Couchbase

WHITEPAPER

Why NoSQL Databases for AI-Powered Apps?

An Architectural Guide

Why successful projects rely on NoSQL database applications

Contents

WHAT IS NOSQL?

NoSQL is a modern database management system that stores information in JSON documents instead of the tables, columns, and rows used by relational databases. It can deliver data for applications in ways that make development easier and more robust. NoSQL databases can also serve as the basis for AI applications that need ultimate adaptability now and into the future.

NoSQL databases are built from the ground up to be flexible, scalable, and capable of rapidly responding to the data management demands of modern businesses.

This paper introduces the modern challenges that NoSQL databases address and shows how multipurpose NoSQL delivers the flexibility that relational lacks and why.

RELATIONAL VS. NOSQL: WHAT'S THE DIFFERENCE?

Relational databases were born in the era of mainframes and back-office business applications – long before the internet, the cloud, artificial intelligence, big data, mobile, artificial intelligence, etc.

Customers demand hyper-personalized experiences from adaptive applications

For modern businesses, customer experience is the most important competitive advantage, driving the need for adaptability in application development. To meet these expectations, businesses must prioritize services that are on demand, real time, resilient, and responsive.

Bringing together all of these new systems requires flexibility, performance, and scalability and can consolidate multiple types of systems to reduce database sprawl.

Relational vs. NoSQL: What's the difference?

Relational databases were born in the era of mainframes and back-office business applications – long before the internet, the cloud, big data, mobile, artificial intelligence, etc. In fact, the first commercial implementation was released by Oracle in 1979. These databases were engineered to run on a single server – the bigger, the better. The only way to increase the capacity of these databases was to upgrade the server – processors, memory, and storage – to continually scale up as Moore's Law allowed.

NoSQL databases emerged as a result of the exponential growth of the internet and the rise of web applications. Google Bigtable research was released in 2006, as well as an Amazon DynamoDB research paper in 2007. Efficient distributed key-value (KV) engines were essential to this evolutionary step and have propelled the technology much further.

New databases were engineered to meet the next generation of enterprise requirements, which companies like Couchbase have taken even further to meet needs going into the future – the need to **develop with agility** and to **operate at any scale**.

Agility means providing flexible schemas, APIs, AI integration, robust SQL-based querying, many types of search, analytics, and more. Scalability allows data to grow without sacrificing performance and stability.

Supporting SQL and NoSQL developers

Traditional relational systems manage tabular data and return it as rows and columns. NoSQL databases can do this without forcing application developers into using a static tabular schema that has to be reworked every time there is a change. Instead, NoSQL databases give developers the flexibility they need to help them excel at their work and create adaptive applications that adjust to future needs.

NoSQL systems hold hierarchical JSON data, but return it to the application as full or partial JSON data structures, full-text search matches, SQL query results, key-based values, big data analytics systems, as well as AI vector searches.

This convergence of the best of relational and the best of modern NoSQL simplifies the information architecture of enterprises and helps developers deliver applications more efficiently with familiar concepts and tooling, without needing to learn and manage a dozen different platforms.

Scaling beyond SQL databases

To operate at scale, NoSQL systems approach cluster-based computing with efficient, automatic cluster management to keep data synchronized and flowing at high speeds.

For example, teams are now expected to build enterprise data management infrastructure that includes the following characteristics:

- Support large numbers of concurrent users (hundreds of thousands to millions)
- Deliver highly responsive experiences to a globally distributed base of users
- Be always available – no downtime
- Handle semi- and unstructured data
- Adapt rapidly to changing requirements with frequent updates and new features

SQL-based relational databases are unable to meet these requirements efficiently and cost-effectively.

Consider just a few examples of Global 2000 enterprises that are deploying NoSQL for mission-critical applications that have been featured in recent news reports:

- **Tesco**, Europe's No. 1 retailer deploys NoSQL for e-commerce, product catalog, and other applications
- **Domino's** uses NoSQL to formulate highly targeted campaigns to incentivize customers
- **Ryanair**, the world's busiest airline uses NoSQL to power its mobile app serving over 3 million users
- **Marriott** hosts 30 million documents, accessed at 4,000 transactions per second
- **GE** deploys NoSQL for its Predix platform to help manage the industrial internet
- **PepsiCo** uses NoSQL so that 30,000 users can perform operations without disruption

TODAY'S TRENDS – TOMORROW'S CHALLENGES

Today's customer experience goals depend on tightly aligned technical integration more than ever before, but it must be able to adapt to trends going forward or risk becoming outdated.

