Strong and Efficient Consistency with Consistency-Aware Durability

Aishwarya Ganesan, Ram Alagappan,

Andrea Arpaci-Dusseau, and Remzi Arpaci-Dusseau

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Distributed Storage Systems

Consistency Models in Distributed Systems

Consistency Models in Distributed Systems

What does a read see given a previous set of reads and writes?

Consistency Models in Distributed Systems

What does a read see given a previous set of reads and writes?

strong ●
linearizability

Consistency Models in Distributed Systems

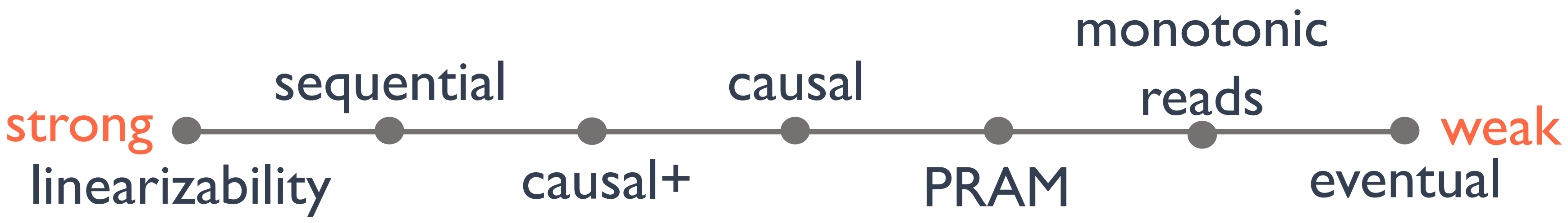
What does a read see given a previous set of reads and writes?

strong — weak

linearizability — eventual

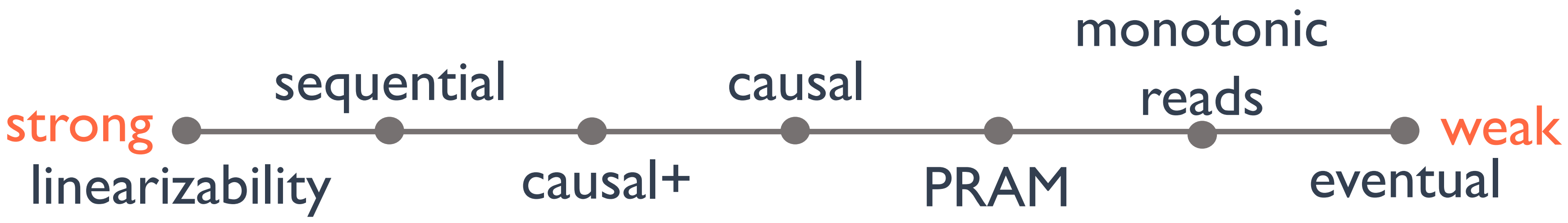
Consistency Models in Distributed Systems

What does a read see given a previous set of reads and writes?



Consistency Models in Distributed Systems

What does a read see given a previous set of reads and writes?



Well studied and understood!

Durability Models

Unlike consistency models, scant attention to durability model!

Durability Models

Unlike consistency models, scant attention to durability model!

How writes are replicated and persisted

Durability Models

Unlike consistency models, scant attention to durability model!

How writes are replicated and persisted

Durability model influences consistency

Durability Models

Unlike consistency models, scant attention to durability model!

How writes are replicated and persisted

Durability model influences consistency

Also determines performance

Durability Models

Unlike consistency models, scant attention to durability model!

How writes are replicated and persisted

Durability model influences consistency

Also determines performance

Despite this importance, often overlooked!

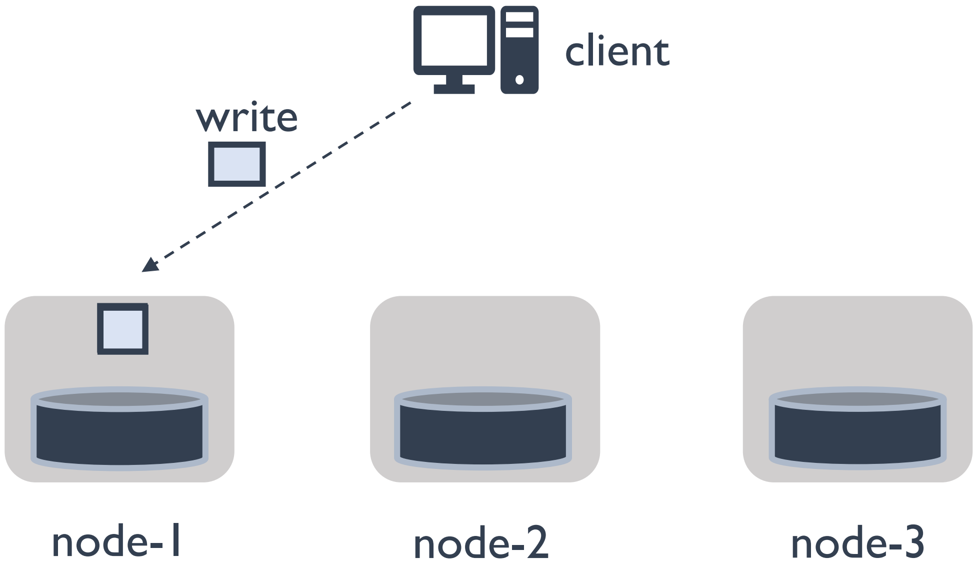
Two Widely Used Durability Models

Two Widely Used Durability Models

Immediate durability

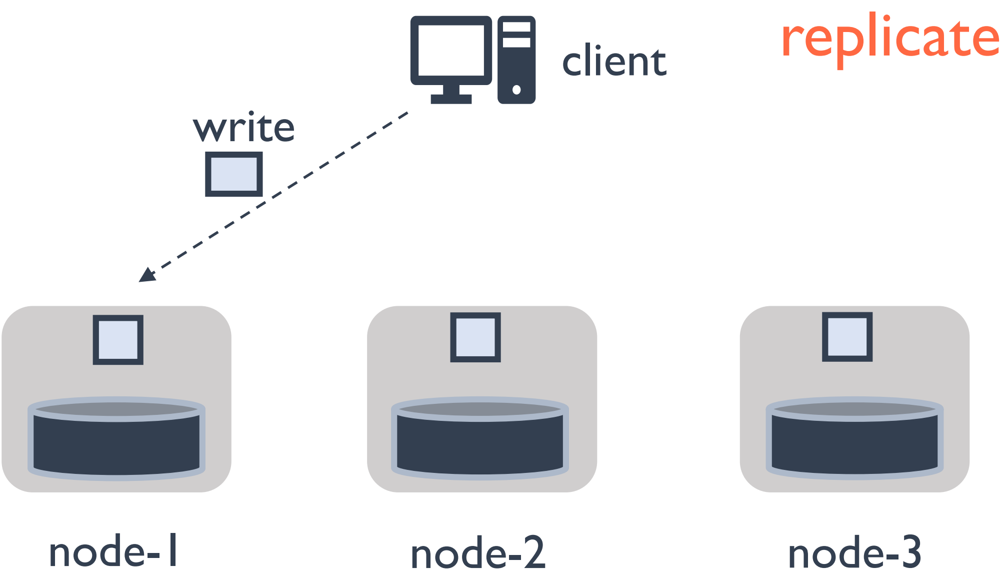
Two Widely Used Durability Models

Immediate durability



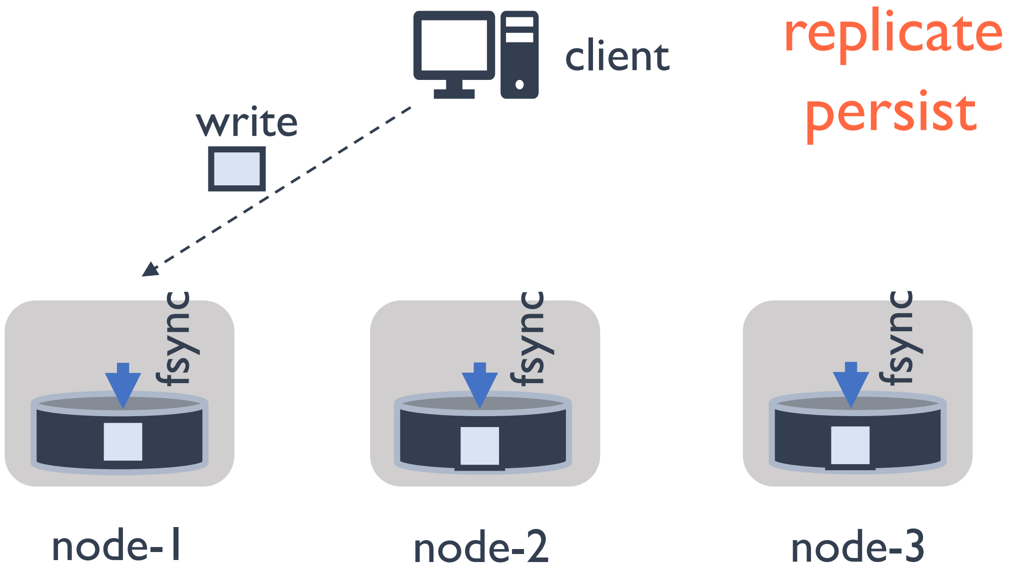
Two Widely Used Durability Models

Immediate durability



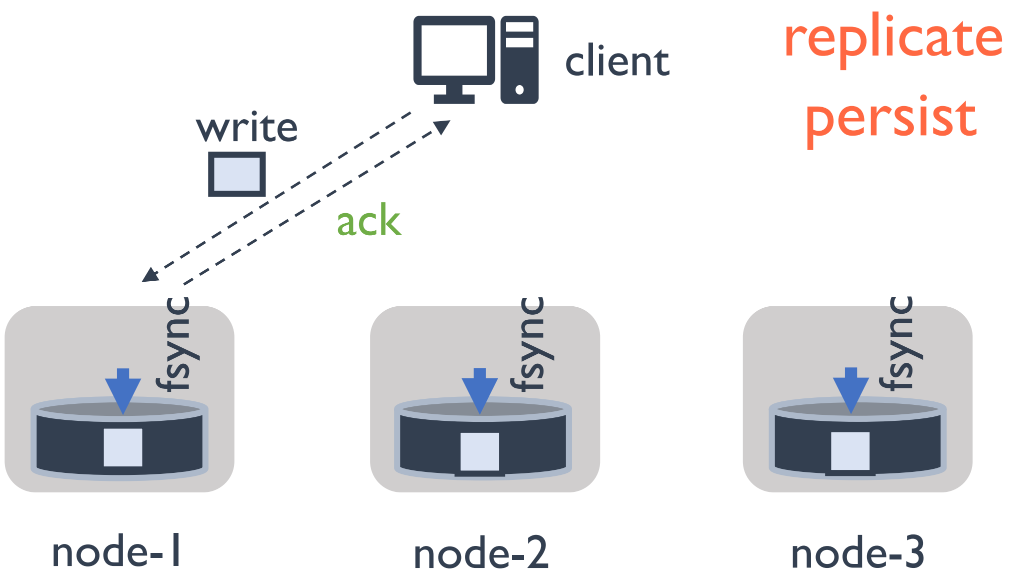
Two Widely Used Durability Models

Immediate durability



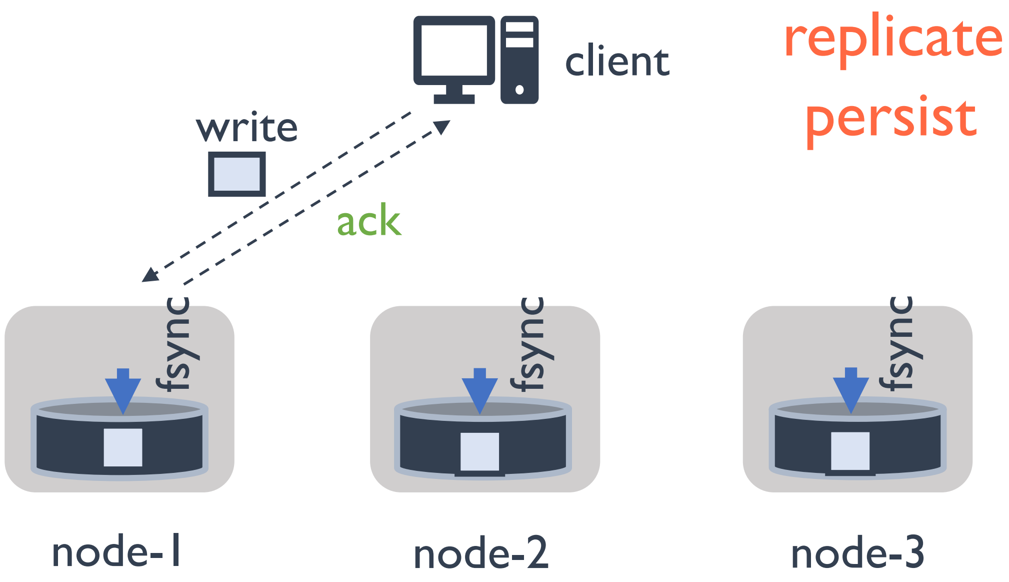
Two Widely Used Durability Models

Immediate durability



Two Widely Used Durability Models

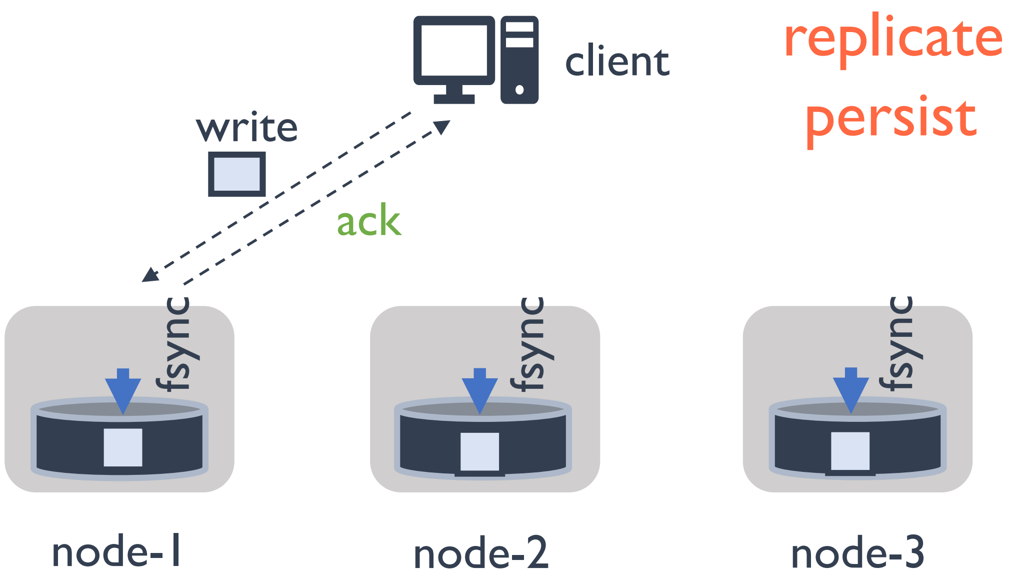
Immediate durability



enables strong consistency
but too slow!

Two Widely Used Durability Models

Immediate durability

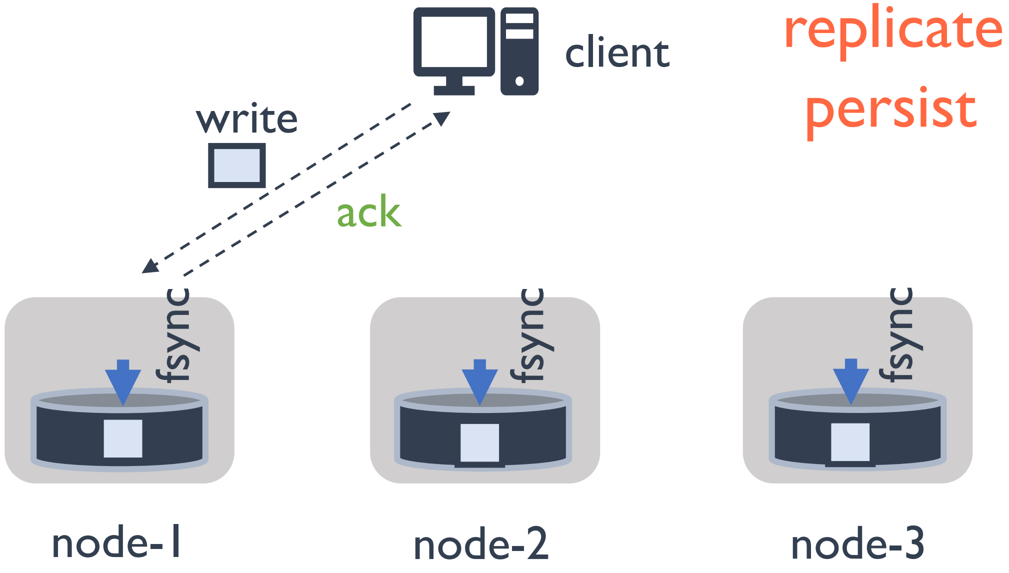


enables strong consistency
but too slow!

Eventual durability

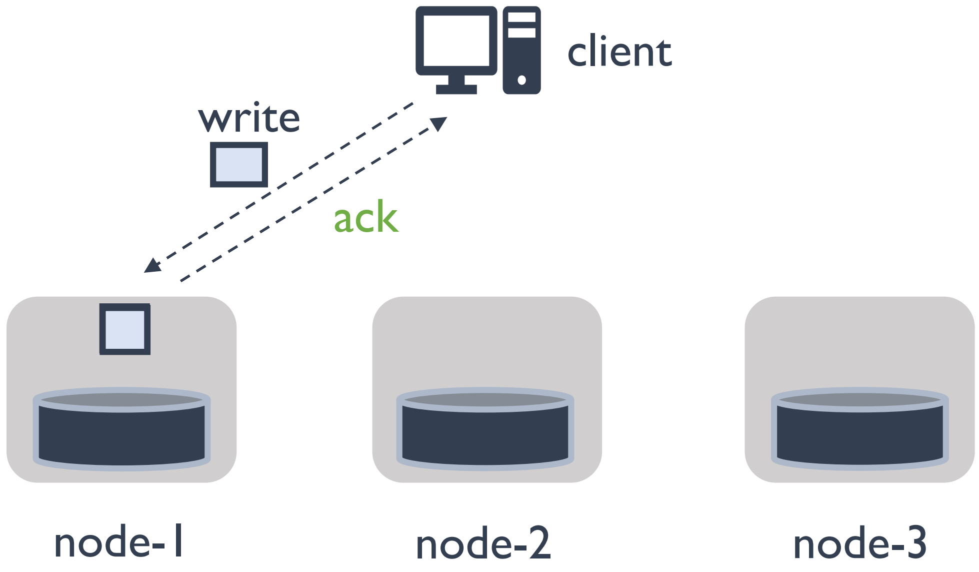
Two Widely Used Durability Models

Immediate durability



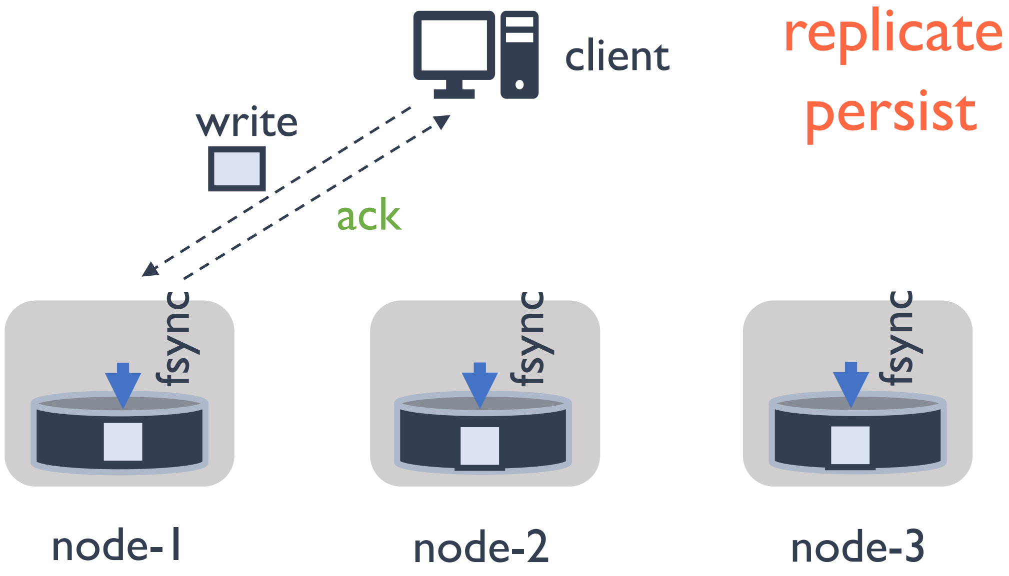
enables strong consistency
but too slow!

Eventual durability



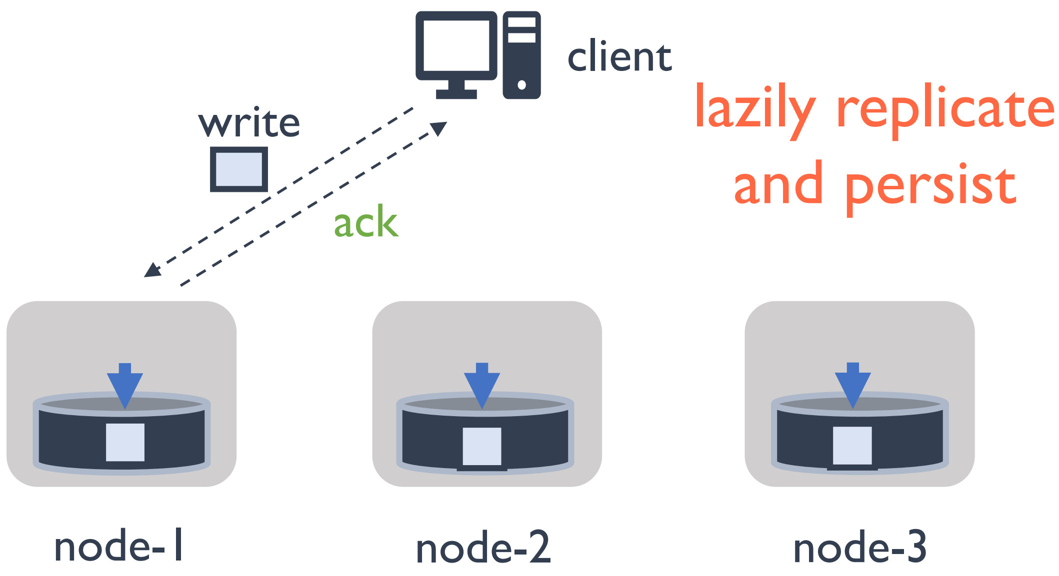
Two Widely Used Durability Models

Immediate durability



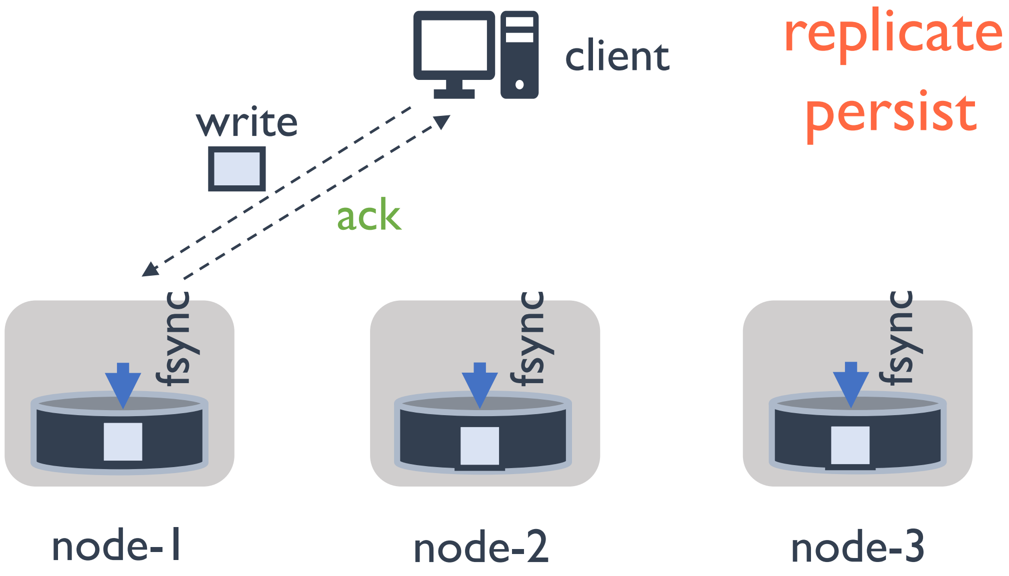
enables strong consistency
but too slow!

Eventual durability



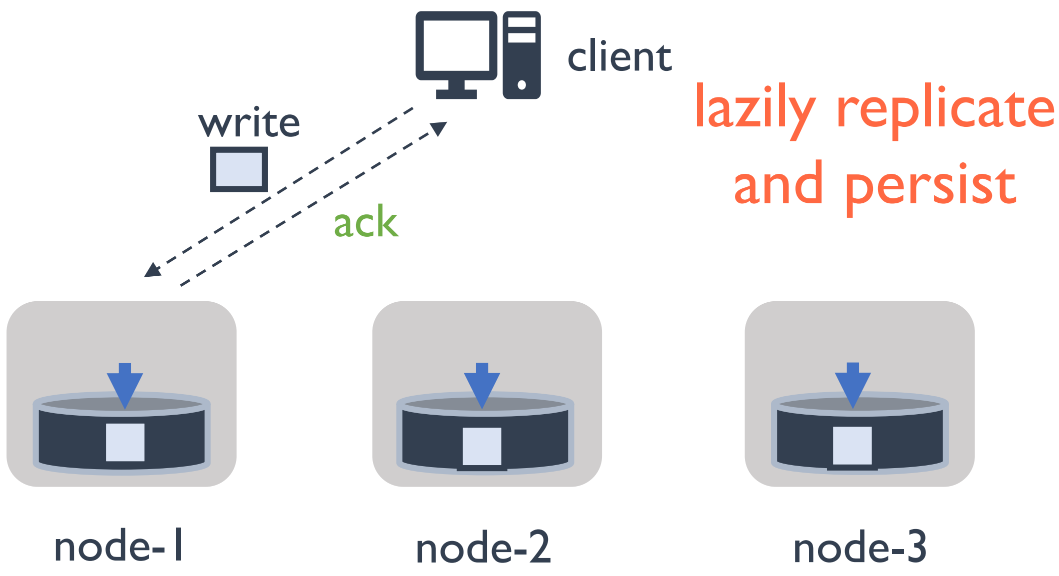
Two Widely Used Durability Models

Immediate durability



enables strong consistency
but too slow!

Eventual durability



fast
but enables only weak consistency due to data loss upon failures!

Is it possible for a durability layer to enable *both* strong consistency and high performance?

CAD: Consistency-aware Durability

CAD: Consistency-aware Durability

Design the durability layer by taking the consistency model into account

CAD: Consistency-aware Durability

Design the durability layer by taking the consistency model into account

Intuition: what a read sees is important for most consistency models

CAD: Consistency-aware Durability

Design the durability layer by taking the consistency model into account

Intuition: what a read sees is important for most consistency models

Key idea: CAD shifts the point of durability to reads from writes

data is replicated and persisted before a read is served

CAD: Consistency-aware Durability

Design the durability layer by taking the consistency model into account

Intuition: what a read sees is important for most consistency models

Key idea: CAD shifts the point of durability to reads from writes

- data is replicated and persisted before a read is served
- delayed writes → high performance

CAD: Consistency-aware Durability

Design the durability layer by taking the consistency model into account

Intuition: what a read sees is important for most consistency models

Key idea: CAD shifts the point of durability to reads from writes

- data is replicated and persisted before a read is served
- delayed writes → high performance
- data durable before it is read → strong consistency even under failures

CAD: Consistency-aware Durability

Design the durability layer by taking the consistency model into account

Intuition: what a read sees is important for most consistency models

Key idea: CAD shifts the point of durability to reads from writes

- data is replicated and persisted before a read is served
- delayed writes → high performance
- data durable before it is read → strong consistency even under failures
- lose some writes if failures arise before read; but, useful for many systems that use eventual durability

CAD: Consistency-aware Durability

Design the durability layer by taking the consistency model into account

Intuition: what a read sees is important for most consistency models

Key idea: CAD shifts the point of durability to reads from writes

- data is replicated and persisted before a read is served
- delayed writes → high performance
- data durable before it is read → strong consistency even under failures
- lose some writes if failures arise before read; but, useful for many systems that use eventual durability

We show efficacy of CAD by providing cross-client monotonic reads

- a new and strong consistency property

Results

Results

ORCA: CAD and cross-client monotonic reads for leader-based systems

implemented in ZooKeeper

Results

ORCA: CAD and cross-client monotonic reads for leader-based systems

- implemented in ZooKeeper

Compared to strongly consistent ZooKeeper

- ORCA is 1.6 – 3.3x faster by using CAD
- higher read throughput by allowing reads at many nodes
- reduces latency in geo-distributed settings by 14x

Results

ORCA: CAD and cross-client monotonic reads for leader-based systems

  implemented in ZooKeeper

Compared to strongly consistent ZooKeeper

  ORCA is 1.6 – 3.3x faster by using CAD

  higher read throughput by allowing reads at many nodes

  reduces latency in geo-distributed settings by 14x

Compared to weakly consistent ZooKeeper

  ORCA provides similar throughput and latency

  but with stronger guarantees

Results

ORCA: CAD and cross-client monotonic reads for leader-based systems

- implemented in ZooKeeper

Compared to strongly consistent ZooKeeper

- ORCA is 1.6 – 3.3x faster by using CAD
- higher read throughput by allowing reads at many nodes
- reduces latency in geo-distributed settings by 14x

Compared to weakly consistent ZooKeeper

- ORCA provides similar throughput and latency
- but with stronger guarantees

Experimentally show ORCA’s guarantees under failures, useful for apps

Outline

Introduction

Motivation

CAD and cross-client monotonic reads

ORCA design

Results

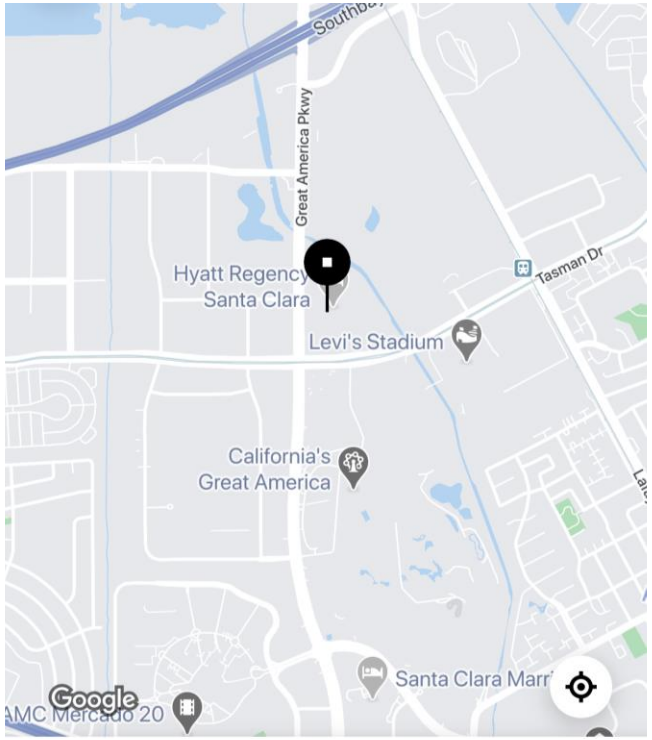
Summary and conclusion

Consistency Models and Guarantees

Example: I’m bored at FAST and want to go home!

