

Toward Coordination-Free and Reconfigurable Mixed Concurrency Control



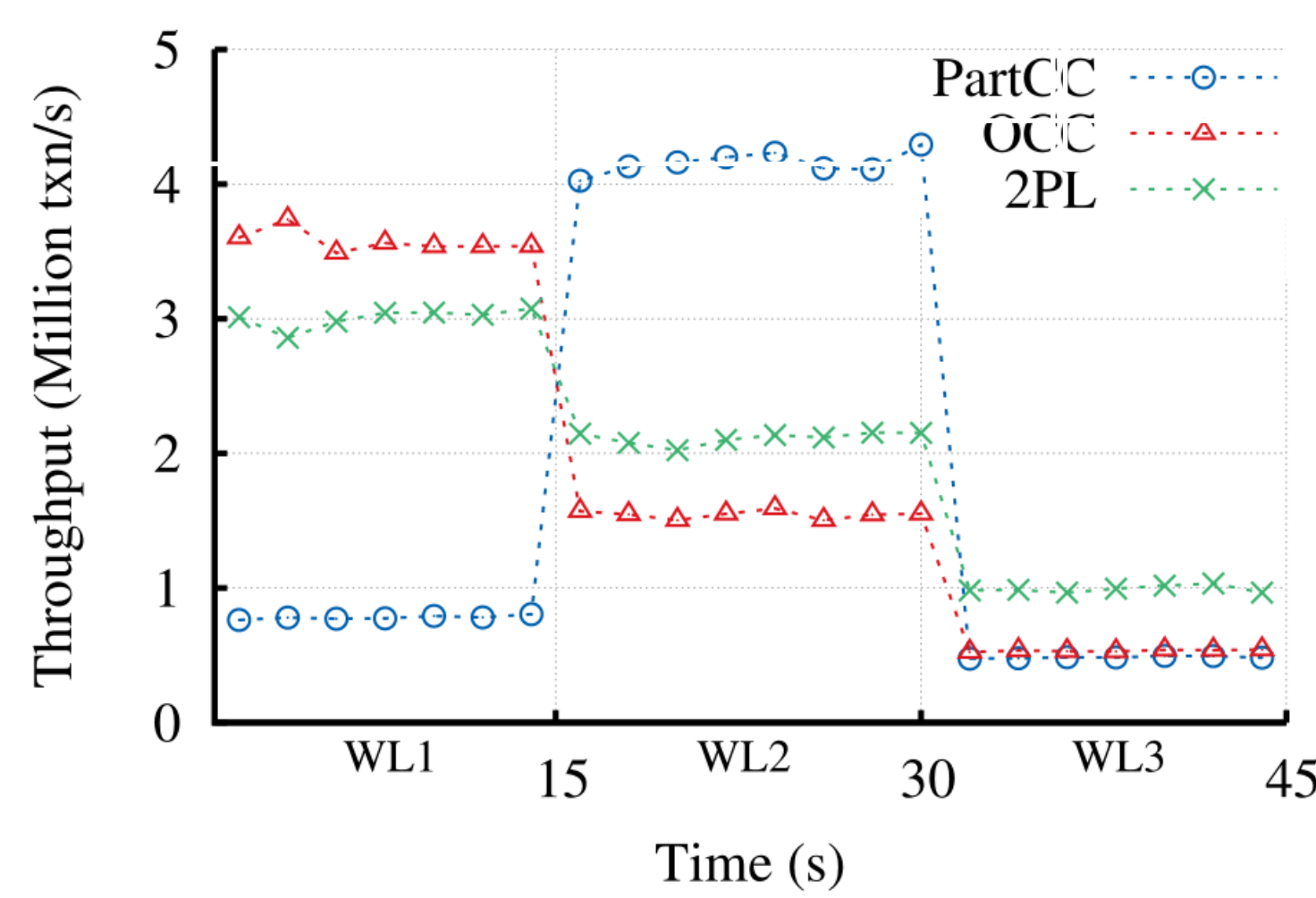
THE UNIVERSITY OF
CHICAGO

Dixin Tang (totemtang@uchicago.edu), Aaron J. Elmore

Motivation

- ❖ One Concurrency Control Protocol Cannot Fit All Workloads

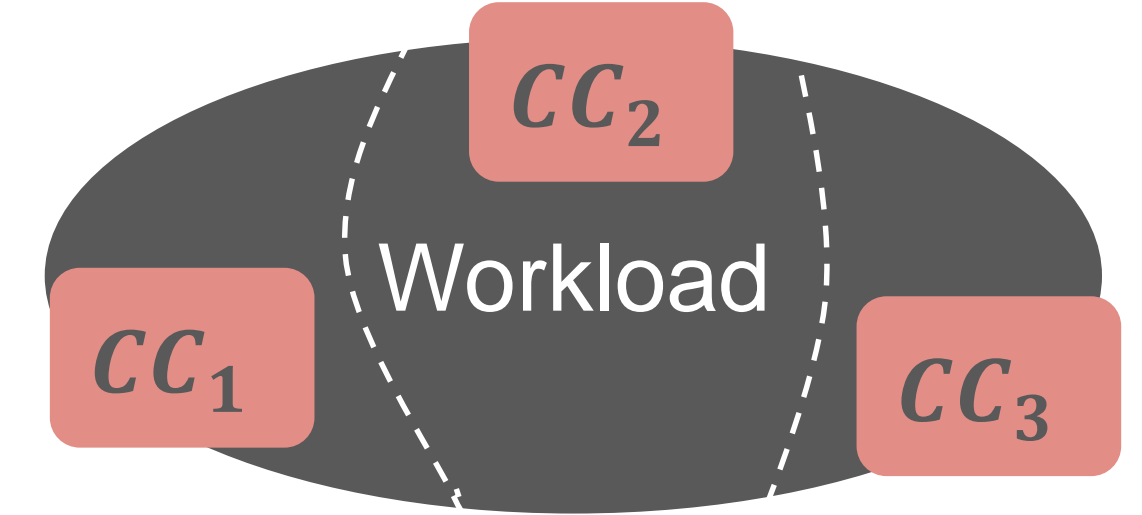
- Testing three protocols using YCSB workloads
- OCC from Silo, 2PL from VLL, PartCC from H-Store



Workload 1	Not partitionable Read Only
Workload 2	Partitionable
