redislabs
HOME OF REDIS

DATA SHEET

RediSearch

RediSearch: Secondary indexing and full-text search

RediSearch is a search and secondary index engine based on the world's fastest in-memory database, Redis. RediSearch allows you to build rich schemas that can drive real-time document indexing, retrieval and aggregations. RediSearch can achieve real-time indexing (<1ms under most conditions) because it leverages in-memory storage and highly optimized data structures built specifically for low latency and rapid query.

Why RediSearch?

Disk-based search engines are inherently inefficient
Legacy search engines are characterized by high latency and slow writes, a function of mechanical limits and OS-level complexities left over from spinning disk solutions. As a result, they require internal caching solutions to achieve high performance which, in turn, layers on additional infrastructure requirements and complexity.

Legacy search engines need many more resources
Legacy search engines are built around burdensome coordinators and inefficient runtimes that suck away compute resources, complicate operation and yield unpredictable performance, especially under garbage collection conditions. Their bloated feature sets often include functionality that is rarely used and overly complex, leading to steep learning curves and high infrastructure costs.

Benefits of RediSearch

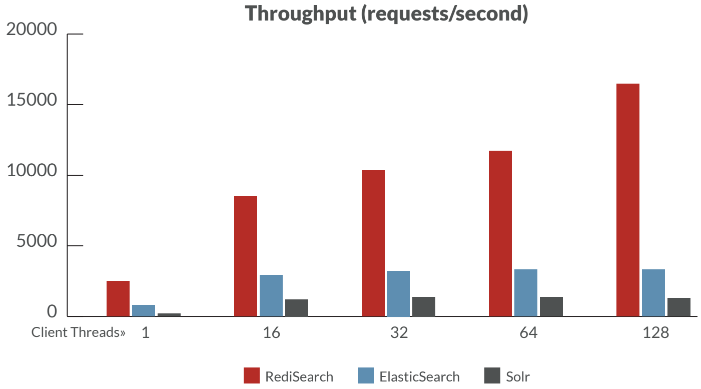
1. **Predictable, high performance with low latency times:** RediSearch delivers continuous results with no performance degradation, while maintaining concurrent loads of both querying and indexing. Redisearch is written directly in C and implemented as a Redis module, so it can write directly to DRAM on a server without any intermediary interpretation or layers. As a Redis module, it implements specialized data types, optimized for index, search and query. This yields extremely high performance for real-time indexing, wherein index items are available for querying within a millisecond.
2. **Reduced infrastructure costs:** Without no need for a cache, RediSearch reduces the overhead needed to run the search engine. Particularly suitable for simultaneous and instantaneous indexing and search scenarios, Redisearch scales to multi-node configurations easily while efficiently indexing and searching billions of documents at high performance.
3. **Operational simplicity:** While RediSearch inherits the performance and operational simplicity of Redis, it doesn't need to deal with years of incremental bloat, thanks to its clean-sheet reimagining of search and secondary indexing.

How it works

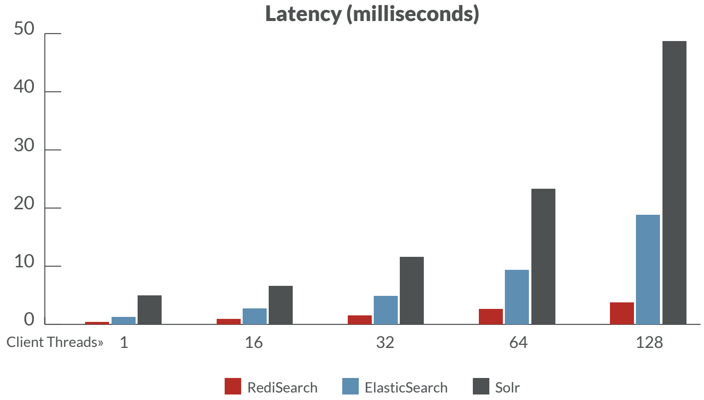
RediSearch starts with a schema that outlines how the data will be incrementally indexed as documents are added. Any field in the index can be one of four broad types:

5x Better Throughput and Latency than Elasticsearch

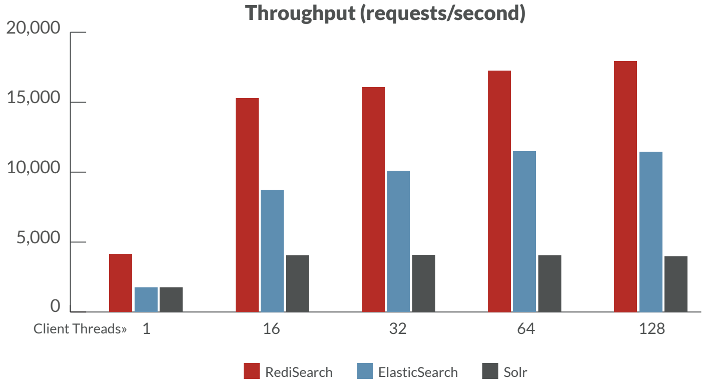
**Simple Two Word Query"barack obama"
Throughput (requests/second)**



**Simple Two Word Query "barack obama"
Latency (milliseconds)**



**Autocomplete -1100 Top 2-3 Letter Prefixes
Throughput (requests/second)**



1. **Text** for character-based data—RediSearch is multilingual and has built-in support for 18 languages (including Chinese)
2. **Numeric** for data that is inherently countable
3. **Geospatial** for data based on real-world coordinates
4. **Tags** for meta-data that labels a particular document

Building a schema can be optimized and fine-tuned by numerous options or it can be as simple as a few characters.

After a schema is established, you can add documents to RediSearch. When documents are added, they are indexed in real time, meaning they are available for query within 1 millisecond. Beyond the fields established in the schema, each document has a unique identifier and a weight that aids in the query process.

To query documents in RediSearch you can provide simple key words or use the built-in query language that allows for very rich refinement and filtering. The query language can natively add clauses based on any of the fields and types, and use logical as well as parenthetical combinations to narrow down the results.

RediSearch also has the ability to aggregate the values stored in an index. The aggregation is based on a pipeline where values are filtered, grouped and reduced, transformed, sorted and limited.

What are the use cases

RediSearch is ideal for search and query of frequently updated datasets. Examples include fraud prevention, predictive alerting, dynamic catalogs, secondary indices of large databases and many more.

No need for batch indexing or service interrupts—RediSearch's continuous indexing with no performance degradation makes it ideal for querying high-volume updates, such as when millions of items need to be searched instantaneously.

beathunter

> *"I would say that RediSearch saved my project. It made it feasible to create a searchable index at a very low cost. I was struggling to decide if I should use a low-cost relational database or a more expensive index engine like Solr or ElasticSearch. Then I read about RediSearch and decided to give it a try. It fit my needs perfectly and it was super easy to set up in production."*

Victor Ruiz

BEATHUNTER.NET

Key features of RediSearch

- Full-text indexing of multiple fields in document, including:
  - Exact phrase matching
  - Stemming in many languages
  - Chinese tokenization support
  - Prefix queries
  - Optional, negative and union queries
- Distributed search on billions of documents
- Numeric property indexing
- Geographical indexing and radius filters
- Incremental indexing without performance loss
- A structured query language for advanced queries, including:
  - Unions and intersections
  - Optional and negative queries
  - Tag filtering
  - Prefix matching
- Integrated aggregations engine that allows for building of pipelinable operations, consisting of:
  - Groups
  - Reductions
  - Sorts
  - Transformations
  - Limits
- A powerful auto-complete engine with fuzzy matching
- Multiple scoring models and sorting by values
- Concurrent, low-latency insertion and updates of documents
- Concurrent searches allowing long-running queries without blocking Redis
- An extension mechanism allowing custom scoring models and query extension
- Support for indexing existing Hash objects in Redis databases
- Support clustering* through a coordinator entity

*Note: clustering is only available in RediSearch's Enterprise version

Get Started with RediSearch Today!

Visit https://redislabs.com/redis-enterprise/software/downloads/.
Talk to a Redis® expert today. Contact expert@redislabs.com.