Build better.
Dream bigger.

#RoachFest22

RoachFest22
A CockroachDB Celebration

twitter

CockroachDB @Twitter

Jigar Bhati, Staff Software Engineer

RoachFest22
A CockroachDB Celebration

Success Stories

- Read-heavy use-case
- Write-heavy use-case
- Multi-region use-case

Success Story: Read-heavy use-case

Single-region CRDB Clusters

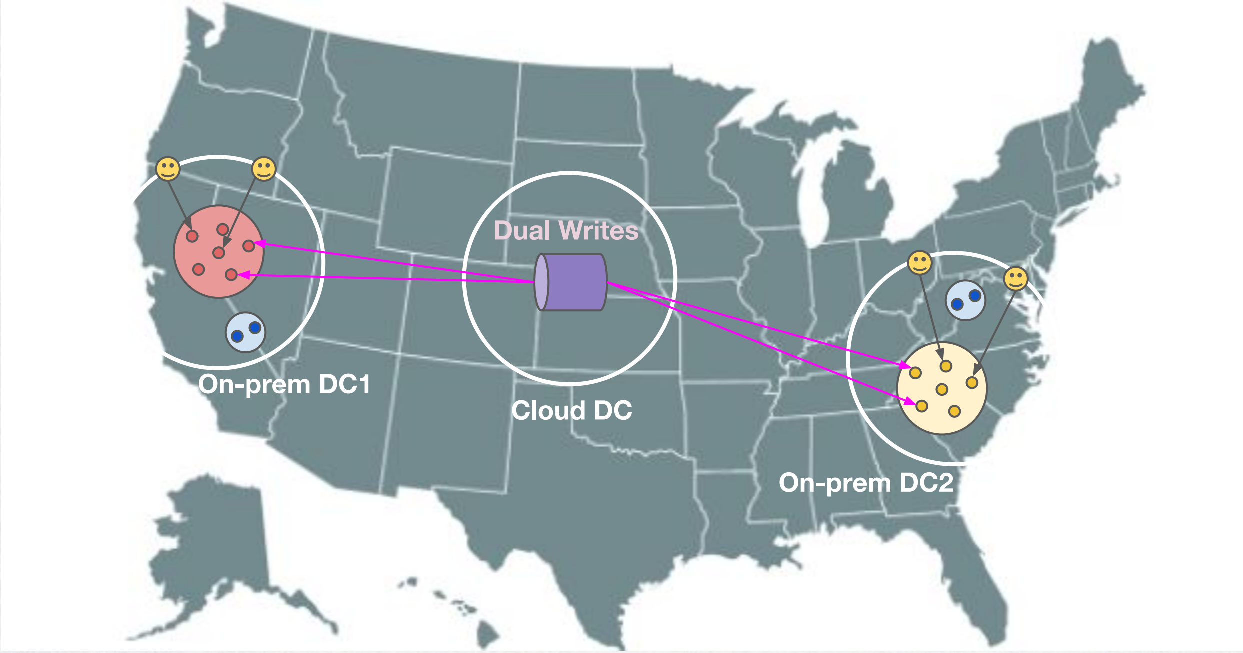
🙂 TwitterDC serving (reads)

🟪 GCP dataflow pipeline (writes)

