ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

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

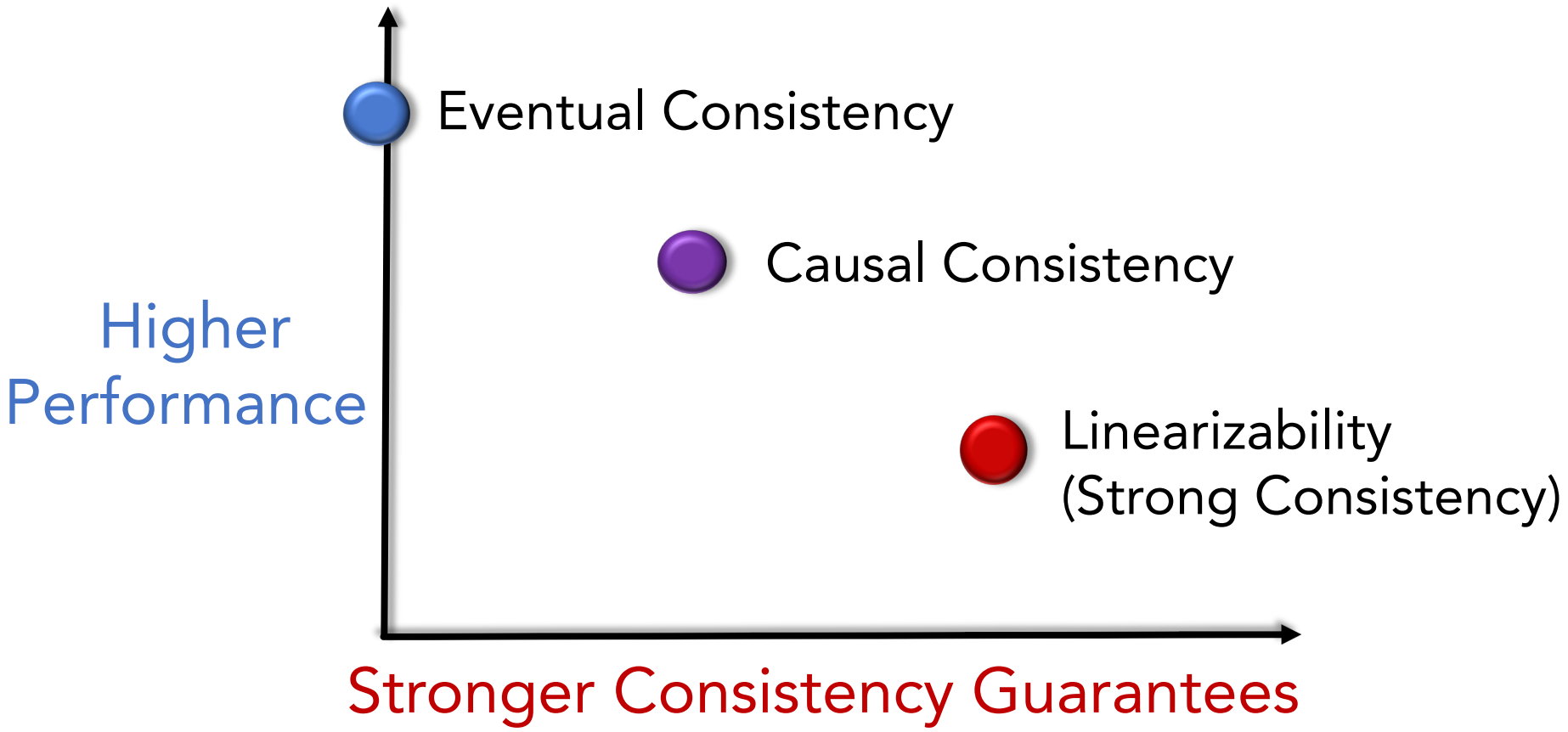
Novel system design that achieves up to:

