High-Concurrent Text Processing System

With Application of gRPC and Redis

Motivation

Cons of Kafka:

- Complicated architecture
- Unpredictable latency
- Not good for real-time processing

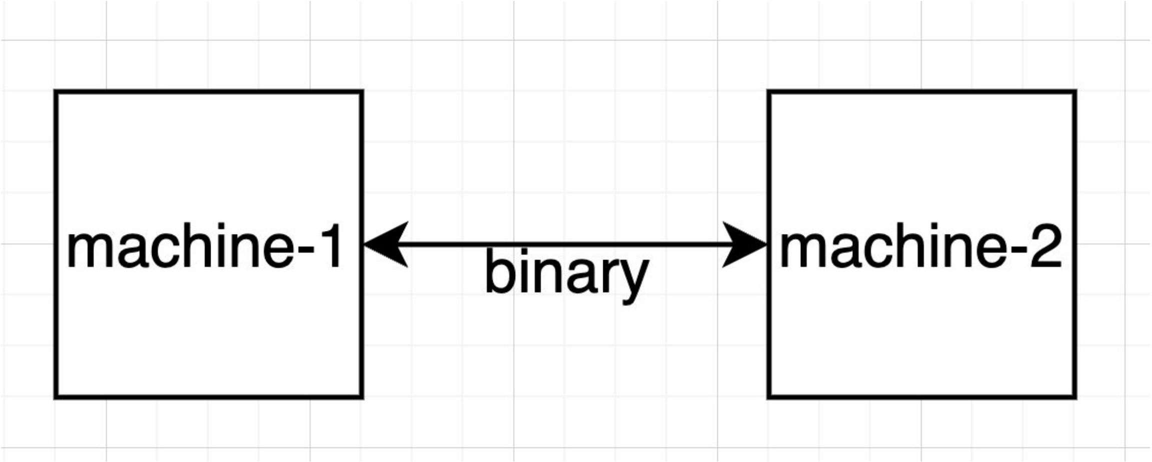
Pros of gRPC:

- Supports streaming transmission based on HTTP 2.0
- Support protobuf for smaller data size
- Supports bi-directional streaming

RPC (Remote Procedure Call)

Definition: RPC allows a program a function from another machine as if it were a local function call.

- Use gzip to compress binary data for faster and more stable text transmission

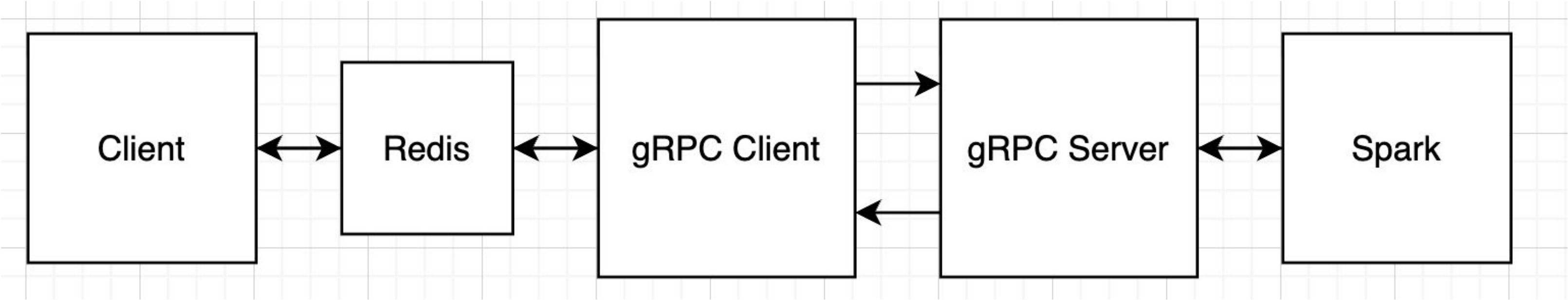


Redis

Definition: An in-memory key-value data store.

- Extremely fast to retrieve data
- Translate each line of text to md5 as key and store the result in Redis
- Check if Redis has the result for this line
- Put result from Spark to Redis

System Architecture



Future Works

- Integrate Redis Cluster for better fault-tolerance

- Support multi-task processing in parallel.

- Implement Docker + Kubernetes for easier service deployment

Thank You