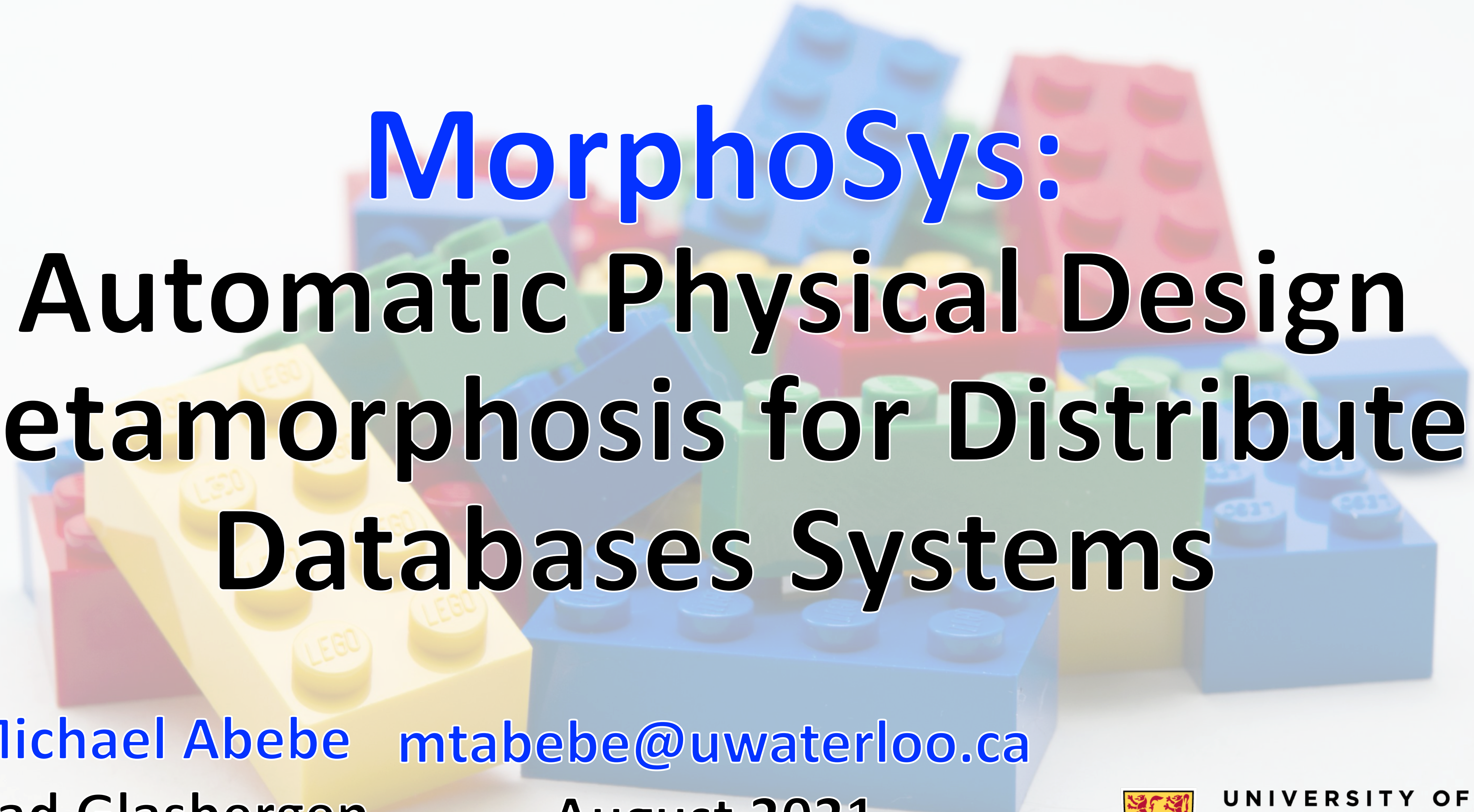


MorphoSys: Automatic Physical Design Metamorphosis for Distributed Databases Systems

Michael Abebe mtabebe@uwaterloo.ca

Brad Glasbergen

Khuzaima Daudjee

UNIVERSITY OF WATERLOO

August 2021

tiny.cc/morphosys

Distributed Databases

How to distribute data?

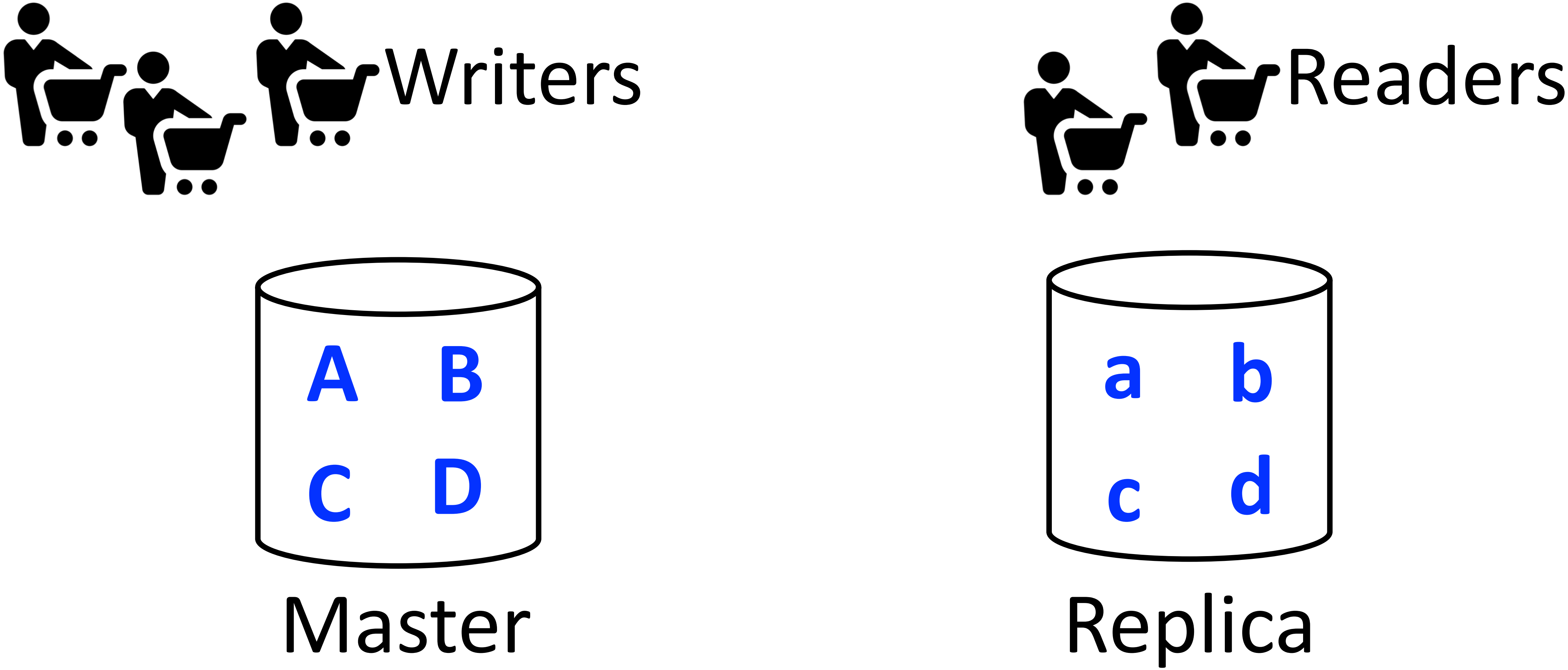
?

Replication

Partitioning

?

