redis

Uninterrupted Availability Anywhere

Executive summary

Ensuring fast application performance and high availability to customers no matter where in the world they are located is a prime challenge for modern enterprises. Making that requirement a reality requires a wide variety of technologies, including a robust data layer that can deliver instant responses to any geographic region, support virtually unlimited scaling to handle planned and unplanned peaks, and deliver five-nines (99.999%) uptime without losing data even in the event of a failure—all without burdening your IT department with unneeded complexity. That's exactly what Redis Enterprise's advanced Active-Active technology is designed to do.

> "Enterprises need database technologies that let them build modern applications while meeting ever-increasing performance, scalability, and regulatory requirements."

Introduction

In today's digital world where competition is always just a click away, applications must be highly responsive and resilient. Application downtime can have dramatic short- and long-term negative effects, so the high availability of the data layer is simply non-negotiable. And high availability goes beyond a simple "on" or "off" state—to be considered "available," your service must meet demanding customer expectations during both normal and peak times.

To a certain extent, of course, in a distributed world failures are inevitable. The goal is to reduce the frequency of any failures and minimize their impact on your business.

Ideally, the modern data layer should enable you to deliver instant[1] experiences to users regardless of region, be hyper scalable to support growth and seasonal surges, consolidate data silos to generate new revenue streams, simplify operational complexities, and future proof your IT strategy.

Redis Enterprise's Active-Active technology is designed to support all of those requirements. Based on conflict-free replicated data types (CRDTs), it unifies data across clouds and geographies to lower operational burden and application latency. Active-Active automatically resolves any data conflicts and effortlessly delivers a consistent view of the datasets.

It supports automated failure detection and single-digit-seconds failover to minimize downtime and data loss. Redis Enterprise's Active-Active capability lets businesses focus on what matters to them most instead of fighting to reduce latency and increase availability.

[1] Nielsen Norman Group: "Response Times: The 3 Important Limits"

Why active-active: The need for speed, availability, and resiliency at global scale

Early on in computing, databases lived on a single machine, which meant they were either "on" or "off," there wasn't another state of operation in between. As more use cases (e-commerce, banking, education, etc.) appeared, the demand for applications to be always available and responsive grew. This increased load on a single machine led to more frequent failures and increased downtime.

To address these outages, improve application response time for users around the world, and to help recover from disasters, organizations developed new architectures that replicated data to geographically distributed locations, continuously synchronized with the primary database. This approach is commonly referred to as active-passive. As the name suggests, active-passive involves one primary data center that actively processes requests, along with one or more secondary sites that either process read requests or are idle unless pressed into action by a primary site's failure.

Active-passive addressed many performance, scalability, availability and resiliency challenges, but not all of them. To start with, write operations still have to be sent to the primary database, which incurs additional latency. Furthermore, when passive databases are in standby mode, their resources are underutilized. Worse, if the primary database fails before any new or updated data is replicated, those values are lost. Lastly, some use cases (such as communication systems for emergency responders) can't afford to be down while the standby database comes online.

The active-active architecture was developed to address these scenarios. In an active-active setup, all instances are online and can execute read and write operations and replicate the updated values to other members. So the chances of downtime and data loss is significantly minimized.

But because active-active implementations allow concurrent write and update operations on the same key across instances, an important question arises: "When conflicts occur, how are they resolved?" There are a number of approaches to this problem, including Last Writer Wins (LWW), consensus driven protocol, quorum-based methods, and others. However these simple solutions are not suitable for real-time use cases because they lead to data loss (as a result of overwriting data) or increase application response time (due to increased communication overhead).

Why consistency matters

In an Active-Active architecture, all database instances are available for concurrent read and write operations. Based on multi-master replication and CRDTs, it delivers local latency regardless of the number of geo-replicated regions and their distance from each other, all with seamless conflict resolution without incurring the latency penalty imposed by other consensus-driven protocols.

Application developers use existing Redis data types and commands, but the Active-Active database understands[2] the developer's intent and automatically resolves conflicts that occur as the result of concurrent writes to the same key across multiple geographies.

This approach delivers strong eventual consistency, meaning that all participating databases that have received the same set of (unordered) updates will eventually converge to the same state without the need of consensus protocol. If the Causal Consistency flag is enabled, the order of operations on a specific key are maintained across all Active-Active database instances. This is an important capability for e-commerce transactions or chat applications, ensuring the order of messages doesn't get mixed up.

