Advanced Database Systems

Introduction

WS07/08

Prerequisite of ADS

- You took the course "Datenmodellierung" and "Datenbank Systeme" (both are offered in DBAI for the students at the first and second year's study), or
- You can prove that you have enough background knowledge which covers the above two courses
- Programming language, such as C and basic knowledge on Linux
- Sufficient knowledge on English for reading and writing research papers

Course Goal

- Get an in-depth knowledge on the implementation of relational database systems
- Learning the current state-of-art research topics in database systems
- Practicing skills in conducting your own research work:
  - Reading papers efficiently
  - Writing reviews and surveys
  - Implementation and writing up reports
  - Presentation

Course roadmap

- Relational Database system internals
  - Storage, indexing, query execution and optimization
  - 2 small projects: (hacking the real DB system)
    - Storage (buffer management)
    - Query execution
- Advanced topics
  - Spatial DB, Similarity search, Data warehousing and data mining, etc.
    - Regularly reading research papers and writing reviews
- Final project
  - There will be a list of topics from which you choose one
  - Write a survey paper, presentation
- There will be no exam!

Course load

- Homework projects (25%) (Nov., Dec.)
- Reading assignments (35%) (Nov., Dec., Jan.)
- Final project (40%) (Dec., Jan.)
  - Report (25%)
  - Presentation (15%)
  - With implementation and analysis of existing algorithms (bonus)
  - With your own idea of improvement, suggestion (bonus)
  - Implementation of your new idea (double bonus)

History of Relational Database Systems

- In the early days, database applications were built directly on top of file systems
- Drawbacks
  - 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

History of Relational Database Systems

- CODASYL: a COBOL extension for manipulating collection of records
- Application programmer needed to know the physical data organization (indexing, etc.)
  - For instance, to join two tables, application programmer had to compose code with nested loops
  - Application programmer carried out the task of query optimization
  - If data changed, the query code had to be changed too!

The relational revolution (1970's)

- A simple data model (relation)
- Declarative query language (SQL)
- Application users specify what answers a query should return, not how
- DBMS picks the best execution strategy based on availability of indexes, data/workload characteristics, etc.

Provides physical data independence

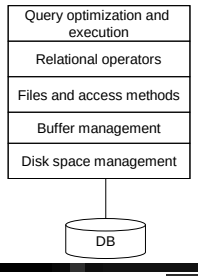
Physical data independence

- Applications should not need worry about how data is physically structured and stored
- Applications should work with a logical data model and declarative query language
- Leave the implementation details and optimization to DBMS
- The single most important reason behind the success of DBMS today

And a Turing Award for E. F. Codd

Structure of a DBMS

- A typical DBMS has a layered architecture.
- The figure does not show the concurrency control and recovery components.
- This is one of several possible architectures; each system has its own variations.
- All the layers except for "optimization and execution" have to consider concurrent control and recovery.



Advanced database technologies

- Revolution on data models:
  - Object-based data models (Object-oriented and Object-relational)
  - Semi-structured data model (XML)
- Solutions:
  - native systems, build everything from scratch
  - extend relational model with object-oriented and XML features

Advanced database technologies

- The traditional accessing method is inefficient
  - Spatial database
    - How to store geographical objects efficiently?
    - Indexing methods as B-tree and hashing are inefficient for many conventional geo-queries like Nearest-Neighbor queries
  - High-dimensional data, time series

Advanced database technologies

- New queries, more efficient queries
  - Similarity queries

```
SELECT *
FROM Movies
WHERE star SIMILARTO 'Schwarrzenger'  AND year
                                    BETWEEN [1980,1999];
```

Advanced database technologies

- New queries, more efficient queries
  - queries with sampling
    - Data volumes are huge: conventional query-processing engines can take hours or even days to compute the exact answer for a very complex SQL query
    - For several application scenarios, exact query answers are not really required
    - users would be much happier with a *fast, approximate answer* to their query

```
SELECT avg(amount)
FROM indivdonations TABLESAMPLE SYSTEM(10)
WHERE committee_id='C00386987';
```

Advanced database technologies

- Discover rules and trends over a huge amount of data
  - Data Warehousing
  - Data mining

Advanced database technologies

- Redesign of the query implementation issues due to the improvement of hardware technology
  - Physical storage of data: is it still efficient with actual CPU and main memory? -- new storage models
  - How to do the optimization if no statistic information of relations is available? -- adaptive query processing

Advanced database technologies

- And with Internet application
  - Data integration
  - Stream systems
  - Publish/subscribe systems
  - Sensor databases
  - RFID databases
  - The list goes on

Course Information

- Textbook
  - Recommended reference:
    - *Database Systems: The Complete Book*, by H. Garcia-Molina, J. D. Ullman, and J. Widom
    - *Database Management Systems*, by Raghu Ramakrishnan, Johannes Gehrke
    - *Readings in Database Systems*" (a.k.a. the "Red Book"), edited by Stonebraker and Hellerstein
  - Major database conferences:
    - Sigmod, VLDB, ICDE
    - DBLP bibliography
    - ACM Digital library
- Web site (http://www.dbai.tuwien.ac.at/staff/wei/teaching/ads0708/)

Assignments

- Reading assignments posted on the website.
- Prepare for your homework projects:
  - Download PostgreSQL (source code)
  - Try to install it on your computer
  - Recall some programming concepts, if you have not written code for some time
- Prepare yourself with reading about the data storage management, buffer management and indexing, so that the basic parts can be skipped quickly
- Here is another interesting article about the history of System R, the first Relational Database System (http://www.mcjones.org/System_R/SQL_Reunion_95/)