Database Replication



Writers

Readers

A B

C D

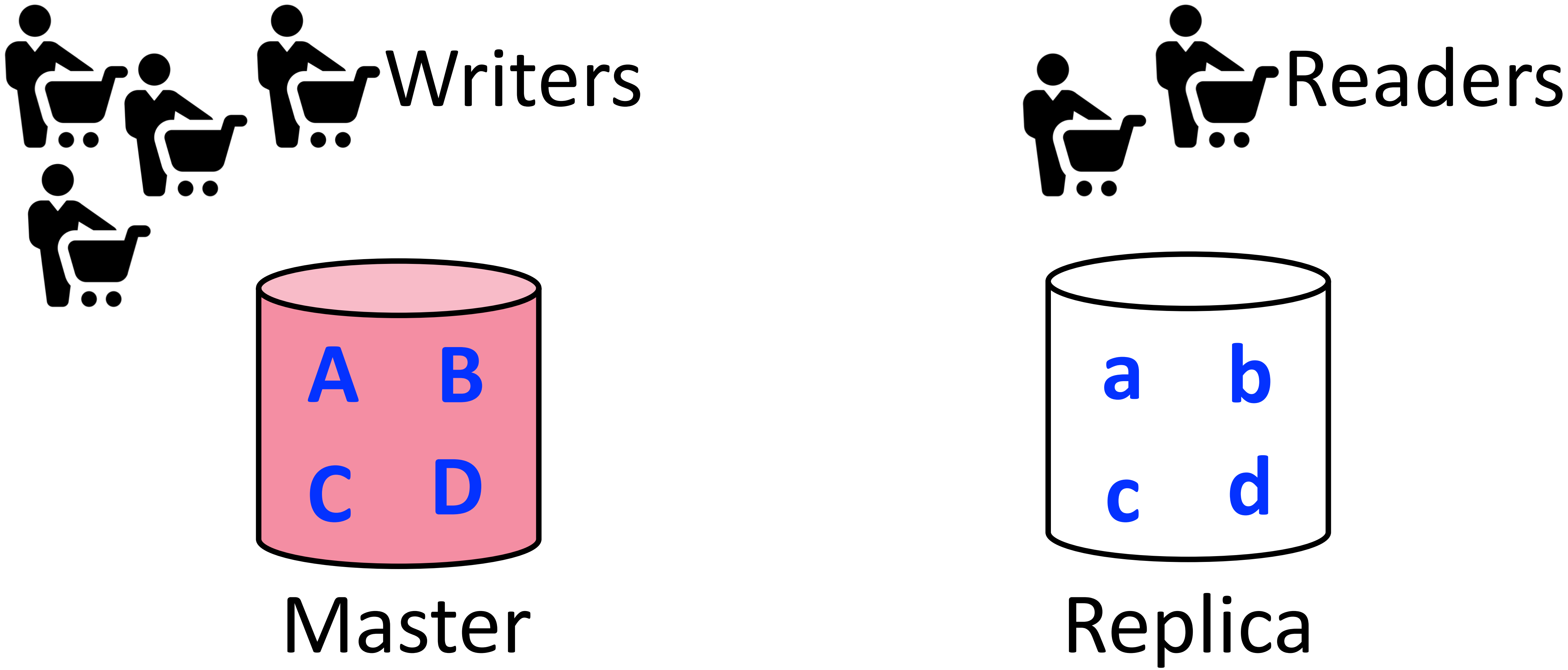
a b

c d

Master

Replica

Database Replication



Writers

Readers

A B

C D

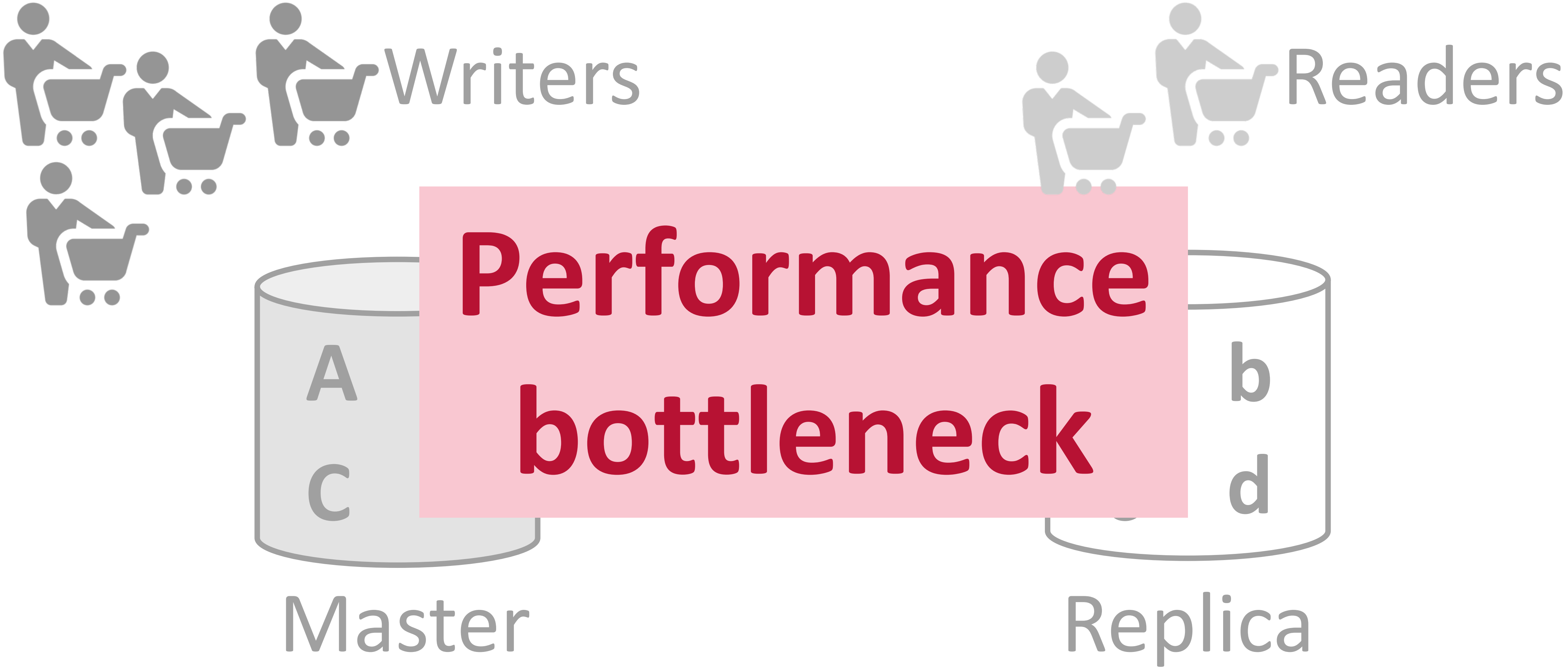
a b

c d

Master

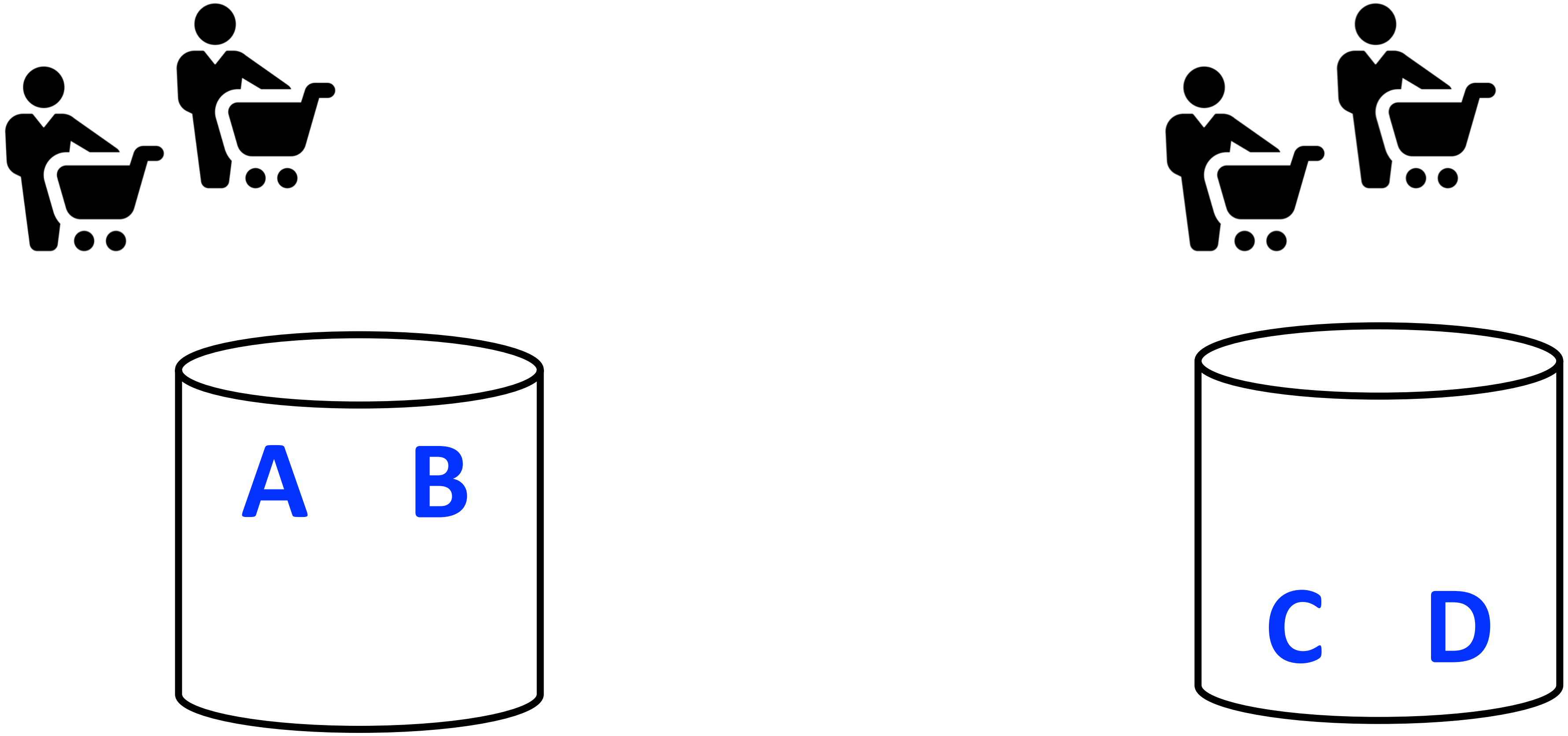
Replica

Database Replication

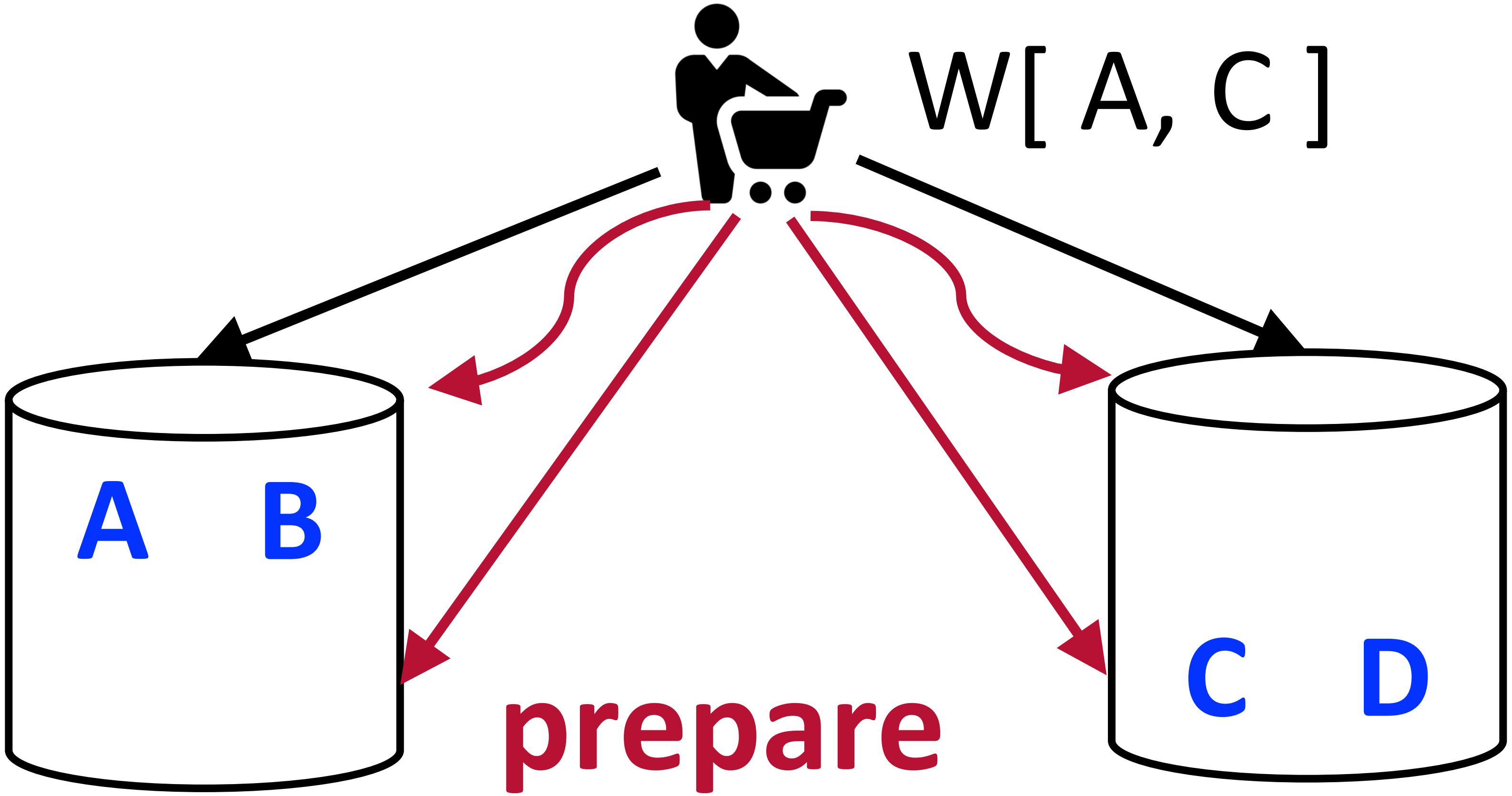


Performance bottleneck

Partitioned Databases



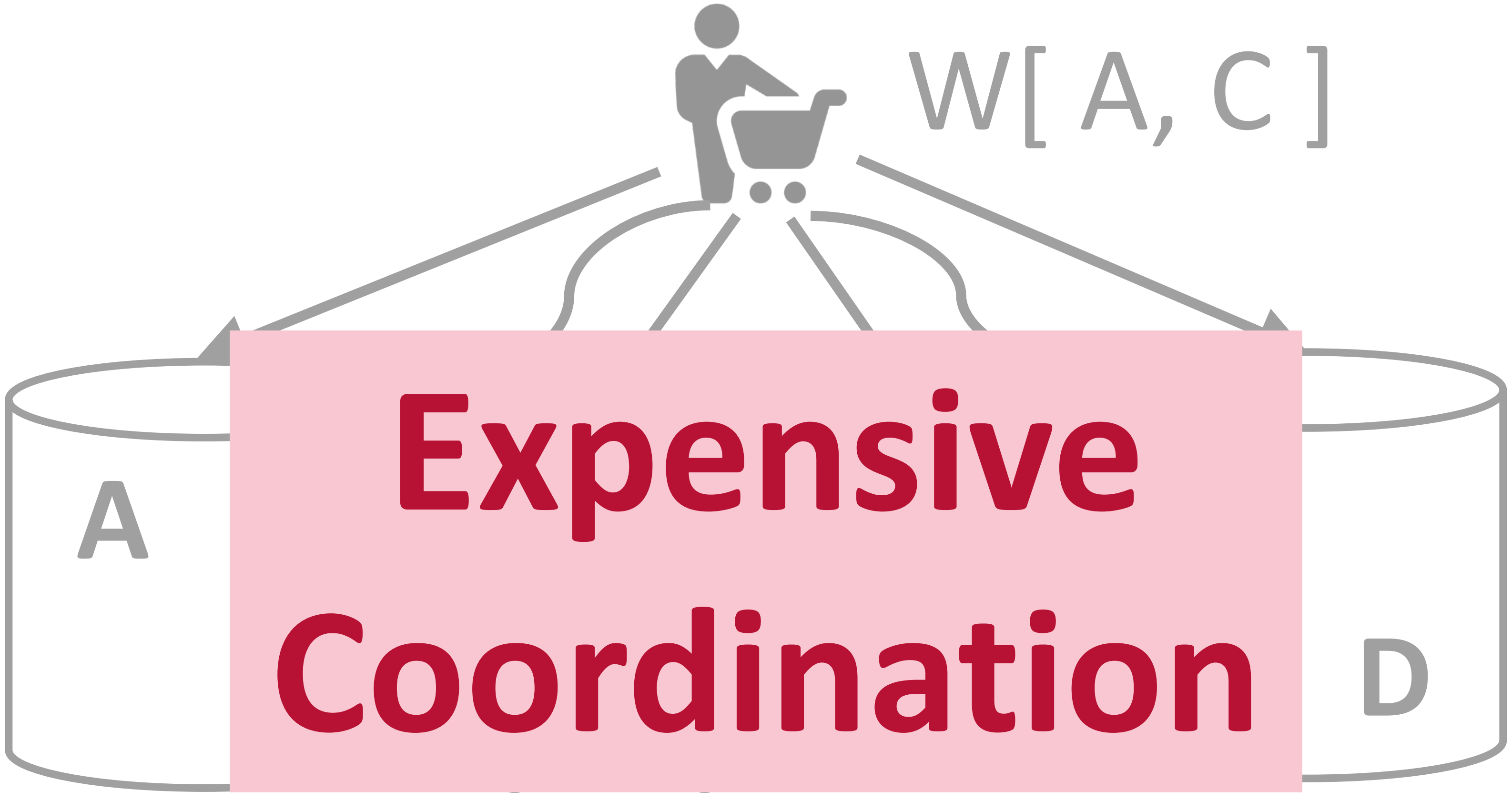
Partitioned Databases



prepare

commit

Partitioned Databases



Expensive Coordination

commit

Distributed Databases

How to distribute data?

