

Adaptive Replication and Partitioning in Data Systems

Data Systems Group



Brad Glasbergen,
Michael Abebe,
Khuzaima Daudjee

Middleware 2018



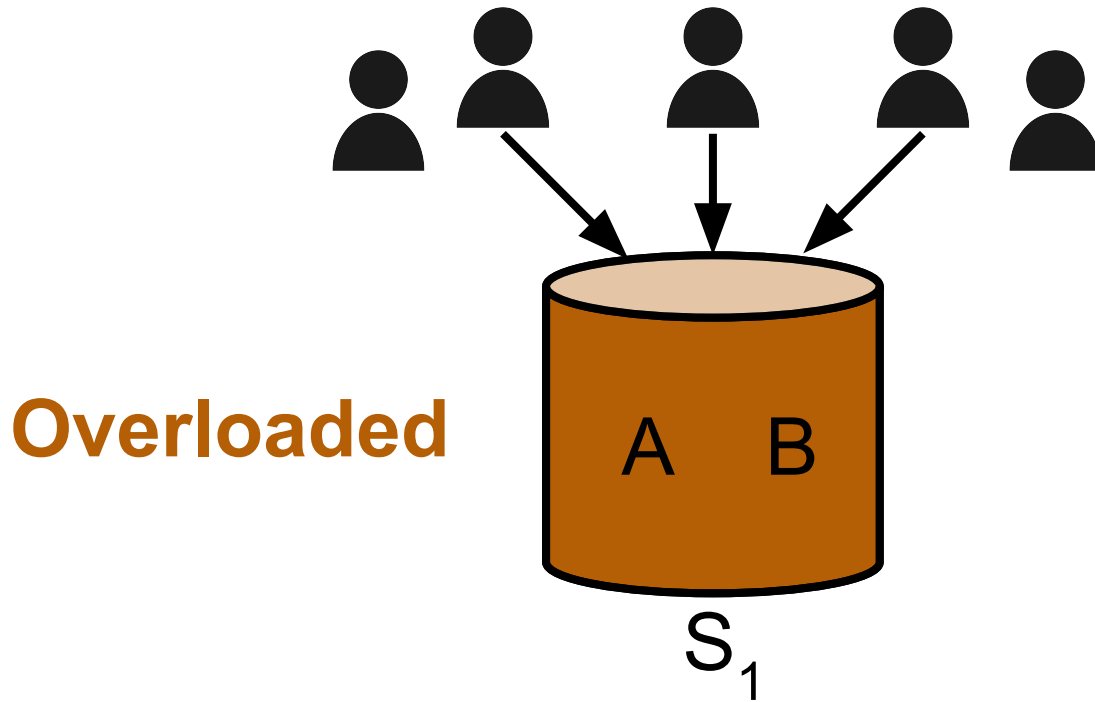
BARK.AI







Single Node Architecture



Single Node Architecture

How to scale beyond a single node?

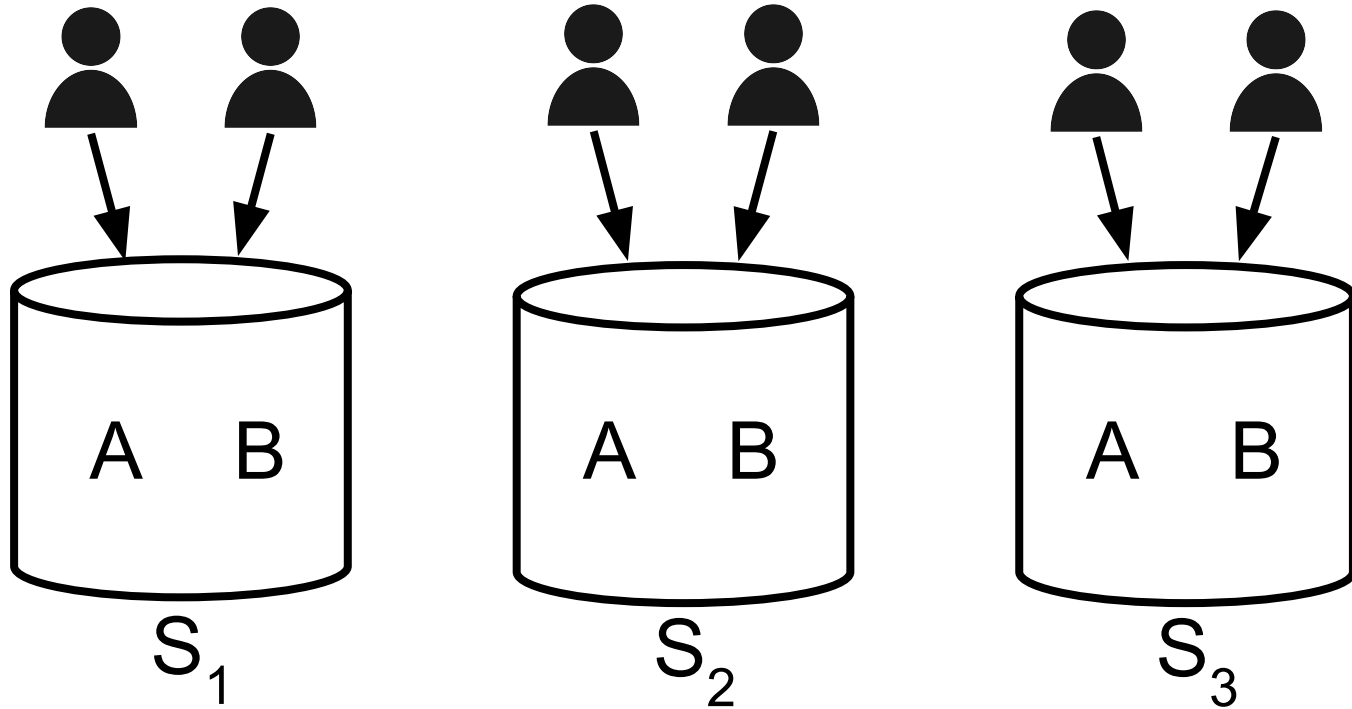
Replicate and partition

Single Node Architecture

How to scale beyond a single node?

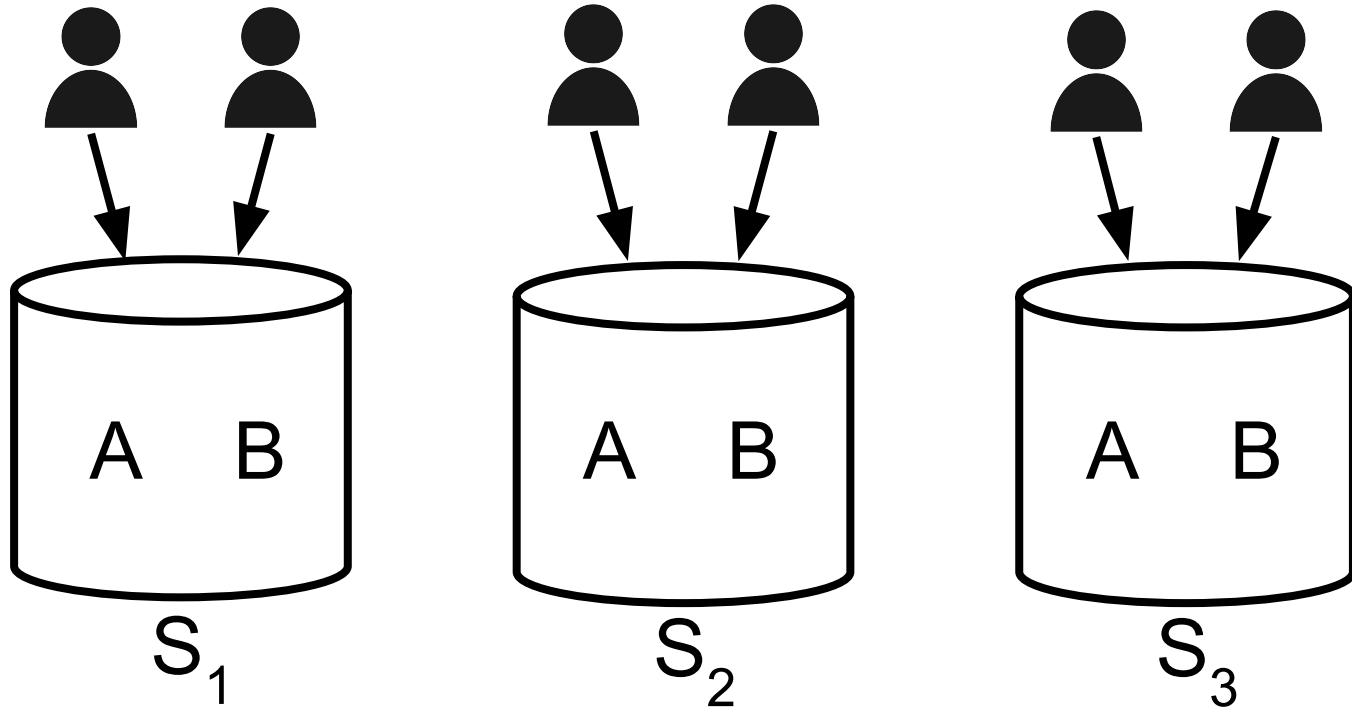
Replicate and partition

Replicated Architecture



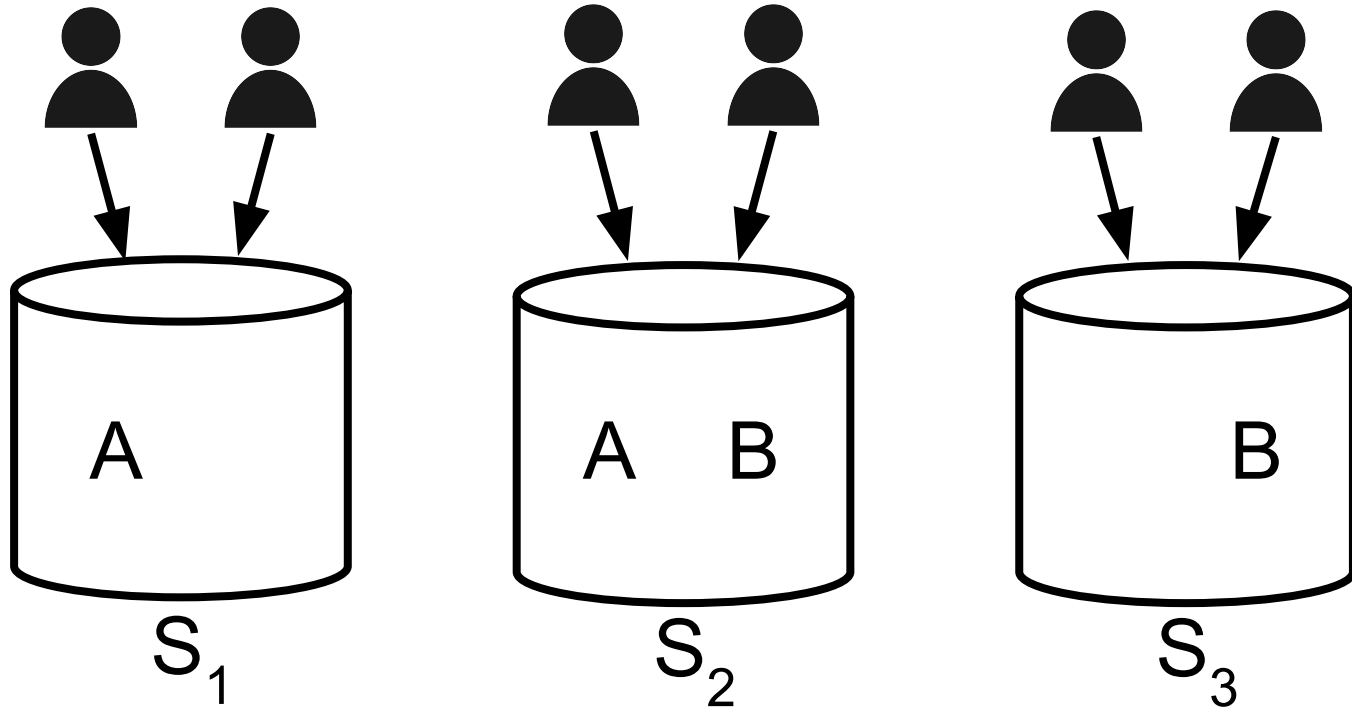
Handle more requests

Replicated Architecture



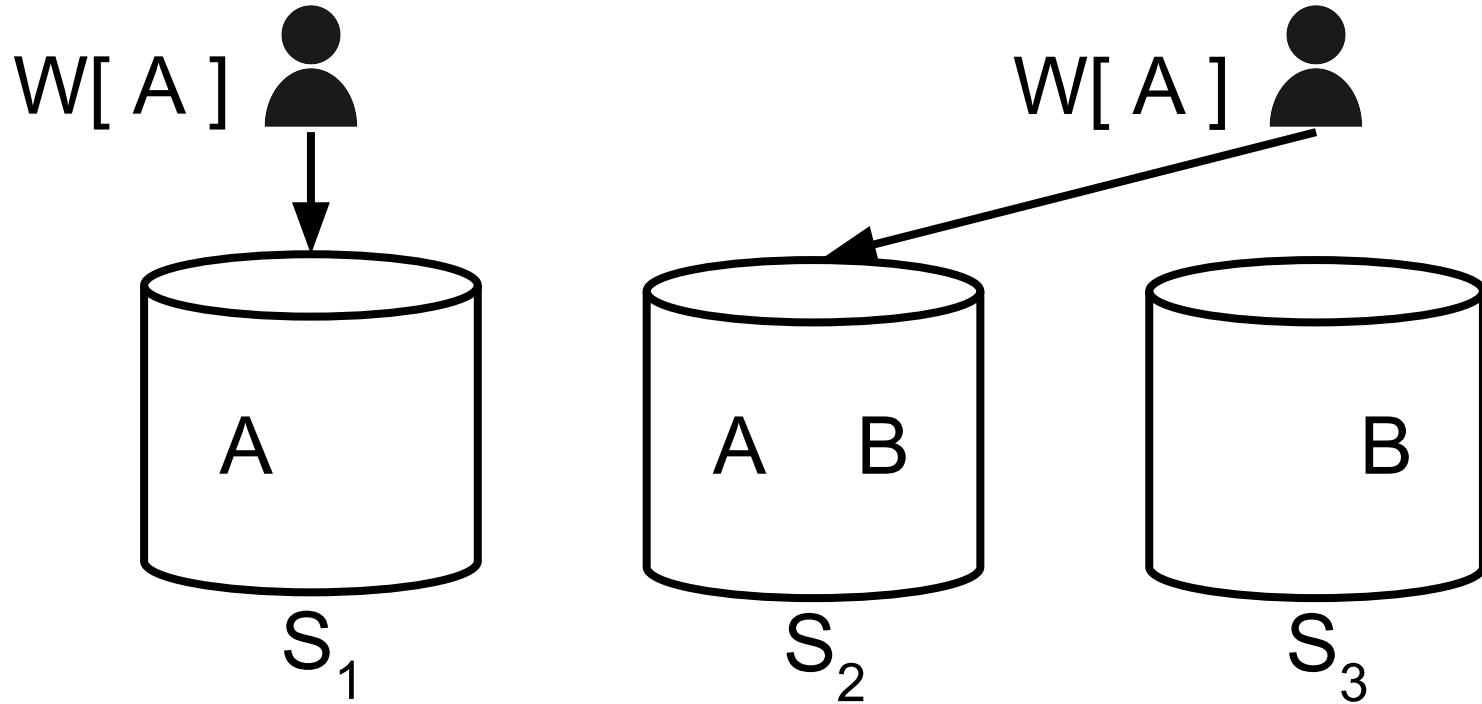
Cost of coordination

Replicated Architecture

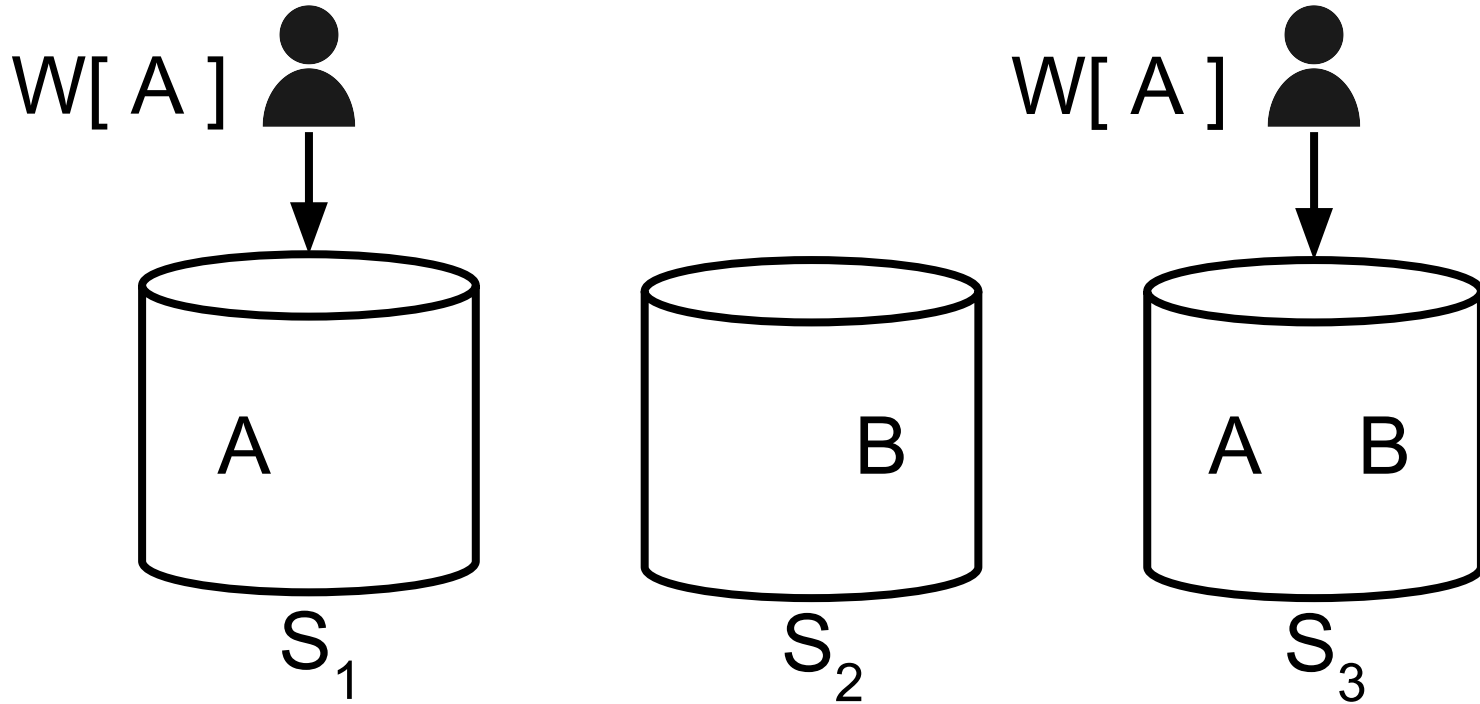


How many replicas?

Replicated Architecture

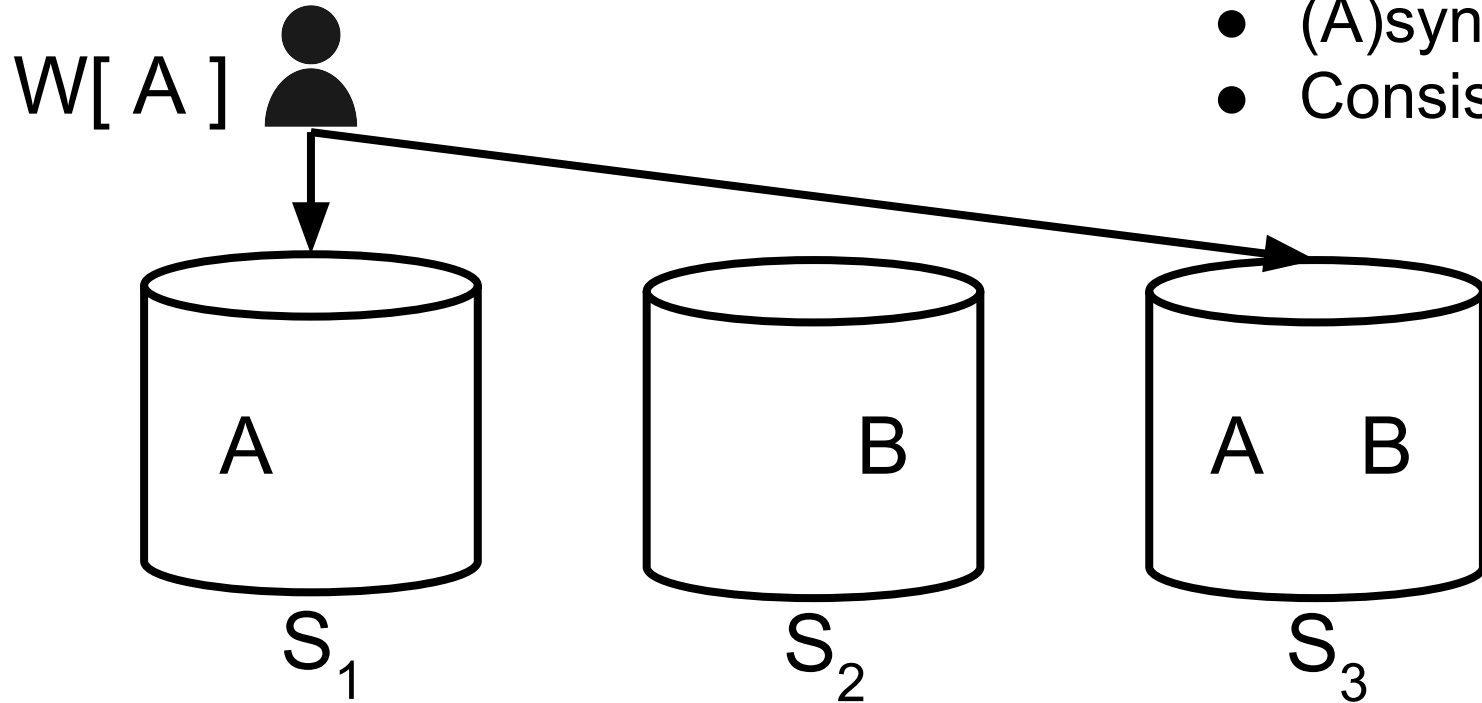


Replicated Architecture



Where to **place** replicas?

Replicated Architecture



- (A)synchronous
- Consistency

How to propagate updates?

Replication Decisions

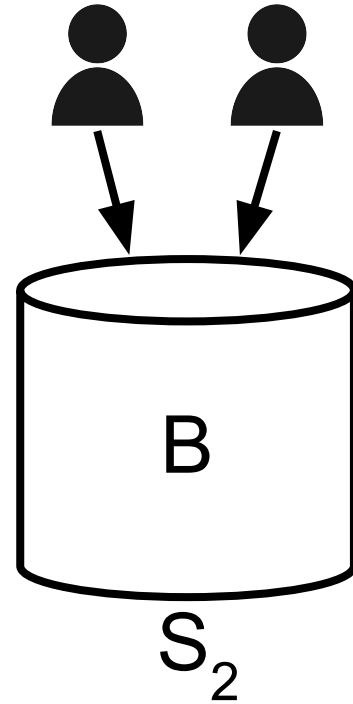
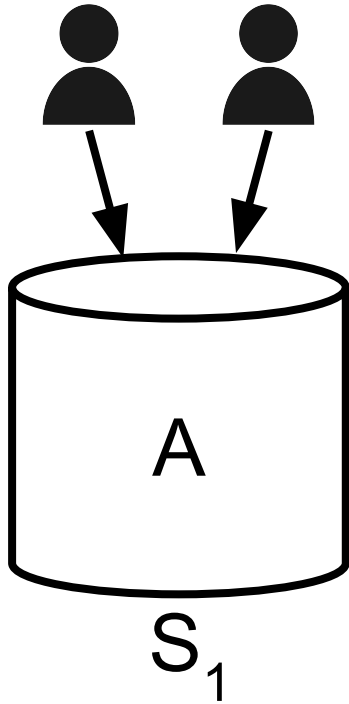
- **How many** replicas?
- **Where to place** replicas?
- **How to propagate** updates?

Single Node Architecture

How to scale beyond a single node?

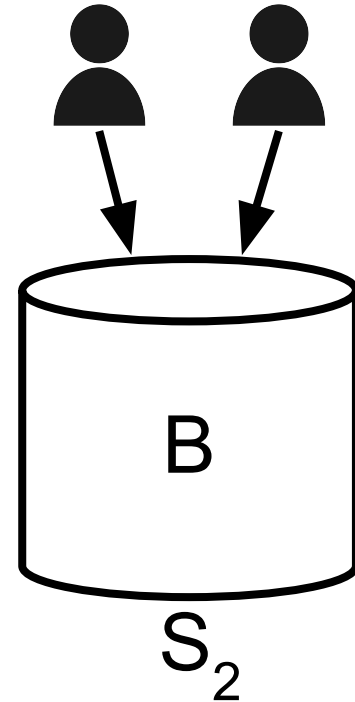
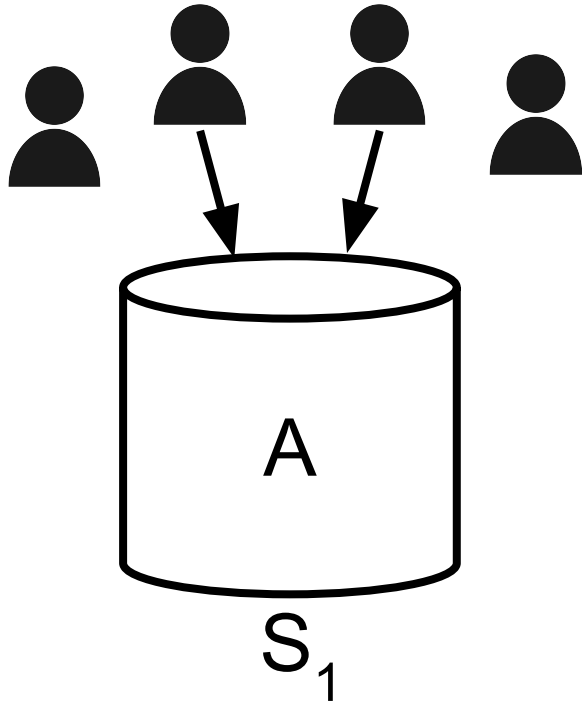
Replicate and partition

Partitioned Architecture

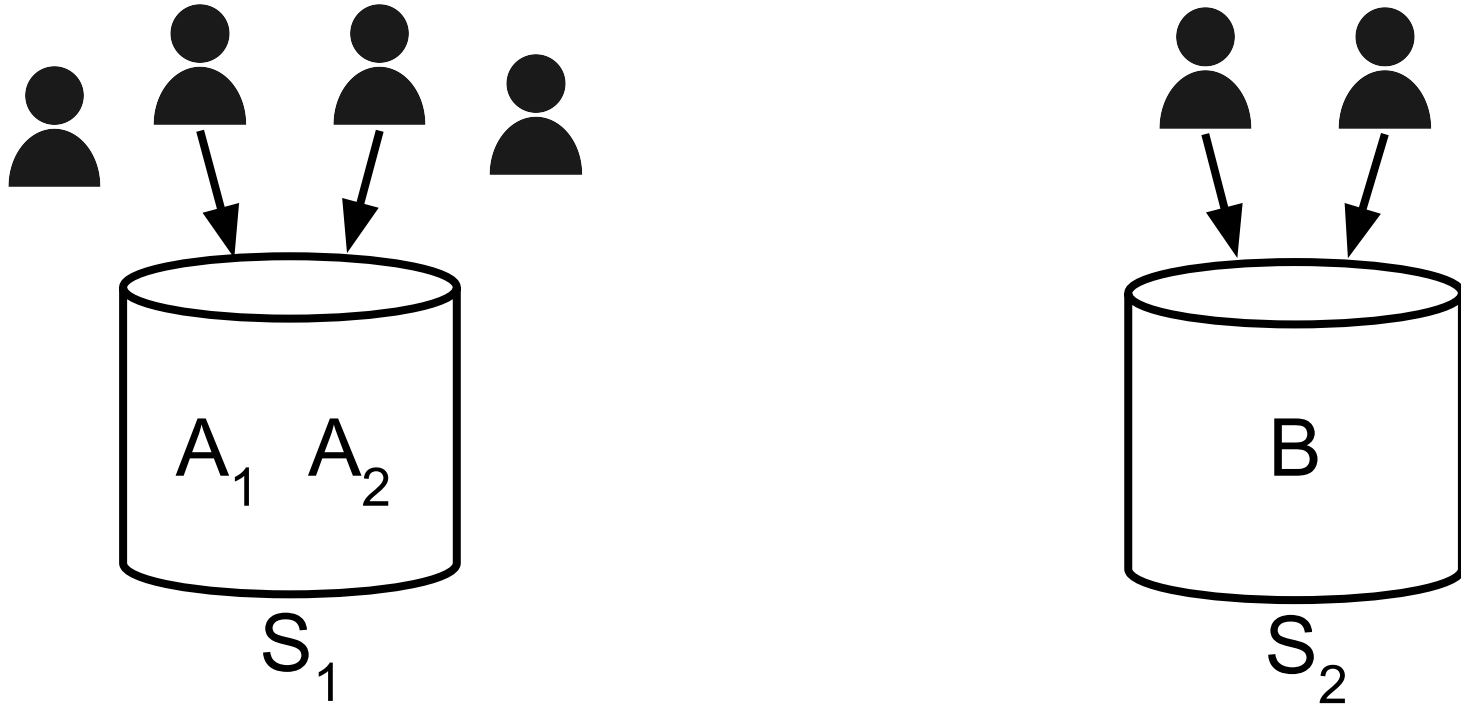


Distributes requests

Partitioned Architecture

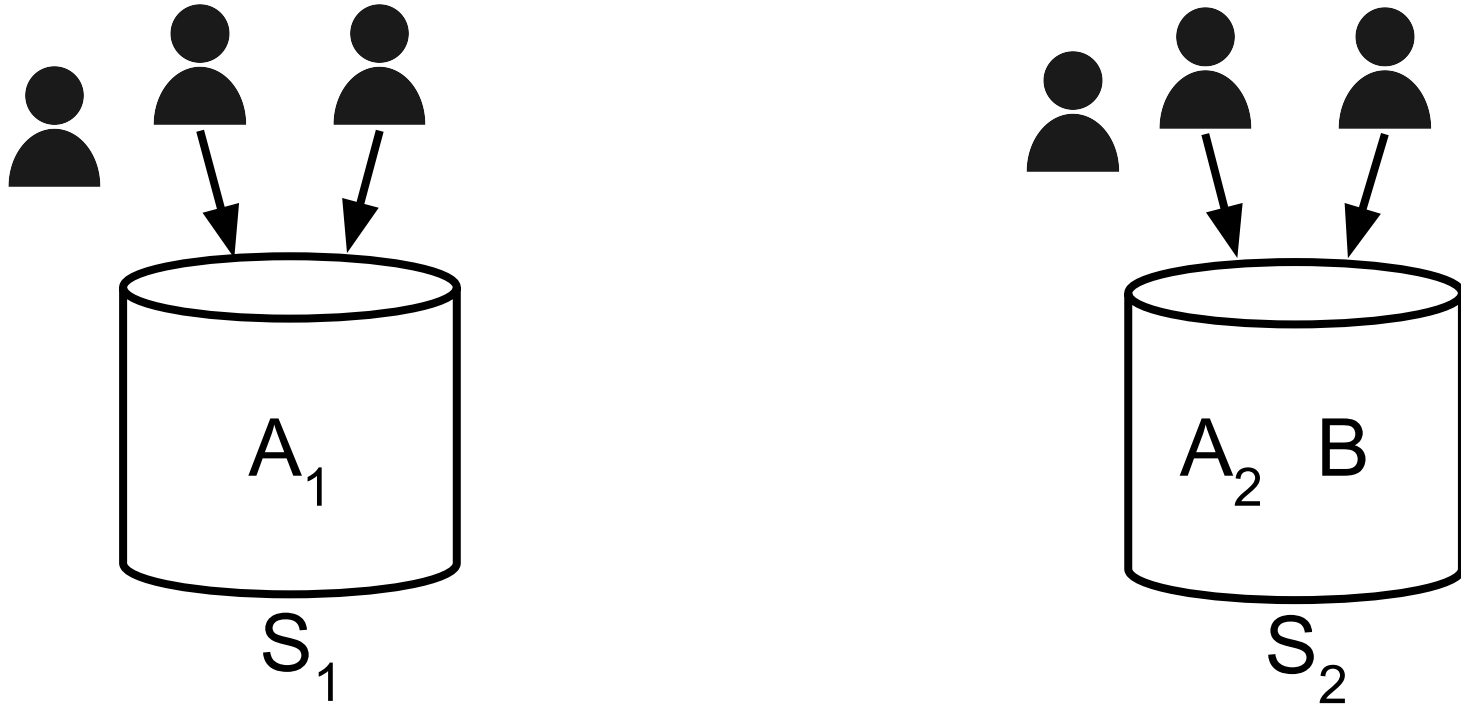


Partitioned Architecture



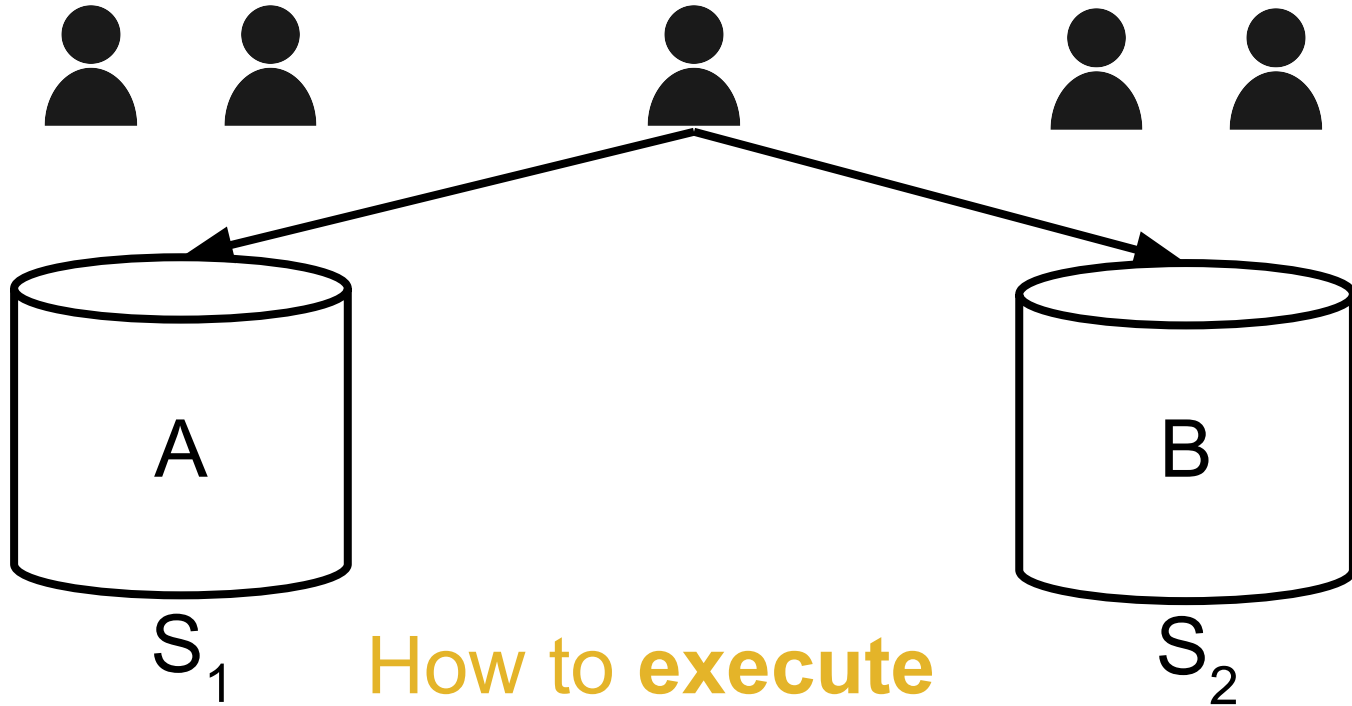
How to form partitions?

Partitioned Architecture



Where to **place** partitions?

Partitioned Architecture

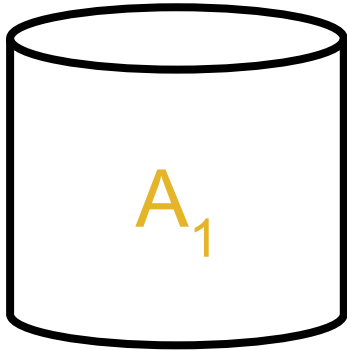


How to **execute**
multi-partition operations?

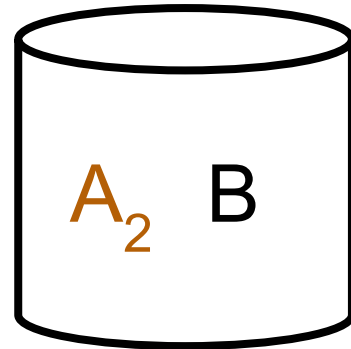
Partitioning Decisions

- **How to form** partitions?
- **Where to place** partitions?
- **How to execute** multi-partition operations?

Where to place partitions?



S_1



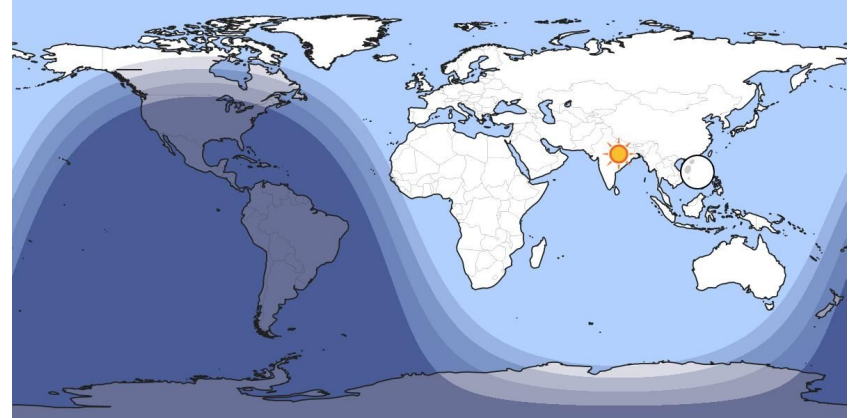
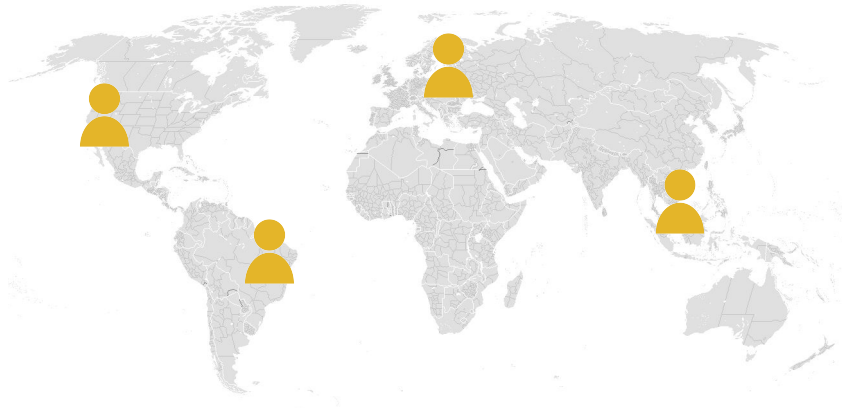
S_3

Static Decisions

How to **make** a partitioning or replication **decision** when access patterns **change**?

Why do access patterns change?

Why do accesses change?



Humans have **follow-the-sun cycles**

Why do accesses change?



reddit

Load bursts

Why do accesses change?



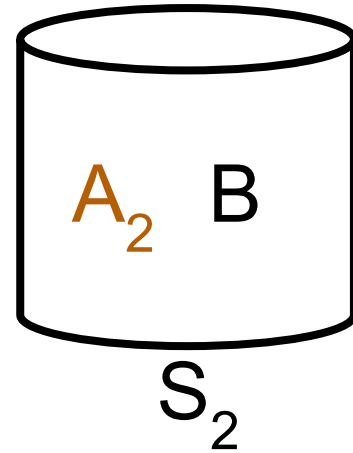
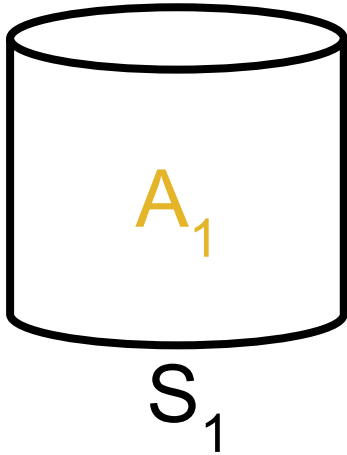
Shifting hot-spots

Static Decisions

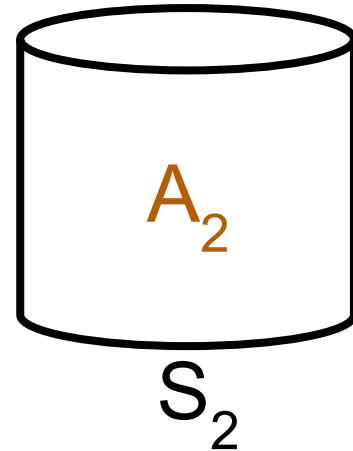
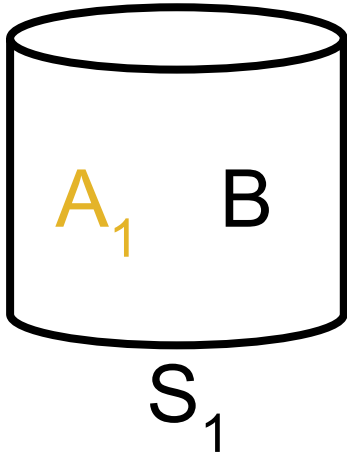
How to **make** a partitioning or replication **decision** when access patterns **change**?

Adaptively replicate and partition

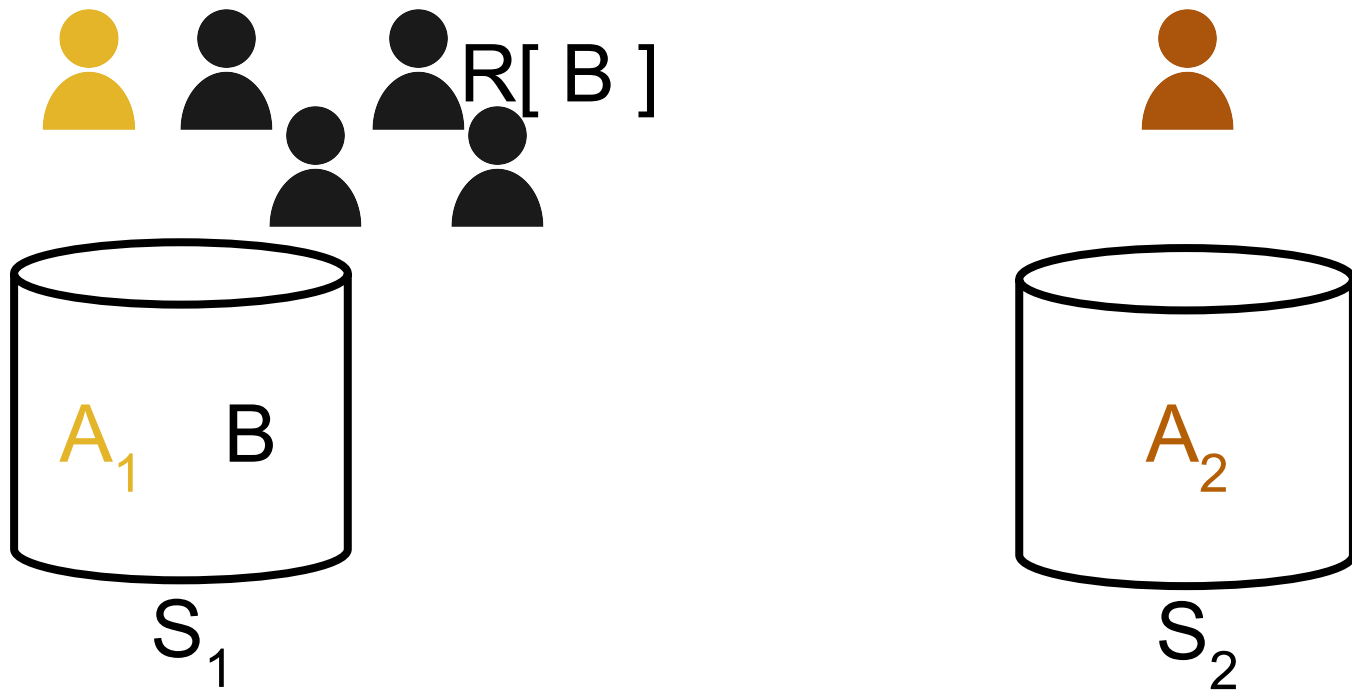
Where to place partitions?



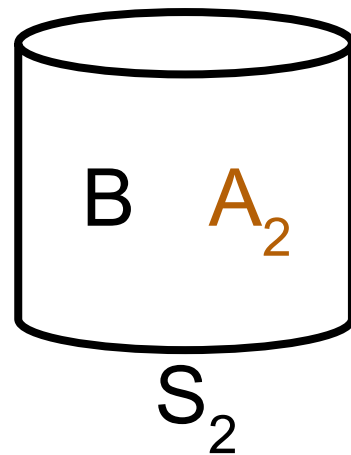
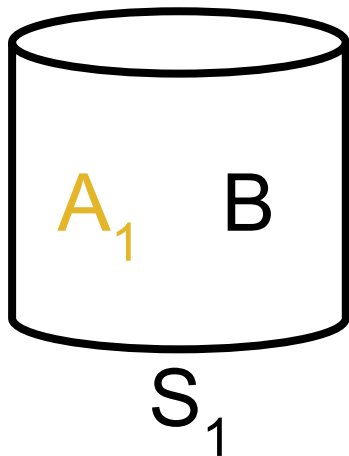
Where to place partitions?



How many replicas?



How many replicas?



Road Map

- Adaptive Replication
- Adaptive Partitioning
- Outlook

Adaptive Replication

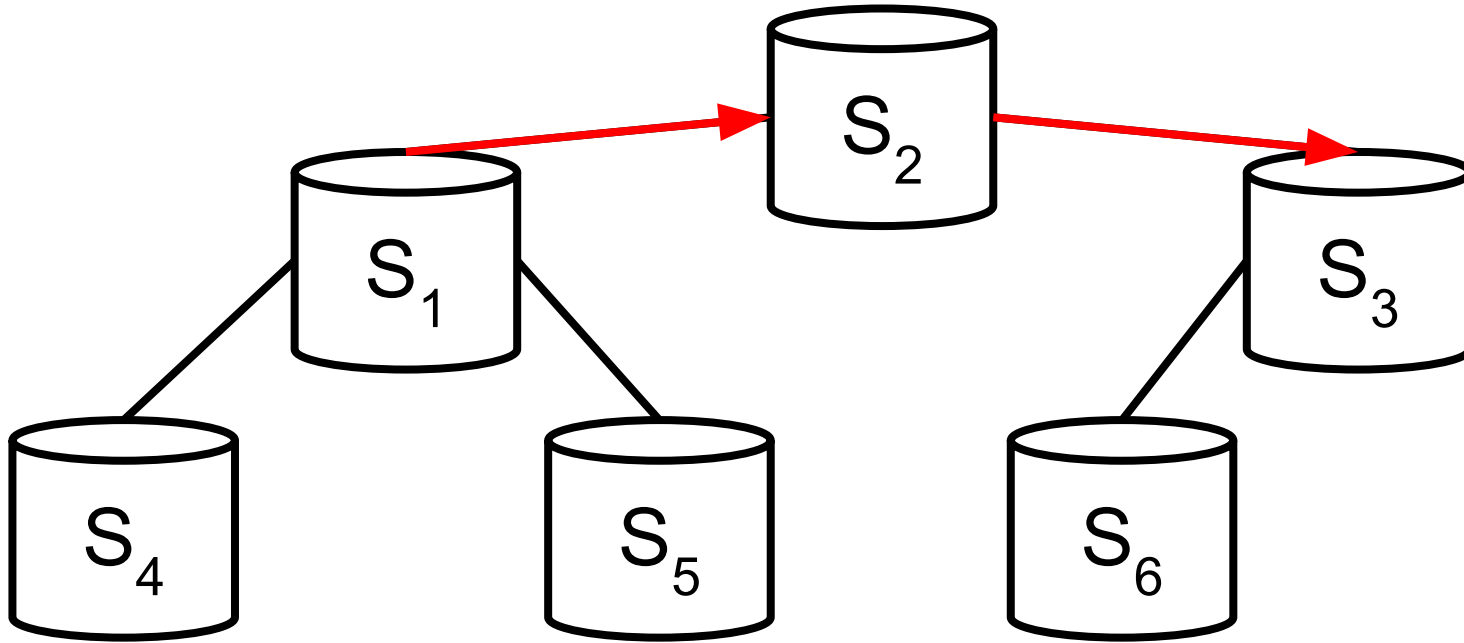
Replication Decisions

- **How many** replicas?
- **Where to place** replicas?
- **How to propagate** updates?

Adaptive Replication

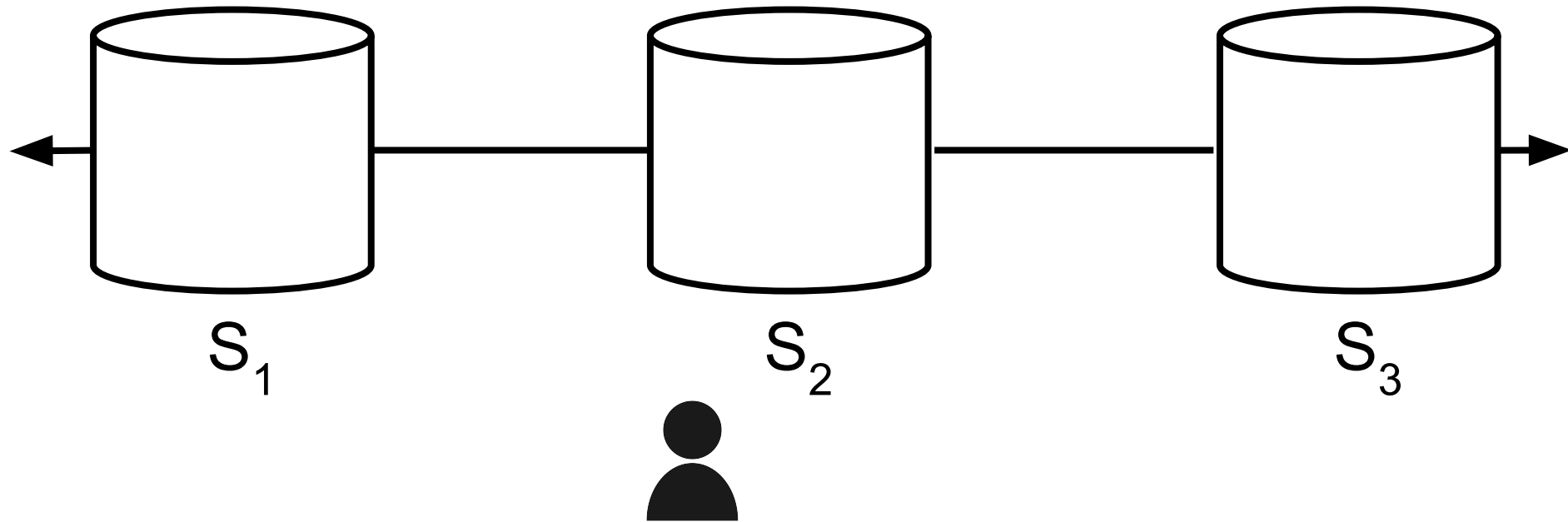
- Decentralized
- Geo-Distributed
- Caching
- Availability

Adaptive Replication (ADR)



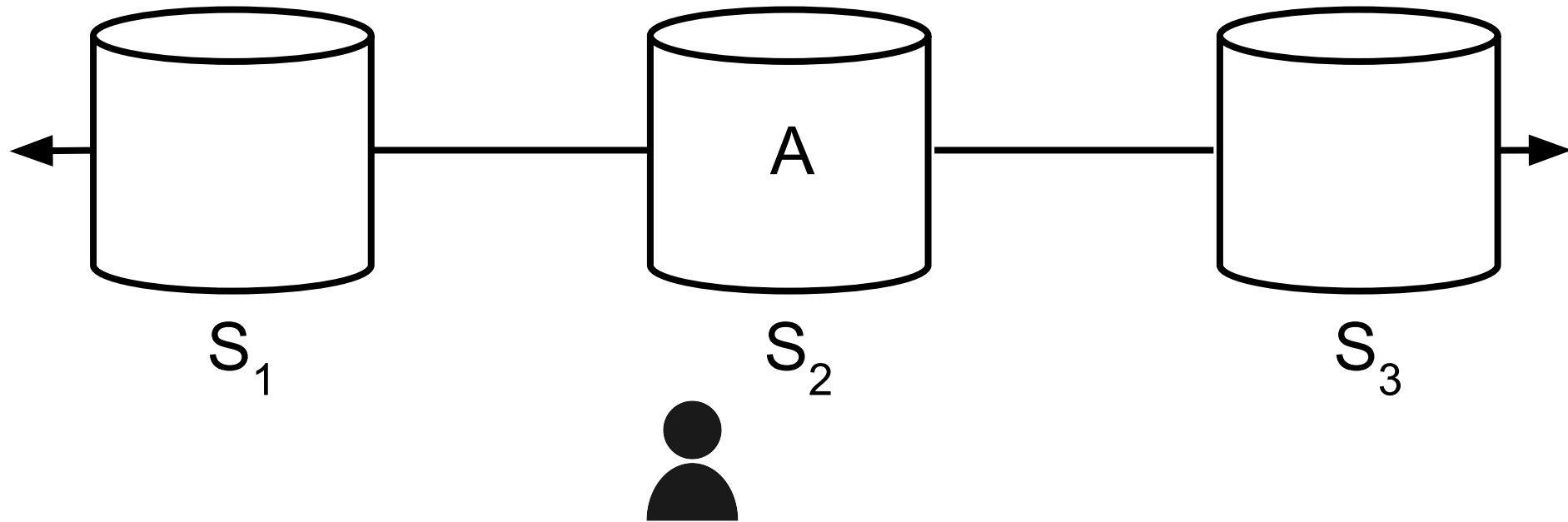
(Wolfson et al., TODS 1997)

Adaptive Replication (ADR)



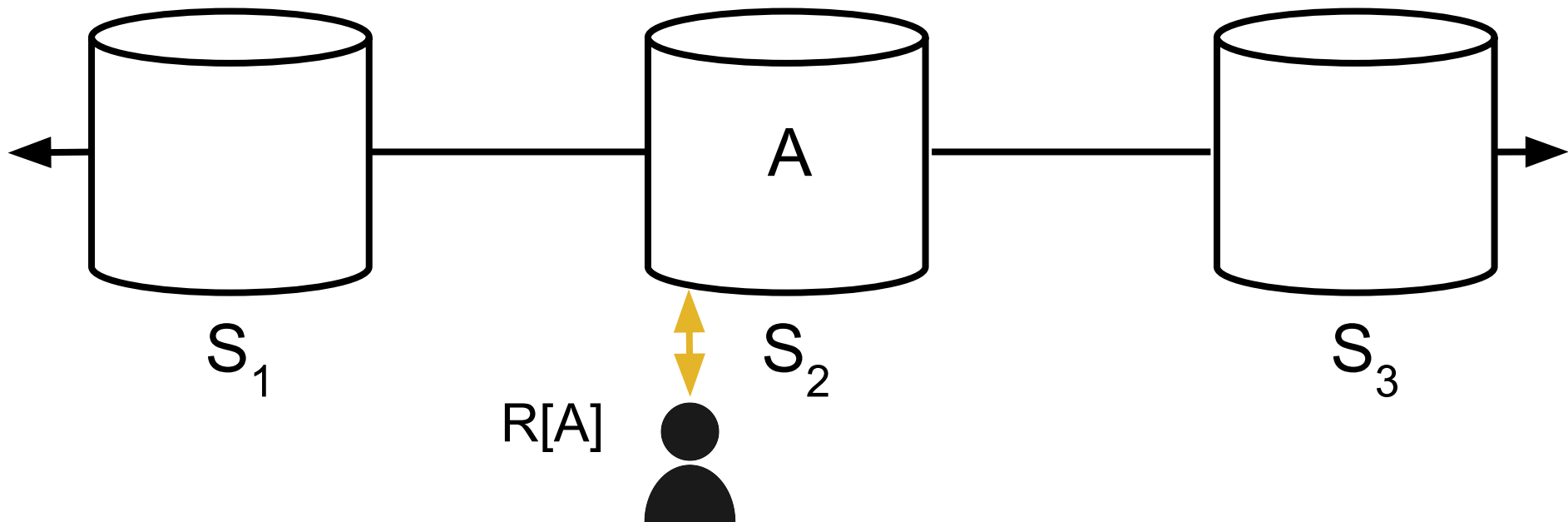
(Wolfson et al., TODS 1997)

Adaptive Replication (ADR)



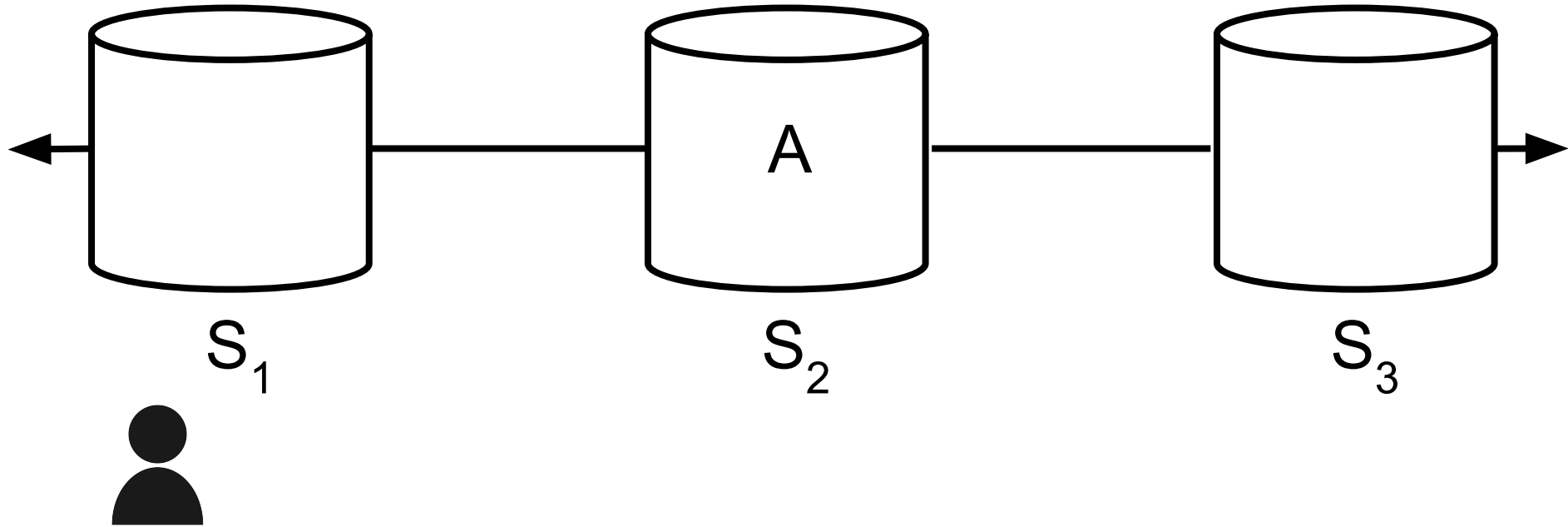
(Wolfson et al., TODS 1997)

Local Read



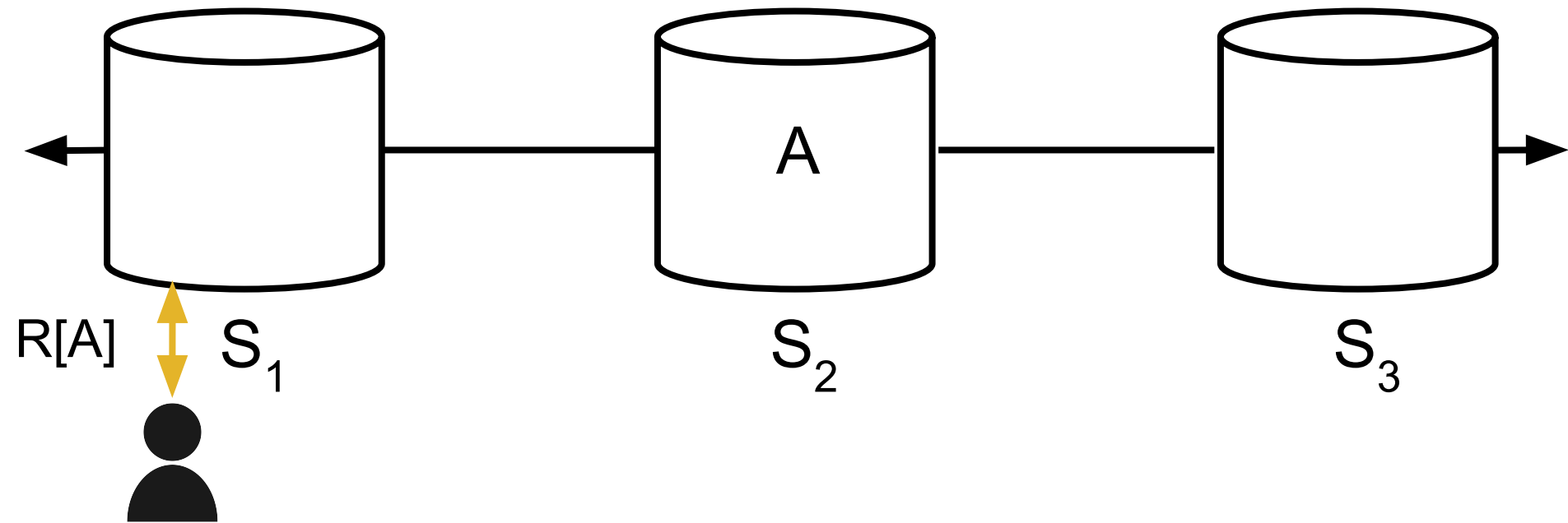
(Wolfson et al., TODS 1997)

Remote Read



(Wolfson et al., TODS 1997)

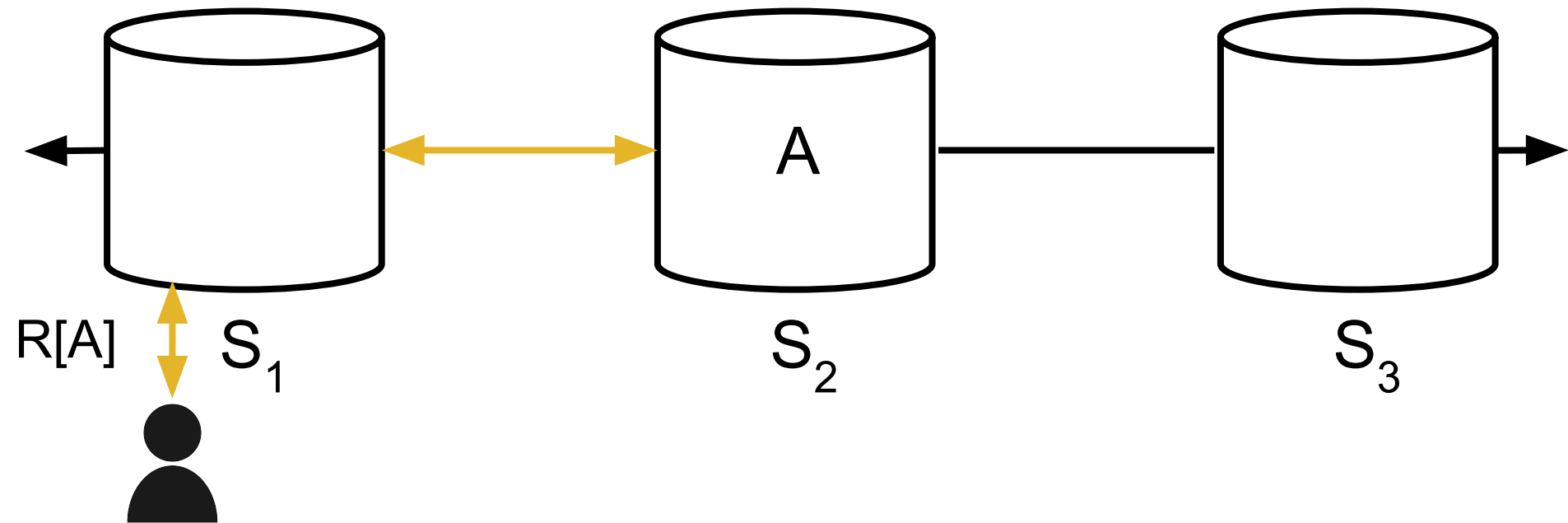
Remote Read



(Wolfson et al., TODS 1997)

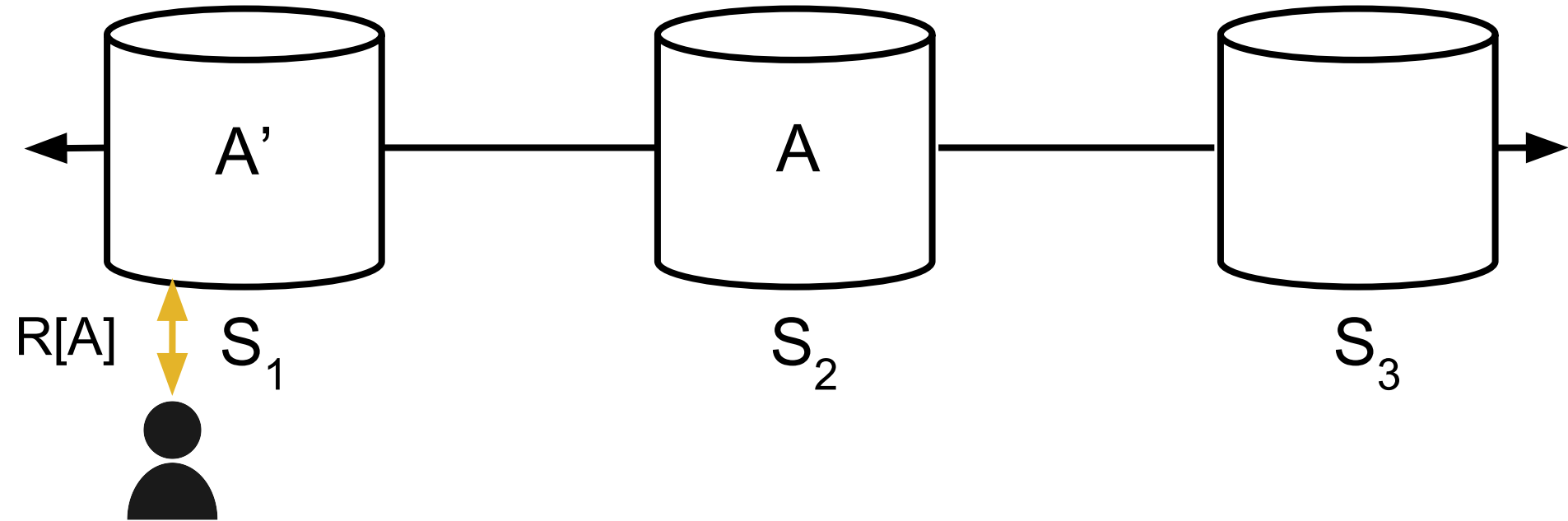
Remote Read

More Messages!