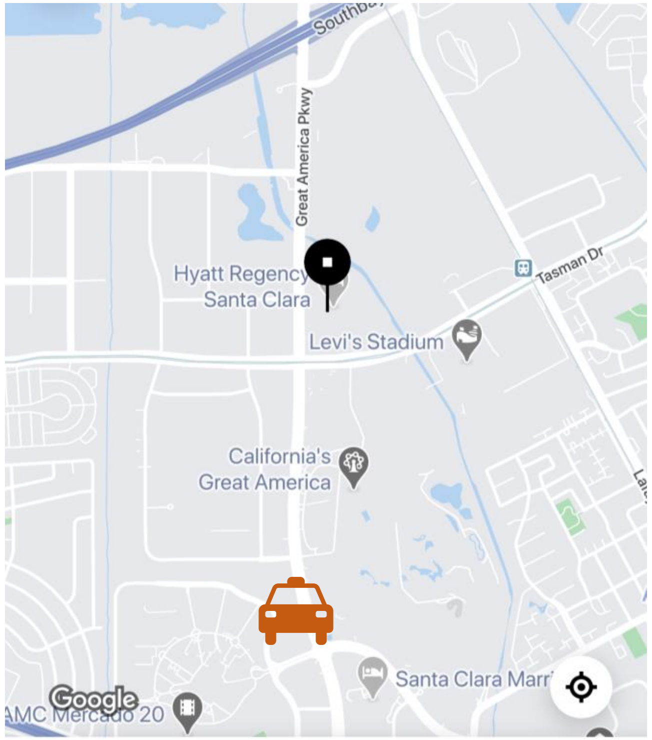
Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!



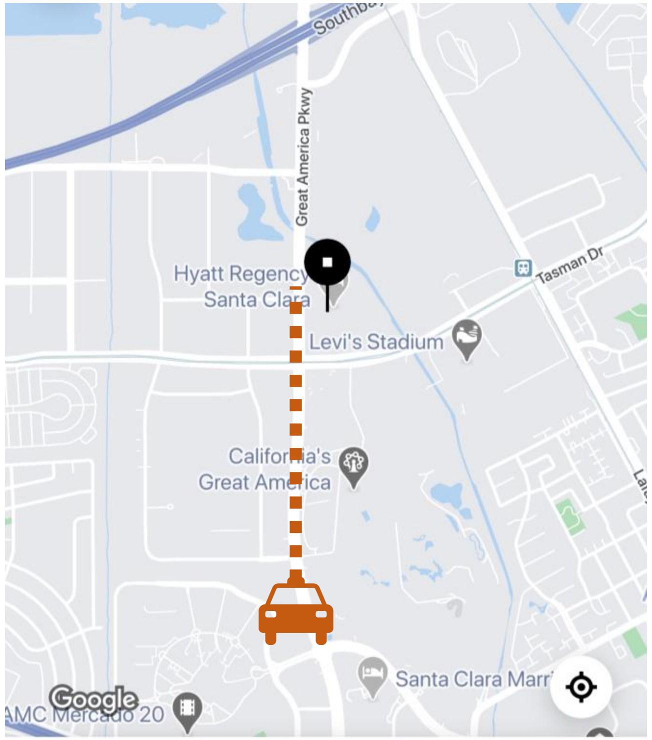
Consistency Models and Guarantees

Example: I’m bored at FAST and want to go home!



Consistency Models and Guarantees

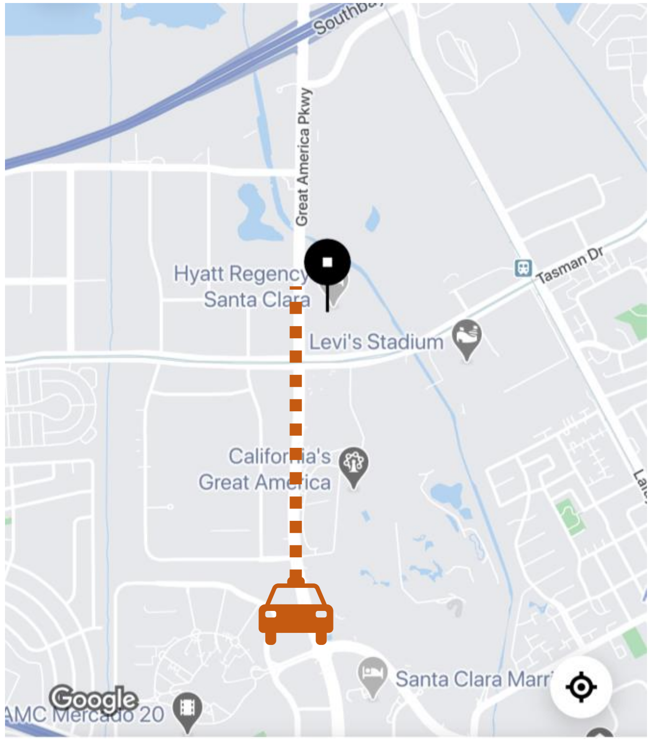
Example: I'm bored at FAST and want to go home!



Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

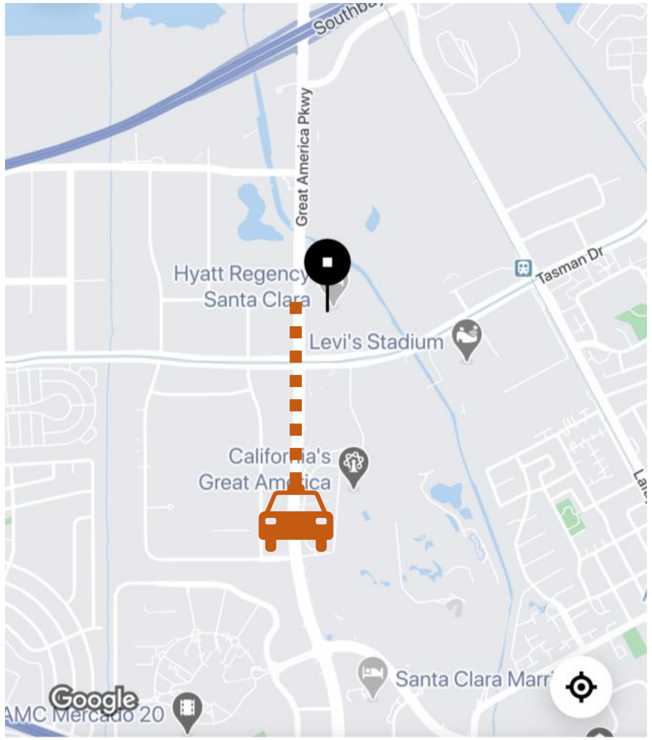
Linearizability



Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

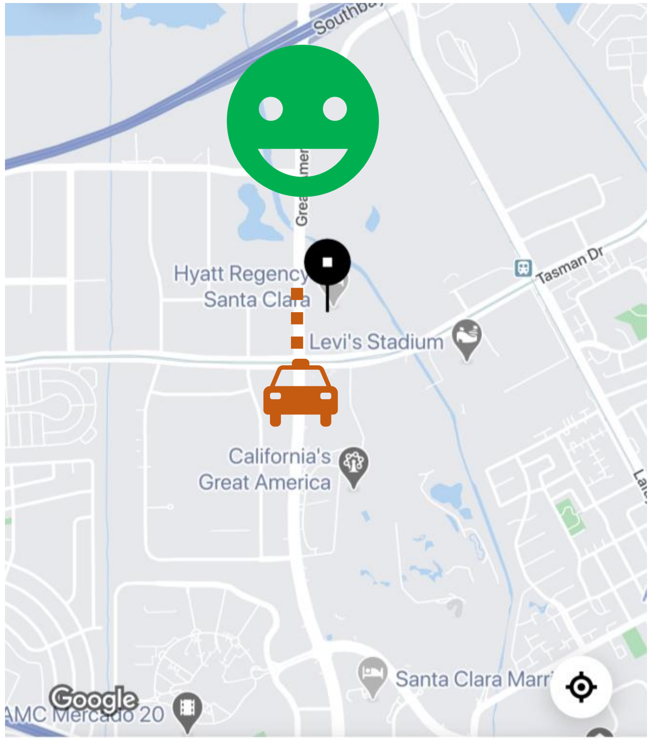
Linearizability



Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

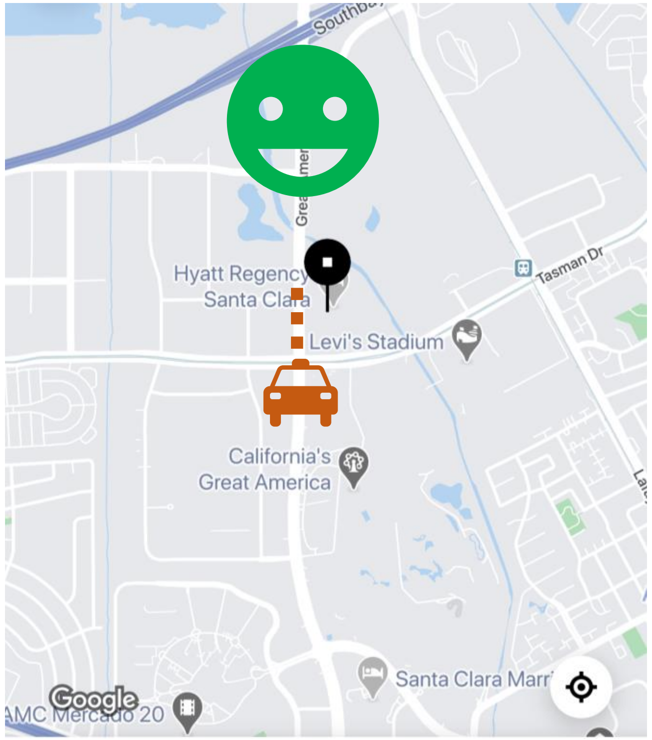
Linearizability



Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

Linearizability

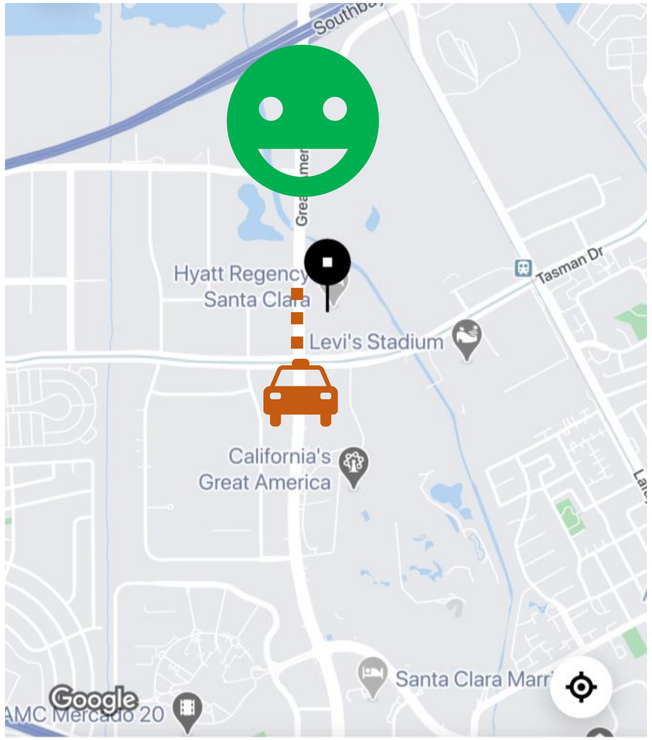


latest data: no staleness

Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

Linearizability



latest data: no staleness

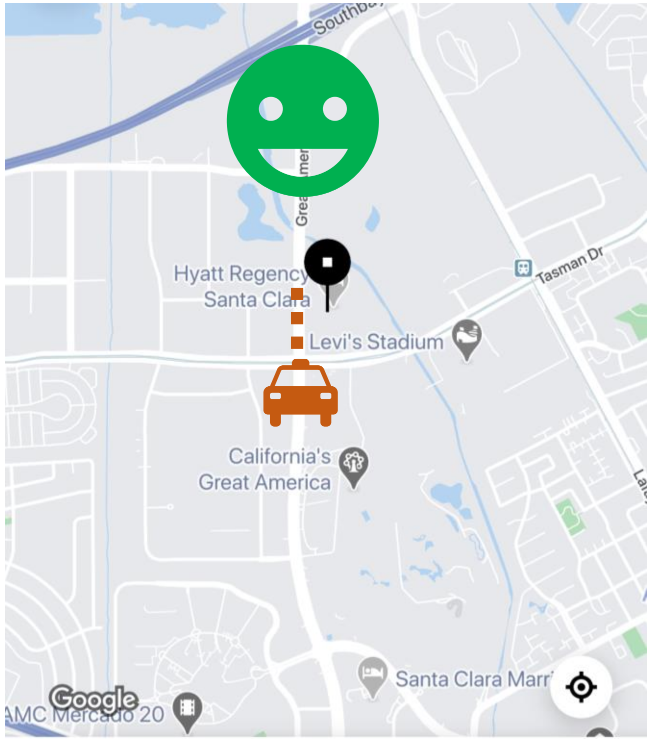
in-order reads across clients

Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

Linearizability

Weaker models



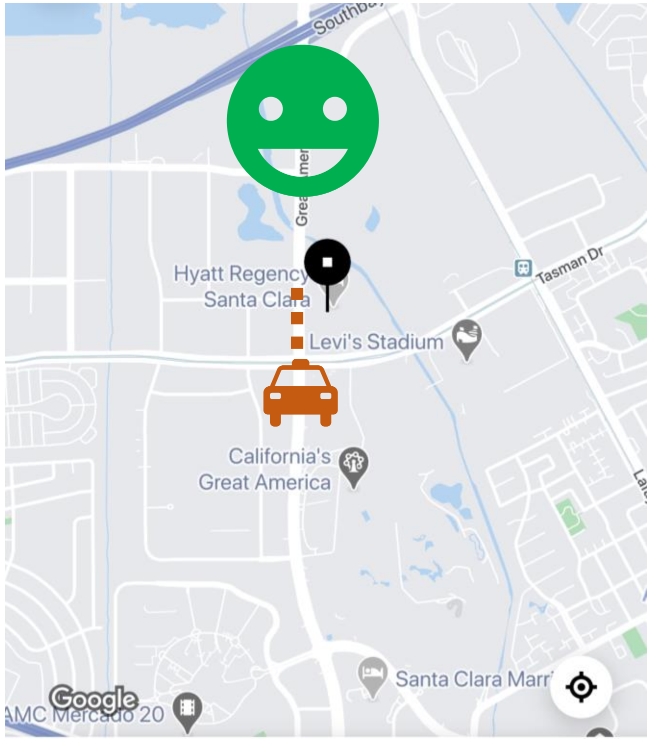
latest data: no staleness

in-order reads across clients

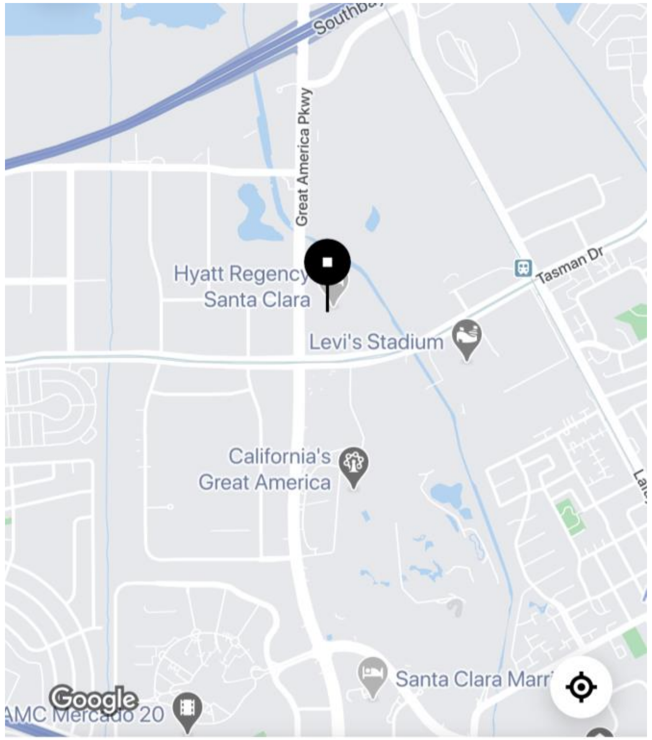
Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

Linearizability



Weaker models



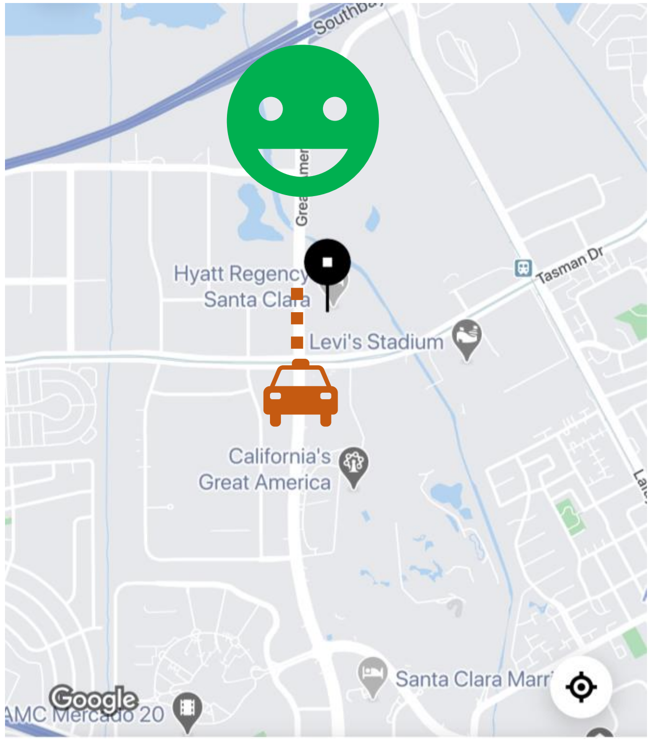
latest data: no staleness

in-order reads across clients

Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

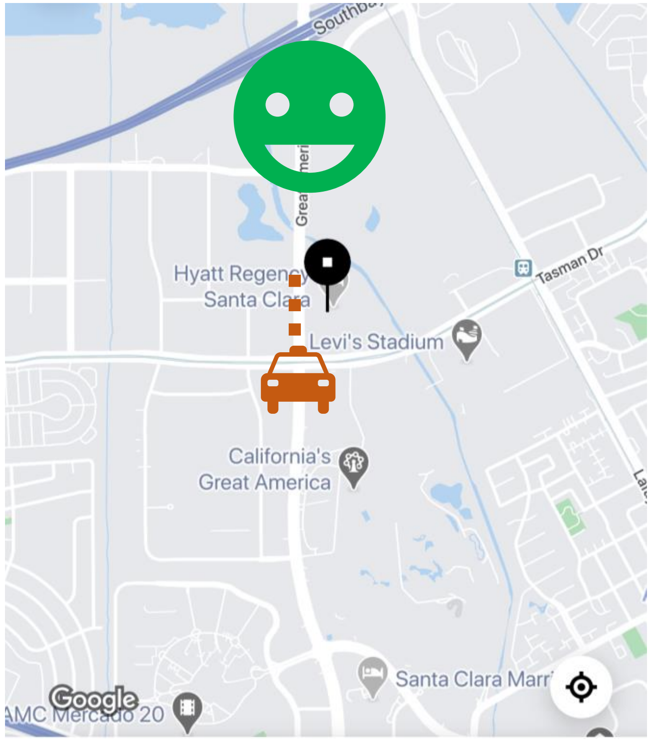
Linearizability



latest data: no staleness

in-order reads across clients

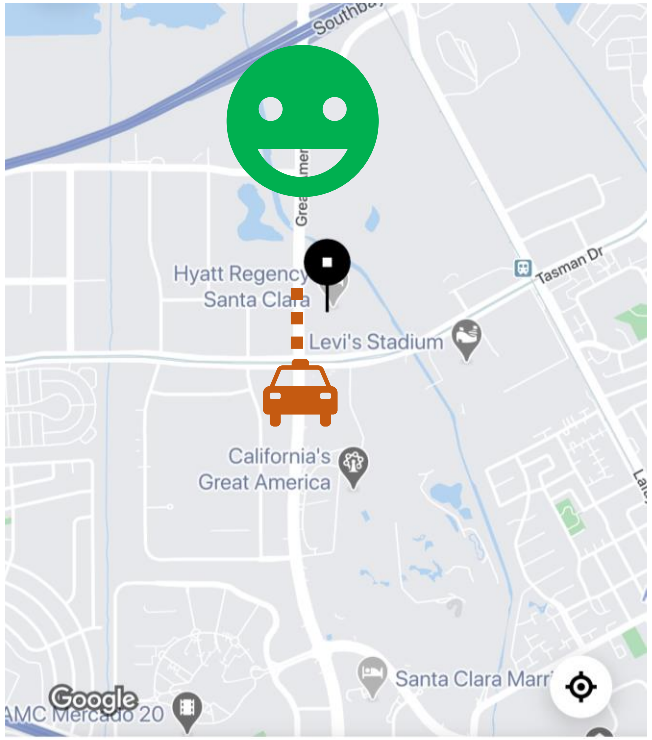
Weaker models



Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

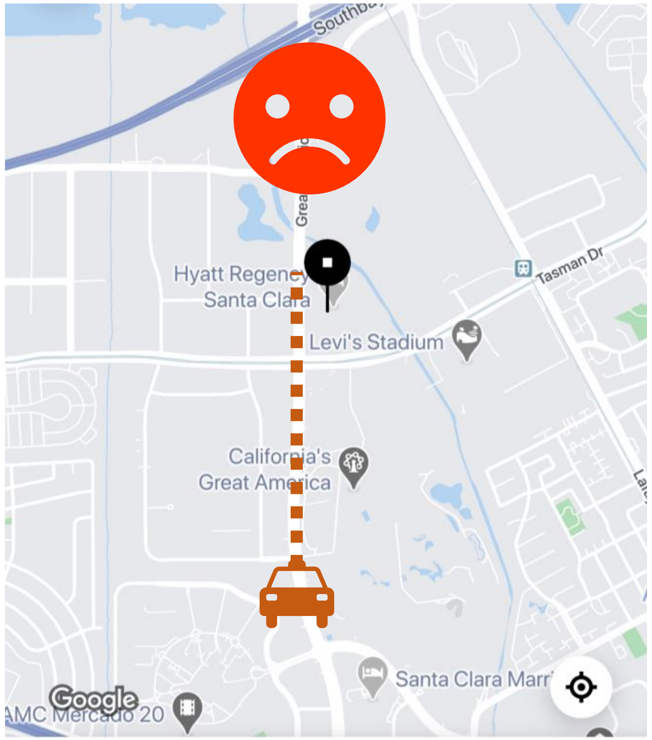
Linearizability



latest data: no staleness

in-order reads across clients

Weaker models



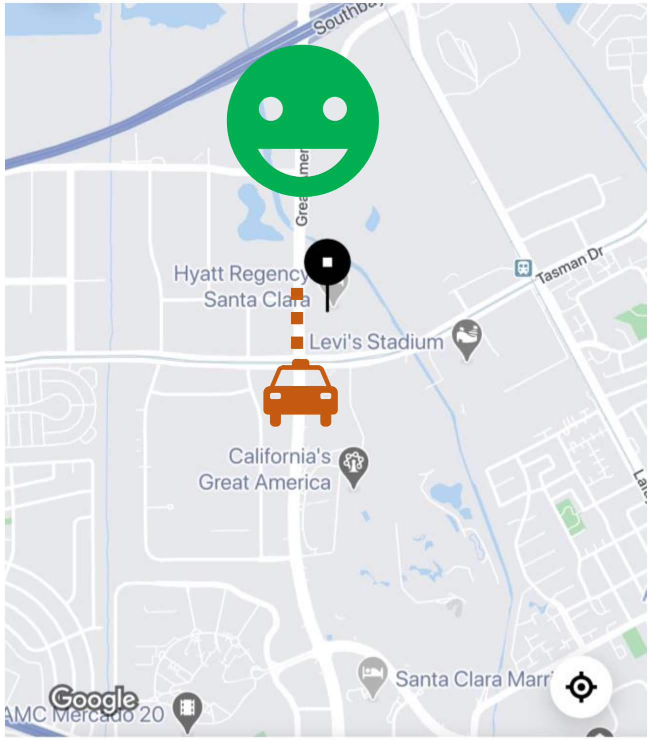
stale reads

out-of-order reads across clients

Consistency Models and Guarantees

Example: I'm bored at FAST and want to go home!

