SOFE 4360U- Cloud Computing

Project Milestone

Project Milestone-- Data Storage Implementation: KV + relational

Alex Dafoe 100599423

Ethan Bautista 100657250

Hamza Farhat Ali 100657374

Nick Kvrgic 100613386

Github Link: https://github.com/HamzaFarhat/CloudProject1Group8
Final Group 4 Videos link:
https://drive.google.com/drive/folders/1Ei8UvX9f8CHs5YQOHjM0oVePOshX78QZ?usp=sharing

Describe the following:

Sink and Source connectors

- A sink exports data from Kafka topics and inserts into a MySQL database
- A source imports data from a MySQL database and produces into Kafka topics

The applications/advantages of using Kafka Connectors with data storage.

- Allows applications using Kafka to talk to existing applications that don't use Kafka or don't integrate with Kafka natively. E.g. a vendor stores all their sensor data in a mysql database but you want users to be able to subscribe to that data using Kafka.
- One could use Kafka Connectors to link two different applications that normally don't "talk" to each other.

How do Kafka connectors maintain availability?

- Data is persisted in a database
- It has many partitions across many different distributed instances

List the popular Kafka converters for values and the properties/advantages of each

- Avro
    - Advantages
        - Default and well supported
        - Space efficient binary encoding
        - Uses Schemas to enforce message formats
    - Disadvantages
        - Serialized messages are not human readable
        - More difficult to implement compared to String/JSON when used for simple tasks
- Protobuf
    - Advantages
        - Space efficient binary encoding of data
        - Supported in many languages
        - Uses Schemas to enforce message formats
    - Disadvantages
        - More difficult to implement compared to String/JSON when used for simple tasks
- String
    - Advantages

        - Simple
        - Well supported in many languages and easily understood by developers
    - Disadvantages
        - Not good when multiple fields are required
- JSON
    - Advantages
        - Easy to implement
        - Supported natively by many databases
    - Disadvantages
        - High overhead (storage wise)

What's a Key-Value (KV) database?

- A Key-Value database is a nonrelational database that uses key-value pairs to store data rather than the row-column structure of tables in a traditional relational database.
- Stores data in collections which are groups of fields which are similar to tables in relational databases.

What are KV databases' advantages and disadvantages?

Advantages:

- KV databases are flexible
- Faster performance than relational databases because of the indexing
- Simple to implement

Disadvantages:

- Not structured
- Inability to do complex queries

List some popular KV databases.

- DynamoDB
- MongoDB
- Redis
- BerkeleyDB

List some possible applications that can be implemented by using the uploaded dataset.

- Application that can predict appearance changes
- Obstacle detection and tracking
- Teach-and -repeat navigation systems