Distributed Systems – Fault Tolerance

Lab 2

2022/2023

Quorum replication

Use quorums for consistent replication with the `lin-kv` service. Store data as $key \rightarrow (value, timestamp, writer)$ and implement the following operations:

- **Write** In two steps:
  - Step 1: Collect $timestamp$ from at least a quorum of servers for the desired key.
  - Step 2: Select the highest $timestamp$ and send $(value, timestamp + 1, writer)$ to the write quorum.
  - In each server, update key if stored $(timestamp, writer)$ is lower and return an acknowledgment.
  - Wait for sufficient acknowledgments and reply to client.
- **Read** Collect $(value, timestamp, writer)$ from a read quorum and return the $value$ with the highest $timestamp$ (and $writer$).
- **CAS** Return `unsupported`.

Steps

1. Implement the quorum protocol.
2. Test with different quorum combinations (ROWA, majority, ...).
3. Retest with increasing request rate (`--rate`) and network latency (`--latency`).
4. Discussion topics: What happens with concurrent writes? Does the protocol tolerate crashes? Can the CAS operation be supported (and tolerate crashes)?

Learning Outcomes Apply quorum replication to build a concurrent linearizable register. Recognize the relevance of quorums for fault tolerance.