Redis v0.3 Q&A

Q&A

0, questions from v0.1

a, what is data point?

So this was actually an imprecise measurement I came up. Data point number = number of keys + number of values + number of any other data object. This is an inaccurate measurement to predict memory size since Redis’ memory size is only directly proportional to key numbers. Twice key numbers means twice memory cost but twice value numbers does not mean twice memory cost. It is due to certain way Reids manage its memory space and explicit implementation is not recorded in documentation.

b,” if I get the result of SORT function, How can I know about the previous

sequence of the items? Because it will be used for the sorting other lists.”

Sort in schema v0.1 was a horrible design, so it is updated to v0.3 to resolve all those issues.

1. What command should I use for Deleting nth item in the List

It will be 2-step operation, including removing key from list and remove the entry from hashes. Let’s use the same example from the PPT and assume we want to remove student with ID 2. Operations would be:

LREM ID 1 2

DEL student \_2

2. If you use a hash set to store data, is there no overhead? For example, the hash set consists of a field using the strings data structure. Therefore, when using hash, contents corresponding to field are stored continuously. Therefore, it seems that memory is used much more than other data structures.

Redis hases are actually implemented with certain encoding method to compress its size in memory. In Reids, the memory cost is more directly correlated with key numbers instead of entry/field numbers. If the key number is doubled, the memory cost is also doubled, but if entries are doubled, the memory cost may not be increased as much as doubled. Hash method is the method with the least keys created in Redis. Hash encoding method is not explicitly explained in documents, but it can be taken as conclusion that hash, at least, has the same memory efficiency as List and Set. Referring to:

<https://redis.io/topics/memory-optimization#use-hashes-when-possible>

<https://redis.io/topics/memory-optimization#using-hashes-to-abstract-a-very-memory-efficient-plain-key-value-store-on-top-of-redis>

3. Are you preparing to answer the two questions Dr. Cho asked?

I have not received any question from Dr.Cho, but I am prepared to answer Dr.cho’s question.

4, RDBMS provides the ability to back up each table. Does Redis provide the ability to back up datasets that correspond to the desired key independently? If you don’t have a function, how can you manage it similarly?

I need more information on definition of “backup”. Is it local back up or back up on another server? Redis has internal local backup to hard driver similar to RDBMS. As for backuping to another server, I do not think there is an internal function for it, but it can be explored. One way I can think of is using stream and send backup data to “consumers”, in this way, you can also define which data you want to back up to which server.

5,  I want to maintain the high availability of the database by constructing one master node and three slave nodes using the Redis Sentinel. Is it possible to test? (Because the Redis database I was managing was down for an unknown reason, and all the information was deleted. So I’m concerned about this and I want to find a way to against it.)

//working

6. What was your way to check Insertion?

Insertion should be checked with the return value. E.g: Inserting 100 entries will return 100 and any other uumber would be a fail. I did not write it in code since I was benchmarking the I/O time.