CS 549 Summer 2020

Distributed Systems and Cloud Computing

Instructor

[Dominic Duggan](#)
Email: Canvas email and Piazza
Office Hours: Friday 1-2pm or by appointment, via Zoom.
Class Materials: http://sit.instructure.com.

Course Objectives

The objective of this course is to give students a basic grounding in designing and implementing distributed and cloud systems. Developers of cloud services question how those services should be implemented. What are global consensus and Paxos, and what are their application in building cloud systems? What are the advantages and disadvantages of using distributed NoSQL stores such as Cassandra instead of relational stores such as MySQL? What are strong and weak consistency, what is the "CAP Theorem," and what is its implication for building highly available services? What are the roles of REST, Websockets and stream processing in cloud applications? This course will combine hands-on experience in developing cloud services, with a firm grounding in the tools and principles of building distributed and cloud applications, including advanced architectures such as peer-to-peer, publish-subscribe and streaming. Besides cloud services, we will also be looking at cloud support for batch processing, such as the Hadoop and Pig frameworks, and their use with NoSQL data stores such as Cassandra.

There are some related courses that you may be interested in.

- **CS526 Enterprise and Cloud Computing:** This course considers cloud computing from the perspective of Web applications and Web services, using the Microsoft Azure platform, including ASP.NET MVC and Windows Communication Foundation. It also goes into depth on virtualization as part of the infrastructure for the cloud. The course considers two examples of NoSQL data stores: Windows Azure Storage, a key-based database model, and MongoDB, a document-oriented database model. CS549 goes much further into the foundations and implementation of NoSQL data stores such as Cassandra.
- **CS548 Enterprise Software Architecture and Design:** This course considers data modeling, software architectures and design patterns for enterprise and cloud applications. It goes into much more depth on aspects of Java Enterprise Edition than CS549, although CS549 does cover some advanced aspects of Java EE (e.g., CDI) not considered in CS548. The focus of CS549 assignments is on frameworks such as Jersey and Atmosphere that go beyond Java EE, and on distributed architectures beyond client-server, such as peer-to-peer. CS548 also considers data modeling for NoSQL data stores, particularly for key-based store models, while CS549 considers the implementation of NoSQL stores.

Course Outcomes

1. **Services -** Develop distributed and Web services, using programming paradigms such as client-server, streaming, peer-to-peer, event-driven and REST.
2. **Agreement** - Explain protocols for achieving agreement in distributed systems, such as 2PC and Paxos, and explain their application in scenarios such as transactional e-commerce and cloud infrastructure.
3. **Availability** - Explain methods for building highly available services, and explain the trade-off between weak and strong consistency as given the CAP theorem. Explain the significance for cloud systems such as Cassandra and Chubby.
4. **Cloud -** Develop cloud applications using frameworks such as Cassandra, Hadoop, Pig and Hive.
5. **Storage -** Describe implementation methods and protocols for distributed storage systems: distributed file systems, distributed hash tables, cloud storage, NoSQL databases. Describe examples such as NFS, Dropbox, HFS, Dynamo and Cassandra.

Course Prerequisites

You are expected to have a certain amount of programming maturity, since you will be developing distributed programs during the course of the term. You should know Java since that will be the language for the assignments. If you know C#, you can learn Java very quickly.

Texts

The following is the required textbook for CS548 Enterprise Software Architecture and Design. We will be covering some of the advanced technical material in the book that is not covered in that course (see the syllabus below for details). **You should not buy the book just for this course.**

- **[D] Recommended:** *Enterprise Software Architecture* by Dominic Duggan. Wiley, 2012.

The following books are helpful references. You are not required to buy any of them.

- **[Hadoop]** *Hadoop: The Definitive Guide, 4th edition* by T. White. ISBN 978-1491901632. O'Reilly, 2015.

In addition, there will be several "classic" papers from the literature, frankly to compensate for the lack of a decent textbook in the area:

