E-Book

Buyer’s Guide for Real-Time Search Engines

Find the right database search tool for your needs

redis

Table of Contents

What is real-time search, and why is it important?

Nearly every application that stores data—which is just about every one of them—also needs a way to find and retrieve that data. Applications regularly perform queries on databases to find specific records so that the application logic can transform data (such as calculate or aggregate it) and then often present it some way (such as via a visual user interface).

There are several ways to look for data programmatically. Sometimes, datasets are small and simple so it's easy for applications to provide acceptable query results. In many cases, however, it's critical to provide results with urgency and to search through massive amounts of structured and unstructured data. This can be accomplished using "real-time search," which refers to high-performance searches or queries that return results with no delay perceived by the person requesting them.

A real-time search engine is an application or service whose only purpose is to search or query rapidly changing data. Search engine databases use indexes to categorize similar characteristics among datasets, which facilitates extremely fast searches and queries.

We use the terms "search" and "query" interchangeably, but there is an important distinction. Searches are typically open ended and seek unknown results; in contrast, queries are formal database expressions used to find expected or exact results.
Search engine databases are highly optimized for search and query by using primary and secondary key indexes. The search engine database typically offers specialized methods such as full-text search, complex search expressions, mathematical functions (e.g., aggregation), and ranking of search results. Developers and architects turn to real-time search for many reasons but mostly to provide the high performance needed to keep users happy.

Four attributes of real-time search solutions

Among those reasons are these four key attributes of real-time search that delivery on their promises:

- Data ingestion performance
- Data freshness
- Concurrent updates
- Query latency

Ingest Performance

However it's accomplished, data ingestion is the process of obtaining and importing data, whether for immediate use or storage in a database. The speed by which raw data is ingested is critical for anything done in real time. Whether the concern is real-time search, online features stores, or training machine-learning models, all of these require fast queries and access to fresh data.

Beyond search and query capabilities, in-memory databases can overcome the following real-time data ingest challenges:

- Ingest massive volumes of data and scale to millions of writes per second with sub-millisecond latency
- Ingest data from multiple sources and format types while retaining source identity
- Filter, analyze, and aggregate massive amounts of data
- Support asynchronous data streams between producers and consumers despite connection loss and hardware failures
- Respond to physical latency issues, such as when data comes from geographically distributed sources

Data freshness

To have value, data must be current and immediately usable. If an application requires a step in which you have to pre-calculate, aggregate, or manipulate data before using it, it may not be fresh, which raises questions about its accuracy. For example, consider inventory: if inventory counts are older than a millisecond, it would be possible to sell two items although only one is in stock. Many developers and database technologies rely on pre-aggregation of data fields to improve query latency, but doing so adds delay and does not support providing metrics in real time.

Concurrent updates

Data concurrency allows multiple users to process reads and writes simultaneously within a single database structure. It's common to find analytic use cases that require 1,000 queries-per-second (QPS) performance. At the same time, a common developer goal is to provide application users with sub 200ms response times to ensure high-quality experiences.

The desired goal is for low P99 latency, which means 99% of requests are faster than 400ms or only 1% of the requests are slower. This is where in-memory databases support real-time search use cases. An in-memory database can consistently scale excellent response times by distributing the concurrent search load across multiple database instances, partitions or shards.

Query latency

Latency measures the delay between a user's action and the response to that action. Traditional relational databases are slow to process complex queries on large datasets, and they bog down in performance. That's particularly so when the databases have keys with high cardinality fields, or columns populated with unique data, acting as a key. These queries are expensive, use lots of IT resources, and are too slow to be valuable. In contrast, in-memory databases can efficiently provide the search and query response times for real-time use cases.

Evaluating different real-time search solutions

Developers and application architects are turning to in-memory databases to provide a frontend caching layer to respond to application or user experience performance problems. Often the caching layer is used to overcome the shortcomings of backend data infrastructure. Savvy developers are realizing the value in-memory databases have beyond caching. They are using in-memory databases to drive real-time search across multiple data types and to transform the data for customer experience or analytic use cases. Here are some questions to ask yourself before building a real-time solution:

- Do you need to perform complex searches, high-speed master data lookups, or aggregations to support real-time applications or operational analytics?
- How does your organization define “real time” for this use case: in minutes, seconds, or zero latency?
- Where does your data come from? Pay attention to disparate sources and data types, user input, Hash files, JSON documents, message streams, graph, time series, Bloom Filters, other databases, or application APIs?
- Are you seeking a real-time search engine for vector similarity, MLOps, or AI/MLsystems?
- Do you support thousands or tens of thousands of input/output operations per second (IOPS)?
- Do you need to store, sort, and search data on secondary keys or dimensions in real time?
- Can you ingest all the data required within the allotted time window for search?
- Are you implementing an enterprise feature store that needs low latency and the ability to scale thousands of features?
- Do you need to replicate your real-time search in multiple regions to ensure performance and availability?
- How important is it for you to eliminate database sprawl and simplify your full stack and data architecture?
- Can your network handle the additional load and still provide low latency and high throughput?

