

Active-Active Geo Distribution Based on CRDTs

Developing globally distributed applications that respond with local latencies, while maintaining consistent views worldwide is challenging. Redis Enterprise introduces the capability to deploy Redis databases in an active-active geographic distribution (i.e. read and write operations on the same dataset) with guaranteed local latencies. It achieves this using a consensus-free protocol with strong eventual consistency characteristics and built-in smart conflict resolution based on the CRDT (Conflict Free Replicated Datatypes) approach.

Conflict Free Replicated Datatypes are a sophisticated technology that provide:

- Uninterrupted availability through the ability to provide reads and writes even when some regions and data centers are completely unavailable
- Simplified development of mission critical applications, with global workloads and complex datatypes, that need concurrent low latency reads and writes

Why CRDT-based Active-Active?

Active-Passive databases are inherently deficient:

- They can geo-distribute reads but writes are always sent to the only active replica, thereby incurring higher latencies
- The passive replicas under utilize their resources since they are required to be on standby to handle full capacity, in case of failures

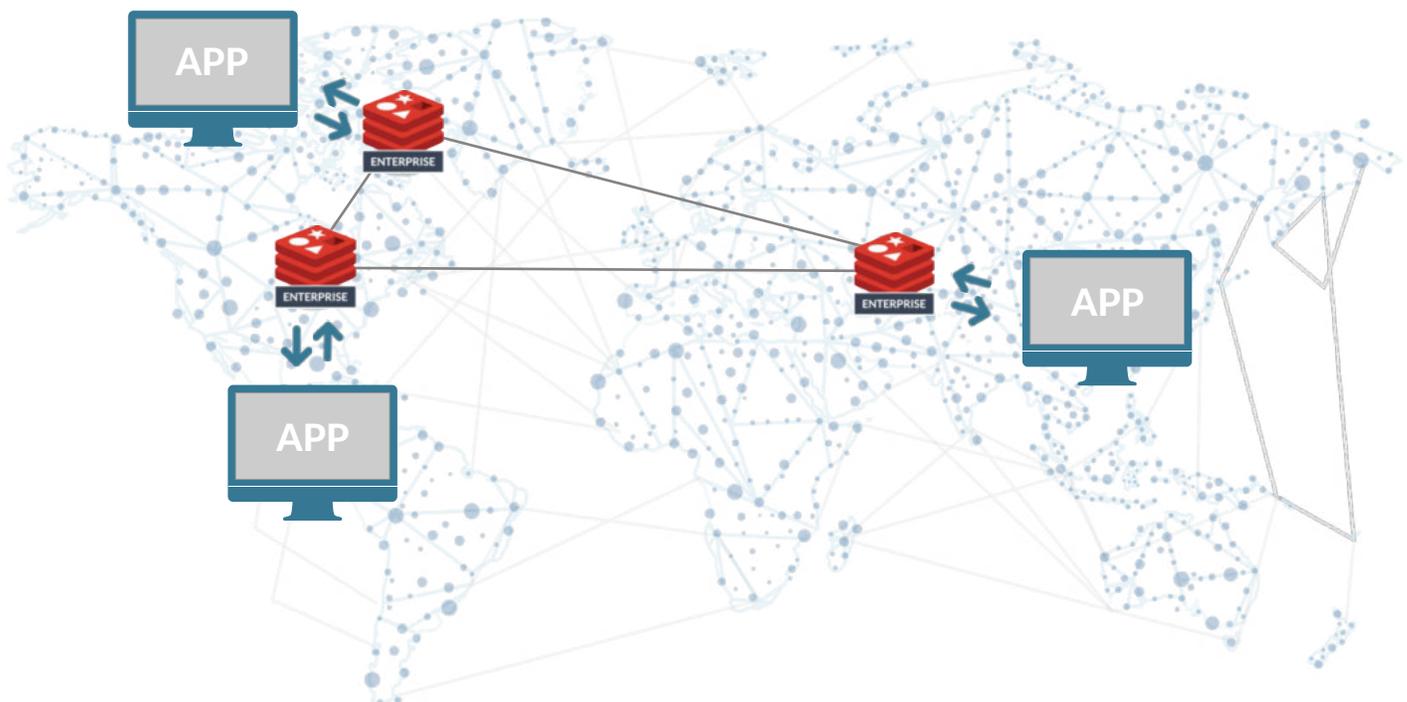
Active-Active databases that don't use CRDTs are also problematic, because they:

- Impose WAN latencies for either read or write operations, regardless of their consistency model
- Utilize crude tools like Last Writer Wins (LWW) which do not provide fine-grained conflict resolution needed by modern applications that use complex data-types like the ones available with Redis

