Large Scale Distributed Systems

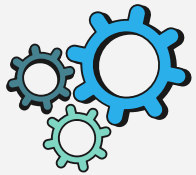
T02G04 - 2023/2024
Gustavo Costa - 202004187
João Oliveira - 202004407
Ricardo Cavalheiro - 202005103

Table of contents

01

Problem Requirements

01 Problem Requirements



Shopping Lists

- Uniqueness
- Create
- Read
- Update (concurrently)

Environment

- Local work
- Cloud Syncing

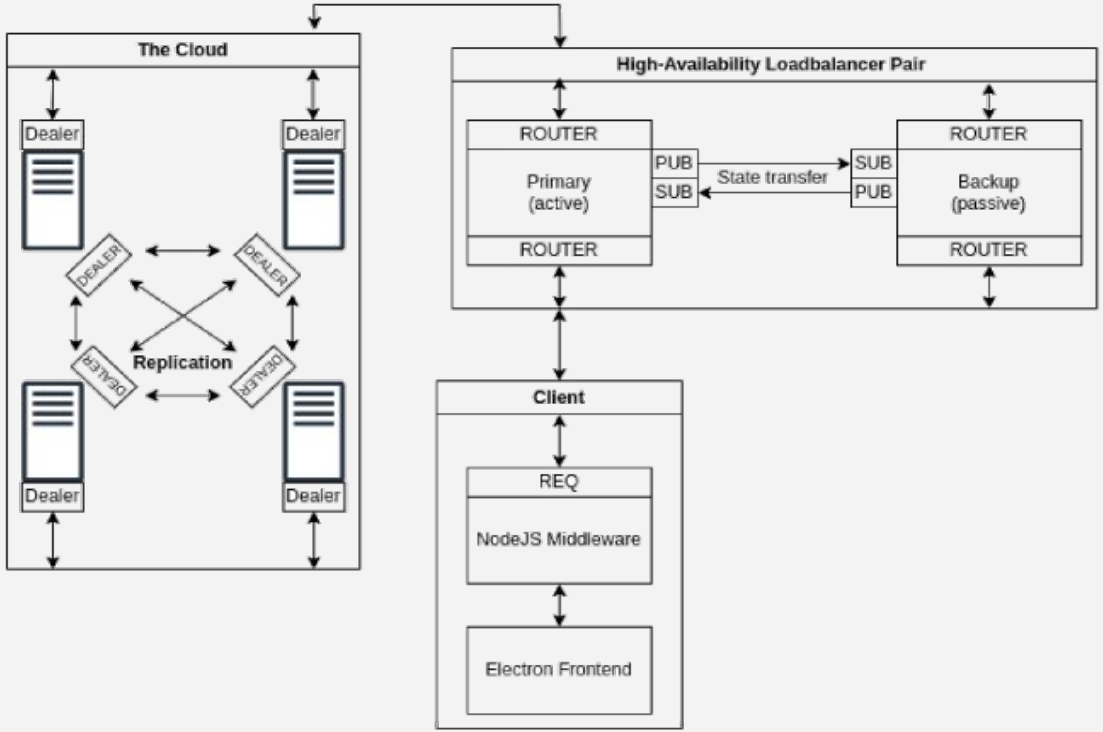
Considerations

- Scalability
- Availability

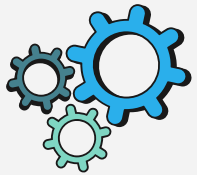
02

Technical Solution

O2 System Architecture



02 Client



Electron + React

- A desktop app made more sense.
- For it we chose Electron, a javascript framework.
- One JS framework is not enough, so we paired it with React.

NodeJS

- NodeJS application, acting as a middleware.
- Responsible for the communication between the client and the cloud.
- Comm. via TCP using a REQ socket.