Key features buyers should look for

Nothing is more critical to application performance than data interchange among applications, machines, and humans. Real-time search solutions support high-performing data interchange with these key capabilities:

Indexing

- Declarative and incremental indexes
- Multiple field indexing
- Secondary index structures to provide rapid search results for any numeric data, text values (tags), and geolocations
- Synchronous document indexing and inverted index structure to support full-text search and stemming
- Automatic indexing and garbage collection

Indexing benefits

□ **Secondary Indexing**

- Find documents without knowing a primary key
- Search on all secondary (non-primary) fields
- Query flexibility with complex nested searches
- Support for duplicate field values

□ **Synchronous document indexing**:

- Once an index is created, every write or update operation is immediately seen in the next query or search operation
- Accurate, fresh search results in real time

□ **Auto Indexing**

- Accurate search results including all recent index updates
- Ability to create thousands of indexes without management headaches

Querying

- Multi-field queries
- Numeric filters and range queries
- Geo-distance queries provides results filtered by location
- Complex Boolean queries with AND, OR, NOT operators between sub-queries
- Optional query clauses
- Aggregations

Querying benefits

□ **Basic and advanced query capabilities**

- Improve customer experiences with fast data lookups.
- Search on any secondary (non-primary) fields.

□ **Faceted search**

- Build applications with multiple filters based on faceted classification of the searched items, allowing users to narrow down the results.

□ **Native JSON ingest, update, and search**

- Native JSON data model offers high-speed search functionality.
- Supports atomic updates of document sub-elements using the JSONPath query language.

Full-text search

- Document ranking and field weights
- Prefix-based, exact phrase search, or slop-based search
- Stem-based query expansion in many languages
- Fuzzy search and fuzzy matching
- Support for custom functions for query expansion and scoring
- Limiting searches to specific document fields
- Spell-check and auto-suggest functions

Full-text search benefits

□ **Full text search**

- Compares every word of the search request against every word within the document for higher-quality text search results.

□ **Stemming**

- A technique wherein a word is reduced to its "stem" or root form, such as taking a plural word to singular, or removing other inflection types. For example, the words "skating," "skates," and "skater" are all based on the word "skate."
- Provides intelligent grouping of text search results.

□ **Fuzzy search**

- Also called approximate string matching, this is the technique of finding strings that match a pattern approximately rather than an exact match.
- Users get relevant results even when the search query is misspelled.

□ **Fuzzy matching and auto-suggestion function**

- Finds approximate matches, even when search words are misspelled or omitted.
- Provides a match score, from which it makes auto-suggestions of appropriate search terms.
- Uses Levenshtein distance, a fuzzy matching technique that measures differences between two strings, with the given number representing how far the two strings are from being an exact match.

Enterprise-grade

- Astonishing performance with sub-millisecond response time
- 99.999% availability
- Highly secure
- Supports large scale data operations; extract, transform, load (ETL); and change data capture

Full-text search benefits

□ **Performance**

- Delivers low-latency, real-time access and searching of data in multiple native formats and can operate solely in-memory.
- Flexibility to operate on NVMe or SATA SSD storage to lower costs.

□ **Enterprise-grade availability and security**

- Provides long term data persistence and global data availability.
- Supports globally distributed active-active database topologies, restore and backup with full data integrity and no data loss.

□ **Change Data Capture (CDC)**

- Tracks all data that has changed in real time so that action on query results can be immediately taken using the new or changed data.
- Supports high-speed extract, transform, and load (ETL) and other DataOps practices.

Comparing databases: SQL and NoSQL disk-based databases vs. in-memory NoSQL databases

If your search queries and data analytics are part of a performance-sensitive application, you should consider the underlying database performance. If you are using or considering using Redis for caching, it makes economic and technical sense to use the Redis built-in search engine. Redis Enterprise real-time search provides flexible secondary indexing, search, query, and aggregation capabilities. See the table below to understand key database differences supporting real-time search and query use cases.

While Redis is not designed for every single database use case, it is a very flexible and versatile database designed to store massive amounts of data. Redis is a perfect database for microservices architecture as a frontend data layer and as a primary database. If there is not a need for high performance or scale, Redis may not be the best fit. But in a microservices architecture, data is handed off from service to service, so Redis' low-latency, in-memory data model is a perfect fit. Redis real-time search ensures the microservices framework intra-communication gets all the data needed fast with no additional latency. And a Redis database has a relatively small footprint that can serve massive throughput even with minimal resources. It can act as a primary database to back a wide range of different microservices, each with its own individual database or data store.

Comparing real-time search solutions

	SQL & NoSQL Disk-based	In-Memory NoSQL Database (Redis)
Supports in-memory, NVMe, and SATA SSD storage types		●
Search, query, and aggregate Redis databases		●
Sub-millisecond latency, engineered for real time		●
Auto-indexing once index is created	●	●
Powerful native search of both hash and JSON document stores		●
Multi-purpose, low latency cache and database, shrinking full stack		●
Ease of Use: less vendors and code to learn and manage		●
Ideal for high-performance application needs; master data tables, operational analytics, and customer lookups		●
Native query syntax based on JSONPath	●	●
Supports active-active database configurations		●
Has highest Trust Radius Score		●
Linearly scale ingest to support real time data analysis		●
Process real-time microservice message streams		●
Supports deterministic Bloom Filters, Cuckoo Filters, and vector similarity		●

Table 1: Key Database Differences Supporting Real Time Search

Popular use cases for real-time search

While there are many profitable uses for real-time search, these are some of the most common applications:

- **Automation & Operational Intelligence**: A category of AI that enables automation and operational processes to be based on real-time data as it's generated or collected, such as real-time streaming of IoT manufacturing tolerance measurements. Simultaneously ingesting machine data and searching disparate data sources (such as Hash and JSON) can be very demanding when running on disk-based SQL and NoSQL databases. Since the quality of operational intelligence is directly correlated to the speed it can access data, in-memory, low-latency NoSQL databases that offer real-time search are a better choice.