Benefits of CRDT-based Active-Active architecture

- Guaranteed local latencies for both read and write operations, utilizing consensus-free protocols to maintain consistency
- Built-in conflict resolution for simple and complex data-types that simplifies app development and global deployment
- Strong eventual consistency delivers converging consistent views of the data, effortlessly
- Streamlined implementation of modern app scenarios such as distributed counters, multi-user metering and many more
- Safer cross-geo failover, with automatic, intelligent sync between active databases, that avoids incorrect overwrites and loss of state

Consensus Free With Strong Eventual Consistency



How It Works

Redis CRDTs use bidirectional replication between multiple Redis databases distributed across multiple data centers to remote geographies. Application developers simply continue coding their applications against the well known Redis data types and commands while Redis CRDTs intelligently resolve conflicts based on data type and commands used by the applications.

Consistency

Any two participating clusters that receive the same set of updates, even if unordered, will be converged to the same state.

Availability

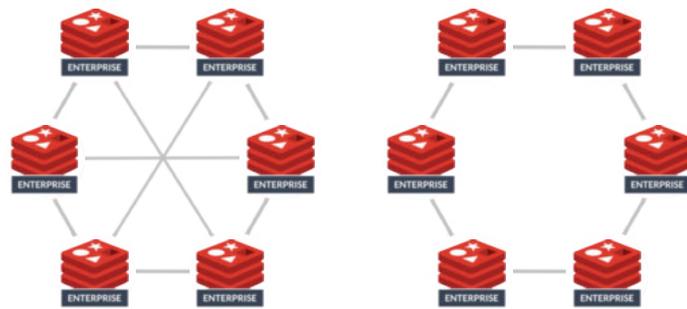
- Automatic failover within clusters provides high availability in each data center under node, rack or zone failures.
- Redis CRDTs provide availability under total regional or data center failures or network communication failures between regions.

Scalability

- Local scalability is achieved by re-sharding and adding nodes to each participating cluster
- Global scalability is achieved by adding new participating clusters

Topology

Redis Enterprise's Active-Active architecture incorporates a fully-connected (mesh) topology where each replica is connected to all other replicas.



Fully-Connected

Ring

Automatic Conflict Resolution

Each Redis data type comes with a set of commands. Redis CRDTs resolve conflict in writes across multiple participating clusters using well defined rules per data type.

What Are The Use Cases?

E-Commerce

Assuring great user experience in ecommerce scenarios needs high availability and resilience to failure scenarios. In the case of a failure of a replica (or a region), user sessions, shopping carts and active wish lists need to be seamlessly picked up by another replica without loss of state and, optionally, transferred back on restore. In the interim, updates made by the user must be handled appropriately (example, adds and removes from the shopping cart must transfer over and must not be overwritten by a previous state). Redis Enterprise CRDTs handle such a scenario gracefully, with a smooth transition between active replicas while maintaining a consistent view from the user standpoint.

Fraud/Anomaly Detection

Often applications that monitor user activity or account activity are globally distributed and must keep track of several events simultaneously. While creating and maintaining scores for a user, activity from different geographic locations must simultaneously update scores to keep risk metrics inline. Redis Enterprise CRDTs handle this at the level of both simple datatypes like counters and complex ones like Sorted Sets, assuring developers don't have to handle conflict scenarios at the application level.

Collaboration

For applications that rank popularity of articles—be they goods on sale or media to be consumed—tracking popularity across locations can get complicated, particularly when royalties need to be paid or real-time advertising decisions to be made. Ranking, sorting and reporting across many geographic locations, in real time with Redis CRDTs allows developers to deliver features with little effort and without compromising on latencies.

Metering

Counting and regulating usage (such as API calls, transactions/second, and minutes used) can be implemented globally with simplicity using Redis CRDTs. Built-in conflict resolution assures both accuracy of counts and regulation in real time.

IoT

Geo-distributed edge deployments that use CRDTs often need to track time-series data from multiple locations. Utilizing common datasets to track such data from multiple locations ensures maintaining data consistency, particularly when aggregating data. Redis Enterprise CRDTs allow for simplified distributed implementations of IoT data processing.

Personalization

Whether you're maintaining geographically distributed counters that trigger actions such as loyalty points awards, or implementing personalized user session views, the high availability and simplified geo-distributed counting provided by Redis CRDTs allow application teams to deliver high performance with simplicity.

Key features of Redis Enterprise CRDTs

- Available with all standard Redis clients
- Available with simple and complex datatypes: Strings, Hash, Counters and Sets
- Maintains sub-millisecond response times even with high throughput
- Automated failure detection, instantaneous failover and re-sync across regions

Get Redis Enterprise Products Today!

Talk to a Redis Enterprise expert today: expert@redislabs.com.
Try Redis Enterprise for free at www.redislabs.com.