Dynamic Replication & Partitioning in Dynamically Mastered DBs

Brad Glasbergen
Michael Abebe
CS 848 (April 2019)
Single Database

![Diagram of a single database with four elements labeled A, B, C, and D, and three customers pushing shopping carts into the database.](image)
Single Database
Single Database

Over-Load
Replicated Databases

Writers

Master

A B
C D

Readers

Replica

a b
c d
Replicated Databases

Writers

Master

A B C D

Readers

Replica

a b c d
Replicated Databases

Writers

Master

Readers

Replica

Over-Load

Update Prop.
Partitioned Databases
Partitioned Databases

W[A]  prepare  W[C]

A  B  C  D

commit
Partitioned Databases

W[A]  

prepare  

W[C]  

Blocking

commit

A  B  C  D
Dynamically Mastered DBs
Dynamic Mastering

release A

\[ A \rightarrow B \]
\[ A \rightarrow c \]
\[ A \rightarrow d \]

grant A

\[ a \rightarrow A \rightarrow b \]
\[ A \rightarrow C \]
\[ A \rightarrow D \]
Dynamic Mastering

Single site txns

Complete replication
Dynamic Replication & Partitioning
Dynamic Replication
Dynamic Replication

add replica

remove replica

A
C
D
Dynamic Replication

Store more data

Distribute load

add replica
remove replica

A
C
D
Dynamic Partitioning

 contention

 W[A]

 W[A]
Dynamic Partitioning

contention

W[A₁]

W[A₂]
Dynamic Partitioning

$W[A_1]$  $W[A_2]$
Dynamic Partitioning

split partition
merge partition

A_1  A_2
C    D
Dynamic Partitioning

Mitigate contention

split partition
merge partition
Architecture

Data Site

W[A]

Site Selector

Route

Propagate

Data Site

a
B

A
C
D

d
DRP Challenges

How to **execute** operations efficiently?

How to **decide** which operations to use?
DRP Challenges

How to **execute** operations efficiently?

How to **decide** which operations to use?
Decouple partition reads & writes

Partition based multi-version concurrency control
Update Propagation

Propagate

W[A]
Update Propagation

Subscribe to partition updates
Update Propagation

Subscribe to partition updates

Built in redo-log
Update Propagation

**Exploit** multi-versioning to apply updates

**Multiplex** partition updates to Kafka

Remastering & repartitioning requires **changing subscriptions**
Adding replicas

**Exploit** multi-versioning & Kafka log

Take a **read-only** partition **snapshot**

Install snapshot & **subscribe** to Kafka
DRP Challenges

How to execute operations efficiently?

How to decide which operations to use?
The Cost Model

1: Upfront Transfers
The Cost Model

2: Queue Time

Queue
T1
T2
...

2: Queue Time
The Cost Model

3: Update Time

Apply:
T15
T22
The Cost Model

3: Lock Time

Waiters
A: T1, T2, T3
B:
In Short:

\[ \arg \min_D \left[ \mathbb{E}_T \left[ C_D(T) \right] \right] \]

ILP? Offline/Expensive

Online/Iterative Approach
Ex: Adding a Replica

Add Replica of A?

Only affects reads!
Ex: Adding a Replica

Add Replica of A?

Splits R[A] load
Ex: Adding a Replica

Add Replica of A?

Splits R[A] load

Reduces Queue Time
Ex: Adding a Replica

Add Replica of A?

Apply a’s updates!

Increases Queue Time
Ex: Adding a Replica

Add Replica of A?

Does not split R[A,B] load!
Add Replica Strategy

Compare estimated load balance before and after proposed replica placement.
DRP Takeaways

Avoid distributed coordination

Dynamic replication and partitioning

Online iterative physical design adjustments
What’s Done:

- Update Propagation and Infrastructure Support
- Basic Underlying Cost Model
- Strategy Design for Split/Merge Partitions, Add/Remove Replicas, Remastering, Transaction Routing
- Statistics Support, Tracking, Sampling
What’s Left:

- Implementing Strategies into DRP
- Comparisons against alternative strategies/baselines
- Comprehensive Experimental Evaluation
- Beyonds this course: Optimization