Replication

Partitioning

Distributed Database Physical Design

Distributed Database Physical Design

For each **data item**

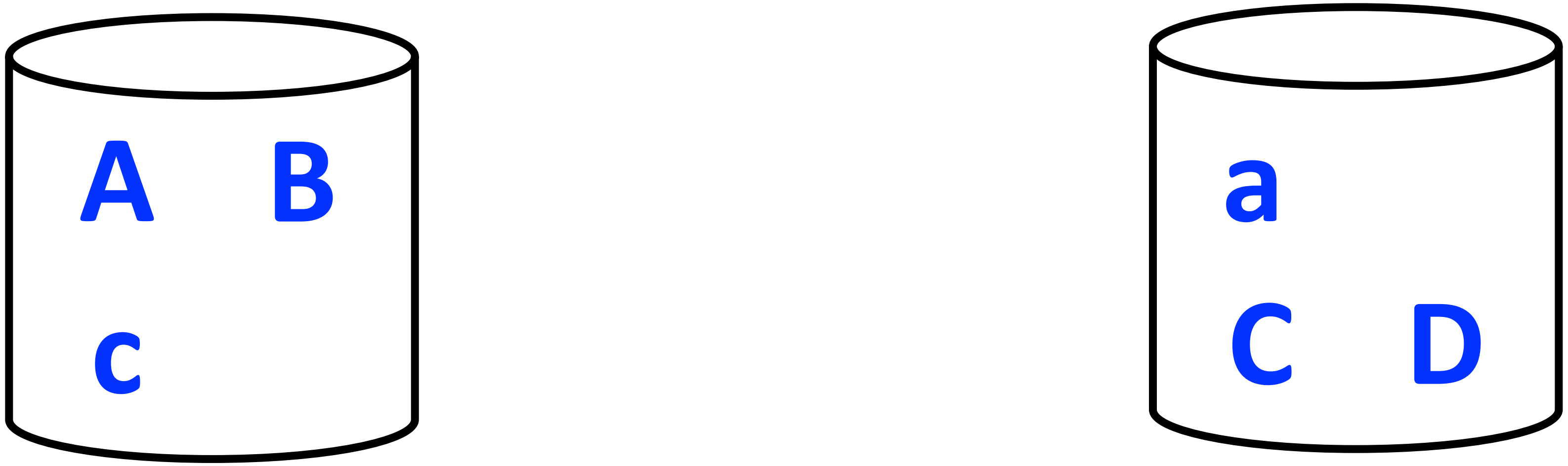
Where is the **master**?

What nodes **replicate** it?

How is it **grouped (partitioned)** with other data items?

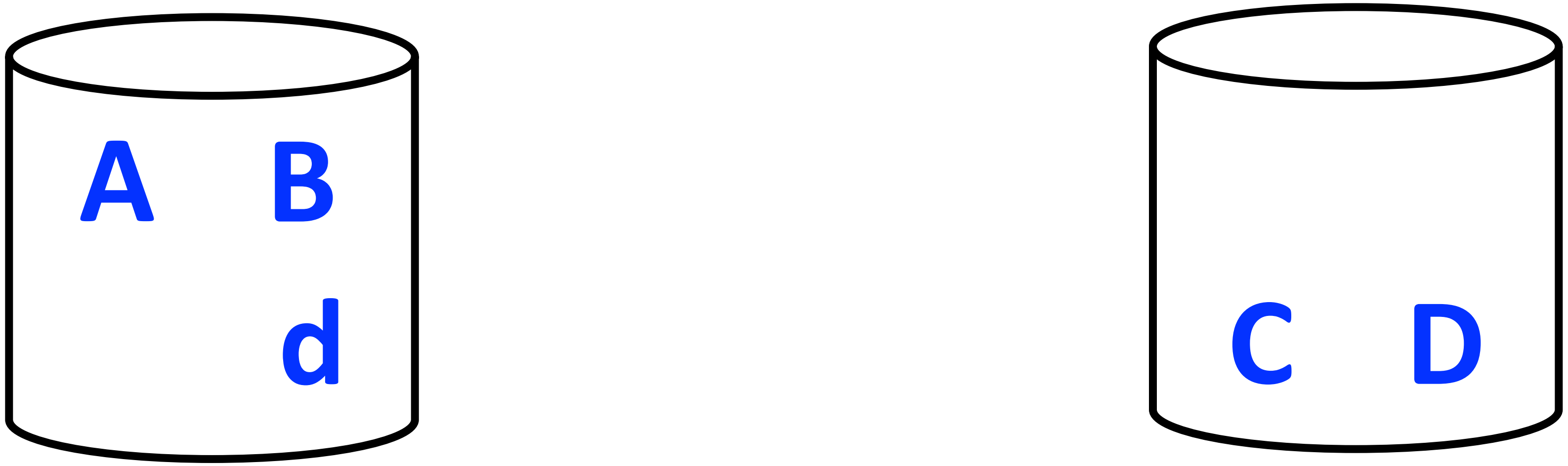
Physical designs

Any combination of master data placement, replication, & data partitioning



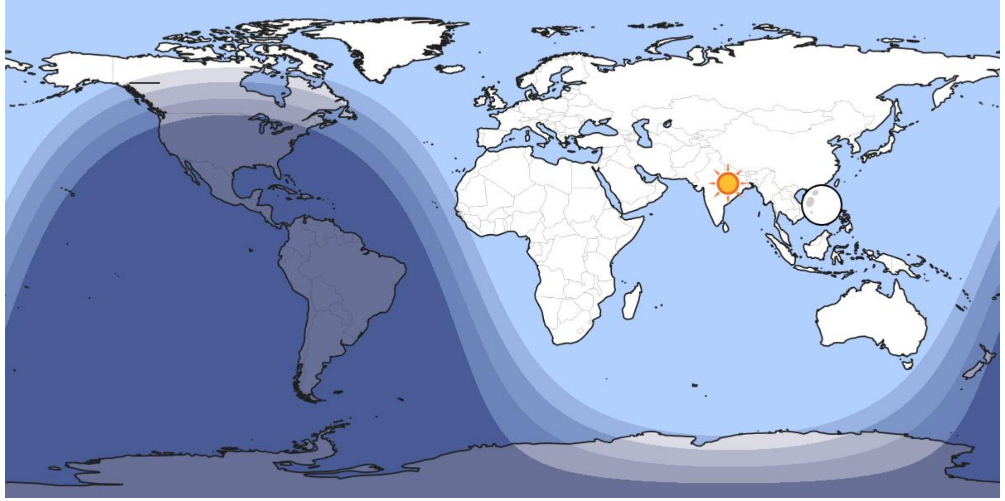
Physical designs

Any combination of master data placement, replication, & data partitioning



Which physical design?