- **Online Feature Stores**: Feature stores pull data from systems of record, data warehouses, and data streams and then transform that data in real time to produce features for ML models and applications. With a feature store, ML pipelines and online applications have repeatable ways to easily access data in real time to perform inferences or estimates. For example, a bank could use an offline feature store that includes the frequency and size of user transactions and IP addresses. The historical offline feature store is then compared in real time against the current online feature store to look for trends, outliers, and odd transaction patterns. In such a case, the queries and set comparisons need to happen in real time in order for humans or applications) to react and stop any fraudulent activity immediately.

- **Anomaly or Fraud Detection**: Along with feature stores, many types of real-time data models are used in the security space. Graph data models represent and store data as a series of nodes and edges that model relationships between data points, making them very useful for uncovering connections between suspicious parties. Bloom Filters check if something is present in a given set of items. For example, a Bloom Filter can be used to determine whether a given transaction was in a list of known fraud patterns or even to check a user's new password against their list of old passwords to ensure that none of them are reused.

- **Insight Engines & Real-Time Business Intelligence (RTBI)**: Leverages real-time search to provide metrics for actionable insights based on data science that can promote tactical and strategic decision-making. In-memory databases enable insight engines or RTBI tools to analyze data sets in real time and present timely analytical findings for humans to consume in dashboards, graphs, charts, or other applications, or for automated systems to leverage via application programming interfaces (APIs).
- **Personalization & Real Time Inventories**: Helps to better tailor and enhance the customer experience. Real-time search can provide an analysis of past customer behavior or current session stores, and offer helpful product suggestions, coupons, discounts, reminders, or up-to-the-millisecond accurate inventory information.

- **Telemetry & Performance optimization**: Real-time search provides instant feedback to event or incident managers if thresholds or high watermarks are breached. IoT telemetry data is ingested from core to edge, searched, and analyzed to provide real-time insights into mechanized processes.

- **Preemptive maintenance**: Real-time search aids in optimizing systems and machines by providing data for predictive failure analysis in order to make proactive suggestions to reduce or eliminate the chance of costly unplanned downtime.

How real businesses benefit from real-time search

Faster results for travelers on the go

HolidayME is a leading India-based global travel booking site that provides customized holiday packages, itineraries, flights for the Middle East and Southeast Asia. They are running more than 800 microservices, and more than 90% of them store data in Redis Enterprise with sub-millisecond latency.

> “We got better performance after making the switch to Redis. We were able to deliver results to the customer in one-third of the time. From a latency perspective, what we used to have earlier from our autocomplete was around 400 to 600 milliseconds. Then, when we used Redis, we were able to reduce that to around 200 milliseconds.”
>
> — Rajat Pandar, CTO, HolidayME

