Adaptive
Data Storage & Placement in Distributed Database Systems

Michael Abebe

mtabebe@uwaterloo.ca

UNIVERSITY OF WATERLOO

August 2022

CockroachDB

TiDB

Distributed DBMSs are widely used

SQL

Amazon Aurora

VOLTDB

Distributed Databases

How and where to store data?

?

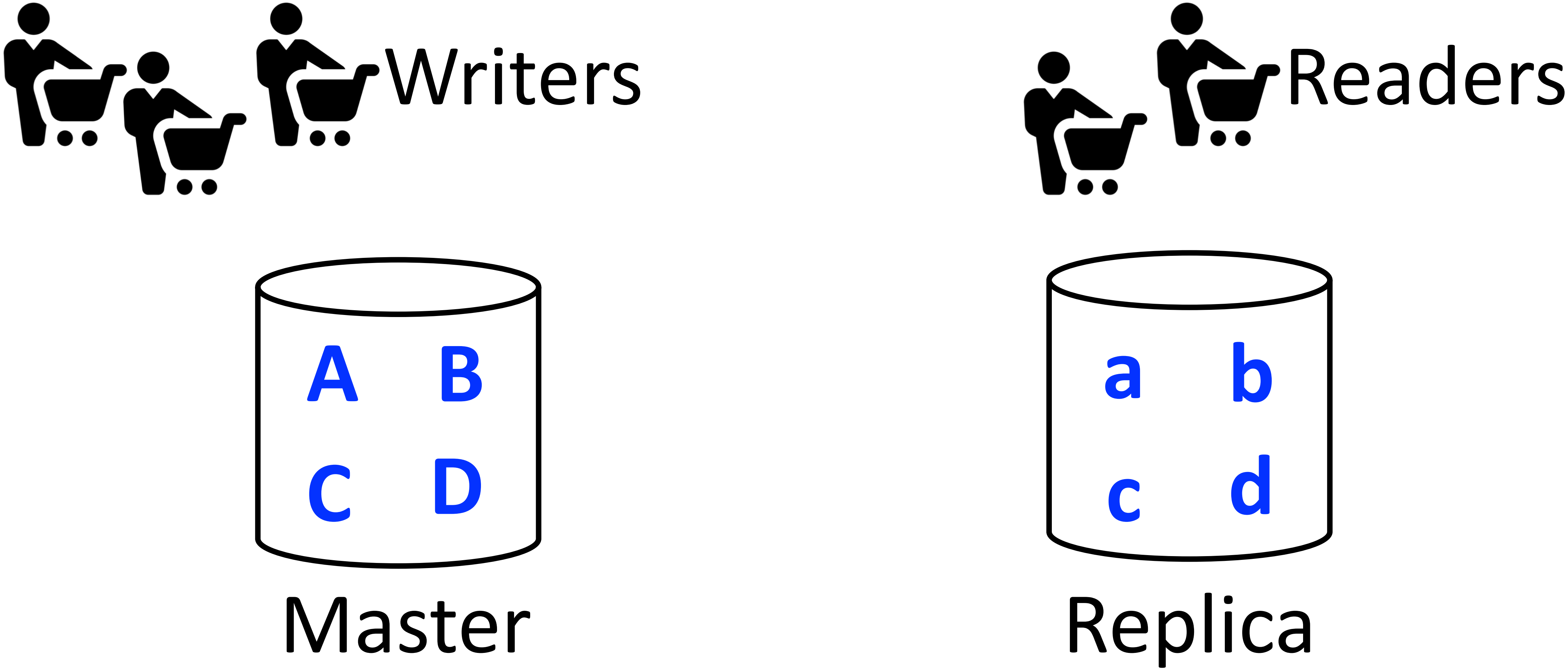
Replication

Partitioning

Format

?

