Assignment -3

The assignment creates the pub/sub and broker demonstration using a multithreaded architecture and zmq router-dealer pattern for broker.

The unit test that the pub/sub has 2 parts - 1) topic test – which tests how the selection of number of topics by pub/sub affect the latencies of message dissemination,Ownership strength QoS and History QoS. 2) Data size test here the publisher sends data of various size like 1k 10k and 100k to the subscriber and latency of message dissemination is calculated.

To run the test first start the broker by typing the python broker\_1.py in one console. Then start the subscriber on the other console by typing python sub\_1.py and finally start the publisher on another console by typing python pub\_1.py.

The Unit test runs 4 test modules -

1)pub1(topic Weather , pubID:met-010,having highest ownership strength ) -> sub1 (topic Weather)

2)pub2 (topic news, Weather,pubID:met-011,having lower ownership strength ) -> sub2 (topic News)

3)pub3-> sub3(topic Stock and topic movie) and prints the time stamp at pub and sub . We need to calculate the delta of time from Pub to sub as shown in excel sheet by entering the timestamp values of pub and sub .

4) Ownership\_strength:Pub1 publishes Weather with pubID met-010 (Nashville weather of 5 samples) and Pub2 also publishes with pubID met-011 (Knoxville Weather of single sample). Since Pub1 registers with broker earlier its ownership strength is more and it suppresses the Pub2 messages as long as its alive and subscriber Sub1 keep receiving pubID met-010 messages only for Topic weather

5) History:Pub1 with pubID met-010 maintains a History buffer of length 5.Sub1 requests retransmission of weather data after 2nd data it receives. So the Pub1 retransmits the data.

6)Data size test – This prints the timestamps for pub and sub for 1k 10k and 100k data size. The timestamp at both pub and sub sides are noted entered in excel sheet to calculate the graphs .

7) All synchronization is happening via zookeeper. Pub/Sub can join and leave at any time. This can be tested by running the pub\_1.py again and again with the broker running and by also by terminating the sub\_1.py using ctrl-c and restarting the sub\_1.py again and again.