Summary of Spanner: Becoming a SQL System

By S. Xiao Fernández Marín

1 Summary

- Distributed relational-like DBMS, behaviour like a centralised DB system and it is
- Highly available as it also store a lot of data so they need a lot of servers
- ACID transactions and external consistency: Transaction history is serializable in wall-clock commit order, if you see the history as a external observer, the same order you observe the commit
-MVCC (Multiversion concurrency control): Snapshot reads data at the same time
- CA; P?
- Before spanner: Google bogtable-to much work into app, google megastore - did not have great performance (slow w) and google spanner,

Distributes architecture:
universemaster: displays status information about all the zones for interactive debugging
Placement driver: handles automated movement of data across zones on the timescale of minutes, periodically communicates with the spanservers to find data that needs to be moved
Zone X (independent part, networking,...;; 1 or more zones in a datacenter):
- Zonemaster: assigns data to spanservers
- Location proxy: used by clients to locate the spanservers assigned to serve their data
- Spanserver: serve data to clients

Spanserver software stack

row>lsm>column

Consistency
• Explicit read-only or read-write transactions
• Read-only transactions execute at given TrueTime timestamp -> External consistency: Transactions are globally ordered by commit time
– Multiple values kept for each row
• Read-write transactions use strict 2PL
– 2PC on commit
- Effectively session consistency: Read/write transactions are always executed with current timestamp and default for read-only transactions is current timestamp
• Values consistency over availability

Availability
High availability
– In practice better than five-nines
• Multiple replicas
– Paxos group for each replica of a ”tablet”
• Fate sharing
– Application is typically hosted in same location as Spanner
– Only difference in availability is interesting

Partitioning
Private global network
– Privately controlled fiber
– Privately controlled network equipment
• Redundant network links
• Redundant networking equipment
• Largest risk is config error or software bug
• Efficiently CA because P is highly controlled

Query processing

- SQL parsed into relational algebra (AST)
- Optimized by relational algebra transformations
  - Push down selection
  - Unnest subqueries
  - Select access method
- Distributed query plans
  - Run-time adaption to ongoing repartitioning
  - DistributedUnion operator

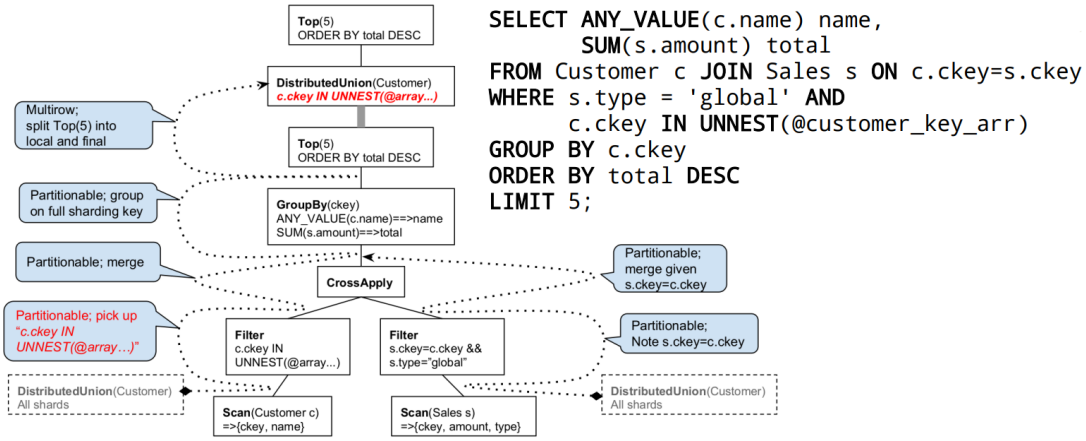

Figure 1: Query processing