# Spanner

Vladimír Míč
Filip Nálepa

# Outline

- What is Spanner
- Data model
- Data distribution
- Architecture
- Schema Example
- Transactions
- TrueTime API
- Future work

# What is Spanner

- Globally-distributed database
  - Datacenters all over the world
- Made by Google
- Successor of Bigtable
- Semi-relational database
  - Each table has a primary key

# Data model (1)

- Scales up to:
  - Millions of machines
  - Trillions of database rows
- Query language
  - SQL-based
- Application creates one or more database in a universe

# Data model (2)

- Data
  - Stored in semi-relational tables
  - Versioned (time-stamps)
    - Multi-versioned database
  - Distributed file system Colossus
    - The successor to Google File System
- Mapping
  - (key, timestamp) → value

# Global Distribution

- Replication used for:
  - Global availability
  - Geographic locality
- Data moved dynamically
  - Even between datacenters
  - To balance load
  - As response to failures
- Applications control their data locality

# Architecture

- Universe – Spanner deployment
  - Running 3 universes
- Zones
  - Physical isolation
  - Unit of administrative deployment
- Zonemaster – one per zone
  - Assigns data to spanservers
- Spanserver – one hundred – several thousand
  - Serves data to clients
- Directory
  - Unit of data placement

# Schema Example

- Hierarchies of tables
  - CREATE TABLE Users

    { uid INT64 NOT NULL, email STRING }

    PRIMARY KEY (uid), DIRECTORY;

  - CREATE TABLE Albums

    { uid INT64 NOT NULL,
    aid INT64 NOT NULL, name STRING}
    PRIMARY KEY (uid, aid),
    INTERLEAVE IN PARENT Users ON DELETE CASCADE;

| | |
|---|---|
| Users (1)<br>Albums (1, 1)<br>Albums (1, 2) | Directory 1 |
| Users (2)<br>Albums (2, 1)<br>Albums (2, 2)<br>Albums (2, 3) | Directory 2 |

# Transactions

- Distributed
- Commit timestamps
  - Serialization order
- Externally-consistent
  - Transaction T1 commits before another transaction T2 starts => T1's commit timestamp is smaller than T2's
- First system to provide the guarantees at global scale
- TrueTime API

# TrueTime API

- Uses GPS and atomic clocks
- Clock uncertainty
  - TT.now() returns TTinterval: [earliest, latest]

# Transaction Types

- Read-write
  - Two-phase locking
- Read-only
  - No locking
  - System-chosen timestamp
- Snapshot reads
  - No locking
  - Client-specified timestamp in the past
- Schema changes

# Future Work

- Secondary indexes
- Performance improvement on complex queries
- Automatic load-based resharding

# Summary

- Spanner – globally distributed semi-relational database
- Versioned data in schematized tables
- Distributed transactions – TrueTime API

# Thank you for your attention