[2] Redis Enterprise documentation: Active-Active Geo-Distributed Redis

> "Redis Enterprise's Active-Active technology guarantees 99.999% uptime for global applications."

Redis Enterprise's Active-Active technology solves app development and operations challenges

In the words of Arthur C. Clarke "Any sufficiently advanced technology is indistinguishable from magic", but rest assured Redis Enterprise Active-Active technology isn't magic. This sophisticated technology is built using leading edge and academically proven methodologies to deliver unprecedented performance and scalability for globally distributed applications without increasing operational complexities.

Improves application response time for end users around the globe

Applications connect to the participating member of an Active-Active database that is closest to them, which guarantees a local level of latency for both read and write operations. With bi-directional replication, changes made to each participant are replicated to all the other members and conflicting writes are automatically resolved based on predefined rules. Without Active-Active capabilities, these queries must be sent to remote data centers, significantly increasing application response time, making it impossible to meet performance and availability requirements.

Unifies the data layer across regions and clouds

Redis Enterprise can be deployed on-premises or on any cloud, or in a hybrid model. Participating members of an Active-Active deployment can reside on-premises or in a multi-cloud deployment without creating data silos or fragmentation. Hybrid deployment paired with seamless conflict resolution ensures your applications have access to consistent datasets while you are migrating to the cloud or between them. This is crucial to a successful digital transformation as 91%[3] enterprises cite data fragmentation as the main reason they can't not fully realize cloud benefits.

Faster time to market for modern and resilient applications

A Redis Enterprise Active-Active database is a single logical database that spans across data centers, so application developers don't have to worry about understanding a large number of race conditions between updates to various sites, network, and cluster failures that could change the outcome of the updates performed across geo-distributed writes.

Developers simply choose from existing Redis data types and commands and any conflict is automatically resolved based on well-defined rules by Redis CRDTs. Non-CRDTs based active-active databases either rely on one method to resolve all conflicts (for example Last Writer Wins) or offload this complex task to the developer, which significantly lengthens development cycles and adds architectural complexity.

Simply put, these databases don't have what it takes to power highly distributed applications around the world.

---

[3] Cohesity: "3 Must Haves to Effectively Manage Data in the Cloud"

Five Active-Active use cases

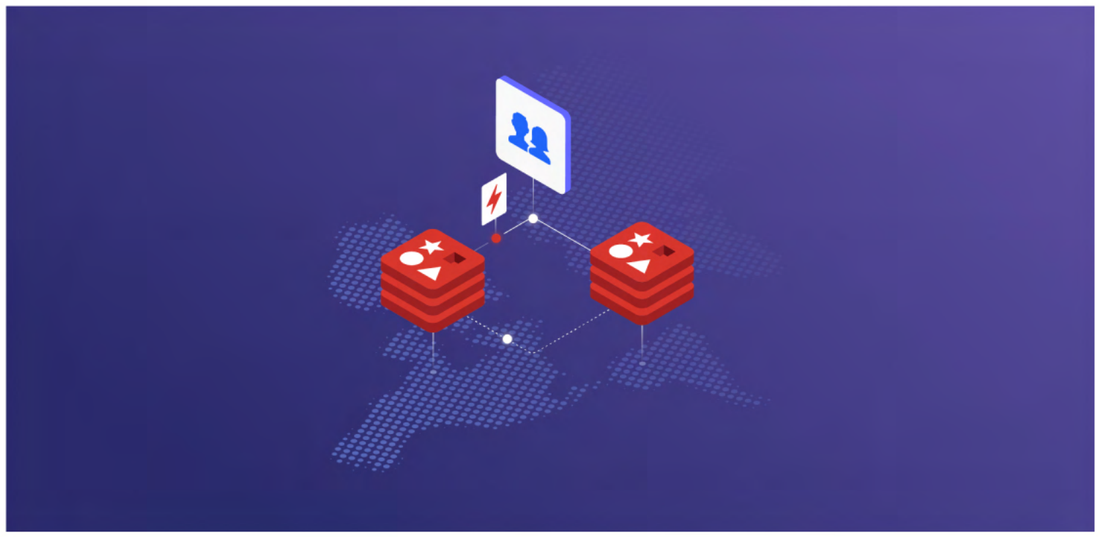
To see how Redis Enterprise Active-Active database is the ideal solution for building modern applications, let's take a closer look at five popular use cases.