- **[BT]** F. Chang et al. *BigTable: A Distributed Storage System for Structured Data.* Proc. 7th Symp. Operating System Design and Implementation (OSDI 06), Usenix Assoc., 2006; www.usenix.org/events/osdi06/tech/chang.html.
- **[CL85]** K.M. Chandy and L. Lamport. *Distributed snapshots: Determining global states of distributed systems.* ACM Transactions on Computer Systems Vol. 3, No. 1, pp 63-75.
- **[Dyn]** G. DeCandia et al. *Dynamo: Amazon's Highly Available Key-value Store.* ACM SIGOPS Operating Systems Rev., vol. 41, no. 6, 2007, pp. 205–220; http://s3.amazonaws.com/AllThingsDistributed/sosp/amazon-dynamo-sosp2007.pdf.
- **[E2E]** Jerome H. Saltzer, David P. Reed, and David D. Clark. *End-to-End Arguments in System Design.* Second International Conference on Distributed Computing Systems (April 1981) pages 509-512.
- **[GFS]** Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. *The Google File System.* ACM Symposium on Operating Systems Principles, 2003.
- **[L01]** L. Lamport, *Paxos Made Simple.* ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001) 51-58.
- **[L78]** L. Lamport. *Time, Clocks and the Ordering of Events in a Distributed System.* Communications of the ACM 21, 7 (July 1978), 558-565.
- **[MR]** Jeffrey Dean and Sanjay Ghemawat. *MapReduce: Simplified Data Processing on Large Clusters.* Operating Systems Design and Implementation (OSDI), 2004.
- **[NFSv4]** B. Pawlowski, S. Shepler, D. Noveck, D. Robinson and R. Thurlow. *The NFS Version 4 Protocol.* 2nd International System Administration and Networking Conference (SANE), 2000.

Grading

The breakdown of grades is as follows:

- Assignments: 45%
- Mini-Exams: 45%
- Final Exam: 10%

Late Policy

Assignments may be submitted after the due date, but up until the cutoff date, with a penalty of -5%. Assignments may be submitted up until the second cutoff date (usually two weeks after the original due date), with an additional penalty of -20%. Assignments may be submitted up until the third cutoff date (usually three weeks after the original due date), with an additional penalty of -25%. There will be no extensions past the third cutoff date. If the cutoff date is the same as the due date, no late extensions are allowed. Please note that an assignment with a penalty of -50% still carries a much higher grade than no assignment at all. *Please also note that resubmission of assignments is not allowed. We do not have the resources to regrade assignments, so please be sure to submit the final version when it is ready.*

All exam and quizzes ("mini-exams") must be submitted by the time and date posted. There will be no extensions and no late submissions allowed. However, assuming that N mini-exams are administered during the semester, your mini-exam grade will be based on the best N-1 of your mini-exam scores. Therefore you maximize your possible grade by taking all mini-exams, but you do not suffer any penalty if you miss a single mini-exam. It is your responsibility not to miss more than one mini-exam.

Software

All assignments will be done in the Java programming language. Assignments will involve the use of tools such as:

- Sockets and Java RMI (standard APIs in Java).
- Jersey for Web services.
- Websockets and Embedded Glassfish.

In addition, there will be an assignment involving the following:

- Hadoop (programming environments for cloud data processing).

Course Schedule

Week	Date	Topics Covered	Reading	Assign
0	5/11	**ORIENTATION**		
1	5/19	Introduction to distributed systems and cloud computing. Cloud architectures: SaaS, PaaS, IaaS. End-to-end system design. Networks and protocol stacks.	D 2.1-2, 2.7. E2E	
2	5/25	Client-server computing. Sockets and remote procedure call.	D 2.4. RPC	
3	6/1	Distributed file systems and cache consistency. NFS, AFS. Storage for Cloud Computing: Google/Hadoop file system. Web services and REST. Example: Amazon S3. The JAX-RS API.	D 6.9.4. NFS. GFS. D 7.5	A1: RMI and sockets.
4	6/8	Failure models and failure detectors.	D 2.3.	
5	6/15	Distributed debugging. Time and ordering of Events. Causal broadcasts.	D B.2, B.3, L78.	A2: P2P & REST.
6	6/22	Asynchrony: Publish-subscribe. Server-side events. Web sockets. Distributed snapshots.	D 2.5, B.3.5. CL85.	
7	6/29	Transactions. Serializability and recoverability.	Bien.	
8	7/6	Long-lived transactions. Atomic commitment protocols: 2PC and 3PC.		A3: Server-Side Events.
9	7/13	Highly available services. Replicated services and quorum consensus. Introduction to Map-Reduce.	D 6.9.1-2, B.3.5. VS.	
10	7/20	Batch cloud computing: Map-Reduce and Hadoop. Domain-specific languages for cloud data processing: Pig Latin.	MR Pig	A4: Websockets
11	7/27	NoSQL data stores. Table-based (Google BigTable), key-based (Amazon Dynamo), and Cassandra. Eventual consistency and the CAP Theorem.	BT, Dyn, Cass.	
12	8/3	Consensus and the Paxos algorithm. Applications in the cloud: Google Chubby, Yahoo Zookeeper.	L01.	A5: Hadoop
13	8/10	Peer-to-peer systems. Distributed hash tables. Applications in multiplayer game-playing.	Chord, Mercury.	
	8/22-8/26	**FINAL EXAM**		

