Spanner: Google's Globally-Distributed Database

Cheng Shiyang
Samuel Cherinet

Introduction

What is Spanner?

- Highly Scalable
- Semi-Relational

Features

- Replication Configuration
- Externally consistent read/write
- Globally meaningful commit timestamps
- SQL

True Time API

Cloud Spanner: The best of the relational and NoSQL worlds

	CLOUD SPANNER	TRADITIONAL RELATIONAL	TRADITIONAL NON-RELATIONAL
Schema	✓ **Yes**	✓ Yes	✗ No
SQL	✓ **Yes**	✓ Yes	✗ No
Consistency	✓ **Strong**	✓ Strong	✗ Eventual
Availability	✓ **High**	✗ Failover	✓ High
Scalability	✓ **Horizontal**	✗ Vertical	✓ Horizontal
Replication	✓ **Automatic**	Configurable	Configurable

Implementation : Universe

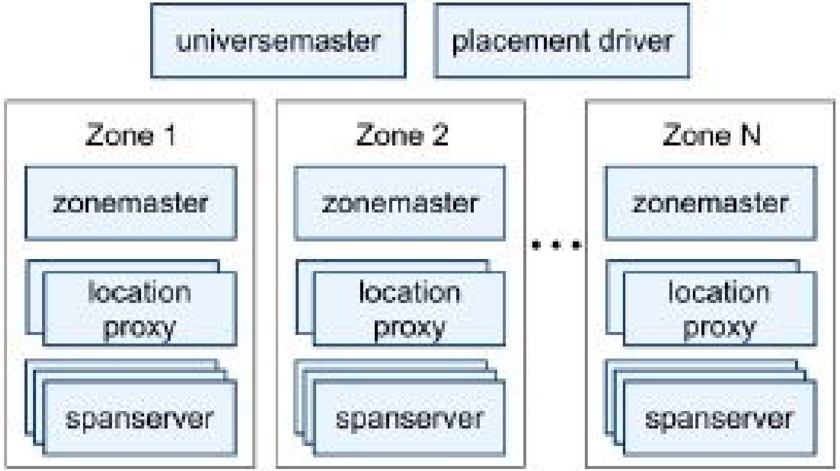


Figure 1: Spanner server organization.

- ❏ Spanserver
- ❏ Zonemaster
- ❏ Location Proxy

Spanserver

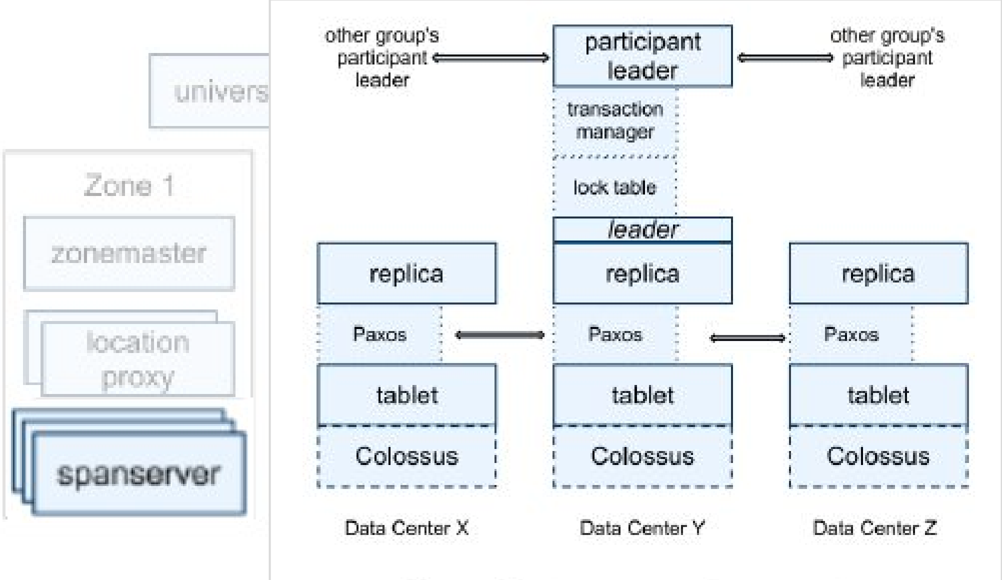


Figure 2: Spanserver software stack.

