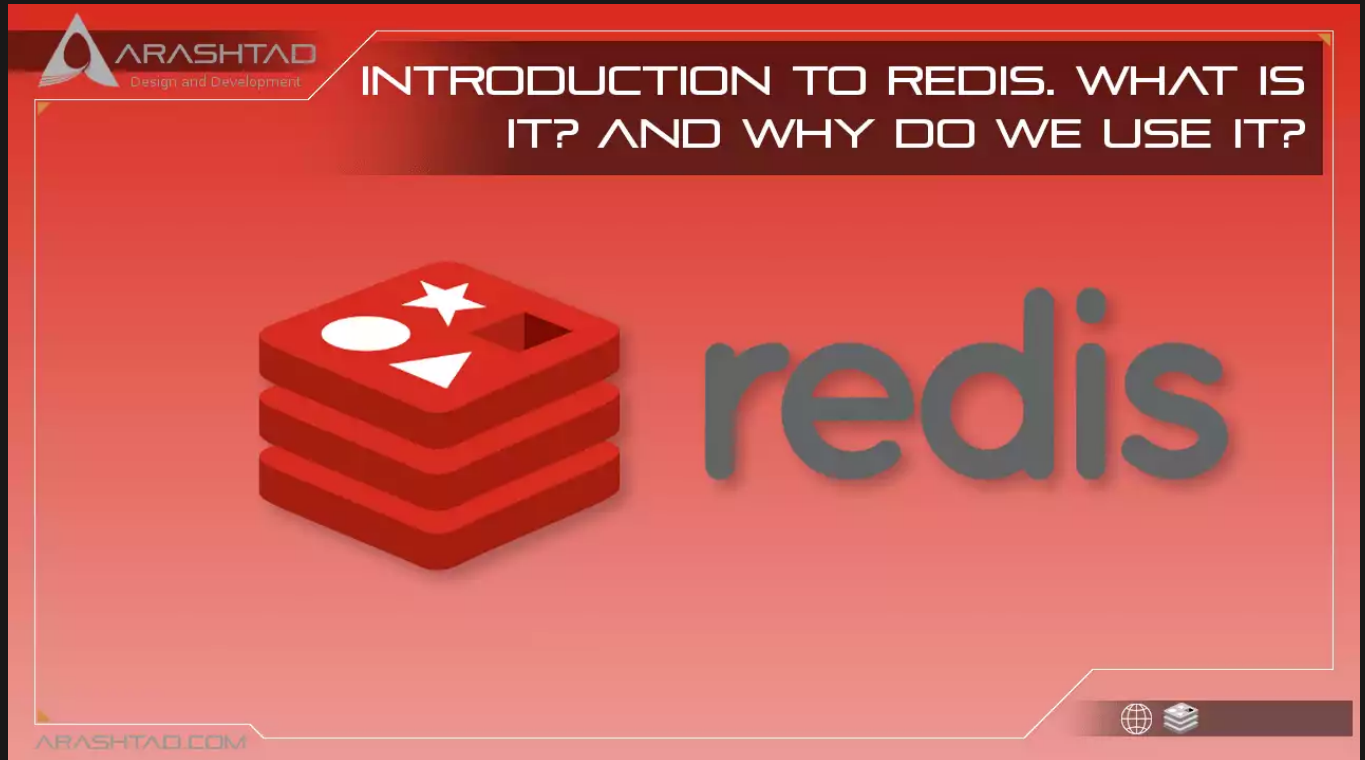


Introduction to Redis. What is it? And Why Do we Use it?

No comments



A NoSQL database known as Redis (which stands for Remote Dictionary Serve) is an in-memory key-value data store, cache, message broker, and queue that runs quickly. This project was initiated by Salvatore Antirez Sanfilippo, the original developer. He developed a real-time web log analyzer and wanted to increase the scalability of his startup in Italy. In response to significant difficulties scaling some types of workloads using traditional database systems, he began prototyping the first proof of concept version of Redis. Github and Instagram were among the first companies to adopt this technology. Therefore, it's no surprise that the technology has been adopted and patronized not just by large companies, but also by developers.

What is Redis?

The official Redis document describes it as an in-memory data structure store (BSD licensed) that serves as a database, cache, and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, and geospatial indexes with radius queries and streams. In short, Redis is a tool in the In-Memory

Databases category of a tech stack. There are no external dependencies for Redis since it was written in ANSI C. It works on most of the POSIX systems, including BSD, Linux, and OS X. OS X and Linux are considered to be the two operating systems for which Redis has been developed and tested the most. However, Linux has been used to deploy it. Despite not being officially supported by Microsoft for the Windows builds, it does run on Solaris-derived systems such as SmartOS.

Why Do we Use Redis?

This software allows for caching. Caching is the process of storing data in temporary storage components which are used in the future to serve data faster by storing temporary data in temporary storage components. When this happens, Redis is used. It is compatible with most major programming languages and protocols, such as Python, Java, PHP, Perl, Go, Ruby, C/C#/C++, JavaScript, and Node.js.

Use Cases of Redis:

It is common for many people to use Redis for caching data and for session storage (i.e. web sessions). Redis has a lot of uses that can be applied and is useful for any situation, especially when it comes to speed and scalability, it is easy to scale up and down. Redis is commonly used for session caching, page caching, message queue applications, leaderboards, and many other tasks by experienced users or companies that have been using it for a long time.

Redis is also popular among large cloud companies offering fully-managed databases or DBaaS. Among the companies using Redis are Twitter, Amazon Web Services, Microsoft, Alibaba, and Microsoft Azure Cache for Redis in Azure, and Amazon Web Services is offering ElastiCache for Redis. Use cases of Redis to describe the types of scenarios you can apply, implement, and integrate with your business applications and environment. As the Redis delivers super-fast performance, it is frequently used for real-time applications. We'll examine each of these where it is ideally applicable.

