LWTs in practice

Christopher Batey
@chbatey
The Last Pickle


SPRINGFIELD

Overview

Review of Cassandra’s consistency model

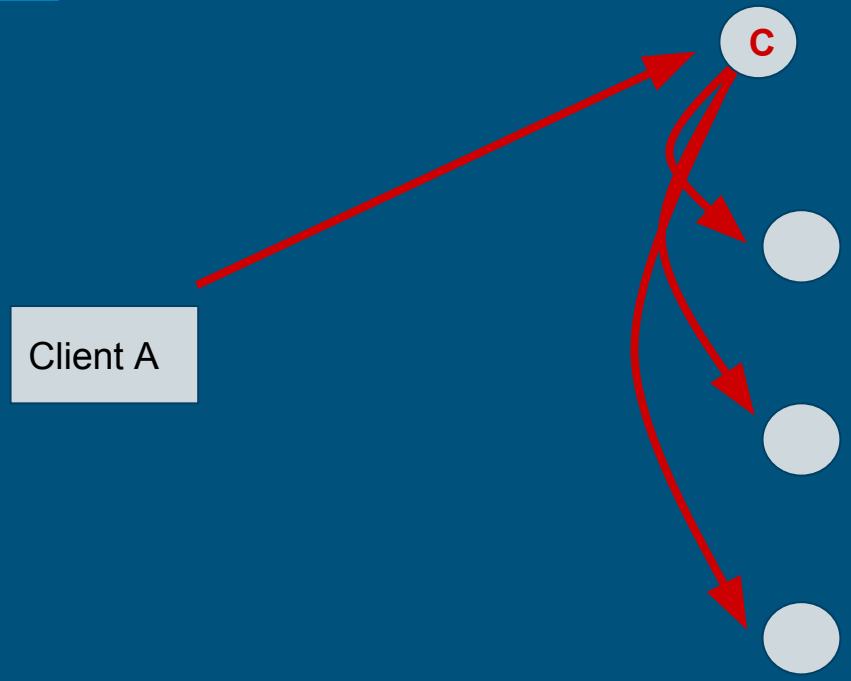
What are LWTs?

Why do we need them?

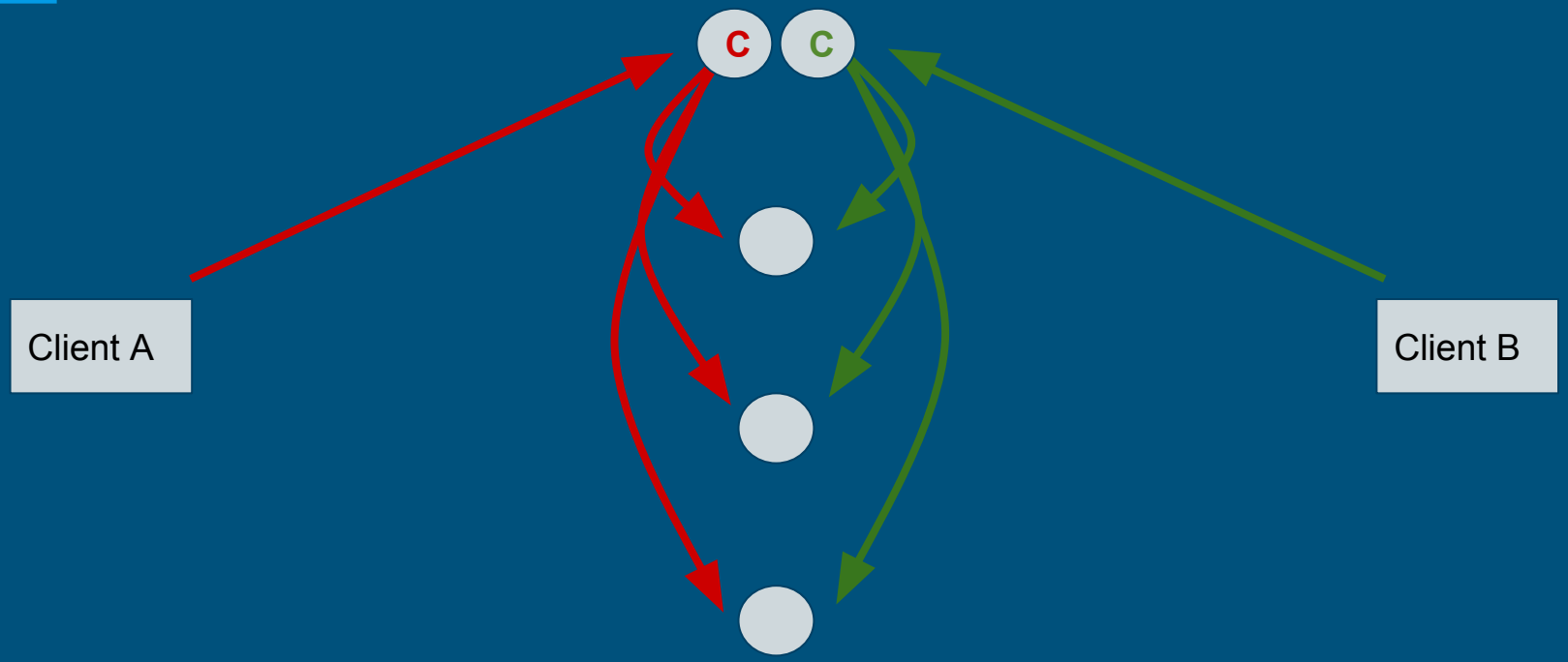
How do they work?

How do you use them?