- ❑ Tablet
- ❑ Distributed Transaction
- ❑ Replication
- ❑ Colossus

Directories

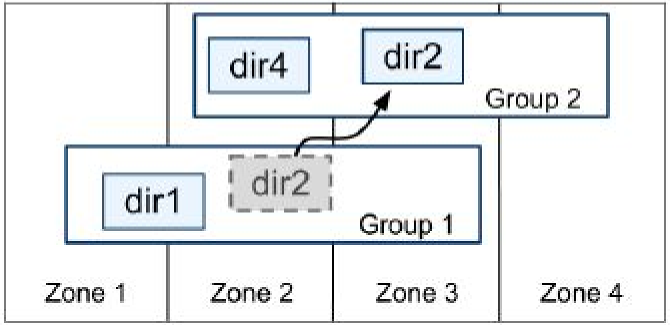


Figure 3: Directories are the unit of data movement between Paxos groups.

- ❑ Prefixing for locality
- ❑ Movedir
- ❑ Administration
  - ❑ number/type of replicas
  - ❑ Geographic placement

Data Model

- Schematized semi-relational tables :
  - Rows must have name (one or more primary key columns)
- Query Language
  - Similar to SQL but has support for protocol-buffer-values fields
- Strongly Typed
- Layered on top of the directory-bucketed key-value mappings supported by the implementation

Example Physical Layout of Data

SingerId	Fir
1	"Ma
2	"Ca
3	"Ali
4	"Le
5	"Da

Singers(1)
Albums(1, 1)
Albums(1, 2)
Singers(2)
Albums(2, 1)
Albums(2, 2)
Albums(2, 3)

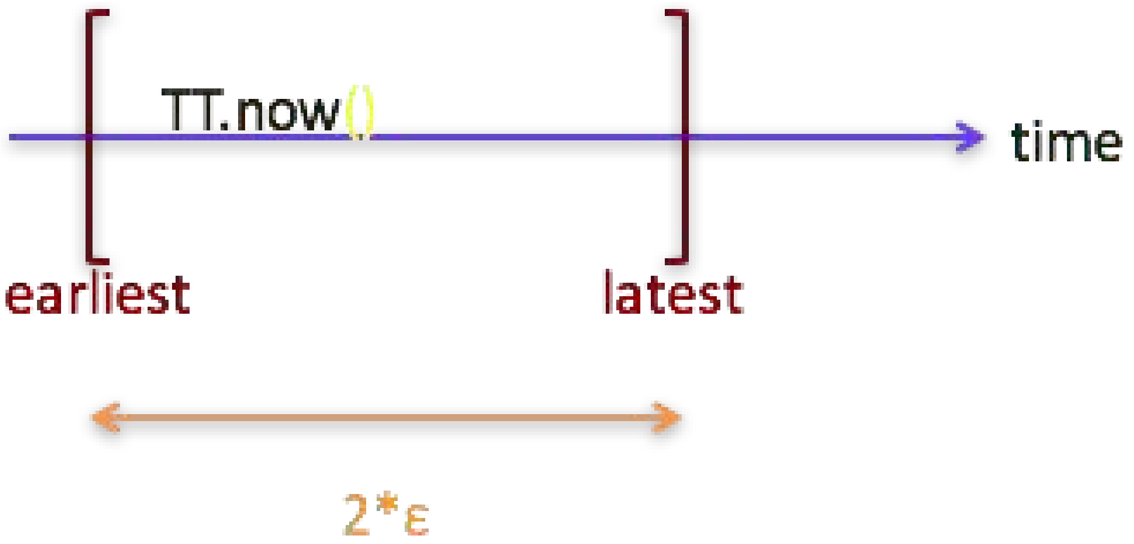
Singers(1)	"Marc"	"Richards"	<Bytes>		
Albums(1, 1)				"Total Junk"	
Albums(1, 2)				"Go, Go, Go"	
Songs(1, 2, 1)					"42"
Songs(1, 2, 2)					"Nothing Is The Same"
Singers(2)	"Catalina"	"Smith"	<Bytes>		
Albums(2, 1)				"Green"	
Songs(2, 1, 1)					"Let's Get Back Together"
Songs(2, 1, 2)					"Starting Again"
Songs(2, 1, 3)					"I Knew You Were Magic"
Albums(2, 2)				"Forever Hold Your Peace"	
Albums(2, 3)				"Terrified"	
Songs(2, 3, 1)					"Fight Story"

