COSC6339: Big Data Analytics  
Future name: Big Data Systems OR Big Data 

Instructor: Carlos Ordonez

1 Mode of Instruction

The course will be hybrid, with 40%-50% online lectures. Lectures will cover theory, principles and programming practices.

2 Course Contents

This is a ”systems” + ”data science” graduate course, which covers theory and programming to store and analyze big data, covering all available systems, platforms and technologies today: data science languages, parallel database systems, NoSQL and Hadoop. Past data mining research has been expanded to analyze big data, mixing structured and semi-structured content, with more complex statistical and machine learning models and exploiting parallel processing.

1. Big Data overview: Evolution from 3 Vs to 5 Vs (3+2), Data Lakes vs Data Warehouses vs Data Swamps, Data integration, pre-processing and cleaning, Parallel and Distributed Computing, Big Data Analytics problems: machine learning, graphs, streams.


3. Data Science programming languages: programming language definition and philosophy. Tasks: exploring data, data pre-processing and data cleaning, functional constructs, runtime in the OS, parallel processing in multicore CPUs and GPUs, main data structures (data frames, matrices).

4. Parallel DBMSs: parallel processing in multicore CPUs, parallel processing in a cluster with distributed storage, OLTP versus cubes (3+2), Data warehousing and denormalization, advanced SQL queries (pivoting, horizontal aggregation, keyword search, recursive queries), advanced programming (embedded SQL, UDFs, internal C), row/column/tile storage, indexing versus row ordering, ETL and pre-processing, machine learning and graph algorithms.

5. Hadoop ecosystem: file systems (HDFS, Linux Posix, Google), subsystems (Yarn, Storm, Zookeeper), storage (Hive, SPARQL, Parquet), containers vs virtual machines, text versus numeric processing, NoSQL and non-DBMS (MapReduce, Spark, MongoDB), Graph analytic systems, Search Engines (mainly Google and Yahoo, IR models, architecture, keywords, page rank, web spider), programming language definition and philosophy.

The course will require reading CS research papers, mainly from Big Data and Database systems conferences and journals, available on DBLP, IEEE and ACM digital libraries. There is no textbook, but [2] and [1] are recommended.
3 Academic Background

This is a course combining "systems", "machine learning" and "data science". Therefore, it is desirable students have background on most of these areas.

Pre-requisites: It is strongly encouraged, but not required, that the equivalent of COSC2436 (basic data structures and algorithms), COSC3380 (Database Systems), COSC3360 (Operating Systems), Machine Learning or Data Science, were taken before.

Familiarity with Linux, Python and C++ development. All programming homeworks must work in Linux (read below).

4 Grading

This is a tentative plan, which may change, depending on students CS background, performance, interests and COVID complications. Keep in mind that this course will require a significant programming effort, which may be harder and will take more time than a test-based course, where tests have a higher weight.

• 60%: 2 programming homeworks (team-based):
  HW1 (40%): fast/scalable machine learning or graph analytics, comparing a data science language (Python) and a Database System (SQL, MQL, or GQL). Potential ML projects: data summarization, correlation, gradient descent, regression, classification, time series and deep neural networks on large data sets that cannot fit in RAM. Potential graph projects: TC, neighborhood analysis, clique detection. Programming language: Python, calling C++ code wrapped as Python libraries. System: your own computer with minimal hardware (multicore CPU, at least 4 GB RAM, 256 GBs storage); using a more powerful computer is acceptable, but will not produce any significant improvement in programming effort. OS: Windows to get started, Linux for final version. DB system: Postgres, Vertica or MongoDB (depending on students background and interest). All software can be installed on your computer to get started. However, your program will be tested on a Linux server in the cloud.
  HW2 (40%): Hadoop Stack. Projects: computing an interesting machine learning model combining text, images and databases. Model: AI predictive models including: deep neural networks, transformers, generative models. Programming language: Python+SQL, calling, numerical, machine learning, text, image and NLP libraries, as needed. System: a traditional “Big Data” system with data stored on HDFS: Spark. OS: Linux; this Linux system will be provided by UH.

• 20%: midterm exam around 11th week of classes; the date will be announced in advance. The exam will be in the classroom during official lecture time. This exam will have 10 questions, which require a short answer or writing a few lines of source code (Python, C++, SQL). There is no final exam. The exam will be closed-everything (no computer, smartphone, notes or textbook allowed). Attending class, solving HWs and reading papers/textbook throughout the semester will be enough preparation.

• Extra points (up to 5), for class participation: answering instructor theory or ”systems” programming questions. Typically, students who participate a lot will push their grade one level (e.g. B to B+).

Homework details: There will be 2 different analytic problems for each HW. In general, each analytic problem will be solved in the typical data science language or system used today (Python, R). That is, there will be 4 programming homeworks, where each team will solve 2. The homeworks will be assigned by the instructor, including analytic goal, programming language and target system. The instructor will randomly assign each homework to each team.

HW grading: homeworks will be graded in 2 ways. TAs will run each program to verify results and each team will prepare a 5-minute video. Each submission should have comments and instructions to run from the command line (no GUI).
Exam topics: parallel efficiency, external algorithms, distributed architectures, memory management, I/O optimization, efficient linear algebra, accelerating numerical methods and graph algorithms.

Data: We will use traditional data sets coming from the UCI machine learning repository or the web.

Teams: Programs for each HW will be developed in teams of 2 students. Students can choose their team partner, but the instructor can assign students to teams if needed.

Final grade: all programs must be submitted and work in order to get at least C+. The scale to assign letter grades will be standard (A is 90, B is 80, and so on).

References