Joint Network Selection and Service Placement Based on Particle Swarm Optimization for Multi-Access Edge Computing

Due to growing number of mobile devices quality it’s become mandatory to improve quality of services which can enhance mobile battery life time and has to reduce response delay and to achieve this output they have introduce TASK OFFLOADING to cloud servers where mobile will offload heavy computation task such as analysing video frames to identify suspicious activity or to detect faces from images or any other task and cloud will accept that task and then process and send response back to client and this process will improve mobile battery life and processing will be faster. This advantage raise another issue called response delay or degrade in quality if cloud is far away from user then it will take lots of time to send request to cloud and to receive response.

To overcome from above issue another technique was introduce called Edge Computing and using this technique mobile will offload task to nearest Edge resources and then Edge will compute request and send response back to mobile and if edge cannot process request then it will offload task to cloud server and send response back to mobile. Edge computing able to improve quality services by reducing response time and by decreasing mobile battery consumption. Edge computing some time may face another issue called task overloading of more number of mobile send request to same edge server then edge server will get overloaded and then quality of services will be affected.

To overcome from thus edge server overloading author of this paper has introduces concept called Network selection & service placement based on PSO algorithm under multiple access edge servers. In propose algorithm all edge servers, services and users request will forms a particles and then PSO algorithm will evaluate each resource to find optimal solution also called as optimal edge server selection. PSO will evaluate each edge server in terms of physical distance between users mobile and edge server and then check available load on each edge servers and the edge server with least distance and least load will be selected as optimal network selection and then place service on that selected edge server.

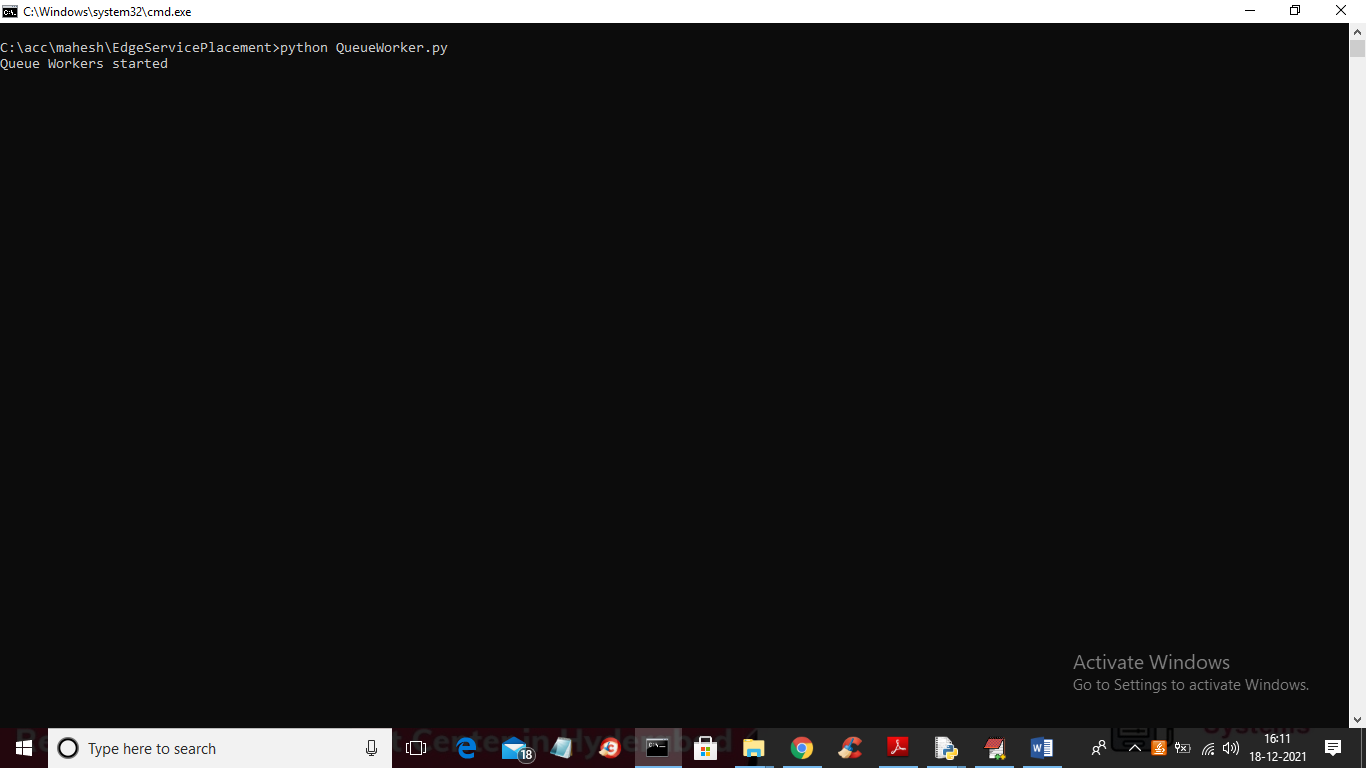
Existing technique were always sending request to nearest edge while propose technique will find optimal network by using PSO algorithm and if selected edge overloaded then another edge server will be selected.