🟥 CockroachDB west region cluster

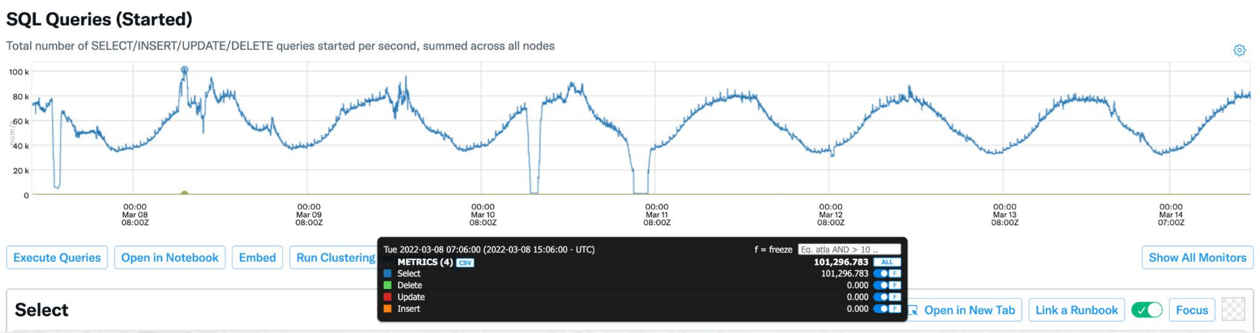
🟨 CockroachDB east region cluster

🟦 CockroachDB system partitions (high available)



Workload

- 36 nodes cluster/DC with 40 cores/node
- ~100k read qps at peak, 100 keys per query



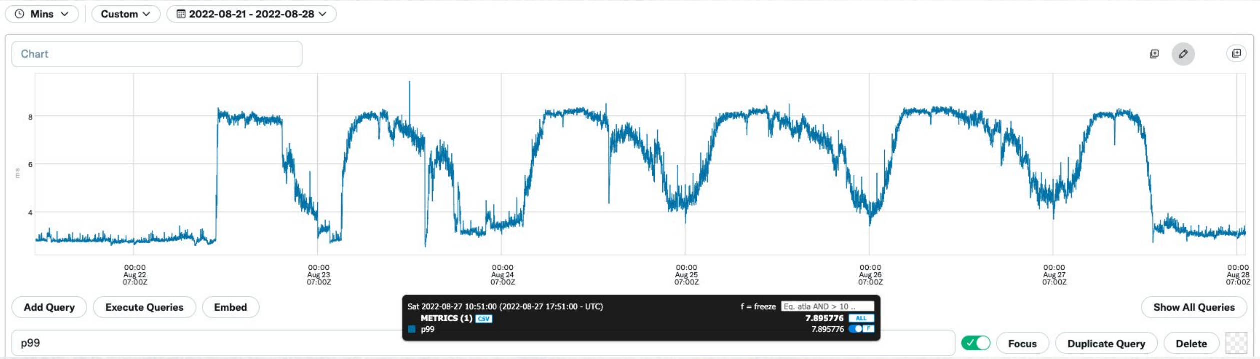
SQL Queries (Started)

Total number of SELECT/INSERT/UPDATE/DELETE queries started per second, summed across all nodes

Tue 2022-03-08 07:06:00 (2022-03-08 15:06:00 - UTC) f = freeze Eq. atla AND > 10 ..

METRICS (4) CSV	101,296.783 ALL
🟦 Select	101,296.783
🟩 Delete	0.000
🟥 Update	0.000
🟧 Insert	0.000

- < 10 ms p99 follower read latency



Sat 2022-08-27 10:51:00 (2022-08-27 17:51:00 - UTC) f = freeze Eq. atla AND > 10 ..

METRICS (1) CSV	7.895776 ALL
🟦 p99	7.895776

Challenges/Learnings

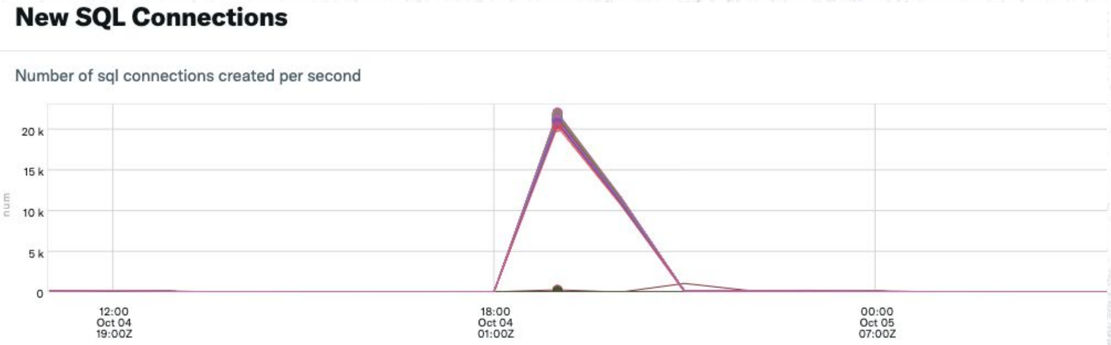
Success Criteria: 20 ms p99 read latency and 99% read SR (Including failover)

