Internet of Things

Assignment 1

Report:

Programming Language Used: **Python**

Operating System Used: **Windows 10**

Broker: **HIVEMQ**

Approach:

First, PowerShell script is made which will ask user to enter how many UAV are available. Script will open HQ.py and uav.py file all instances. HQ will read vehicle location from input.txt file and sends vehicle locations to broker every 10 seconds, all UAV's are subscribed topic preference which will help to receive locations of vehicles.

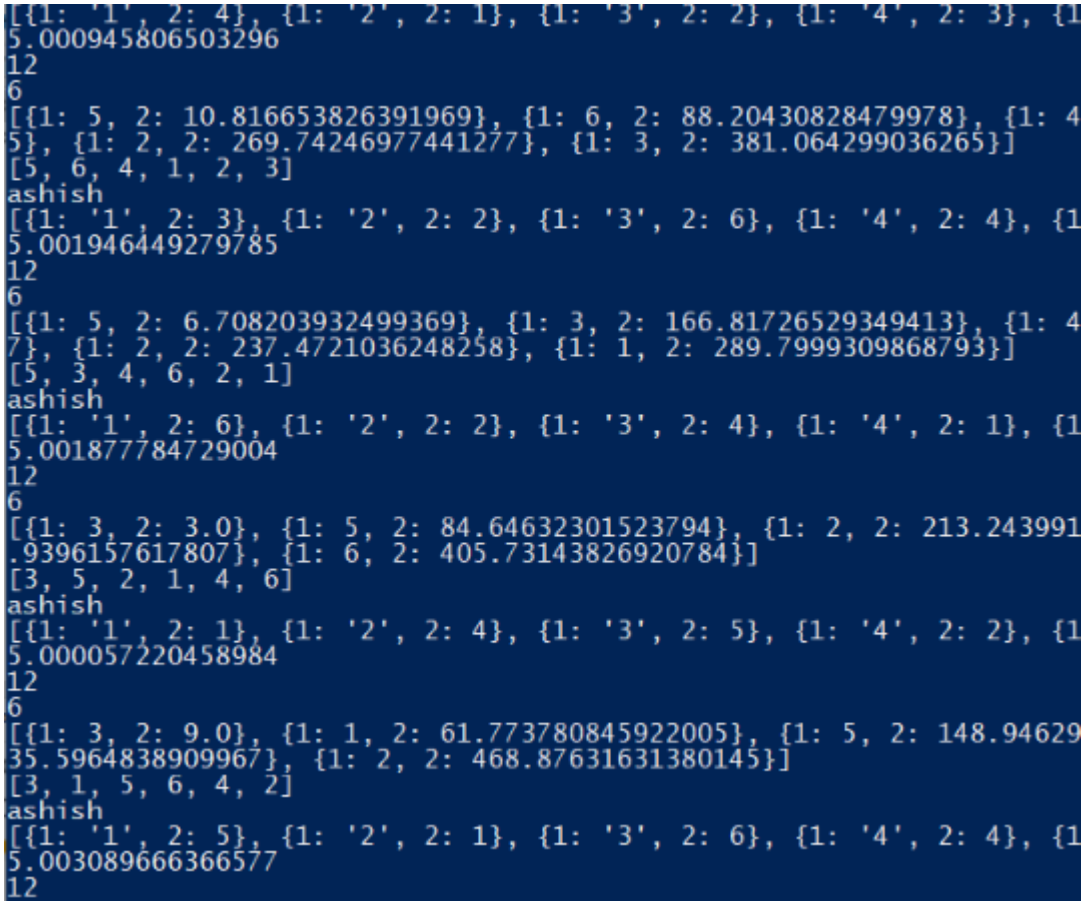
All UAV first reads there locations from their respective "uav.txt" and calculates the distance of all vehicles from their location and store it into a list. The list is then sorted on the basis of minimum distance of a UAV to any one of vehicle, It might happen that two or more UAV have their minimum distance to same vehicle so to assign a vehicle to only one UAV, we have to communicate with other UAV's so that no 2 UAV picks same vehicle.

To deal with breaking Ties, preference is given to lower ID ie. 1 has higher preference than 2,so if 2's 1st preference is same as 1st UAV ,then it has to select other vehicle which is next minimum distance in the preference list, for achieving this all UAV sends there preference list to all other UAV and then all UAV calculates which vehicle they should assign to because now they have preferences of all UAV and they can decide based on the ID preference to break the tie if any occur and then all UAV sends there assigned vehicle to back to HQ via broker.

Then HQ receives all the assigned vehicle information from UAV and now it write this information into an output.txt file.

