First Assignment - Building Your Big Data Platforms

Last modified: 24.09.2019 By Linh Truong(linh.truong@aalto.fi)

1 Introduction

The goal of this assignment is to help students to understand system design and provisioning for big data platforms.

This is the first assignment in which our assumption is that **you** (the student doing this assignment) design a simple big data platform. The big data platform to be designed will have a set of minimum features built from some key components. We assume that you do not have depth knowledge about some technologies to be used in the design of your big data platform but this will not prevent you to design and run a simple big data platform.

2 Constraints and inputs for the assignment

The simple big data platform to be designed, called **mysimbdp**, will have the following key components:

- a key component to store and manage data called **mysimbdp-coredms**. This component is a platform-as-a-service.
- a key component, called **mysimbdp-daas**, of which APIs can be called by external data producers/consumers to store/read data into/from **mysimbdp-coredms**. This component is a platform-as-a-service.
- a key component, called **mysimbdp-dataingest**, to read data from data sources (files/external databases) of the tenant/user and then store the data by calling APIs of **mysimbdp-coredms**.

In this assignment, students will be asked to select **one** of the following technologies for **mysimpbdp-coredms**:

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

you must select **one** of the following datasets as input data

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

and you can only use the following programming languages:

- Python
- Scala
- JavaScript/NodeJS
- Java

You might also need to use shell scripts together with shell scripts to design and develop **mysimbdp**.

3 Requirements and delivery

The deliverable of this assignment includes three parts

Part 1 - Design (weighted factor for grades = 2)

Address the following points:

1. Design and explain interactions between main components in your architecture of **mysimbdp** (1 point)
2. Explain how many nodes are needed in the deployment of **mysimbdp-coredms** so that this component can work property (theoretically based on the selected technology) (1 point)
3. Will you use VMs or containers for **mysimbdp** and explain the reasons for each component (1 point)
4. Explain how would you scale **mysimbdp** to allow a lot of users using **mysimbdp-dataingest** to push data into **mysimbdp** (1 point)
5. Explain your choice of industrial cloud infrastructure and/or **mysimbdp-coredms** provider, when you do not have enough infrastructural resources for provisioning **mysimbdp** (1 point)

Part 2 - Development and deployment (weighted factor for grades = 2)

Address the following points:

1. Design and explain the data schema/structure for **mysimbdp-coredms** (1 point)
2. Explain how would you partition the data in **mysimbdp-coredms** into different shards/partitions (1 point)
3. Write a **mysimbdp-dataingest** that takes data from your selected sources and stores the data into **mysimbdp-coredms** (1 point)

4. Given your deployment environment, show the uploading performance (response time and failure) of the tests for 1,5, 10, .., **n** of concurrent **mysimbdp-dataingest** pushing data into **mysimbdp-coredms** (1 point)
5. Observing the performance and failure problems when you push a lot of data into **mysimbdp-coredms** (you do not need to worry about duplicated data in **mysimbdp**), propose the change of your deployment to avoid such problems (or explain why you do not have any problem with your deployment) (1 point)

Part 3 Extension with discovery and (weighted factor for grades = 1)

Address the following points:

1. Assume that each of your tenants/users will need a dedicated **mysimbdp-coredms**. Design the data schema of service information for **mysimbdp-coredms** that can be published into an existing registry (like ZooKeeper, consul or etcd) so that you can find information about which **mysimbdp-coredms** for which tenants/users. (1 point)
2. Assume that the service information about **mysimbdp-coredms** for a tenant/users is in a file, write a program that can be used to publish the service information of **mysimbdp-coredms** into either etcd, consul or Zookeeper (1 point)
3. Explain how you would change the implementation of **mysimbdp-dataingest** (in Part 2) to integrate a service discovery feature (no implementation is required) (1 point)
4. Explain APIs you would design for **mysimbdp-daas** so that any other developer who wants to implement **mysimbdp-dataingest** can write his/her own ingestion program to write the data into **mysimbdp-coredms** by calling **mysimbdp-daas** (1 point)
5. Assume that now only **mysimbdp-daas** can read and write data into **mysimbdp-coredms**, how would you change your **mysimbdp-dataingest** (in Part 2) to work with **mysimbdp-daas** (1 point)

> You will address the above-mentioned points by writing them into the design document (template: Assignment1-Design.MD, see the git assignment template) and provide source files. Using the template: Assignment1-Deployment.MD (see the git assignment template) for describing how to run/deploy your code, whereas code/scripts and logs will be organized into appropriate directories.

Bonus points

In this assignment, you do not have to develop **mysimbdp-daas** but if you do the implementation and test the performance of ingesting data into **mybdp-coredms** through **mysimbdp-daas**, you get **5 bonus points**.

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.