WishCloud: Local-First Shopping List Application

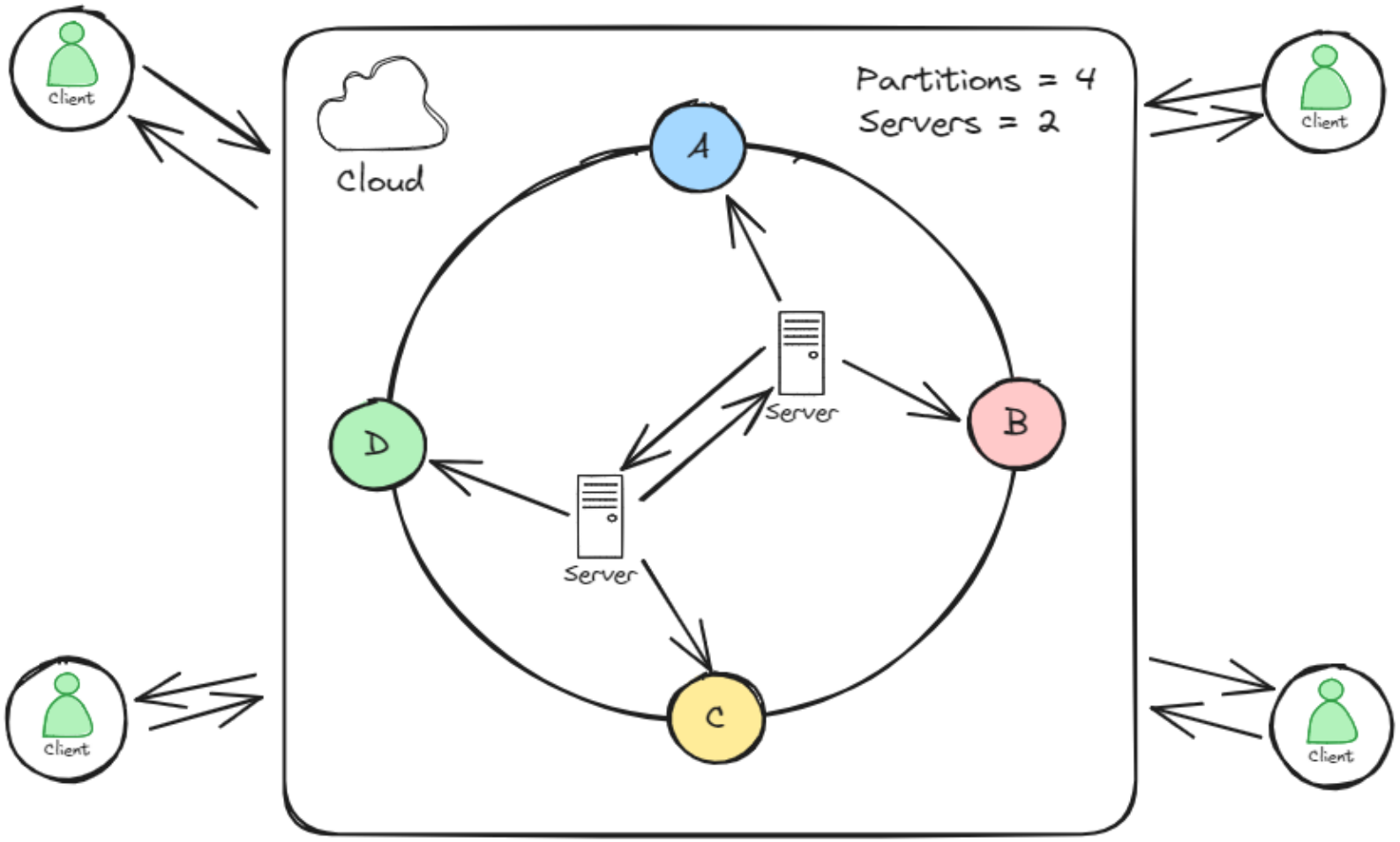
LARGE SCALE DISTRIBUTED SYSTEMS

Gonçalo Ferreira, up202004761
Ricardo Ribeiro, up202310095
Vítor Cavaleiro, up202004724

General Requirements

- Local First List Storage
- Cloud List Backup
- Handle Concurrent List Changes
- Solve Conflicts

Architecture



Client

- Create and store lists locally
- Local First List Storage – SQLite database
- Connect to cloud with hardcoded seeds
- Is award of the ring partitions
- Synchronize locally created and updated lists with the cloud
- Get shopping list’s preference list
- Access and store lists present in the cloud through their UUID
- Add and update items in the lists
- Merge local list with those coming from the cloud

Cloud

- Store seeds
- Instantiate and start servers

Server

- Configurable number of virtual nodes
- Handle requests from client:
  - Upload Request
    - Store a copy of the shopping list in the cluster
  - Read Request
    - Return the most up to date state of the shopping list
- Coordinator Actions:
  - Put and Get
- If server is the coordinator of a list, replicate the changes in other nodes in the shopping list
- Hinted Handoff (database to store misplaced list)

Consistent Hashing

- Hash function: MD5
- Uniform distribution of lists in the hash ring

Conflict Resolution

- Last-writer-wins CRDT Implementation
- Merge between Item CRDT's
- Merge between Lists (maps of items and CRDT's)

Evaluation of solution

STRONG POINTS

- Hinted Handoff (handling server failures)
- Load balance on client side
- Database Consistency
- Synchronized Database Writes

LIMITATIONS

- Inability to add or remove servers that are not seeds
- Weak handling of read quorum failure (only notifies client of the error)
- Size of partitions is not equal between virtual nodes
- Unscalable retrieval of hinted lists from database