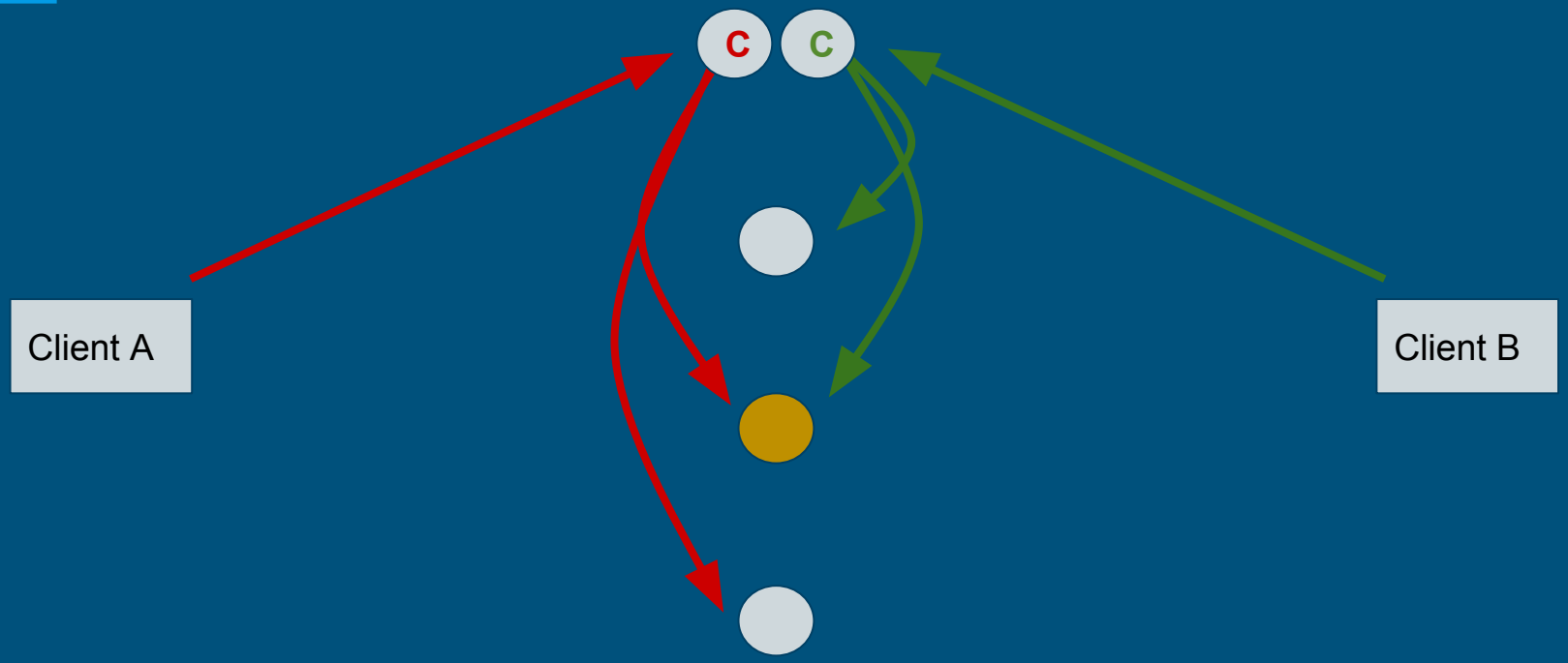
Writes



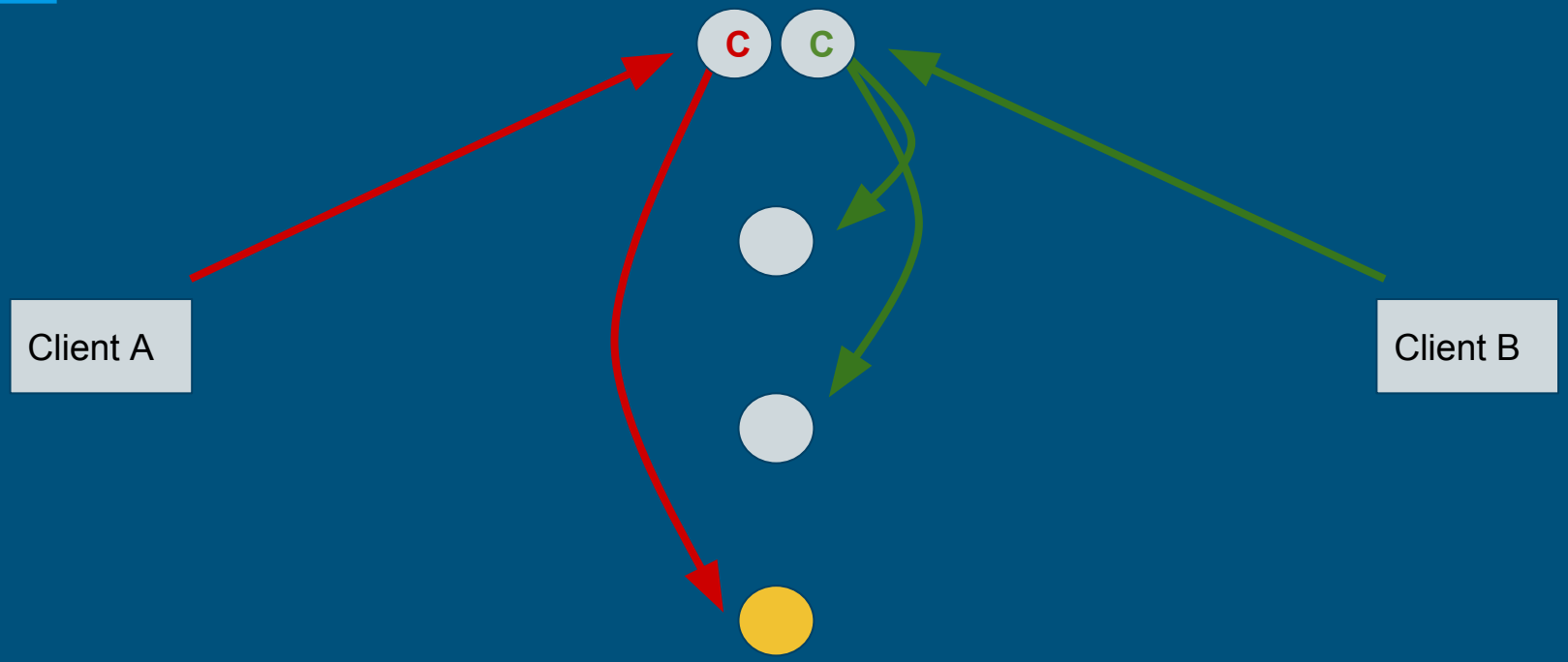
Concurrent writes



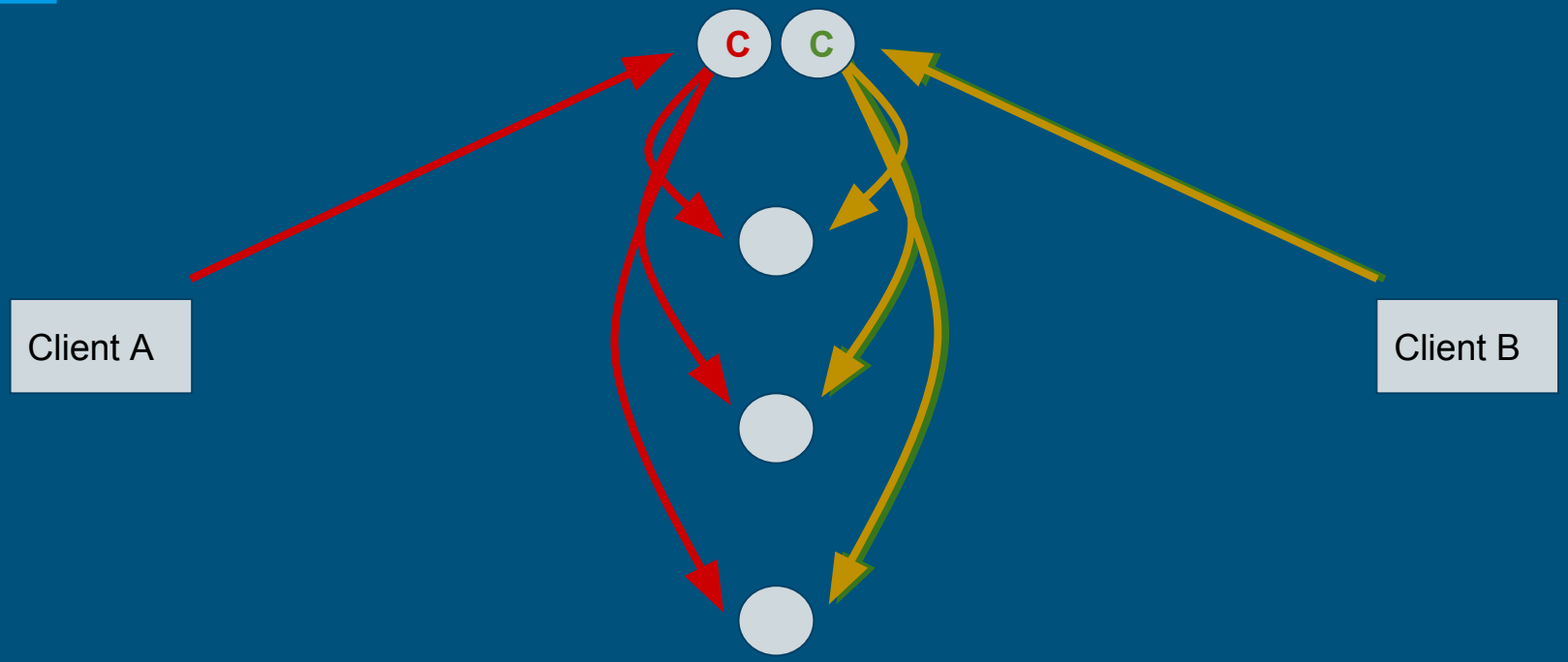
QUORUM based consistency



QUORUM based consistency



QUORUM based consistency



Voucher example

```
CREATE TABLE vouchers_mutable (
    name text PRIMARY KEY,
    sold int
)
```