To implement this project we have designed 3 applications such as

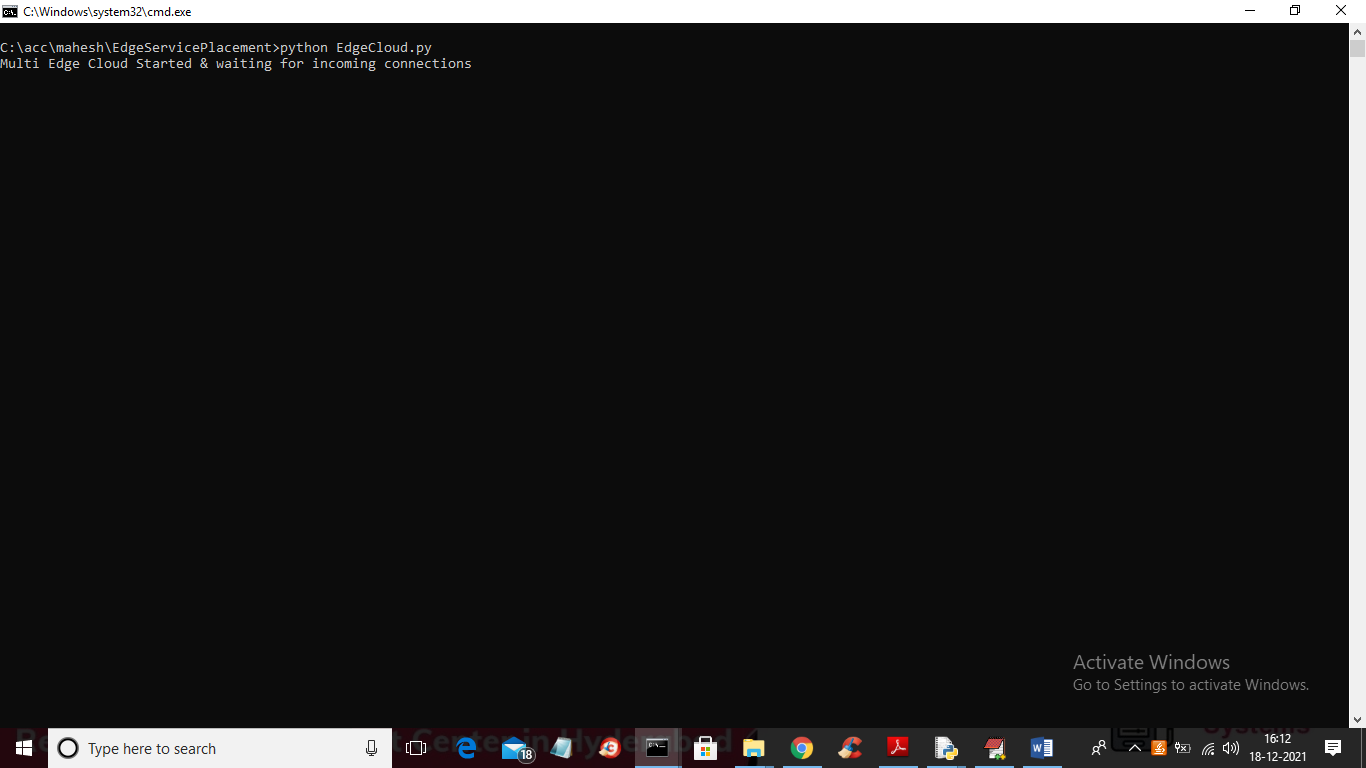
1. Queue Worker: this will queue all request evenly (one by one or smooth request sending) and then send to edge cloud and edge cloud will process request and send response back to client. PSO will evaluate each edge server for optimal network selection
2. Edge Cloud: this will queue and send request unevenly every second to selected edge server and edge server will process request and send response back to client. By using above two technique we will calculate response or computation cost delay.
3. Local Run Task: using this we will run task locally as task I will upload person image and then mobile will run task to detect faces from uploaded image and then calculate computation time or cost.
4. Offload Evenly Task: using this we will offload task to selected edge as task I will upload person image and then this image will be offload to edge cloud for face detection and detected face will be returned to mobile and then compute total time taken by task offload technique
5. Unevenly Task Computation Cost Graph: using this we will plot graph for local task computation cost and offload task computation task in unevenly task execution where request will send and process unevenly
6. Evenly Task Computation Cost Graph: using this we will plot graph of local task execution and evenly send request to edge cloud as one by one and then compute computation cost.

SCREEN SHOTS

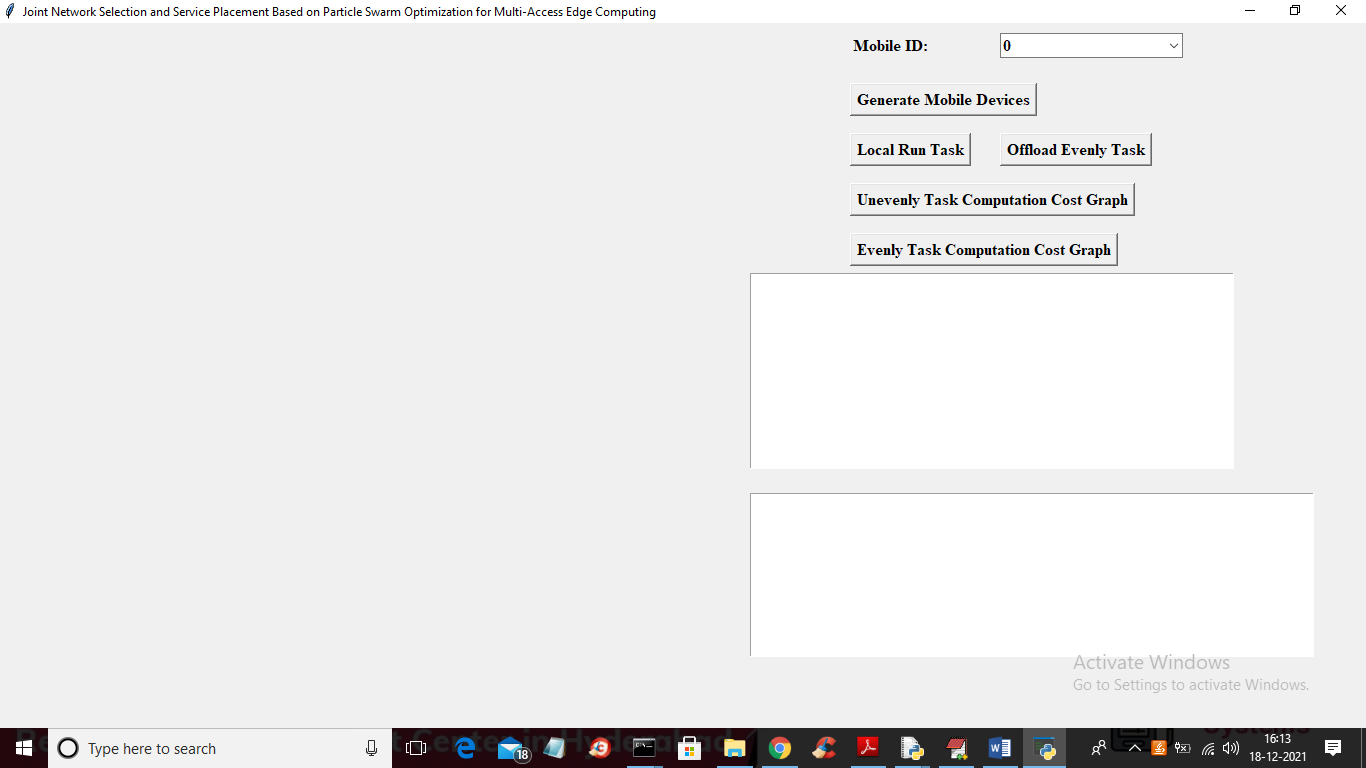
To run project first double click on ‘run\_worker.bat’ to start edge servers thread with queue and to gte below screen. This will process task evenly



