YSQL Follower Reads

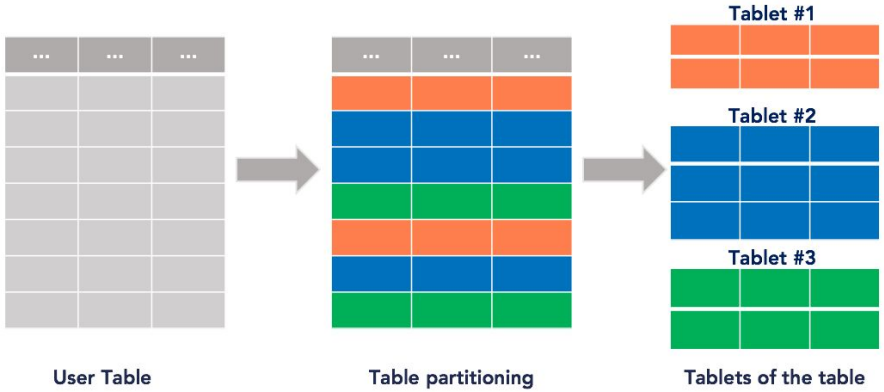
Amitanand Aiyer

Friday, Feb 18 2022

YFTT
YUGABYTEDB
FRIDAY
TECH TALKS

yugabyteDB

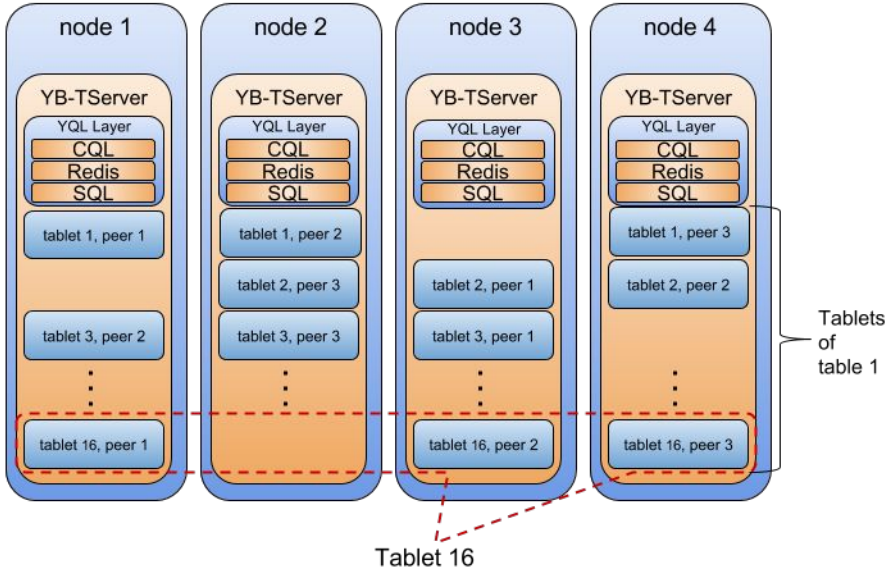
Yugabyte: Data sharding



- Table is partitioned into tablets based on primary key (hash / range sharding).
- Tablet = group of rows.

