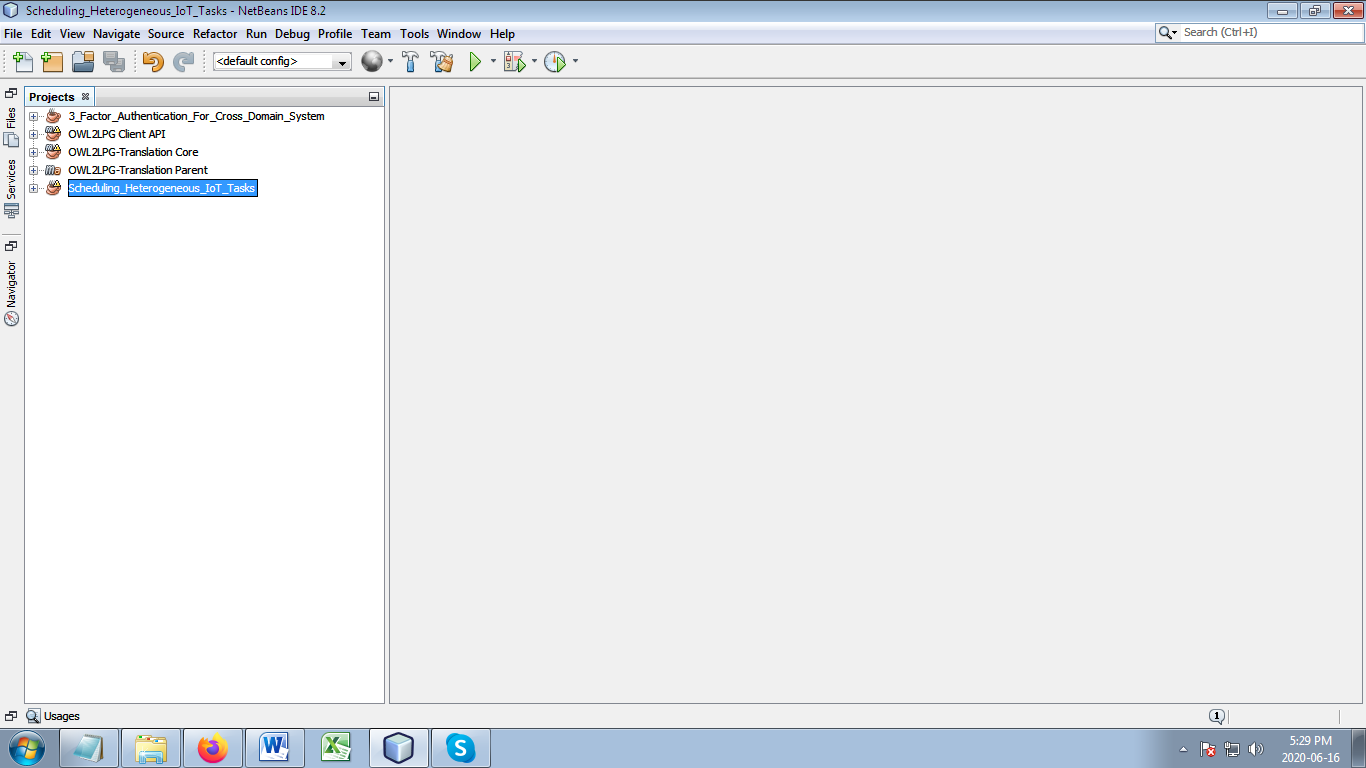
**Scheduling Heterogeneous IoT Tasks in Fog Cloud Model for QoS aware Semantic based Services Discovery**

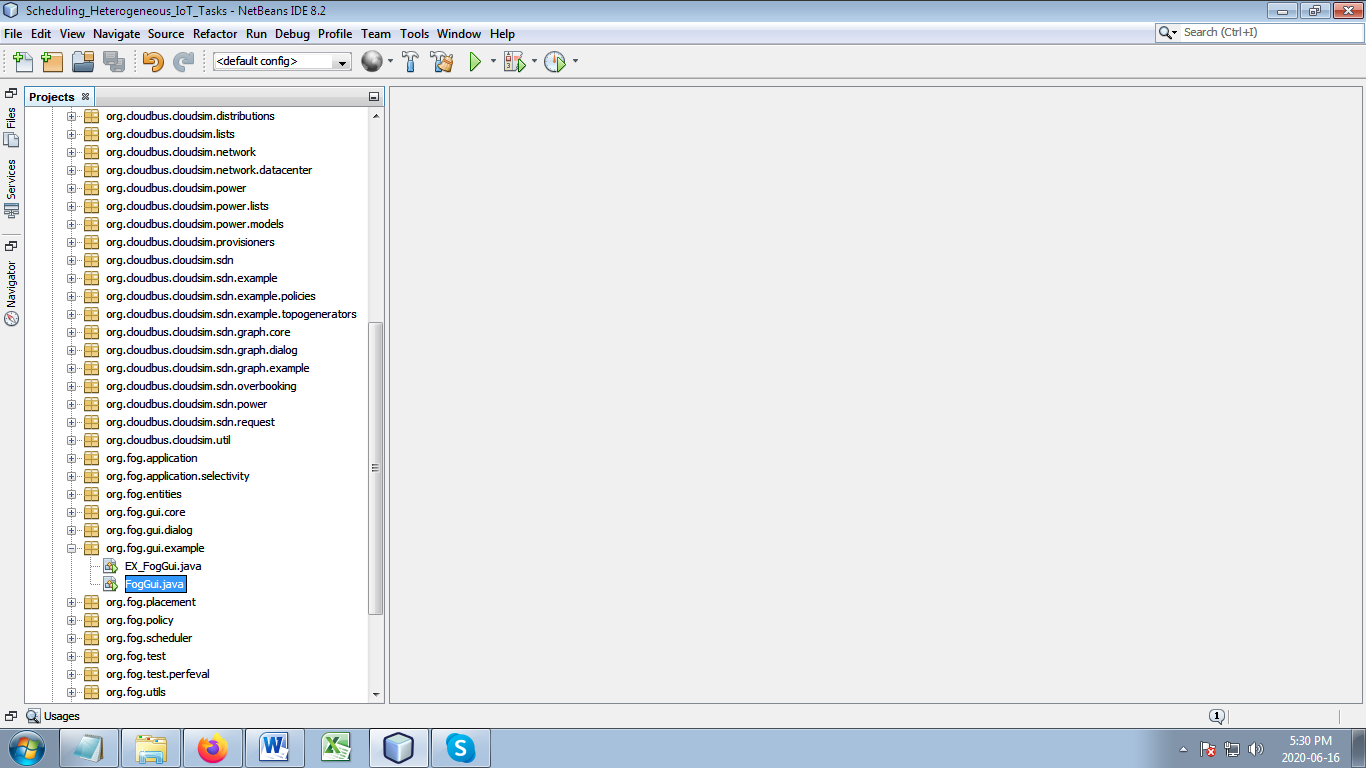
**Introduction**

We create a network topology with 30 IoT Users,20 IoT Sensor Devices,1 Task Manager, 10 fog nodes and 1 cloud server. Our proposed system has the following three layers are considered, i) IoT layer, ii) Edge layer and ii) Cloud. The implementations of IoT layer have the IoT users and devices. It has following the process are considered. IoT users to access the stored data from the cloud and The IoT devices collect the data and transfer via fog to the cloud. Next we apply the Edge layer by using two-tier task managers and fog devices. In this layer, we considered as the following process, First Tier – The received tasks from the IoT users are entered into queue1 and scheduled by using Policy-based AHP scheduling. Then it is assigned to the nearest fog. It is based on the task deadline, QoS, and response time. And Second Tier – Using QoS based Double Q-learning algorithm that receives tasks from queue 2 and applies the IoT devices. It’s considered the metrics are load, task energy, and completion time. The fog is composed of Class-based Splay tree algorithm, using which the requested service will be in search. Then the cloud collected heterogeneous data is formed into ontology from the ontology of each class a dependency graph is constructed as per the semantic similarity using Switchable RNN Ontology Matching. The threshold defines the number of arrived search tasks. If the number of requests is beyond the threshold then RNN activates GRU otherwise LSTM will be used. The overall system is evaluated in terms of following performance metrics, Number of Tasks with respect to Response time, Number of Tasks with respect to Success rate, Number of Tasks with respect to Satisfaction rate, Number of Tasks with respect to Service latency, Number of fogs with respect to Network Usage and Number of Requests with respect to Accuracy in retrieval,.. we make a Existing process based on An efficient indexing for Internet of Things massive data based on cloud-fog computing.