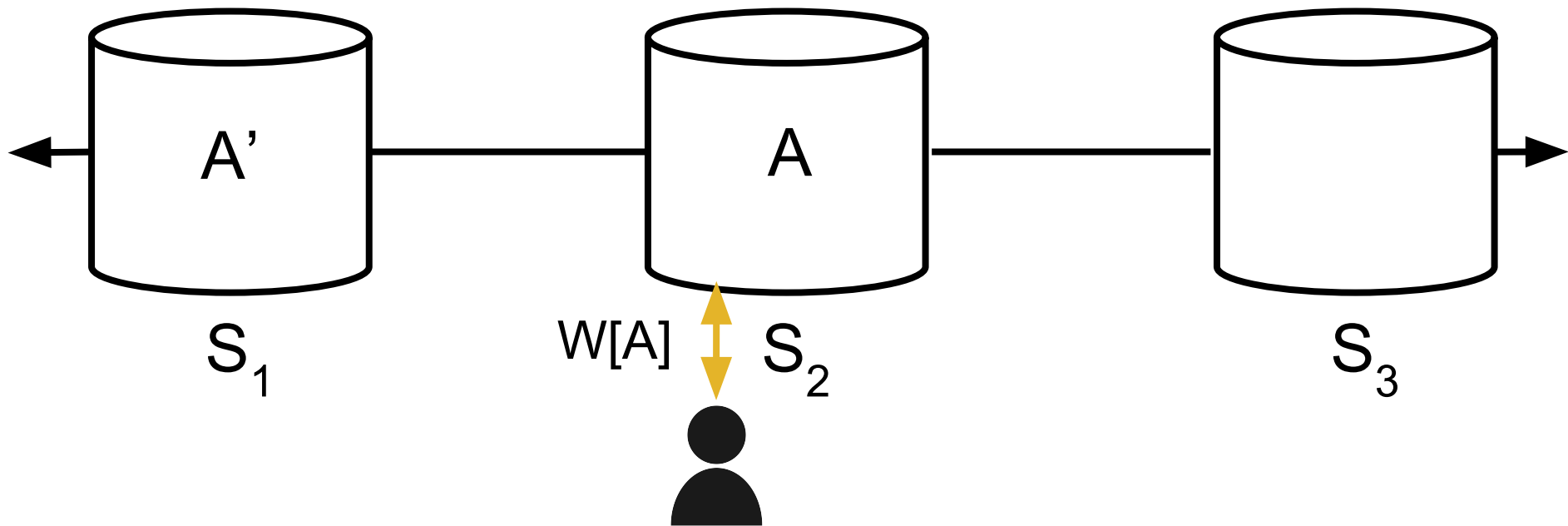
(Wolfson et al., TODS 1997)

Replication for Reads



(Wolfson et al., TODS 1997)

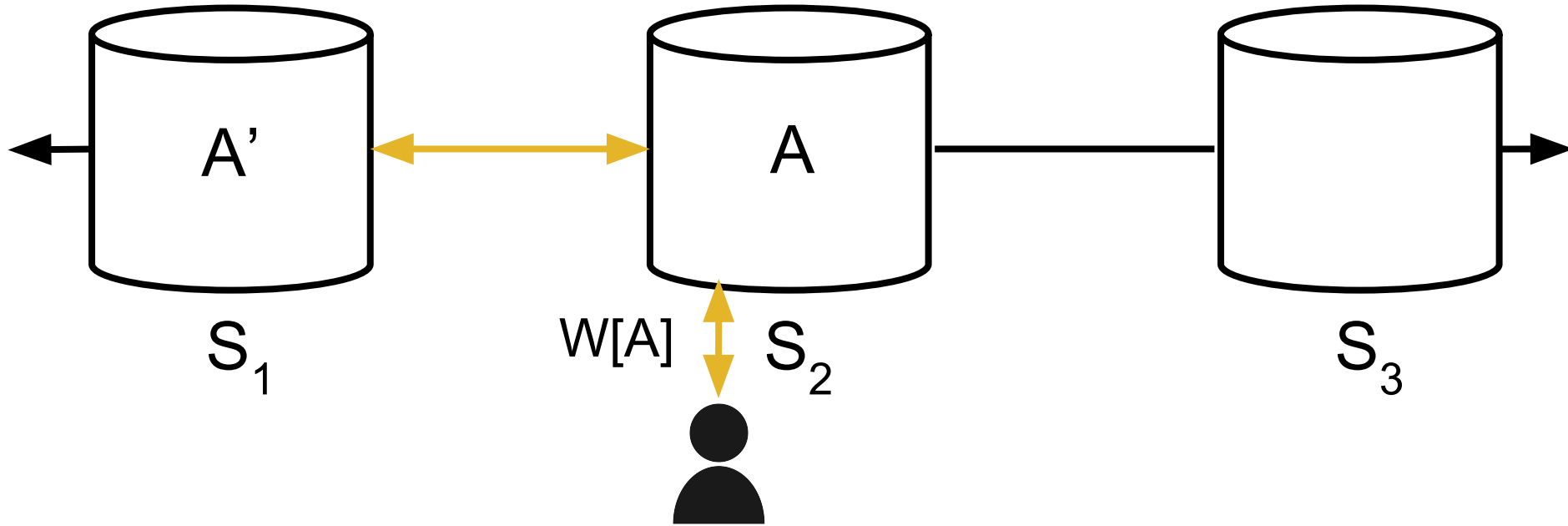
No Free Lunch



(Wolfson et al., TODS 1997)

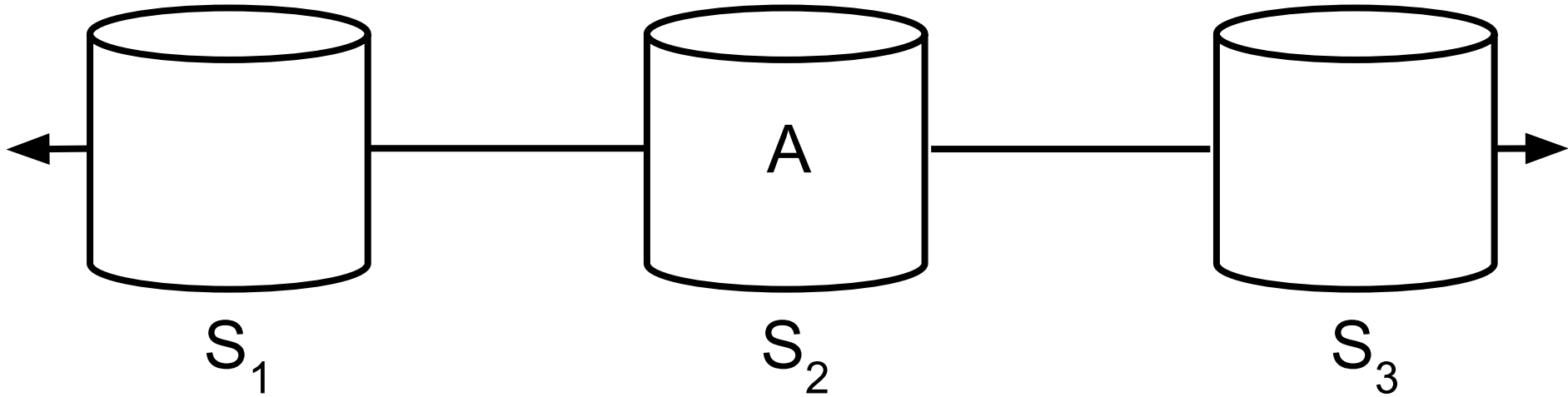
No Free Lunch

Reduces read cost,
Increases write cost



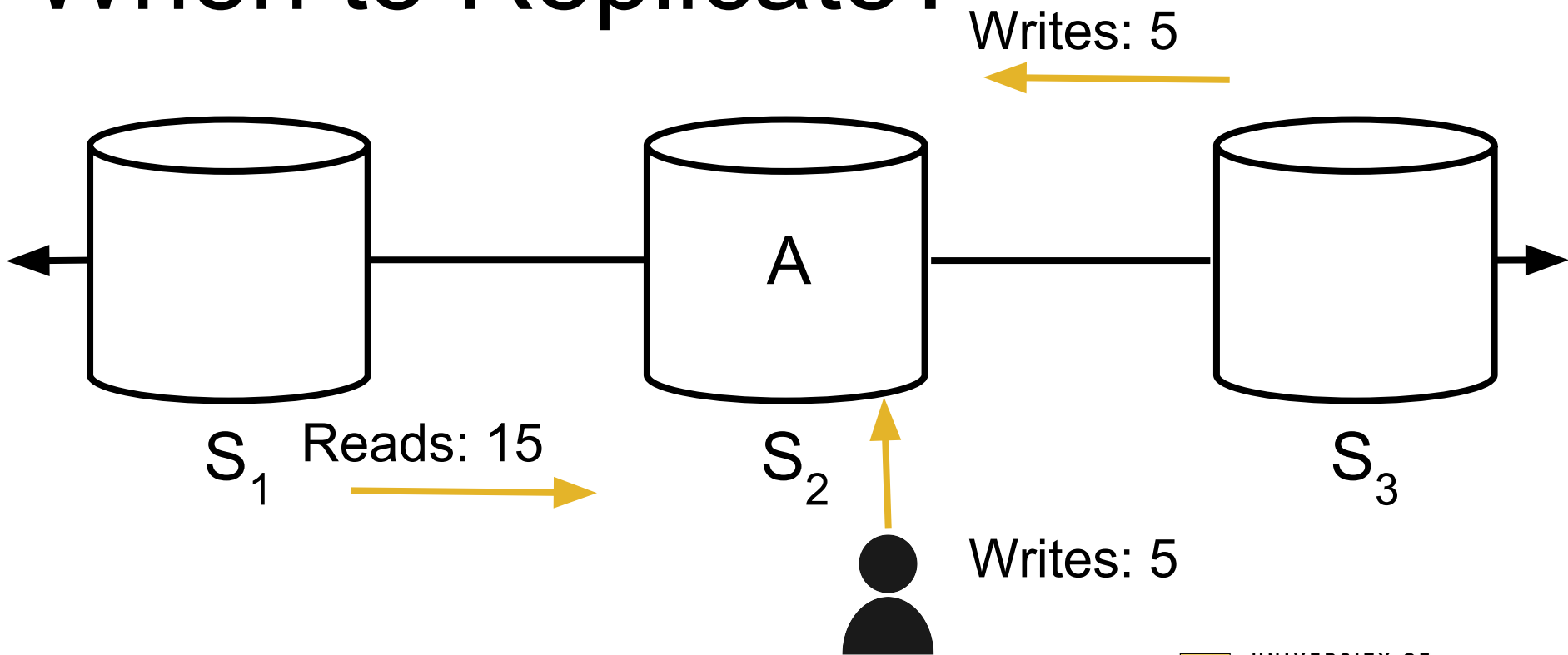
(Wolfson et al., TODS 1997)

When to Replicate?



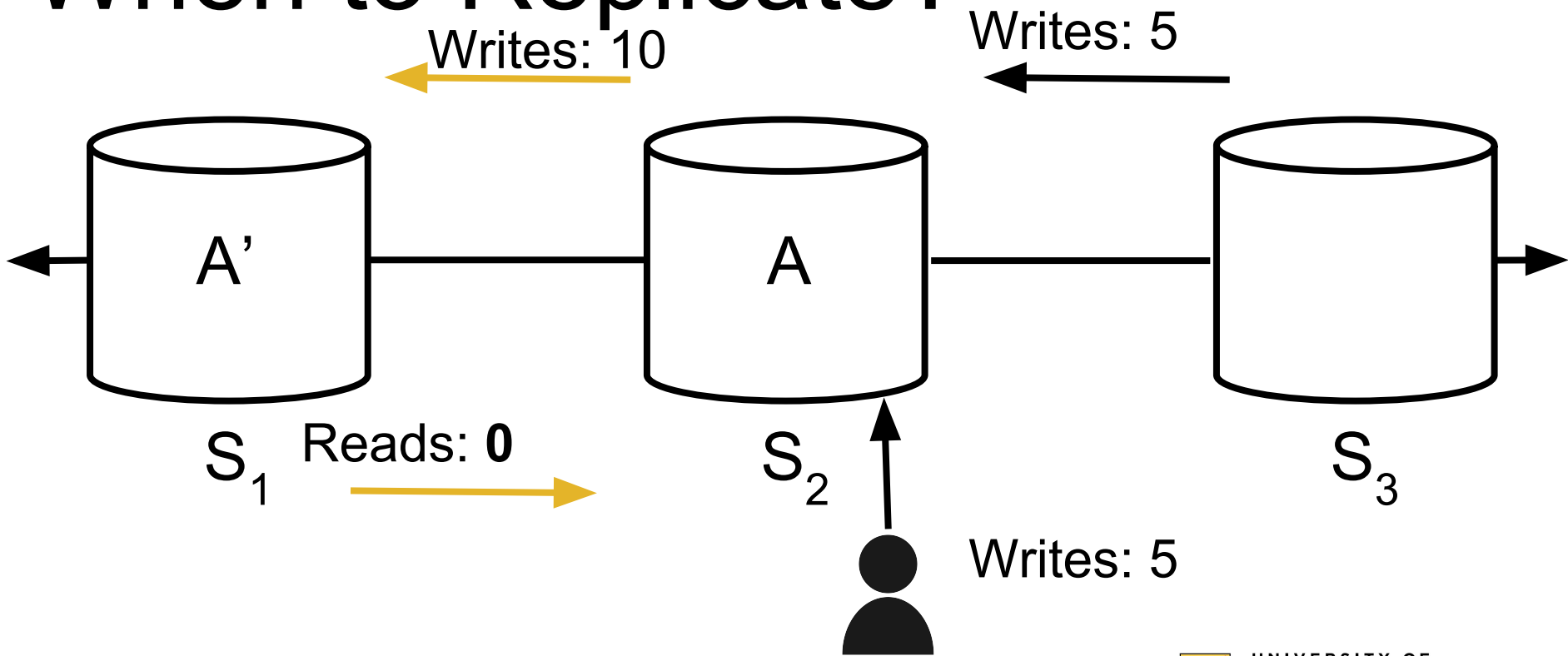
(Wolfson et al., TODS 1997)

When to Replicate?



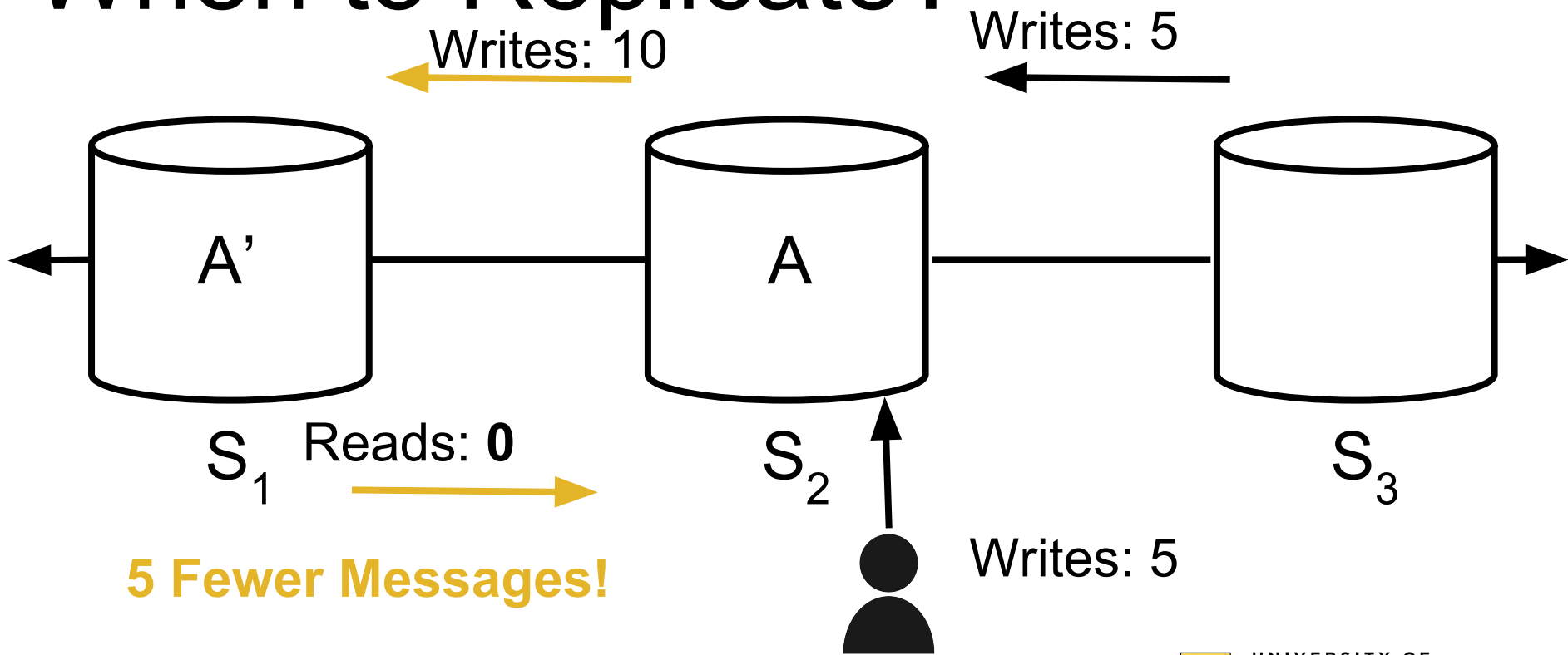
(Wolfson et al., TODS 1997)

When to Replicate?



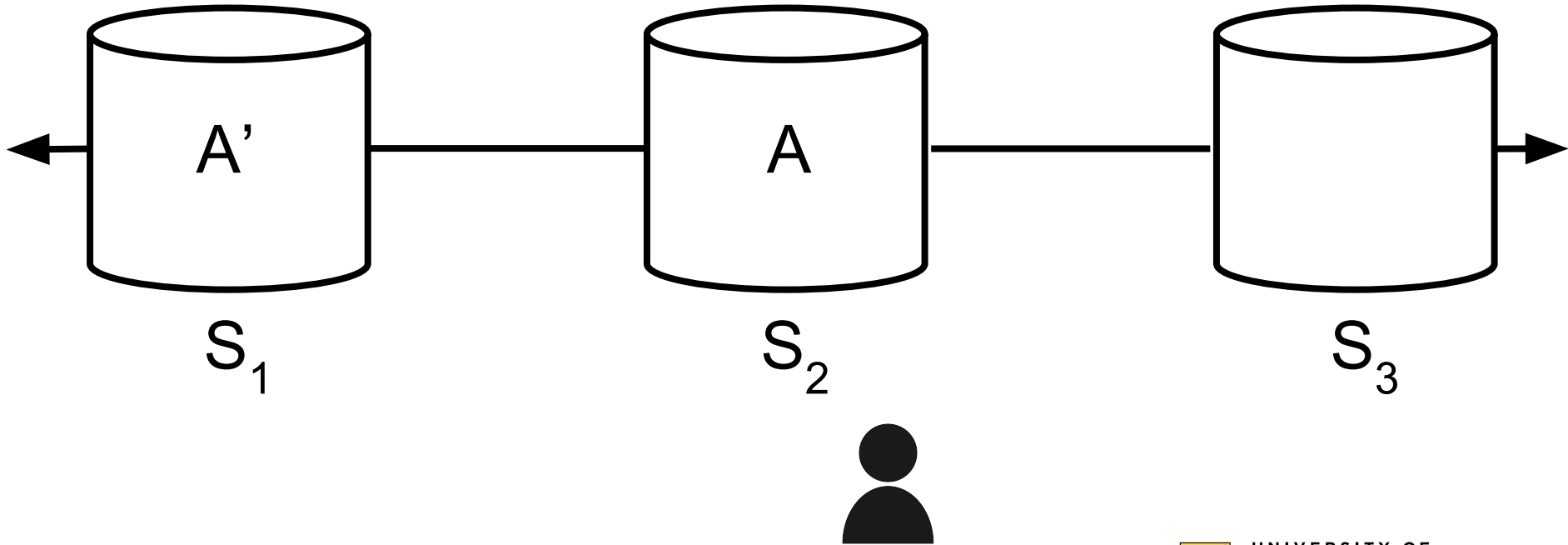
(Wolfson et al., TODS 1997)

When to Replicate?



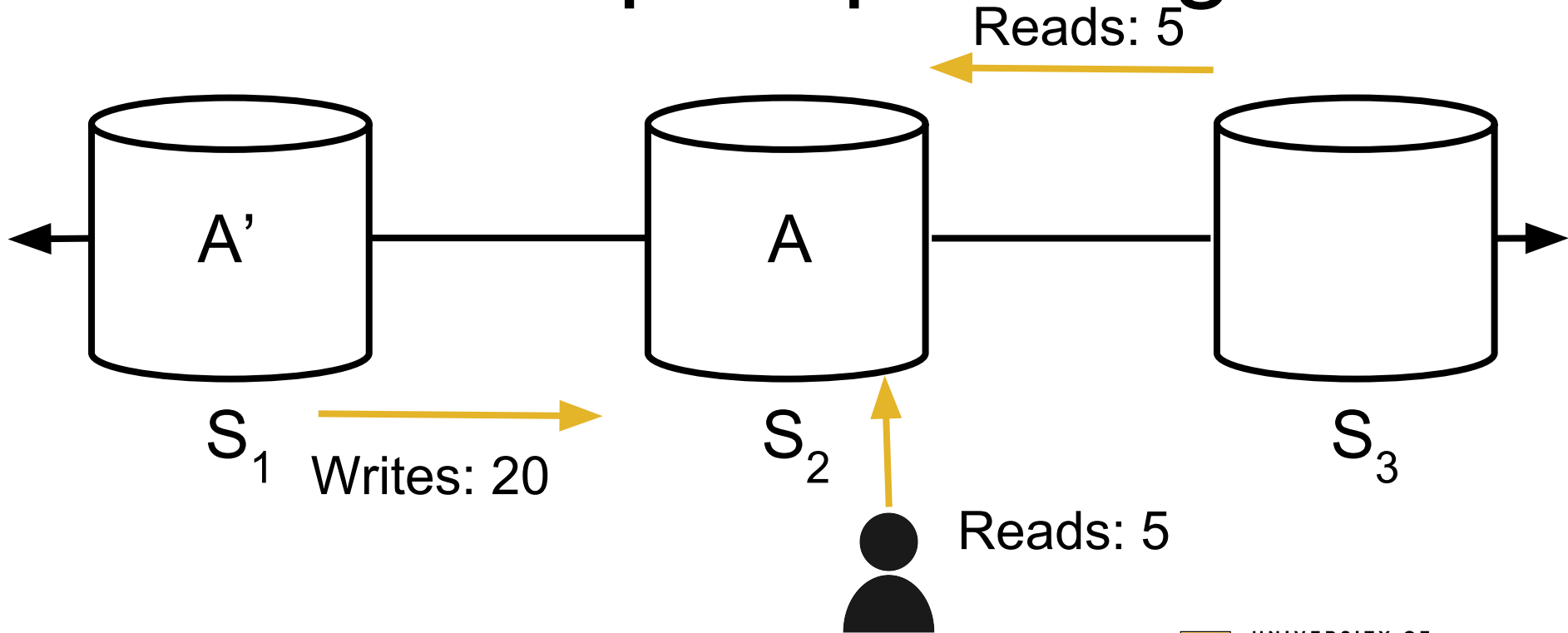
(Wolfson et al., TODS 1997)

When to Stop Replicating?



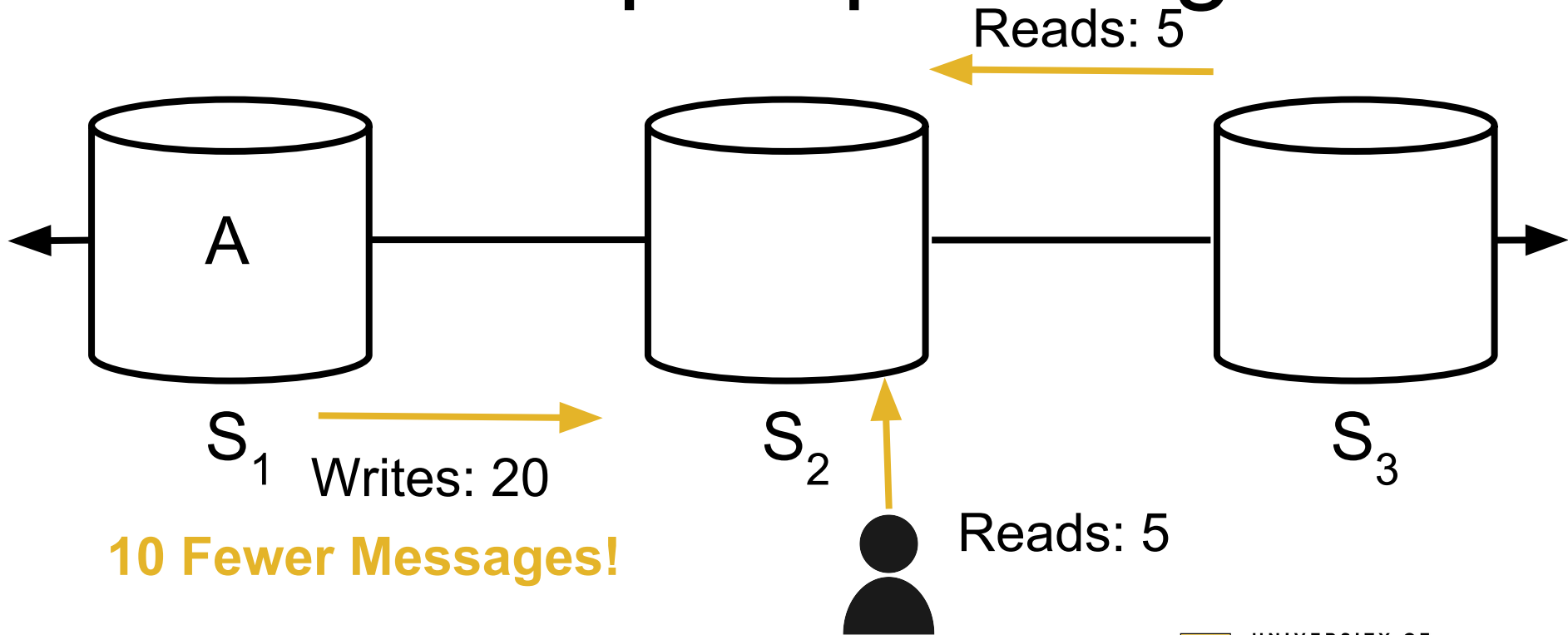
(Wolfson et al., TODS 1997)

When to Stop Replicating?



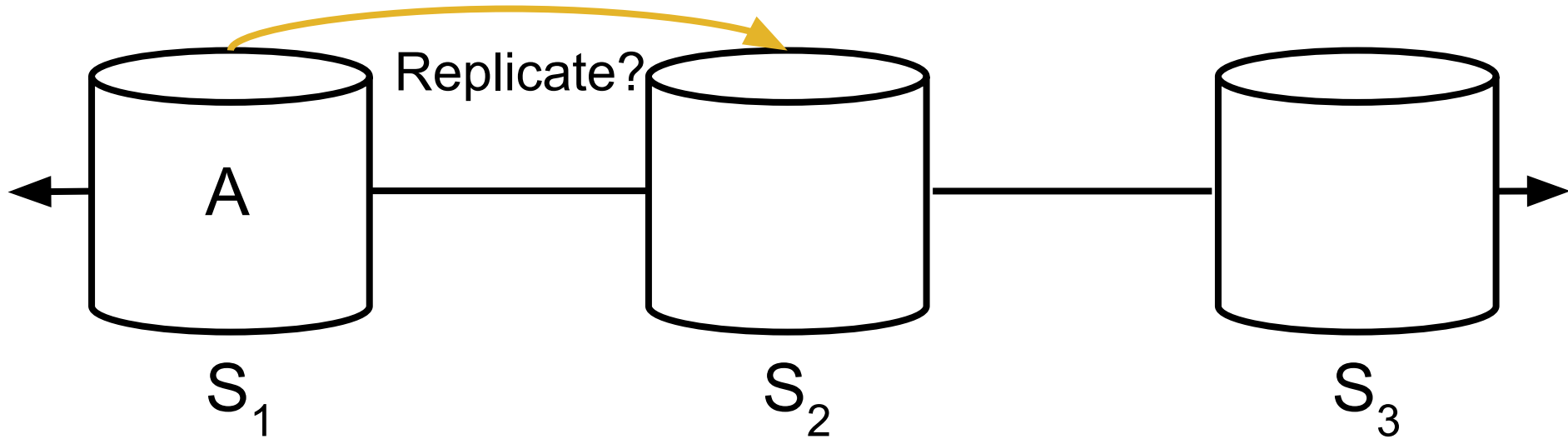
(Wolfson et al., TODS 1997)

When to Stop Replicating?



(Wolfson et al., TODS 1997)

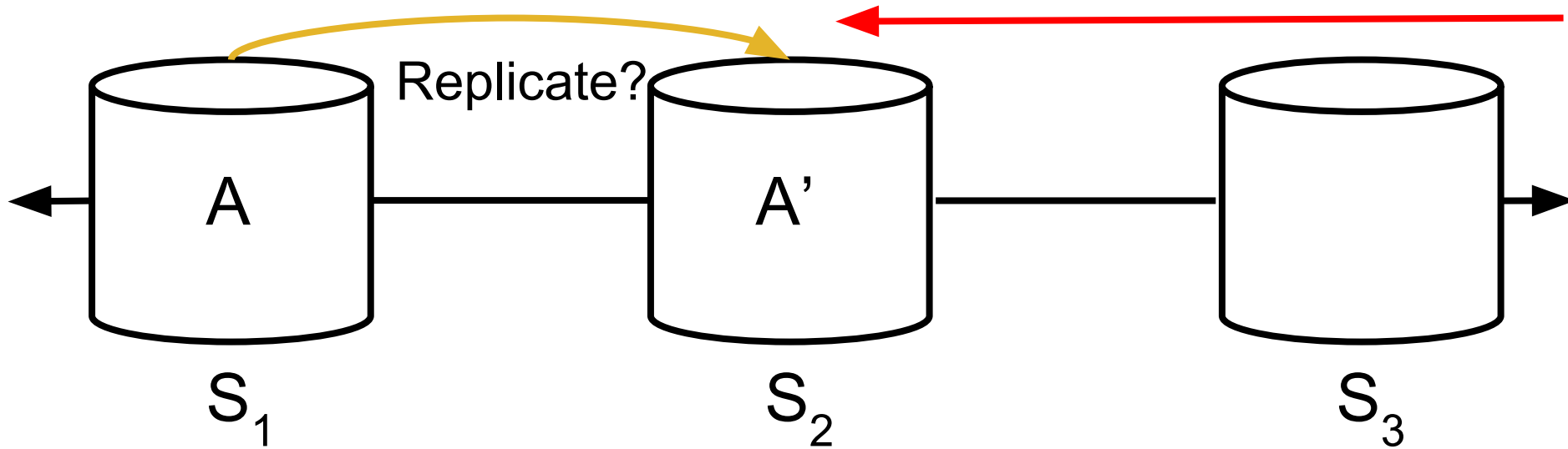
Decentralized Decisions



(Wolfson et al., TODS 1997)

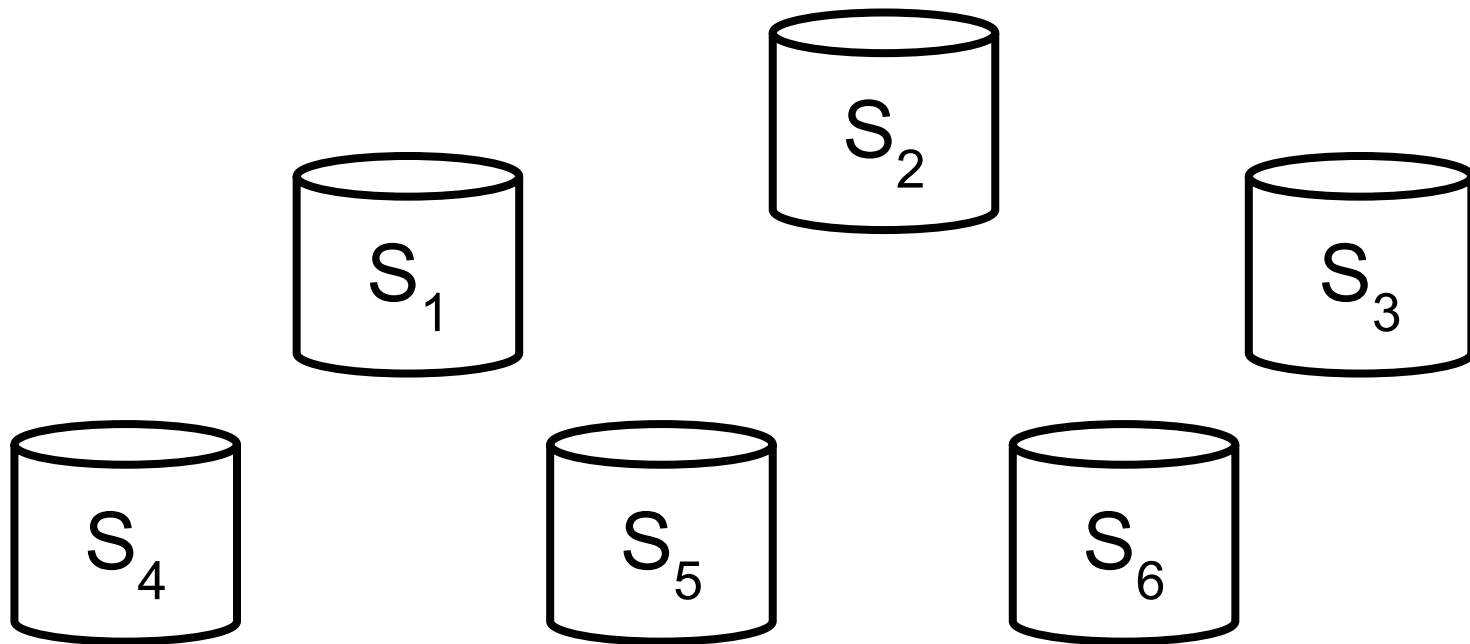
Adapts to Changing Workloads

More Reads



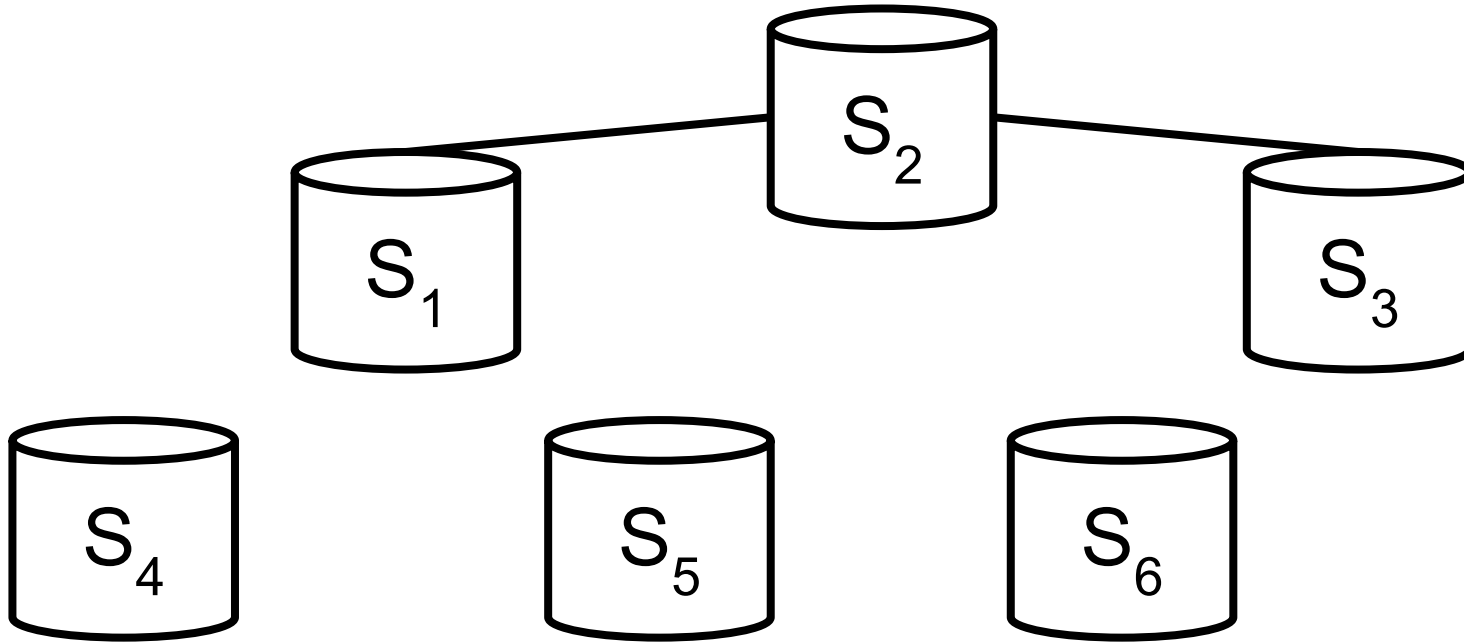
(Wolfson et al., TODS 1997)

Extensions (Network Topology)



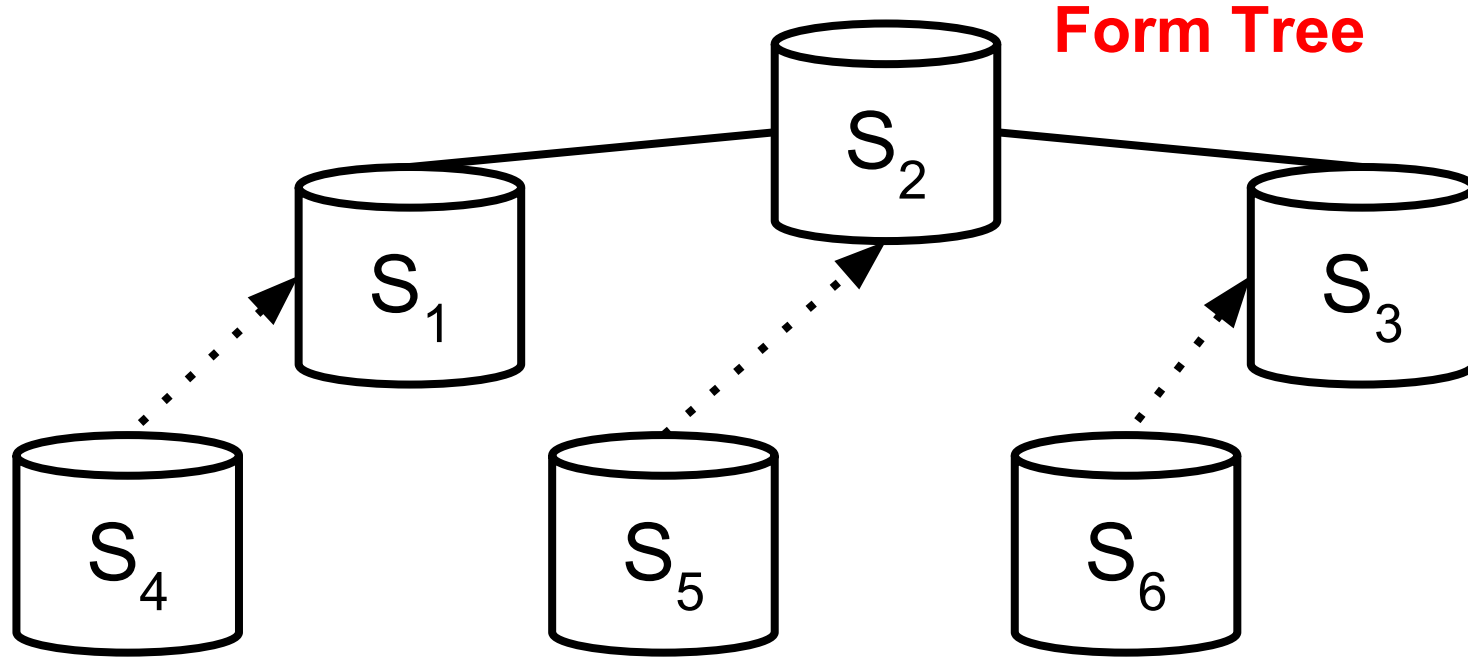
(Wolfson et al., TODS 1997)

Extensions (Network Topology)



(Wolfson et al., TODS 1997)

Extensions (Network Topology)

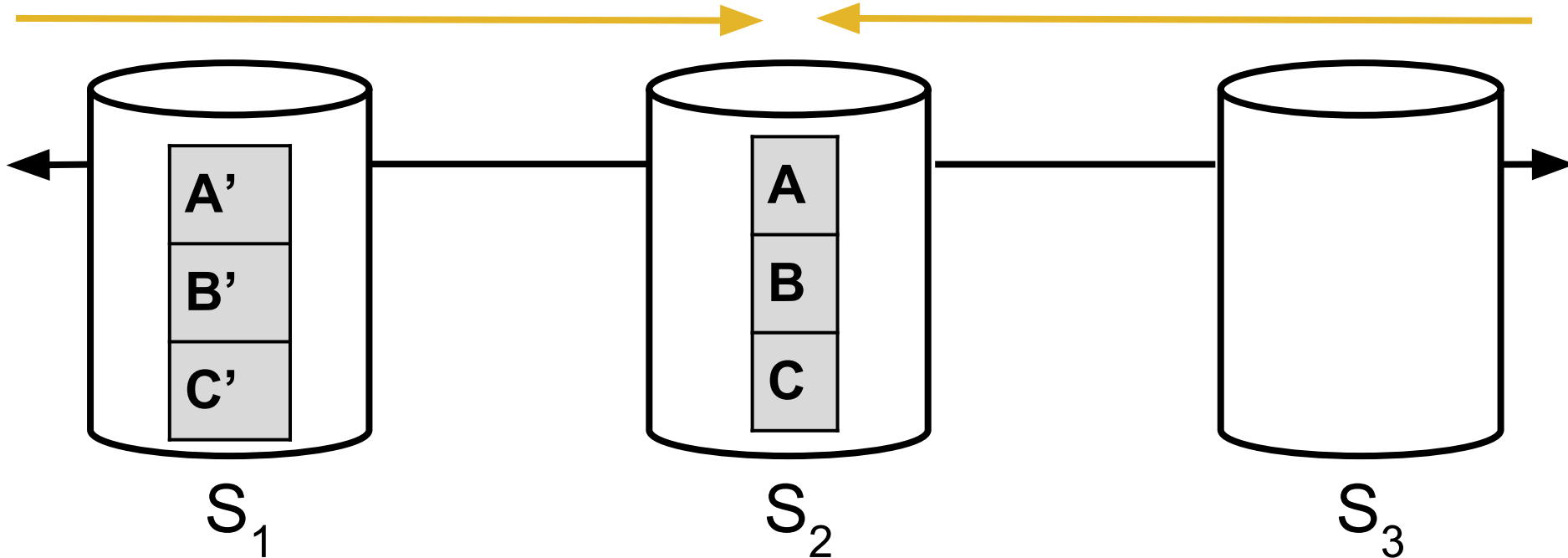


(Wolfson et al., TODS 1997)

Extensions (Blocks)

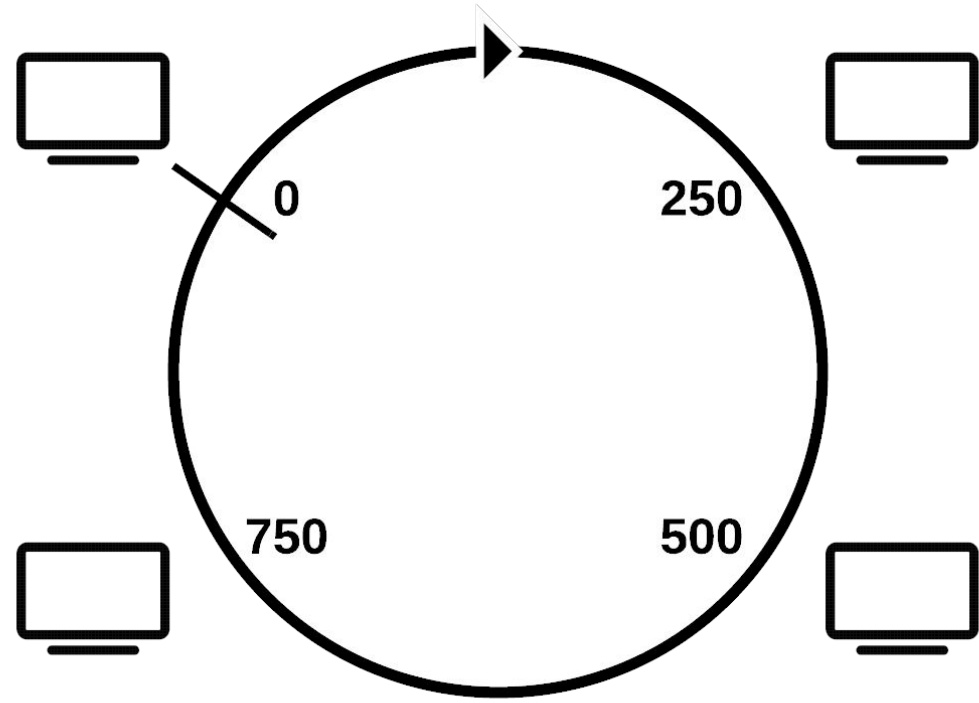
$$\frac{1}{3} R[A] + \frac{1}{3} R[B] + \frac{1}{3} R[C]$$

$$\frac{1}{3} W[A] + \frac{1}{3} W[B] + \frac{1}{3} W[C]$$

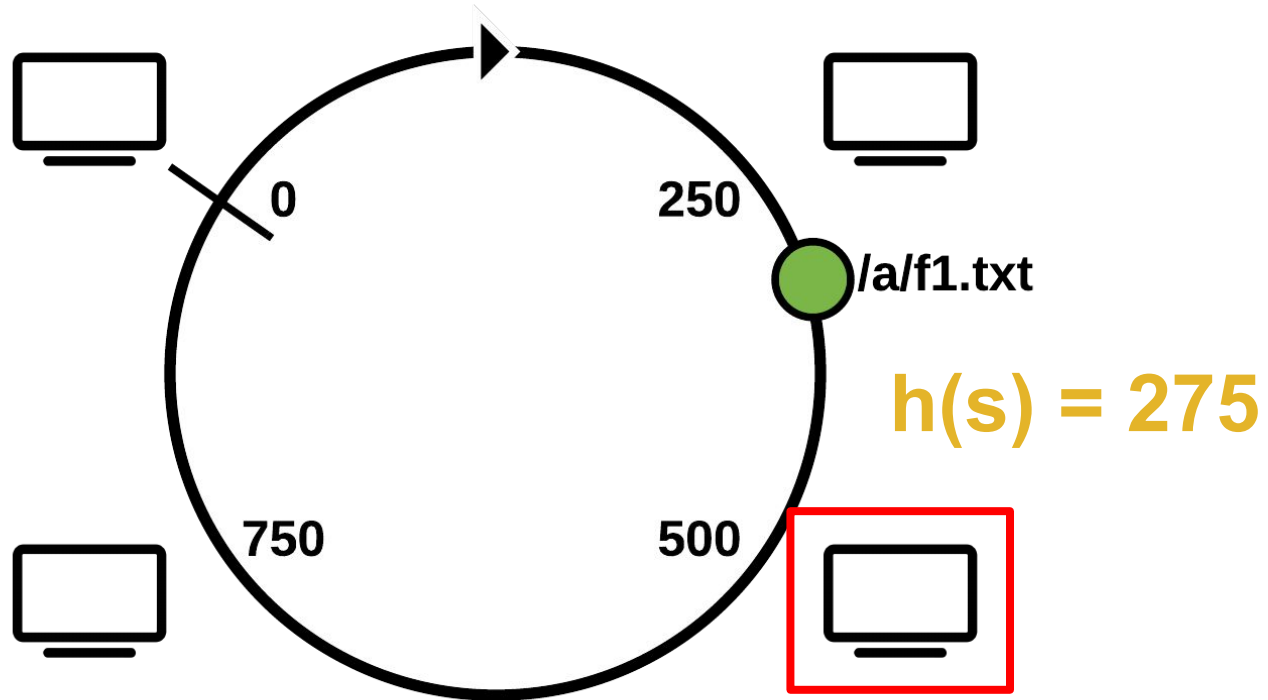


(Wolfson et al., TODS 1997)

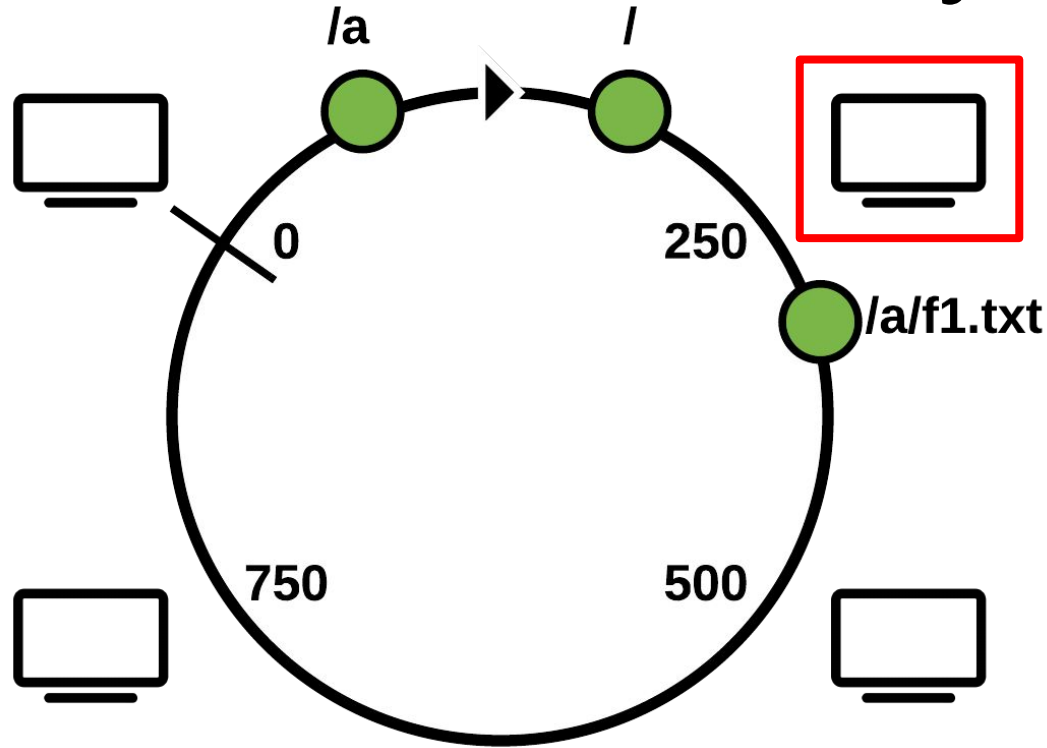
Peer-to-Peer File Systems



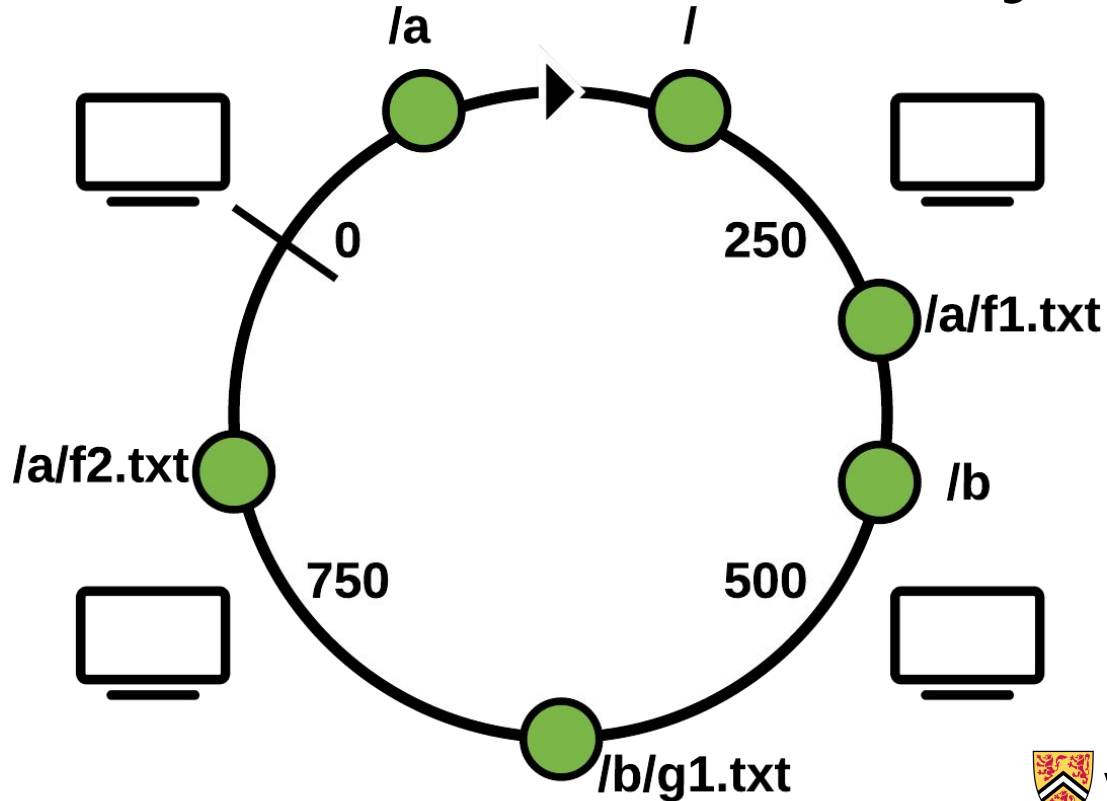
Hash-Based P2P File System



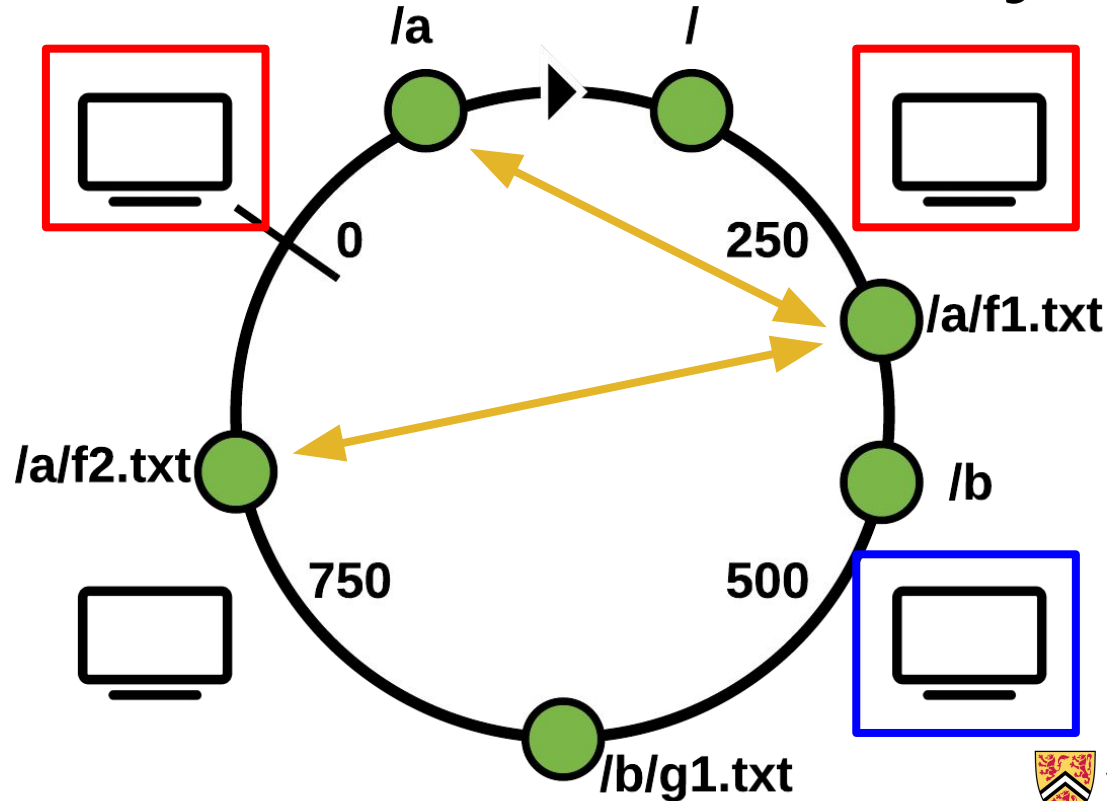
Hash-Based P2P File System



Hash-Based P2P File System

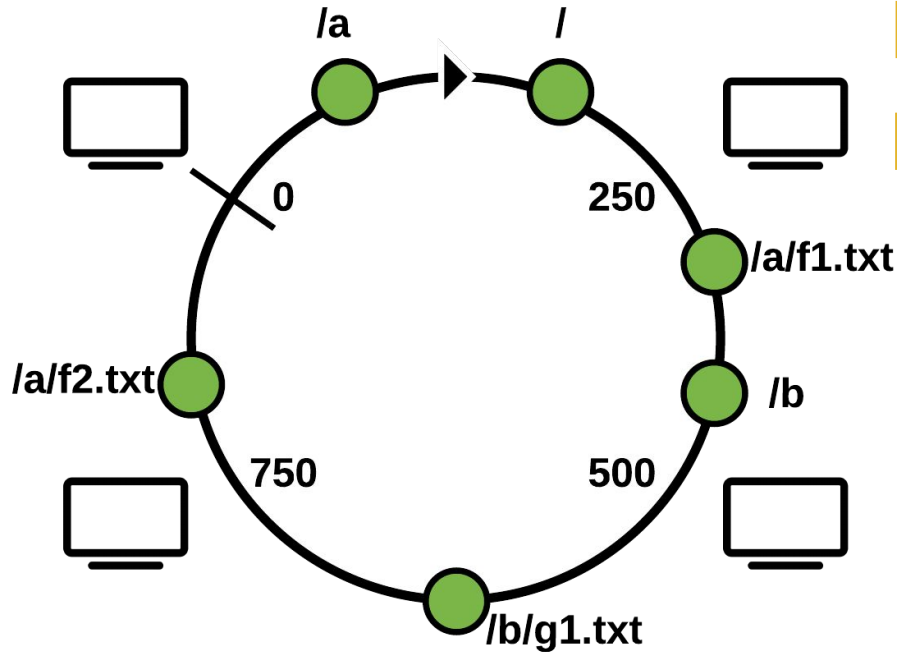


Hash-Based P2P File System

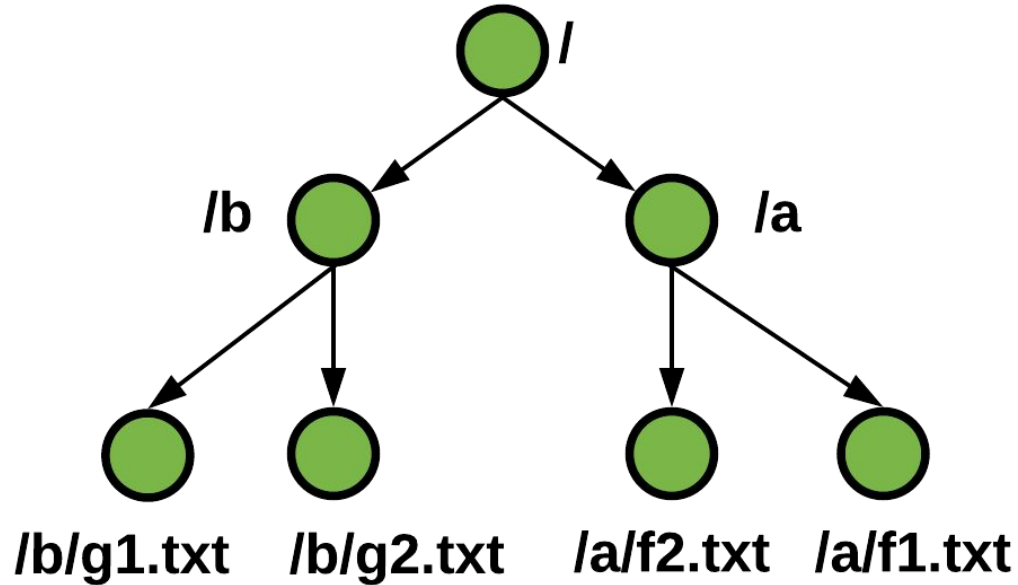


Hash-Based P2P FS

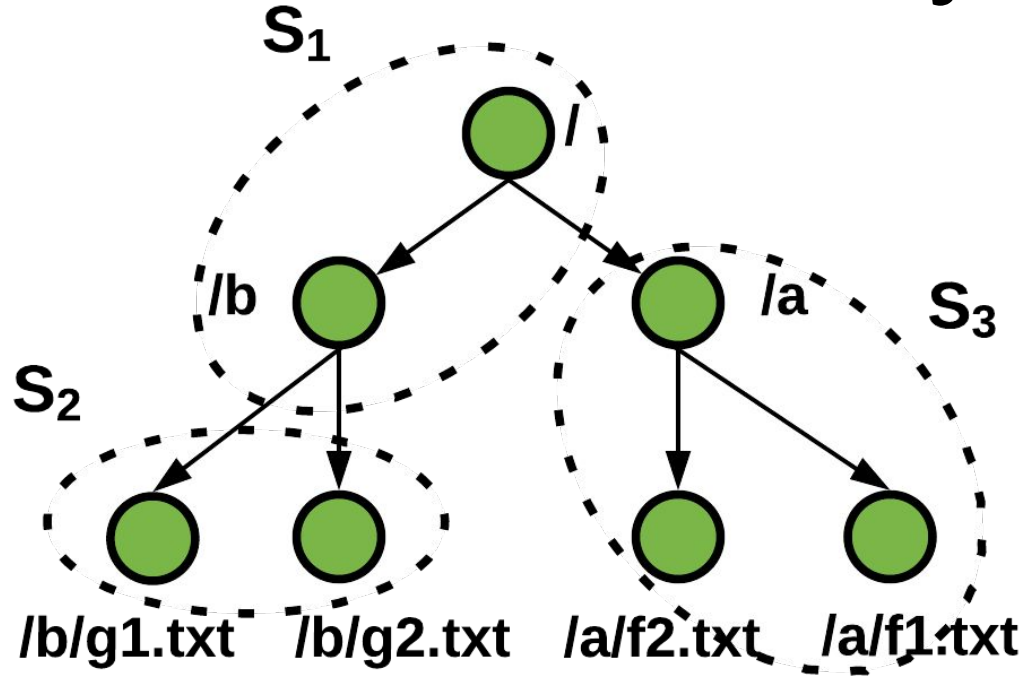
Balanced load, but
poor access locality!



