Couchbase
NoEQUAL

REDIS + MONGODB VS COUCHBASE PERFORMANCE COMPARISON

Superior performance and manageability

Watch Webinar Get started now >

The best way to learn is to download the software and try it out Get It Now

MongoDB is a popular NoSQL document database that many developers have played around with. It has some nice features, and can be a good fit for use cases – typically small-scale applications running on a few nodes. But companies report problems when trying to scale MongoDB to support more users and bigger workloads on clusters with multiple nodes.

Redis is a well-known caching solution that some companies use to alleviate the performance and scalability issues that arise with MongoDB. Using Redis in conjunction with MongoDB can improve performance and scalability, but leads to greater operational complexity and cost for database architects and operations professionals.

MongoDB's "master-slave" architecture is a major ca of its scaling limitations.

Many of the issues that businesses face with MongoDB stem from its fundamental design: MongoDB was originally engineered for single-node deployments rather than a distributed database. MongoDB uses a "master-slave" architecture, which critically limits its ability to perform at scale. This master-slave architecture is hardwired into MongoDB's design, so it's not something that can be easily changed. With MongoDB, you have two approaches to try and scale – you can scale read operations by increasing the number of secondaries in each replica set (data shard) or scale both reads and writes by increasing the total number of data shards.

Viber switched to Couchbase and for less than half the number of Redis/MongoDB nodes

switched to Couchbase, such as electronics and internet communications giant Rakuten Viber. Viber is a cross-platform instant messaging and voice-over application used by over 800 million subscribers, and competes with Facebook Messenger and WhatsApp for both audience share and revenues. Viber needed a single robust data platform that could perform millions of operations per second with consistent low latency on huge data sets; a database that could scale easily without any performance impact, and was easy and inexpensive to monitor and manage. To power potential user growth and support hundreds of millions of subscribers, Viber chose to switch from its MongoDB and Redis system to the Couchbase Data Platform. In doing so it moved from an estate of 150 MongoDB servers and 100 Redis servers to one of 120 Couchbase servers, because Couchbase takes care of the caching duties originally performed by Redis. In Couchbase, Viber found a single platform that provides internet-scale performance, elastic scalability, five nines availability, manageability, and lower cost of operations.

Learn More

7 frequent challenges with Redis/MongoDB and how the Couchbase Data Platform avoids them

very difficult to scale from a single replica set to a fully sharded environment.

you have to manually partition your data based on IDs embedded in the keys, hash of keys, or some combination of the two. MongoDB struggles with the same scalability challenges. Scaling the Redis/MongoDB stack requires a complex process and manual configuration to scale from a single replica set to a fully sharded environment.

elastic architecture, where both the integrated object cache and document database are supported with a single node type. Scaling Couchbase on demand is a simple, push-button process – add nodes and rebalance data with a few clicks, and you're done.

Challenge 2: Cost

MongoDB's inability to scale increases operational costs.

Redis/MongoDB

Both Redis and MongoDB use a master-slave architecture, that limits the ability to efficiently support many concurrent users with a single node. Beyond a few dozen concurrent users per node with MongoDB, performance rapidly degrades. Some businesses try to optimize MongoDB database operations

Couchbase

Couchbase is purpose-built as a memory-first architecture with key-value and blazing-fast cache integrated with a document database that is designed to efficiently handle large numbers of concurrent users. Couchbase easily supports hundreds of concurrent users on a single node with to no impact o performance.

caching solution in front of MongoDB.

Challenge 3: Replication and failover

MongoDB is significantly susceptible to data loss and inconsistency.

Redis/MongoDB

The way MongoDB handles replication and failover (by promoting “secondaries” to “primaries” automatically) can result in lost writes and inconsistent reads. So if a primary goes down, MongoDB is likely to suffer data loss or inconsistency. This risk is one reason why MongoDB deployments need to be closely monitored at all times.

Couchbase

The Couchbase Data Platform, which was purpose-built for distributed environments, has an entirely different approach to replication and failover, with cross datacenter replication (XDCR) that significantly reduces the potential of data loss and inconsistency. During a failover, Couchbase prevents two distinct nodes from accepting simultaneous reads or writes of the same data.

