Course Introduction & History of Database Systems

CMPT 843, SPRING 2019

JIANNAN WANG

https://sfu-db.github.io/dbsystems/

Introduce Yourself

What's your name?

Where are you from?

M.Sc. or Ph.D.? Which year?

What do you want to get out of the course?

Course Introduction

Why this course?

Objective 1. Stand on the shoulders of giants!

- A Half Century DB Research
- VLDB = Very Large Data Base (founded in 1975)
- Four Turing Awards
- $50 billion Market Per Year

Objective 2. Master essential skills for being a researcher

- Reading Papers
- Giving Talks
- Reviewing Papers
- Asking Questions

Prerequisites

Undergraduate Database Systems Courses

- **Testing yourself:** Relational Model, SQL, Query Optimization, Transaction, Concurrency Control, ACID, etc.

If not, you have to spend extra time on

- Textbook: "Database Management Systems"
- Online Courses: Stanford, Berkeley

Part 1: Traditional Database Systems and Techniques (20 papers)

Part 2: Modern Database Systems and Techniques (18 papers)

39 papers

Part 1: Traditional Database Systems and Techniques (before 2000)

Background

1. Database Systems: Achievements and Opportunities (1990)
2. The Asilomar Report on Database Research (1998)

Part 1: Traditional Database Systems and Techniques (before 2000)

Data Model

3. A Relational Model of Data for Large Shared Data Banks (1970)
4. What Goes Around Comes Around (1960-1970, Sec I~IV only)

Part 1: Traditional Database Systems and Techniques (before 2000)

Traditional DBMS

5. A History and Evaluation of System R (1981)
6. The Design of Postgres (1986)
7. The Gamma database machine project (1990)
8. An Overview of Data Warehousing and OLAP Technology (1997)

Part 1: Traditional Database Systems and Techniques (before 2000)

Transaction Management

9. Granularity of Locks and Degrees of Consistency in a Shared Data Base (1976, Part 1 only)
10. Granularity of Locks and Degrees of Consistency in a Shared Data Base (1976, Part 2 only)
11. On Optimistic Methods for Concurrency Control (1981)
12. Concurrency control performance modeling: alternatives and implications (1987)

Part 1: Traditional Database Systems and Techniques (before 2000)

Query Optimization

13. Access Path Selection in a Relational Database Management System (1979)
14. The Volcano Optimizer Generator: Extensibility and Efficient Search (1993)
15. Efficient mid-query re-optimization of sub-optimal query execution plans (1998)
16. Eddies: Continuously Adaptive Query Processing (2000)

Part 1: Traditional Database Systems and Techniques (before 2000)

Interactive Analytics

17. Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-tab, and Sub-totals (1997)
18. An Array-Based Algorithm for Simultaneous Multidimensional Aggregates (1997)
19. New Sampling-Based Summary Statistics for Improving Approximate Query Answers (1998)
20. Informix under CONTROL: Online Query Processing (2000)

Part 2: Modern Database Systems and Techniques (after 2000)

Background

21. Challenges and Opportunities with Big Data (2011)

Part 2: Modern Database Systems and Techniques (after 2000)

MapReduce and Beyond

22. MapReduce: Simplified Data Processing on Large Clusters (2003)
23. A Comparison of Approaches to Large-Scale Data Analysis (2009)
24. Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing (2012)
25. Spark SQL: Relational Data Processing in Spark (2015)

Part 2: Modern Database Systems and Techniques (after 2000)

Column Store

26. C-store: A Column-oriented DBMS (2005)
27. Dremel: Interactive Analysis Of Web-Scale Datasets (2010)
28. Column-Stores vs. Row-Stores: How Different Are They Really?(2012)

Part 2: Modern Database Systems and Techniques (after 2000)

NoSQL

29. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications (2003)
30. Bigtable: A Distributed Storage System for Structured Data (2006)
31. Dynamo: Amazon's Highly Available Key-Value Store (2007)
32. CAP Twelve Years Later: How the "Rules" Have Changed (2012)

Part 2: Modern Database Systems and Techniques (after 2000)

NewSQL

33. OLTP Through the Looking Glass, and What We Found There (2008)
34. Hekaton: SQL Server's Memory-optimized OLTP Engine (2013)
35. Efficiently Compiling Efficient Query Plans for Modern Hardware (2011)
36. Scalable SQL and NoSQL data stores (2010)

Part 2: Modern Database Systems and Techniques (after 2000)

ML and SQL

37. Accelerating Machine Learning Inference with Probabilistic Predicates (2018)
38. NoScope: Optimizing Deep CNN-Based Queries over Video Streams at Scale (2017)
39. Learning to Optimize Join Queries With Deep Reinforcement Learning (2018)
40. The Case for Learned Index Structures (2018)

References

Papers to read

- Redbook - 5th Edition (Peter Bailis, Joseph M. Hellerstein, Michael Stonebraker)
- Readings in Database (Reynold Xin)

Related Graduate Courses

- CS286: Implementation of Database Systems (UC Berkeley, Fall 2014)
- EECS 584: Advanced Database Management Systems (UMichgan, 2015 Fall)
- Big Data Systems (Columbia, 2016 Spring)
- 15-799: Advanced Topics in Database Systems (CMU, 2013 Fall)

