First Assignment - Building Your Big Data Platforms

Last modified: 20.01.2026 By Linh Truong(linh.truong@aalto.fi)

> **You are not allowed to share/publish this assignment description**.

> Make sure you follow the rule to avoid academic violation.

The goal of this assignment is to help students to understand the basic system design and provisioning for the core components related to data storage (databases/data services) in a big data platform.

1 Introduction

In this first assignment, our assumption is that **you** (the student doing this assignment) design a simple big data platform. You will play two main roles in this assignment:

- platform designer/provider: provides key big data services for tenants
- tenant developer/users: designs tenant data structures and performs basic data ingestion/tests.

2 Constraints and inputs for the assignment

The simple big data platform to be designed and developed in this assignment, called **mysimbdp**, will have the following key components:

- a key component to store and manage data called **mysimbdp-coredms**. Tenants can get shared or dedicated instances of **mysimbdp-coredms** for their usage.
- a key component, called **mysimbdp-dataingest**, to read data from data sources (files/external databases/messaging systems) of the tenant/customer and then store the data by calling APIs of **mysimbdp-coredms**.

> **Recall**: A platform can be used by a set of tenants. One tenant can have different users and run many data producers/consumers, whereas each producer and consumer might use a set of concurrent tasks for writing and reading data. In this assignment, we do not focus on specific requirements about data from tenants. Thus, we can also assume that the tenants of **mysimbdp** will store similar types of data using similar data models (at the data structure or semantic levels).

Students will be asked to select **one** of the following technologies for **mysimpbdp-coredms**:

- MongoDB, ElasticSearch, Cassandra, Scylla, CockroachDB, Apache Druid, Clickhouse, Apache Pinot

> *If you want to work with other databases, you can discuss with the responsible teacher.*

you must select real **datasets from** the following datasets as input data for examples/testings

- The list of datasets: https://github.com/rdsea/bigdataplatforms/blob/master/data/README.md

> *You can bring your own data through a discussion with the responsible teacher. It is up to you to decide if you want to have an advanced scenario of multiple types of databases/data storage.*

and you can only use the following programming languages:

- Python, JavaScript/TypeScript, Java, GoLang

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. Explain your choice of the application domain, the *generic* types of data to be supported, and the technologies used for **mysimbdp-coredms**. Explain your assumption about the tenant data sources and how the data from the sources can be accessed. Explain the situations/assumptions under which your platform serves for big data workloads. (1 point)
2. Design and explain the interactions among main platform components in your architecture of **mysimbdp**. Explain how the data from the sources will be ingested into the platform. Explain the third parties (services/infrastructures) that you do not develop for your platform. (1 point)
3. Explain a configuration of a set of data nodes for **mysimbdp-coredms** so that you prevent a single-point-of-failure problem for **mysimbdp-coredms** for your tenants. (1 point)
4. You decide a pre-defined level of data replication for your tenants/customers. Explain the required number of data nodes in the deployment of **mysimbdp-coredms** for your choice so that **mysimbdp-coredms** can work properly according to the choice of replication. (1 point)
5. Consider the data center hosting your platform, the locations of tenant data sources and the network between them. Explain where you would deploy **mysimbdp-dataingest** to allow your tenants using **mysimbdp-dataingest** to push data into **mysimbdp**, and which assumptions you have for the deployment. Explain the performance pros and cons of the deployment place. (1 point)

Part 2 - Implementation (weighted factor for grades = 2)

Address the following points:

1. Design, implement and explain one example of the data schema/structure for a tenant whose data will be stored into **mysimbdp-coredms**. (1 point)
2. Given the data schema/structure of the tenant (from the previous point), design a strategy for data partitioning/sharding, explain the goal of the strategy (performance, data regulation, etc.), and explain your implementation for data partitioning/sharding together with your design for replication in Part 1, Point 4, in **mysimbdp-coredms**. (1 point)
3. Assume that you play the role of the tenant, emulate the data sources with the real selected dataset and write a **mysimbdp-dataingest** that takes data from your selected sources and stores the data into **mysimbdp-coredms**. Explain the atomic data element/unit to be stored. Explain possible consistency options for writing data in your **mysimdbp-dataingest**. (1 point)
4. Given your deployment environment, measure and show the performance (e.g., response time, throughput, and failure) of the tests for 1,5, 10, .., **n** of concurrent **mysimbdp-dataingest** writing data into **mysimbdp-coredms** with different *speeds/velocities* together with *the change of the number of nodes* of **mysimbdp-coredms**. Indicate any performance differences due to the choice of consistency options. (1 point)
5. Write a data consumer by querying/retrieving data in **mysimbdp-coredms**. Observe and present the performance and failure problems when you increase the number of concurrent data producers and

consumers and their ingested data and queries. Propose the change of your deployment to avoid such problems (or explain why you do not have any problem with your deployment). (1 point)

> Note: A condition in performance testing is to change the number of nodes of **mysimbdp-coredms**. Therefore, if you use public cloud deployment of your selected data nodes/databases, you must be able to manipulate the number of nodes.

Part 3 Extension (weighted factor for grades = 1)

Address the following points:

1. Using your **mysimbdp-coredms**, a single tenant can run **mysimbdp-dataingest** to create many different databases/datasets. The tenant would like to record basic lineage of the ingested data. Explain the types of metadata about data lineage you would like to support. Provide one example of lineage data. Explain from where and how you can have such lineage data. (1 point)
2. Assume that each of your tenants/users will need a dedicated **mysimbdp-coredms**. Design the data schema of service and data discovery information for **mysimbdp-coredms** that can be published into an existing registry (like Redis, ZooKeeper, consul or etcd) so that you can find information about which **mysimbdp-coredms** is for which tenants/users. (1 point)
3. Explain how you would change the implementation of **mysimbdp-dataingest** (in Part 2) to integrate a service and data discovery feature (no implementation is required). (1 point)
4. Assume that you have to introduce a new key component, called **mysimbdp-daas**, of which APIs can be called by external data producers/consumers to store/read data into/from **mysimbdp-coredms**. Tenants can get shared or dedicated instances of **mysimbdp-daas** for their usage. Assume that only **mysimbdp-daas** can read and write data into **mysimbdp-coredms**. Explain how you would change your **mysimbdp-dataingest** (in Part 2) to work with **mysimbdp-daas**. Draw the updated architecture of your mysimbdp. (1 point)
5. Assume that the platform allows the tenant to define which types of data should be stored in a **hot** space and which in a **cold** space in the **mysimbdp-coredms**. Provide one example of constraints based on characteristics of data for data in a **hot** space vs in a **cold** space. Explain how you would support *automatically moving data* from a **hot** space to a **cold** space. Explain also possible inconsistencies that may happen when accessing hot and cold data. (1 point)

> You will address the above-mentioned points by writing your solutions into the design document (template: Assignment-1-Report.md, see the git assignment template https://version.aalto.fi/gitlab/bigdataplatforms/assignment-nr-studentid and provide source files. See the guideline of the template for organizing code and document how to run the code (Assignment-1-Deployment.md)

4 Other notes

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

- Include the (adapted) deployment scripts/code you used for your installation/deployment
- Explain steps that one can follow in doing the deployment
- Include logs to show successful or failed tests/deployments
- Include git logs to show that you have incrementally solved questions in the assignment