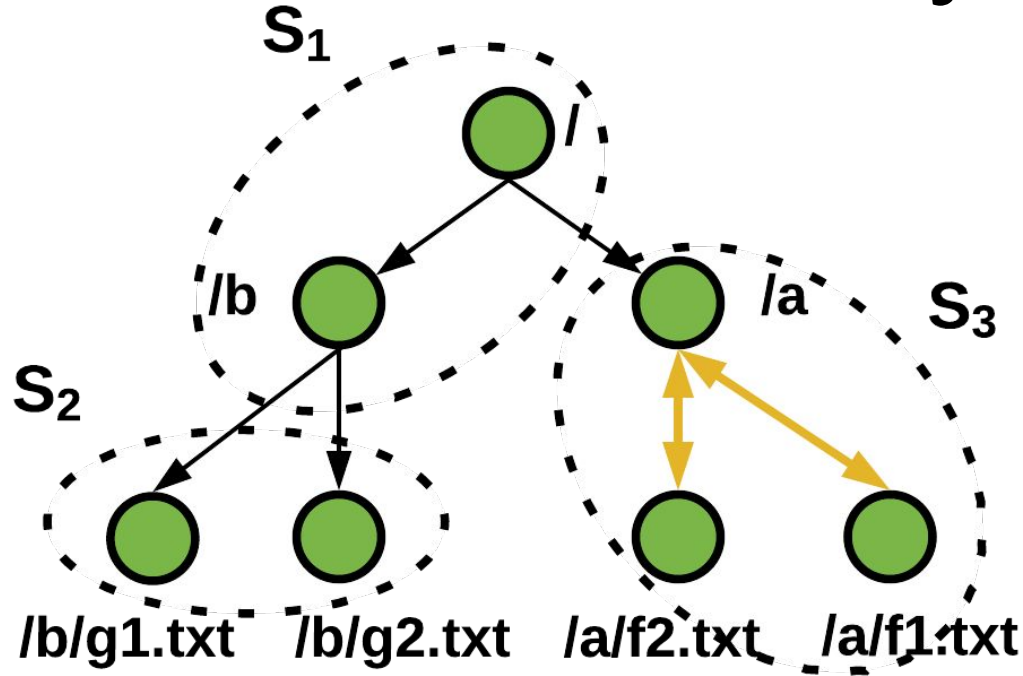
Hierarchical P2P File Systems



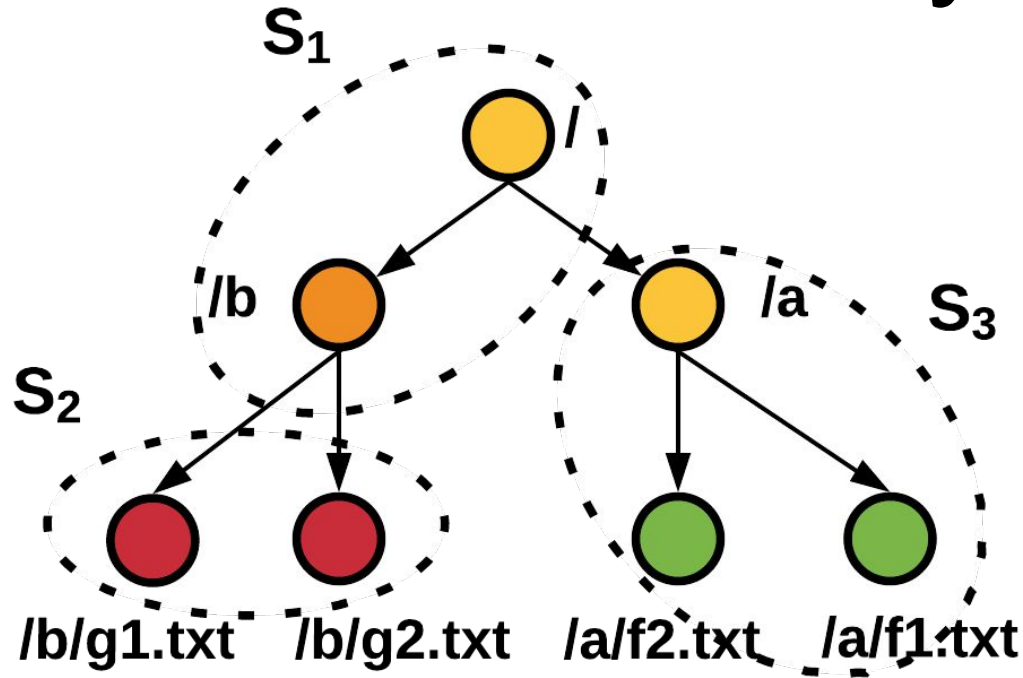
Hierarchical P2P File Systems



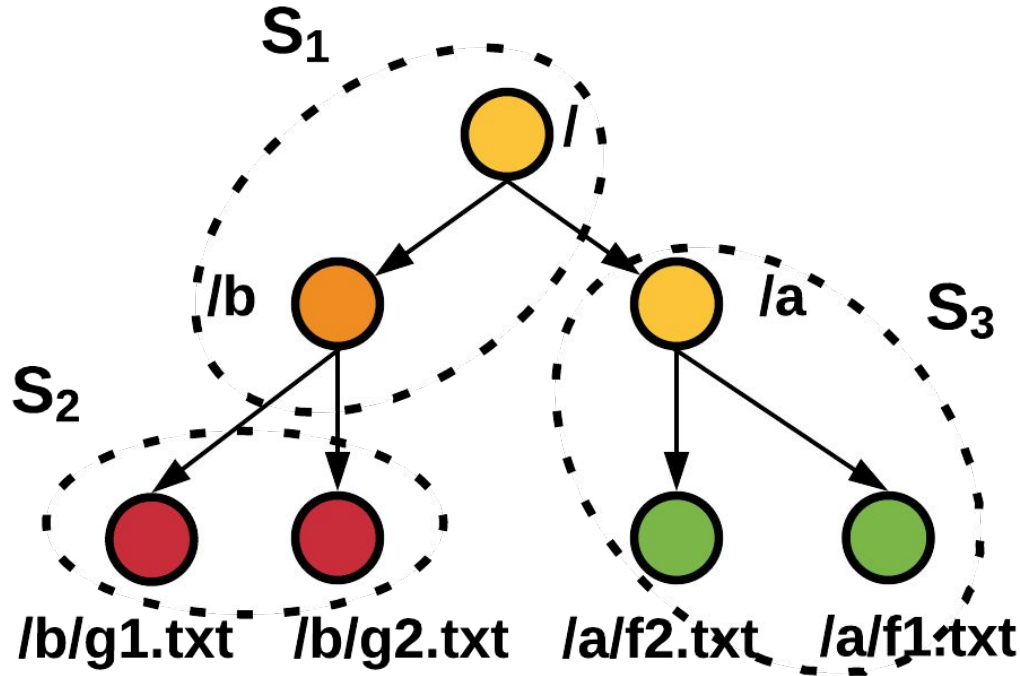
Hierarchical P2P File Systems



Hierarchical P2P File Systems

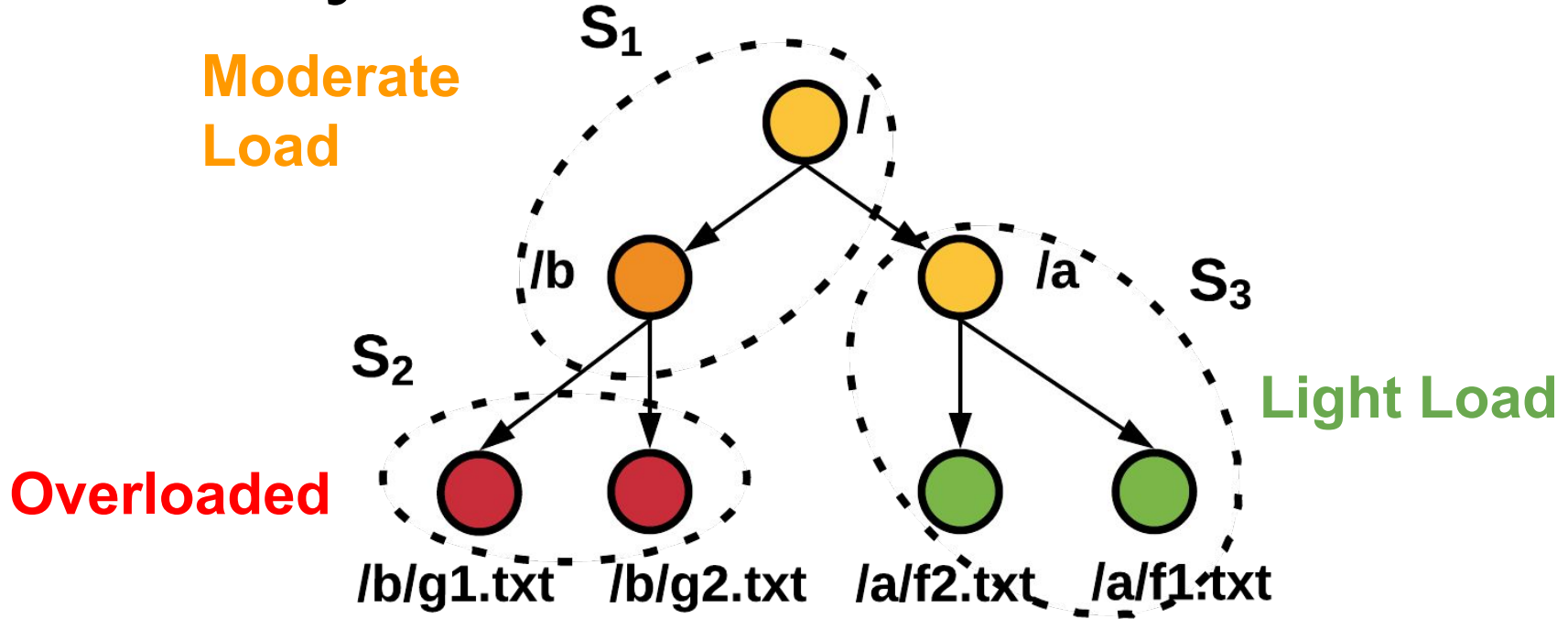


Hierarchical P2P File System

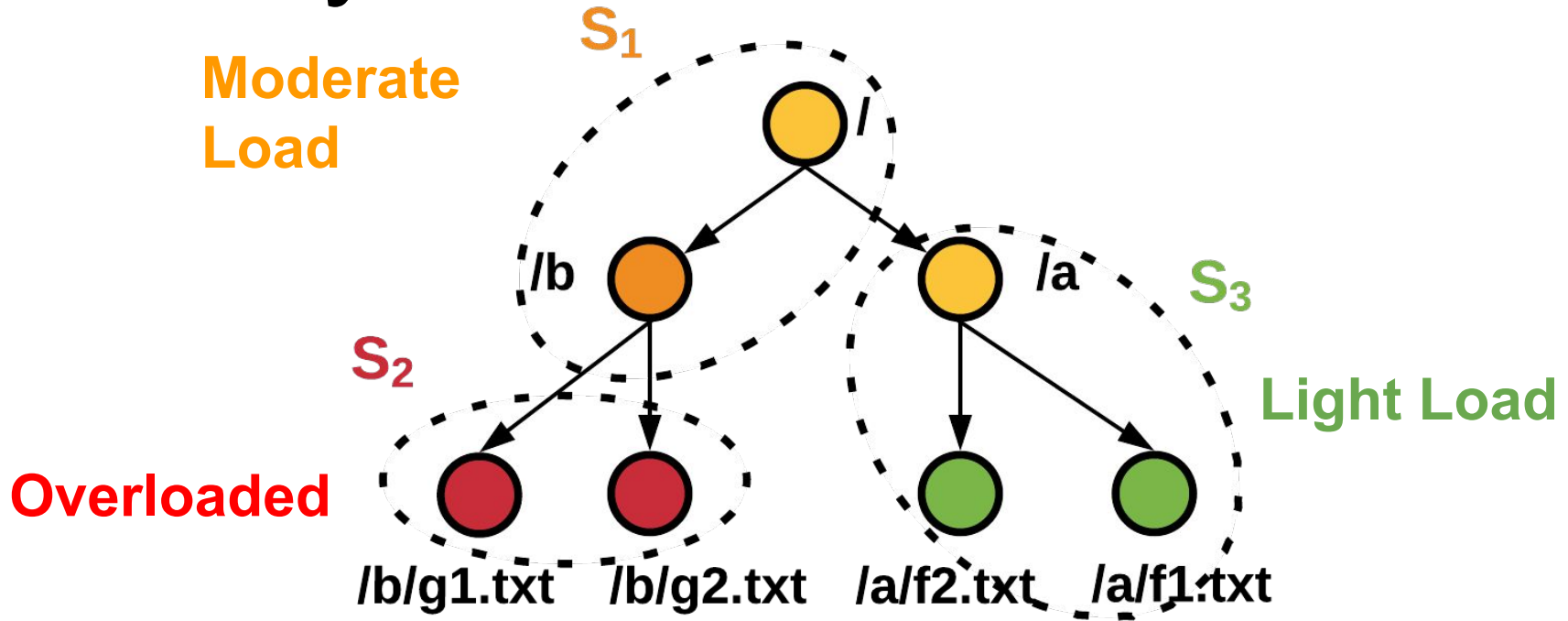


**Good locality,
but poor
balance**

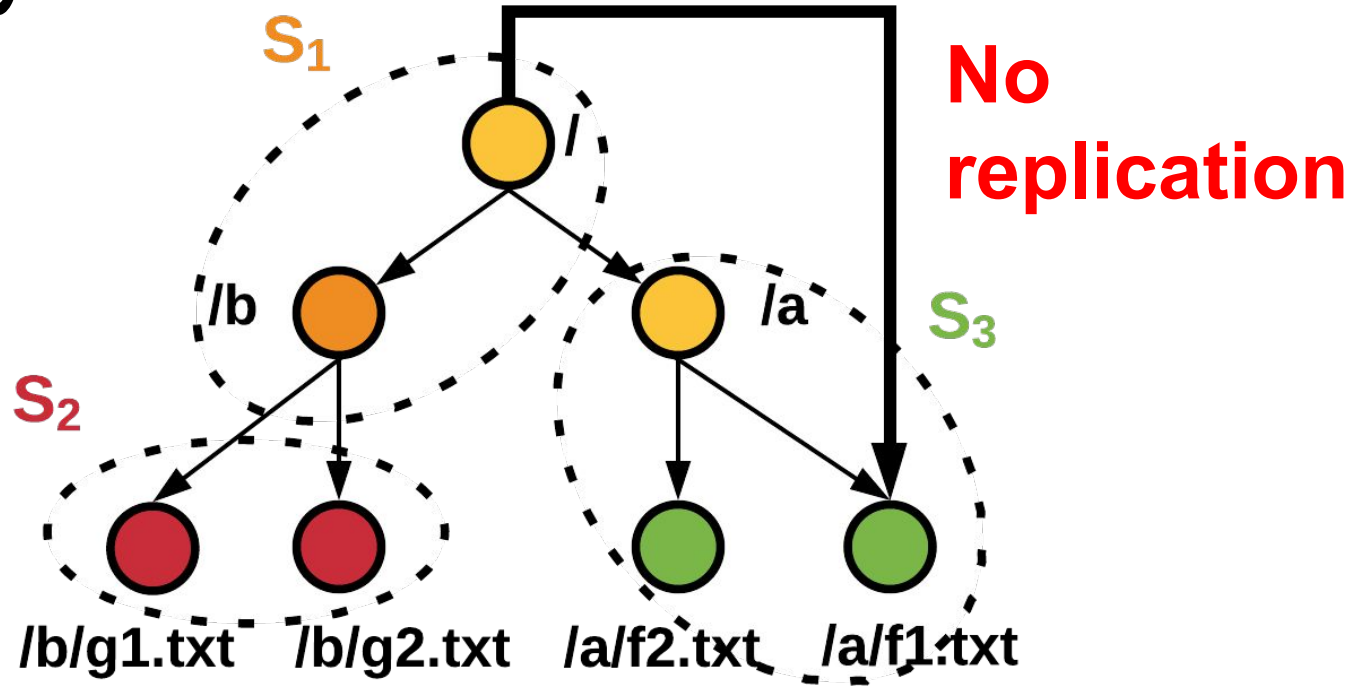
Locality and Load Balance



Locality and Load Balance

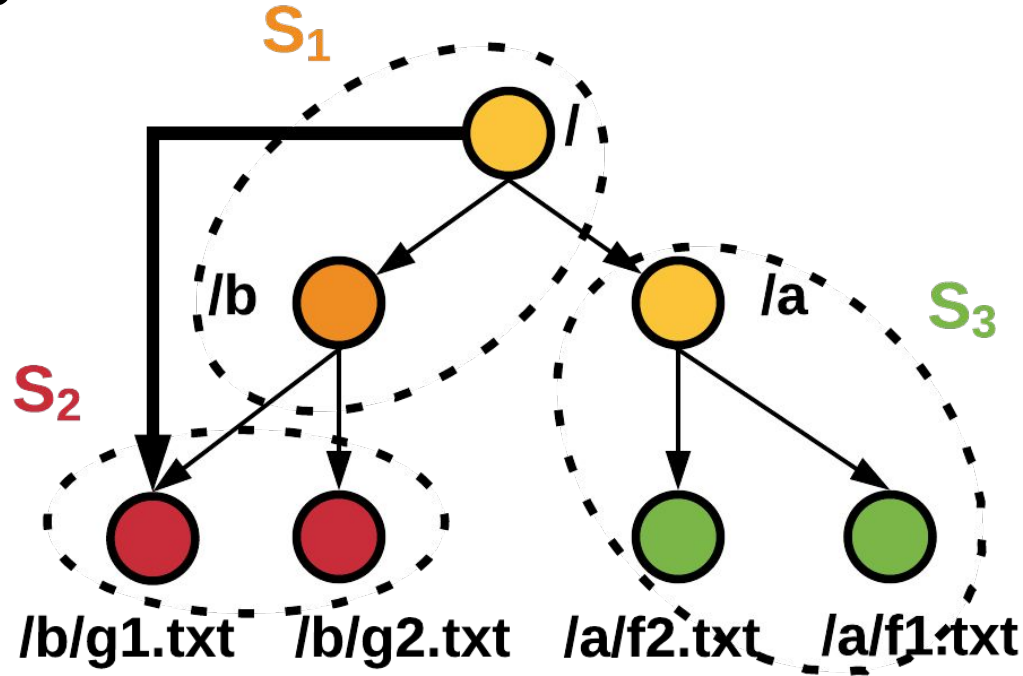


Locality and Load Balance

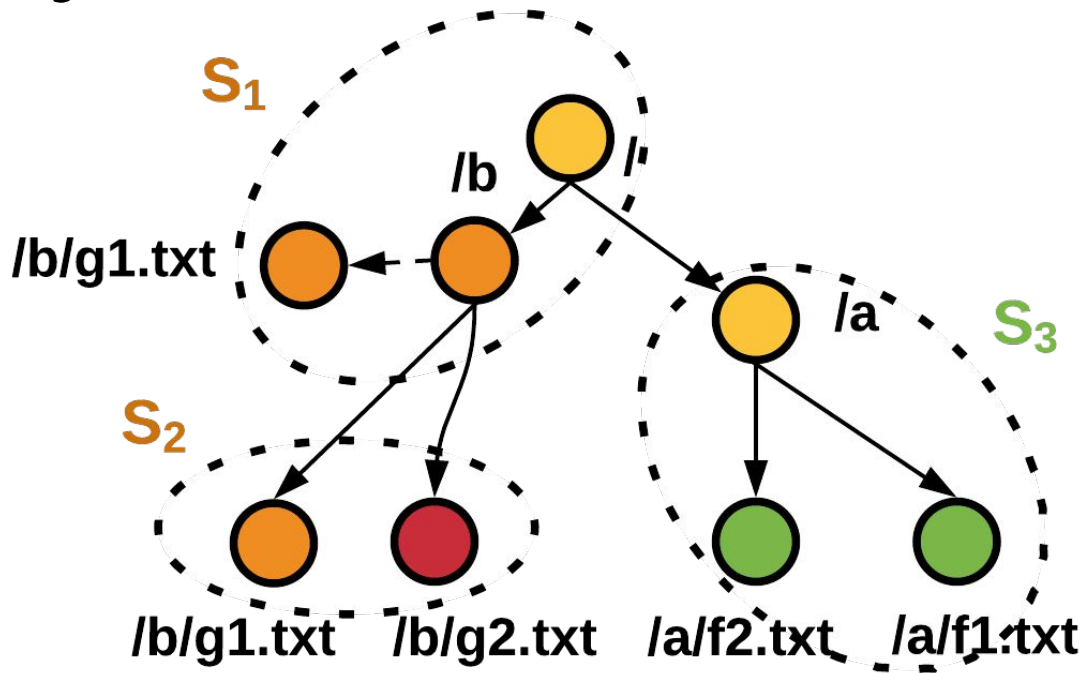


Locality and Load Balance

**Replicate
to
balance
load!**

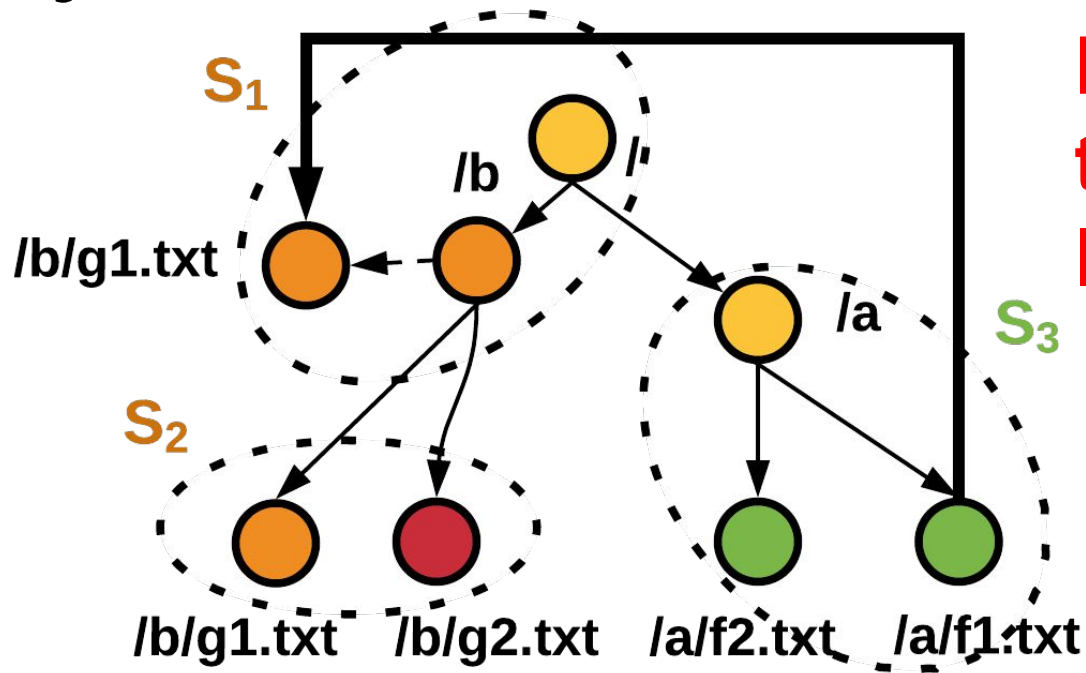


Locality and Load Balance

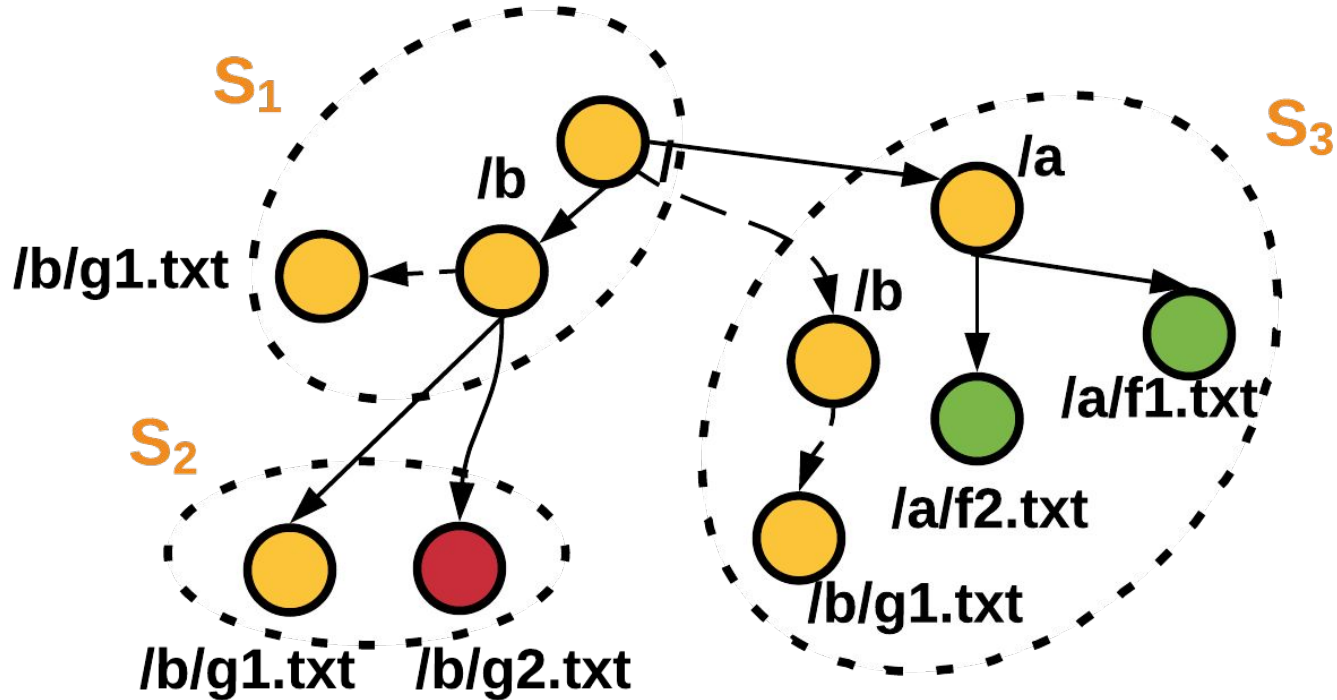


(Gopalakrishnan et al., ICDCS'04)

Locality and Load Balance

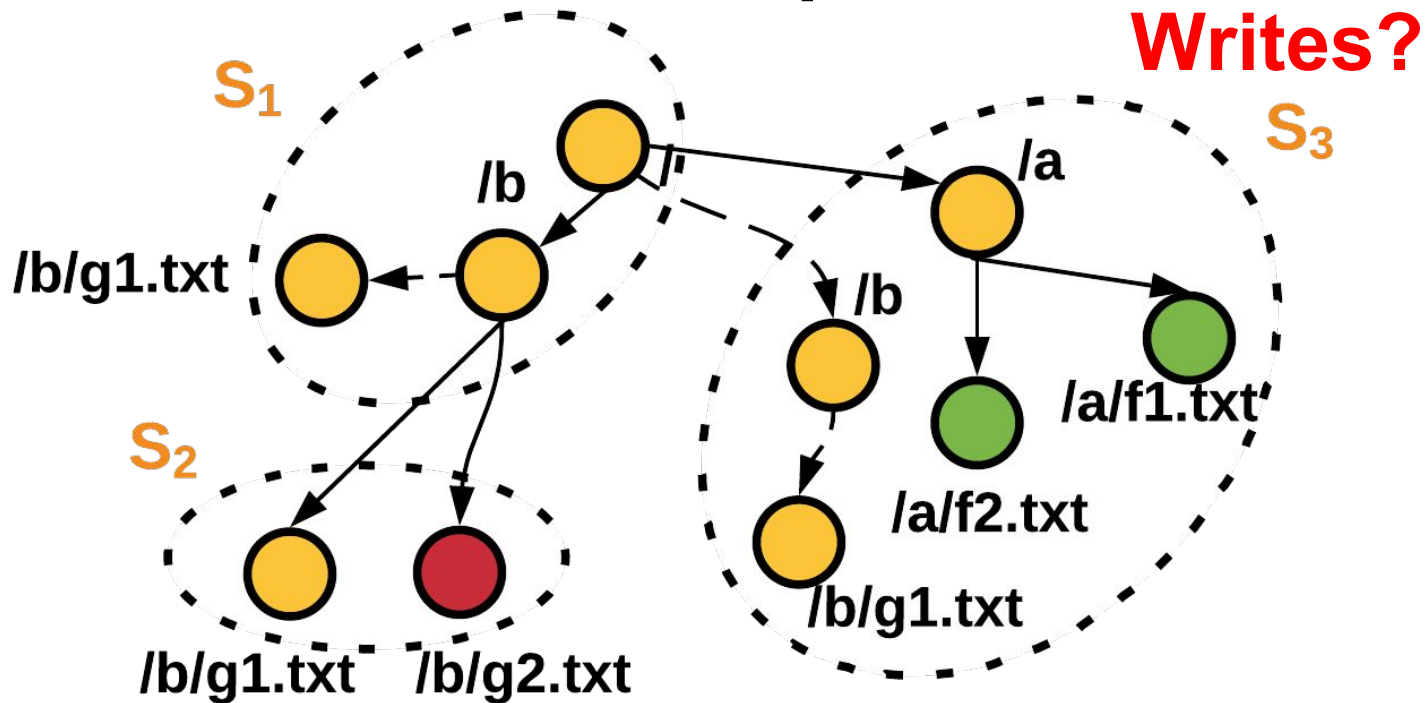


Locality and Load Balance



(Gopalakrishnan et al., ICDCS'04)

Decentralized Replicas



(Gopalakrishnan et al., ICDCS'04)

Replication Decisions

- **How many** replicas?

Decentralized decision

- **Where to place** replicas?

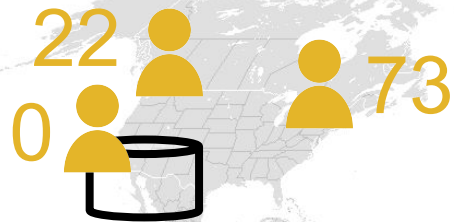
At the requester

- **How to propagate** updates?

ADR: Synchronous, P2P: read-only



Global Scale Replication

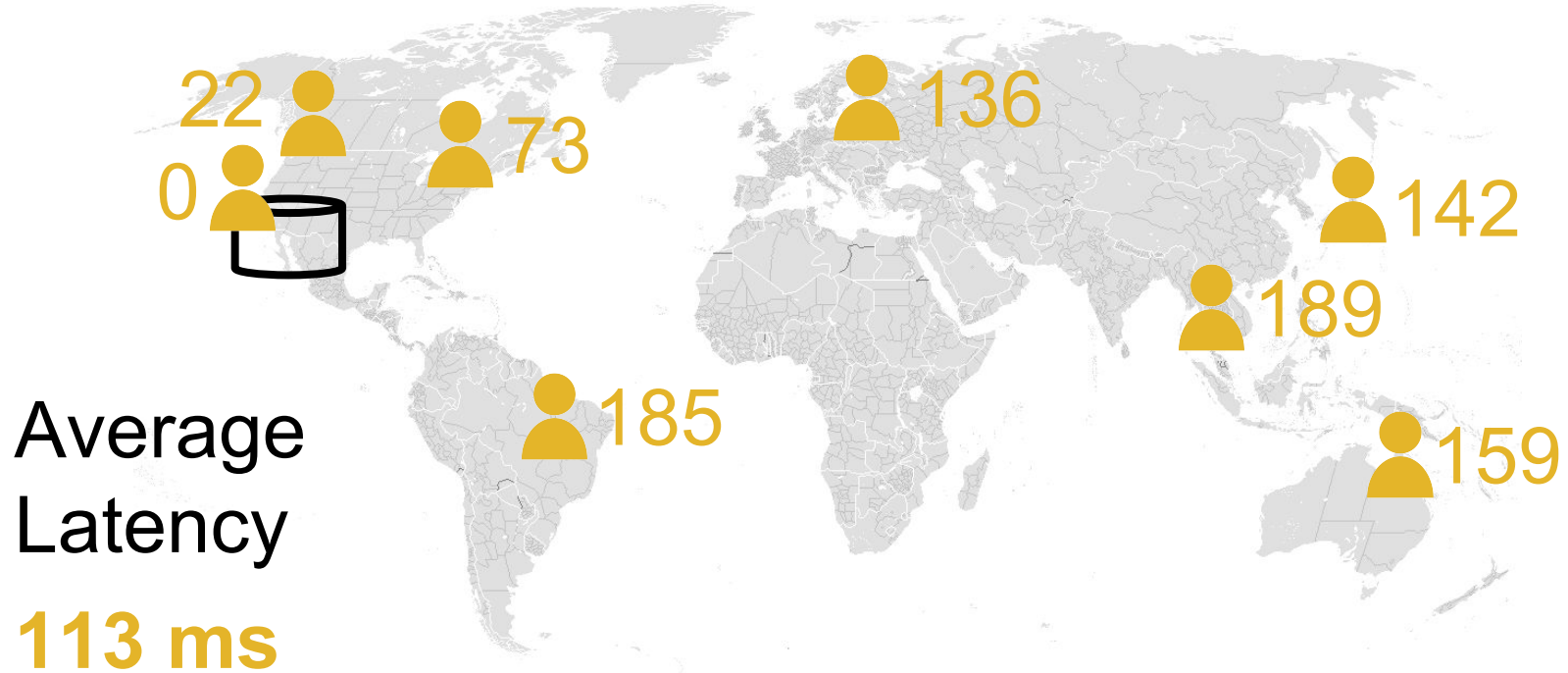


Average
Latency

31 ms

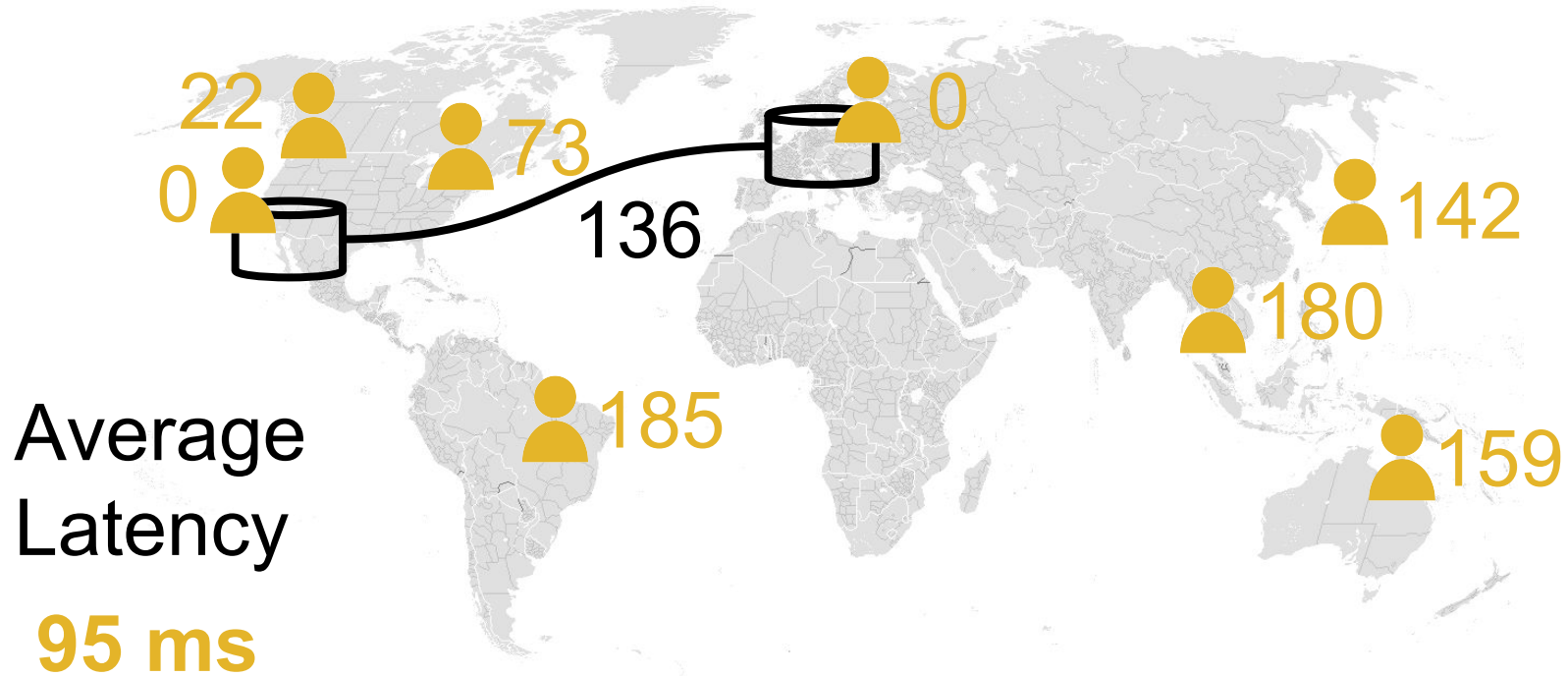


Global Scale Replication

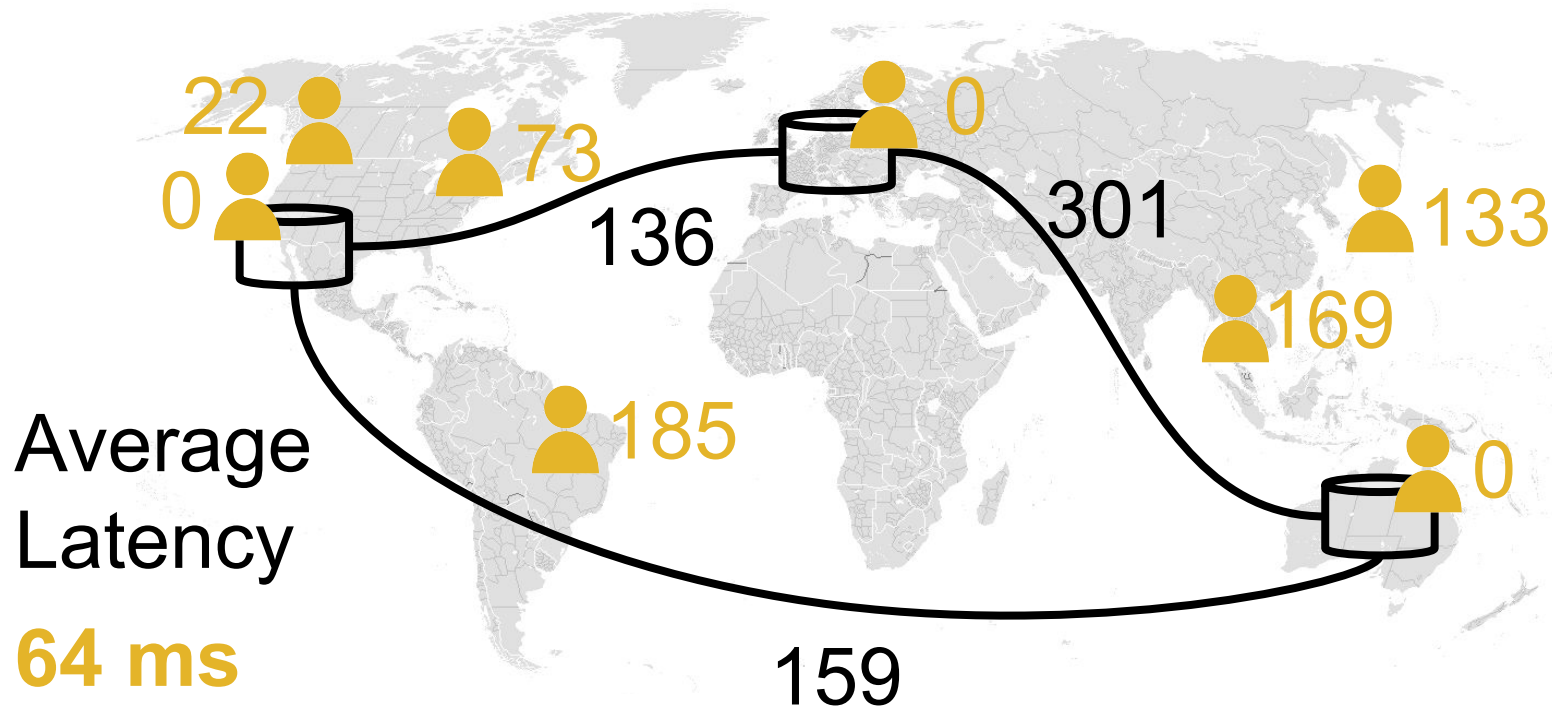




Global Scale Replication



Global Scale Replication



Global Scale Replication

Place data around the world

Minimize cost of access

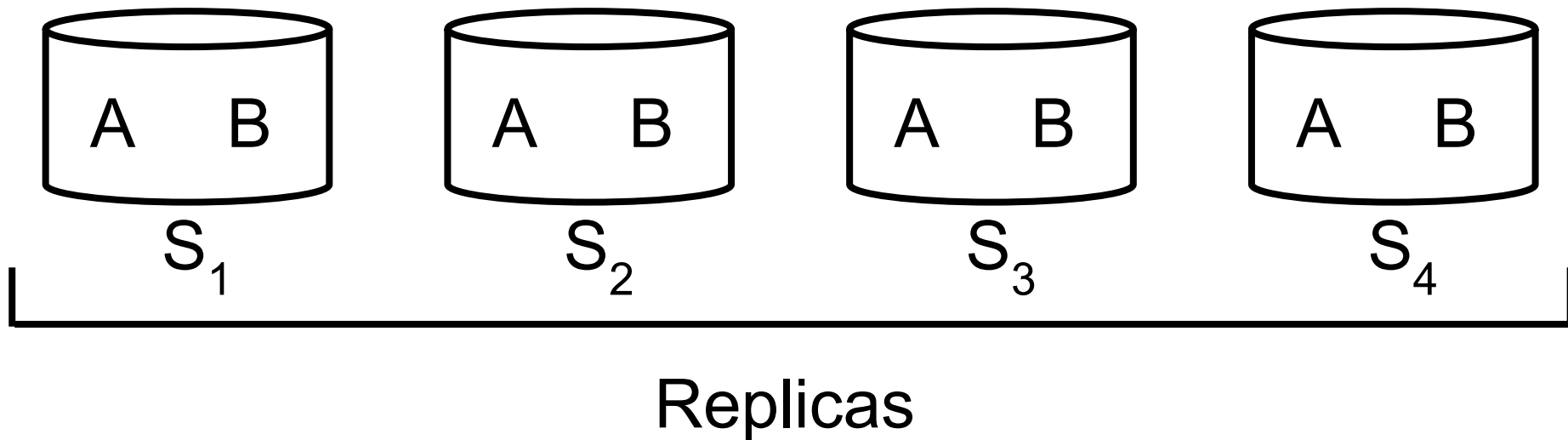
Take me to your
leader!

(Sharov et al., VLDB 2015)

GPlacer

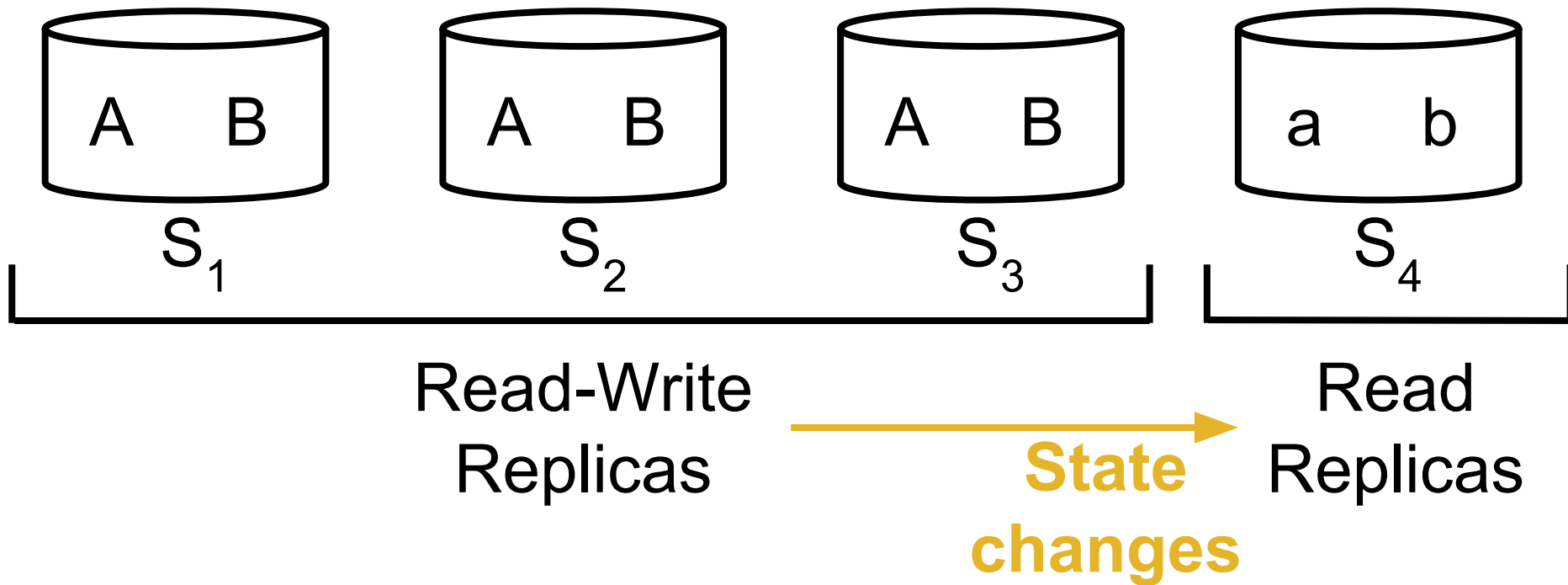
(Zakhary et al., EDBT 2018)

Data Distribution Model



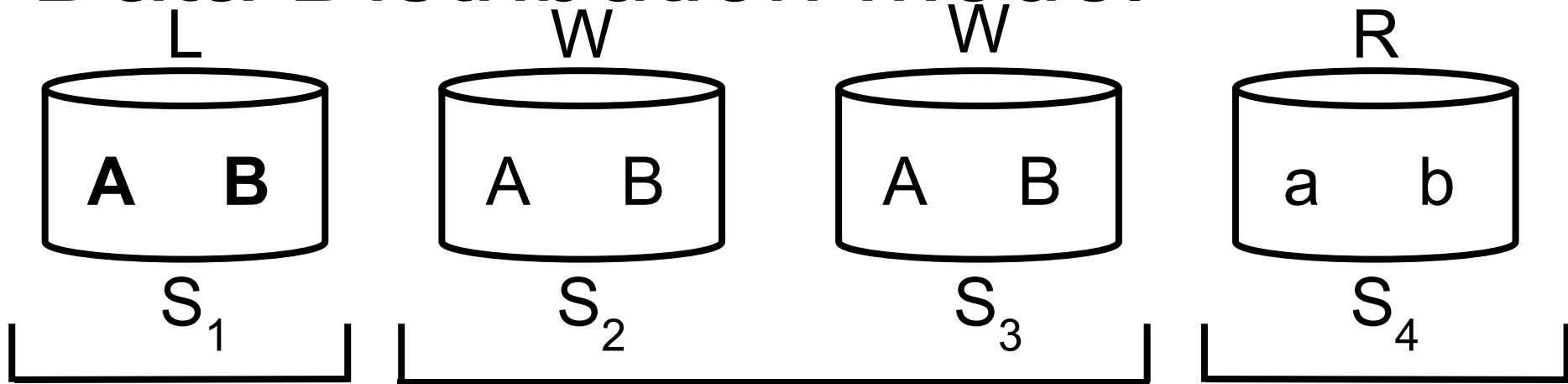
(Sharov et al., VLDB 2015)

Data Distribution Model



(Sharov et al., VLDB 2015)

Data Distribution Model



(Sharov et al., VLDB 2015)

Global Replication Problem

- Select replicas
- Assign replica roles (read or read-write)
- Assign leader

(Sharov et al., VLDB 2015)

Global Replication Problem

- Select replicas
- Assign replica roles (read or read-write)
- Assign leader

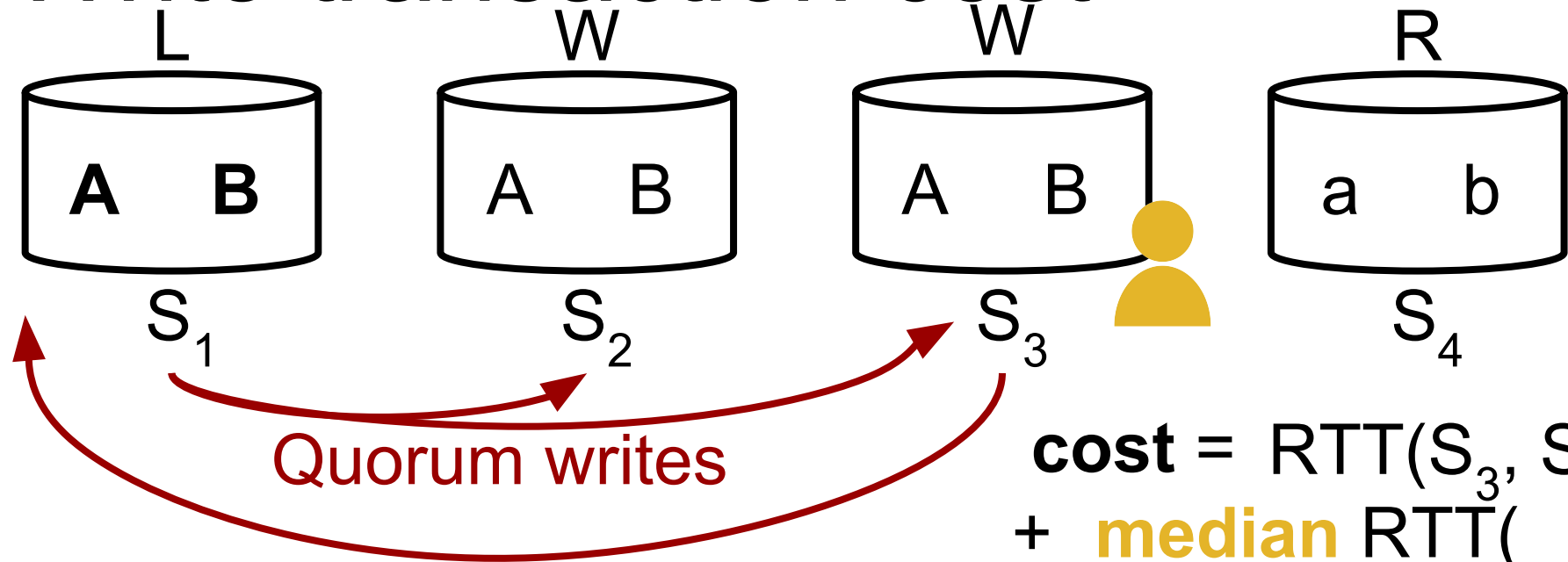
(Sharov et al., VLDB 2015)

Assign Leader

Leader: site that **minimizes access costs**

(Sharov et al., VLDB 2015)

Write transaction cost



Send write to leader

$$\text{cost} = \text{RTT}(S_3, S_1) + \text{median RTT}(S_1, \{S_1, S_2, S_3\})$$

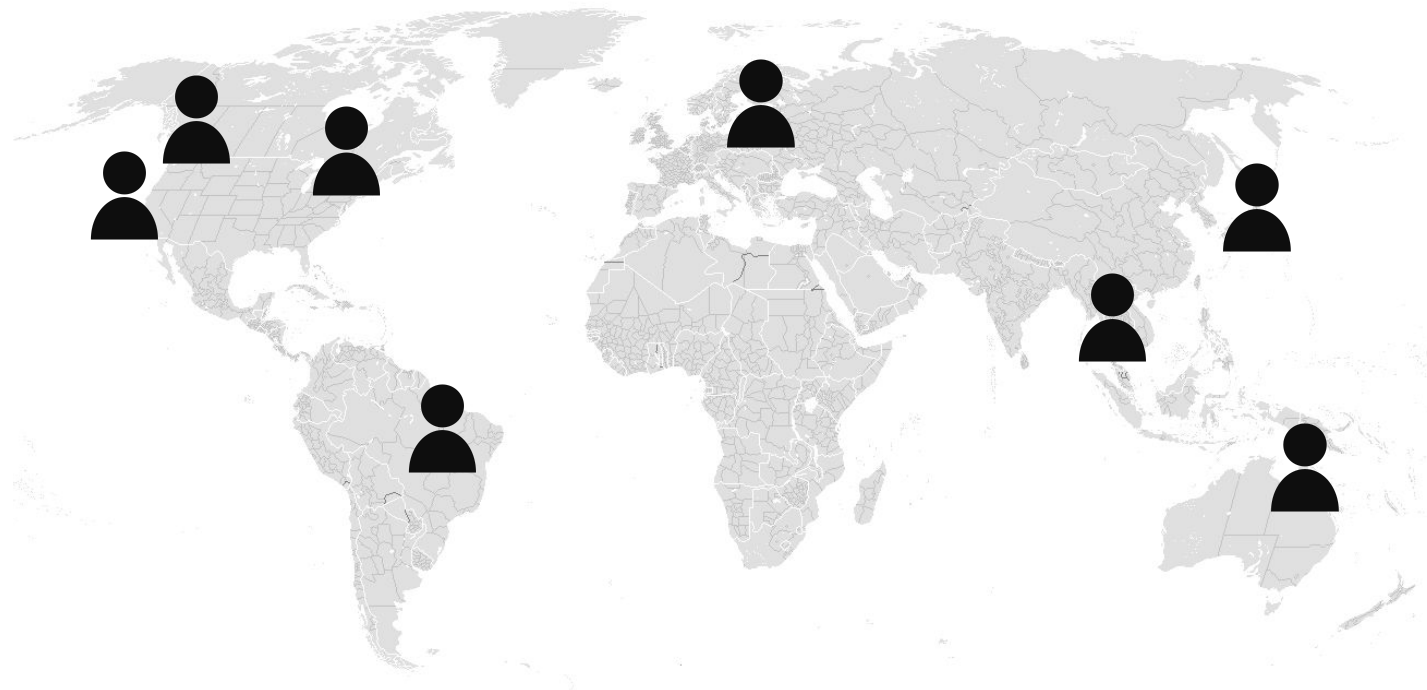
(Sharov et al., VLDB 2015)

Assign Leader

Leader: site that **minimizes access costs**

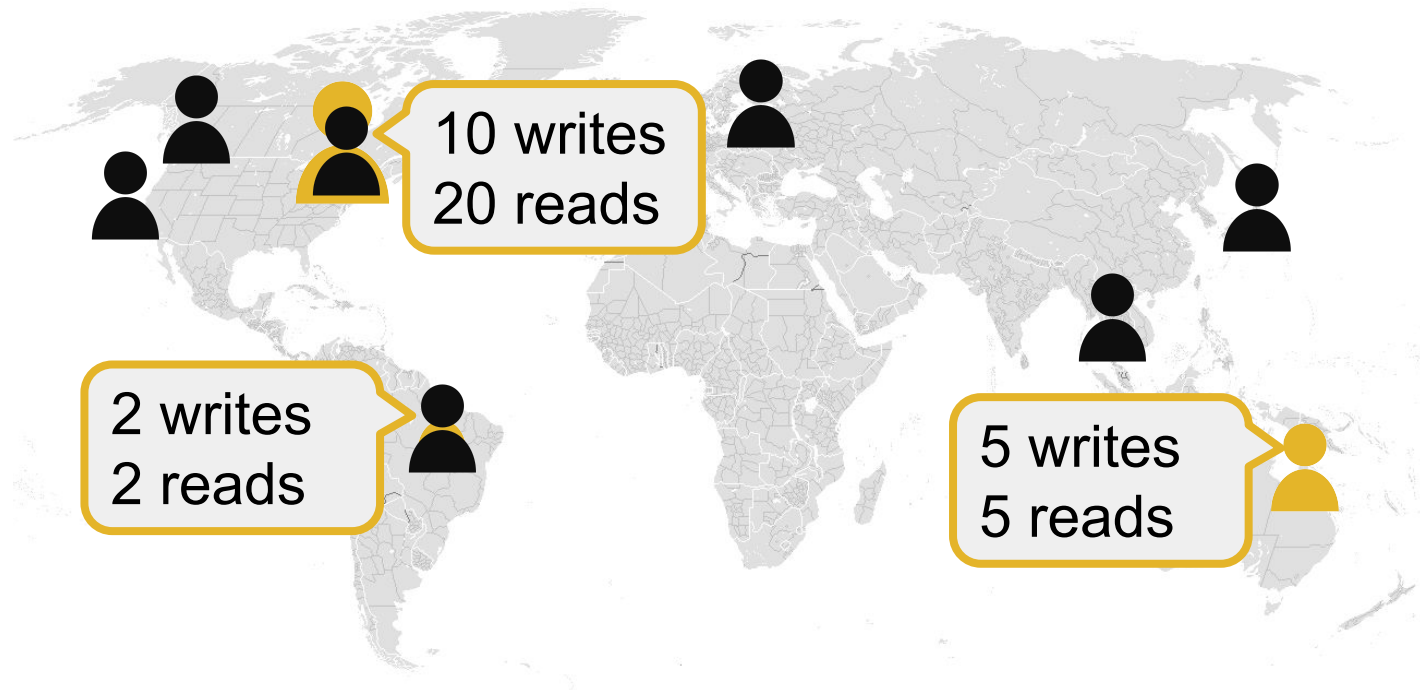
Client cost: $\text{RTT}(\text{client}, \text{replica}) +$
 $\text{cost}(\text{ transaction })$

Weighting Client Cost



(Sharov et al., VLDB 2015)

Weighting Client Cost



(Sharov et al., VLDB 2015)

Assign Leader

Leader: site that **minimizes access costs**

Client cost: $\text{RTT}(\text{client, replica}) +$
 $\text{cost}(\text{ transaction })$

Cost: Weighted average of client costs

Global Replication Problem

- Select replicas
- Assign replica roles (read or read-write)
- Assign leader

(Sharov et al., VLDB 2015)

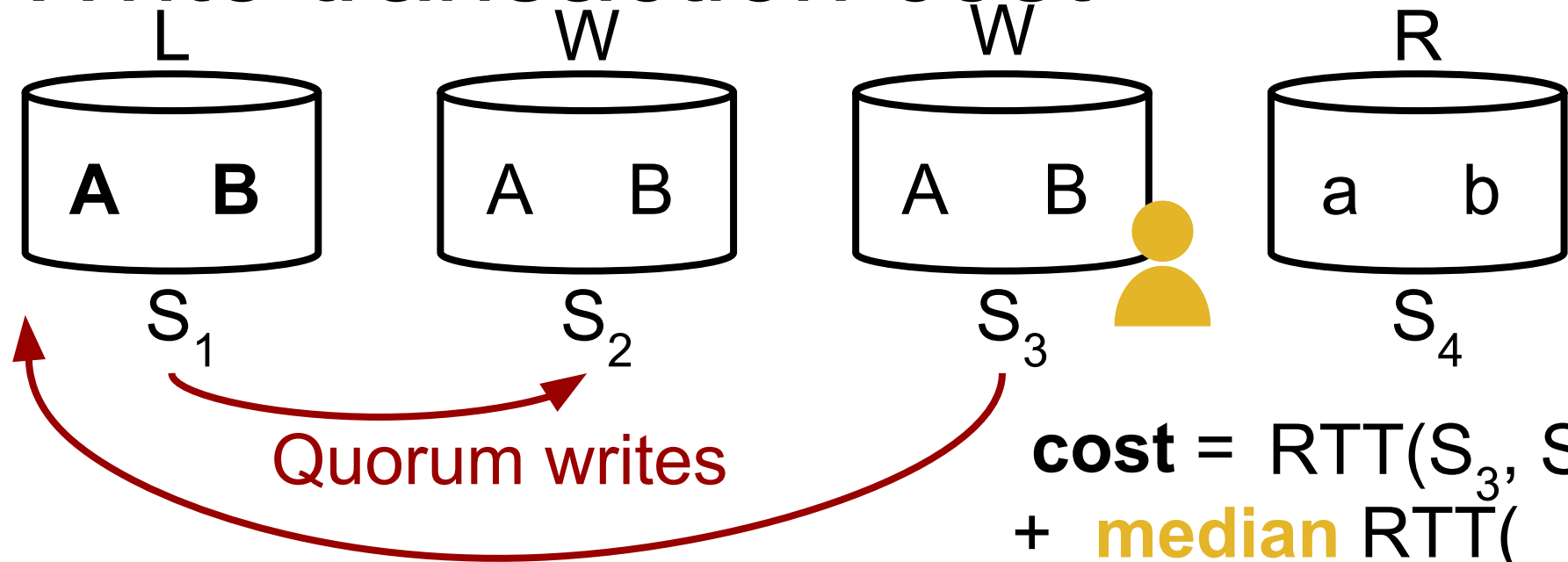
Assign Replica Roles

Leader: **minimizes median RTT** to read-write replicas

Read-write replicas:

