E-Book

Real-Time Applications Need Real-Time Data: The Importance of In-Memory NoSQL Databases

Table of Contents

Introduction

Recommending or choosing a data platform?

This e-book is for you. Whether you're an IT decision maker, DevOps engineer, architect, or other IT professional, this e-book was designed with your questions in mind. We'll help you understand what it takes to deliver real-time data today and why an in-memory NoSQL database is best suited to do so.

While data is an asset, it has no inherent value just sitting in a database. Instead, your business creates value by generating, collecting, and using data, and the faster you can do that, the more value you can create in today's digital world—which translates into more customers, revenue, competitive advantage, operational efficiencies, and market opportunities.

Customers expect instantaneous responses, with very few people these days willing to tolerate slow digital experiences. In fact, a lack of speed is the number one driver of consumer dissatisfaction with digital services. Which is another reason why real-time data—at scale—is fundamental for success.

What is real-time data? It's when a database responds to an application in less than 1 millisecond.

You read that right. The database must be so fast as to not add any delay in the end-to-end response time to the user.

It's real-time data that powers real-time applications such as gaming and entertainment, online shopping, fraud detection, customer 360, personalization, and many others. Can your current databases provide real-time data at scale or is it time to look for a much faster alternative for your projects and applications?

Chapter 1

Slow Response Is the Top Reason Customers Leave

What's slow and what's fast in real life is vastly different from what people expect today in the digital world.

Think about it: If you waited one second for coffee at Starbucks, that would be extraordinarily fast. If it took one second to purchase a new t-shirt at your favorite retail store, that would be incredibly fast. A one-second wait for your doctor's appointment? That would make anyone happy.

In the digital world, waiting one second for an application to respond feels so slow. We expect digital services to feel instantaneous and one second doesn't even come close.

How bad is it to deliver slower than expected digital experiences? Of the top five problems that customers encounter with digital services, slow response is the driving factor for two of them and is the number one problem reported by customers.

Top five problems encountered with digital services

1. Slow page/screen loading
2. Poor connectivity
3. Crashing/stopped working
4. Slow streaming of content/media
5. Slows down device/drains battery

Consumers are not tolerant of slow experiences

61% of people state their expectation of digital services has changed forever and they will no longer tolerate poor performance.

Source: "The App Attention Index 2021," AppDynamics

Chapter 2

Developers' Nemesis: Too Much Latency and Not Enough Throughput

While latency has always been an issue for application performance, it takes on even more importance today when modern use cases and customer expectations demand data in real time.

In its simplest definition, latency is the time (delay) between an action and a response. In the real-time data discussion, latency represents the time between a user's action and a response back to the user from the application. Total latency is the time elapsed for the roundtrip from user to application to database and back again to the user. Throughput is the rate of production or the rate at which something is processed.

How much latency is too much? User performance data shows that even the *smallest fraction of a second* in page load time can change customer focus and interaction.[2]

Latency is the new outage

For more insight into why slow products and services can be damaging to the bottom line, read this **white paper**.