Open project in NetBeans

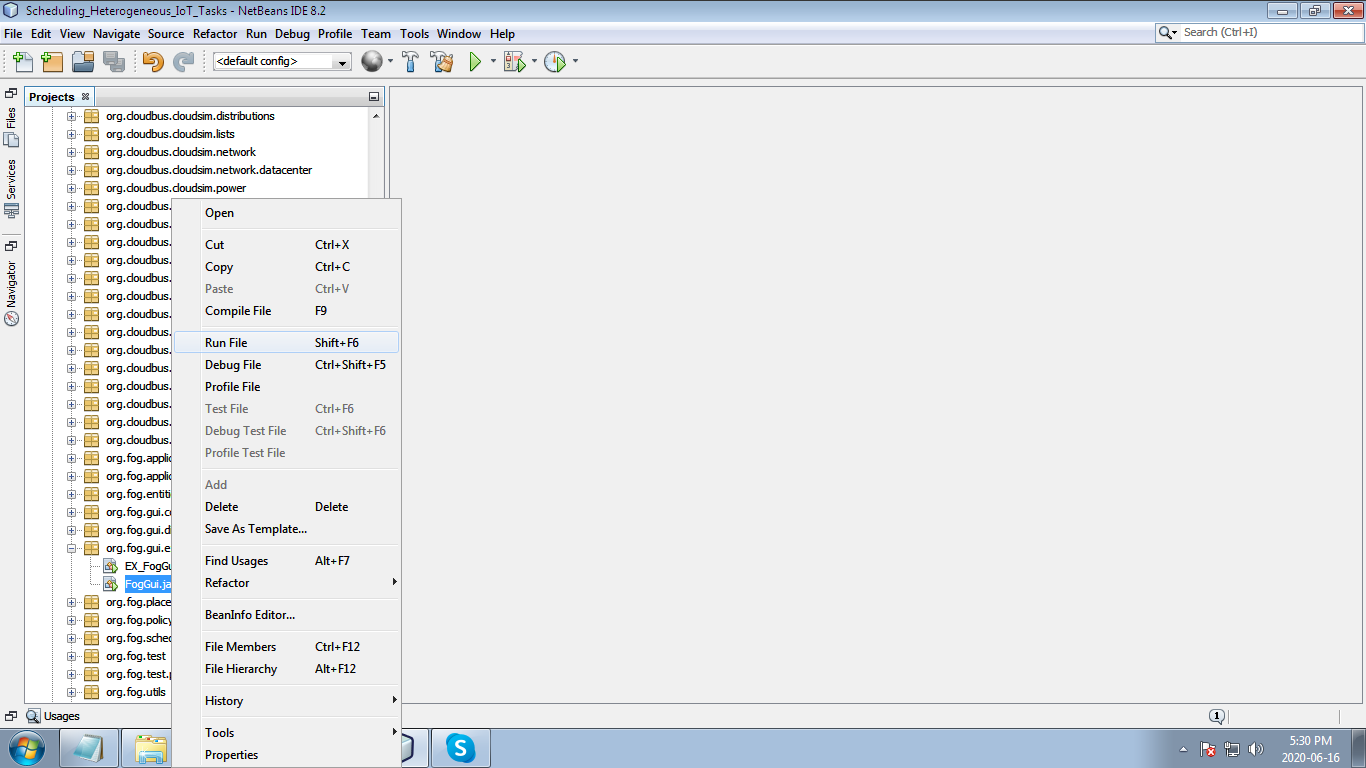


Choose Source Packages -> org.fog.gui.example

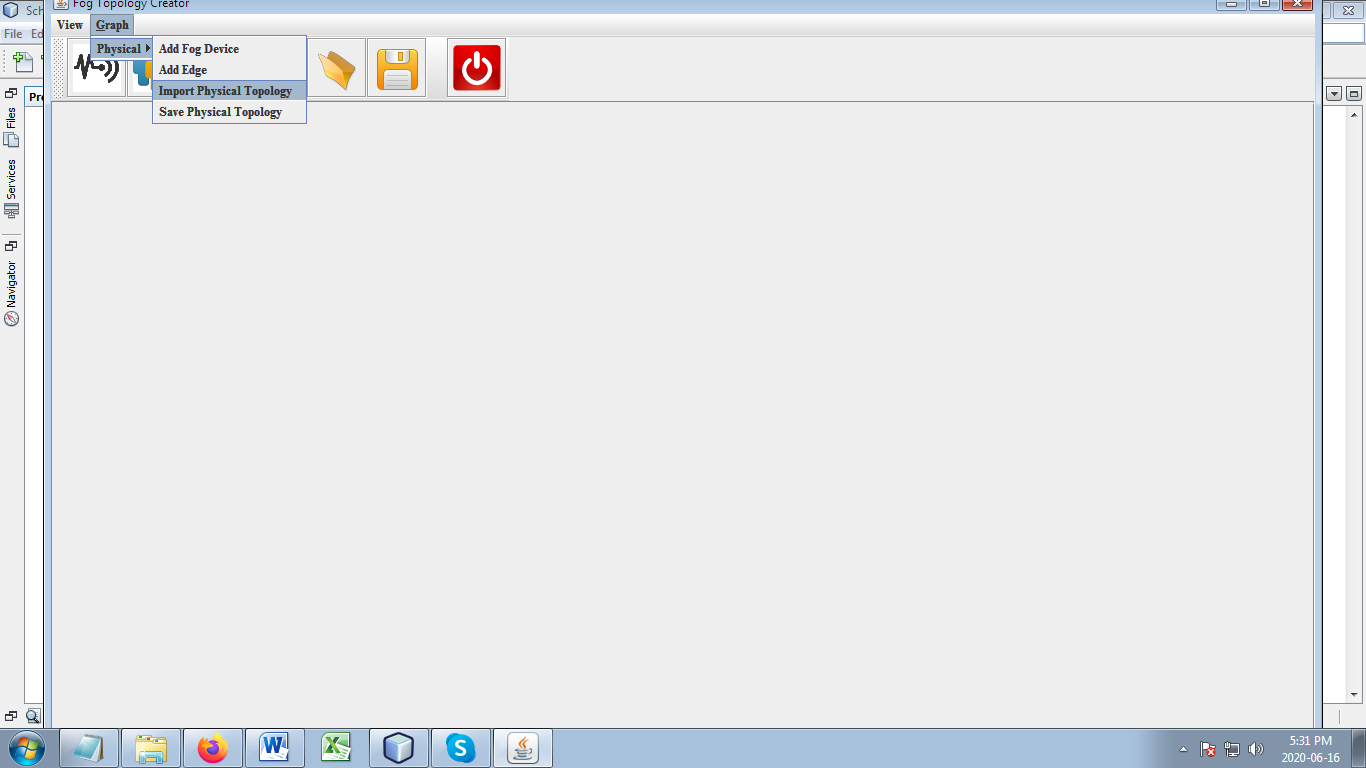


Right Click on FogGUI.java under org.fog.gui.example package

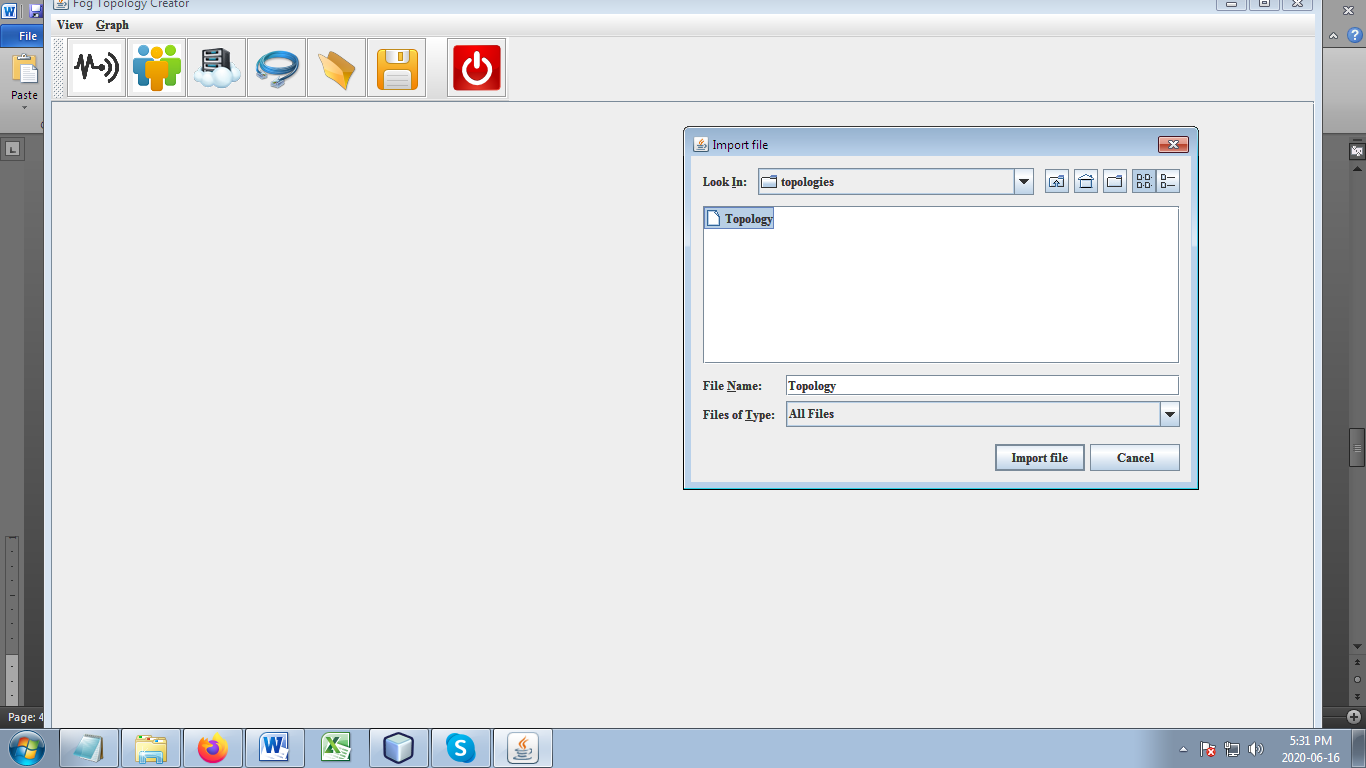
Then Choose “Run File” to run your project



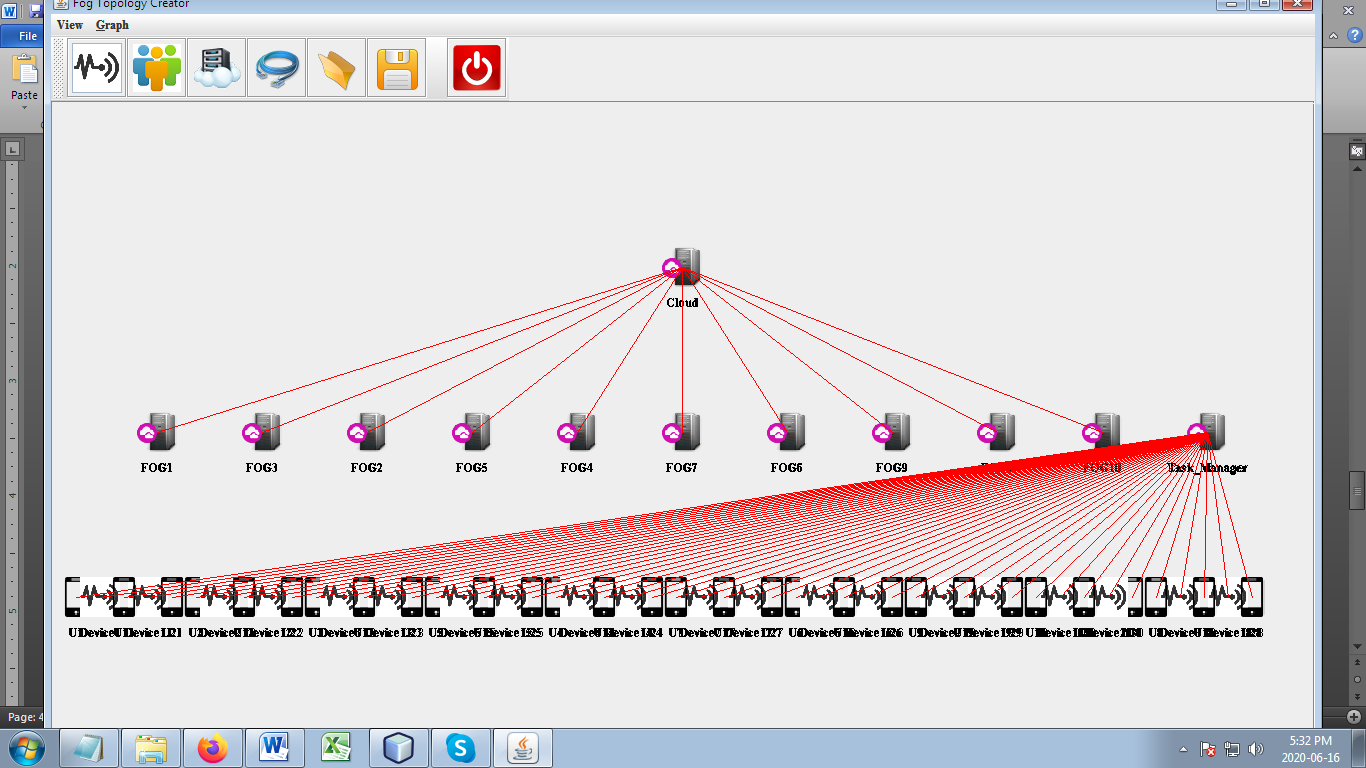
This is your start form Create Topology here (import physically or manually)



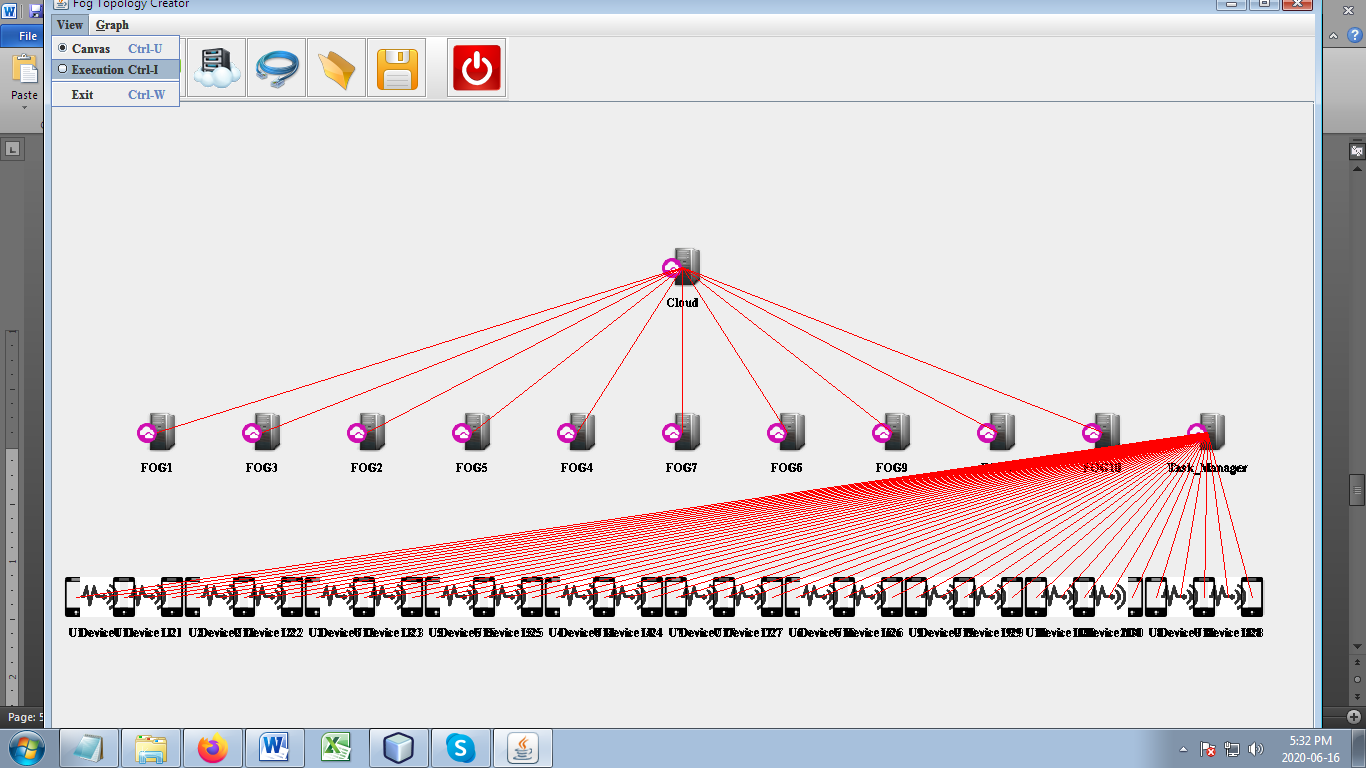
Already we create a physical topology network named as ‘Topology’, to import our project.



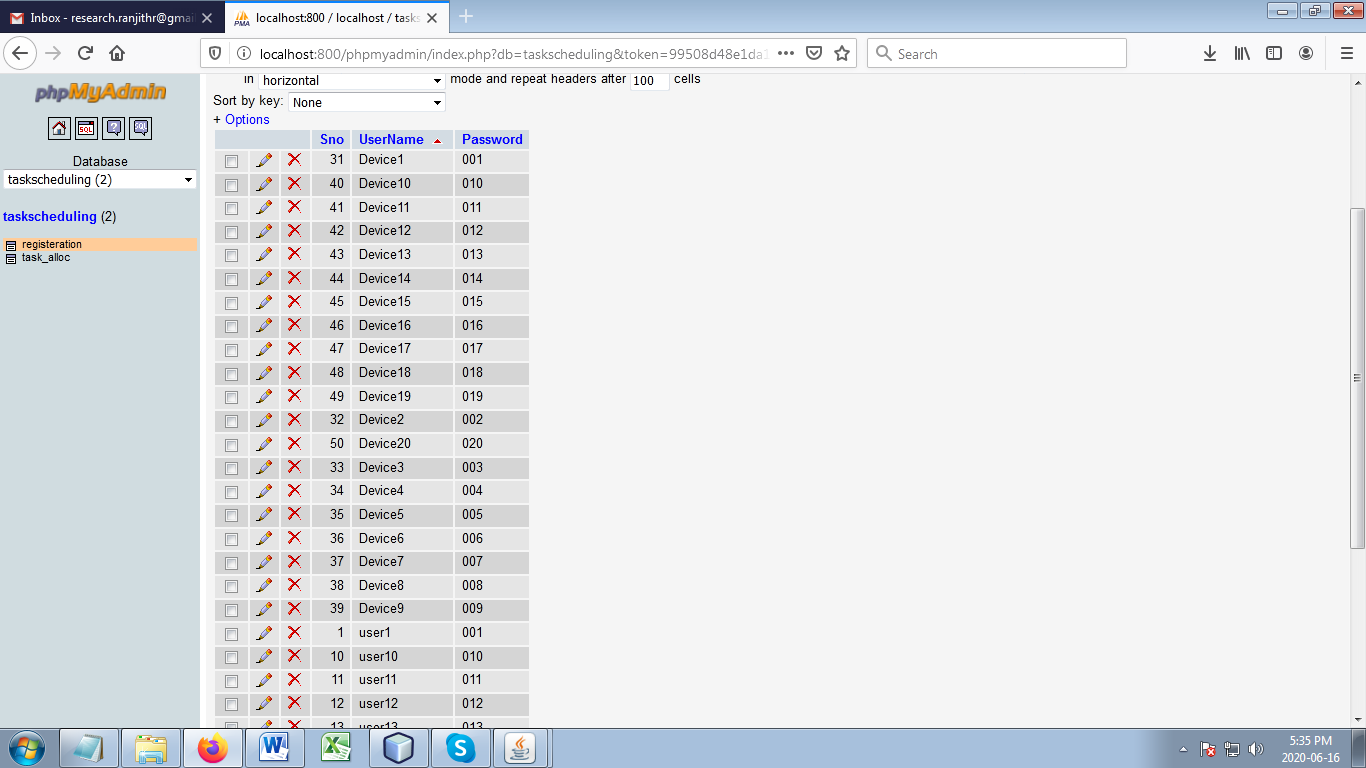
We create a network topology with 30 IoT Users,20 IoT Sensor Devices,1 Task Manager, 10 fog nodes and 1 cloud server. Our proposed system has the following three layers are considered, i)IoT layer, ii)Edge layer and ii) Cloud.



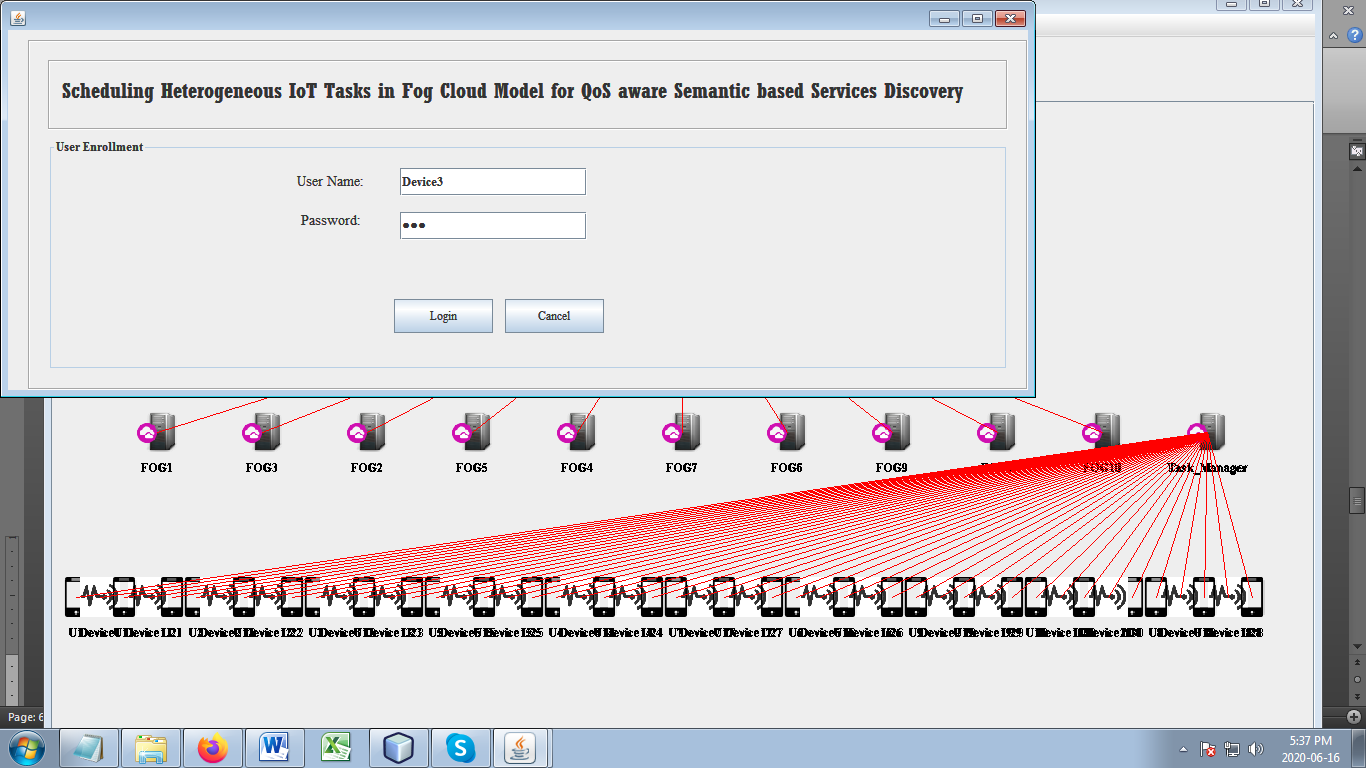
After successful Topology imported, Choose View-> Execution

