Cheap and Available State Machine Replication

Rong Shi, Yang Wang

The Ohio State University — Department of Computer Science and Engineering

Motivation: Replication Cost vs. Guarantees

- **Primary-backup:** f+1 replicas to tolerate f crash failures
- **Paxos:** 2f+1 replicas to tolerate asynchronous events and f crash failures
  - **Cheap Paxos:** f+1 replicas, sacrifice availability
  - **Gnothi**: f+1 replicas, sacrifice generality
  - ……
- **BFT**: 3f+1 replicas to tolerate arbitrary failures

Can we reduce replication cost without sacrificing other properties?

YES

Solution: Combine on-demand instantiation and lazy recovery

A. On-demand Instantiation

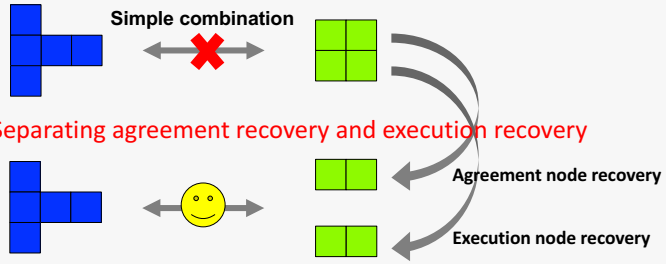
Activate a subset of replicas first and activate backup ones when active ones fail

Low cost

B. Lazy Recovery

Keep processing requests while recovering a backup replica in the background during recovery

High availability



Agreement nodes decide the next request to execute

Key observation: they do not need to know prior requests

Solution: when an agreement node fails, ask a blank one to join instantly

Execution nodes run the application's logic to execute tasks

Key observation: critical tasks may require fewer replicas than flexible tasks

Solution: activate sufficient replicas for critical tasks; use lazy recovery for flexible tasks

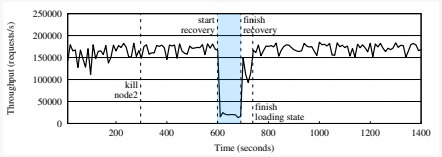
Solution: Adaptive Recovery

Complete recovery in a timely manner and make a balance between recovery and executing new requests

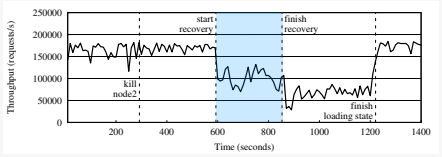
Why necessary?

- Increasing recovery time results in higher probability of data loss
- Long delayed flexible tasks (e.g. garbage collection) will eventually block the system

Method: compare $\frac{fetched\ Data}{total\ Data}$ and $\frac{elapsed\ Time}{deadline}$; if the former is smaller, increases recovery speed



Deadline = 100 seconds
Complete recovery in 92 seconds

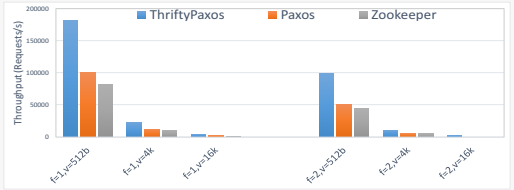


Deadline = 300 seconds
Complete recovery in 265 seconds

Evaluation

We apply the proposed techniques to Paxos (2f+1) and build ThriftyPaxos (f+1)

A. Throughput: fewer replicas lead to fewer messages and **higher throughput**



Throughput of writing to replicated RemoteHashMap

~80% improvement in TPS

B. Availability: ThriftPaxos achieves **same availability** as Paxos during failure recovery