DS-593: Data Engineering at Scale

General Information

Course

- All course communication should take place on Piazza: https://piazza.com/bu/spring2024/ds593
- Class Location: CGS 523
- Class Time: 11-12:15pm
- Course Dates: Spring 2024
- Course Credits: 4

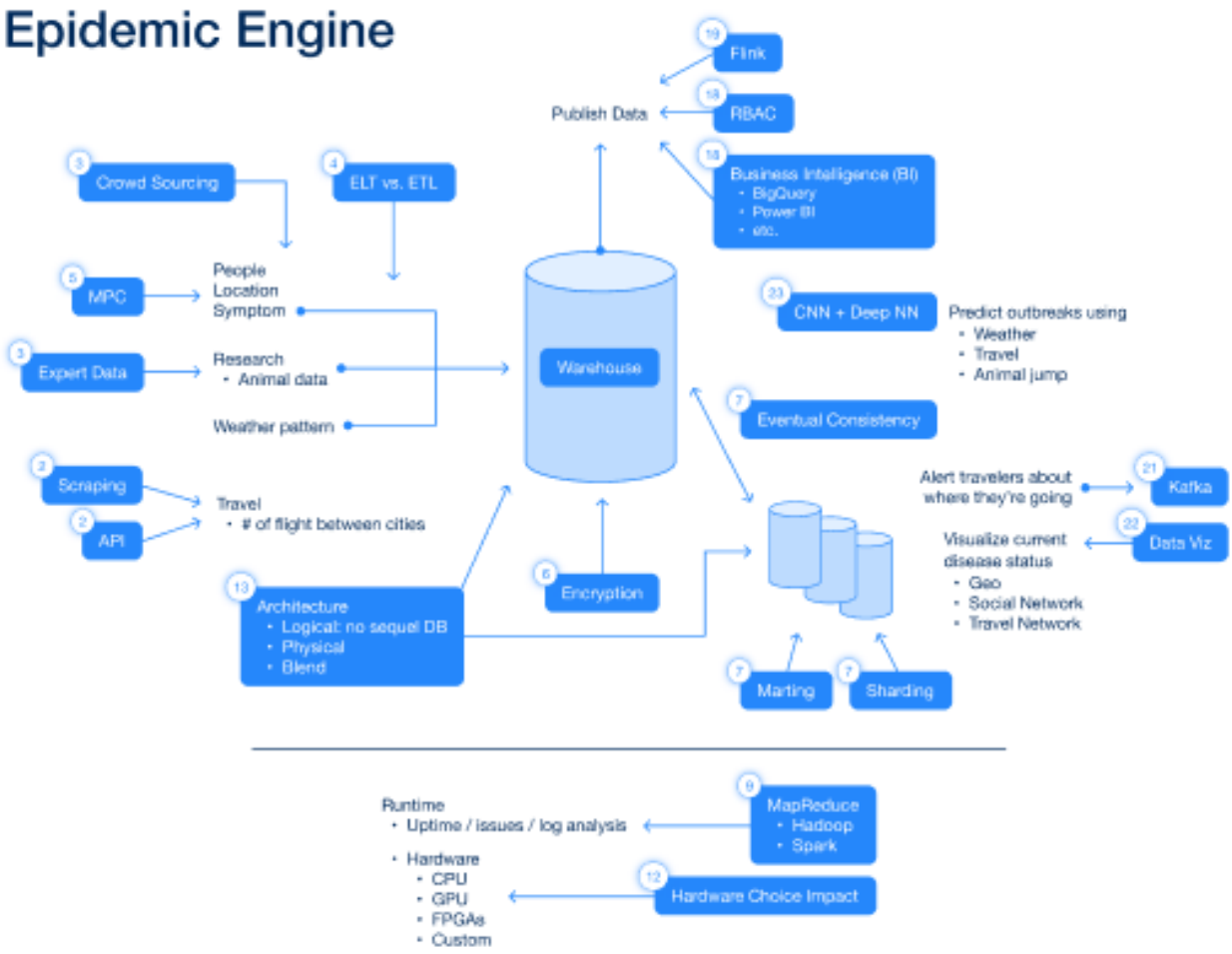
Instructors

Name	Role	Office
Langdon White	Prof	CCDS 1306
Bargav Jagatha	TA	TBD

Course Description

Summary

Welcome to "Data Engineering at Scale," a course designed to immerse you into the fascinating world of large-scale data management, processing, and analytics. Throughout this course, we will focus on a mythical but powerful application called the "Epidemic Engine". This application gathers information about potential health events, aggregates this data, publishes it in diverse ways, and ultimately attempts to predict epidemics. The Epidemic Engine, while hypothetical, embodies the principles and challenges of real-world data engineering systems that power today's most innovative technologies, from social networks to streaming platforms to cutting-edge AI research.



In our journey, we will explore the building blocks of the Epidemic Engine. This will include understanding various data collection methods, using MapReduce systems like Hadoop and Spark for data processing, managing large datasets with NoSQL and RDBMS, and navigating the challenges of uptime and log analysis. We will also dive deep into the intriguing world of data publishing and visualization, using tools like Flink, Kafka, and PowerBI. The course will provide you with a hands-on understanding of different data types, including geospatial data, document-based data, and graph-based data, all essential parts of the Epidemic Engine's architecture. Throughout the course we will be considering how to best support the use cases for our data be they BI, Alerting or doing predictions with Neural Networks.

We will explore how these techniques can be applied to the vast and complex data processed by our Epidemic Engine, offering us a glimpse into future epi-

demics. This course isn't just about learning data engineering – it's about applying it to real-world, scalable applications that can make a difference. So, get ready to embark on a thrilling journey into the heart of large-scale data engineering, unraveling the potential to harness data in ways that could revolutionize our world!

Prerequisites:

- DS310 required or equivalent.
- Strong Python skills
- Exposure to SQL & DDL
- A GitHub Account

Learning Outcomes

After completing this course students will:

- Understand the complexity of the data ecosystem for modern, enterprise-scale, applications
- Be familiar with the concepts involved in resolving the complexity and the existing tools that implement them
- Understand how to differentiate between understanding a concept and the tool that implements it