- No range splitting allowed (Single range solution)
  - *kv.range_split.by_load_enabled=false* to avoid read fanouts
  - *gc.ttlseconds* = *60* for keeping only fresh data (purge overwritten data)
  - < 500 MB range size (total storage)
- *num_replicas* = *#nodes* (= 36), *num_voters* = *7*
  - Ensures local reads from all gateway nodes.
- Majority reads served out of memory

Challenges/Learnings

Cluster meltdown: Large number of client instances/connections

- Scale: ~15k client instances making a total of ~360k connections to 36 node CRDB cluster
- Problem: CRDB lacks pre-auth connection level rate limiting (ticket).
  - At peak load, request timeout & interrupts lead to connection churn leading to increased connection requests to CRDB.
  - High connection requests caused CPU overload leading to cluster meltdown.
- Fix:
  - Client side rate limiting to restrict max connection requests/s/host.
  - Grace period before connection cancellation.
  - Statement timeout (server-side) - Prevents long running queries.



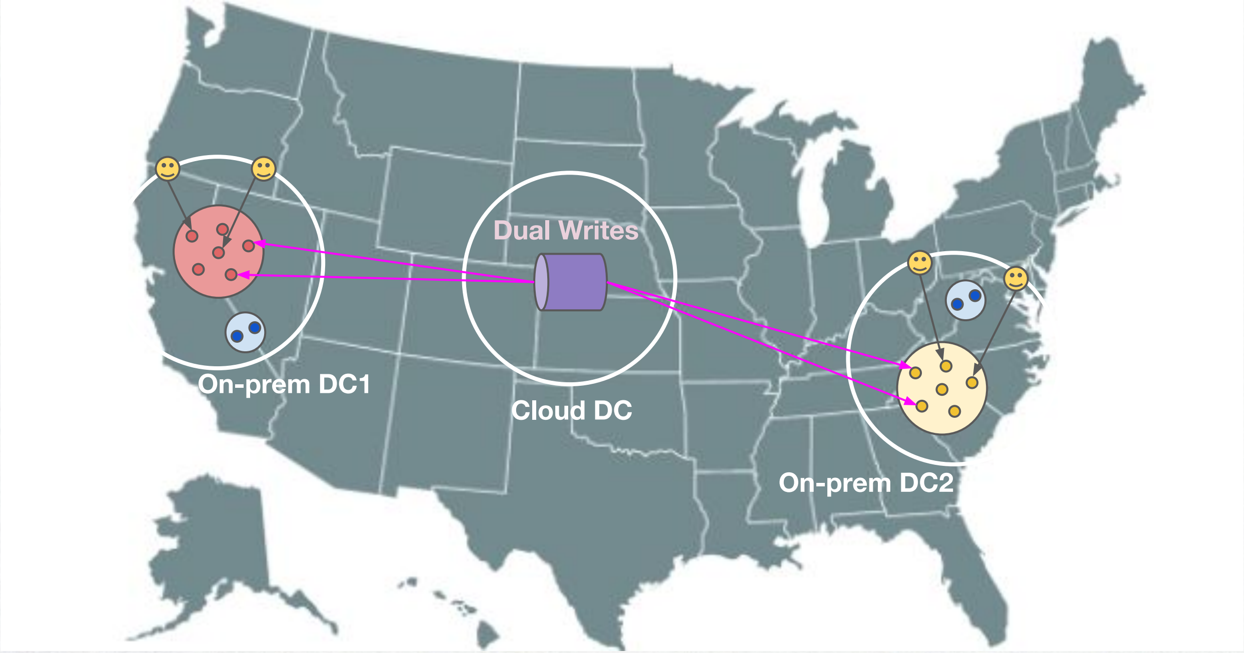
New SQL Connections

Number of sql connections created per second

Success Story: Write-heavy use-case

Single-region CRDB Clusters

- 🙂 TwitterDC serving (reads)
- ▭ GCP dataflow pipeline (writes)
- 🟥 CockroachDB west region cluster
- 🟨 CockroachDB east region cluster
- ▯ CockroachDB system partitions (high available)



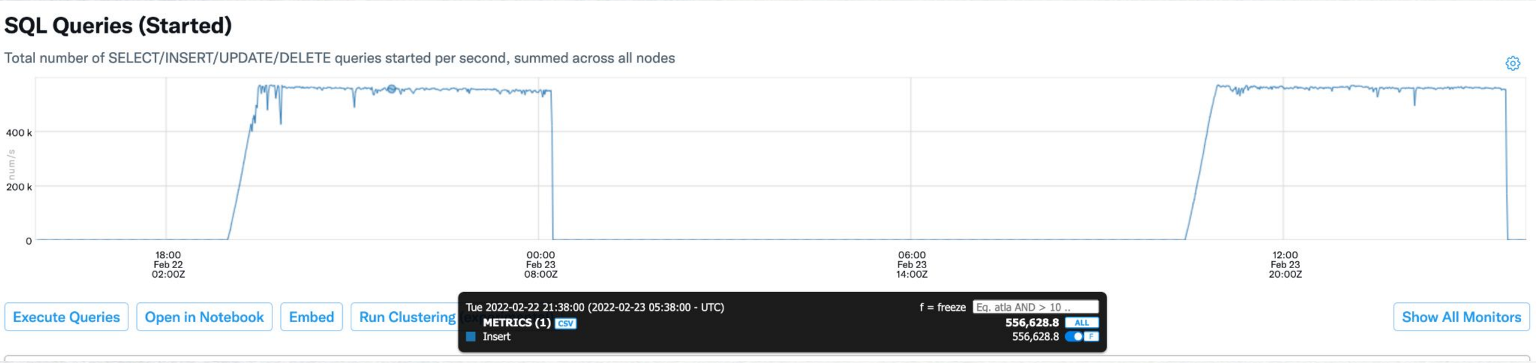
Workload

Setup:

- 64 nodes cluster/DC with 32 cores/node
- 21.2.5 CRDB version
- Default RF, 3 for user tables and 5 for system tables

Throughput Success Criteria:

- ~500k write qps - continuous import within 6 hours
- ~1 TB storage



SQL Queries (Started)