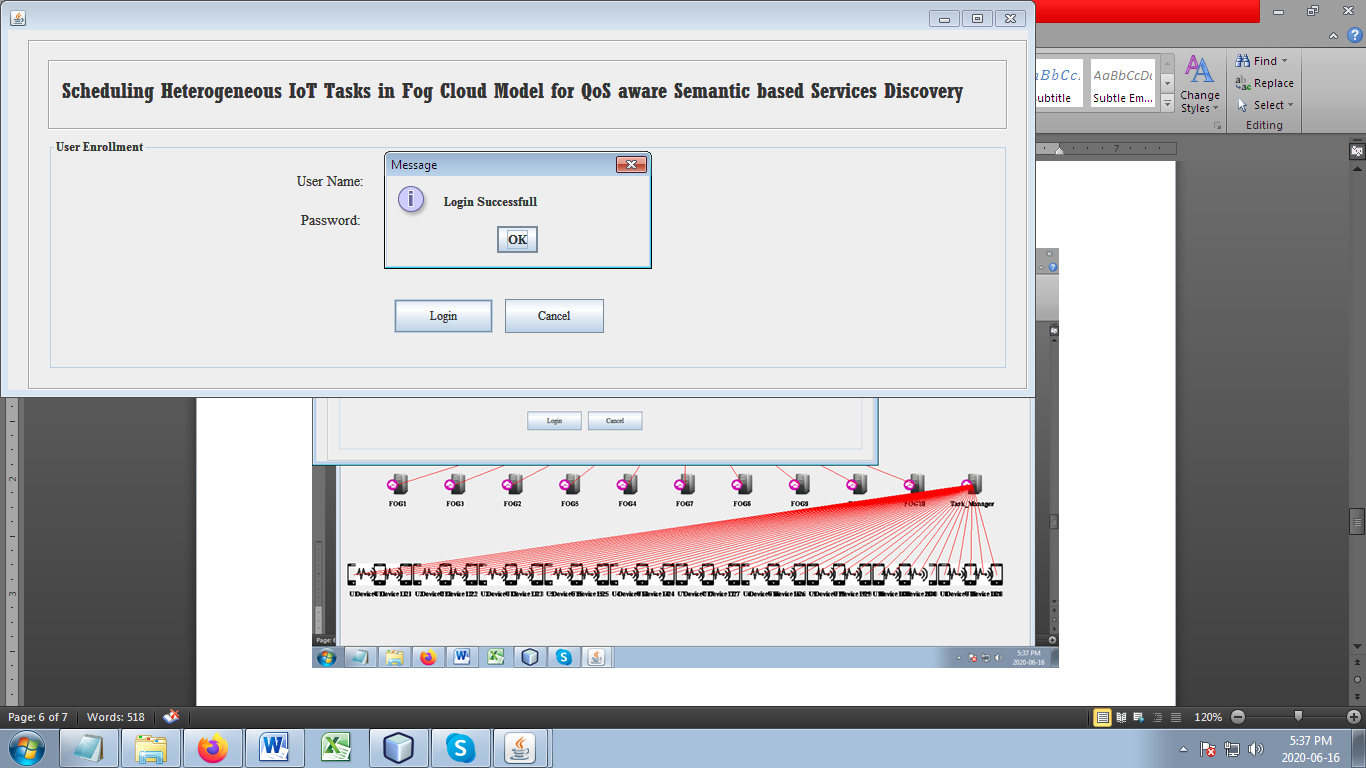


User and device registration details are already stored in the Database, by use the values the user and device make login process.

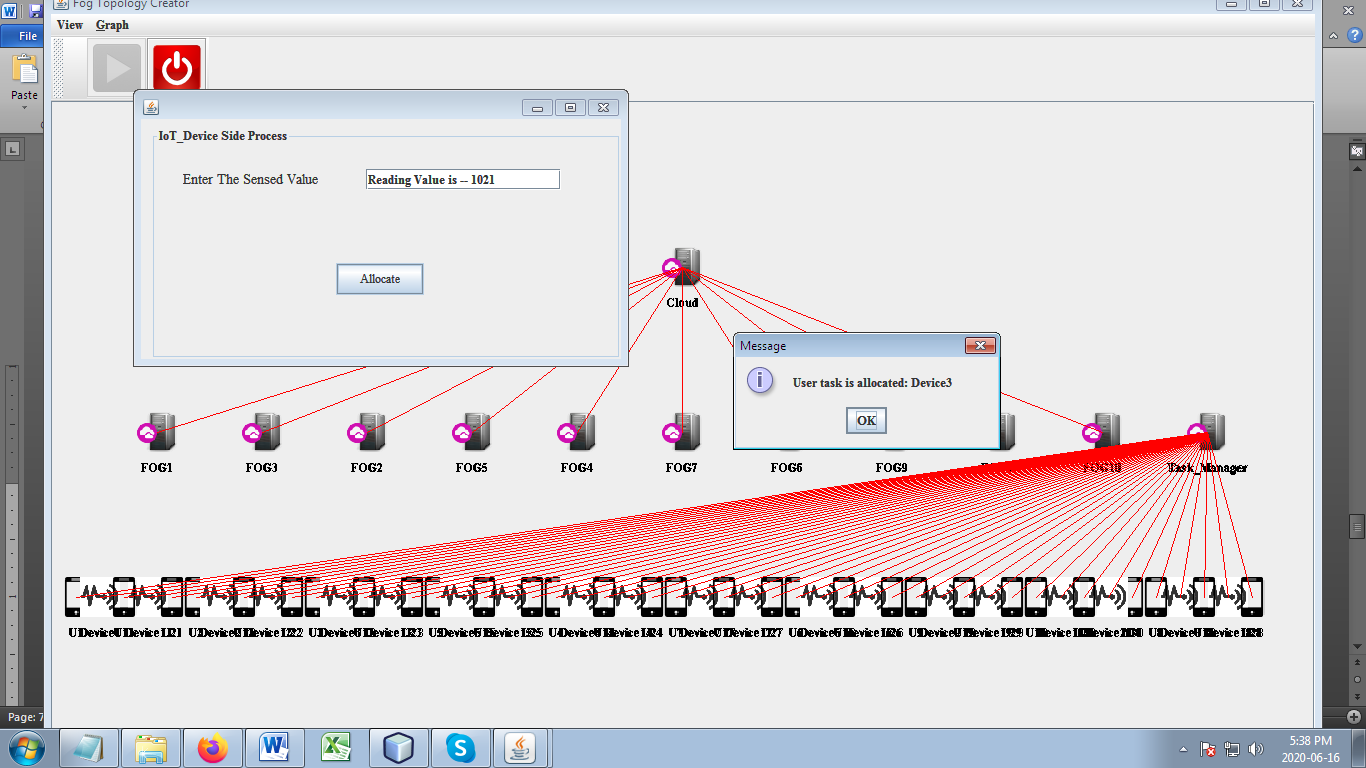


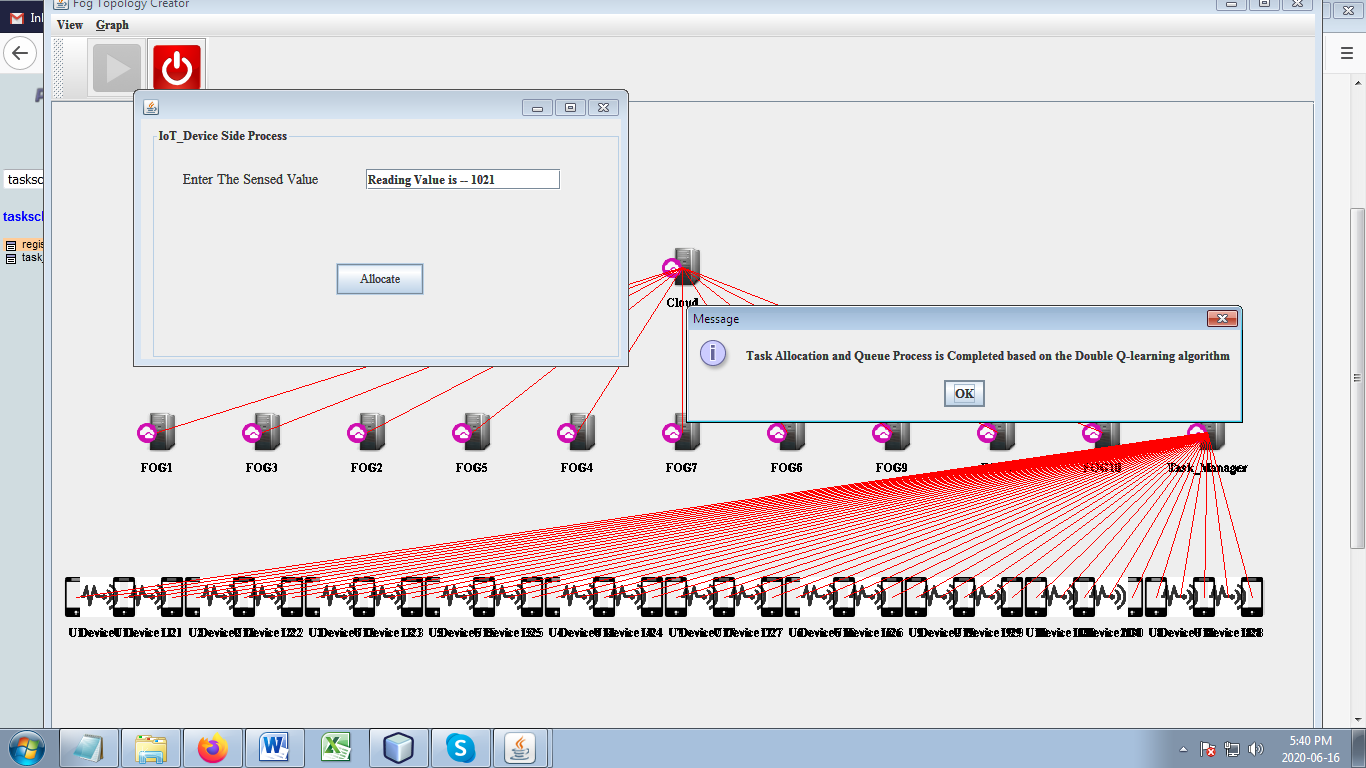
IoT Device login

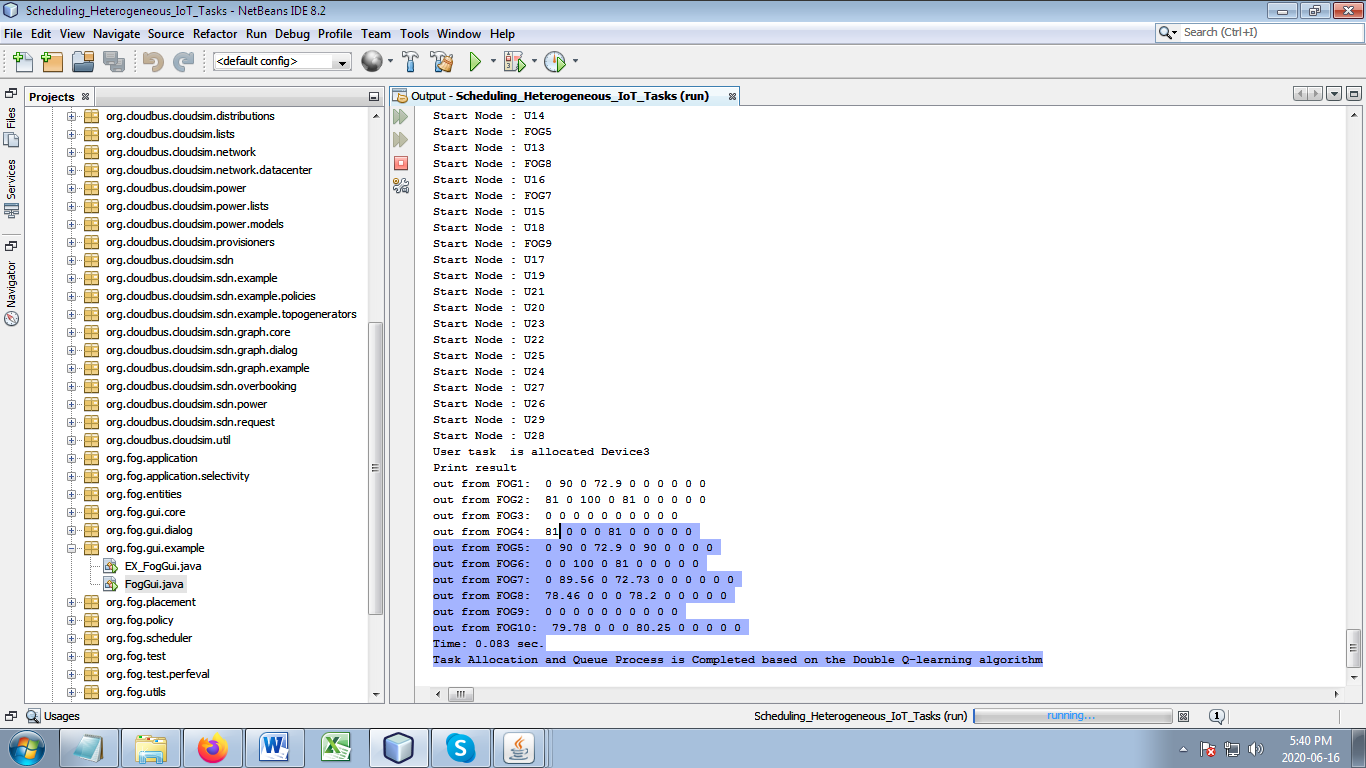




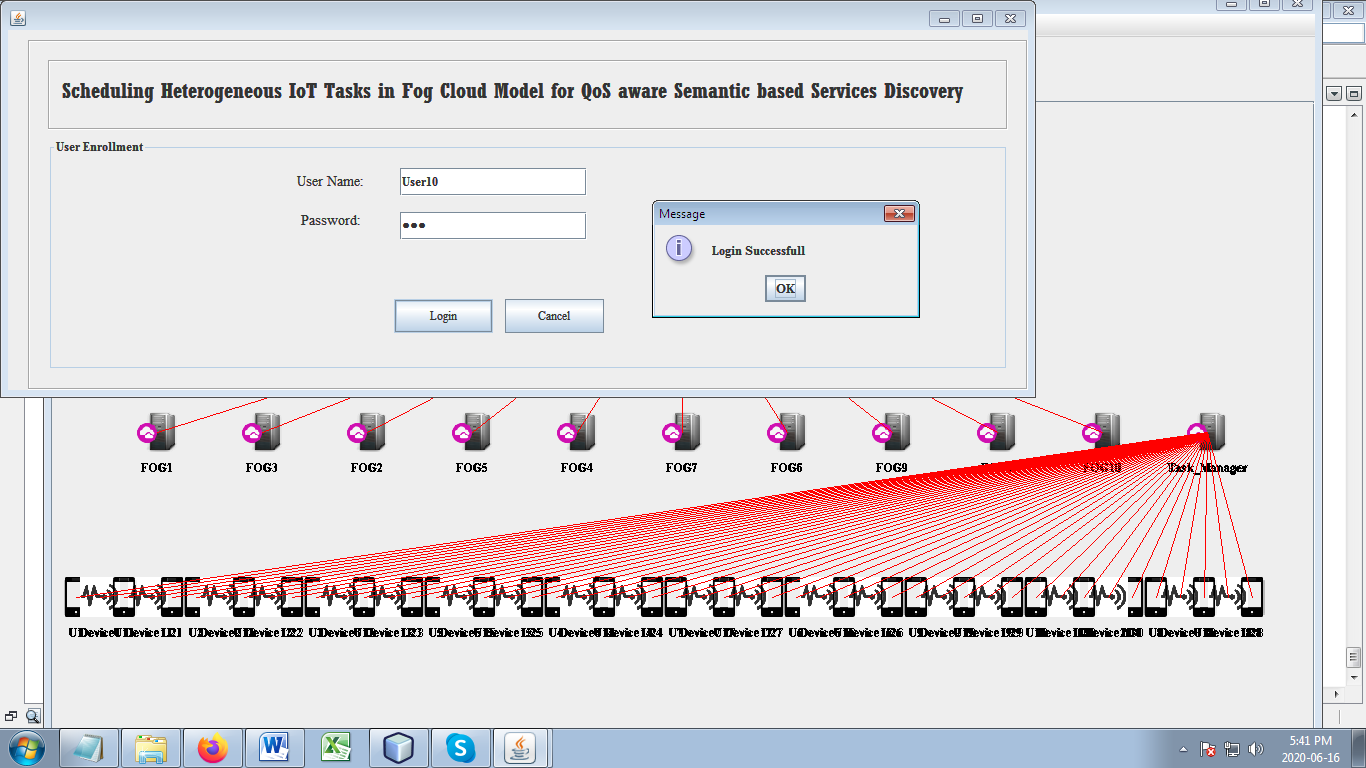
Using QoS based Double Q-learning algorithm that receives tasks from queue 2 and applies the IoT devices. It’s considered the metrics are load, task energy, and completion time.