Workload 3	Not partitionable High skew, Write heavy

Our Solution – Mixed Concurrency Control

- ❖ Each protocol can process the part of workload it is optimized for
- ❖ Each protocol can avoid being brittle to workload where it suffers



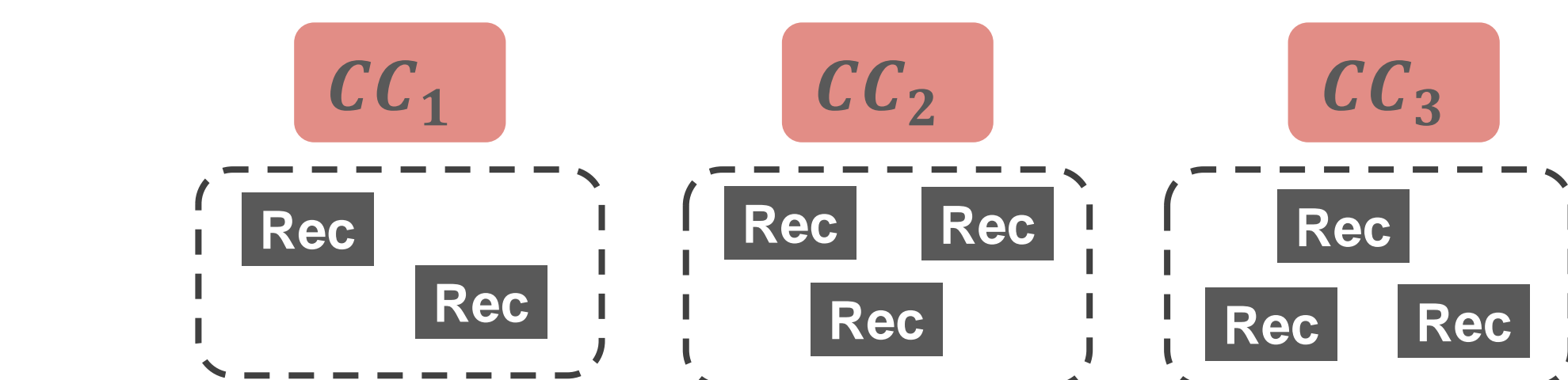
Our Goals

- ❖ Achieve mixed concurrency control with low overhead
- ❖ Support online protocol reconfiguration
- ❖ Guarantee correctness (Serializability and Deadlock Free)

