CSCI E-88 Principles Of Big Data Processing

Harvard Extension School, Fall 2017

Syllabus

DRAFT - SUBJECT TO CHANGE

Instructor: Marina Popova, M.S., ALM, Principal Software Engineer, Yottaa, Inc.
Teaching Assistants: TBD
Location/Time: Harvard Hall 104, Thursday 7:40pm - 9:40pm
Optional Weekly Sections: TBD

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:

- programming in at least one language, preferably Java, Python, or Scala. Most of examples in the lectures will be in Java. Please complete the Assignment #0 - the Self-assessment assignment, not graded, to determine whether are you ready to take on this course
- basic usage, package/software installations and administration on Unix-like systems (Linux, any flavor, MacOS), as we will run most of the programs on Linux/MacOS. For

example: when we ask you to install a basic Cassandra or Kafka cluster - we expect you to do it by using online tutorials and your own research. We will not be providing step-by-step instructions; only pointers to the important required configuration settings. Of course, you are welcome to ask questions/clarifications on the discussion board

- cloud environments like AWS cloud and/or virtualization frameworks (like VMWare, VirtualBox, Docker). We will be using AWS services quite extensively for Lecture and Section demos as well as homeworks. For homeworks - you are free to use the other mentioned virtualization tools if you are comfortable with them. If you are going to use VMs/Docker containers on your local laptop/server- make sure it is powerful enough to run 3-4 VMs simultaneously

Courses such as CSCI E-7 and CSCI E-55 are strongly recommended. CSCI E-28 and CSCI E-90, offered previously, or equivalent are also strongly 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"

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. Most sections will be held on **Saturdays, 9AM or Sundays, 9AM**, subject to change, based on TAs availability - 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. Details - TBD.
Contact info:

- Marina Popova: map685@g.harvard.edu
- Contact info of Teaching Assistants will be posted when the course begins

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

Grading and Late Policies:

- There will be up to 3 late days allowed for each assignment, with 15% grade penalty per day. After 3 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

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

Week	Topic
Week 1 - 08/31/2017	● Introduction - what is Big Data processing? ● Evolution of applications: from classic 3-tier to massively distributed Big Data ones; ● Common types and architectural blueprints of big data processing systems; ● Common building blocks of the end-to-end processing pipelines ● Administrative details
Week 2 - 09/07/2017	Scaling Concepts and Parallel Processing ● Vertical and horizontal scaling - hardware and software based (CPU cores, hyper-threading, processes, threads) ● Shared state and shared data management ● Basics of Parallel Processing: how to make your processing parallelizable; multi-threading in Java (and maybe Python) as illustration ● MapReduce as a language-agnostic processing model ● Example implementations: Java 8 Streams, Hadoop MR
Week 3 - 09/14/2017	Distributed Data Persistence 1 ● Core concepts (partitioning, replication, consistency) ● Deep dive: Hadoop - as a distributed FS ● Hadoop data modeling, HDFS and HBase schema design ● Data storage options (Parquet, Avro, …)
Week 4 - 09/21/2017	Distributed Data Persistence 2 ● Core RDBMS concepts (ACID, TX, isolation); scaling challenges ● NoSQL storage systems - core concepts: CAP and

	PACELC; partitioning, sharding, replication, storage management ● Classification of NoSQL systems by requirements and target application classes; overview of the NoSQL landscape
Week 5 - 09/28/2017	Intro to Lambda Architecture ● Batch vs. Real-Time vs. Stream processing ● the quest for answers to Ad-Hoc queries over unlimited historical and real-time data Collection Tier and Data Ingestion ○ Common patterns and concerns (push vs pull, data loss, duplicates, re-processing) ○ Example frameworks (Fluentd, Scoop, …) ○ Deep dive: Flume (sources and sinks, overflows)
Week 6 - 10/05/2017	Batch Tier: Master Datasets - core concepts ● Data modeling for Master datasets (de-normalization; example models for counting uniques and memberships) ● loading data into Master datasets: requirements, storage options (HDFS, S3), processing options ● Illustration with HDFS
Week 7 - 10/12/2017	Batch Tier: Processing and Batch Views ● Data Modeling for Batch Views - next iteration of uniques and membership counting models; ● Batch views storage options - Cassandra, Postgres, AWS Redshift/ RDS ● Batch processing requirements (fault-tolerance, scalability, no data loss, others) ● Batch processing algorithms and techniques (MapReduce again, Spark micro-batching)
Week 8 - 10/19/2017	Messaging Tier ● Data buffering and back-pressure handling ● traditional message system (like MessageQ, Tibco) vs Big Data - centric ones (Kafka) ● Kafka - deep dive (replication, partitioning, data consistency guarantees, operations) ● Kafka Connectors
Week 9 - 10/26/2017	Stream Processing and Real-Time (RT) Views ● Stream vs static data processing - core concepts, requirements, concerns ● Modeling and storage options for RT Views (Cassandra,

	RDS, ElasticSearch - as a special case) ● Algorithms for RT processing (HyperLogLog, aggregations)
Week 10 - 11/02/2017	Streaming frameworks ● Overview of the landscape (Spark, Kafka Streams, Storm, Samza, Flink) ● Kafka Streams - deep dive ● Spark Streaming - deep dive
Week 11 - 11/09/2017	Time Series as a specialized type of Big Data processing ● Modeling for Time Series ● Cassandra - deep dive
Week 12 - 11/16/2017	Data Access Tier, Distributed Indexing and Search ● Distributed Indexing - core concepts ● Anatomy of queries (filters, facets, aggregations, …) ● Illustration with ElasticSearch: cluster anatomy, index sharding, routing, percolation ● Data visualization - ES Kibana, DIY - D3, HighCharts/HighMaps
Thanksgiving Break - 11/23/2017	
Week 13 - 11/30/2017	DevOps in the Distributed World ● Containerization with Docker ● Continuous Delivery systems ● Autoscaling and monitoring
Week 14 - 12/07/2017	Putting it All Together ● Review Architectural Blueprints for Big Data processing systems again - with the gained insight: Lambda, Kappa, Hybrids ● Best Industry practices and example architectural solutions (LinkedIn, Facebook, others) ● Where to go from here
Week 15 - 12/14/2017	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