Database Replication



Writers

Readers

A B

C D

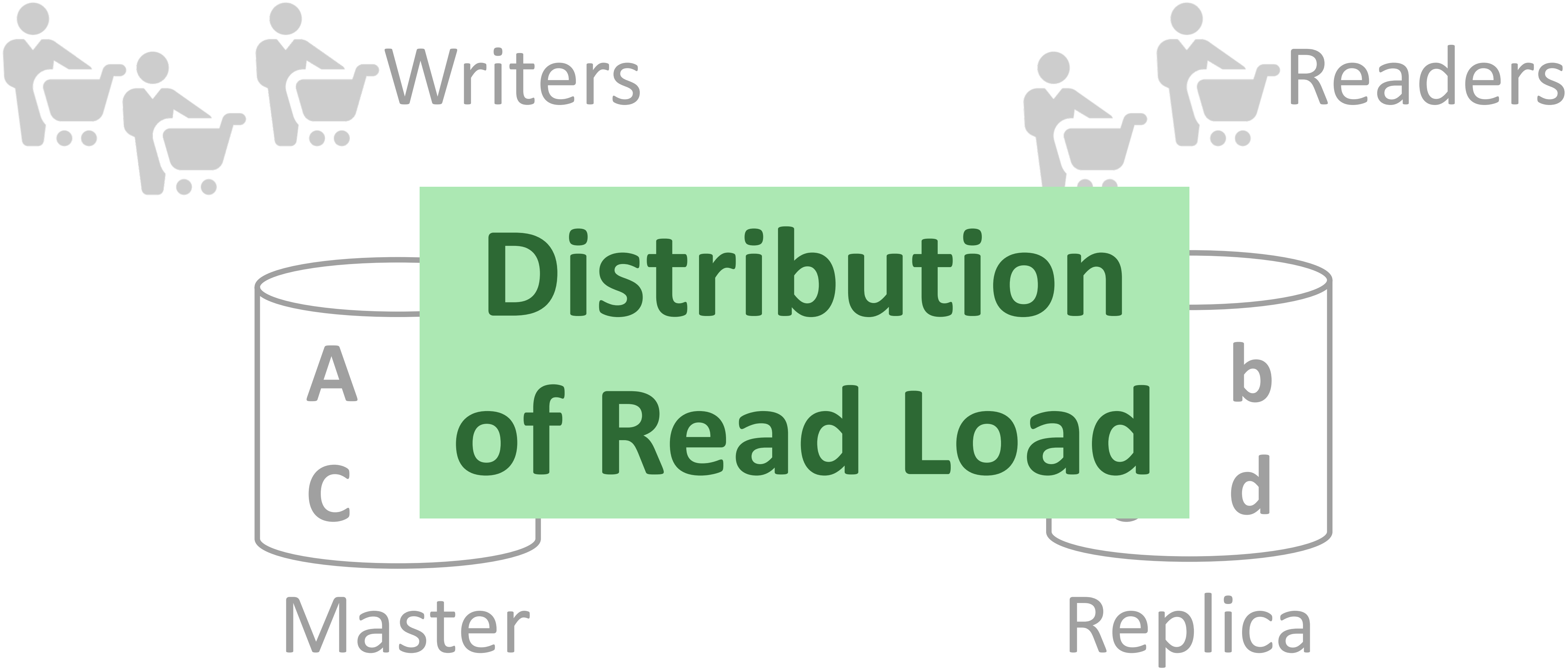
a b

c d

Master

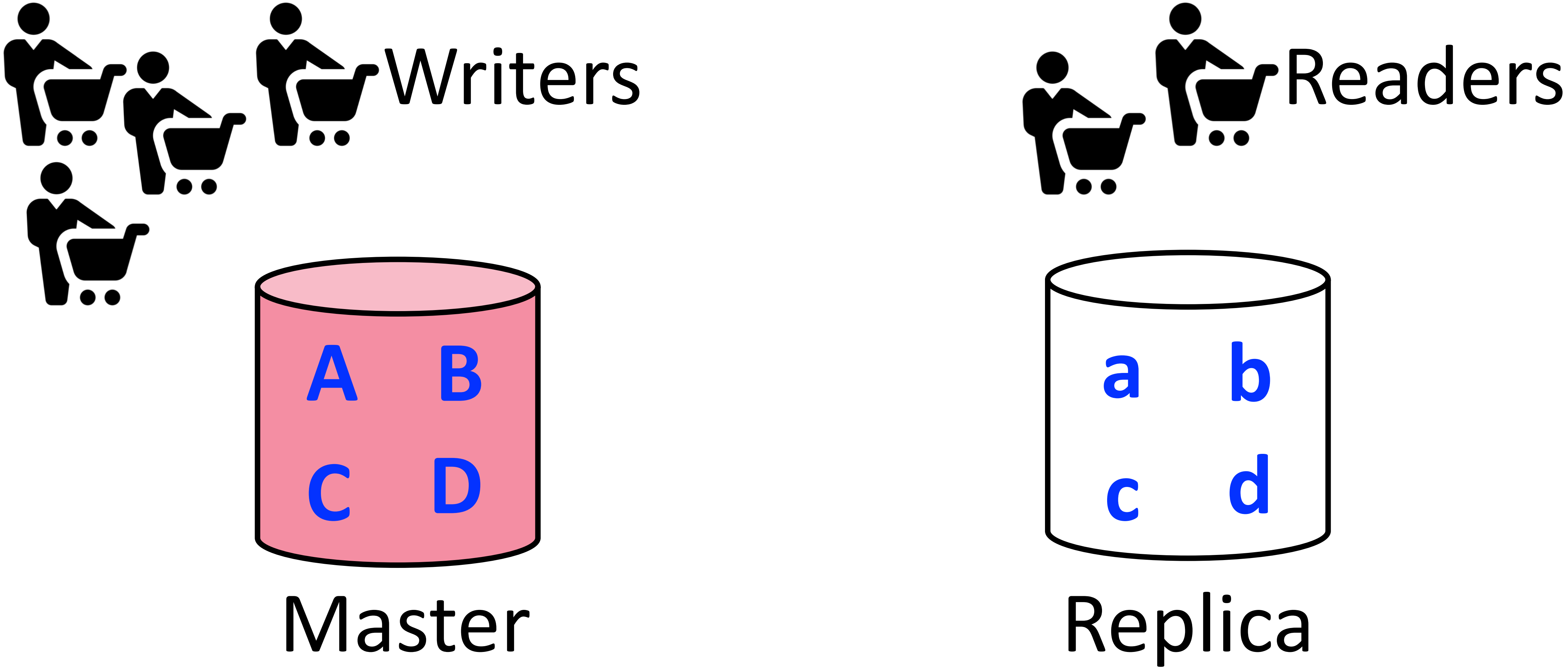
Replica

Database Replication



Distribution of Read Load

Database Replication



Writers

Readers

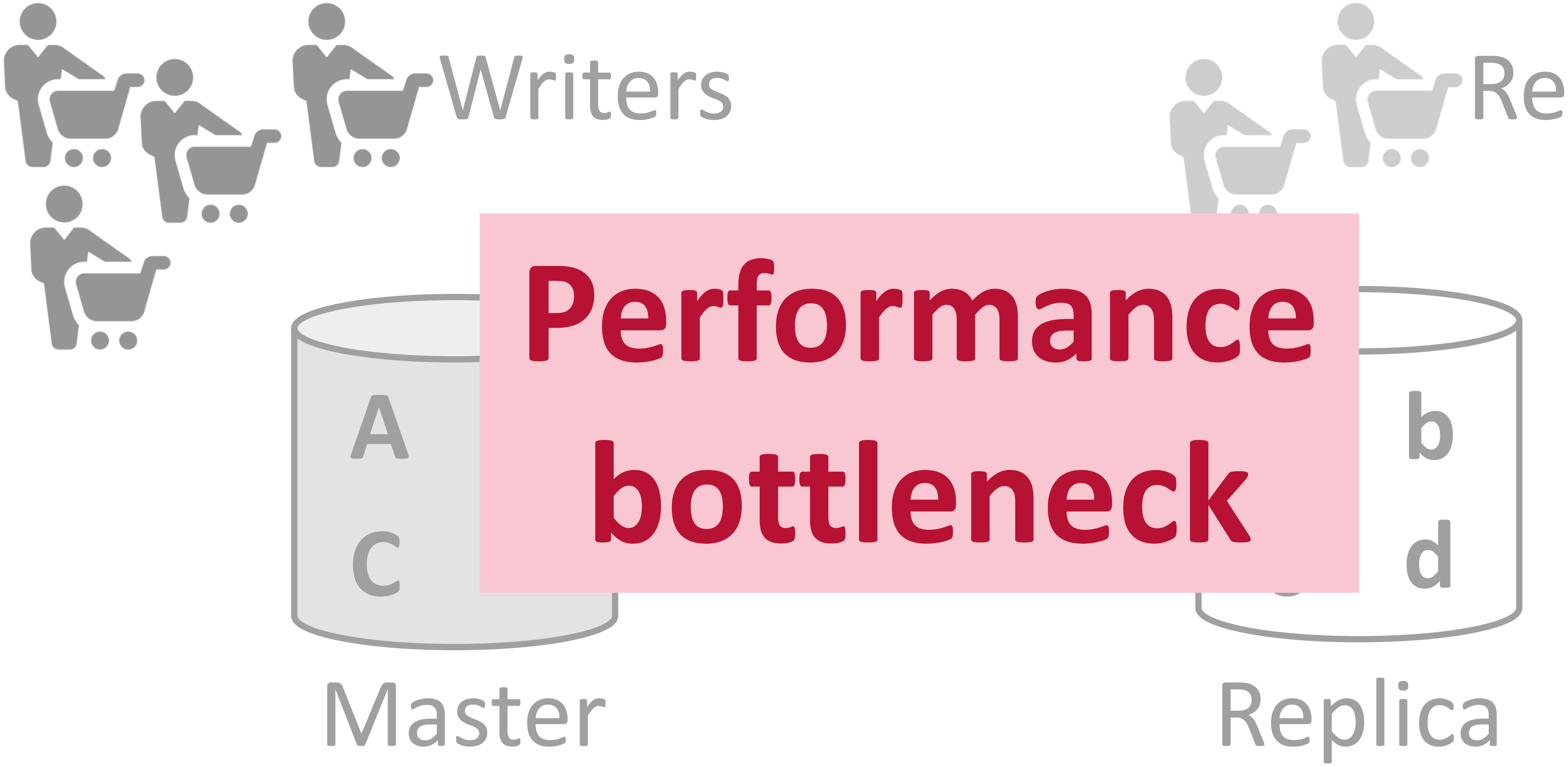
A B
C D

a b
c d

Master

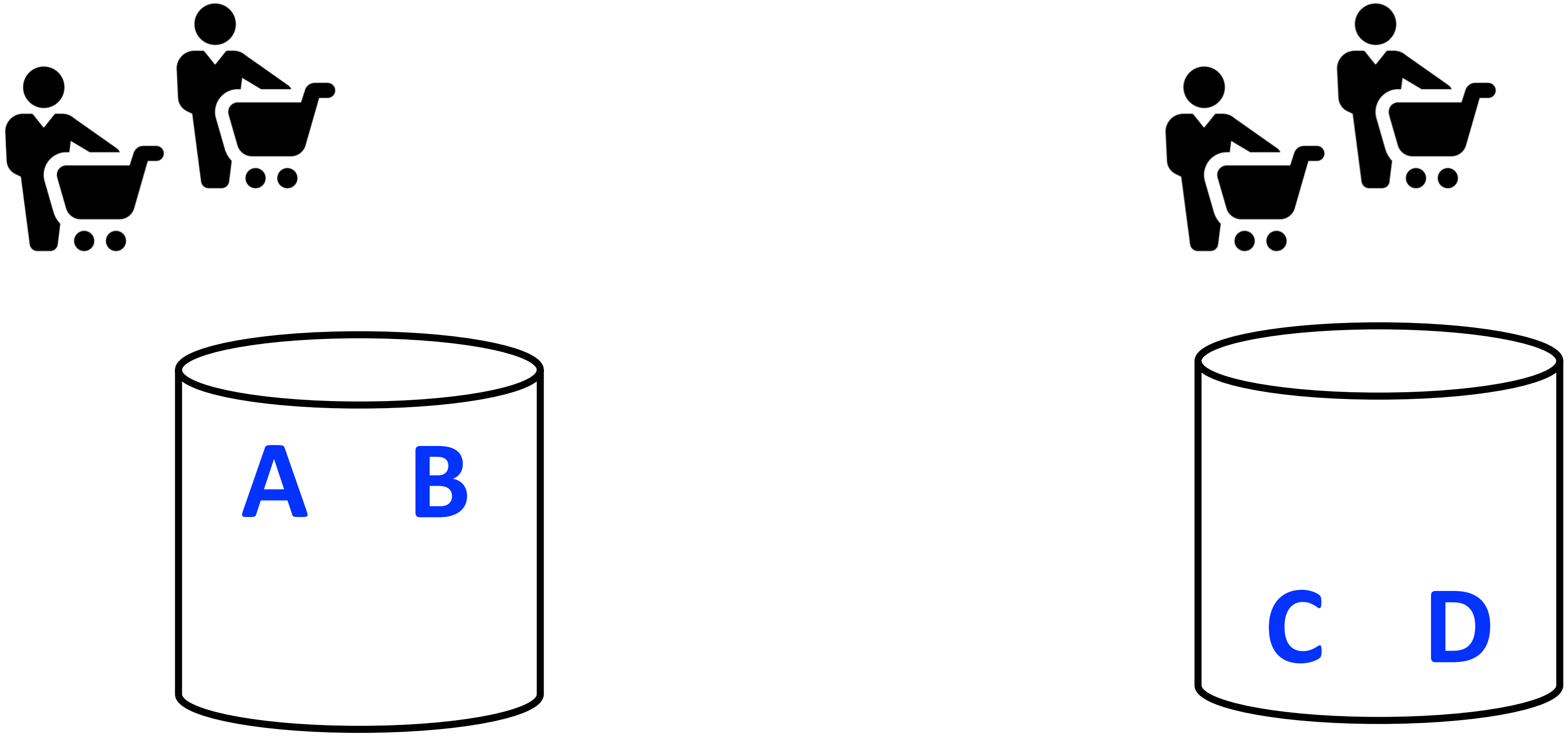
Replica

Database Replication

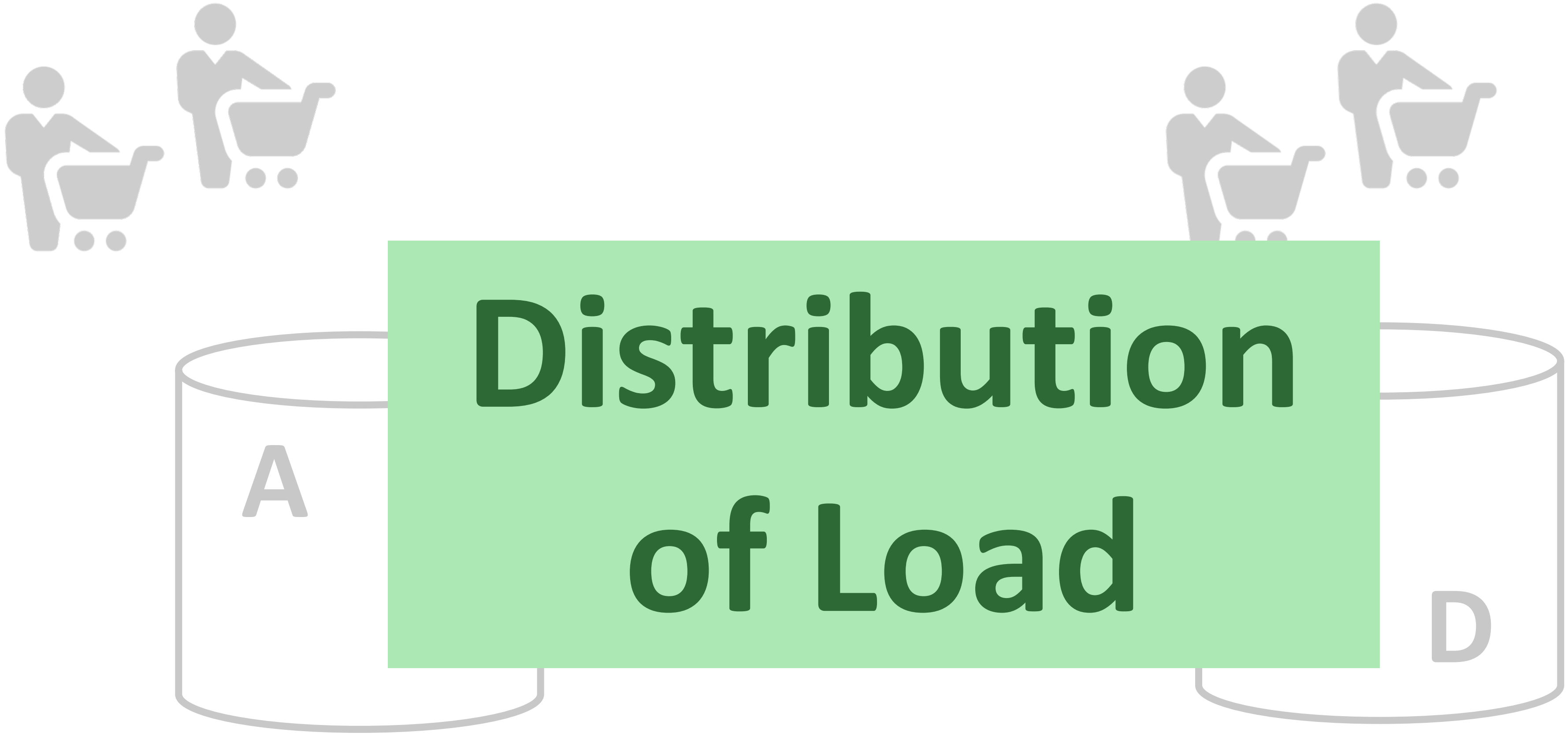


Performance bottleneck

Database Partitioning

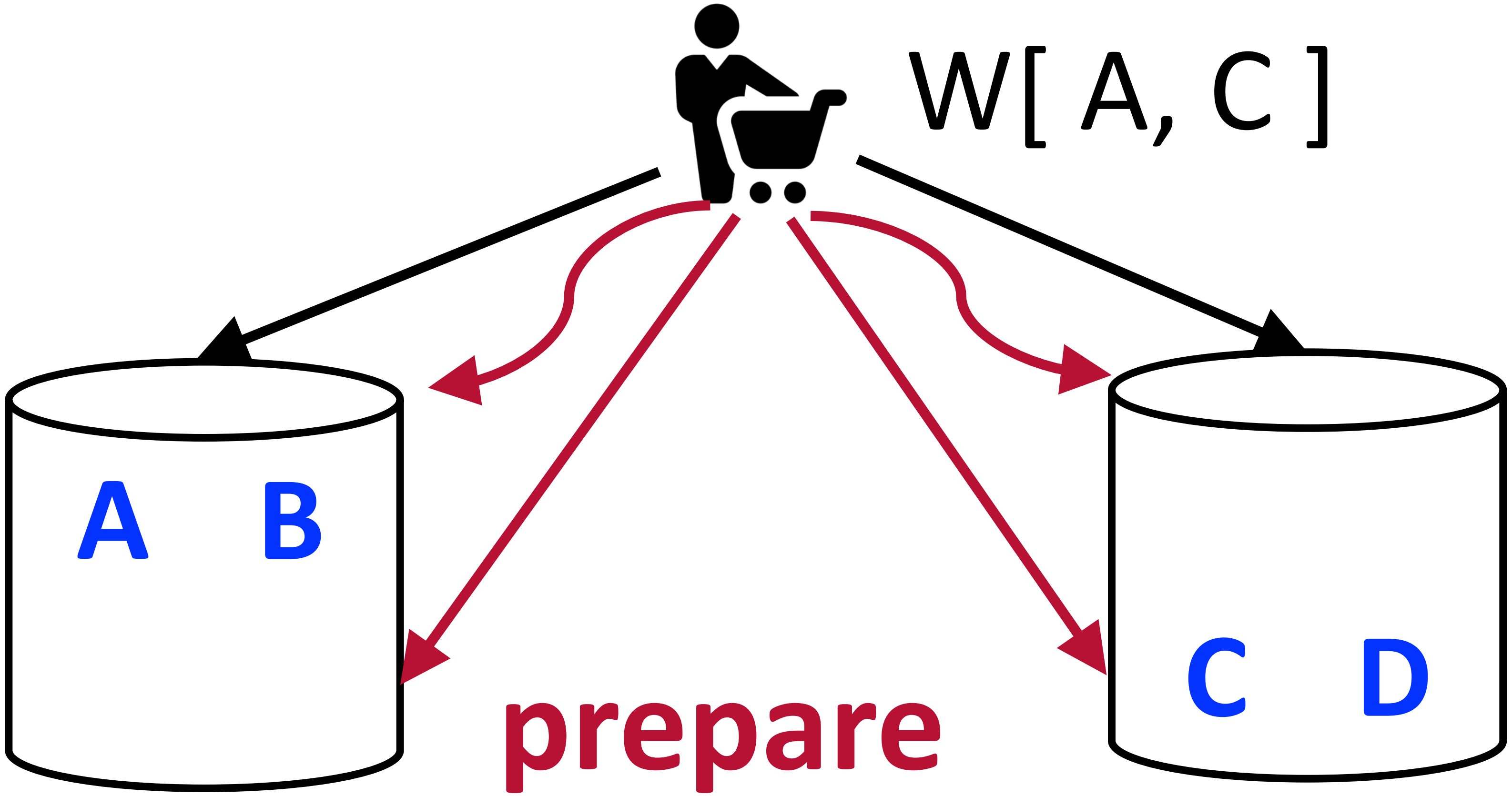


Database Partitioning



Distribution of Load

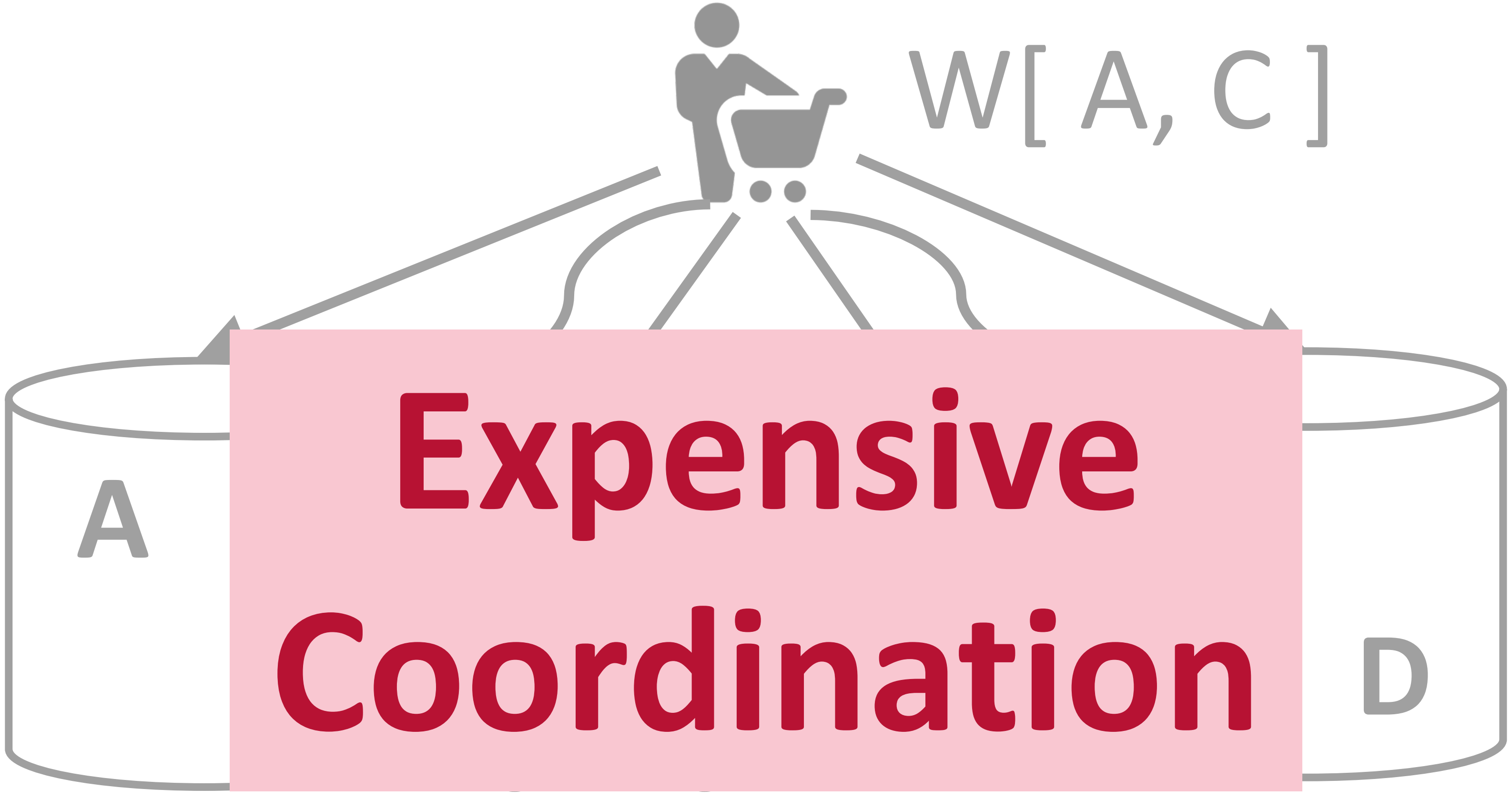