Traditionally: **offline** workload knowledge



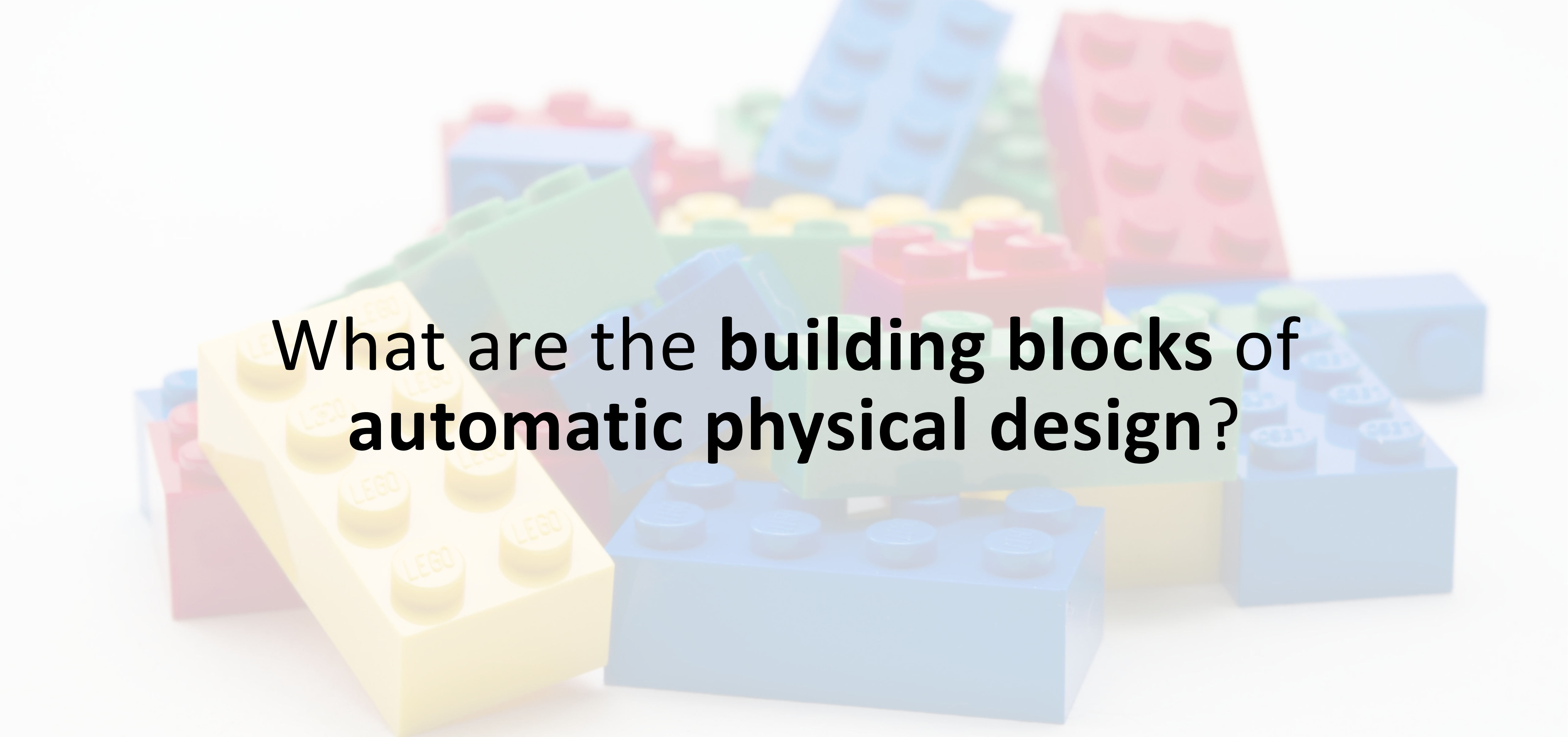
Physical design should change with workload

MorphoSys

Automatically chooses a physical design

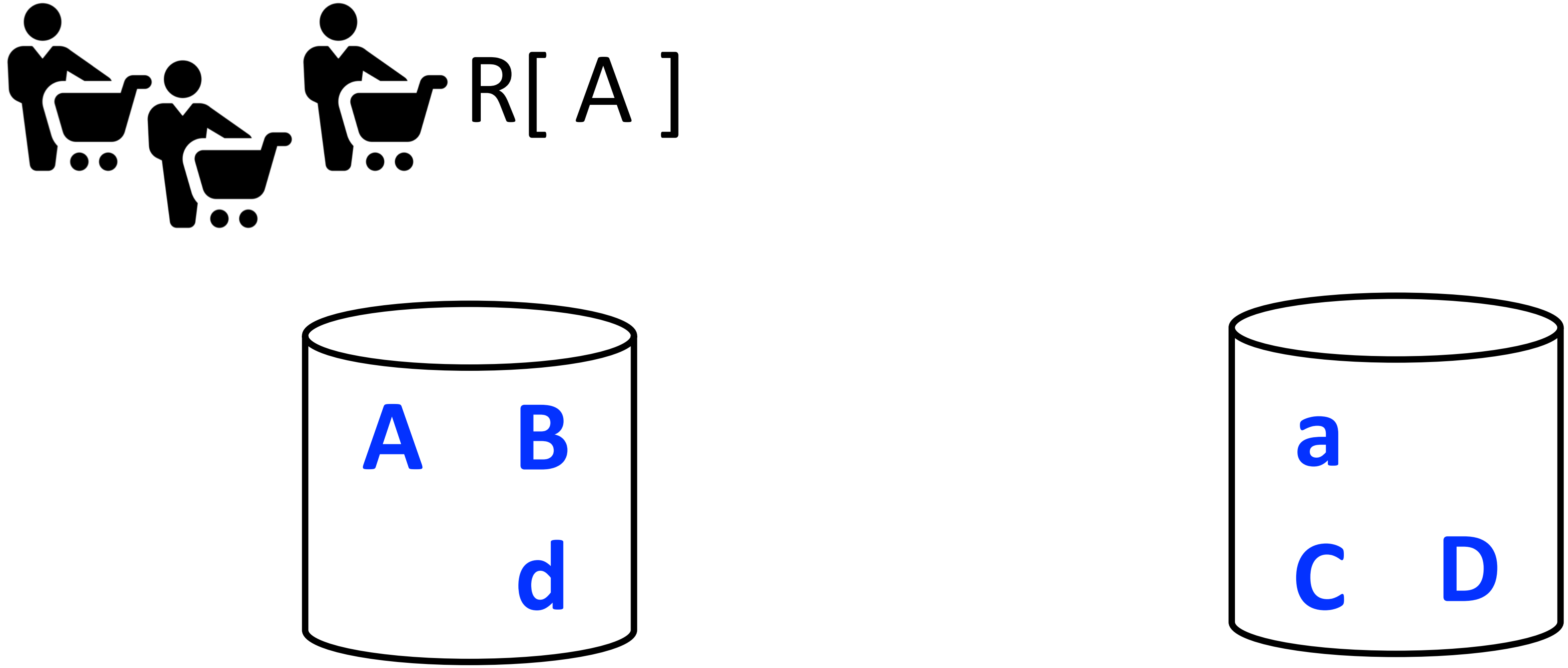
Automatically adapts the physical design

Aim: improve database system performance



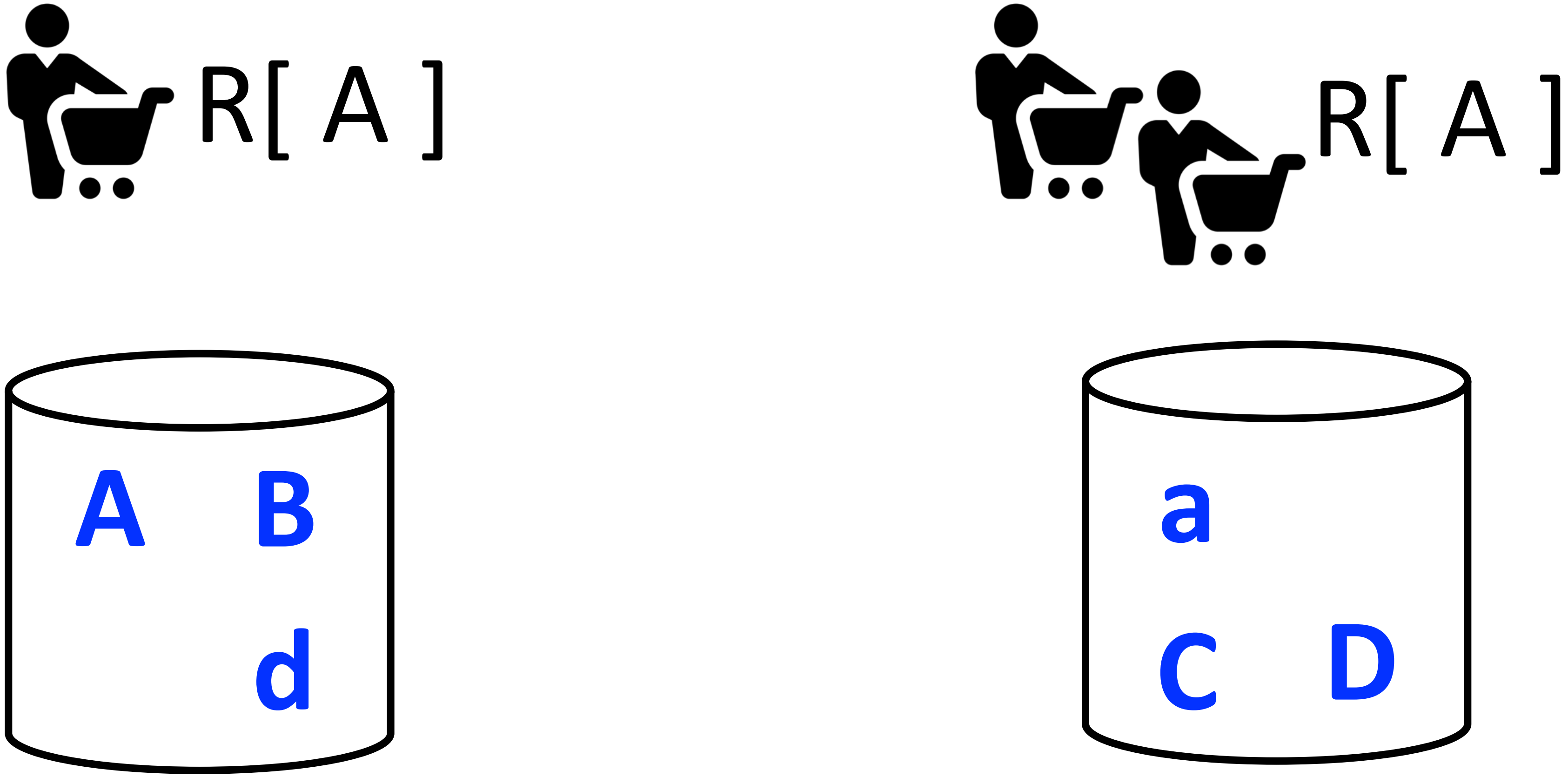
What are the **building blocks** of **automatic physical design**?

