CS244b - spanner

Learning goals:

- Putting it all together
  (2PL, Paxos, linearizability, witnesses)
- The power of real-time clocks...
  or how to change your assumptions when
  faced with a hard problem.

operation	latency (ms)		count
	mean	std dev	
all reads	8.7	376.4	21.5B
single-site commit	72.3	112.8	31.2M
multi-site commit	103.0	52.2	32.1M

Table 6: F1-perceived operation latencies measured over the course of 24 hours.

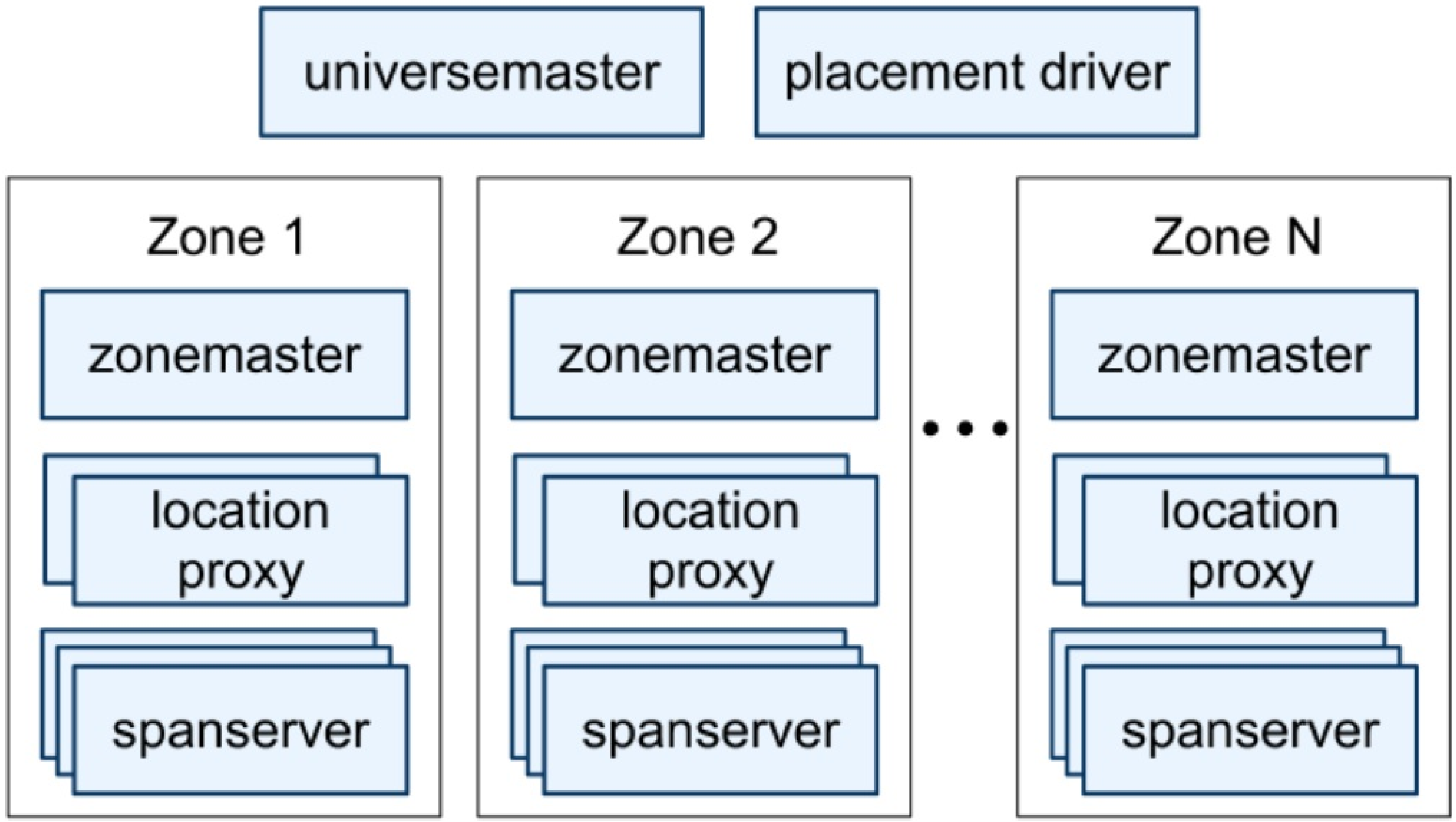
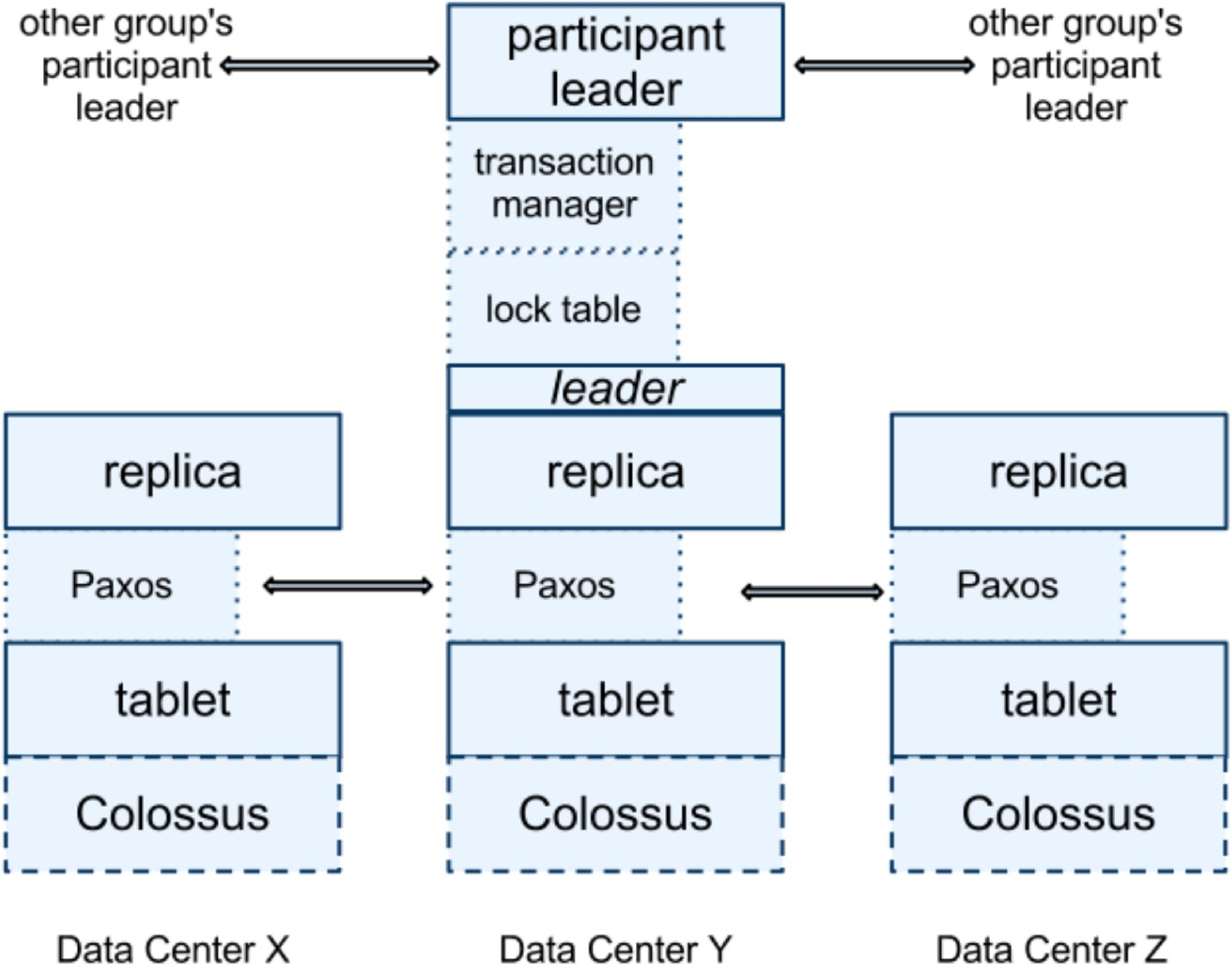


Figure 1: Spanner server organization.