Database Partitioning



prepare

commit

Database Partitioning

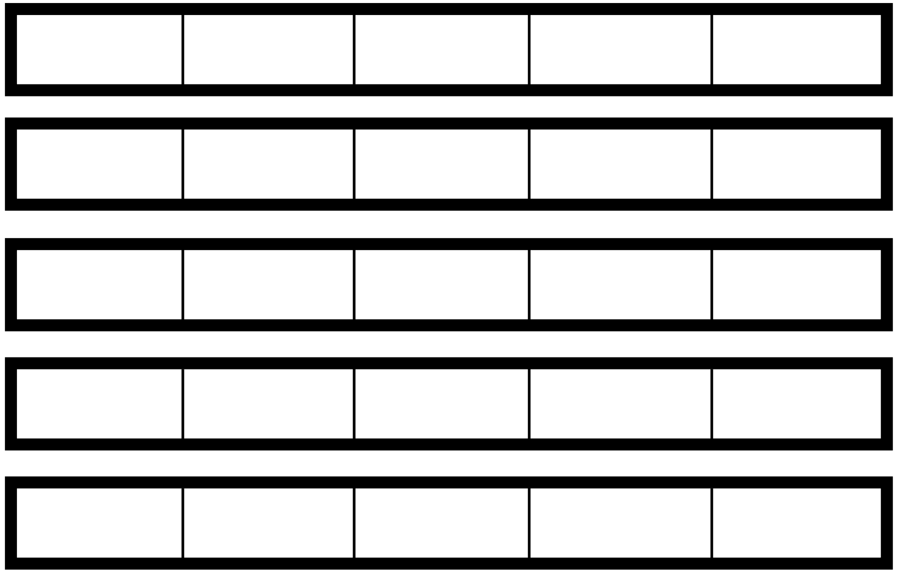


Expensive Coordination

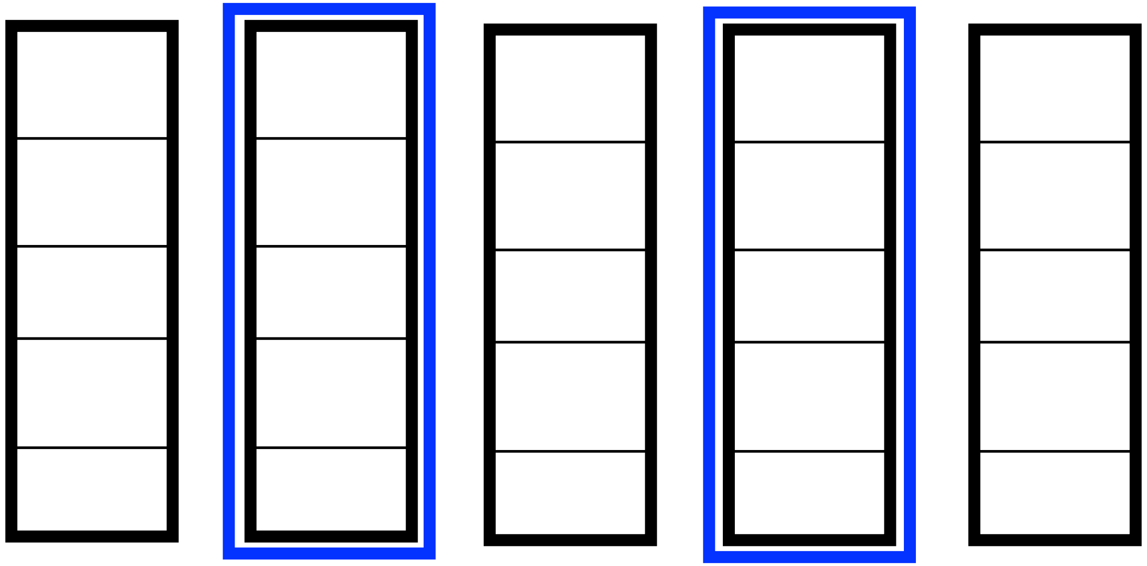
commit

Storage Formats

Row Layout



Columnar Layout

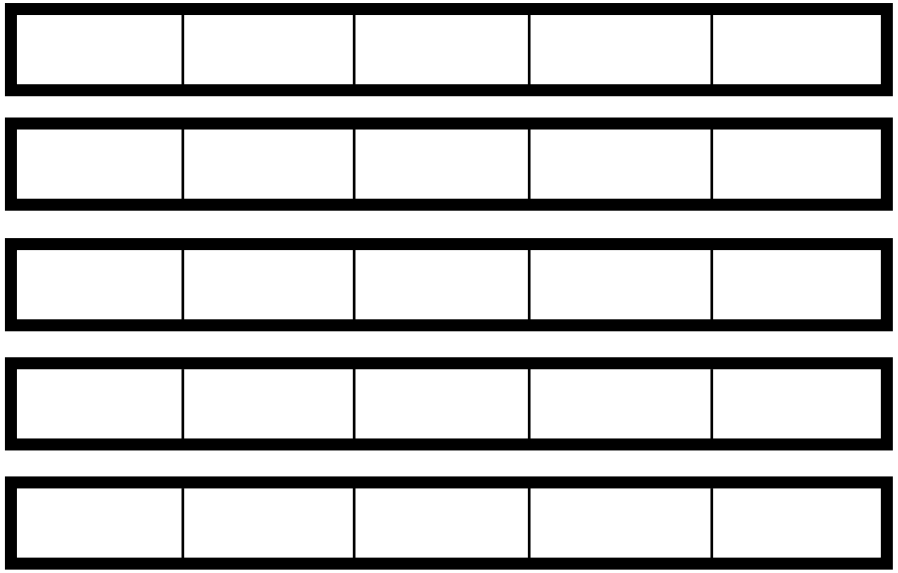


Updates

Analytics

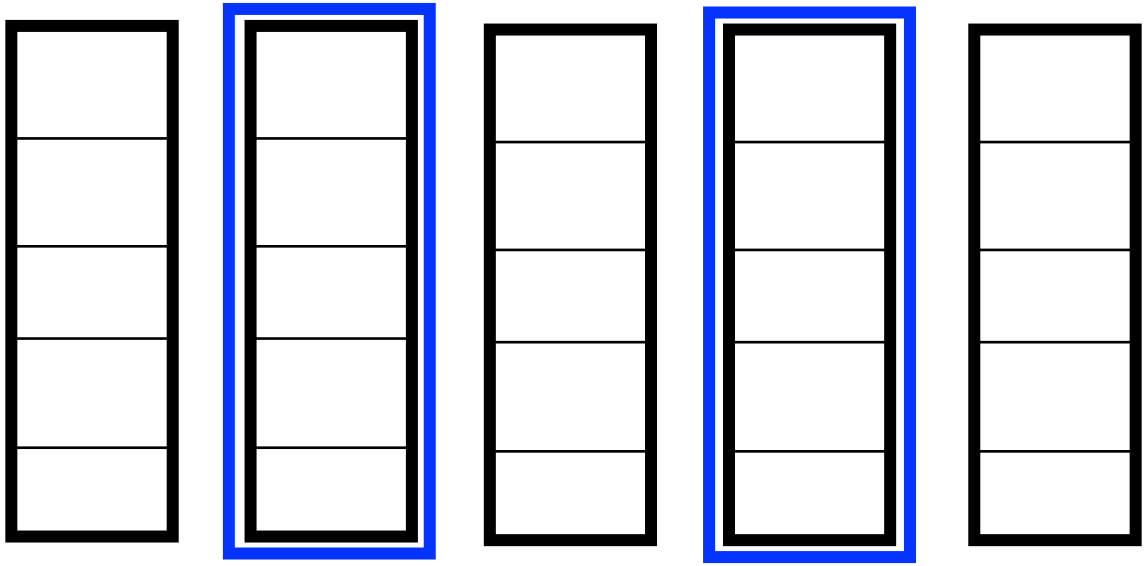
Storage Formats and HTAP

Row Layout



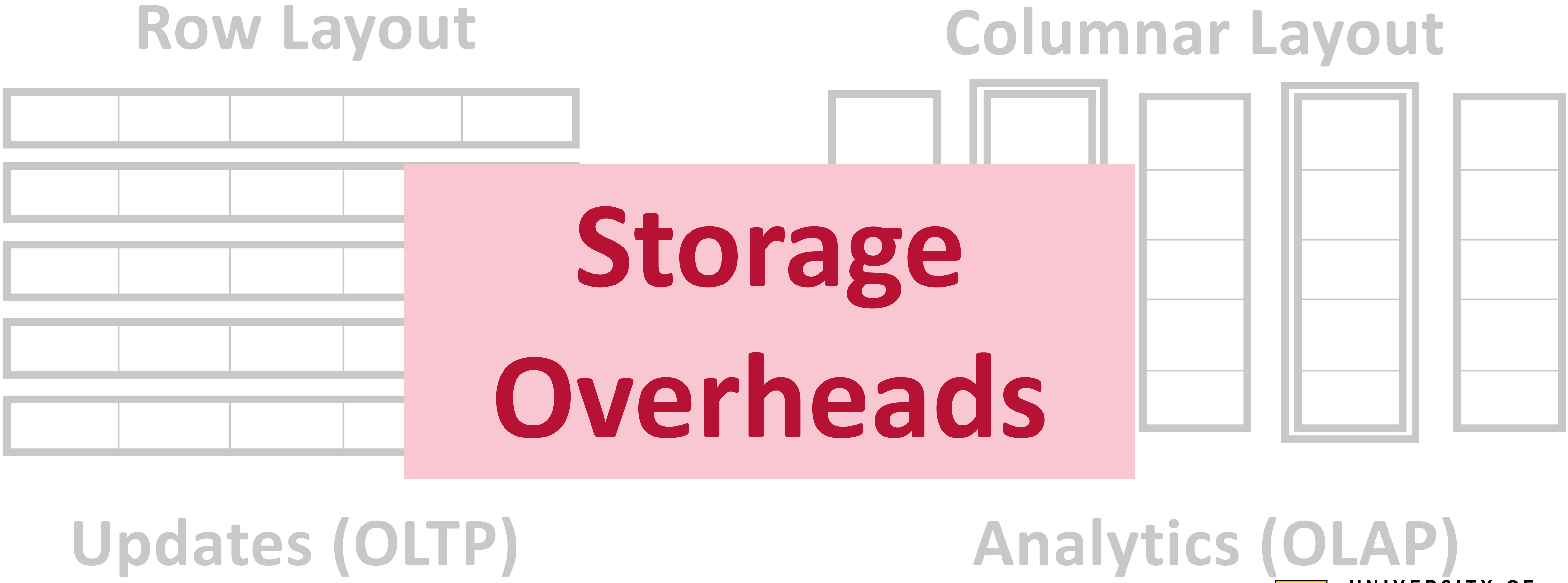
Updates (OLTP)

Columnar Layout



Analytics (OLAP)

Storage Formats and HTAP



Storage Overheads

Distributed Databases

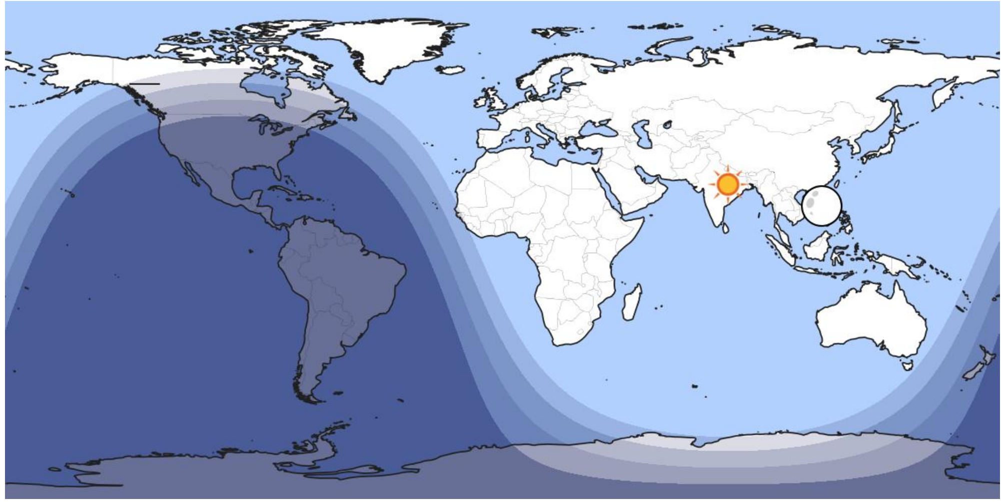
How and where to store data?