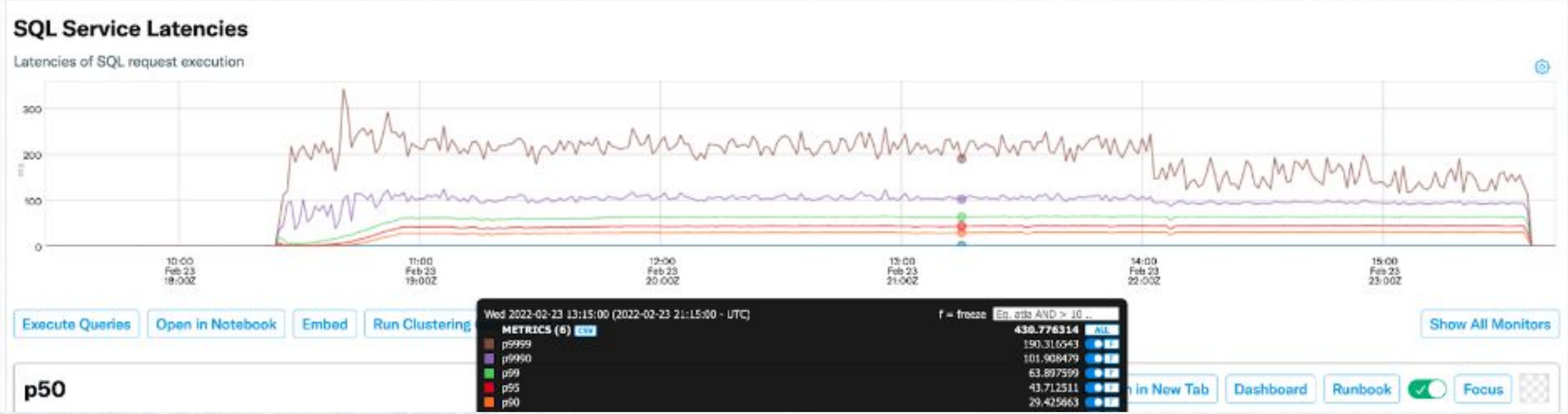
Total number of SELECT/INSERT/UPDATE/DELETE queries started per second, summed across all nodes

Tue 2022-02-22 21:38:00 (2022-02-23 05:38:00 - UTC)
METRICS (1) 556,628.8
Insert 556,628.8

Write Performance

Latency Success Criteria:

- < 100 ms p99 write latency and 99% Success Rate
    - Achieved < 65 ms p99 latency



SQL Service Latencies

Latencies of SQL request execution

```
Wed 2022-02-23 13:15:00 (2022-02-23 21:15:00 - UTC)                f = freeze  Eg. attr AND > 10 ..
METRICS (6) CSV                                                    430.776314  ALL
🟫 p9999                                                           190.316543
🟪 p9990                                                           101.908479
🟩 p99                                                              63.897599
🟥 p95                                                              43.712511
🟧 p90                                                              29.425663
```

Challenges/Learnings

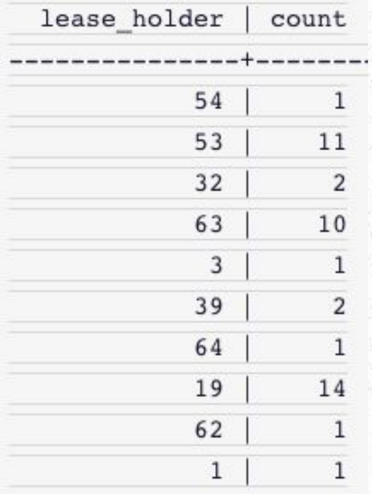
Cold Start Issue

- Problem
    - New table has 1 range to begin with, causing ~90% traffic to be sent to 1 set of replicas (p99: ~5s).

Challenges/Learnings

Cold Start Issue

- What was tried but didn’t work well independently.
  - Tuning range split qps threshold to trigger faster range splits.
  - Tuning snapshot rebalance rate to move ranges faster between nodes.
  - *SPLIT-AT* to pre-split tables into a pre-defined set of ranges
  - *ALTER TABLE $TABLE SCATTER* to force trigger distribution of leases and replicas