[2] "Milliseconds Make Millions," Deloitte, 2020

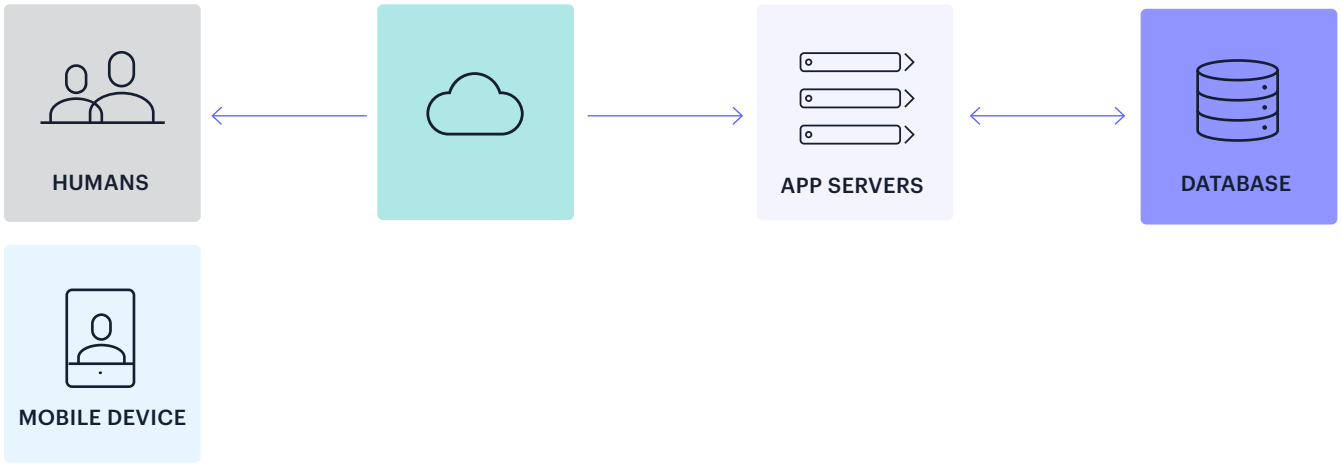
Chapter 2

In fact, the goal should be for total turnaround time to be 100 milliseconds or less. Why? Usability expert Jakob Nielsen states that 0.1 second (100 milliseconds) is the upper limit for the user feeling that a system is responding instantaneously. Users will perceive having a real-time experience when the end-to-end response time is less than 100 milliseconds.

[3] "Usability Engineering," Jakob Nielsen, 1993

Figure 1: Latency is the time between a user's action and a response back to the user from the application, which includes the roundtrip from user to application to database and back again to the user.

The Components of Latency



Chapter 3

Google ranking: 100 milliseconds or less for interactivity

Google's page experience update in June 2021 was designed to highlight pages that offer "great user experiences" based on signals that include Google's three Core Web Vitals—metrics that measure loading, interactivity, and visual stability.

One of those metrics is first input delay (FID), which measures interactivity (the time from when a user inputs an action and the page executes it). An FID of 100 milliseconds or less is considered to provide a good user experience. Anything greater than 300 milliseconds is considered to be a poor experience.

Source: "More Time, Tools, and Details on the Page Experience Update," Google Search Central blog, April 2021; "Evaluating Page Experience for a Better Web," Google Search Central blog, May 2020

One Millisecond Response– Can Your Database Do That?

What does 100 milliseconds look like for today's applications? In general, we can assume that half of that time is network time (50 milliseconds from the user to the application and back again) and the other half is the datacenter/cloud infrastructure time (50 milliseconds between the application server and the database).[4]

But what about the database? As a rule of thumb when running applications in production, every millisecond spent in the database means 100 milliseconds felt at the application level. From a real-time data perspective, this means that the database needs to be so fast that it doesn't add any delay in the communication to and from the application because latency is already at the maximum to appear instantaneous for users.

