CASE STUDY

Top U.S. financial software company turns to CockroachDB to improve its application login experience

A financial software company using Oracle GoldenGate for its identity access management layer was drawn to CockroachDB's high availability, low latency, and consistency

Overview

A U.S. financial software company was seeking a new database solution for its customer identity access management (CIAM). The CIAM layer for all their applications was built on Oracle using GoldenGate for high availability, but the company found that GoldenGate couldn't provide the fast, always-available login experience it needed. The team turned to CockroachDB for its resilience to outages, ability to achieve high performance in a multi-region setting, and consistency.

Challenge

One of the largest financial software companies in the U.S. built a single CIAM layer that spanned all of its product lines. CIAM is used when customers create accounts and log into the company's various products. The team deployed their authentication layer on Oracle with GoldenGate on Amazon Web Services (AWS), and they replicated their data across multiple AWS regions to improve fault tolerance. However, the team found that multi-region replication using GoldenGate caused a lag in updates to the data. The lag was anywhere from 30 seconds to up to five minutes long. Authentication often failed for a period of time after a customer created an account, resulting in a poor user experience. Customers might think they had access to their new account, but then their login would be denied. The team decided to address these issues by evaluating new database solutions.

Requirements

The financial software team had several key requirements for their new database. First, they needed to continue replicating their data across multiple regions to ensure resiliency. Their app requires high availability so customers can log into services at all times. Next, the team didn't want to sacrifice performance for availability. They hoped to achieve very high read performance (under 5 milliseconds), because each and every customer login requires reads. They could tolerate slightly lower write performance (several hundred milliseconds), because users generally create accounts only once. The team also hoped to find a database with stronger consistency than the "eventual consistency" they experienced when using cross-region replication with GoldenGate.

Solution: A Multi-Region Deployment with Duplicate Indexes

CockroachDB met all of the requirements that the financial software company had. With synchronous writes and stronger consistency, it eliminated the lag in writes, so customers could access their accounts immediately after creating them.

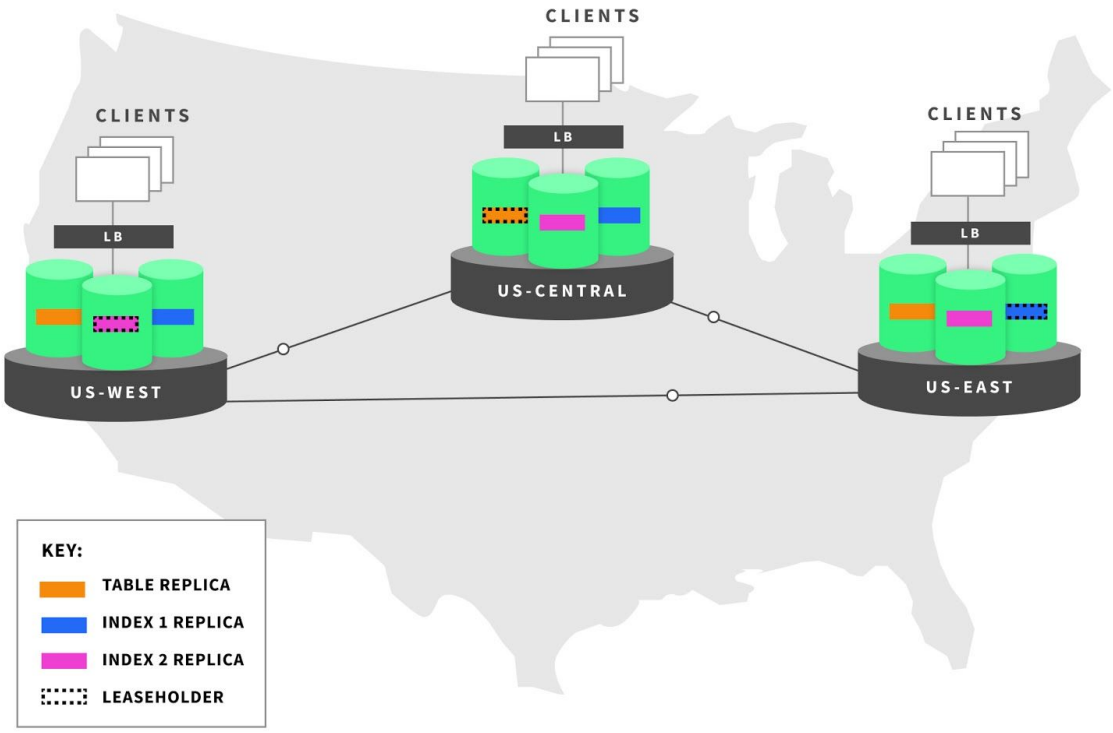
CockroachDB also provides strong resiliency. It replicates data three times by default and distributes the replicas in a way that maximizes geo-diversity. The financial software team decided to deploy CockroachDB across three AWS regions in the U.S. With their replicas of data spread across these regions, their CockroachDB deployment could survive an entire region failure without any downtime.

Now that their consistency and resiliency needs were met, the team needed to achieve low latency reads for their CIAM app. They decided to take advantage of CockroachDB's Enterprise duplicate indexes feature. This feature is ideal for tables that are rarely updated and require high read performance. Using this feature, the company could create additional indexes with stored columns that would serve local reads with very low latency in each region.

By default, only one of any three replicas in CockroachDB is allowed to handle reads and writes for its corresponding segment of data. This replica is called the leaseholder, and it retains control over reads and writes. Different rules apply to reads versus writes. Reads can return from the leaseholder directly without the need for communication with other replicas. However, for a write to be acknowledged, the leaseholder must communicate with the other replicas, and the majority of replicas must achieve quorum before the value is

written. The location of leaseholders can increase latency in a multi-region cluster that is not properly configured. For example, when users from the West Coast access data with leaseholders on the East Coast, they must pay a cross-country latency penalty.

CockroachDB's duplicate indexes feature solves the latency issue for reads. With this feature, the user pins the leaseholder for a table to one region. Then the user creates two secondary indexes on a table, essentially replicating that table, and pins the leaseholders for each secondary index to each of the other regions.



An example of a multi-region CockroachDB configuration that uses the duplicate indexes feature. Two secondary indexes are created for a table, and the leaseholders for the table itself and the two indexes are pinned to different regions. Clients then access the nearest leaseholder for reads.

After that, CockroachDB's Cost-Based Optimizer (CBO) directs queries to the nearest leaseholder. The CBO is aware of locality constraints and leaseholder preferences for tables and indexes. It automatically chooses to use either the underlying table or one of the two indexes, depending on which will be fastest based on where the query was executed. With this pattern, reads are local and extremely fast, taking only about one millisecond. This high read performance allowed the financial software company to improve their customer login experience. In this configuration, writes still need to leave the region to get

consensus from the other replicas, and they take about 280 milliseconds. This time is orders of magnitude faster than the 30 seconds to five minutes that the financial software team had endured when using GoldenGate.

Results and what's next

With CockroachDB, the software company found an ideal database for their customer access and identity management. They achieved the consistency, resiliency, and quick data retrieval they needed, at a multi-region scale.