Read and write race condition with Quorum

- Client A read the number of ticket sales at 299
- Client B read the number of ticket sales at 299
- Client A sells ticket 300
- Client B sells ticket 300

Compare and set

Enter Light Weight Transactions

- Client A read the number of ticket sales at 299
- Client B read the number of ticket sales at 299
- Client A sells ticket 300 if total sold is 299
- ~~Client B sells ticket 300 if total sold is 299~~

Examples

Uniqueness

```
CREATE TABLE users (
    user_name text PRIMARY KEY,
    email text,
    password text
)
```

```
INSERT INTO users (user_name, password, email )
    VALUES ( 'chbatey', 'different',
'adifferentchris@gmail.com' ) IF NOT EXISTS;
```

Finite resource

```
CREATE TABLE vouchers_mutable (
    name text PRIMARY KEY,
    sold int
)
```

```
UPDATE vouchers_mutable SET sold = 1
WHERE name = 'free tv' IF sold = 0;
```

Immutable events

```
CREATE TABLE vouchers (
    name text,
    when timeuuid,
    who text,
    PRIMARY KEY (name, when)
);
```

Batches + LWTs

```
CREATE TABLE vouchers (
    name text,
    when timeuuid,
    sold int static,
    who text,
    PRIMARY KEY (name, when)
);
```

```
INSERT INTO vouchers (name, sold) VALUES
( 'free tv', 0);
```

Batches + LWTs

```
BEGIN BATCH
    UPDATE vouchers SET sold = 1 WHERE name = 'free tv' IF sold  = 0
    INSERT INTO vouchers (name, when, who) VALUES ( 'free tv', now(), 'chris')
APPLY BATCH ;

 [applied]
-----------
      True
```

Batches + LWTs

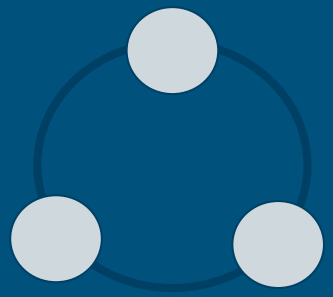
```
BEGIN BATCH
    UPDATE vouchers SET sold = 1 WHERE name = 'free tv' IF sold = 0
    INSERT INTO vouchers (name, when, who) VALUES ( 'free tv', now(), 'charlie')
APPLY BATCH ;

 [applied] | name    | when | sold
-----------+---------+------+------
     False | free tv | null |    1
```

How they work

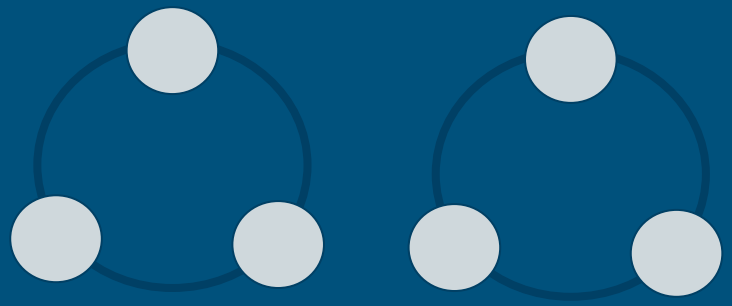
LWTs be puzzling

1. Why does a LWT have two consistency levels?
2. What is this SERIAL consistency I keep hearing about?
3. What are SERIAL reads?
4. Why does my LWT fail but the value still get written?
5. Why are they so damn slow?