Predecessor/Other Data Models

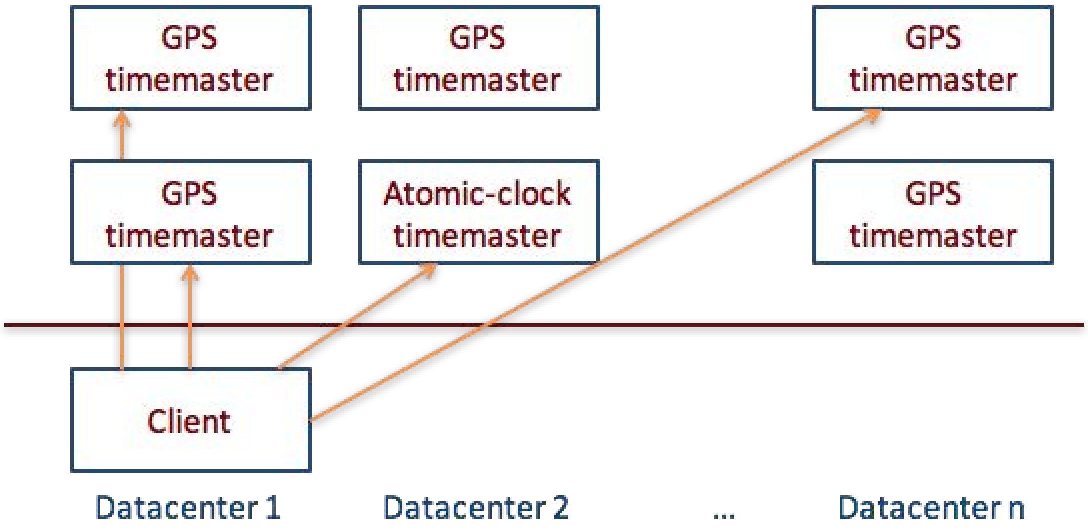
- **Big Table**
- **Megastore**
  - Over 300 application use it in google
  - Relative low performance
  - Synchronous replication

True Time

True Time APIs:
TT.now() //[earliest, latest]
TT.after(t)
TT.before(t)