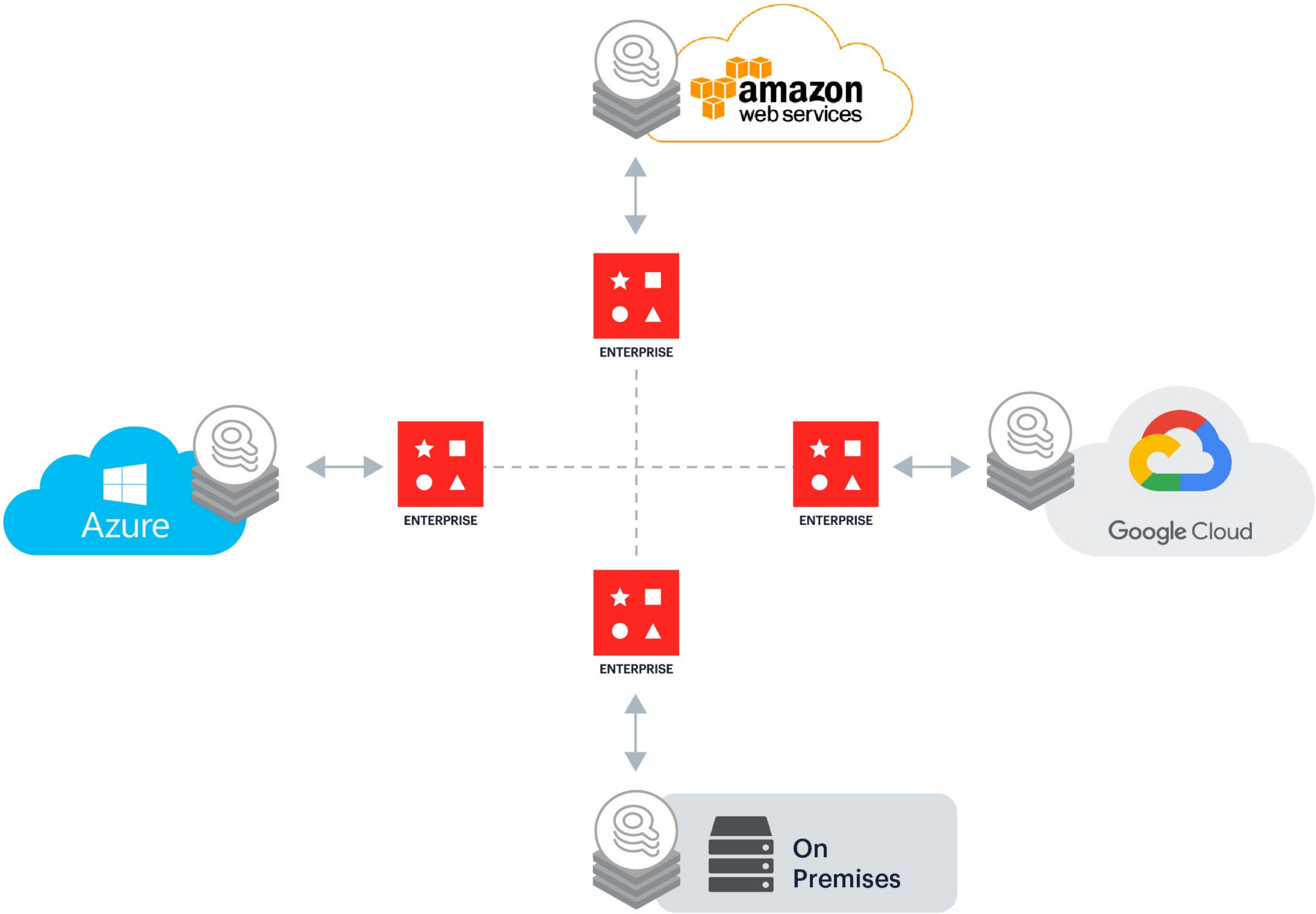
Challenges

- HolidayME sought to build a search autocomplete mechanism with low latency. The company’s existing stack of MongoDB, Apache Lucene, and Elasticsearch were too slow so it turned to Redis Enterprise real-time search to achieve that functionality.
- The company wanted to transition from an API framework to a microservices architecture in order to streamline application configuration and to enable its developers to focus on singular functionalities, not huge portions of difficult-to-understand code.

Redis Enterprise Results

- Met customer experience expectations with low-latency search, completing more than 60,000 operations within 200ms.
- Delivered fast search service that works with the company’s microservices architecture.
- With its active-active capabilities, Redis scaled global real-time search and acted as a primary database for HolidayME.

Diagram 1: Redis Enterprise's active-active implementation allows Holiday.ME to scale low latency search globally.



Brand management in real time

Yext is a NYC-based technology company offering online brand management. It provides a modern, global, AI-powered answers platform that understands natural language and delivers real-time, marketing-related analytics dashboards.

> “[Redis] is pushing out features very important to us. We use the hosted version of Redis search and because they’re able to operate in multiple clouds like we are, we can use them anywhere we have a point of presence.”
>
> — Josh Blender, Vice President of Infrastructure, Yext

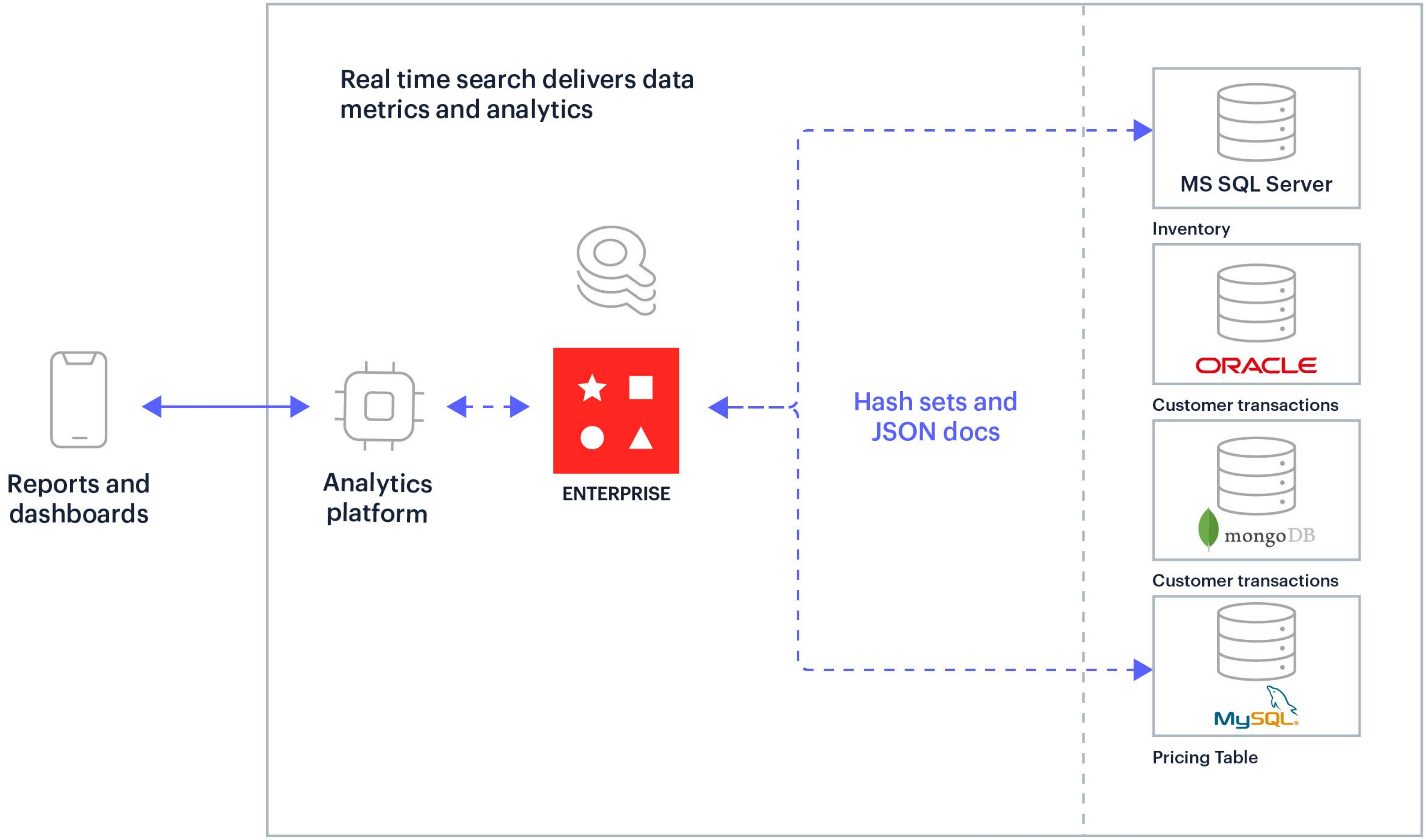
Challenges

- As a SaaS vendor, Yext needed a low-latency data platform to consolidate disparate data sources and to support distributed search and query.
- The company required low-latency query results for real-time analysis, training auto-suggest NLP engines, and other ML models to enhance the customer website experience.