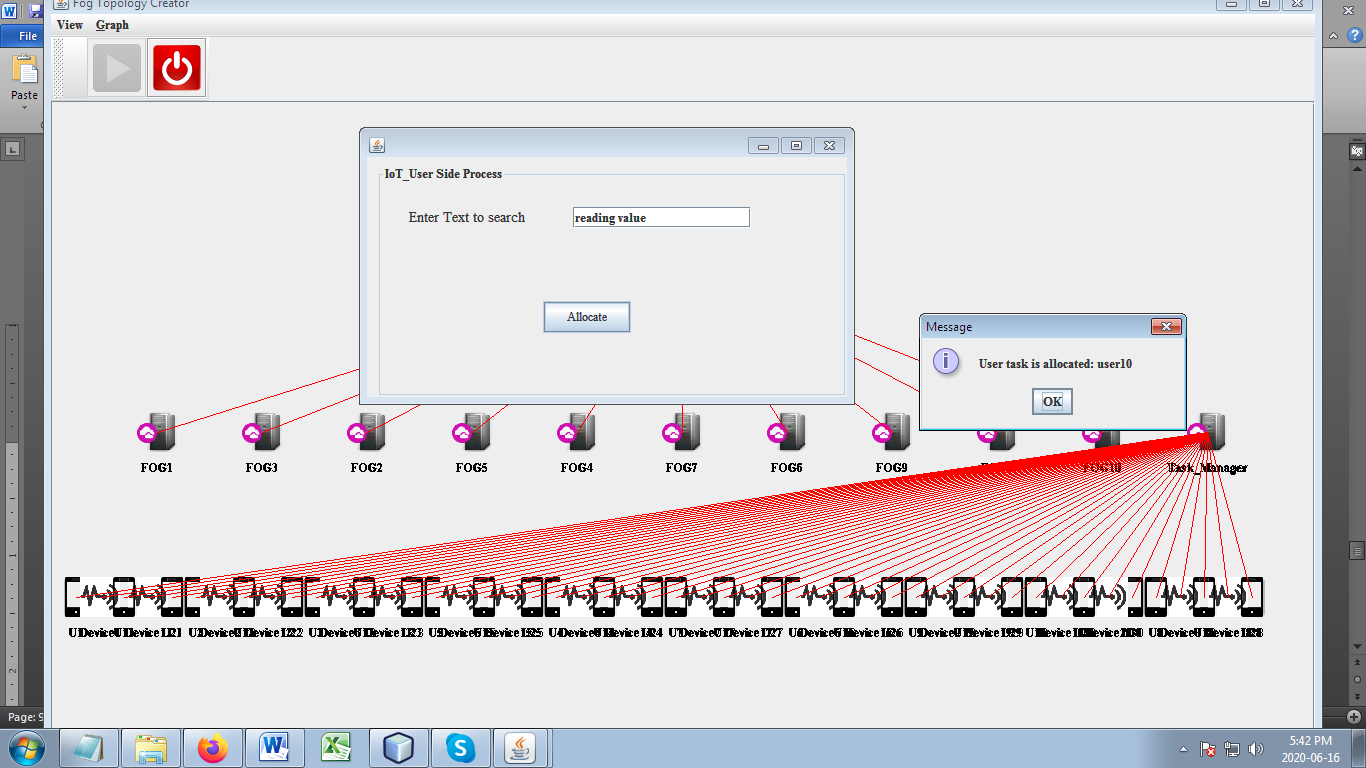


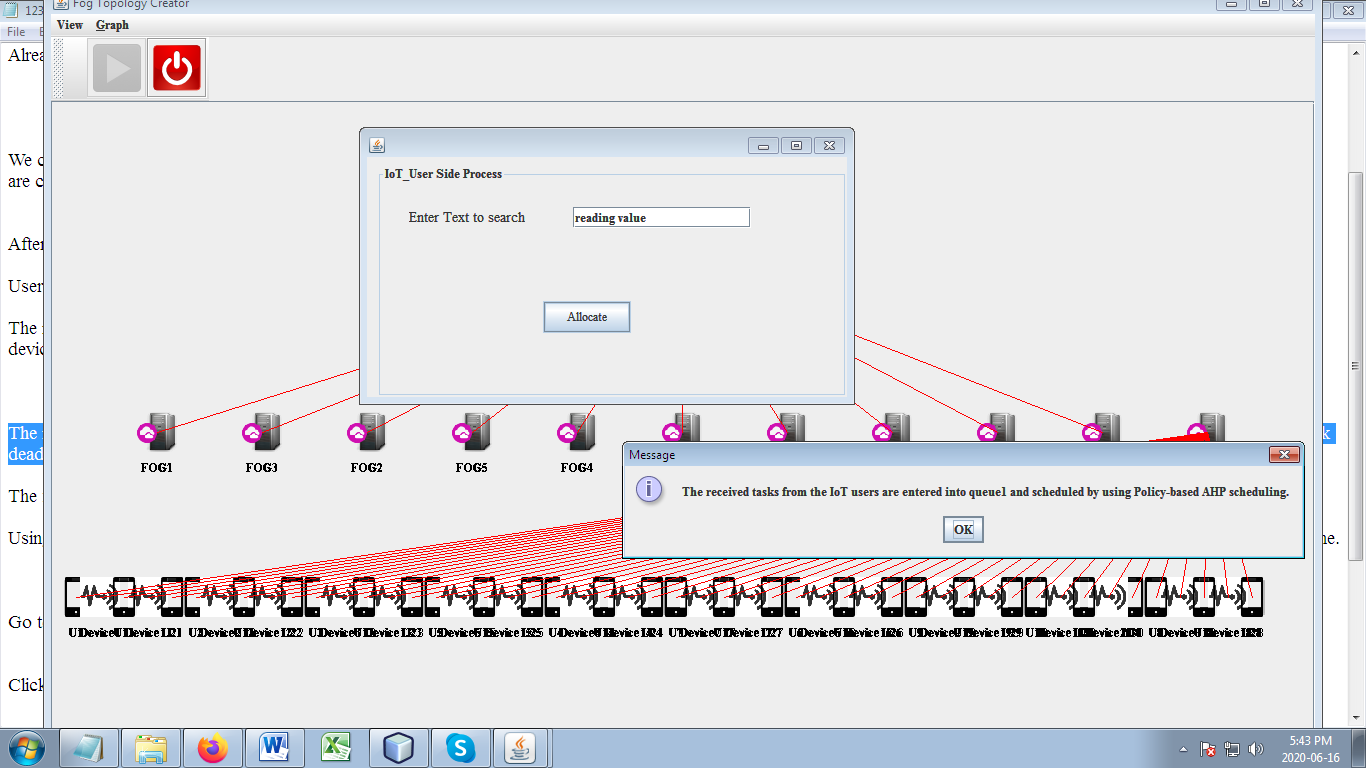


IoT User login

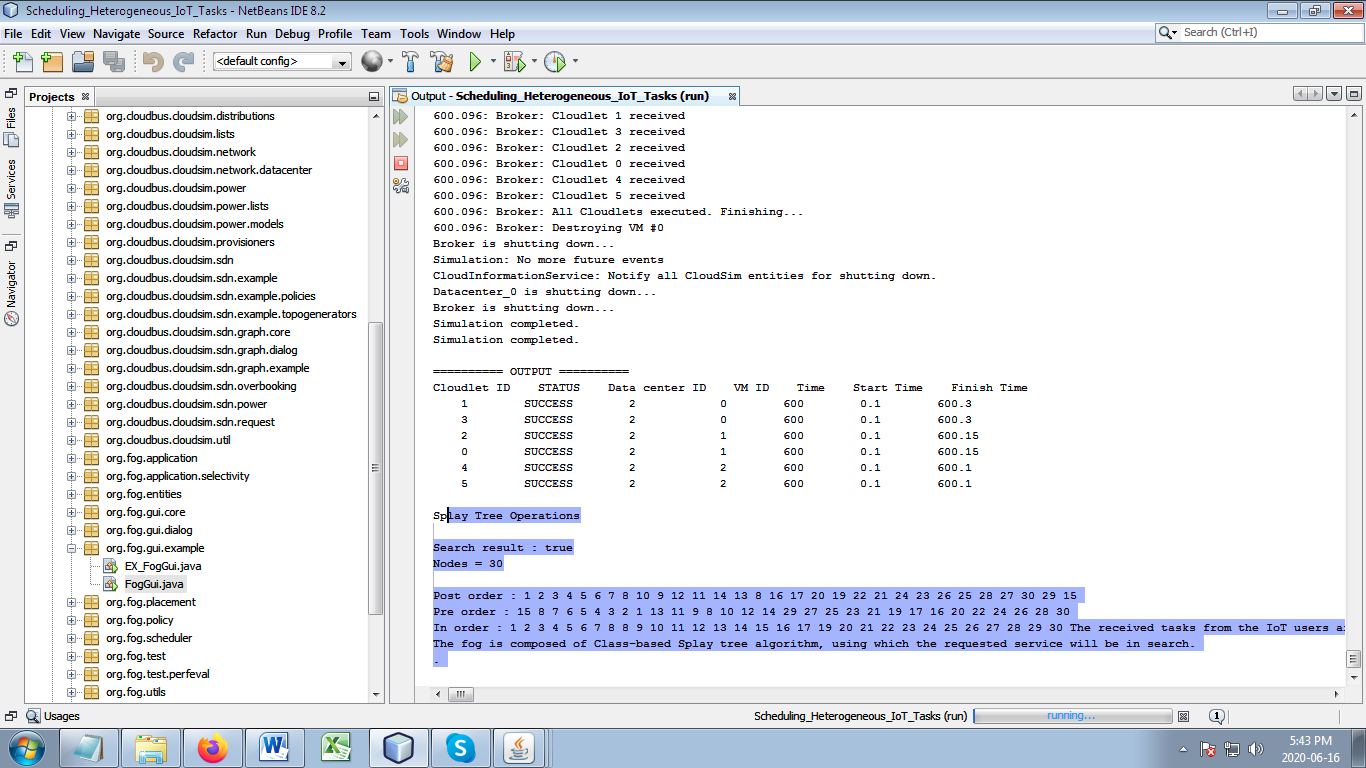


The received tasks from the IoT users are entered into queue1 and scheduled by using Policy-based AHP scheduling. Then it is assigned to the nearest fog. It is based on the task deadline, QoS, and response time.

