ECE 494-516 Fall 2023 – Cloud Computing and Big Data

Professor: Maria Striki

Syllabus

Instructor:

Name: Maria Striki

Email: maria.striki@rutgers.edu

Office: ECE 115

Teaching Assistant: TBA

Class Meets:

Time: Tuesday/Thursday @ 2.00-3.20pm

Place: Core 538

Final Exam: No Final – Main Project Presentations only

Course Outline (Tentative)

1: Introduction to cloud computing

2: Overview of parallel and distributed computing

3: The MapReduce programming model and intro to Hadoop

4. Overview of locality and introduction to Spark

5. Advanced topics in Spark and applications

4: Hadoop examples

5. The CAP theorem

6: Distributed systems concepts: synchronization, mutual exclusion, termination detection, clocks, event ordering (time and ordering), locking, multicast communications, failure detectors, memberships, leader election, … and distributed systems paradigms: Gossip protocols, grids, P2P systems, Paxos and Consensus, sensor networks, …

7: SQL, NoSQL (tentatively)

8: Students Paper Presentations and Reviewing

9: Mid-way Projects Presentations from class (end Oct-start Nov)

10: Data analytics on cloud

11a: The seven dwarfs of computing: sparse and dense linear algebra and Machine learning in the cloud
11b: The seven dwarfs of computing: structured and unstructured grids

11c: The seven dwarfs of computing: graphs and graph processing in the cloud (you will have the chance to see how this works at least)

12: Data streaming in the cloud: Spark Streaming, Storm… (tentative, if we have enough time)

13: Distributed File Systems, distributed Memory Systems (tentative, if we have enough time)

14: Cloud Computing Programming Platforms Selected and Presented by Groups of Students throughout the semester.

15: Final Project presentations (beginning to mid Dec)

Pre-requisites:

Prerequisites:

Data-structures, programming (preferably C, Python, Java) and **Systems Design** are very important. You must have good programming and hacking skills.

- E.g.: 16:332:351 (Programming Methodology-II & Data Structures for Undergrads and Grads).

Helpful past courses/experience:

The following taken courses can facilitate your understanding of many concepts and techniques to be discussed in this course:

- Operating Systems, Computer Architecture, Parallel and Distributed Computing.

Computer Architecture Literature

No official Text book : Lecture Notes and handouts (required)

Random Reads:

- **Distributed Systems: Concepts and Design**, by George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, Fifth Edition, Addison-Wesley, Morgan Kaufman, ISBN 13: 978-0-13-214301 (not required),
- **Distributed Operating Systems, by Andrew S. Tanenbaum** (not required -- lecture notes and handouts are adequate),
- **Specialized Textbooks or Lecture Notes (or Seminar Series) on Cloud Computing:** TBA

Must do and Must Haves for ECE 494-516

1: Good programming and hacking skills (you may try to install and run Hadoop on your own and see where you stand…). However, for those that do not want to get involved with deep hands-on work, I will be providing a docker-container where Hadoop and Spark is installed and is easy to add to your machine.

2: Will need to obtain familiarity with a few Cloud programming languages such as Spark and Hadoop. We will do so through lecture discussions and corresponding literature, but must be willing to place some effort to get good familiarity following the pace of the class.

3: Be comfortable reading, presenting, and reviewing research papers and bold enough to introduce or present new technologies and tutorials to the class.

4: Be a good team player within your group and serve as a fair and smart judge/reviewer for your fellow groups.

Course Goal and Description

This course will introduce students to fundamentals of Cloud Computing Concepts. It will emphasize both in the distributed communication and coordination aspects of clouds and the new parallel and distributed technologies, concepts, techniques, and algorithms that make-up the Cloud Infrastructure as we have built it up or … envision it today.

We will investigate more or less in depth how each individual component works and contributes to the Cloud and also how it works in symphony with all the remaining components of the Cloud Framework.

We will ask ourselves at the beginning of the class… What is a Cloud? We will attempt to describe. We will ask ourselves again at the end of the course: What is a Cloud after all? And the goal is that everyone now will color this answer with his/her own personal experience on working, touching, approaching, altering some parts of the Cloud!

Key Words: heavily and pleasantly collaborative, flexible w.r.t. syllabus, hands-on: programming assignments and one long ongoing project, recitations and mini-labs during many of the lectures.

Mode of learning: As there is a lot of new material introduced for many of you, and some of them are rather "complicated" to use, I am always going to gauge the needs of the class before moving onwards. My own goal for you taking this class is to learn while practicing, to enjoy while learning, to get confidence, and find special topics that interest you, to have the initiative to research and hack on concepts that you select!

Positive Side-Effect: The last five years I have been teaching this class, numerous students got employed or got numerous interviews, as they worked on concepts and projects that happen to be hot topics in the industry right now!

Assignments and Time-Lines

1: 3-4 programming assignments/homework and lab recitations done in groups of 1-3. In class presentation-discussion of selected assignments. **It can be more assignments if we break them to shorter pieces. Note that the labs merge with the 3-4 programming assignments at some point and they become the same assignment.

2: A half-semester long on-going project of your selection with your team: midway presentation and final presentation from ALL group members.

3: A number of in class (a few many be take-home) quizzes (notified beforehand). Longer quizzes may serve as mini-midterms.

4: No in class half-semester Midterm, No Final Exam.

5: 1-2 paper presentation per group, a number of paper reviews of other peers during their presentation, and a number of paper summary reviews and answers to questions (randomly selected from your peers).

6: Programming/Platforms/Technologies presentation of a number of selected or requested tutorials on programming languages, tools, technologies (e.g., SQL, noSQL, Pig, Hive, etc).

7: Class Participation: contribution to our class discussions, responsible, fair, and educated reviews and discussion of the work of your peers, detailed presentations... This class is highly collaborative and your active participation is very much appreciated and rewarded…

Group Project:

October 5th-12th: submit your idea to get GO or no GO): Write a **short proposal** of the deployment of your idea and tools and methods required, group members' specific tasks and timelines described.

November 2th – 9th: midterm report and short (in-class) **presentation of current progress**

December 12th to 15th (or earlier if needed - TBA): Final Reports **and Presentations**

Paper Reviews: start mid September or later

Paper Presentations: start end of September or later

Grading Guidelines and Policies

Grading (to be revisited beginning of October):

- Programming assignment/Labs/Homework: ~22-28%
- Paper Presentations: 8%-12% (depending on whether you present 1 or 2 papers)

- Paper Reviews: ~12%
- Quizzes: ~18 - 28% (depending on how many quizzes we take)
- Mid-Semester Project and Presentation: ~20%
- Participation: X%
- Volunteer Presentations: TBA, part of participation
- NO FINAL EXAM ----
- Undergraduate students will be graded using a slightly different scale on presentations, reviews, and a number of research questions. Regarding quizzes, there may be questions addressing only to undergraduate students and different questions addressing only to graduate students.

In-class participation:

1: In class participation is a very important factor. The teaching style is very interactive and truly based on students' contributions.

2: You assimilate the class material better, clarify concepts, facilitate your colleagues in clarifying their own questions, and help the professor direct the class content better, and tailor it closer to the students' needs and requirements.