Some of the high-level technical goals include:

- Consolidated platforms that work together efficiently
- Simplified system architectures that are easy to manage
- Effective data "plumbing" for real-time web applications and low latency
- Distributed service layer that delivers data as close to the customer as possible
- Enhanced user interactions through integrated enterprise AI solutions

Five advantages of NoSQL databases

Ambitious customers with big ideas for how to use data have unleashed a new set of technology requirements for CIOs and technical leaders.

Here are five trends that play into the advantages of NoSQL databases for addressing the challenges of building software.

Trends	Requirements
Customer shift continues online	• Scaling to support thousands or even millions of users • Meeting UX requirements with consistently high performance • Maintaining availability 24 hours a day, 7 days a week • Deliver hyper-personalized experiences
The internet is connecting everything	• Supporting many different applications with different data structures • Ensuring software is "always on" with no excuse for downtime • Supporting continuous streams of data from the real-time web
Big data is getting bigger with AI	• Storing customer-generated semi- and unstructured data • Storing different types of data from different sources in the same infrastructure or even the same cluster • Storing data generated by thousands or millions of customers, and IoT devices • Integrating with external data systems powered by AI and large language models (LLMs)
Applications are moving to the cloud	• Scaling on demand to support more customers, and store more data • Operating fully managed applications on a global scale to support customers worldwide • Minimizing infrastructure and operating costs, achieving a faster time to market
The world has gone mobile	• Creating "offline-first" apps – network connection not required • Synchronizing mobile/edge data with remote databases in the cloud • Supporting multiple mobile platforms with a single backend

"We chose Couchbase for its memory-first architecture, which enables us to perform writes of our catalog at peak load consistently. Instead of doing it every two hours we can do it every 15 minutes."

—MRITUNJAY SINGH,
LEAD CONSULTANT, BT

Case study:
www.couchbase.com/customers/bt

The above requirements are extensive and challenge even the best systems to do even more with less. Today's extreme requirements can be loosely grouped into two categories that impact two different levels of end users:

- Providing agile platforms for application developers to excel
- Supporting scalable system architectures that outperform others

DEVELOPER AGILITY

"For us to set up a resilient implementation of Couchbase, it took us minutes. We stood up three servers, lit'em up, balanced the load, and instantly we were resilient."

—ROBERT LAWRENCE,
PRODUCT OWNER,
DIGITAL CATALYSRS, PG&E

Case study:
www.couchbase.com/customers/PG&E

To remain competitive, enterprises must innovate – and now they have to do it faster than ever before. Developers are under extraordinary pressure. Speed is critical, but so is agility, since these applications evolve far more rapidly than legacy applications. Relational databases have a restricted, flat data structure and don't respond well to frequent changes in the data model. This often impedes the needs of modern, agile projects, applications, and business requirements.

Application developers that are used to querying and performing transactions with SQL can continue to use the same language in NoSQL platforms but operate against the JSON data that is stored. For example, Couchbase provides a SQL-based query standard known as SQL++ that returns results in JSON with sets of rows and subdocument components where appropriate. This is in contrast to the vast majority of other NoSQL databases (like MongoDB™) that don't use SQL and require developers to climb a new language learning curve.

The combination of transactional capabilities and SQL greatly expands the number of use cases where a NoSQL database can be considered. In the past, the inability to join or handle transactional operations meant that NoSQL databases were only chosen for the highest volume and scale use cases. But the option of using SQL and transactions means that NoSQL databases can also be chosen for traditional database use cases that need more flexibility and power.

NoSQL databases operate as a multipurpose primary content store, meaning you enter the data in one application but can access it multiple ways depending on the use case. For example, developers can use direct API calls to access a specific document using a key or through a SQL query that returns multiple rows of data in a JSON response. This is known as "multi-model."

Other access methods are available depending on the database, including robust search systems for AI vector searching of LLMs that find similarity and context matches in data. There are also full-text search systems that allow natural language search requests. Requests can be made for full or partial "fuzzy" matches, geographic ranges, or wildcard searches. The response includes a JSON document with lists of matching document IDs, contextual information, and a relevancy score.

Search systems are often separate from a database but NoSQL databases like Couchbase include them as part of the underlying system, allowing managers to simplify the overall architecture.

Big data analytics are possible as well, using complementary subsystems that process larger volumes of historical data. Using advanced indexing and query

capabilities, also based on SQL++, more advanced analytics can be done in the same database without needing a separate, external OLAP system.