- Be able to adapt to a changing technical landscape by being able to read and apply software documentation, not just learn from classes or prepared materials

Instructional Format, Course Pedagogy, and Approach to Learning

The course is a fast paced review of many of the major components of modern data engineering. We are using the fictitious Epidemic Engine as a real-world example of the tools in use to give context to all of the individual lectures, assignments and projects.

Typically, each lecture period will have an opening, in class assignment, the assignment will be based on the reading(s) that were due for that lecture. Once the assignment is complete, a lecture will follow focused on the most important and complex aspects of the subject of the lecture. Lecture slides will be made available on Piazza shortly after each lecture.

The out of class assignments will generally connect the in class assignments together and integrate them into the Epidemic Engine. The projects will be larger efforts to provide a feature to the Epidemic Engine and may include all

or some of the other assignments. In other words, none of the components you develop in the class (with very few exceptions) won't, ultimately, be a part of the Epidemic Engine.

Discussions

Students are expected to attend their weekly discussion section. The Teaching Assistant/Fellow (TA) will lead the discussion sessions. During discussion, the TA will present slides covering the most salient topics from the week. Most of the discussion will be focused on questions from students about the lecture topics or any assignments. Questions asked during discussion can have a big, positive impact on your participation grade. The TA will post slides or other information to Piazza as necessary. In addition to the discussions, the TA will hold weekly Office Hours.

Classroom recordings

Class sessions will be recorded on a best effort basis, primarily, for the benefit of registered students who are unable to attend live sessions (either in person or remotely). The recordings will also be made available as soon as possible to students for review. Students will not appear in the recordings. However, if students do, accidentally, appear, the recording will only be available upon request and may not be distributed.

Books and Other Course Materials

For this course there will be one required book and then a number of online resources. The book, *Designing Data-Intensive Applications*, by Martin Kleppmann is available digitally or hard copy and does cost money (~$40 at the time of writing). However, all the other resources should be free of charge.

See the Course Schedule (tentative) for reading schedule and where to find them. The expectation is that the reading has been completed **before** the lecture for which it is marked as "due."