?
Replication
Partitioning
Format
?

Trade-offs dependent on workload

Distributed DBMSs must adapt

Workloads Can Change



Distributed DBMSs must adapt

Thesis Statement

Automatic **adaptation** of **how & where** data stored

Using online workload information

Improves performance of distributed DBMSs

Thesis Contributions

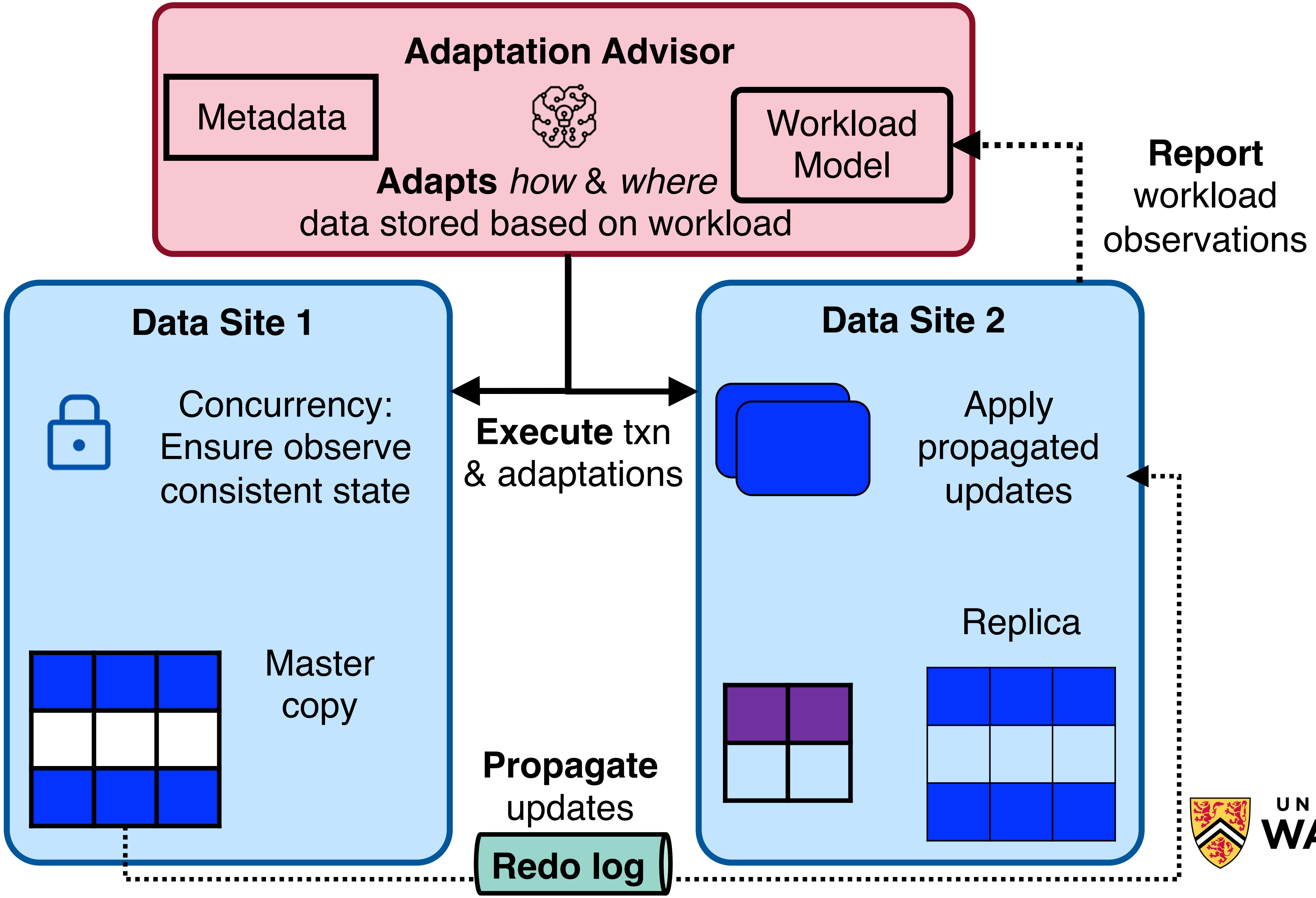
Automatic **adaptation** of **how & where** data stored

DynaMast (ICDE’20)
Dynamic transfer data mastership to reduces overhead of coordination

MorphoSys (PVLDB’20)
Automatically select physical design: partitioning, & data placement

Proteus (SIGMOD’22) (PVLDB’22)
Adapt data storage (formats & tiers) for HTAP workloads

Architecture



Adaptation Advisor

Metadata

Workload Model

Adapts *how* & *where* data stored based on workload

Report workload observations

Data Site 1

Concurrency: Ensure observe consistent state

Master copy

Execute txn & adaptations

Data Site 2

Apply propagated updates

Replica

Propagate updates

Redo log

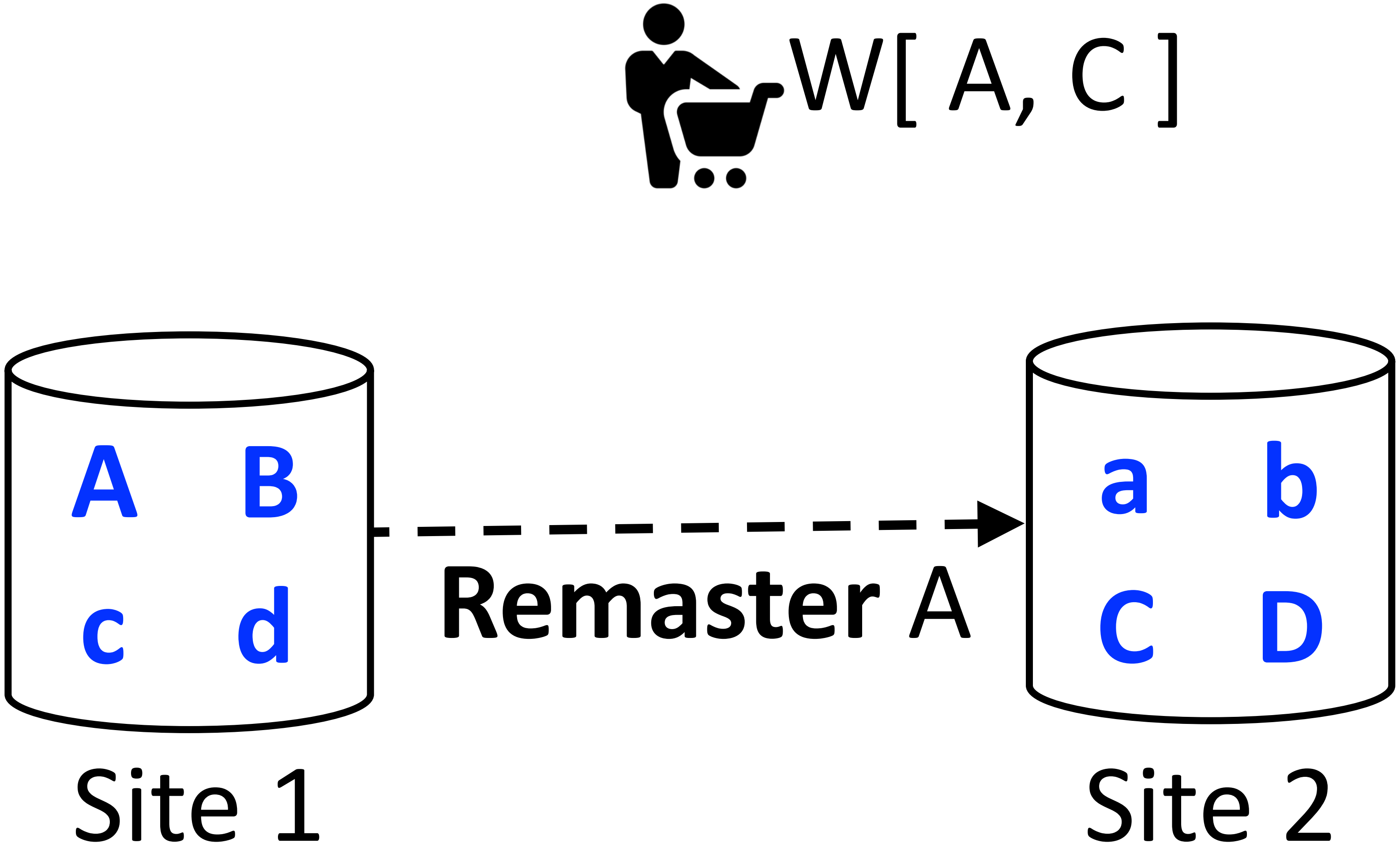
Thesis Contributions

Automatic **adaptation** of **how & where** data stored

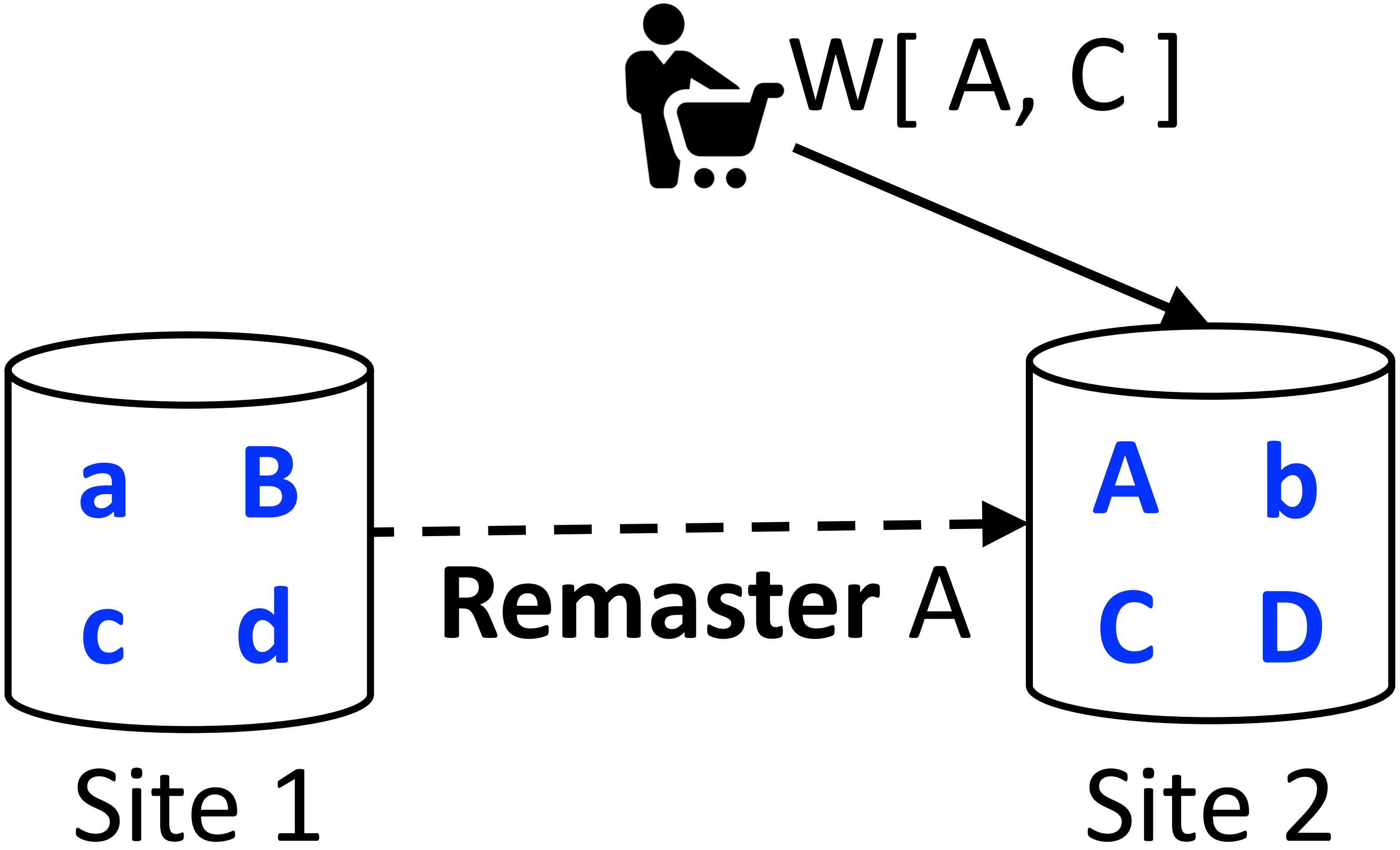
DynaMast	Dynamic transfer data mastership to reduces overhead of coordination
MorphoSys	Automatically select physical design: partitioning, & data placement
Proteus	Adapt data storage (formats & tiers) for HTAP workloads