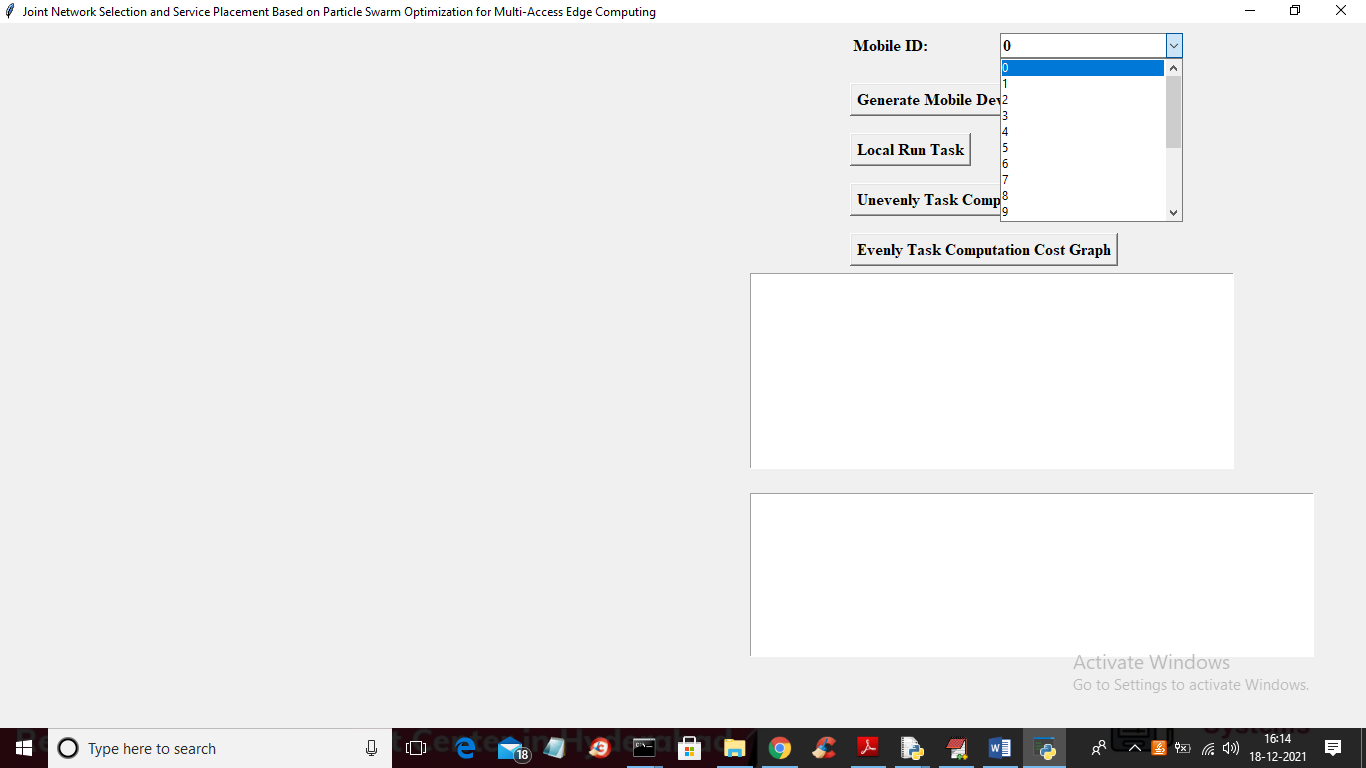
In above screen queue worker started and now double click on ‘runEdgeCloud.bat’ file to start edge server which will process request unevenly and to get below screen



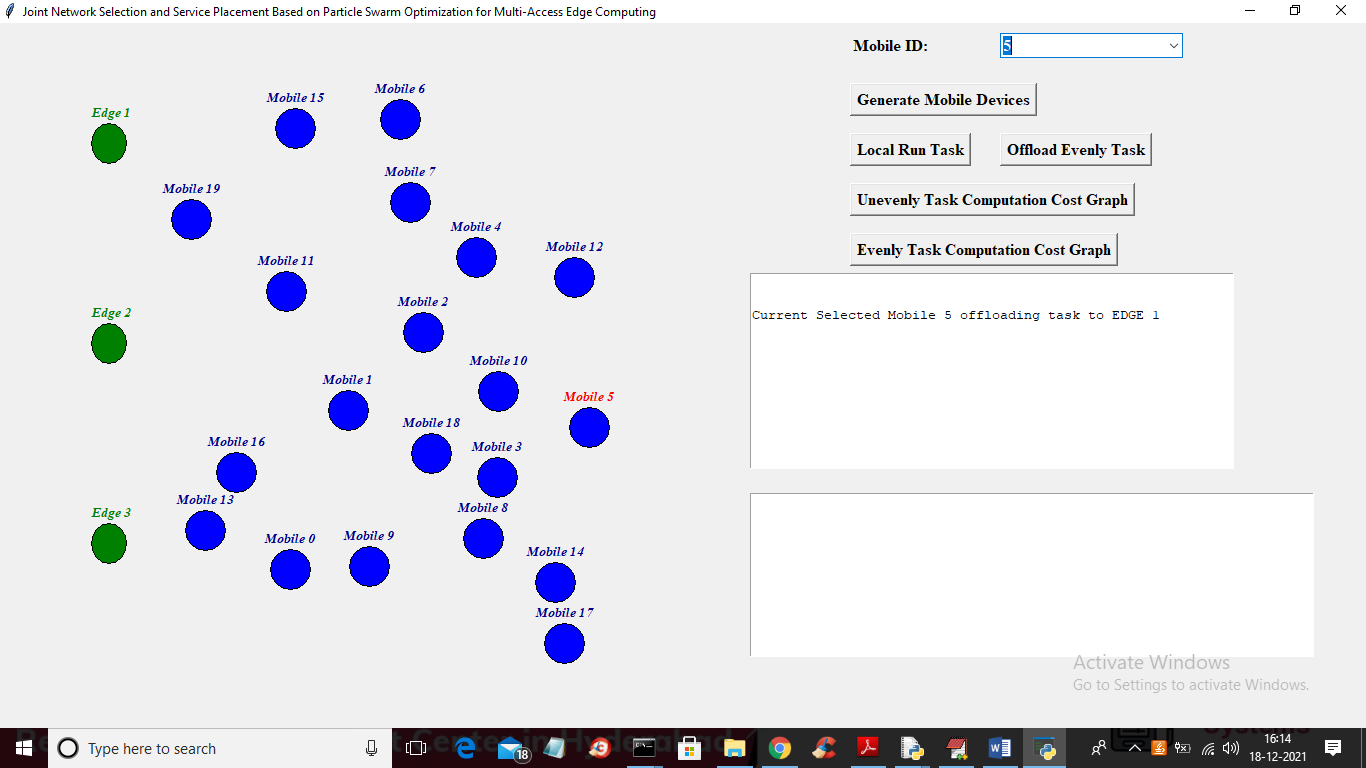
In above screen multi edge thread based edge cloud started which will process request unevenly from any mobile at any time. Now double click on ‘run.bat’ to start mobile application simulation screen like below one



