

Geo-replication integrated all the way to the client machine

Marek Zawirski¹ Annette Bieniusa² Valter Balegas³ Sérgio Duarte³ Carlos Baquero⁴ Marc Shapiro¹ Nuno Preguiça³
¹Inria & UPMC-LIP6 ²U. Kaiserslautern ³U. Nova de Lisboa ⁴INESC Tec. & U. Minho

Motivation

limitations of server-centric geo-replication

Ad-hoc client-side caching – today’s solution plagued with issues:
 => **Error-prone** application-level logic
 => **Inconsistent** on partial cache misses or failures (no metadata/updates)

Goal

extend geo-replication to the client machine

Integrated solution – expectations:

- **Lower latency and improved availability** for some operations
- **(Causally) consistent access to partial replicas despite faults**

Problem (1) with naïve approach

liveness of causal consistency w/o full (meta)data

example execution: replicated updates and causal dependencies

Failover problem: reads in **NY** blocked ($y=7$) or inconsistent ($y=0?$)
Cause: non-replicated causal dependency $y:=7$

Multi-versioned approach

read stable updates of all clients + own updates

Read/depend on slightly old version:

- **stable updates** (> 1 server replica)
- **own recoverable updates**

- **Consistent access on failover** vs. inconsistent in asynchronous systems
- **No added WAN latency** vs. high latency in quorum-synchronous systems
- **Low staleness increase:** $\leq 1\%$ more stale reads under contention

Problem (2) with naïve approach

inefficient or insufficient metadata

Approach A
 client-assigned update id + Version Vectors encoding causal dependencies

id: $(C_1, 17)$ depends on: $[C_1=16, C_2=5, \dots, C_n=3]$

Efficiency problem: unsustainable vector size

Approach B
 server-assigned update id + any efficient encoding

depends on: $[WA=77, NY=58]$

Safety problem: ≥ 1 update id multiple execution of non-idempotent updates

e.g. transient WA fault

Hybrid approach

separated concerns: update identity and summary

update lifecycle

1. Client assigns a unique id: prevents duplicated update log entries in case of failures
- 2a. Server assigns an alias id: reference for efficient summary of updates
- 2b. Rare failure path (failover/retry): Assign new server id if needed. Identify any duplicates by client id
3. Metadata compaction upon eventual full replication

last id: $x:=1$