(Sharov et al., VLDB 2015)

Write transaction cost



Send write to leader

$$\text{cost} = \text{RTT}(S_3, S_1) + \text{median RTT}(S_1, \{S_1, S_2, S_3\})$$

(Sharov et al., VLDB 2015)

Assign Replica Roles

Leader: **minimizes median RTT** to read-write replicas

Read-write replicas: Lowest RTT to leader

Global Replication Problem

- Select replicas
- Assign replica roles (read or read-write)
- Assign leader

(Sharov et al., VLDB 2015)

Replica selection

Leader: **minimizes median RTT** to read-write replicas

Read-write replicas: Lowest RTT to leader

Read replicas:

(Sharov et al., VLDB 2015)

Replica selection

Client cost: $\text{RTT}(\text{client}, \text{replica}) + \text{cost}(\text{transaction})$

(Sharov et al., VLDB 2015)

Replica selection

Client cost: $RTT(\text{client}, \text{replica}) +$
 $\text{cost}(\text{ transaction })$

Read replicas: Lowest RTT to **clients**

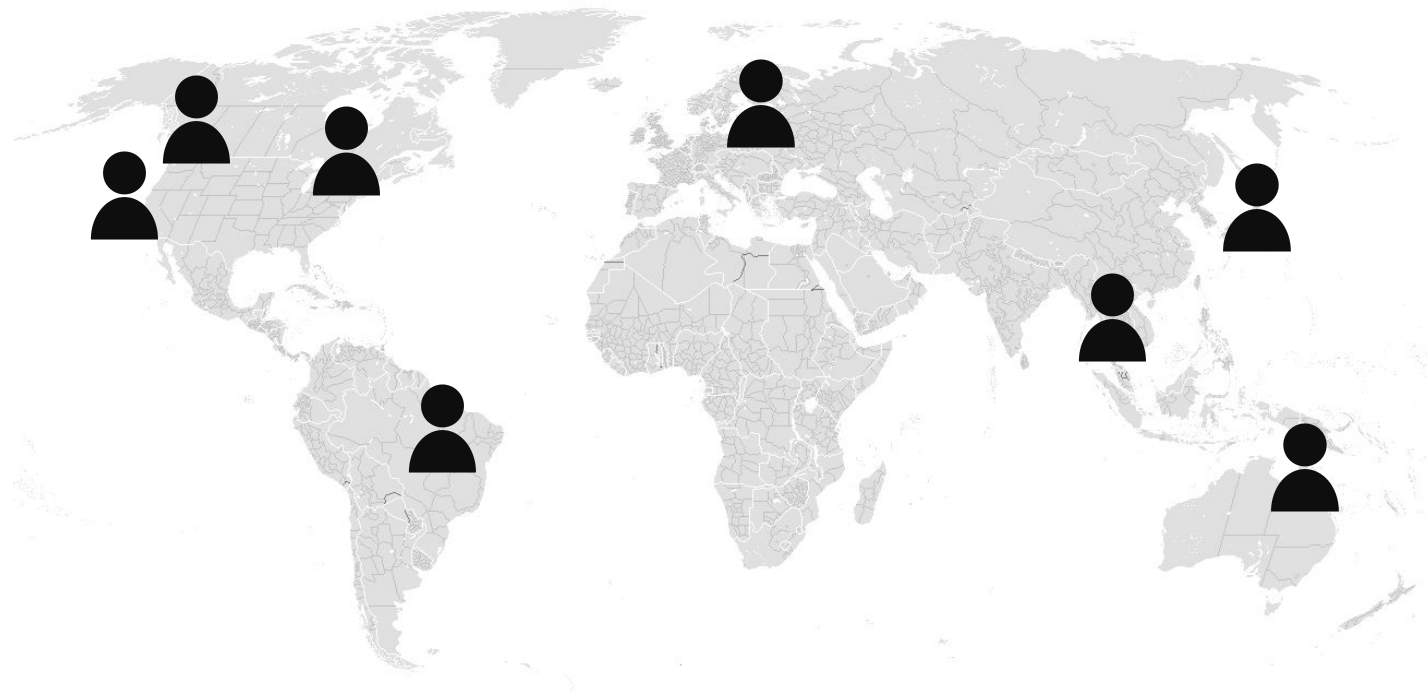
Replica selection

Leader: **minimizes median RTT** to read-write replicas

Read-write replicas: Lowest RTT to leader

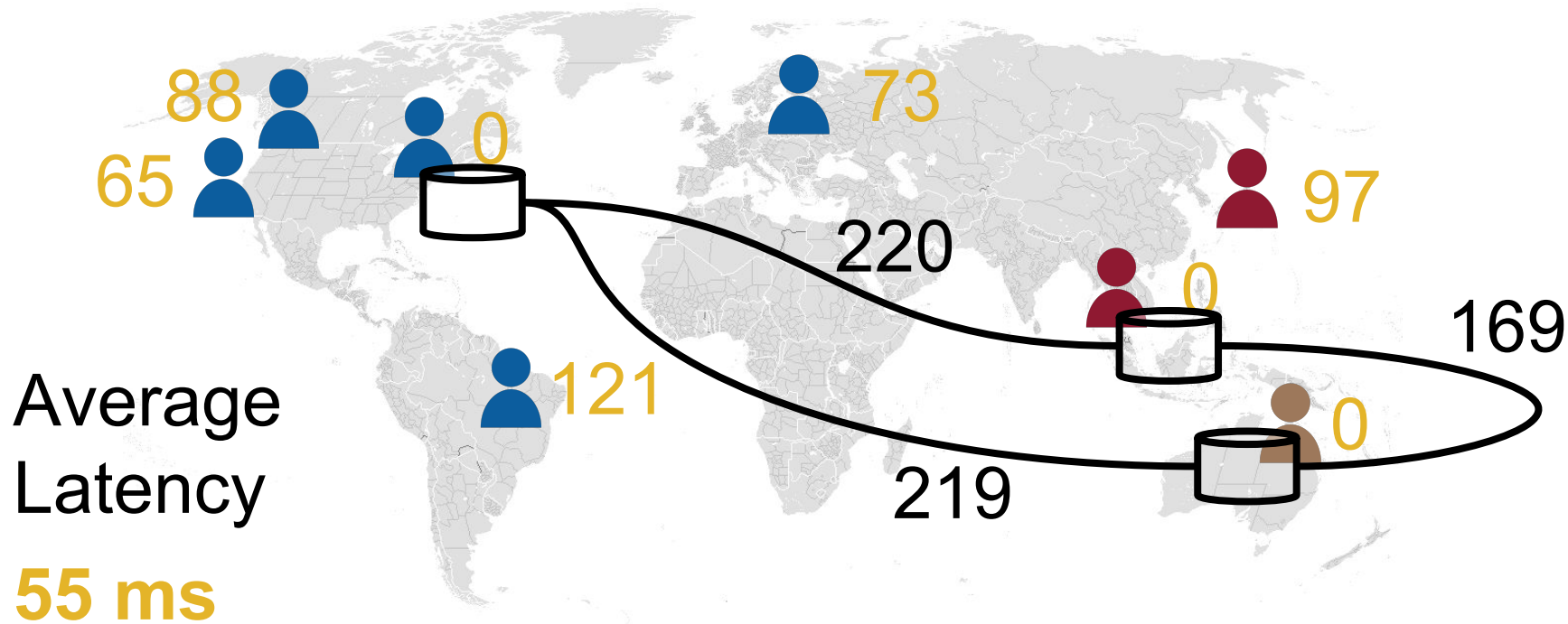
Read replicas: Lowest RTT to **clients**

K-Means Replica selection



(Sharov et al., VLDB 2015)

K-Means Replica selection



(Sharov et al., VLDB 2015)

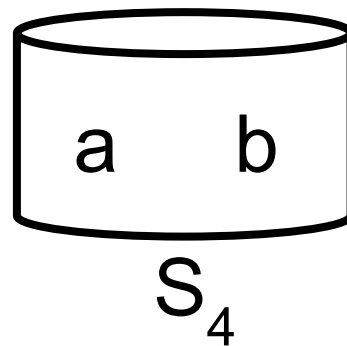
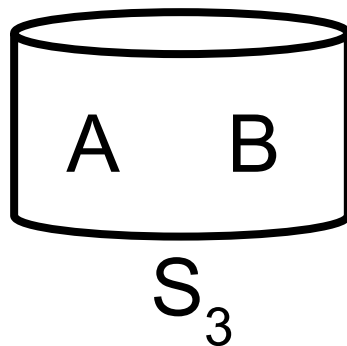
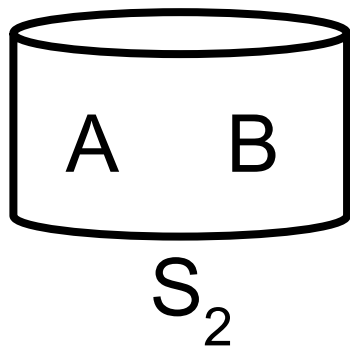
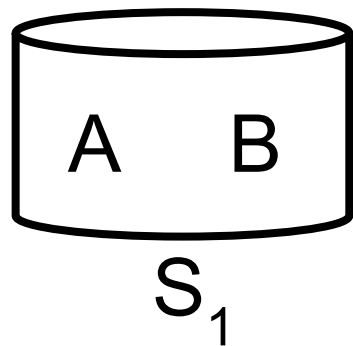
K-Means Replica selection

Select replicas

Assign leader and read-write replicas

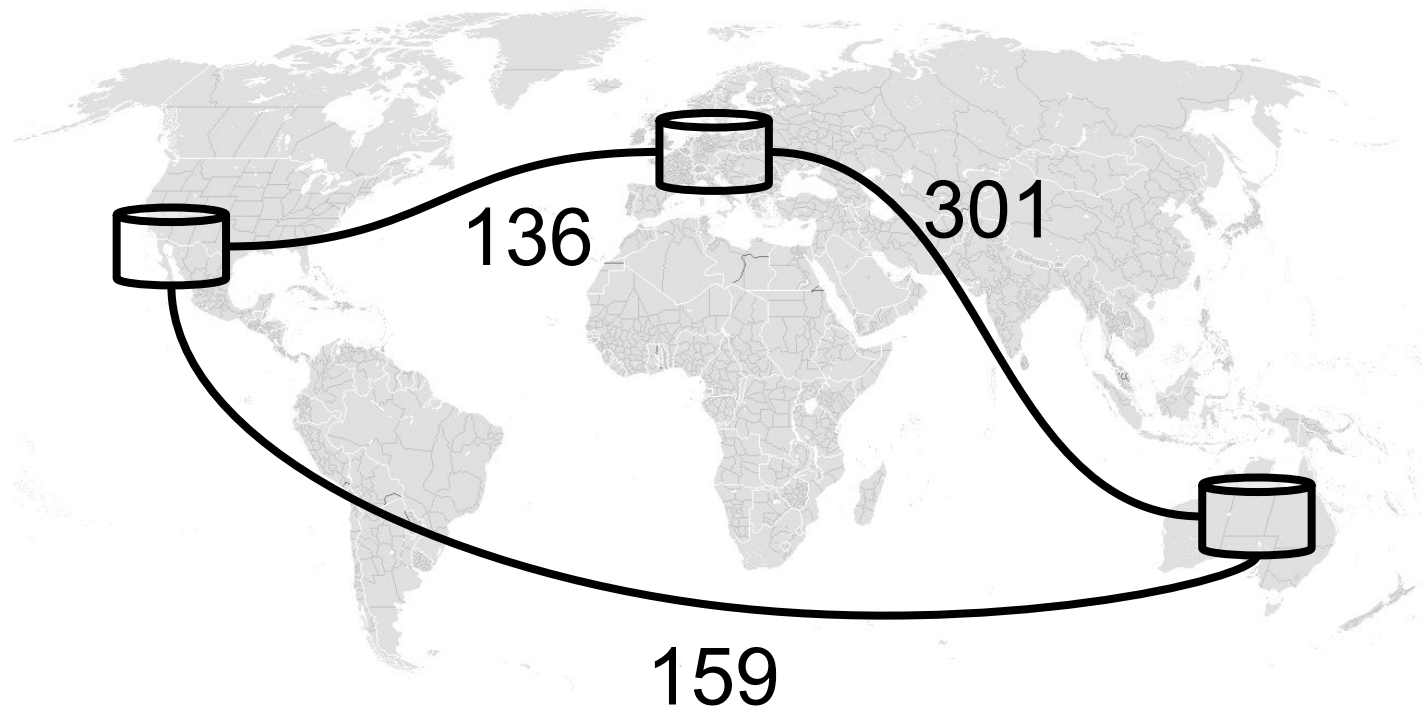
(Sharov et al., VLDB 2015)

Leaderless Protocols



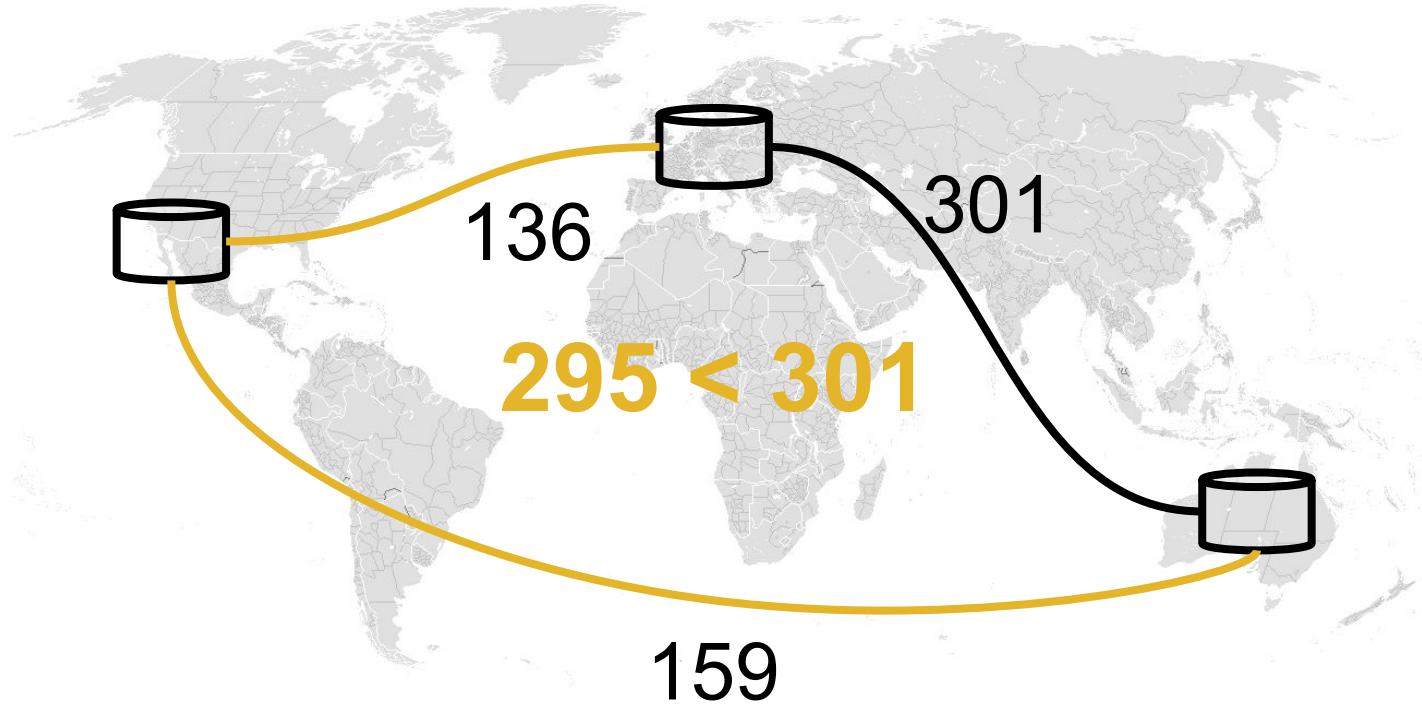
Any quorum member can coordinate

Hinted Hand off



(Zakhary et al., EDBT 2018)

Hinted Hand off



(Zakhary et al., EDBT 2018)

Hinted Hand off

cost(S) = cost of executing request at S

Hand off request from S_1 to S_2 if:

$$\text{cost}(S_1) > \text{RTT}(S_1, S_2) + \text{cost}(S_2)$$

Replication Decisions

- **How many** replicas?

Centralized, given client workload

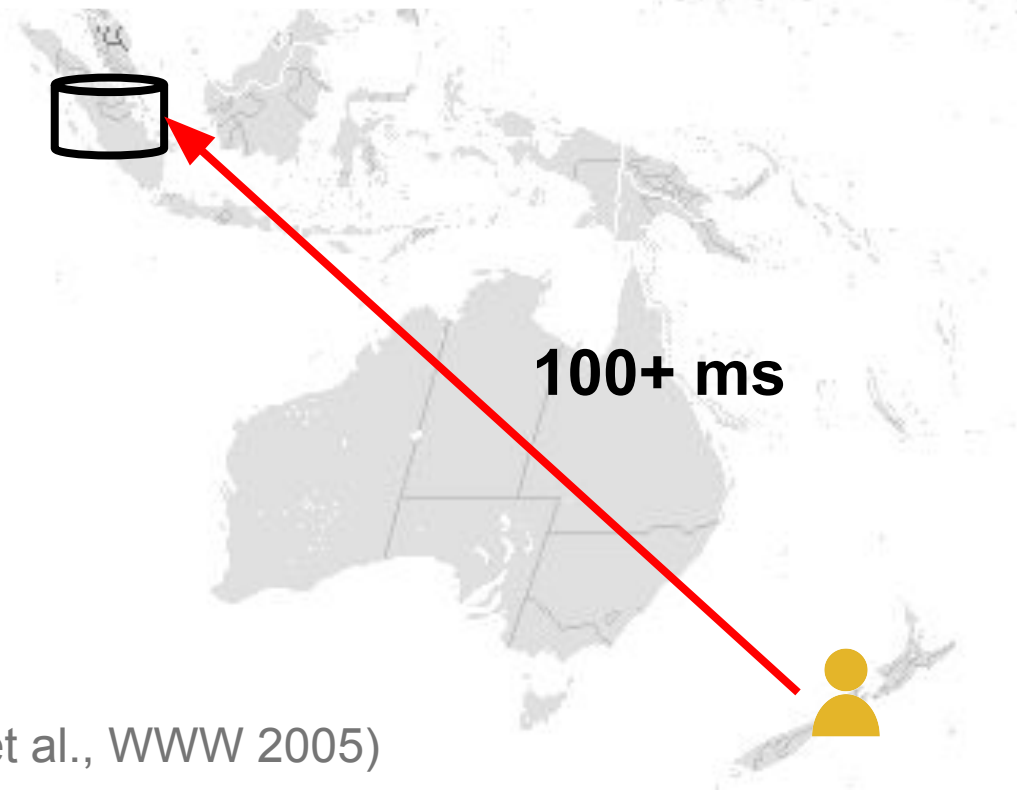
- **Where to place** replicas?

Heuristic (clustering)

- **How to propagate** updates?

Quorums / Leader-based (Sharov)

Intra-Region Latency



(Sivasubramanian et al., WWW 2005)

Edge Nodes

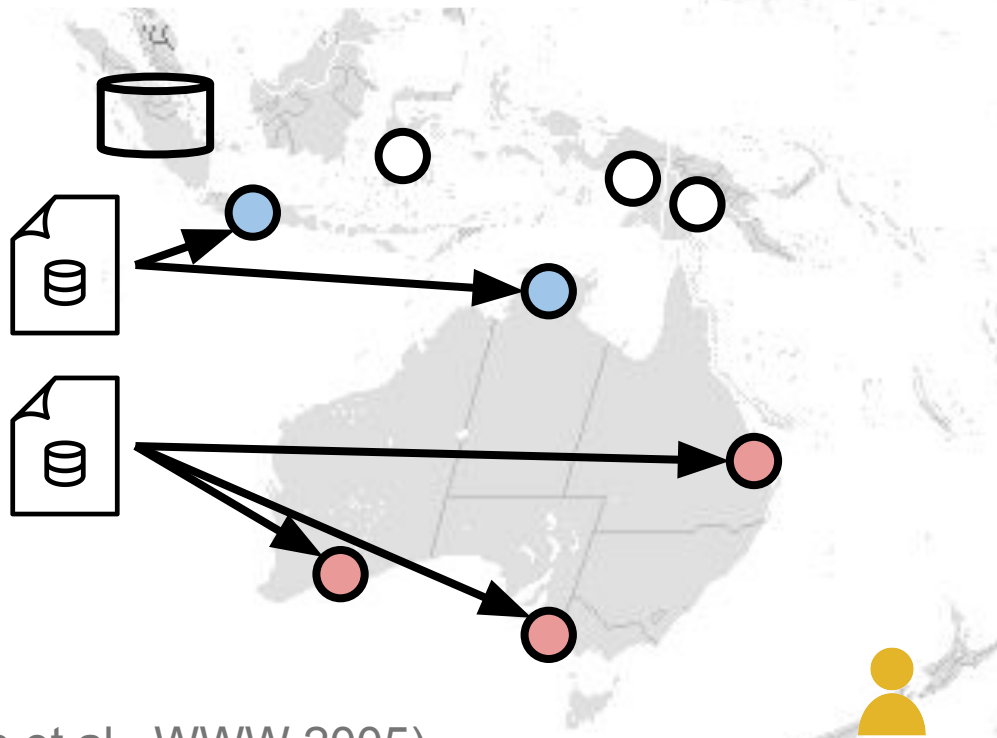


Supports Static Data

Dynamic Data?

(Sivasubramanian et al., WWW 2005)

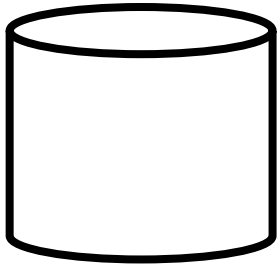
GlobeDB



(Sivasubramanian et al., WWW 2005)

Replication Granularity

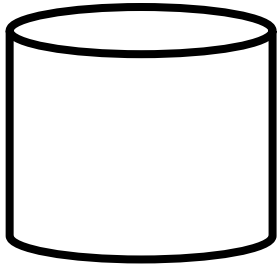
ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne



(Sivasubramanian et al., WWW 2005)

Replication Granularity

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne



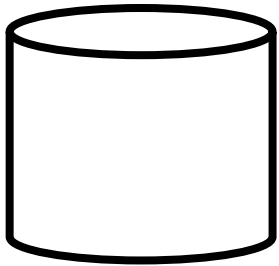
Per-Record?

High Overhead

(Sivasubramanian et al., WWW 2005)

Replication Granularity

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne

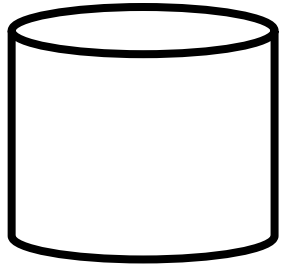


Per-Table?
Inflexible

(Sivasubramanian et al., WWW 2005)

Access-Driven Replicas

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne



When would these be replicated together?

(Sivasubramanian et al., WWW 2005)

Access-Driven Replicas

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne

$$A_{\text{bryan}} = \langle r_1, \dots, r_n, w_1, \dots, w_n \rangle$$

(Sivasubramanian et al., WWW 2005)

Access-Driven Replicas

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne

$$A_{\text{justin}} = \langle r_1, \dots, r_n, w_1, \dots, w_n \rangle$$

$$\text{Sim}(A_{\text{bryan}}, A_{\text{justin}}) \geq T?$$

(Sivasubramanian et al., WWW 2005)

Access-Driven Replicas

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne

Shared Replication Scheme

(Sivasubramanian et al., WWW 2005)

Access-Driven Replicas

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne

$$A_{p1} = \langle r_1, \dots, r_n, w_1, \dots, w_n \rangle$$

(Sivasubramanian et al., WWW 2005)

Access-Driven Replicas

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne

$$A_{\text{avril}} = \langle r_1, \dots, r_n, w_1, \dots, w_n \rangle$$

$$\text{Sim}(A_{p1}, A_{\text{avril}}) \geq \tau?$$

(Sivasubramanian et al., WWW 2005)

Access-Driven Replicas

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne

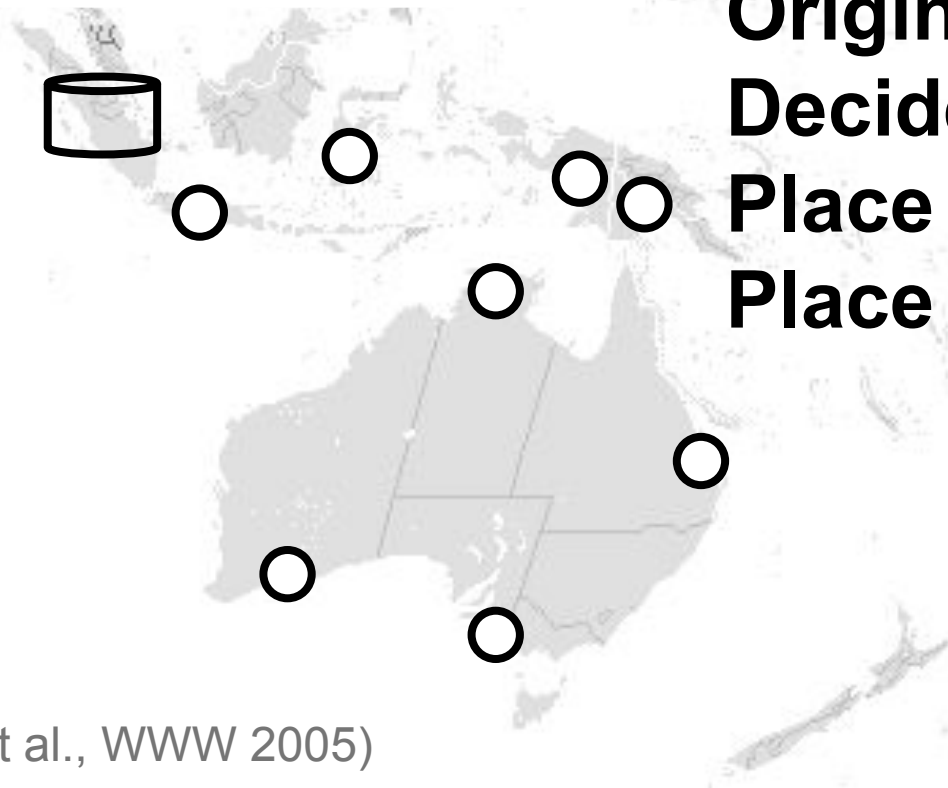
(Sivasubramanian et al., WWW 2005)

Access-Driven Replicas

ID	ARTIST
1	Bryan Adams
2	Justin Bieber
3	Avril Lavigne
4	Kanye West
5	Drake
6	David Guetta
7	Ed Sheeran

(Sivasubramanian et al., WWW 2005)

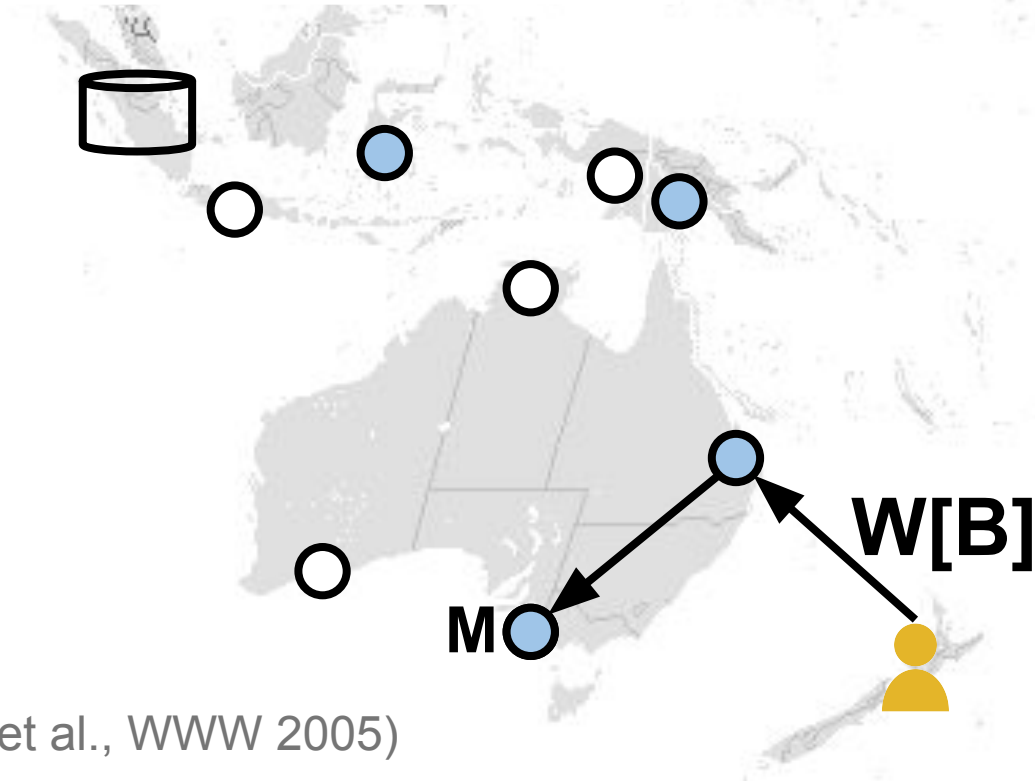
Transaction Processing



**Origin Server:
Decide Partitions,
Place Replicas,
Place Master**

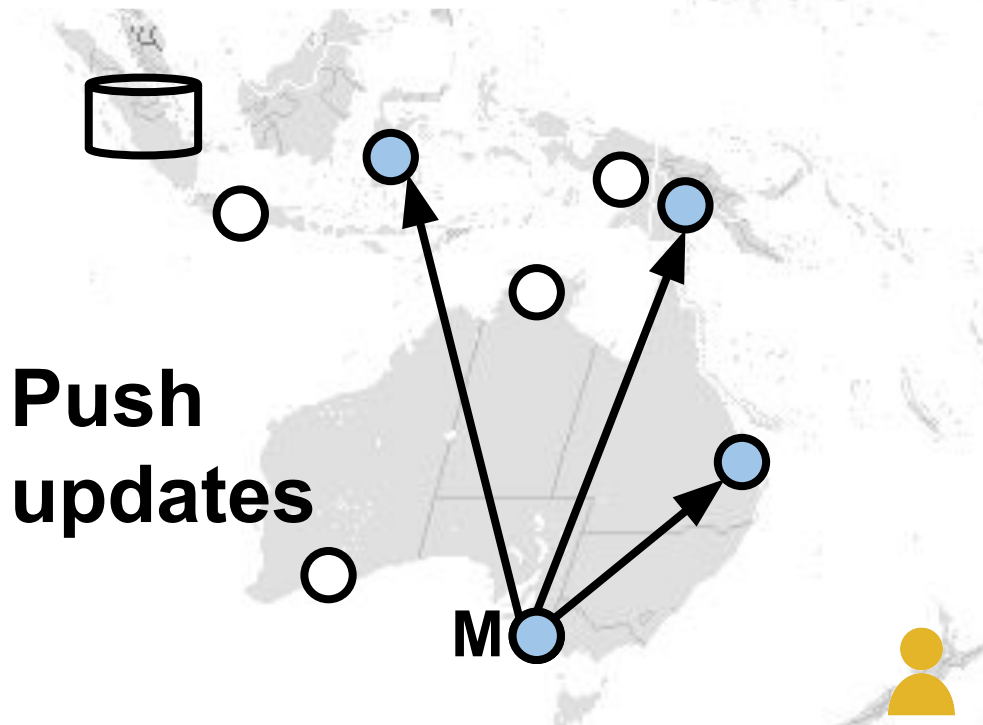
(Sivasubramanian et al., WWW 2005)

Transaction Processing



(Sivasubramanian et al., WWW 2005)

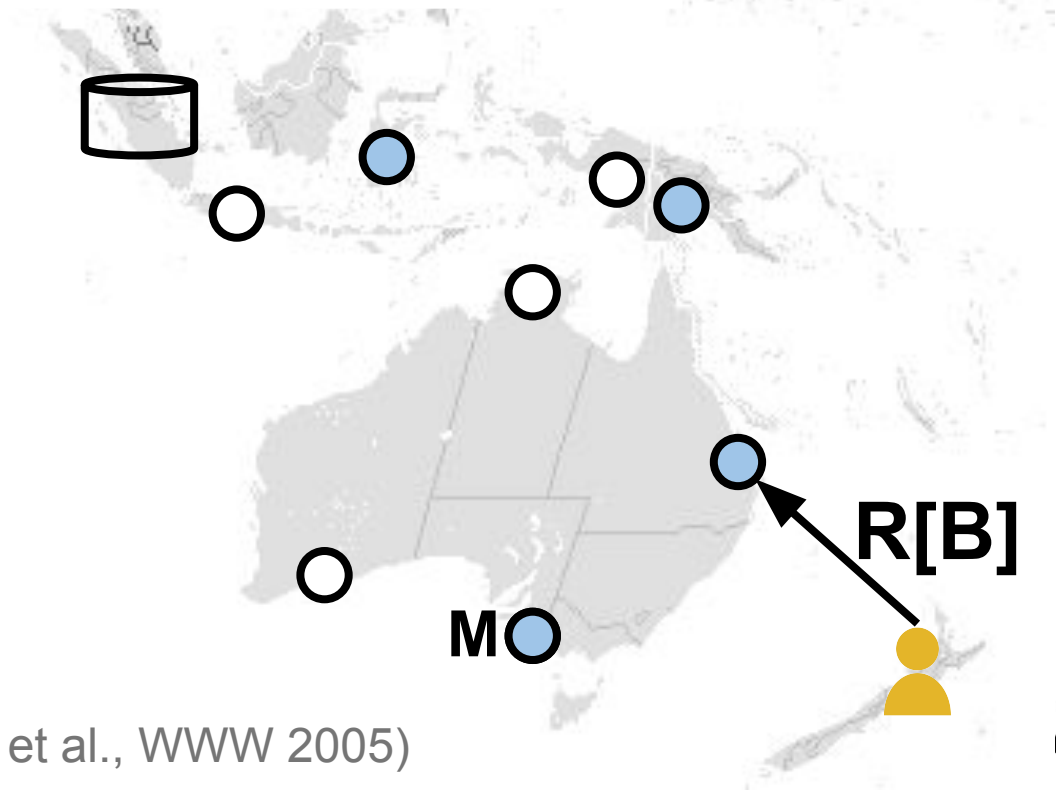
Transaction Processing



**Push
updates**

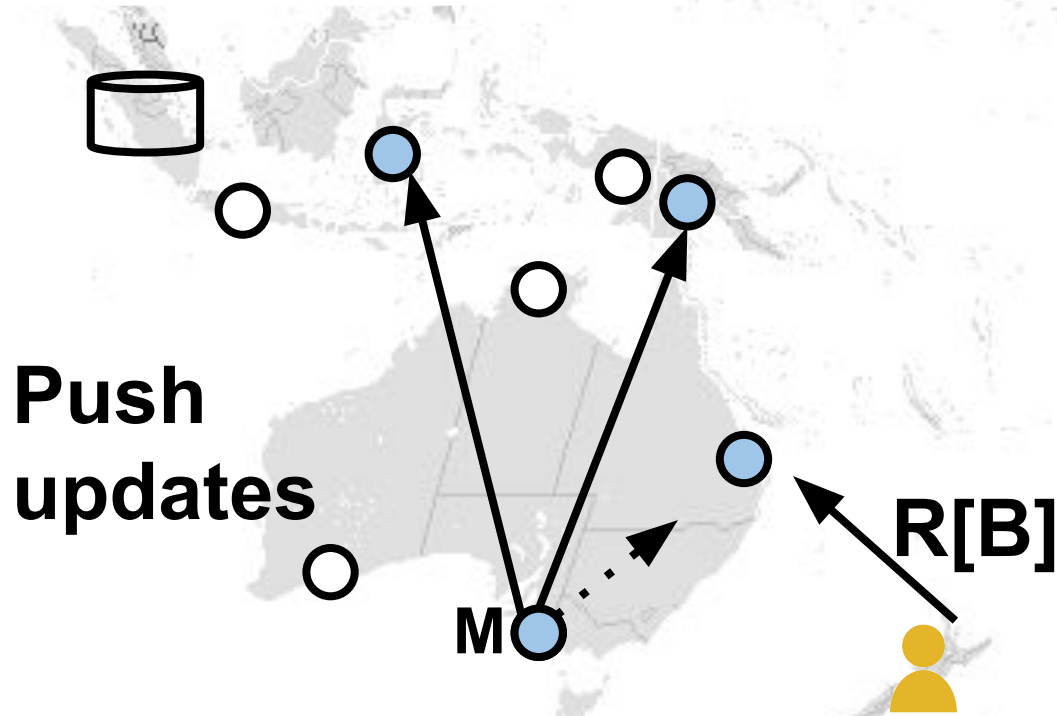
(Sivasubramanian et al., WWW 2005)

Transaction Processing



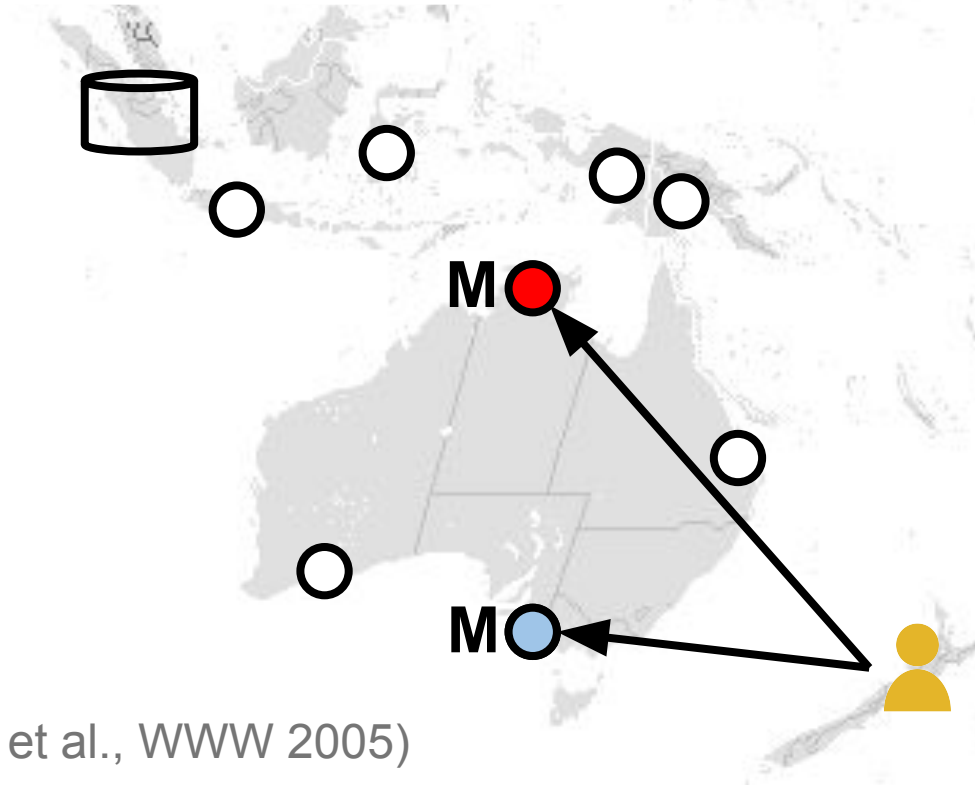
(Sivasubramanian et al., WWW 2005)

Transaction Processing



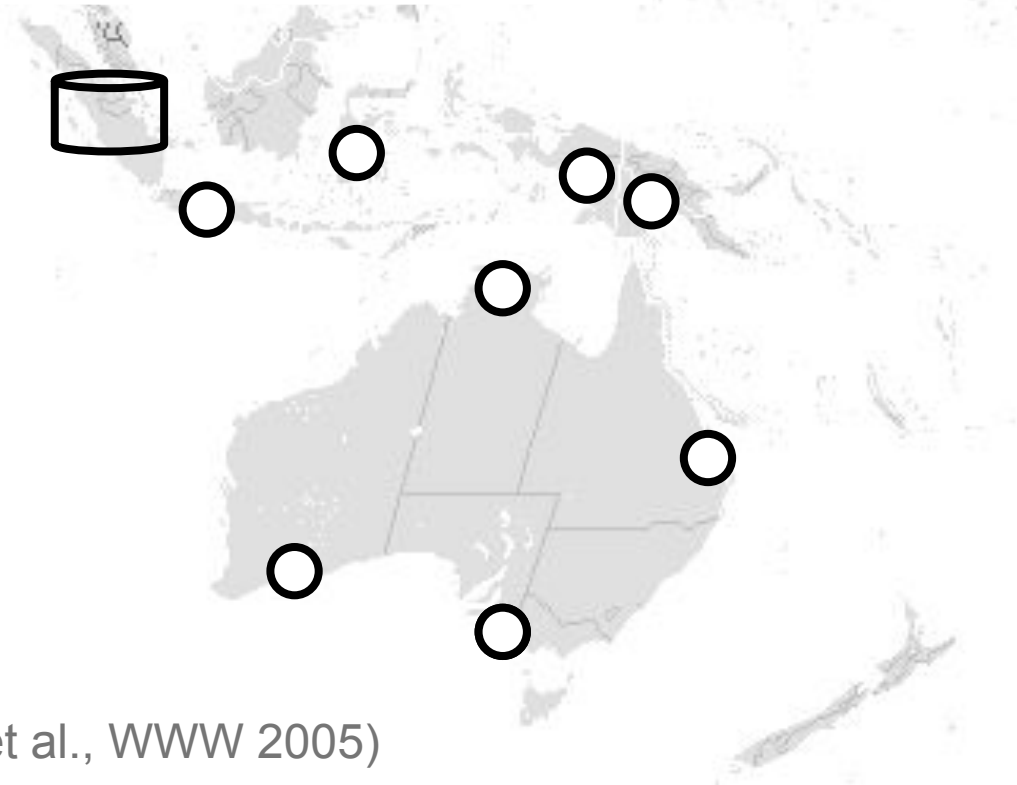
(Sivasubramanian et al., WWW 2005)

Transaction Processing



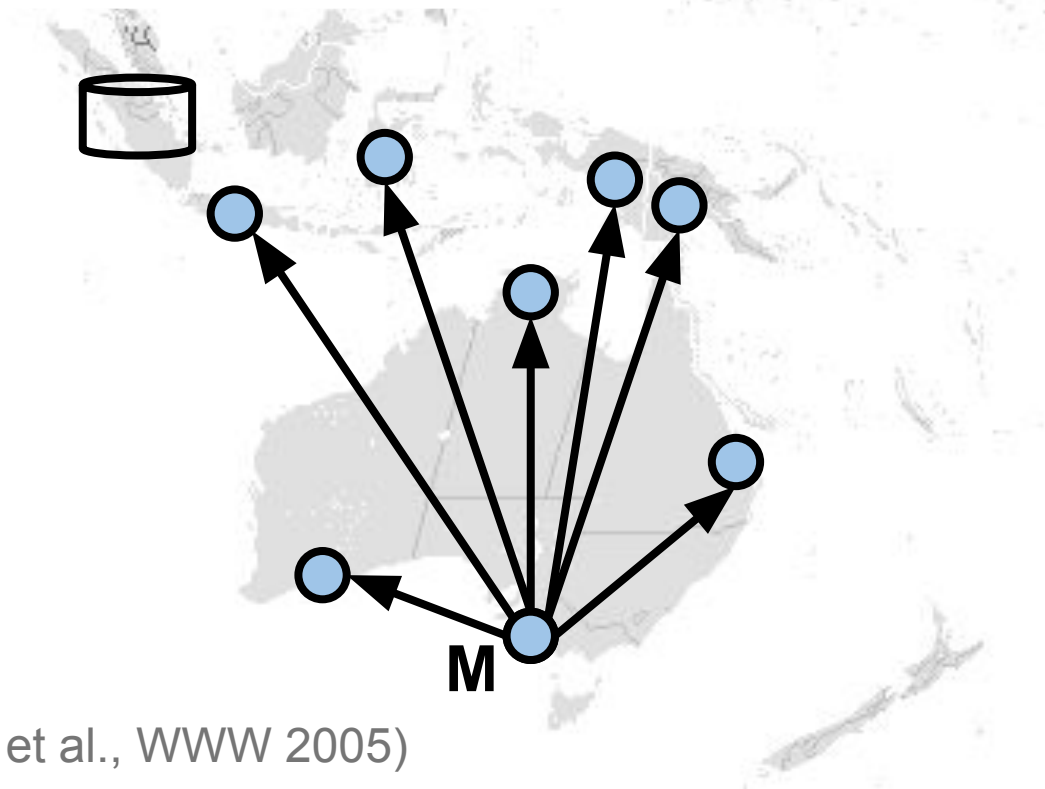
(Sivasubramanian et al., WWW 2005)

Replica and Master Placement



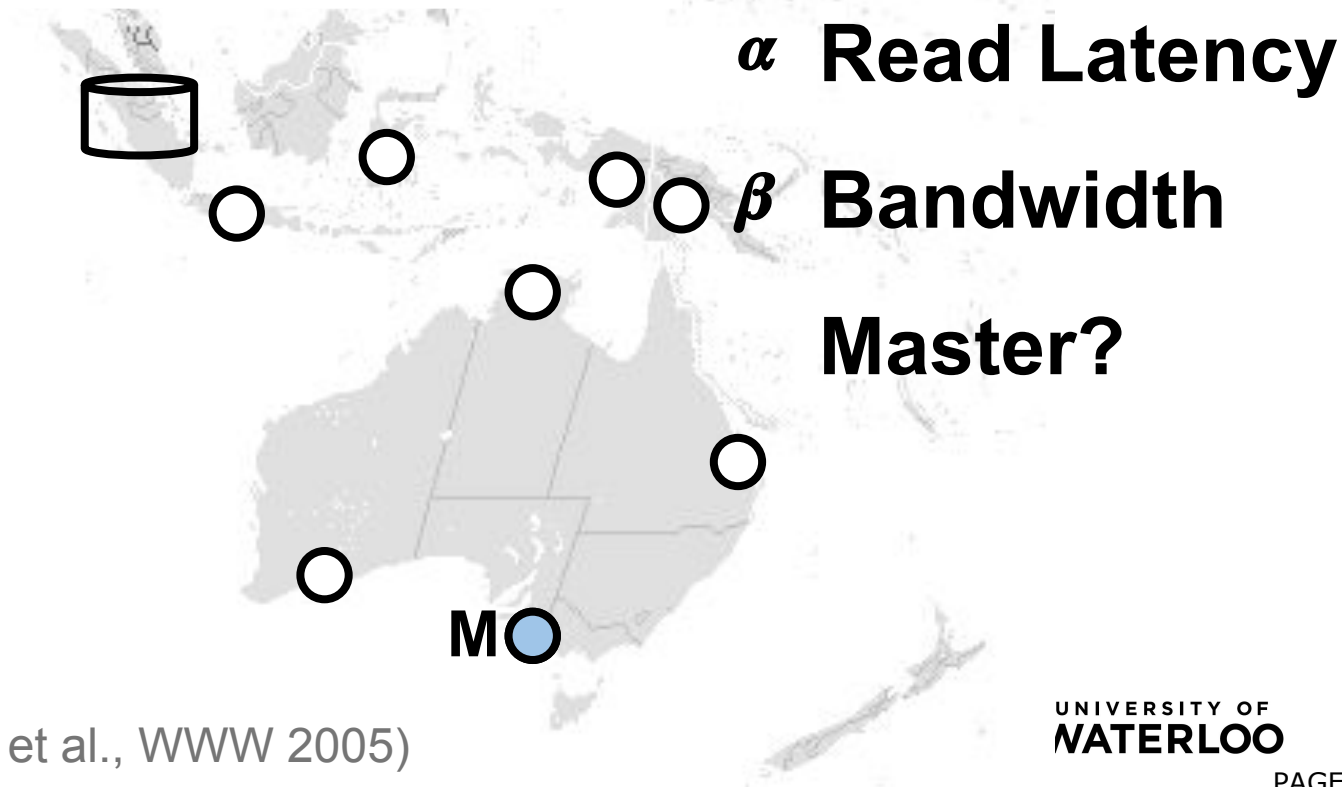
(Sivasubramanian et al., WWW 2005)

Read Latency vs. Bandwidth



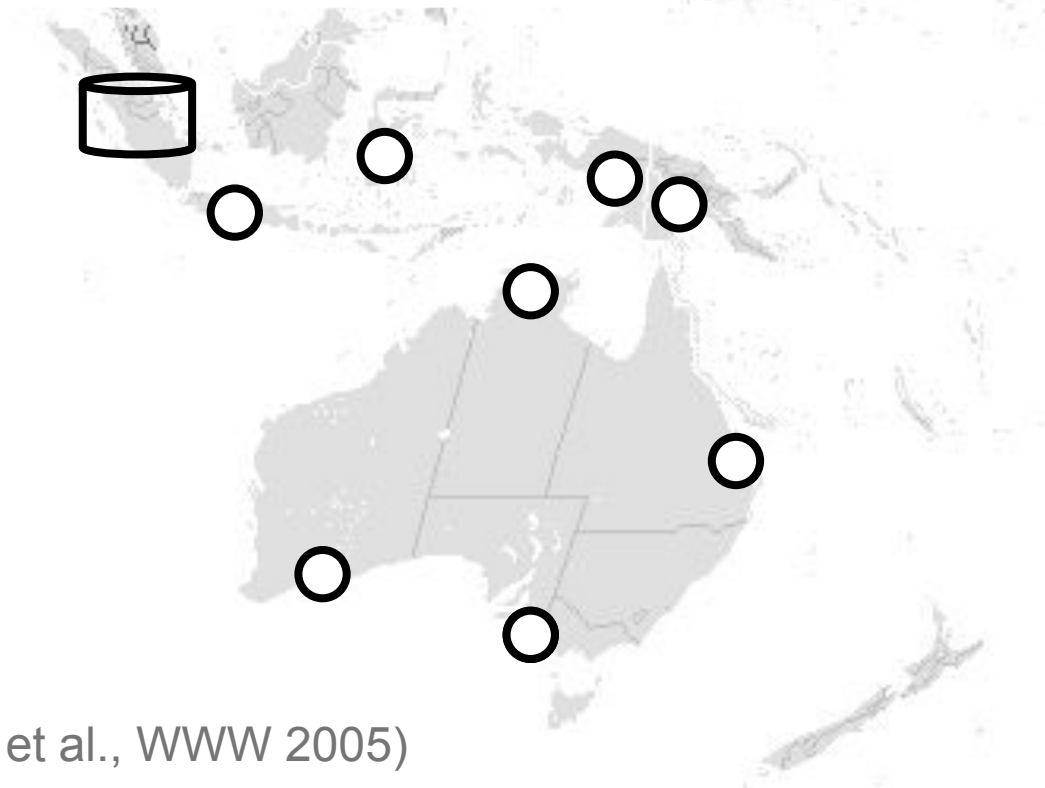
(Sivasubramanian et al., WWW 2005)

Read Latency vs. Bandwidth



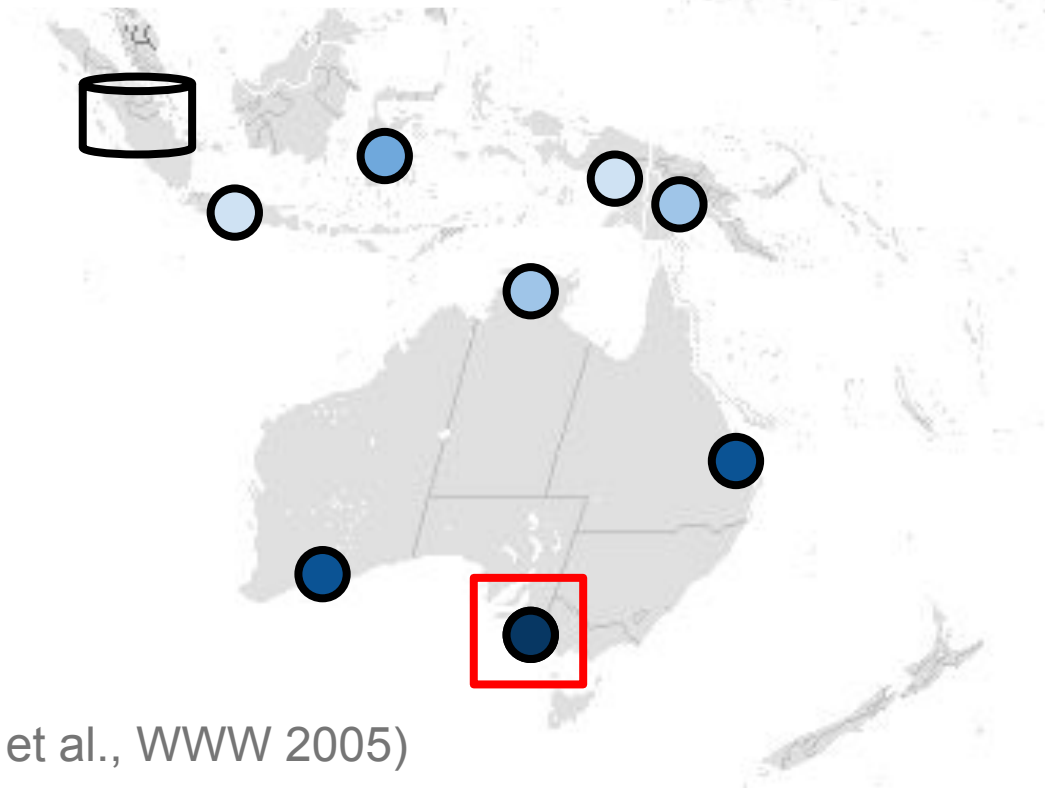
(Sivasubramanian et al., WWW 2005)

Master Partition Placement



(Sivasubramanian et al., WWW 2005)

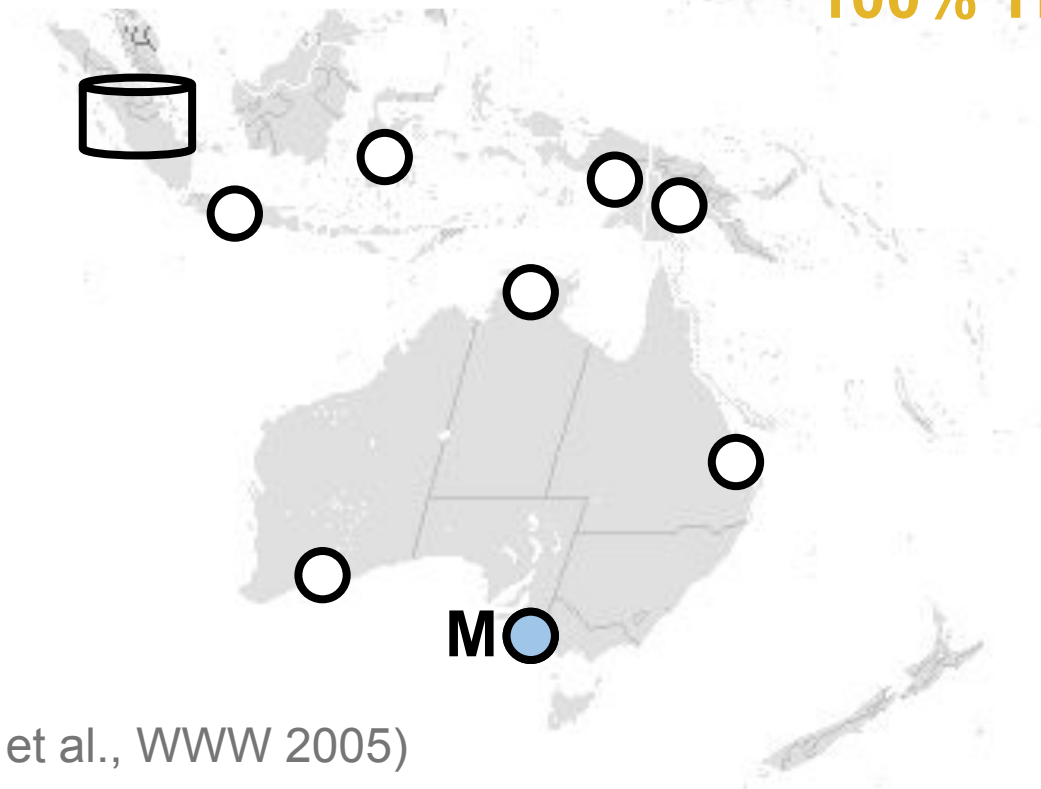
Master Partition Placement



(Sivasubramanian et al., WWW 2005)

Placement Heuristic

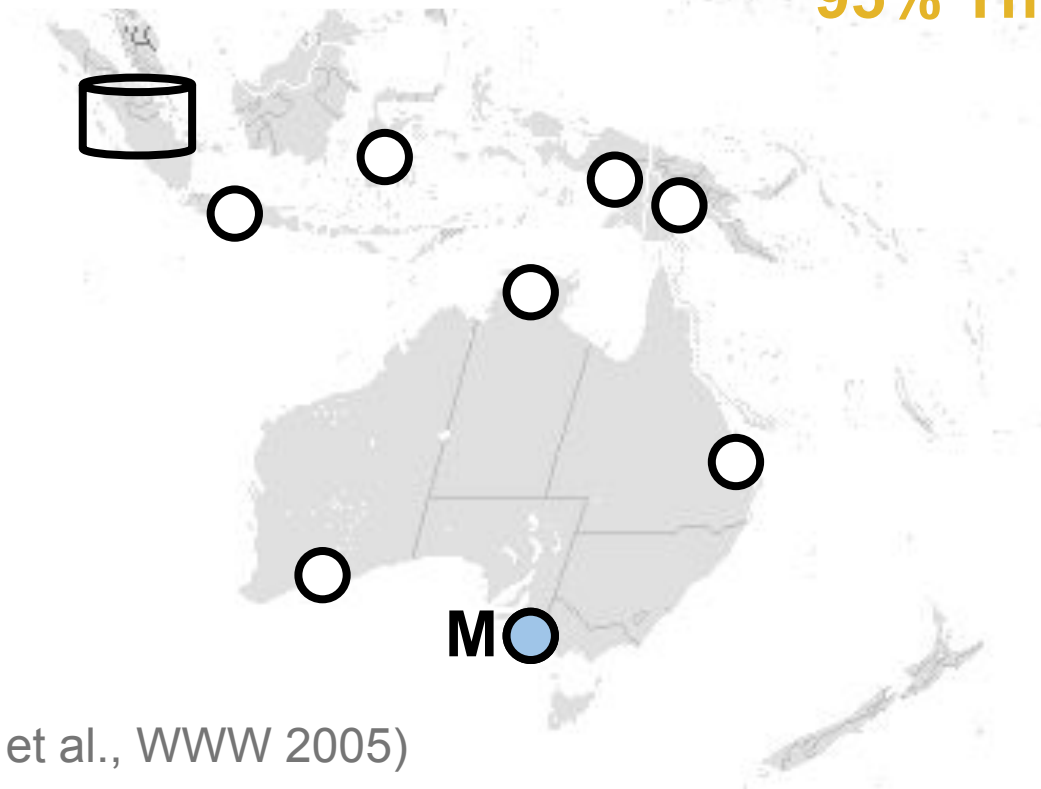
100% Threshold



(Sivasubramanian et al., WWW 2005)

Placement Heuristic

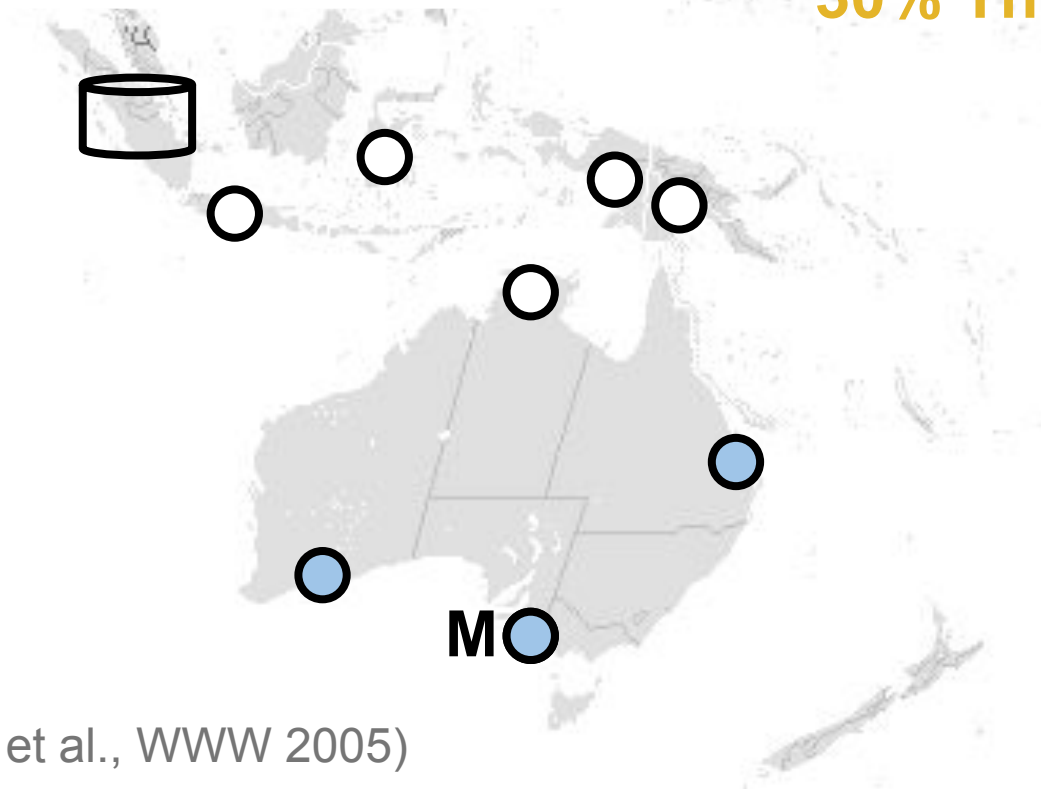
95% Threshold



(Sivasubramanian et al., WWW 2005)

Placement Heuristic

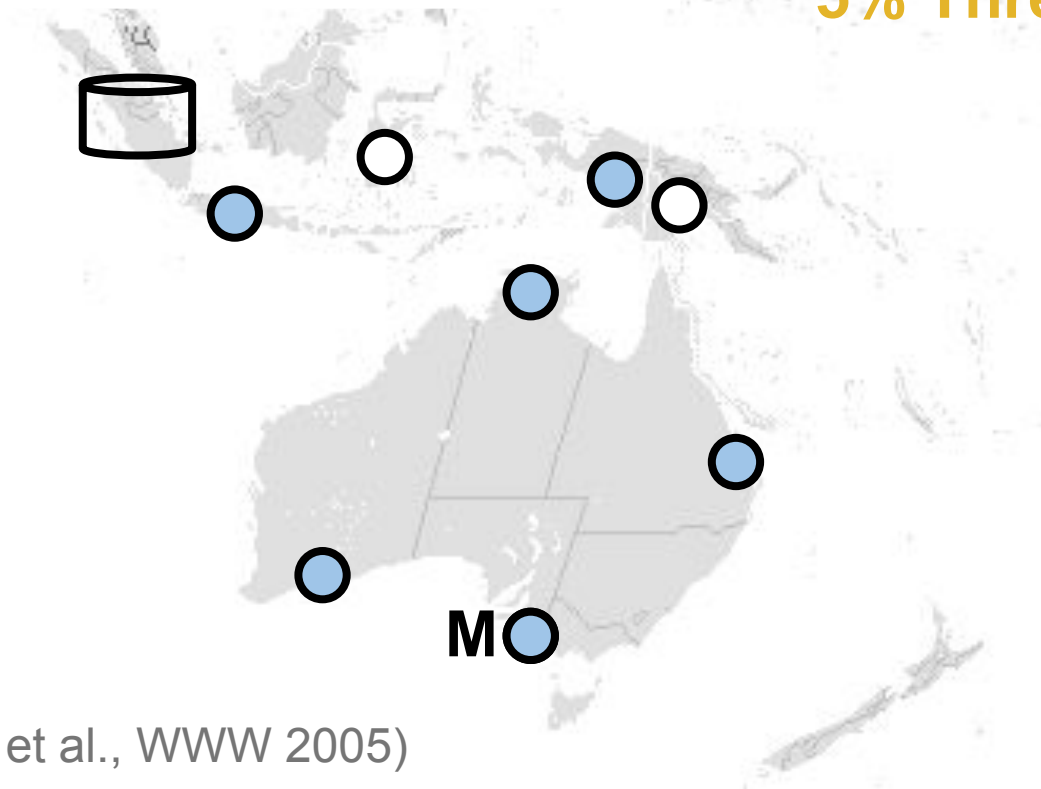
30% Threshold



(Sivasubramanian et al., WWW 2005)

