# Assignment 2: Design and Implementation of Data Processing Pipelines

| | |
|---|---|
| Deliverables | A report (less than 10 pages) - upload to Canvas.<br>A working application (demo).<br>Source code: a link to a GitHub project should be provided. |
| Submission Deadline | 4th November 23:59 |
| Demonstration | 6th November (During Labs) |
| Late Submission Penalty | 10% reduction (0 < delay < 7 days)<br>50% reduction (delay > 7 days) |
| Grade Percentage | 24% from the course grade (i.e., 24 points from 100 points). |

## Goals

- Design, implement, deploy, execute, and monitor data processing pipelines using Google Dataflow (Beam)

## Skills Required

- Ability to create, configure, and use a data architecture in the GCP
- Ability to create, deploy, and execute a data processing pipeline with Dataflow (Beam)
- Ability to measure the performance of a data processing pipeline

## Assignment Description

### Goals of the Data Pipeline

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

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

**Goal Example 2:** Understand the correlation between stock market movements of a company and sentiments in tweets

There may be many ways/approaches to answer the question indicated by the goal, for example,

1. Some SQL queries over the pre-processed data stored in a data store (i.e., ad-hoc queries)
2. Using a machine learning model (i.e., prediction results)
3. ...

The students can select the most appropriate approach for answering their goals.

## 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.

## Design and Implementation of the Data Pipeline

The design of a data pipeline includes two aspects: data and platform architecture, and data processing workflows or pipelines.

The architecture needs to have the components for the phases/layers of *Ingest*, *Process and analyze*, *Store*, and *Explore and visualize* [1]. In terms of Lambda architecture, they are *ingestion*, *processing*, *storage*, and *serving*. The key focus is on *data processing pipelines*. We attempt to use the systems and tools from the GCP (see Appendix and [1]). Don't be alarmed! Our main focus is on processing, and thus on Dataflow (Beam). A few technology choices for each phase/layer:

1. **Ingest**: Google Storage or Google Pub/Sub (Message Broker)
2. **Store**: Google Storage or BigQuery
3. **Process**: Dataflow (Beam)
4. **Visualization**: Any tool (e.g., a Notebook or a flask web page showing some visualizations). Note that if the calculation of the desired results requires a pipeline (e.g., a prediction pipeline), and Dataflow (Beam) should be used.

We can use different data architectures. The following are two examples. The students can use any data architecture, provided that it includes the above-mentioned four phases/layers.

The simplest architecture is shown in Figure 1. The simplest implementation is to use Google Storage and Dataflow (as in Lab 7).

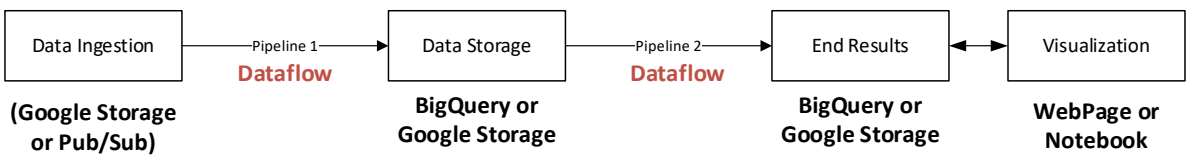



**Figure 1. The data architecture variant 1**

In the above figure, Pipeline 1 can be typical ETL batch or streaming (or both) pipelines, focusing on data preprocessing, and even training an ML model. Pipeline 1 is a typical serving pipeline, which answers to an end-user query (i.e., the goal of the data pipeline). It may do ad-hoc data processing (ah-hoc queries) or predicting an outcome using an ML model

An interesting architecture variant is in Figure 2.