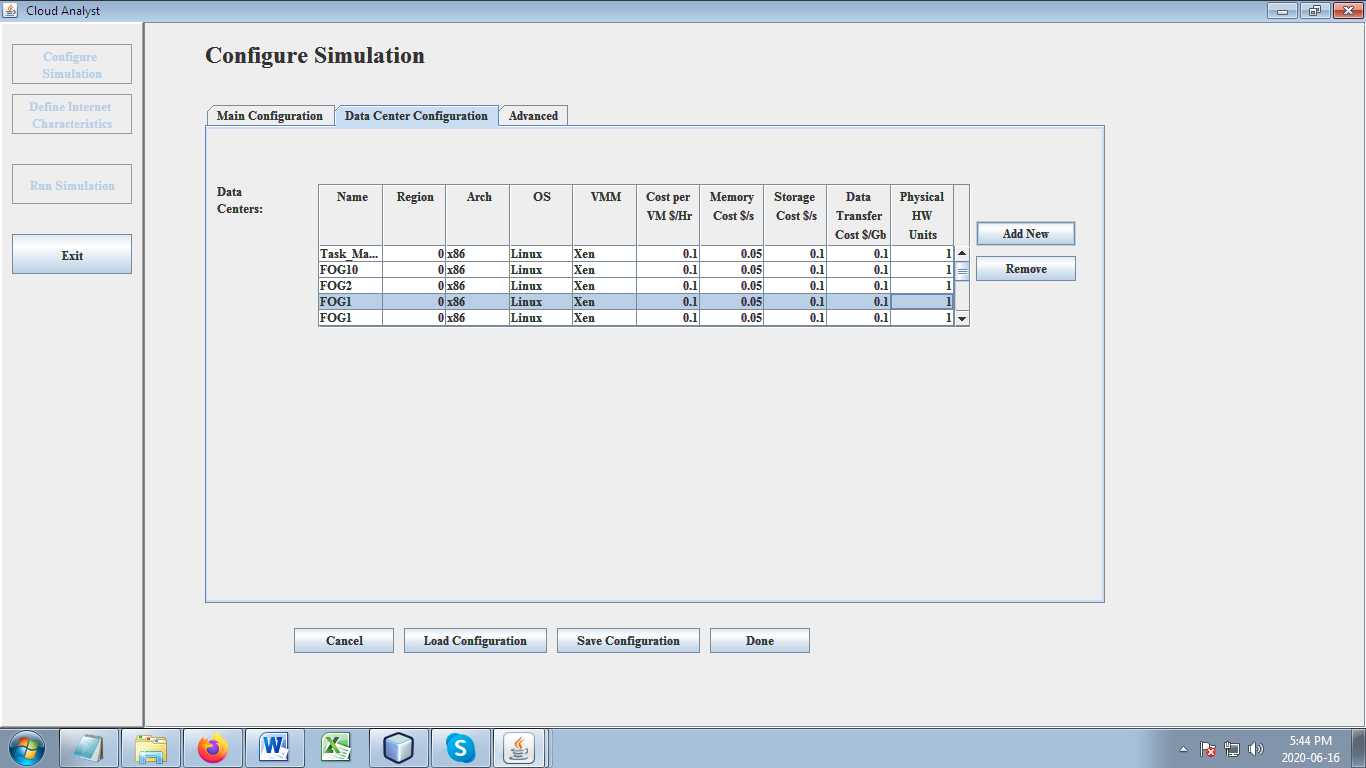


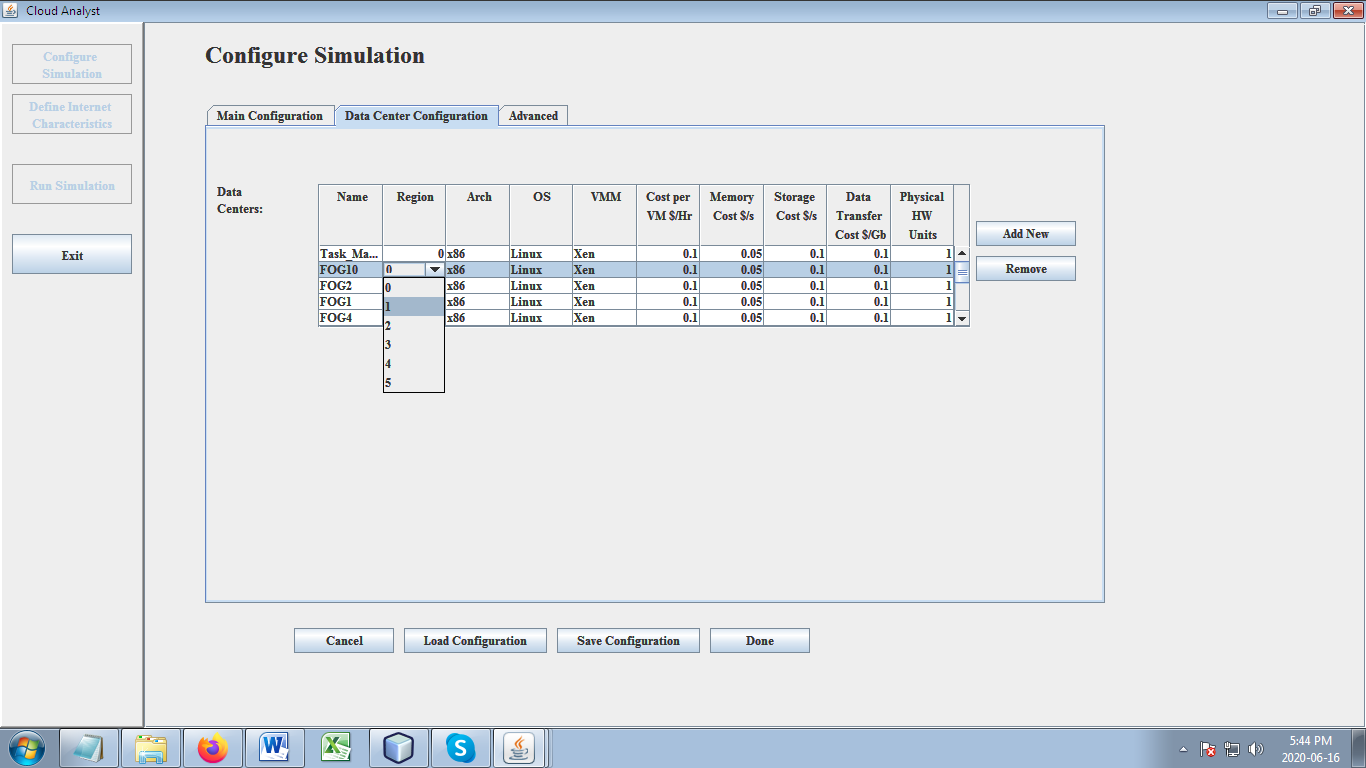


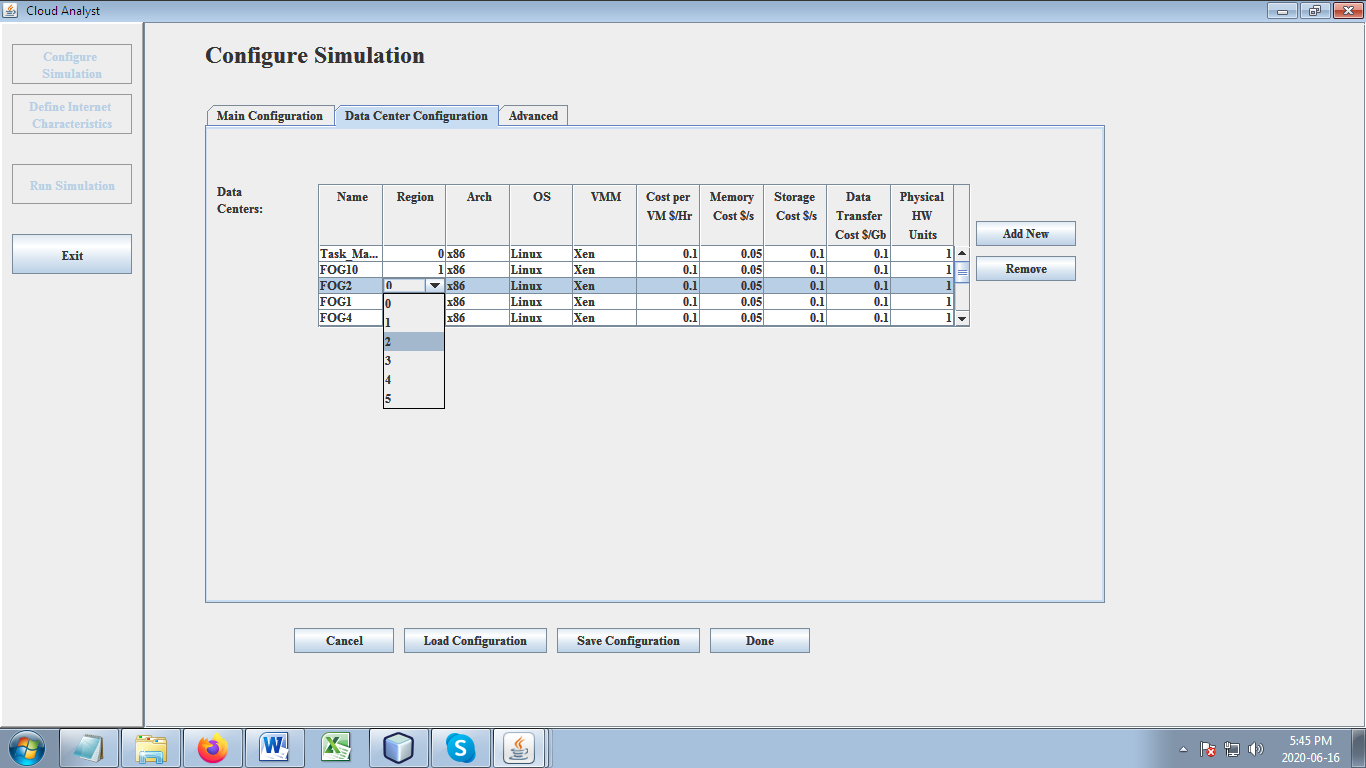
The fog is composed of Class-based Splay tree algorithm, using which the requested service will be in search.

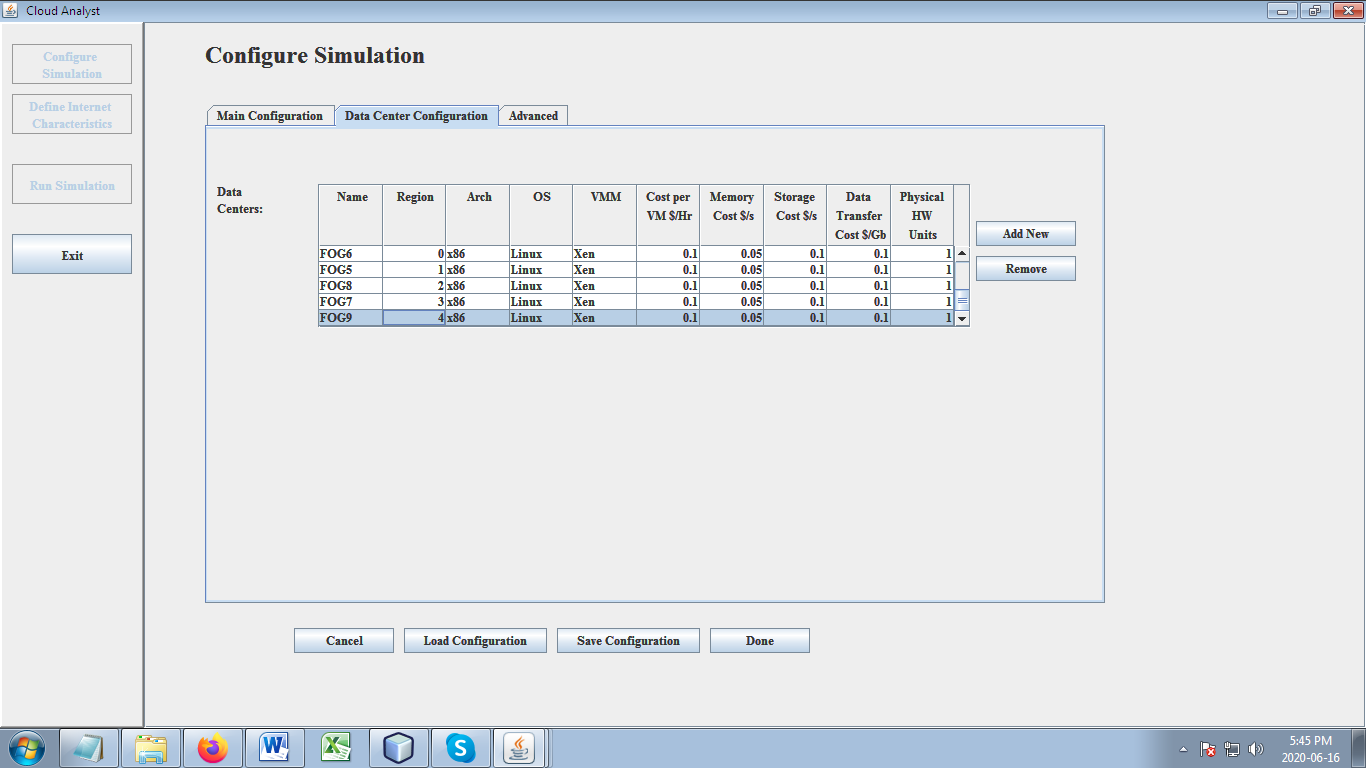


Go to the configure simulation, Add device, users,task manager, fog and set region in Main Configuration.