Placement Heuristic

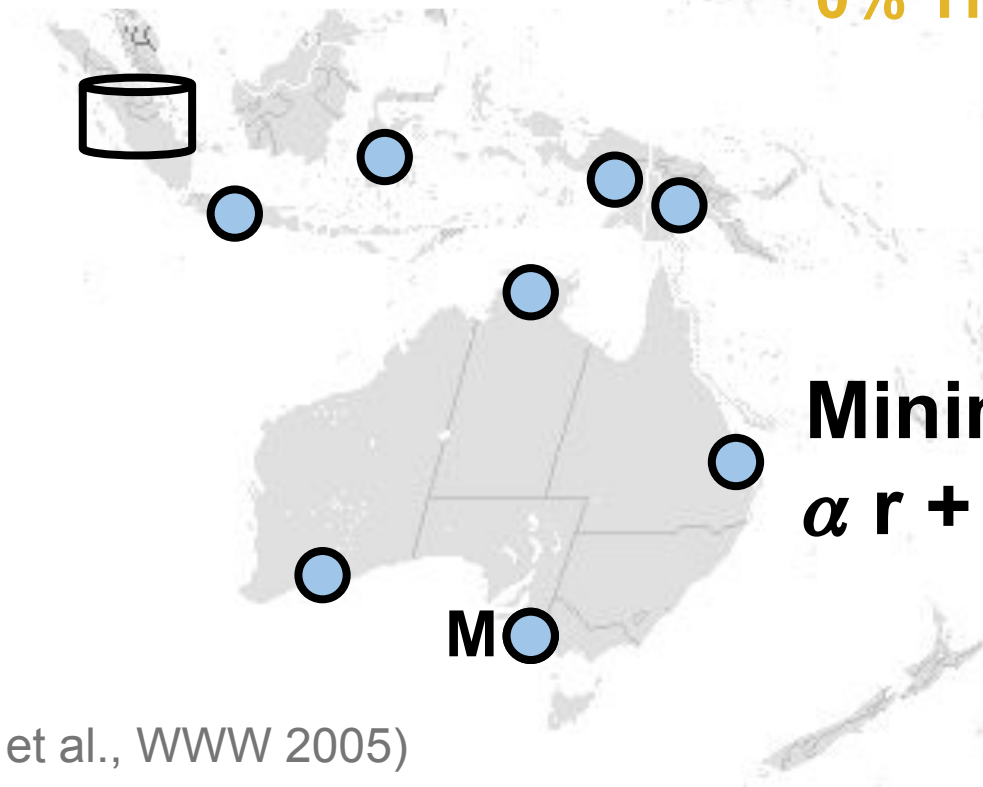
5% Threshold



(Sivasubramanian et al., WWW 2005)

Placement Heuristic

0% Threshold



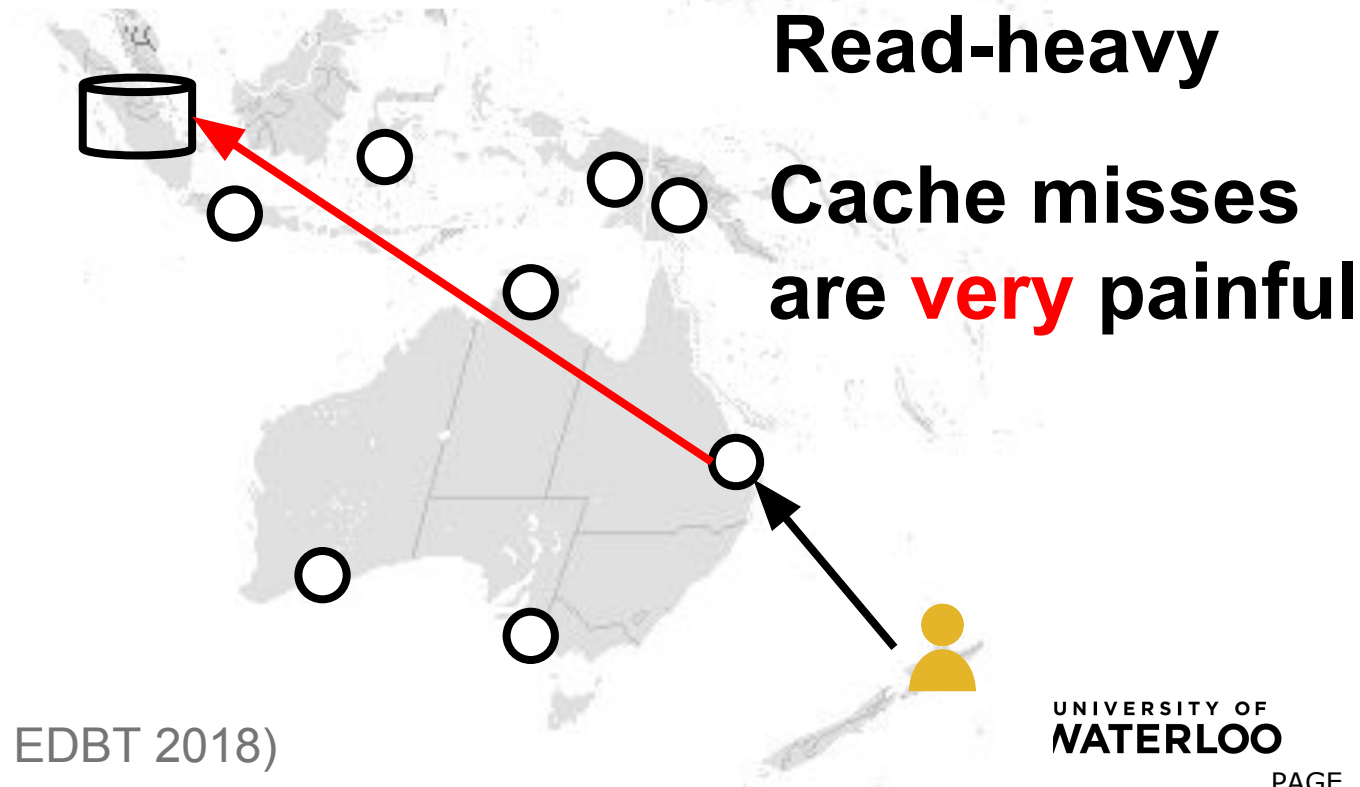
Minimize:
 $\alpha r + \beta b$

(Sivasubramanian et al., WWW 2005)

Replication Decisions

- **How many** replicas?
Cost-based given requests
- **Where to place** replicas?
Cost-based given requests
- **How to propagate** updates?
Single-master, eventual consistency

Web Workload Characteristics



(Glasbergen et al., EDBT 2018)

Web Workload Characteristics

The screenshot shows the Amazon.ca search results for "dog food". The search bar at the top contains "dog food" and is highlighted with a red box. Below the search bar, the results are sorted by "Featured". The main product displayed is "Purina ONE® SmartBlend Dry Dog Food" by ONE, with a price of CDN\$ 34.71 (Subscribe & Save) and a 5-star rating from 91 reviews. The product image shows a bag of "LAMB & RICE FORMULA FORMULE AGNEAU ET RIZ". Below the main product, there is a "Shop by Category" section with five categories: Dog Food, Home & Kitchen, Dog Treat Cookies, Biscuits & Snacks, Dog Feeding & Watering Supplies, and Bath Tissue. The left sidebar shows filters for "Pet Supplies", "Books", and "Amazon Prime".

amazon.ca prime

All Departments dog food

Deliver to Brad

Shop by Department

Brad's Store Black Friday Gift Guides Gift Cards Sell Help

EN Hello, Brad Your Account Your Prime Wish List Cart

1-16 of over 100,000 results for "dog food" Sort by Featured

Show results for

Pet Supplies

- Dog Food
- Dog Feeding & Watering Supplies
- Dog Treat Cookies, Biscuits & Snacks

Books

- Animal & Pet Care
- Dog Care

See All 13 Departments

Refine by

Amazon Prime

- prime

Dog Life Stage

- Adult
- Puppy
- Young Adult
- Senior

Pet Food Flavour

Amazon's Choice

Purina ONE® SmartBlend Dry Dog Food by ONE

CDN\$ 34.71 **Subscribe & Save**

Get scheduled, repeat delivery

★★★★☆ 91

CDN\$ 36.54 ~~CDN\$ 42.99~~ prime

Shop by Category

- Dog Food
- Home & Kitchen
- Dog Treat Cookies, Biscuits & Snacks
- Dog Feeding & Watering Supplies
- Bath Tissue

(Glasbergen et al., EDBT 2018)

Web Workload Characteristics

The screenshot shows the Amazon.ca search results page for 'dog food'. The search bar at the top contains 'dog food' and a magnifying glass icon. Below the search bar, there are navigation links for 'Deliver to Brad', 'Shop by Department', 'Brad's Store', 'Black Friday', 'Gift Guides', 'Gift Cards', 'Sell', and 'Help'. The user is logged in as 'Brad' and has a shopping cart with 0 items. The search results are sorted by 'Featured' and show 1-16 of over 20,000 results for 'Prime Eligible: "dog food"'. On the left side, there are filters for 'Pet Supplies', 'Books', and 'Refine by'. The 'Amazon Prime' filter is highlighted with a red box and is checked. The 'Dog Life Stage' filter is also visible. The main content area displays two product listings: 'Purina ONE® SmartBlend Dry Dog Food' and 'Pedigree Vitality+ Dry Food for Dogs'. Each listing includes a product image, the product name, the brand, the price (CDN\$), and the number of reviews (91 for Purina ONE and 22 for Pedigree).

amazon.ca prime

All Departments dog food

Deliver to Brad

Shop by Department

Brad's Store Black Friday Gift Guides Gift Cards Sell Help

EN Hello, Brad Your Account Your Prime Wish List Cart

1-16 of over 20,000 results for Prime Eligible: "dog food" Sort by Featured

Show results for

Amazon's Choice

Pet Supplies

- Dog Food
- Dog Feeding & Watering Supplies
- Dog Treat Cookies, Biscuits & Snacks

Books

- Animal & Pet Care
- Dog Care

See All 12 Departments

Refine by

Amazon Prime

- Clear
- prime

Dog Life Stage

- Adult
- Puppy
- Young Adult
- Senior

Purina ONE® SmartBlend Dry Dog Food

by ONE

CDN\$ 34.71 [Subscribe & Save](#)

Get scheduled, repeat delivery

★★★★☆ 91

CDN\$ 36.54 ~~CDN\$ 42.99~~ [prime](#)

FREE Delivery by Tuesday, Nov 27

Pedigree Vitality+ Dry Food for Dogs

by Pedigree

CDN\$ 26.58 [Subscribe & Save](#)


Get scheduled, repeat delivery

★★★★☆ 22


CDN\$ 27.98 ~~CDN\$ 29.98~~ [prime](#)

(Glasbergen et al., EDBT 2018)

Web Workload Characteristics



★☆☆☆☆ **Five Stars**
May 10, 2018
Flavor Name: Hearty Beef & Vegetable | Size: 14kg | **Verified Purchase**
Arrived damaged. Bag was torn. Not happy



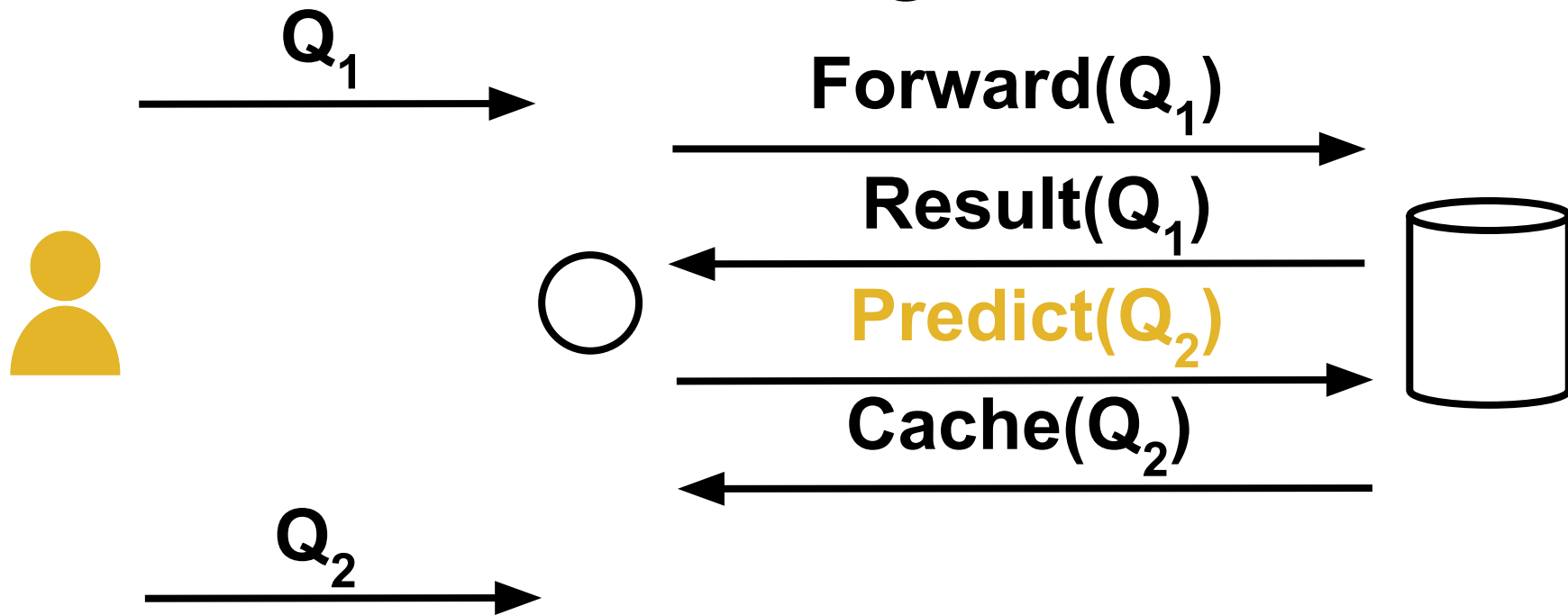
Helpful

▼ [Comment](#)

[Report abuse](#)

(Glasbergen et al., EDBT 2018)

Predictive Caching



(Glasbergen et al., EDBT 2018)

Query Patterns (TPC-W)

```
SELECT C_ID FROM CUSTOMER WHERE C_UNAME = ? and  
C_PASSWD = ?
```

```
SELECT MAX(O_ID) FROM ORDERS WHERE O_C_ID = ?
```



```
SELECT C_ID FROM CUSTOMER WHERE C_UNAME = 'Alice' and  
C_PASSWD = 'pass'
```

```
SELECT MAX(O_ID) FROM ORDERS WHERE O_C_ID = 3
```

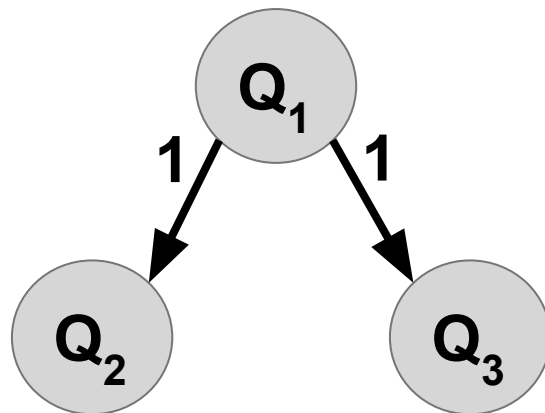
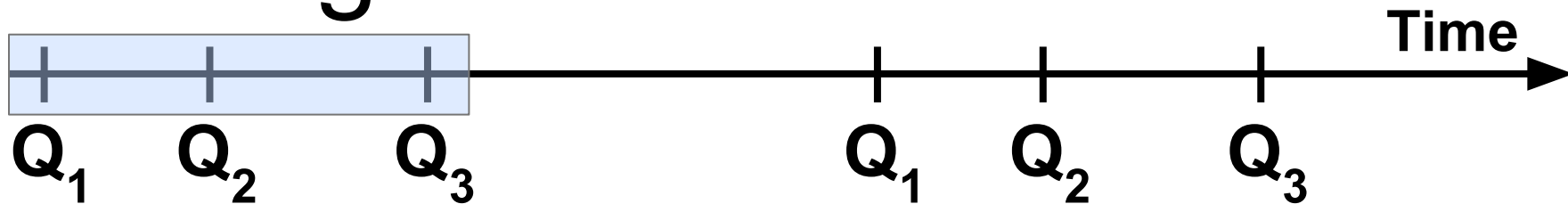


```
SELECT C_ID FROM CUSTOMER WHERE C_UNAME = ? and  
C_PASSWD = ?
```

```
SELECT MAX(O_ID) FROM ORDERS WHERE O_C_ID = ?
```

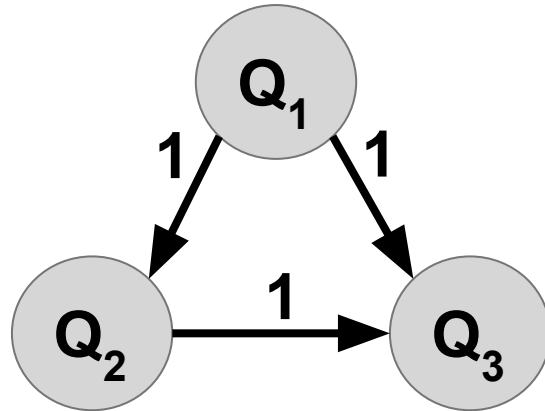
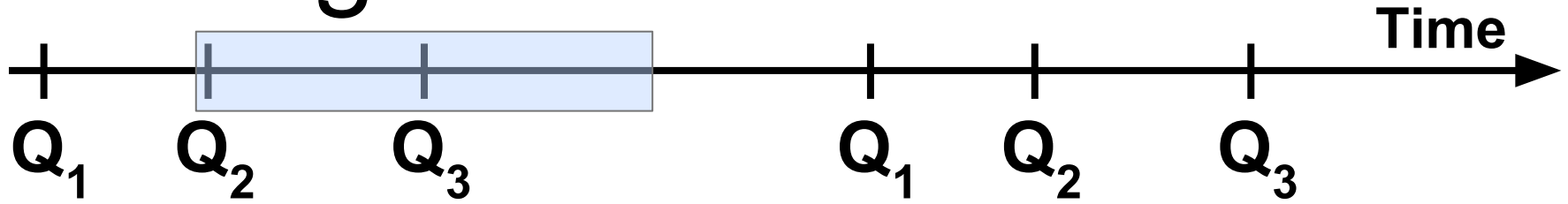


Building a Predictive Model



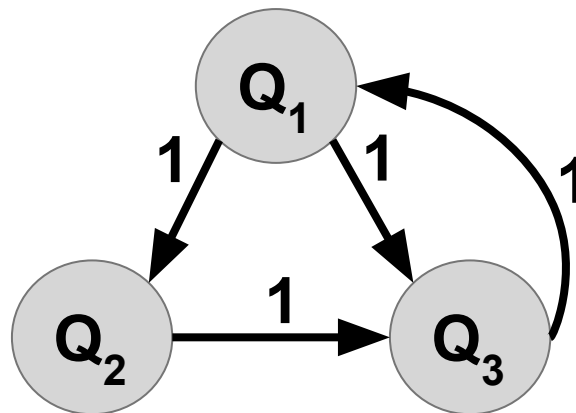
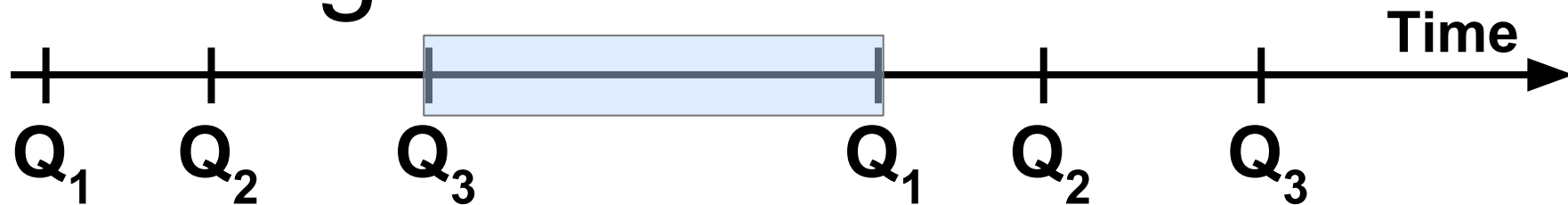
(Glasbergen et al., EDBT 2018)

Building a Predictive Model



(Glasbergen et al., EDBT 2018)

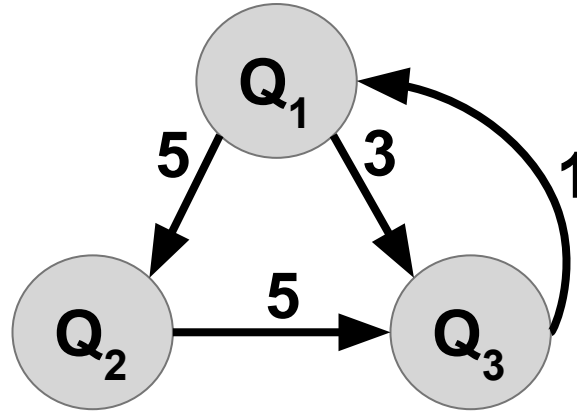
Building a Predictive Model



(Glasbergen et al., EDBT 2018)

Building a Predictive Model

All executed 5 times



100% probability Q_2 follows Q_1

20% probability Q_1 follows Q_3

Finding Parameter Mappings

```
SELECT C_ID FROM CUSTOMER WHERE C_UNAME = ? and  
C_PASSWD = ?
```

```
SELECT MAX(O_ID) FROM ORDERS WHERE O_C_ID = ?
```

(Glasbergen et al., EDBT 2018)

Finding Parameter Mappings

```
3  
SELECT C.ID FROM CUSTOMER WHERE C.UNAME = ? and  
       C.PASSWD = ?  
  
SELECT MAX(O.ID) FROM ORDERS WHERE O.C.ID = ?
```

(Glasbergen et al., EDBT 2018)

Finding Parameter Mappings

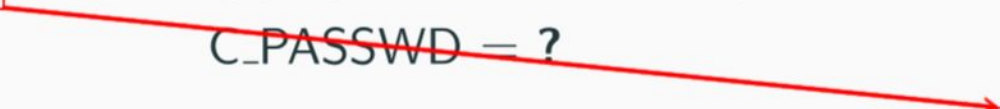
3
SELECT C_ID FROM CUSTOMER WHERE C_UNAME = ? and
C_PASSWD = ?

SELECT MAX(O_ID) FROM ORDERS WHERE O_C_ID = ?

3

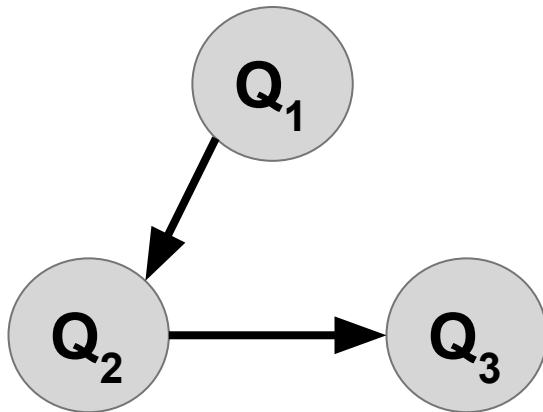
Finding Parameter Mappings

```
SELECT C.ID FROM CUSTOMER WHERE C.UNAME = ? and  
C.PASSWD = ?  
  
SELECT MAX(O.ID) FROM ORDERS WHERE O.C.ID = ?
```



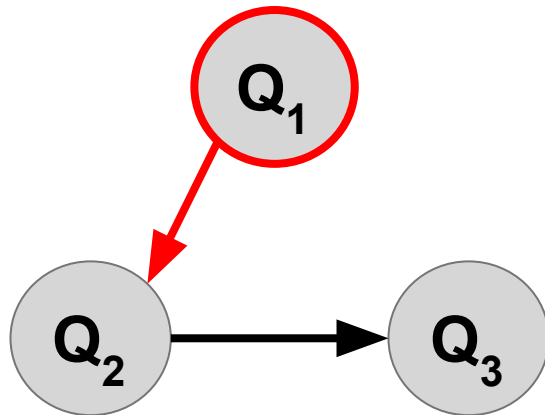
(Glasbergen et al., EDBT 2018)

Predictive Caching



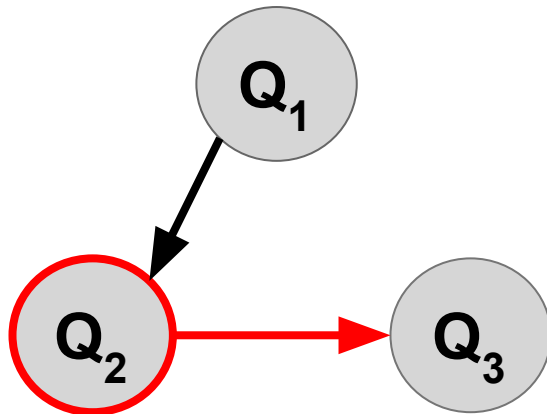
(Glasbergen et al., EDBT 2018)

Predictive Caching



Predictively Cache Q_2

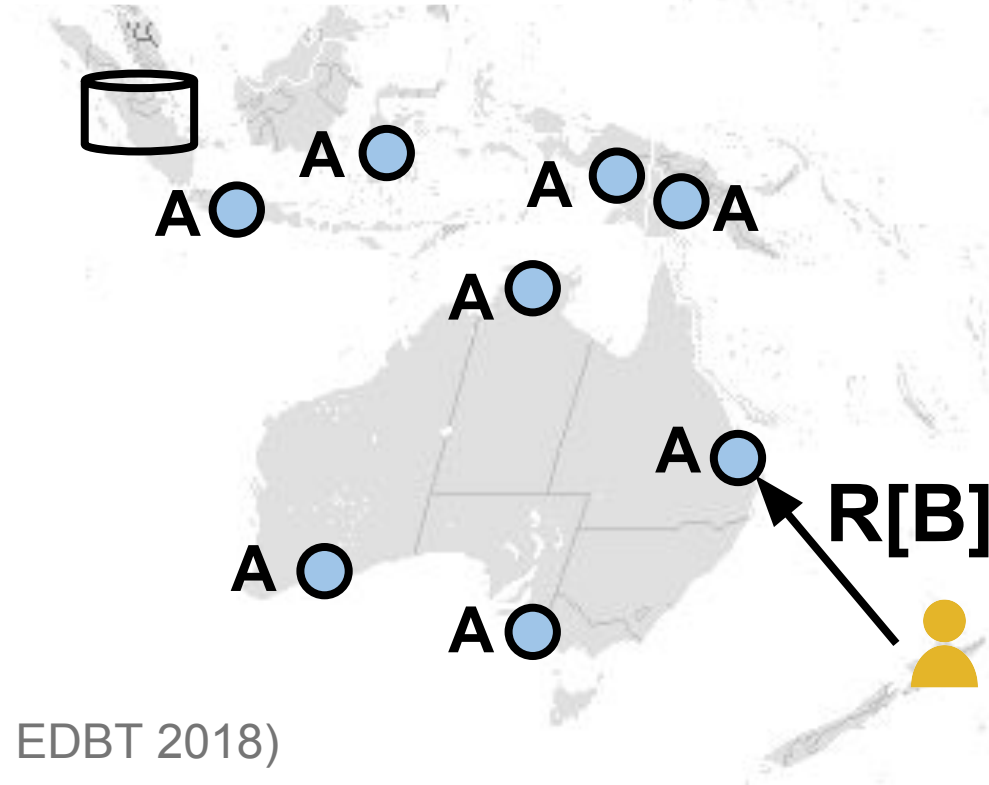
Predictive Caching



Predictively Cache Q₃

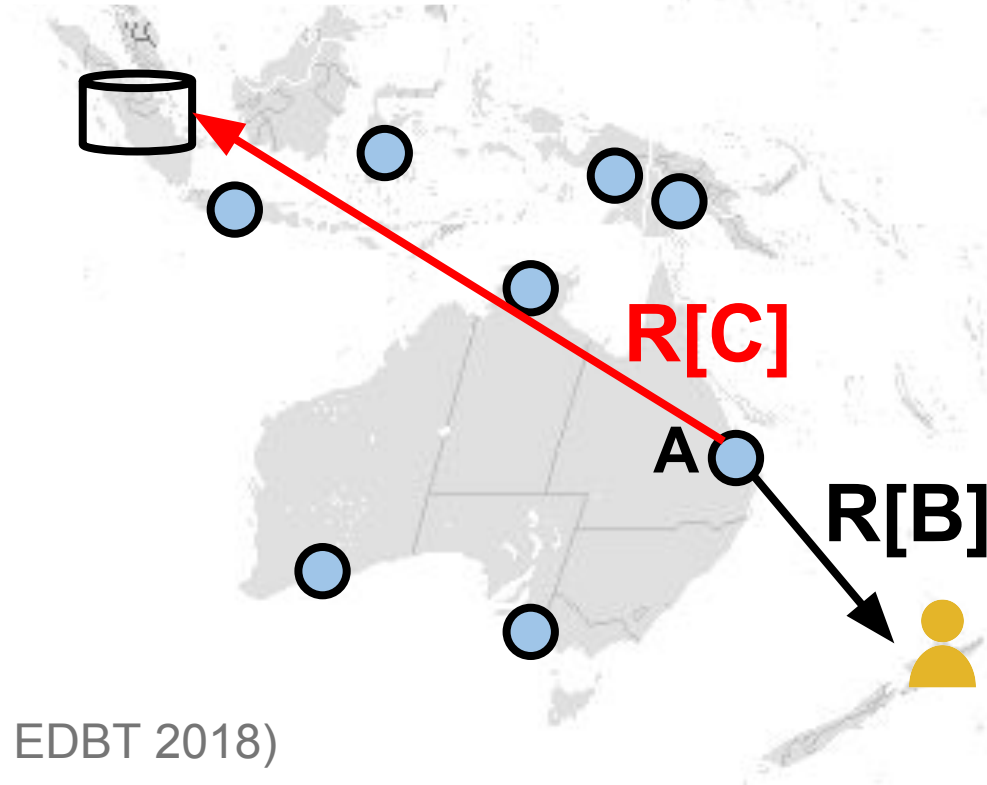
(Glasbergen et al., EDBT 2018)

Apollo Deployment



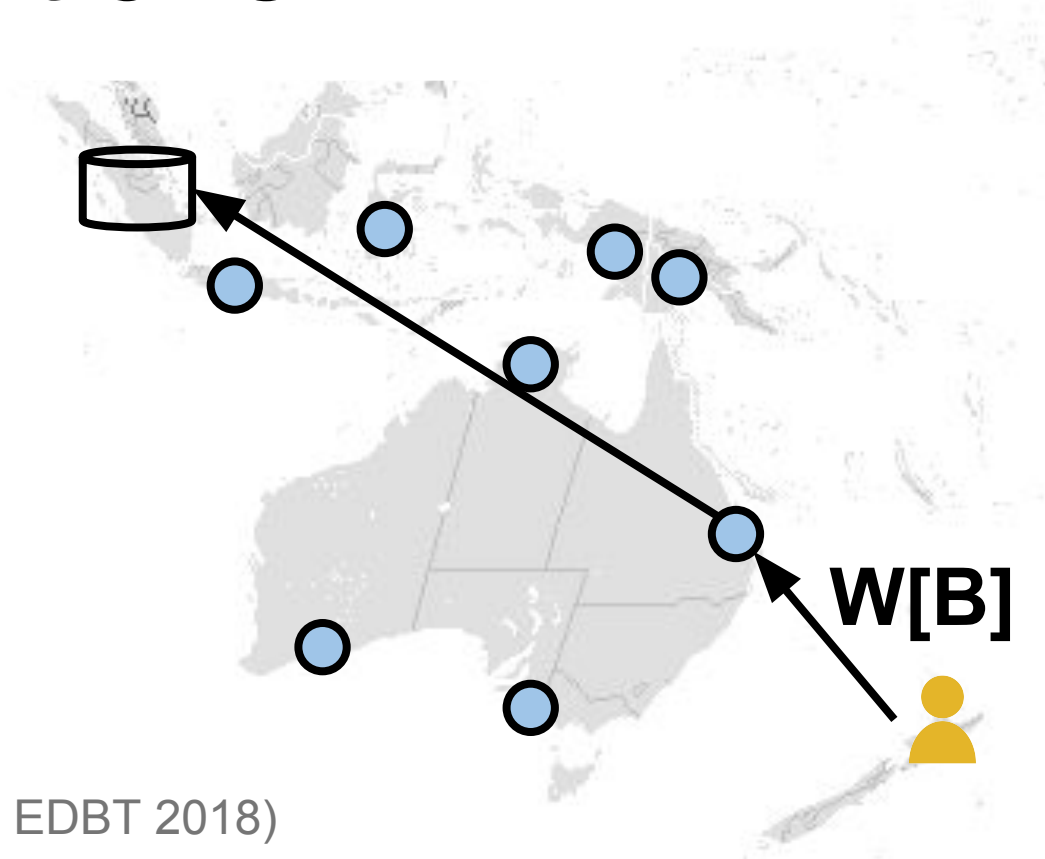
(Glasbergen et al., EDBT 2018)

Apollo Deployment



(Glasbergen et al., EDBT 2018)

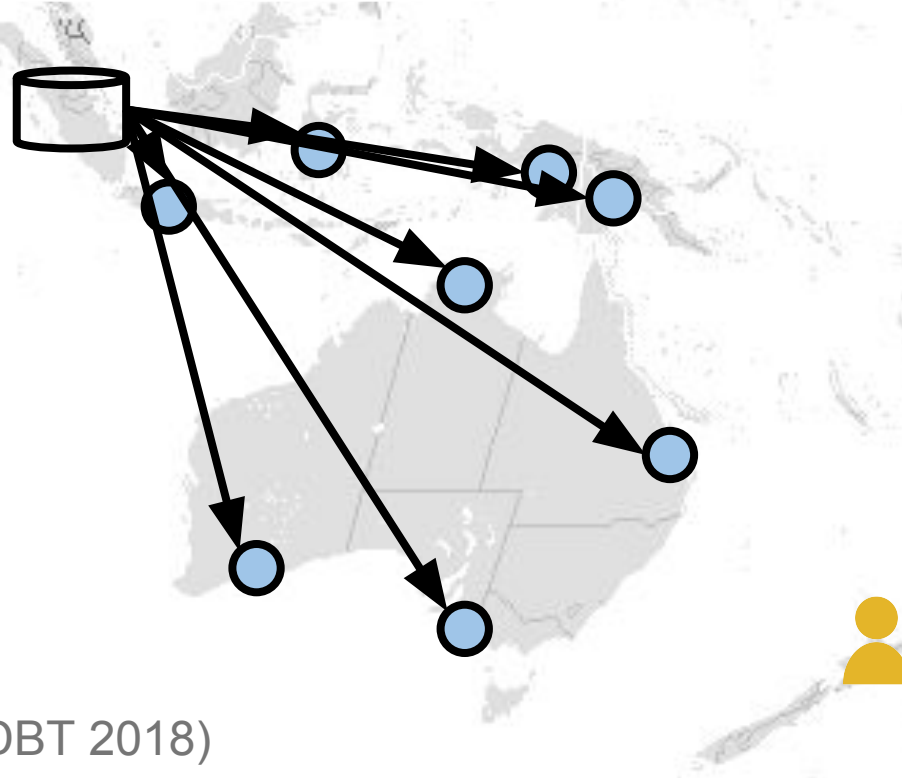
Invalidations



(Glasbergen et al., EDBT 2018)

Invalidations

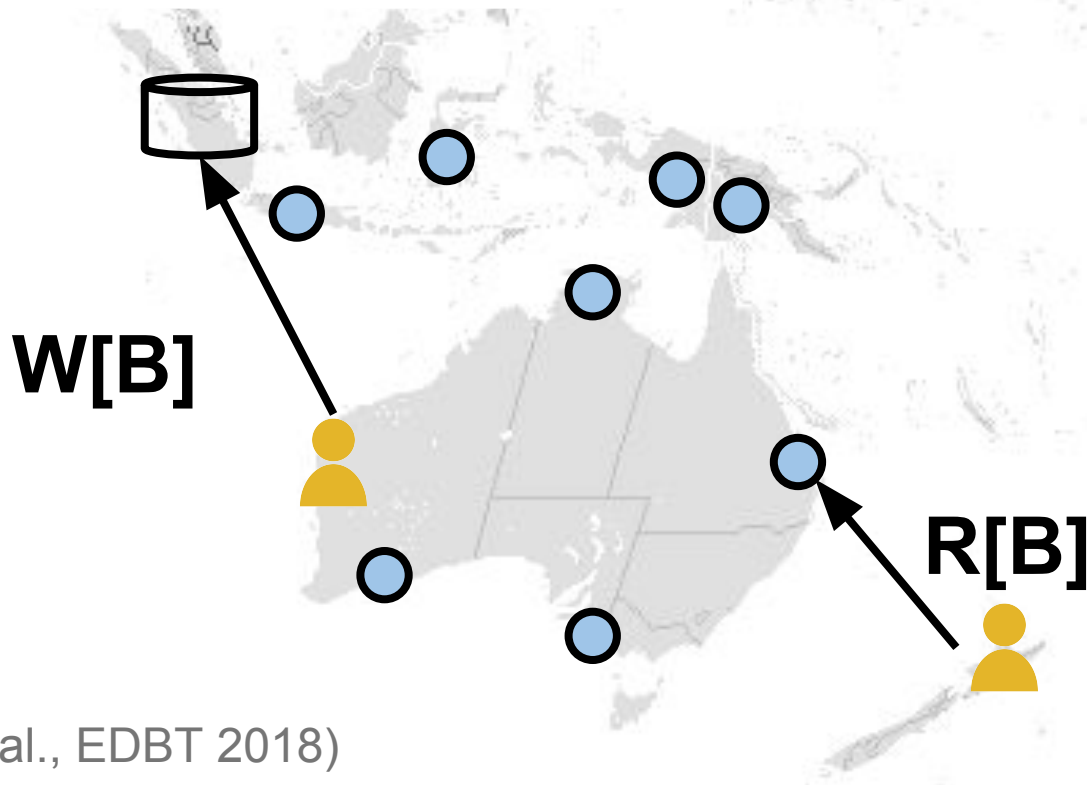
Invalidations Limit Cache Effectiveness



(Glasbergen et al., EDBT 2018)



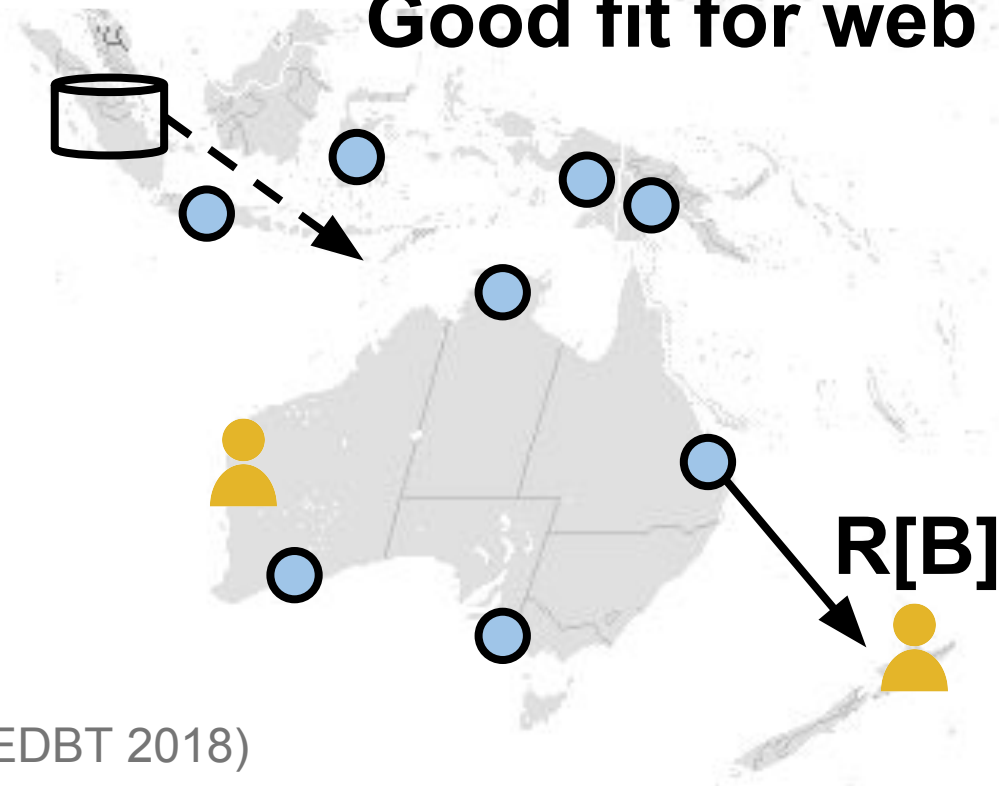
Session Semantics



(Glasbergen et al., EDBT 2018)

Session Semantics

Good fit for web data!



(Glasbergen et al., EDBT 2018)

Replication Decisions

- **How many** replicas?

Predictively based on requests

- **Where to place** replicas?

Client edge cache, predictively

- **How to propagate** updates?

Cache updates with sessions







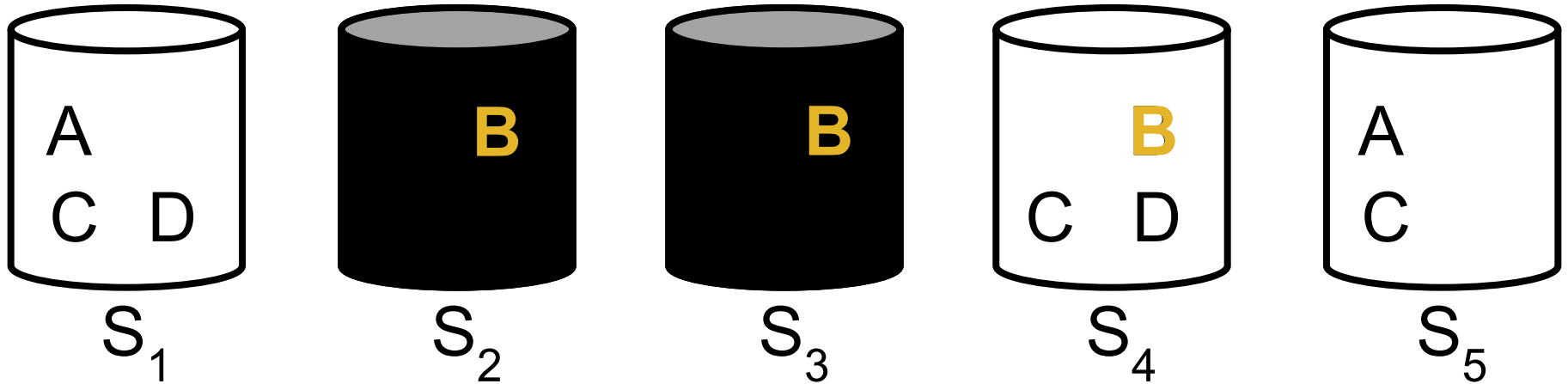
Replication for Availability

Failures are common

Data systems must remain **available**

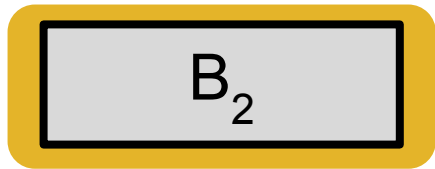
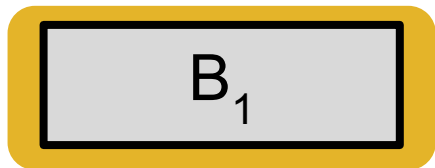
Replication for Availability

Tolerating **r faults** requires **$r + 1$ replicas**

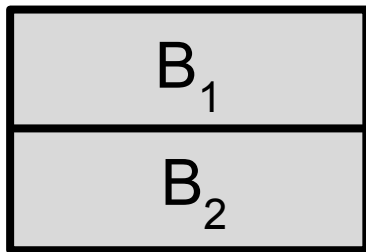


Lower Overhead

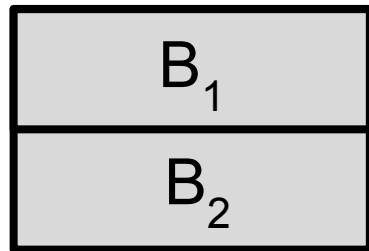
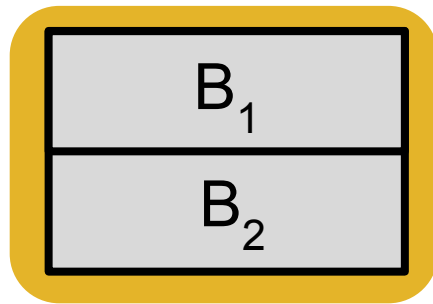
XOR



Data



Replicas

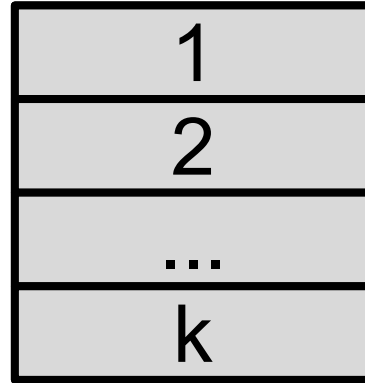
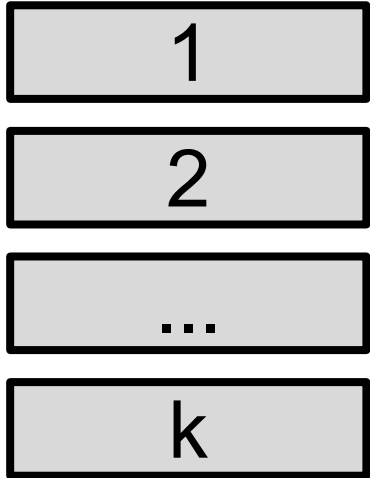


$$B_2 \text{ xor } (B_1 \text{ xor } B_2) = B_1$$

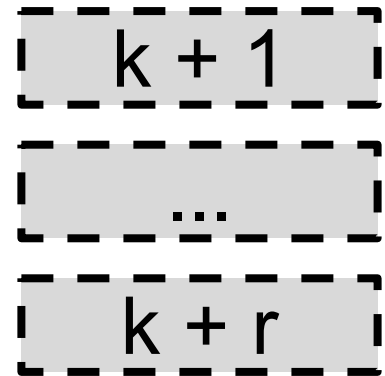
Erasure Coding

Data

k partitions



r parity partitions



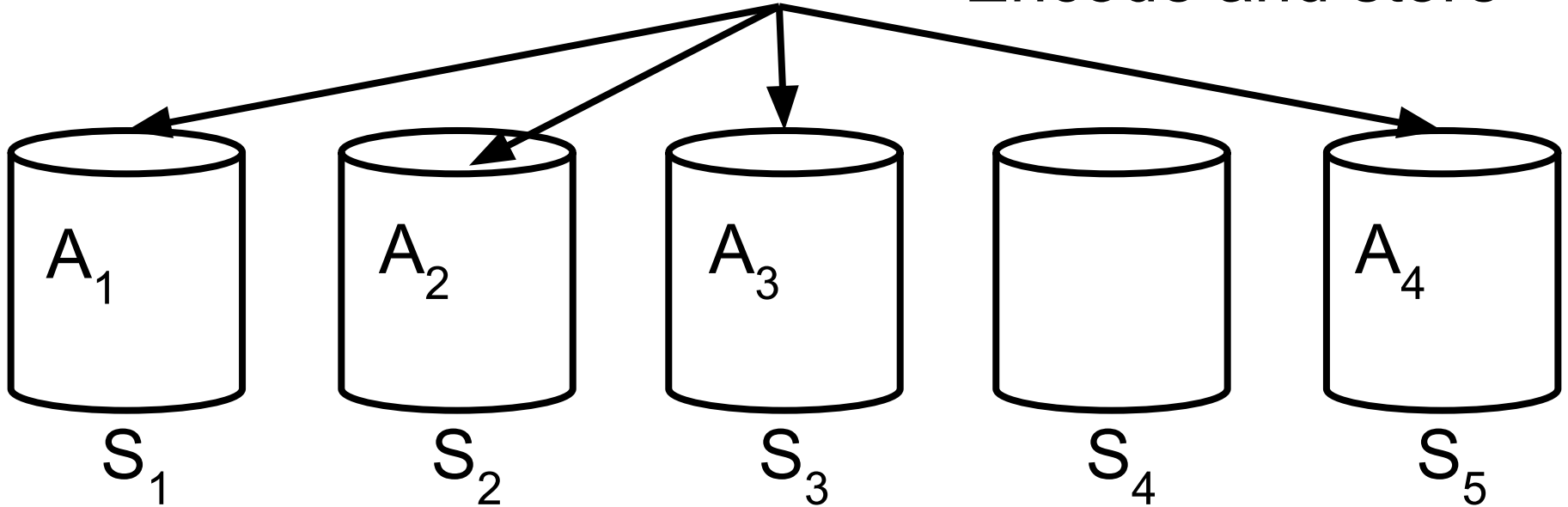
Tolerating **r faults**
requires **$(k+r)/k$** space

Erasure Coding

$k = 2, r = 2$

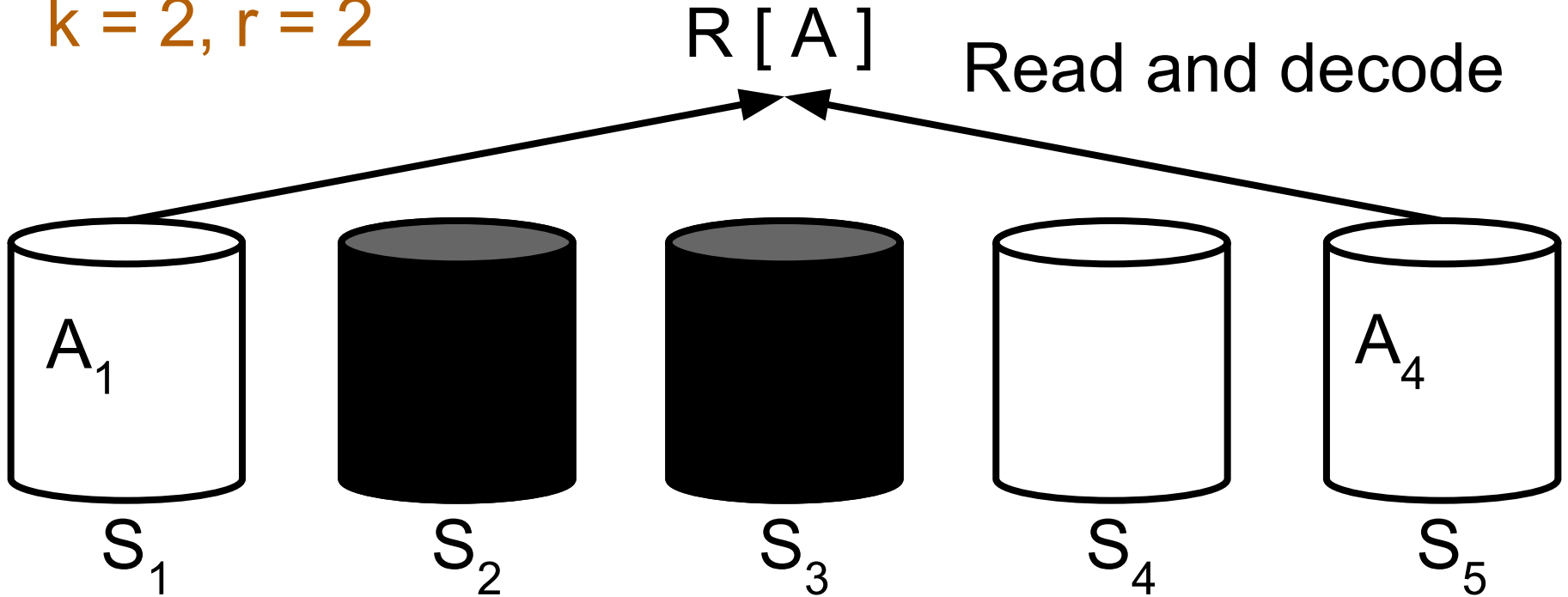
$W[A]$

Encode and store



Erasure Coding

$k = 2, r = 2$



Erasure Coding

Reduces storage overhead

Requires **parallel** retrieval

Erasure Coded Storage

Where to **place** data

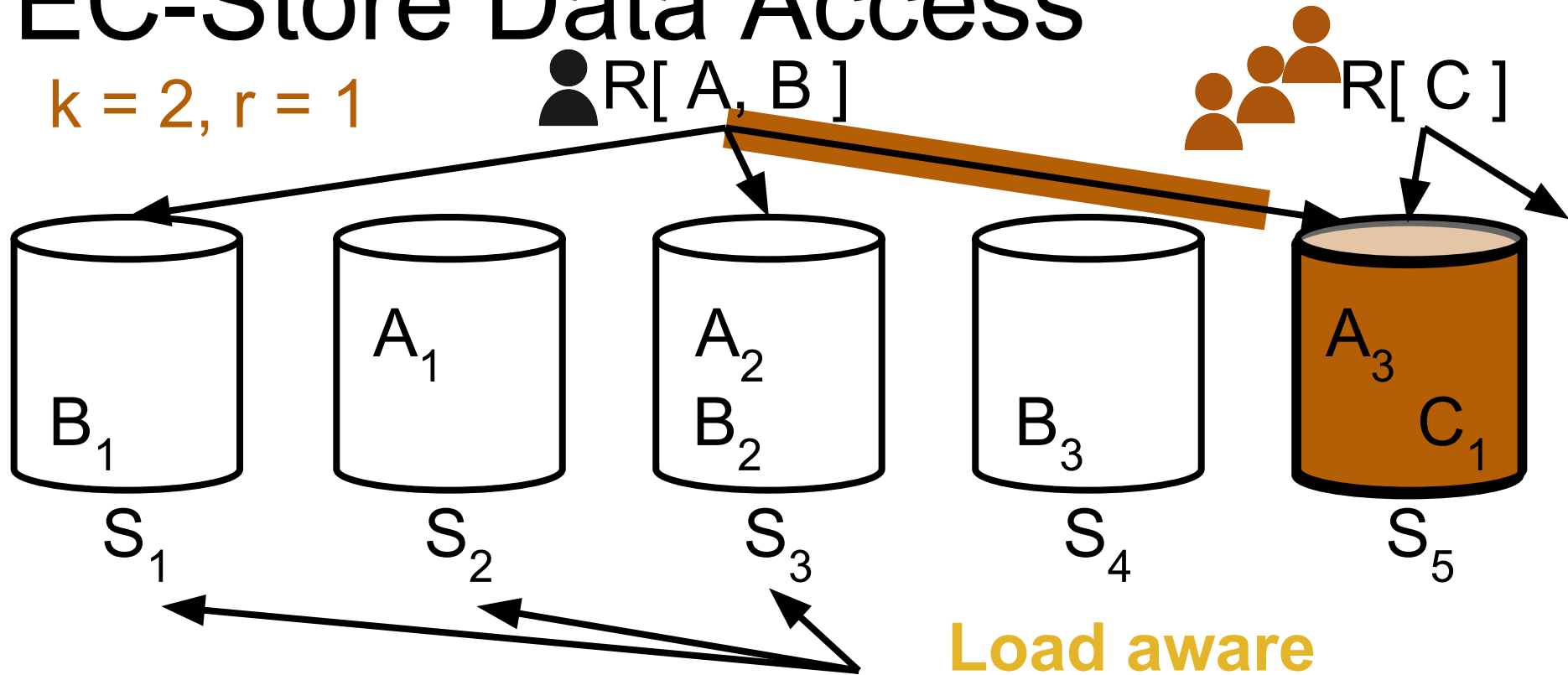
How to **access** data

EC-Store

(Abebe, ICDCS 2018)

EC-Store Data Access

$k = 2, r = 1$



(Abebe et al., ICDCS 2018)

 $R[A, B]$

EC-Store Data Access

Access Strategy: Minimize **cost of access**

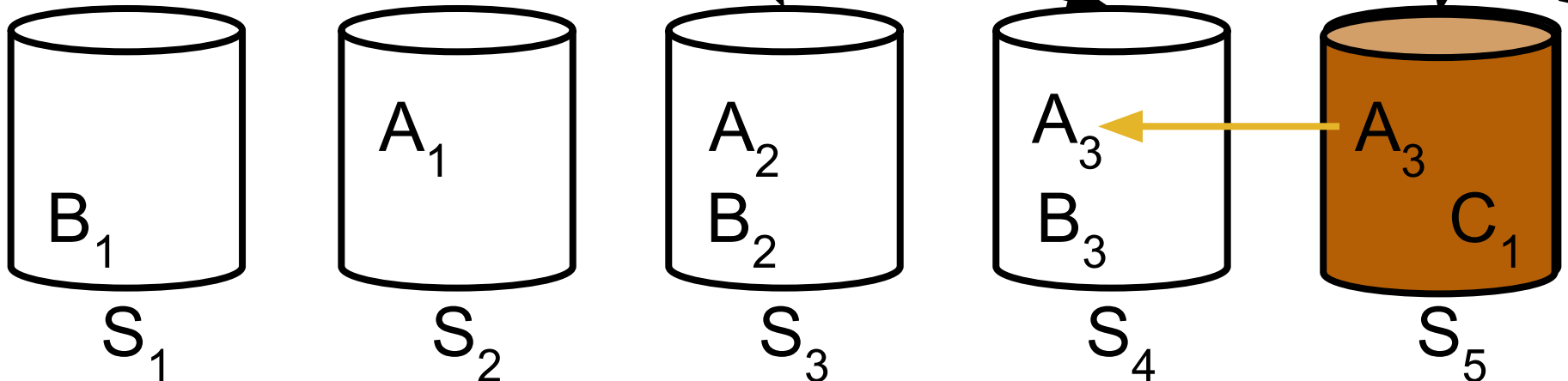
Cost of site access: **load** at site + **I/O** at site

EC-Store Data Movement

$k = 2, r = 1$

 R[A, B]

 R[C]



Load aware

(Abebe et al., ICDCS 2018)

 R[A, B]

EC-Store Data Movement

Move data to **minimize** cost of **future accesses** and **balance** **system load**

Model access patterns to **predict** **future accesses**

(Abebe et al., ICDCS 2018)

Replication Decisions

- **How many** replicas?

Fault tolerance requirements

- **Where to place** replicas?

Dynamic movement, using access costs

- **How to propagate** updates?

Synchronous updates

Road Map

- Adaptive Replication
- Adaptive Partitioning
- Outlook

Adaptive Partitioning

Partitioning Decisions

- **How to form** partitions?
- **Where to place** partitions?
- **How to execute** multi-partition operations?