Consensus for a partition



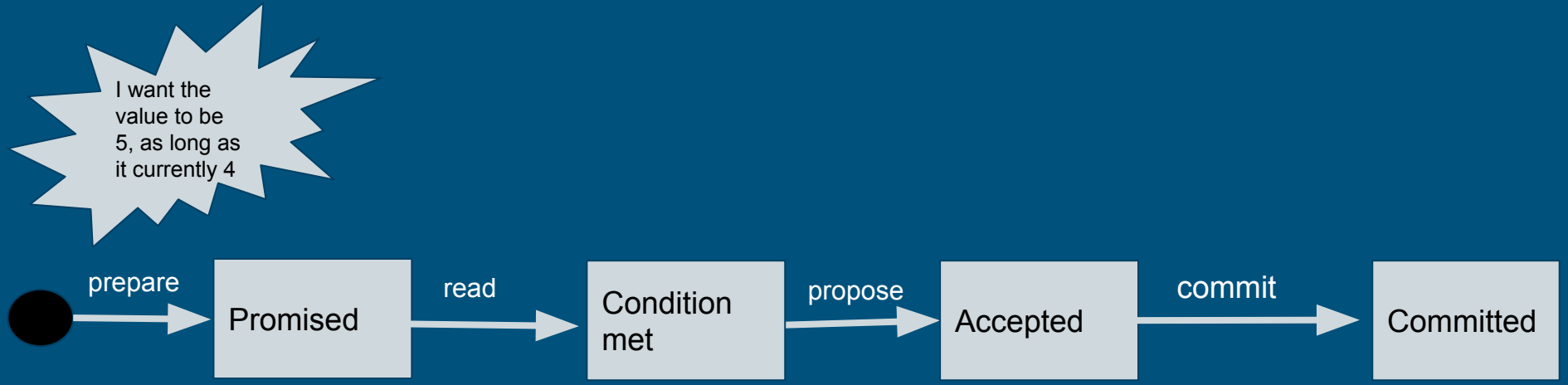
Consensus for a partition



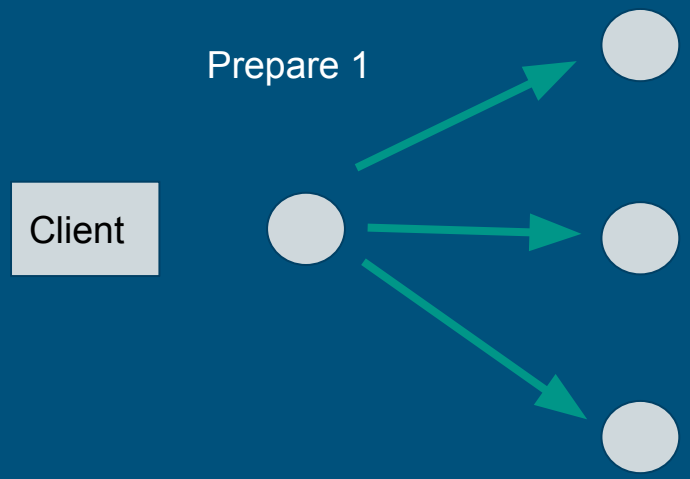
Stages of a LWT

- Prepare and promise
- Read existing value
- Propose and accept
- Commit

Consensus for a partition

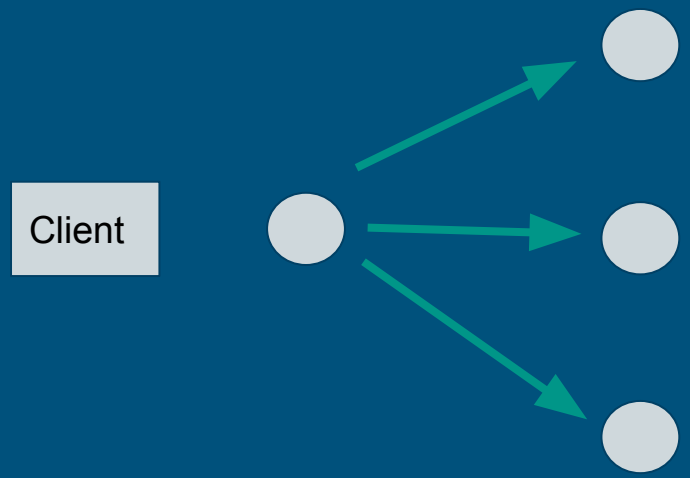


Prepare and promise 1



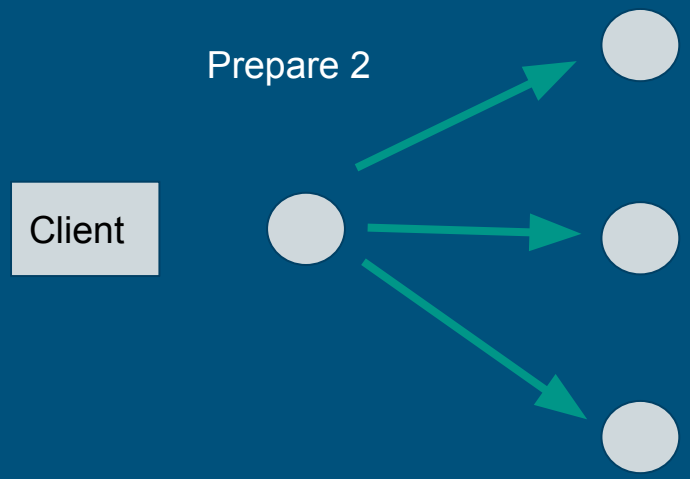
partition	table	promised	accepted	committed

Prepare and promise 1



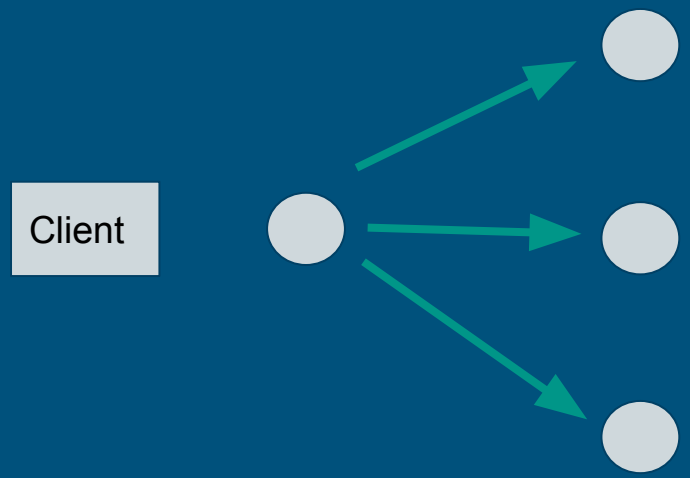
partition	table	promised	accepted	committed
A	vouchers	1		

Prepare and promise 2



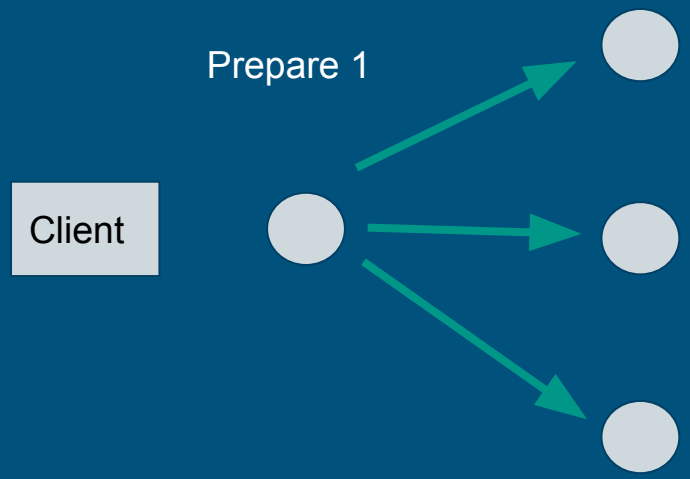
partition	table	promised	accepted	committed
A	vouchers	1		

Prepare and promise 2



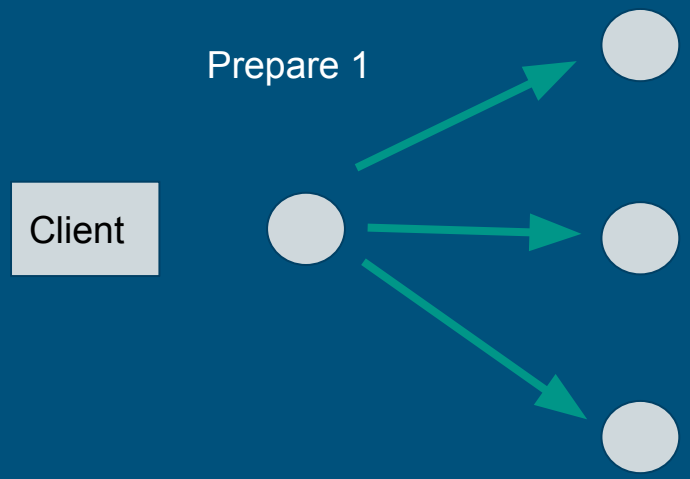
partition	table	promised	accepted	committed
A	vouchers	2		

Prepare and promise - rejection



partition	table	promised	accepted	committed
A	vouchers	2		

Prepare and promise - rejection



partition	table	promised	accepted	committed
A	vouchers	2		

Rejected

- ClientRequest.CASWrite.contentions

Prepare and promise - trace

Parsing insert into users (user_name, password, email) values ('chbatey', 'chrisrocks', 'christopher.batey@gmail.com') if not exists; [SharedPool-Worker-1] | 2016-08-22 12:38:44.132000 | 127.0.0.1 | 1125

Sending PAXOS_PREPARE message to /127.0.0.3 [MessagingService-Outgoing-/127.0.0.3] | 2016-08-22 12:38:44.141000 | 127.0.0.1 | 10414

Sending PAXOS_PREPARE message to /127.0.0.2 [MessagingService-Outgoing-/127.0.0.2] | 2016-08-22 12:38:44.142000 | 127.0.0.1 | 10908

Promising ballot fb282190-685c-11e6-71a2-e0d2d098d5d6 [SharedPool-Worker-1] | 2016-08-22 12:38:44.147000 | 127.0.0.3 | 4325

Prepare and promise - trace

Promising ballot fb282190-685c-11e6-71a2-e0d2d098d5d6 [SharedPool-Worker-1] | 2016-08-22 12:38:44.147000 | 127.0.0.3 | 4325

