Wren: Nonblocking Reads in a Partitioned Transactional Causally Consistent Data Store

Kristina Spirovska

Advisor: Willy Zwaenepoel

Diego Didona

Research area:
Causal Consistency in Distributed Data Stores

PhD Stage: Finisher

EuroDW’18, April 23, Porto, Portugal
Existing geo-replicated, causally consistent data stores are sub-optimal (performance, scalability, resource efficiency)

Performance, scalability and resource efficiency matter in the real world

Novel system design that achieves up to:

- 3.6x lower latency than state of the art
- 1.4x higher throughput

Trade-off: reading slightly from the past
Causal Consistency

Higher Performance

Stronger Consistency Guarantees

Eventual Consistency

Causal Consistency

Linearizability (Strong Consistency)

Strongest consistency model compatible with availability
Transactional Causal Consistency

Causal consistency

= 

Interactive Read-Write Transactions
Transactional Causal Consistency

- Reads from causal snapshot
- Writes are atomic

Challenge under sharding
Wren vs. Cure [ICDCS’16]

Read-heavy workload

- **Cure**
- **Wren**

**Response time (msec)** vs **Throughput (TX/sec)**

Lower and more to the right is better.
Our solution: Wren

- Achieves nonblocking reads
  - Low latency
- Scales horizontally by sharding
  - Scalability
- Tolerates network partitions between DCs
  - Availability

Trade-off: reading slightly from the past
Atomic writes + Sharding = 2PC

UNCERTAINTY PERIODS

PREPARE

COMMIT

$P_x$

$P_y$

$C_2$

$C_1$
Cure [ICDCS'16]
Our solution: **Wren**
Wren vs. Cure

Diagram showing the process of preparing and committing in a distributed system.
Contributions:

**CANToR: Client-Assisted Nonblocking Transactional Reads**
- New dependency tracking protocol
- Novel transactional protocol

**BDT: Binary Dependency Time**
- New dependency tracking protocol

**BiST: Binary Stable Time**
- New stabilization protocol