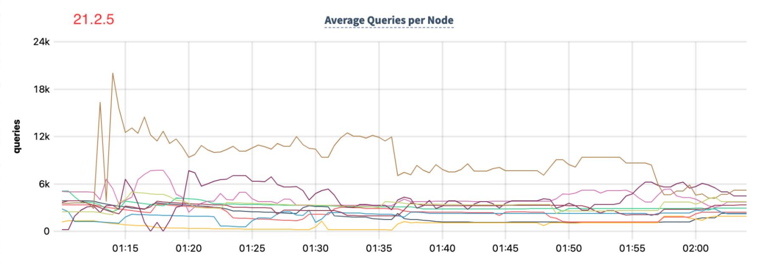


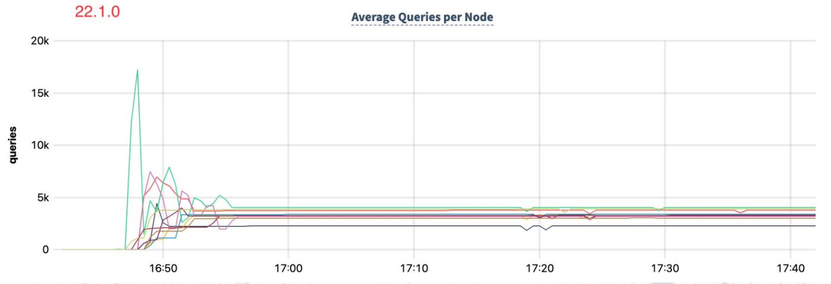
```
  lease_holder | count
---------------+-------
            54 |     1
            53 |    11
            32 |     2
            63 |    10
             3 |     1
            39 |     2
            64 |     1
            19 |    14
            62 |     1
             1 |     1
```

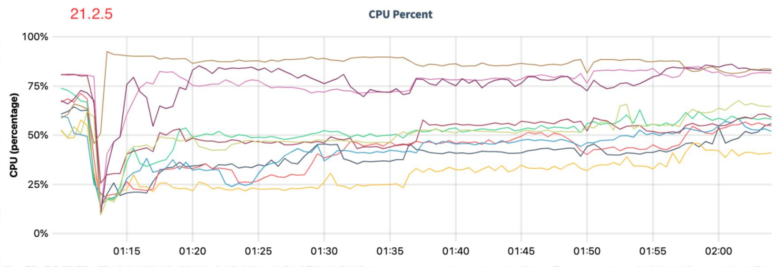
Challenges/Learnings

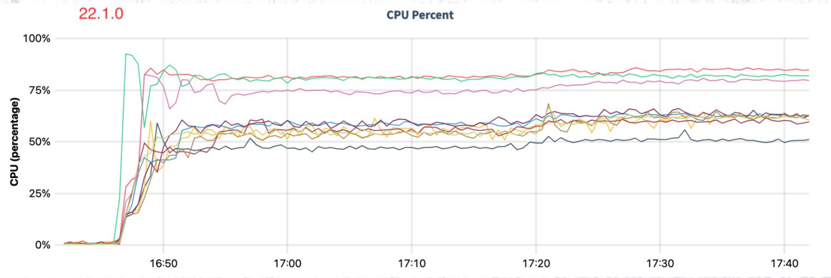
Cold Start Issue

- Possible Solution
  - 22.1.0 CRDB version with fix improved query distribution but still some nodes’ CPU were running hotter than others.
    - Fix: Rebalance ranges to minimize QPS delta among stores