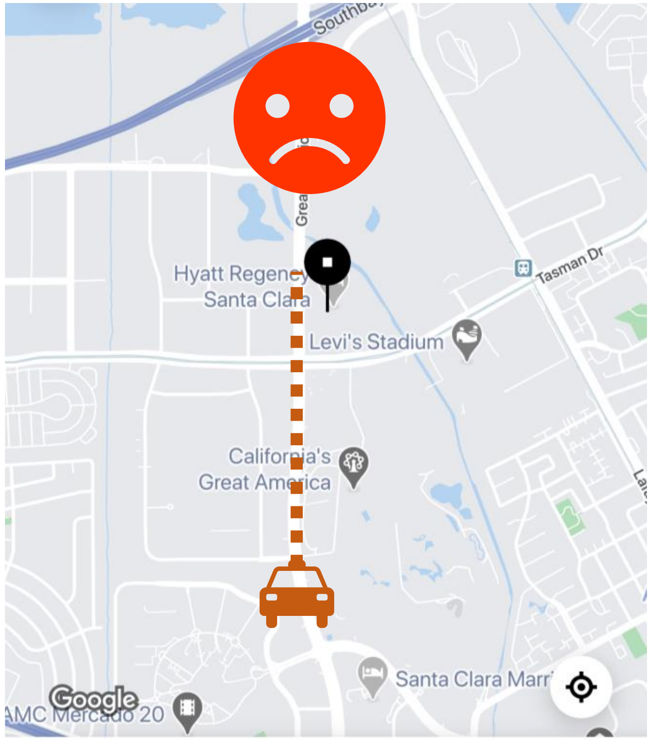
Linearizability



latest data: no staleness

in-order reads across clients

Weaker models



stale reads

out-of-order reads across clients

even with monotonic reads and causal

Realizing Strong Consistency

- Linearizability requires immediate durability
  - must synchronously replicate and persist data on a majority to tolerate failures

Realizing Strong Consistency

Linearizability requires immediate durability

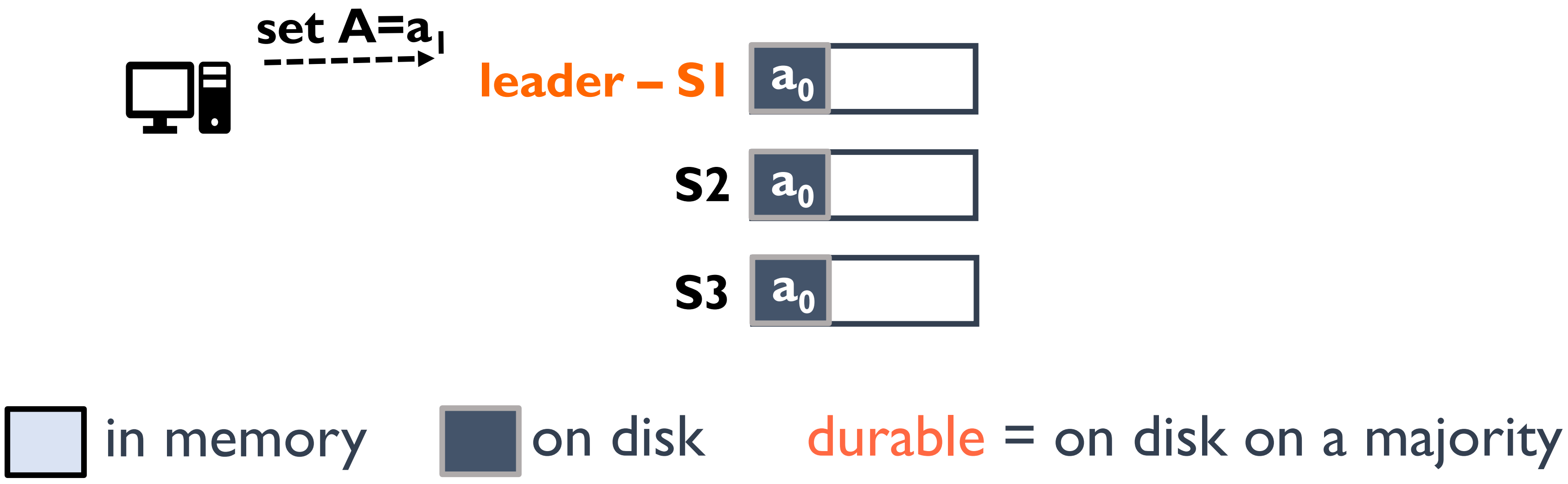
must synchronously replicate and persist data on a majority to tolerate failures

durable = on disk on a majority

Realizing Strong Consistency

Linearizability requires immediate durability

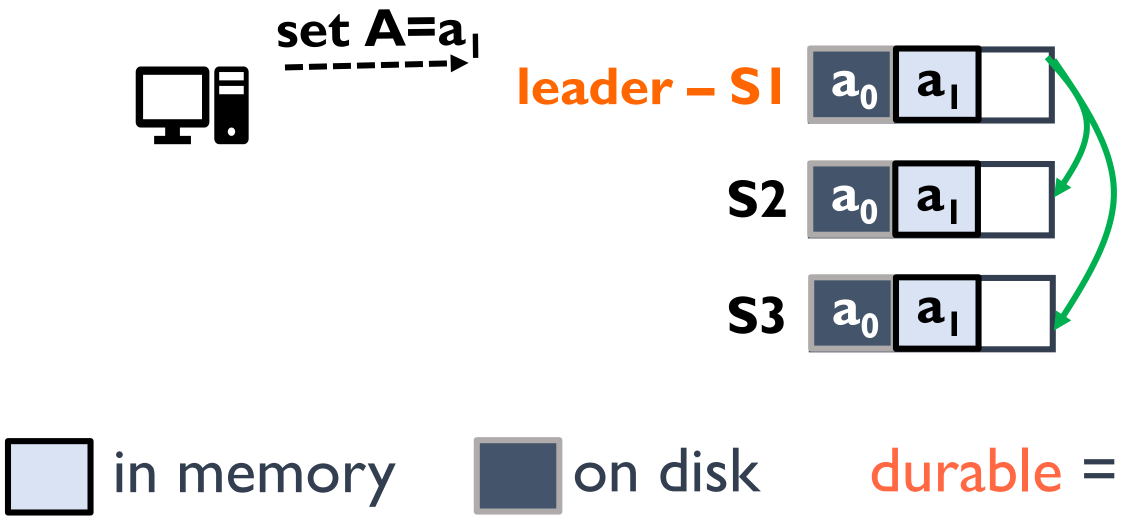
must synchronously replicate and persist data on a majority to tolerate failures



Realizing Strong Consistency

Linearizability requires immediate durability

must synchronously replicate and persist data on a majority to tolerate failures

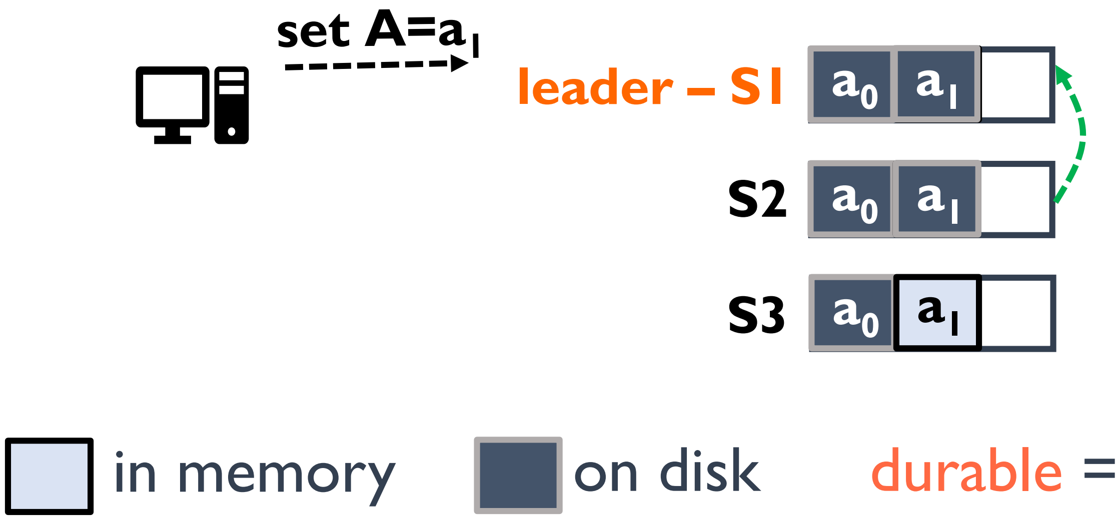


durable = on disk on a majority

Realizing Strong Consistency

Linearizability requires immediate durability

must synchronously replicate and persist data on a majority to tolerate failures

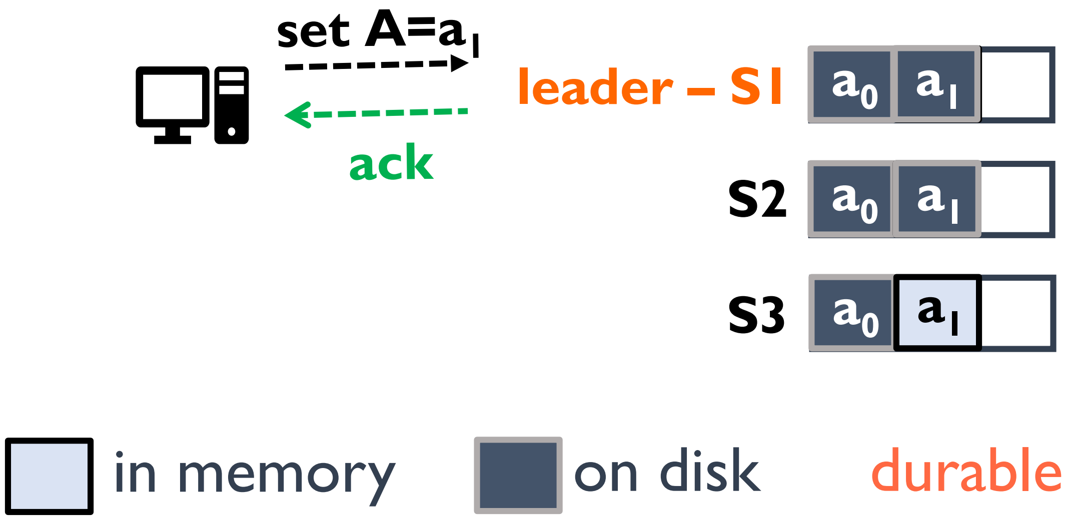


durable = on disk on a majority

Realizing Strong Consistency

Linearizability requires immediate durability

must synchronously replicate and persist data on a majority to tolerate failures

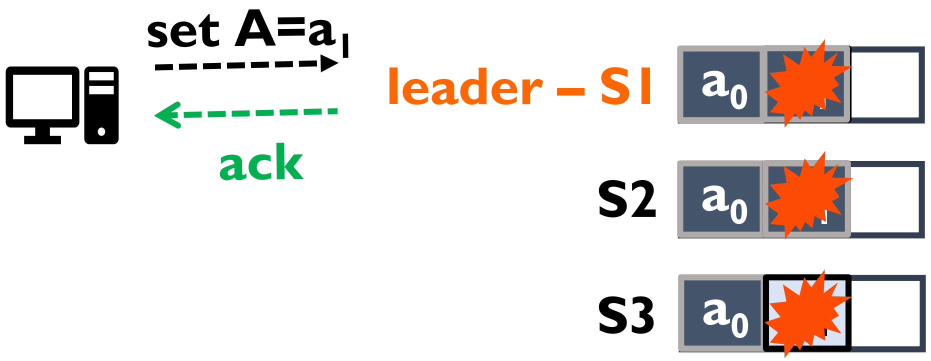


durable = on disk on a majority

Realizing Strong Consistency

Linearizability requires immediate durability

must synchronously replicate and persist data on a majority to tolerate failures

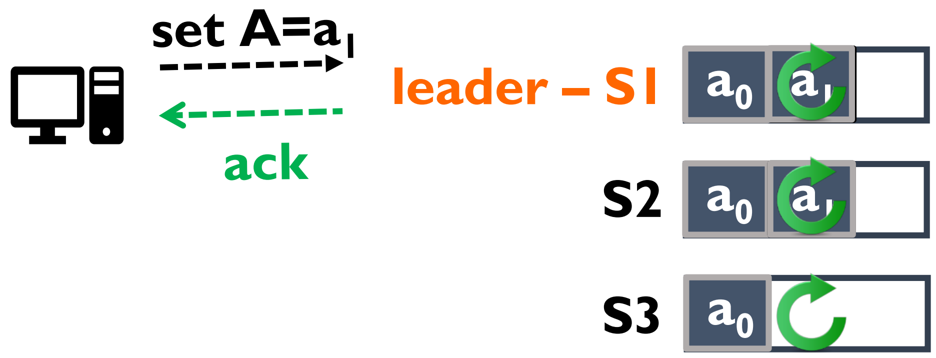


in memory   on disk   durable = on disk on a majority

Realizing Strong Consistency

Linearizability requires immediate durability

must synchronously replicate and persist data on a majority to tolerate failures

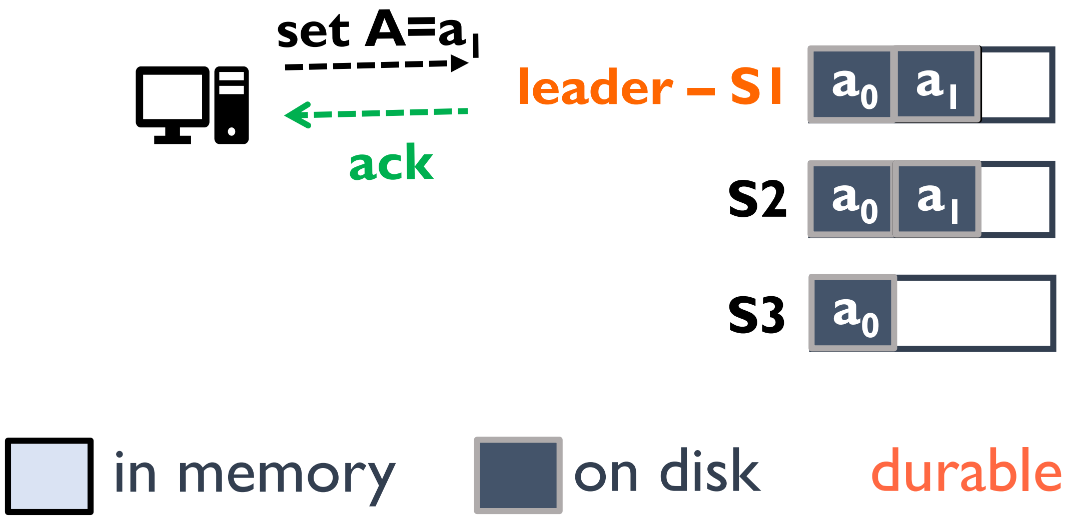


in memory on disk durable = on disk on a majority

Realizing Strong Consistency

Linearizability requires immediate durability

must synchronously replicate and persist data on a majority to tolerate failures

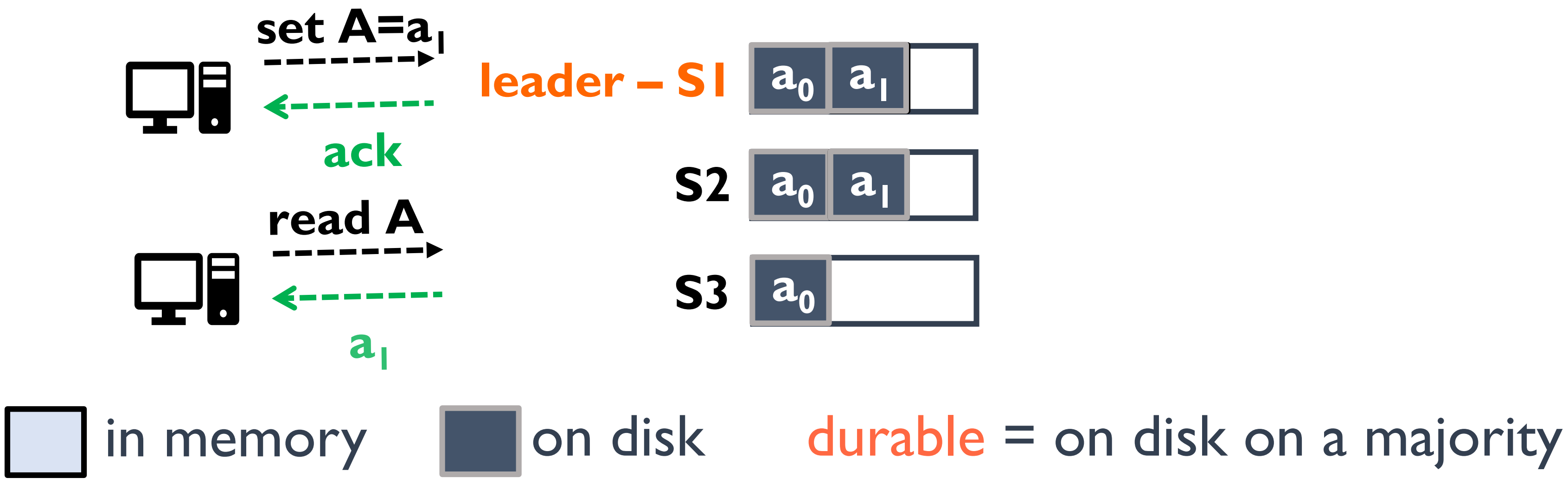


durable = on disk on a majority

Realizing Strong Consistency

Linearizability requires immediate durability

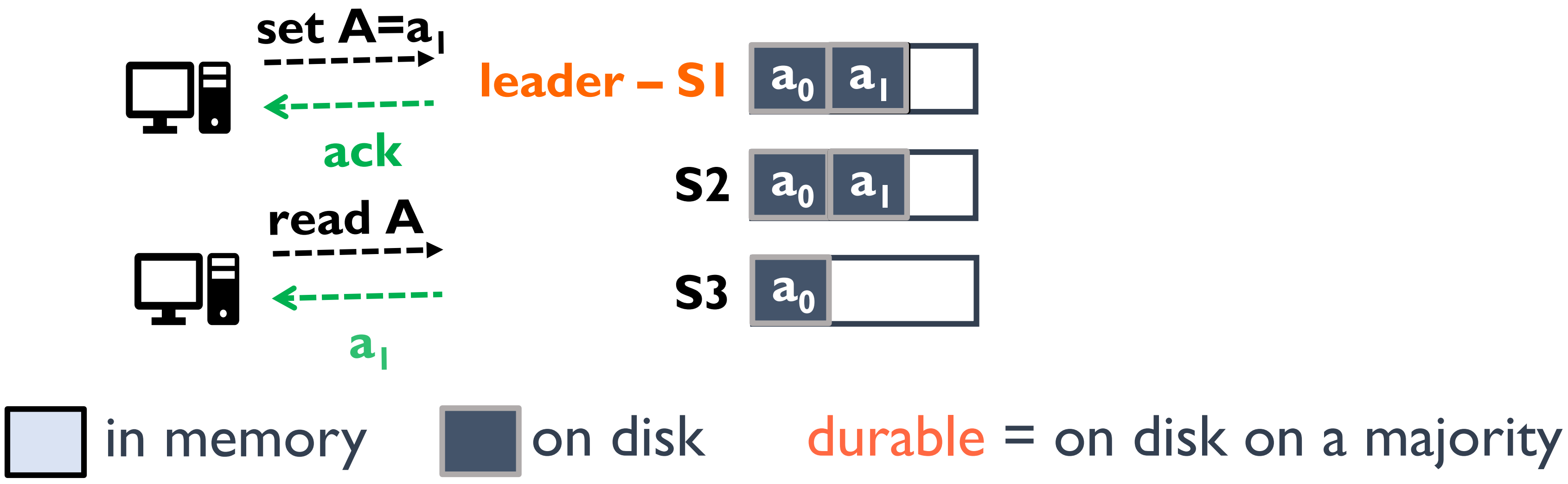
must synchronously replicate and persist data on a majority to tolerate failures



Realizing Strong Consistency

Linearizability requires immediate durability

must synchronously replicate and persist data on a majority to tolerate failures



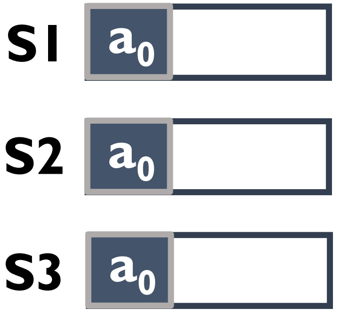
Poor performance due to synchronous operations

10x slower within data center

Realizing Weaker Models

Weaker models only require eventual durability

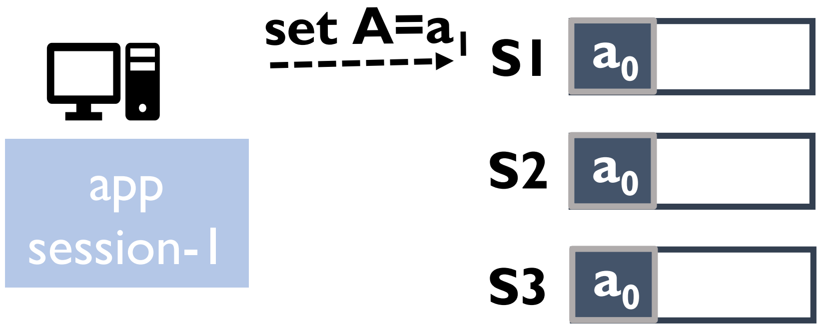
data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

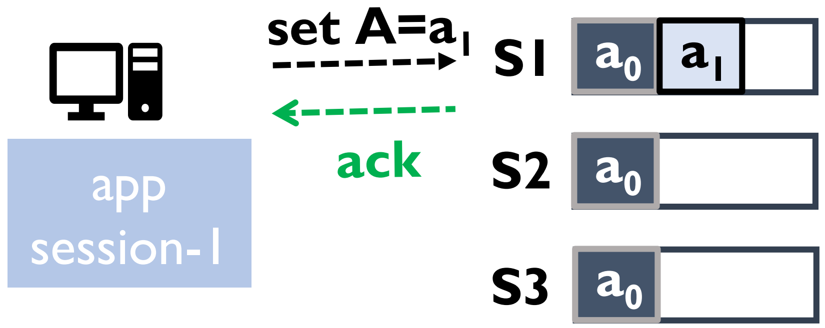
data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

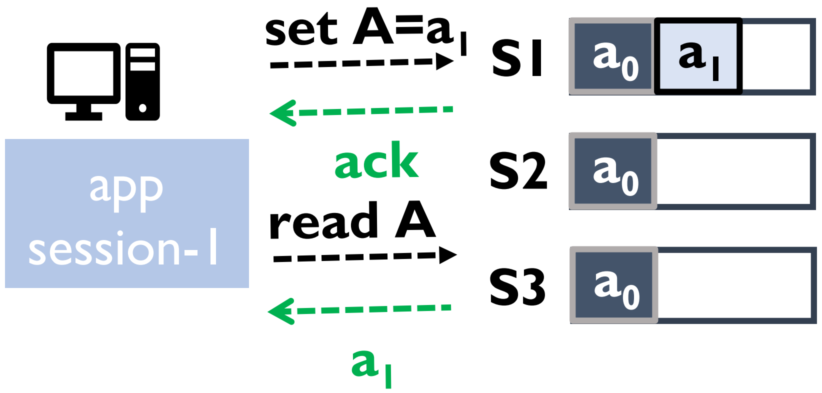
- data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

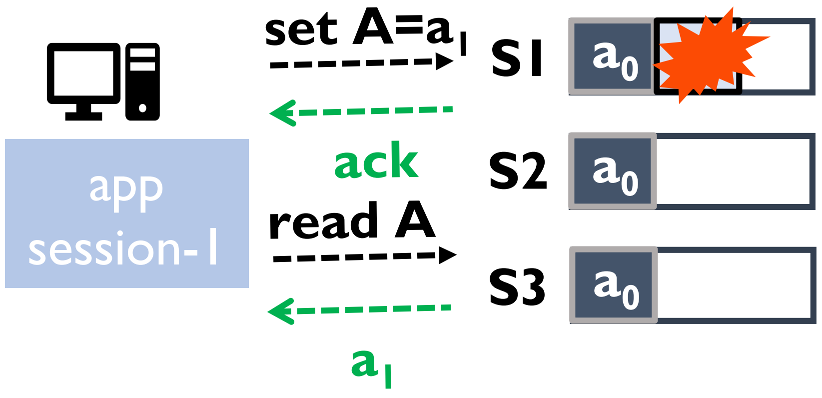
data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

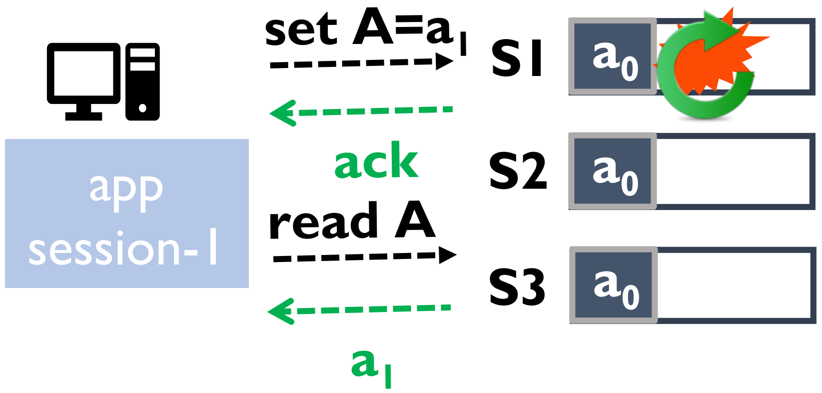
- data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

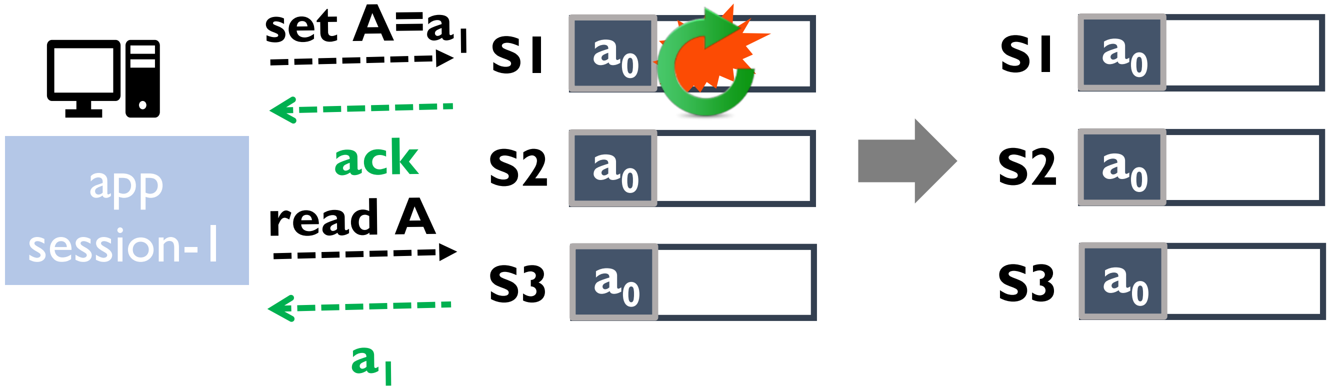
- data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

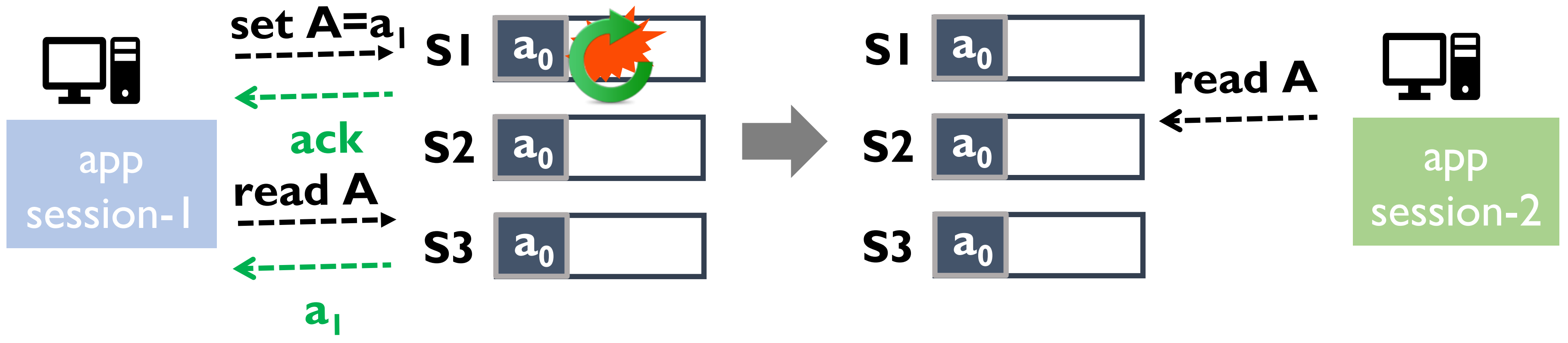
data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

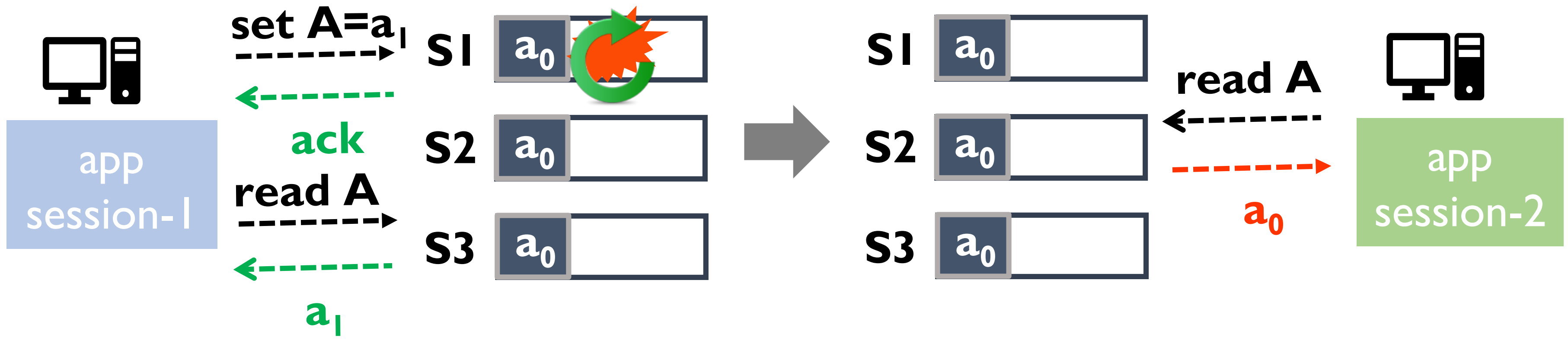
- data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

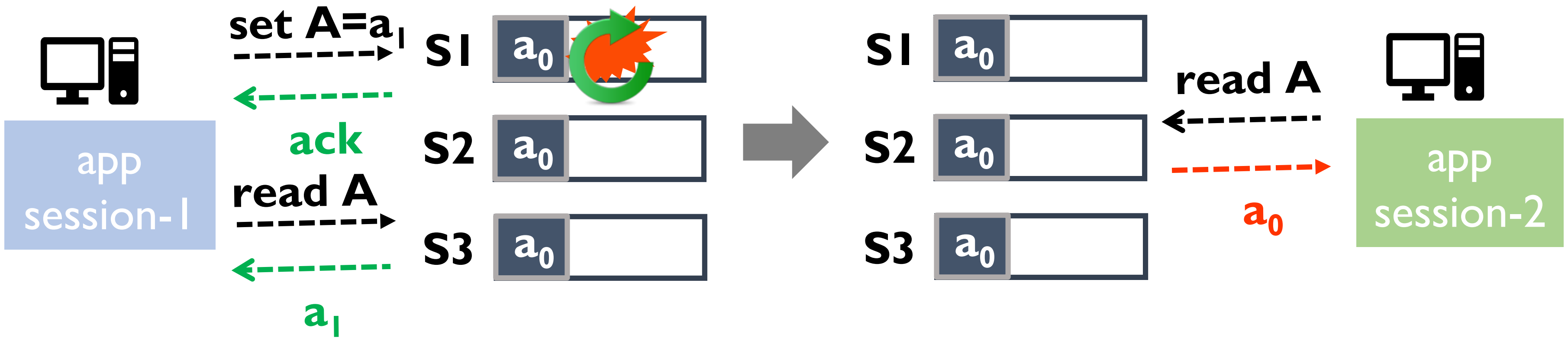
data buffered on one node, replication and persistence in background