- **3.6x lower latency**
- **1.4x higher throughput**

than state of the art

Trade-off: reading slightly from the past

Causal Consistency



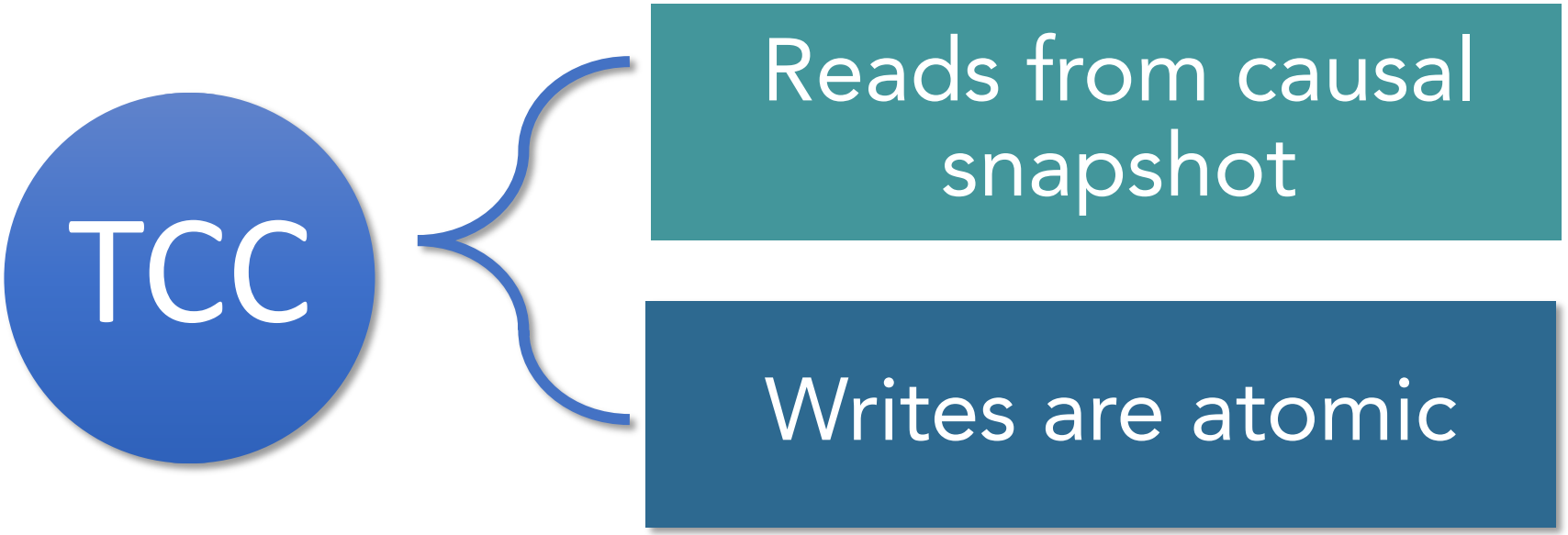
Strongest consistency model compatible with availability