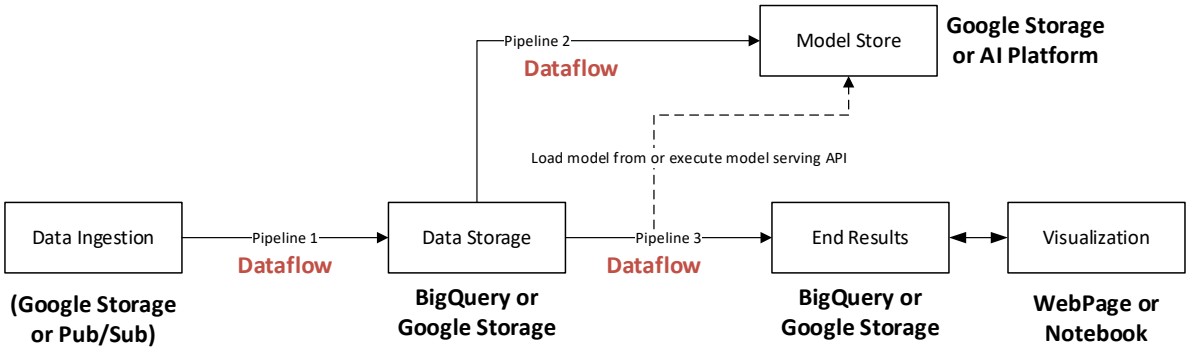



**Figure 2. The data architecture variant 2**

In Figure2, Pipeline 1 is a typical ETL pipeline focusing on cleaning and preparing the data for training ML models. BigQuery is a good choice as the data storage (as a data warehouse). Pipeline 2 is a training pipeline, and Pipeline 3 is a prediction pipeline.

The data processing pipelines should be presented as a flowchart or workflow. The students can use any flowchart or workflow notation - see the visualization of data pipelines in Google Dataflow. In general, a pipeline is first designed, and then implemented. A nice modeling/design of a pipeline using a flowchart can be found at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4921999/ (see Figure 2).

## Evaluation of the Data Pipeline

To measure the performance of a data pipeline, the students can simply use the Google Dataflow monitoring interface. The students can report information such as task execution time and resource usage. However, the students must report the performance data for three different machine types. See machine_type parameter at https://cloud.google.com/dataflow/docs/guides/specifying-exec-params I.e., execute the same dataflow pipeline with three different machine types (by simply changing machine_type parameter).

See https://cloud.google.com/dataflow/docs/guides/using-monitoring-intf

## Reuse of Lab Source Codes

The students can reuse the code from the labs.

## Some Good Examples

The students can find a good example for a data pipeline at

Game of Thrones Twitter Sentiment with Google Cloud Platform and Keras

https://towardsdatascience.com/game-of-thrones-twitter-sentiment-with-keras-apache-beam-bigquery-and-pubsub-382a770f6583

Note that the students cannot use this pipeline. Moreover, the design of the data architecture in your implementation can be simpler than the one used in the above article.

## Report Guidelines

| | |
|---|---|
| Page Limit | Less than 10 pages (excluding front page, references, and appendix). |
| Report Content | 1. Overview of the Data Pipeline<br><br>What does the data pipeline do? Describe its goals/requirements<br><br>2. Design and Implementation of Data Pipeline<br>Describe both data architectures and data processing workflows in detail. Use diagrams.<br><br>3. Evaluation of Data Pipeline<br><br>Report and discuss the performance of the data pipeline. Use Google Dataflow monitoring data. Experiment with at least 3 different machine types.<br><br>Use plots/graphs.<br><br>4. Individual Contributions of Students<br><br>Briefly describe the individual contributions of each student in the group. |

## Marking Scheme

The grading considers implementation, demonstration, and report.