Realizing Weaker Models

Weaker models only require eventual durability

data buffered on one node, replication and persistence in background



out-of-order across clients

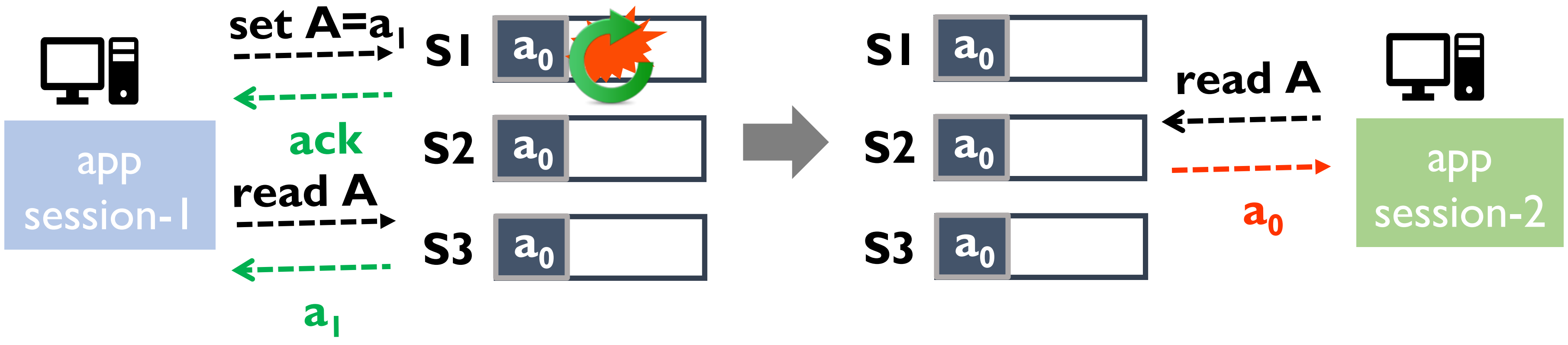
valid under causal and monotonic reads

but confusing semantics

Realizing Weaker Models

Weaker models only require eventual durability

data buffered on one node, replication and persistence in background



out-of-order across clients

valid under causal and monotonic reads

but confusing semantics

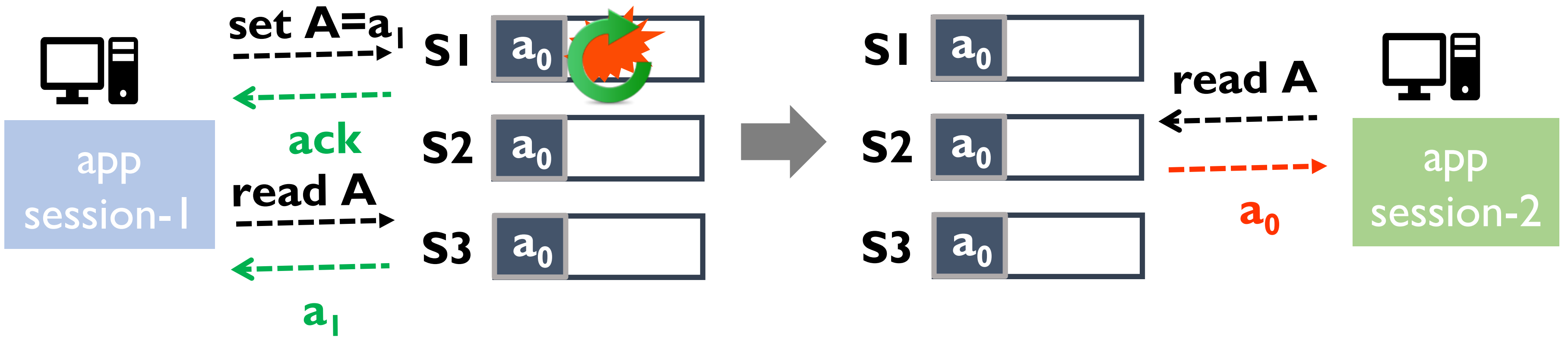
Many deployments prefer eventual durability for performance

in fact, it is the default (e.g., MongoDB, Redis)

Realizing Weaker Models

Weaker models only require eventual durability

data buffered on one node, replication and persistence in background



out-of-order across clients

valid under causal and monotonic reads

but confusing semantics

Many deployments prefer eventual durability for performance

in fact, it is the default (e.g., MongoDB, Redis)

Thus settle for weak consistency

Realizing Weaker Models

Weaker models only require eventual durability

data buffered on one node, replication and persistence in background

Immediate durability enables strong consistency but is slow

Eventual durability is fast but enables only weaker consistency

Many deployments prefer eventual durability for performance

in fact, it is the default (e.g., MongoDB, Redis)

Thus settle for weak consistency

Outline

Introduction

Motivation

CAD and cross-client monotonic reads

ORCA design

Results

Summary and conclusion

Consistency-aware Durability

Consistency-aware Durability

Most consistency models care about what reads see

Consistency-aware Durability

Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

Consistency-aware Durability

Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

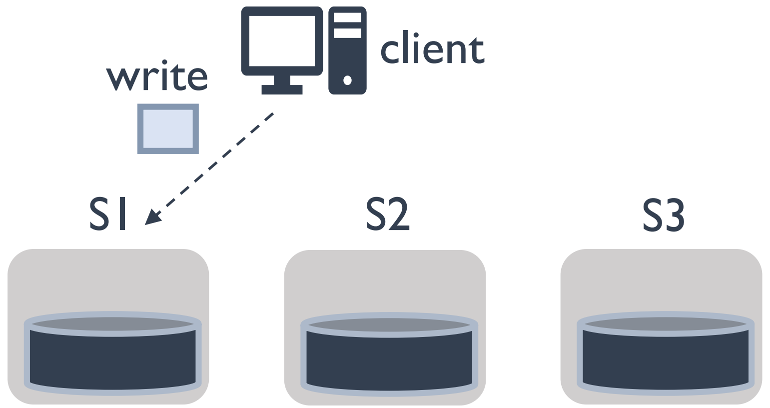
delay durability of writes

Consistency-aware Durability

Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes

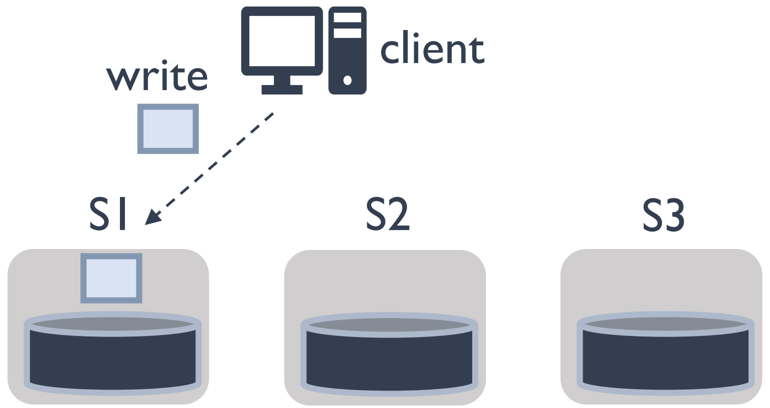


Consistency-aware Durability

Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes

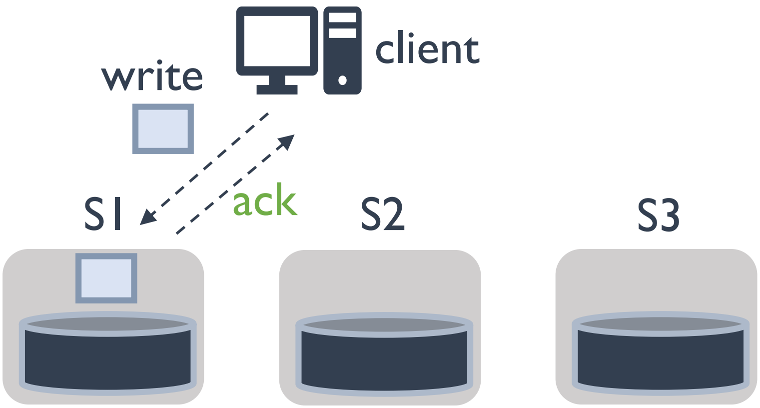


Consistency-aware Durability

Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes

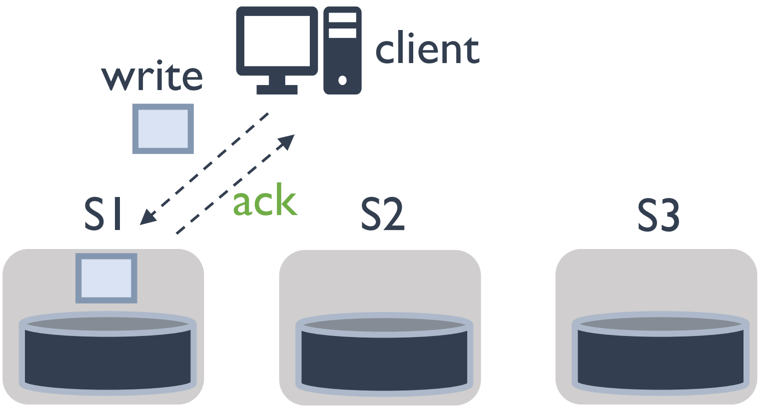


Consistency-aware Durability

Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes



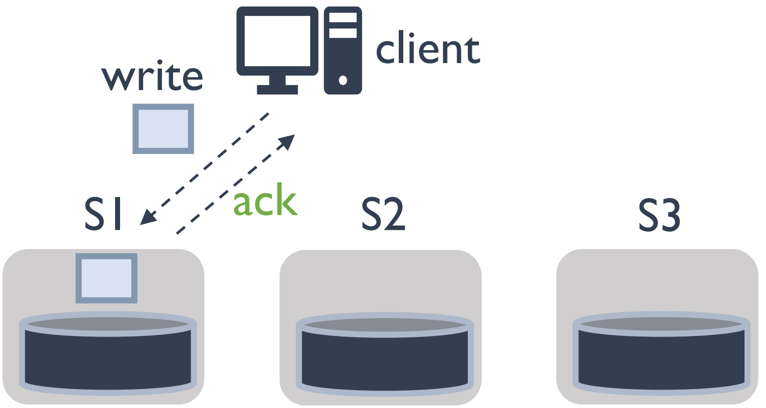
good performance

Consistency-aware Durability

Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes	make data durable before serving reads



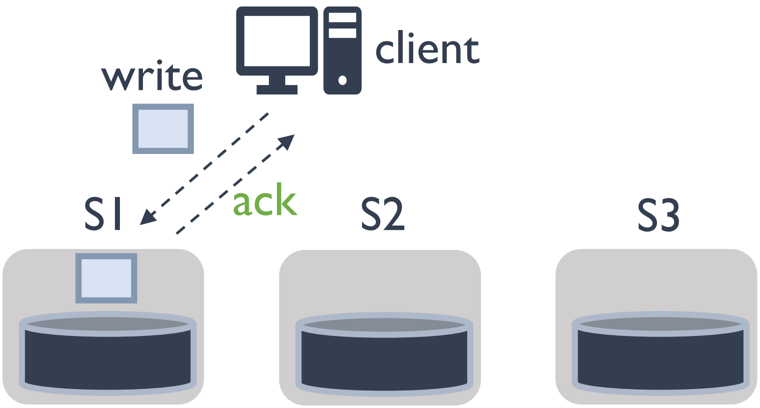
good performance

Consistency-aware Durability

Most consistency models care about what reads see

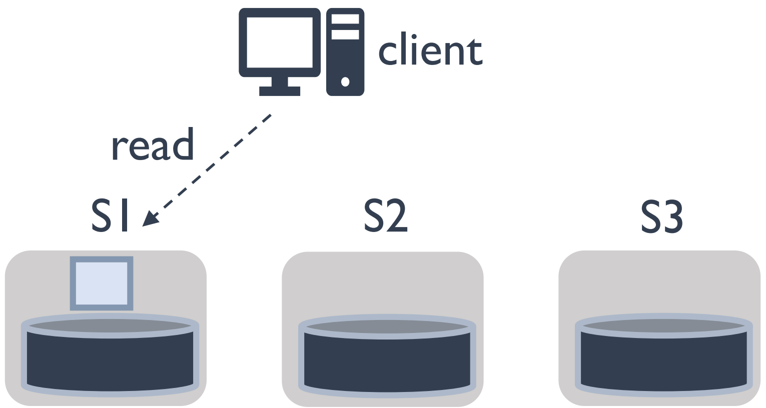
Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes



good performance

make data durable before serving reads

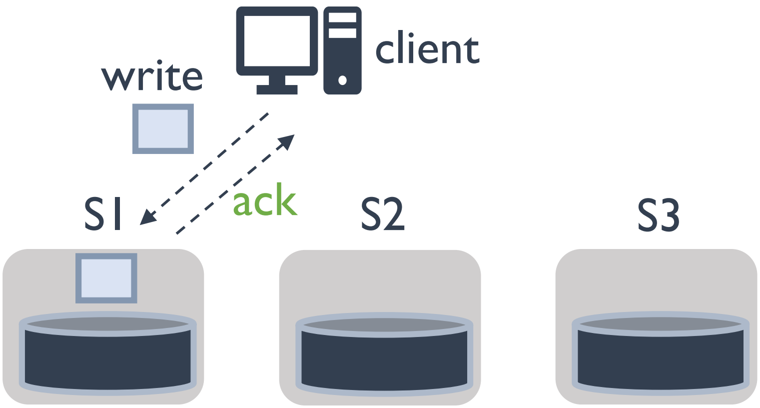


Consistency-aware Durability

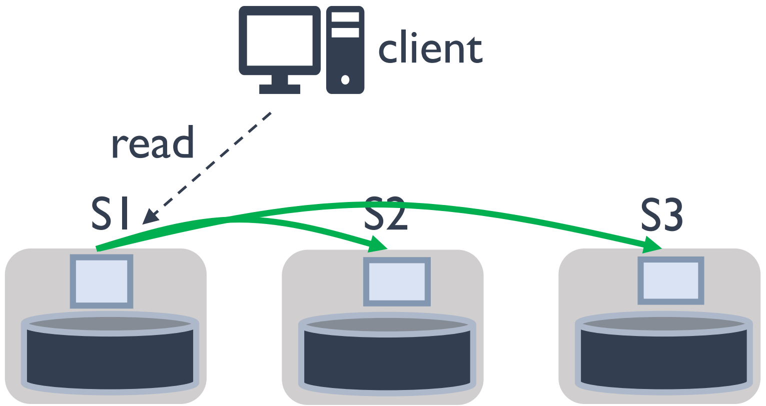
Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes	make data durable before serving reads



good performance

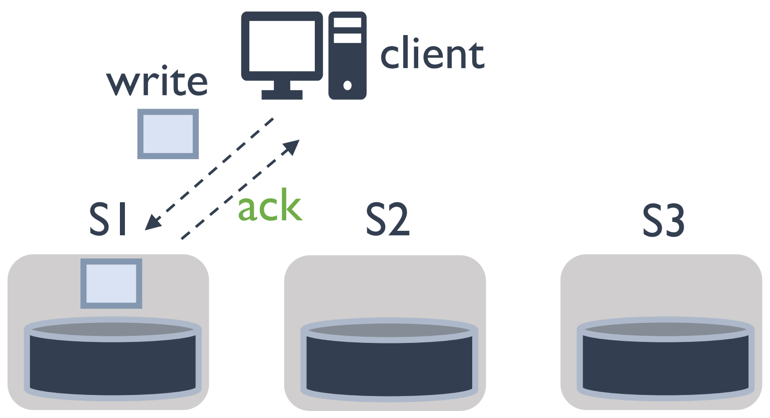


Consistency-aware Durability

Most consistency models care about what reads see

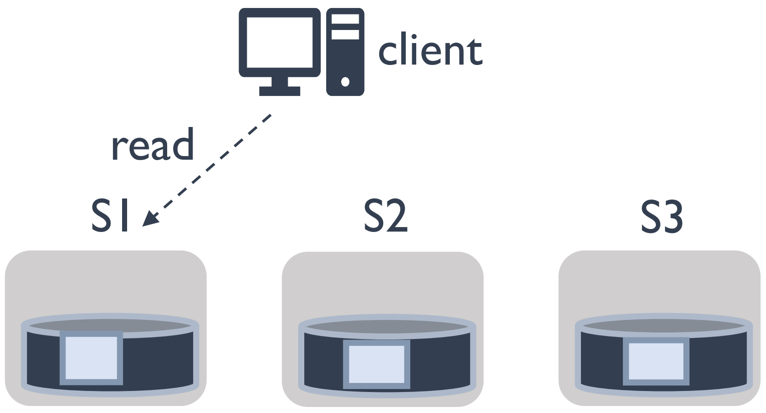
Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes



good performance

make data durable before serving reads

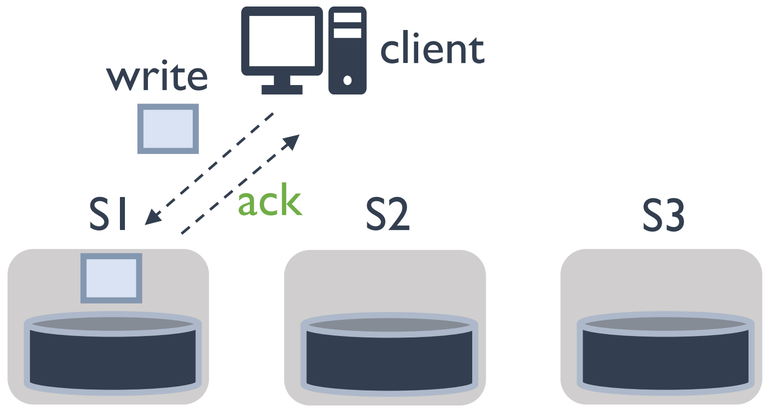


Consistency-aware Durability

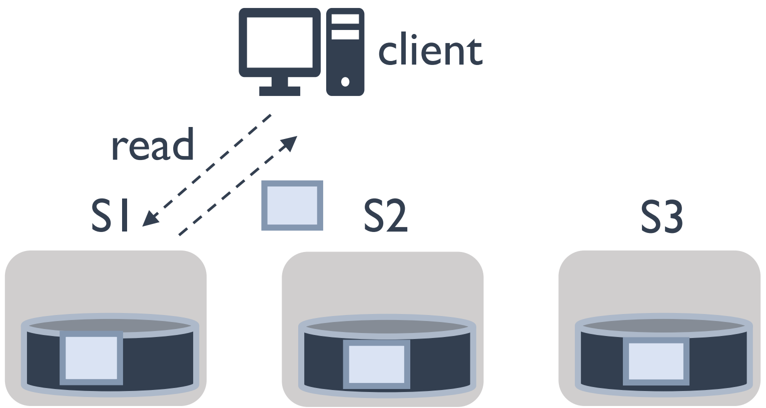
Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes	make data durable before serving reads



good performance

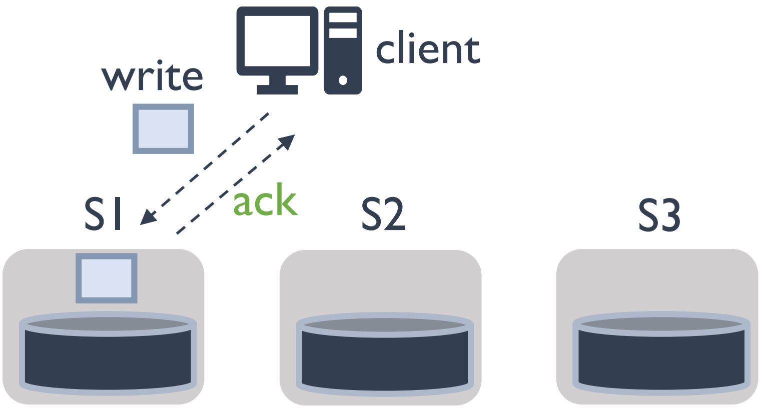


Consistency-aware Durability

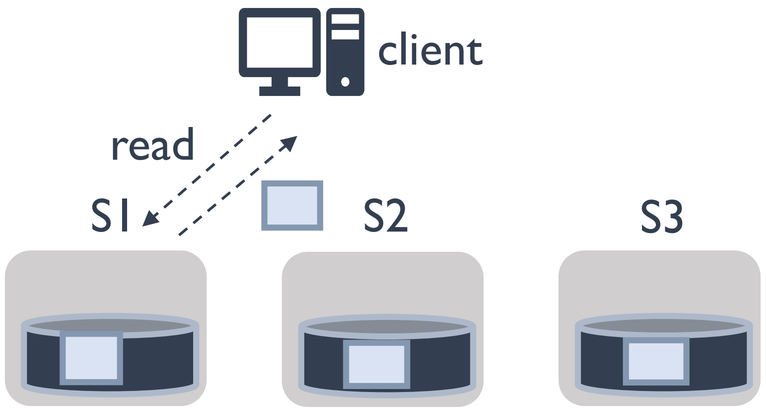
Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes	make data durable before serving reads



good performance



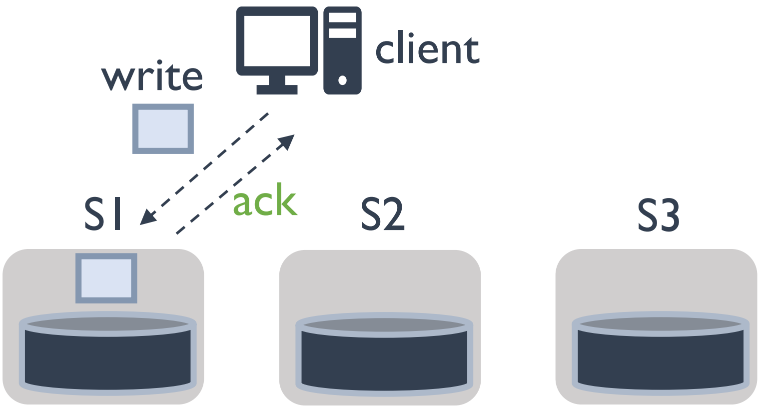
prevents out-of-order data across failures
strong consistency

Consistency-aware Durability

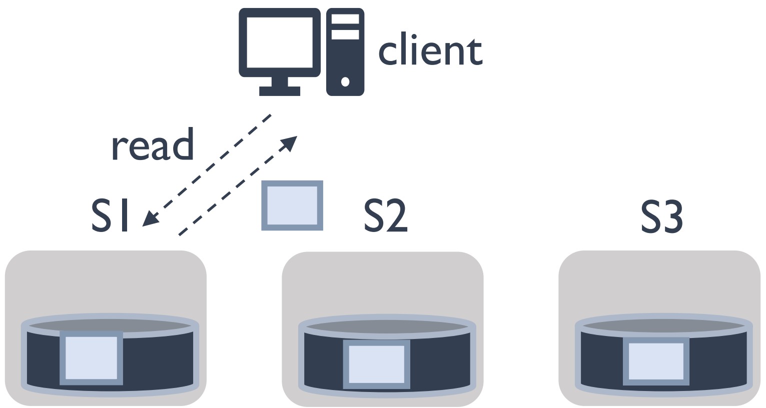
Most consistency models care about what reads see

Key idea: CAD shifts the point of durability to reads from writes

delay durability of writes	make data durable before serving reads



good performance



prevents out-of-order data across failures
strong consistency

CAD does not always incur overheads on reads

- reads do not immediately follow writes – natural in many workloads
- common case: data already durable well before applications access it

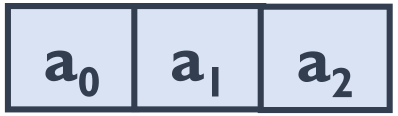
Cross-client Monotonic Reads upon CAD

Cross-client Monotonic Reads upon CAD

A read from a client guaranteed to return at least the latest state returned to a previous read from any client

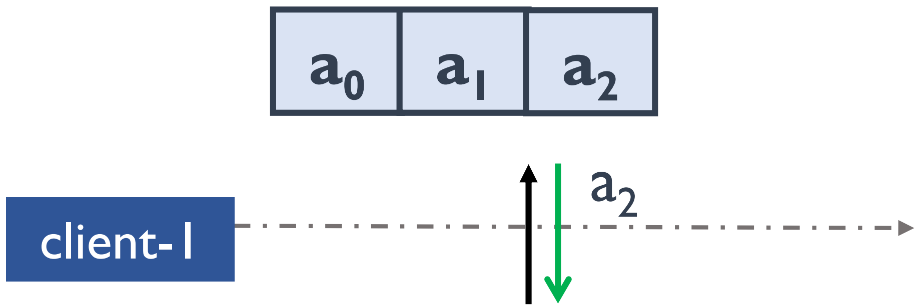
Cross-client Monotonic Reads upon CAD

A read from a client guaranteed to return at least the latest state returned to a previous read from any client



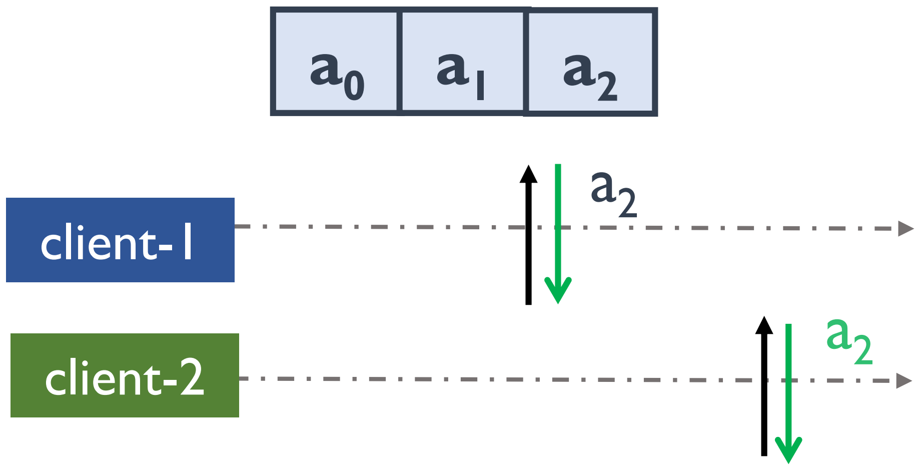
Cross-client Monotonic Reads upon CAD

A read from a client guaranteed to return at least the latest state returned to a previous read from any client



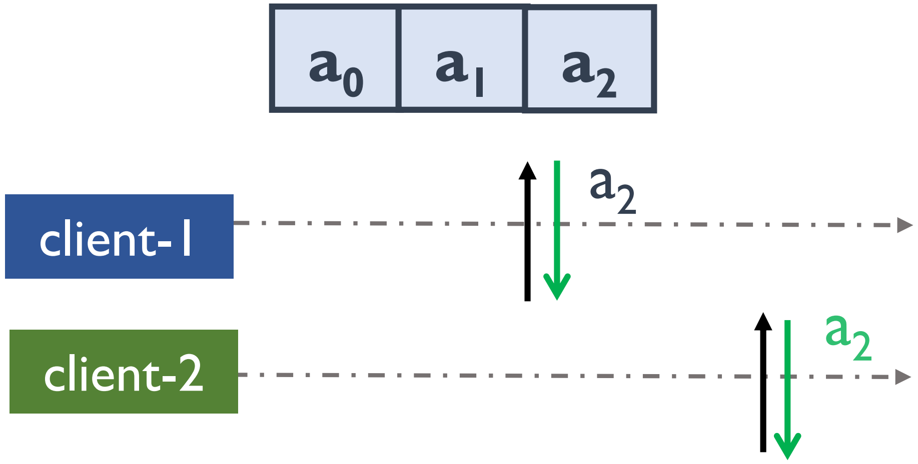
Cross-client Monotonic Reads upon CAD

A read from a client guaranteed to return at least the latest state returned to a previous read from any client



Cross-client Monotonic Reads upon CAD

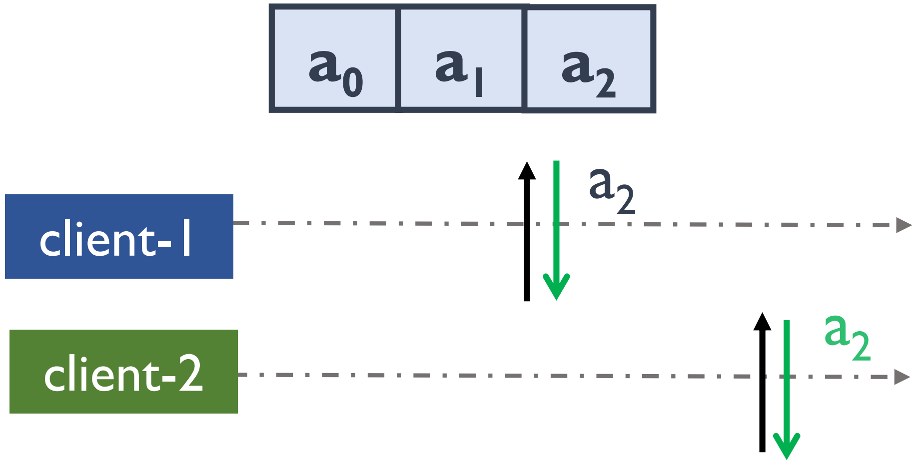
A read from a client guaranteed to return at least the latest state returned to a previous read from any client



Even in the presence of failures and across client sessions

Cross-client Monotonic Reads upon CAD

A read from a client guaranteed to return at least the latest state returned to a previous read from any client

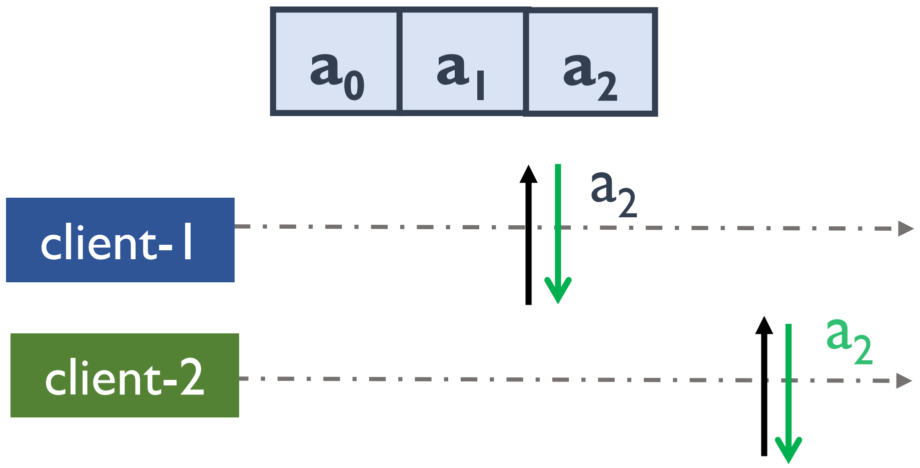


Even in the presence of failures and across client sessions

No existing model provides this guarantee except linearizability but not with high performance

Cross-client Monotonic Reads upon CAD

A read from a client guaranteed to return at least the latest state returned to a previous read from any client



