EECS4415 Big Data Systems

Fall 2019

Course Website

www.eecs.yorku.ca/~papaggel/courses/eecs4415/

Course Description

Storing, managing, and processing datasets are foundational to both computer science and data science. The enormous size of today's data sets and the specific requirements of modern applications, necessitated the growth of a new generation of data management systems, where the emphasis is put on distributed and fault-tolerant processing. New programming paradigms have evolved, an abundance of information platforms offering data management and analysis solutions appeared and a number of novel methods and tools have been developed. This course introduces the fundamentals of big data storage, retrieval, and processing systems. As these fundamentals are introduced, exemplary technologies are used to illustrate how big data systems can leverage very large data sets that become available through multiple sources and are characterized by diverse levels of volume (terabytes; billion records), velocity (batch; real-time; streaming) and variety (structured; semi-structured; unstructured). The course aims to provide students with both theoretical knowledge and practical experience of the field by covering recent research on big data systems and their basic properties. Students consider both small and large datasets because both are equally important and justify different trade-offs. Topics include: software frameworks for distributed storage and processing of very large data sets, MapReduce programming model, querying of structured data sets, column stores, key-value stores, document stores, graph databases, distributed stream processing frameworks.

Topics

Topics include:

- data-driven organizations
- data ingestion
- data quality
- data storage (data lakes, RDBMS, columnar DBMS, NoSQL, HDFS, Key-Value stores, object storage)
- data definition (CAP theorem, schema-on-read, schema-on-write)
- big data analytics architectures
- batch processing
- interactive query processing
- data stream processing
- unified processing engines
- tools/systems for big data analytics (examples: OpenRefine, Apache Hadoop/MapReduce, Google BigTable/BigQuery, Twitter Storm/Huron, Apache Spark)

Instructor

Manos Papagelis
Email: papaggel@eecs.yorku.ca, papaggel@gmail.com
Website: http://www.eecs.yorku.ca/ ~papaggel

Class Hours

Lectures: *Wed, 17:30pm-20:30pm at VH 3006 (Vari Hall)*
Office Hours: *Tue, 13:00pm-14:00pm or by appointment (LAS3050, Lassonde building)*

Class Attendance
Attendance of lectures is expected but not required.

Prerequisite Courses
The following are prerequisites for the course:

- EECS-3421 (Introduction to Database Systems)
- EECS3101 (Design and Analysis of Algorithms)
- General Prerequisites

Textbooks
There is no single text for this course. The course will rely mainly on the following suggested textbooks:

- (**mmdb**) Mining of Massive Datasets, 2nd Edition. Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman. 2014. Cambridge University Press. ISBN: 9781107077232
  Note: The aforementioned book is freely available online.
- (**bdb**) Big Data 2.0 Processing Systems: A Survey. Sherif Sakr. 2016. SpringerBriefs in Computer Science. ISBN: 978-3-319-38776-5
- (**ncmb**) Networks, Crowds, and Markets: Reasoning About a Highly Connected World. David Easley, Jon Kleinberg. 2010. Cambridge University Press. ISBN: 9780521195331
  Note: The aforementioned book is freely available online.

In addition, a number of recent research papers in the area of big data systems will be distributed every week.

Communication
The main communication tools for the class will be the *course website* and *Piazza*.

- *Course Website*: Most class materials are available on the course web site; be sure to check regularly. The page has also a link to a discussion board. We are using Piazza.
- *Piazza*: Instead of a discussion board, we are using Piazza, a free Q&A platform. Piazza can get you fast, accurate response to your questions – but it only works if everyone participates! We will also use Piazza to post announcements and updates, so both the website and Piazza is required reading. See below for Piazza signup and class links:

  Signup link: http://piazza.com/yorku.ca/fall2019/eecs4415

  Note: You will need to sign up with your school email, ending in *yorku.ca*. If you do not have a school email address, please contact your instructor and request to be enrolled with your personal email.
- *Email*: Please use email for personal issues and the discussion board to ask general course-related questions. Include "eecs4415" in all email subject lines to ensure your message is correctly filtered and filed. An informative subject line like "eecs4415: Question related to X" really helps. I try to respond to email frequently. However, due to volume, it may take longer, especially on weekends and near due dates.

Grading Policy

Work	Weight	Comment
Readings	10%	1% each
3 Assignments	30%	10% each
Project	30%	Large project consisting of: proposal, milestone report, final presentation, and final report
Final Exam	30%	You must get >=40% to pass the course

Project Grading Policy (updated)

Milestone	Weight
Project proposal	20%
Project final report and source code	50%
Project final in-class presentation	30%

Final Examination
A written final exam will be given between 6-21 Dec (to be determined during the term).

Working with a Partner
You have the option of partnering with other (currently enrolled) students for your assignment and project, and we encourage you to do so. The ability to work effectively in a team will be very important in your career, and that involves many skills beyond the purely technical aspect of creating working code. You may choose your own partner(s). If you do have a partner, submit only a single copy of your work. Jointly submitted assignments will be graded in the usual way and all partners will receive the same mark. Working with a partner has the potential to lighten your workload or to increase it, depending on how well you work together. Be aware that simply dividing the work and assembling your separate pieces at the end is a poor strategy for completing successful assignments. And of course, you are responsible for learning the course material underlying all parts of the assignments. You will have the most success if you truly work together.

Assignment Policies
Here assignment refers to project milestones. You must make sure that all your assignments are running and are sufficiently documented. Code that doesn't compile, fails to run or lacks documentation, will be marked as not working.

Late Work Policy
Here assignment refers to project milestones. The late policy is strict. All assignments will be submitted electronically. Late assignments will be handled based on a system of "*grace days*", as follows: Each student begins the term with 3 *grace days*. One grace day is 24 hours. If an assignment is due at 10:00 p.m. on a Friday then an assignment handed in by 10:00 p.m. on Saturday uses one grace day. The grace days are intended for use in emergencies (e.g., system failure or illness). Do not use all of them to buy an extension because of a busy week or you will be out of luck in a true

emergency. Assignments submitted after the due date when all grace days have been used will receive a grade of 0.

If you are at risk of missing a deadline due to a busy week, rather than using your grace days you should hand in a working (and tested) version of a simpler program. In the event of an illness or other catastrophe, get proper documentation (e.g., medical certificate), and contact me (by email or in person) as soon as possible. Do not wait until the due date has passed. It is always easier to make alternate arrangements before the due date or test day.

Assignments are submitted electronically and will often be tested using an automated testing program; you must follow the submission instructions exactly. If you do not, you will most likely lose substantial marks on the assignment. If you find you have submitted the wrong file or omitted a file, please notify your instructor as soon as possible.

Remarking
Here assignment refers to project milestones. If you feel an error was made in marking an assignment or test please submit a remark request. Requests for remarking must be submitted using a university remarking request form explaining what your concern is **no later than a week after** the assignment (or test) has been returned back.

Academic Offenses
All of the work you submit must be done by you and your work must not be submitted by someone else. Plagiarism is academic fraud and is taken very seriously. The department uses software that compares programs for evidence of similar code. Please read the Rules and Regulations from the York University's Academic Integrity and the York University's Senate Policy on Academic Honesty documents.

Accessibility Needs
York University is committed to accessibility. If you require accommodations for a disability, or have any accessibility concerns about the course, the classroom or course materials, please contact York University's Counselling & Disability Services.