CSCI 550: Advanced Data Stores

Basic Information

Place and time: Spring 2014, Tue/Thu 9:30-10:50 am

Instructor: Prof. Shahram Ghandeharizadeh, shahram@usc.edu, 213-740-4781

ITS Help: E-mail: consult@usc.edu
Phone: 213-740-5555
Web: http://www.usc.edu/its/csc/
Note: Software for programming assignments is available in ITS user rooms.

Prerequisites: CSCI 485 or 585

Course Description

The landscape of Database Management Systems (DBMSs) has expanded to include SQL, NoSQL and NewSQL systems. Some are hybrids that offer expressiveness of SQL extended with simple functionalities such as put, get, and delete calls of a key-value store. The purpose of this course is to provide an overview of these DBMSs along with their design and implementation. More specifically, we present and discuss:

1. Key-value stores such as Berkeley DB, LinkedIn's Voldermort, Riak, Redis, and Amazon's Dynamo,
2. Key-value SQL DBMSs such as USC's SQLTrig and MIT's CacheGenie and TxCache,
3. Document stores such as CouchDB, MongoDB, RavenDB, Jackrabbit, and SimpleDB,
4. Extensible record stores such as HBase, Facebook's Cassandra, and Google's BigTable,
5. Scalable RDMBS such as VoltDB, Clustrix, and ScaleDB.

In addition, we explore applications of these systems and their definition of consistency (strong, weak, and eventual). Other topics include: Scalable data stores, benchmarking frameworks to evaluate data stores, data center scale database management systems, and middle-tier caches.

This course assumes student familiarity with a conceptual data modeling tool such as Entity-Relationship (ER) data model, a logical data model such as the relational data model, SQL query language, normal forms and logical data design, physical characteristics of mass storage devices such as magnetic disks and memory, physical data design and index structures such as B+-tree and hash indexes, concurrency control and crash recovery protocols.

Course Requirements and Grades

- There are no required text books. The reading material is based on recently published technical papers available via the ACM/IEEE/Springer digital libraries. All USC students have automatic access to these digital archives.
- Grading breakdown
  - Exam 1: 30%
  - Exam 2: 30%
  - Class project: 20%
  - Class participation: 10%
  - Homework assignment: 10%

- In class attendance is mandatory due to the nature of the technical papers discussed and presented. Absence from lecture will almost certainly impact student performance for both exams.

Course Project
Sample class projects may include a paper that surveys a technique or a system, an application of either a NewSQL or a NoSQL system, use of a benchmark to evaluate a system, development of a new benchmark that captures the essence of an application, and design of a consistency technique for next generation data stores.

Spring 2014 At a Glance
Here is an overview of the Spring 2014 semester calendar and its reading material. Papers covered by each topic are detailed below.

Jan 13, 15: Introduction
Jan 20: Martin Luther King Day
Jan 22: Introduction Continued
Jan 27, 29: Cache Augmented SQL Systems
Feb 3, 5: NewSQL
Feb 5: Projects start
Feb 10, 12: Benchmarks
Feb 17: President's Day
Feb 19, 24: Web, Databases, and Caches
Feb 26: Key-Value Stores
March 3: Review for Exam 1
March 5: Exam 1 covers material presented from Jan 13 to Feb 26
March 10, 12: Scalable RDBMS
March 17, 19: Spring Break
March 24, 26, 31: Data Center Scale Stores
April 2, 7: In-Class presentation on projects
April 9, 14: Extensible Stores
April 16, 21: Application Transparent Mid-Tier Database Caching
April 23: Project reviews
April 28: Review for Exam 2
April 30: Exam 2 covers material presented from March 24 to April 21

Course Readings/Class Sessions
The reading material for the course is organized chronologically and based on a specific theme. This material is tentative and based on recent publications. We will make adjustments to the list as new manuscripts (currently under review) are accepted for publication.

Introduction
- R. Cattell. Scalable SQL and NoSQL Data Stores. White Paper, June 12, 2011.
- M. Stonebraker. New Opportunities for New SQL. Communications of the ACM, November 2012, Vol. 55, No. 11.
- C. Mohan. History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla. EDBT 2013.

Cache Augmented SQL Systems

- R. Nishtala. Scaling Memcache at Facebook. NSDI 2013.
- S. Ghandeharizadeh and J. Yap. Gumball: A Race Condition Prevention Technique for Cache Augmented SQL Database Management Systems. In the Second ACM SIGMOD Workshop on Databases and Social Networks (DBSocial), May 2012.
- P. Gupta, N. Zeldovich and S. Madden. CacheGenie: A Trigger-Based Middleware Cache for ORMs. Middleware 2011, December 2011.

Suggested readings:

- A. Labrindis and N. Roussopoulos. Exploring the Trade-off Between Performance and Data Freshness in Database-Driven Web Servers. VLDB Journal, Volume 13, 2004.
- S. Ghandeharizadeh, J. Yap, and S. Barahmand. COSAR-CQN: An Application Transparent Approach to Cache Consistency. In Twenty First International Conference On Software Engineering and Data Engineering, June 2012.

NewSQL

- Pavlo, et. al. Skew-Aware Automatic Database Partitioning in Shared-Nothing Parallel OLTP Systems. SIGMOD 2012.
- A. Narayan et. al. DJoin: Differentially Private Join Queries over Distributed Databases. OSDI 2012.

