Syllabus subject to change

CS 498: Cloud Computing Applications Syllabus

Course Description

Welcome to CS 498: Cloud Computing Applications! This 16-week course was designed to give you a comprehensive view of the world of cloud computing and Big Data. **You should expect to spend 10-12 hours per week on this course**.

In this first half of the course, we cover a multitude of technologies that comprise the modern concept of cloud computing. Cloud computing is an information technology revolution that has just started to impact many enterprise computing systems in major ways, and it will change the face of computing in the years to come.

We start by introducing some major concepts in cloud computing, the economics foundations of cloud computing, and the concept of Big Data. We also cover the concept of software defined architectures, how virtualization results in cloud infrastructure, and how cloud service providers organize their offerings. After this, we cover virtualization and containers with a deeper focus, including lectures on Docker, JVM, and Kubernates. We also will compare Infrastructure as a Service offered by the big three: Amazon, Google, and Microsoft.

We then move to a higher level of cloud offering, including Platform as a Service, Mobile Backend as a Service, and even serverless architectures. We also talk about some of the Cloud middleware technologies that are fundamental to cloud-based applications such as RPC, REST, JSON, and load balancing. We'll also cover Metal as a Service (MaaS), where physical machines are provisioned in a cloud environment. Additionally, we'll introduce higher level cloud services with special focus on cloud storage services. We introduce Hive, HDFS, and Ceph as pure Big Data storage and file systems, and move on to cloud object storage systems, virtual hard drives, and virtual archival storage options, including a discussion of Dropbox.

In the second half of the course, we continue by exploring how the Cloud opens up data analytics to huge volumes of data that are static or streamed at high velocity and represent an enormous variety of information. Cloud applications and data analytics represent a disruptive change in the ways that society is informed by, and uses information. We'll discuss some major systems for data analysis, including Spark, and the major frameworks and distributions of analytics applications, including Hortonworks, Cloudera, and MapR. Additionally, we'll introduce the HDFS distributed and robust file system that is used in many applications like Hadoop, explore the powerful MapReduce programming model, and see how distributed operating systems like YARN and Mesos support a flexible and scalable environment for Big Data analytics.

This course also introduces large-scale data storage, along with the difficulties and problems of consensus in enormous stores that use large quantities of processors, memories, and disks. We discuss eventual consistency, ACID, and BASE and the consensus algorithms used in data centers, including Paxos and Zookeeper. We'll also present distributed key-value stores and in-memory databases like Redis used in data centers for performance. Next we present NOSQL Databases. We visit HBase, the scalable, low latency database that supports database operations in applications that use Hadoop. Then, we will show how Spark SQL can program SQL queries on huge data and present Distributed Publish/Subscribe systems using Kafka, a distributed log messaging system that is finding wide use in connecting Big Data and streaming applications together to form complex systems.

Near the end of the course, we'll discuss data real-time streaming and introduce Storm technology that is used widely in industries such as Yahoo. We continue with Spark Streaming, Lambda and Kappa architectures, and a presentation of the streaming ecosystem.

At the end of the course, we'll focus on graph processing, machine learning, and deep learning. We introduce the ideas of graph processing and present Pregel, Giraph, and Spark GraphX, as well as machine learning with examples from Mahout and Spark. Kmeans, Naive Bayes, and fpm are given as examples. Spark ML and Mllib continue the theme of programmability and application construction. The last topic we cover in the course is deep learning technologies, including Theano, Tensor Flow, CNTK, MXnet, and Caffe on Spark.

Course Goals and Objectives

Upon successful completion of this course, you will be able to:

- Understand what cloud computing is and why it is important.
- Get a picture of the economics of cloud computing.
- Describe Big Data and the challenges of working with it.
- Learn about many fundamental technologies that enable cloud computing, such as software defined architectures, virtualization, and containers.
- Learn about many “glue” technologies that enable access to clouds, such as web middleware, JSON, REST API, RPC, etc.
- Learn about the different levels of clouds services, which include IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service), MaaS (Metal as a Service), FaaS (Function as a Service (server-less architecture)), MBaaS (Mobile Backend as a Service (server-less architecture)), and Amazon Lambda.
- Learn about many types of cloud-based storage services, including object storage, block-level storage, archival storage, and Big Data file systems.
- Become familiar with the key concepts underlying Big Data and data streaming applications on the Cloud.

- Describe the concerns of storage, processing, parallelism, distribution, consensus, and scalability as they relate to the Cloud.
- Understand key benefits and limitations of the various technologies available in the Cloud.
- Utilize the course content to select technologies you wish to use in your work or company.

Textbook

There are no required textbooks or readings for this course.

