CSCI E-88 Principles Of Big Data Processing

Harvard Extension School, Fall 2019

Syllabus

DRAFT - SUBJECT TO CHANGE

Instructor: Marina Popova, M.S., ALM, Principal Software Engineer, Yottaa, Inc.
Teaching Assistants: TBD
Location/Time: TBD, Tuesdays 5:40pm - 7:40pm
Optional Weekly Sections: to be arranged (online only, via Zoom)

Course Description:
The goal of this course is to learn core principles of building highly distributed, highly available systems for processing large volumes of data with historical and near real-time querying capabilities. We cover the stages of data processing that are common to most real-world systems, including high-volume, high-speed data ingestion, historical and real-time metrics aggregation, unique counts, data de-duplication and reprocessing, storage options for different operations, and principles of distributed data indexing and search. We review approaches to solving common challenges of such systems and implement some of them. The focus of this course is on understanding the challenges and core principles of big data processing, not on specific frameworks or technologies used for implementation. We review a few notable technologies for each area with a deeper dive into a few select ones. The course is structured as a progression of topics covering the full, end-to-end data processing pipeline typical in real-world scenarios.

Prerequisites:

- Students must be comfortable with intermediate programming in at least one language, preferably Java, Python, or Scala. Students should be comfortable with basic data structures, functions, basic multi-threading, and build and dependency management (Maven or Gradle for Java, virtualenv for Python). Most of the examples in lectures are in Java and Python.
- Students should be comfortable with basic usage, package/software installations, and administration and troubleshooting on Unix-like systems (Linux, any flavor, MacOS).

- Students should be comfortable with cloud environments like Amazon web services (AWS) cloud and container frameworks like Docker (or VMware, VirtualBox).
- Their laptops should have 64-bit operating systems, and have at least 8 central processing units (CPU) and 8G random-access memory (RAM).
- Students should complete the self-assessment assignment zero, available on the syllabus, to determine if they are ready to take the course.
- Courses such as CSCI E-7, CSCI E-10a, CSCI E-50, and CSCI E-90, or equivalents, are also recommended.

Assignment #0 - Self-Assessment

The goal of this assignment is to determine whether you have enough programming experience to complete this course assignments - successfully and stress-free :) .
It is not to be submitted - just completed as an exercise before you decide to enroll into the class.

Problem 1.a

- setup AWS account (if you don't have one already) and create an S3 bucket
- write a program (Java, Python or Scala) that will do the following:
  - generate a file with 100 lines:
  - each line should have 3 random numbers in the range [0-10]

    your lines would look like:
    "1 7 3"
    "2 1 3"
    ...

Problem 1.b

- using AWS APIs, upload created file into your S3 bucket - verify the content of the file is correct there
- next, also using AWS APIs, download the file from your S3 bucket - verify the downloaded local file is the same as the original created file

Problem 2

write a program (Java, Python or Scala) that will do the following:

- use file generated in Problem 1 as input
- for each line, calculate its "key" as following: key = sum of all three numbers from the line
- your 'key' is a number; find the max and min key of your data set
- create a file on your local file system and write each line into this file, prepending it with the calculated 'key'
- write the lines in the descending order by 'key'

your resulting local file content should look like:

"29: 10 9 10"
"29: 10 10 9"
"25: 8 7 10"
...
"1: 0 0 1"

Problem 3:

- Create a small EC2 instance from an AWS AMI with CentOS or a Docker/ other VM
- Deploy (copy) your Problem 1 and Problem 2 applications on that instance
- Make sure they run Ok

Problem 4:

- install Docker on your laptop/PC
- pull a Tomcat image
- create a simple html page with an image of your pet (or any other image)
- start the Tomcat docker container and replace the standard Tomcat's index.html page with your own one
- make sure you can access your index.html page

Lectures and Sections:
This class can be taken 100% remotely, on-campus or as a mix of on-campus and remote access. All lectures and sections will be available as live streams online and will be recorded, with access to all recordings online. Section will be conducted online only via Zoom, with the recordings available right after the section. Details/agenda/timing of each section will be announced in advance.
More details about access options will be posted closer to the course start.

Communication:
Class communication will be done via Piazza Discussions and Canvas.
Contact info:

- Marina Popova: map685@g.harvard.edu
- TAs: TBD

Piazza link and signup:

- TBD

Assignments:
There will be weekly assignments, a small Mid-term Quiz (open-book, un-proctored) and a Final Project. Detailed requirements for each assignment will be posted on Canvas and communicated in the Lectures and Sections.
Weights towards the final class grade:

- Assignments: 80%
- Mid-term Quiz: 5%
- Final Project: 15%

Grading and Late Policies:

- There will be up to 2 late days allowed for each assignment, with 15% grade penalty per day. After 2 days - assignments are not accepted anymore.
- For the final grade - the lowest grade from any of the assignments will be dropped (does not apply to Mid-Term Quiz and Final Project)
- For some assignments - there will be options to earn extra points
- Grades are not curved
- There will be no extensions/late days allowed for the Final Project, and no EXT grades
- Students enrolled as **Noncredit**: you are **not** expected to complete any of the homeworks , quiz and Final Project. You are welcome to work on them - but they will not be graded
- Students enrolled as **Undergraduate**: you are required to finish all homeworks (subject to the same policies outlined above), but are not required to do the Quiz and the Final project

Texbooks and Reference Materials:

There are no required textbooks. References to optional online readings and books will be provided in each lecture. Some materials from the following books will be used in the Lectures (with full credits to the original source of information, in-line with copyright laws and in some cases with an explicit permission from the authors!):

- "Big Data" by Nathan Marz and James Warren (https://www.manning.com/books/big-data)
- "Hadoop Application Architecture" by Jonathan Seidman, Gwen Shapira, Ted Malaska, Mark Grover (http://shop.oreilly.com/product/0636920033196.do)

They are not a required reading since all relevant materials are presented in the Lectures, but are highly recommended for those who would like to dive deeper into the subject

Weekly Schedule of Topics:

this list is not final and topics can shift around and/or change - please re-visit Canvas site for the latest information often

Weeks	Topics
Week 1 - Sept 3	• Introduction - what is Big Data processing? • Evolution of applications: from classic 3-tier to massively distributed Big Data ones; • Classification of Big Data processing pipelines
Week 2 - Sept 10	Scaling Concepts and Parallel Processing • Vertical and horizontal scaling • Shared state and shared data management • Distributed cache concepts (Redis as an example)

	● Basics of Parallel Processing
Week 3 - 4 - Sept 17-24	MapReduce as a generic parallel processing model ● Example implementations: Java 8 Streams ● Hadoop MR Distributed Data Persistence 1 ● Master Datasets: core concepts, modeling, requirements, storage formats (Avro, Parquet, ..) ● HDFS
Week 5 - Oct 1	Batch Tier: Processing and Batch Views ● Batch processing concepts, requirements, algorithms and techniques ● Batch views data modeling, storage options ● Illustration with Spark batch processing
Week 6 - Oct 8	Distributed Data Persistence 2 ● RDBMS vs NoSQL storage systems - core concepts ● CAP and PACELC ● Overview and classification of NoSQL systems
Week 7 - Oct 15	Collection Tier and Data Ingestion ○ Core concepts, common patterns and concerns ○ Example frameworks (Fluentd, Scoop, …) ○ Deep dive: Flume
Week 8 - Oct 22	Messaging Tier ● traditional message system (like MessageQ, Tibco) vs Big Data - centric ones (Kafka) ● Kafka: deep dive
Week 9 - Oct 29	Stream Processing ● Stream vs static data processing - core concepts, requirements, concerns, storage options for RT Views ● Algorithms for RT processing ● Stream processing with Spark and Kafka Streams **Mid-Term Quiz is given out**
Week 10 - Nov 5	Stream: windowed processing with watermarks Time Series processing with Cassandra
Week 11 - 12 - Nov 12-19	Distributed Indexing, Search and Visualization ● core indexing and search concepts and algorithms ● Data ingestion with Connectors ● Illustration with ElasticSearch: deep dive ● Visualization with ES Kibana **Final Project is given out**

Nov 26	TBD
Nov 27 - Dec 1	Thanksgiving Break
Week 13 - 14 Dec 3 - Dec 10	Putting it All Together - Real-World Big Data Processing use cases and example architectures - up-to-the-moment hot trends overview
Week 15 - Dec 17	Final Projects Review and Presentations

Harvard Extension School Standards

Academic Conduct
Unless otherwise stated, all work submitted as part of this course is expected to be your own.
You may discuss main ideas of problem with other students on Piazza but you must implement the actual solution by yourself.
Prohibited behaviors include:

- copying all or part of another person's work, even if you subsequently modify it
- posting full source code of your solution on Piazza (or any other way of sharing solutions)
- copying solutions from the Web

You are also responsible for understanding Harvard Extension School policies on academic integrity: www.extension.harvard.edu/resources-policies/student-conduct/academic-integrity

Not knowing the rules, misunderstanding the rules, running out of time, submitting "the wrong version", or being overwhelmed with multiple demands are not acceptable excuses. There are no excuses for failure to uphold academic integrity. If we believe that a student is guilty of academic dishonesty, we will refer the matter to the Administrative Board of the Extension School, who could require withdrawal from the course and suspension from all future work at the School.

We also expect you to know and adhere to the general policies and procedures of the Extension School. You can find more information here:
http://www.extension.harvard.edu/resources-policies

Accessibility Services
The Extension School is committed to providing an accessible academic community. The Accessibility Services Office offers a variety of accommodations and services to students with

documented accessibility issues. For more information, please visit:
www.extension.harvard.edu/resources-policies/resources/disability-services-accessibility