Skills

Reading Papers

Giving Talks

Reviewing Papers

Asking Questions

How you will be trained

Reading Papers

- 40 Papers

Giving Talks

- 2 Paper (20min+10 min Q&A)

Reviewing Papers

- 2 Papers

Asking Questions

- Asking at least 10 questions in the Q&A sessions

Grading

Paper Presentation: 20%

Paper Review: 20%

Participation: 10%

Questions: 10%

Blog Post: 10%

Final Project: 30% (2% plan + 14% poster + 14% report)

What's next

Fill in the form by the end of Monday 1/7

https://goo.gl/g3y735

History of Database Systems

Database Systems in a Half Century

(1960s – 2010s)

When	What
Early 1960 – Early 1970	The Navigational Database Empire
Mid 1970 – Mid 1980	The Database World War I
Mid 1980 – Early 2000	The Relational Database Empire
Mid 2000 – Now	The Database World War II

References.

- https://en.wikipedia.org/wiki/Database#History
- What Goes Around Comes Around (Michael Stonebraker, Joe Hellerstein)
- 40 Years VLDB Panel

The Navigational Database Empire

(Early1960 – Early 1970)

Data Model

1. How to organize data
2. How to access data

Navigational Data Model

1. Organize data into a multi-dimensional space (i.e., A space of records)
2. Access data by following pointers between records

Inventor: Charles Bachman

1. The 1973 ACM Turing Award
2. Turing Lecture: "The Programmer As Navigator"

The Navigational Database Empire

(Early1960 – Early 1970)

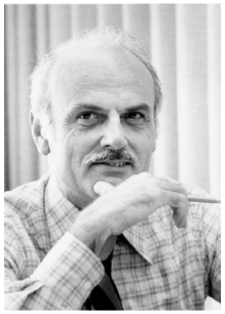
Representative Navigational Database Systems

- Integrated Data Store (IDS), 1964, GE
- Information Management System (IMS), 1966, IBM
- Integrated Database Management System (IDMS), 1973, Goodrich

CODASYL

- Short for "Conference/Committee on Data Systems Languages"
- Define navigational data model as standard database interface (1969)

The Birth of Relational Model



Ted Codd

- Born in 1923
- PHD in 1965
- “A Relational Model of Data for Large Shared Data Banks” in 1970

Relational Model

- Organize data into a collection of relations
- Access data by a declarative language (i.e., tell me what you want, not how to find it)

Data Independence

The Database World War I: Background

One Slide (Navigational Model)

- Led by Charles Bachman (1973 ACM Turing Award)
- Has built mature systems
- Dominated the database market

The other Slide (Relational Model)

- Led by Ted Codd (mathematical programmer, IBM)
- A theoretical paper with no system built
- Little support from IBM

The Database World War I:
Three Big Campaigns

1. Which data model is better in theory?
   (Mid 1970)

2. Which data model is better in practice?
   (Late 1970 – Early 1980)

3. Which data model is better in business?
   (Early 1980 – Mid 1980)

The "Theory" Campaign

A **debate** at ACM SIGFIDET (precursor of SIGMOD) 1974

Navigational model is bad

- Data Organization: So complex
- Data Access: No declarative language

Relational model is bad

- Data Organization: A special case of navigational model
- Data Access: No system proof that declarative language is viable

The “Practice” Campaign

The Big Question

- Can a relational database system perform as good as a navigational system?

System prototypes

- Ingres at UC Berkeley (early and pioneering)
- System R at IBM (arguably got more stuff “right”)

The System R Team

- Query Optimization (Patricia P. Griffiths et al.)
- SQL (Donald D. Chamberlin et al.)
- Transaction (Jim Gray et al.)

The “Business” Campaign

Commercialization of Relational Database Systems

Not as easy as we thought

Three reasons (required) that led relational database systems to won

- The minicomputer revolution (1977)
- Competing products (e.g. IDMS) could not be ported to the minicomputer
- Relational front end was not added to navigational database systems

What Can We Learn?

Lesson 1.

The winning of theory $\neq$ The winning of practice

Lesson 2.

The winning of practice $\neq$ The winning of business

Lesson 3.

Everyone can get a chance to win

The Relational Database Empire
(Mid1980 – Early 2000)

Parallel and distributed DBs (1980 – 1990)
- SystemR*, Distributed Ingres, Gamma, etc.

Objected-oriented DB (1980 – 1990)
- Objects: Data/Code Integration
- Extensibility: User-defined functions, User-defined data types

MySQL and PostgresSQL (1990s)
- Widely used open-source relational DB systems

The Database World War II: Background

Internet Boom (Early 2000)

- Larger data volume that cannot be fit in a single machine
- Faster data updates that cannot be handled by a single machine

Commercial distributed database systems are expensive ☹

Open-source database systems do not support distributed computing well ☹

The Database World War II:
Two Big Campaigns

1. Which is better for large-scale data analysis:
   MapReduce vs. Traditional Database Systems?

2. Which is better for distributed transaction processing:
   NoSQL vs. Traditional Database Systems?

Let's find the answers through this course! ☺