Dynamic Replication



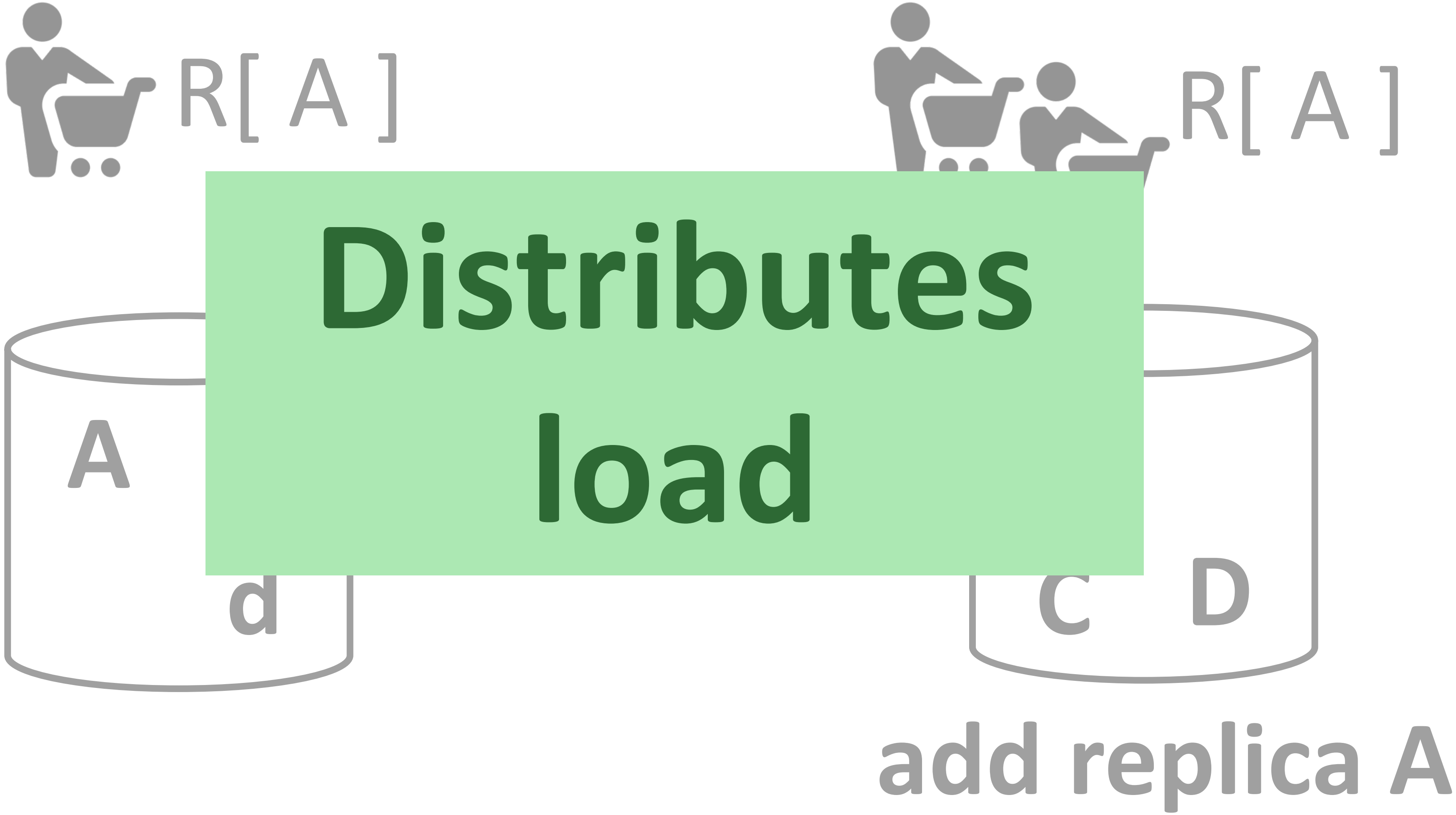
add replica A

Dynamic Replication



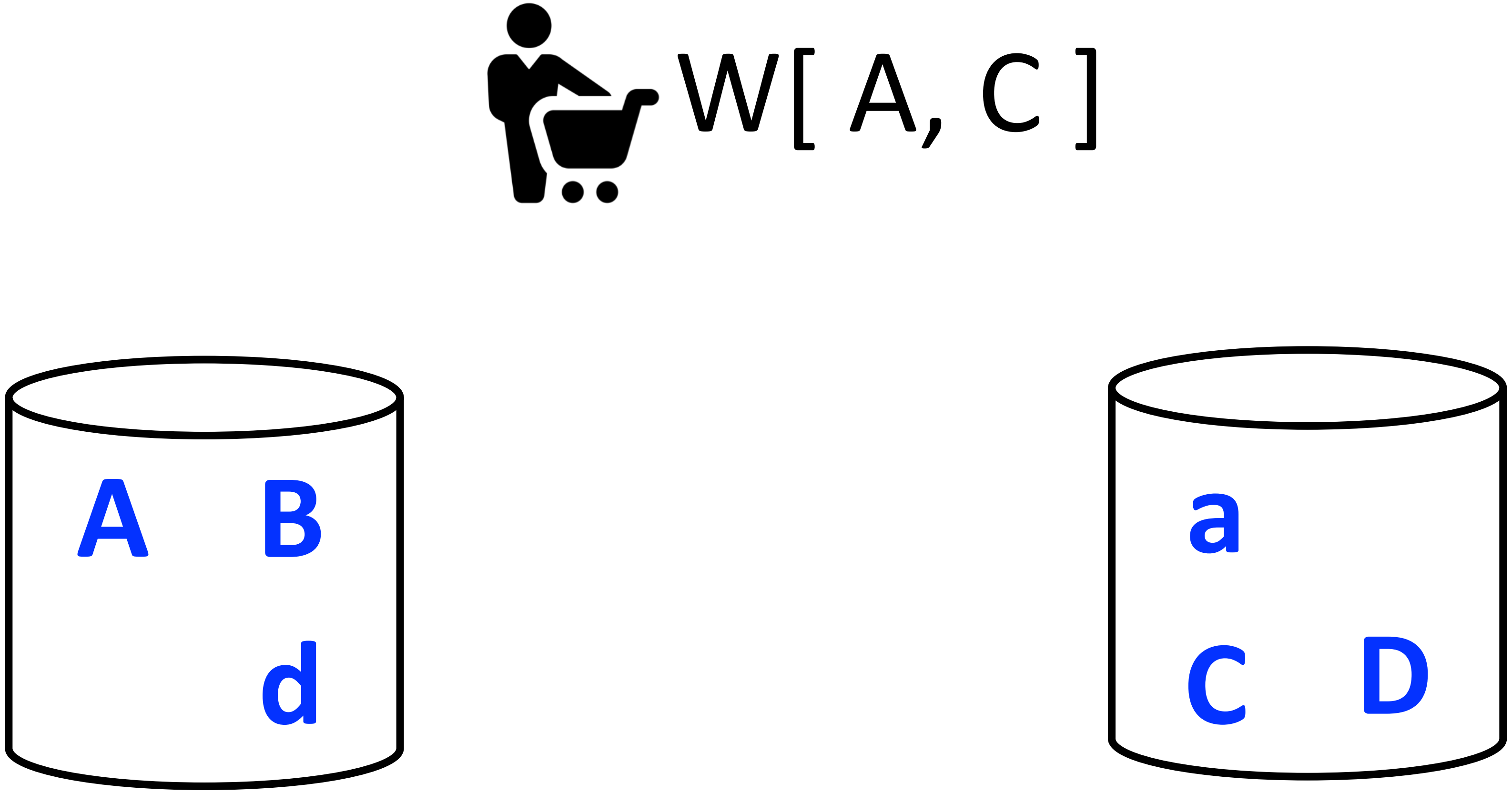
add replica A

Dynamic Replication



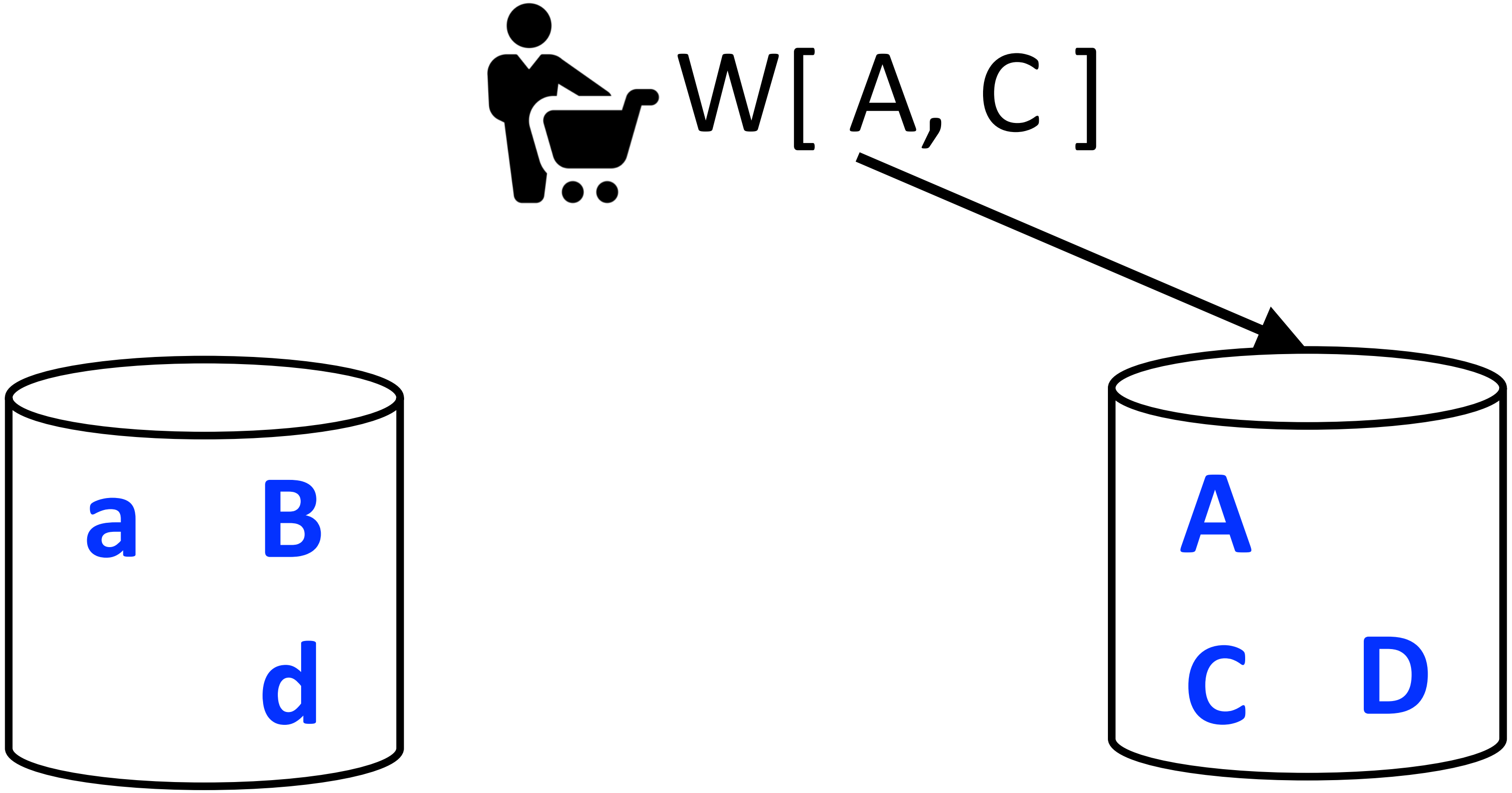
Distributes load

Dynamic Mastering



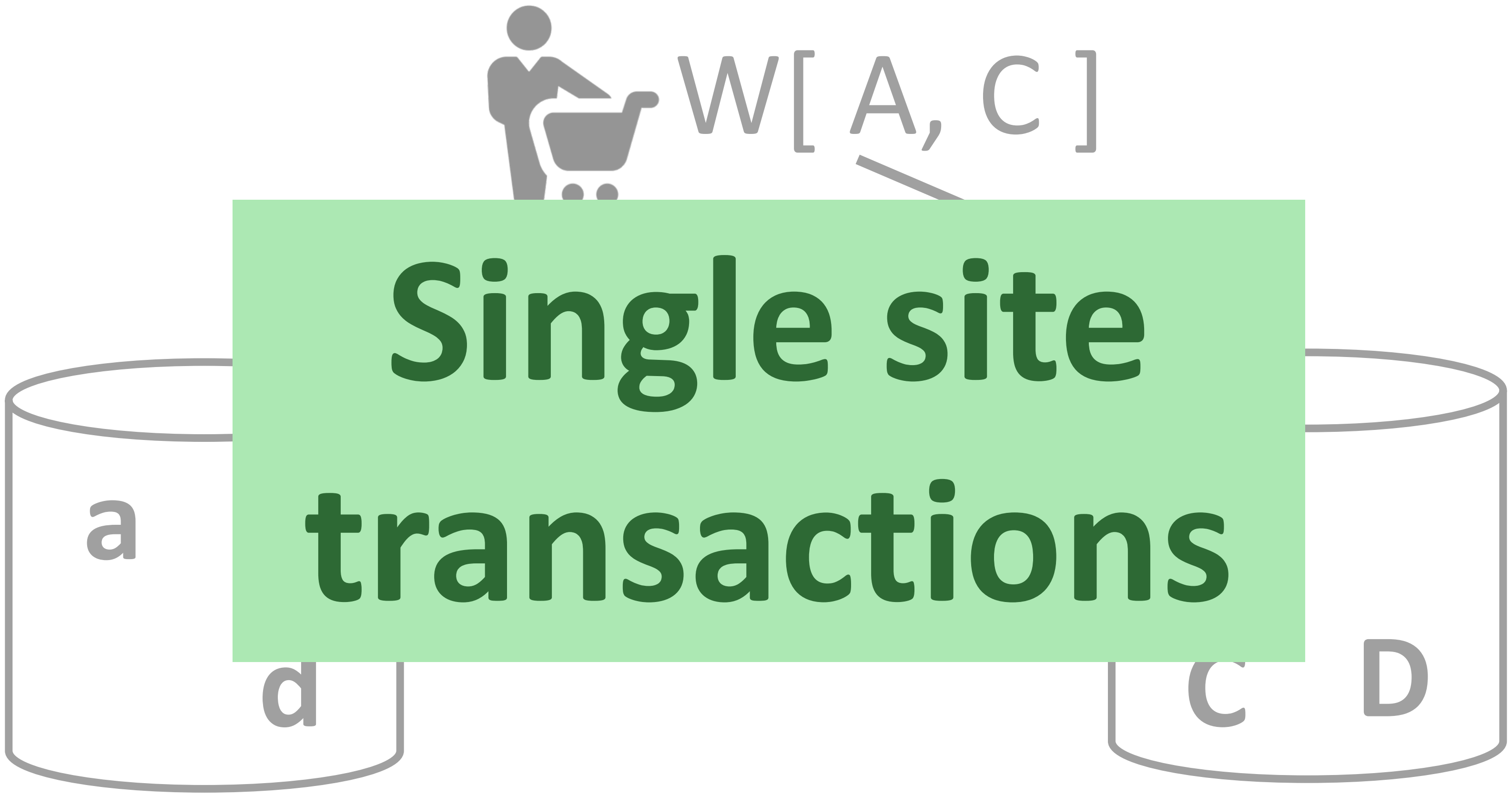
remaster A

Dynamic Mastering



remaster A

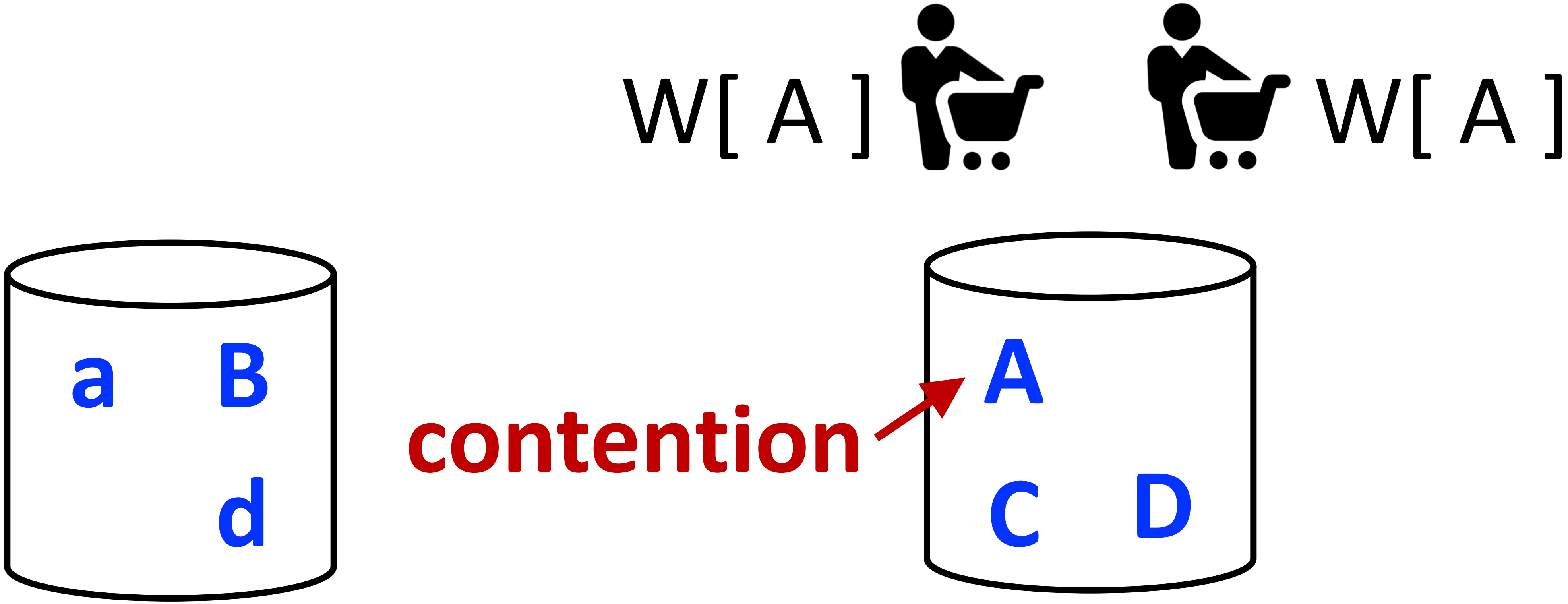
Dynamic Mastering



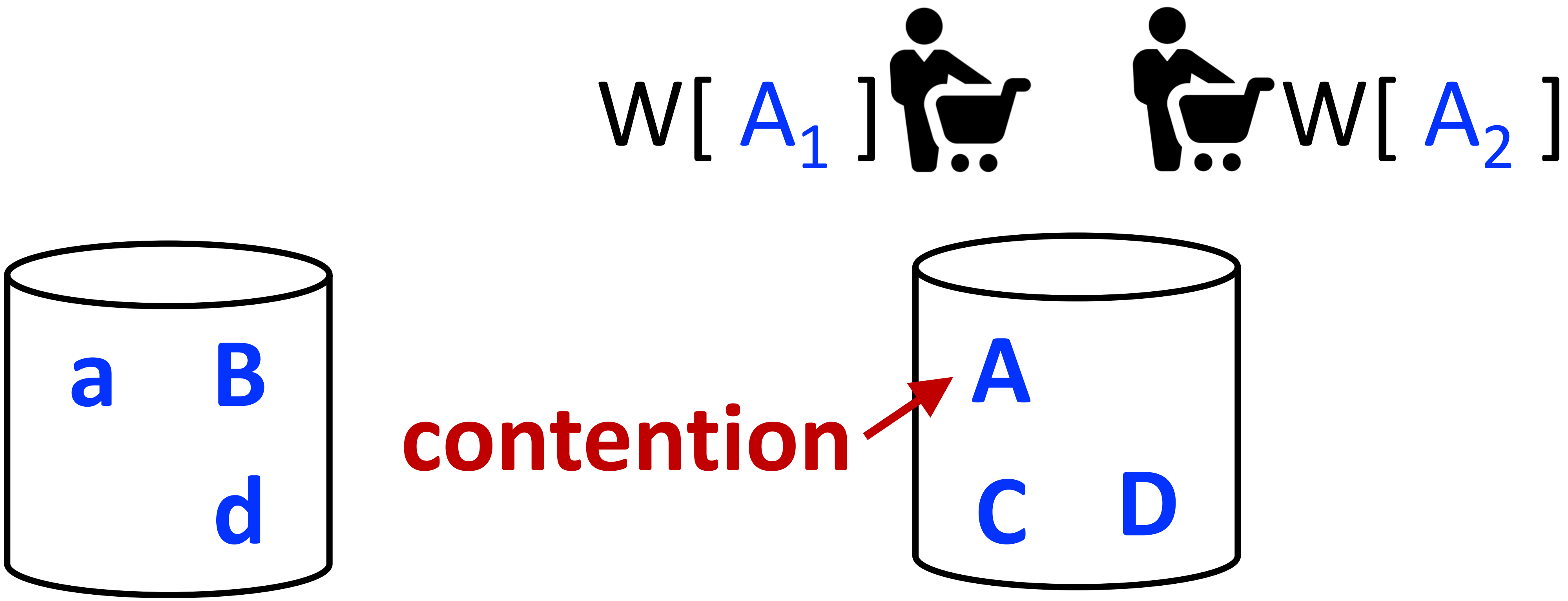
Single site transactions

remaster A

Dynamic Partitioning