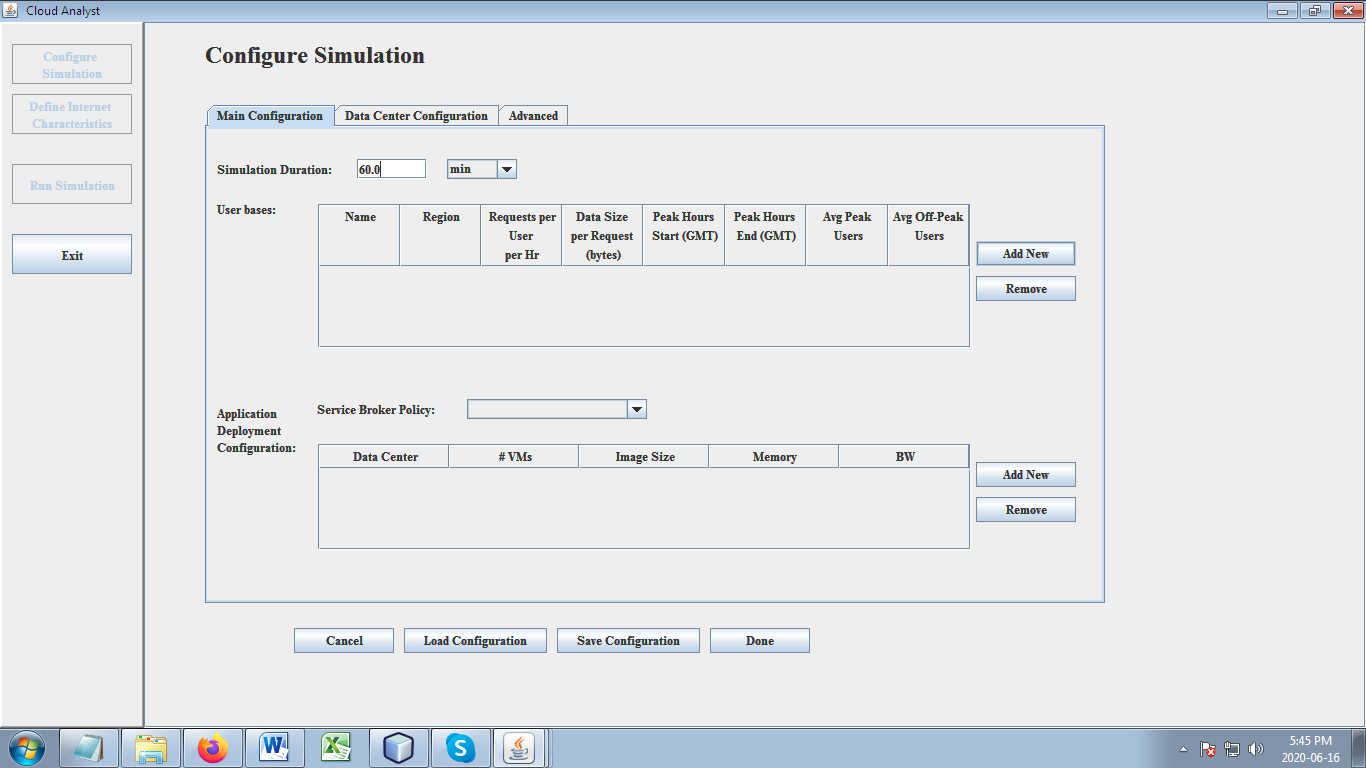


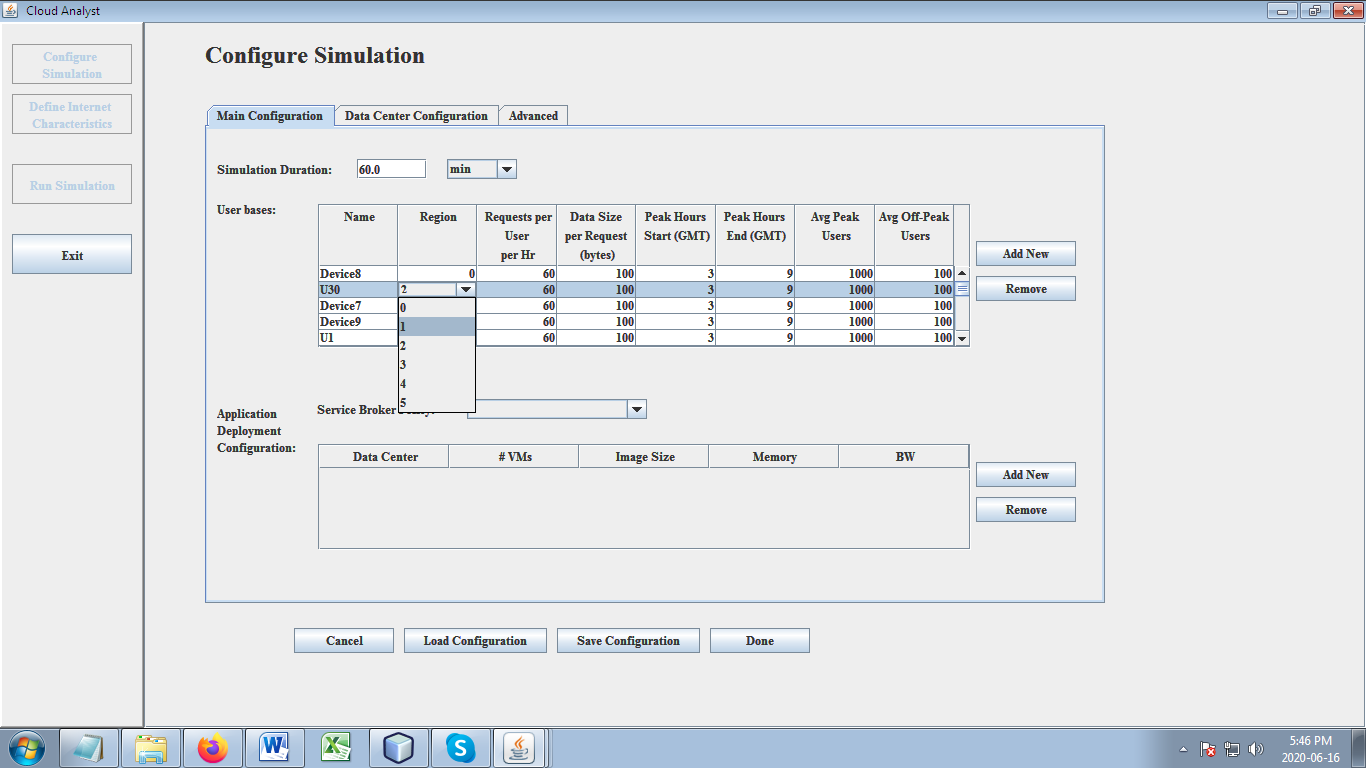


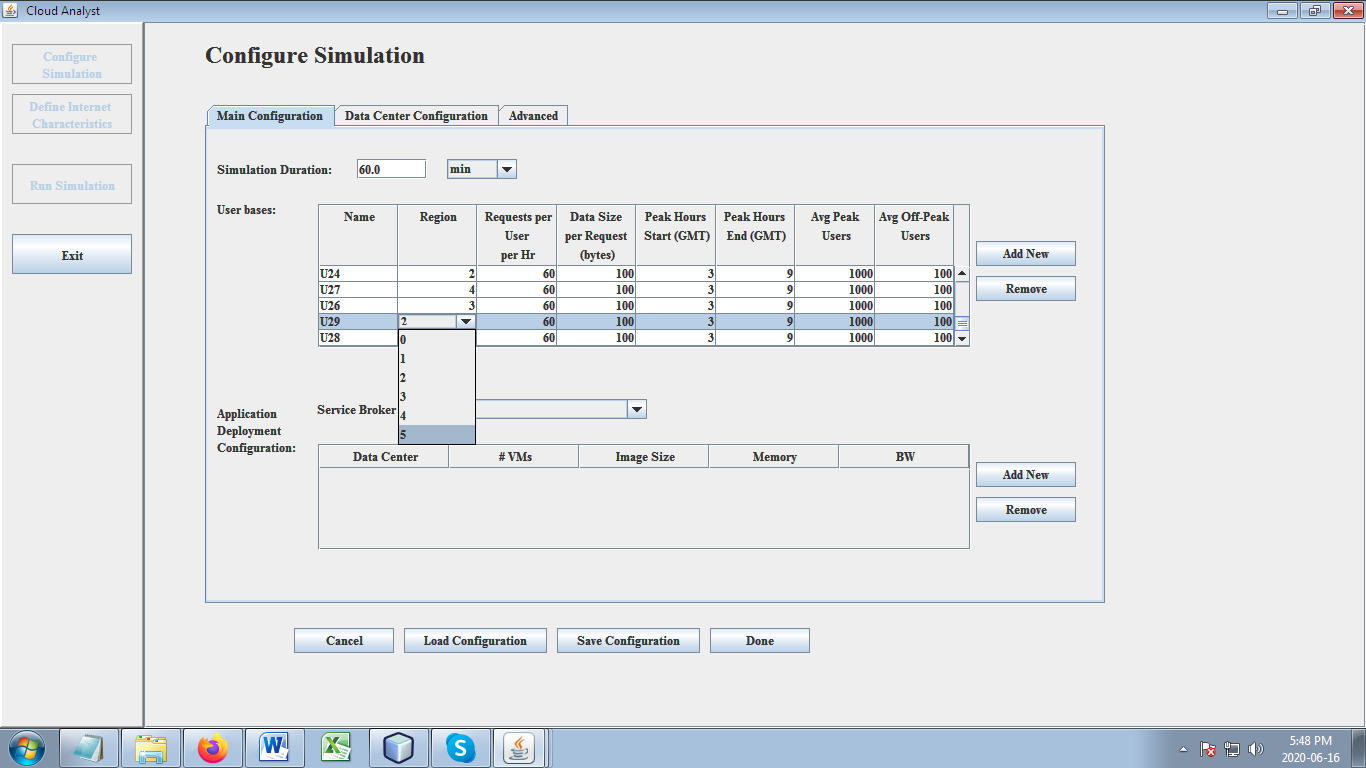




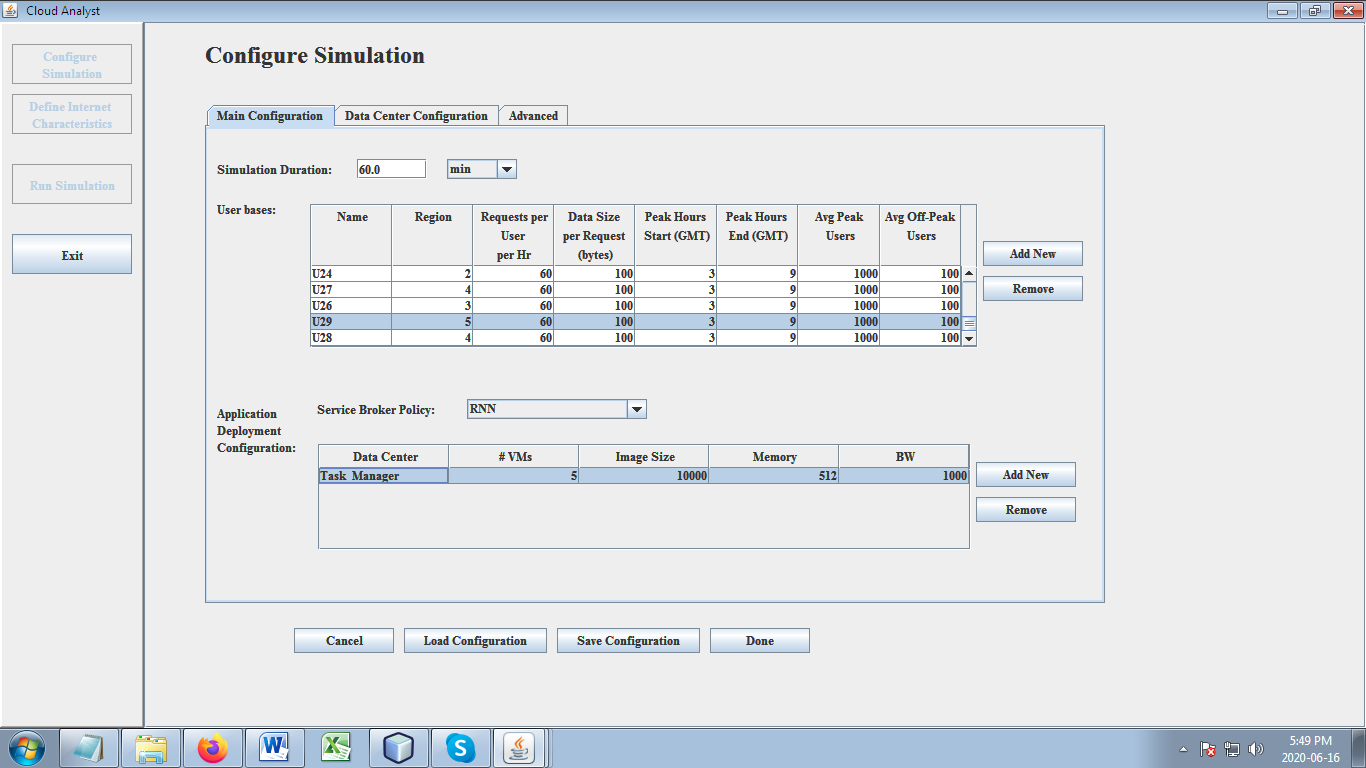
Click Add new button in user bases



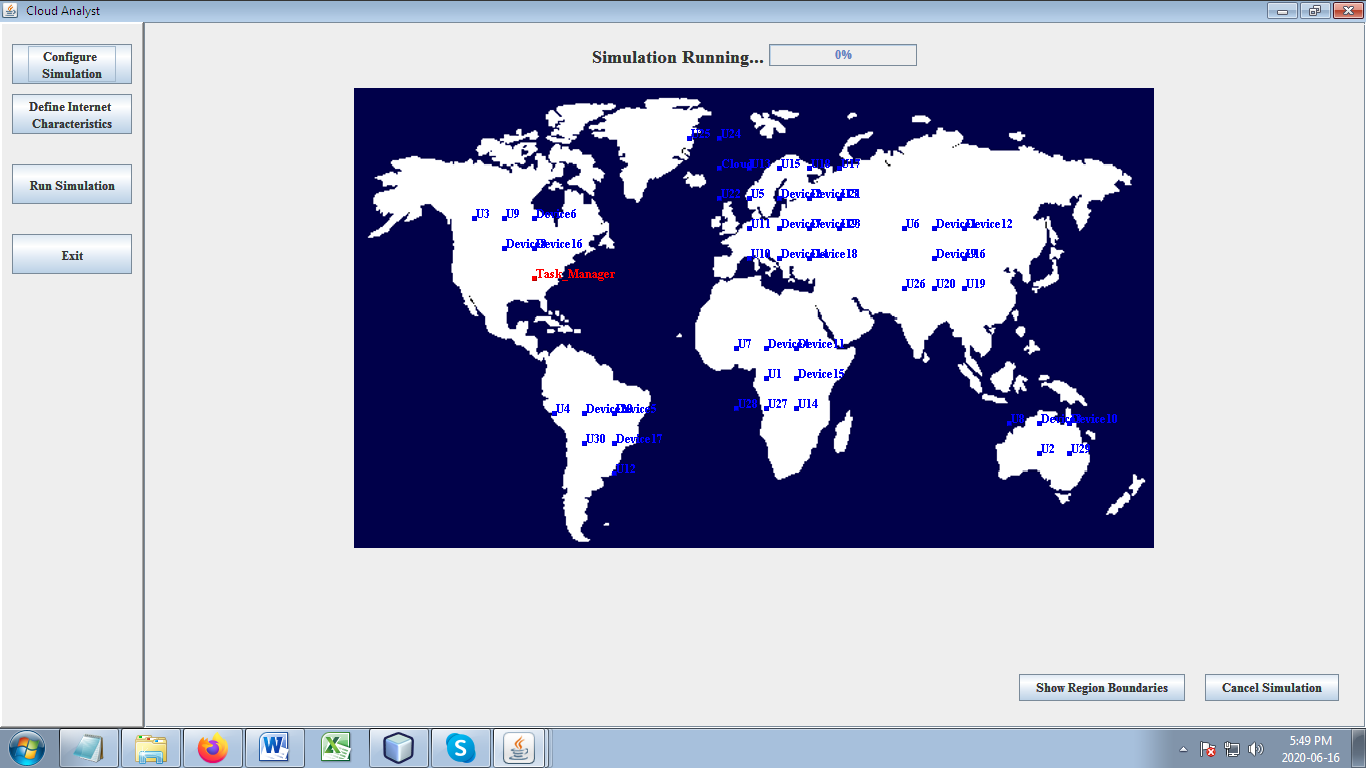




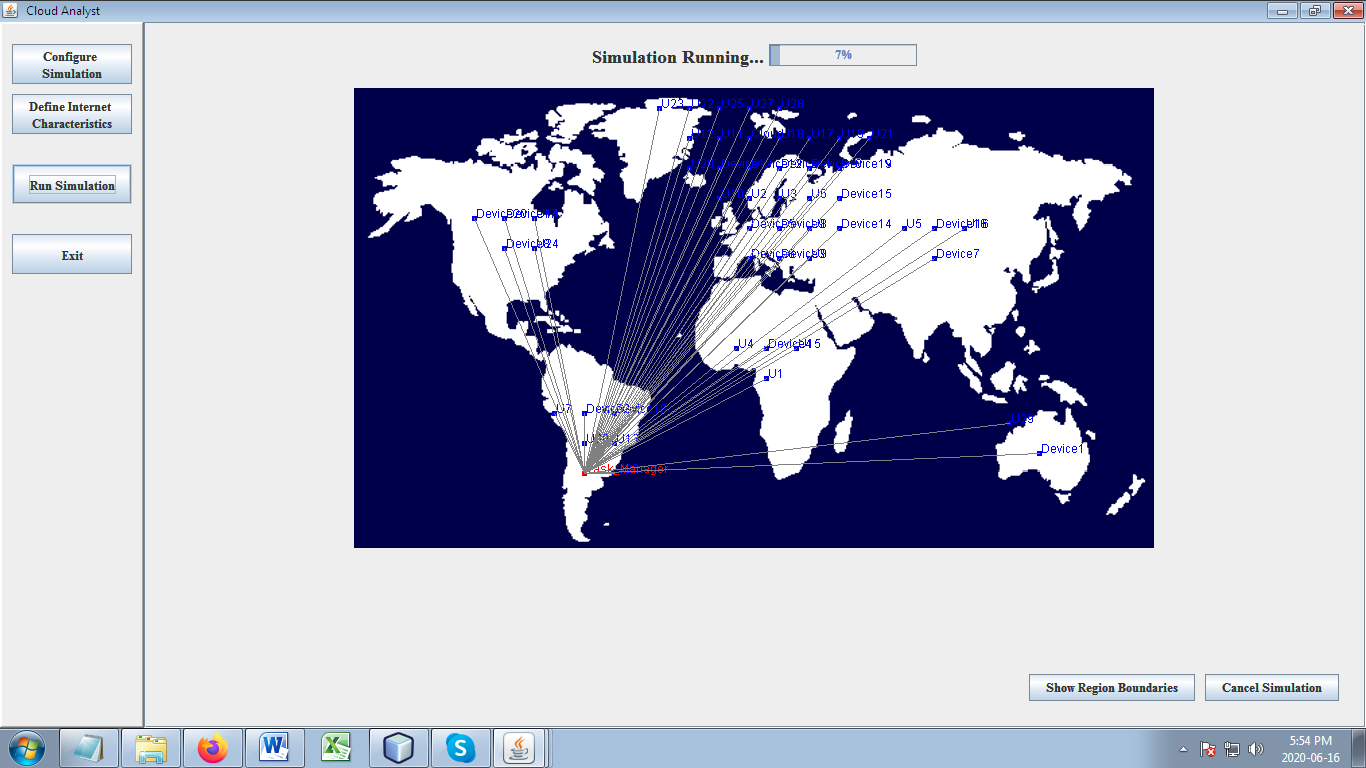
Then the cloud collected heterogeneous data is formed into ontology from the ontology of each class a dependency graph is constructed as per the semantic similarity using Switchable RNN Ontology Matching. The threshold defines the number of arrived search tasks. If the number of requests is beyond the threshold then RNN activates GRU otherwise LSTM will be used.