- https://docs.yugabyte.com/latest/architecture/docdb-sharding/sharding

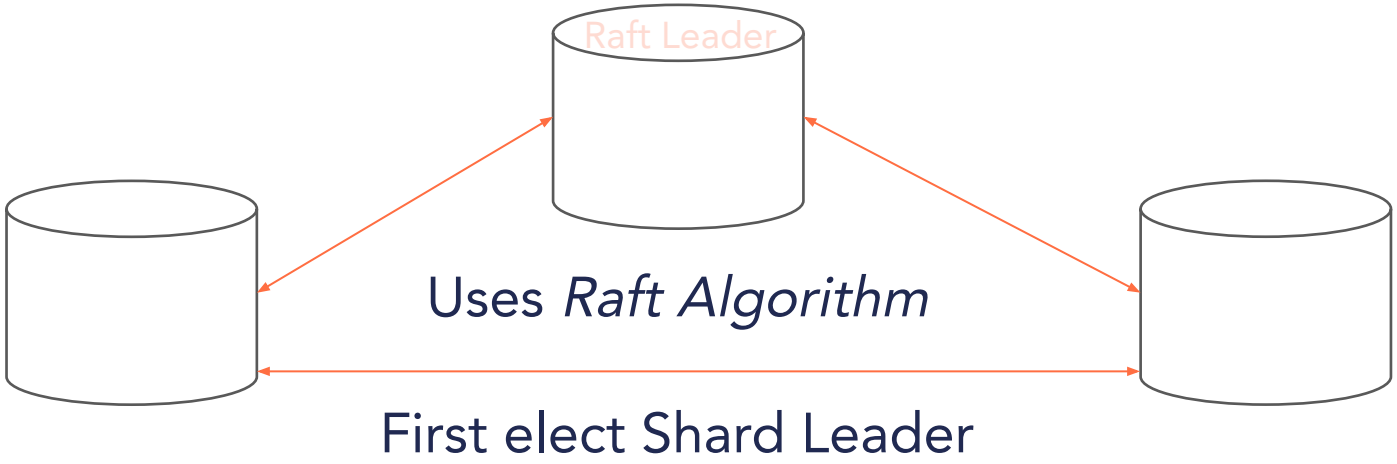
TServer process



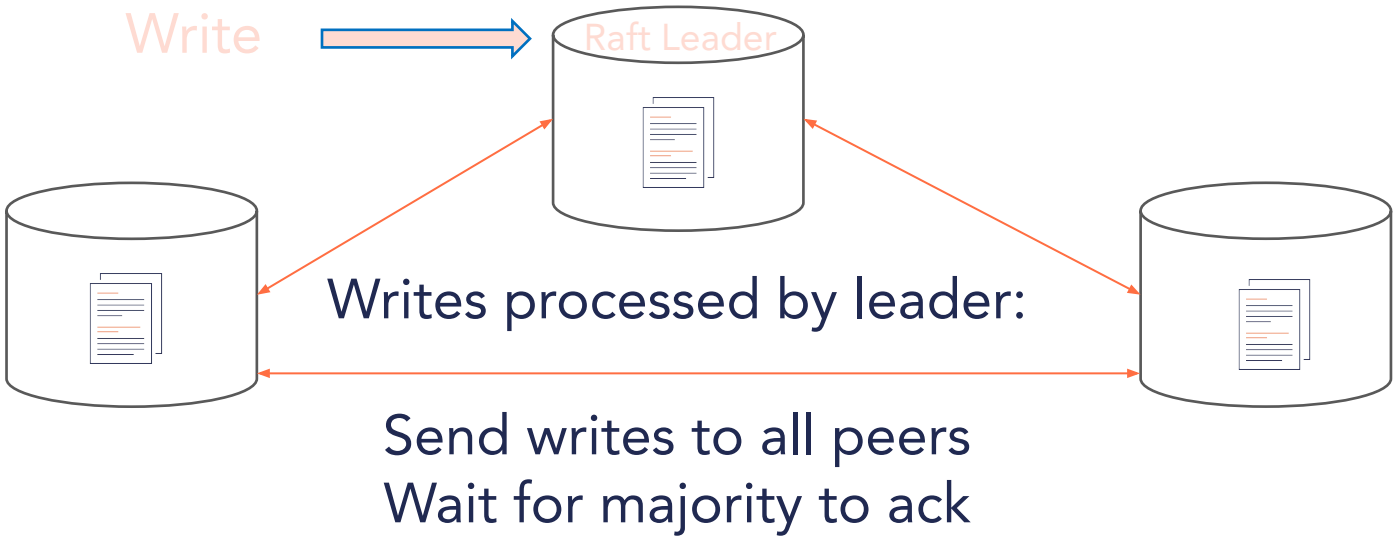
- TServer is responsible for the actual IO for end user requests.
- Tablet peers corresponding to each tablet hosted on different TServers form a Raft group and replicate data between each other.

