P3 - Replicated Block Store

Yilei Hu, Wiley Corning, Yatharth Bindal

Replication Strategy

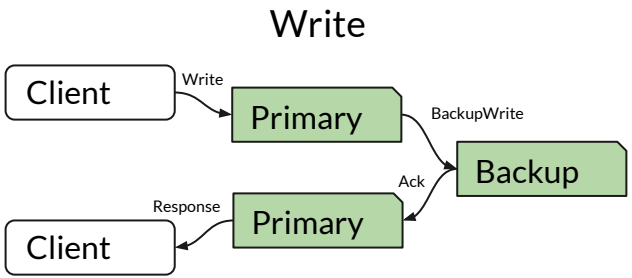
- Primary / Backup with two nodes.
  - Client interacts with Primary
  - Primary backs up data to Backup
- Client retries with Backup if Primary is unavailable, and vice versa
- The client switches to the backup when a crash is detected in the primary, where it has access until the primary server has fully recovered
- The client only receives the confirmation after the data has been committed to the backup

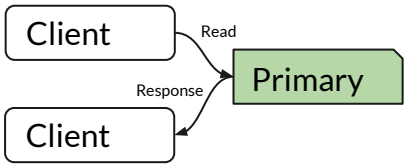
Fault Handling

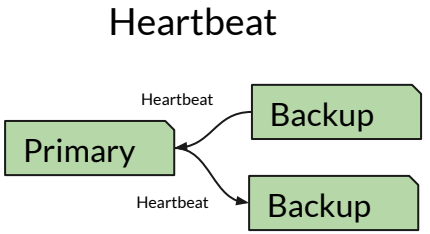
Each server can be in one of three states:

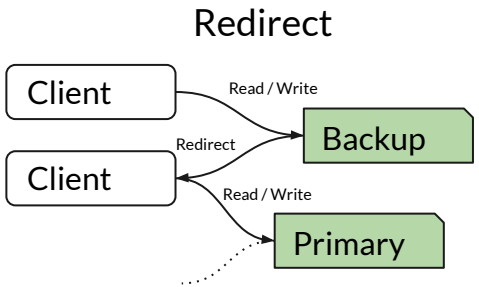
- **Normal.**
  - System is fully online
  - Server is acting as a replicated primary or as a pure backup
- **Standalone.**
  - Other server is inaccessible
  - Server will handle requests locally until the other has recovered
- **Recovering.**
  - Server is restarting after crashed
  - Server will request resynchronization

Protocols: Normal Operation



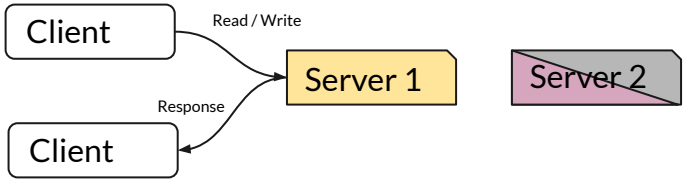




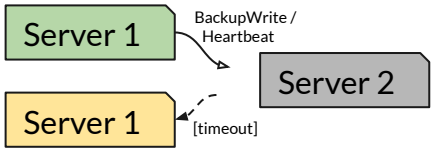


Protocols: Fault Behavior

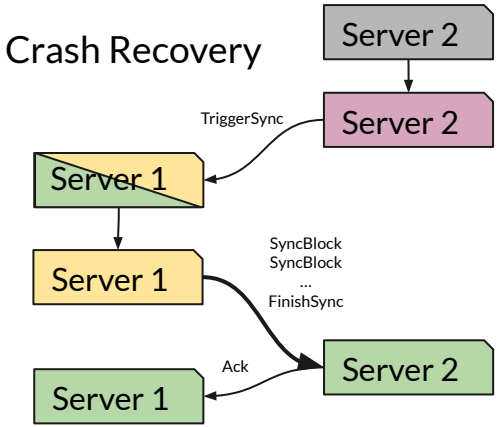
Standalone Operation



Crash Detection



Crash Recovery



Assumptions and Invariants

- Assume at least one server is healthy at all times
- Assume no network partitions (i.e., every message sent is received)
    - Given a partition, both servers would act as standalone

Given that the above hold, then

- At most one server will accept Read() or Write() at a time
- Strong consistency is guaranteed
- The system will eventually recover from any transient faults

Consistency Testing Method

- Server watches for specific addresses to be written
- One address primes the server to crash
- Others crash it in specific places
  - Primary: during write, before backing up
  - Primary: between writes
  - Backup: while backing up
  - Backup: between backups
  - Both: while recovering

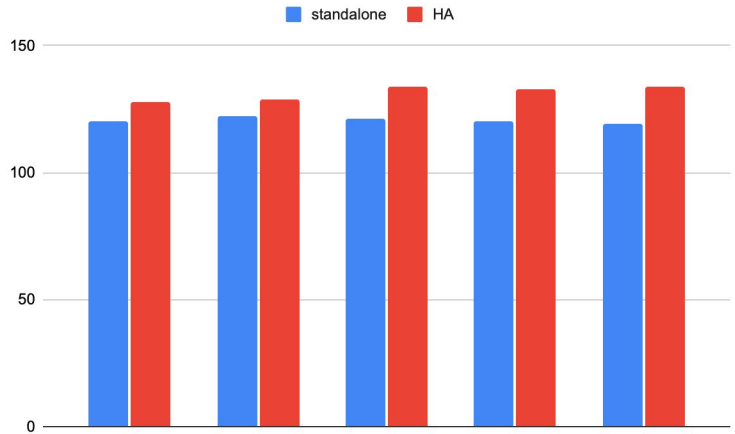
Crash Recovery: Live Demo

Cloud Configuration

- Client, Primary and Backup hosted on separate Google Cloud VMs
- Each VM has 2 cores and 4GB memory

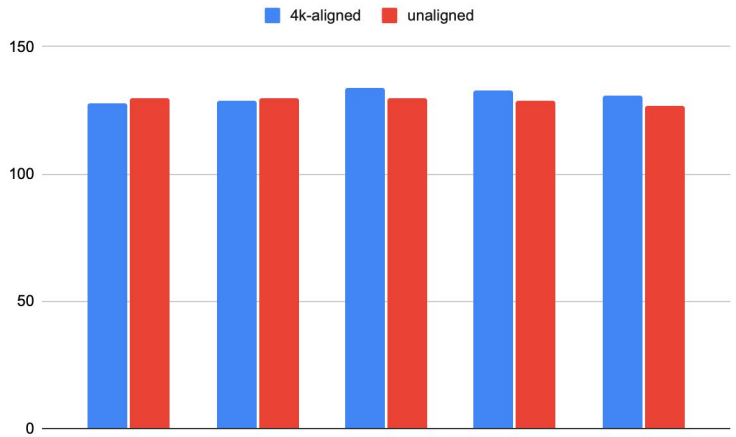
Write/Read performance

Time(ms) spent on writing different 4k Blocks to random addresses and reading them back:



4k aligned v.s. Unaligned addresses

Time(ms) spent on writing 4k Blocks to 4k aligned and unaligned addresses and reading them back:



Q&A

Thank you!