User session migration across data centers

To deliver the best end user experience, there are times when it would be ideal to shift users from one data center to another. For example, if a user is shopping online while riding a train, at some point in their journey they could get closer to a different data center than the one they were connected to when they started their trip. It is challenging to migrate all of a user's session information in the middle of active sessions without losing any data (items in the cart, credit- card information, billing and shipping addresses, and so on).

In an active-passive model, users remain connected to the original data center even as they move farther away from it, which can increase latencies.

Redis Enterprise's Active-Active approach seamlessly routes user session information from one data center to another in real time. All session states are preserved during the transition and both databases converge automatically to the same state with strong eventual consistency.



Disaster recovery

In a distributed, always-available world, failure (whether of a node, zone, region, etc.) is inevitable. The impact of downtime includes loss of revenue and employees' productivity, data corruption and data loss, disruption to the supply chain, damage to brand reputation, and so on. That makes it business critical to reduce the number of failures and the severity of their impact.

In a Redis Enterprise Active-Active deployment, if an application can't connect to the local server, it automatically gets redirected to the next nearest one. When the original server is back online, the connection between that server and application is resumed and all data is immediately consolidated, without data loss, between the two servers.

Distributed querying, indexing, and search

Based on a recent Forrester report[4] , 43% of retail website visitors go directly to the search bar, and these browsers are 2-3 times more likely to become buyers if they are served relevant results fast. RediSearch is a distributed query, indexing, and full-text search engine for Redis. It can be deployed as a multi-site, Active-Active search engine, it lets developers automatically index and replicate data across regions to enable on-demand scaling of their databases to easily handle billions of documents on hundreds of servers and provide business continuity in any scenario.

RediSearch allows you to create secondary and full-text indexes on datasets hosted in Redis. These indices let you query your Redis data at lightning speed, perform sophisticated aggregations, and run faceted searches. It includes a rich query language that can perform simple full-text searches, as well as complex structured queries, filtering by numeric properties and geographical distances. Further, you can do "fuzzy" searches to implement auto-complete search experiences.

RediSearch is able to do quick document indexing by using an incremental indexing approach. This makes it ideal for searching through frequently updated datasets, delivering real-time performance for use cases such as real-time inventory, fraud prevention, predictive alerting, dynamic catalogs, secondary indices of large databases, and more.

Geo-distributed application functionality

Rating and ranking are the heart of social media, content recommendation, and gaming applications. But it's not easy to accurately track scores and leading topics or players. You need either have one central database, which imposes availability and response-time challenges, or a data platform that can immediately resolve conflicts and consolidate data among multiple databases.

Redis Enterprise's Active-Active database is the ideal choice for keeping track of changes in content rank and user engagements, as it seamlessly resolves any conflict without overwriting any data.