(key, timestamp) -> value

Fig. 2



other group's participant leader
participant leader
other group's participant leader
transaction manager
lock table
leader
replica
replica
replica
Paxos
Paxos
Paxos
tablet
tablet
tablet
Colossus
Colossus
Colossus
Data Center X
Data Center Y
Data Center Z

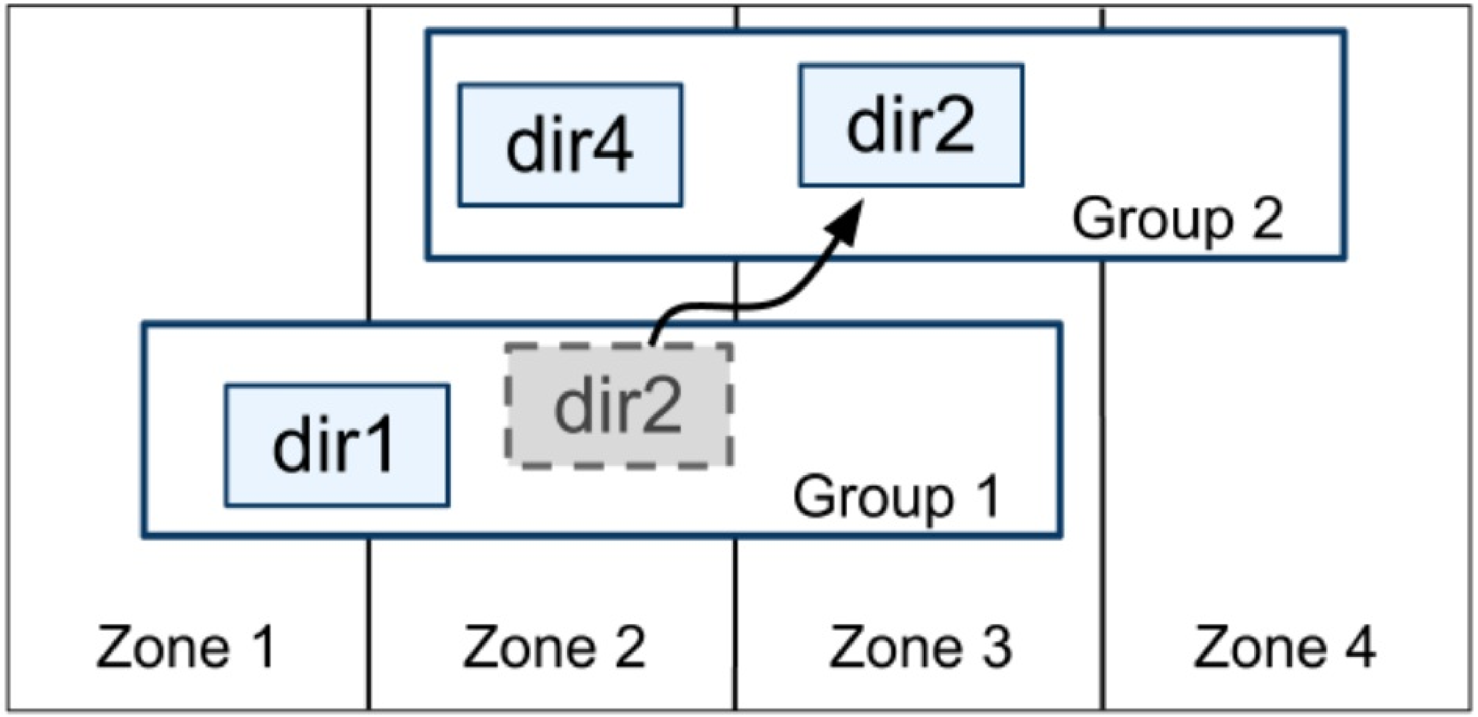


Figure 3: Directories are the unit of data movement between Paxos groups.