Suggested readings:

- J. Shute, et. al. F1-The Fault-Tolerant Distributed RDBMS Supporting Google's Ad Business. SIGMOD 2012.

Benchmarks

- S. Barahmand and S. Ghandeharizadeh. BG: A Benchmark to Evaluate Performance of Interactive Social Networking Actions. CIDR 2013.
- Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, R. Sears. Benchmarking Cloud Serving Systems with YCSB. SoCC 2010.
- S. Patil, M. Polte, K. Ren, W. Tantisiriroj, L. Xiao, J. Lopez, G. Gibson, A. Fuchs, B. Rinaldi. YCSB++: Benchmarking and Performance Debugging Advanced Features in Scalable Table Stores, SoCC 2011.

Suggested Reading:

- A. Floratou, et. al. Can the Elephants Handle the NoSQL OnSlaught? VLDB 2012.

Web, Databases, and Caches

- K. S. Candan, W.-S Li, Q. Lou, W.P.Hsiung, D. Agrawal. Enabling Dynamic Content Caching for Database-Driven Web Sites. SIGMOD 2001.
- J. Challenger, A. Iyengar, P. Dantzig. A Scalable System for Consistently Caching Dynamic Web Data. IEEE INFOCOM, 1999.

Key-Value Stores

- B. Atikoglu, et. al. Workload Analysis of a Large-Scale Key-Value Store. ACM SIGMETRICS 2012.

- J. Rao, E. J. Shekita, S. Tata: Using Paxos to Build a Scalable, Consistent, and Highly Available Datastore. PVLDB 4(4): 243-254 (2011).
- G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's Highly Available Key-Value Store. In SOSP 2007.

Suggested Reading:

- M. Seltzer. Beyond Relational Databases. Communications of the ACM, July 2008, Vol. 51, No. 7.
- B. Debnath, S. Sengupta, Jin Li. SkimpyStash: RAM Space Skimpy Key-Value Store on Flash-based Storage. SIGMOD 2011.

Data Center Scale Stores

- F. Nawab, D. Agrawal, A. El Abbadi. Message Futures: Fast Commitment of Transactions in Multi-Datacenter Environments. CIDR 2013.
- S. Kadambi, J. Chen, B. F. Cooper, D. Lomax, R. Ramakrishnan, A. Silberstein, H. Garcia-Molina. *Where in the World is My Data?* VLDB Conference, August 2011.
- J. Corbett, et. al. Spanner: Google's Globally-Distributed Database. OSDI 2012.
- B. F. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H. Jacobsen, N. Puz, D. Weaver and R. Yerneni. *PNUTS: Yahoo!'s Hosted Data Serving Platform*. VLDB Conference (industry track), 2008.

Suggested reading:

- M. Stonebraker, S. Madden, D. Abadi, S. Harizopoulos, N. Hachem and P. Helland. The End of an Architectural Era (It's Time for a Complete Rewrite). In VLDB 2007.
- P. Helland. Life Beyond Distributed Transactions: An Apostate's Opinion. In CIDR 2007.
- R. Kallman, J. Natkins, H. Kimura, A. Pavlo, A. Rasin, S. Zdonik, E. Jone, S. Madden, M. Stonebraker, D. Abadi. H-Store: A High-Performance, Distributed Main Memory Transaction Processing System. VLDB 2008.

Extensible Stores

- F. Chang et al. Bigtable: A Distributed Storage System for Structured Data. In OSDI 2006.
- O. Kennedy, Y. Ahmad, C. Koch. DBToaster: Agile Views for a Dynamic Data Management System. CIDR 2011.

Application Transparent Mid-Tier Database Caching

- P. Larson, J. Goldstein, and J. Zhou. MTCache: Transparent Mid-Tier Database Caching in SQL Server. In Proceedings of the 20th International Conference on Data Engineering. Page 177-189, April 2004.
- C. Bornhovd, M. Altinel, S. Krinshnamurthy, C. Mohan, H. Pirahesh, B. Reinwald. DBCache: Middle-Tier database Caching for Highly Scalable e-Business Architectures. SIGMOD 2003.
- C. Garrod et. al. Scalable Query Result Caching for Web Applications. VLDB 1998.

Suggested Reading:

- The Times Ten Team. High Performance and Scalability Through Application-Tier In-Memory Data Management. VLDB 2000.
- Lou and J. Naughton. Form-Based Proxy Caching for Database Backed Web Sites. VLDB 2001.

Statement for Students with Disabilities

Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.–5:00 p.m., Monday through Friday. Website and contact information for DSP: http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html, (213) 740-0776 (Phone), (213) 740-6948 (TDD only), (213) 740-8216 (FAX) ability@usc.edu.

Statement on Academic Integrity

USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All students are expected to understand and abide by these principles. *SCampus*, the Student Guidebook, (www.usc.edu/scampus or http://scampus.usc.edu) contains the University Student Conduct Code (see University Governance, Section 11.00), while the recommended sanctions are located in Appendix A.

Emergency Preparedness/Course Continuity in a Crisis

In case of a declared emergency if travel to campus is not feasible, USC executive leadership will announce an electronic way for instructors to teach students in their residence halls or homes using a combination of Blackboard, teleconferencing, and other technologies.