Transactional Causal Consistency

Causal consistency

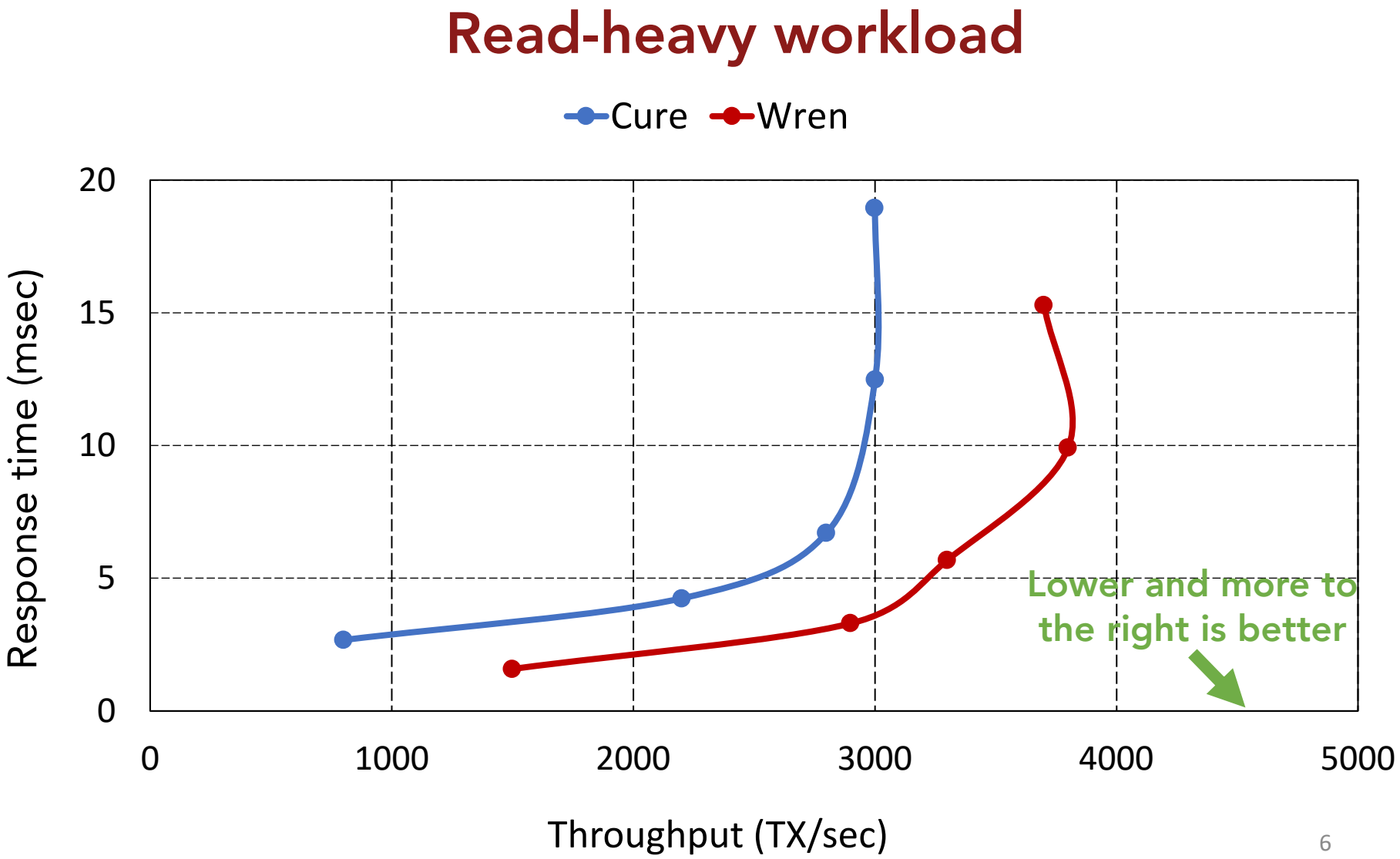
Interactive Read-Write Transactions

Transactional Causal Consistency



Challenge under sharding

Wren vs. Cure [ICDCS'16]