Adaptive Partitioning

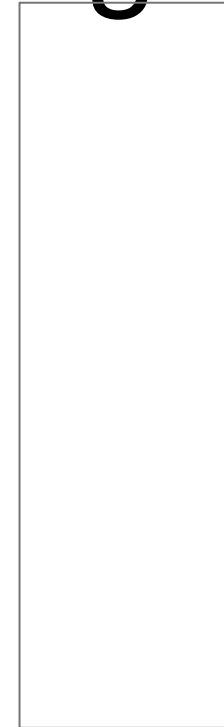
- Iterative improvements
- Partitioning per request
- Considering the overall workload
 - Heuristics





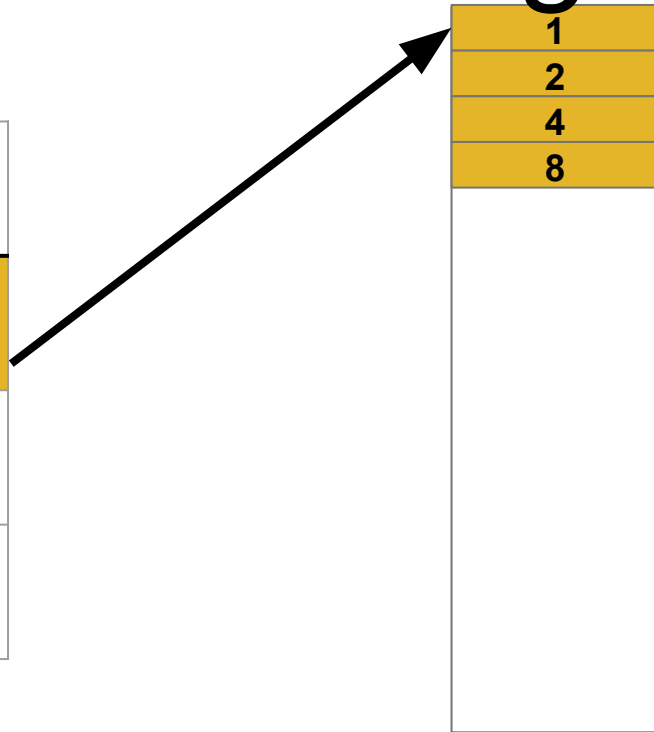
Physical Database Design

A	B	C	D
1	2	4	2
2	4	6	8
3	6	7	5



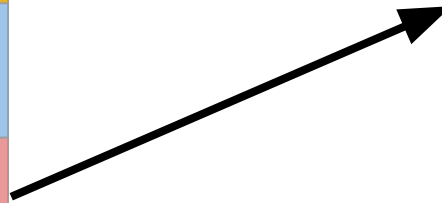
Physical Database Design

A	B	C	D
1	2	4	8
2	4	6	3
3	6	7	10



Physical Database Design

A	B	C	D
1	2	4	8
2	4	6	3
3	6	7	10



1
2
4
8
2
4
6
3
3
6
7
10

Physical Database Design

**SELECT AVERAGE(C)
FROM R WHERE R.D > 5;**

A	B	C	D
1	2	4	8
2	4	6	3
3	6	7	10

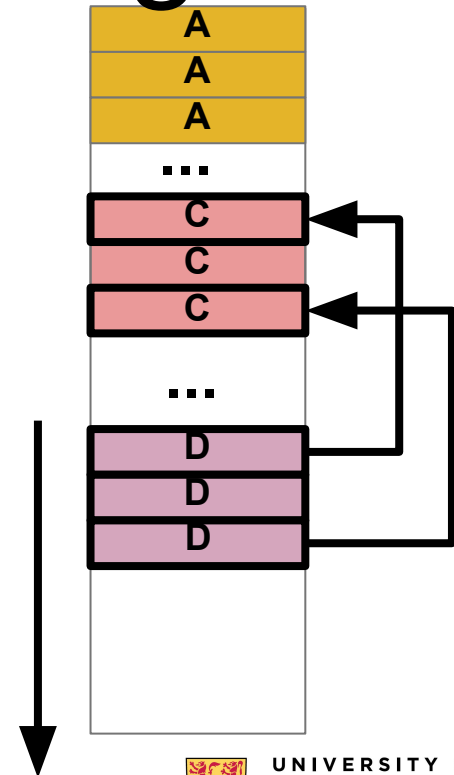


Analytic Database Design

**SELECT AVERAGE(C) FROM
R WHERE R.D > 5;**

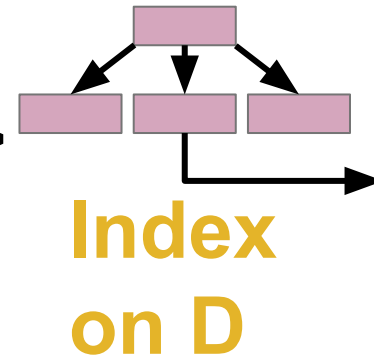
A	B	C	D
1	2	4	8
2	4	6	3
3	6	7	10

Scan



Analytic Database Design

```
SELECT AVERAGE(C)
FROM R WHERE R.D >
5;
```



A
A
A
...
C
C
C
...
D
D
D

**Need to know what to
index upfront**

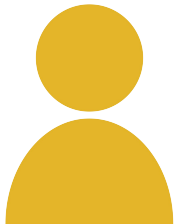


Adaptive Range Indexing

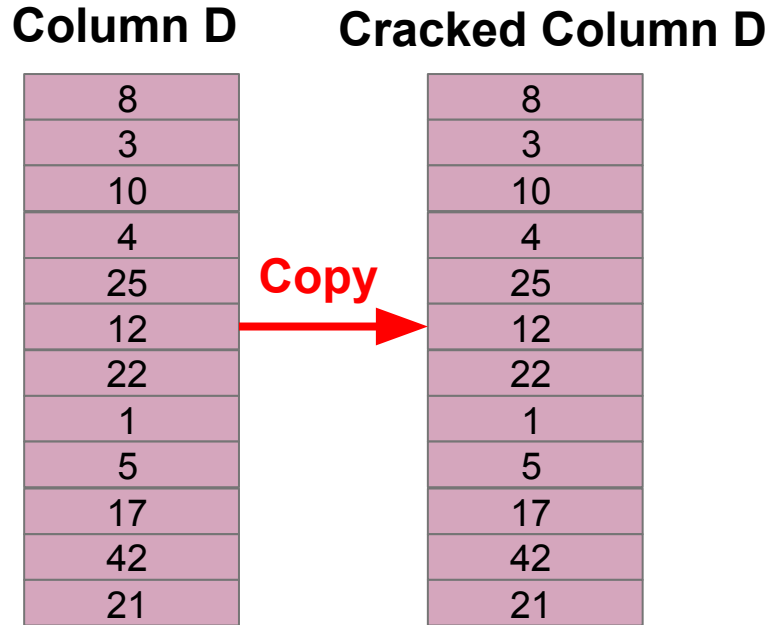
SELECT ... FROM R WHERE
R.D > 5;

SELECT ... FROM R WHERE
R.D > 5 AND R.D < 10;

SELECT ... FROM R WHERE
R.D > 10 AND R.D < 20;



Database Cracking



(Idreos et al., CIDR 2007)

Indexes via Partitioning

Cracked Column D

8
3
10
4
25
12
22
1
5
17
42
21

SELECT ... WHERE
R.D > 5

(Idreos et al., CIDR 2007)

Indexes via Partitioning

SELECT ... WHERE
R.D > 5

Cracked Column D

8	←
3	
10	
4	
25	
12	
22	
1	
5	
17	
42	
21	←

Indexes via Partitioning

SELECT ... WHERE
R.D > 5

Cracked Column D

8	←
3	
10	
4	
25	
12	
22	
1	
5	
17	
42	←
21	

Indexes via Partitioning

**SELECT ... WHERE
R.D > 5**

Cracked Column D

8	←
3	
10	
4	
25	
12	
22	
1	
5	←
17	
42	
21	

Swap

Indexes via Partitioning

**SELECT ... WHERE
R.D > 5**

Cracked Column D

5	←
3	
10	
4	
25	
12	
22	
1	
8	←
17	
42	
21	

Swap

Indexes via Partitioning

Cracked Column D

5
3
10
4
25
12
22
1
8
17
42
21

← Swap

**SELECT ... WHERE
R.D > 5**

Indexes via Partitioning

Cracked Column D

5
3
1
4
25
12
22
10
8
17
42
21

← Swap

SELECT ... WHERE
R.D > 5

Indexes via Partitioning

Cracked Column D

5
3
1
4
25
12
22
10
8
17
42
21

SELECT ... WHERE
R.D > 5

Indexes via Partitioning

**SELECT ... WHERE
R.D > 5 AND R.D <
10**

Cracked Column D

5
3
1
4
25
12
22
10
8
17
42
21

**Only need to
consider
these**

Indexes via Partitioning

**SELECT ... WHERE
R.D > 5 AND R.D <
10**

Cracked Column D

5
3
1
4
8
10
17
12
25
22
42
21

Indexes via Partitioning

**SELECT ... WHERE
R.D > 10 AND R.D <
20**

Cracked Column D

5
3
1
4
8
10
17
12
25
22
42
21

Only need to
consider
these

Indexes via Partitioning

**SELECT ... WHERE
R.D > 10 AND R.D <
20**

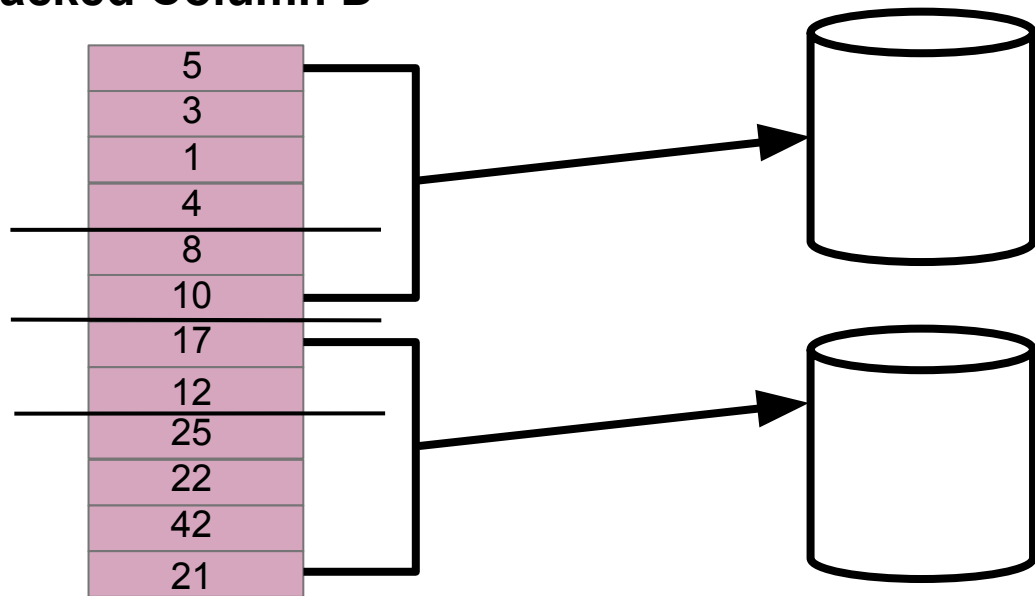
Cracked Column D

5
3
1
4
8
10
17
12
25
22
42
21

**Iterative
Partitioning for
Indexing**

Cracking: Extensions

Cracked Column D



**Advanced
Cracking
Methods**

Distribution

(Idreos et al., CIDR 2007)

Partitioning Decisions

- How to **form** partitions?

Iteratively, based on queries

- Where to **place** partitions?

Sorted in memory

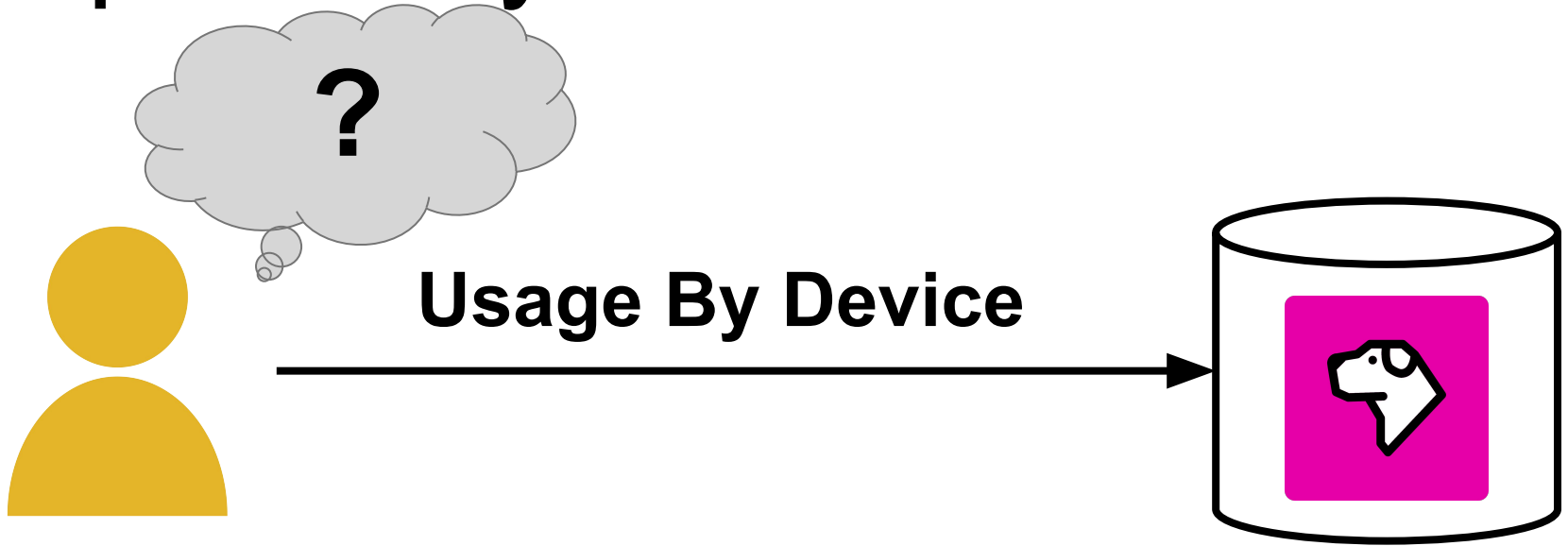
- How to **execute** multi-partition operations?

N/A

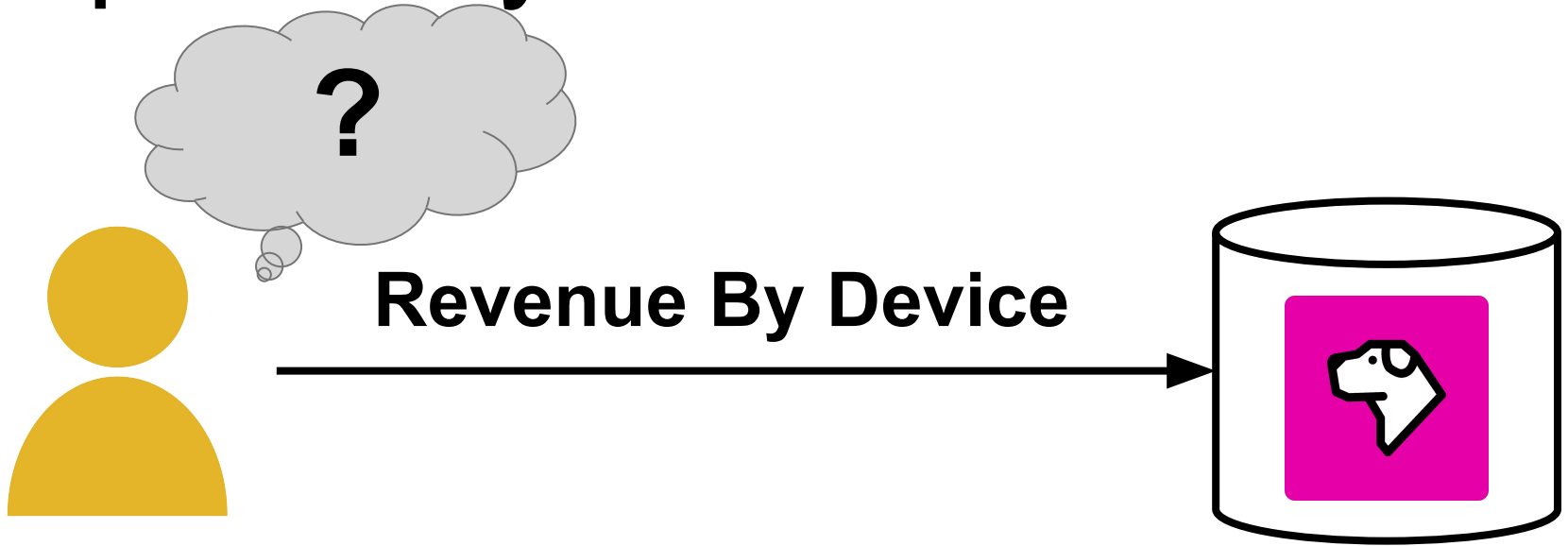
Exploratory Workloads



Exploratory Workloads



Exploratory Workloads



Exploratory Workloads



No upfront information, need generic partitioning!

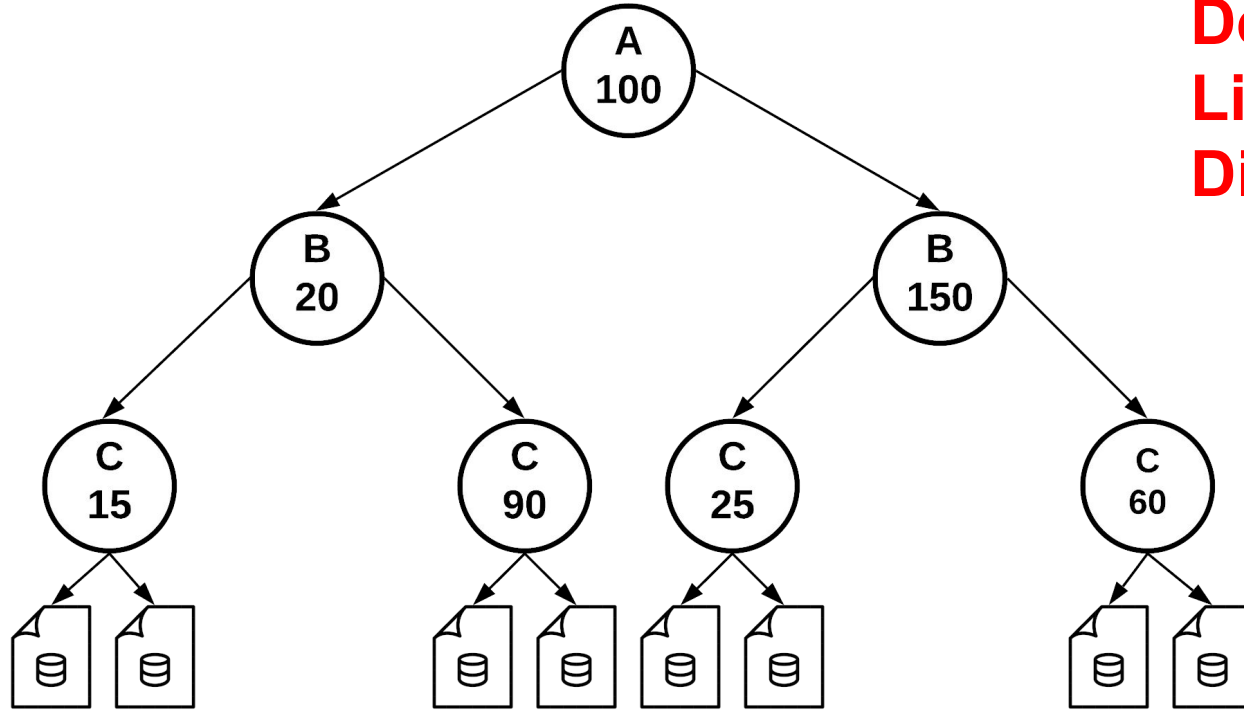
Initial Partitioning (KD-Tree)

512 MB

256 MB

128 MB

64 MB

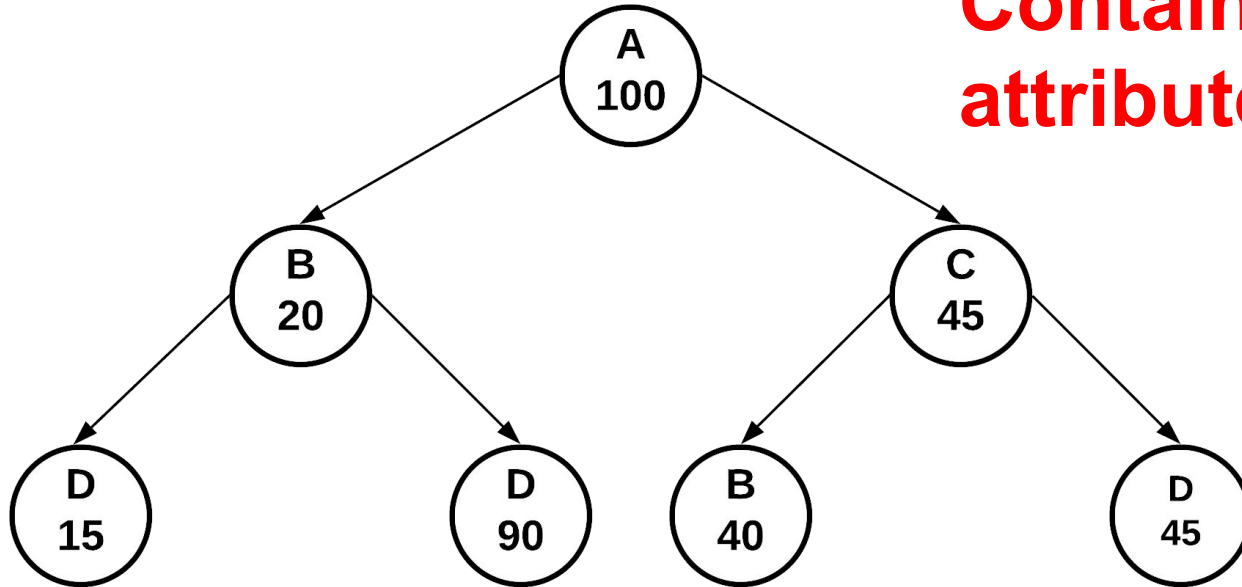


Depth
Limits
Division

(Shanbhag et al., SoCC 2017)

Heterogeneous Tree

Contains more attributes!



Building the Partitioning

A: 0.0

B: 0.0

C: 0.0

D: 0.0

(Shanbhag et al., SoCC 2017)

Building the Partitioning

A
100

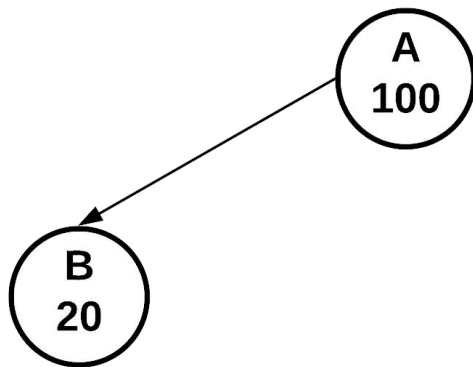
A: 1.0

B: 0.0

C: 0.0

D: 0.0

Building the Partitioning



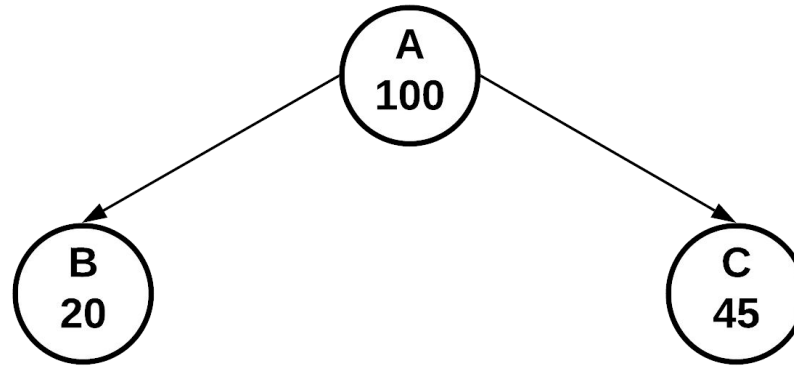
A: 1.0

B: 0.5

C: 0.0

D: 0.0

Building the Partitioning



A: 1.0

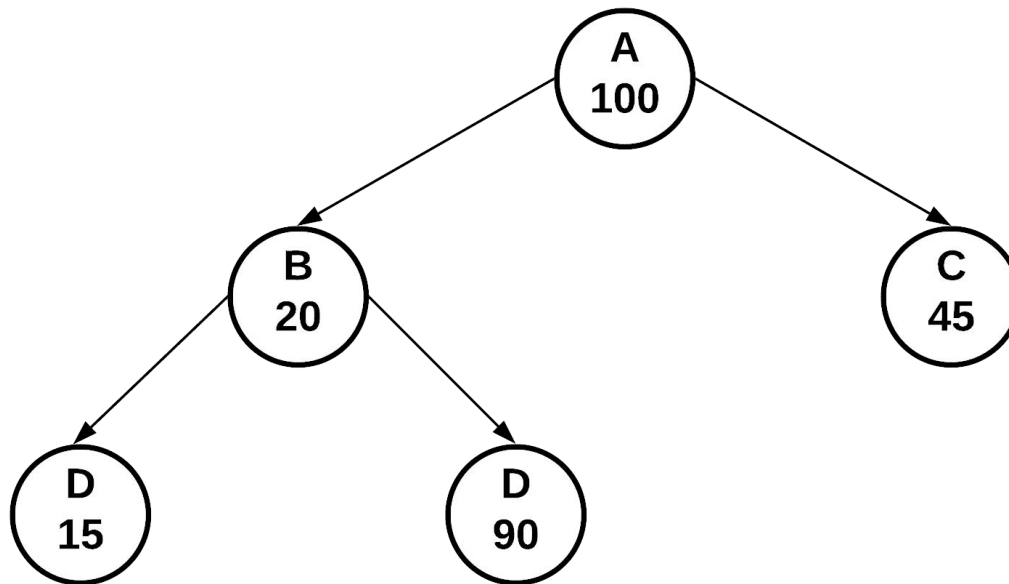
B: 0.5

C: 0.5

D: 0.0

(Shanbhag et al., SoCC 2017)

Building the Partitioning



A: 1.0

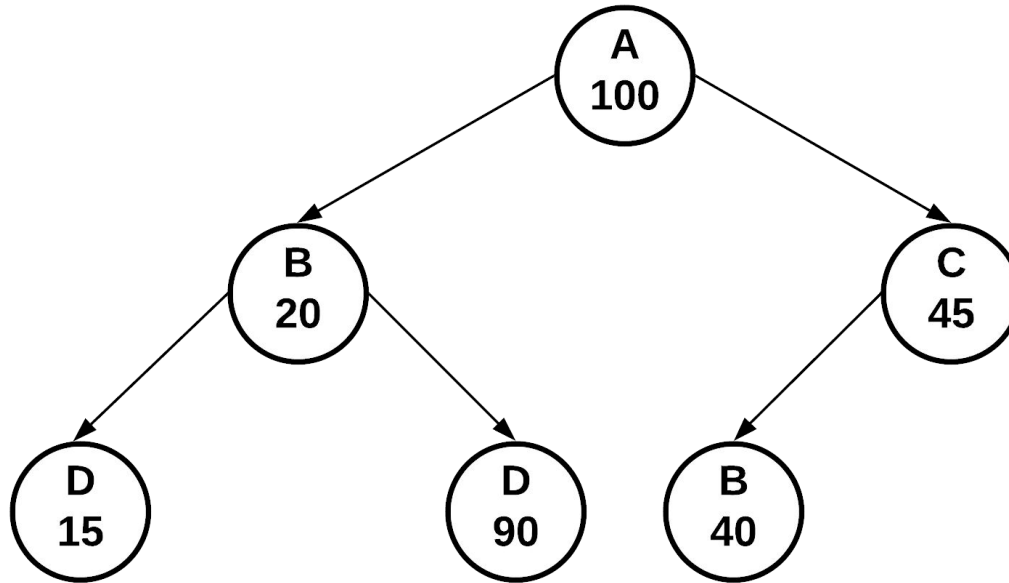
B: 0.5

C: 0.5

D: 0.5

(Shanbhag et al., SoCC 2017)

Building the Partitioning



A: 1.0

B: 0.75

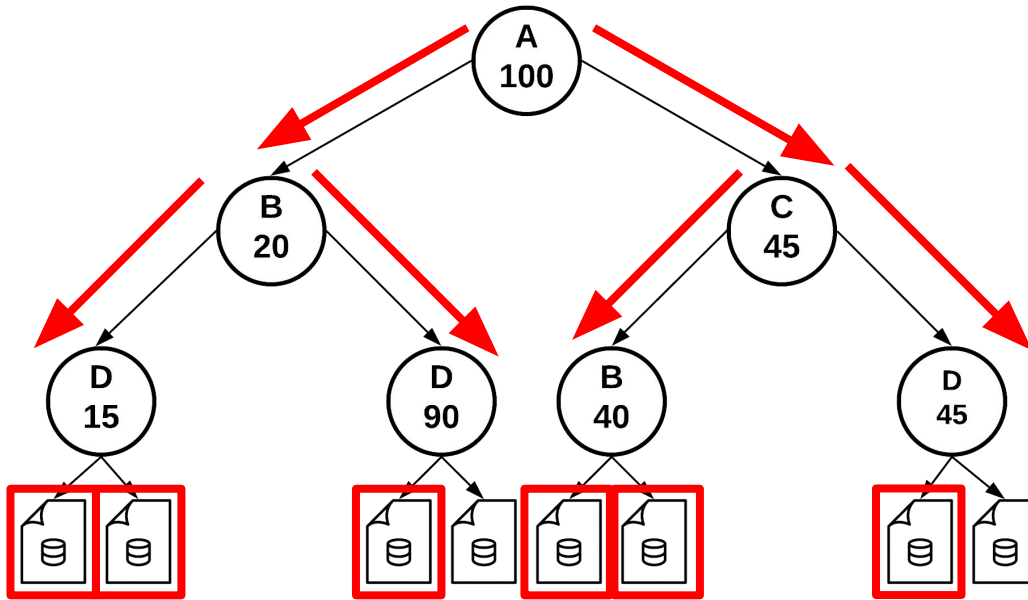
C: 0.5

D: 0.5

(Shanbhag et al., SoCC 2017)

Adaptive Partitioning

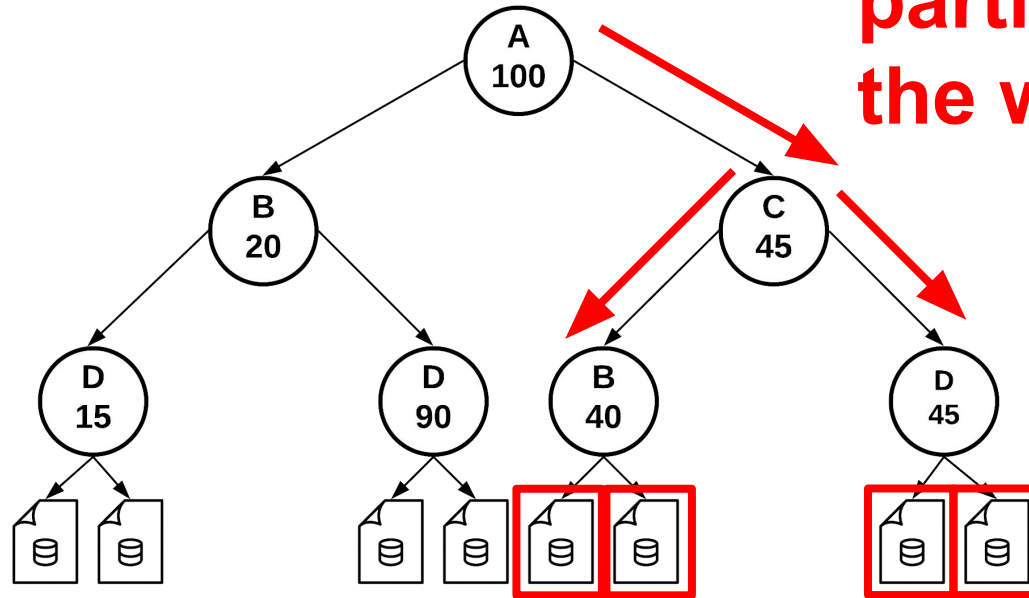
$$Q_1 = \sigma_{D \leq 45}$$



(Shanbhag et al., SoCC 2017)

Adaptive Partitioning

$$Q_2 = \sigma_{A \geq 125}$$

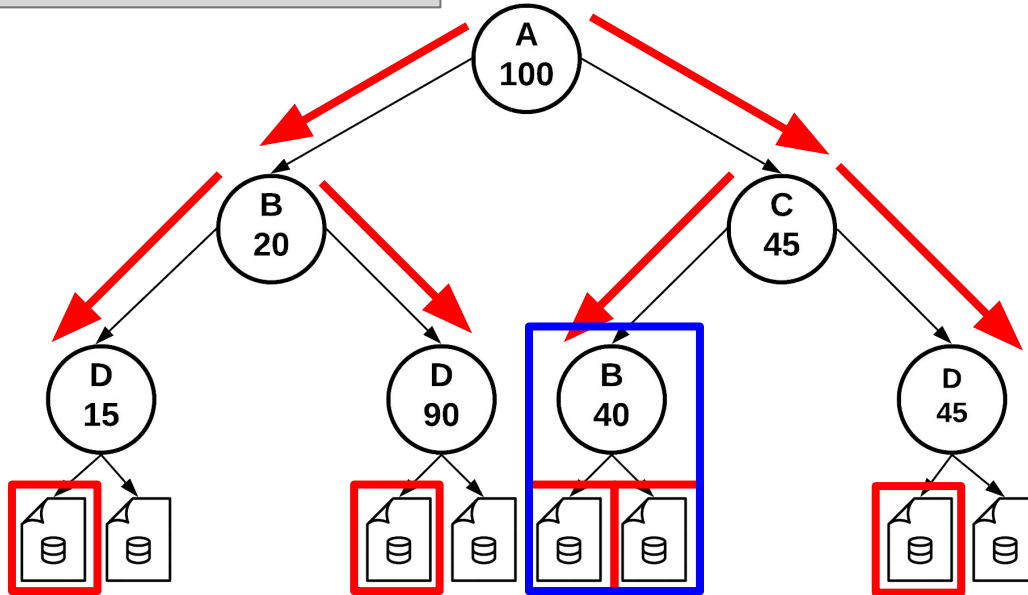


**Refine
partitioning per
the workload!**

Adaptive Partitioning: When?

$Q_1, Q_2, Q_3, Q_1, \dots$

$$Q_1 = \sigma_{D \leq 45}$$



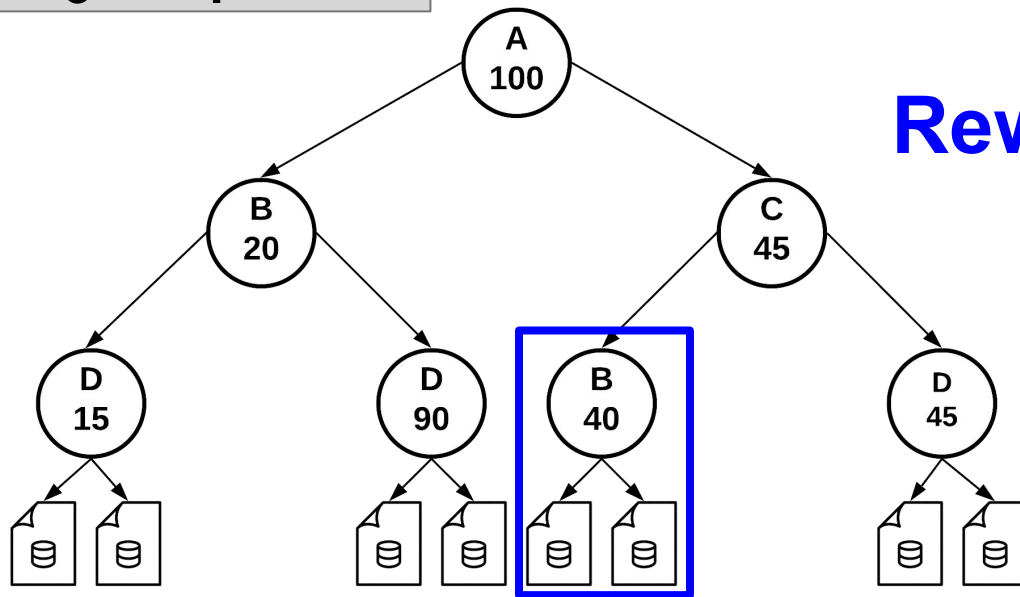
(Shanbhag et al., SoCC 2017)

Swap Operation

$Q_1, Q_2, Q_3, Q_1, \dots$

$$Q_1 = \sigma_{D \leq 45}$$

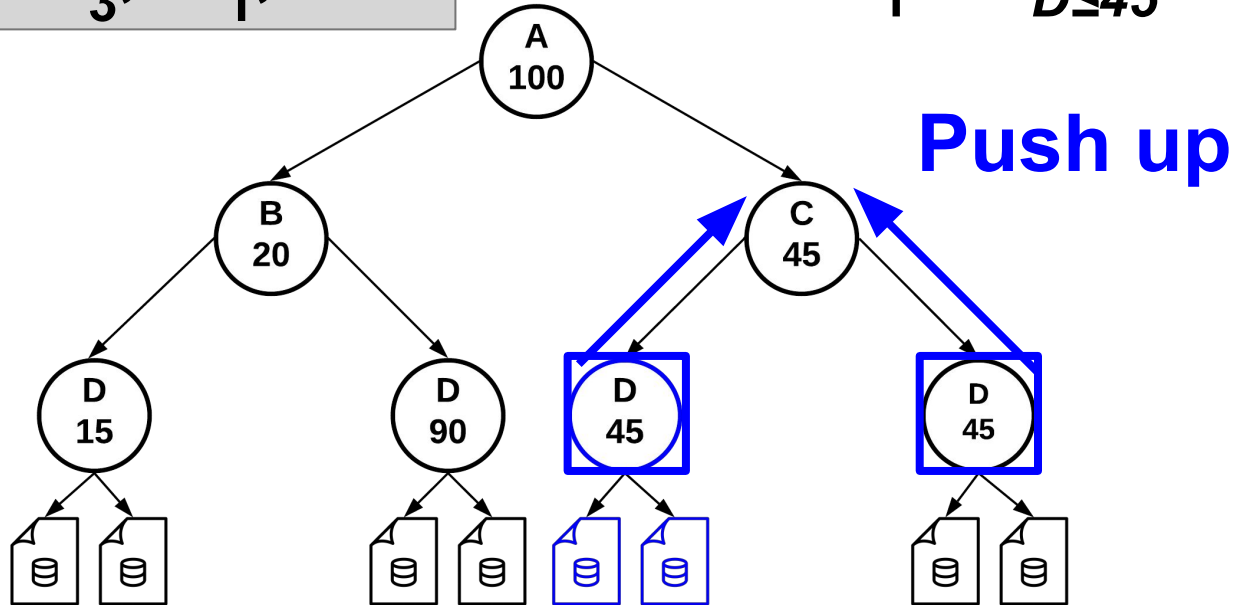
Rewrite Tree



Push Up Operation

$Q_1, Q_2, Q_3, Q_1, \dots$

$$Q_1 = \sigma_{D \leq 45}$$

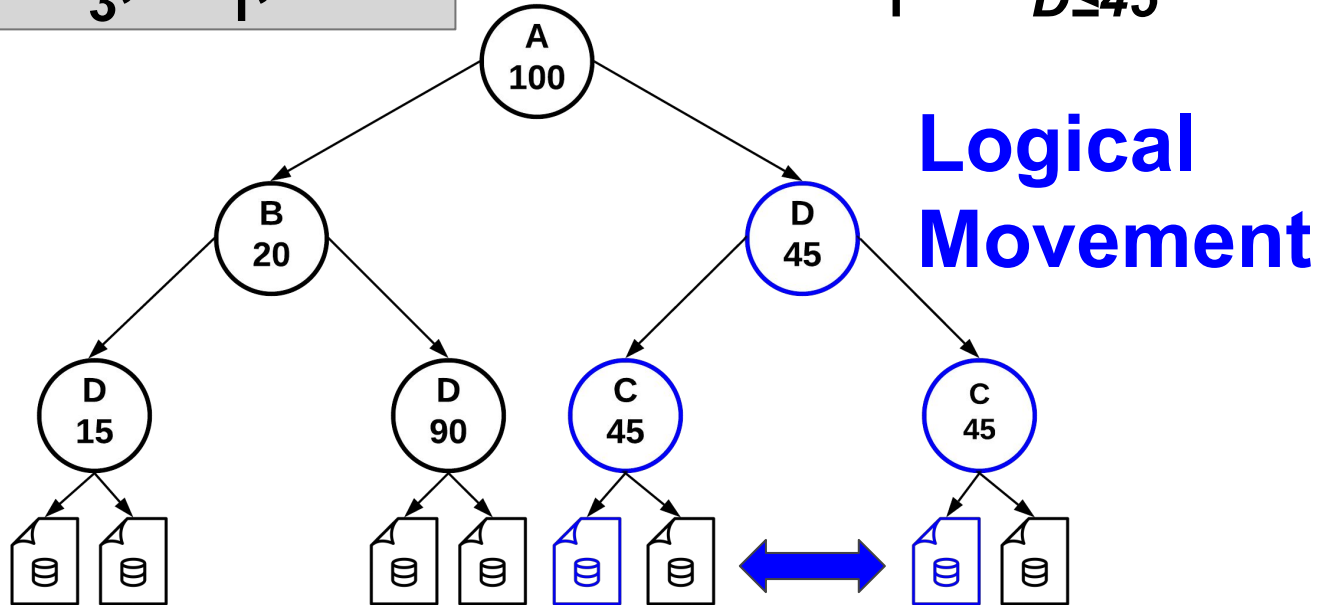


(Shanbhag et al., SoCC 2017)

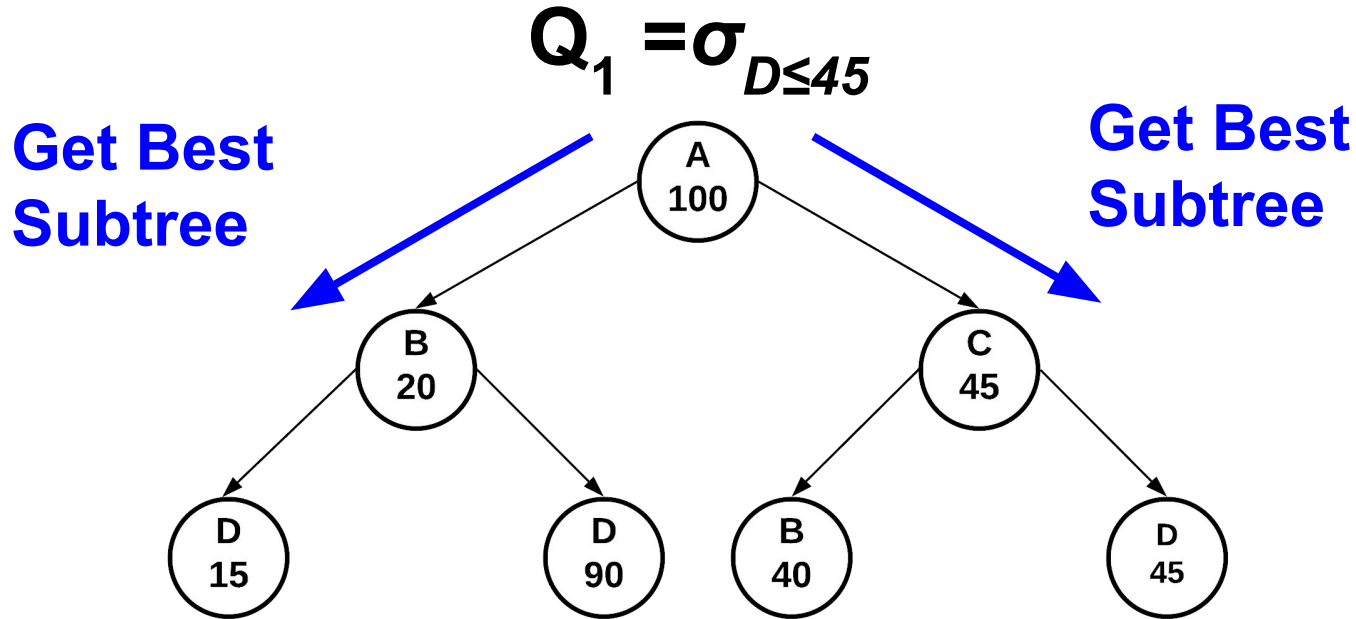
Push Up Operation

$Q_1, Q_2, Q_3, Q_1, \dots$

$$Q_1 = \sigma_{D \leq 45}$$



Divide and Conquer



Partitioning Decisions

- **How to form** partitions?

Upfront then iteratively, based on queries

- **Where to place** partitions?

Rely on HDFS

- **How to execute** multi-partition operations?

Rely on HDFS



Exploiting Workloads

Training & Behaviour Aids
See more

Refine by

Subscription Option
Subscribe & Save

Amazon Prime
Prime

Brand

- OneTigris
- Kong
- Kurgo
- Zichao
- CNATTAGS
- Pawz Road
- cadrim
- AmazonBasics
- Antspark
- Furbo
- Starmark
- Dexas
- SUNREEK
- idepet
- FATCHOI

See more

Avg. Customer Review

- ★★★★★ & Up
- ★★★★☆ & Up
- ★★★☆☆ & Up
- ★★☆☆☆ & Up

New Arrivals

Last 30 days
Last 90 days

Price

Under \$25
\$25 to \$50
\$50 to \$100
\$100 to \$200
\$200 & Above

\$ Min \$ Max Go

Seller

- Amazon.ca
- Northern Shipments
- UrbanInspirations



More options available

Sponsored ⓘ
AmazonBasics Single-Door Folding Metal Dog Crate - Medium (36x23x25 Inches)
by AmazonBasics
CDN\$ 52.90 Prime | FREE One-Day
FREE Delivery by Tomorrow, Nov 22
★★★★★ · 904



More options available

Sponsored ⓘ
Blueberry Pet Classic Dog Collar, Medium Turquoise, Medium, Neck 14.5"-20", Nylon Collars for Dogs by Blueberry Pet
CDN\$ 16.99 Prime
FREE Delivery by Monday, Nov 26
★★★★★ · 99



More options available

Sponsored ⓘ
Tractive 3G Dog GPS Tracker and pet Finder - The GPS Dog Collar Attachment for Dog Tracking
by TRACTIVE
CDN\$ 71.00 CDN\$ 67.00 Prime
FREE Delivery by Tuesday, Nov 27
★★★★☆ · 119



More options available

Amazon's Choice
AmazonBasics Dog Waste Bags with Dispenser and Leash Clip - 900-Count
by AmazonBasics
CDN\$ 18.99 Subscribe & Save
Get scheduled, repeat delivery
CDN\$ 19.99 Prime | FREE One-Day
FREE Delivery by Tomorrow, Nov 22
Get it tomorrow for FREE on qualifying orders over \$25
★★★★★ · 1,840



Dog Car Seat Covers, EVELTEK Universal Fit Waterproof Nonslip and Machine-Washable Pet Hammock for Cars, SUV, Vans & Trucks, With Side Flaps, Pockets and Hammock Front Zipper Design - Black
by EVELTEK
CDN\$ 29.99 Prime
FREE Delivery by Monday, Nov 26
★★★★★ · 434



More options available

LumoLeaf Portable Pet Water Bottle, Reversible & Lightweight Water Dispenser Dogs Cats, Made Food-Grade Silicone (20 Oz) - Green
by LumoLeaf
CDN\$ 16.79 CDN\$ 33.00 Prime | FREE One-Day
FREE Delivery by Tomorrow, Nov 22
FREE One-Day Delivery on qualifying orders over CDN\$ 25
See Details
Save CDN\$ 1.00 with coupon
★★★★☆ · 128



Furbo Dog Camera: Treat Tossing, Full HD Wifi Pet



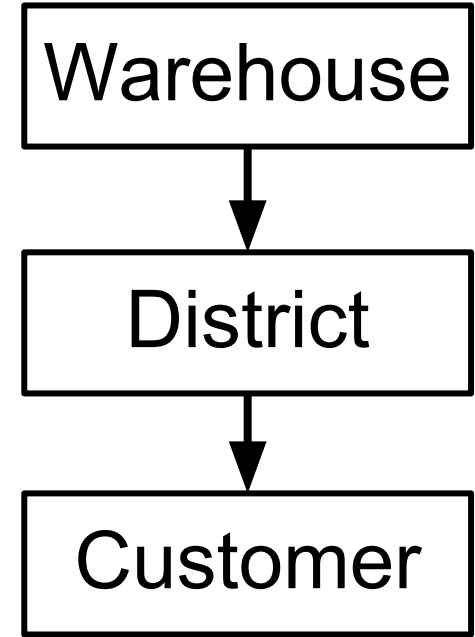
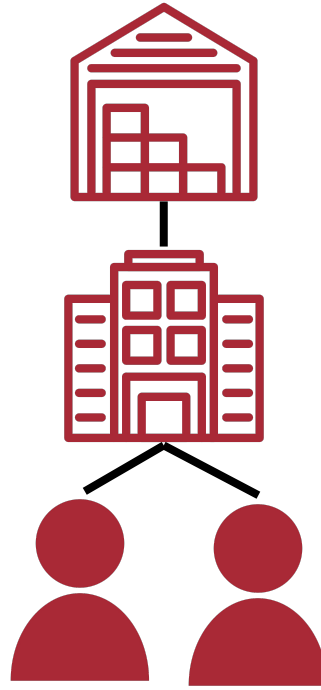
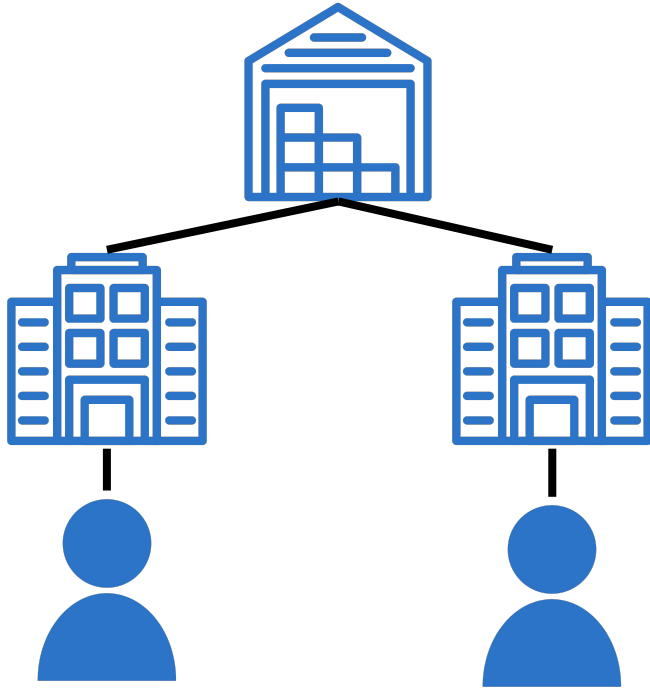
More options available




More options available

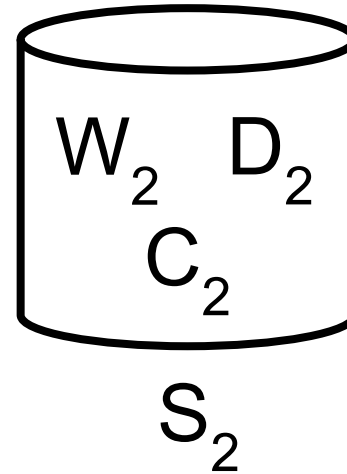
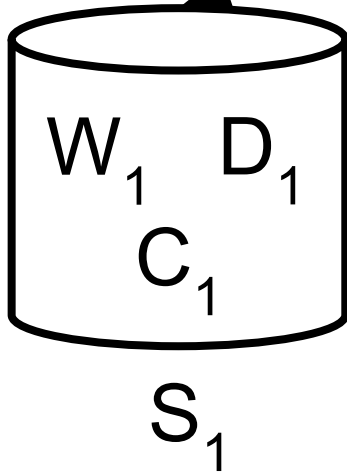
- Known ahead of time
- Parameterized
- Repetitive

Exploiting Workloads - OLTP



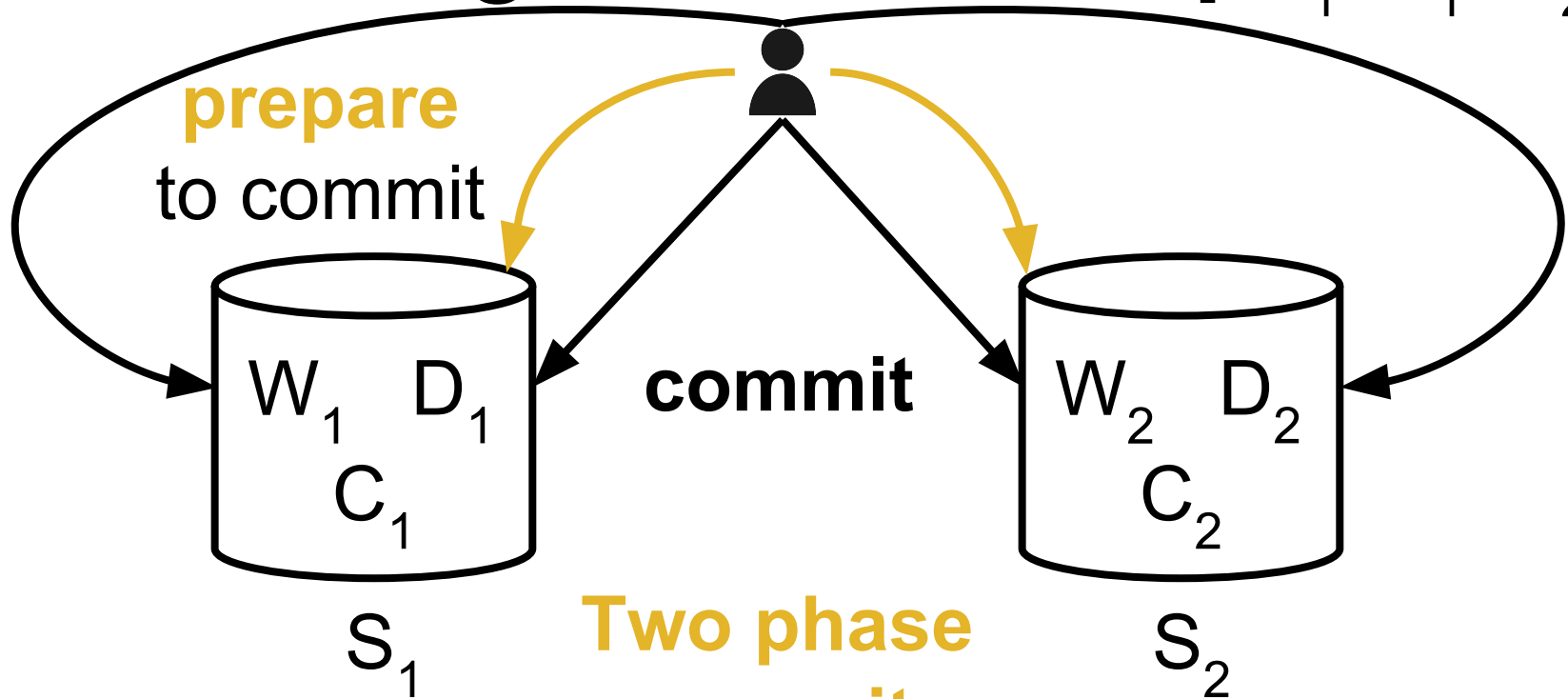
Partitioning OLTP

Write [W_1 , D_1 , C_1] 



Partitioning OLTP

Write [W_1 , D_1 , C_2]



**Two phase
commit**

Partitioning OLTP

Workload based
repartitioning

Per transaction
partitioning

Later in the tutorial

G-Store

(Das et al., SoCC 2010)

L-Store

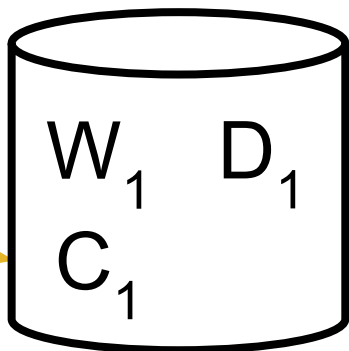
(Lin et al., SIGMOD 2016)

Key Grouping

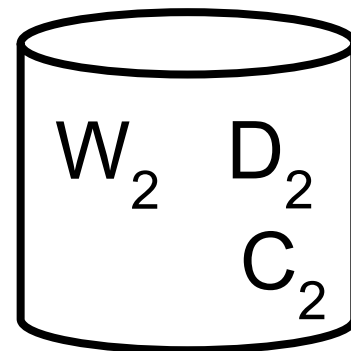
Create
group



Write[W_1 , D_1 , C_2]



S_1



S_2

(Das et al., SoCC 2010)

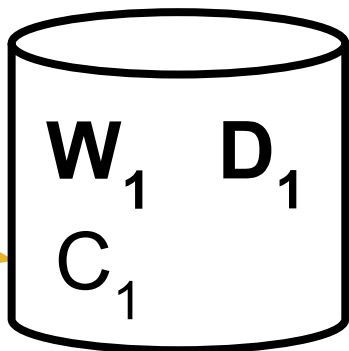
Key Grouping

Create
group

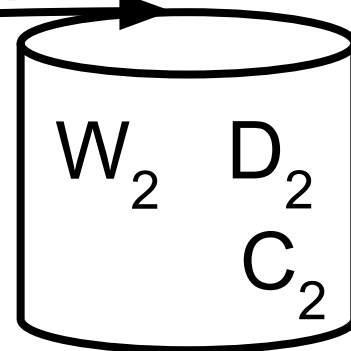


Write[W_1 , D_1 , C_2]

Join request



S_1



S_2

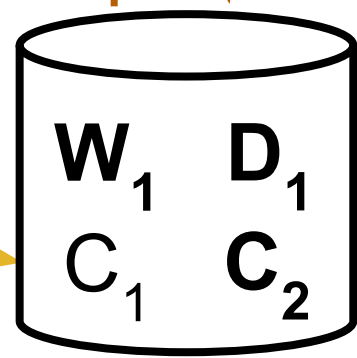
Key Grouping

Create
group

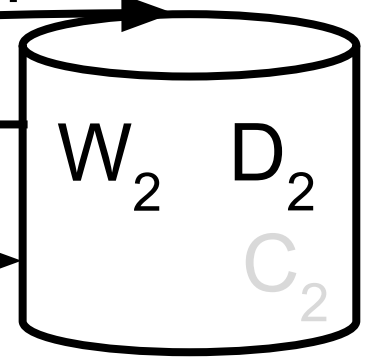
Write[W_1, D_1, C_2]

Txn
ops

Join request



S_1



S_2

Joined

Propagate

Key Grouping

Create group

Txn ops

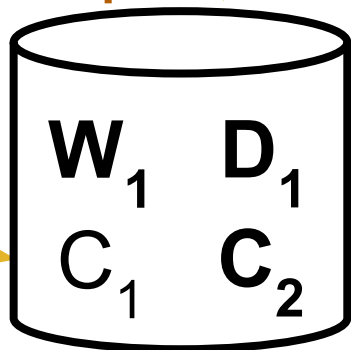
Delete group

Write[W_1, D_1, C_2]

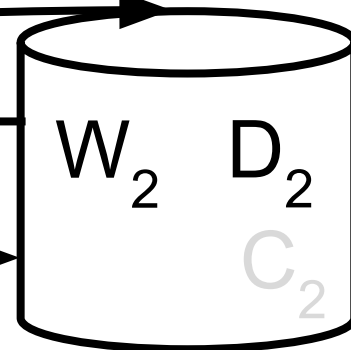
Join request

Joined

Propagate



S_1



S_2

Key Grouping

Create
group

Txn
ops

Delete
group

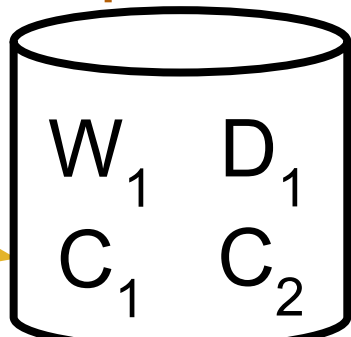
Write[W_1, D_1, C_2]

Join request

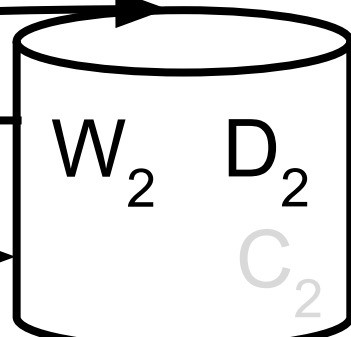
Joined

Propagate

Free



S_1



S_2

Key Grouping

Create
group

Txn
ops

Delete
group

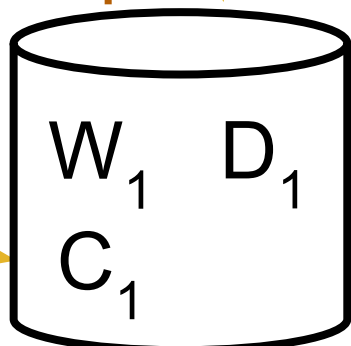
Write[W_1, D_1, C_2]

Join request

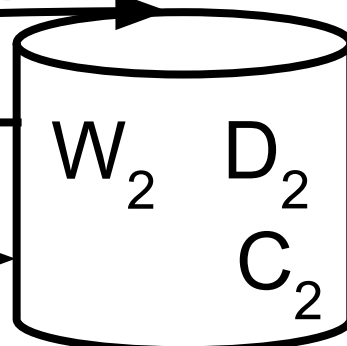
Joined

Propagate

Free



S_1



S_2

Key Grouping

On demand transactional partitioning

Works **best** when groups are small and transactions contain **multiple operations**

But groups are **transient**

(Das et al., SoCC 2010)

Localizing Execution

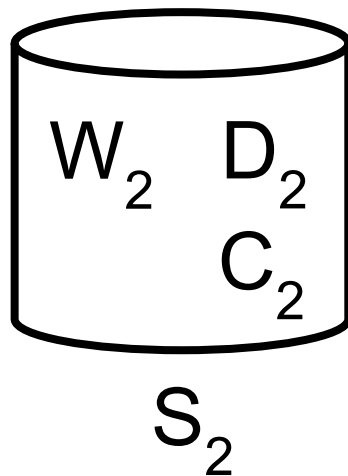
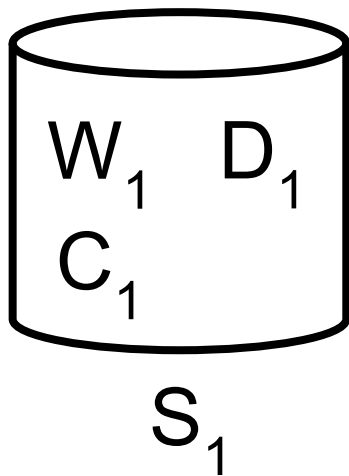
Repartition data via **localization** for **single site** execution

Dynamic partitioning based on **transaction patterns**

(Lin et al., SIGMOD 2016)

Localizing Execution

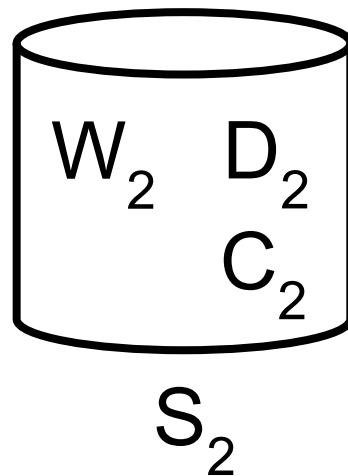
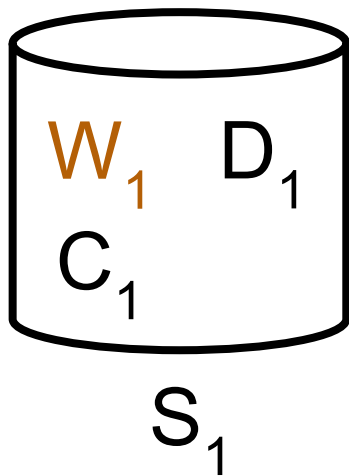
Ownership
information



W_1	S_1
W_2	S_2
D_1	S_1
D_2	S_2
C_1	S_1
C_2	S_2

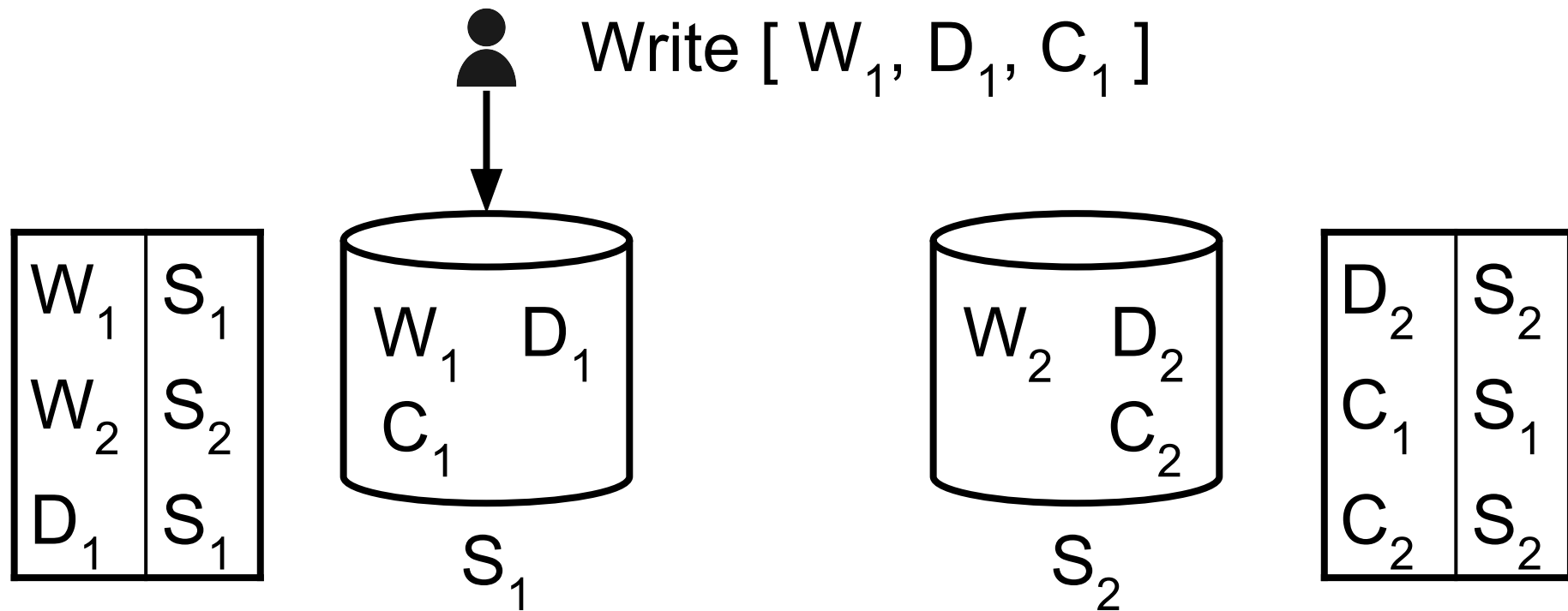
Localizing Execution

Ownership
information



W_1	S_1
W_2	S_2
D_1	S_1
D_2	S_2
C_1	S_1
C_2	S_2

Localizing Execution



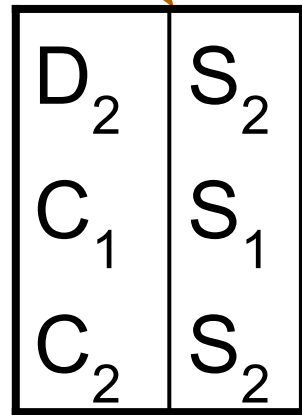
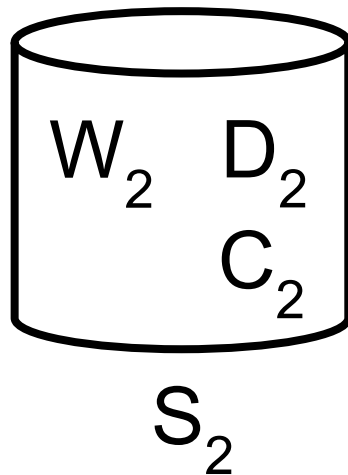
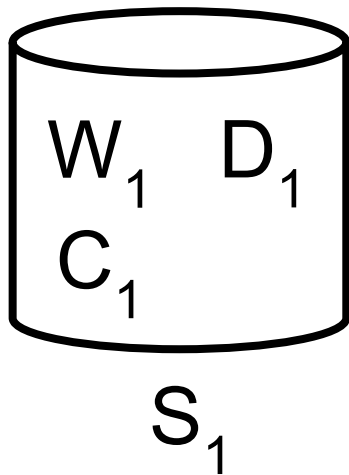
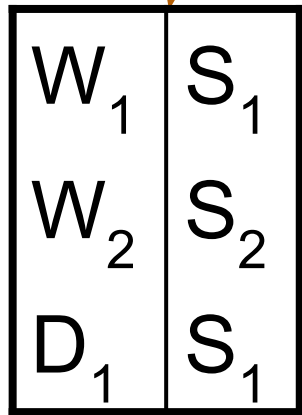
(Lin et al., SIGMOD 2016)