UNIVERSITY OF WATERLOO

Dynamic Mastering

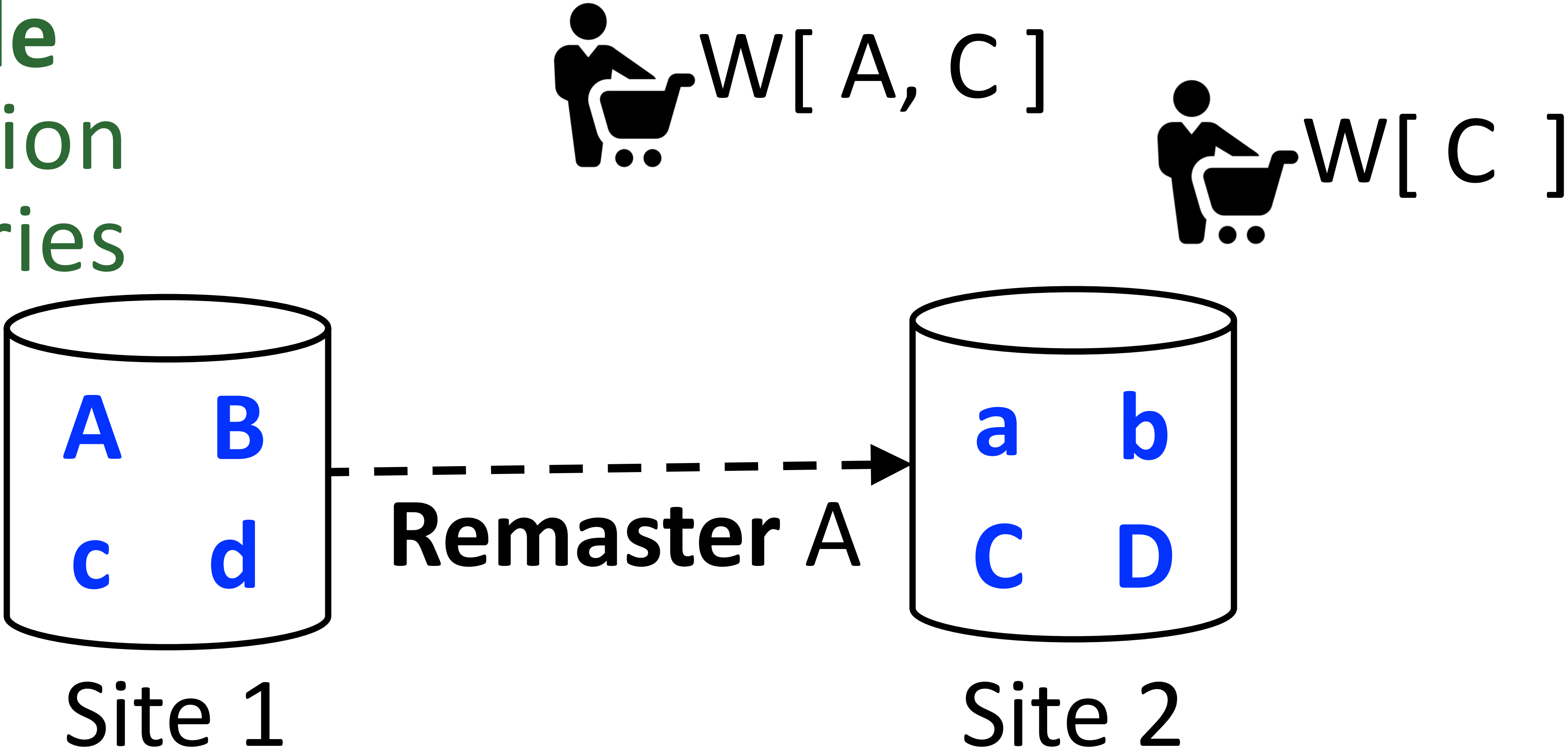


Dynamic Mastering

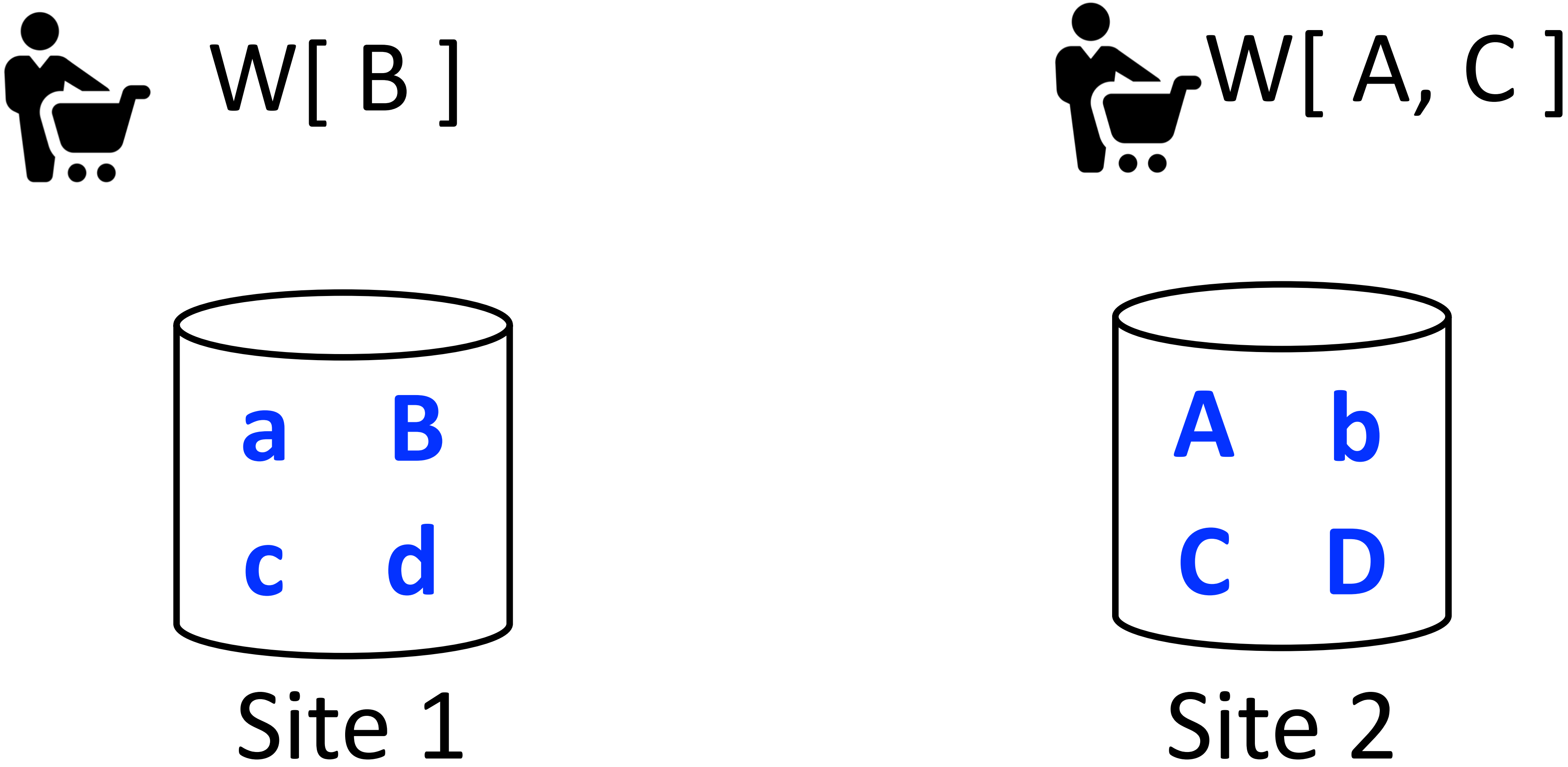


Dynamic Mastering

Outside transaction boundaries



Dynamic Mastering



W[B]

W[A, C]

a B

c d

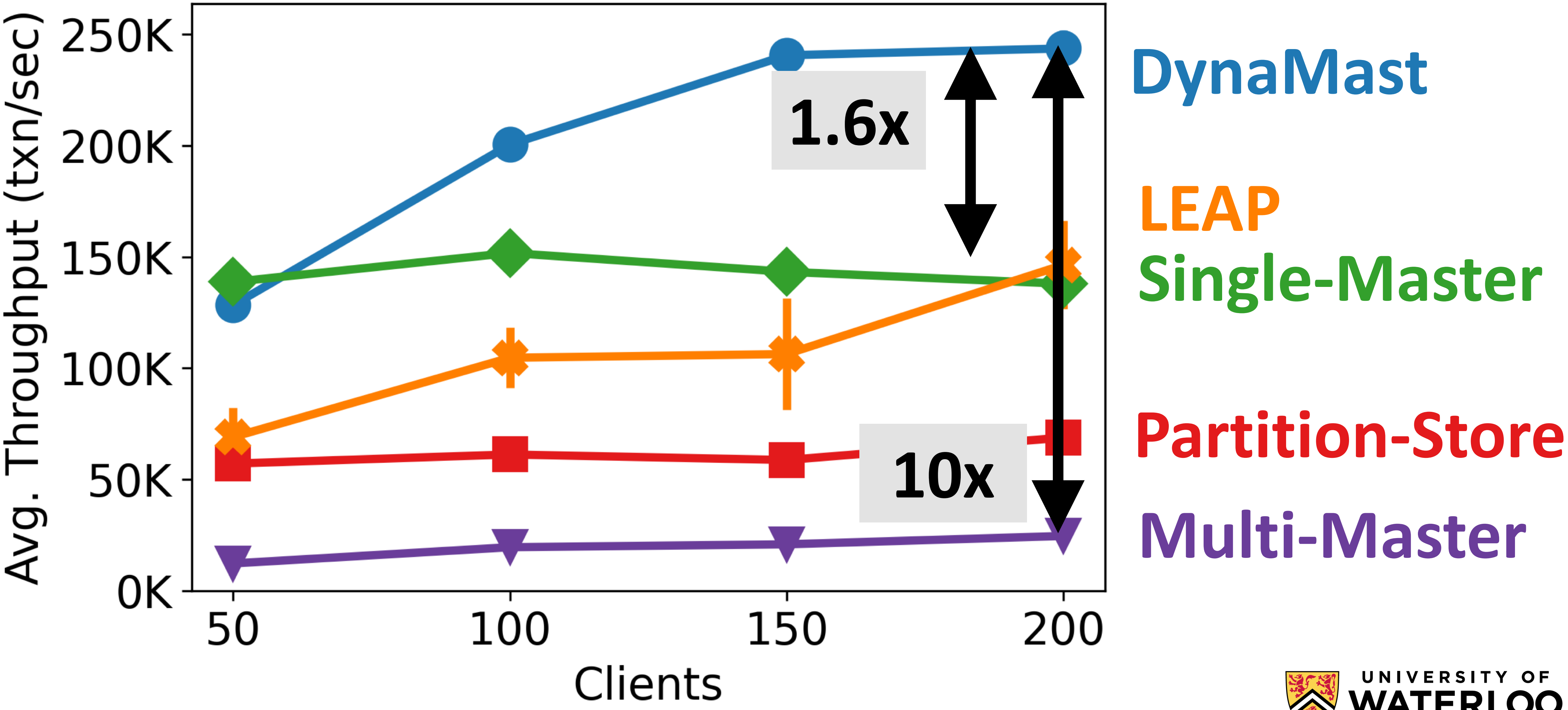
Site 1

A b

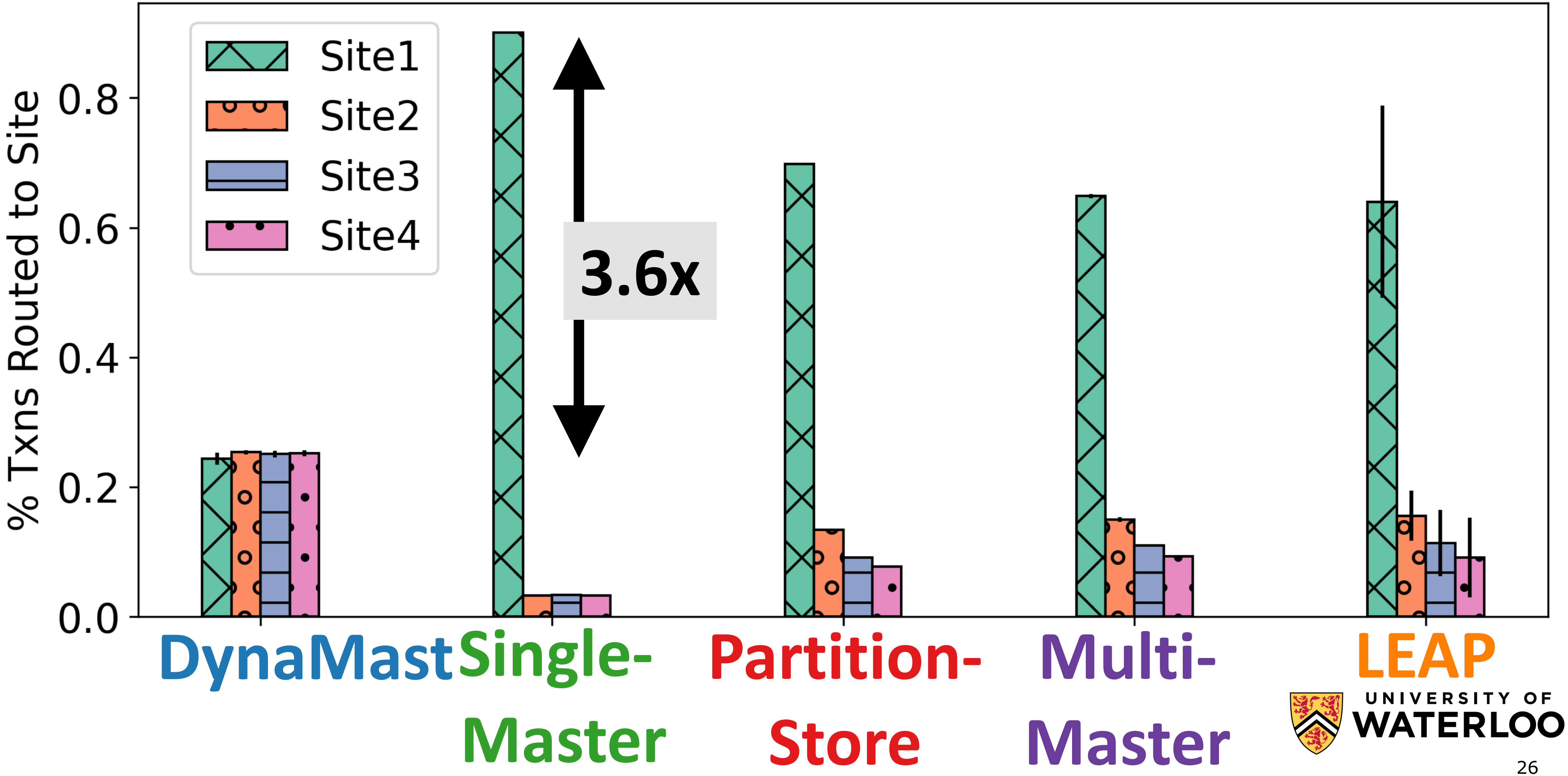
C D

Site 2

YCSB with Skew - Throughput



YCSB with Skew - Routing



UNIVERSITY OF WATERLOO

Thesis Contributions

Automatic **adaptation** of **how & where** data stored

DynaMast Dynamic transfer data mastership to reduces overhead of coordination

MorphoSys Automatically select physical design: partitioning, & data placement

Proteus Adapt data storage (formats & tiers) for HTAP workloads

UNIVERSITY OF WATERLOO

Distributed DBMS Physical Design

For each **data item**

- Where is the **master**?
- What nodes **replicate** it?
- How is it **grouped (partitioned)** with other data items?