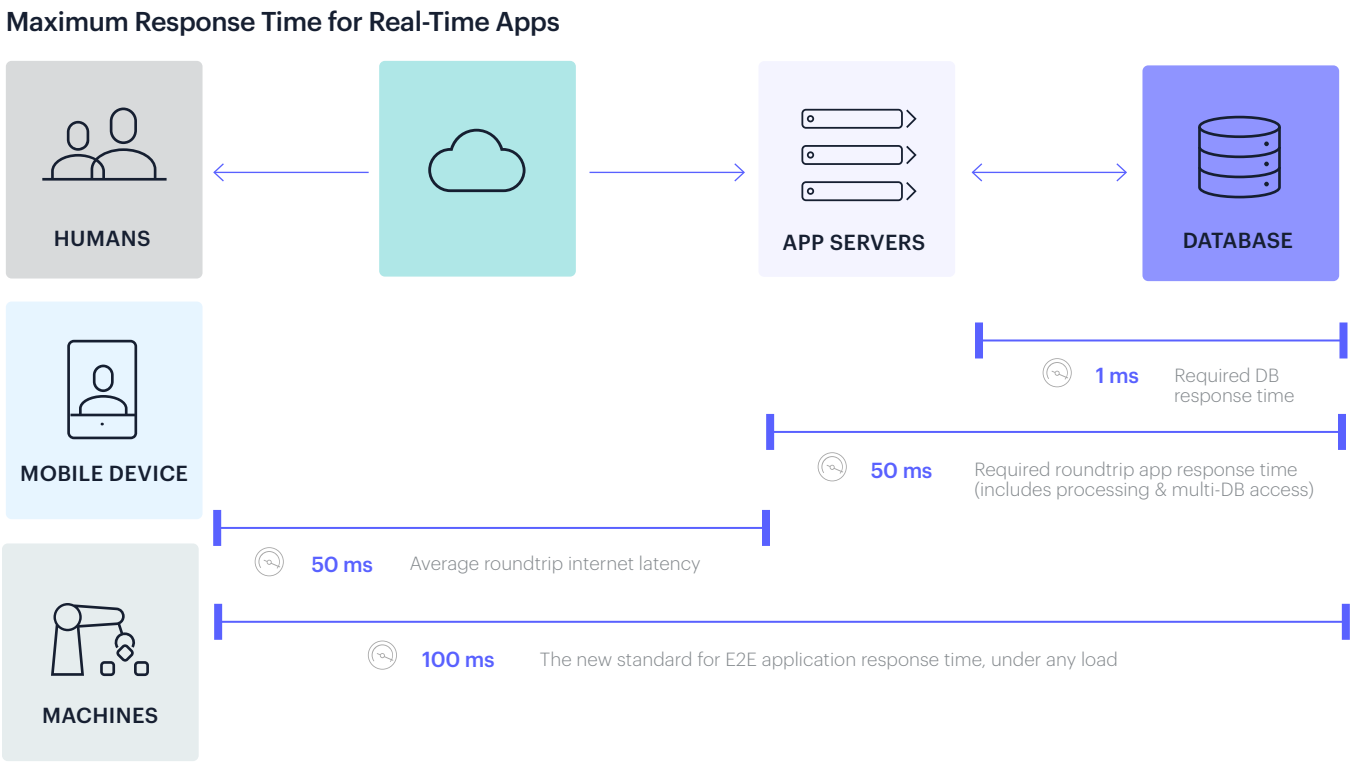
Therefore, the database must respond in less than one millisecond. And it must do so at scale, supporting millions of operations per second to meet the demands of today's use cases such as gaming or online shopping.

As many software teams are discovering, this is simply not feasible with legacy or disk-based databases.

[4] Based on data and analysis from Redis from more than 10 years in production across all public clouds, 1.5 million databases under management, and more than 8,500 paying customers

Chapter 3

Figure 2: In the modern application equation, databases have to respond in less than 1 millisecond to give users end-to-end real-time responsiveness.



Chapter 4

A Job for NoSQL Databases

What is a NoSQL database?

A NoSQL database is a distributed, non-relational database designed for large-scale data storage and for massively parallel, high-performance data processing using a scaleout architecture. It is a modern data storage paradigm that provides data persistence for environments where high performance is a primary requirement. Within a NoSQL database, data is stored in such a way that both writing and reading are fast, even under heavy load.

What kind of database can deliver less than one millisecond response time? It's certainly not a legacy relational database. Raise your hand if you've had slow queries against relational databases dragging down digital performance.

Traditional SQL/relational databases were designed for legacy data types that are highly structured and easily organized into defined fields, columns, and tables. These types of databases don't lend themselves to efficient or effective support of unstructured and semi-structured data relied upon by modern applications.

That's why NoSQL, non-relational databases have become far more popular. In fact, thanks to a major acceleration in development and expansion in the past decade or more, today NoSQL databases are the first choice of developers and architects for applications that require:

Superior performance

A flexible data model supporting different data types

High scalability

True high availability with built-in replication

Interconnectivity between applications and between microservices

Chapter 5

A NoSQL Database in Memory? Now That's Real-Time Data

Even with the speed advantages of a NoSQL database, if it's a pure disk-based system, it can't deliver the less than one millisecond latency needed to meet the expectations of instantaneous response times for customers.