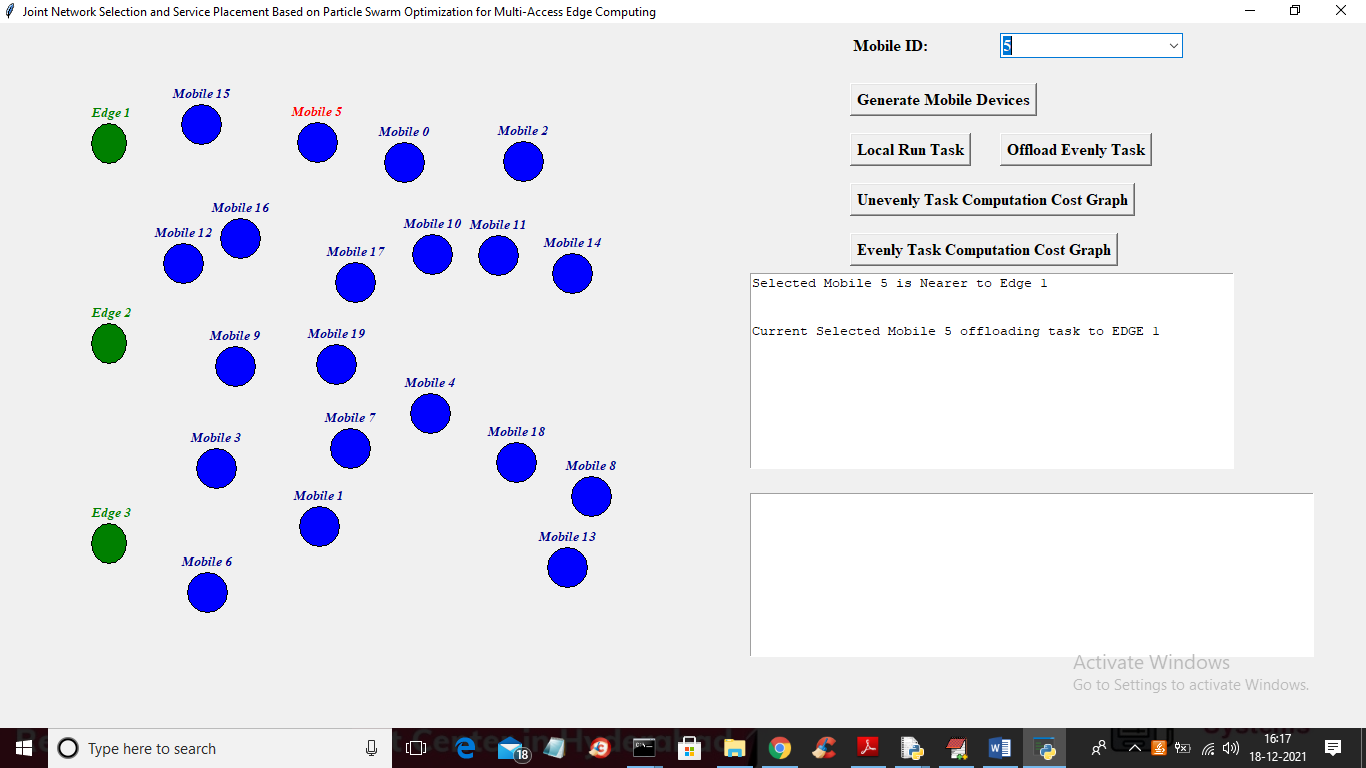
In above screen ‘Mobile ID’ contains all mobile’s ID like below screen