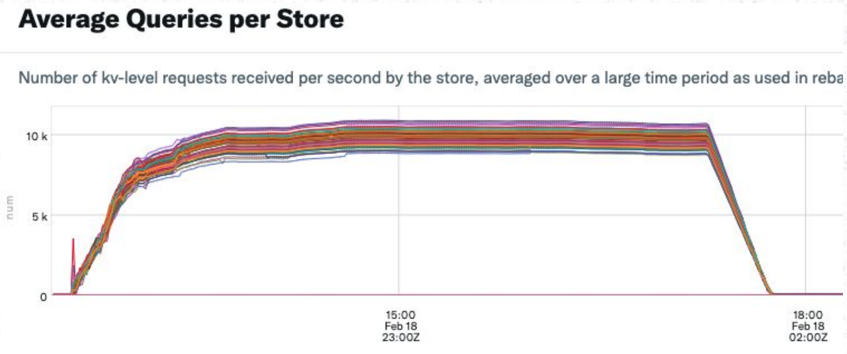




Challenges/Learnings

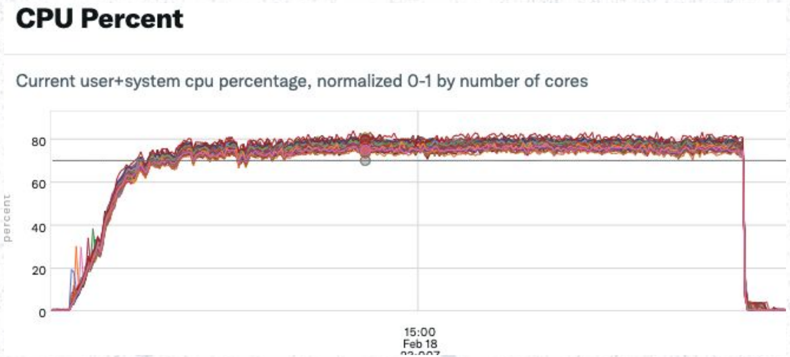
Cold Start Issue

- Chosen Solution
  - Automated slow rampup on client side to give CRDB enough time to do auto range splits and range distribution before taking in 500k wps traffic.



Average Queries per Store

Number of kv-level requests received per second by the store, averaged over a large time period as used in reba



CPU Percent

Current user+system cpu percentage, normalized 0-1 by number of cores

Challenges/Learnings

Liveness Probe disabled - improved cluster stability (CRLabs reco).

Success Story: Multi-region use-case

Multi-region CRDB Cluster

☺ Clients (writer) ☺ Clients (reader)

🟥 CockroachDB red partition

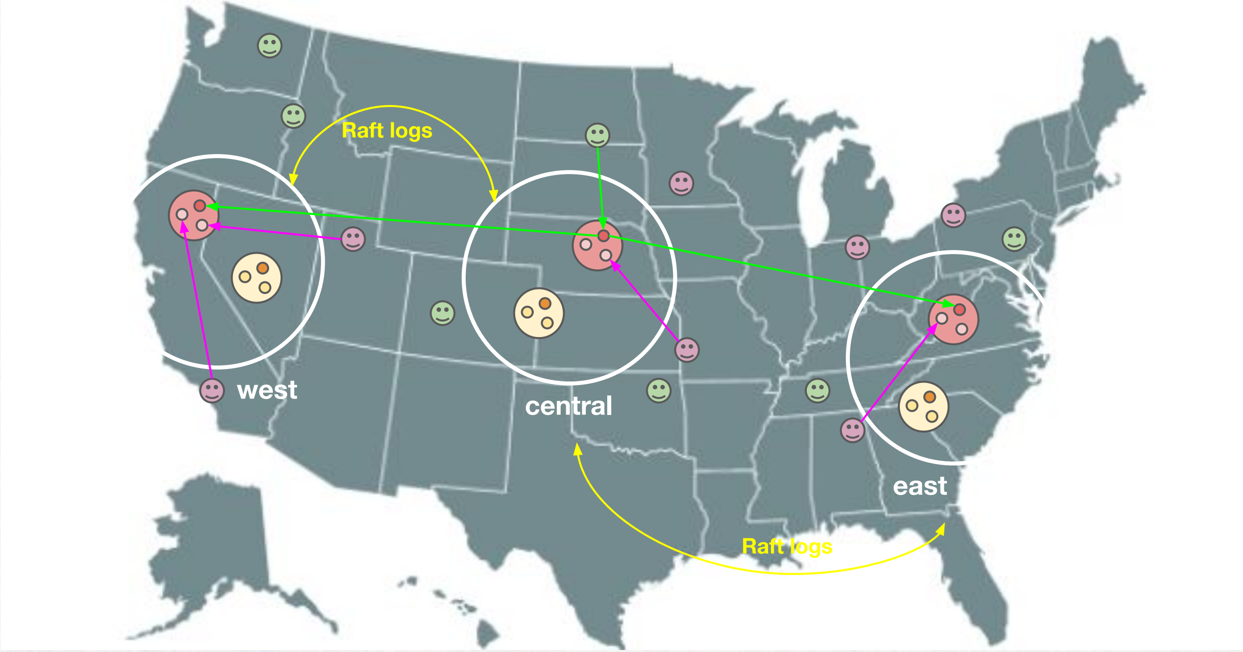
● CockroachDB red replica (voting)