Redis Enterprise Results

- Improved business and customer process for each user thanks to real-time metrics and data analysis.
- Achieved globally scalable, low-latency data delivery for fast results anywhere.
- Gained powerful, programmable auto-suggestion with intelligent fuzzy matching.
- Consolidated multiple systems of records into one real-time data platform.

Diagram 2: Deliver fresh, quality data for analytics with one real-time data platform and search engine.



On the road to a better customer experience

GoMechanic is India's largest network of technology-enabled car service centers, offering a seamless experience for car service, repairs, cleaning, and spare parts and accessories. GoMechanic is also India's #1 car servicing mobile app, trusted by over 1 million users and 1,000+ car service centers across India providing hassle-free car services and repairs with a single tap.

> "We need to load our spare parts and accessories retail sites and our customer service chats instantaneously, and RediSearch does that with ease. We tried many solutions, but Redis Enterprise on Google Cloud delivers the best performance for its cost."
>
> — Nagendra Kumar, Senior Manager of Technology, GoMechanic

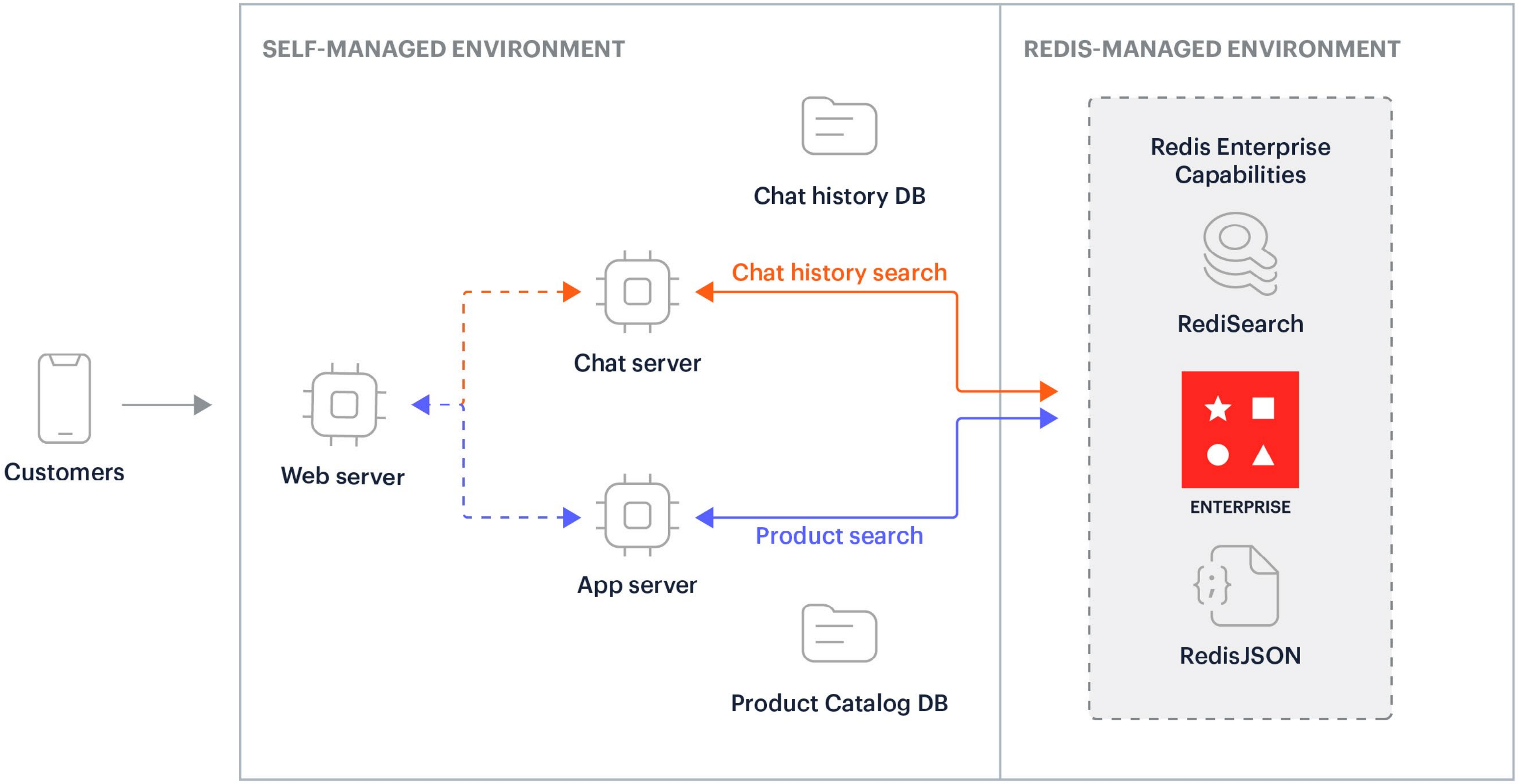
Challenges

- GoMechanic wanted to achieve real-time search and instant master data lookup for over 10 million unique auto parts.
- The company needed the flexibility to query more than primary keys.
- It also wanted to search chat history in real time.

Redis Enterprise Results

- Scaled to tens of millions of keys delivering query and exact match performance capabilities.
- Increased the speed of repetitive master data lookups, eliminating application performance issues.
- Permitted queries on secondary keys to enable retrieval of information such as product description, price, user session data, purchase history, stock price, and location.