- https://docs.yugabyte.com/latest/architecture/concepts/yb-tserver

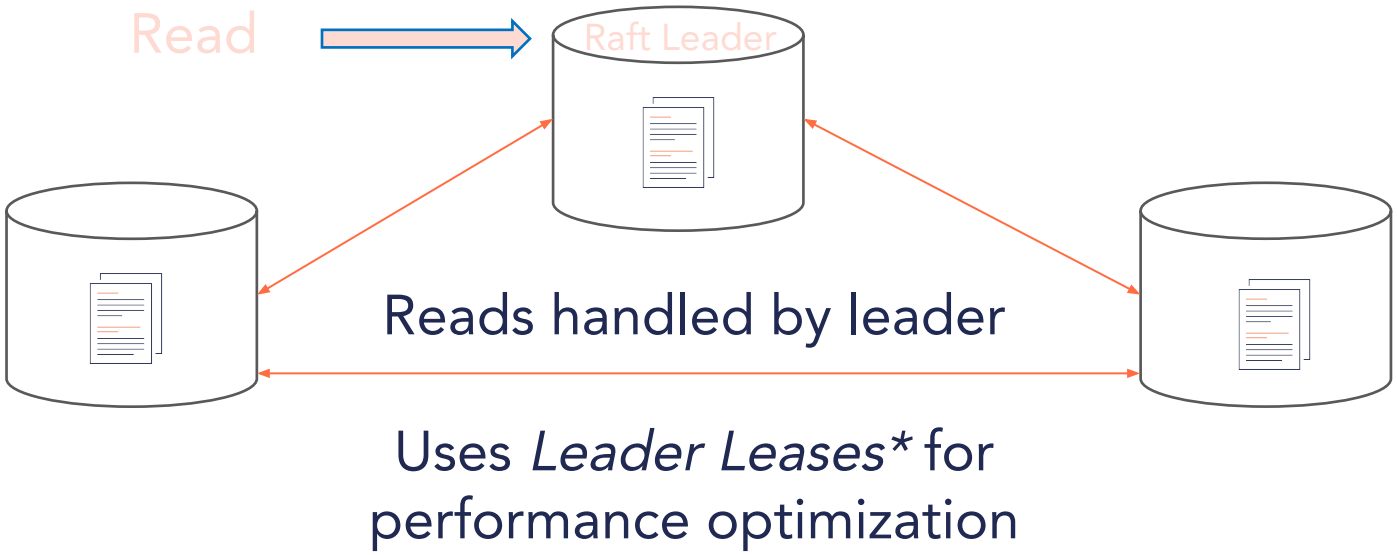
Replication uses a Consensus algorithm



Writes in Raft Consensus



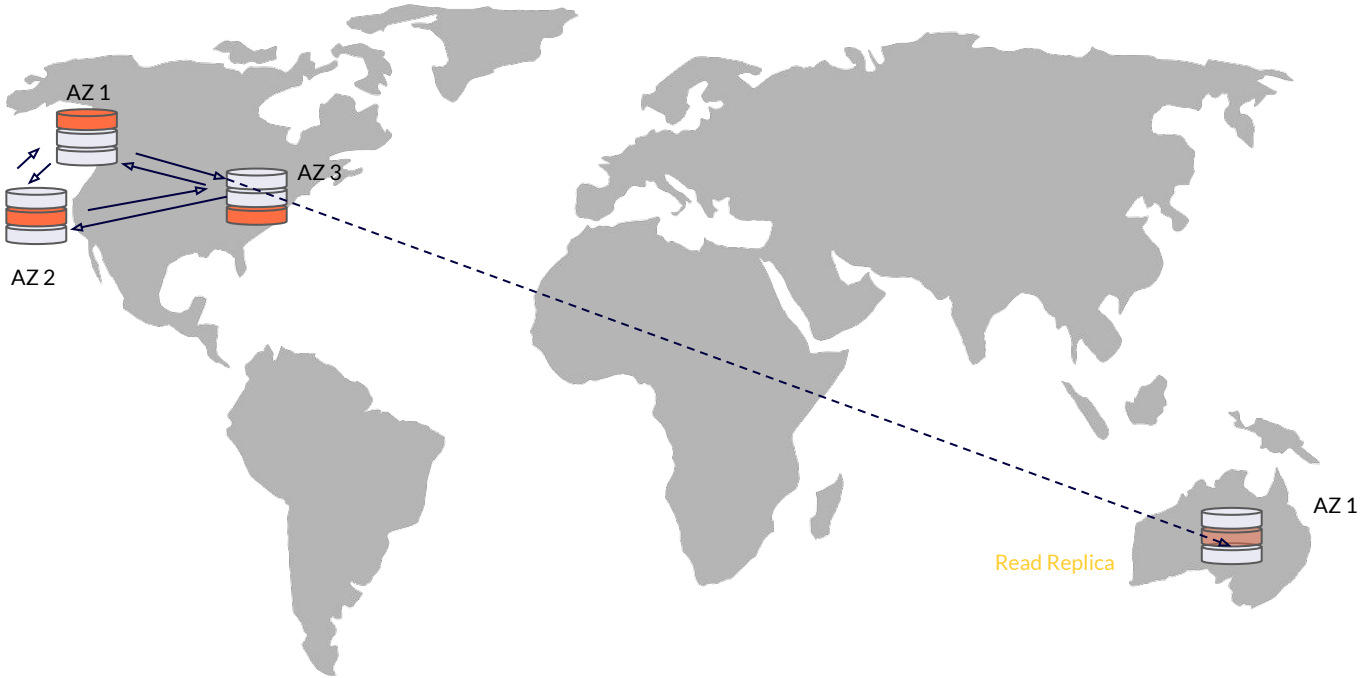
Reads in Raft Consensus



*Leader Leases: https://blog.yugabyte.com/low-latency-reads-in-geo-distributed-sql-with-raft-leader-leases/

Follower Reads

Geo-distributed deployments



- Strong reads
  - Go to the leader
  - High latency
- Follower reads
  - Closest replica
  - Read older values
  - Low latency

Follower Reads

- Only for applications that
  - can tolerate staleness

- Go to the closest replica
- Read at *<current_time> - <delta>*
  - Can be enabled through a pg session variable
  - Any isolation level
  - Use the same timestamp consistently across tablets

Follower Reads: GUC session variables

- Enable follower reads
  - Set ***yb_read_from_followers*** to ***true***
- Set desired staleness (optional)
  - Set ***yb_follower_read_staleness_ms*** accordingly (default ***30s)***
- Start a read-only transaction
  - Txn Block: Begin transaction ***read only***
  - Single stmt: Use ***pg_hint*** to set ***transaction_read_only*** to ***true***
  - Session level: Set ***default_transaction_read_only*** to ***true***

Demo

Thank You

Join us on Slack: yugabyte.com/slack (#yftt channel)

Star us on Github: github.com/yugabyte/yugabyte-db

YFTT YUGABYTEDB FRIDAY TECH TALKS

yugabyteDB