CS 549 Distributed Systems and Cloud Computing Syllabus

The syllabus below describes a recent offering of the course, but it may not be completely up to date. For current details about this course, please contact the course coordinator. Course coordinators are listed on the course listing for undergraduate courses and graduate courses.

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.

Text Books

Required

Various authors , *Various papers.* , We will rely on some tutorial papers, and sections from the recommended text.

Recommended

Dominic Duggan , *Enterprise Software Architecture and Design* , This book is used in CS548. In this course, we will use some of the more advanced material in the book, that is not covered in CS548.
Antonio Goncalves , *Beginning Java EE 6 with Glassfish 3* , Springer Apress, 2009

Week-by-Week Schedule

Week	Topics Covered	Reading	Assignments
1	Introduction to distributed systems and cloud computing. Cloud architectures: SaaS, PaaS, IaaS. End-to-end system design. Networks and protocol stacks.	Duggan 2.1-2, 2.7. E2E.	
2	Client-server computing. Sockets and remote procedure call.	Duggan 2.4. RPC.	
3	Distributed file systems and cache consistency. NFS, AFS. Storage in the Cloud: Google/ Hadoop file system.	Duggan 6.9.4. NFS. GFS.	A1: RMI and sockets.
4	Web services and REST. Example: Amazon S3. The JAX-RS API. Persistent cloud services.	Duggan 7.5.	
5	Failure models and failure detectors.	Duggan 2.3.	
6	Asynchrony: publish-subscribe. Server-side events and REST. Web sockets. Vert.x: Node.js for Java. Distributed snapshots.	Duggan 2.5; B 3.5; CL85	A2: P2P and REST
7	Distributed debugging. Time and ordering of events. Causal broadcasts.	B 2, B 3; L78	

Week	Topics Covered	Reading	Assignments
8	Batch cloud computing: map-reduce and Hadoop. Domain-specific languages for cloud data processing: Pig and Hive.	MR, Pig, Hive	A3: Server-side events
9	Transactions. Serializability and recoverability. Long-lived transactions.	Bien	
10	Transactions. Atomic commitment protocols: 2PC and 3PC.	Bien	
11	Highly available services. Replicated services and quorum consensus. The CAP Theorem.	D 6.9.1-2; B 3.5; VS	A4: Hadoop and Pig
12	NoSQL data stores. Table-based (Google BigTable), key-based (Amazon Dynamo), and Cassandra. The Hector API. Query processing with Map-reduce.	BT, Dyn, Cass	
13	Consensus and the Paxos algorithm. Applications in the cloud: Google Chubby, Yahoo Zookeeper.	L01	
14	Peer-to-peer systems. Distributed hash tables. Applications in multiplayer game-playing.	Chord, Mercury.	A5: Websockets