Third assignment - Understanding Big Data Processing

Last modified: 06.11.2019

By Linh Truong (linh.truong@aalto.fi)

1 Introduction

The goal of this assignment is to design and develop stream processing services for big data platforms.

2 Constraints and inputs for the assignment

In this assignment, we assume that you develop and operate a big data platform **mysimbdp**, of which **mysimbpd-coredms** is the component for big data store (databases and storage). You might also have **mysimbdp-daas**, which provides REST API for **mysimbdp-coredms**, already implemented. Overall, you have a set of APIs for your customers to store data into **mysimbdp** and the data will be stored in **mysimbdp-coredms** as the final sink.

> Note: **mysimbdp** is the name of your big data platform. It has many components.

Furthermore, we assume that you have different customers who need to ingest their data into **mysimbdp**. Each customer has its own data (different syntaxes and/or semantics) but we assume that all your customers use the same database/storage model, e.g., document-based, column family -based or file -based models. Your **mysimbdp** supports multi-tenancy, thus you can have one deployment of **mysimbdp** for many customers.

With the above assumptions, you will support your customers to send data through platform message brokers (messaging systems) for performing near-real time streaming analytics. The results from near real time analytics will be (1) disseminated back to the customers in near-realtime and (2), *optionally in this assignment*, stored into **mysimbdp-coredms**, using the APIs and components you offer in your platform. Some components will be designed and implemented in this assignment.

> **Important note**: in this view, you can reuse your previous implementation of **mysimbdp-coredms**, **mysimbdp-daas**, brokers, dataset, etc. However, we consider them as external, reusable components. To simplify the test and development, you can have their code within this assignment but you need to arrange their code in clear, separate modules, as they

> cannot be taken into account for grading. This assignment, however, does not reply on the previous assignments. You can select suitable technologies for your **mysimbdp**.

We will continue with the set of technologies and programming languages like in the previous assignments. You can continue to use previous datasets for testing:

> Note: for streaming data processing, your selected data might not have timestamps associated with events. If needed, you can add timestamps into events for your tests by modifying the data.

> You might also look for other datasets which can help to run some interesting analytics, e.g., inappropriate comments in Twitter messages/product reviews

In this assignment, all customers of your platform will send their data into **mysimbdp** for streaming data analytics and they will receive the analytics results back via corresponding message brokers/messaging systems. You can use one of the following frameworks/technologies for message brokers/messaging systems in **mysimbdp**:

- AMQP with RabbitMQ
- MQTT with Mosquitto
- Apache Kafka

It is important to note that customer data sources and platforms are distributed in different administrative domains.

3 Requirements and delivery

Your goal is to provide a simple streaming data analytics service in your platform that enables some streaming analytics applications for customers.

> Important note: **implementation** should not be interpreted that the implementation starts from scratch. Reuse is important. But you should check the requirements carefully, because often you cannot reuse existing tools by just **configuring** them. Usually you need to implement extras to work with existing tools.

Part 1 - Design for streaming analytics (weighted factor for grades = 4)

1. Select a dataset suitable for streaming analytics for a customer as a running example (thus the basic unit of the data should be a discrete record/event data). Explain the dataset and at least two different analytics for the customer: (i) a streaming analytics which analyzes streaming data from the customer (**customerstreamapp**) and (ii) a batch analytics which analyzes historical results outputted by the streaming analytics. The explanation should be at

a high level to allow us to understand the data and possible analytics so that, later on, you can implement and use them in answering other questions. (1 point)

2. Customers will send data through message brokers/messaging systems which become data stream sources. Discuss and explain the following aspects for the streaming analytics: (i) should the analytics handle keyed or non-keyed data streams for the customer data, and (ii) which types of delivery guarantees should be suitable. (1 point)