○ CockroachDB red replica (non-voting)

🟨 CockroachDB yellow partition

● CockroachDB yellow replica (voting)

○ CockroachDB yellow replica (non-voting)



Workload

Setup:

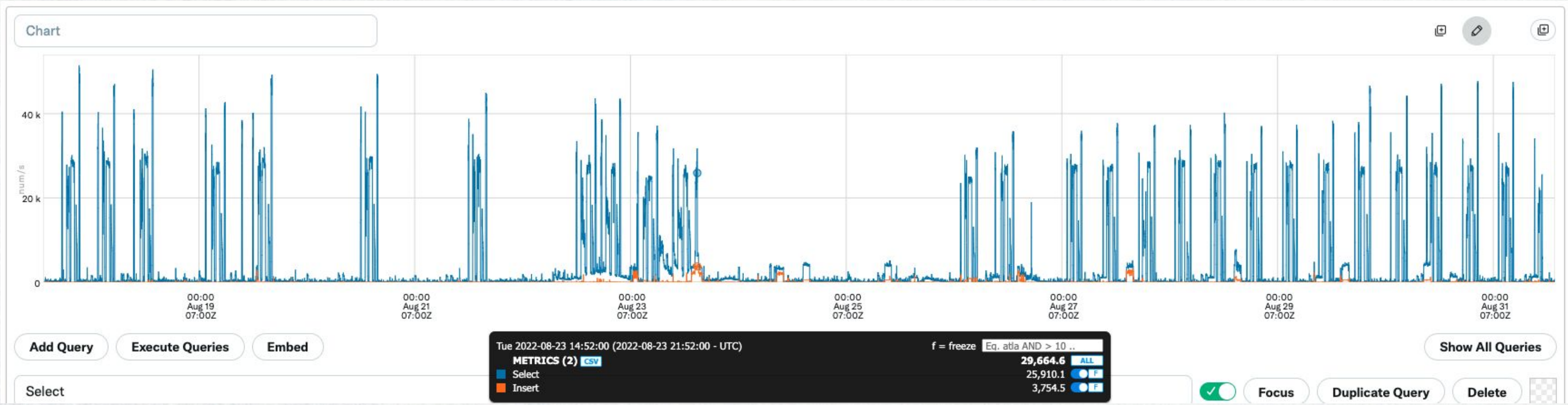
- 36 nodes cluster across 3 DCs with 32 cores/node
- 22.1.3 CRDB version
- RF=7 for both user and system tables

Throughput Success Criteria:

- ~40k reads (Mostly follower)
- ~4k writes

Latency Success Criteria:

- < 200ms p99 reads
- < 500ms p99 writes



Tue 2022-08-23 14:52:00 (2022-08-23 21:52:00 - UTC) f = freeze Eq. atla AND > 10 ..

METRICS (2) CSV 29,664.6 ALL

- 🟦 Select 25,910.1
- 🟧 Insert 3,754.5

Challenges/Learnings

- Connection Tuning
  - Find the right optimal connection count per client instance to not affect the cluster stability.
- Primary Region Setting
  - Default is Follow-the-workload topology
  - Usage: Dominant traffic in one region compared to others and this region is closer to one other region.
- Remove Foreign Key Constraints
  - For inserts, this avoids locking on foreign key referencing table whose leaseholder lives on another node.
- Kubernetes
  - Pod Disruption Budget (PDB) works at a single region. Minimum RF = 7.
  - Alternative Option: Build custom Pod Disruption Coordinator

Thank you!