Our solution : Wren

Achieves nonblocking reads

- Low latency

Scales horizontally by sharding

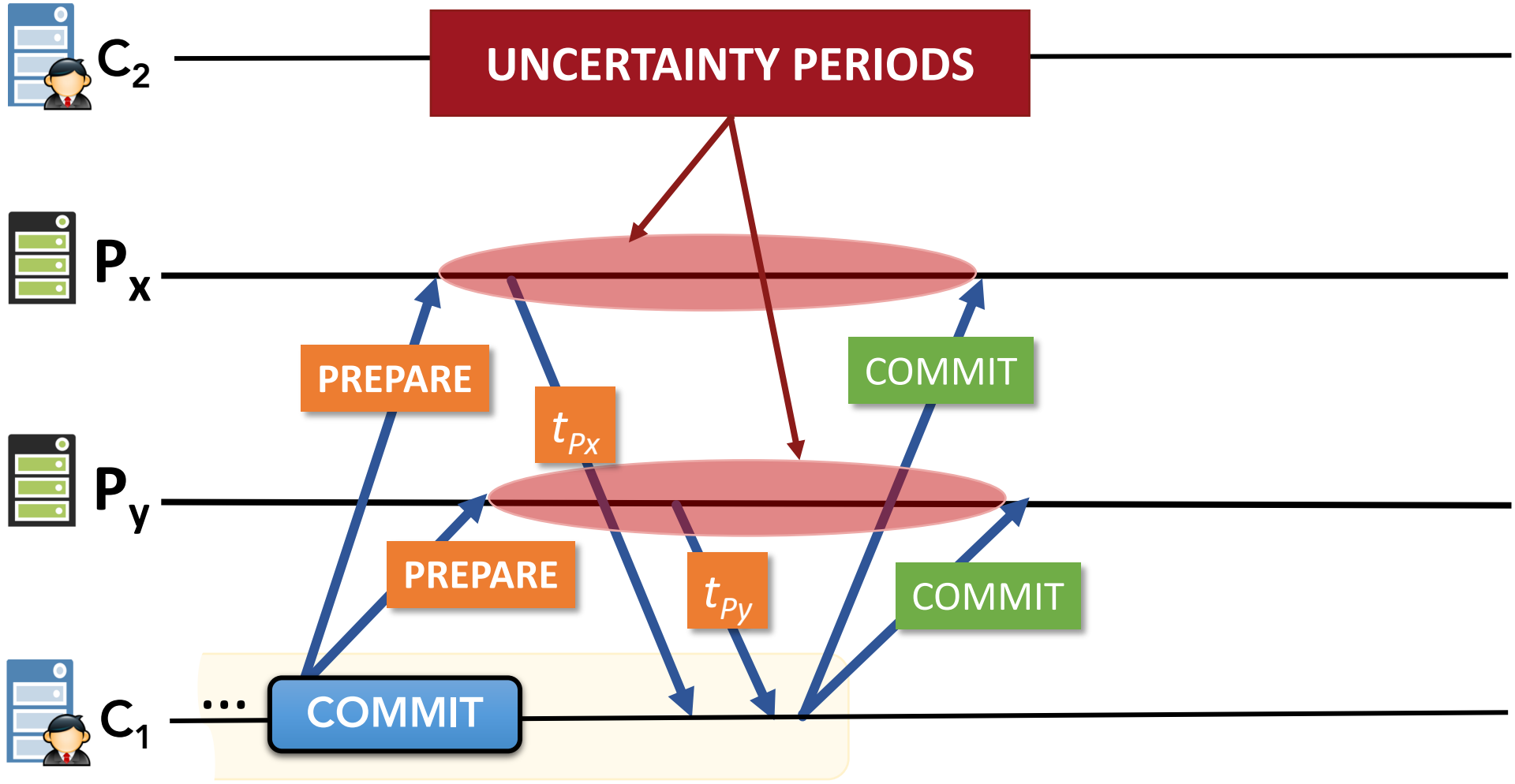
- Scalability

Tolerates network partitions between DCs