Localizing Execution

Owner
request



Write [W_1 , D_1 , C_2]



(Lin et al., SIGMOD 2016)



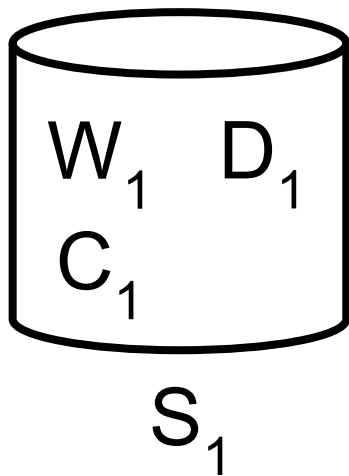
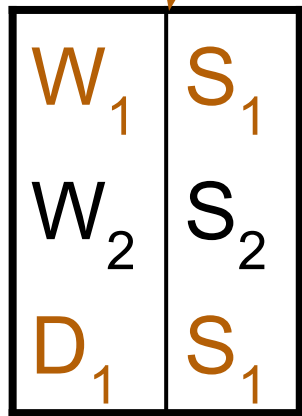
UNIVERSITY OF
WATERLOO

Localizing Execution

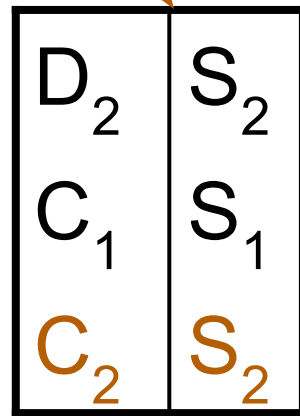
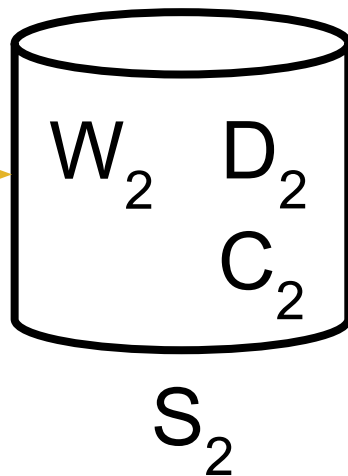
Owner
request



Write [W_1 , D_1 , C_2]



Transfer



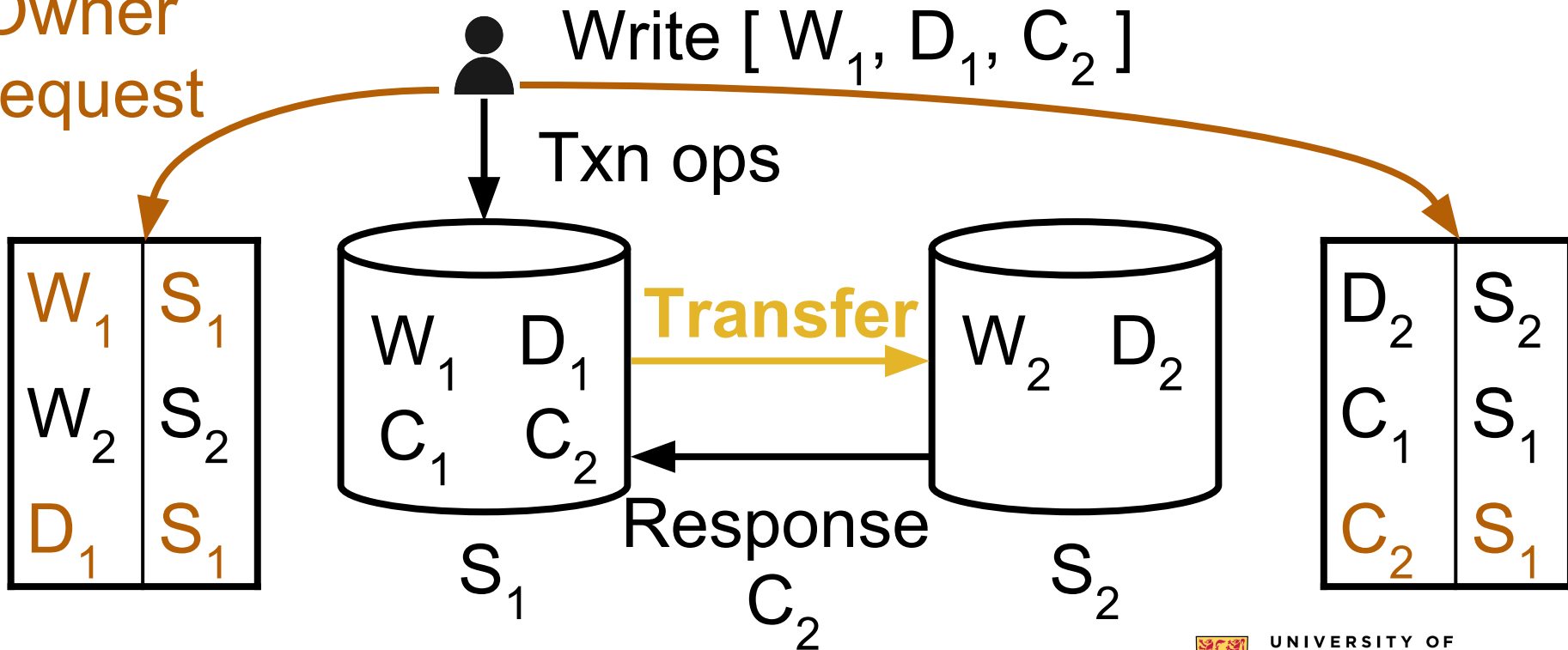
(Lin et al., SIGMOD 2016)



UNIVERSITY OF
WATERLOO

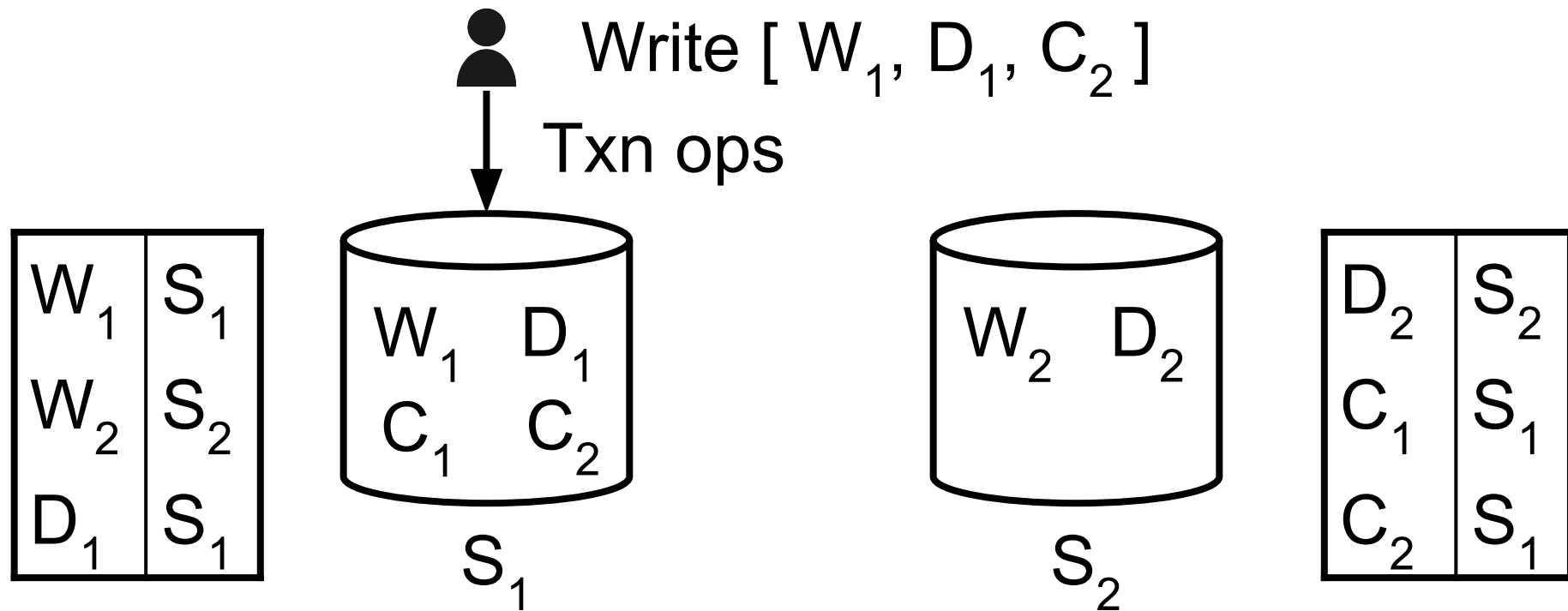
Localizing Execution

Owner
request



(Lin et al., SIGMOD 2016)

Localizing Execution



(Lin et al., SIGMOD 2016)

Localizing Execution

Dynamic partitioning based on **per transaction patterns**

Does **not** consider **workload overall**

Partitioning Decisions

- **How to form** partitions?

Transaction localization

- **Where to place** partitions?

At requester

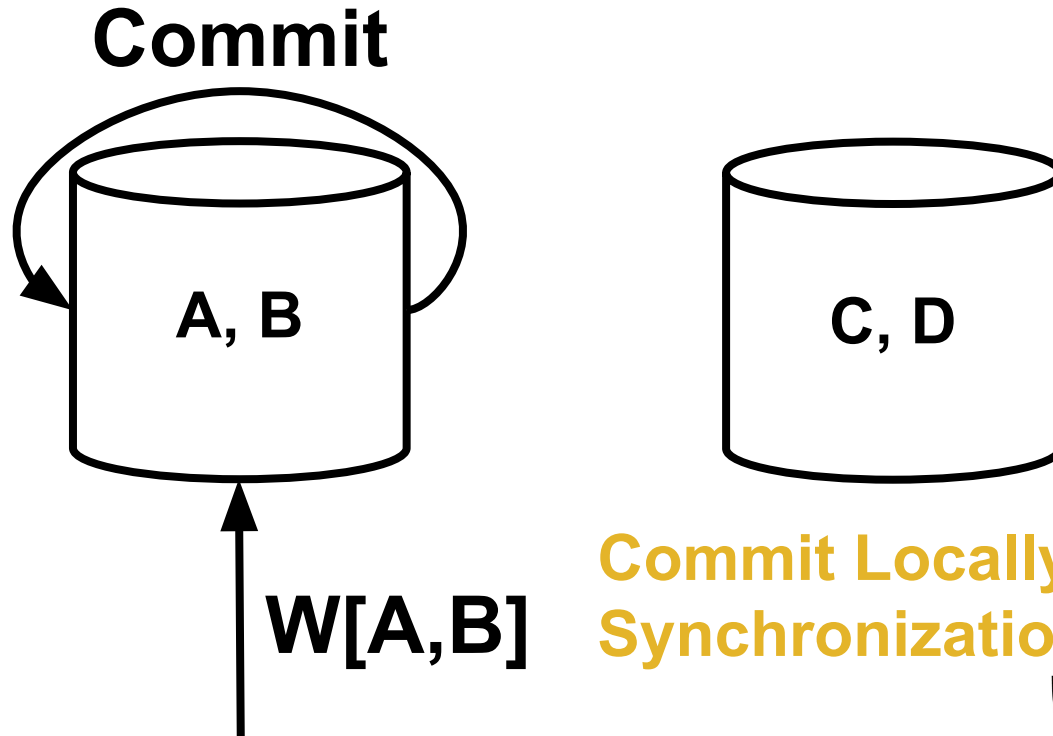
- **How to execute** multi-partition operations?

L-Store protocol

Partitioning Decisions

- **How to form** partitions?
Key groups, temporarily
- **Where to place** partitions?
Key group leader
- **How to execute** multi-partition operations?
Key group protocol

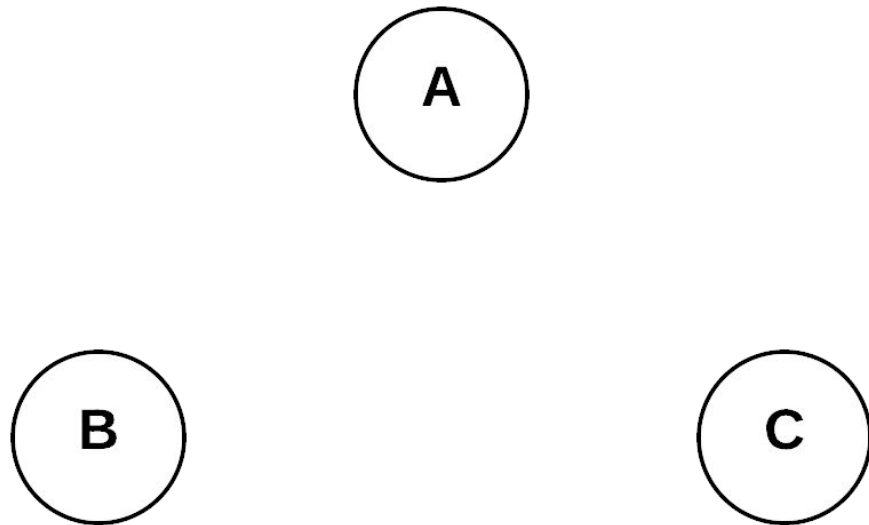
Localizing Transactions



Constructing the Graph

From a workload trace

ID	Name
A	Alice
B	Bob
C	Carol

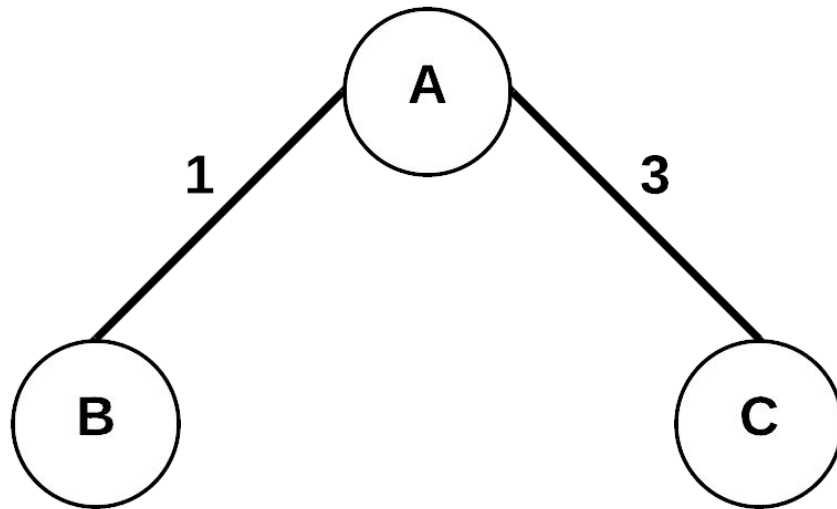


(Curino et al., VLDB 2010)

Constructing the Graph

Add traced transactions: $R[A,B]$, $3x W[A,C]$

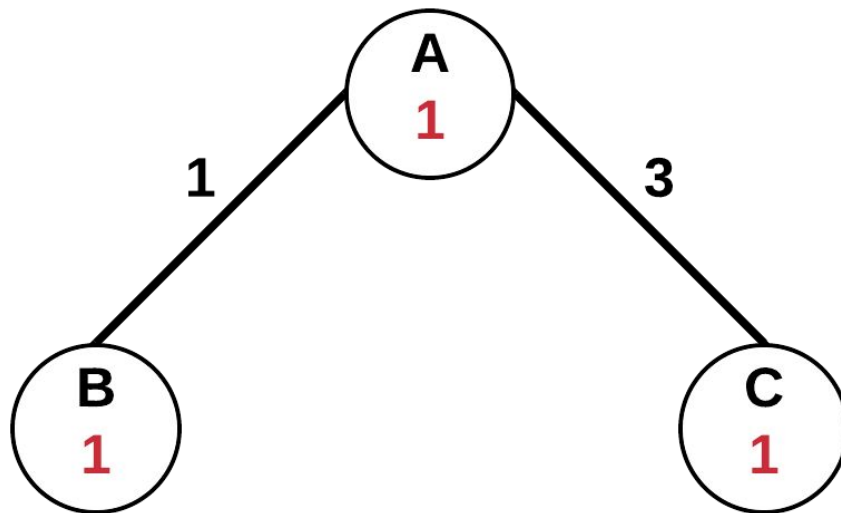
ID	Name
A	Alice
B	Bob
C	Carol



Constructing the Graph

Add node weights (size, load)

ID	Name
A	Alice
B	Bob
C	Carol

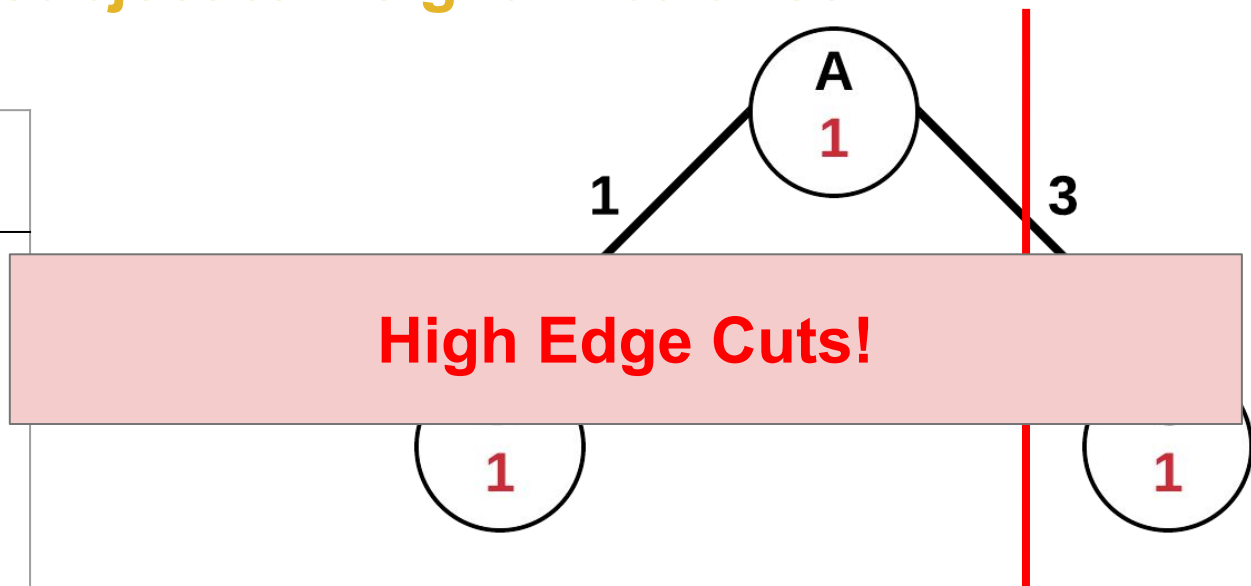


(Curino et al., VLDB 2010)

Constructing the Graph

Min-cut edges subject to weight imbalance

ID	Name
A	Alice
B	Bob
C	Carol

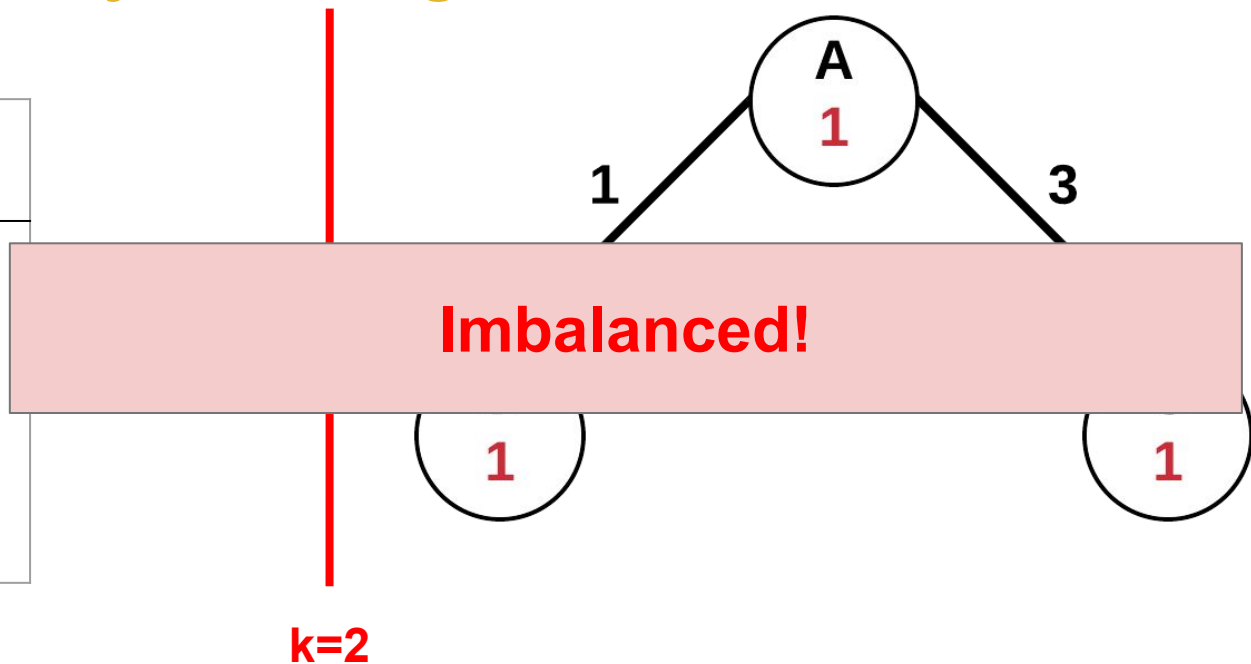


k=2

Constructing the Graph

Min-cut edges subject to weight imbalance

ID	Name
A	Alice
B	Bob
C	Carol

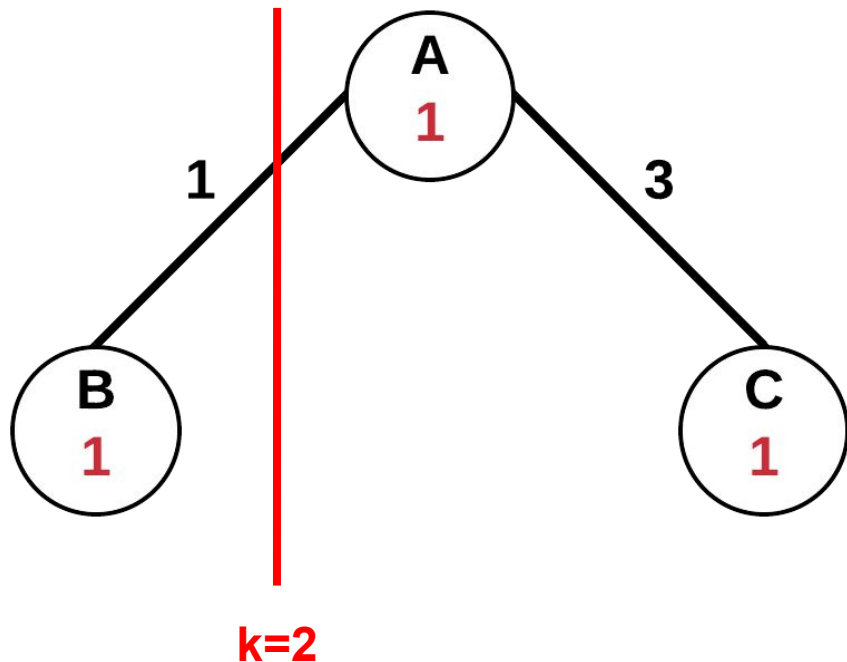


(Curino et al., VLDB 2010)

Constructing the Graph

Min-cut edges subject to weight imbalance

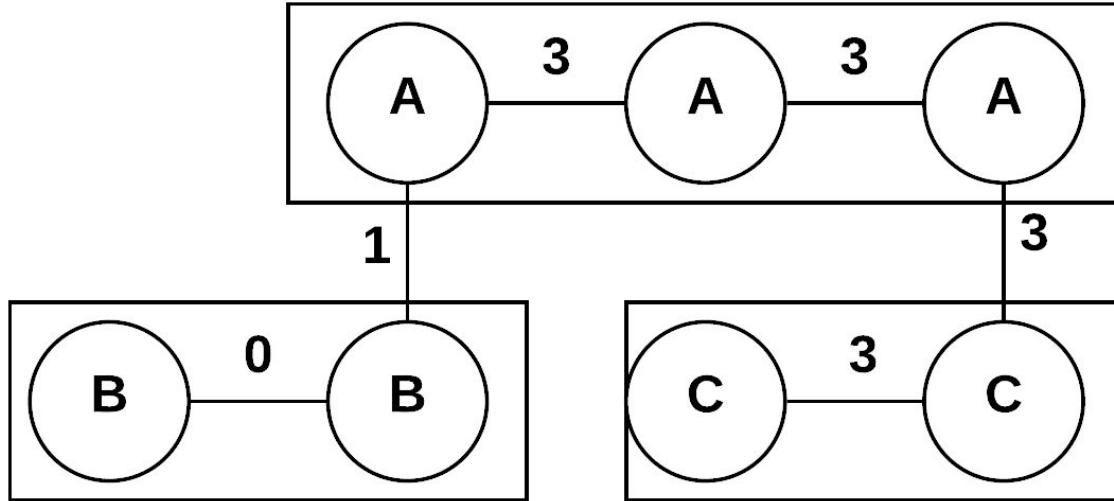
ID	Name
A	Alice
B	Bob
C	Carol



(Curino et al., VLDB 2010)

Adding Replica Support

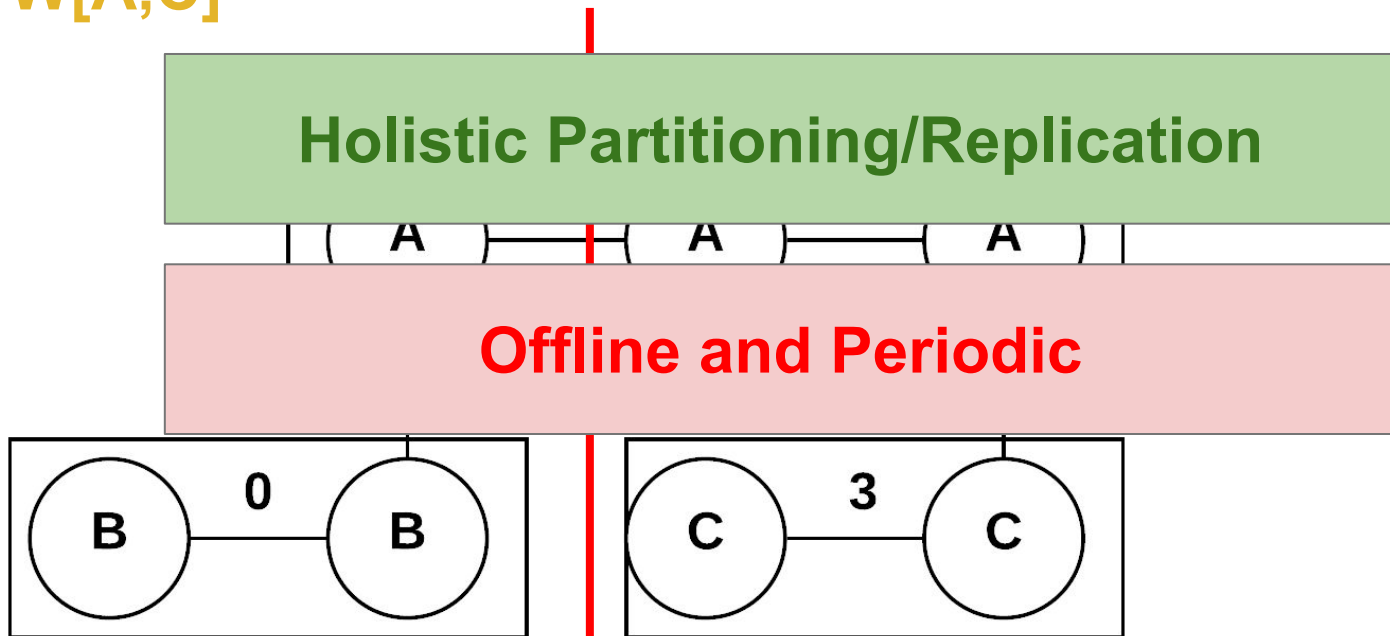
R[A,B] 3x W[A,C]



(Curino et al., VLDB 2010)

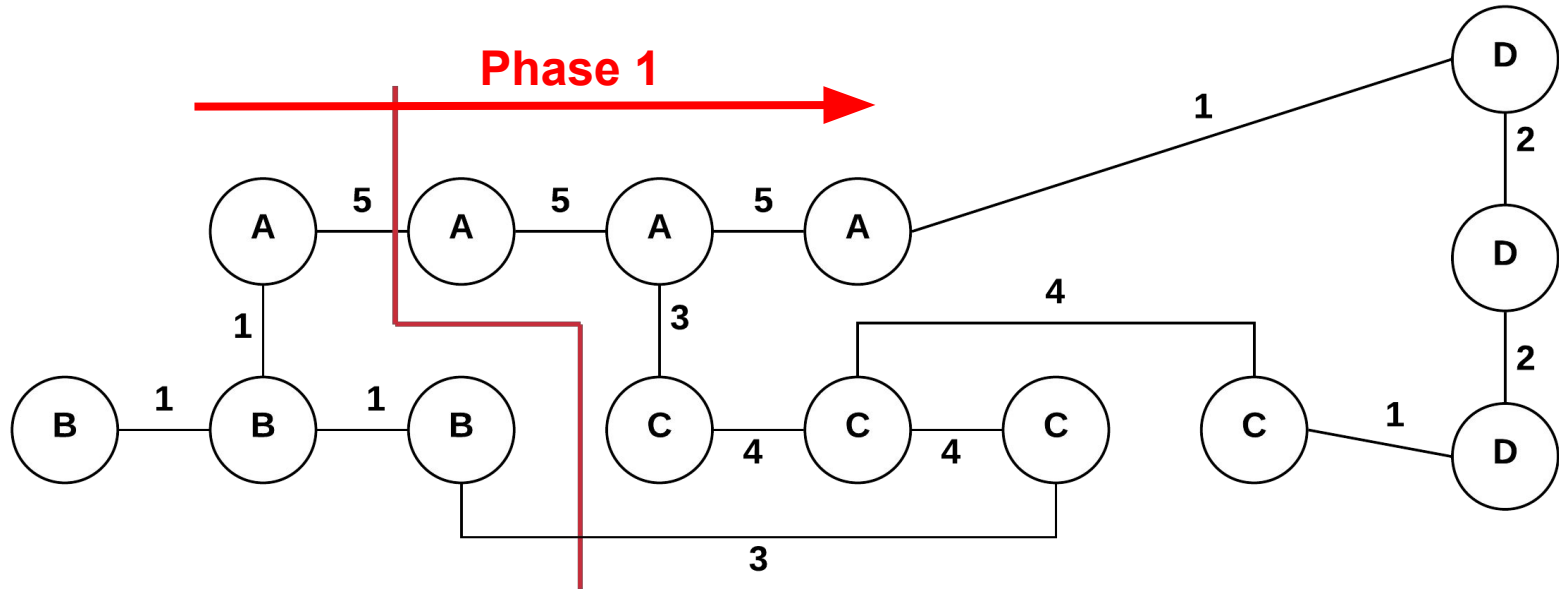
Adding Replica Support

R[A,B] 3x W[A,C]



(Curino et al., VLDB 2010)

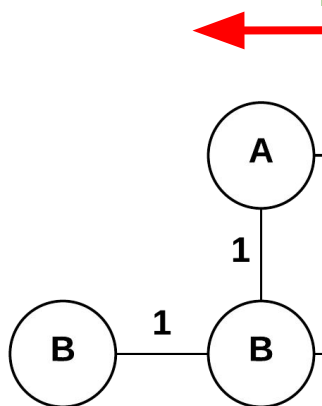
Two Phases



(Nicoara et al., EDBT 2015)

Two Phases

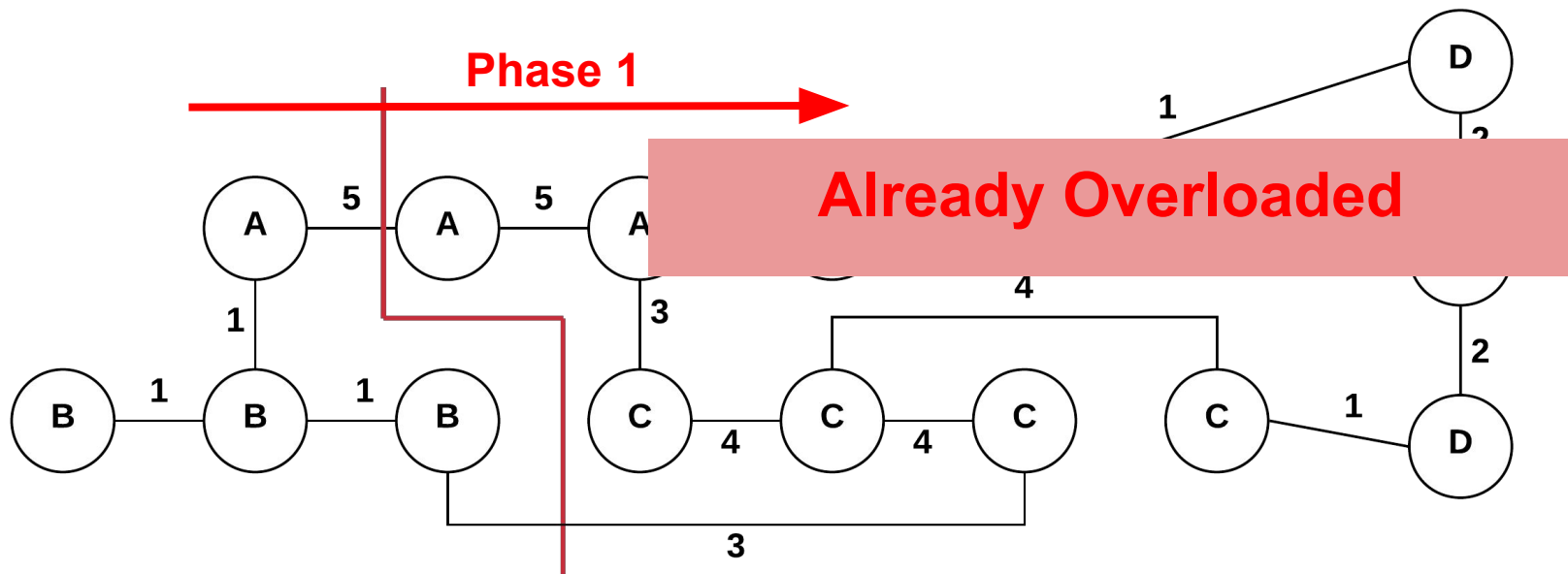
Logical Movement, then Migrate



Rule:

- Movement doesn't overload
- Move best-gain candidates
- If overloaded, must move!

Two Phases

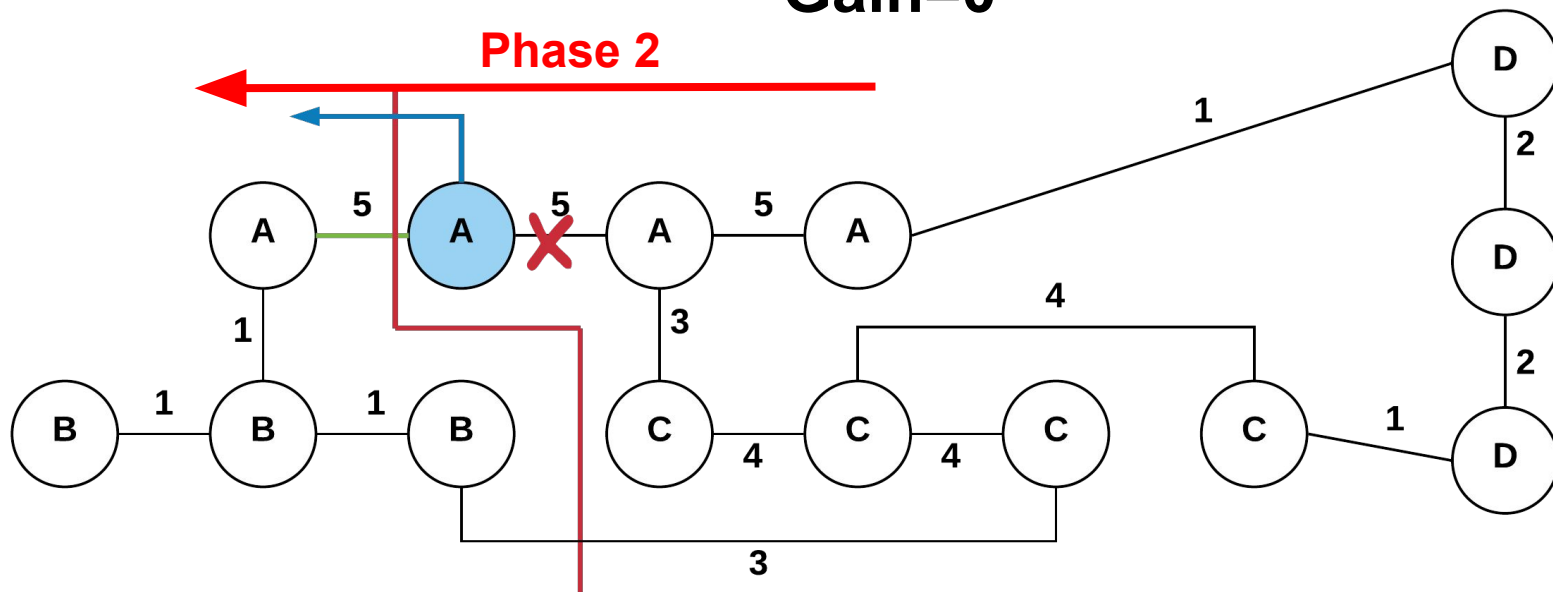


$W(P1)=4$, $W(P2)=10$, $EC=8$, Bounds: (6,8)

(Nicoara et al., EDBT 2015)

Two Phases

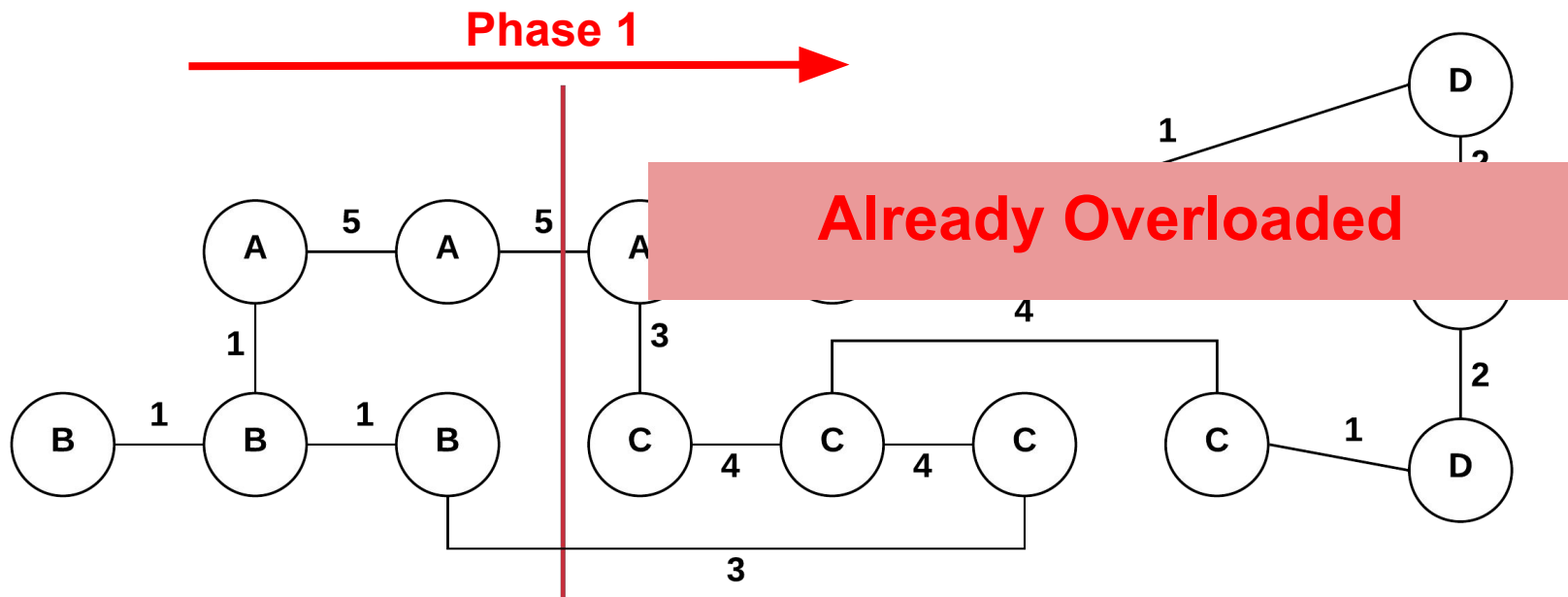
Gain=0



$W(P1)=4$, $W(P2)=10$, $EC=8$, Bounds: (6,8)

(Nicoara et al., EDBT 2015)

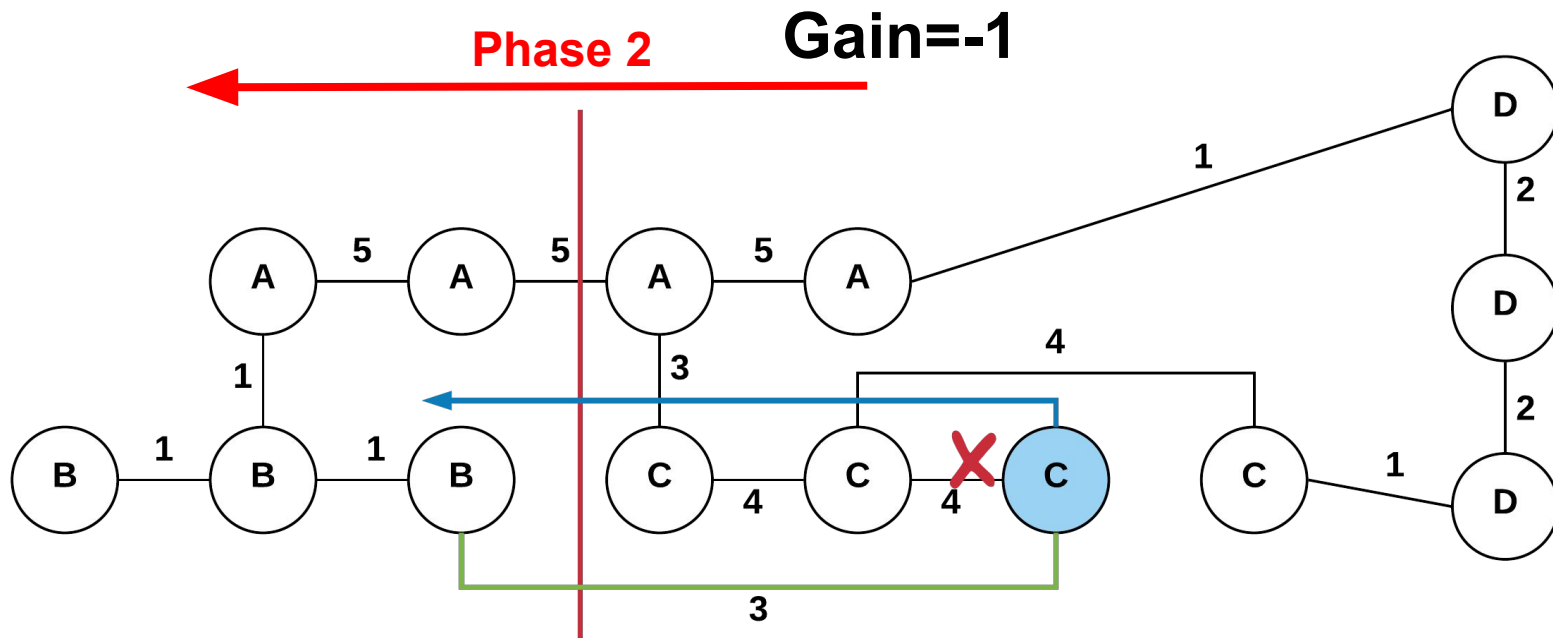
Two Phases



$W(P1)=5$, $W(P2)=9$, $EC=8$, Bounds: (6,8)

(Nicoara et al., EDBT 2015)

Two Phases



$W(P1)=5$, $W(P2)=9$, $EC=8$, Bounds: (6,8)

(Nicoara et al., EDBT 2015)

Partitioning Decisions

- **How to form** partitions?

Graph partitioning

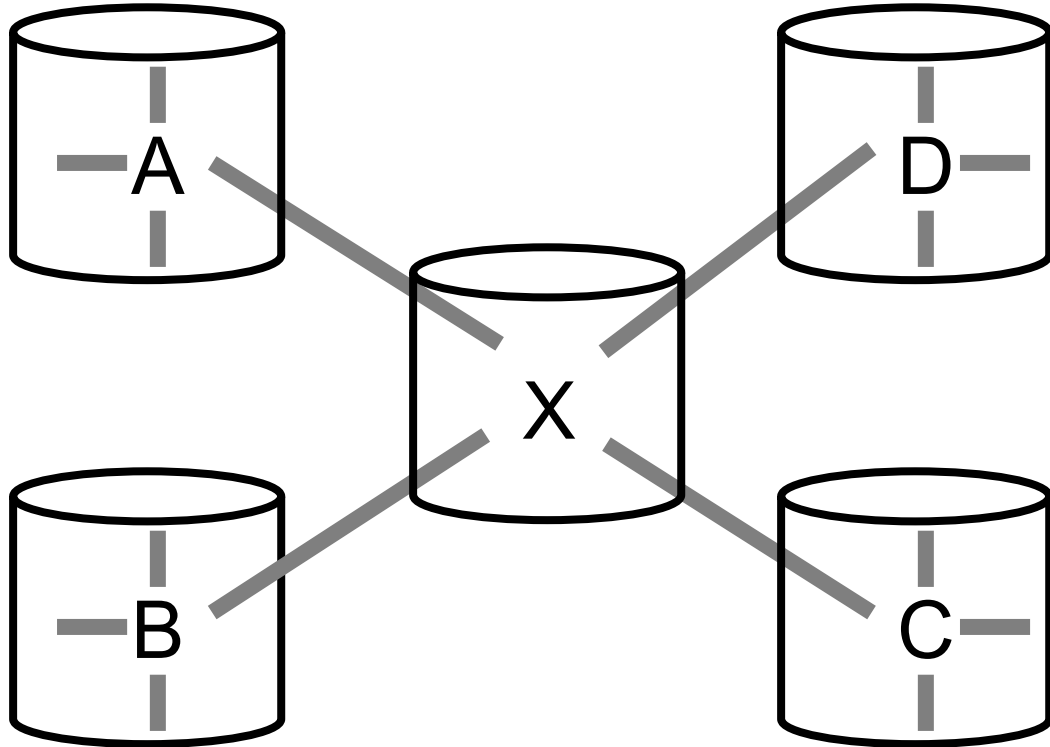
- **Where to place** partitions?

Based on partitioning

- **How to execute** multi-partition operations?

2PC

Graph Partitioning



Minimizes **total**
number of distributed
transactions

Ignores **per node**
involvement

Adaptive Database Partitioning

Balance load and minimize distributed transactions

Database elasticity

E-Store

(Taft et al., VLDB 2014)

Clay

(Serafini et al., VLDB 2015)

P-Store

(Taft et al., SIGMOD 2018)



Considering Distributed Cost

General Graph Partitioning: minimize # edge cuts

such that: $\text{load}(S_i) < (1 + \epsilon) \text{avg load}(S)$
load balanced

General: $\text{load}(S_i) = \sum w(v)$ (v at S_i)

Clay: $\text{load}(S_i) = \sum w(v) + k \sum w(uv)$ (v at S_i)
(u not at S_i)

Distributed cost

(Serafini et al., VLDB 2016)



Repartitioning Cost

General Graph Partitioning: minimize # edge cuts

Clay: minimize # edge cuts

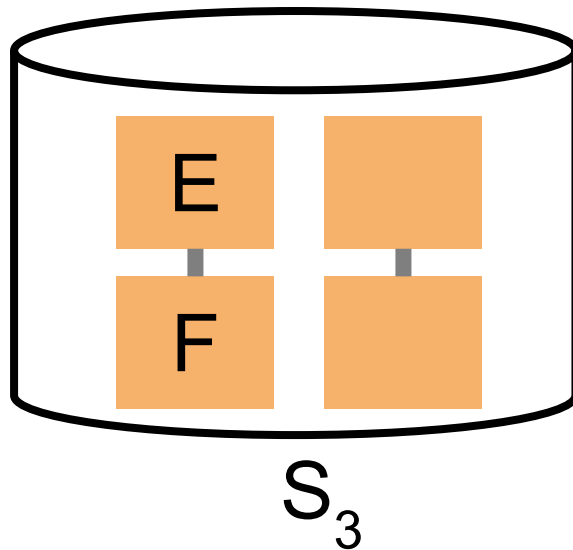
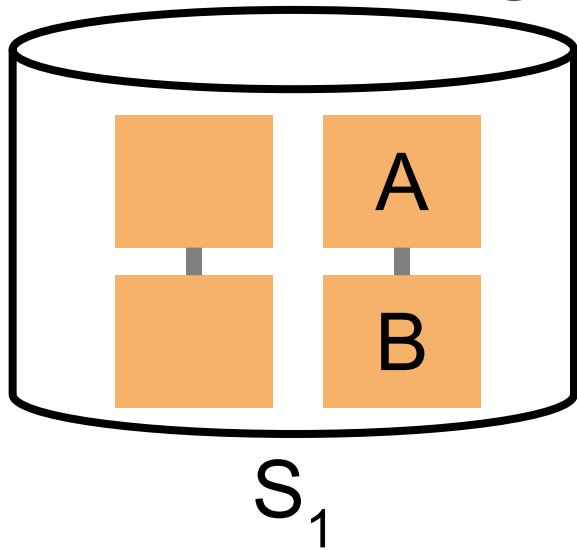
and

of vertices mapped to new partitions

cost of repartitioning

(Serafini et al., VLDB 2016)

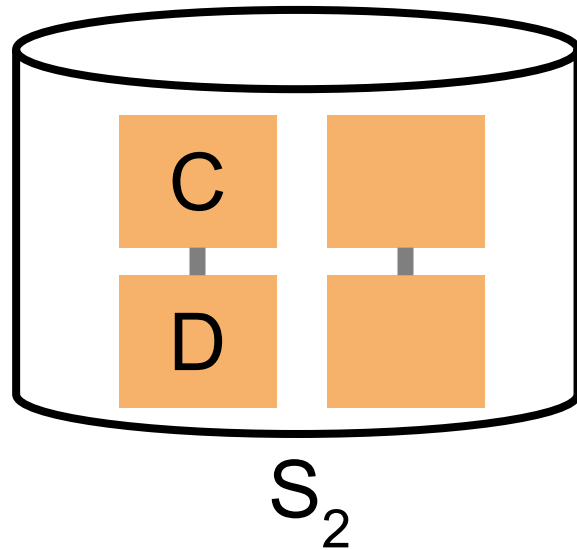
Clumping



Low

Med

High



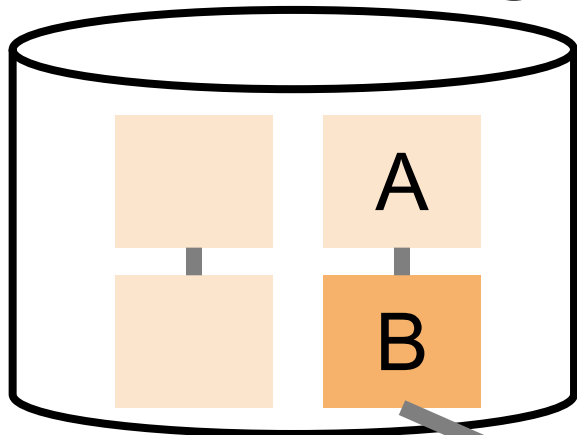
(Serafini et al., VLDB 2016)

Clumping

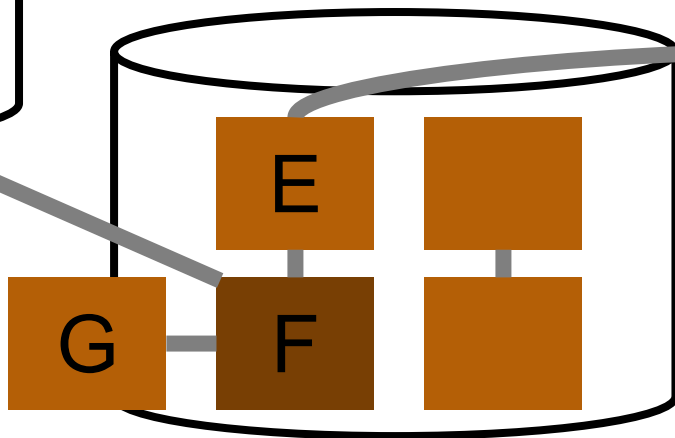
Low

Med

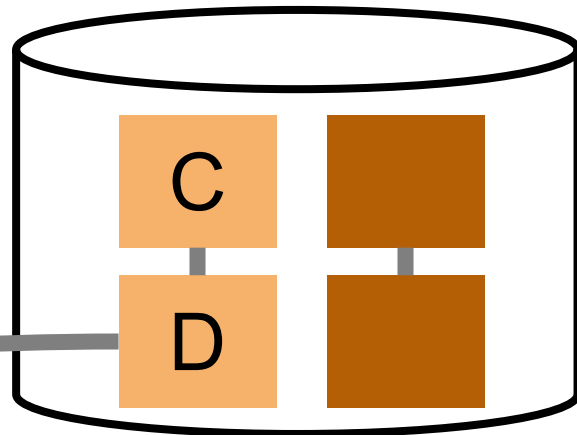
High



S_1



S_3

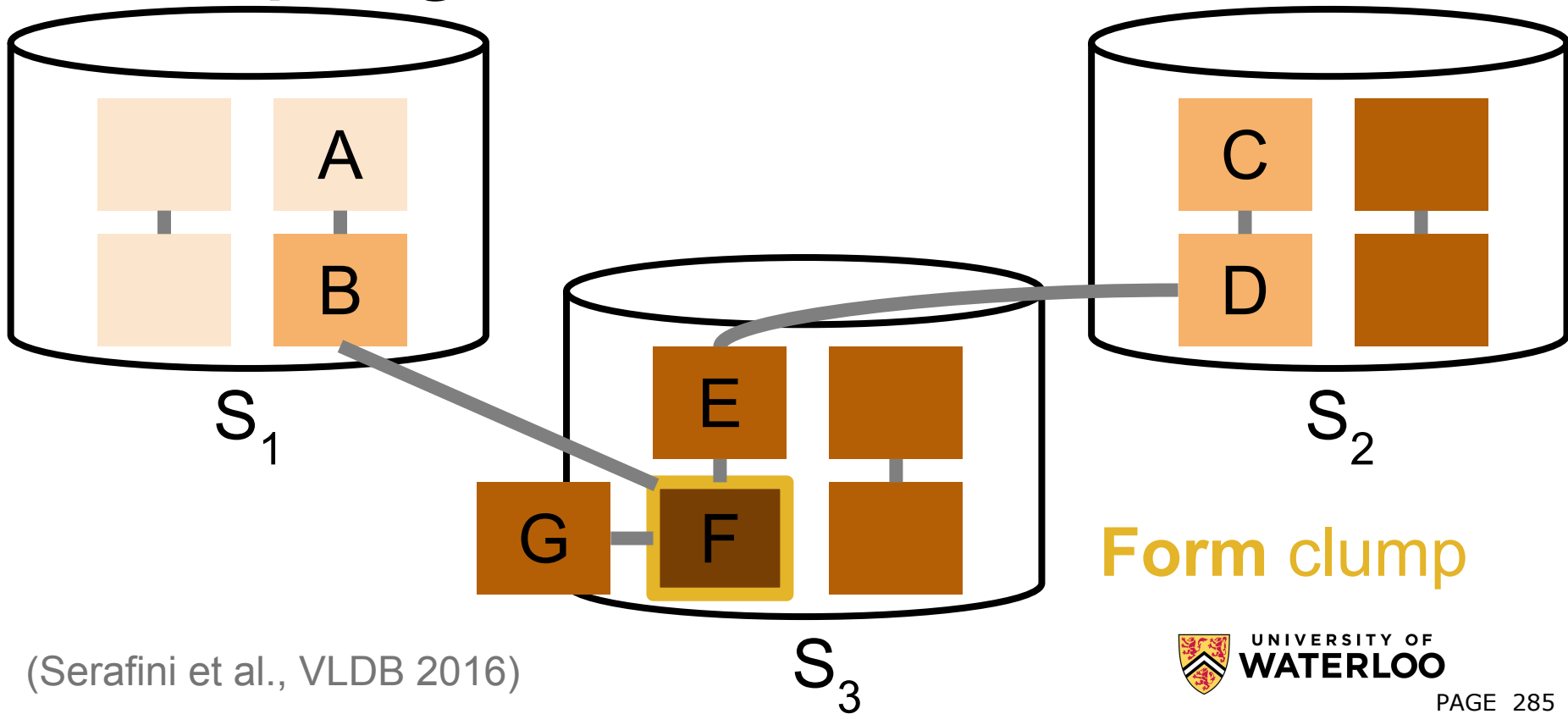


S_2

(Serafini et al., VLDB 2016)

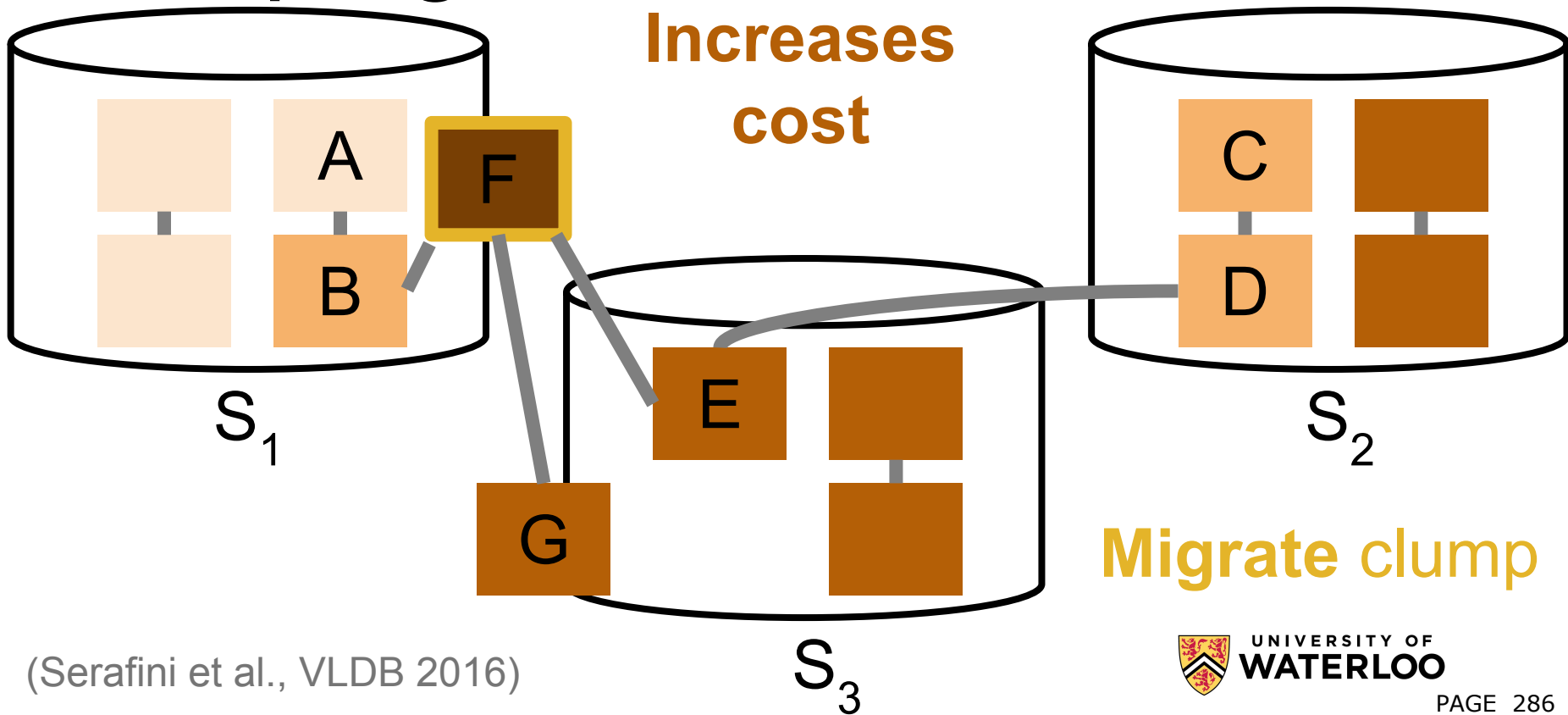


Clumping



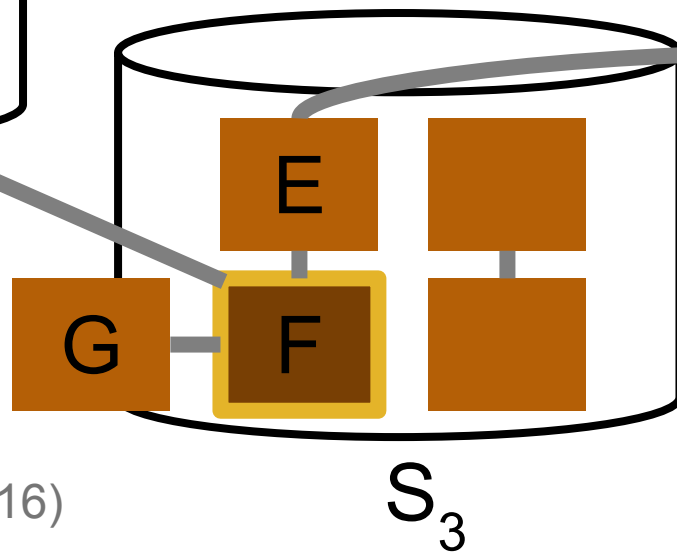
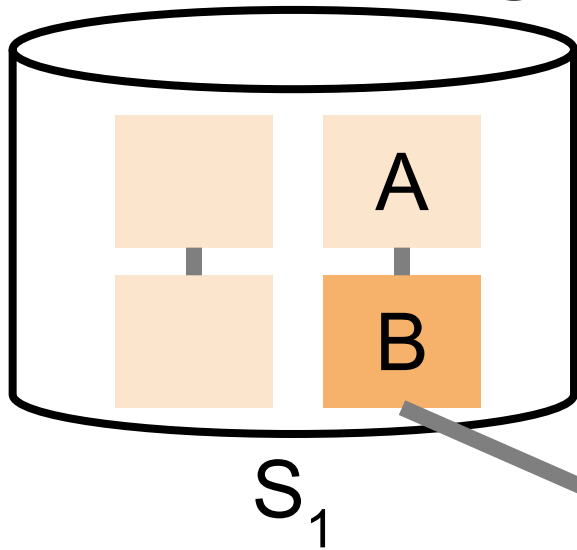
(Serafini et al., VLDB 2016)