Because the data is stored and indexed all within the one database product, it allows developers to connect to one multipurpose system and pass through the relevant requests.

OPERATE AT ANY SCALE

Databases that support web, mobile, AI, and IoT applications must be able to operate at any scale. While it is possible to scale a relational database like Oracle (using, for example, Oracle RAC), doing so is typically complex, expensive, and not fully reliable. With Oracle, for example, scaling out using RAC technology requires numerous components and creates a single point of failure that jeopardizes availability.

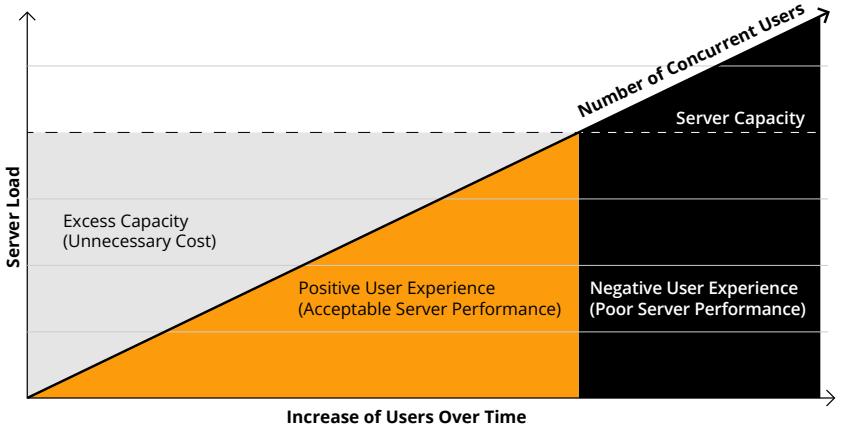
By comparison, a NoSQL distributed database – designed with a scale-out architecture and no single point of failure – provides compelling operational advantages.

Elasticity for performance at scale

Applications and services have to support an ever-increasing amount of users and data – hundreds to thousands to millions of users, and gigabytes to terabytes of operational data. At the same time, they have to efficiently scale to maintain performance.

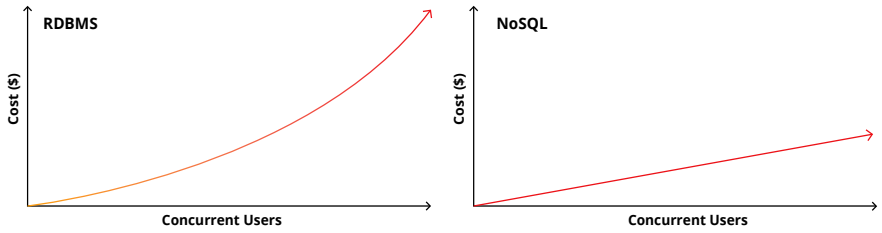
The database has to be able to scale reads, writes, and storage. This is a problem for relational databases that are limited to scaling up – i.e., only being able to scale by adding more processors, memory, and storage to a single physical server (sometimes known as "vertical scaling"). As a result, the ability to scale efficiently, and on demand, is a challenge. It becomes increasingly expensive because companies have to purchase bigger and bigger servers to accommodate more users and more data. In addition, it can result in downtime if the database has to be taken offline to perform hardware upgrades.

RDBMS – The server is too big or too small, leading to unnecessary costs or poor performance



A distributed NoSQL database runs on commodity hardware to **scale out** – i.e., add more resources simply by adding more servers to a cluster (sometimes known as "horizontal scaling"). The ability to scale out enables companies to scale more efficiently by (a) deploying no more hardware than is required to meet the current load; (b) applying less expensive hardware and/or cloud infrastructure; and (c) scaling on demand and without downtime.

NoSQL – Add commodity servers on demand so the hardware resources match the application load



ALWAYS-ON AVAILABILITY

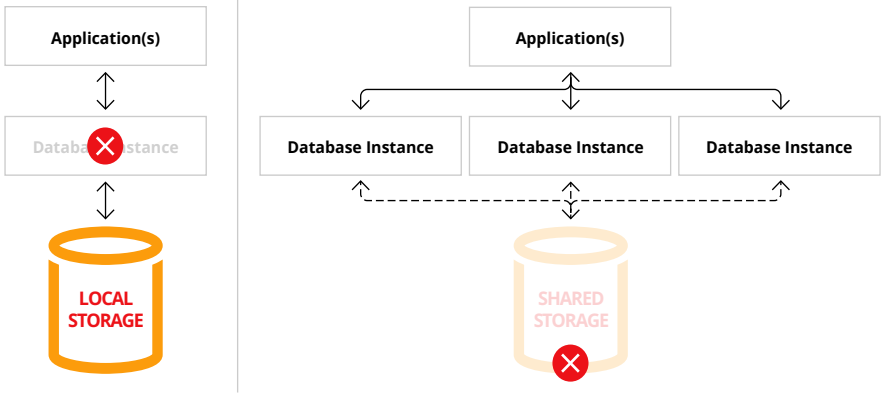
In contrast to relational technology, a distributed, NoSQL database partitions and distributes data to multiple database instances with no shared resources. Couchbase goes a step further and does this automatically.

