CLOUD COMPUTING APPLICATIONS

Cloud Databases – Google Cloud Spanner

Prof. Reza Farivar

Google Cloud Spanner

- Spanner is a distributed data layer that uses optimized sharded Paxos to guarantee consistency even in a system that spans multiple geographic regions
  - Query API is SQL (i.e. SELECT)
  - Insert and updates are done through a specialized GRPC interface
- Two-phase commit to achieve serializability
- TrueTime for external consistency, consistent reads without locking, and consistent snapshots
  - *External > Strong > Weak*

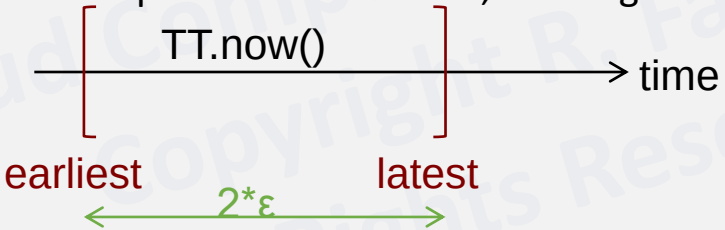
Spanner and CAP

- Is Spanner C+A+P?
  - No, It is CP
  - during (some) partitions, Spanner chooses C and forfeits A
- Availability is in the 5 nines range
  - Is this acceptable to your application?
  - Effectively CA
- Spanner uses the Paxos algorithm as part of its operation to shard (partition) data across hundreds of servers
- 2PC known as the anti-availability protocol
  - because all members must be up for it to work
  - In Spanner, each member is a Paxos group
  - ensures each 2PC “member” is highly available even if some of its Paxos participants are down
- Cloud Spanner provides stale reads, which offer similar performance benefits as eventual consistency but with much stronger consistency guarantees
  - A stale read returns data from an "old" timestamp, which cannot block writes because old versions of data are immutable

TrueTime

- Heavy use of hardware-assisted clock synchronization using GPS clocks and atomic clocks to ensure global consistency
  - avoid communication in a distributed system
  - GPS and Atomic clock have different failure modes
- "Global wall-clock time" with bounded uncertainty
  - ε is worst-case clock divergence
  - Timestamps become intervals, not single values

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



- Consider event e_{now} which invoked tt = TT.now():
  - Guarantee: tt.earliest <= $t_{abs}(e_{now})$ <= tt.latest

Spanner

- TrueTime exposes clock uncertainty
  - Commit wait ensures transactions end after their commit time
  - Read at TT.now.latest()
- Reads dominant, make them lock-free
  - Read-Only Transaction
    - A replica can satisfy a read at a timestamp t if t <= t_{safe}.
  - Snapshot Read, client-provided timestamp
    - read at a particular time in the past
  - Snapshot Read, client-provided bound
- Read-Write Transaction less common
  - Pessimistic, use 2 phase locking
- Globally-distributed database
  - 2PL w/ 2PC over Paxos!

$$t_{safe} = min(t_{safe}^{Paxos}, t_{safe}^{TM})$$

Paxos State Machine Safe Time

Transaction Manager Safe Time

HW-based Time Synchronization

- NTP
  - Software time synchronization with one server
  - Stratum 2
  - Network delays
  - 1/2 to 100 ms accuracy
- Google Spanner
- Microsoft Azure now offers GPS clock synchronization
  - VMICTimeSync provider
  - Precision Time Protocol
  - Stratum 1 devices
    - I.e. direct connection, not through shared network, to a reference time server (Stratum 0)
    - 10 microseconds accuracy
      - For reference, GPS accuracy < 1 us, 95% of the time ≤40 nanoseconds
- Amazon Time Sync Service
  - Chrony vs. NTP