- fine grained locking
- placement flexibility

# fragments	# directories
1	>100M
2–4	341
5–9	5336
10–14	232
15–99	34
100–500	7

Table 5: Distribution of directory-fragment counts in F1.

Double log

- Paxos
- Tablet

A directory is also the smallest unit whose geographic-
p. 254 replication properties (or *placement*, for short) can be specified by an application. The design of our placement-specification language separates responsibilities for managing replication configurations. Administrators control two dimensions: the number and types of replicas, and the geographic placement of those replicas. They create a menu of named options in these two dimensions (e.g., *North America, replicated 5 ways with 1 witness*). An application controls how data is replicated, by tagging each database and/or individual directories with a combination of those options. For example, an application might store each end-user's data in its own directory, which would enable user *A*'s data to have three replicas in Europe, and user *B*'s data to have five replicas in North America.

Leases – assumes bounded clock drift

strawman #1

2-phase locking

Paxos leader keep lock table

What goes wrong?

A, B concurrently write to dif. Paxos groups

C, D concurrently reading both

Strawman #2

Acquire lock in multiple Paxos groups

Use 2PC to commit across groups

Drawbacks

Lots of locking

Lots of reads at leader

Transactions might not make sense w. locking

Method	Returns
TT.now()	*TTinterval*: [*earliest*, *latest*]
TT.after(t)	true if t has definitely passed
TT.before(t)	true if t has definitely not arrived

Table 1: TrueTime API. The argument t is of type *TTstamp*.

drift $\leq$ 200 µsec / sec 0.02%

A, B – each tx has timestamp

Read-write transaction - 1 Paxos

Client acquires read locks
buffers writes
Send commit request to leader
Leader picks timestamp s
$s >$ previous Paxos writes
has to be in leader's lease term
$s >$ TTnow().latest
Commit to Paxos
Wait until $s <$ TTnow.earliest()

RW transaction to multiple Paxos groups

Client picks coordinator - send req. to leader

Coord. ~~leader~~ client broadcasts VOTE-REQ

send VOTE-COMMIT w. prepare TS, which
> TS of committed
in participant leader's lease term

Coord. will pick s.
- s > TTnow().latest when commit req. received
- s ≥ max(prepare TS)
- s has to be in all leader lease terms

Commit wait

RO transaction

safest : $s = TT.now().latest$

better : 1 Paxos Group

LastTS()