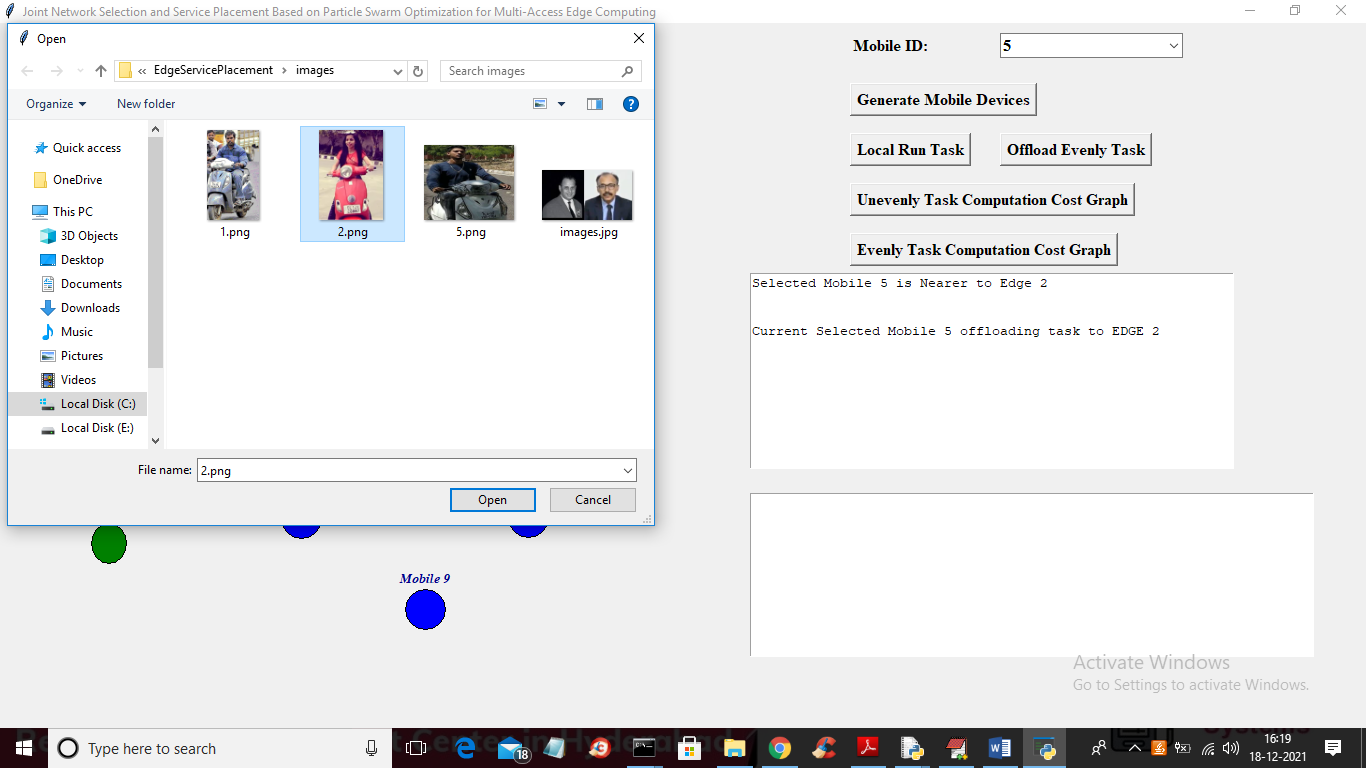
From above drop down box select any mobile ID and then click on ‘Generate Mobile Devices’ button to get below screen



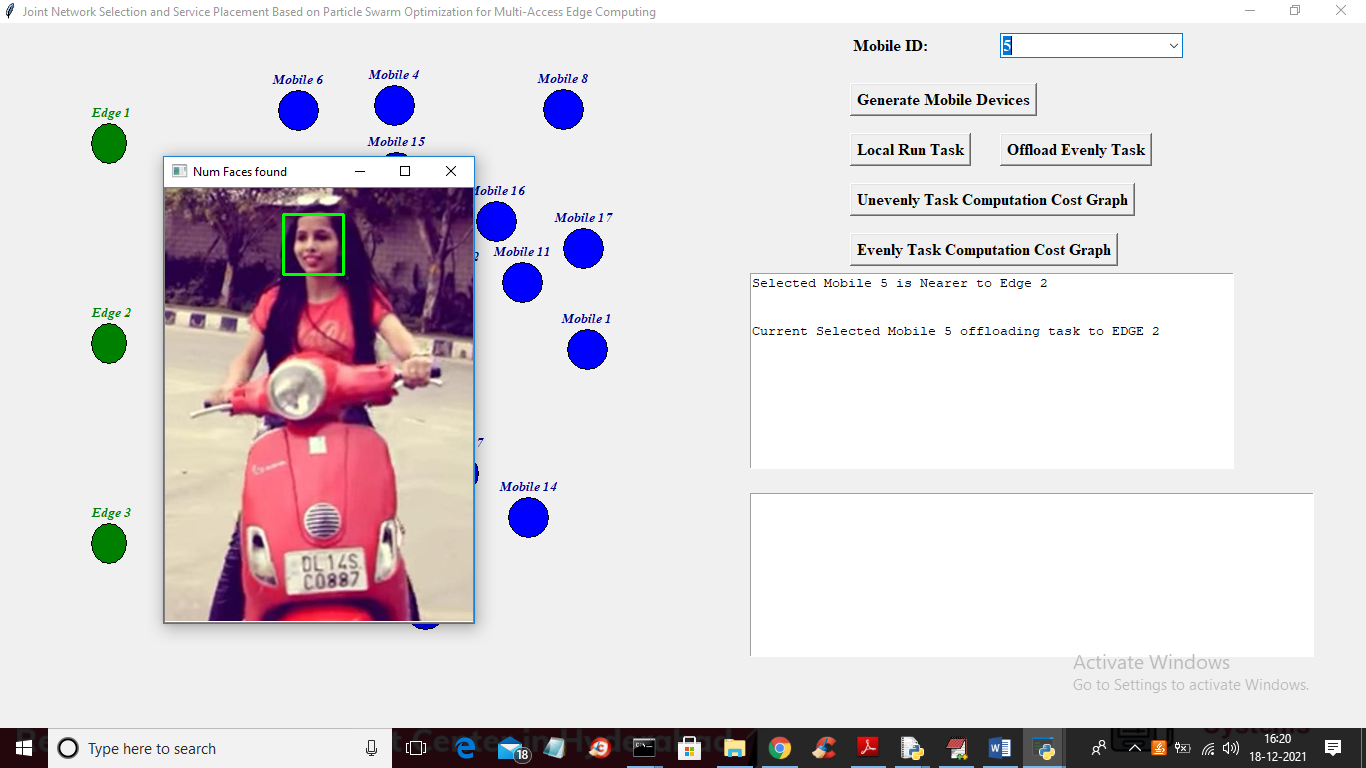
In above screen all blue colour circles are mobile devices and green colour nodes are EDGE servers and every second mobile devices will change their location and then select nearest edge server and in above screen I selected Mobile ID as 5 so every time mobile id 5 will change location and using PSO will find optimal edge server and offload dummy task and in above screen circle with red colour label is the selected mobile ID