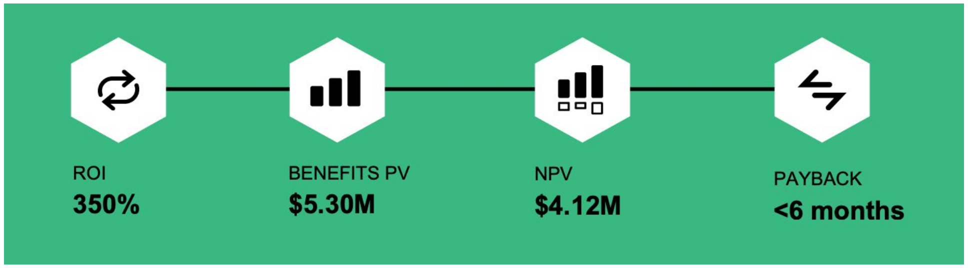
Diagram 3: Boost customer experience with real-time auto-complete and online search of massive tables.



The economic value of in-memory NoSQL databases for real-time search

Recently, Redis commissioned Forrester Consulting to conduct a Total Economic Impact™ (TEI) study and examine the potential return on investment (ROI) that enterprises could achieve with Redis Enterprise. The purpose of this study was to provide critical customer evidence of success along with a framework to evaluate the potential financial impact of Redis Enterprise on their organizations. Read Forrester's Total Economic Impact of Redis Enterprise to discover more about organizations that received $5.3M in business benefits with a 350% ROI in less than 6 months when deploying Redis Enterprise.

Diagram 4: Forrester Total Economic Impact (TEI) of Redis Enterprise Summary of Findings



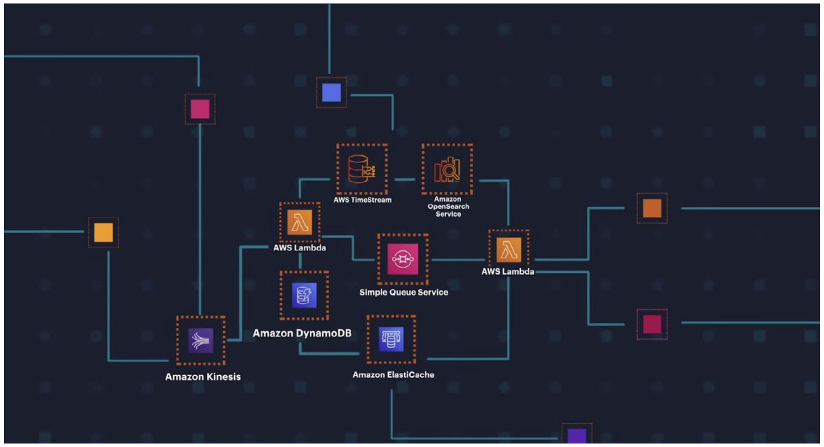
The importance of high-speed multi-model databases for real-time search

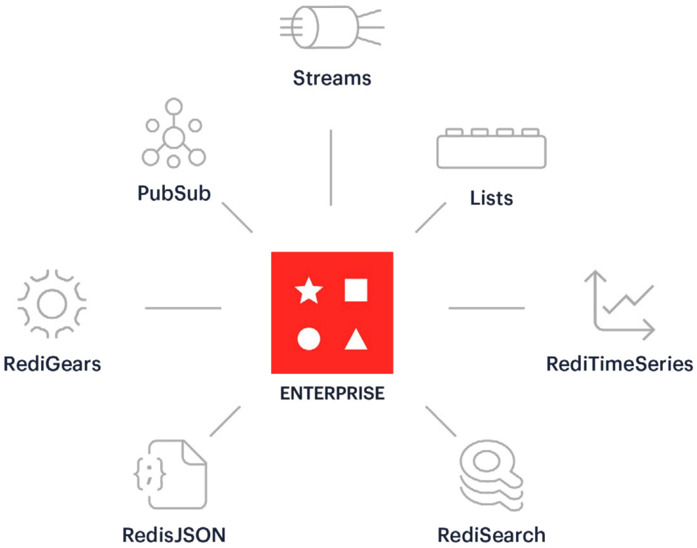
Enterprise-grade, in-memory databases can ingest multiple data models and scale to support millions of simultaneous search and query services for things such as gaming leaderboards, financial services, geospatial mapping, online feature stores, customer lookups, master data lookups, and real-time shopping suggestions. Scaling high-speed search and analytics with traditional SQL or drive-based NoSQL databases is challenging and costly. Often the database overhead will cause severe application latency. Any delay either in-network or application response will cripple business results and drive customers away.

Today, in-memory databases are providing the search and query response time needed to scale where traditional relational database management systems can't. They are most often used as a caching layer to scale NoSQL databases such as MongoDB for JSON documents, online feature stores for machine learning (MLOps), and high-volume message stream applications with Kafka.

But this common best-of-breed, full-stack approach can quickly become overly complex and expensive to operate in order to meet business objectives (image 1).

With a desire to simplify their development environments as well as save time and money, more and more developers are finding additional use cases for Redis. They use Redis for caching, real-time search, and data aggregation as well as a primary database to simplify their entire stack and create a more unified code base (image 2).