---

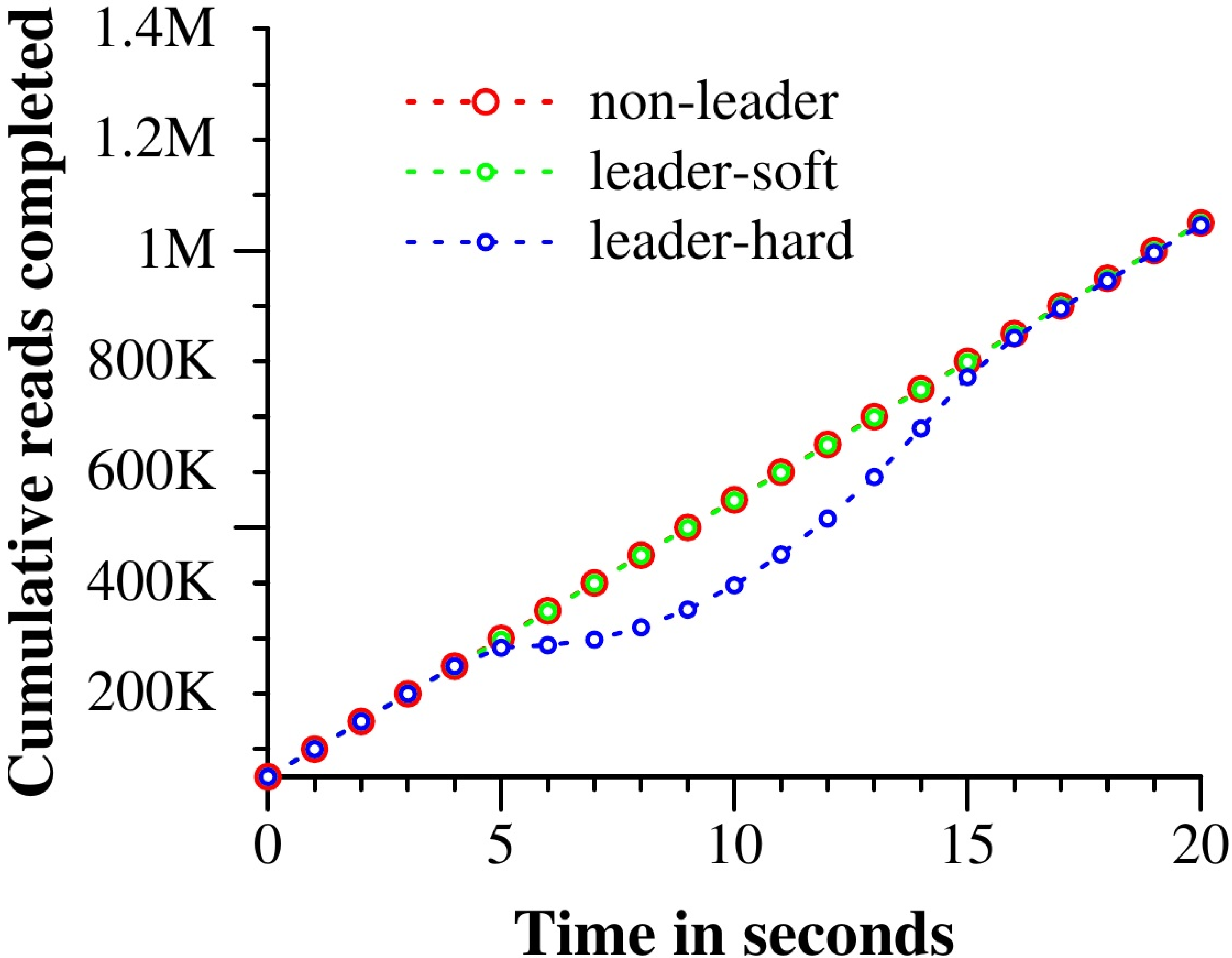
t_{safe} t_{TM}

replicas	latency (ms)			throughput (Kops/sec)		
	write	read-only transaction	snapshot read	write	read-only transaction	snapshot read
1D	9.4±.6	—	—	4.0±.3	—	—
1	14.4±1.0	1.4±.1	1.3±.1	4.1±.05	10.9±.4	13.5±.1
3	13.9±.6	1.3±.1	1.2±.1	2.2±.5	13.8±3.2	38.5±.3
5	14.4±.4	1.4±.05	1.3±.04	2.8±.3	25.3±5.2	50.0±1.1

Table 3: Operation microbenchmarks. Mean and standard deviation over 10 runs. 1D means one replica with commit wait disabled.

operation	latency (ms)		count
	mean	std dev	
all reads	8.7	376.4	21.5B
single-site commit	72.3	112.8	31.2M
multi-site commit	103.0	52.2	32.1M

Table 6: F1-perceived operation latencies measured over the course of 24 hours.



non-leader
leader-soft
leader-hard
Cumulative reads completed
1.4M
1.2M
1M
800K
600K
400K
200K
0
5
10
15
20
Time in seconds