Federating Consistency for Partition Prone Networks

Benjamin Bengfort and Pete Keleher
University of Maryland ● Department of Computer Science

Introduction

Groups of devices coordinated by consensus can efficiently order events under ideal conditions, but become less effective in dynamic and heterogeneous environments. Weakly consistent devices efficiently tolerate faults and dynamic conditions but are slow to converge on a single ordering of system events. Federated consistency combines the strengths of both approaches into a single protocol. Federated systems use a strongly consistent inner core to maintain a totally ordered sequence of accesses. A cloud of weakly-consistent devices disseminates orderings and enables progress despite varying connectivity.

Raft
Strong Consistency

Leader-oriented consensus algorithms nominate a dedicated proposer in a fault-tolerant manner to conduct consistent replication. All accesses go through the leader so linearity or sequential consistency is guaranteed. On a conflict, the leader simply drops the conflicting version.

Federated Tunable Consistency

Hybridization of Consistency by Replica Protocol

Consistency Integration
A vector component called the forti is only incremented by the Raft leader on commit.

Write

Policy
First Write to Leader Wins

Write

Eventual replicas must propagate forti bumps to all derived versions.

Forte Component+

Replicas select their local consistency policy but all Raft replicas must also participate in anti-entropy.

Write

Anti-Entropy
Eventual Consistency

Eventually consistent replication protocols utilize periodic pairwise anti-entropy sessions to synchronize replica states. Because there is no central authority, accesses are highly available and localized. However, anti-entropy tolerates inconsistencies and may even "stomp" writes by not fully replicating them.

Write

Policy
Latest Version Wins

Simulation

Our initial evaluation utilized discrete event simulations with asynchronous processes to investigate the effect of latency, conflict, and network outages on consistency. Every experiment includes multiple simulated runs of 20 processes in varying environments and configurations, accessing multiple objects concurrently.

The primary simulation parameter was variable latency across wide areas, described by the mean and standard deviation of one-way message latency. Other parameters include the likelihood of conflict & outages, timing parameters for consistency protocols & workload, and distributions for anti-entropy selection.