Class Format

1. **Lecture slides and videos:** I will be making slides and short lecture videos available each week, via Canvas. It is your responsibility to review these materials and take the exams (see below). It is important that you keep up with this material.
2. **Reading:** There will sometimes be reading associated with each topic. It is highly recommended that you do the reading. You should view the lectures as intended to draw out what is important in the reading and explain the key points of understanding. By doing the reading, you will get much better depth of understanding in the material than can be made available in the slides alone. Readings will be from the texts and from other on-line materials as the term progresses.
3. **On-line discussion:** The Canvas classrooms include Piazza forums, which you are encouraged to use for on-line discussions of the course material. You will find the Piazza system particularly useful for the assignments.
4. **Exams:** There will weekly “mini-exams” (more extensive than a quiz, but shorter than a full exam). They will be due on Monday the week after the material is covered in on-line lectures.
5. **Homework:** There will be several programming assignments. They will be due at midnight on Sunday at the end of the week in which their due date falls, via Canvas. The first few assignments give you exercise with the Android API, progressing to assignments that develop best practices and design patterns for managing the complexity introduced by these APIs.
6. **Virtual Office Hours** are a weekly synchronous session (through Zoom) to give you the opportunity to ask questions related to course material and/or assignments.
7. I will be available via **Canvas email** and will respond as soon as I am available (generally within 24-48) hours. For the online discussions, I will check in at least 3 times per week. Often I will not need to respond to questions, as students often can answer each others' questions before I have a chance to respond. For urgent matters, feel free to email me; Canvas email is much preferred as the Canvas mailbox is not as cluttered as regular email (and gets forwarded to my regular email anyway).

Ethical Conduct

Undergraduate Honor System

Enrollment into the undergraduate class of Stevens Institute of Technology signifies a student's commitment to the Honor System. Accordingly, the provisions of the Stevens Honor System apply to all undergraduate students in coursework and Honor Board proceedings. It is the responsibility of each student to become acquainted with and to uphold the ideals set forth in the Honor System Constitution. More information about the Honor System including the constitution, bylaws, investigative procedures, and the penalty matrix can be found online at http://web.stevens.edu/honor.

The following pledge shall be written in full and signed by every student on all submitted work (including, but not limited to, homework, projects, lab reports, code, quizzes and exams) that is assigned by the course instructor. No work shall be graded unless the pledge is written in full and signed.

“I pledge my honor that I have abided by the Stevens Honor System.”

Reporting Honor System Violations

Students who believe a violation of the Honor System has been committed should report it within ten business days of the suspected violation. Students have the option to remain anonymous and can report violations online at www.stevens.edu/honor.

Graduate Student Code of Academic Integrity

All Stevens graduate students promise to be fully truthful and avoid dishonesty, fraud, misrepresentation, and deceit of any type in relation to their academic work. A student’s submission of work for academic credit indicates that the work is the student's own. All outside assistance must be acknowledged. Any student who violates this code or who knowingly assists another student in violating this code shall be subject to discipline.

All graduate students are bound to the Graduate Student Code of Academic Integrity by enrollment in graduate coursework at Stevens. It is the responsibility of each graduate student to understand and adhere to the Graduate Student Code of Academic Integrity. More information including types of violations, the process for handling perceived violations, and types of sanctions can be found at www.stevens.edu/provost/graduate-academics.

Special Provisions for Undergraduate Students in 500-level Courses

The general provisions of the Stevens Honor System do not apply fully to graduate courses, 500 level or otherwise. Any student who wishes to report an undergraduate for a violation in a 500-level course shall submit the report to the Honor Board following the protocol for undergraduate courses, and an investigation will be conducted following the same process for an appeal on false accusation described in Section 8.04 of the Bylaws of the Honor System. Any student who wishes to report a graduate student may submit the report to the Dean of Graduate Academics or to the Honor Board, who will refer the report to the Dean. The Honor Board Chairman will give the Dean of Graduate Academics weekly updates on the progress of any casework relating to 500-level courses. For more information about the scope, penalties, and procedures pertaining to undergraduate students in 500-level courses, see Section 9 of the Bylaws of the Honor System document, located on the Honor Board website.