True Time Architecture



Compute reference [earliest, latest] = now ± ε

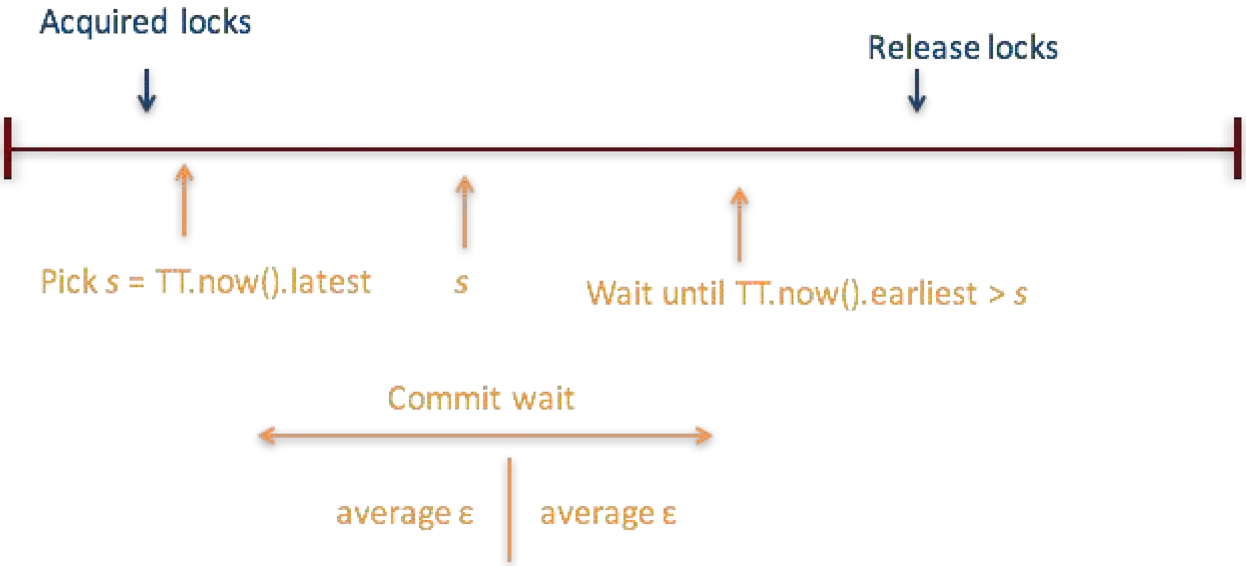
Marzullo’s Algorithm to detect and reject liars.

External Consistency

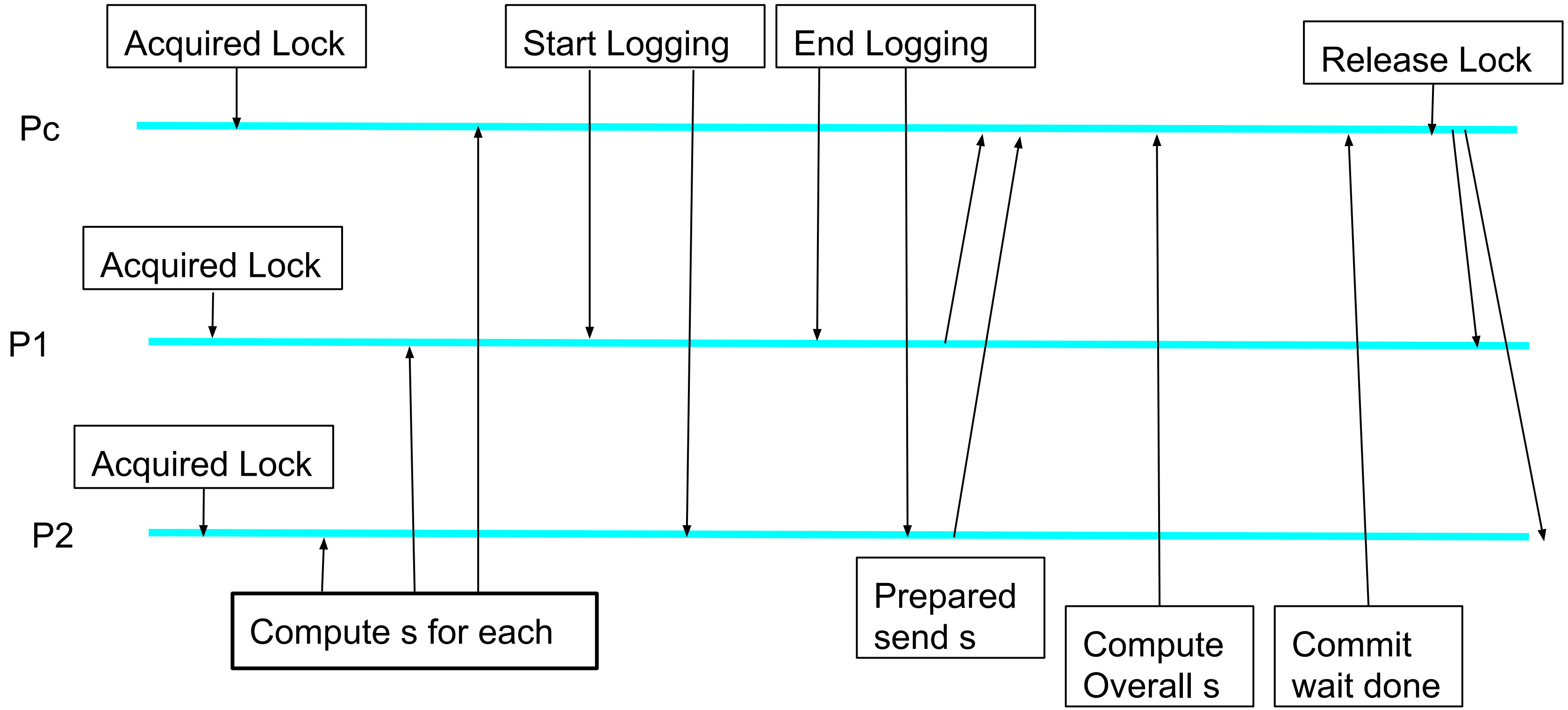
Transaction:T1, T2
If T1 commits before T2’s