Courseware

- We will use **Piazza** for announcements, questions, discussions, and all other communication. Please do not email any of the instructional team directly as it limits our ability to respond quickly.
- We will use **Blackboard** for assignments, grades, your current standing in the class, and attendance
- We will be using GitHub for most of the assignments. Please ensure you have an account before the class starts.

Assignments and Grading

Assignments serve 2 purposes:

- Cement material learned in class
- Provide partial steps to projects

Assignments are due as indicated and may be submitted up to 24 hours late with a 5% late penalty. No late submissions will be accepted after 24 hours. Assignments must be submitted as per the assignment instructions. All in class assignments are expected to be completed within one week of release (in the event of excused absence).

Grading: Final Grade

Your final grade will be a weighted sum of grades received in the following categories:

% of Grade	Category	Notes
35%	Assignments	Grading rubrics are available on individual assignment pages. In class assignments are marked only for completion unless done outside of class.
10%	Attendance and Participation	N/A
45%	Projects	The course has 3 projects, they contribute to your projects grade as follows: Project 1: 25%; Project 2: 35%; Project 3: 40%.
10%	Peer evaluations	For much of the work in this class you will be collaborating with one or two other students. Their evaluations of your work, your contributions, and your teaming will contribute to your grade.

Resources/Support/How to Succeed in This Course:

1. Office hours are available weekly throughout the semester. You may find the hours and location on the Piazza Resources Tab | Section Staff.
2. Ensure you have read and understand the readings. You may need to read them more than once. Discuss the readings with your classmates.
3. In order to show you have read this document, please post a picture of your pet, plant or something else that is in your school or permanent home privately to Piazza.

Community of Learning: Class and University Policies

Course members' are responsible for ensuring a positive learning environment (e.g., participation/ discussion guidelines). Any violation of this tenet will result in severe penalties.

Attendance & Absences Students are expected to attend each class session unless they have a valid reason for being absent. Acceptable excused absences include observing religious holidays and illness. In such cases, students are advised to contact the instructor as soon as possible, so that reasonable accommodations can be provided. Please review the BU attendance policy and the BU Policy on Religious Observance for more information.

Academic Conduct Statement Software & data engineering are an inherently collaborative endeavor. In most cases, you will find open source projects or code snippets on the internet that you might want to use in your own projects. While this is permitted, you must cite your sources appropriately. You are also responsible for ensuring that you have the original author's permission to use their work. Remember, if there is no license, you do **not** have the right to use the content. The Open Source Initiative maintains an excellent page on the different types of software licenses and what you can and cannot do with them. Remember, source code with no mentioned license is, by default, not available for reuse (in the US).

Using code you have borrowed from the internet without permission and/or attribution is an instance of plagiarism, which is a violation of the Academic Code of Conduct. If you are in doubt about whether something might be construed as plagiarism, please check with course staff and in general err on the side of caution.

Collaboration on Assignments and Projects Unless explicitly stated otherwise, collaboration on assignments and projects among teammates is both allowed and encouraged.

Disability Accommodations If you are a student with a disability or believe you might have a disability that requires accommodations, please contact the Office forDisability Services (ODS) at 617-353-3658 to coordinate any reasonable accommodation requests. For more information, please see http://www.bu.edu/disability.

Course Schedule (tentative)

The expectation is that the reading has been completed **before** the next lecture. All assignments may be found in Blackboard.

