

HW1 Task2 Report

Venkata Satya Kavya Sree Bondapalli
725006670

23rd February, 2018

1 Time taken to complete this task

5 hours

2 Task

2.1 Writeup on Architecture & Functionality

Architecture:

- I used a Redis database for write and listen operations. This database is chosen in for this purpose in order to perform publish and subscribe functionality as a part of write and listen operations.
- I used a MongoDB database in order to perform read, write operations for every message board. I chose this because I wanted to retrieve all the messages related to a particular message board for read operation.

Functionality of operations:

- Select <board_name>: Sets a variable name msg_board to <board_name>
- Read: From the mongoDB used, display all the entries of the messages corresponding to msg_board name using `collection.find({"NameOfMessageBoard":msg_board})` command.
- Write: Publish the written message to the redis database. In addition, insert a new element into the MongoDB database using `collection.insert({"NameOfMessageBoard":msg_board, "Message":to_pub})`.
- Listen: Use subscribe functionality on the redis database to subscribe to the channel named msg_board.
- Stop: Set a flag to false saying that the user is not longer listening. This is in order to exit the loop of subscribing.
- For all the above operations, error checks for cases mentioned in the hw2 handout have been taken care of.

2.2 Instructions to run the code

- You need to have Python 2.7.6 version, pymongo 3.6.0, MongoDB 3.6.3, redis 4.0.8, redis-server v=2.8.4 preinstalled on your linux system
- Command to run the code: `python my_message_board.py`

- Note that `my_message_board.py` had dependency on `constants.py`, `mongo_connect.py` files in order to connect to MongoDB
- The host, database, collections parameters used for the MongoDB is: `host = "127.0.0.1"` `db = "task2_725006670"` `collection = "task2_725006670"`
- Note that the MongoDB database is stored on loop back IP address.