Caching

Basically, caching is the process of storing data in a temporary storage location to allow it to be accessed faster in the future. The cache is a temporary storage location for data. In order to reduce latency access with disks or SSDs, increase throughput, and ease database and application load, Redis is a good choice for implementing an in-memory cache. Caching solutions with Redis include web page caching, database query results caching, persistent session caching, and caching of frequently used objects like images, files, and metadata.

Session Store

As a user interacts with applications such as a website or a game, the session state captures the current state of user interaction with the application. As long as a user is logged in to the system, a typical web application keeps a session for each connected user. An application uses a session state to remember user details, login credentials, personalization information, recent actions, shopping cart information, and more.

Each user interaction must be handled without disrupting the user experience by reading and writing session data.

therefore, no round trips to the main database should be required during the live session. when a user is disconnected from the application, the session state life cycle ends. some data will remain in the database after the session ends, but transient information can be discarded after the session ends.

Messaging Applications

As a result of its support for Pub/Sub, pattern matching, and a range of data structures including lists, sorted sets, and hashes. Redis has also been able to support high-performance chat rooms, real-time comment streams, and social media feeds.

Game Applications

Real-time leaderboards and scoreboards are often built using Redis by game developers. To implement this use case, a Redis Sorted Set data structure can be used, which provides uniqueness of elements while keeping the list sorted by the users' scores (points) associated with the key. Update the user's score whenever it changes. Sorted Sets can also be used to handle time-series data by using timestamps for scoring. A popular technology for implementing session stores, Redis has a very high throughput, so it is widely used for implementing session stores. As well as the use cases mentioned above, Redis is used in many other ways in many different ways in machine learning, real-time analysis, media streaming, etc.

Redis Pros and Cons:

Pros

1. It's Faster than nearly all caching we know.
2. It is Simple user friendly due to its easy setup.
3. It supports almost all data structures.
4. It allows storing key and value pairs as large as 512 MB.
5. It has its own hashing mechanism called Redis Hashing.
6. Zero downtime or performance impact while scaling up or down.
7. It is open source and stable.

Cons

1. Due to the fact that the data is sharded based on the hash-slots assigned to each Master, the data to be written into those slots will not be preserved if the Master holding those slots is unavailable.
2. Clients that connect to the Redis cluster need to know the cluster topology, causing extra configuration work.
3. Failover does not happen unless the master has at least one slave.
4. It requires a huge RAM because it is in memory so are not supposed to use it on RAM servers.

How Does Redis Store Data?

There are 5 data types for storage in Redis:

1. String: a text value
2. Hash: A hash table of string keys and values
3. List: A list of string values
4. Set: A non-repeating list of string values
5. Sorted Set: A non-repeating list of string values ordered by a score value

Data type operations are supported in Redis, so you do not have to load an entire data object at an application level, modify it and then restore the modified object. The Redis memory management mechanism is encapsulated, resulting in a much simpler approach than Memcached's Slab mechanism. Redis supports keys and values of up to 512MB in size. this limit applies to aggregate data types (Lists and Sets).

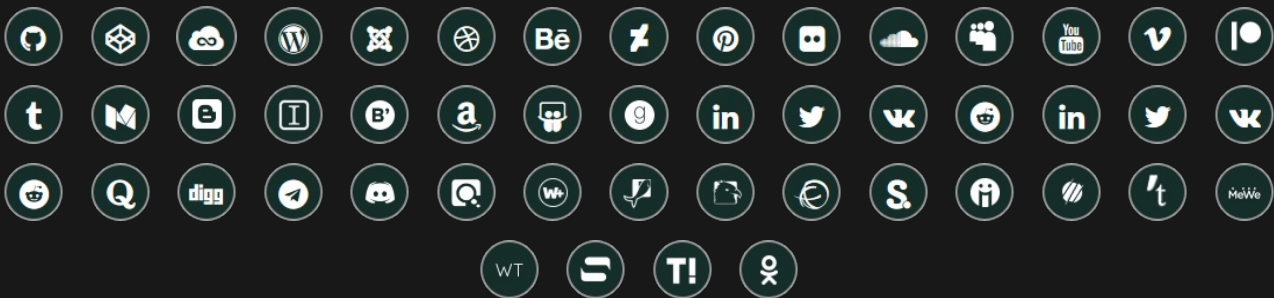
Conclusion

In this article, you learned about Redis, What it is, why we use it and how we do it. Moreover, you learned about the pros and cons of using Redis in addition to its use cases. In summary, Redis (Remote Dictionary Server) is an in-memory data structure store that is used as a distributed, in-memory key-value database, cache, and message broker, with optional durability. In addition to strings, lists, maps, sets, sorted sets, HyperLogLogs, bitmaps, streams, and spatial indexes, Redis supports other types of abstract data structures as well.

Join Arashtad Community

Follow Arashtad on Social Media

We provide variety of content, products, services, tools, tutorials, etc. Each social profile according to its features and purpose can cover only one or few parts of our updates. We can not upload our videos on SoundCloud or provide our eBooks on Youtube. So, for not missing any high quality original content that we provide on various social networks, make sure you follow us on as many social networks as you're active in. You can find out Arashtad's profiles on different social media services.



Get Even Closer!

Did you know that only one universal Arashtad account makes you able to log into all Arashtad network at once? Creating an Arashtad account is free. Why not to try it? Also, we have regular updates on our newsletter and feed entries. Use all these beneficial free features to get more involved with the community and enjoy the many products, services, tools, tutorials, etc. that we provide frequently.

[SIGN UP](#)[NEWSLETTER](#)[RSS FEED](#)