MorphoSys Physical Design Change Operations

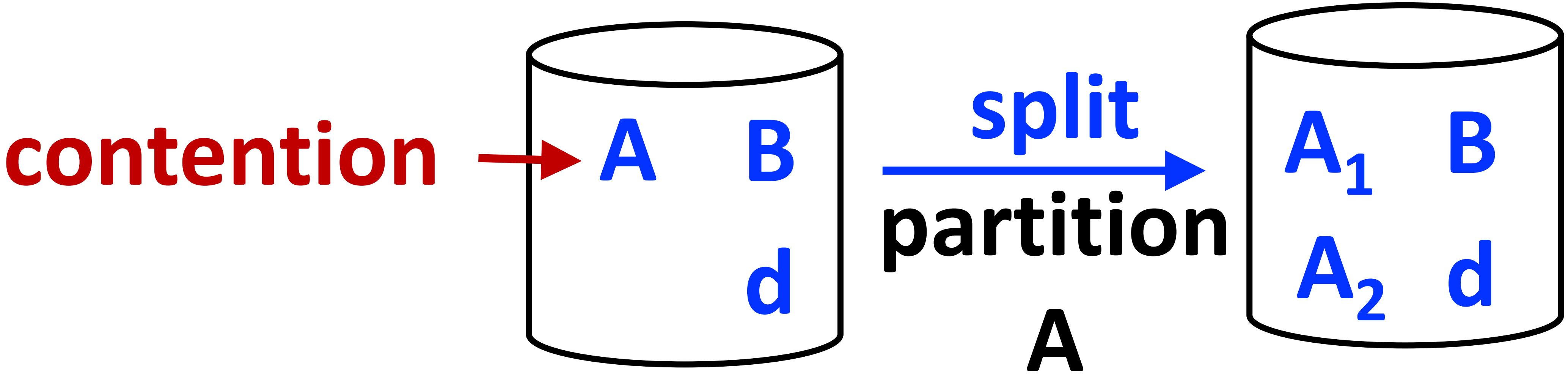
Add or **remove replica** of a **partition**