02 High-availability Load balancer pair

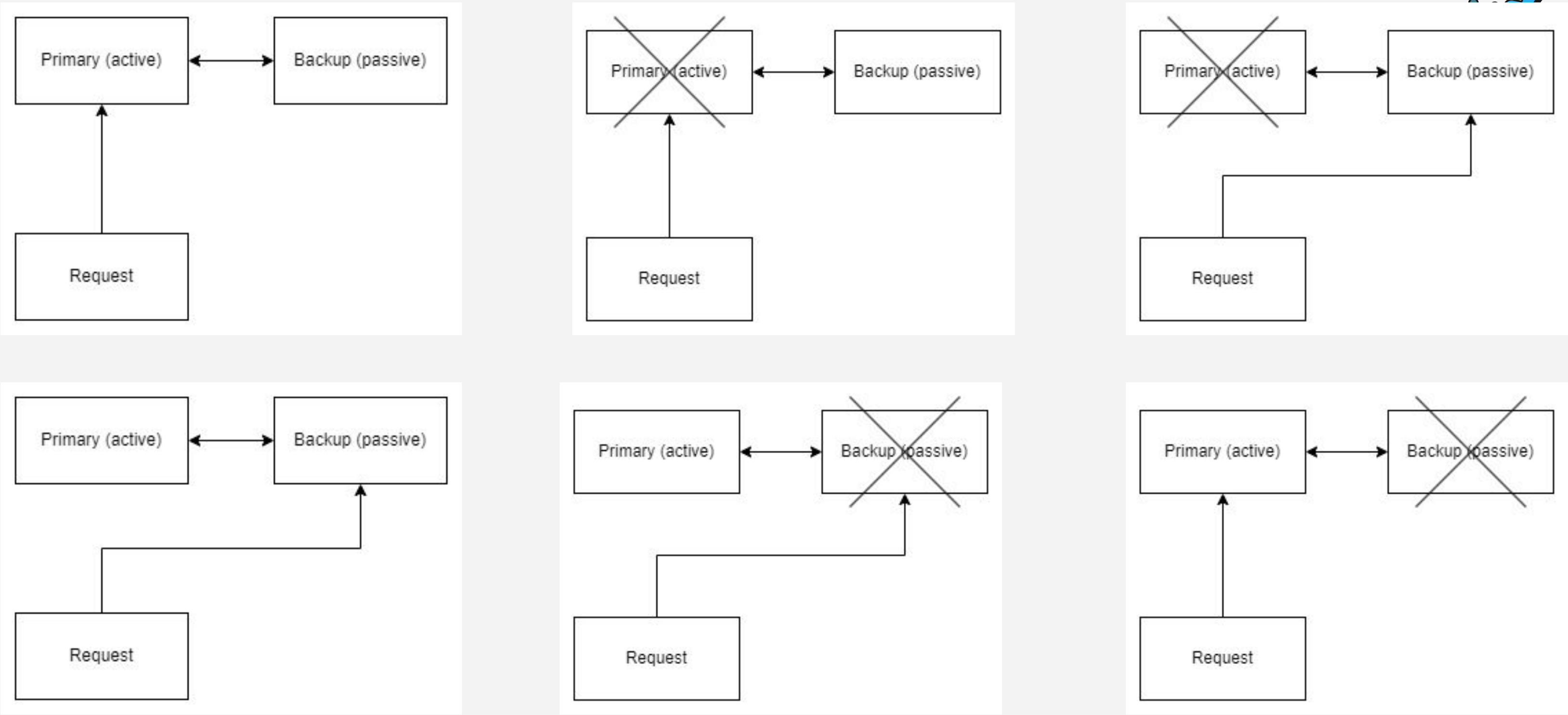
Load balancing

- Consistent Hashing
- Hash Function (SHA-256)
- Virtual Nodes
- NºReplicas = 3
- Shopping list IDs are hashed and the hash ring is queried to know the coordinator node.
- Informs the servers on each update of the ring. (Server Left/Join)

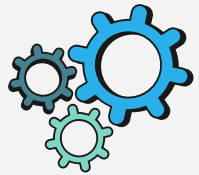
Failure detection

- Primary and backup load balancers.
- Always communicate states between each other.
- 7 step protocol.

02 7 step protocol

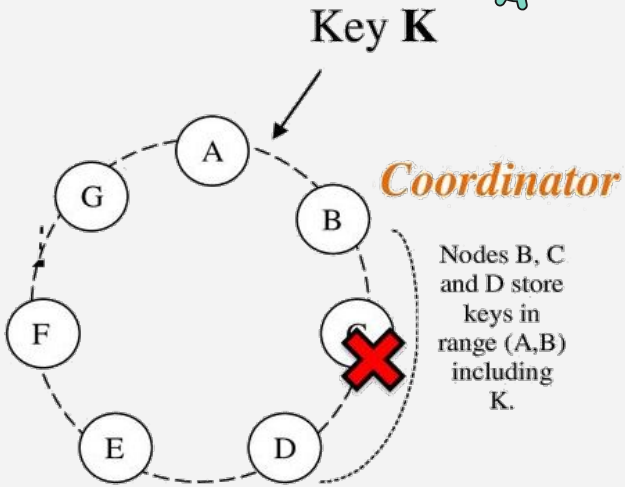


02 Cloud

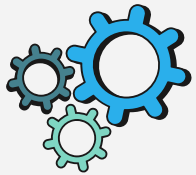


Servers

- Everytime a server joins the ring is rebalanced.
- Hinted Handoff is employed to replicate the data.
- The servers that have backup data send it to the destination upon its connection.
- Data merge is done on the writes using CRDTs
- Data is persisted as a CRDT in JSONs



02 HeartBeating & CRDT



HeartBeat

- Between load balancers so each one knows the state of the other
- Between the load balancers and the servers

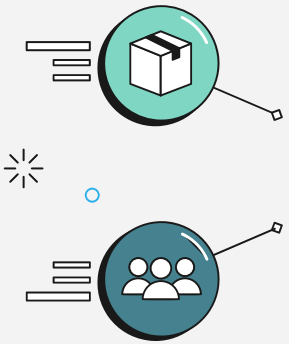
CRDT Format

- Uuid’s are used for the user and shopping lists.
- Each item of the shopping list is a key value pair, where the key is the name of the item and the value is a PNCounter CRDT

03

Solution Assessment

03 Limitations



Server Assumption

- If a server goes down, we assume it will reconnect.
- Not removed from the ring.
- There has to be one server with a given shopping list on at all times.

Brokers

- If both brokers go down at the same time, it won't be possible for the user to communicate with the cloud or for any other servers to connect.