Principles of Data Management

Lecture #1

Instructor: Chen Li
UC Irvine
Overview

- Welcome grads and undergrads alike!
- Course wiki page
  - http://www.ics.ucl.edu/~cs222
  - http://www.ics.ucl.edu/~cs122c
- Piazza page
  - piazza.com/uci/fall2015/cs122ccs222/home
  - Sign up ASAP
DB courses at ICS

- CS122A
- CS122B
- CS122C
- CS222
- CS223
- CS224
Pre-requisite

- CS122A or equivalent
- Data structures, algorithms, OS
- C++ programming skills
Grading

- Class Participation: 5%
- Midterm Exam: 25%
- Final Exam: 25%
- Four-Part Programming Project: 45%
- Undergrads have a different breakdown

- “2-week window” to do a rebuttal
Textbooks

Projects

- 1: file and record management (solo project)
- 2: relation manager (pair project)
- 3: index manager (pair project)
- 4: query engine (pair project)

- C++
- 48-hour grace period with a 5% penalty
Overview of DBMS
Structure of a DBMS

- A typical DBMS has a layered architecture.
- The figure does not show the concurrency control and recovery components (CS 223).
- This is one of several possible architectures; each system has its own variations.
DBMS Structure In More Detail

- Query Parser
- Query Optimizer
- Plan Executor
- Relational Operators (+ Utilities)
- Files of Records
- Access Methods (Indices)
- Buffer Manager
- Disk Space and I/O Manager
- SQL
- Query plans
- API calls

Data Files
Index Files
Catalog Files

Transaction Manager
Lock Manager
Log Manager

WAL

(CS223)
Components’ Roles

- **Query Parser**
  - Parse and analyze SQL query
  - Produce data structure capturing SQL statement and the “objects” that it refers to in the system catalogs

- **Query optimizer (often w/2 steps)**
  - Rewrite query logically
  - Perform cost-based optimization
  - Goal is a “good” query plan considering
    - Physical table structures
    - Available access paths (indexes)
    - Data statistics (if known)
    - Cost model (for relational operations)

(Cost differences can be orders of magnitude!)
Components’ Roles (continued)

- Plan Executor + Relational Operators
  - Runtime side of query processing
  - Usually based on “tree of iterators” model, e.g.:

- Nodes are relational operators (actually they are physical implementations of the various operators)
Components’ Roles (continued)

- **Files of Records**
  - OSs usually have byte-stream based APIs
  - DBMSs instead provide record-based APIs
    - Record = set of fields
    - Fields are typed
    - Records reside on pages of files

- **Access Methods**
  - Index structures for access based on field values
  - We’ll look at tree-based, hash-based, and spatial structures (including the time-tested B+ tree)
  - Peer layer to record-based files (to map from field values to lists of RIDs or lists of primary keys)
Components’ Roles (continued)

- **Buffer Manager**
  - DBMS answer to main memory management
  - Cache of pages from files and indices
  - “DB-oriented” page replacement scheme(s)
  - All disk page accesses go via the buffer pool
  - Also interacts with logging/recovery management (to support undo/redo and thus data consistency)

- **Disk Space and I/O Managers**
  - Manage space on disk (pages), including extents
  - Also manage I/O (sync, async, prefetch, …)
Components’ Roles (continued)

- **System Catalog**
  - Info about physical data (volumes, table spaces, …)
  - Info about tables (name, columns, types, …; also constraints, keys, etc., etc.)
  - Data statistics (e.g., value distributions, counts, …)
  - Info about indexes (types, target tables, …)
  - And so on!
    - Views, triggers, security, …

- **Transaction Management (CS 223)**
  - ACID: Atomicity, Consistency, Isolation, Durability
  - Lock Manager for C+I
  - Log Manager for A+D
A Brief History of Databases

- Pre-relational era: 1960’s, early 1970’s
- Codd’s seminal paper: 1970
- Basic RDBMS R&D: 1970-80 (System R, Ingres)
- RDBMS improvements: 1980-85
- Relational goes mainstream: 1985-90
- Distributed DBMS research: 1980-90
- Parallel DBMS research: 1985-95
- Extensible DBMS research: 1985-95
- OLAP and warehouse research: 1990-2000
- Stream DB and XML DB research: 2000-2010
- Big data R&D: 2005-present
So What’s the Plan?

- We’ll start working our way up the architectural stack next time
- You should also start on the 4-part course project right away
- Immediate to-do’s for you are:
  - Read the materials indicated on the wiki
  - Get yourself signed up on Piazza
  - Review SQL and chapters 1-8 if need be
  - Start on part 1 of the project (solo) today!