Clumping

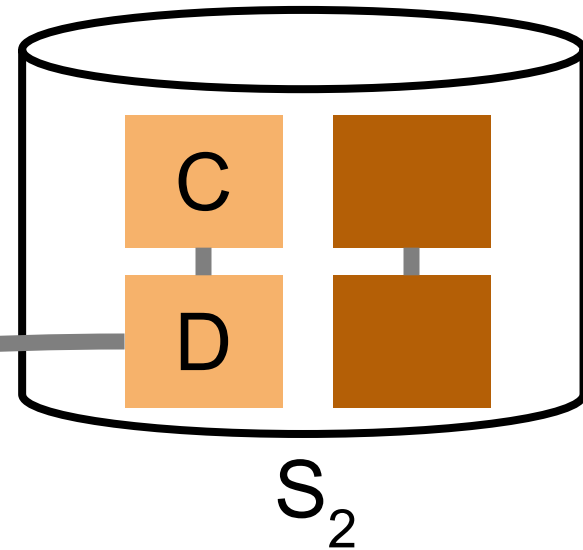


(Serafini et al., VLDB 2016)

Clumping

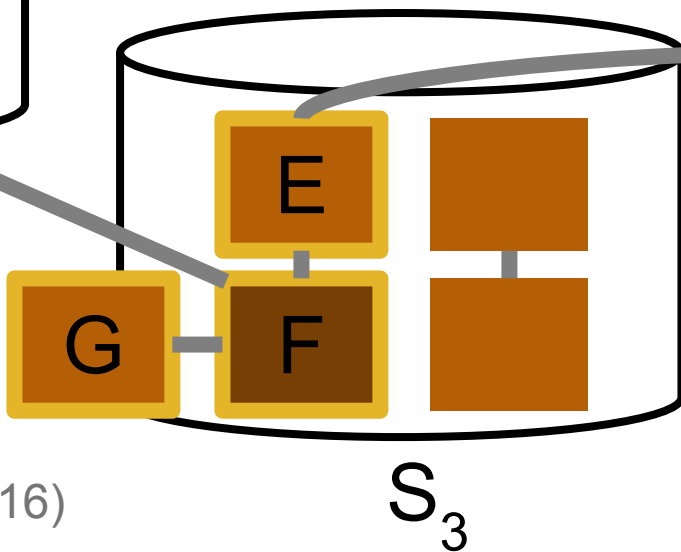
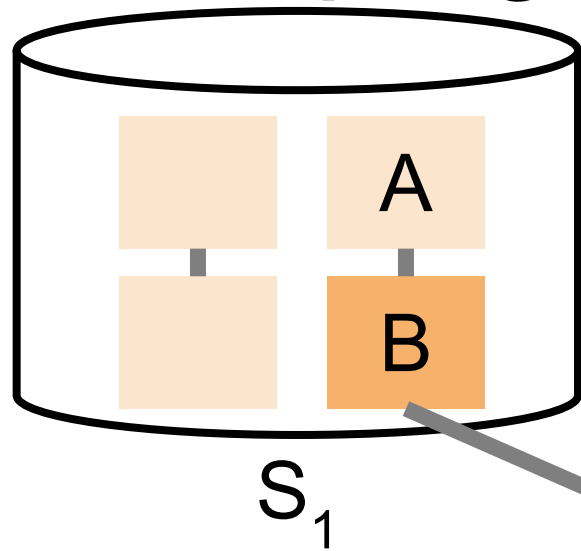


Low Med High

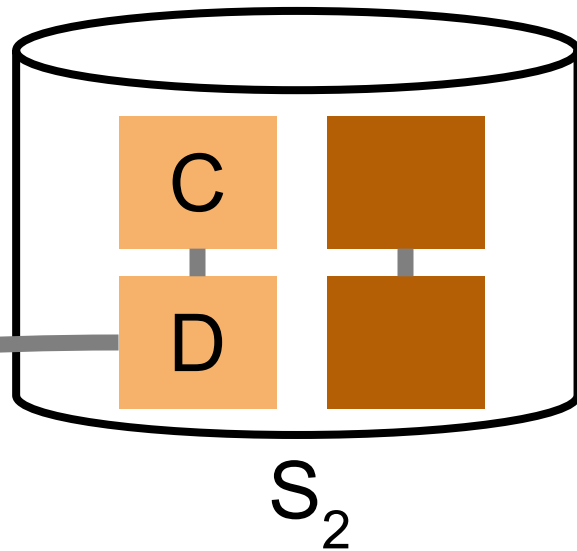


(Serafini et al., VLDB 2016)

Clumping

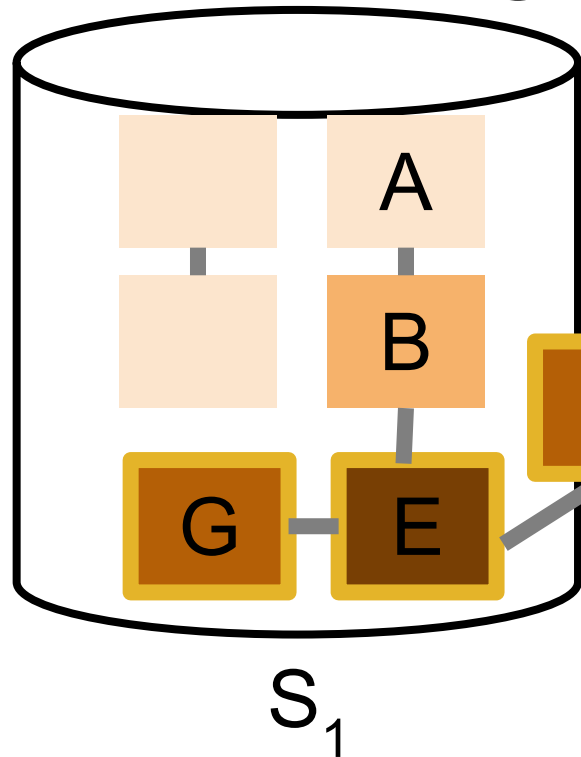


Low Med High



Expand clump

Clumping

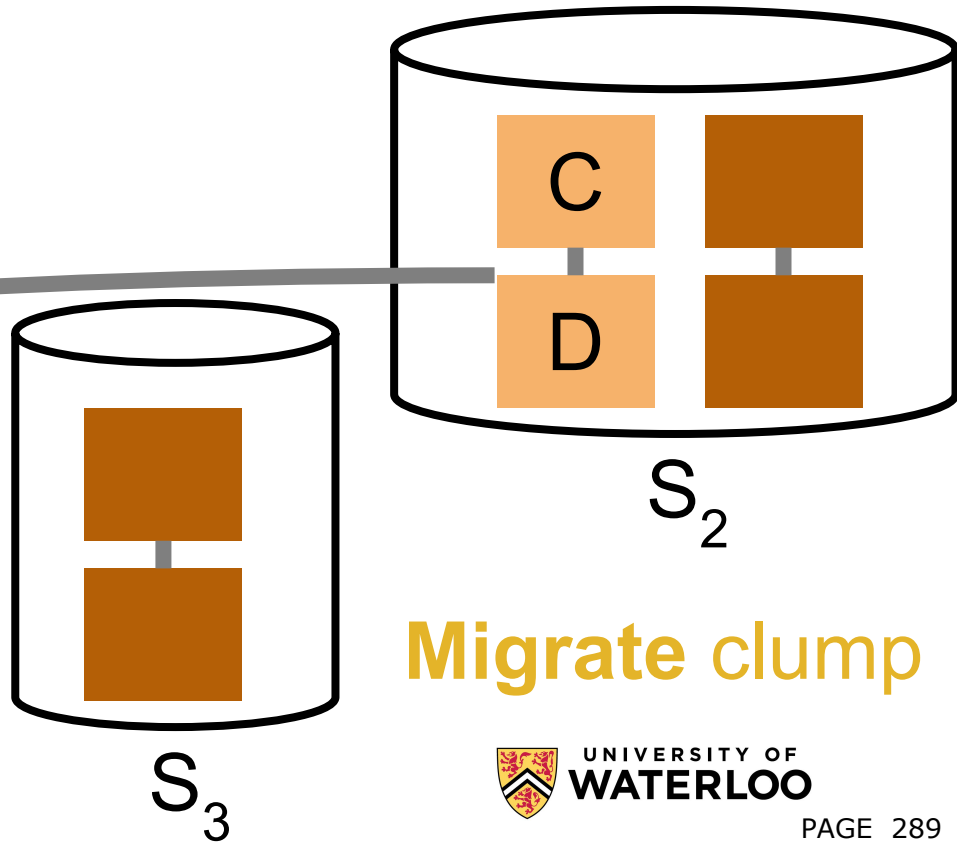


(Serafini et al., VLDB 2016)

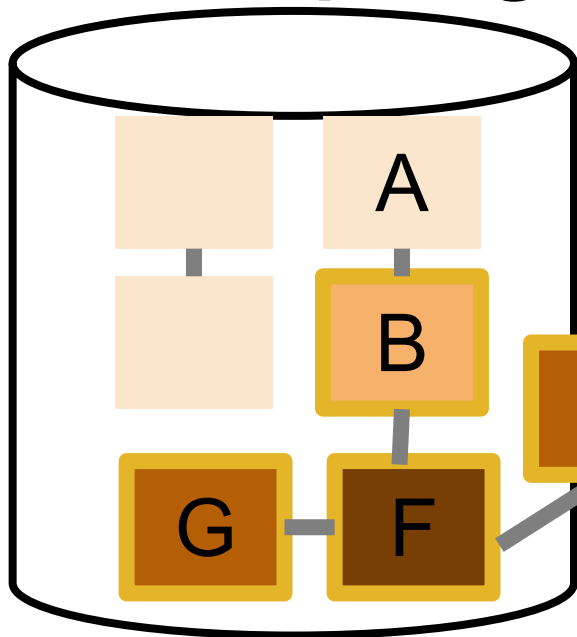
Low

Med

High



Clumping



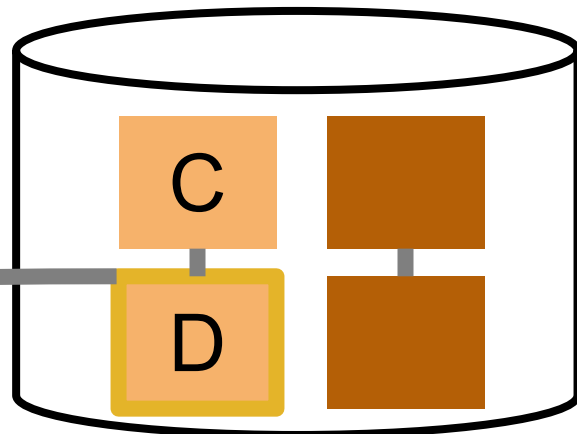
S_1

(Serafini et al., VLDB 2016)

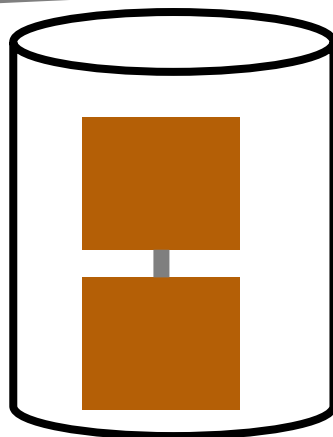
Low

Med

High



S_2



S_3

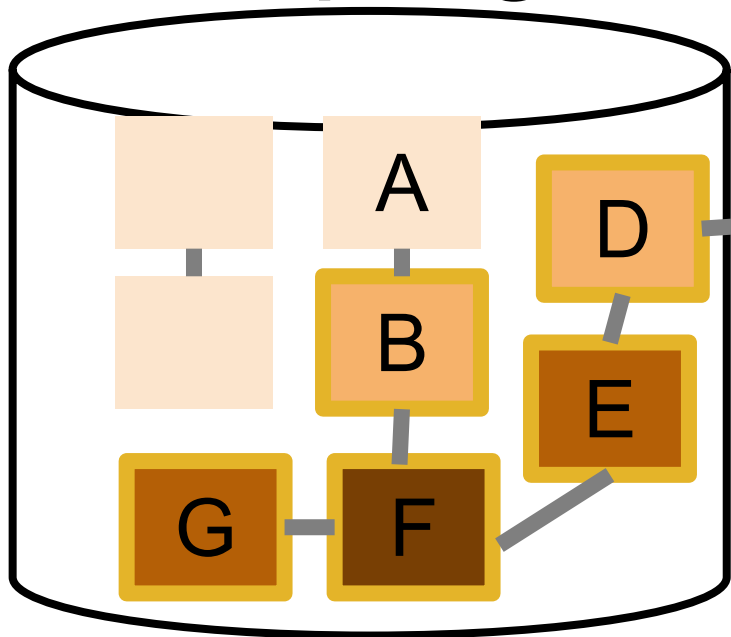
Expand clump

Clumping

Low

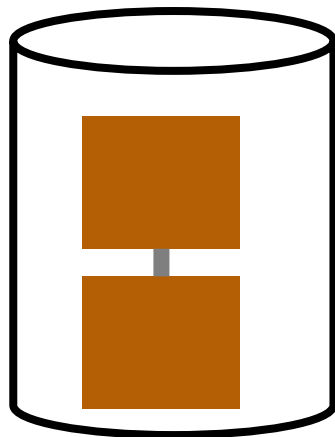
Med

High

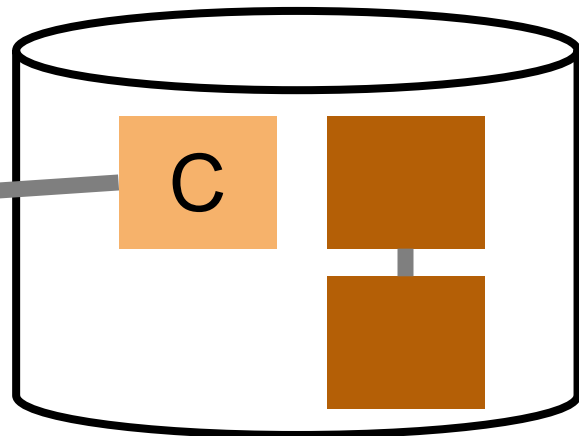


S_1

(Serafini et al., VLDB 2016)



S_3

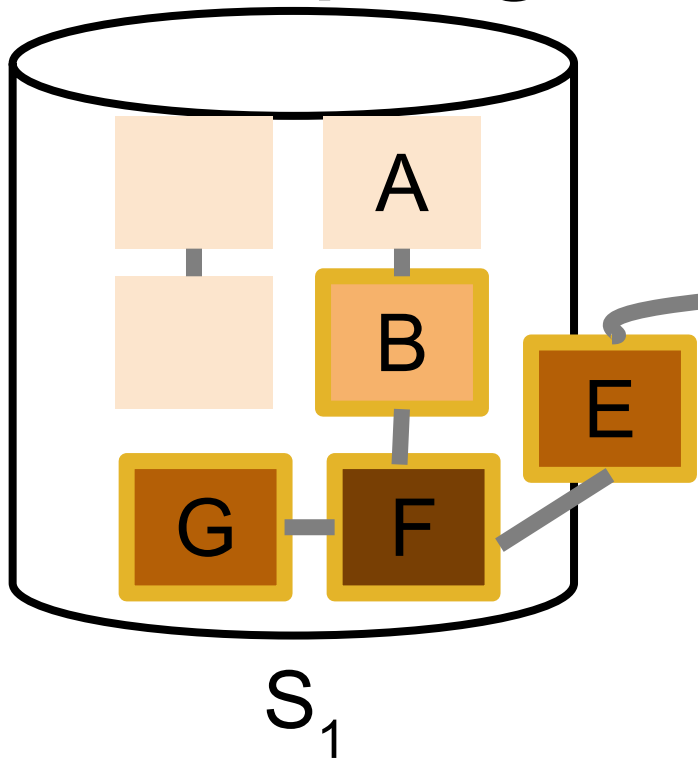


S_2

Migrate clump

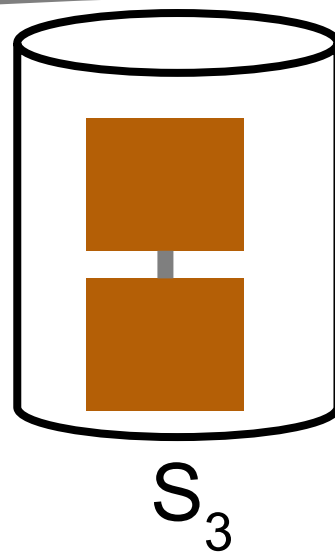


Clumping



(Serafini et al., VLDB 2016)

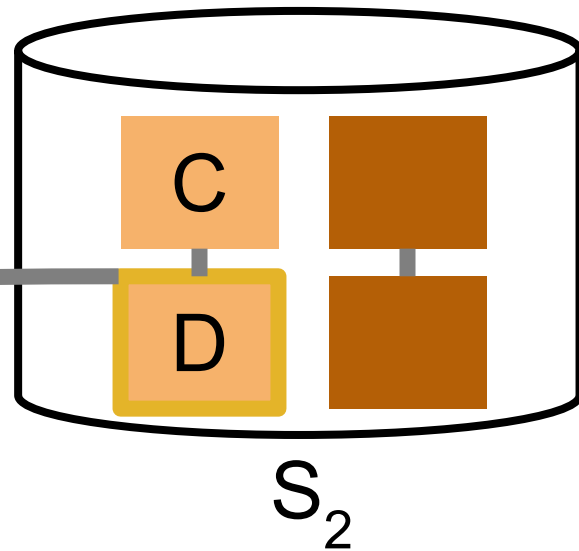
Termination



Low

Med

High



Clumping

Expands clumps to **frequently accessed neighbours**

Consider moving clump to **lightly loaded sites**

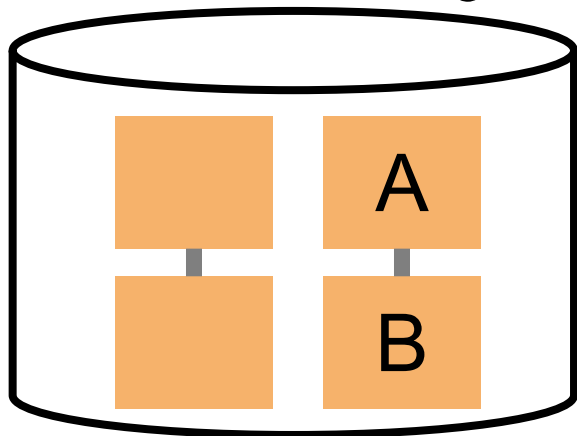
Considers both **re-partitioning and load costs**

Elasticity

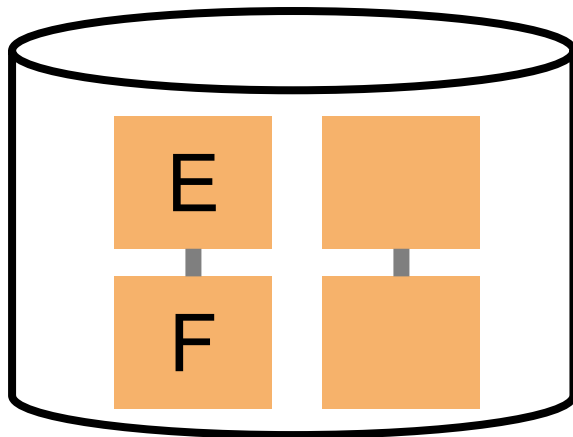
Low

Med

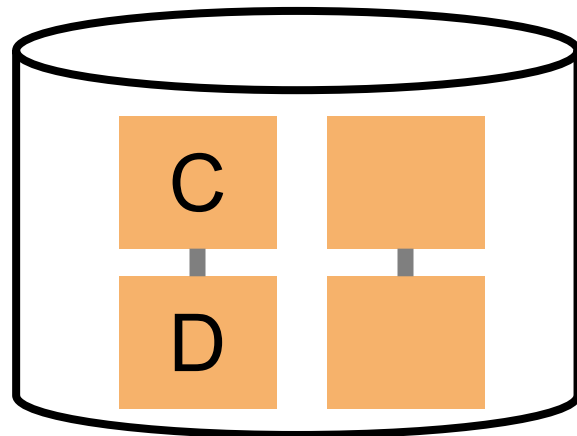
High



S_1



S_3



S_2

(Taft et al., VLDB 2015)

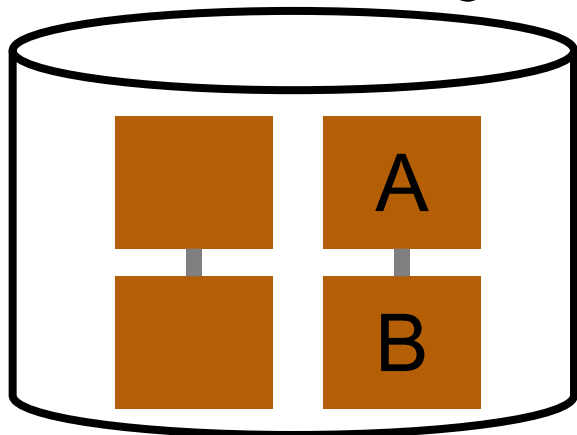


Elasticity

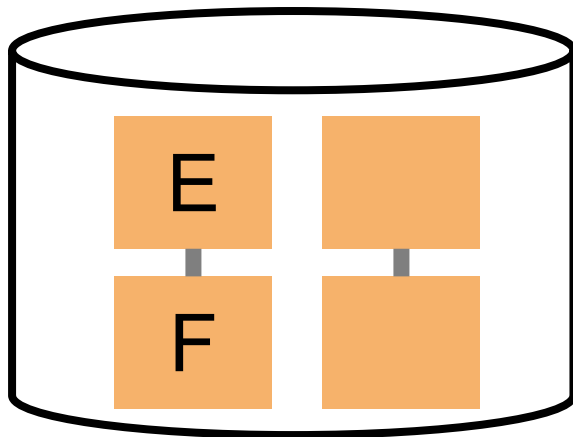
Low

Med

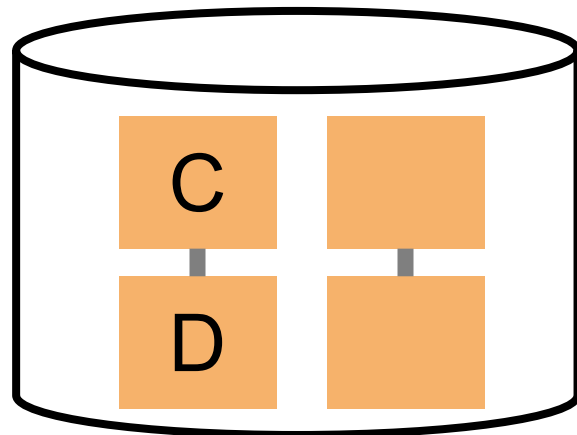
High



S_1



S_3



S_2

(Taft et al., VLDB 2015)

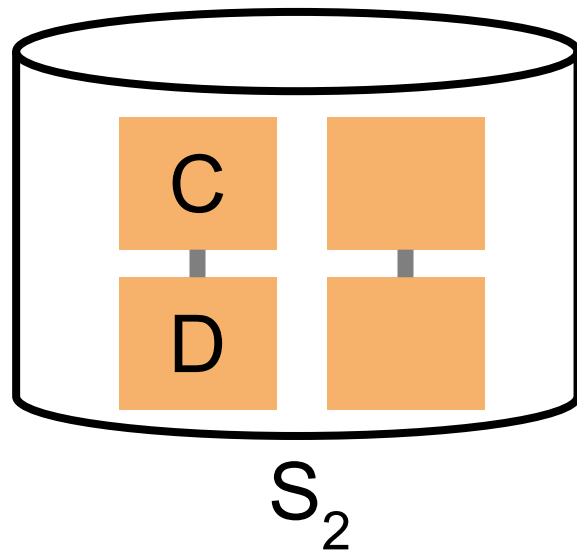
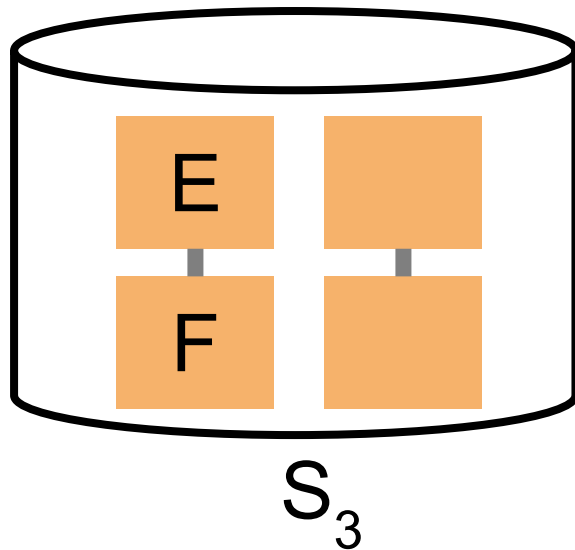
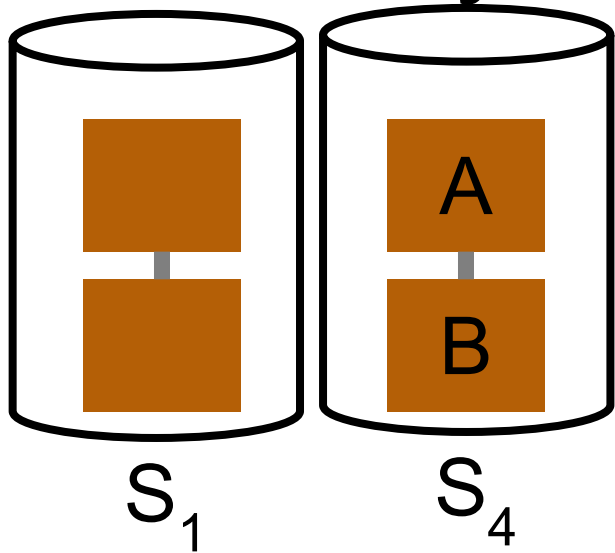


Elasticity

Low

Med

High



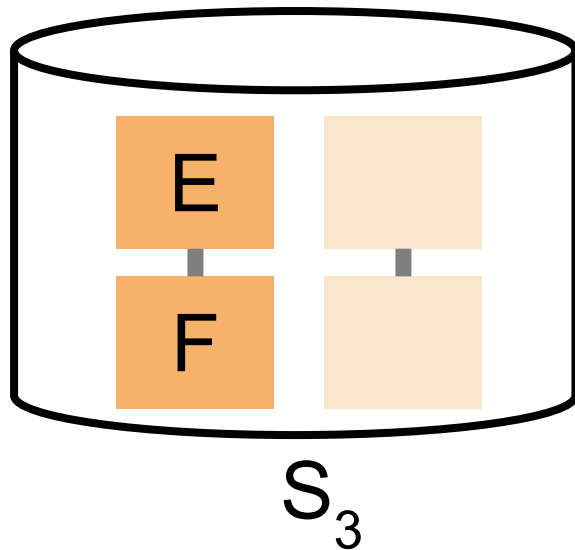
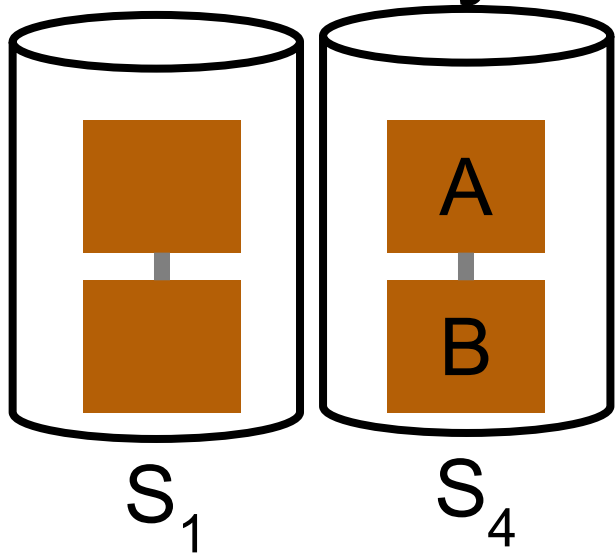
(Taft et al., VLDB 2015)

Elasticity

Repartition to **elastically** add or remove nodes

(Taft et al., VLDB 2015)

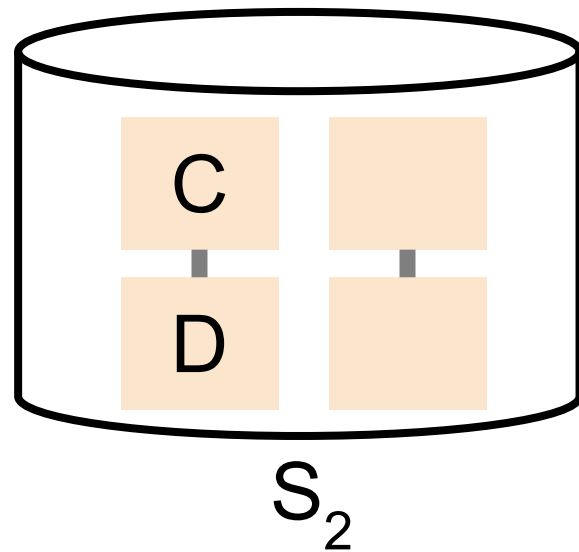
Elasticity



Low

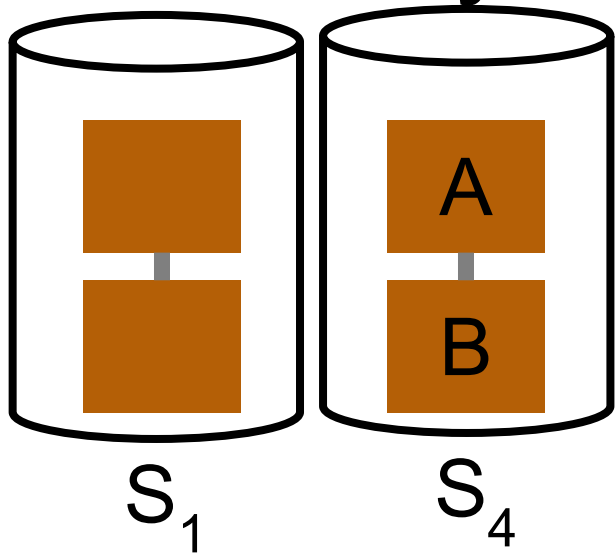
Med

High



(Taft et al., VLDB 2015)

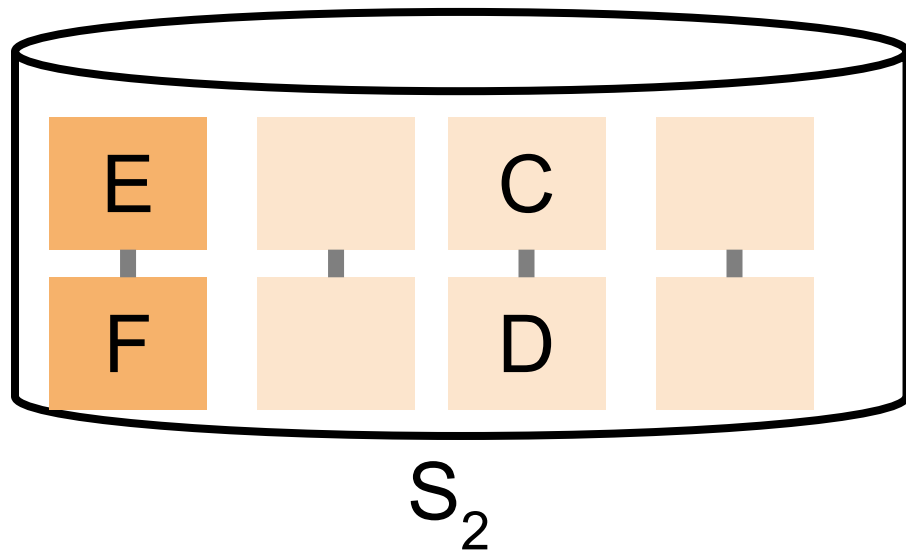
Elasticity



Low

Med

High



(Taft et al., VLDB 2015)

Elasticity Decisions

When the **average load**:

increases: add nodes

decreases: remove nodes

Two Tier Data Placement

Identify **hot** data

Evenly distribute hot data

Distribute cold data over remaining capacity

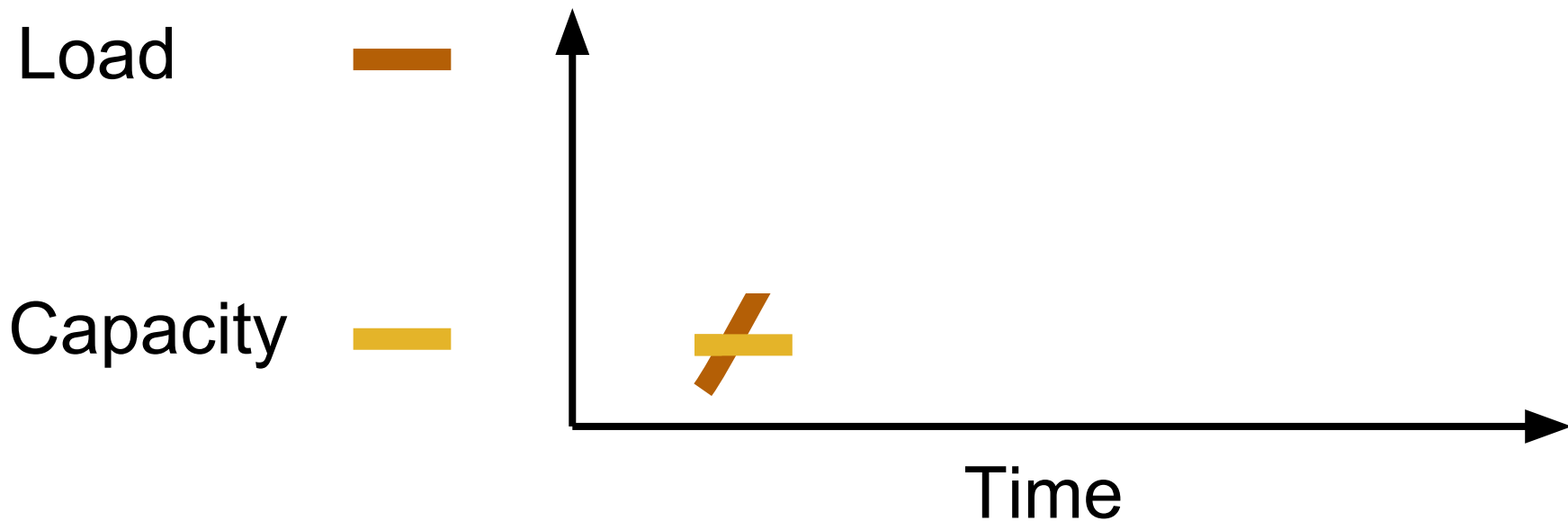
Identifying Hot Data

Monitor **partition level** access frequency

If **hot partition** enable **tuple level monitoring**

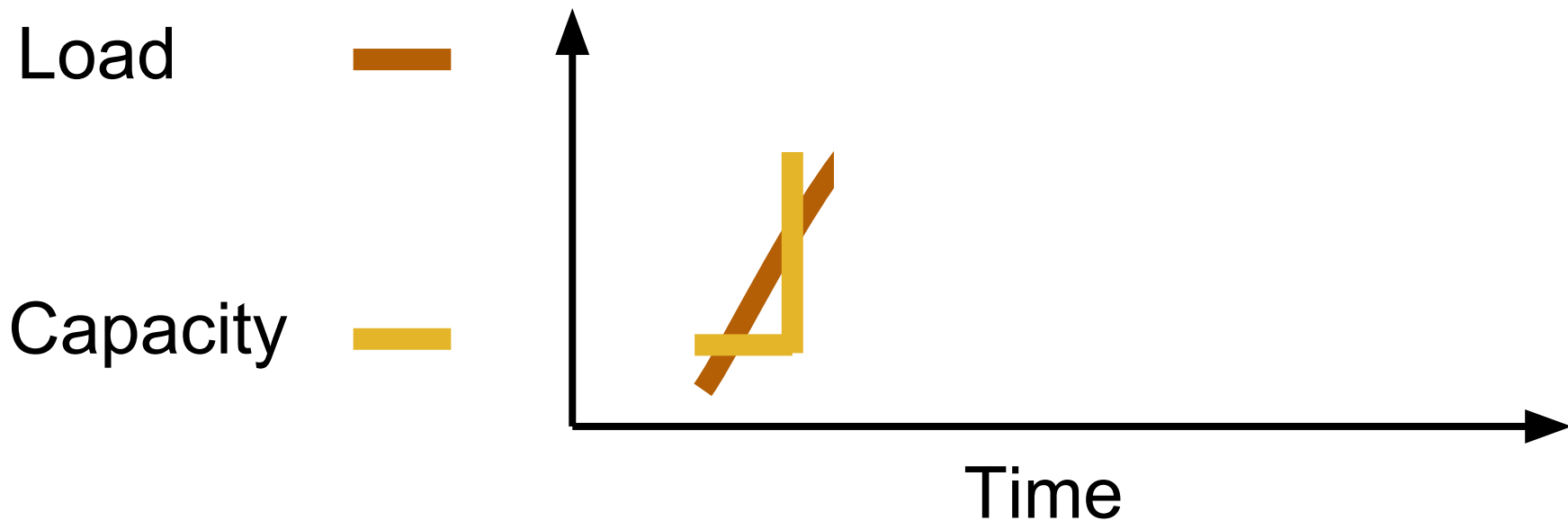
Reacts to changes in load

Reactive Elasticity



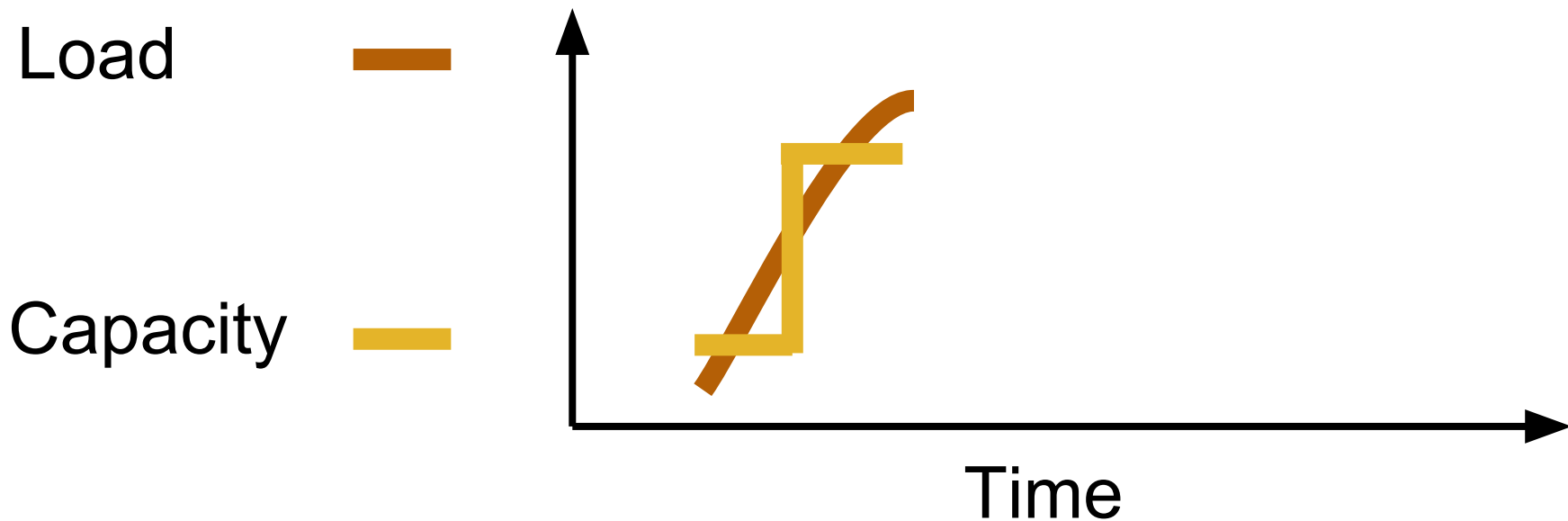
(Taft et al., VLDB 2015)

Reactive Elasticity



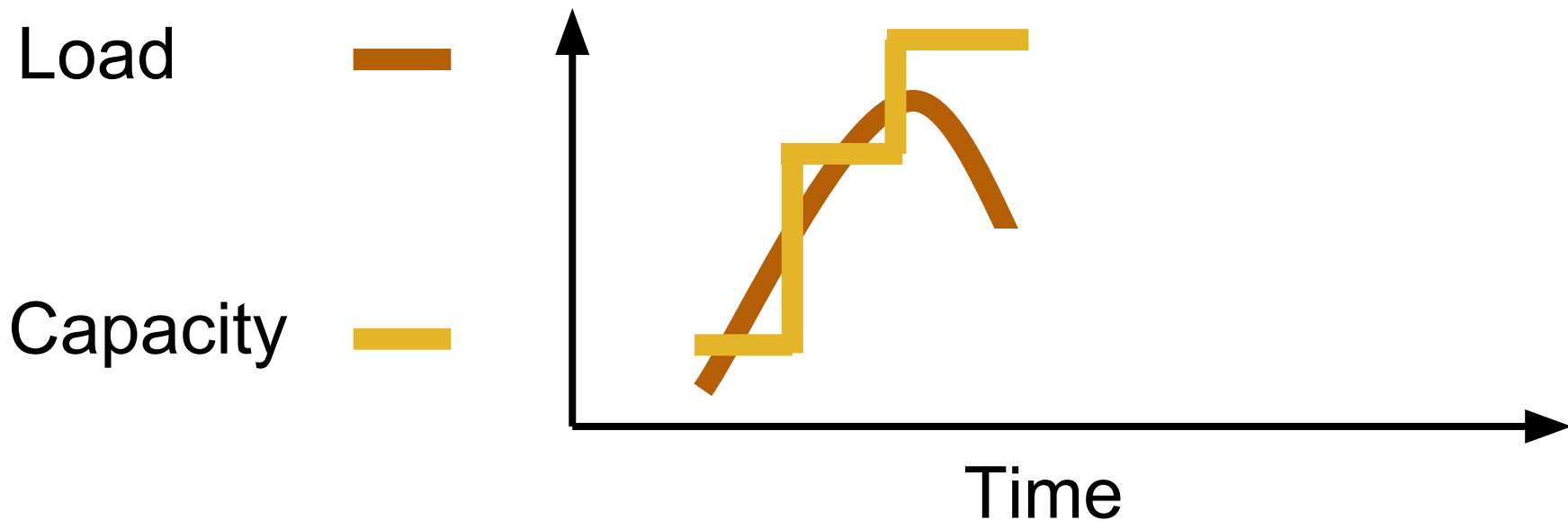
(Taft et al., VLDB 2015)

Reactive Elasticity



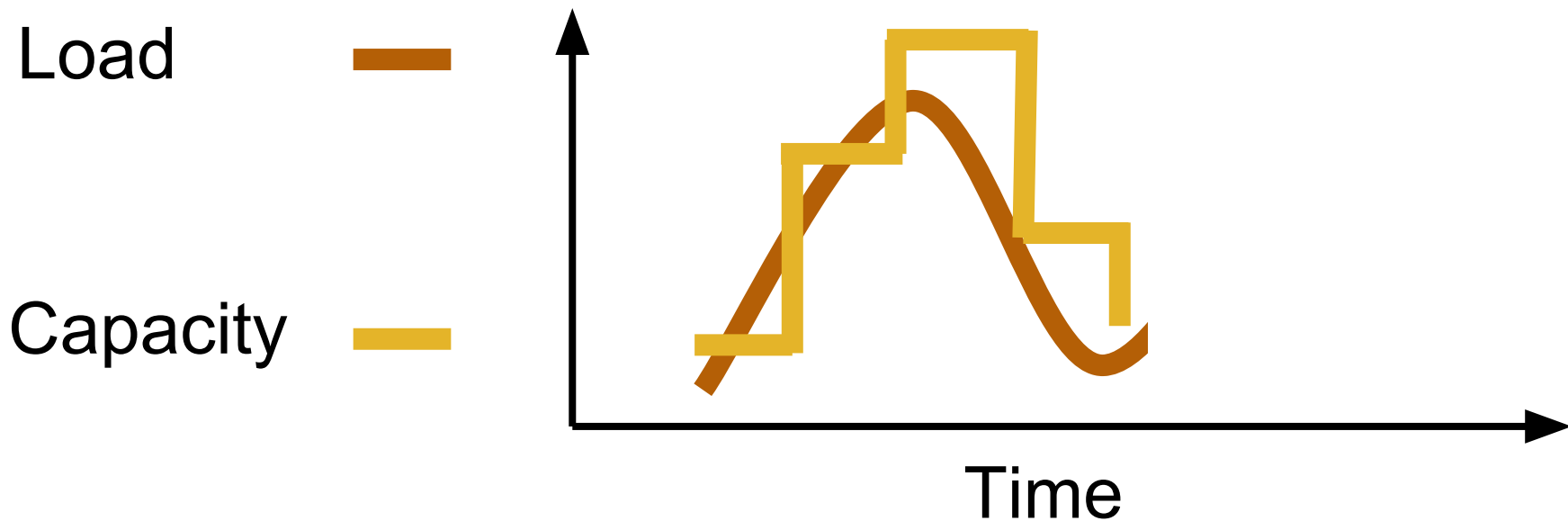
(Taft et al., VLDB 2015)

Reactive Elasticity



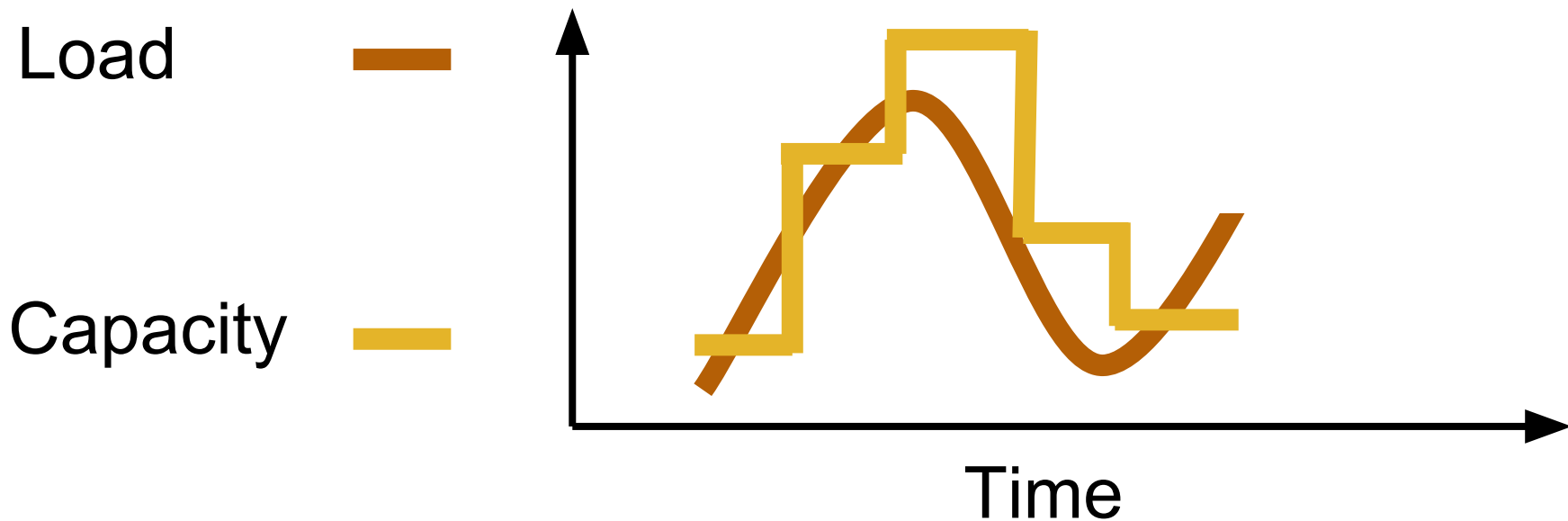
(Taft et al., VLDB 2015)

Reactive Elasticity



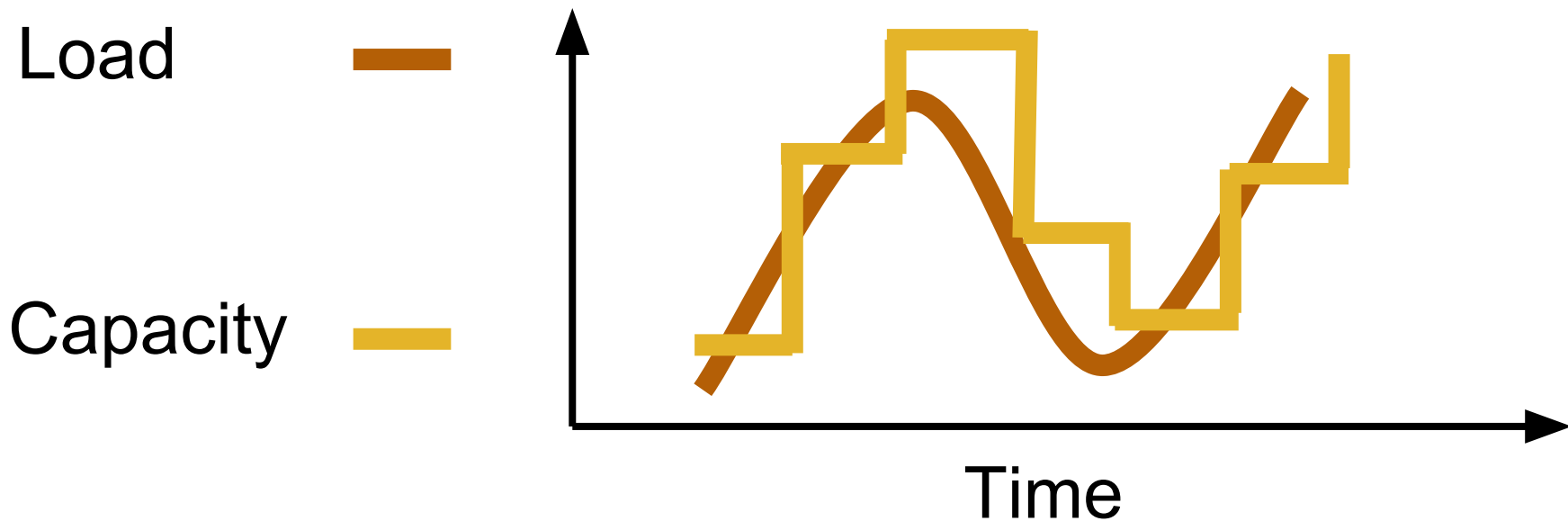
(Taft et al., VLDB 2015)

Reactive Elasticity



(Taft et al., VLDB 2015)

Reactive Elasticity



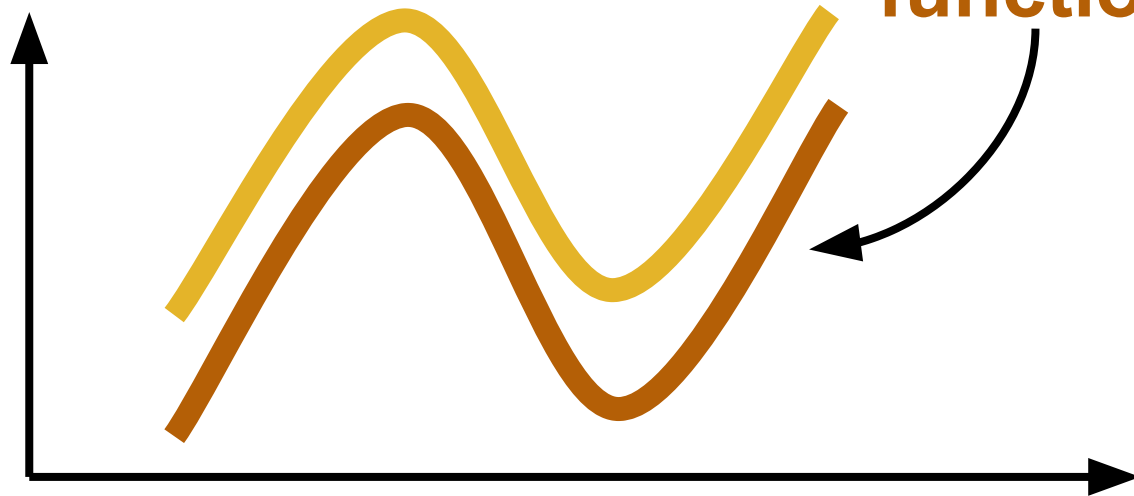
(Taft et al., VLDB 2015)

Ideal Elasticity

Load



Capacity

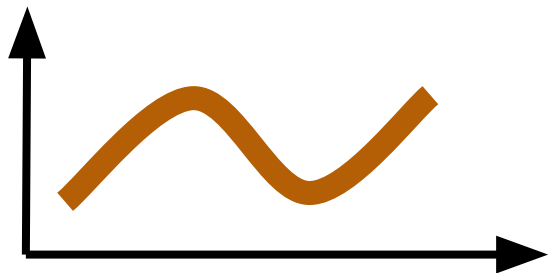


predict the
function

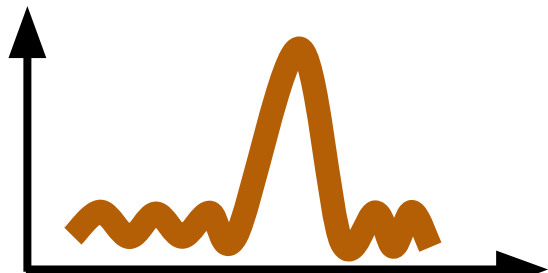
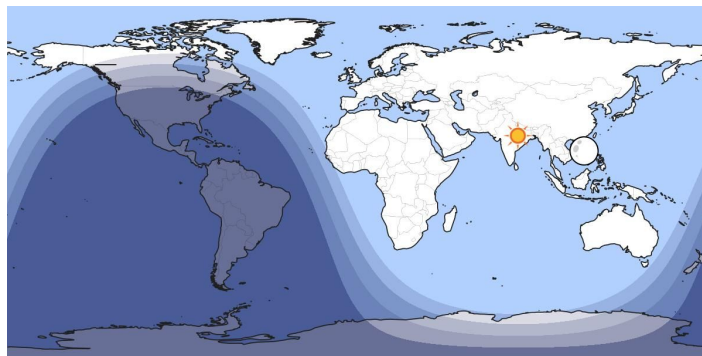
Time

(Taft et al., SIGMOD 2018)

Periodic Workloads



Daily load variations



Seasonal load spikes



(Taft et al., SIGMOD 2018)

How to Predict Load



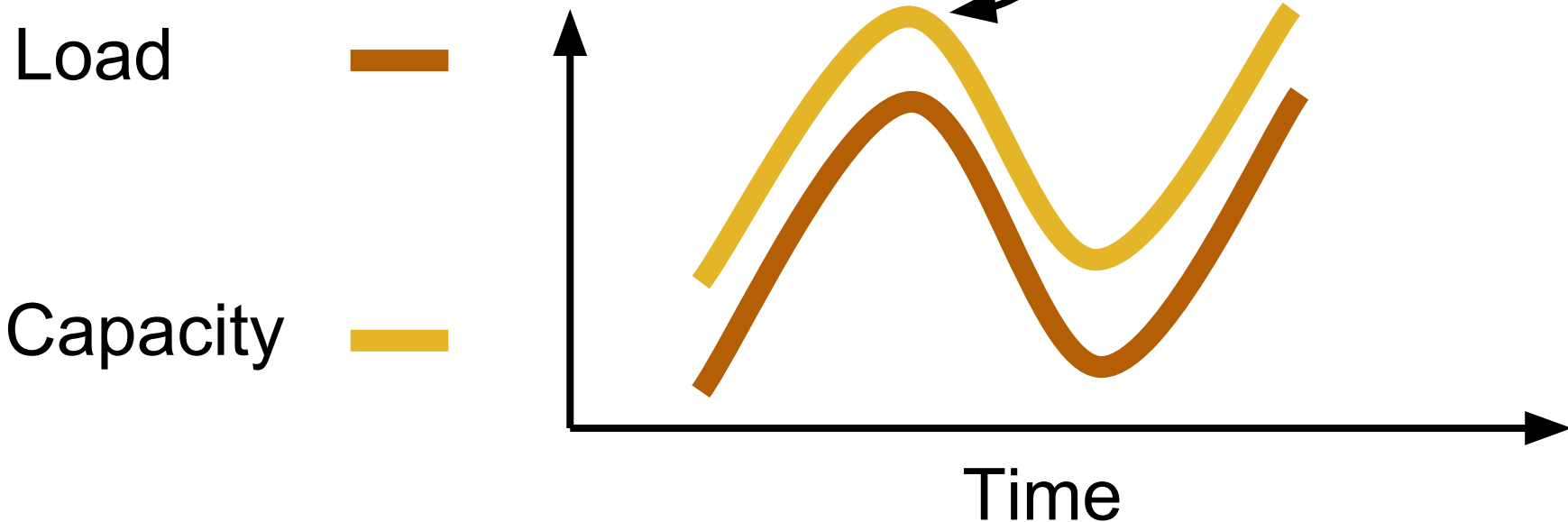
SPAR: Sparse Periodic Auto-Regression

$$\text{load}(t) = \text{avg_load}(t - p_i) + \text{change_in_load}(t - j_i)$$

(Taft et al., SIGMOD 2018)

Ideal Elasticity

decide the
of nodes

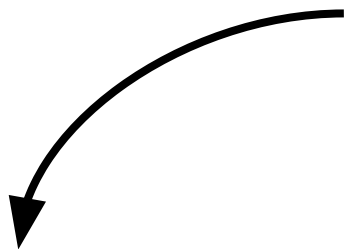


(Taft et al., SIGMOD 2018)

Number of Nodes

Assuming **partitionable**

of nodes = $\frac{\text{Predicted Load}}{\text{Load per Server}}$



(Taft et al., SIGMOD 2018)

Partitioning Decisions

- **How to form** partitions?

Heuristically (Clumping versus 2 Tier)

- **Where to place** partitions?

React or predict based on load

- **How to execute** multi-partition operations?

2PC

(Serafini et al., VLDB 2016)

(Taft et al., SIGMOD 2018)

(Taft et al., VLDB 2015)

Road Map

- Adaptive Replication
- Adaptive Partitioning
- Outlook

Outlook

Adaptive Systems

How to **make** a partitioning or replication **decision** **when** access patterns **change**?

Adaptively replicate and partition

Partitioning Decisions

- **How to form partitions?**
Iterative, Temporarily, Graph partitioning, Heuristic
- **Where to place partitions?**
Sorted, Leader, At requester, Graph partitioning, Reactively, Predictively
- **How to execute multi-partition operations?**
Novel protocols, 2PC

Replication Decisions

- **How many replicas?**
Decentralized, Client workload,
Cost-based, Predictive, Fault tolerance
- **Where to place replicas?**
At requester, Heuristic, Cost-based,
Predictive, Dynamic
- **How to propagate updates?**
Synchronous, Quorums,
Single-master, Cache

Decisions

- **How many** replicas? **Predictively**
- **Where to place** replicas? **Predictively**
- **Where to place** partitions? **Predictively**

Adaptive & Predictive Systems

How to **make** a partitioning or replication **decision** **when** access patterns **change**?

Adaptively and predictively
replicate and partition

Predicting the Future

How can your system **predict** its future workload?

Apollo: Predict future queries (Markov Model)

P-Store: Predict future load (SPAR)

Predicting the Future QB5000

When, how many, and what queries will arrive?

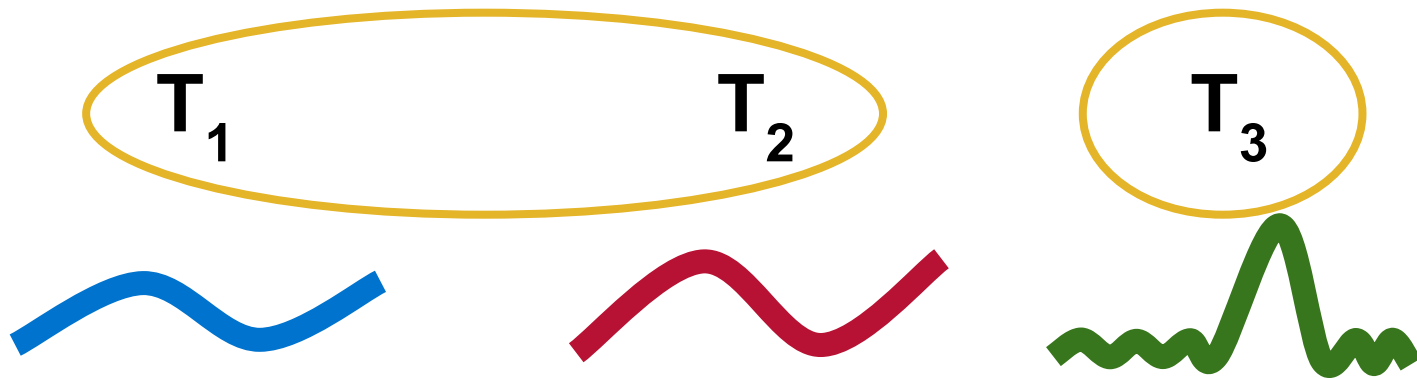
```
SELECT * FROM C WHERE id = "C1"
```

```
SELECT * FROM C WHERE id = $
```

Pre-process: **remove parameters**, creating templates

Predicting the Future QB5000

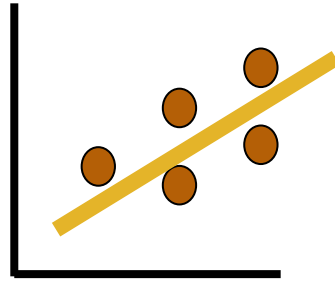
When, how many, and what queries will arrive?



Cluster: group templates by **arrival rate**

Predicting the Future QB5000

When, how many, and **what** queries will arrive?



Forecast: Predict clusters arrival rate
(**Ensemble** of RNN, LR, KR)

(Ma et al., SIGMOD 2018)

Predicting the Future QB5000

When, how many, and what queries will arrive?

Pre-process: **remove parameters**, creating templates

Cluster: group templates by **arrival rate**

Forecast: Predict clusters arrival rate
(**Ensemble** of RNN, LR, KR)

Predicting the Future

How can your system **predict** its future workload?

Apollo: Predict future queries (Markov Model)

P-Store: Predict future load (SPAR)

QB5000: Predict query workloads
(Ensemble of RNN, LR, KR)

Predicting the Future

How can your system **predict** its future workload?

If your system **knew the future workload**, how could it **partition and replicate** data?