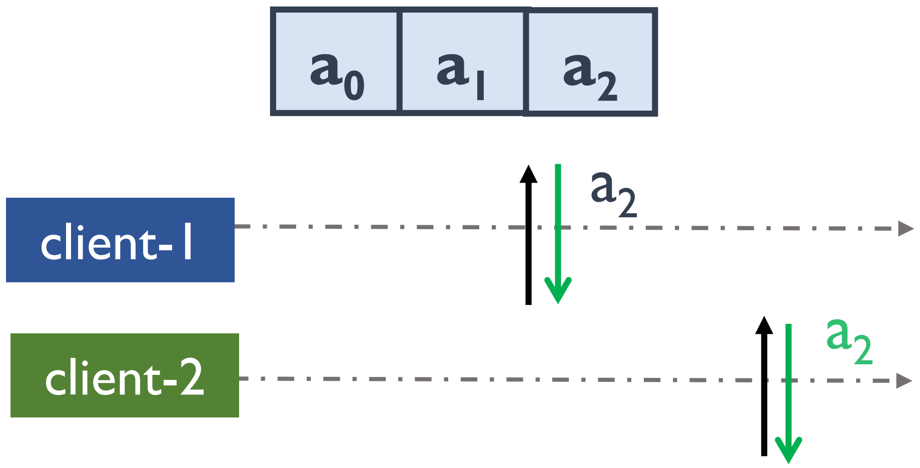
Even in the presence of failures and across client sessions

No existing model provides this guarantee except linearizability but not with high performance

CAD enables this property with high performance

Cross-client Monotonic Reads upon CAD

A read from a client guaranteed to return at least the latest state returned to a previous read from any client



Even in the presence of failures and across client sessions

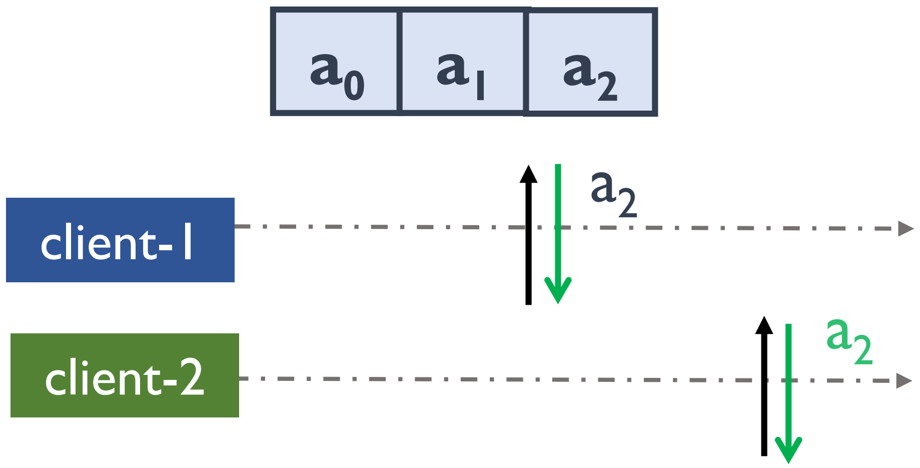
No existing model provides this guarantee except linearizability but not with high performance

CAD enables this property with high performance

Does not prevent staleness like many weaker models

Cross-client Monotonic Reads upon CAD

A read from a client guaranteed to return at least the latest state returned to a previous read from any client



Even in the presence of failures and across client sessions

No existing model provides this guarantee except linearizability but not with high performance

CAD enables this property with high performance

Does not prevent staleness like many weaker models

However, avoids out-of-order data, useful in many app scenarios

- e.g., location-sharing, twitter timelines

Outline

Introduction

Motivation

CAD and cross-client monotonic reads

ORCA design

Results

Summary and conclusion

ORCA

ORCA

Implementation of consistency-aware durability and cross-client monotonic reads in leader-based majority systems

ORCA

Implementation of consistency-aware durability and cross-client monotonic reads in leader-based majority systems

Leader-based systems (e.g., MongoDB, ZooKeeper)

- leader – a dedicated node
- others are followers
- writes flow through leader, establishes a single order

ORCA

Implementation of consistency-aware durability and cross-client monotonic reads in leader-based majority systems

Leader-based systems (e.g., MongoDB, ZooKeeper)

- leader – a dedicated node
- others are followers
- writes flow through leader, establishes a single order

Majority

- data is safe when persisted on majority nodes (e.g., 3 out of 5 servers)

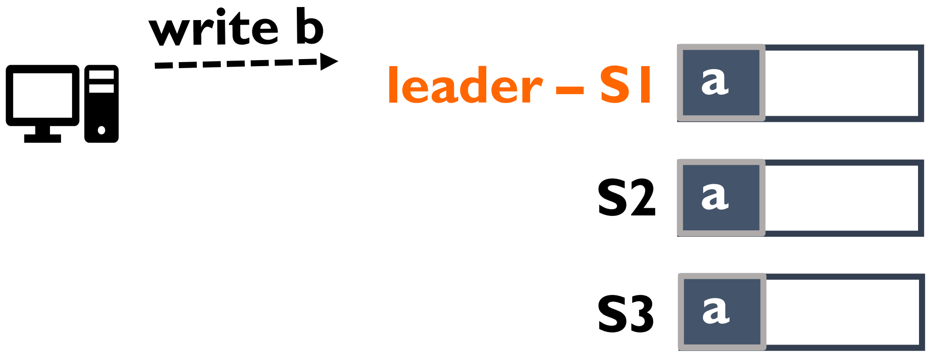
ORCA Write Path

ORCA Write Path

Same as an eventually durable system

ORCA Write Path

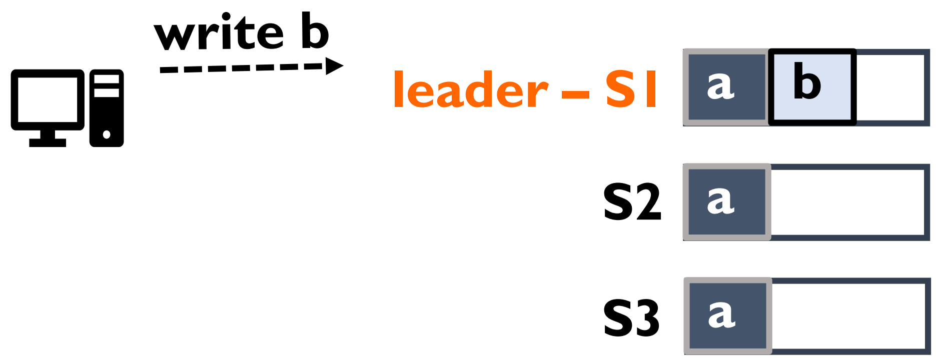
Same as an eventually durable system



in memory a_0 on disk durable = on disk on a majority

ORCA Write Path

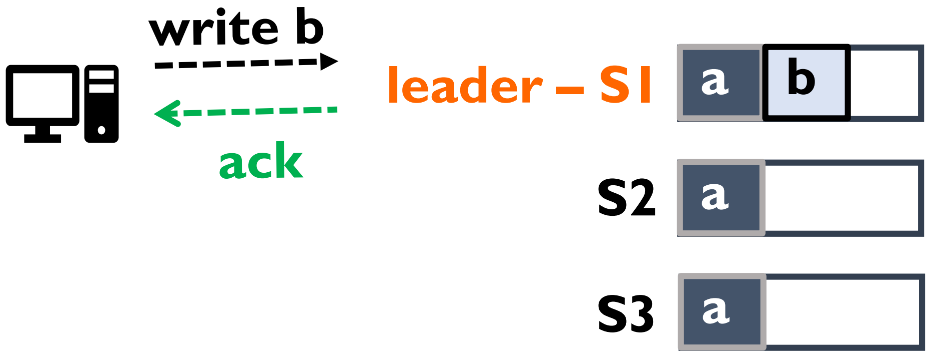
Same as an eventually durable system



in memory a_0 on disk durable = on disk on a majority

ORCA Write Path

Same as an eventually durable system



immediately acknowledge writes → high performance

durable = on disk on a majority

ORCA Write Path

Same as an eventually durable system

immediately acknowledge writes → high performance

replication and persistence in background

in memory a_0 on disk   durable = on disk on a majority

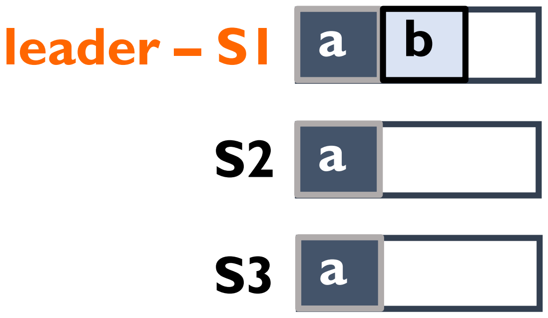
ORCA Write Path

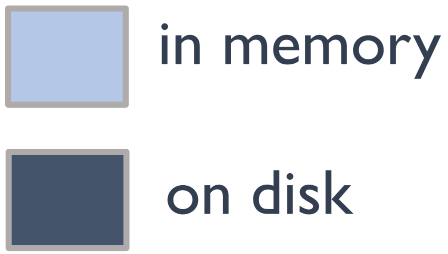
Same as an eventually durable system

immediately acknowledge writes → high performance

replication and persistence in background

ORCA Read Path





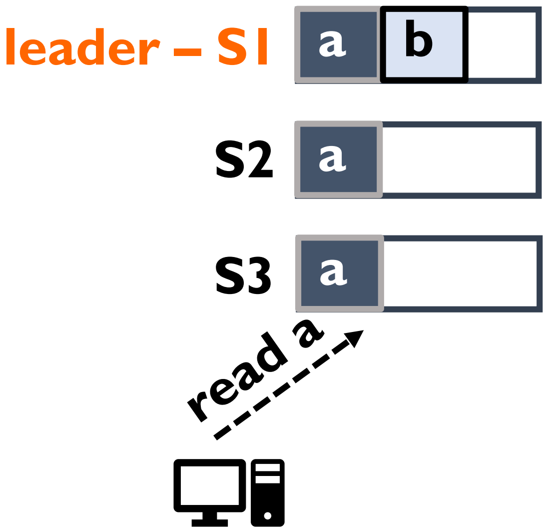
durable = on disk on a majority

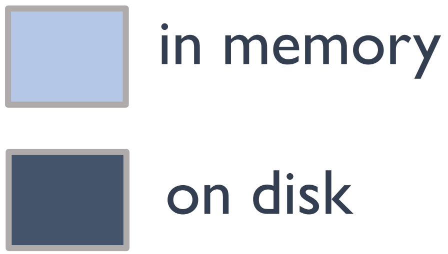
ORCA Read Path

durable = on disk on a majority

ORCA Read Path

Durable-index – index of the latest durable item in the system





durable = on disk on a majority

ORCA Read Path

Durable-index – index of the latest durable item in the system

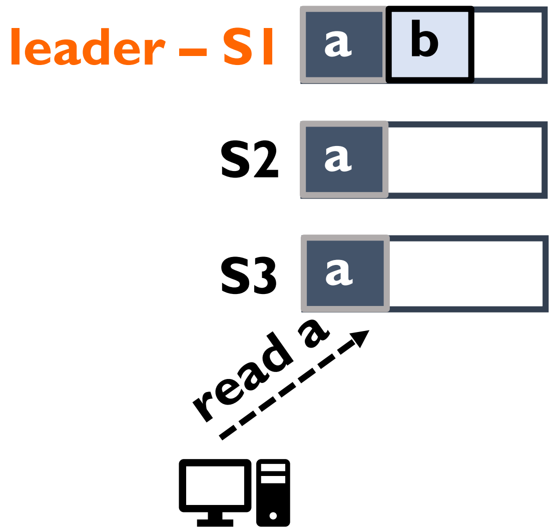
Update-index of item *i* -- index of the last update that modified *i*

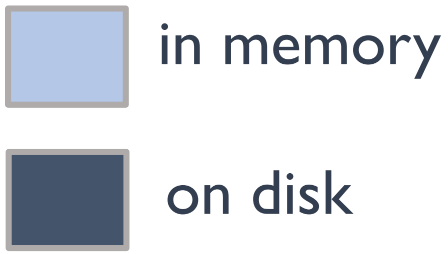
ORCA Read Path

Durable-index – index of the latest durable item in the system

Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system



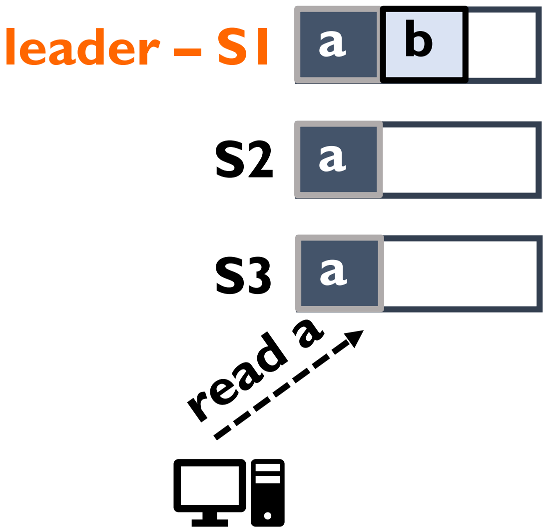


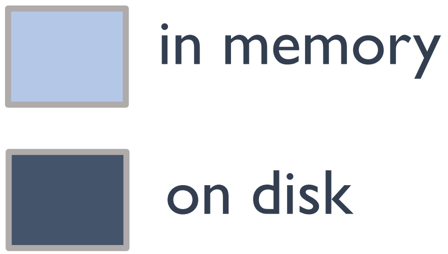
durable = on disk on a majority

ORCA Read Path

Durable-index – index of the latest durable item in the system
Update-index of item *i* -- index of the last update that modified *i*
Durability check – *i* durable if update-index of *i* ≤ durable-index of system

durable-index: 1
a's update-index: 1
a is durable





durable = on disk on a majority

ORCA Read Path

Durable-index – index of the latest durable item in the system

Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system

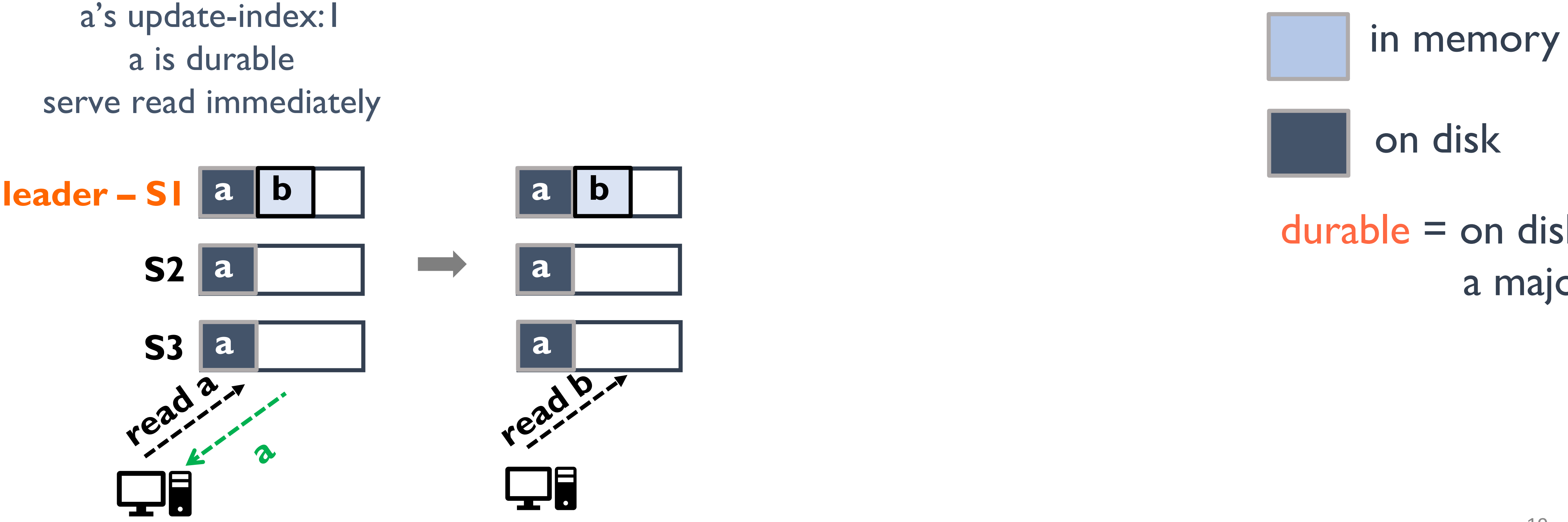
ORCA Read Path

Durable-index – index of the latest durable item in the system

Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system

durable-index: 1
a's update-index: 1
a is durable
serve read immediately



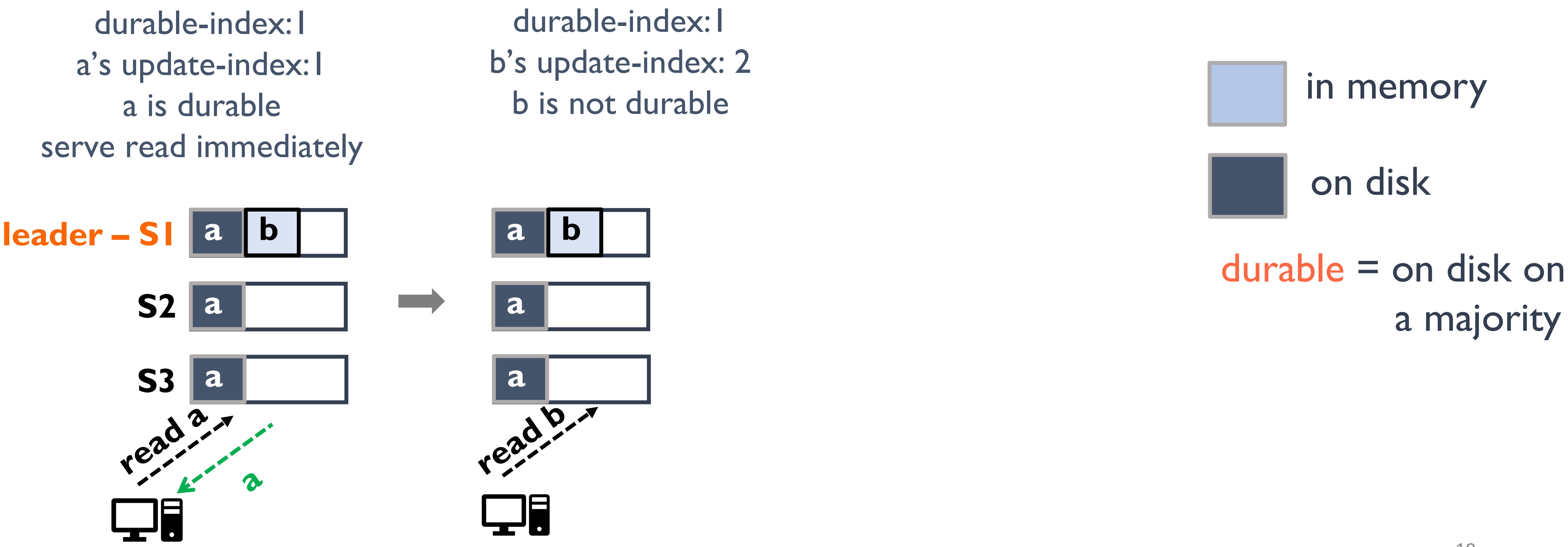
durable = on disk on a majority

ORCA Read Path

Durable-index – index of the latest durable item in the system

Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system

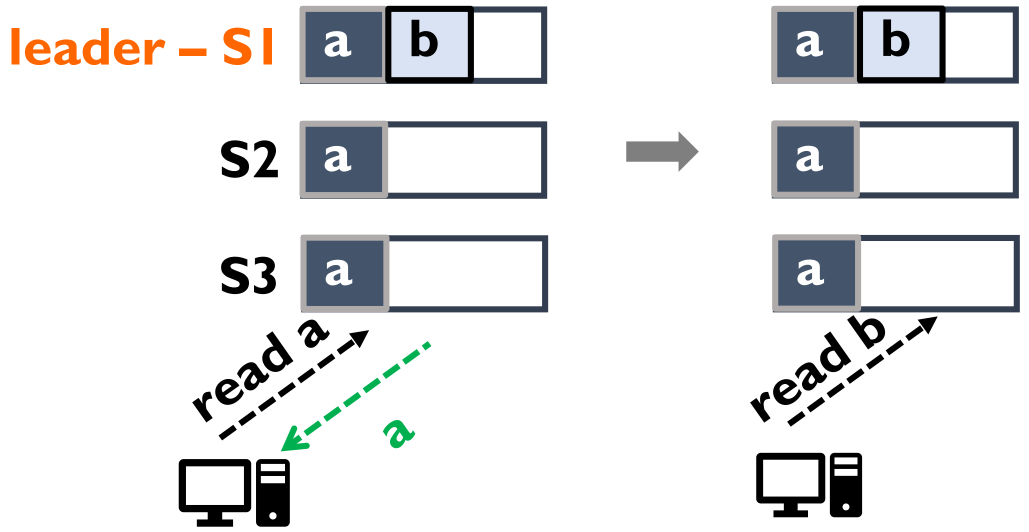


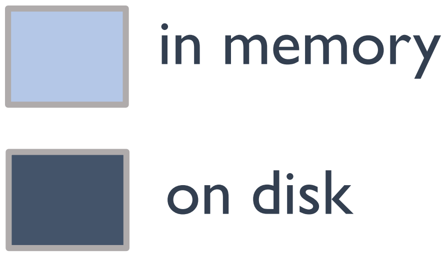
ORCA Read Path

Durable-index – index of the latest durable item in the system

Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system





durable = on disk on a majority

ORCA Read Path

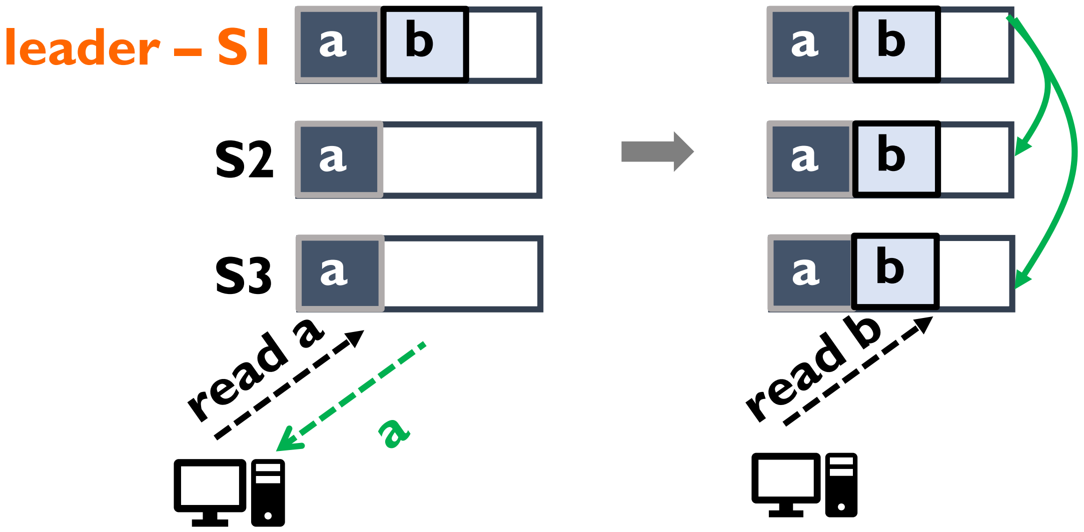
Durable-index – index of the latest durable item in the system

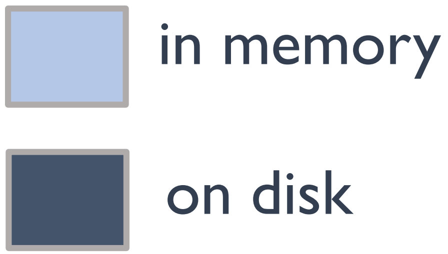
Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system

durable-index: 1
a's update-index: 1
a is durable
serve read immediately

durable-index: 1
b's update-index: 2
b is not durable
make b durable before serving





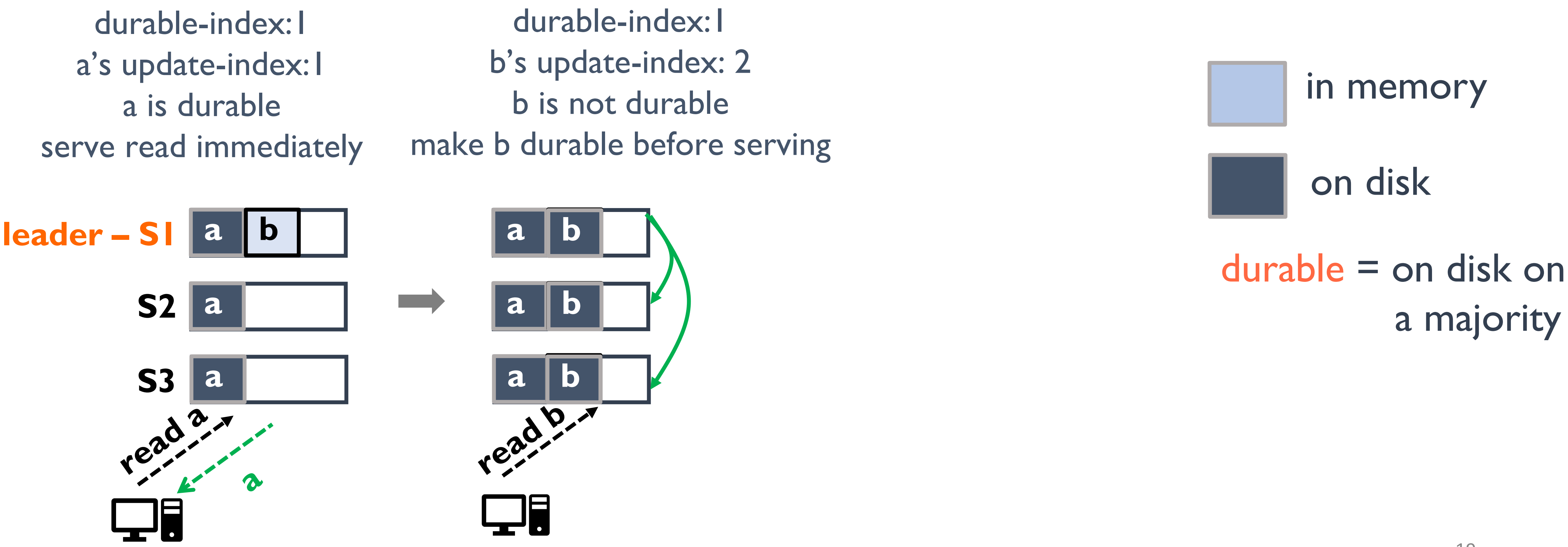
durable = on disk on a majority

ORCA Read Path

Durable-index – index of the latest durable item in the system

Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system



ORCA Read Path

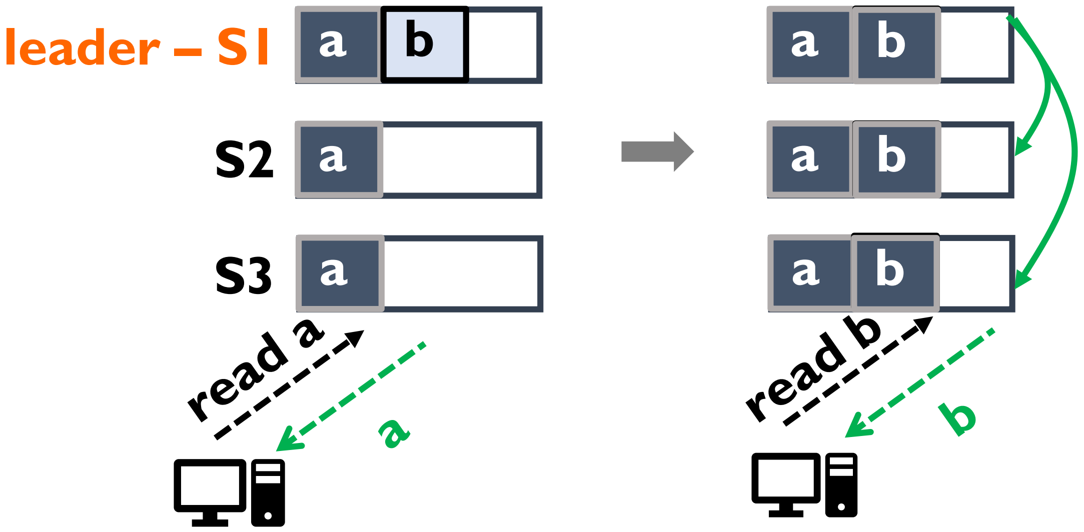
Durable-index – index of the latest durable item in the system