T1’s ts < T2’s ts

Timestamps and TrueTime



Commit Wait and 2-Phase Commit



Read-Write Transaction

- Writes occur in a transaction are buffered at the client until commit
- Read transaction will acquires read locks and then read most recent data
- After read transaction client will apply two phase-commit
- A non-coordinator-participant leader first acquires write locks, choose a timestamp then logs the prepare record
- Coordinator participant first acquire the lock, skip the prepare phase and log the record
- Before allowing any coordinator commit, it will wait for TT.after(s)
- Any coordinator commit and release the lock

Read-Only Transaction

- Define LastTS() to be the timestamp of the last committed
- In single-Paxos group, If there is no prepare transactions, the assignment s = LastTS()
- In multi-Paxos group, assign s = TT.now().latest

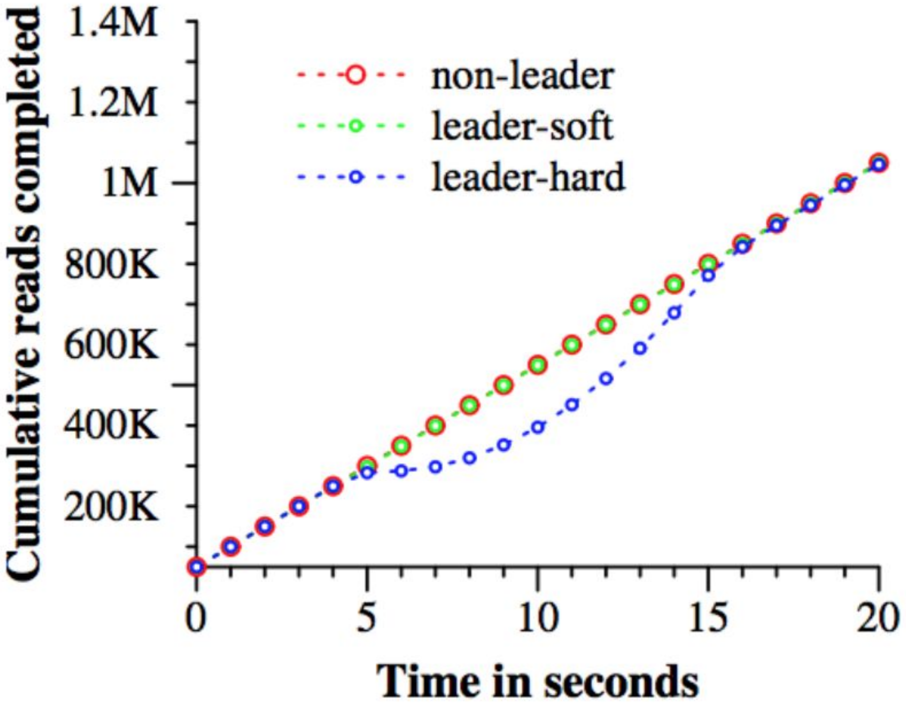
Schema-Change Transaction

- Assign a timestamp in future, which registered in the prepare timestamp at time t
- Reads and writes may proceed if their timestamp procede t
- Reads and writes must block behind the schema-change transaction if their timestamp are after t

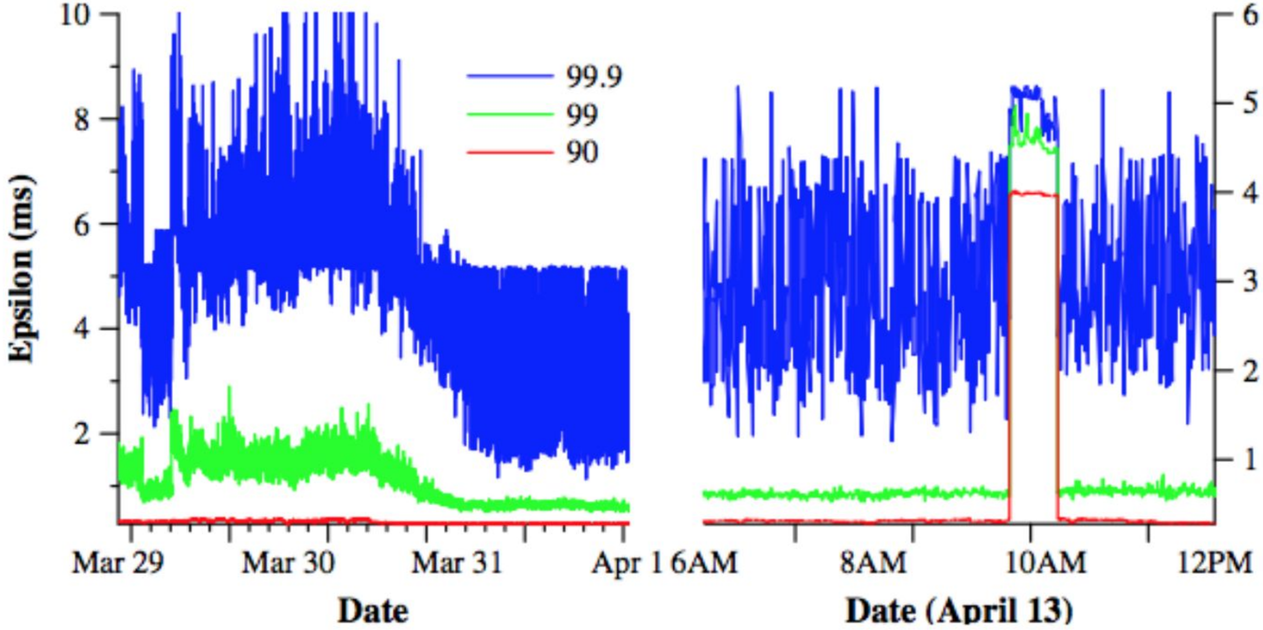
Evaluation - Two phase commit Scalability

participants	latency (ms)	
	mean	99th percentile
1	17.0 ±1.4	75.0 ±34.9
2	24.5 ±2.5	87.6 ±35.9
5	31.5 ±6.2	104.5 ±52.2
10	30.0 ±3.7	95.6 ±25.4
25	35.5 ±5.6	100.4 ±42.7
50	42.7 ±4.1	93.7 ±22.9
100	71.4 ±7.6	131.2 ±17.6
200	150.5 ±11.0	320.3 ±35.1

Evaluation - Availability



Evaluation - True Time



Question?