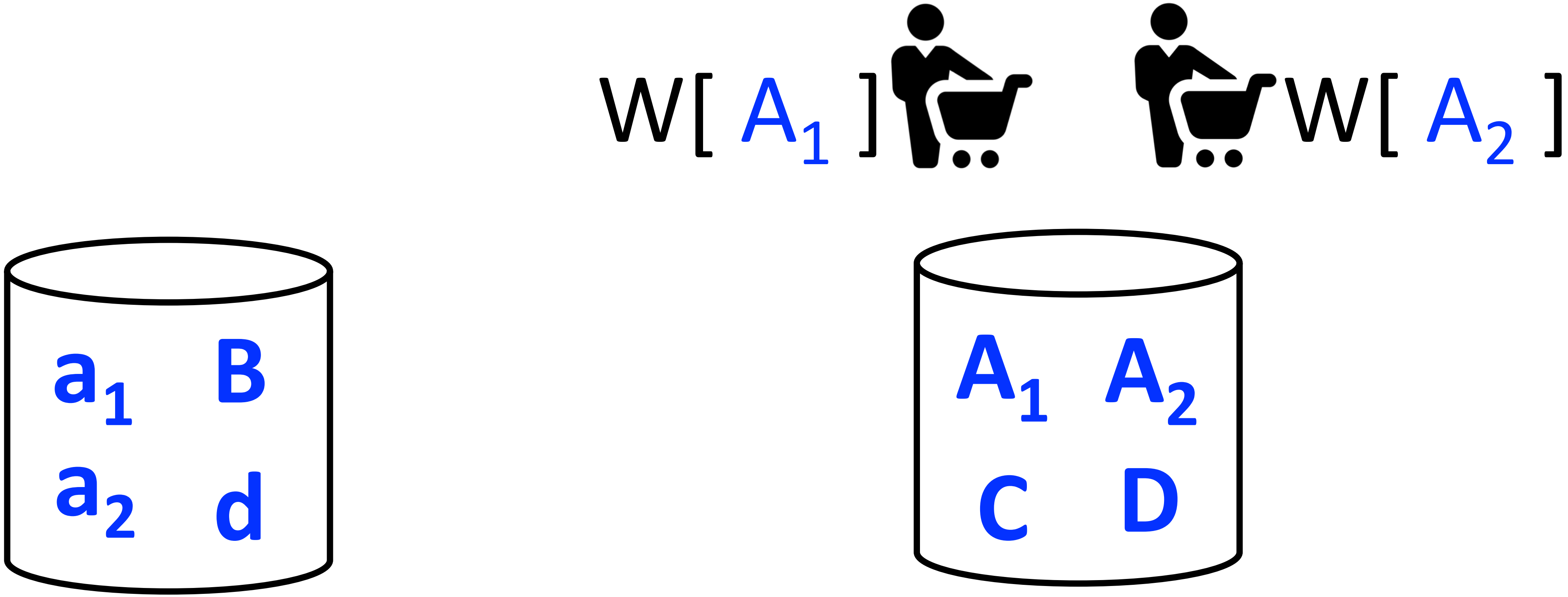


Dynamic Partitioning



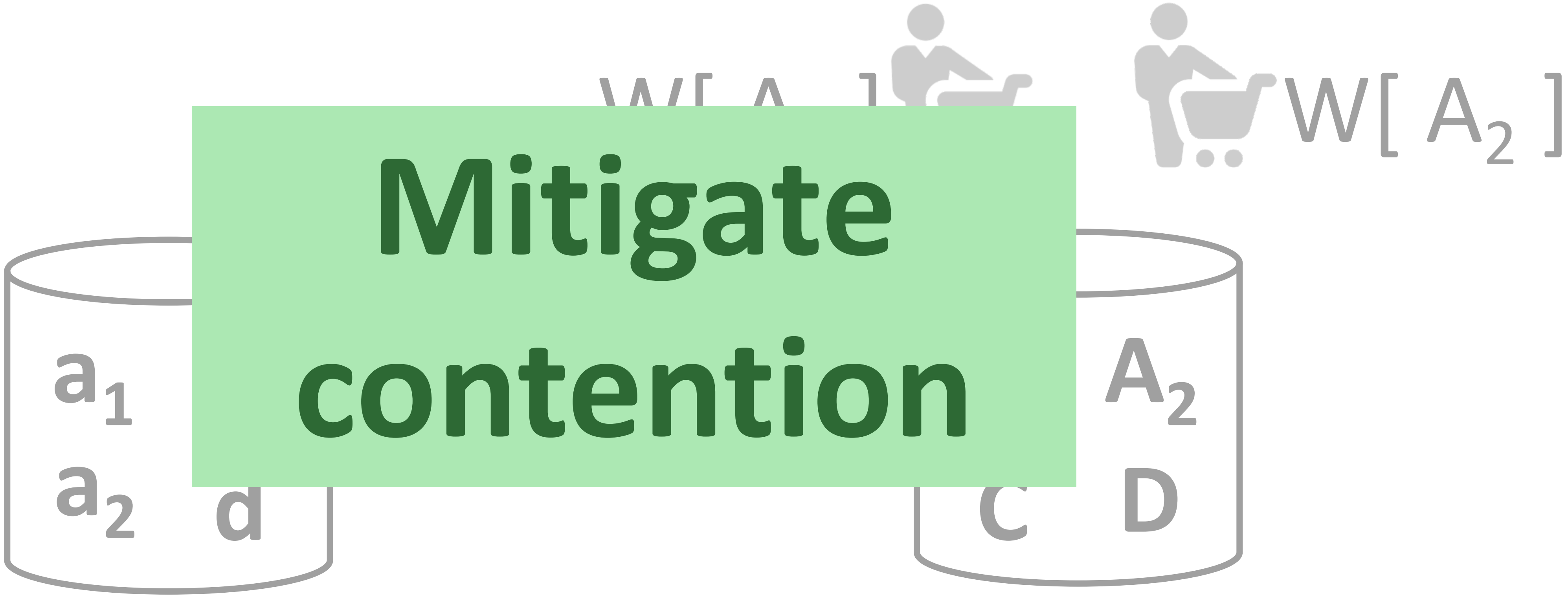
split partition A

Dynamic Partitioning



split partition A

Dynamic Partitioning



Mitigate contention

split partition A

MorphoSys Physical Design Change Operations

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

Remaster a **partition**

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

Challenges of Automatic Physical Design

How to **execute** both transactions and **design changes efficiently**

