* **What is Redis?**

*Redis, which stands for Remote Dictionary Server, is a fast, open-source, in-memory key-value data store for use as a database, cache, message broker, and queue.*

*----------------------------------------------------------------------*

* **Why do we need Redis in our microservice?**

In our microservice our goal is to check for the presence of duplicate trades among millions of trades coming to the database, so we need a checkpoint where trades are analysed whether they are already existing in the database!

But we don’t want to keep all trades at that checkpoint for a long time, all we need is a differential amount of time to check for their prior presence in the database.

For that purpose we need a faster memory where trades will be parallely consumed as well as processed before we insert them in the database.And this necessity brings the Redis into the picture.We will be storing the trades in redis cache for a short period of time during processing it.

-----------------------------------------------------------------------------

* **More insights on why do we choose Redis over memcached and Ehcache**

❖DB-Engines-Rankings specified properties:

# One of the most searched and system-mentioned cache management system.DB-Engine mines data from search engines like Google, Bing and Yandex(Yandex is a russian search engine).

# Most profiles on professional networks mention this as a better in-memory data structure store which can be used as cache as well as database.

❖Most variety of data structures supported:

# Strings

# Sets

# Sorted Sets

#Hashes

#Lists…….all of them store value in the form of a contiguous sequence of chars.

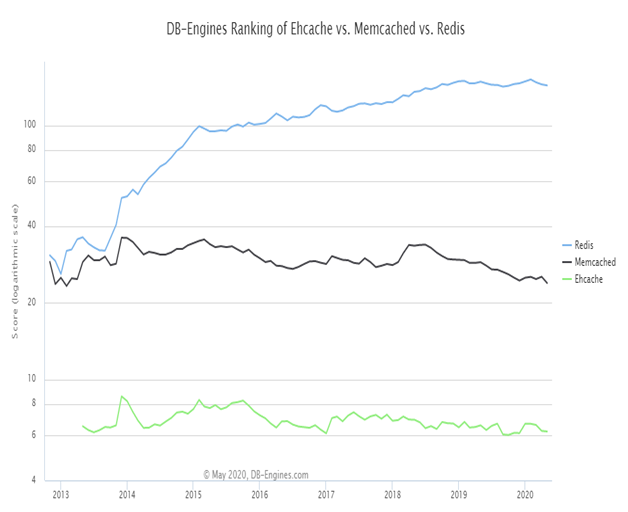
❖One of the fastest cache tools.

-----------------------------------------------------------------------------

* **A systematic comparison for more clarification:-**

|  |  |  |
| --- | --- | --- |
| **Category** | **Redis** | **Other Cache Mechanisms** |
| **Memory** | Used as database,cache and message broker. | Cache with only tiered data storage |
| **Implementation** | Implemented in C therefore the speed is better. | Developed in java,therefore relatively slow. |
| **Engines** | DB-Engines Ranking:  Score/Rank-143.48/8  source:[https://db-engines.com/en/system/Ehcache%3BMemcached%3BRedis](https://db-engines.com/en/system/Ehcache;Memcached;Redis) | DB-Engines Ranking:  Score/Rank-6.23/51 |
| **Supported Languages** | Supporting programming Languages:Almost Every high level programming language. | Supporting Programming Languages:  Only Java. |

* **Comparison Graph**

**----------------------------------------------------------------------------------------------------------**

* **Limitations and Solution**

●Redis is a NoSQL Database so there is no notion of ID and you have to rely onKEY for data storage.But so is the case of Cassandra and MongoDb and still we can use the KEY as ID or program the database to configure our own custom ID.

●Redis is an in-memory database i.e. it stores data in RAM while doing query and updates in database, So there is a risk of data loss if server crashes or system turns off accidently.

But Redis provides us with two ways to configure it to achieve data

Persistency:-

1.Dumping in-memory data to disk in compact format for each transaction (in form of log files).

2.Writing/Appending a file with every command which alters the data on redis.

**-----------------------------------------------------------------------------------------------------------**

* **Installing redis in Ubuntu 18.04 :**

|  |
| --- |
| **$ sudo apt-get update $ sudo apt-get install build-essential tcl $ mkdir redis $ cd redis $ wget http://download.redis.io/releases/redis-6.0.3.tar.gz $ tar xzf redis-6.0.3.tar.gz $ cd redis-6.0.3 $ make $ make test $ src/redis-server # after this, redis server will get activated on port 6379 # now open other terminal(don't close previous terminal as it has server running) #follow following commands in new terminal $ cd redis $ cd redis-6.0.3 $ src/redis-cli #this will open redis client...Now you can communicate between server and client.** |