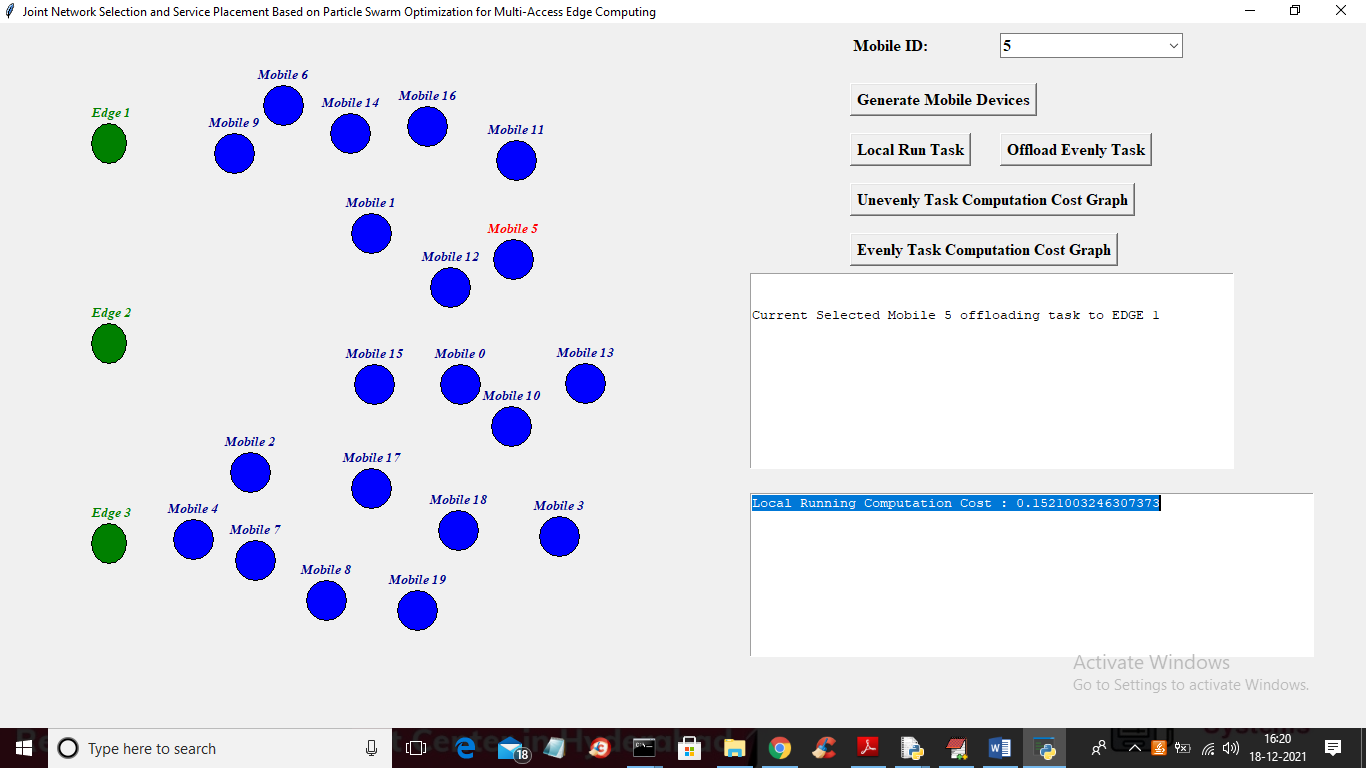
In above screen you can see Mobile 5 has changed its location and then calculated its nearest and optimal edge server and then display this output in text area. In above screen in first text area you can see Mobile 5 unevenly selected and offload task to nearest edge server. Now click on ‘Local Run Task’ button to upload image like below screen



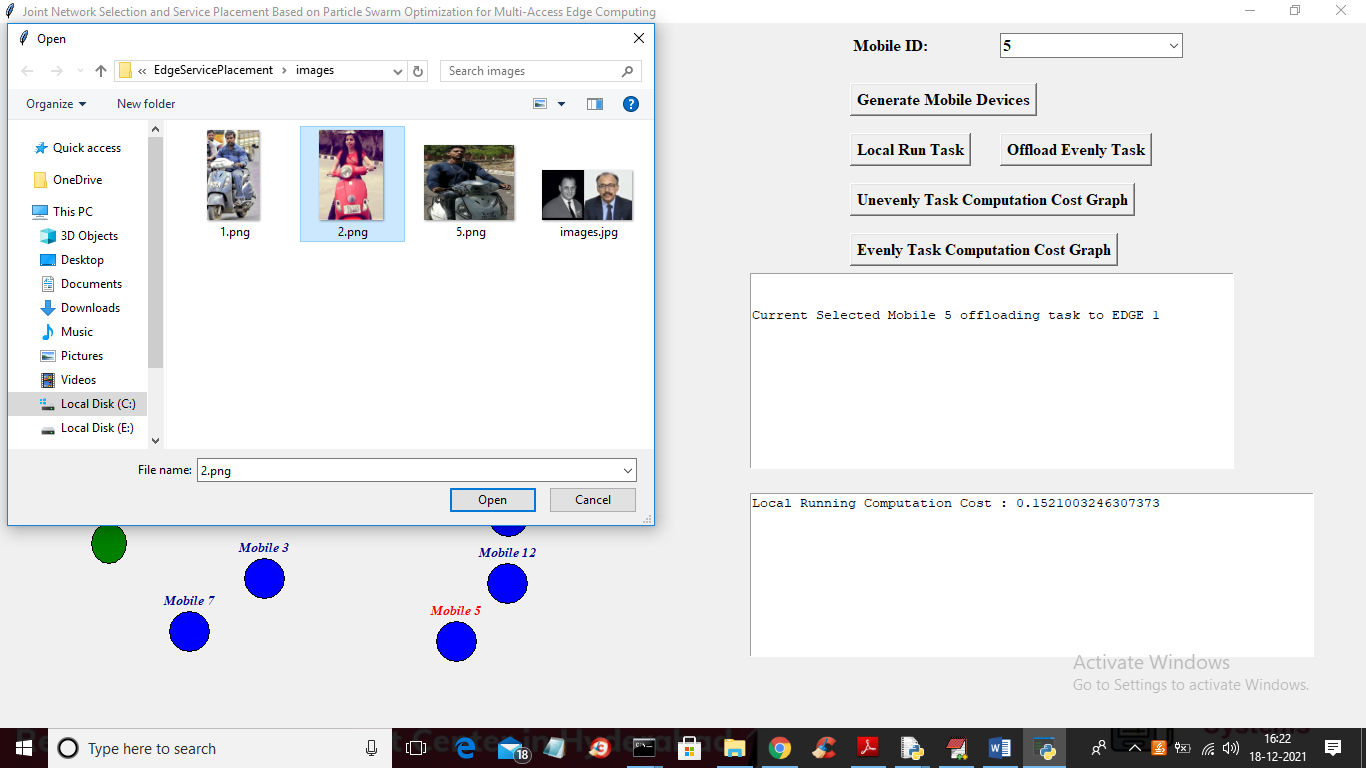
In above screen I am selecting and uploading ‘2.png’ file and then click open button to run task of face detection locally and compute computational task