But what if you put the NoSQL database in random access memory (RAM)? What if you had the entire dataset in RAM?

You'd have no more hard-disk drives that cost hundreds of milliseconds or more of delay. Combining the benefits of NoSQL databases with the ability to keep data in-memory gives you the less than one millisecond response time your company needs for real-time data.

That's why businesses of all kinds—financial services, e-commerce and retail, healthcare, logistics, entertainment and gaming, and others—are embracing in-memory NoSQL databases to power their real-time applications, deliver the kind of fast experience customers expect, and outperform the competition.

Advantages of an in-memory NoSQL database

Incredible speed

Linear scalability with consistent performance (real-time data at scale)

High availability

Plus all the other benefits of a NoSQL database

Chapter 6

Powered by Real-Time Data

While relational databases will certainly continue to have a place in your technology stack, usage of in-memory NoSQL databases is growing at a faster pace. That's because they deliver the speed at scale that's needed for a broad set of digital use cases.

Popular use cases for real-time data powered by in-memory NoSQL

Enterprise caching	Improve response time for existing applications by storing copies of the most frequently used data in memory instead of on disk.
Session management	Read and write session data at each user interaction without slowing down the user experience.
Gaming	Deliver fast performance from anywhere in the world regard less of the number of simultaneous players.
Leaderboards	Update, track, and calculate all the data and statistics users want to see on a leaderboard in real time across hundreds of thousands or millions of users.
Real-time inventory	Survive seasonal peaks, maintain data consistency, and deliver instantaneous results with a real-time inventory system.
Fraud detection	Analyze transactions in real time and compare them with geodata and previous spending patterns to rapidly identify potential fraud without impacting the customer experience.
Claims processing	Power real-time claims inquiries, fast member lookups, auto claims adjudication, and more in memory to deliver a great customer experience.
Personalization	Update customer profiles in real time for dynamic pricing, credit-risk analysis, targeted advertising, credit card promotions, and more.
Real-time analytics	Unlock insights from large volumes of data, drive powerful artificial intelligence applications, and respond quickly to trends and drivers.

Chapter 7

Beyond Speed and Scale: Other Enterprise Requirements

While an in-memory NoSQL database is the only way to achieve the high throughput and low latency required for real-time data, there are other requirements that are also important when considering a database platform for the enterprise.

Here's your checklist for an enterprise-ready, in-memory NoSQL database. To get the most from your real-time database, it should be able to check all of these boxes.

Is the in-memory NoSQL database ...

Capability	Why this capability is important
☐ **Durable?**	To make sure transactions that have been committed will survive permanently, your in-memory NoSQL database needs to be fully durable, without affecting performance even under heavy write loads. Look for support for data persistence using network-attached storage and mechanisms such as Append-Only File (AOF) data persistence and snapshot persistence.
☐ **Globally distributed?**	For local high throughput and low latency as well as a disaster-proof architecture, your in-memory NoSQL database should support geo-distributing applications in both Active-Active and Active-Passive replication deployments.
☐ **Highly available?**	In today's digital world, highly available is defined as five-nines, available 99.999% of the time. To prevent loss of data or operations, an in-memory NoSQL database needs to offer uninterrupted high availability with diskless replication, instant failure detection, and single-digit-seconds failover across racks, zones, and geographies.
☐ **Flexible to support a variety of data models?**	Supporting different real-time data use cases requires a variety of ways to model the data. To minimize any impact on performance, throughput, and latency, your in-memory NoSQL database should support multiple data types and models—such as Strings, Hashes, Lists, Sets, Sorted Sets, Bitmaps, Bitfields, HyperLogLog, Geospatial indexes, and Streams—in a single platform.
☐ **Flexible in its deployment model?**	Flexibility also applies to deployment, with support for any cloud platform, on premises, as well as multi-cloud and hybrid architectures.
☐ **Secure?**	Every database platform should offer robust, multi-layer security and compliance safeguards, including role-based access control, authentication, authorization, and encryption of data in flight and at rest.

