- Projects: 40%
  - C++ assignment (out now): 5%
  - Two BadgerDB Assignments: 23%
    - Buffer Manager: 10%
    - B+-tree: 13%
  - Two Assignments (SQL + ML): 12%

Tentative

All assignments are individual assignments.

Most assignments are in C++.

No late days!

Course Contents

- Database management systems – "under the hood" perspective
  - Algorithms, data structures, storage organization, that make data management systems work
- How to use a database system
  - A smaller focus of this course

- Textbook: "Database Management Systems," by Raghu Ramakrishnan and Johannes Gehrke

Database Management System (DBMS)

- A DBMS ***manages*** a ***database***.
- A database is a collection of data, usually with some description of the structure of the data.
  - Structure description, if present, is described using a ***schema***. e.g. the CREATE TABLE command in SQL

Data Storage Management

- Store and retrieve data in an efficient way
  - Organize data in blocks called pages on disk
  - "Index" data for efficient retrieval
- Make efficient use of memory hierarchy
  - Cache frequently used data in a main memory **buffer pool**
- Safely allow concurrent access to the data
- Make sure updates are "committed"

i.e. provide transactional semantics

Describe and Query the Data

Describe the data

- **Data model** is the abstraction to describe the data
- A **schema** describes a specific database using the "language" of the data model
  - E.g. In the relational data model the CREATE TABLE command is used to express the schema of a table

Query the data

- Provide a high-level language to allow a user to pose queries easily
  - Declarative languages are preferred, e.g. SQL
- Need **query processing algorithms** to evaluate the query and techniques to **optimize** the query

Data Management Systems: Three Common Types

1. Relational Database Management Systems: **RDBMS**
   - e.g. PostgreSQL, Sqlite3, MySQL, Oracle, SQL Server …
   - Store data as tuples in tables. Query using SQL.
2. **Key-value (KV) stores**
   - e.g. BigTable, Hbase, Dynamo, Cassandra, …
   - Store data as "key, value" pairs. Retrieve data based on keys.
3. **MapReduce (MR)**
   - Works on top of a key-value distributed file system (DFS).
   - Invented by Google to run data processing on large clusters. Open-source version is called **Hadoop**.
   - A new trend is to put a SQL interface on top of MR. e.g. Hive

KV-store interface:
- Put (key, value)
- Get (key) → value

MR interface:
- Data = Set of <k1, v1> pairs
- Map(k1, v1) → <k2, v2> // for every key-value pair in the input, output 0 or more key-values
- Reduce (k2, list-values-with-key-k2) → <k3, v3> // Final result is also key-value pairs

RDBMS

```
-------------------------------------------------------------------------
-- Section 1: Creating schemas in SQL, i.e. SQL DDL
-------------------------------------------------------------------------

-- Create a table to store student information
CREATE TABLE Students ( name VARCHAR(80),
                        bday DATE,
                        hobbies VARCHAR(100),
                        uwid INTEGER,
                        PRIMARY KEY (uwid) -- Do not allow two tuples with the same uwid
    );

-- Add sample tuples to the Student table
INSERT INTO Students VALUES ('Jane Doe', '1990-03-01', 'sailing', 111);
INSERT INTO Students VALUES ('Joe Smith', '1991-05-12', 'dancing', 222);
INSERT INTO Students VALUES ('Goof Ball', '1992-12-31', 'watching TV', 333);

-------------------------------------------------------------------------
-- Section 2: Querying in SQL -- i.e. SQL DML
-------------------------------------------------------------------------

SELECT * FROM Students WHERE bday > '1991-01-01' AND hobbies <> 'watching TV';
```

Key-value store example: MongoDB

```
MongoDB browser shell version: 0.1.0
connecting to random database
type "help" for help
type "tutorial" to start the tutorial
> db.students.save({name: "James Bond", age: 21, uwid: 111})
"ok"
> db.students.save({name: "Jane Cool", age: 20, uwid: 222})
"ok"
> db.students.find({age: 21})

[
  {   "name" : "James Bond",   "_id" : {   "$oid" : "50fdffdbcc93742c16007880"   },   "uwid" : 111,   "age" : 21
  }
]
> |
```

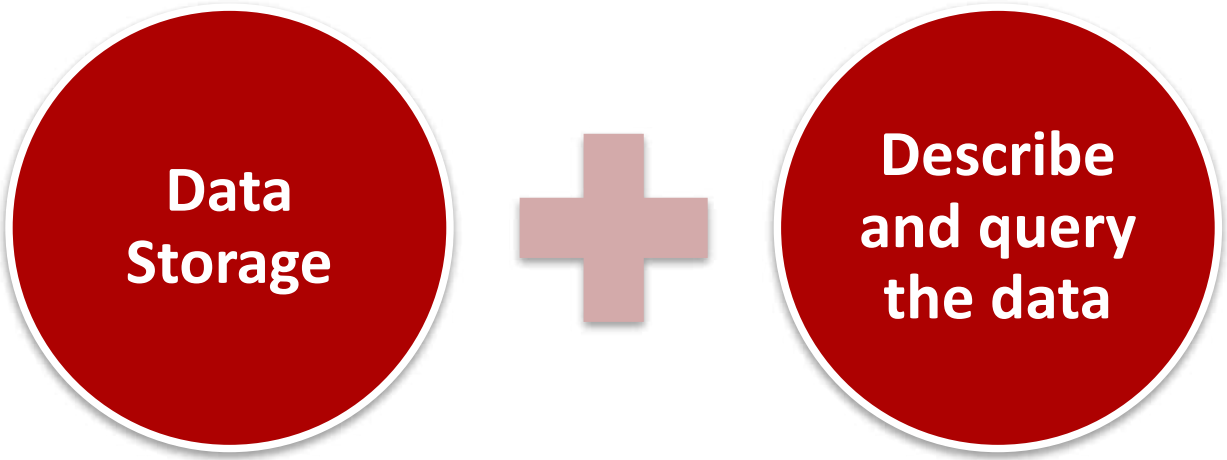
MR System: example Hive

```
CREATE TABLE page_views ( viewTime INT,
                          userid BIGINT,
                          page_url STRING,
                          referrer_url STRING,
                          ip STRING COMMENT 'IP Address of the User')
COMMENT 'This is the page view table'
PARTITIONED BY(dt STRING, country STRING)
STORED AS SEQUENCEFILE;
```

