TU Munich, Department of Informatics
Chair III: Database Systems
Prof. Dr. Jana Giceva
Winter Term 2023/24: Cloud-Based Data Processing

Assignment 1 - High Level System Design (I)

1 Point
Handout date: 26.10.2023
Due date: **02.11.2023 at 2pm**
Discussion dates: 26.10.2023, 02.11.2023
Michalis Georgoulakis, Tobias Götz ({geom,goetzt}@in.tum.de)

Introduction

The first assignment is organized around one paper which you will find on Moodle. It introduces the analytical, cloud-native data warehouse Snowflake [Dageville et al.(2016)], an influential platform for later cloud systems.

Section 1 involves questions related to high level analysis of Snowflake, whereas the 2nd section involves distributed systems challenges, the handling of which is crucial in high level system design.

Focus only on what is written in the paper for your answers, there's no need for further research on use-cases or requirements not named in the paper. However, if you are not familiar with the workloads targeted or some of the terms used in the papers, a background research is necessary. Feel free to ask in class or in Mattermost.

Submission guidelines

Answer the questions briefly, in several sentences. In many of the questions, there are many possible answers and we find that discussion is more important than a single concrete answer. Please be ready to present your answer in class, and outline the advantages and disadvantages of your choices.

Submit your answers in **Moodle** in a PDF file. They can either be typed, hand-written, and drawings can also be included for the diagram questions.

Please include your name in the first page of the PDF.

Section 1: Snowflake - Architecture and design decisions

Snowflake - High Level System Design

(i) What are the **functional** and **non-functional** requirements of Snowflake?

(ii) For **what workloads** is Snowflake built for/best-suited?

(iii) What design decision allowed Snowflake to handle heterogeneous workloads, that vary in terms of resource demand and concurrency?

(iv) Draw a **high-level diagram** visualizing how a submitted query is handled by the system, showing the interaction between the different layers.

Section 2: Distributed Systems Questions

CAP Theorem

(i) Where do the following stand in terms of the CAP theorem, and why?

1 Snowflake

2 Relational database management systems (e.g. PostgreSQL)

Failure - Disaster Recovery

(i) How does Snowflake handle (i) a single node failure (ii) a full AZ failure?

(ii) Snowflake guarantees 99.99% availability. What is the maximum accepted down time of the system per year in minutes? Please round the result to the next full minute.

(iii) Describe a scenario when your Snowflake virtual warehouse becomes unavailable.

Section 3: Storage

(i) Name two reasons on why having dedicated storage drives in the compute nodes can be useful.

(ii) Name two limitations of the S3 Storage Service.

Section 4: Economics

(i) Cost provisioning is important when working in the Cloud.

1. What are the **main cost elements** included in Snowflake? Which is the most expensive? (hint)

2.What are the costs included in the storage layer of Snowflake? (hint)

(ii) Optional: How does separation of compute and storage pay off financially?

References

[Dageville et al.(2016)] Benoit Dageville, Thierry Cruanes, Marcin Zukowski, Vadim Antonov, Artin Avanes, Jon Bock, Jonathan Claybaugh, Daniel Engovatov, Martin Hentschel, Jiansheng Huang, Allison W. Lee, Ashish Motivala, Abdul Q. Munir, Steven Pelley, Peter Povinec, Greg Rahn, Spyridon Triantafyllis, and Philipp Unterbrunner. 2016. The Snowflake Elastic Data Warehouse. In *Proceedings of the 2016 International Conference on Management of Data* (San Francisco, California, USA) *(SIGMOD '16)*. Association for Computing Machinery, New York, NY, USA, 215–226. https://doi.org/10.1145/2882903.2903741