COPS: Scalable Causal Consistency for Wide-Area Storage

Wyatt Lloyd, Michael J. Freedman, Michael Kaminsky, and David G. Andersen

Presented by Troy Stacer

Diagrams and Figures from COPS
Motivation
COPS (Cluster of Order-Preserving Servers) is a key value store for wide area networks that provides causal+ consistency, is always on and is scalable.
Ideal Properties

- Strongly consistent (linearizability)
- Always Available
- Work even with network partitions

Impossible to have all three (CAP theorem)
Always On and Scalable - ALPS

- **Availability**
  - Operations always complete and do not block indefinitely
- **Low Latency**
  - Operations occur quickly
- **Partition Tolerance**
  - Service still works if data centers are separated
- **High Scalability**
  - Adding N resources increases capacity by $O(n)$
Consistency

High Consistency

Strong (linearizability) (impossible with ALPS)
Sequential (impossible with ALPS)
**Causal + (new for this paper)**
Causal
Per-Key Sequential
FIFO
Eventual

Low Consistency
Causal Consistency

All causally related events are in the same order on all servers.
Conflict Handling: Convergence

For events that are not causally related, the data centers should agree on the value.

COPS defines Causal consistency with conflict handling as Causal+. 

\[
\text{Put}(x, 5) \quad X=5 \quad \text{Put}(x, 2)
\]
COPS Overview
COPS Key Points

• Get and Put operations
  • Goes through client library to update/attach dependency metadata
• Each data center contains all the data partitioned among servers
• Tracks Causality with metadata
• Don’t expose Puts until all dependencies are satisfied
Get

Local Datacenter

Client Library

get

get
Dependencies

Client 2

get(K) -> value, version, deps'

value -> get(K)

deps

K_{version}

L_{337}

M_{195}
Dependencies

Client 1

- put(Key, Val)
- True/False

put_after(Key, Val, deps)

deps
K
version

version
Causal+ Replication

Exposing values after dep_checks return ensures causal+
Causal Dependencies

Only need to check Nearest Dependencies
COPS GT OVERVIEW
Get Transaction Motivation

isBobFriend=True

Server A: Start

isBobFriend=False

Server A: after 2 put actions
Get Transaction Motivation

isBobFriend=True

Server B: get(friendsList)

isBobFriend =False

Server B: get(album)
COPS-GT

• Adds get transactions
• Provides a causal consistent view of multiple key value pairs in a single call
• Less efficient for write heavy workloads
• Must store all dependencies and not just nearest dependencies
Evaluations
Evaluation
Questions