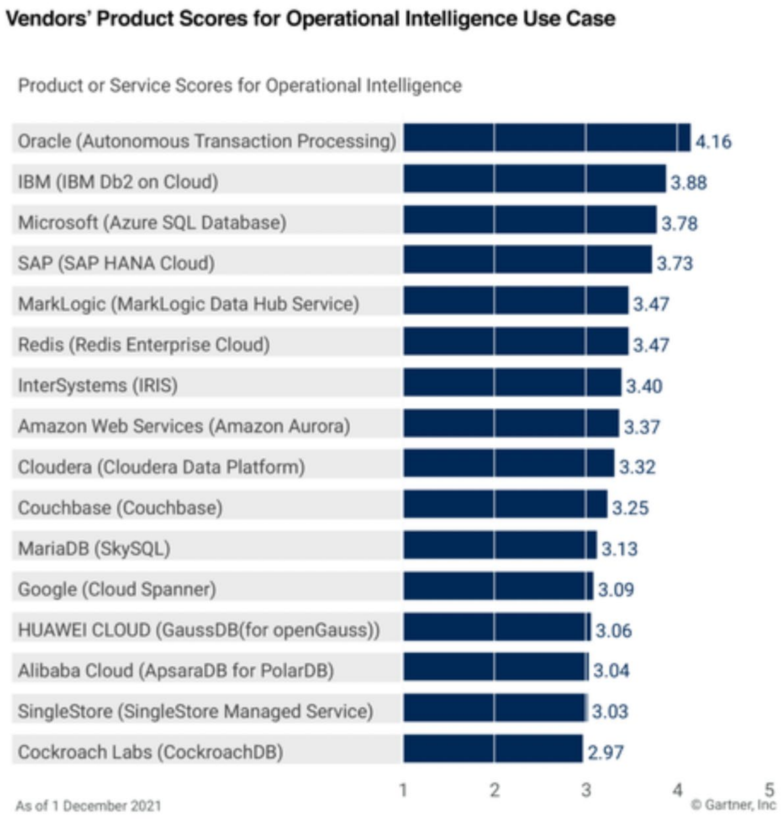
Redis outperforms others for real-time search solutions

Not all versions of cloud or on-premises databases are created equal. Redis is available as open source through the 3-ClauseBSD license, which allows the software to be modified by the developer and redistributed. This is the edition of Redis that most third-party providers deliver as a managed cloud service (e.g., ElastiCache and Memorystore).

The challenge with most third-party provider Redis services is that the open-source version is limited, compared to the Enterprise version. The open-source version can restrict available use cases, making it impossible for businesses to fully use Redis as a global search engine.

Recently, Gartner reviewed leading cloud databases. Redis Enterprise Cloud met the inclusion requirements for all four-cloud database use cases, and also ranked within the top 10 vendors in each use case. Redis is also listed in the top five vendors for the Operational Intelligence use case, with a score of 3.47 out of 5. This validates Redis Enterprise as a real-time search solution among the top vendors in the world.

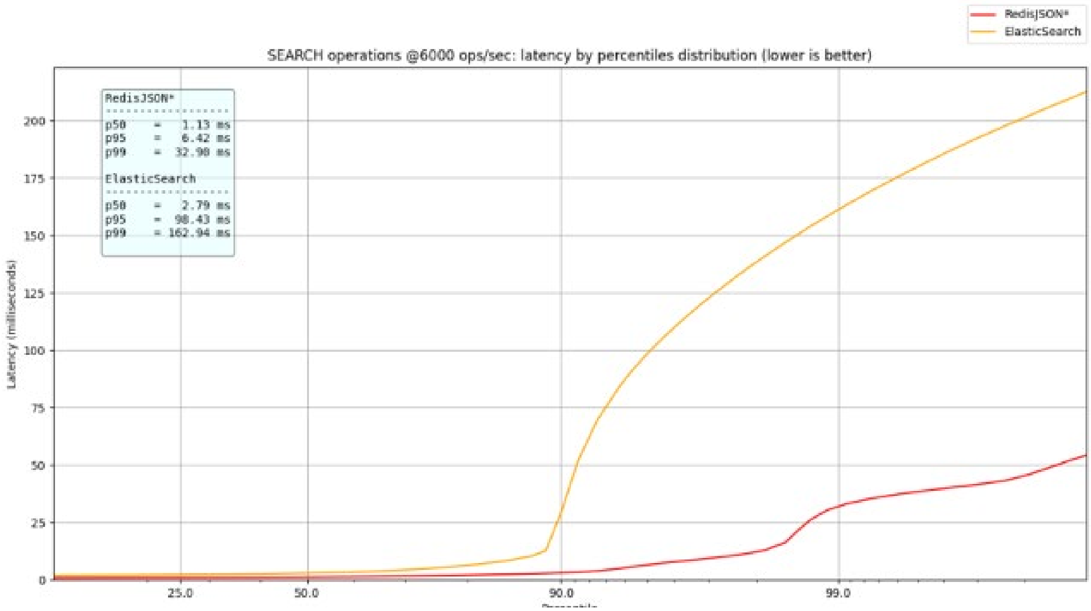
Image 3: Gartner Operational Intelligence Score Card



Source: Gartner (December 2021)

Focusing on search database operations, Redis compared ElasticSearch to Redis Enterprise search with a JSON data model at 6,000 operations per second. They both had single-digit p50 latencies but ElasticSearch was already 247% slower (p50 RedisJSON* of 1.13ms vs. p50 of ElasticSearch of 2.79ms). Latency rises dramatically with ElasticSearch due most likely to garbage collection processes triggering and query cache misses on the higher percentiles, as is clearly visible on the >= p90 percentiles in this graph. Redis searching JSON retained a p99 below 33ms vs. the 5x higher p99 percentile of 163ms for ElasticSearch.

Image 4: Latency Analysis for MongoDB vs. ElasticSearch vs. RedisJSON



The value of Redis: Cache, real-time search, document store, database, and more

As a business leader, the impact Redis has on real-time search and data analytics gives organizations a competitive edge, allowing them to make timely and informed decisions. It can save money, too. The positive impact on application performance Redis delivers can eliminate or greatly reduce costly network, storage, or compute infrastructure upgrade costs.