Remaster a **partition**

Split or **merge** **partition(s)**

Making design decisions

Learned cost model quantifies
design change effects



contention

A B d

split

partition A

A_1 B A_2 d

Design change cost < Expected Benefit

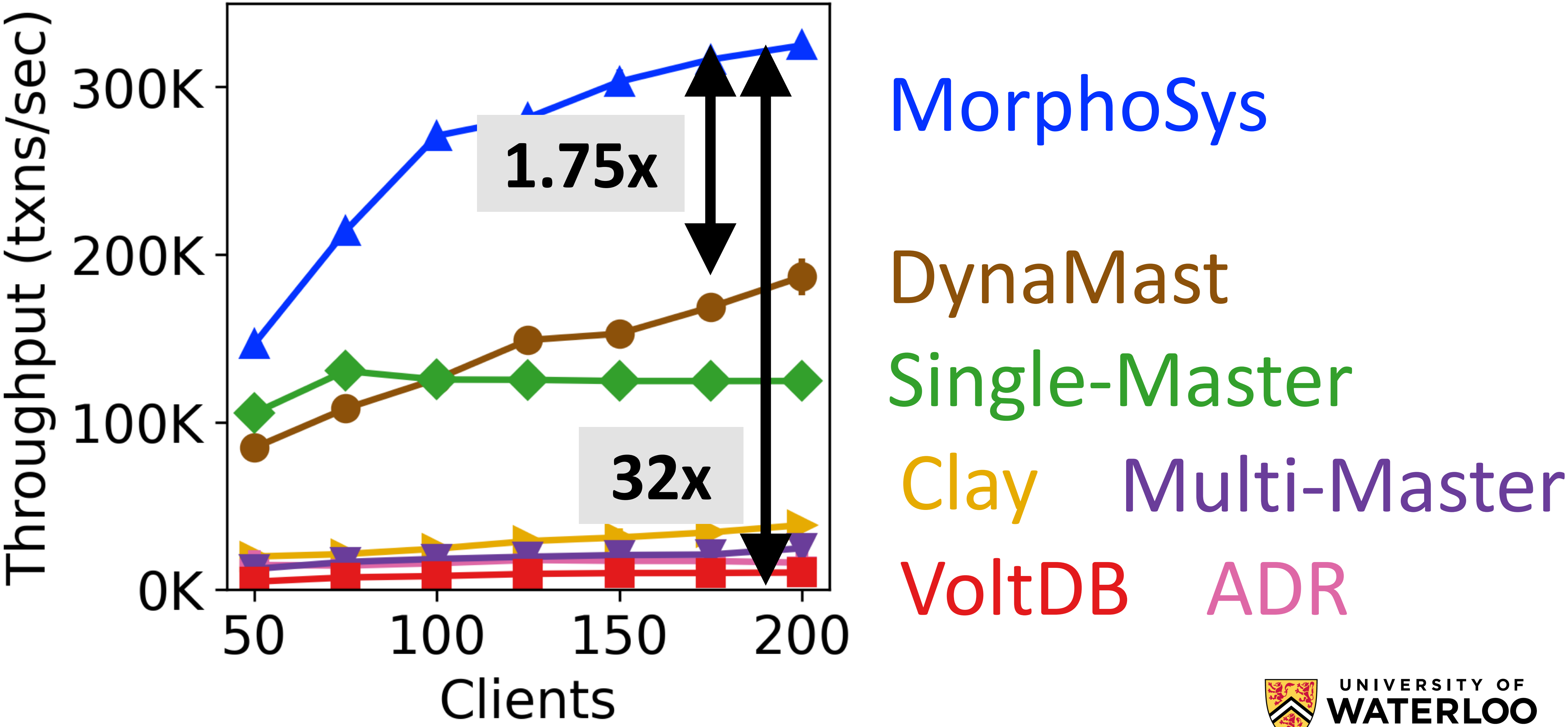
Physical design cost model

Design change cost < Expected benefit

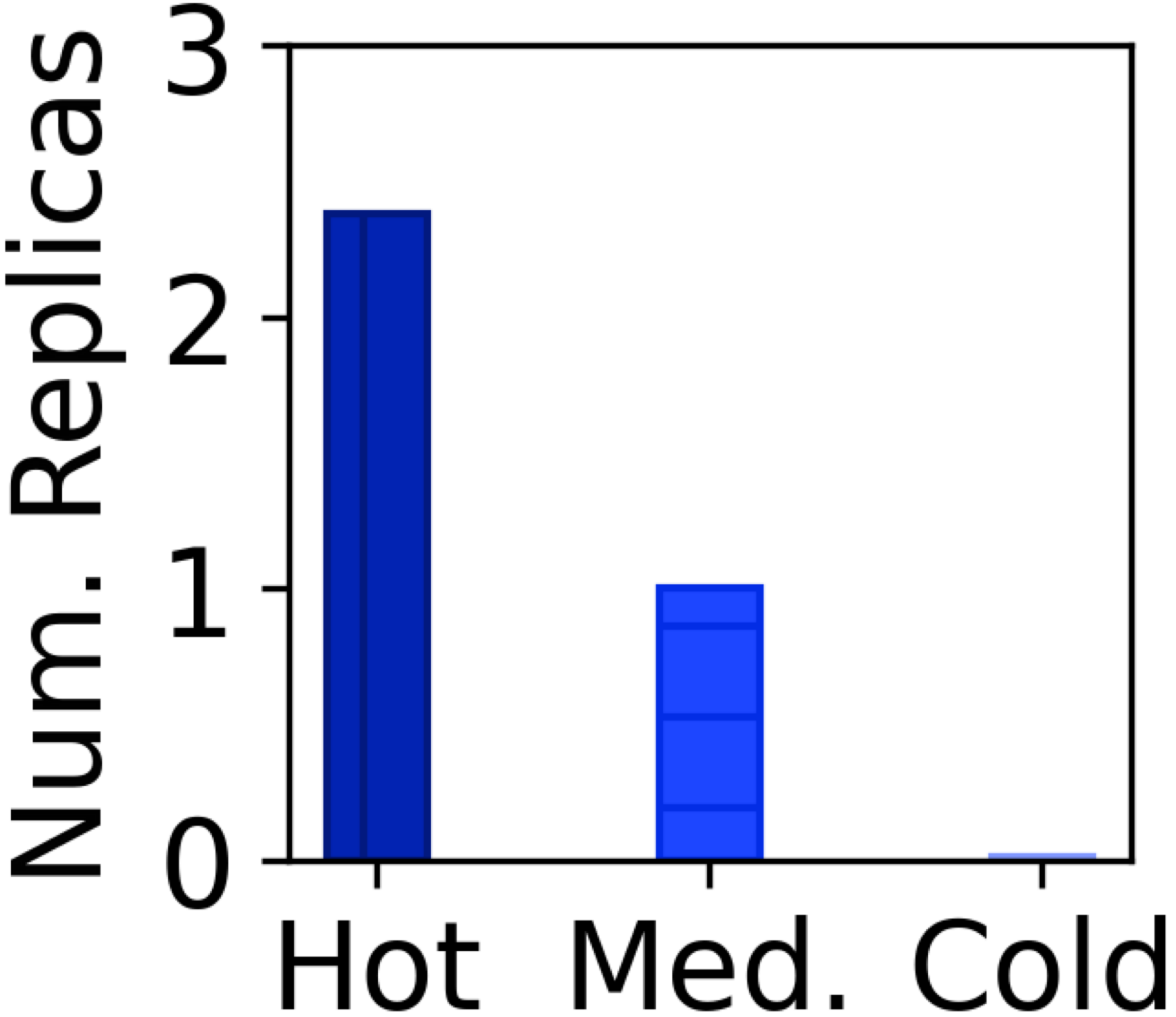
Decompose operators into **key costs**

Predict benefit based on workload history

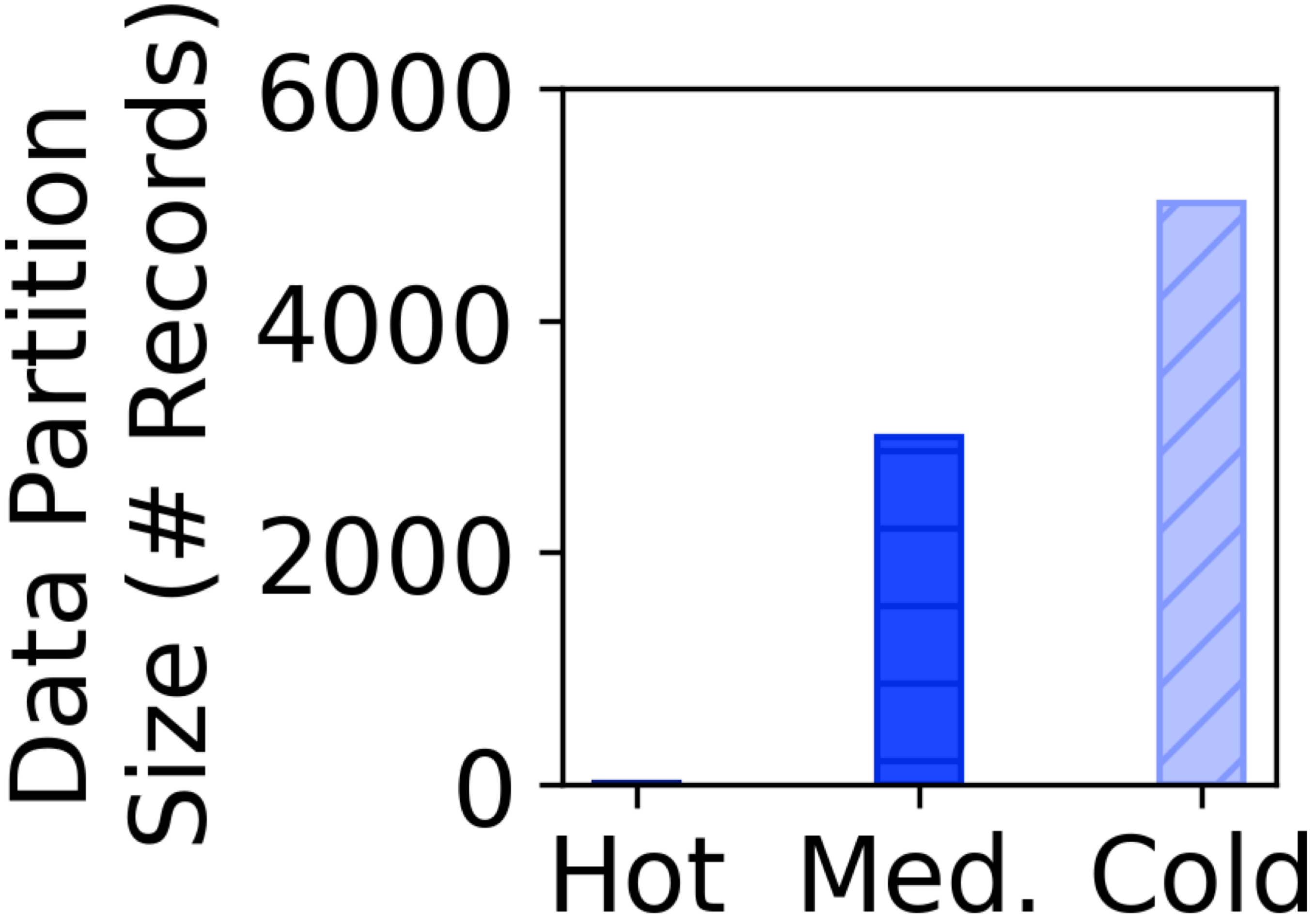
Skewed YCSB - Throughput



Number of Replicas



Partition Sizes



Thesis Contributions

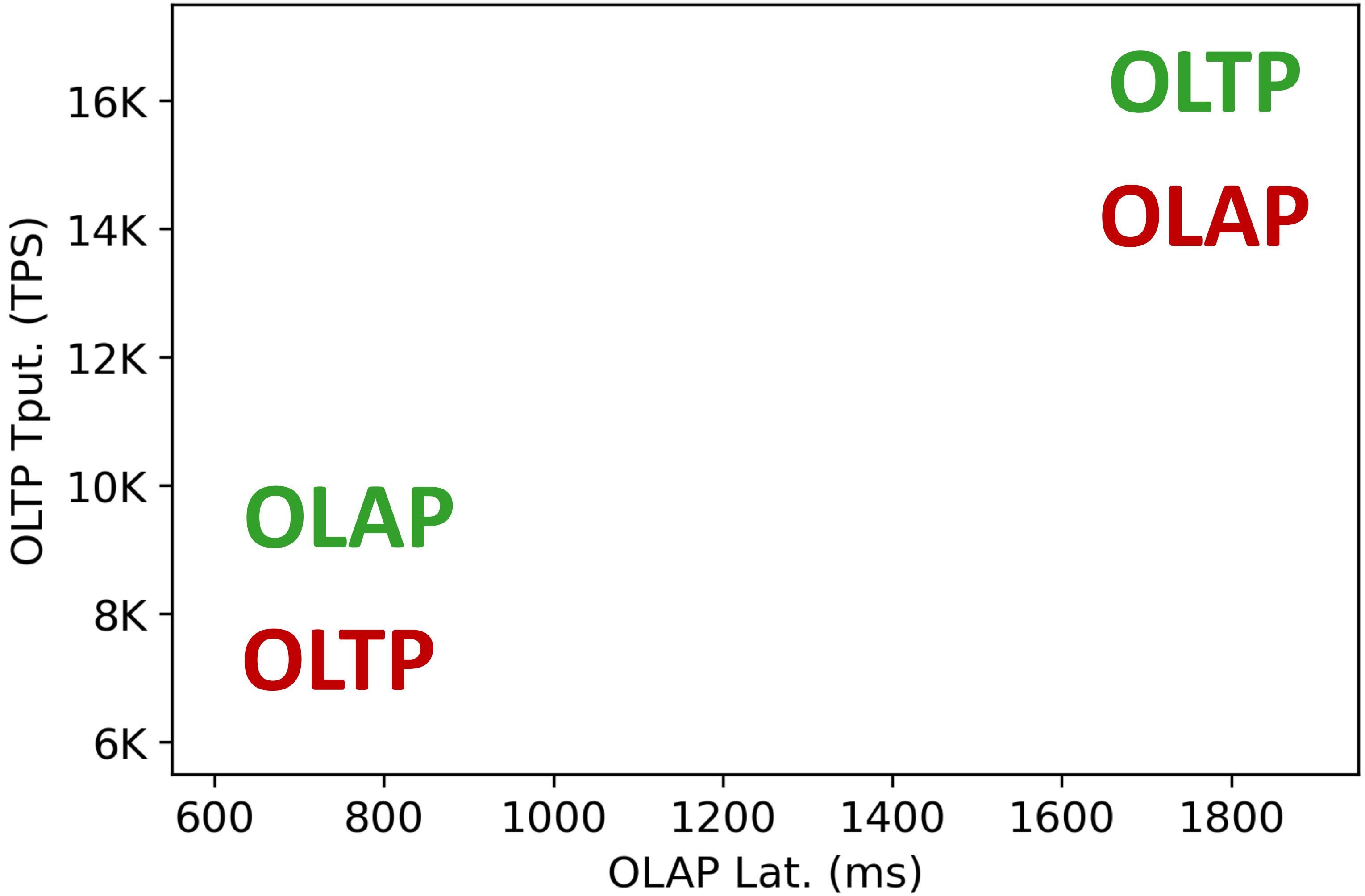
Automatic **adaptation** of **how & where** data stored

DynaMast Dynamic transfer data mastership to reduces overhead of coordination

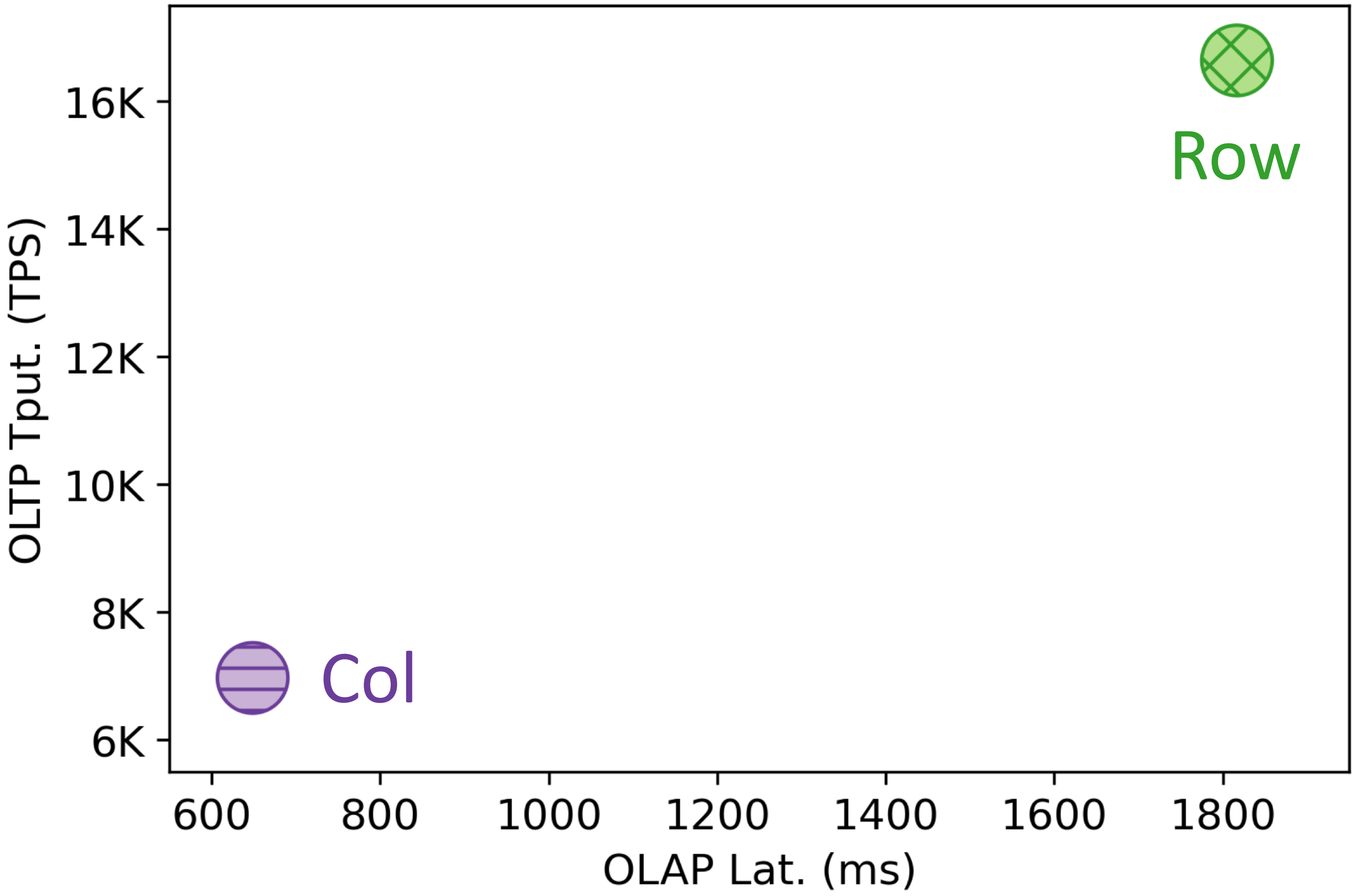
MorphoSys Automatically select physical design: partitioning, & data placement

Proteus Adapt data storage (formats & tiers) for HTAP workloads

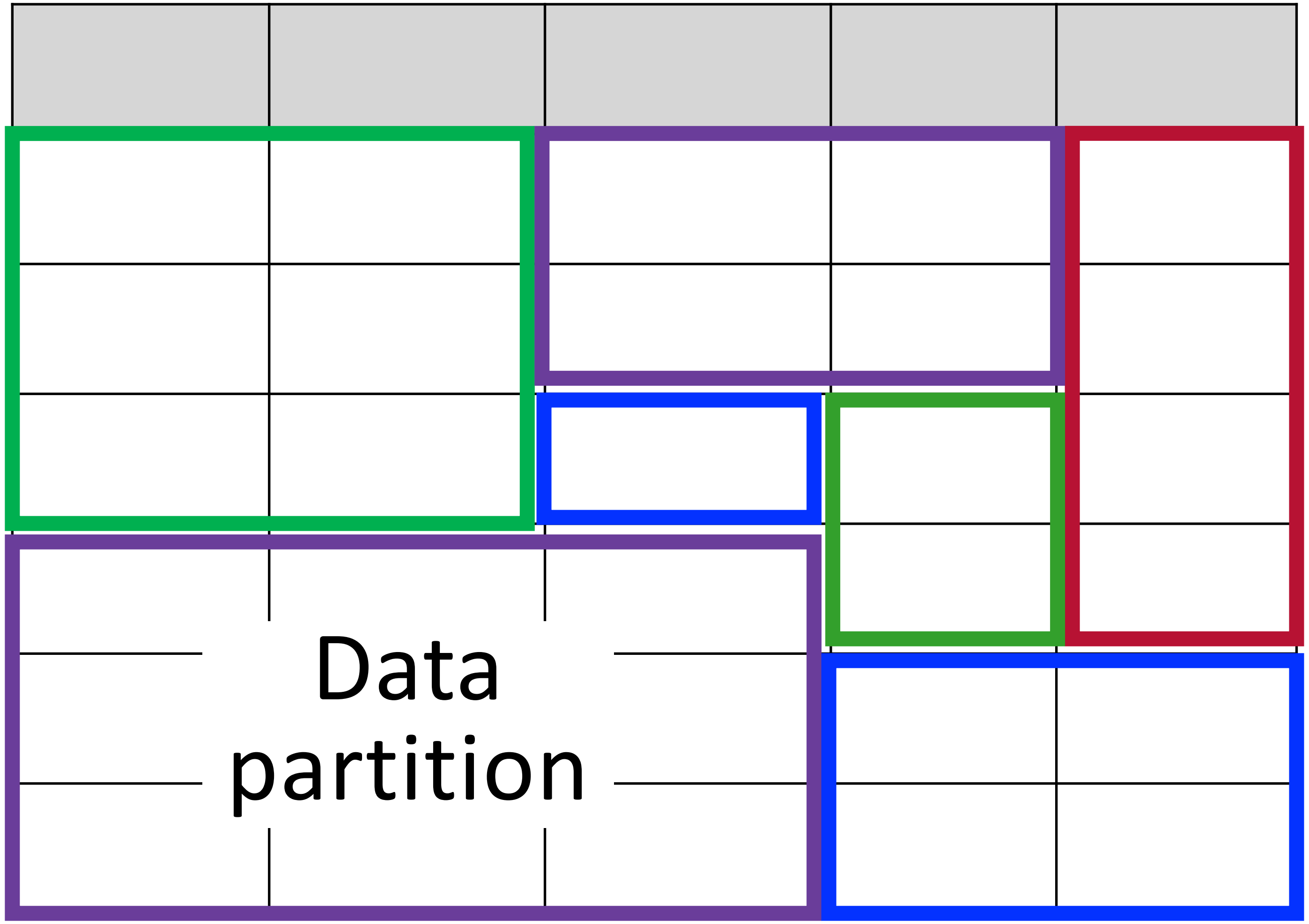
Performance Trade-Off



Performance Trade-Off



Proteus Decisions



Storage layout

Master/replica(s)