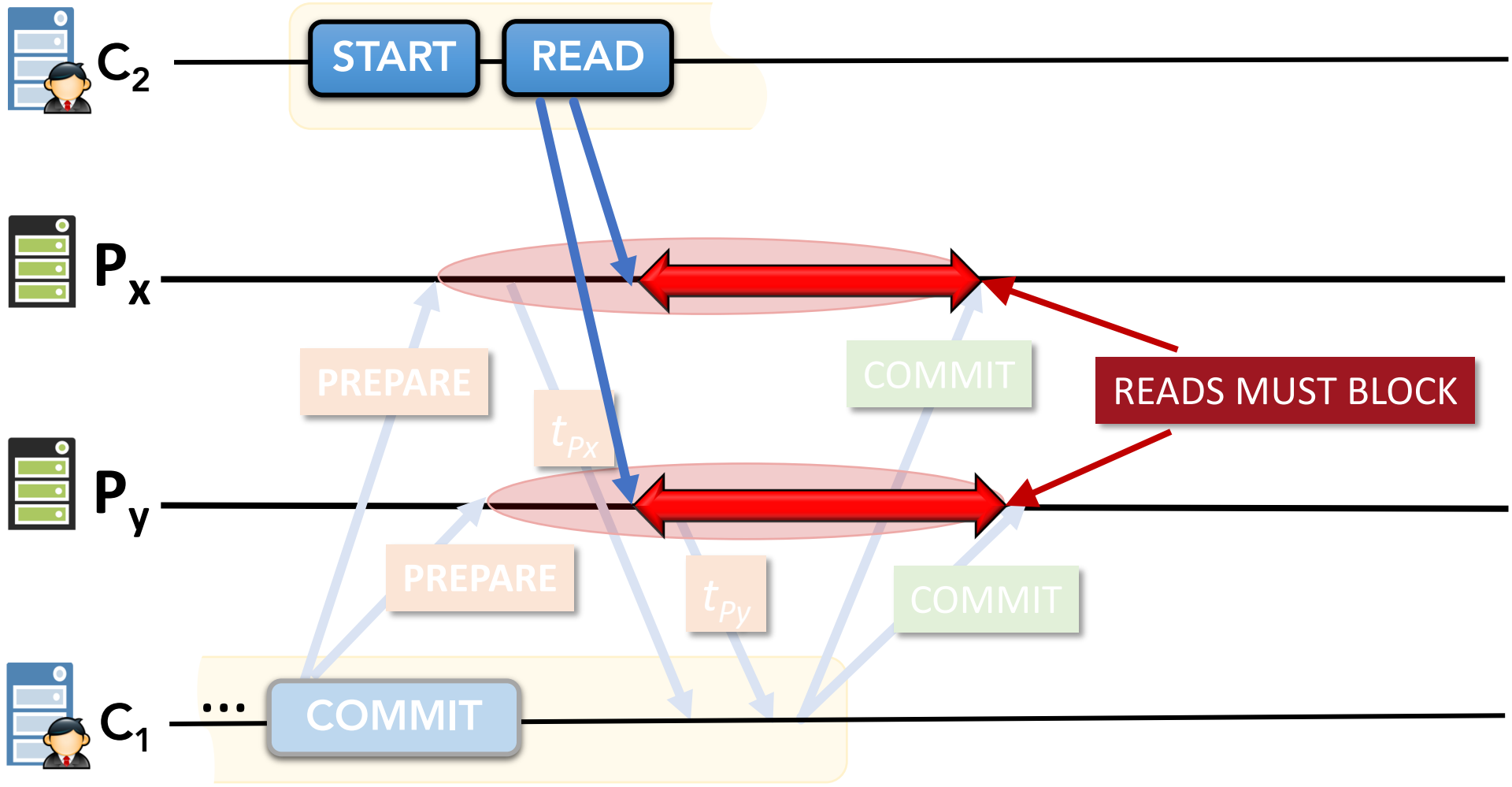
- Availability

Trade-off:
reading slightly from the past

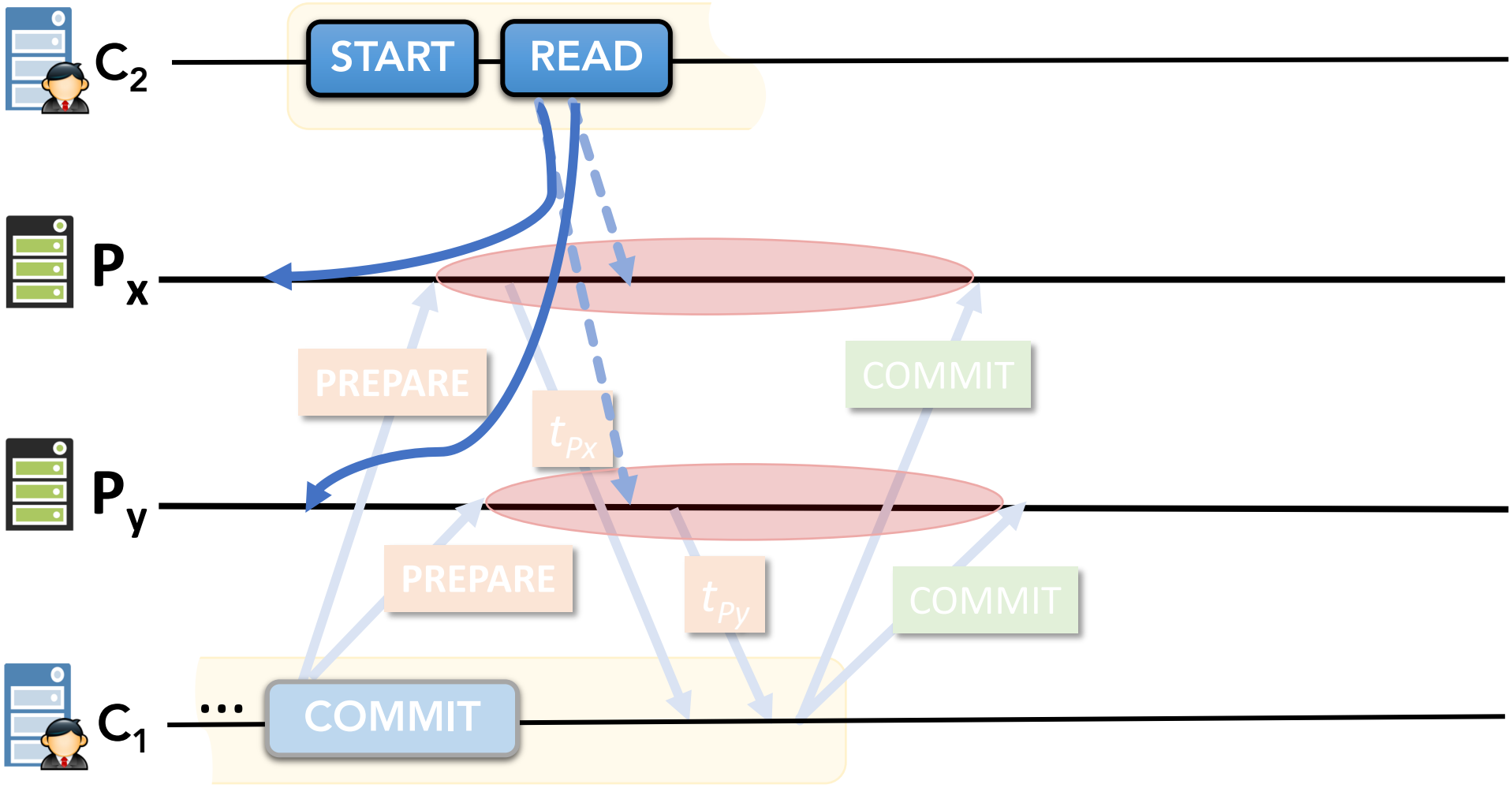
Atomic writes + Sharding = 2PC



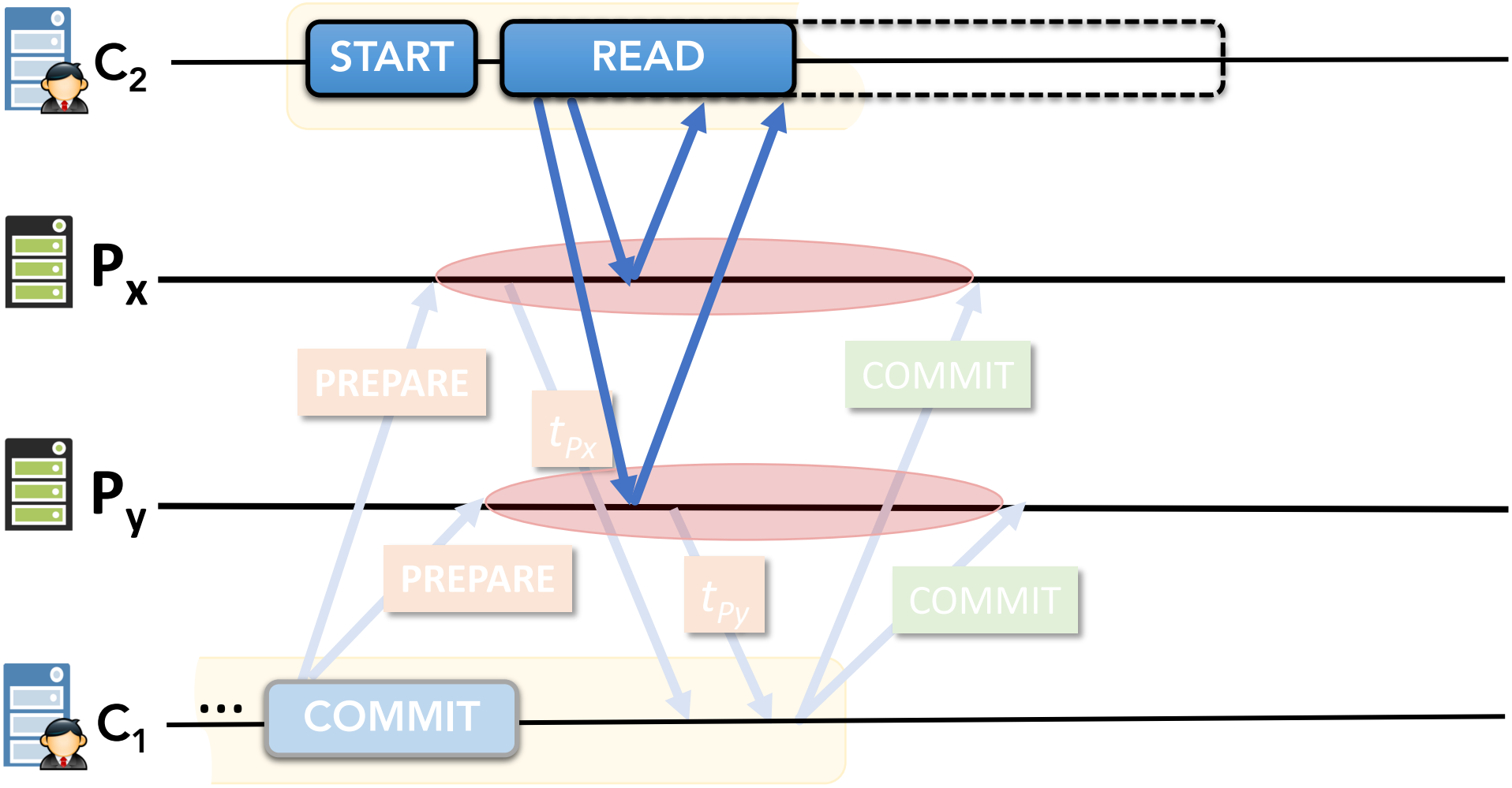