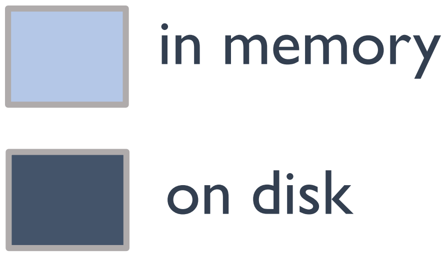
Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system

durable-index: 1
a's update-index: 1
a is durable
serve read immediately

durable-index: 1
b's update-index: 2
b is not durable
make b durable before serving





durable = on disk on a majority

ORCA Read Path

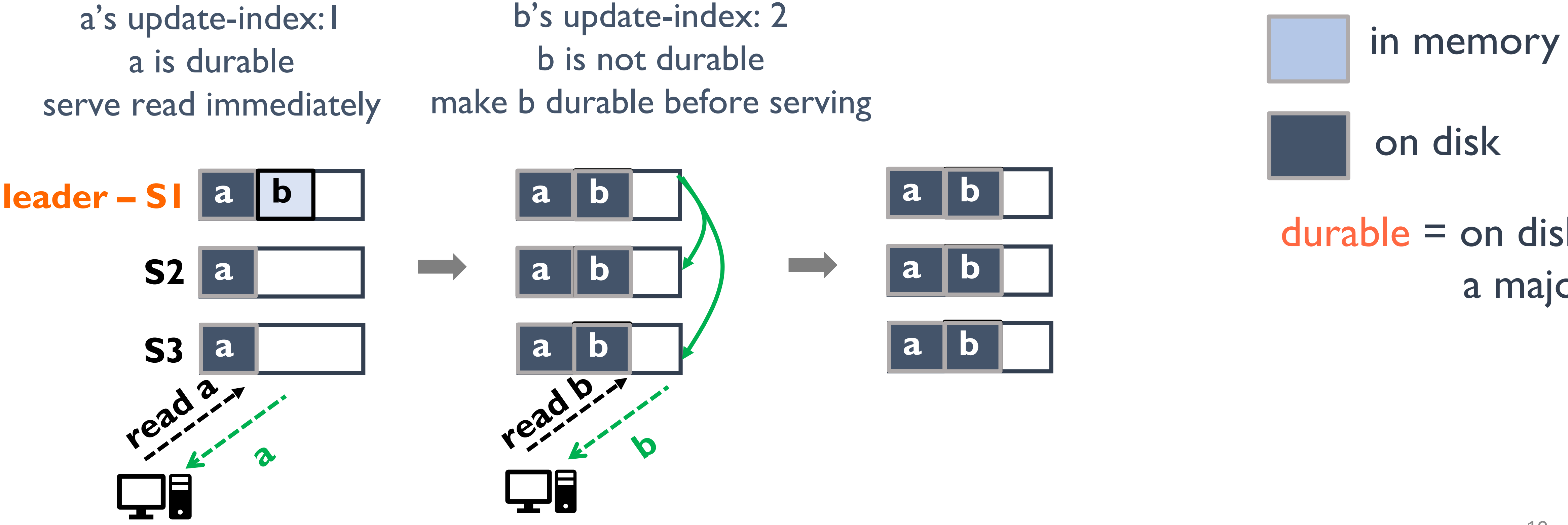
Durable-index – index of the latest durable item in the system

Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system

durable-index: 1
a's update-index: 1
a is durable
serve read immediately

durable-index: 1
b's update-index: 2
b is not durable
make b durable before serving



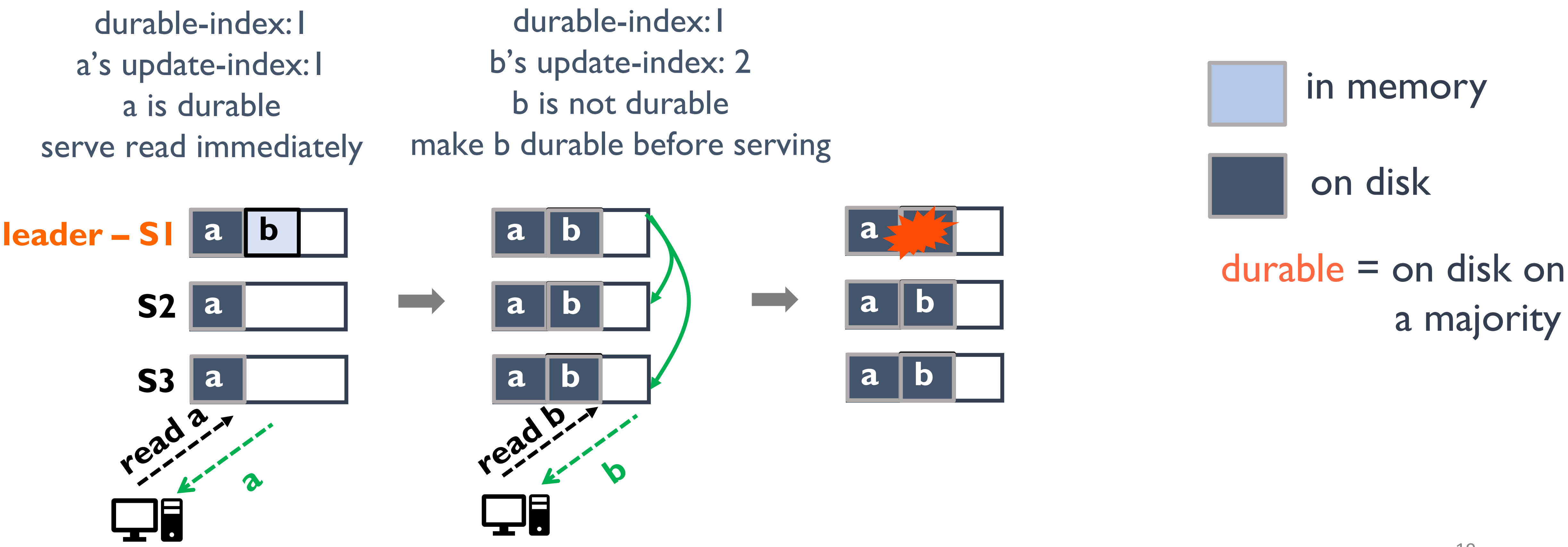
durable = on disk on a majority

ORCA Read Path

Durable-index – index of the latest durable item in the system

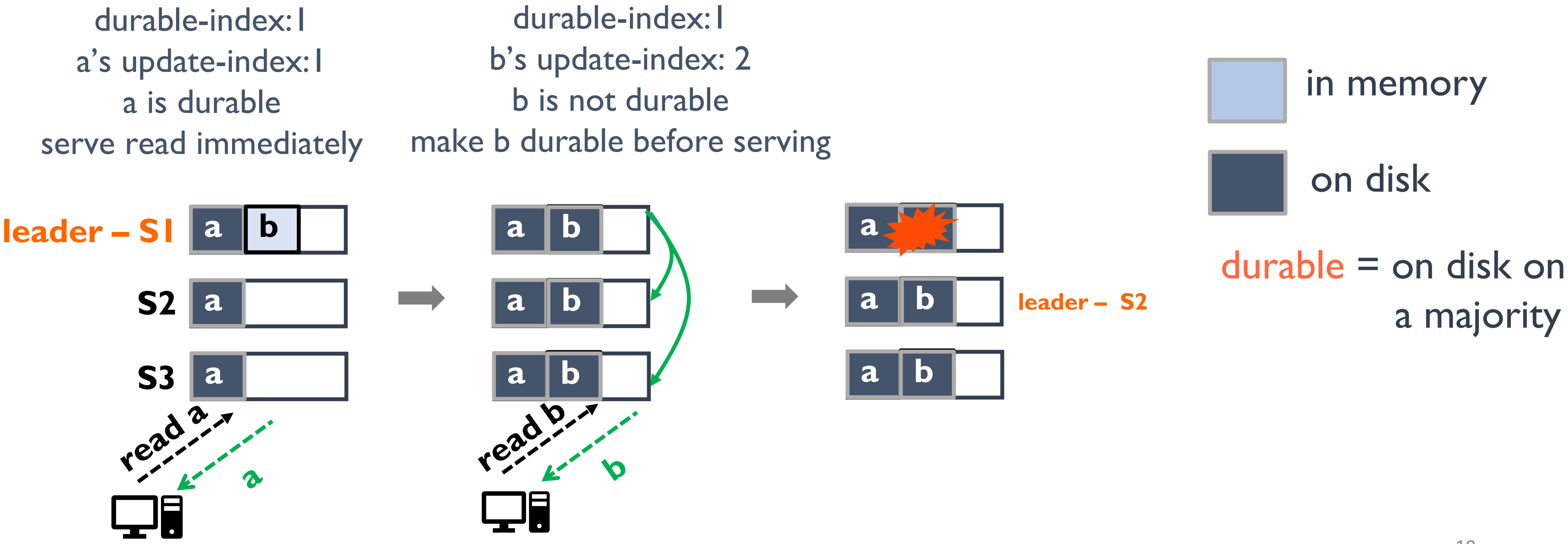
Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system



ORCA Read Path

Durable-index – index of the latest durable item in the system
Update-index of item *i* -- index of the last update that modified *i*
Durability check – *i* durable if update-index of *i* ≤ durable-index of system

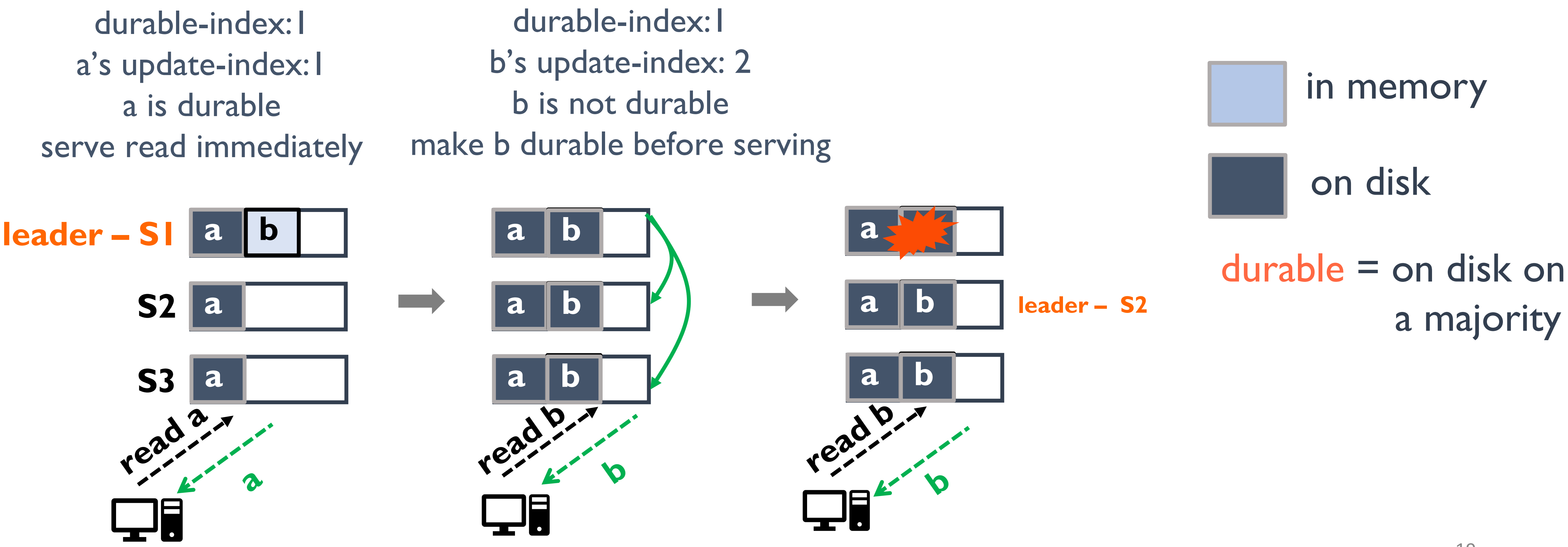


ORCA Read Path

Durable-index – index of the latest durable item in the system

Update-index of item *i* -- index of the last update that modified *i*

Durability check – *i* durable if update-index of *i* ≤ durable-index of system



Cross-Client Monotonic Reads in ORCA

Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

- not scalable

Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

- not scalable

Allow reads at followers

- lagging followers could cause out-of-order states, CAD is not sufficient

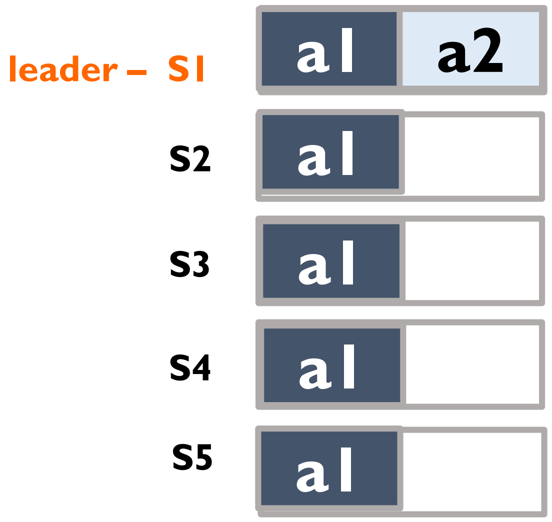
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

- not scalable

Allow reads at followers

- lagging followers could cause out-of-order states, CAD is not sufficient



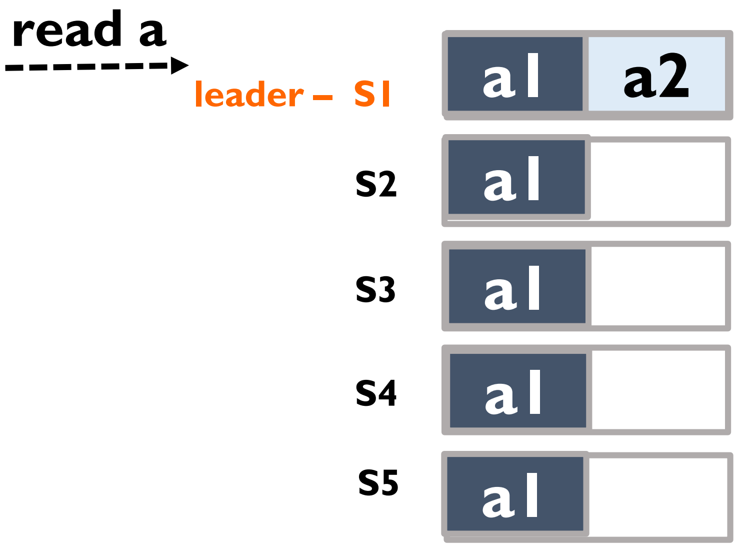
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

- not scalable

Allow reads at followers

- lagging followers could cause out-of-order states, CAD is not sufficient



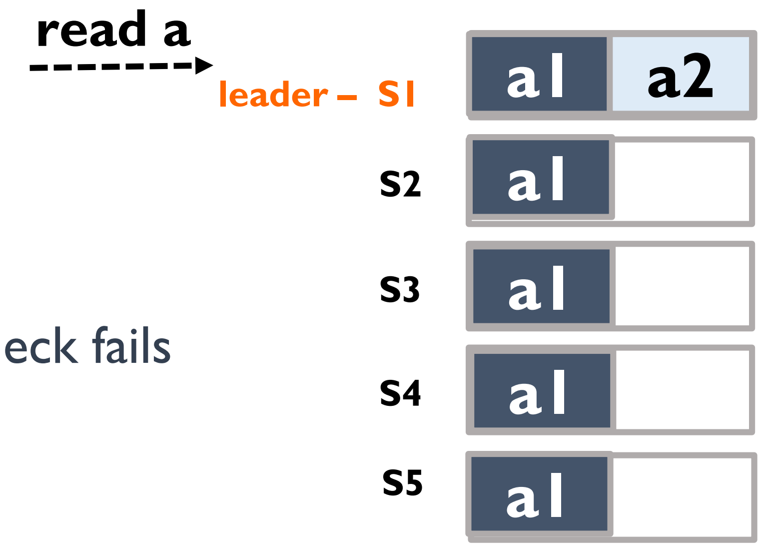
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient



durability check fails

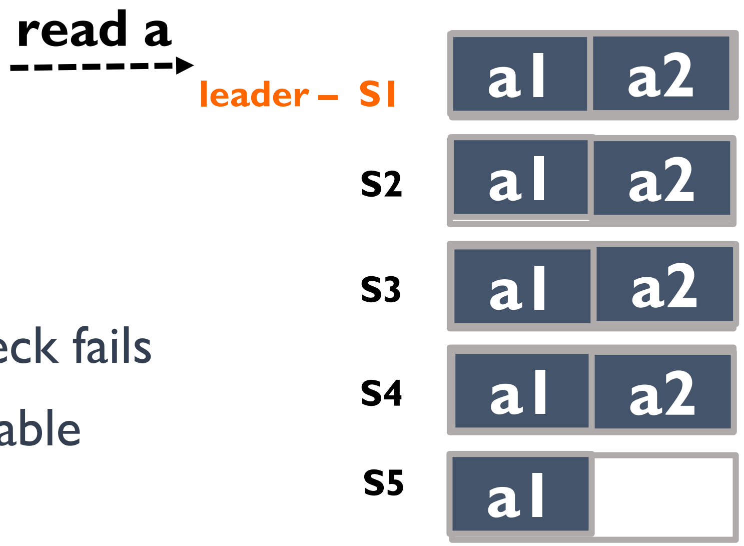
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient



durability check fails

make durable

Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

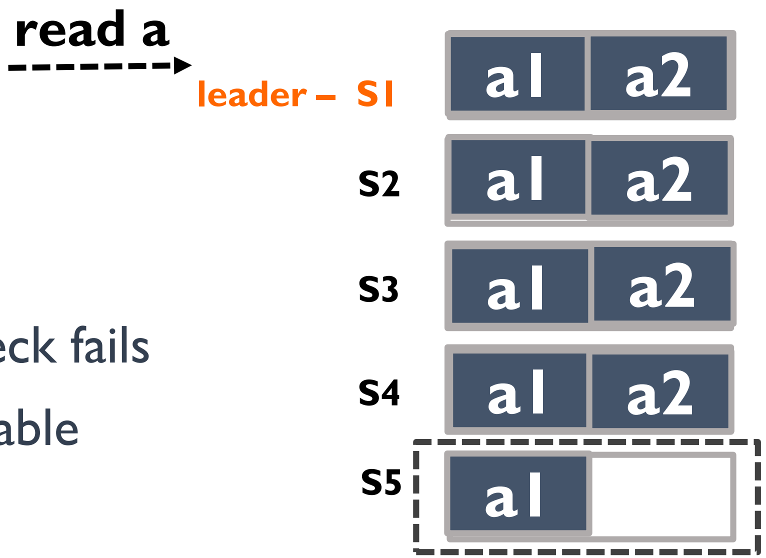
not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient

durability check fails

make durable



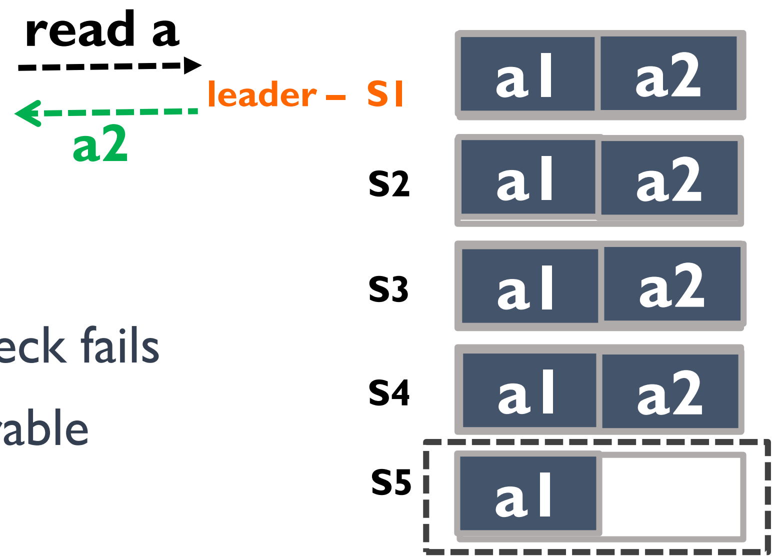
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient



durability check fails

make durable

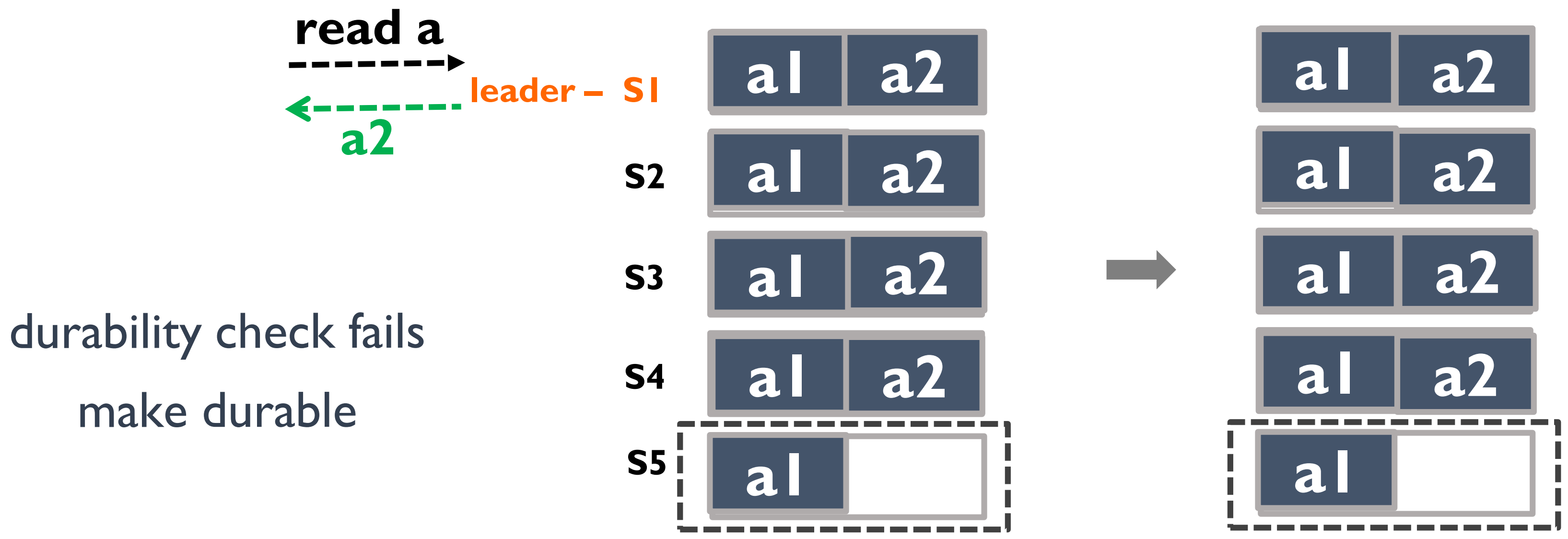
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient



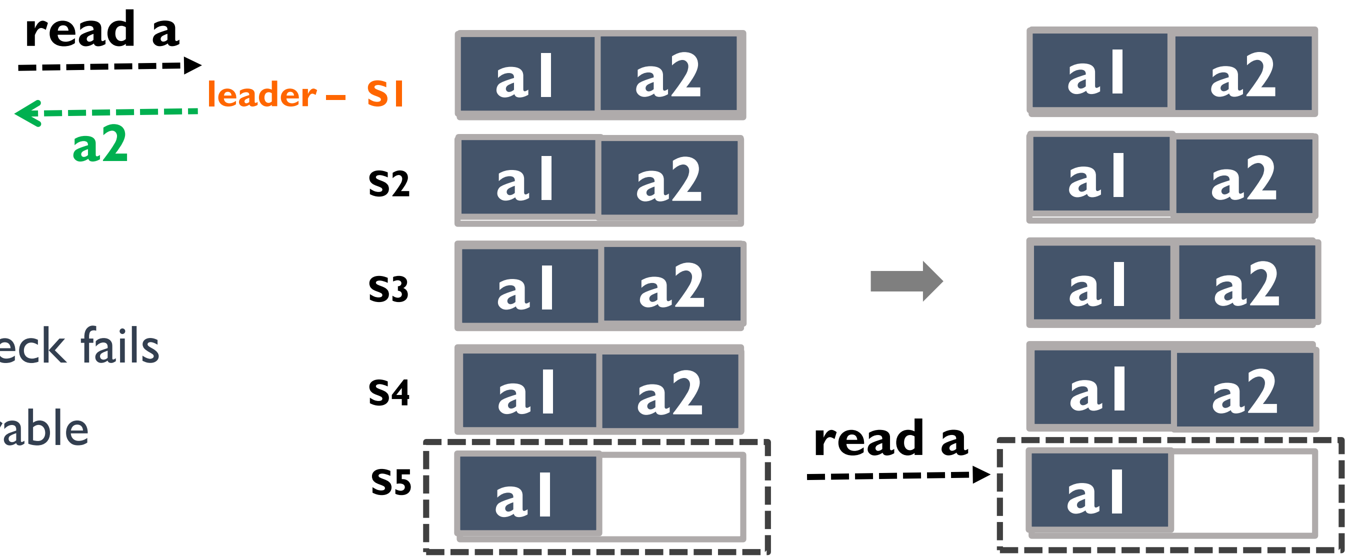
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient



durability check fails

make durable

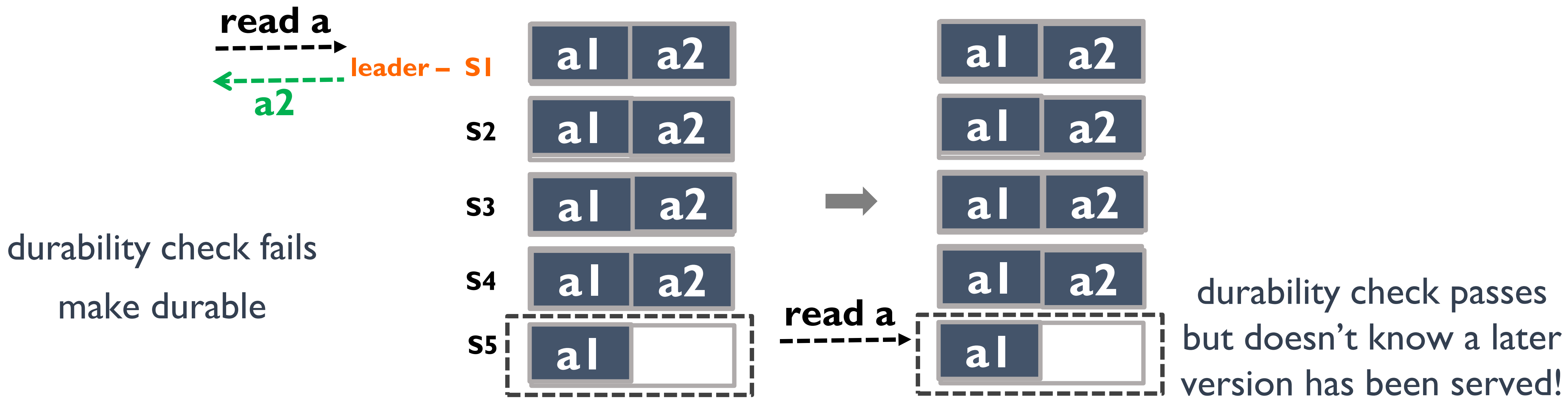
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient



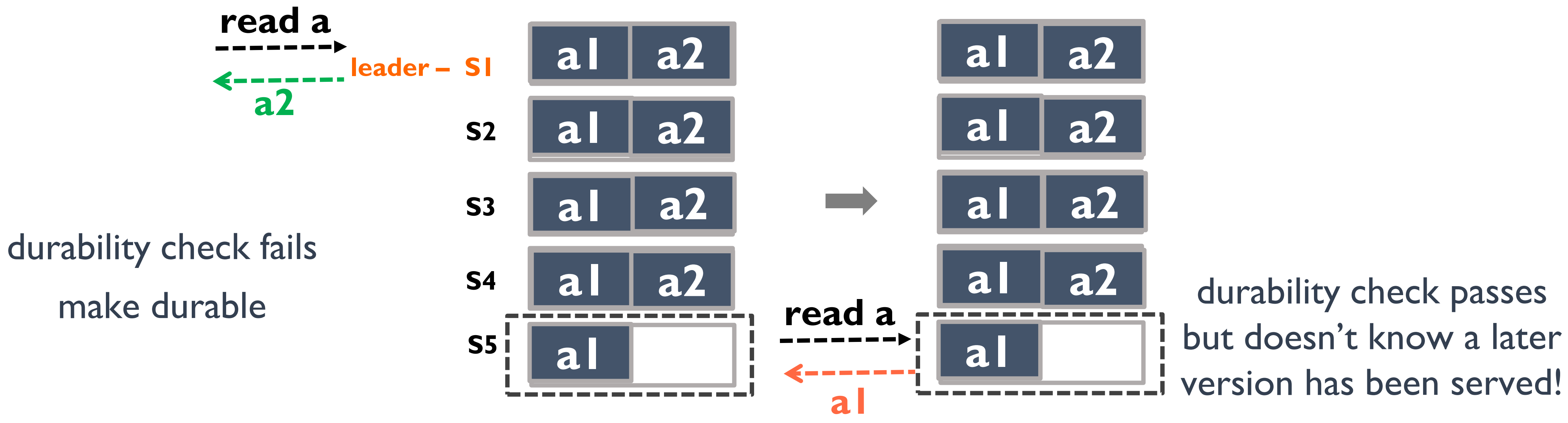
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient



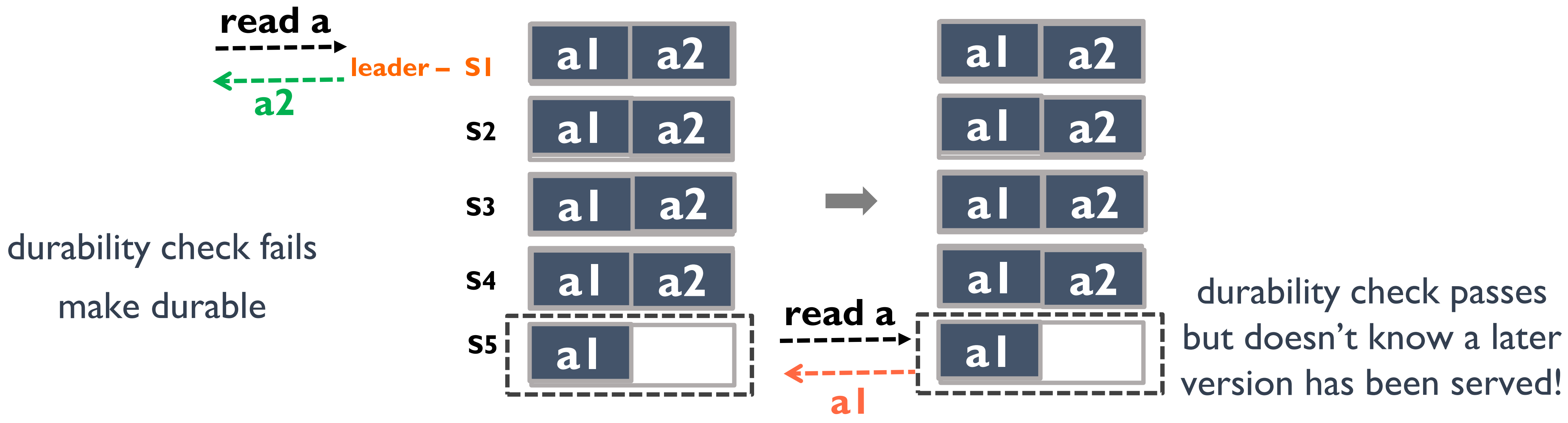
Cross-Client Monotonic Reads in ORCA

If reads restricted to leader, CAD provides cross-client monotonic reads

not scalable

Allow reads at followers

lagging followers could cause out-of-order states, CAD is not sufficient



Additional mechanisms: Active sets (lease-based mechanism), not in this talk…

Outline

Introduction

Motivation

CAD and cross-client monotonic reads

ORCA design

Results

Summary and conclusion

Evaluation

Evaluation

Implemented in ZooKeeper

Evaluation

Implemented in ZooKeeper

Evaluate different durability models in isolation

- compare CAD against immediate and eventual durability

Evaluation

Implemented in ZooKeeper

Evaluate different durability models in isolation

- compare CAD against immediate and eventual durability

Evaluate overall system performance

- ORCA against strong and weakly consistent ZooKeeper

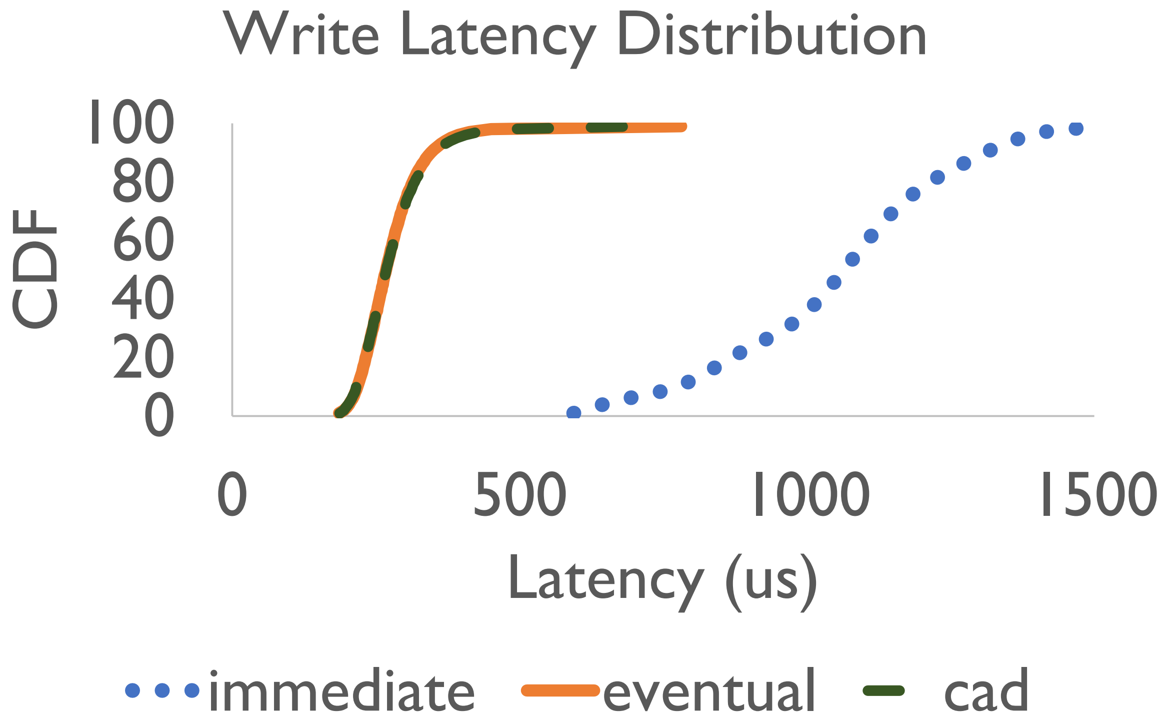
CAD Durability Layer Performance

CAD Durability Layer Performance

YCSB-A: 50% W, 50% R

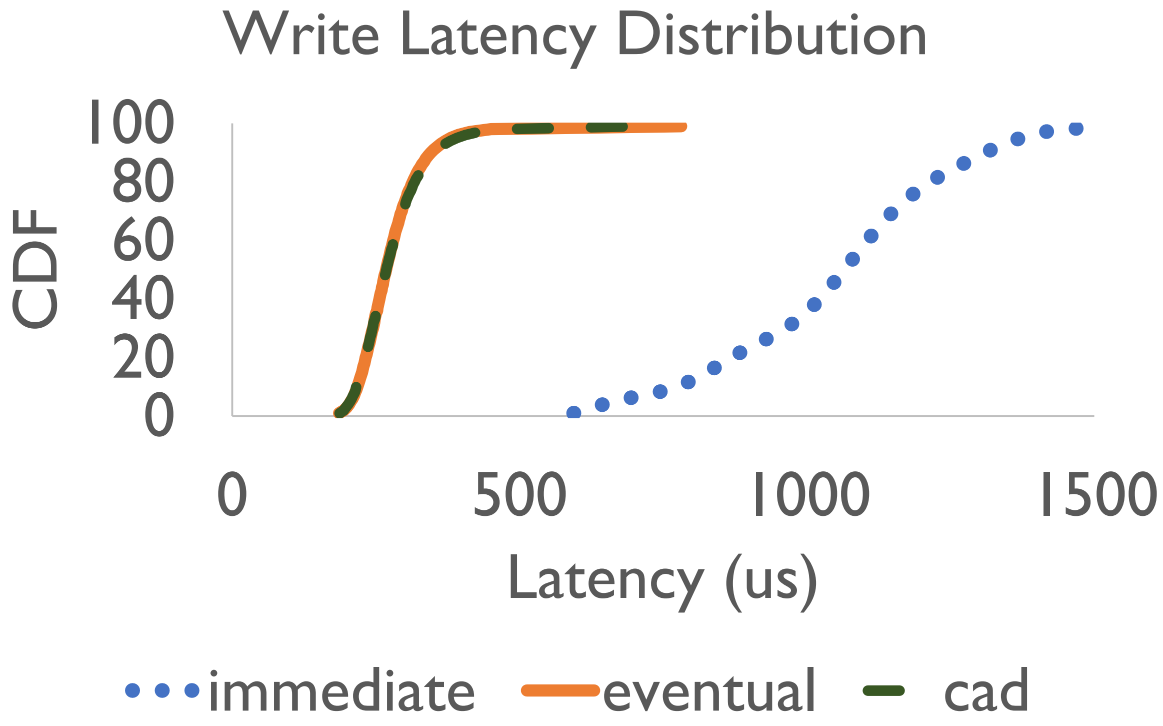
CAD Durability Layer Performance

YCSB-A: 50% W, 50% R



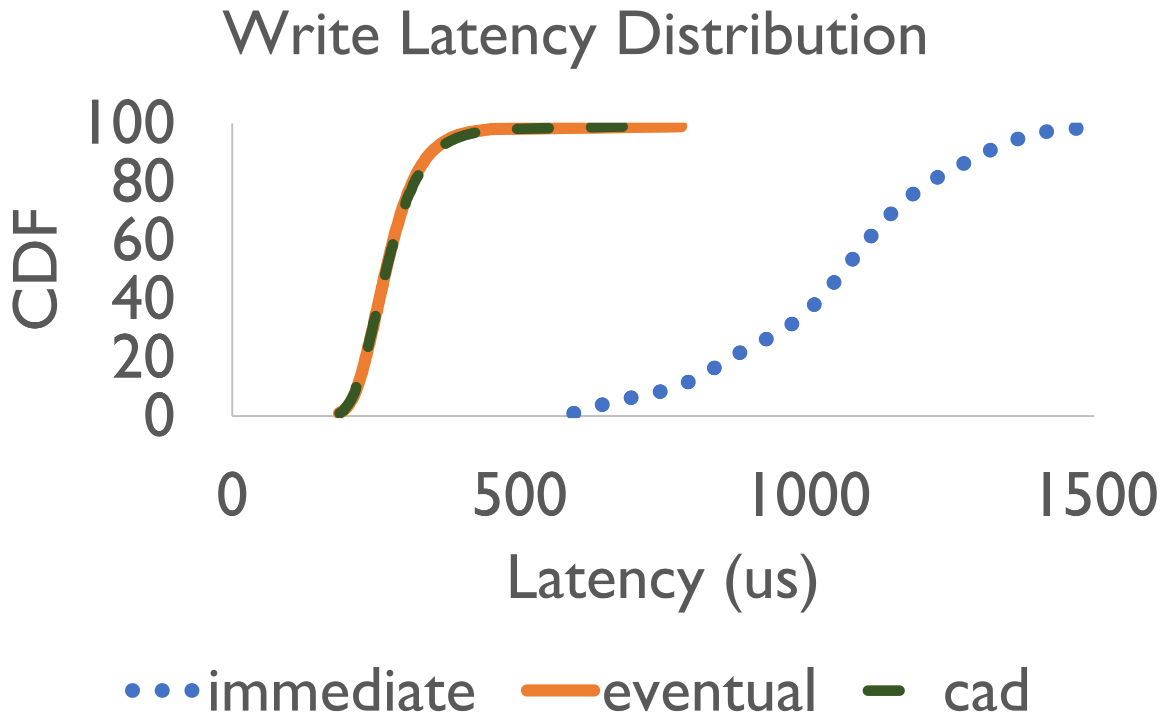
CAD Durability Layer Performance

YCSB-A: 50% W, 50% R



CAD Durability Layer Performance

YCSB-A: 50% W, 50% R

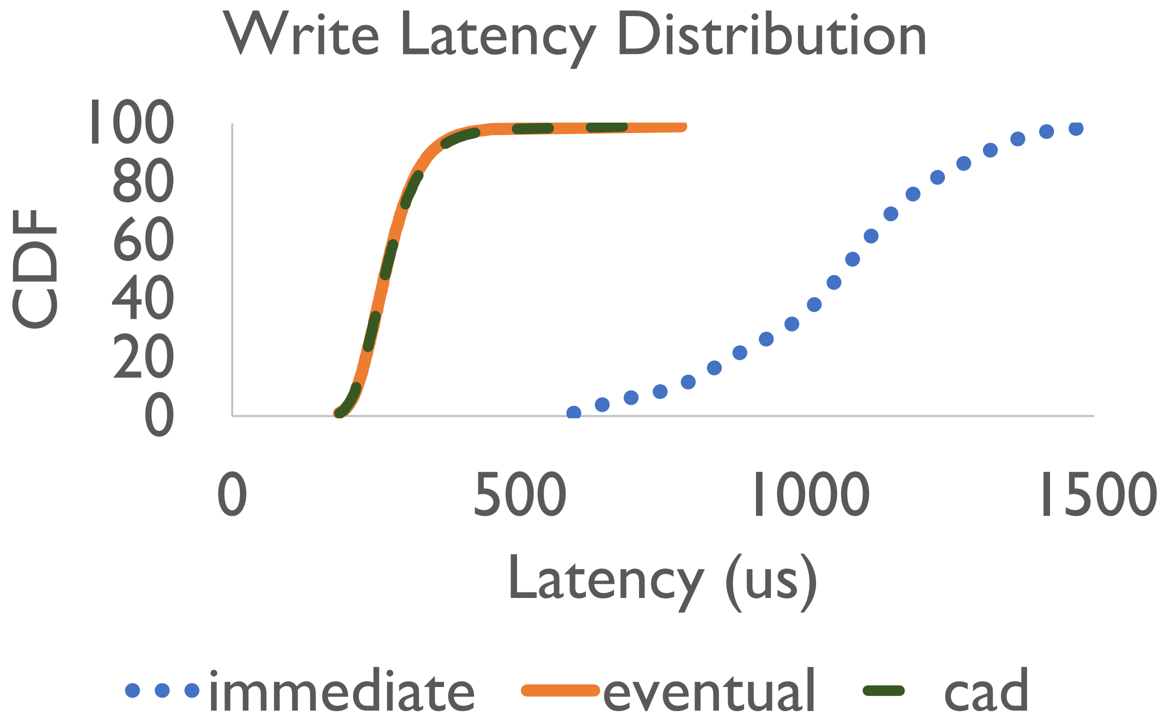


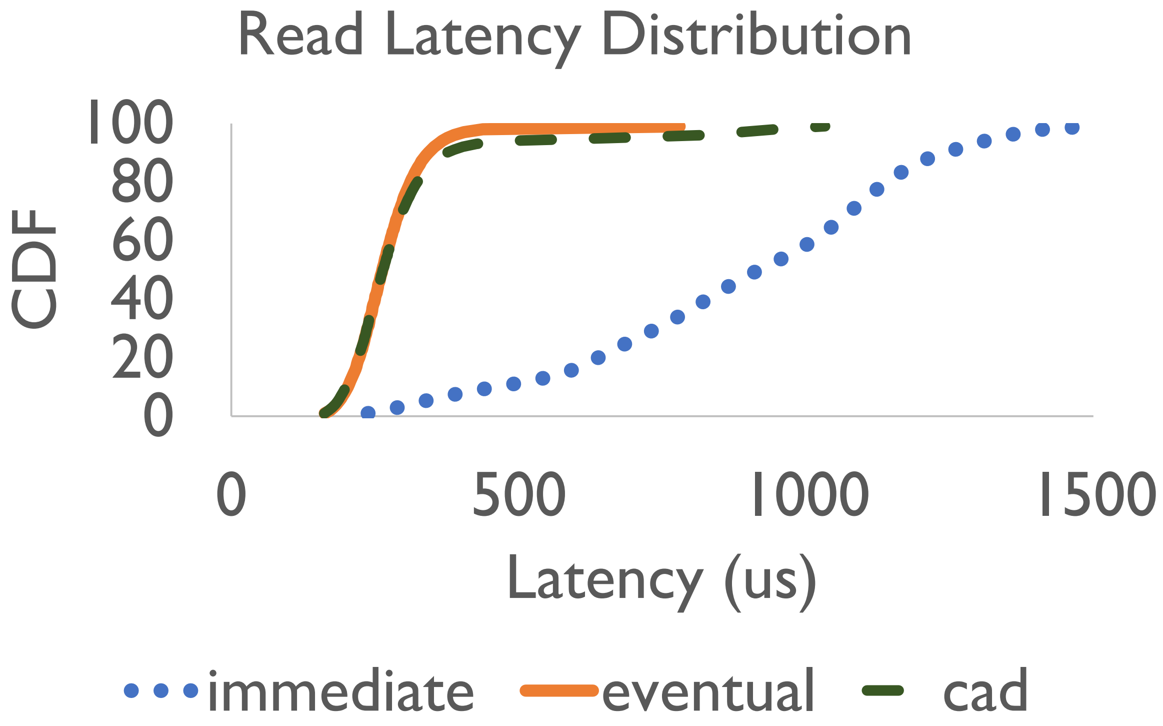
CAD writes faster than immediate durability

CAD matches performance of eventual

CAD Durability Layer Performance

YCSB-A: 50% W, 50% R

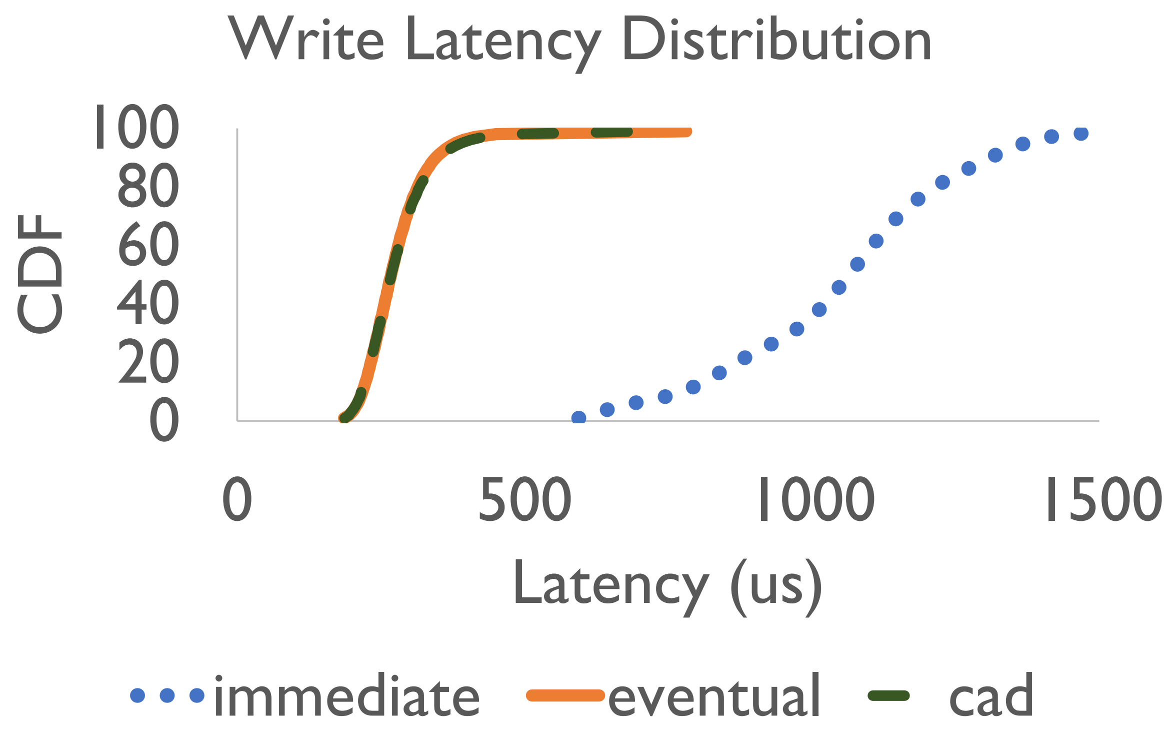


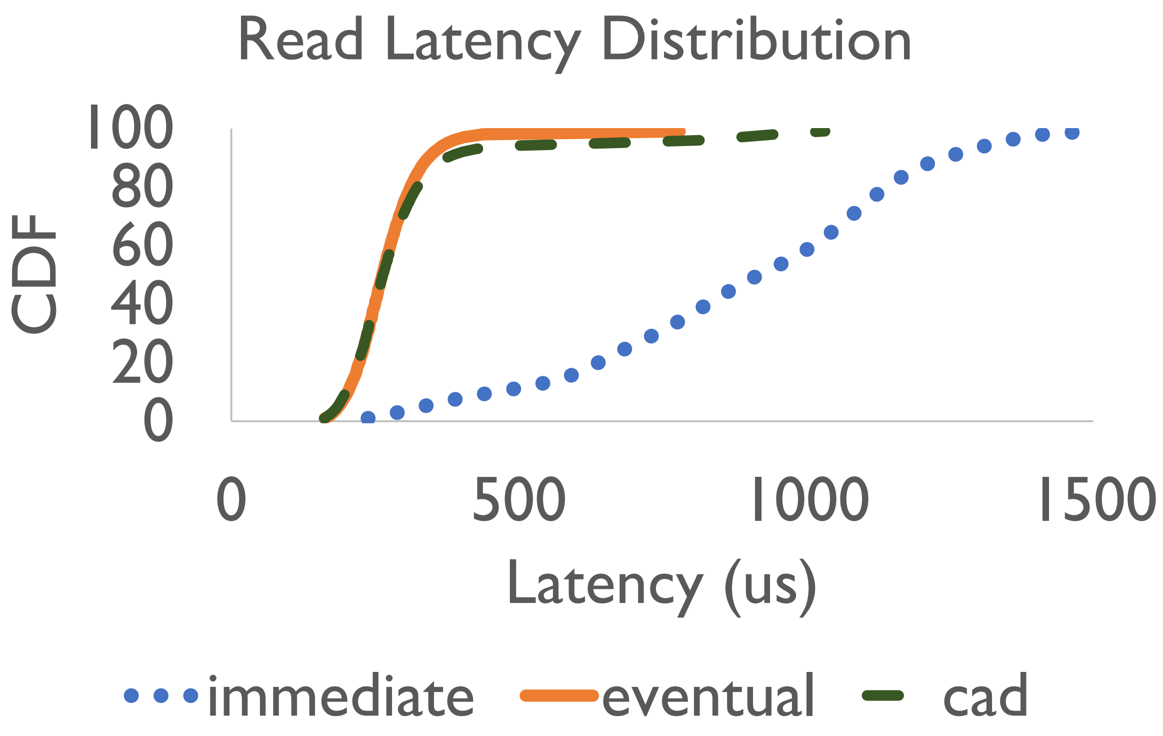


CAD writes faster than immediate durability
CAD matches performance of eventual

CAD Durability Layer Performance

YCSB-A: 50% W, 50% R



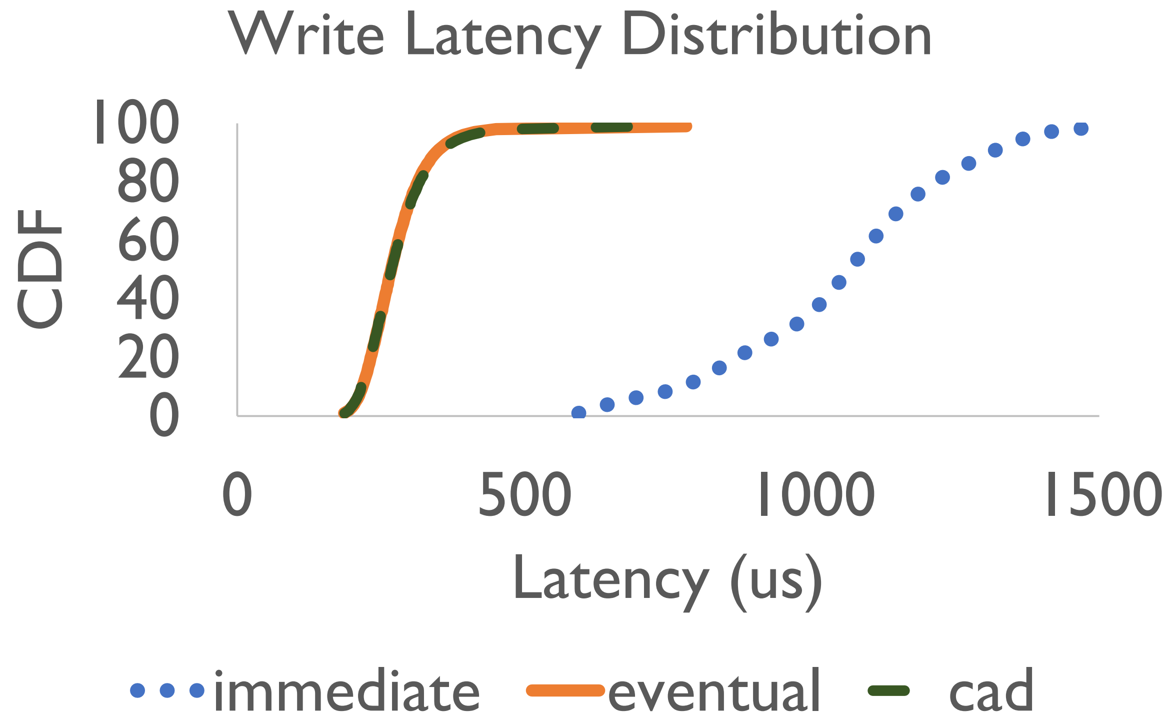


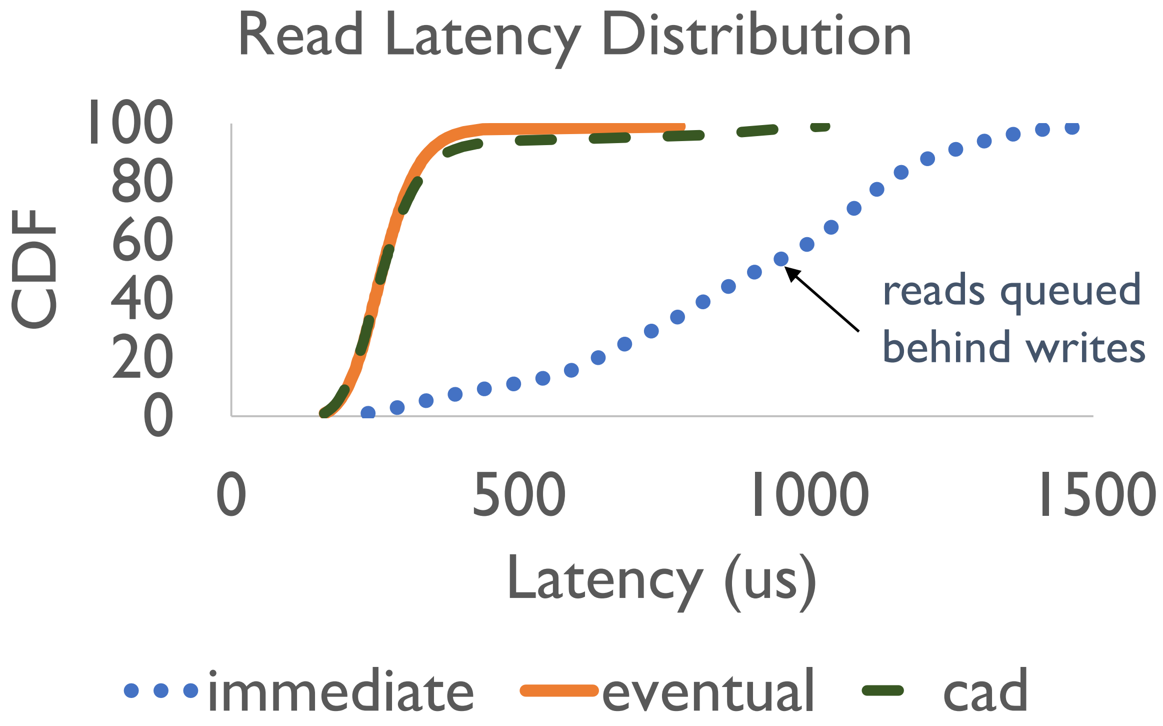
CAD writes faster than immediate durability

CAD matches performance of eventual

CAD Durability Layer Performance

YCSB-A: 50% W, 50% R



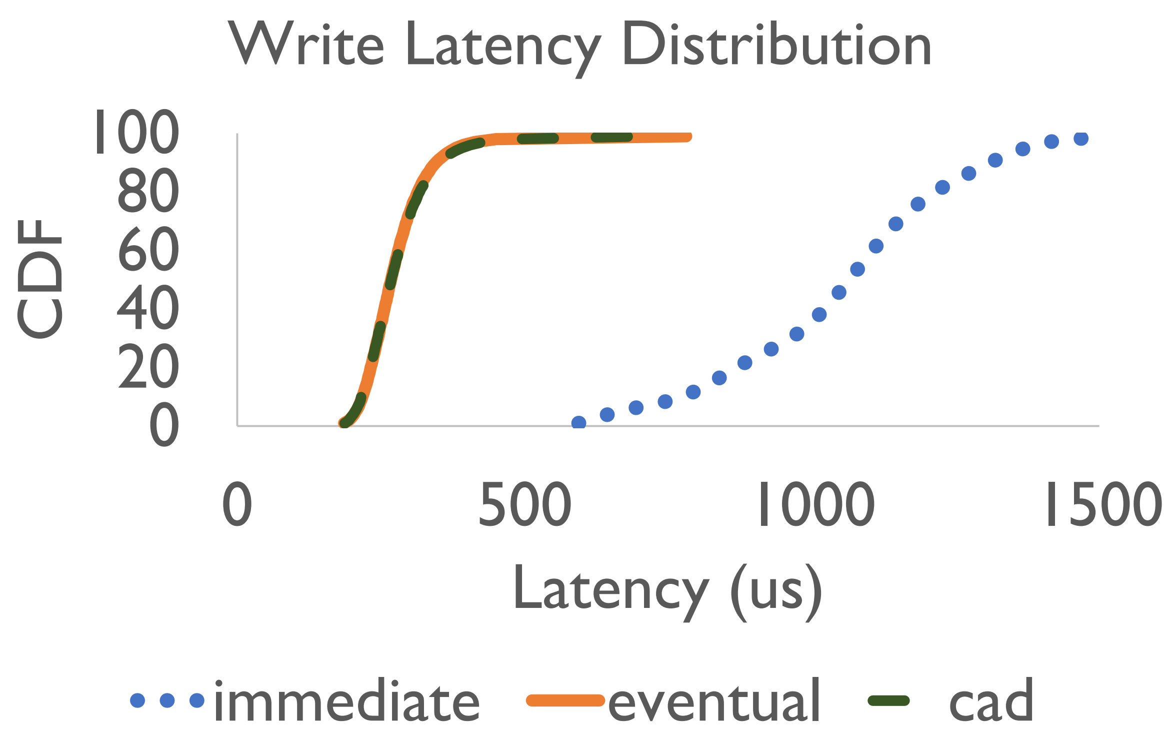


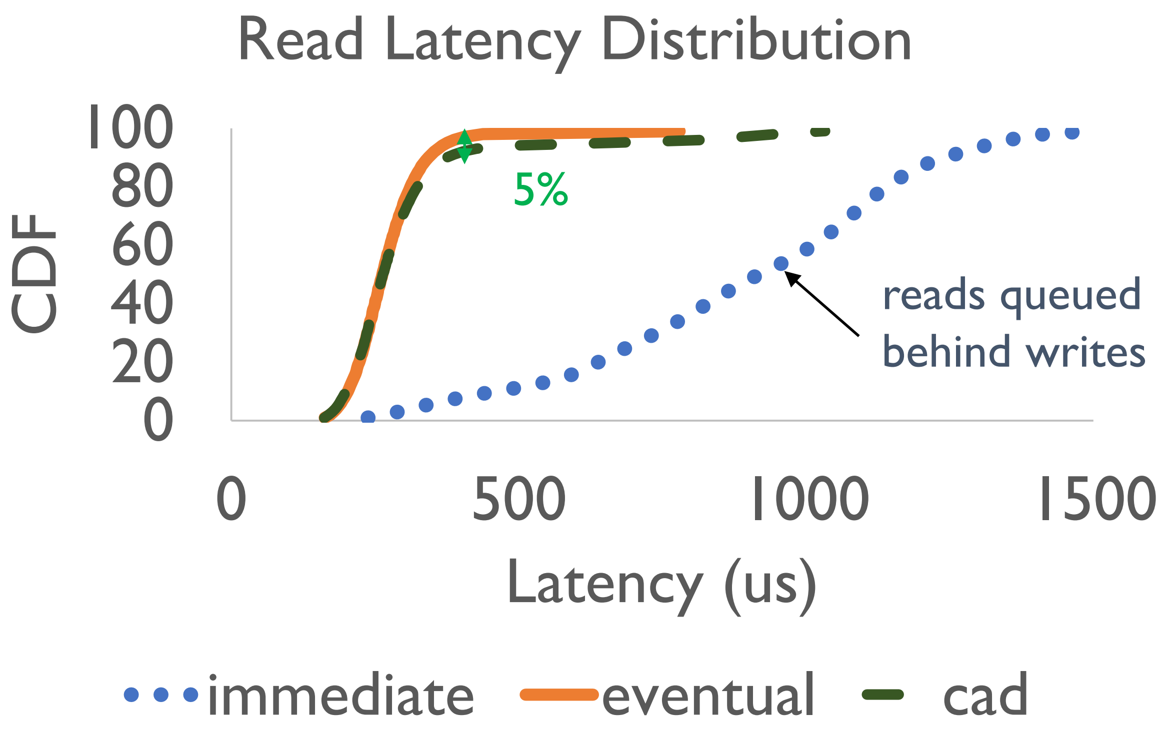
CAD writes faster than immediate durability