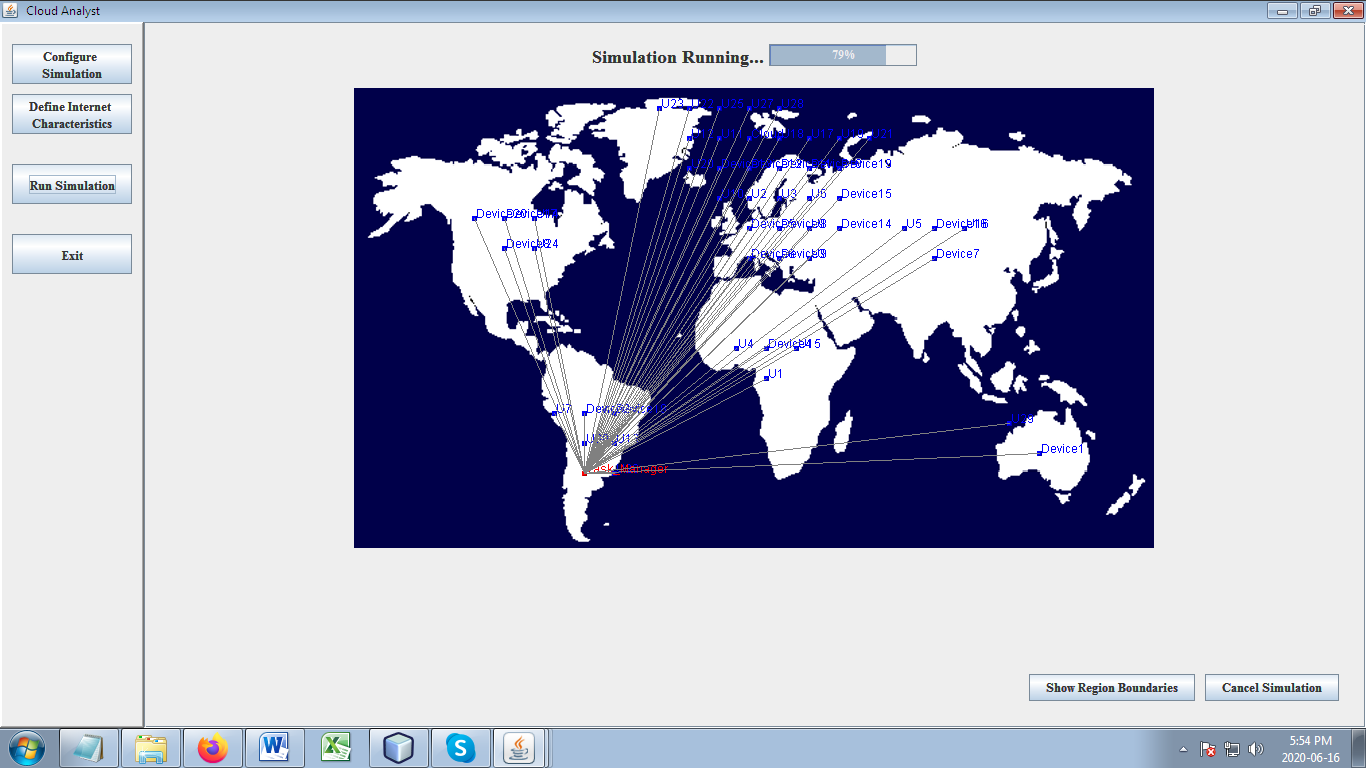


Click on done button



Click on Run Simulation





The overall system is evaluated in terms of following performance metrics,

Number of Tasks with respect to Response time

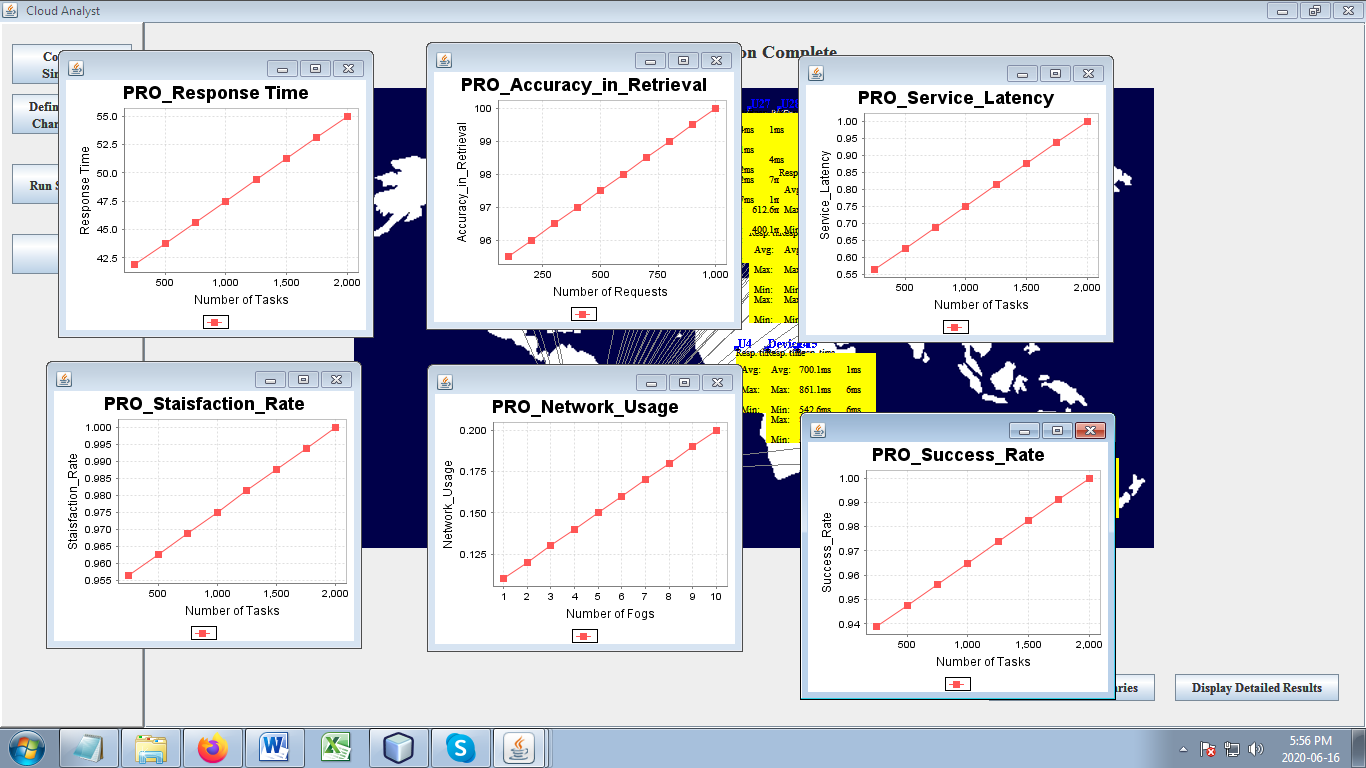
Number of Tasks with respect to Success rate

Number of Tasks with respect to Satisfaction rate

Number of Tasks with respect to Service latency

Number of fogs with respect to Network Usage

Number of Requests with respect to Accuracy in retrieval,..

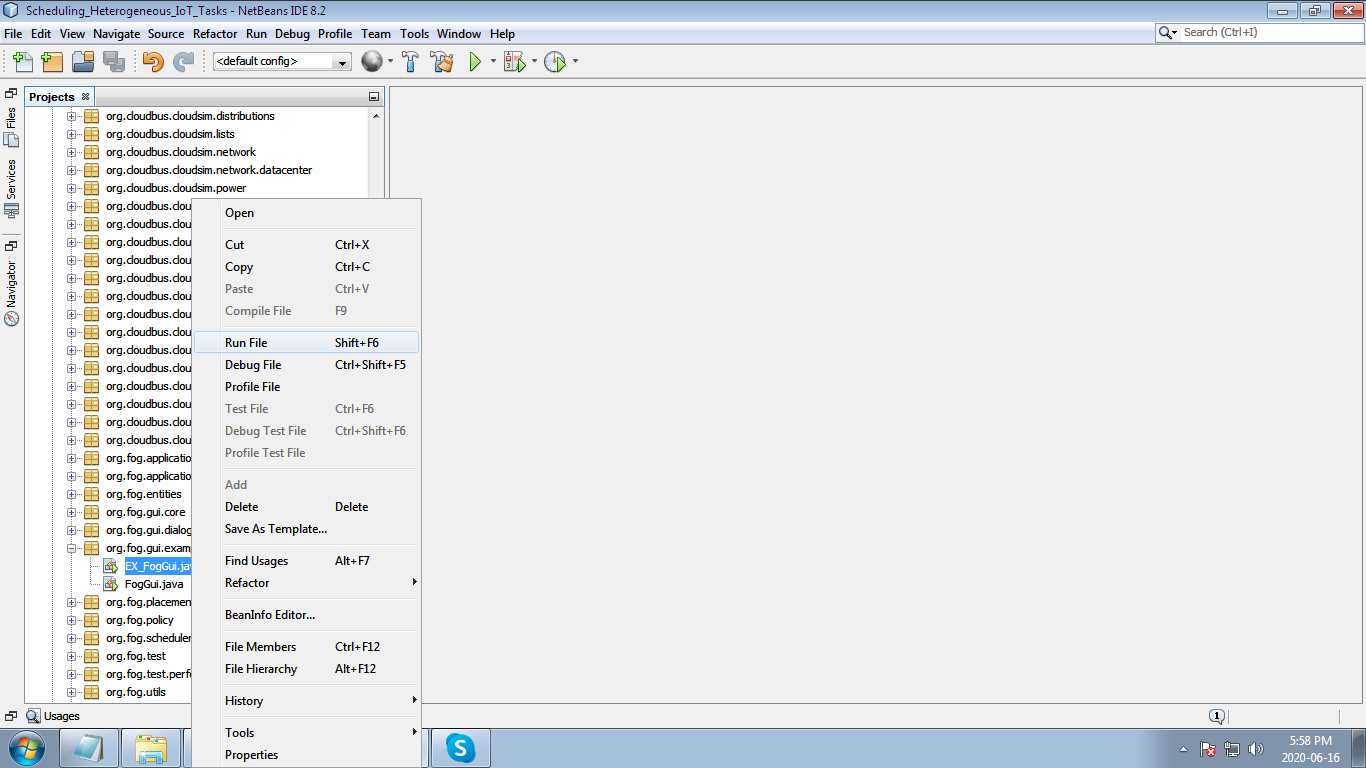


**Existing:** An efficient indexing for Internet of Things massive data based on cloud-fog computing.

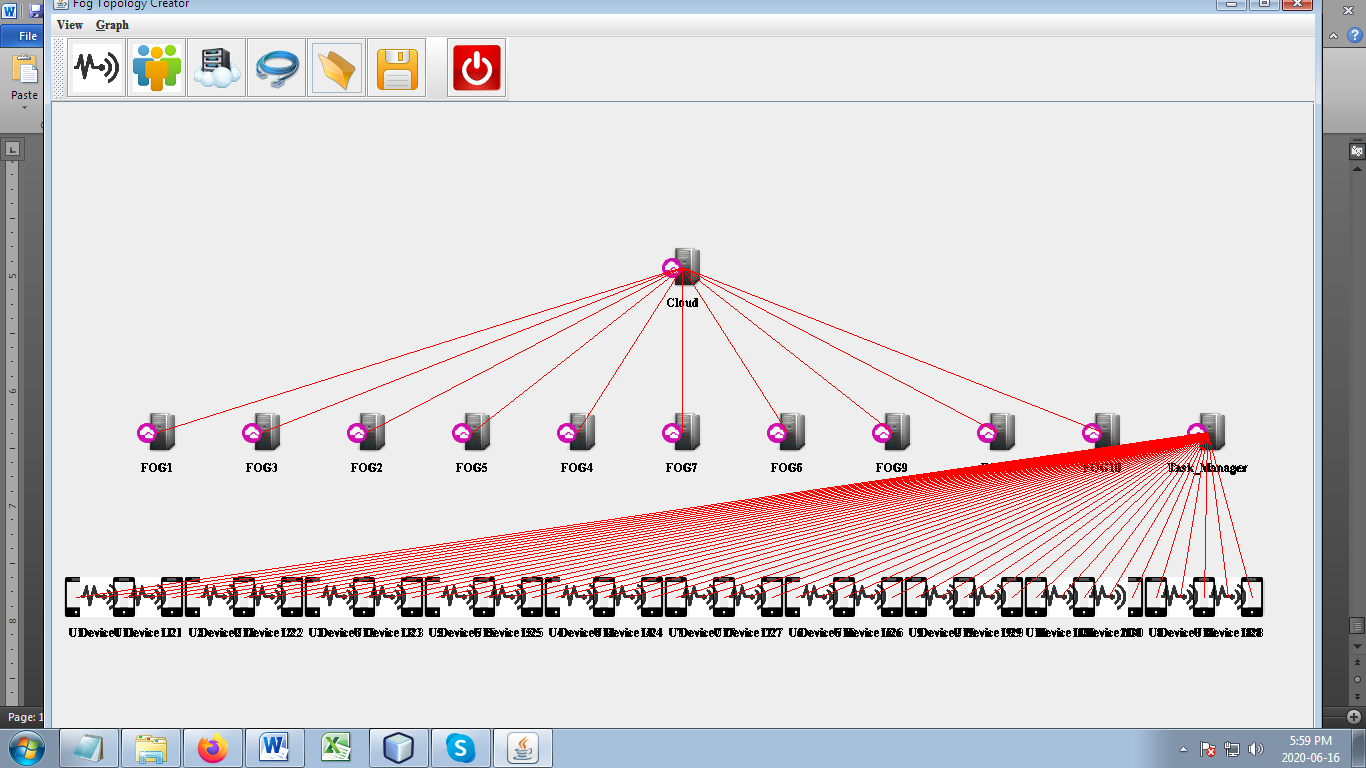
Choose Source Packages -> org.fog.gui.example

Right Click on EX\_FogGUI.java under org.fog.gui.example package

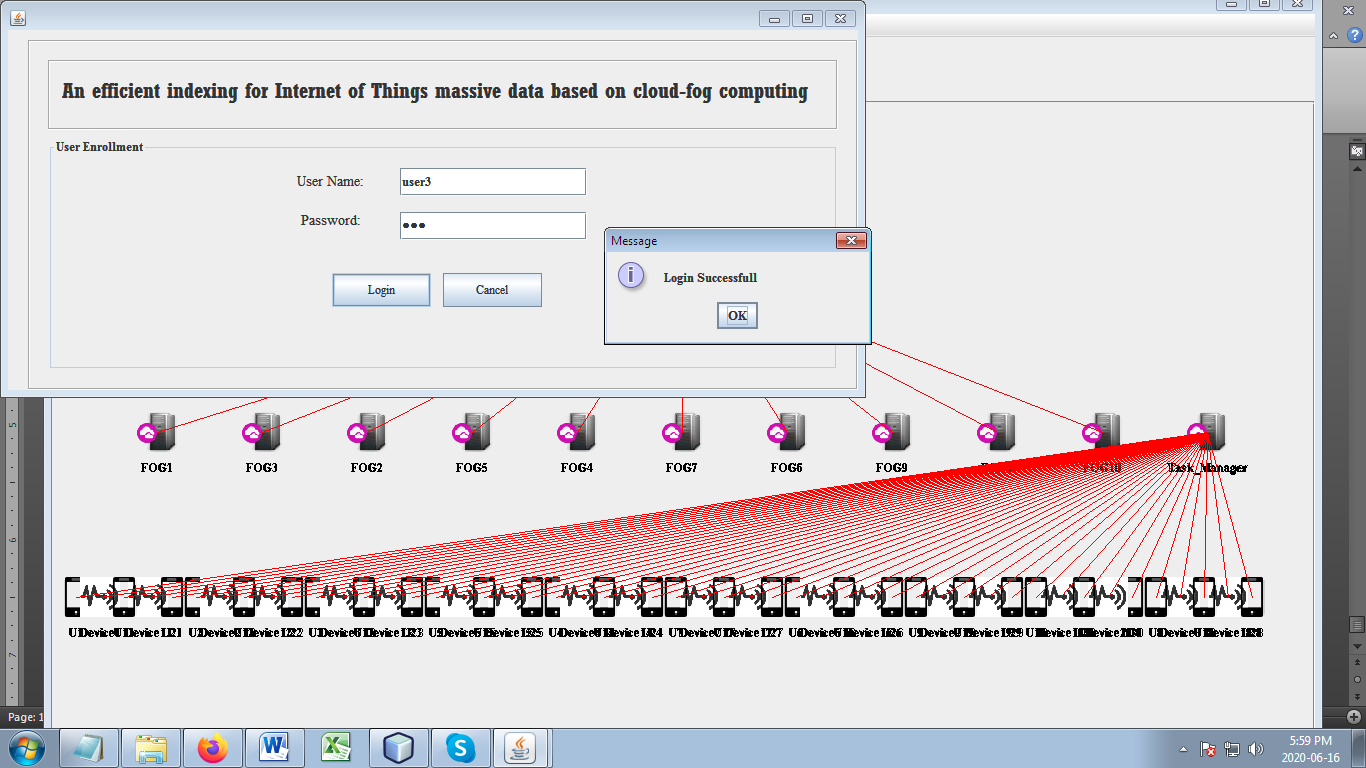
Then Choose “Run File” to run your project and Import the topology