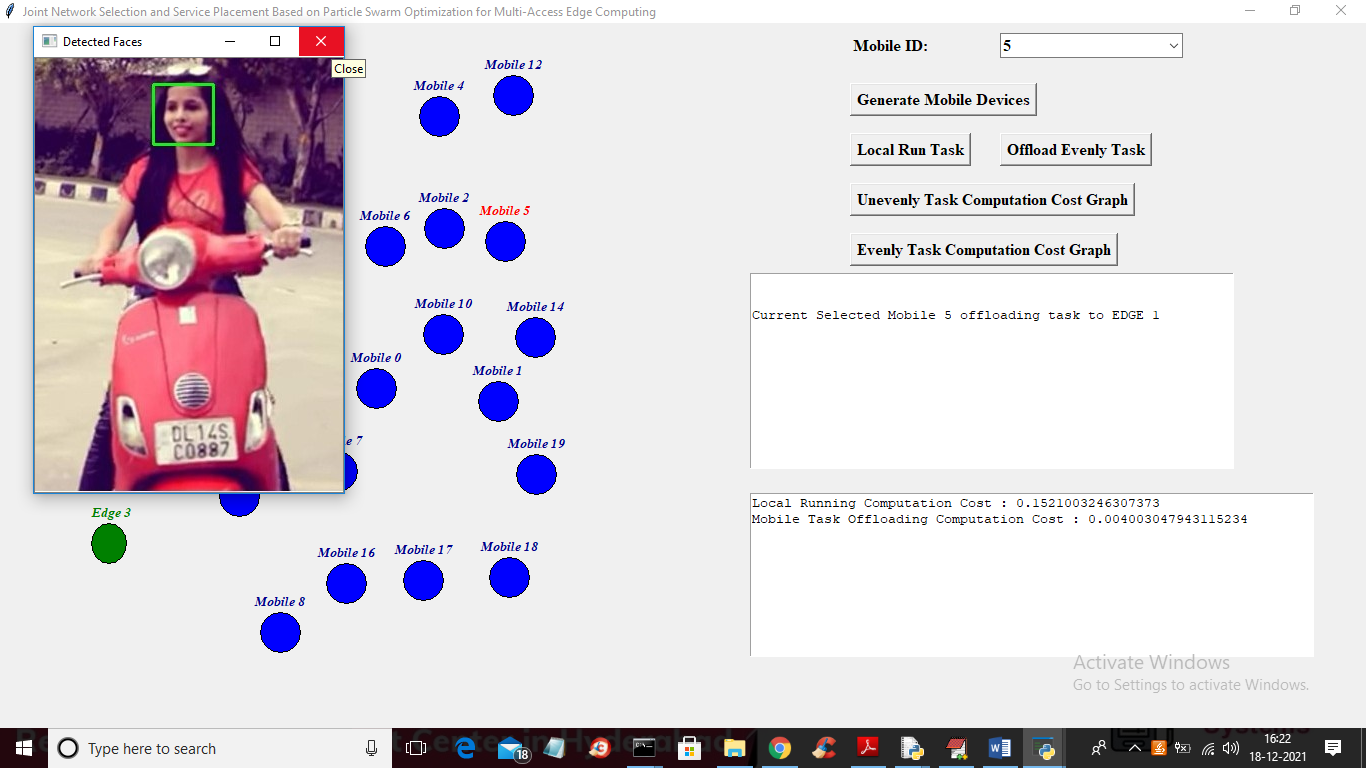
In above screen we can see mobile run task locally and successfully detected face and now close above image to get below task cost