How to decide **which physical design operations** to use, and **when**

Efficient Execution

Perform all **operations** at the **partition level**

Decouple partition **reads** and **writes**

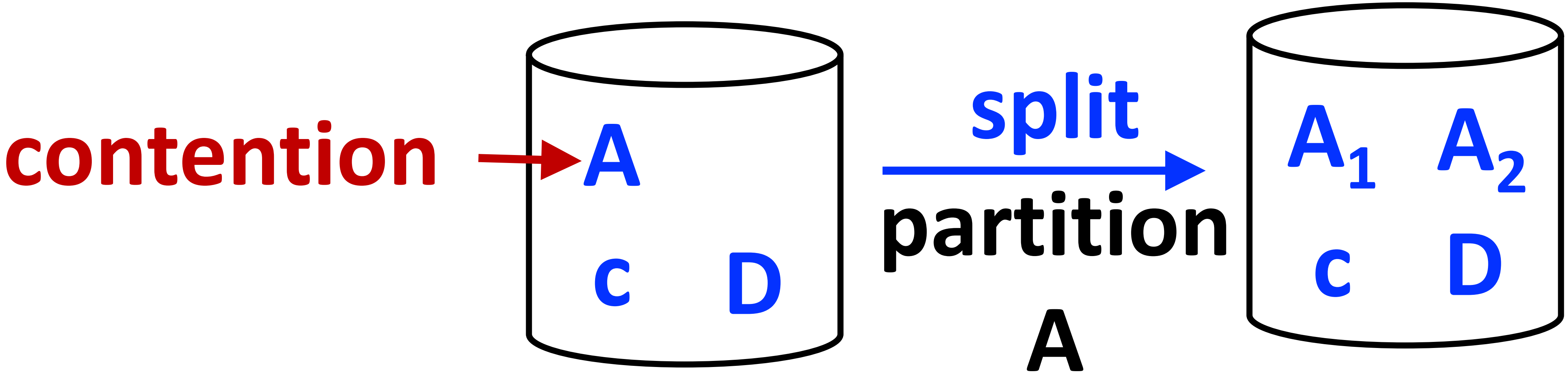
Partition based **multi-version** concurrency control

tiny.cc/morphosys

UNIVERSITY OF WATERLOO

Making design decisions

Learned cost model quantifies
design change effects



contention → A

c D

split

partition A

A_1 A_2

c 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

How well does MorphoSys work?

