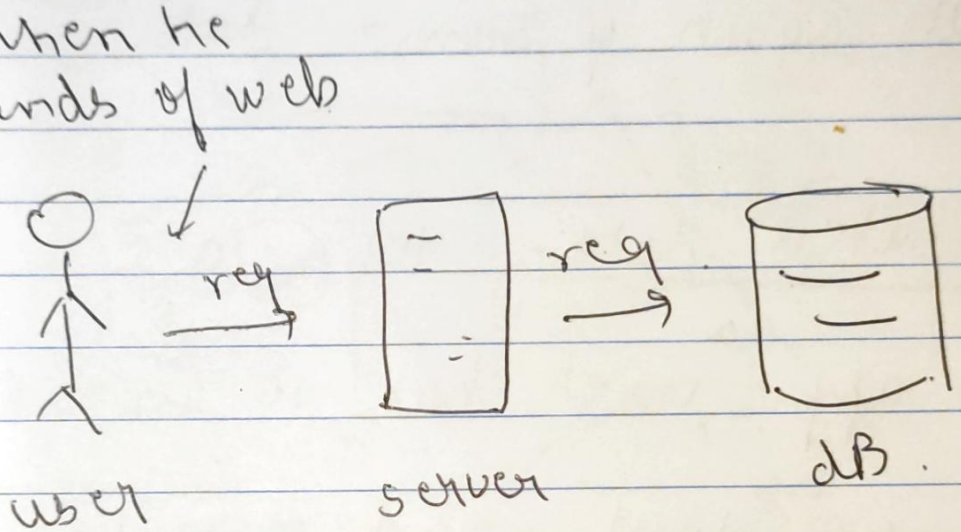
Redis.

What problem is solves?



When user lands on website server makes a query to dB let's say 6 commands are run. what if he again presses refresh, and again 6 commands are run, increasing bill amount of dB and uncessary request are made again of for same data.

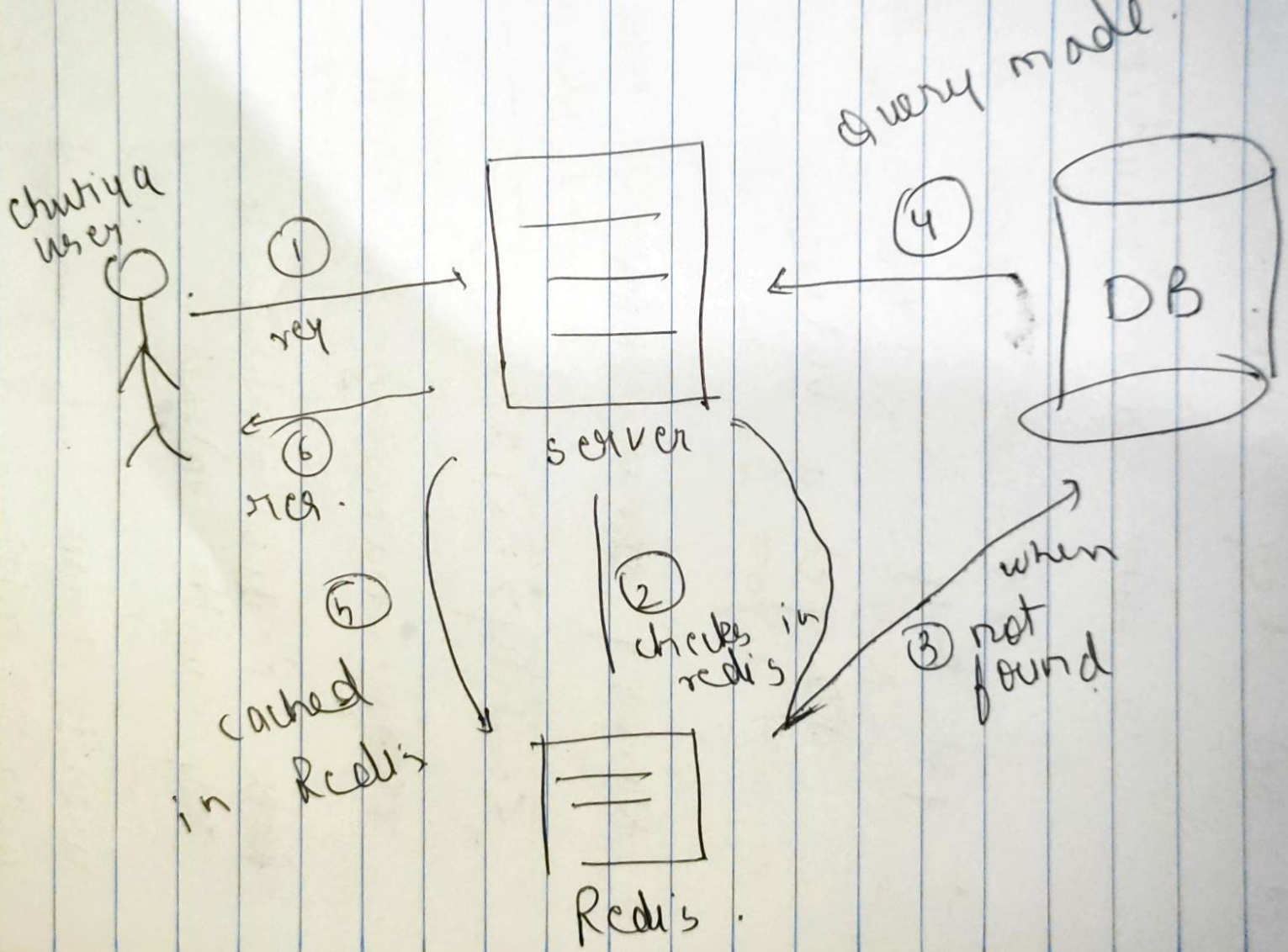
In memory data store. It caches the computed data (Ram) so that next time we don't need to run Query on dB again of same data.

