Lab IST769 Unit H - Key Value

## Agenda

1. Your Questions
2. Go over Problem Set

## 1. Your Questions

Ask any questions you have here!

* Can the question-1,2,3 done using Pubsub
* Problems with configuring Redis Retwis

## 2. Go Over Problem Set H

Snapchat clone! Let’s use Redis to create a data model like Snapchat. Basically, users send messages to each other and once the message is accessed by the receiver it expires in 60 seconds. The rules:

A. Each **message** should be keyed by an id (you can use an integer and control the ID yourself)

B. Each message key should be namespaced, like so: **snap:msg:1** where **1** is the ID in this case.

C. Each **message** has 3 hash fields:

a. **To**: username of the recipient e.g bob

b. **From**: username of the sender e.g mary

c. **Text**: the message itself.

D. When a user ***sends a*** ***message,*** perform these Redis commands:

a. A new key is added to namespace **snap:msg:*id*** with the fields set in the hash..

b. Add the ***id*** of the message to the user’s inbox key, queue, which is a list.

For example, mary’s inbox key is **snap:inbox:mary**

E. When a ***user reads a message,*** we:

a. Remove it from the end of their inbox key list, a FIFO queue

b. Set the message id key to expire in 60 seconds.

1. Using the Redis CLI, send these messages in the order they are listed with Redis commands. Make sure to perform both steps D.a and D.b as separate commands.

**To From Text**

Bob Art You owe me $50

Che Bob Hello there!!!

Che Dax Is this thing on?

Dax Art When is the meet-up?

Che Art What is Bob doing. OMG.

Bob Dax Who?!?!?

Answer:

hset snap:msg:1 to "Bob" from "Art" text "You owe me $50"  
lpush snap:inbox:Bob 1  
  
hset snap:msg:2 to "Che" from "Bob" text "Hello there!!!"  
lpush snap:inbox:Che 2  
  
hset snap:msg:3 to "Che" from "Dax" text "Is this thing on?"  
lpush snap:inbox:Che 3  
  
hset snap:msg:4 to "Dax" from "Art" text "When is the meet-up?"  
lpush snap:inbox:Dax 4  
  
hset snap:msg:5 to "Che" from "Art" text "What is Bob doing. OMG."  
lpush snap:inbox:Che 5

hset snap:msg:6 to "Bob" from "Dax" text "Who?!?!?"  
lpush snap:inbox:Bob 6

2. Using the Redis CLI, read messages for the following users, in the order listed. Make sure to perform both steps E.a and E.b.

Bob

Che

Art

Bob

rpop snap:inbox:Bob  
expire snap:msg:1 60  
  
rpop snap:inbox:Che  
expire snap:msg:2 60

rpop snap:inbox:Art

rpop snap:inbox:Bob

expire snap:msg:6 60

3. Provide a current state of the Redis database after Questions 1 and 2.

Display the current keys under the **snap:** namespace.

Display the messages which have not been read (and therefore have not expired)

Display the message ID’s in each users’ inbox.

Display the current keys under the snap: namespace.

keys snap:\*

Display the messages which have not been read (and therefore have not expired)

keys snap:msg:\*

hgetall snap:msg:4

hgetall snap:msg:5

hgetall snap:msg:3

Display the message ID’s in each users’ inbox.

lrange snap:inbox:\*

lrange snap:inbox:\* 0 -1

The Department of Motor Vehicles has hired you to build a queue management system. You have decided the best system for this is Redis (a good choice, BTW). The system needs to manage a single queue of users, by username. Queued users can be served at one of 4 windows, A,B,C or D. The structure you build in Redis should support the queue and be able to display who is waiting in the queue. As people go to the window they should be removed from the queue and assigned to one of the 4 windows. You should be able to display who is at each window at any time.

Namespace all keys with **dmv:**

**Example:**

Users In queue: Tom, Bill, Bart

Being Served at windows: A: Carl, B: Steve, C: Chuck, D: Dave

Event: When Dave is done at the window D, Bart is served next:

Users In queue: Tom, Bill

Being Served at windows: A: Carl, B: Steve, C: Chuck, D: Bart

Event: Mary arrives

Users In queue: Mary, Tom, Bill

Being Served at windows: A: Carl, B: Steve, C: Chuck, D: Bart

4. Its first thing in the morning and 8 people are waiting outside for the department to open! Add them to your queue: **amy, beth, chris, dee, erin, fran, greg, hela**

Provide all the command required to accomplish this and a view of the queue

lpush dmv:queue amy beth chris dee erin fran greg hela

lrange dmv:queue 0 -1

5. The department is now open! Assign the first 4 people to windows A,B,C and D respectively. Oh, and Don’t forget to remove them from the Queue!

Provide all the steps required to accomplish these steps and a view of the queue and windows.

# adding names to queue

lpush dmv:queue amy beth chris dee erin fran greg hela

lrange dmv:queue 0 -1

# Adding list of windows to queue

lpush dmv:window A B C D

lrange dmv:window 0 -1

# Poping to get open window and the person on the queue

# assigning the window to the person, and displaying the assignment

rpop dmv:window

rpop dmv:queue

hset dmv:display A amy

hgetall dmv:display

# repeat

rpop dmv:window

rpop dmv:queue

hset dmv:display B beth

hgetall dmv:display

# repeat

rpop dmv:window

rpop dmv:queue

hset dmv:display C chris

hgetall dmv:display

# repeat

rpop dmv:window

rpop dmv:queue

hset dmv:display D dee

hgetall dmv:display

6. Next, the following events occur:

a. iris arrives

b. window C becomes available – move the next person from the queue to this window!

c. window B becomes available – move the next person from the queue to this window!

d. jake arrives

e. window C becomes available – move the next person from the queue to this window!

Provide all the steps required to accomplish these steps and a view of the queue and windows after the events.

**Answer:**

# a) add iris to queue

lpush dmv:queue iris

lrange dmv:queue 0 -1

# b) window C becomes available – move the next person from the queue to this window!  
lpush dmv:window C --- Add C to empty window queue

hdel dmv:display C --- Update assignment

hgetall dmv:display --- Show Updated display

rpop dmv:window --- Get next open window

rpop dmv:queue --- Get next person on queue

hset dmv:display C erin --- Assign the person to the Window

hgetall dmv:display --- Show Updated display

# c) window B becomes available – move the next person from the queue to this window!

lpush dmv:queue iris  
lpush dmv:window B  
hdel dmv:display B  
hgetall dmv:display  
  
rpop dmv:window  
rpop dmv:queue  
hset dmv:display B fran  
hgetall dmv:display

# d) jake arrives

lpush dmv:queue jake

lrange dmv:queue 0 -1

# e) window C becomes available – move the next person from the queue to this window!

lpush dmv:window C

hdel dmv:display C

hgetall dmv:display

rpop dmv:window

rpop dmv:queue

hset dmv:display C erin

hgetall dmv:display

7. Use spark to load the exam scores dataset `/home/jovyan/datasets/exam-scores/\*.csv` into Redis under the namesspace **examscores**. Use spark to Demonstrate the data is there by querying it back out.

8. In Spark SQL, read the Redis **examscore** data into a temp view and get the min, max, and average exam score across all students. Write the data back out to Redis as **examscoresummary**, finally query the key in redis showing all values in the hash!