Date	Topics	Notes
Jan. 18	Lecture: Introduction; API Overview	Reading, Requests Due: Jan 25, Reading, Beautiful Soup Due: Jan 25
Jan. 23	Lecture: Python Quiz; API Group Exercise	Assignment, Scrape a Website Due: Feb 06
Jan. 25	Lecture: API Group Exercise (Cont'd)	Reading, What is ETL Due: Jan 30 Reading, ORM & SQLAlchemy Due: Jan 30
Jan. 30	Lecture: Object Relational Mapping & ETL/ELT	Reading, Data Collection Due: Feb 06
Feb. 01	Lecture: MPC, Crowd vs Expert; In class MPC exercise	Reading, 1-4th Normal Forms Due: Feb 06
Feb. 06	Lecture: Relational Design & Containers	
Feb. 08	Lecture: Containers (Cont'd)	Reading, *Reliable, Scalable, and Maintainable Applications*
Feb. 13	Lecture: Distributed Systems	Reading, The Trouble with Distributed Systems
Feb. 15	Lecture: Architecture: Containers; Lecture: Architecture: Deployment; In Class Assignment: Container Development	
Feb. 20	Lecture: Architecture: Distributed Systems Solutions; In Class Assignment: Demonstrate an Error	
Feb. 22	Lecture: Event Driven Architecture; Assignment: Develop a Solution (Collect & Load)	Reading, MapReduce Due: Feb 29, Assignment, Project: Collect & Load Due: Mar 02
Feb. 27	Lecture: Architecture (overview of physical hardware choices):; Lecture: Architecture: Deep dive in to graph databases & gql/sparql; Reading: Mahoot, Beam, Tekton	
Feb. 29	Lecture: Architecture: MapReduce & log analysis; In Class Assignment: Project Work Time;	
Mar. 05	Lecture: Architecture: Data Pipelines; In Class Assignment: Create simple pipeline	
Mar. 07	Lecture: Architecture: GIS / Geospatial Data; Reading: Event Driven Architecture	Reading, Event Driven Architecture Due: Mar 21
~~Mar. 12~~	Spring Break	
~~Mar. 14~~	Spring Break	
Mar. 19	Lecture: Architecture: Managing change; In Class Assignment: Design a Change Process	
Mar. 21	Lecture: Architecture: Event Driven; Assignment: Project 2: Data Deployment	Assignment, Project 2: Data Deployment Due: Apr 05
Mar.		

Date	Topics	Notes
Jan. 18	Lecture: Introduction; API Overview	Reading, Requests Due: Jan 25, Reading, Beautiful Soup Due: Jan 25
Jan. 23	Lecture: Python Quiz; API Group Exercise	Assignment, Scrape a Website Due: Feb 06
Jan. 25	Lecture: API Group Exercise (Cont'd)	Reading, What is ETL Due: Jan 30 Reading, ORM & SQLAlchemy Due: Jan 30
Jan. 30	Lecture: Object Relational Mapping & ETL/ELT	Reading, Data Collection Due: Feb 06
Feb. 01	Lecture: MPC, Crowd vs Expert; In class MPC exercise	Reading, 1-4th Normal Forms Due: Feb 06
Feb. 06	Lecture: Relational Design & Containers	
Feb. 08	Lecture: Containers (Cont'd)	Reading, *Reliable, Scalable, and Maintainable Applications*
Feb. 13	Lecture: Distributed Systems	Reading, The Trouble with Distributed Systems
Feb. 15	Lecture: Architecture: Containers; Lecture: Architecture: Deployment; In Class Assignment: Container Development	
Feb. 20	Lecture: Architecture: Distributed Systems Solutions; In Class Assignment: Demonstrate an Error	
Feb. 22	Lecture: Event Driven Architecture; Assignment: Develop a Solution (Collect & Load)	Reading, MapReduce Due: Feb 29, Assignment, Project: Collect & Load Due: Mar 02
Feb. 27	Lecture: Architecture (overview of physical hardware choices):; Lecture: Architecture: Deep dive in to graph databases & gql/sparql; Reading: Mahoot, Beam, Tekton	
Feb. 29	Lecture: Architecture: MapReduce & log analysis; In Class Assignment: Project Work Time;	
Mar. 05	Lecture: Architecture: Data Pipelines; In Class Assignment: Create simple pipeline	
Mar. 07	Lecture: Architecture: GIS / Geospatial Data; Reading: Event Driven Architecture	Reading, Event Driven Architecture Due: Mar 21
~~Mar. 12~~	~~Spring Break~~	
~~Mar. 14~~	~~Spring Break~~	
Mar. 19	Lecture: Architecture: Managing change; In Class Assignment: Design a Change Process	
Mar. 21	Lecture: Architecture: Event Driven; Assignment: Project 2: Data Deployment	Assignment, Project 2: Data Deployment Due: Apr 05
Mar.		