Promising ballot fb282190-685c-11e6-71a2-e0d2d098d5d6 [SharedPool-Worker-3] | 2016-08-22 12:38:44.166000 | 127.0.0.1 | 35282

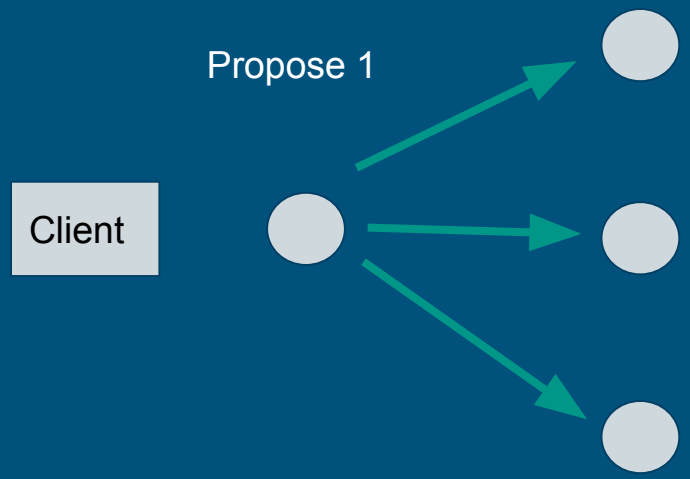
Read

LOCAL_SERIAL => LOCAL_QUORUM

SERIAL => QUORUM

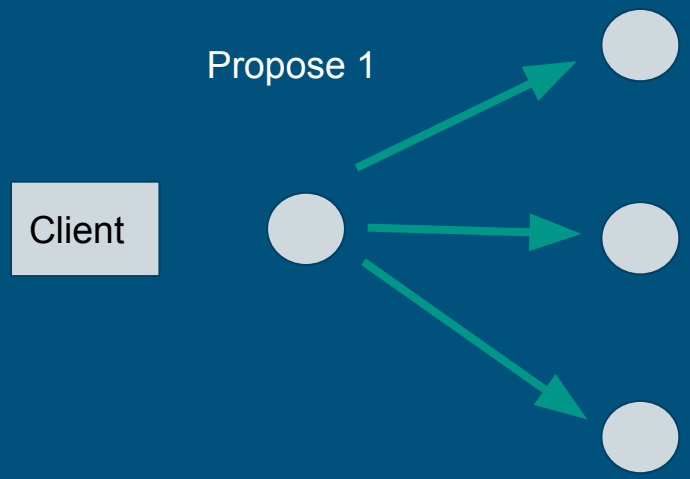
ClinetRequest.CASWrite.conditionNotMet

Propose and accept



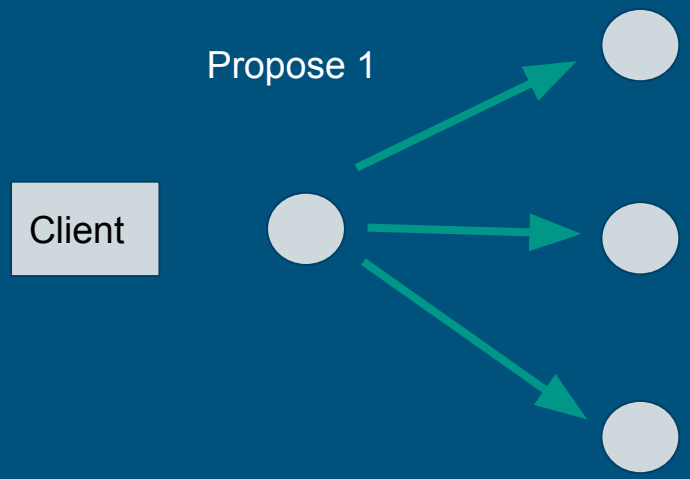
partition	table	promised	accepted	committed
A	vouchers	1		

Propose and accept



partition	table	promised	accepted	committed
A	vouchers	1	1	

Propose and accept - rejection



partition	table	promised	accepted	committed
A	vouchers	2		

Rejected

- ClientRequest.CASWrite.contentions

Propose and accept - trace

Sending PAXOS_PROPOSE message to /127.0.0.2 [MessagingService-Outgoing-/127.0.0.2] | 2016-08-22 12:38:44.196000 | 127.0.0.1 | 65606

Sending PAXOS_PROPOSE message to /127.0.0.1 [MessagingService-Outgoing-/127.0.0.1] | 2016-08-22 12:38:44.196000 | 127.0.0.1 | 65606

PAXOS_PROPOSE message received from /127.0.0.1 [MessagingService-Incoming-/127.0.0.1] | 2016-08-22 12:38:44.197000 | 127.0.0.1 | 65986

Sending PAXOS_PROPOSE message to /127.0.0.3 [MessagingService-Outgoing-/127.0.0.3] | 2016-08-22 12:38:44.197000 | 127.0.0.1 | 66139

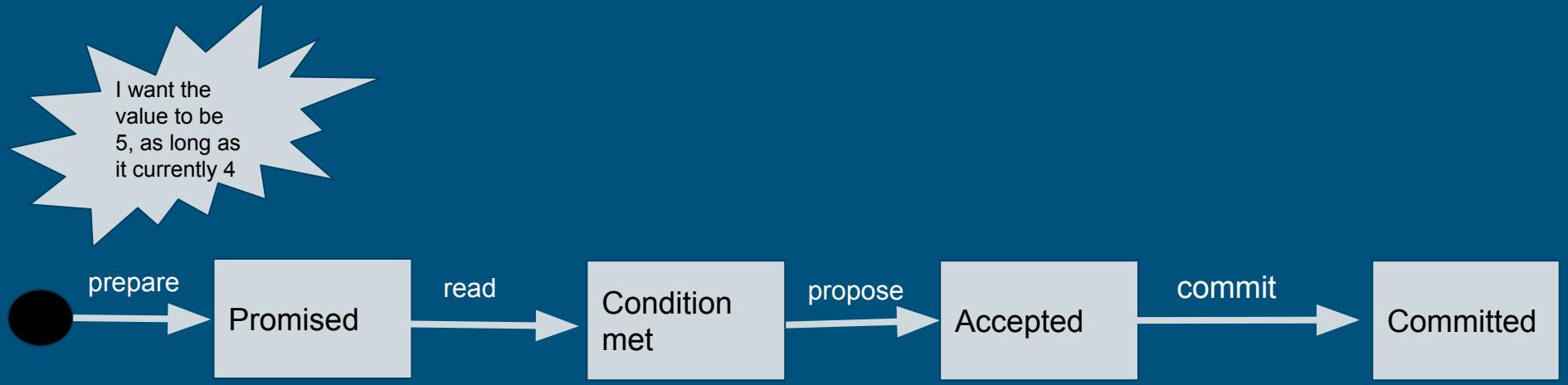
Propose and accept - trace

Accepting proposal Commit(fb282190-685c-11e6-71a2-e0d2d098d5d6, [lwts.users] key=chbatey columns=[[] | [email password]]\n Row: EMPTY | email=christopher.batey@gmail.com, password=chrisrocks) [SharedPool-Worker-2] | 2016-08-22 12:38:44.199000 | 127.0.0.1 | 67804