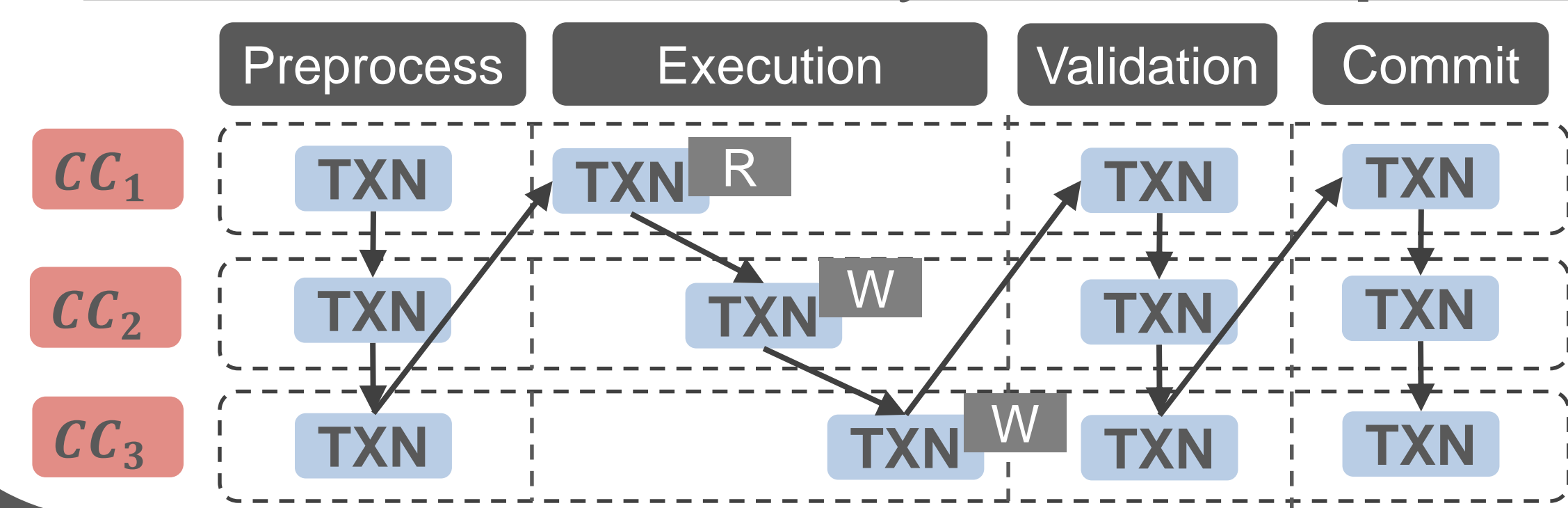
Our Approach – CormCC

Partition Workload and Assign Protocols

- ❖ Partition by record access characteristics
- ❖ Assign a single protocol to process all read/write operations to that partition