3. Given streaming data from the customer (selected before). Explain the following issues: (i) which types of **time** should be associated with stream sources for the analytics and be considered in stream processing (if the data sources have no timestamps associated with events, then what would be your solution), and (ii) which types of **windows** should be developed for the analytics (if no window, then why). Explain these aspects and give examples. (1 point)

4. Explain which performance metrics would be important for the streaming analytics for your customer cases. (1 point)

5. Provide a design of your architecture for the streaming analytics service in which you clarify: **customer** data sources, **mysimbdp** message brokers, **mysimbdp** streaming computing service, **customer** streaming analytics app, **mysimbdp-coredms**, and other components, if needed. Explain your choices of technologies for implementing your design and reusability of existing assignment works. Note that the result from **customerstreamapp** will be sent back to the customer in near real-time. (1 point)

> Note that questions 1-4 are very much based on the selected datasets and customers. Thus you must give concrete examples based on the data and customer.

Part 2 - Implementation of streaming analytics (weighted factor for grades = 4)

> Note: implementation part we expect to see real code, real numbers, real logs, real flows, etc. in your implementation. So try to answer and give examples with concrete results from your implementation.

1. Explain the implemented structures of the input streaming data and the output result, and the data serialization/deserialization, for the streaming analytics application (**customerstreamapp**) for customers. (1 point)

2. Explain the key logic of functions for processing events/records in **customerstreamapp** in your implementation. (1 point)

3. Run **customerstreamapp** and show the operation of the **customerstreamapp** with your test environments. Explain the test environments. Discuss the analytics and its performance

observations. (1 point)

4. Present your tests and explain them for the situation in which wrong data is sent from or is within data sources. Report how your implementation deals with that (e.g., exceptions, failures, and decreasing performance). You should test with different error rates. (1 point)

5. Explain parallelism settings in your implementation and test with different (higher) degrees of parallelism. Report the performance and issues you have observed in your testing environments. (1 point).

Part 3 - Connection (weighted factor for grades = 2)

> Notes: no software implementation is required for this part

1. If you would like the analytics results to be stored also into **mysimbdp-coredms** as the final sink, how would you modify the design and implement this (better to use a figure to explain your design). (1 point)

2. Given the output of streaming analytics stored in **mysimbdp-coredms** for a long time. Explain a batch analytics (see also Part 1, question 1) that could be used to analyze such historical data. How would you implement it? (1 point)

3. Assume that the streaming analytics detects a critical condition (e.g., a very high rate of alerts) that should trigger the execution of a batch analytics to analyze historical data. How would you extend your architecture in Part 1 to support this (use a figure to explain your work)?. (1 point)

4. If you want to scale your streaming analytics service for many customers and data, which components would you focus and which techniques you want to use? (1 point)

5. Is it possible to achieve end-to-end exactly once delivery in your current implementation? If yes, explain why. If not, what could be conditions and changes to make it happen? If it is impossible to have end-to-end exactly once delivery in your view, explain why. (1 point)

4 Other notes

Remember that we need to **reproduce** your work. Thus:

- Remember to include the (adapted) deployment scripts/code you used for your installation/deployment
- Explain steps that one can follow in doing the deployment (e.g. using which version of which databases)
- Include logs to show successful or failed tests/deployments
- Include git logs to show that you have incrementally solved questions in the assignment

- etc.

Appendix

We suggest you to follow one of the implementation paths mentioned in the lecture to work on your assignment. The following information is extracted from the first assignment.

Technologies for data store:

- [MongoDB] (https://www.mongodb.com/)
- [ElasticSearch] (https://www.elastic.co/)
- [Hadoop File System] (https://hadoop.apache.org/)
- [Cassandra] (http://cassandra.apache.org/)

Datasets as input data

- The list of datasets: (https://version.aalto.fi/gitlab/bigdataplatforms/cs-e4640-2019/tree/master/data)

  Programming languages:

- Python

- Scala
- JavaScript/NodeJS
- Java
- GoLang
  > Note that when doing stream processing, not all major languages are supported.