Only Redis delivers over a million read/write operations per second, with sub-millisecond latency on a modest-sized commodity cloud instance, making it the most resource-efficient database for large volumes of data. It is also highly flexible, supporting many data models, messaging services, and client libraries in all popular programming languages—all of which makes Redis ideally suited for combining high-speed data ingestion and real-time search.

Flexibility is crucial since once a data model is selected, it's very challenging to access a different model on the same database. To solve this problem, users usually develop separate logic running outside the database, either at the application tier or over a serverless infrastructure, acting as a kind of glue between the different data models. Such logic, external to the database, creates significant execution overhead and cannot meet today's requirements for the instant experience. It also adds complexity and makes

Redis Enterprise solves this complex problem by supporting multi-model database operations across and between many modern data models in a fully programmable and distributed manner, all while maintaining instant, sub-millisecond latency.

Redis' Proven ROI: delivers real-time caching, indexing, search, and query database capabilities

Redis Enterprise is a multi-model database that eliminates the need to purchase additional middleware or backend database infrastructure. This reduces costs across the design, build, and run application phases by reducing complexity as well as the number of vendors and software languages needed. For example, Redis Enterprise provides native APIs to ingest, index, query, and run full-text search on JSON documents. It's the unmatched performance and flexibility of Redis Enterprise that makes it a very popular database, documented by many sources and, most importantly, by developers themselves (as shown in Table 2).

Table 2: Redis Popularity

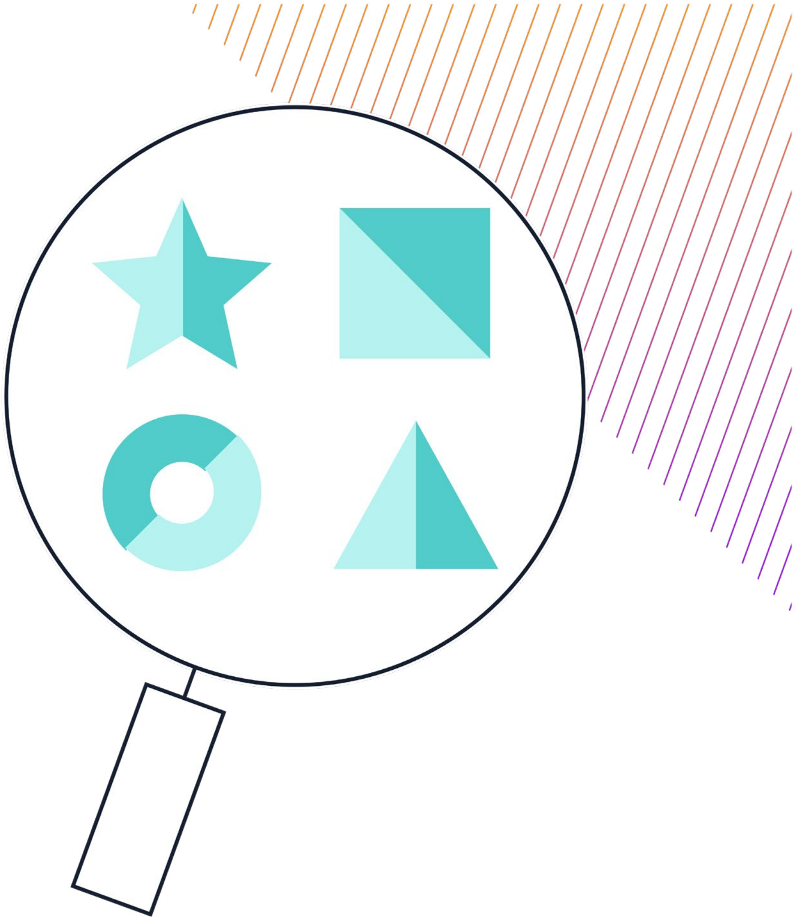
Source	Year	Category
Db-Engines	2021	Top 10 database
Docker Hub	April 2021	7.2M launches/day
Sumologic	November 2020	Most popular database deployed on AWS
DatDocker Hubadog	November 2020	The largest share of container images running in Kubernetes Stateful Sets
Stack Overflow	2017–present	"Most loved" database

Summary

Redis is best known as a caching layer and is cited by developers as the "most loved" database in the world. It has earned its popularity because of its simplicity, versatility, performance, and rapid return on investment. Now innovators are depending on Redis capabilities, such as real-time search, to enable new use cases that extend the value of Redis beyond caching. They are using Redis Enterprise to perform key application functions including real-time search, data enrichment and aggregation, real-time metrics, customer and master data lookups, and as a primary store for JSON documents.

Businesses today operate and manage a perpetual series of real-time events. But what separates the good from the extraordinary is how businesses capture and operationalize that data. Data Operations (DataOps) is a critical practice for organizations to master. Good businesses use data to make informed decisions over time. Great businesses operationalize data to automatically take actions in real time. Redis is a key component to real-time customer experiences, AI, operational analytics, and creating an autonomous business.

Easy to understand and implement, the Redis Enterprise database and its capabilities are fully supported worldwide by Redis. Developers get timely assistance when they need it. And business leaders can limit risk, reduce costs, and simultaneously enhance their customer experience and bottom line.



About Redis

Data is the lifeline of every business, and Redis helps organizations reimagine how fast they can process, analyze, make predictions, and act 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 and now are finding new use cases for the world leader in-memory database.

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 how Redis can give you this edge at redis.com

redis