SOFE 4630U: Cloud Computing
February 15, 2022
Group 11 - Group Report
Project Milestone - Data Storage Implementation: KV + relational
Fajer Zayed (100672347), Ireni Ruthirakuhan (100657302), Raveenth Maheswaran (100704540), Yale Wang (100673933)

GitHub Link: https://github.com/fzayed/Project-Milestone-Group-11.git

Sink and Source Connectors

- Sink Connector
    - Upload data into a source
    - No limit on number of places data can come from
        - Loosely coupled from the data sources
    - Data is generically taken in
        - Then, the system rewrites
        - Can be changed on the fly
        - Many formats accepted
        - Scalable
- Source Connector
    - Takes data from e.g. database
    - Database isn't affected and does not need to know what's taken
        - Loosely coupled from the database
    - Quickly takes updates and generalizes
        - Messages are format agnostic
        - Does not have limits on source

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

- Applications use Kafka Connectors in the case where you need to receive data from external systems
- Advantages of using Kafka Connectors are:
    - Flexibility
        - Decouples source and target
        - Changes in source/sink can be done without impacting the other
    - Scalability
        - Buffer for the data
        - When there is too much happening it provides a basic queuing functionality
    - Fault tolerance
        - Connection to sink/source may go down, with knowing that you are still producing data
            - Data is stored into kafka
    - Building pipelines
        - With the data in some exterior location and you want to through kafka to another location → transactional db to an object store

How do Kafka connectors maintain availability?

- Many clusters, varied environments
    - Ensures that one machine going down does not affect others
- No dependency on original source or user
- High variance on data
    - Lots of contingencies
    - Doesn't care about data changing

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

- Converters receive connect records and turns them into bytes
    - Writes it as key and value into a kafka cluster
- Serializer in a regular kafka producer

- Popular converters
    - Json Schema
        - Advantages include that it generates clear and readable documentation
    - Avro
        - Advantages:
            - It's binary format
            - Fast as it doesn't require code generation
            - Flexibility as it has wide variety of programming languages
    - ProtoBuf
        - Advantages include
            - that it is faster, simpler and smaller
            - Has RPC support
            - Structure validation allows a predefined + larger structure

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

- Stores messages as key and value pairs
    - Ability to store, retrieve and update data
- Non relational database

What are KV databases' advantages and disadvantages?

- Advantages of Key-Value databases include:
    - Scalable
        - Increase on database load/data has no negative impact on performance
        - Infinitely scalable horizontally
    - Speed/responsiveness
    - Reliability
    - Flexibility
        - DB can be easily relocated w/o change in structure
- Disadvantages of Key-Value databases include:
    - Single key value
    - No Query language
        - May not be able to import data into a different KV database
        - Not optimized for lookup
    - Values cannot be filtered

List some popular KV databases.

- Amazon DynamoDB
- Aerospike

- Redis

Video Link
https://drive.google.com/file/d/1AHV-axfHvXO0PNC1w0eL6AVXv5Hetz_Z/view?usp=sharing

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

- Ensure a robot is working
- Measure accuracy of implemented robotics algorithms
- Adjust robot movements on the fly