Cure [ICDCS'16]



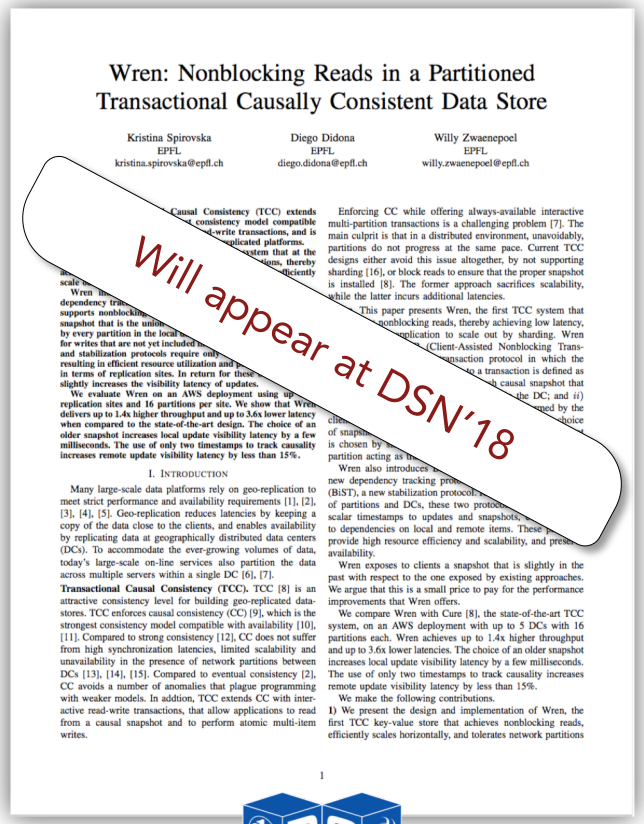
Our solution: Wren



Wren vs. Cure


C2
START
READ
Px
PREPARE
tPx
COMMIT
Py
PREPARE
tPy
COMMIT
C1
...
COMMIT

Wren – DSN'18



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

Kristina Spirovska, EPFL, kristina.spirovska@epfl.ch
Diego Didona, EPFL, diego.didona@epfl.ch
Willy Zwaenepoel, EPFL, willy.zwaenepoel@epfl.ch

