Assignment 2: Design and Implementation of Data Architecture and Data Processing Pipelines

Deliverables	A report (less than 7 pages) - upload to Canvas. A working application (demo). Source code: a link to a GitHub project should be provided.
Submission Deadline	1st December 23:59
Demonstration	3rd December (During Labs)
Late Submission Penalty	10% reduction (0 < delay < 7 days) 50% reduction (delay > 7 days)
Grade Percentage	20% from the course grade (i.e., 20 points from 100 points).

Goals

- Design and implement data architectures and data processing pipelines using Apache Spark and GCP services.

Skills Required

- Ability to create, configure, and use a data architecture in the GCP
- Ability to create, deploy, and execute batch and stream data processing jobs with Apache Spark

Assignment Description

Definition of a Data Pipeline

We use the definition from https://martinfowler.com/articles/cd4ml.html#DataPipelines

“Pipeline is an overloaded term, especially in ML applications. We want to define a "data pipeline" as the process that takes input data through a series of transformation stages, producing data as output. Both the input and output data can be fetched and stored in different locations, such as a database, a stream, a file, etc. The transformation stages are usually defined in code, although some ETL tools allow you to represent them in a graphical form. They can be executed either as a batch job, or as a long-running streaming application.”

Spark program/job defines a data pipeline programmatically (code).

Assignment Constraints

- The data architecture must be based on recognized data architectures such as Microsoft Big Data Architecture, Lambda, Kappa, and Data Warehouse.
- The data architecture must be implemented with the most appropriate tools such as Spark, BigQuery, Google Cloud Storage, and Google Pub/Sub.
- Two separate data pipelines should be implemented. The pipelines can be batch processing and/or stream processing pipelines. Each pipeline will be a Spark program/job.
- The students cannot use the datasets used in the labs.
- The students need to develop their own Spark programs. The students cannot use the complete spark programs from the articles or Github as-is. For example, the students cannot simply copy and use a spark program implemented for a specific use case (e.g., Exercise #3 from https://towardsdatascience.com/six-spark-exercises-to-rule-them-all-242445b24565)
- The data pipelines should implement a sufficient number of data processing steps (see the examples given later for the level of the complexity expected). Machine learning is not the focus of this assignment. While the students can use machine learning in their pipelines, simply training models and making predictions are not sufficient. For example, there should be reasonably complex data preprocessing. The students are encouraged to focus on exciting batch and stream data processing use cases.

Goals of the Data Pipelines

A data pipeline is designed to support some end-user goals. Each data pipeline should support at least one business-related or end-user goal. The following are three goal examples (hypothetical).

Goal Example 1: Understand the customer satisfaction trends (for Dutch railway) based on tweets

Goal Example 2: *Clean* and *integrate* tweets and stock data, and then *extract* the features that can be used to build a model to understand the correlation between stock market movements of a company and sentiments in tweets.

Goal Example 3: Analyze and produce the statistics (e.g., average car model prices, filtered car model listings, state-level average prices, and manufacturer listings) for the current used cars market in the United States.

Goal Example 3 is from one of the student projects from the last year's course. It used the dataset from https://www.kaggle.com/austinreese/craigslist-carstrucks-data

Note: Two data pipelines in combination may implement a single goal.

Number of Data Datasets

The students can use any number of datasets, including only a single dataset. The students can use any publicly available datasets.

The students can also generate and use synthetic datasets. A good tool for data generation is https://www.mockaroo.com/

Examples for Data Architecture Implementations

Figure 1 shows the data architecture we used in the labs.

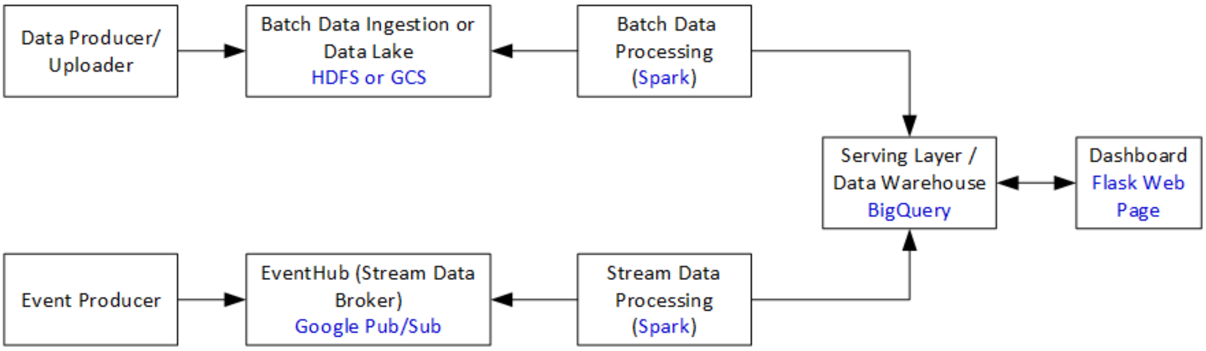
Data Producer/ Uploader
Batch Data Ingestion or Data Lake HDFS or GCS
Batch Data Processing (Spark)
Serving Layer / Data Warehouse BigQuery
Dashboard Flask Web Page
Event Producer
EventHub (Stream Data Broker) Google Pub/Sub
Stream Data Processing (Spark)