Commit

- The normal consistency is now used for the commit

Consensus for a partition

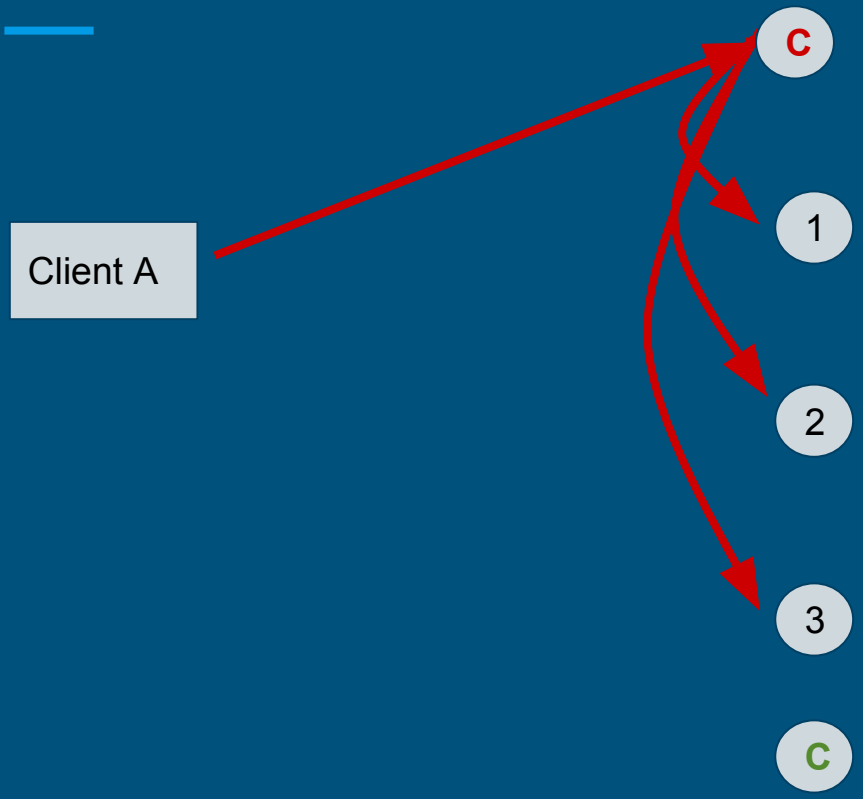


SERIAL reads

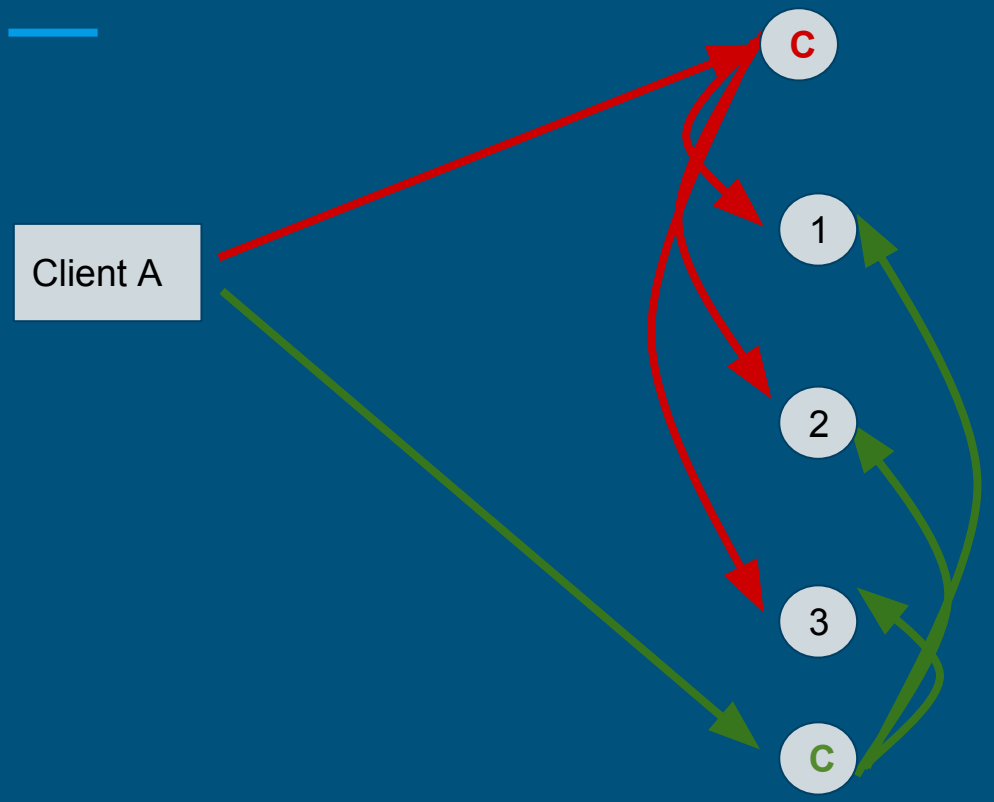
o.a.c.s.StorageProxy.readWithPaxos

- For a single partition
- Runs a prepare and ensures all replicas have the latest commit
- Then runs the read at either Q or LQ

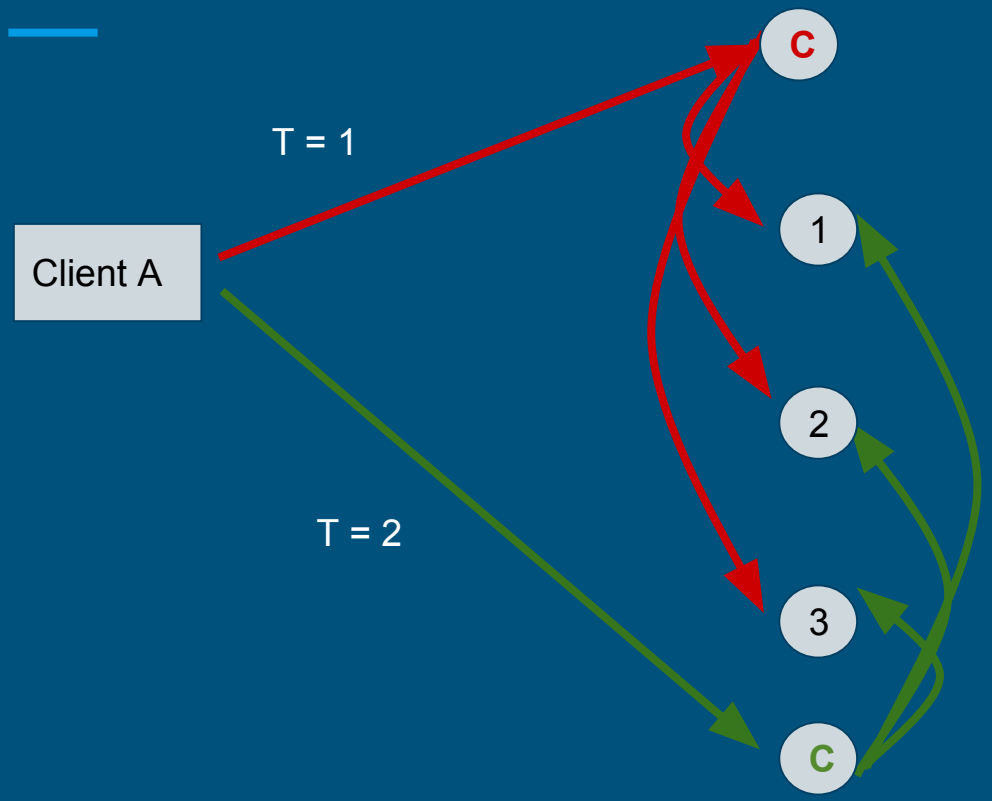
Write timestamps



Write timestamps



Write timestamps



Some numbers

Setup

4 * i2xLarge

RF = 3

10 clients trying to buy 1000 vouchers each - 10k total operations

Contention: all clients buying the same voucher (same partitoin)