CAD matches performance of eventual

CAD Durability Layer Performance

YCSB-A: 50% W, 50% R



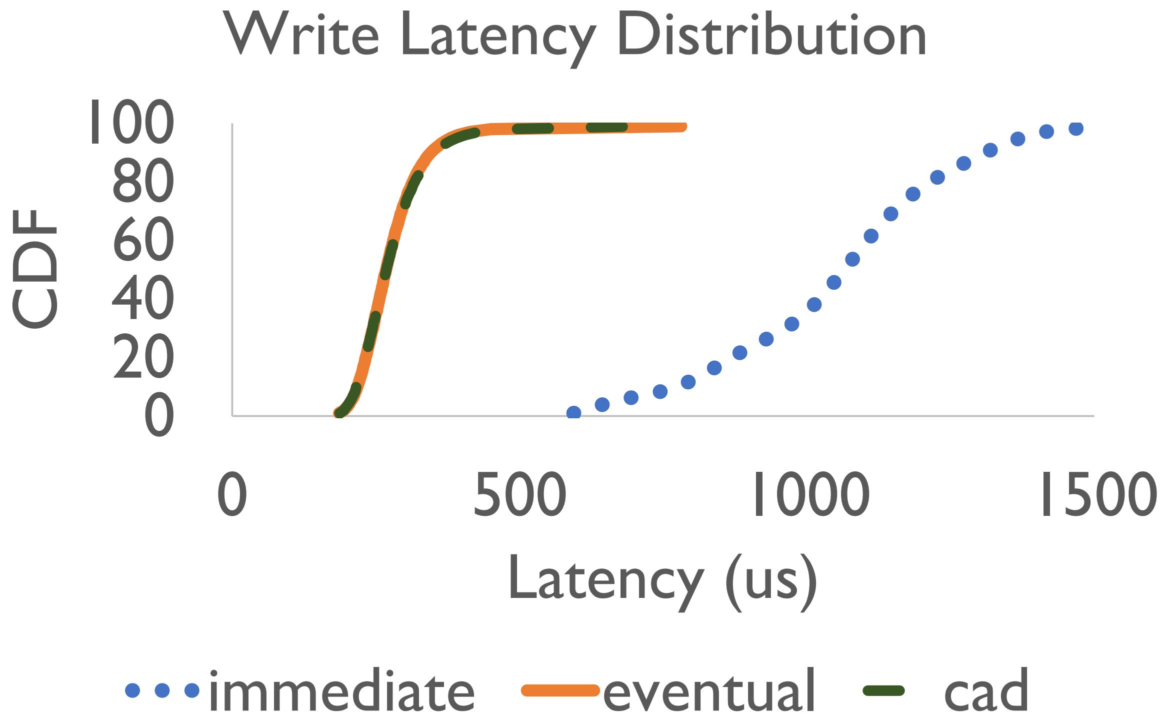


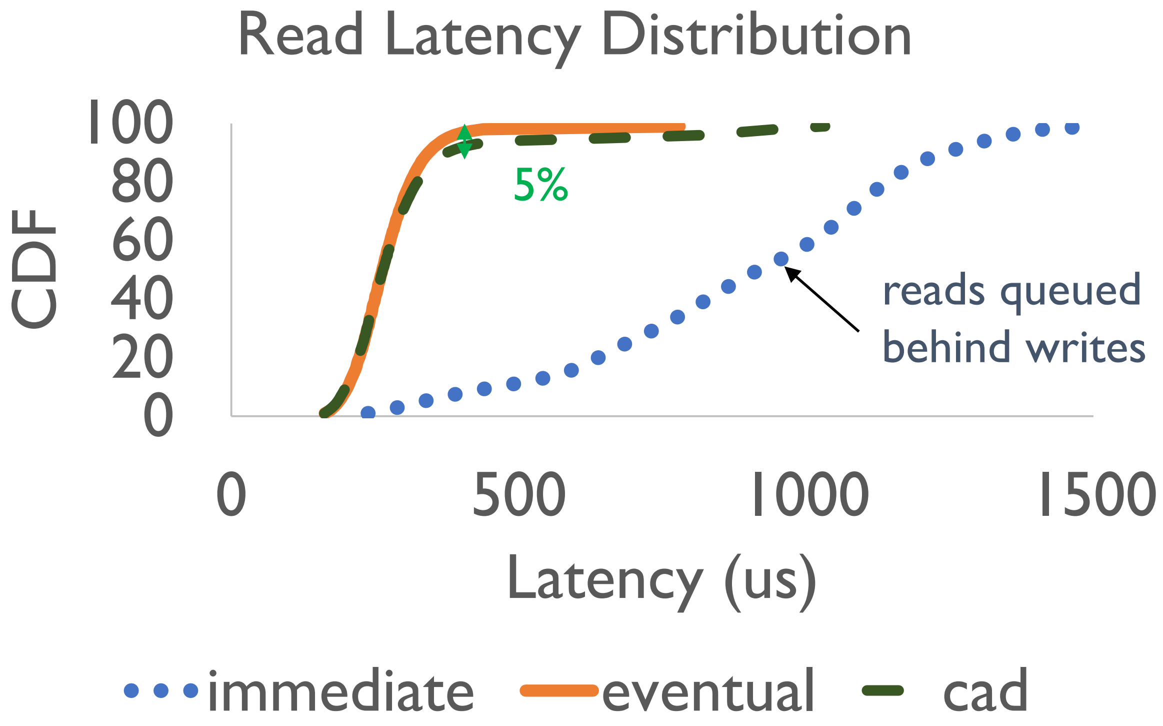
CAD writes faster than immediate durability
CAD matches performance of eventual

Most reads in CAD fast
Only 5% slow due to synchronous ops

CAD Durability Layer Performance

YCSB-A: 50% W, 50% R





CAD writes faster than immediate durability
CAD matches performance of eventual

Most reads in CAD fast
Only 5% slow due to synchronous ops

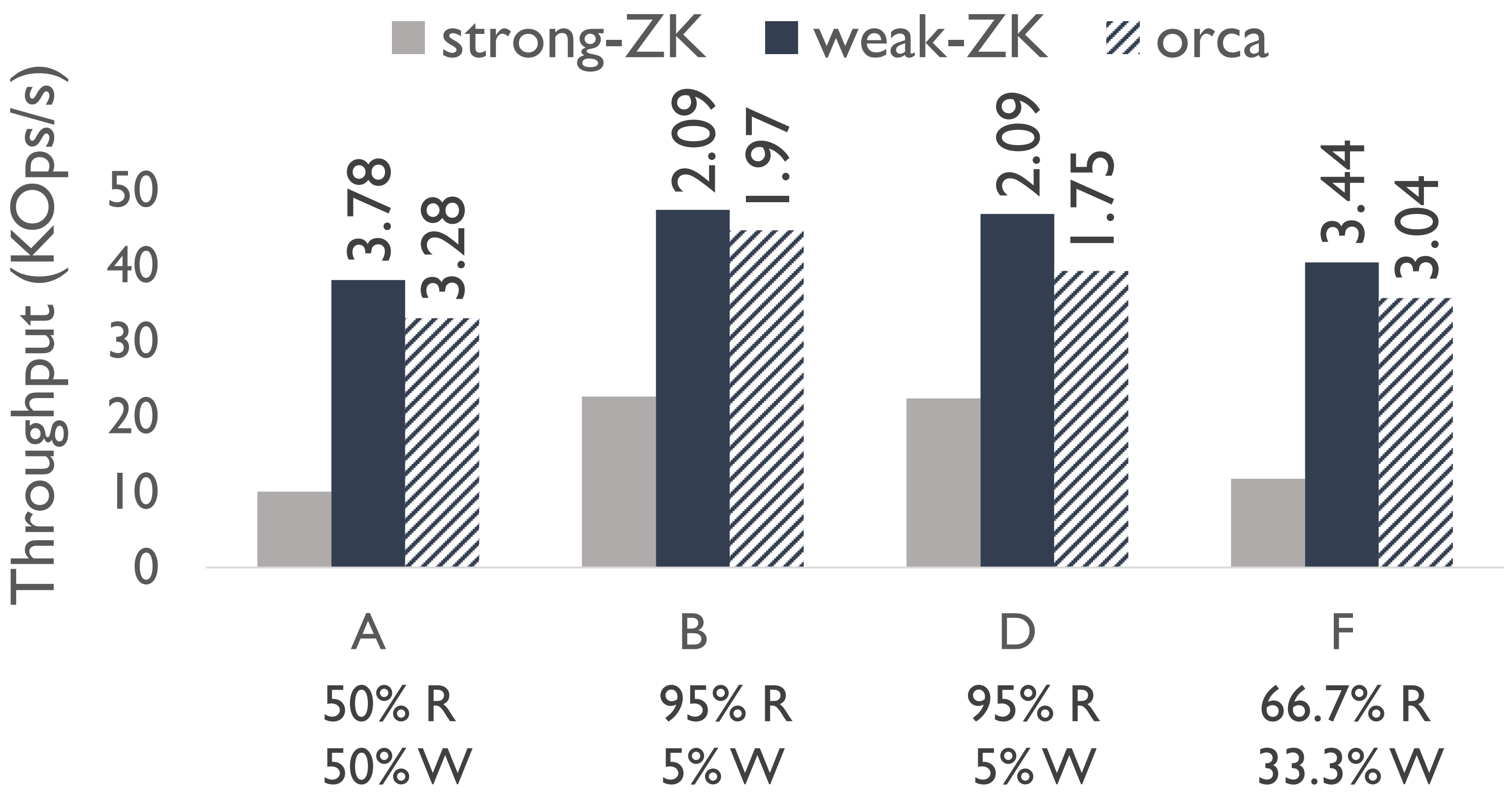
CAD performs similar to eventual and is faster than immediate

ORCA System Performance

Strong-ZK – uses immediate durability, reads only at leader
Weak-ZK – uses eventual durability, reads at many nodes
ORCA – uses CAD, reads at many nodes

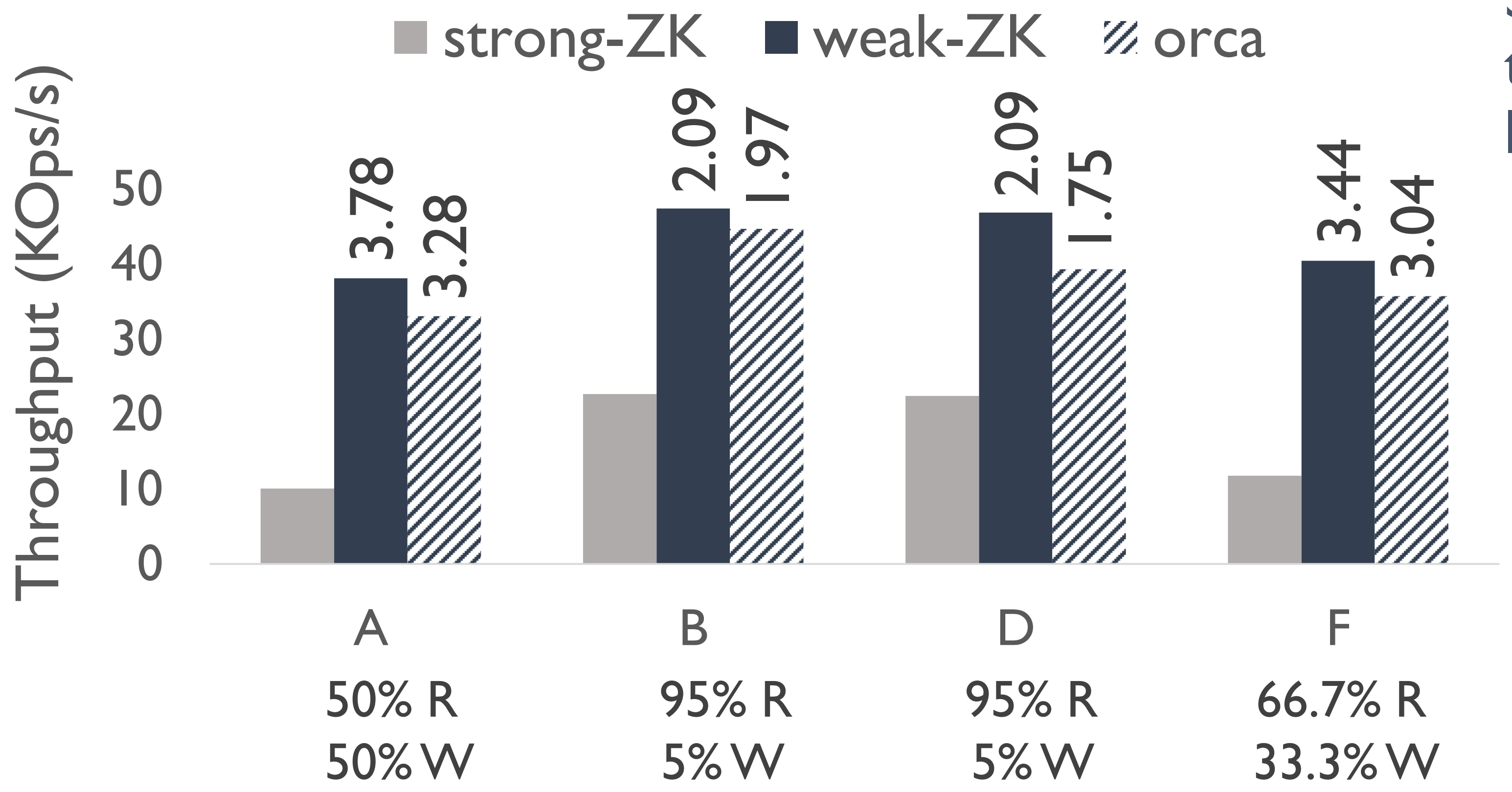
ORCA System Performance

Strong-ZK – uses immediate durability, reads only at leader
Weak-ZK – uses eventual durability, reads at many nodes
ORCA – uses CAD, reads at many nodes



ORCA System Performance

Strong-ZK – uses immediate durability, reads only at leader
Weak-ZK – uses eventual durability, reads at many nodes
ORCA – uses CAD, reads at many nodes



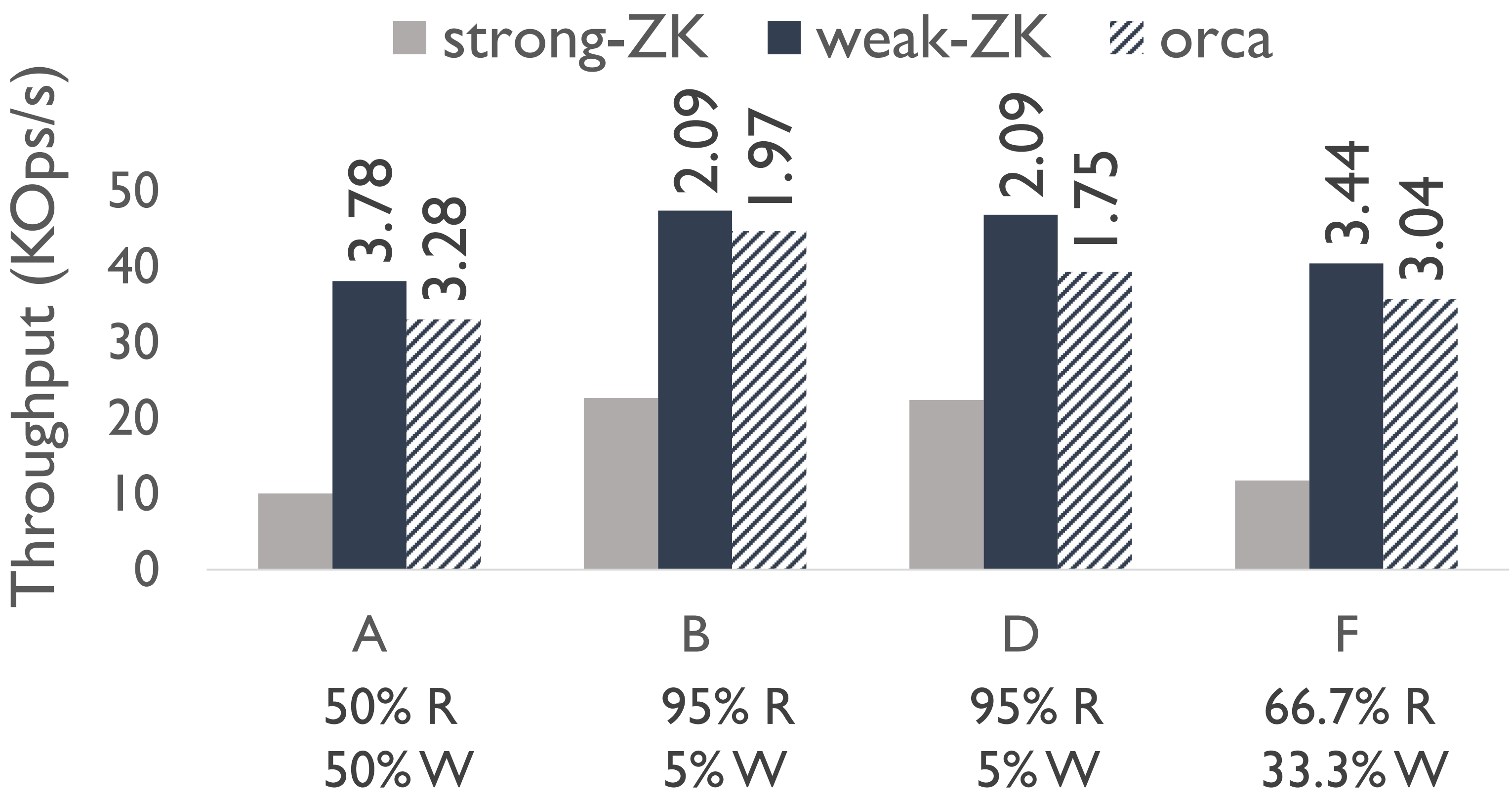
Strong-ZK performs poorly due to immediate durability and leader-restricted reads

ORCA System Performance

Strong-ZK – uses immediate durability, reads only at leader

Weak-ZK – uses eventual durability, reads at many nodes

ORCA – uses CAD, reads at many nodes



Strong-ZK performs poorly due to immediate durability and leader-restricted reads

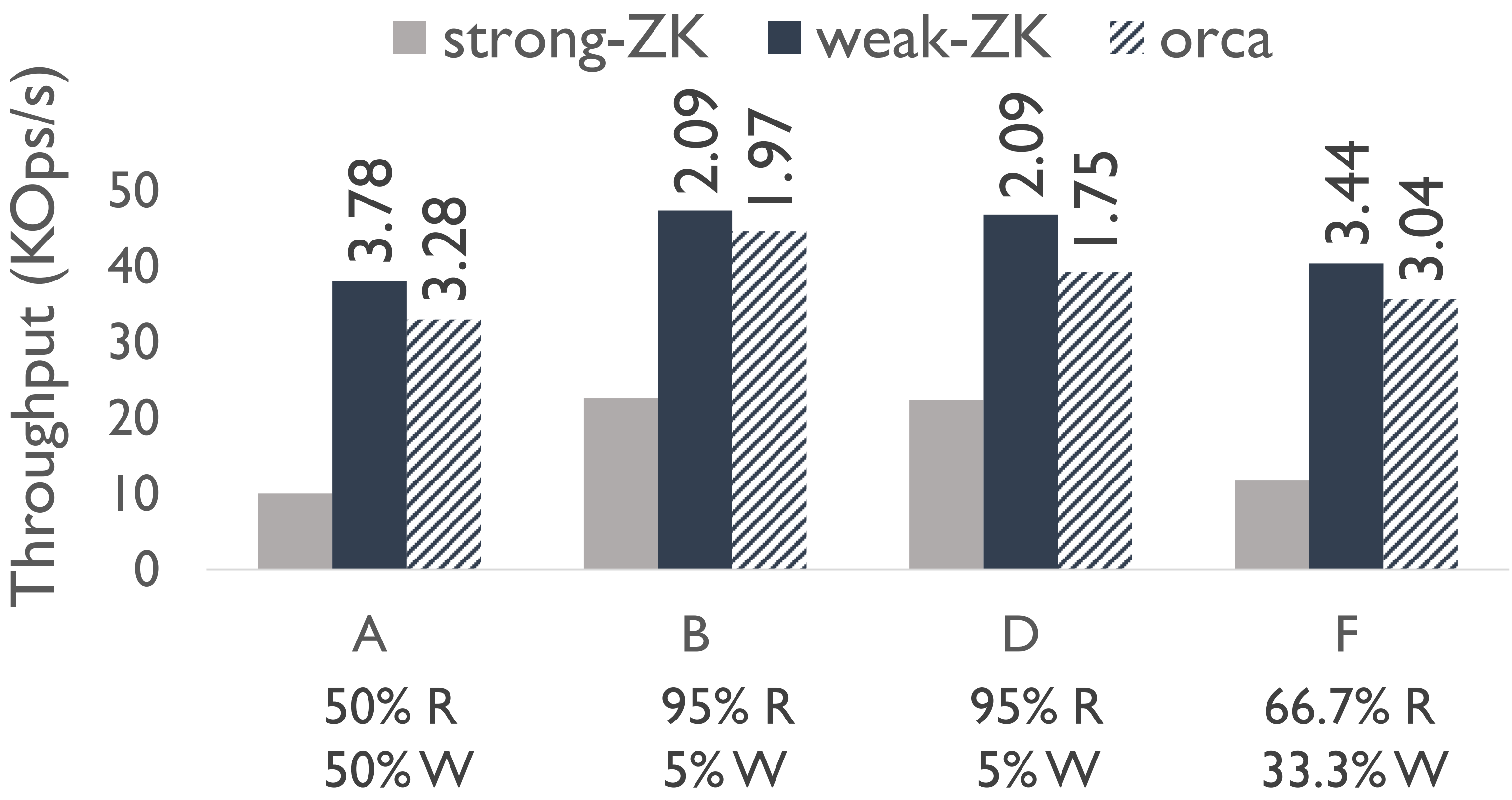
Weak-ZK performs well due to eventual durability and scalable reads

ORCA System Performance

Strong-ZK – uses immediate durability, reads only at leader

Weak-ZK – uses eventual durability, reads at many nodes

ORCA – uses CAD, reads at many nodes



Strong-ZK performs poorly due to immediate durability and leader-restricted reads

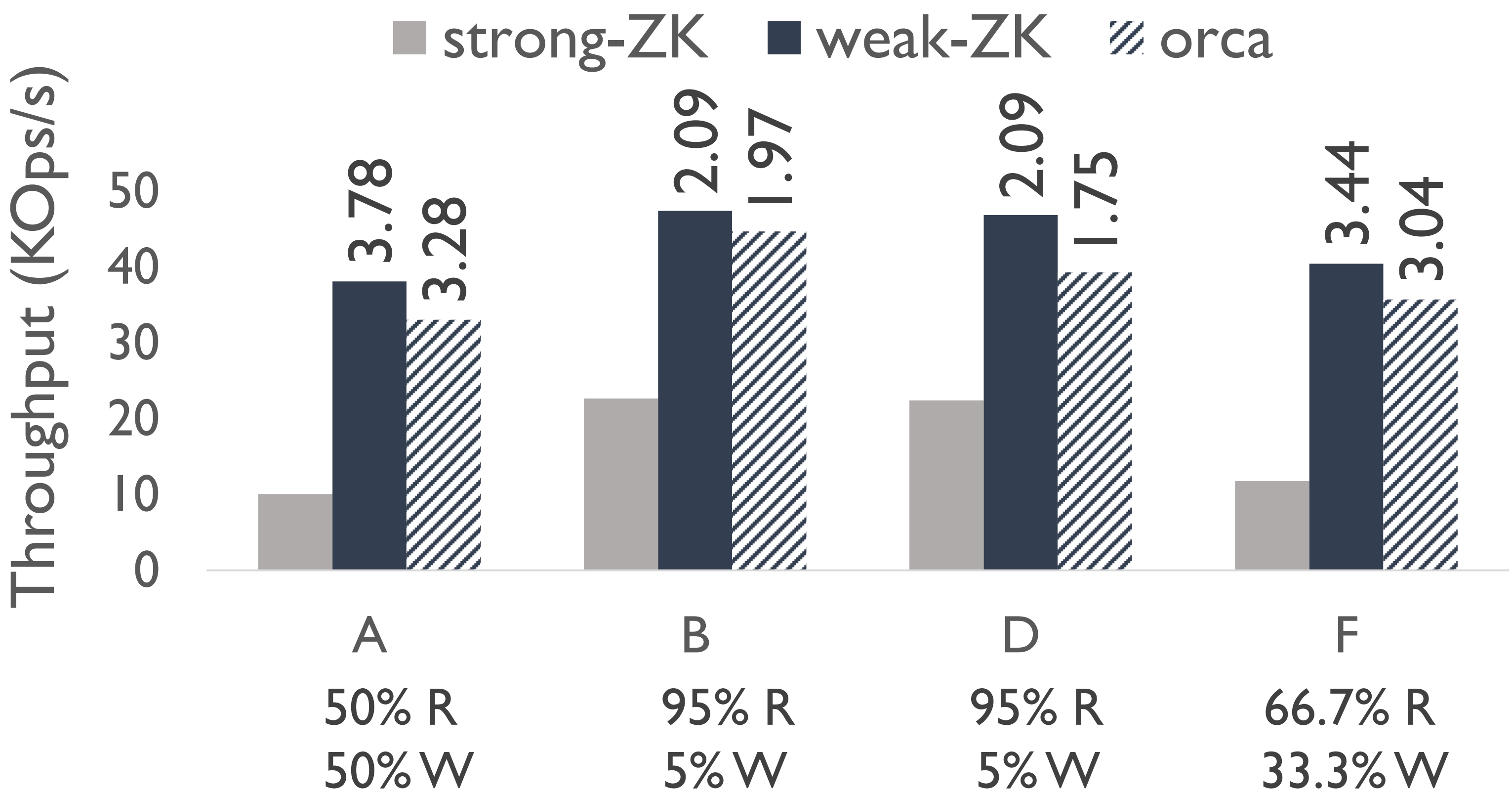
Weak-ZK performs well due to eventual durability and scalable reads

ORCA adds little overheads compared to weak-ZK

reads that access non-durable data

ORCA System Performance

Strong-ZK – uses immediate durability, reads only at leader
Weak-ZK – uses eventual durability, reads at many nodes
ORCA – uses CAD, reads at many nodes



Strong-ZK performs poorly due to immediate durability and leader-restricted reads

Weak-ZK performs well due to eventual durability and scalable reads

ORCA adds little overheads compared to weak-ZK

reads that access non-durable data

More experiments in the paper…

Evaluation

- correctness testing using a cluster crash-testing framework
- geo-replicated setting
- micro-benchmarks

Application case studies

- location-tracking
- social-media timeline

Summary and Conclusions

Summary and Conclusions

Surprisingly, durability models are overlooked

Summary and Conclusions

Surprisingly, durability models are overlooked

Immediate durability enables strong consistency but is slow

Summary and Conclusions

Surprisingly, durability models are overlooked

Immediate durability enables strong consistency but is slow

Eventual durability is fast but only enables weak consistency

Summary and Conclusions

Surprisingly, durability models are overlooked

Immediate durability enables strong consistency but is slow

Eventual durability is fast but only enables weak consistency

CAD – consistency-aware durability, a new way of thinking about durability

- enables both strong consistency and high performance

Summary and Conclusions

Surprisingly, durability models are overlooked

Immediate durability enables strong consistency but is slow

Eventual durability is fast but only enables weak consistency

CAD – consistency-aware durability, a new way of thinking about durability

- enables both strong consistency and high performance
- CAD is useful for many deployments that currently adopt eventual durability

Summary and Conclusions

Surprisingly, durability models are overlooked

Immediate durability enables strong consistency but is slow

Eventual durability is fast but only enables weak consistency

CAD – consistency-aware durability, a new way of thinking about durability

- enables both strong consistency and high performance
- CAD is useful for many deployments that currently adopt eventual durability

Consistency and performance are seemingly at odds – by carefully examining the underlying layer, achieve both

Summary and Conclusions

Surprisingly, durability models are overlooked

Immediate durability enables strong consistency but is slow

Eventual durability is fast but only enables weak consistency

CAD – consistency-aware durability, a new way of thinking about durability

- enables both strong consistency and high performance
- CAD is useful for many deployments that currently adopt eventual durability

Consistency and performance are seemingly at odds – by carefully examining the underlying layer, achieve both

Thank you!