By distributing reads, writes, and storage across a cluster of nodes, NoSQL databases are able to operate at any scale. Additionally, they are designed to be easy to configure, install, and manage both small and large clusters.

Availability for always-on, global deployment

As more and more customer engagements take place online via web and mobile apps, availability becomes a major concern. These mission-critical applications have to be available 24 hours a day, 7 days a week – no exceptions. Delivering 24x7 availability is a challenge for relational databases that are deployed to a single physical server or that rely on clustering with shared storage. If the single server or shared storage fails, the database becomes unavailable, applications stop, and customers disengage.

RDBMS – The failure of a server or storage device brings down the entire database

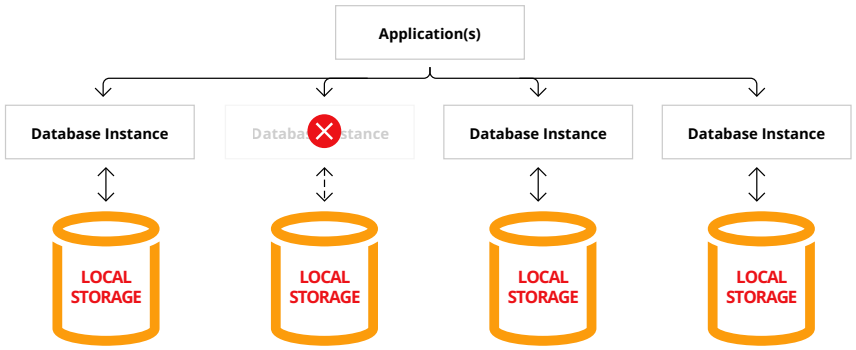


In contrast to relational technology, a distributed, NoSQL database partitions and distributes data to multiple database instances with no shared resources. Couchbase goes a step further and does this automatically.

In addition, the data can be replicated to one or more instances for high availability (intercluster replication) and in different geographic locations. While relational databases like Oracle require separate software for replication, for example, Oracle Active Data Guard, NoSQL databases do not – it's built in and it's automatic. Couchbase's cross data center replication (XDCR) feature also provides this automatically.

In addition, automatic failover ensures that if a node fails, the database can continue to perform reads and writes by sending the requests to a different node. Again, Couchbase will perform this failover recovery and rebalancing automatically.

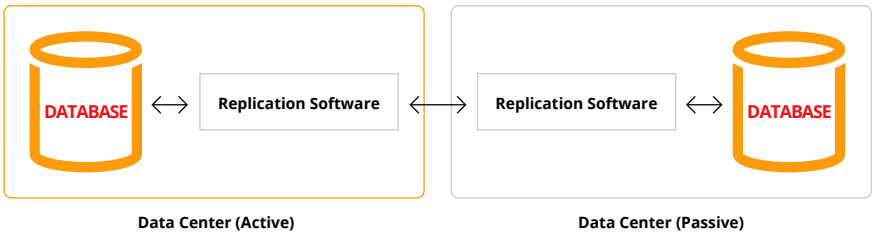
NoSQL – If an instance fails, the application can send requests to a different one



Customer behavior often requires organizations to support multiple physical, online, and mobile channels in multiple regions and often multiple countries. While deploying a database to multiple data centers increases availability and helps with disaster recovery, it also has the benefit of increasing performance too. All reads and writes can be executed on the nearest data center, thereby reducing latency.

Ensuring global availability is difficult for relational databases where separate add-ons are required – which increase complexity – or where replication between multiple data centers can only be used for failover, because only one data center is active at a time. Oracle, for example, requires Oracle GoldenGate. When replicating between data centers, applications built on relational databases can experience performance degradation or find that the data centers are severely out of sync.