[4] Forrester: Must-Have E-Commerce Features

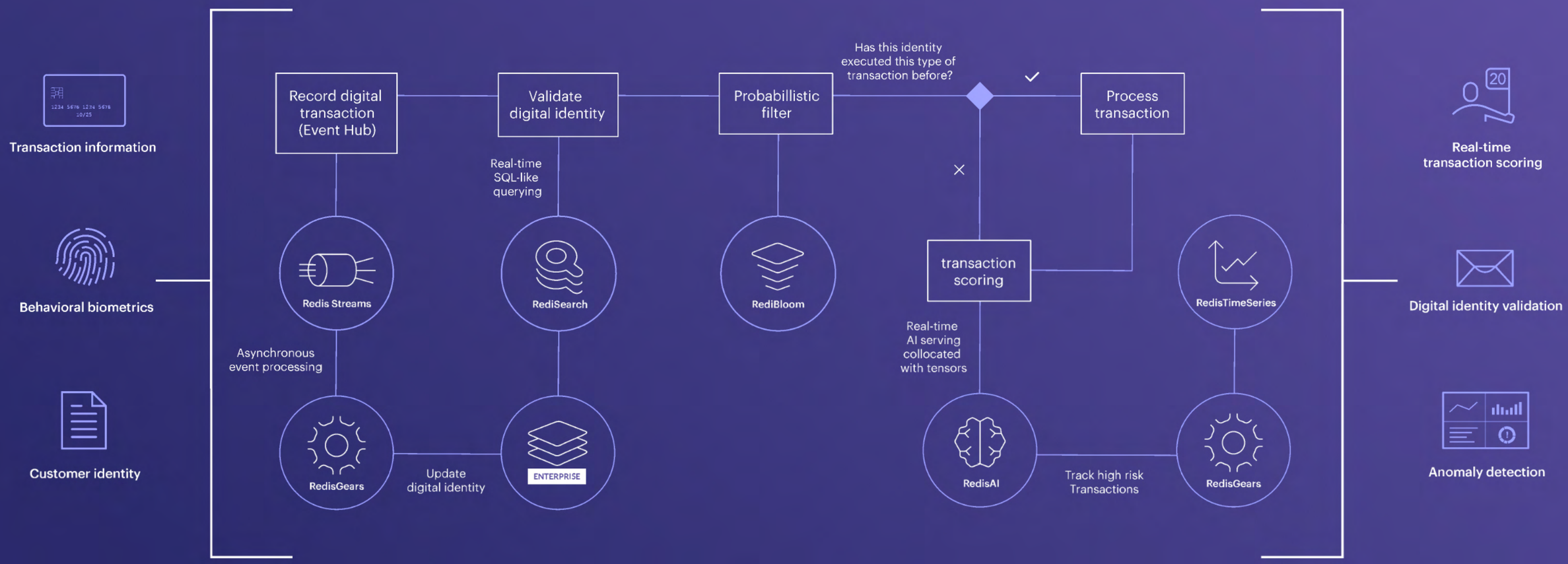
Microservices environment

According to a recent Kong report[5] , 84% of organizations are using microservices architectures to accelerate innovation and stay competitive. A major difference between microservices and monolithic architectures is that with a microservices approach , each individual service can stand on its own without being tightly coupled with other services.

When adopting a microservices architecture, it is important to consider whether data is shared among these services to avoid issues with data consistency and performance bottlenecks.

Whether there is a shared dataset among services (user data, for example) or one database for each service (for example, an event-driven architecture where services transfer their datasets among themselves), Redis Enterprise's CRDTs based Active-Active technology immediately resolves data conflicts to ensure all services are operating with real-time data.

[5] Kong: 2020 Digital Innovation Benchmark



Thank You

But wait, there's more!

In this paper we've shown how Redis Enterprise's CRDTs-based Active-Active technology can be used to solve business challenges in five popular use cases. But that's only a small selection of the many scenarios where Active-Active deployments can deliver fast, local-latency performance to huge numbers of users around the globe while bridging data silos, cutting through operational complexities, and empowering winning technology strategies.

Want to know more? Watch a short video on How to Use Active-Active for E-commerce, learn how conflict resolution works, dive deeper into CRDTs, and see how to create an Active-Active database and develop applications.

Or read an exclusive case study detailing how Mutualink leverages Redis Enterprise Active-Active technology for immediate data conflict resolution and seamless support of microservices architectures to build and deploy its advanced technology for emergency management communications.

About Redis

Modern businesses depend on the power of real-time data. With Redis, organizations deliver instant experiences in a highly reliable and scalable manner.

Redis is the world's most popular in-memory database, and commercial provider of Redis Enterprise, which delivers superior performance, matchless reliability, and unparalleled flexibility for personalization, machine learning, IoT, search, e-commerce, social, and metering solutions worldwide.

Redis, consistently ranked as a leader in top analyst reports on NoSQL, in-memory databases, operational databases, and database-as-a-service (DBaaS), is trusted by more than 7,400 enterprise customers, including five Fortune 10 companies, three of the four credit card issuers, three of the top five communication companies, three of the top five healthcare companies, six of the top eight technology companies, and four of the top seven retailers.

Redis Enterprise, available as a service in public and private clouds, as downloadable software, in containers, and for hybrid cloud/on-premises deployments, powers popular Redis use cases such as high-speed transactions, job and queue management, user session stores, real time data ingest, notifications, content caching, and time-series data.

Follow us:

redis.com

©2021 Redis