Shopping Lists on the Cloud

Large Scale Distributed Systems

Gonçalo Domingues (202007914)
José Araújo (202007921)
Marcos Aires (202006888)

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Introduction

- Local-first application
- Users can create a new shopping list via the user interface
- Each shopping lists exists under a unique ID
- Users who are connected to a shopping list can add and delete items from it
- Consistent hashing
- Use of Conflict-Free Data Types (CRDTs) to maintain data integrity (PN-counter)
- Shopping lists replication
- Architecture based on Amazon Dynamo paper (https://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf)

Technical Solution (General)

- Database: SQLite
- ZeroMQ

Technical Solution (Client)

- Create new shopping lists
- Edit shopping lists (add and remove items)
- Client can create and edit both online and offline
- Many clients can edit the same shopping list at the same time
- Client can choose between offline and online mode
- Client can join other shopping lists when online
- Version resolution using CRDTs
- Due to implementation of database, clients can reconnect and still maintain their shopping lists

Technical Solution (Broker)

- Receives requests from Clients, re-routes them to corresponding Server, and sends the reply back
- Responsible for calculating hash key of shopping lists and servers (building hash ring)
- Possesses the current state of the hash ring
- Included in gossip protocol to maintain updated state
- Sloppy Quorum for sending shopping lists to Servers
- Responsible for adding new servers to hash ring

Technical Solution (Server)

- Gossip protocol
- Hinted-Handoff for shopping lists between servers
- When in a situation where Hinted-Handoff is working, Server is checking if other server is alive, if not removes it from ring
- Replicas are sent to at least 2 servers (if possible, if not only one or none)
- Replicas consistency
- When a new Server is added to hash ring, Servers exchange information between them in a decentralized way to send shopping lists from one Server to another
- Version resolution using CRDTs
- Server additions and removals supported

Limitations

- Not using virtual nodes, for this case due to stability, predictability and finding that our application would not benefit massively from it
- Sloppy Quorum and Hinted-Handoff working for shopping lists updating requests, but lacking in replication of server contents

CRDT

```python
class ShoppingListCRDT:
    def __init__(self, url, additions=None, removals=None):
        if additions is None:
            additions = {}
        if removals is None:
            removals = {}

        self.additions = additions
        self.removals = removals
        self.url = url
        self.key = None

    def merge(self, remote):
        merged_additions = {item: max(self.additions.get(item, 0), remote.addit
        merged_removals = {item: max(self.removals.get(item, 0), remote.removal

        return ShoppingListCRDT(self.url, merged_additions, merged_removals)

    def add_item(self, item):
        self.additions[item] = self.additions.get(item, 0) + 1

    def remove_item(self, item):
        self.removals[item] = self.removals.get(item, 0) + 1

    def get_items(self):
```