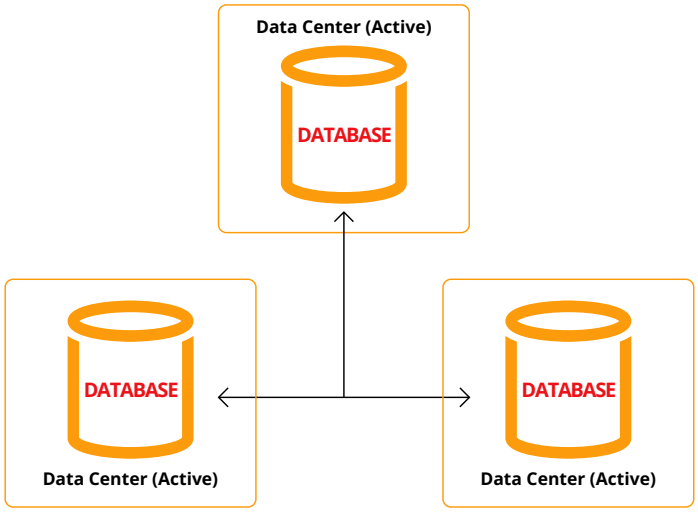
RDBMS – Requires separate software to replicate data to other data centers



A distributed, NoSQL database includes built-in replication between data centers – no separate software is required. In addition, some include bi-directional replication enabling full active-active deployments to multiple data centers. This enables the database to be deployed in multiple countries or regions while providing local data access to local applications and their users.

Deploying to multiple data centers not only improves performance, but enables immediate failover via hardware routers. Applications don't have to wait for the database to discover the failure and perform its own failover.

NoSQL – Replication between data centers is fully built-in and can be bi-directional



DATABASE-AS-A-SERVICE

Many companies are looking to reduce their operational efforts and costs of running software and hardware, with databases increasingly becoming a common area for consideration. Typically a Database-as-a-Service, or DBaaS, streamlines and improves operations in areas like IaaS setup and configuration, database provisioning, operations management, scaling automation, monitoring, and security.

Operational management reduces many of the tasks of maintaining a database environment, allowing companies to focus more time and effort on core business activities. These operational processes can include ongoing configuration, patching, upgrades, backup and recovery activities, and overall system monitoring.

BENEFITS OF DBAAS

BENEFITS OF DBAAS

- Rapid setup
- Easily scale or evolve configurations
- High service levels
- Security automation

DBaaS offerings limit the work of often overstretched IT teams, providing convenience and opportunities to focus on more high-value projects. New databases can often be spun up in minutes instead of a traditional multi-week provisioning process.

From a financial and business perspective, companies see benefits like:

- **Rapid setup –** DBaaS capabilities allow users to provision new database instances when needed in a self-service, highly automated fashion. Developers can more quickly test out new projects to drive company innovation.
- **Easily scale or evolve configurations –** As the needs of users and applications change, modify the configuration of clusters to match those needs. The database makes it easy to match regional needs and keep up with regulatory changes.
- **High service levels –** Most DBaaS systems provide at least a 99% uptime SLA, and often much higher. Replication and redundancy architecture improve quality even further as deployments go global.
- **Security automation –** Advanced DBaaS systems build in multiple levels of security and encryption to protect data at rest, in transit, and throughout the data's lifecycle.

NOSQL IS A BETTER FIT FOR LARGE-SCALE REQUIREMENTS

As enterprises shift to cloud, mobile, social media, AI, and big data technologies, developers, architects, and operations teams have to build and maintain web, mobile, and IoT applications faster, and at a greater scale. NoSQL is increasingly the preferred database technology to power today's web, mobile, IoT, and AI-powered applications.

Hundreds of Global 2000 enterprises, along with tens of thousands of smaller businesses and startups, have adopted NoSQL. For many, the use of NoSQL started with caching, a proof of concept, or a small application, then expanded to targeted mission-critical applications. Today, the Couchbase NoSQL database serves thousands of these types of customers.

With NoSQL, enterprises are better able to both develop with agility and operate at any scale – and deliver the performance and availability required to meet the demands of businesses.

Couchbase

Modern customer experiences need a flexible database platform that can power applications spanning from cloud to edge and everything in between. Couchbase's mission is to simplify how developers and architects develop, deploy and consume modern applications wherever they are. We have reimagined the database with our fast, flexible and affordable cloud database platform Capella, allowing organizations to quickly build applications that deliver premium experiences to their customers – all with best-in-class price performance. More than 30% of the Fortune 100 trust Couchbase to power their modern applications.

For more information, visit www.couchbase.com and follow us on Twitter.