Figure 1. A Data Architecture Implementation based on Microsoft Big Data Architecture

Figure 2 shows an instantiation of the Lambda architecture style for an IoT data processing use case.   Kafka is a message broker, similar to Google Pub/Sub. The serving layer can be replaced with BigQuery. As an alternative to HDFS, GCS or local file systems can be used.

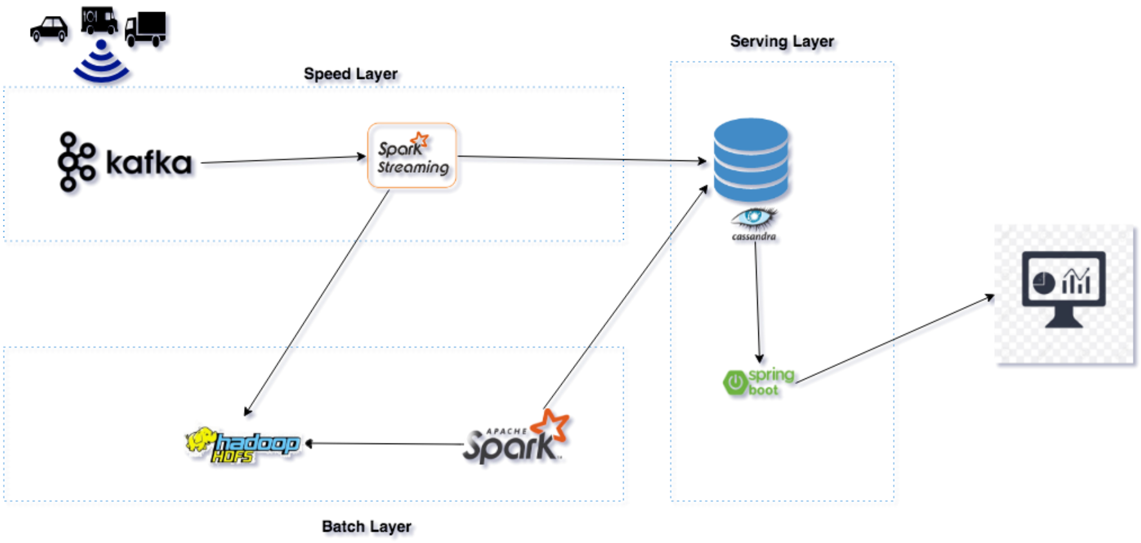

Figure 2. Lambda Architecture Used by https://github.com/apssouza22/lambda-arch

Examples for Data Pipelines

These examples can be used to assess the complexity of the pipelines you need to develop (as expected by the assignment.)

1. Data Integration and ad-hoc queries: "Exercise #3" from https://towardsdatascience.com/six-spark-exercises-to-rule-them-all-242445b24565
2. Capturing hashtag traffic for Texas Senate Elections - https://github.com/nilabja9/pyspark-twitter

Report Guidelines

Page Limit	Less than 7 pages (excluding front page, references, and appendix)
Report Content	1. Overview of the Data Pipelines What do the data pipelines do? Describe their goals/requirements 2. Design and Implementation of Data Architecture Describe data architecture and its implementation. Use diagrams as necessary. 3. Design and Implementation of Data Pipelines Describe data processing pipelines. As necessary, use diagrams, e.g., flow charts, to show the logical design of the data processing pipelines (i.e., processing steps and their flow). 4. Reflection on Design and Implementation of Data Architecture and Data Pipelines What are the possible alternative designs for your data architecture and data pipelines? First, identify at least one alternative design. Then, compare it (them) with the designs you have implemented in terms of potential strengths and weaknesses. Reflect on the process of implementing your data architecture and pipelines. What were the difficulties you faced? How did you overcome them? 5. Individual Contributions of Students Briefly describe the individual contributions of each student in the group.

Marking Scheme

The grading considers implementation, demonstration, and report.

Data Architecture (design, implementation, and demonstration)	25%
Two Data Processing Pipelines (design, implementation, and demonstration)	25*2 = 50%
Report	25%
Total	100%