ECE 1779 Assignment3

Junhong Chen

Q1. In the context of data warehousing, describe briefly what Mesa would be; identify why it is needed:

Mesa is a highly scalable analytic data warehousing system designed by Google to store critical measurement data related to Google's Internet advertising business. It's designed to address the challenges of handling massive-scale data processing and analysis within Google's infrastructure. It is needed because it has the following capabilities:

Scalability: Mesa is built to handle extremely large volumes of data generated by various Google services. It can scale horizontally across numerous machines, allowing it to accommodate the ever-increasing amounts of data.

Low Latency: Mesa prioritizes low-latency query processing, enabling near real-time analysis of data. This is crucial for applications where timely insights are essential, such as monitoring system performance or analyzing user behavior.

Fault Tolerance: Being a distributed system, Mesa is designed with fault tolerance in mind. It can handle hardware failures and network disruptions gracefully, ensuring continuous operation without data loss.

Flexibility: Mesa supports both batch and stream processing, catering to different types of data analysis requirements. It can handle structured and semi-structured data efficiently, providing flexibility in data modeling and analysis.

Q2. Identify the technical requirements that must be deployed in a data warehouse core.

The data warehouse core needs to meet the following requirements: atomic updates, consistency and correctness, availability, near real-time update throughput, query performance, scalability, online data and metadata transformation.

Q3. Identify the operational infrastructure components that Mesa utilizes.

Mesa utilizes many infrastructure components and services such as Colossus, BigTable, MapReduce, a distributed synchronization protocol based on Paxos, the controller/worker framework, query servers, committer, and versions databases.

Q4. Identify the ACID properties in the context of data processing.

Atomicity: This property ensures that each transaction is treated as a single unit of work, which either succeeds completely or fails completely. In other words, if any part of the transaction fails, the entire transaction is rolled back, and the database is left unchanged. This prevents the database from being left in a partially updated or inconsistent state.

Consistency: Consistency ensures that the database remains in a valid state before and after each transaction. Transactions must adhere to all defined rules, constraints, and relationships in the database. Thus, any transaction that begins with the database in a consistent state will leave it in a consistent state, regardless of its success or failure.

Isolation: Isolation ensures that the execution of transactions is independent of each other. Transactions should operate as if they were the only ones being executed, even if multiple transactions are occurring concurrently.

Durability: Durability ensures that once a transaction is committed, its effects are permanently stored in the database and will persist even in the event of system failures such as power outages or crashes.

Q5. Explain how Mesa maintains data.

All persistent metadata is stored in BigTable and all data files are stored in Colossus. Data is stored and maintained in tables, each of which has a table schema that specifies the structure of the table. A table schema specifies the key space K and the corresponding value space V. Values with the same key can be aggregated.

Q6. Identify how updates are applied.

Mesa updates in batches. An update specifies a version number n and a set of rows of the form (table name, key, value). Each update contains at most one aggregated value for every (table name, key). Updates are applied to the database sequentially, according to their version numbers. The next update will no start until the previous update completes entirely to ensure atomicity.

Q7. What is the rôle of Deltas in Meas.

Deltas are used to pre-aggregate and store contain versioned data. It consists of a set of rows and a delta version. Deltas enable Mesa to perform cumulative updates rather than reprocessing entire datasets whenever there is a change. Deltas allow Mesa to maintain data consistency and integrity, as well as support queries based on specific versions of the data. It also facilitates efficient storage and retrieval of data by storing only the changes or differences between consecutive versions of the dataset.

Q8. Refer to figure 7 and explain the update time performance:
- How often does Mesa receive batches from the sources, has Mesa been capable of consuming the data without neither running out of resources nor occurrence of update backlog; if so, how would Mesa guarantees its commit latency.

Mesa receives update batches from the sources every 5 minutes. Mesa is capable of consuming the data without running out of resources or occurrence of update blocklog. It guarantees its commit latency by dynamically scaling resources and the batch processing approach.