- ❖ Break a transaction life cycle into four phases



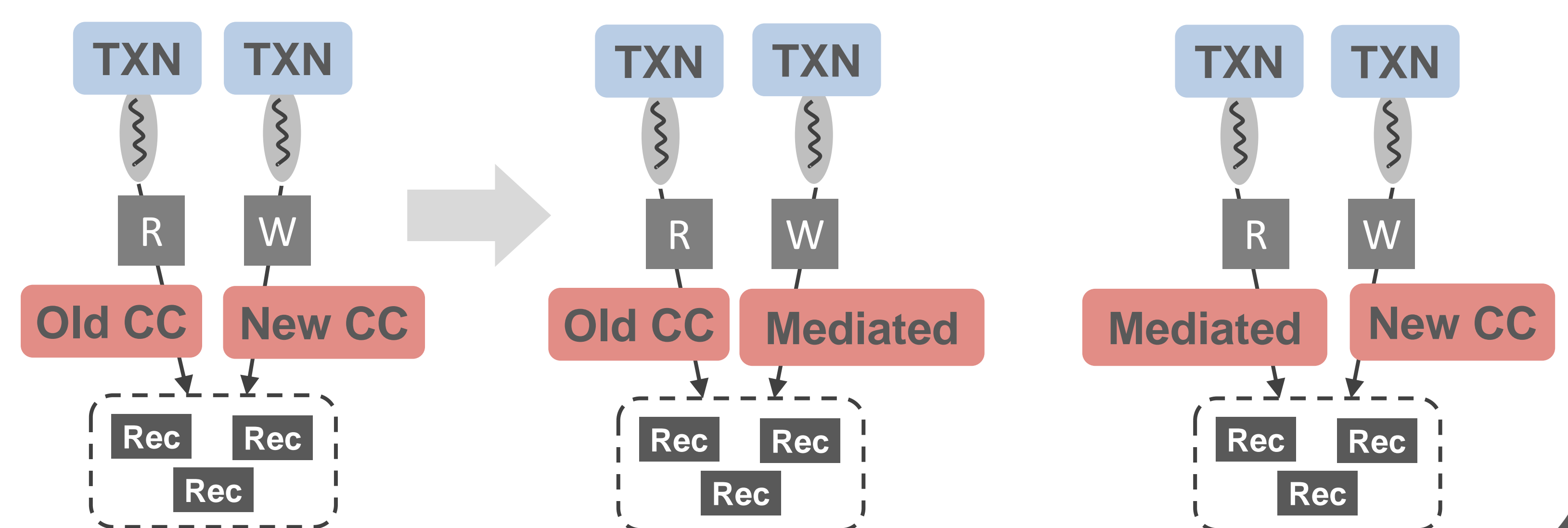
Deadlock Free

- ❖ We require each protocol can exclusively let transactions wait in no more than one phase
 - No deadlock within one phase
 - Transactions in earlier phases can wait for later phases, but not the other way around

Online Reconfiguration via Mediated Protocol

Challenge of online reconfiguration

Our solution: using a mediated protocol that executes logics of both old and new CC



Serializability via COCSR

- ❖ COCSR: Commit Ordering Conflict Serializable
- ❖ Commit ordering respects conflicts

If all protocols are COCSR, then CormCC is COCSR

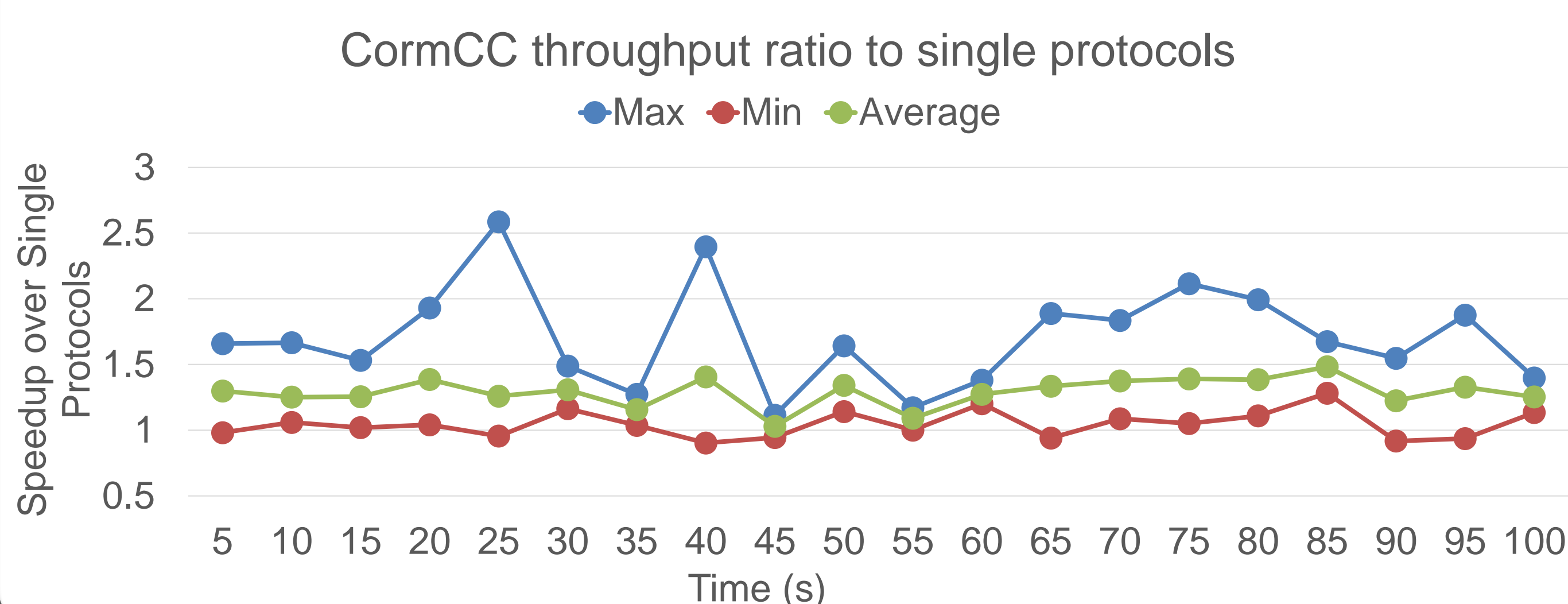
Experiments – TPC-C

Prototype

- ❖ Supporting PartCC from H-Store, OCC from Silo, and 2PL from VLL
- ❖ Partition the whole database and apply each partition a single protocol
- ❖ Selecting the ideal protocol based on feature engineering:
 - Partition Conflicts, Record Contention, Read Rate, and Transaction Length

