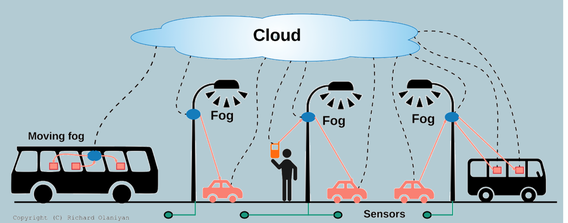
Implementation of Worker Nodes Data Interchange in JAMScript

Alvin Tan | BA Software Engineering
Prof. Muthucumaru Maheswaran | Dept. of Electrical Engineering

Background

A distributed computing topology refers to information processing/collection at different levels, namely, edge, fog and cloud. The levels are a general proximity, respectively having a further distance from the end users. JAMScript is a programming language that is oriented specifically around edge computing. The JAMScript database system used for storing the data was long due for a close inspection and redesigning.



Objectives

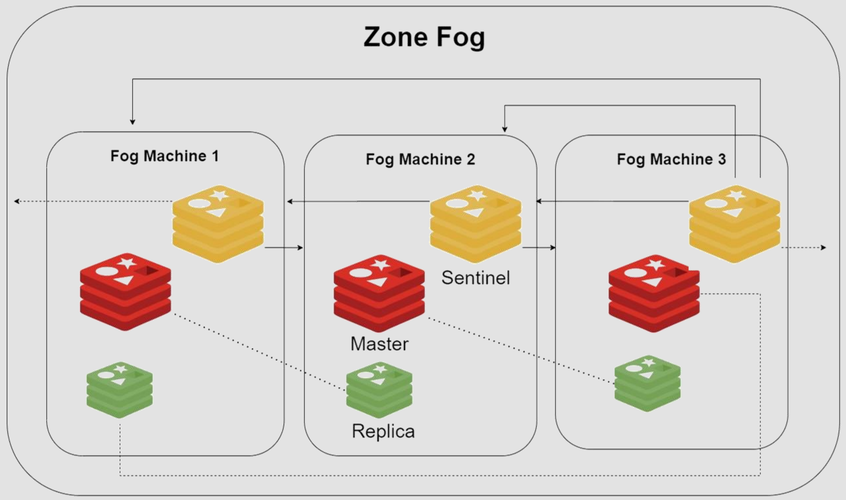
The following are the system requirements:

1. Resilient to random fault occurrences
2. Low memory usage and latency
3. High throughput and efficiency
4. Time series data compatible

System Design

To achieve fault tolerance, the following system was introduced:

- Each master data store in machine is backed up by a replica at another machine
- Sentinels monitors the local, left and right machines, replacing failed any master by its replica

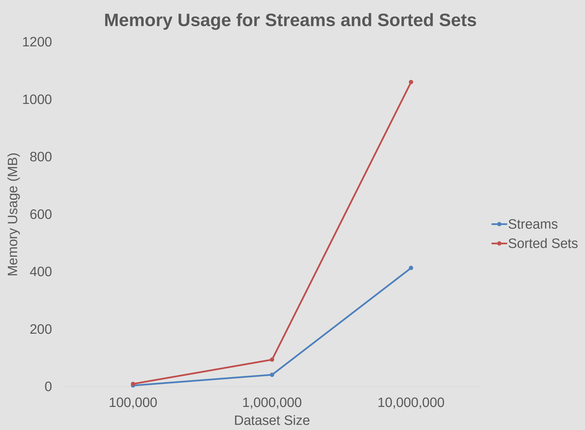


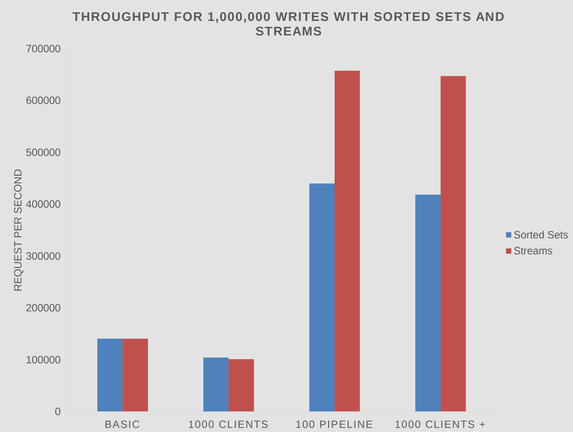
A stream data structure replaced the former sorted sets data structure due to:

- Lower memory usage
- Low read and write time

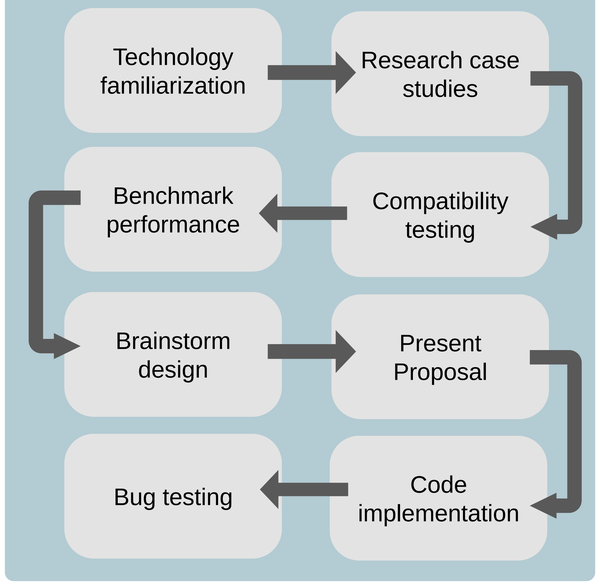
A pipeline which aggregates information flow was implemented, which provides:

- Higher efficiency
- Lower network latency





Methodology



Future Work

- Implement a buffer system for synchronously retrieve sequential data without duplicates
- Explore other database system that enables direct interaction rather than through a process
- Discover solutions regarding potential clock skew between machines
- Optimization to reduce memory usage for the edge device with limited memory

Acknowledgements

This research is supported by NSERC, Ericsson (GAIA), and Mitacs. A big thanks to the following JAMScript members: Yuxiang Ma and Ross Vladimirov.