ITCS-6190/8190: Cloud Computing for Data Analysis
Spring 2017

1 Course Information

Instructor:	Srinivas Akella
Email:	`sakella@uncc.edu`
Office:	Woodward 205B, x7-8573
Office hours:	Monday 11:00am-12:00pm

Lectures:	Monday, Wednesday 9:30am–10:45pm
Classroom:	Woodward 130
Credits:	3
Prerequisites:	ITCS 6114 or permission of instructor.

Teaching Assistant:	Walid Shalaby
Email:	`wshalaby@uncc.edu`
Office:	Woodward 432
Office hours:	Friday 12:00pm-1:00pm

Course information: Course announcements and information will be available on Canvas.

Readings:
There are **four** textbooks:

Hadoop: The Definitive Guide, fourth edition, by Tom White, O'Reilly, 2015.

Learning Spark: Lightning-Fast Big Data Analysis, by Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia, O'Reilly Media, 2015.

Introduction to Information Retrieval, by Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, Cambridge University Press, 2008.

Mining of Massive Datasets, second edition, by Jure Leskovec, Anand Rajaraman, and Jeff Ullman, Cambridge University Press, 2014.

We will supplement the textbook readings with additional readings and research papers for broader coverage of the course topics and to reflect recent work in the field.

2 Course Overview

This course will introduce the basic principles of cloud computing for data-intensive applications. It will focus on parallel computing using Google's MapReduce paradigm on Linux clusters, and algorithms for large-scale data processing applications in web search, information retrieval, computational advertising, and scientific data analysis. Students will read and present research papers on these topics, and implement programming projects using Hadoop, an open source implementation of Google's MapReduce technology, and Apache Spark.

Prerequisites: ITCS 6114 or permission of instructor. Prerequisites for the course are familiarity with Java and Python/Scala, Unix, Data structures and Algorithms, Linear Algebra, and Probability and Statistics. Students are expected to have good programming skills including knowledge of data structures and algorithms, and a solid mathematical background.

3 Topics

This is the tentative schedule of topics to be covered.

Class(es)	**Topic**
1-6	Distributed computing
7-14	Information retrieval, web search
15-20	Data analysis algorithms (clustering, classification)
21-28	Advanced topics, student presentations

4 Grading

The course activities consist of homework and occasional in-class quizzes, programming assignments, a final course project, and a midterm and a final exam. Students taking the Ph.D. version of the course (ITCS-8190) will also be required to make a classroom presentation on a topic to be selected with the instructor. Homework assignments will be a combination of written homeworks and reading reports on research papers. There will be five or six programming assignments, which will involve a substantial amount of programming. Students will have one or two weeks to do each programming assignment. Programming assignments must be submitted by 11:59:59 pm on the scheduled due date. The final course project will be a significant implementation project, to be selected in consultation with the instructor. In addition to a demonstration and presentation of the project, students must submit a written summary of the project.

Graduate students taking ITCS-6190 and ITCS-8190 will be graded separately. The tentative grading scheme for the course is as follows:

ITCS-6190
Homeworks and quizzes: 10%
Programming assignments: 40%
Course project: 20%
Class participation: 5%
Midterm and Final exams: 25%

ITCS-8190
Homeworks and quizzes: 10%
Programming assignments: 35%
Course project: 20%
Class presentation and participation: 10%
Midterm and Final exams: 25%

The maximum lower bound cutoffs for A, B, and C grades will be 90%, 80%, and 70% respectively. Lower cutoffs may be established at the end of the semester when assigning grades. If you feel there is a grading error on a homework, quiz, programming assignment, or exam, you should bring this to the attention of the instructor as soon as you receive your grade and no later than a week after you receive the grade. **All grades will be treated as final two weeks after they are issued.**

Attendance Policy: Attendance in class is expected and students are responsible for knowing all material covered in class. A portion of the grade will be determined by class attendance and participation.

Lateness Policy: If there is a good reason you will need an extension on a homework or a programming assignment, contact me **in advance**. Each student will be given one day (whole or partial) of grace for late homeworks. Each student will be given three days (whole or partial) of grace for late programming assignments. Use these late days carefully. Once you have exhausted these days, **late assignments will not be accepted** without a written excuse from the Dean of Students' office. This includes late days for illnesses, plant trips, etc. USE YOUR LATE DAYS WISELY, IF AT ALL.

As an example, if you submit your 1st program 26 hours late, you will have used two late days and have only one day left. If you submit your 2nd program 5 hours late, you will have used your last late day. If you then submit your 3rd program 1 minute late, it will not be accepted. USE YOUR LATE DAYS WISELY, IF AT ALL. **Students are responsible for keeping track of their usage of late days.**

5 Academic Honesty

Students are encouraged to discuss course material to improve their understanding of the subject. However they must submit their own work for homeworks, quizzes, programming assignments, exams, and the final course project.

Students are allowed to discuss homeworks. Students must however write their homeworks individually. Copying on this course work is not allowed and will result in a 0 for the submission plus a 5 percentage point penalty on the semester grade and a report to the Dean of Students office.

To ensure academic integrity on programming assignments while permitting students to learn and get help from each other, the following rules will be in force. Students are allowed to work together in interpreting error messages and in finding bugs, but NOT in writing code. Students may not share code, copy code, or discuss code in detail while it is being written or afterwards. Students may not use code obtained on the web or from other sources (unless permitted by the instructor). Shared or copied code is easy to spot manually and is easily detected using a variety of software tools. Students caught illegally collaborating in writing code or violating the above rules will receive a 0 on the assignment plus a 5 percentage point penalty on their semester grade and a report to the Dean of Students office. Students caught a second time will receive a U (failing grade) in the course and will be reported to the Dean of Students office.

Copying, sharing answers, or using disallowed materials during a quiz or an exam is cheating, of course, and will result in a 0 on the exam, plus a 5 percentage point penalty on the semester grade and a report to the Dean of Students office.

Any student violating these academic honesty rules for a second time will receive a U in the course and will be reported to the Dean of Students office.

Students are responsible for knowing and observing the UNC Charlotte Code of Student Academic Integrity, which defines various forms of academic dishonesty and procedures for responding to them. The Academic Integrity Code is available online at
`http://legal.uncc.edu/policies/up-407`. Students found in violation of academic honesty policies will receive a failing grade for the course.