Computed data:- let's a in a chat app, user has 10 unread msg, we dont need to cache whole dB, we only need 10, so that's why, its computed data.

Redis by default port :-

6397.

Redis stack:- to visualize Redis server.


Query made.
chutiya user.
1
req
4
DB
server
6
res.
5
cached in Redis
2
checks in redis
when
3 not found
Redis.

→ set Key value

To ~~cache~~ cache anything.

→ get Key

TO get a value.

→ Naming convention

<entity>:<id>

eg → name:1.

~~Redis~~ Redis groups it Key with values.

user:1 hello } grouped.
user:2 hello

→ about [nx].

nx is used to eradicate data duplicacy.

Ex:-

→ set msg:1 hello nx

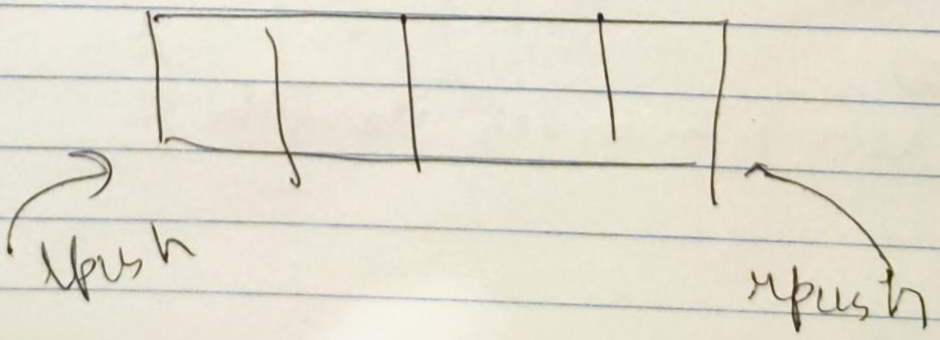
this will set (since no msg:1 is prest)

if we again do set msg:1 hello nx

→ ~~xxxxx~~ Return nil, since msg:1 is already present.

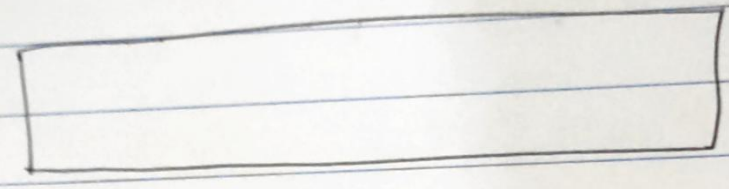
List

→ lpush → pushing messages.
→ rpush.



Caps Lock

stack using list



if we ~~list~~ push and Remove from only left side it becomes a stack.

```
Upush <> ...
Upop <> ...
```

Queue

if we push from a side and remove from right side its Queue.

Rpop and Upop (same).

(llen) - gives length

list pushes from left side and jumps the other side

.push("hello")

.push("manes")