Transactional Causal Consistency (TCC) extends … **consistency model compatible** … **-write transactions, and is** … **plicated platforms.** … **system that at the** … **tions, thereby** … **ficiently**

Wren … **dependency tra** … **supports nonblockin** … **snapshot that is the union** … **by every partition in the local** … **for writes that are not yet included** … **and stabilization protocols require only** … **resulting in efficient resource utilization and** … **in terms of replication sites. In return for these** … **slightly increases the visibility latency of updates.**

We evaluate Wren on an AWS deployment using up … **replication sites and 16 partitions per site. We show that Wren delivers up to 1.4x higher throughput and up to 3.6x lower latency when compared to the state-of-the-art design. The choice of an older snapshot increases local update visibility latency by a few milliseconds. The use of only two timestamps to track causality increases remote update visibility latency by less than 15%.**

I. Introduction

Many large-scale data platforms rely on geo-replication to meet strict performance and availability requirements [1], [2], [3], [4], [5]. Geo-replication reduces latencies by keeping a copy of the data close to the clients, and enables availability by replicating data at geographically distributed data centers (DCs). To accommodate the ever-growing volumes of data, today's large-scale on-line services also partition the data across multiple servers within a single DC [6], [7].

Transactional Causal Consistency (TCC). TCC [8] is an attractive consistency level for building geo-replicated data-stores. TCC enforces causal consistency (CC) [9], which is the strongest consistency model compatible with availability [10], [11]. Compared to strong consistency [12], CC does not suffer from high synchronization latencies, limited scalability and unavailability in the presence of network partitions between DCs [13], [14], [15]. Compared to eventual consistency [2], CC avoids a number of anomalies that plague programming with weaker models. In addition, TCC extends CC with interactive read-write transactions, that allow applications to read from a causal snapshot and to perform atomic multi-item writes.

Enforcing CC while offering always-available interactive multi-partition transactions is a challenging problem [7]. The main culprit is that in a distributed environment, unavoidably, partitions do not progress at the same pace. Current TCC designs either avoid this issue altogether, by not supporting sharding [16], or block reads to ensure that the proper snapshot is installed [8]. The former approach sacrifices scalability, while the latter incurs additional latencies.

This paper presents Wren, the first TCC system that … nonblocking reads, thereby achieving low latency, … plication to scale out by sharding. Wren … (Client-Assisted Nonblocking Trans- … nsaction protocol in which the … to a transaction is defined as … h causal snapshot that … the DC; and *ii*) … med by the … choice … of snapsh … is chosen by … partition acting as …

Wren also introduces … new dependency tracking proto… (BiST), a new stabilization protocol … of partitions and DCs, these two protoc… scalar timestamps to updates and snapshots, … to dependencies on local and remote items. These … provide high resource efficiency and scalability, and pres… availability.

Wren exposes to clients a snapshot that is slightly in the past with respect to the one exposed by existing approaches. We argue that this is a small price to pay for the performance improvements that Wren offers.

We compare Wren with Cure [8], the state-of-the-art TCC system, on an AWS deployment with up to 5 DCs with 16 partitions each. Wren achieves up to 1.4x higher throughput and up to 3.6x lower latencies. The choice of an older snapshot increases local update visibility latency by a few milliseconds. The use of only two timestamps to track causality increases remote update visibility latency by less than 15%.

We make the following contributions.

1) We present the design and implementation of Wren, the first TCC key-value store that achieves nonblocking reads, efficiently scales horizontally, and tolerates network partitions

1

Will appear at DSN'18

Contributions:

CANToR: Client-Assisted Nonblocking Transactional Reads

- Novel transactional protocol

BDT: Binary Dependency Time

- New dependency tracking protocol

BiST: Binary Stable Time

- New stabilization protocol