Tests over varied workloads

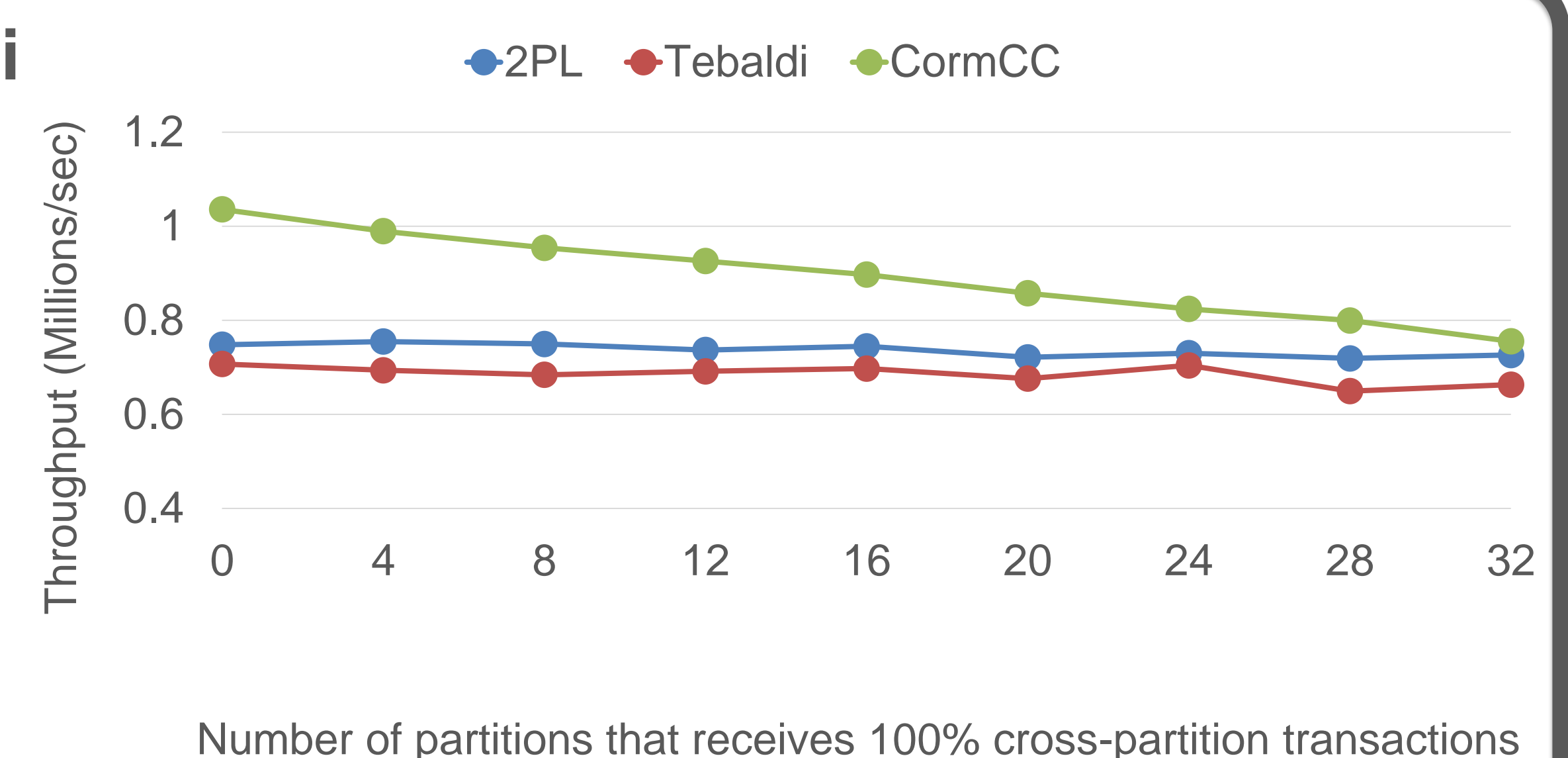
- ❖ Vary parameters every 5 seconds
- ❖ Parameters: Transactions mix, Percentages of cross-partition transactions, Access skewness (i.e. theta of Zipf)



Compared with Tebaldi

- ❖ Start with well-partitionable workloads

- ❖ Increase the number of partitions receiving 100% cross-partition transactions



Testing Mediated Switching

- ❖ Switch from OCC to 2PL

- ❖ Test a short-only workload and workloads with one long transaction of different duration

- ❖ Test different switching points