| | |
|---|---|
| Data architecture | 20% |

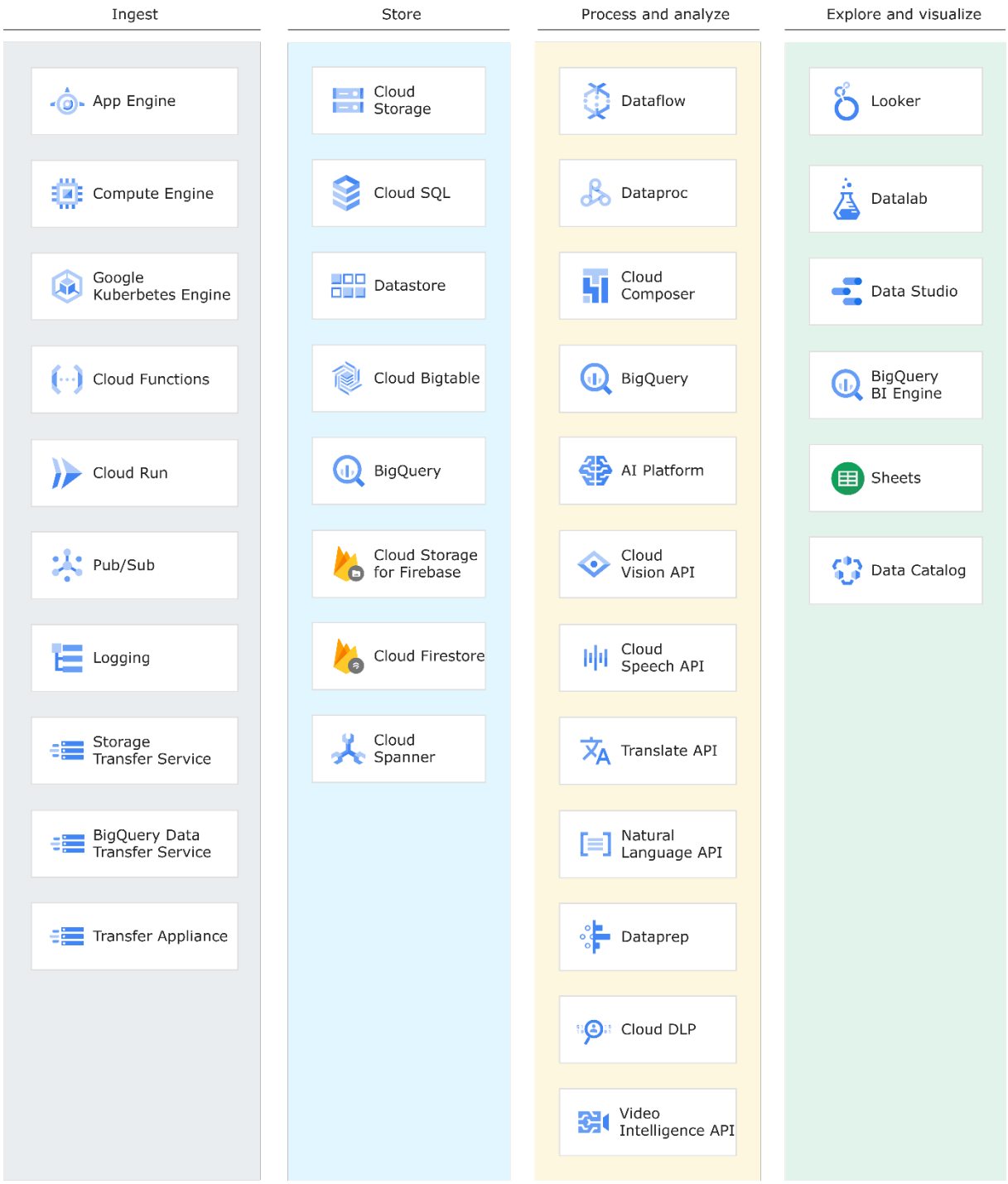
| | |
|---|---|
| Data processing pipelines | 60% |
| Performance evaluation of data pipelines | 20% |
| Total | 100% |

## References

[1] https://cloud.google.com/solutions/data-lifecycle-cloud-platform

## Appendix

https://cloud.google.com/solutions/data-lifecycle-cloud-platform


Ingest
App Engine
Compute Engine
Google Kuberbetes Engine
Cloud Functions
Cloud Run
Pub/Sub
Logging
Storage Transfer Service
BigQuery Data Transfer Service
Transfer Appliance
Store
Cloud Storage
Cloud SQL
Datastore
Cloud Bigtable
BigQuery
Cloud Storage for Firebase
Cloud Firestore
Cloud Spanner
Process and analyze
Dataflow
Dataproc
Cloud Composer
BigQuery
AI Platform
Cloud Vision API
Cloud Speech API
Translate API
Natural Language API
Dataprep
Cloud DLP
Video Intelligence API
Explore and visualize
Looker
Datalab
Data Studio
BigQuery BI Engine
Sheets
Data Catalog