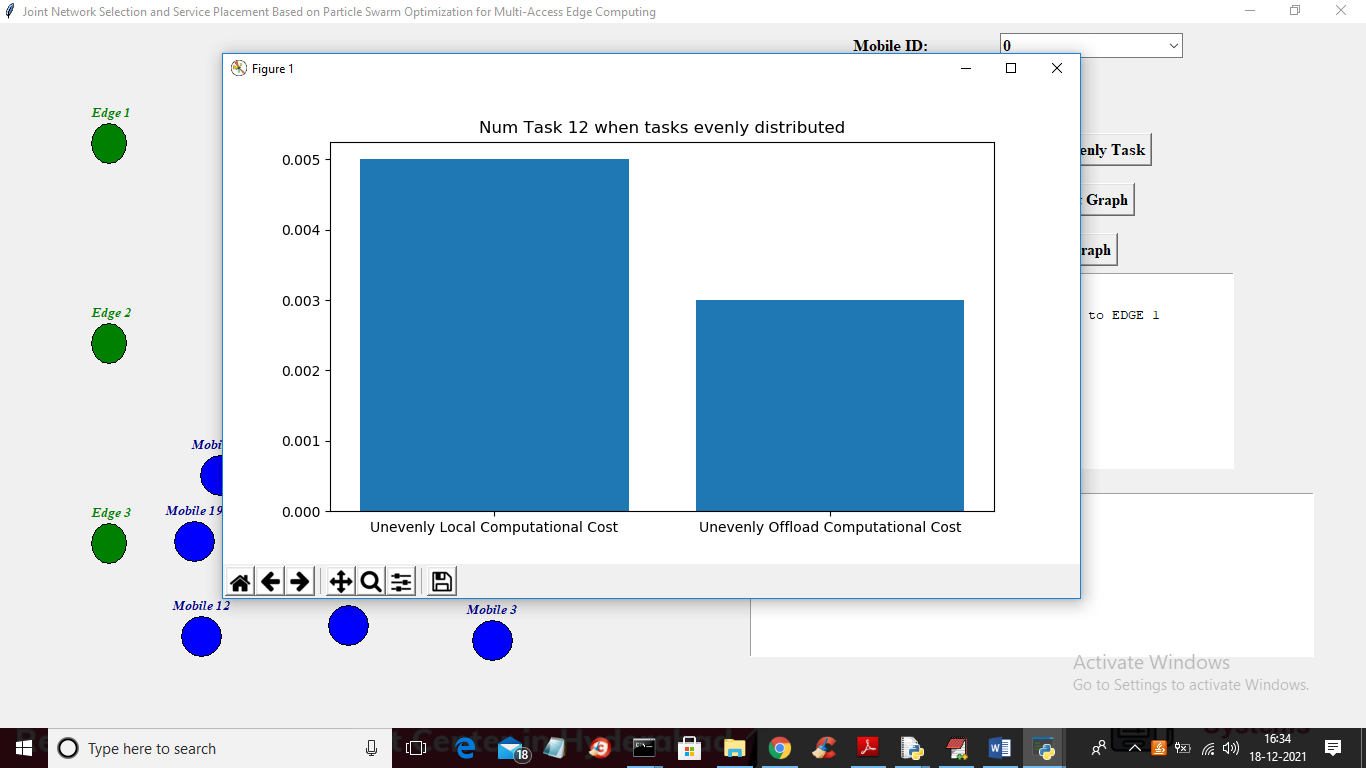
In above screen in second text area in blue colour text you can see mobile took 0.152 seconds as computation task to detect face and now run same task by offloading so click on ‘Offload Evenly Task’ button to offload task



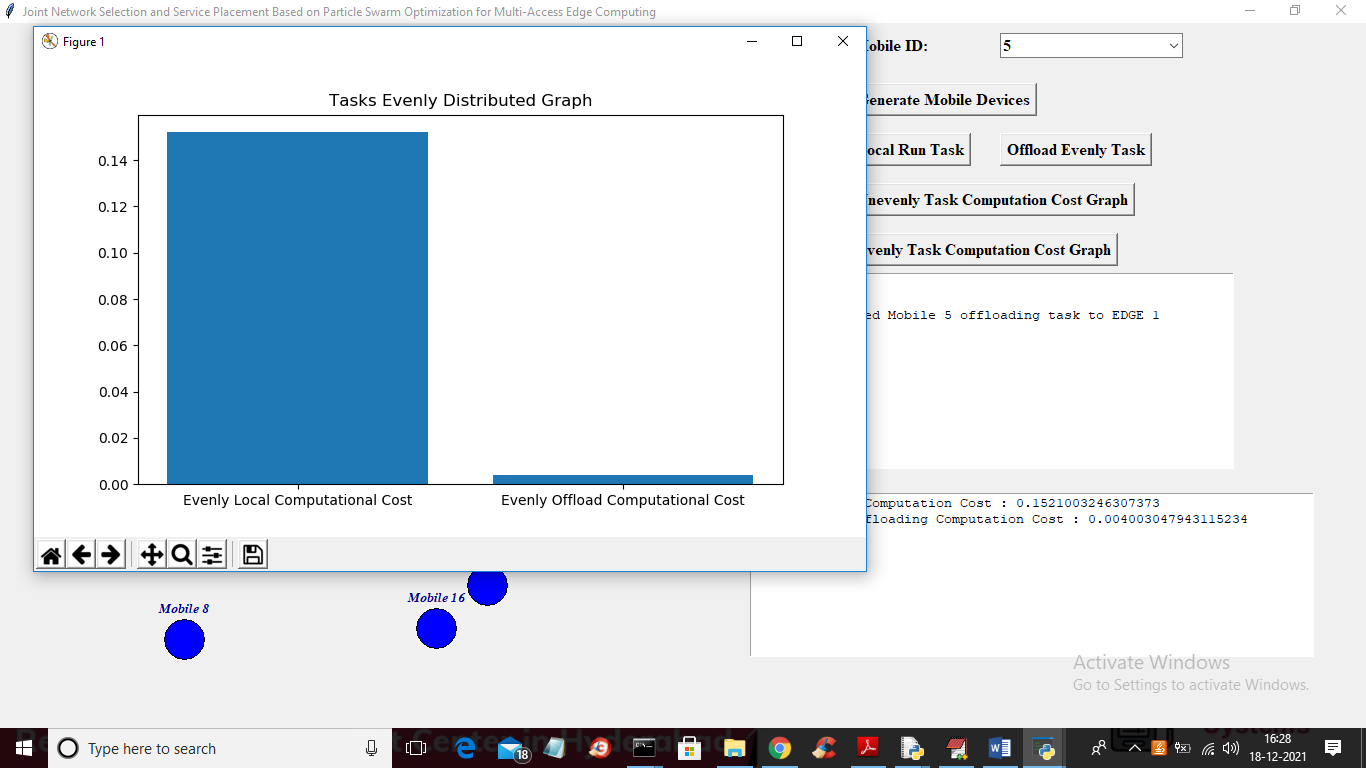
In above screen for task offload also I uploading image and then click on ‘Open’ button to offload task and to get below output



In above screen offload task also detected face and in second text area we can see for same output offloading task took 0.004 cost and now close above image and then click on ‘Unevenly Task Computation Cost Graph’ button to get below graph



In above graph x-axis represents technique names distributed task unevenly and y-axis represents computation cost of that technique. Now click on ‘Evenly Task Computation Cost Graph’ button to get below graph



In above graph x-axis represents technique names and y-axis represents computation task and if task offloaded using propose technique then must execution time can be reduce and response can be faster. In below edge cloud and queue worker we can see request receiving and processing evenly and unevenly