Challenge 4: Operational overhead

Redis/MongoDB deployments require operations teams with mastery of disparate skill sets.

Redis/MongoDB

Because MongoDB cannot handle many connections and many concurrent users per node, companies often need to add a dedicated third-party cache like Redis, with Redis-capable engineers to help MongoDB engineers meet their performance

Couchbase

The Couchbase Data Platform is self-managing for most applications. It has a fully integrated managed object cache that completely removes the need for a third-party cache like Redis. The Couchbase Engagement Database combines a applications. The need for a third-party cache can add both cost and complexity to a Redis/MongoDB deployment.

document database, eliminating the need to hire operations teams with knowledge of multiple technologies.

Challenge 5: Multi-data center deployment

When deployed in multiple geographies, MongoDB can't perform all writes locally.

Redis/MongoDB

Due to its master-slave architecture, and its use of "primaries" and "secondaries," MongoDB cannot perform all writes locally when it's deployed across multiple data centers in different geographies. This means it has to perform at least some writes to remote locations, which can increase latency and degrade application performance.

Couchbase

The Couchbase Data Platform's masterless, memory-first distributed architecture, and its built-in support for cross datacenter replication (XDCR), means that all writes can be performed locally rather than to a remote location. This capability minimizes latency to help optimize application performance.

Challenge 6: Mobile

Redis/MongoDB lacks a mobile solution.

Redis/MongoDB

Redis/MongoDB does not have a complete solution to support mobile applications. As a result, you have to write custom services for your mobile apps to access

Couchbase

The Couchbase Data Platform provides a complete mobile solution – Couchbase Mobile – that includes an embedded JS database (Couch Lite) and a pre-built

	write your own code to synchronize data on the mobile device with data in a remote server, or forego synchronization and instead pull data from the remote server for every request. This means that your app always needs a	use Couchbase to easily build mobile apps that always work, with or without an internet connection.
Challenge 7: Time	**Redis/MongoDB**	**Couchbase**
Redis/MongoDB is time-consuming to manage and maintain.	Due to its master-slave architecture, the topology of a MongoDB deployment can be complex and increasingly complicated as you try to scale it. Deploying MongoDB requires lots of manual configurations and includes numerous components: routers, config servers, arbiters, primary instances, and secondary instances.	Couchbase's masterless, distributed architecture is elegantly simple. Installing Couchbase and setting up a cluster is a fast and easy process, with minimal components and configuration requirements. A multi-node Couchbase cluster can be set up in minutes, and adding or removing nodes is a simple, push-button process.

Register for the webinar Couchbase Data Platform vs. Redis/MongoDB stack

scalability, but it also adds cost and complexity. In this webinar, we'll compare the Couchbase Data Platform versus the Redis/MongoDB stack and discuss how the different solutions handle common operational challenges, including: architecture, data consistency, deployments, data reads, writes, and queries, support for multiple data centers, support for offline mobile use cases, SQL on JSON, and agility and time-to-market for internet-scale applications. Join us, and see for yourself how Couchbase reduces cost, complexity, and database sprawl with a purpose-built data platform that includes caching, cross datacenter replication, Full Text Search, analytics, multi-dimensional scaling, and N1QL (SQL for JSON).

Register For U.S. Register For EMEA

Talk to a Couchbase Solutions Engineer **Arrange A Call**

Couchbase

3250 Olcott Street
Santa Clara, CA 95054
United States

Contact Us

1-650-417-7500

COMPANY

ABOUT

LEADERSHIP

NEWS & PRESS

CAREERS

EVENTS

LEGAL

SUPPORT

DEVELOPER PORTAL

DOCUMENTATION

FORUMS

PROFESSIONAL SERVICES

SUPPORT LOGIN

SUPPORT POLICY

PRICING

QUICKLINKS

BLOG

DOWNLOADS

GET STARTED

ONLINE TRAINING

RESOURCES

WHY NOSQL

COVID-19 FAQ

FOLLOW US

FACEBOOK

GITHUB

TWITTER

LINKEDIN

STACKOVERFLOW

YOUTUBE

Terms of Use **Privacy Policy** **Cookie Policy** **Support Policy**

Marketing Preference Center