No contention: all clients after different vouchers (different partition)

Mutable field

```
CREATE TABLE vouchers_mutable (
    name text PRIMARY KEY,
    sold int
)
```

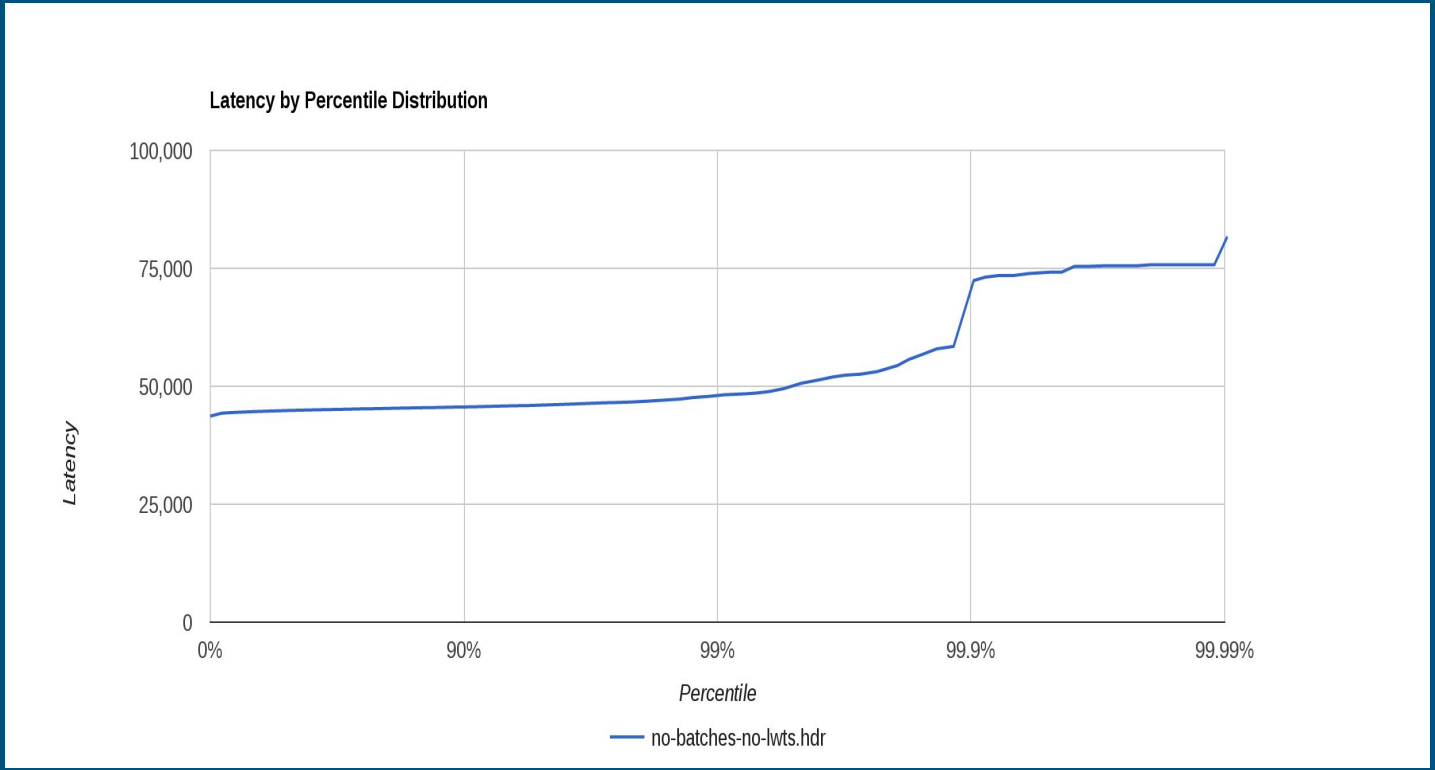
1)

```
UPDATE vouchers_mutable SET sold = 1
WHERE name = 'free tv'
```

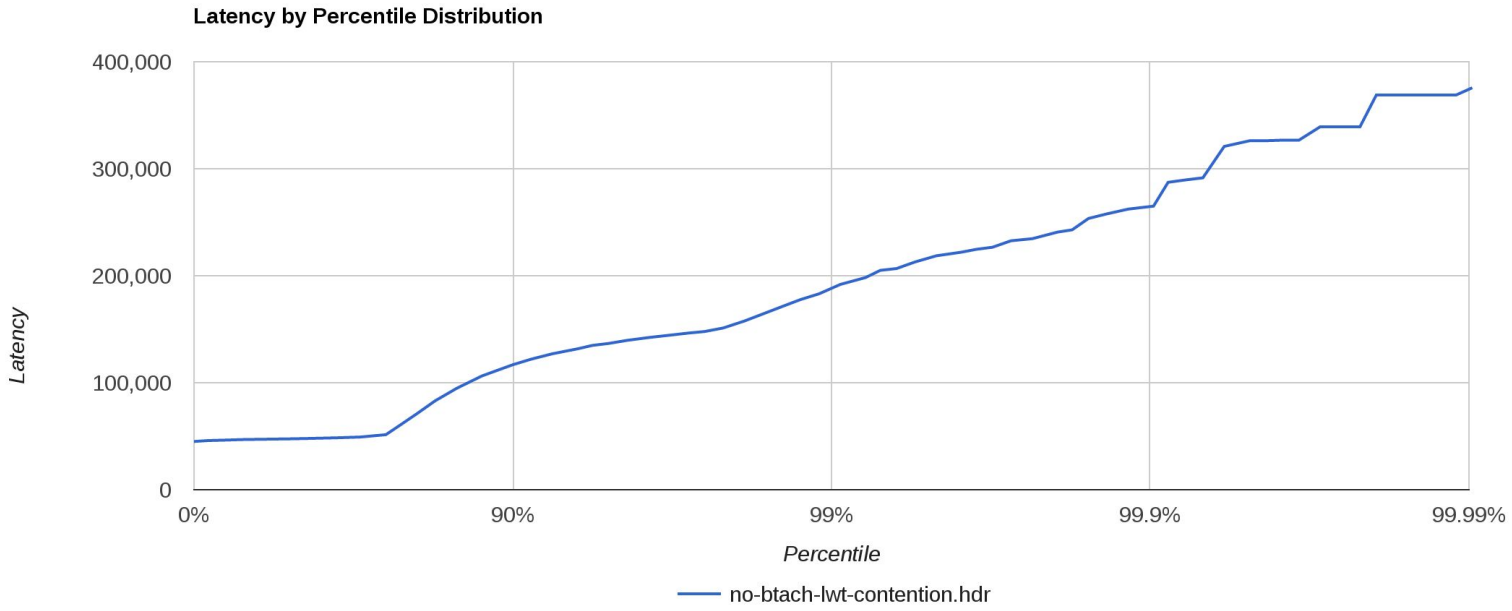
2)

```
UPDATE vouchers_mutable SET sold = 1
WHERE name = 'free tv' IF sold = 0;
```

Histogram


Latency by Percentile Distribution
100,000
75,000
50,000
25,000
0
Latency
0%
90%
99%
99.9%
99.99%
Percentile
no-batches-no-lwts.hdr