Course Outline

Week	Dates	Topics
Week 1	January 17–22	Cloud Computing Foundations
Week 2	January 23–29	MAAS, PAAS, Web Services
Week 3	January 30–February 5	Hadoop, MapReduce
Week 4	February 6–12	Spark, Big Data Distros, HDFS
Week 5	February 13–19	Virtualization, OS Based Virtualization
Week 6	February 20–26	Containers, IAAS
Week 7	February 27–March 5	Large Scale Data Storage, CAP Theorem, Eventual Consistency

Week 8	March 6–12	Large Scale Data Storage, Distributed Key-Value Store, Scalable Databases, Publish/Subscribe Queues
Week 9	March 13–19	*No New Content – Midterm Exam preparation and Exam*
Week 10	March 20–26	*Spring Break*
Week 11	March 27–April 2	Storage Using Ceph, HIVE, Tez, SWIFT, S3, Amazon EBS, Glacier, Dropbox
Week 12	April 3–9	Streaming Foundations, Advanced Storm
Week 13	April 10–16	Streaming: Storm Internals, Spark Streaming
Week 14	April 17–23	Graph Processing
Week 15	April 24–30	Machine Learning
Week 16	May 1–May 7	*No New Content – Submit Course Project*
Week 17	May 8–May 14	*No New Content – Final Exam preparation and Exam*

Course Components

Lecture videos

Each week, your instructors will teach you the concepts you need to know through a collection of short video lectures. You may either stream these videos for playback within the browser by clicking on their titles, or you can download each video for later offline playback by clicking the download icon.

The videos usually total 1.5 to 2.5 hours each week, but you generally need to spend at least the same amount of time digesting the content in the videos. The actual amount of time needed to digest the content will naturally vary according to your background.

Quizzes

Most weeks will include 1 for-credit quiz. You will have 2 attempts for each quiz, with your highest score used toward your final grade. Your top 12 quiz scores will be used to calculate your final grade (i.e., we will drop your lowest quiz score).

Exams

This course will have 2 90-minute exams – 1 midterm exam and 1 final exam. The exams will be taken using ProctorU. More information on ProctorU can be found on the **ProctorU Exams** page.

Machine Problems

This course consists of 6 Machine Problems, which are an opportunity for you to practice your programming skills and apply what you've learned in the course. Set aside at most 8 hours to work on each of the MPs. You may need to budget more time for this if you are not familiar with Java.

Course Project

There will also be 1 culminating project due at the end of course, which you will work on in small groups. More information on the course project can be found in Week 16 of the course. You can also **click here** to access information on the course project.

Grading

Your final grade will be calculated based on the activities listed in the table below.

Activity	Percent of Final Grade
Quizzes	15%
Machine Problems	30%
Course Project	25%
Midterm Exam	15%
Final Exam	15%

Letter Grade	Percent Needed	Letter Grade	Percent Needed	Letter Grade	Percent Needed
A+	95	**B+**	80	**C**	60
A	90	**B**	75	**D**	55
A-	85	**B-**	70	**F**	<55

Additional Course Policies

Student Code and Policies

A student at the University of Illinois at the Urbana-Champaign campus is a member of a University community of which all members have at least the rights and responsibilities common to all citizens, free from institutional censorship; affiliation with the University as a student does not diminish the rights or responsibilities held by a student or any other community member as a citizen of larger communities of the state, the nation, and the world. See the University of Illinois Student Code for more information.

The CS department also maintains a policies handbook for graduate student. For more information, see the Graduate Student Handbook.

Additionally, all Coursera learners are required to follow an Honor Code and a Code of Conduct. Please review both of these items before commencing your studies.

Academic Integrity

All students are expected to abide by the campus regulations on academic integrity found in the Student Code of Conduct. These standards will be enforced and infractions of these rules will not be tolerated in this course. Sharing, copying, or providing any part of a homework solution or code is an infraction of the University's rules on academic integrity. We will be actively looking for violations of this policy in homework and project submissions. Any violation will be punished as severely as possible with sanctions and penalties typically ranging from a failing grade on this assignment up to a failing grade in the course, including a letter of the offending infraction kept in the student's permanent university record.

Again, a good rule of thumb: *Keep every typed word and piece of code your own*. If you think you are operating in a gray area, you probably are. If you would like clarification on specifics, please contact the course staff.

Disability Accommodations

Students with learning, physical, or other disabilities requiring assistance should contact the instructor as soon as possible. If you're unsure if this applies to you or think it may, please contact the instructor and Disability Resources and Educational Services (DRES) as soon as possible. You can contact DRES at 1207 S. Oak Street, Champaign, via phone at (217) 333-1970, or via email at disability@illinois.edu.

Late Policy

Late homework and homework by email will not be accepted by the TA or the instructors under any circumstances.