The following Hive query finds all page_views in the month of 03/2013 referred from domain xyz.com:

```
SELECT page_views.*
FROM page_views
WHERE page_views.date >= '2013-03-01'
AND page_views.date <= '2013-03-31'
AND page_views.referrer_url like '%xyz.com';
```

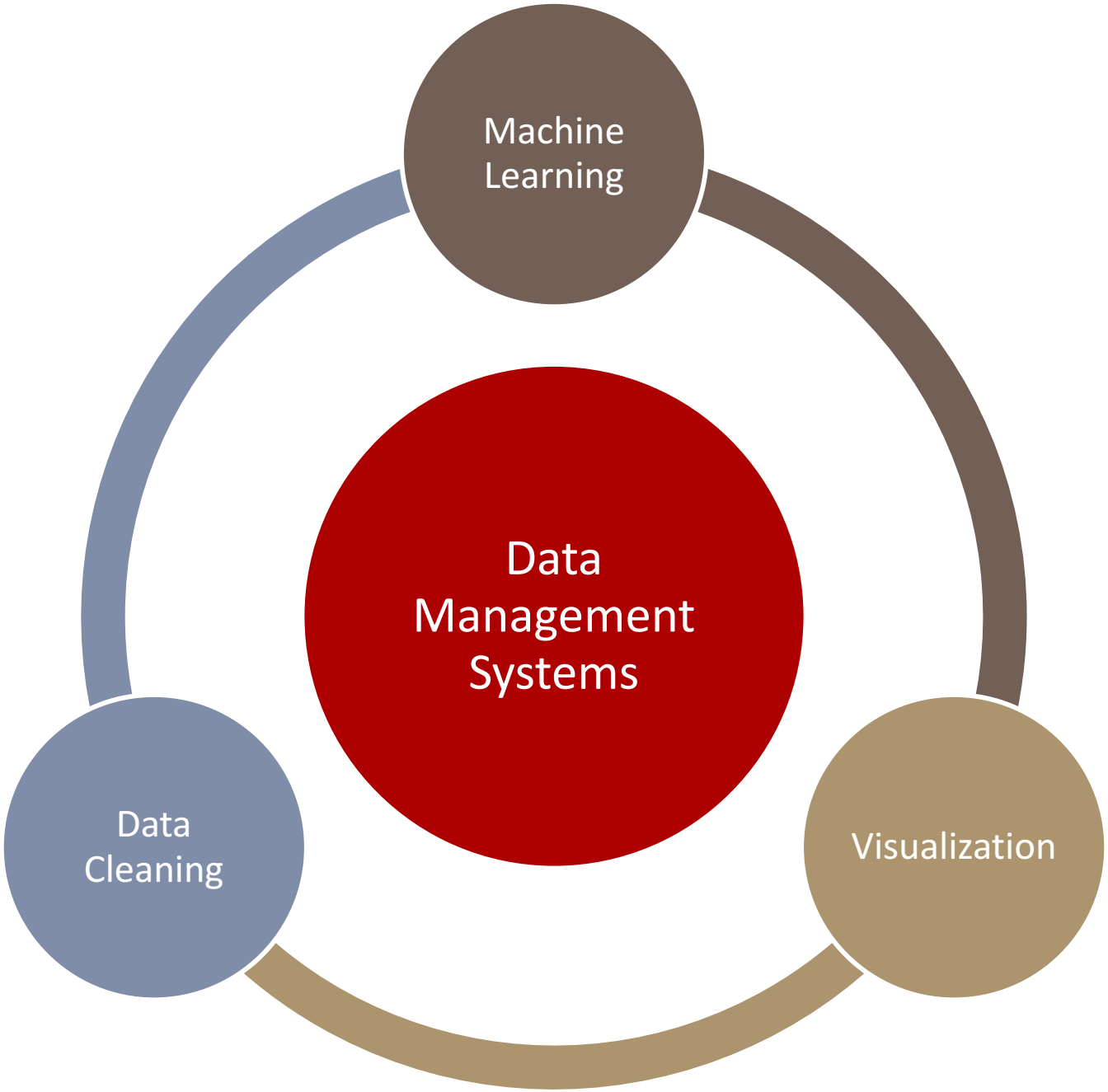
Common Requirement Across All Data Management Systems



Quick note on MapReduce

(I pulled a quick one a few slides ago …)

- SQL-like interfaces (e.g. Hive) are increasingly being used for structured data processing
- … but you can also put unstructured data in MR, e.g. web pages
- … and do complex processing directly on the data, e.g. run a machine learning module to find correlation patterns in the data
- For this course, we will only focus on structured data processing


Machine Learning
Data Management Systems
Data Cleaning
Visualization

Assignment 1: Word count in C++

Do not take this class if you can't make the Friday meetings

- Assignment 1 is now posted
  - C++ warm up: Due 1/27 2PM
  - No late days
- Discussions lead by your TA
- Must attend discussion sessions on Friday: Primary venue for project discussions
- Friday meeting usually used for the discussion, but will be used occasionally for regular lecture