

G



D



D

Redis

What is Redis?
How is it used?

How does it help?

open source, inmemory

User


server

Postgres

* a lot of data gets to us from different tables and when website is refreshed, our server computes all data from the DB

→ this increases waiting time for the user and reads into the DB:

→ increases response time slowing down the application

To solve such problems we have services like Redis

What is Redis?

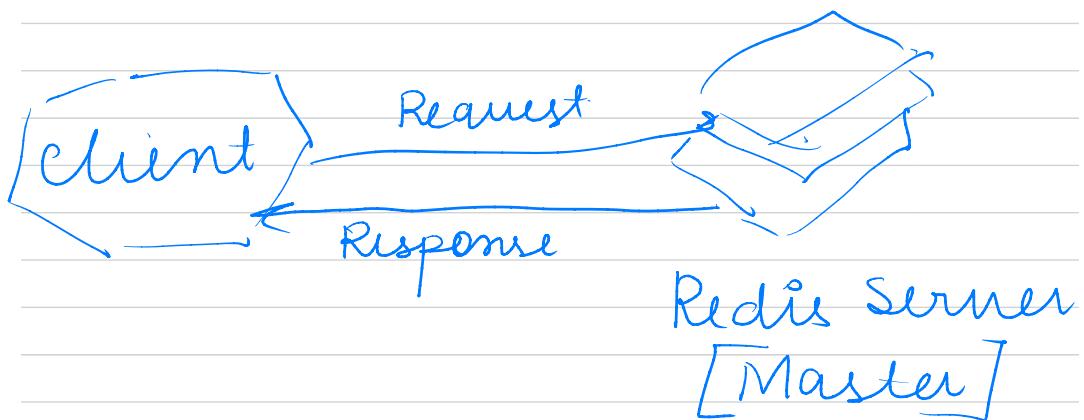
- An in-memory data structure store that can be used as data base, cache, and message broker
- Redis supports various data structures such as strings, hashes, lists, sets, sorted sets, bitmaps, hyperloglogs and geospatial indexes.

Features of Redis:

- In memory storage
- Persistence → option to dump data
- Data structures [support string, Hashed, list, Bitmap]
- Replication
- Pub/Sub messaging

fast read
and write
ideal for
real time requirements

Redis Architecture



Running Redis using terminal on docker

- Install docker on your device
- Search Redis Stack on docker
- copy and paste the command given in

redis/redis-stack

to start Redis stack server:

redis-stack-server [command for terminal]

to launch Redis UI

redisinsight

Now we have Redis
we can access it using 2
ways

Either on local host:8001

or using terminal



Step 1: docker ps

Step 2: copy ID of mediastack

Step 3: command:

docker exec -it ID bash

Step 4: to check whether we are in to
redis

type medi-cli ping # this
returns PONG

Step 4: to go into cli type:

medi-cli [then press enter]

Values can be set and
changed using GUI as well

Redis Cli DATA TYPES

String: At these are cli commands

→ Set Key value

Ex: Set name dakesh

This returns
OK if every
thing
works well

→ get name

Returns name

Ex: "Dakesh"

Here we can declare keys
and they get automatically
grouped, to declare them we
use:

Ex:

set name:1 dat
set name:2 ABC
set name:3 XYZ

} they will be stored in a name folder:

- if we add nx at the end of a set command it sets a value only if another key of the same name doesn't exist.
- If we want to get multiple values we use mget: key key key
- We can also use mset to set multiple values at the same command
→ mset Key:1 "Daksh" Key:2 "Laksh"

incr → to increment

incrby → to increment using some value

REDIS with Nodejs

- create folder
 - npm init
 - install package ioredis
- make a file named client.js
- ↳ con require ioredis

We can expire a set of data, set max limiting in Redis

Redis Lists:

We can use queue and stack in list:

commands:

lpush → push from left side

rpush → push from right side

Redis list

LEFT

RIGHT

- * if we insert from left and remove from right it is a queue
- * if we insert from left and remove from left we get stack.

lpop } remove from left and
rpop } right respectively

llen → length of list

Blpop [group name / list name] timeout

This returns value
from head of the list
and waits till the timeout

for returning message

ltrim

range [list name] index range

an:

range messages 0 - 1 → [gets from 0 to end of list]

SetS

doesn't allow duplicate items

command

sadd

semmone

sissemleer

sinter [intersection b/w 2 sets]

[if operation is performed successfully it returns
else return zero]

hashes:

- used to store key value pair

Ex:

```
const fieldsAdded = await client.hset(
```

'like:1',

{

model: " ",

brand: "xyz",

type: "xyz",

price: "xyz",

)

},

```
console.log('number of fields were added', field  
added);
```

Sorted order &
Priority!]

zadd add priority to dataset

$\text{zrange score } 0\text{-}1$

↳ sorts score:

$\text{zrevrange score } 0\text{-}1$

↳ sorts score in
reverse order

We get rank/position using rzang

Redis Streams

⇒ Event sourcing

⇒ Sensor monitoring

⇒ notification

Commands:

- xADD → adds new entry to stream

- xREAD → reads one or more entries
- xRange → returns a range of entries

b/w two supplied entry ID's

- ~~xlm~~ ~~length~~ length of stream

Redis geospatial Data: !!

lets you store coordinates and search for them

Commands:

GEOADD: adds a location to a given geospatial index

GEOSEARCH: return locations with a given radius

Redis Bit Map:

Bitmaps are not an actual data type but a set of bit oriented operations defined on the string type

Commands:

Pub-Sub

We can publish / subscribe / multichannel subscribe

