Cockroach Labs

CockroachDB vs Oracle Globally Distributed Database

A comparison of enterprise-grade distributed relational database management systems.

www.cockroachlabs.com

Executive Summary

Mission-critical, data-intensive applications need the reliability and consistency of a relational data store and the flexible scale, high availability, and global coverage of a distributed system. Distributed SQL combines the consistency and structure of the early relational databases with the scalability, reliability, and performance first pioneered in NoSQL. Distributed SQL databases like CockroachDB use this architecture to provide a single logical database that replicates data across physical nodes at scale, on any infrastructure, anywhere in the world.

Oracle Globally Distributed Database (Oracle GDD) claims to surpass leading Distributed SQL databases like CockroachDB by finally capturing the benefits of Distributed SQL[1] that traditional SQL databases struggle to provide: horizontal scale, fault tolerance, and high availability while maintaining data sovereignty and ACID transactions. Brand new to the distributed database market, Oracle GDD is an attempt to copy and match up to — without success — proven solutions like CockroachDB, which have been deployed for years in global enterprise production workloads.

Retrofitted legacy SQL vs cloud native distributed SQL

Released to general availability in March[2] 2024, Oracle GDD incorporates the Raft consensus protocol[3] to extend Oracle's legacy partitioning methods across multiple Oracle databases. Oracle itself, however, was built for single-tier applications — not the interconnected, virtualized, and abstracted systems that define cloud computing. Getting this legacy database to perform in an even semi-distributed fashion requires inserting a control plane between the application and multiple physical Oracle database instances ("shards").

Bolting a sharding coordination layer on top is the only way to retrofit a legacy database to handle data distribution and replication functionalities, and to approximate horizontal scalability. (This extra layer also means Oracle GDD cannot offer important SQL features like global primary keys, unique constraints, and referential integrity between tables). Oracle recognizes the customer shift to strongly consistent distributed databases built to be cloud native; Oracle GDD is a repackaging of existing software components in an attempt to find relevance in this space.

By comparison, CockroachDB was designed as a distributed relational database from the ground up. CockroachDB is a highly available, ACID compliant, elastically scalable database and has been consistently chosen by enterprises building mission-critical applications. CockroachDB has been in the market since 2015 and has been tested in many different enterprise production deployments, including hyperscale workloads.

[1] https://www.cockroachlabs.com/blog/what-is-distributed-sql. Last accessed June 6, 2024.
[2] https://www.oracle.com/news/announcement/oracle-announces-availability-globally-distributed-autonomous-database-2024-03-04. Last accessed June 6, 2024.
[3] https://docs.oracle.com/en/database/oracle/oracle-database/23/shard/raft-replication.html. Last accessed June 6, 2024.

Due diligence and disclosures

In this document, we will delve into the practical and technical aspects of both these platforms and highlight why CockroachDB could be the best database of choice for you and your team. It's important to note, however, that, although it’s in General Availability, Oracle Globally Distributed Database documentation is still a work in progress. Our team encountered numerous recursive loops, conflicting guidance, and outright gaps in the documentation. This comparative document has been written to the truest and fairest of our ability with the information that is available as of the date of publication (May 2024).

Architectural simplicity

The fundamental difference between CockroachDB and Oracle GDD is architectural simplicity vs. complexity, and the resulting stability vs. inherent fragility.

CockroachDB is a distributed SQL database designed for scalability, consistency, and high availability. CockroachDB's distributed SQL architecture allows an application to connect to and execute SQL queries on any node, anywhere in the world, as though it is a single logical database instance.

Using the Raft consensus protocol allows CockroachDB to replicate data while surviving failures. Transaction protocols built with distributed data in mind allow CockroachDB to provide guaranteed ACID characteristics without manual sharding or knowledge of shard placement. Since CockroachDB does not rely on explicit data placement to provide ACID transactions, CockroachDB can automatically rebalance replica placement to avoid load hot spots. This is all with no manual intervention, using only a single binary and single running process – fewer services to manage. Schema, replication, and transactions are all managed within the database as part of the database, with no extra configuration required.

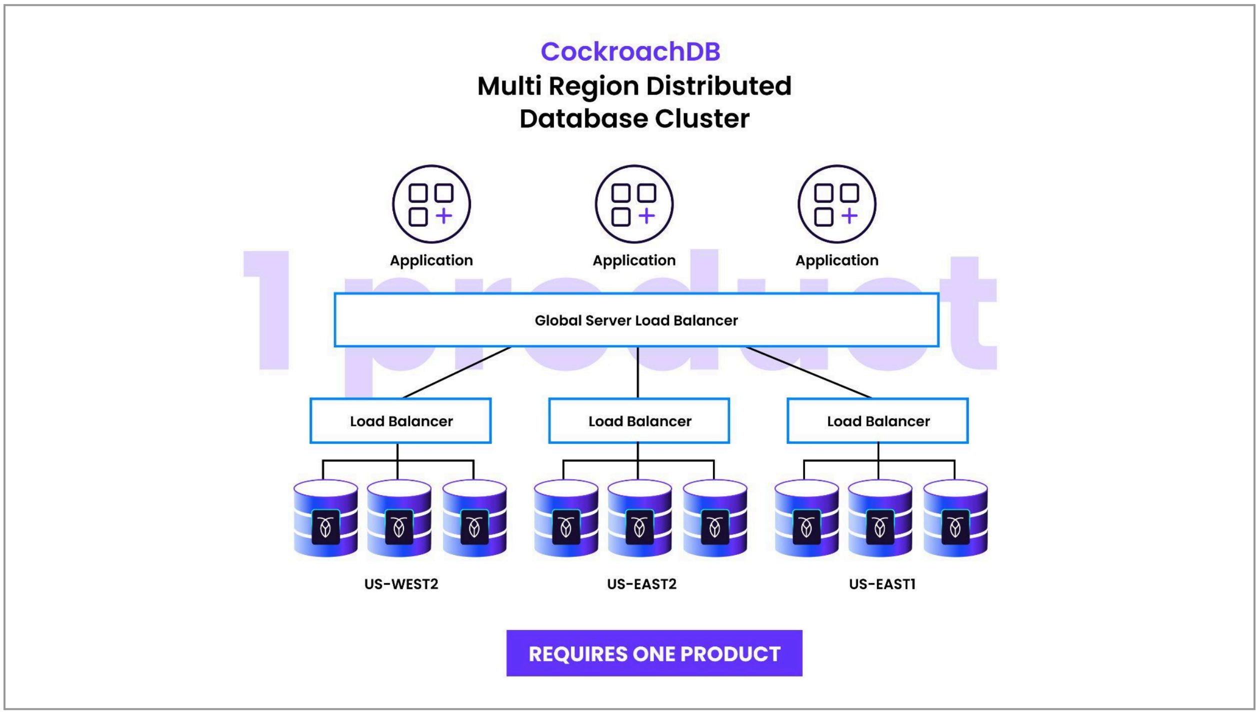

Fig 1: CockroachDB Distributed SQL architecture