Txn execution

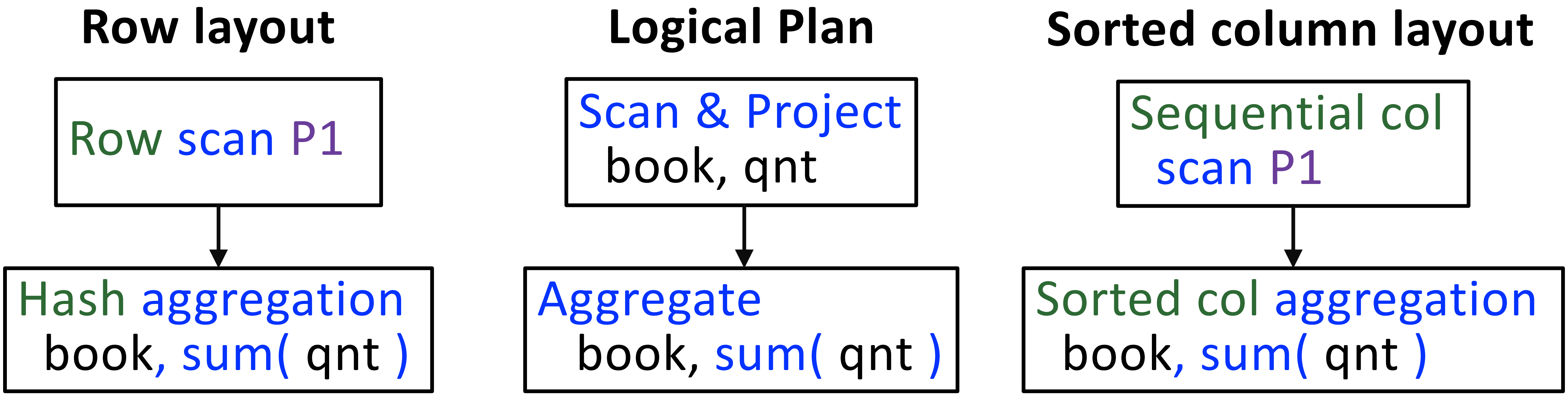
How to **partition?**

When & what to change

Transactions in Proteus

Breakdown transaction into **physical** operators

SELECT book, **SUM**(qnt) **GROUP BY** book



Row layout

Row scan P1

Hash aggregation book, sum(qnt)

Logical Plan

Scan & Project book, qnt

Aggregate book, sum(qnt)

Sorted column layout

Sequential col scan P1

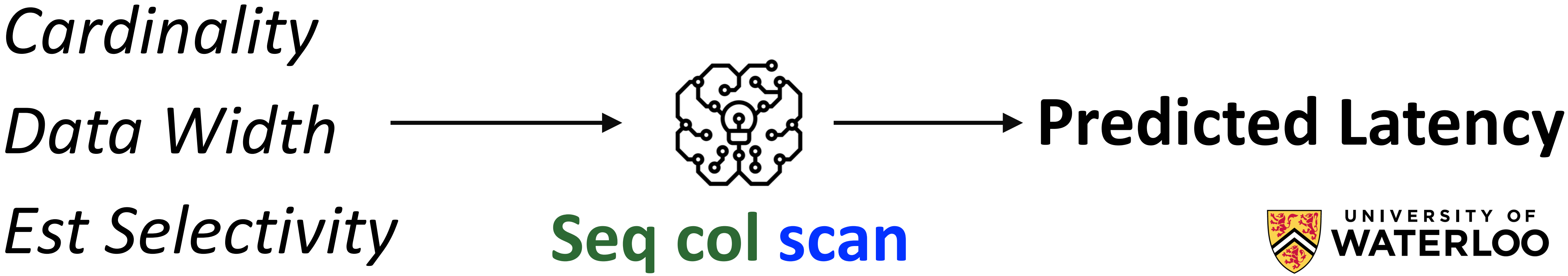
Sorted col aggregation book, sum(qnt)

Storage-Aware Operators

Per layout implementation of operators

Operate **directly** over columnar, sorted, compressed data

Predict physical operator latency

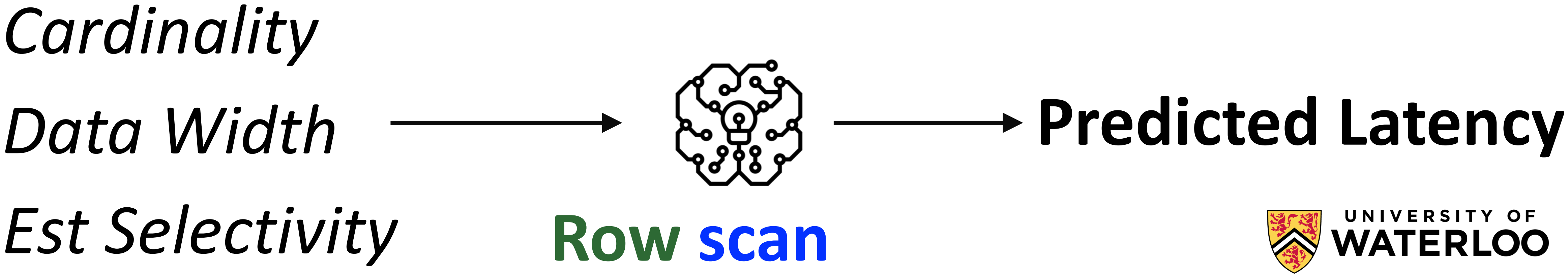


Storage-Aware Operators

Per layout implementation of operators

Operate **directly** over columnar, sorted, compressed data

Predict physical operator latency



Cardinality

Data Width

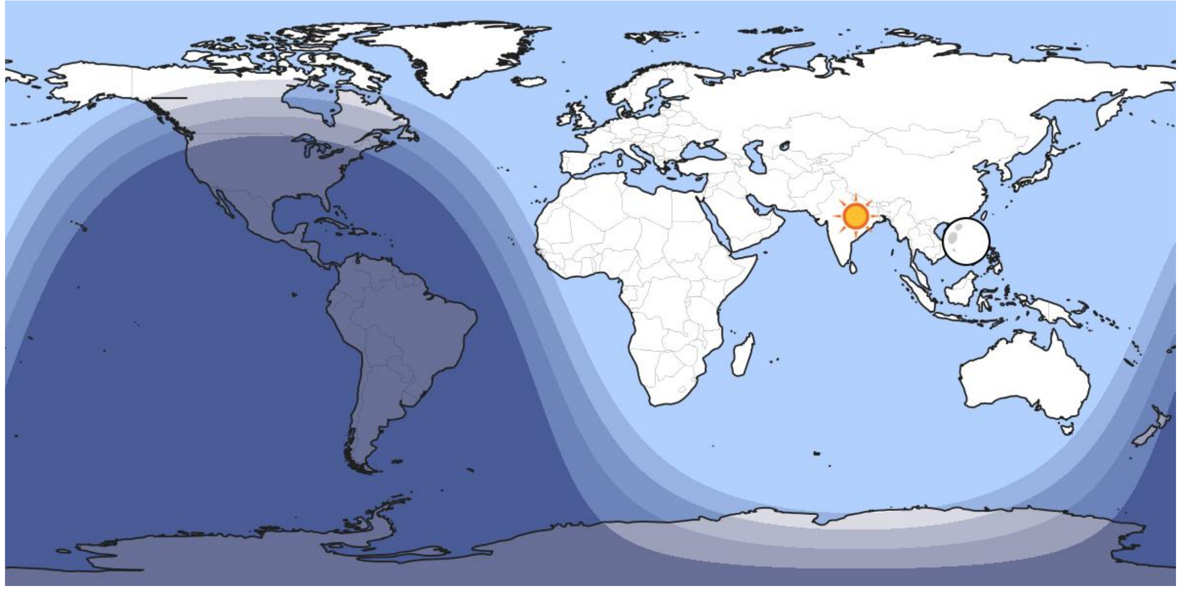
Est Selectivity

Row scan

Predicted Latency

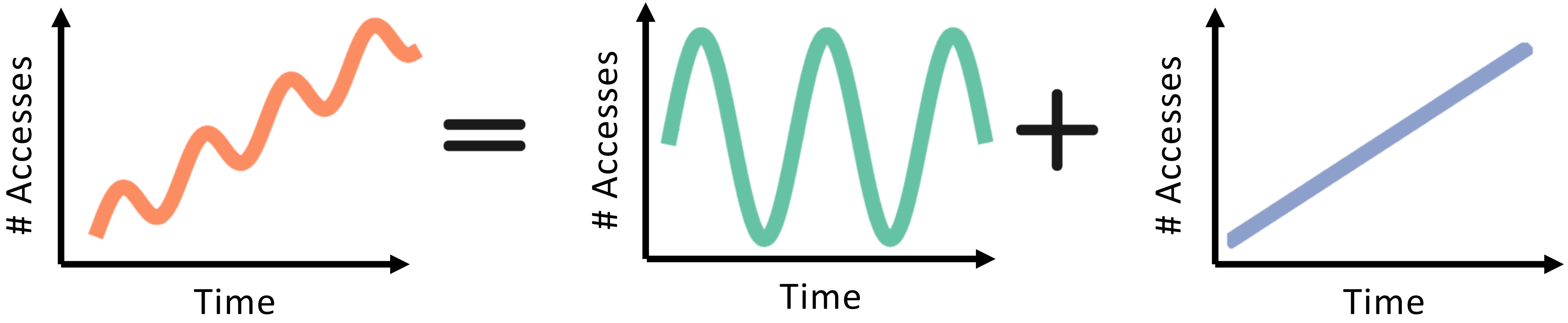
Likelihood of a Transaction

Data **accesses** to **storage** often follow **predictable** pattern



Likelihood of a Transaction

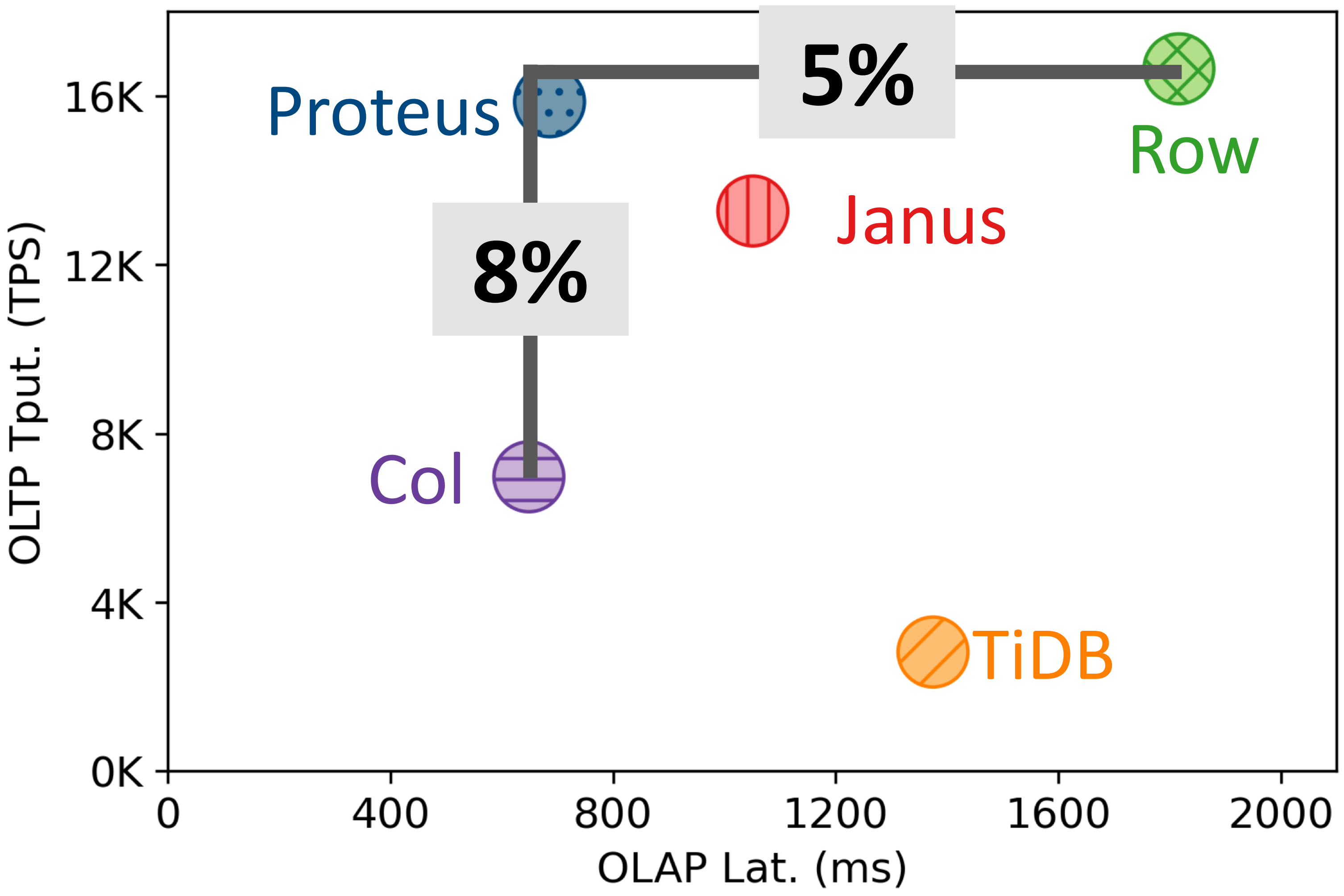
Data **accesses** to **storage**
often follow **predictable** pattern



Periodicity

Trend

CH BenCHmark



Distributed DBMSs are widely used

Distributed DBMSs must adapt

Adaptation of **how** & **where** data stored **improves performance**

DynaMast **MorphoSys** **Proteus**

UNIVERSITY OF WATERLOO

Extra Slides