Advantages of My approach:

1. HQ doesn't have to do any processing for breaking ties, all UAV by intercommunicating decide and come up with one preferences for each vehicle.
2. It avoids **single point of failure**, if any UAV breaks down, code still manages to come up with the preferences as all UAV are calculating their own preferences ,there is no single thread or process which is doing all the assignments.
3. If FCFS has been used like if 1 and 2 is sending their preference to 3 and 2 message came before one but in global time 1 computed before than 2 so it gives ambiguity to decide for 3 so we need time sync or global snapshots like using vector clocks in distributed computing which can increase computation cost and unnecessary delay because they TCP connection.
4. Minimum no. of message exchanged, all UAV's have to send their preference only one time to broker, so no requirement of message sending more than once.
5. Only UAV's first preference has to be sent not full list so size of packet is also less.
6. Average number of message exchanged by any UAV for each iteration of calculating vehicle=n+2(n=no. of UAVs)
7. Message size is 52 bytes (in case for 6 UAV), basically only 1 entry is sent with its ID concatenated.

8. 

As we can see it take on average around 5 second to exchange message with all UAV and sending it's preference to HQ where I am using time.sleep(5) for safe side so that all the message between threads are delivered before any further computation, so it take only on average microsecond to do the computation so total time =5 second plus micro seconds.

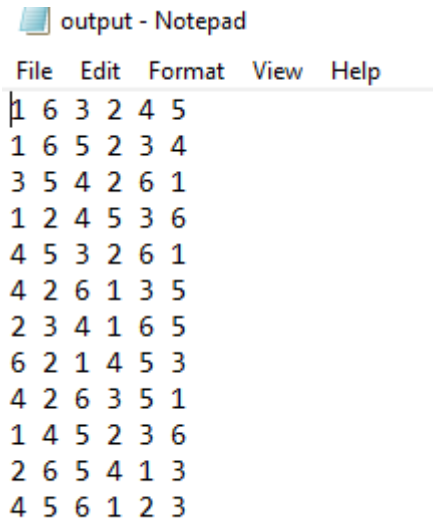
Screenshots

Power shell script

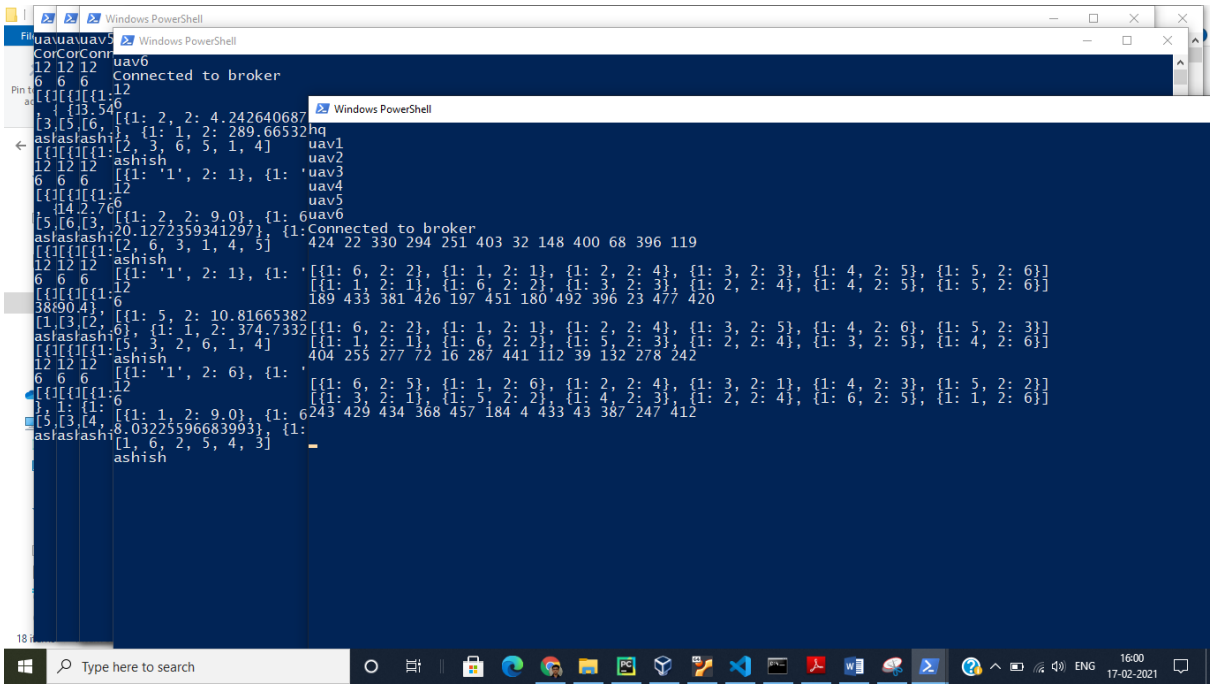
File Edit Format View Help

```
[int]$name = Read-Host -Prompt 'Enter the numbers of UAV?'
$name+=1
$i=1;
Start-Process powershell "python hq.py hq $name"
while($i -lt $name){
    Start-Process powershell "python uav.py $i $name"
    $i += 1
 }
 Read-Host -Prompt "Press Enter to exit"
```

Output



Termial output



Ashish Kishor

Roll No.CS20mtech11002