Chapter 8

Customer Engagement at Real-Time Speed

"We have access to the highest in-memory performance available on the market today, flexible data structures for extreme efficiency across a wide variety of use cases, and fully managed operations that speed up—rather than slow down—application delivery."

— Naren Janakiraman, Director of Engineering, Freshworks

A leader in customer engagement software, Freshworks offers a suite of cloud-based business products. Its flagship offering Freshdesk is used by more than 50,000 organizations worldwide, including Bridgestone, NYU, and American Express. The company's initial public offering saw the company value reach $12 billion.

Freshworks has seen impressive year-over-year growth, with more than $300 million in annual recurring revenue. This extraordinary growth, spurred by the rapid adoption of Freshdesk and the addition of seven other new products, strained the capabilities of the company's application architecture and development operations and led it to adopt Redis Enterprise Cloud, a fully managed in-memory NoSQL database.

Today, Freshworks relies on Redis Enterprise to support many different use cases, including:

- **Caching:** Provides a front-end cache for Freshworks' MySQL database
- **Metering:** Meters API requests coming into Freshdesk for rate-limiting purposes
- **Persistent data store:** Serves as a persistent store for background jobs, stored on disk, as well as the rate-limiting meter
- **Session store:** Provides a session store for an authentication microservice
- **Real-time analytics:** Delivers real-time data as a frontend database for user analytics

Chapter 9

The Last Word

Choosing a new database is not a trivial exercise—nor should it be. With all the choices available, it can feel overwhelming.

However, given the state of customer expectations today and the knowledge that they'll continue to rise even higher, the choice of database has never been more important.

Before you begin your next project, think about the throughput and latency requirements. What would real-time response mean for your application? For your customers? For your business?

Then learn more about in-memory NoSQL database platforms by starting with these resources:

Nine Essential Database Capabilities
redis.com/docs/9-essential-database-capabilities

Databases are expected to do more than ever. They have to work with mountains of disparate data, return results in the blink of an eye, and cope with hugely complex technology environments, all while remaining easy to work with and highly available. While many databases exist that have just one or even a few of these critical features, having all nine is vital to successfully manage today's challenges.

Make sure your database has what it takes to meet all your demands—no matter how tough.

Linear Scaling Benchmark
redis.com/docs/linear-scaling-benchmark-50m-ops-sec

Redis Enterprise supports the open source Cluster API, enabling infinite, linear scaling by simply adding shards and nodes. This, combined with the shared-nothing symmetric architecture of Redis Enterprise cluster, ensures that data and control path are separate and the control path does not impose non-linear overhead in a scaled-out environment.

Chapter 10

The World's Most Loved NoSQL Real-Time Database

Redis Enterprise is a robust in-memory NoSQL database platform built by the people who develop open source Redis. It maintains the simplicity and high performance of Redis, while adding many enterprise-grade capabilities, such as linear scaling to hundreds of millions of operations per second, Active-Active geo-replication with local high throughput and low latency, Redis on Flash to tier data across dynamic and persistent memory and solid-state disk (SSD) to reduce total cost of ownership, and five-nines (99.999%) uptime based on built-in durability and single-digit-seconds failover.

Redis Enterprise supports many data modeling methods with modules such as RediSearch, RedisJSON, RedisGraph, RedisTimeSeries, RedisBloom, and RedisAI, and allows operations to be executed across and between modules and core Redis functionality using RedisGears, a serverless engine that runs across shards and nodes of Redis Enterprise cluster. All this while keeping database latency under one millisecond, so your application can respond in real time.

Find out more at **Redis.com**

About Redis

Data is the lifeline of every business, and Redis helps organizations reimagine how fast they can process, analyze, make predictions, and take action on the data they generate. Redis provides a competitive edge to any business by delivering **open source** and **enterprise-grade** data platforms to power applications that drive real-time experiences at any scale. Developers rely on Redis to build performance, scalability, reliability, and security into their applications.

Born in the cloud-native era, Redis uniquely enables users to unify data across multi-cloud, hybrid and global applications to maximize business potential. Learn more about Redis at **redis.com** and sign up for **your free trial**.