Histogram

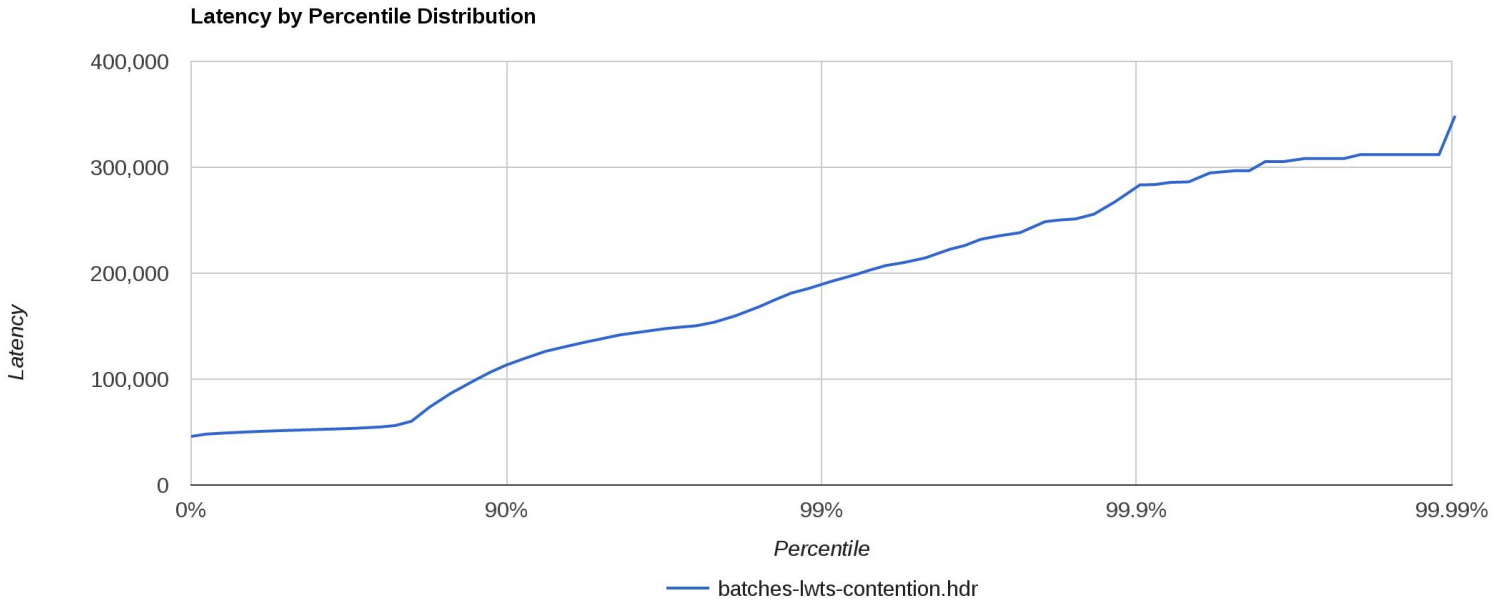

Latency by Percentile Distribution
400,000
300,000
200,000
100,000
0
Latency
0%
90%
99%
99.9%
99.99%
Percentile
no-btach-lwt-contention.hdr

Batches

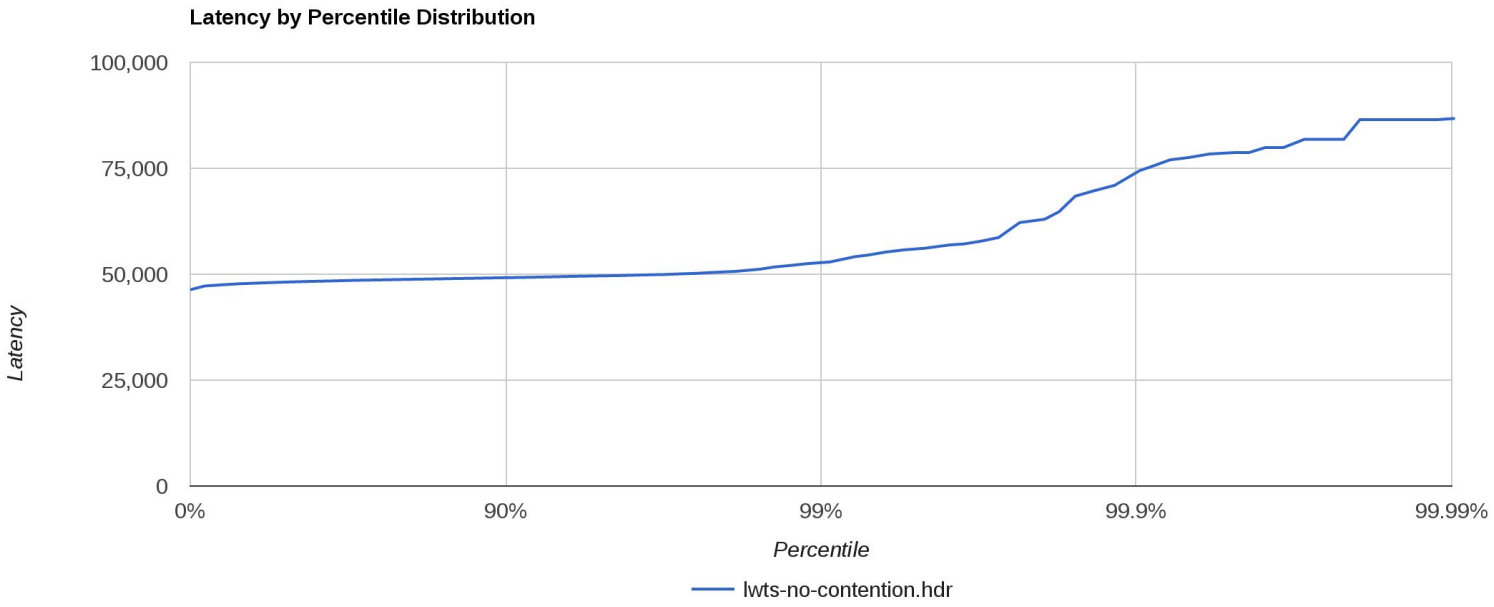
```
CREATE TABLE vouchers (
    name text,
    when timeuuid,
    sold int static,
    who text,
    PRIMARY KEY (name, when)
);
```

```
BEGIN BATCH
    UPDATE vouchers SET sold = 1 WHERE name = 'free tv' IF sold = 0
    INSERT INTO vouchers (name, when, who) VALUES ( 'free tv', now(), 'charlie')
APPLY BATCH ;
```

Histogram


Latency by Percentile Distribution
400,000
300,000
200,000
100,000
0
Latency
0%
90%
99%
99.9%
99.99%
Percentile
batches-lwts-contention.hdr

Histogram


Latency by Percentile Distribution
100,000
75,000
50,000
25,000
0
Latency
0%
90%
99%
99.9%
99.99%
Percentile
lwts-no-contention.hdr

Summary

LWT	Batch	Contention	Incorrect results	99th %ile (milliseconds)
N	N	Y	87% Lost	48
Y	N	Y	0% Lost 1% Unknown 81% CNM	191
Y	N	N	0% Lost 0% Unknown 0% CNM	52
Y	Y	Y	0% Lost <1% Unknown 82% CNM	192

Summary

Summary

- LWTs are expensive
- They are more complex and less mature than the regular read and write path
- Might be a lot easier than bringing in a second technology

Questions?

Christopher Batey

@chbatey

The Last Pickle