Comparisons

Single-Master
Multi-Master
VoltDB

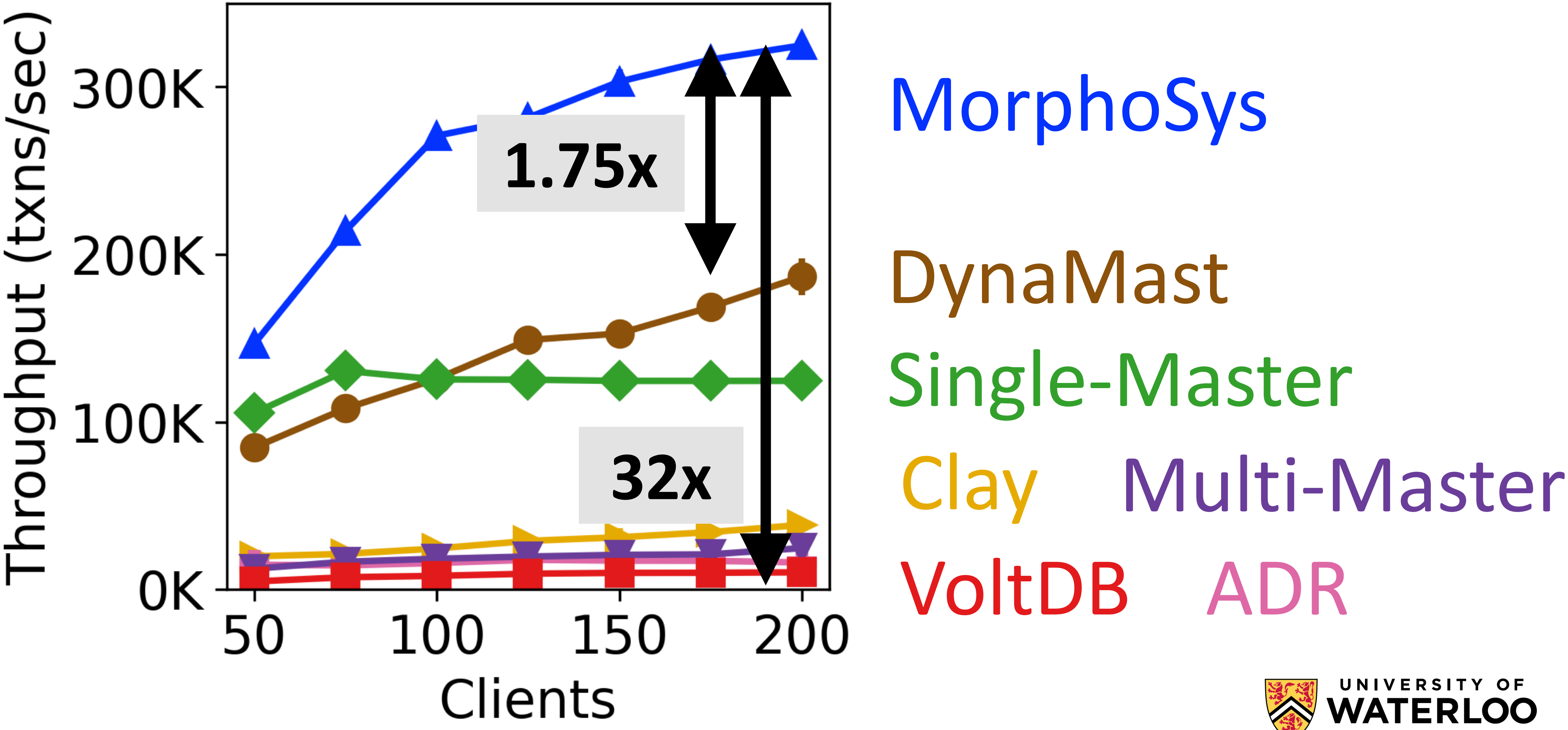
} *Static Designs*

DynaMast
Adaptive Replication (ADR)
Clay

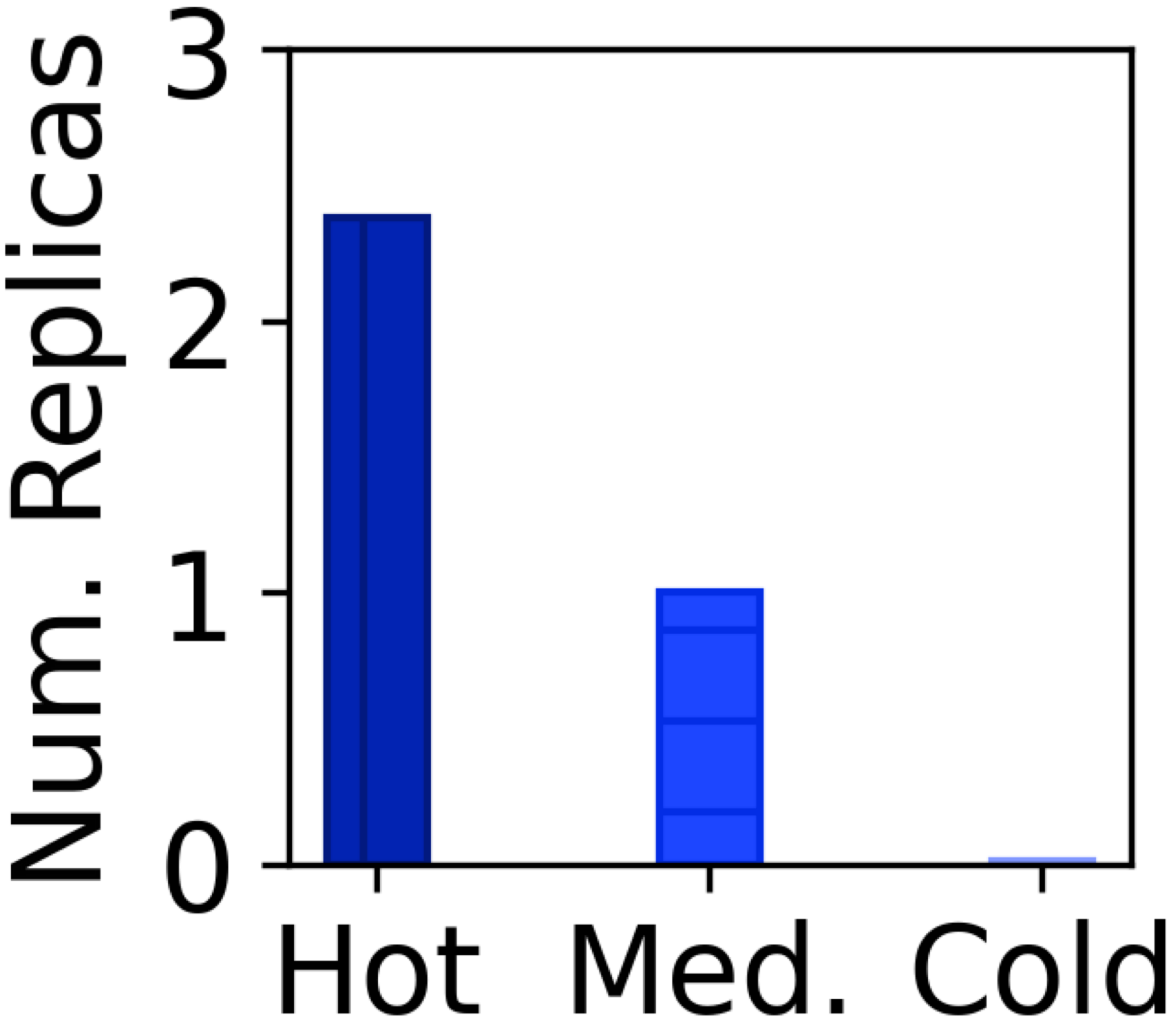
} *Dynamic Designs*

UNIVERSITY OF WATERLOO

Skewed YCSB - Throughput

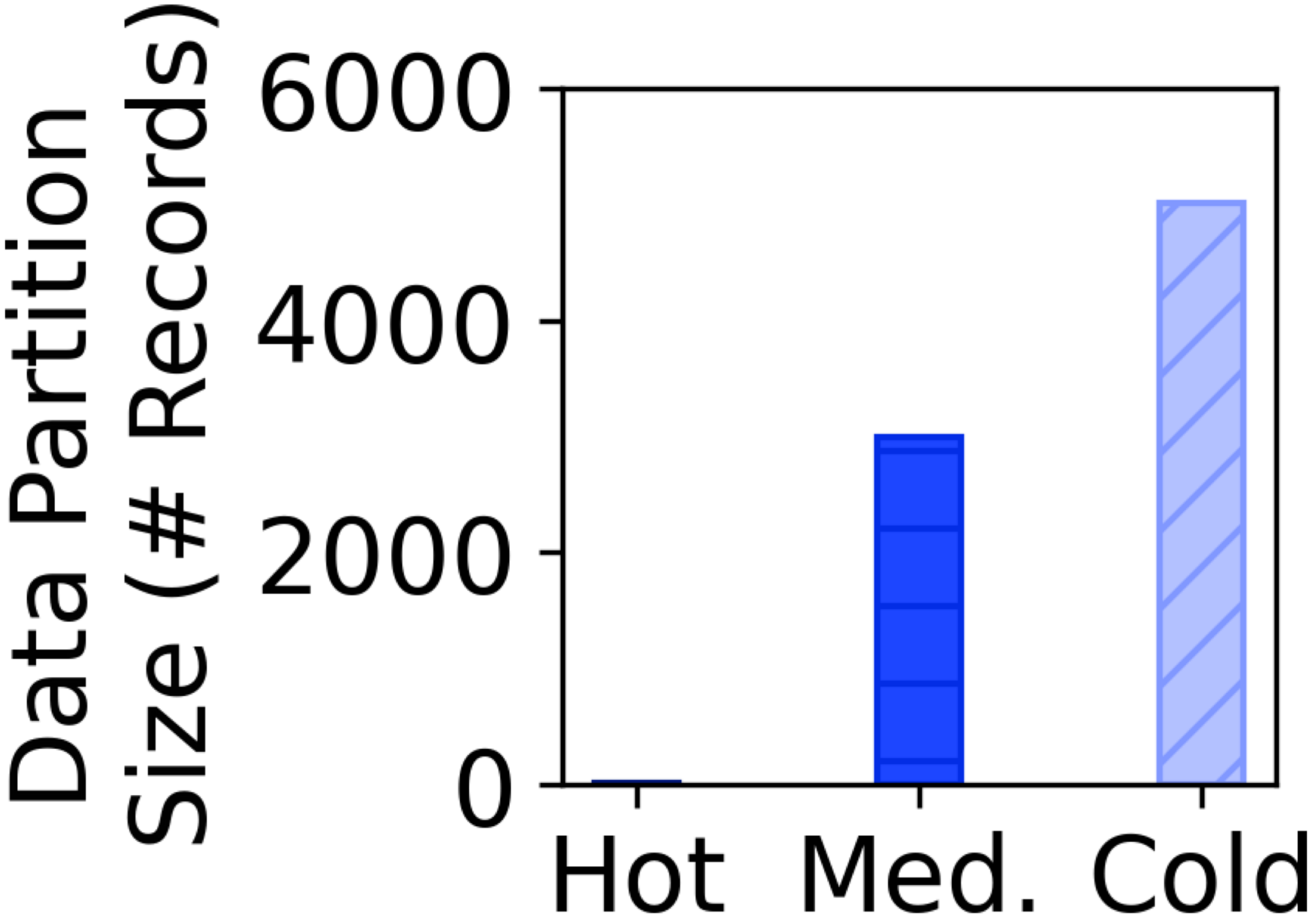


Number of Replicas

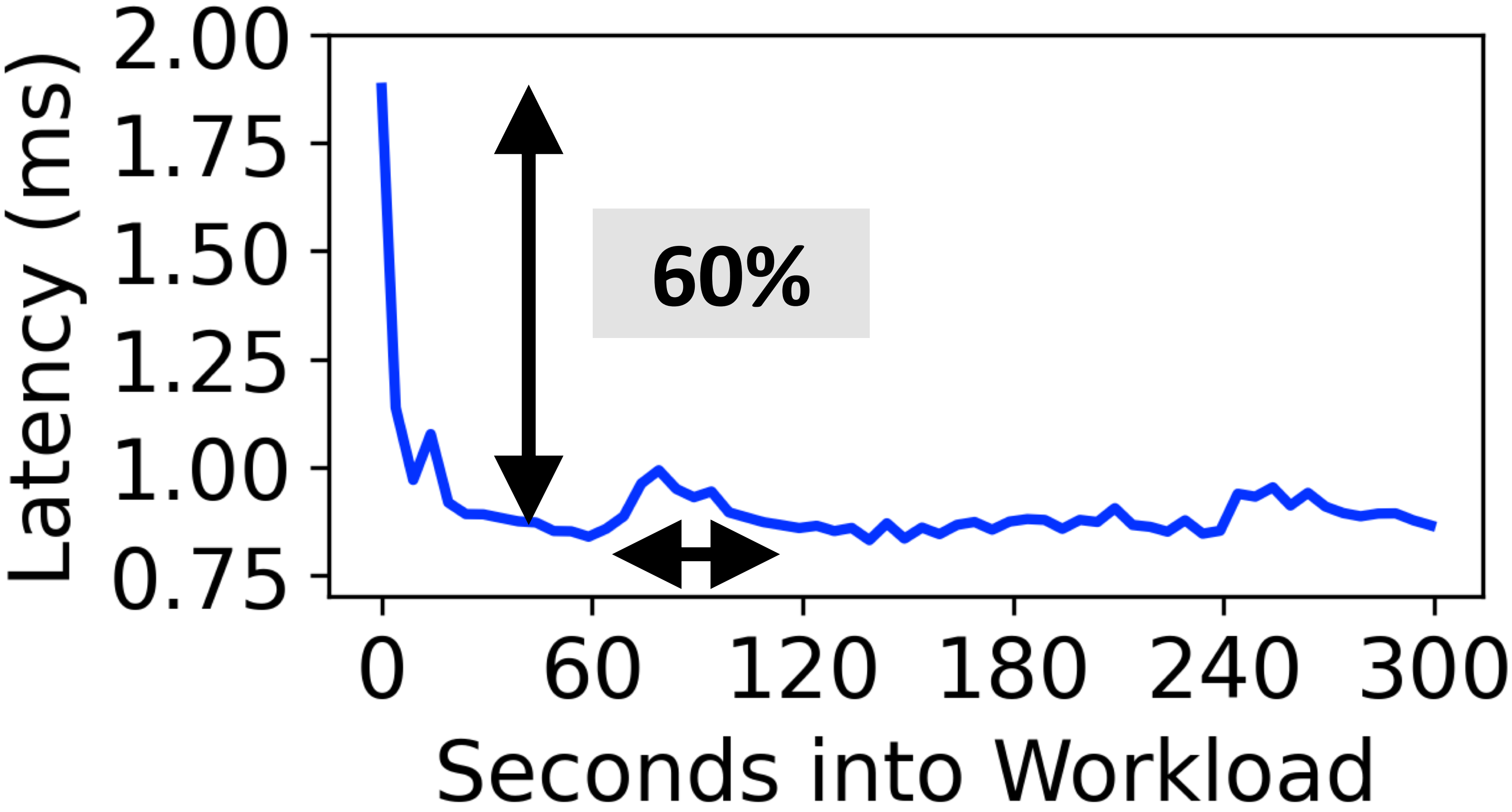


UNIVERSITY OF WATERLOO

Partition Sizes



Adapts to Workload Changes



60%

More Details in the Paper

Formalism of concurrency control

Role in replica maintenance &
design change operator execution

Generating design change plans

Learned cost functions &
building a workload model

Additional Experiments

MorphoSys Takeaways

Automatic physical design changes for distributed databases

tiny.cc/morphosys

Efficiently execute using **partition level operations**

Learned **cost model quantifies** physical design

UNIVERSITY OF WATERLOO