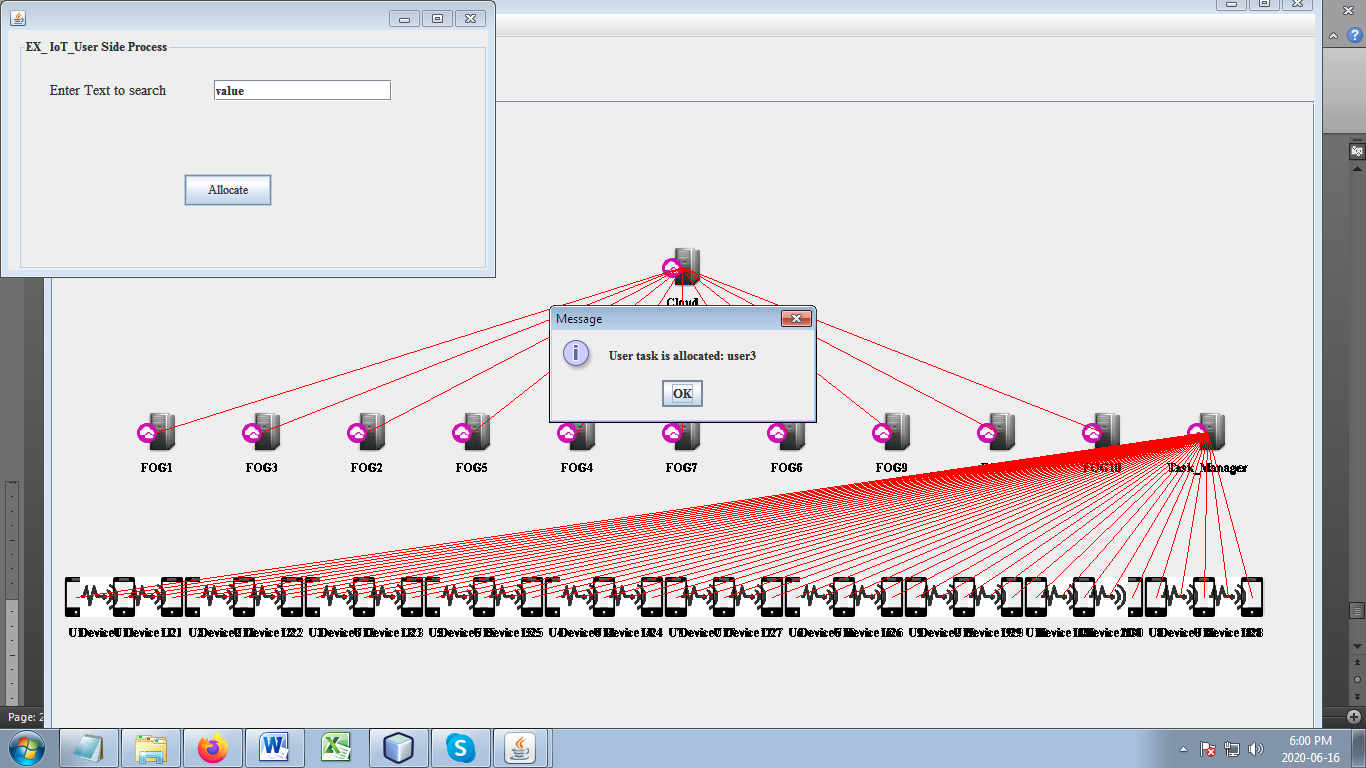


Import the topology

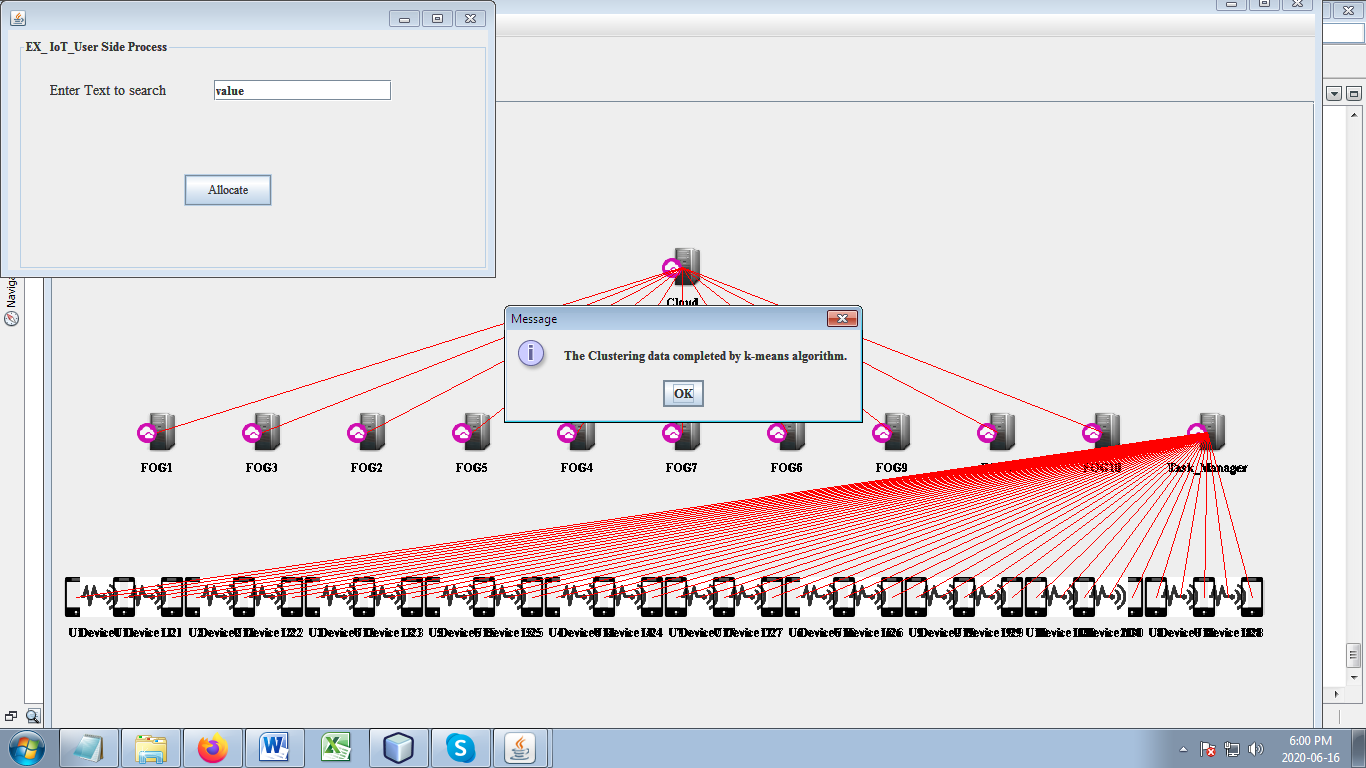


Execution 🡪 user login

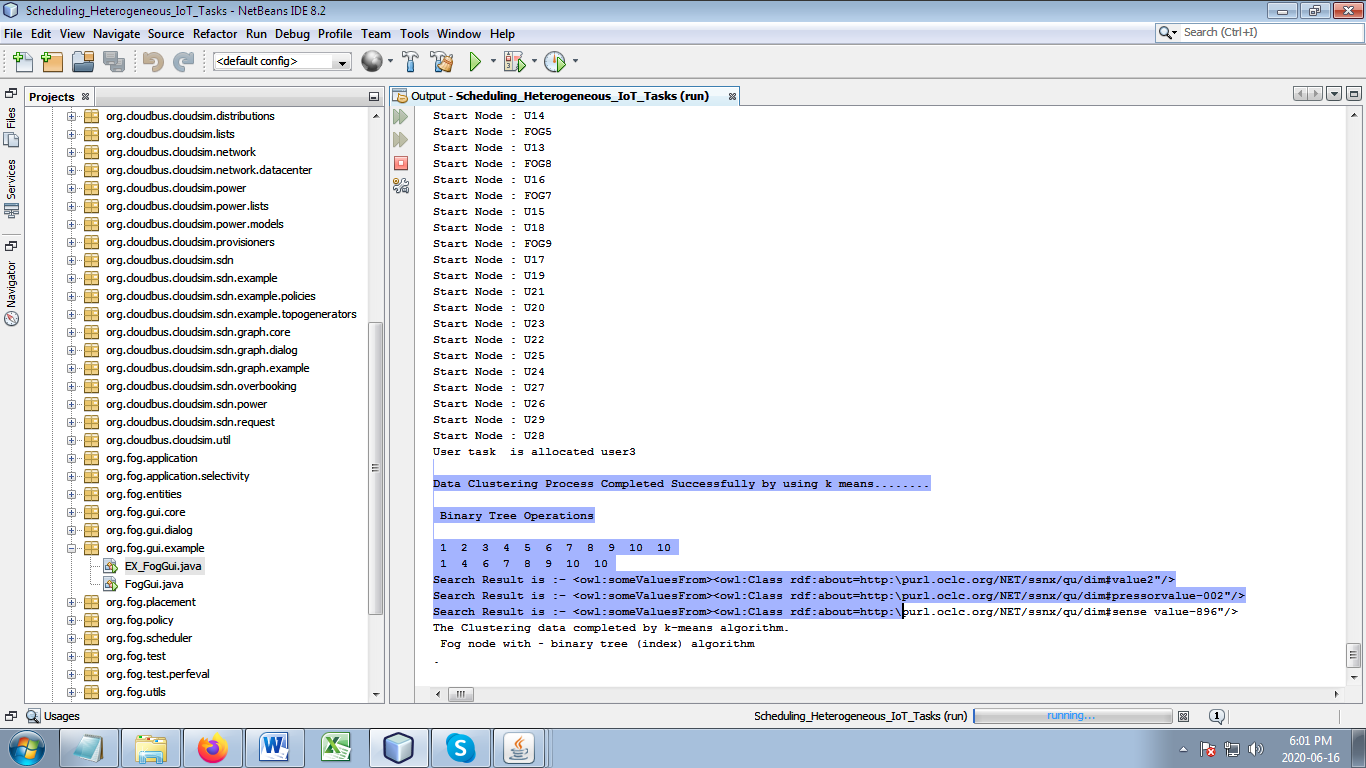


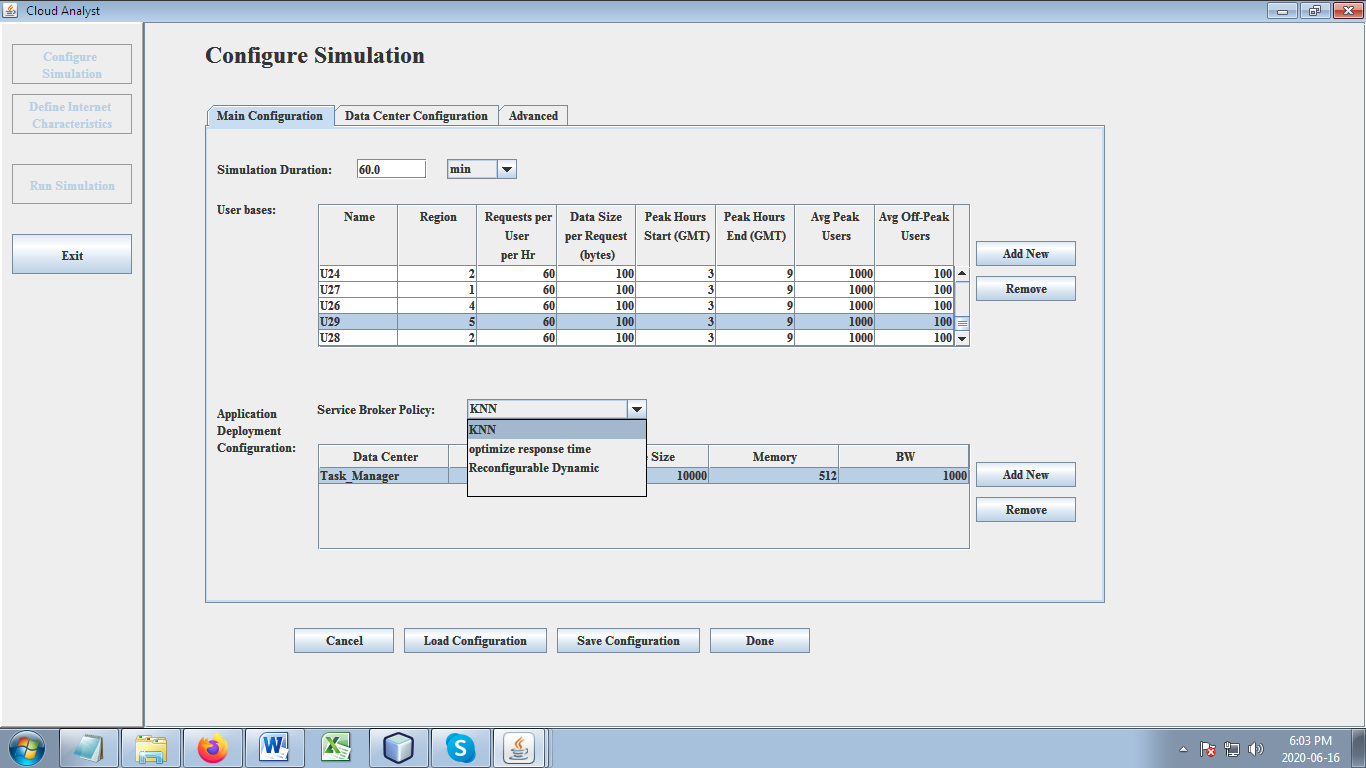


Make a data clustering using k means

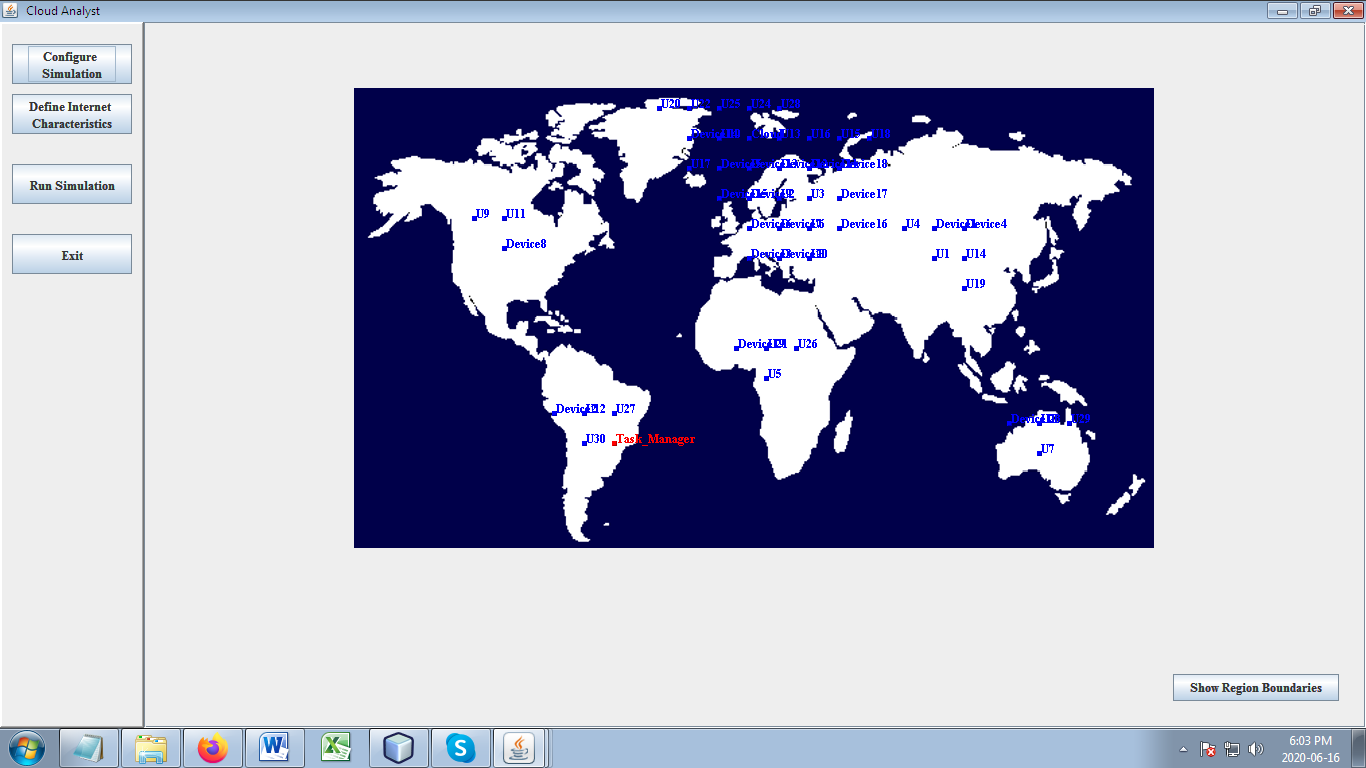


Fog constructs the tree using binary tree

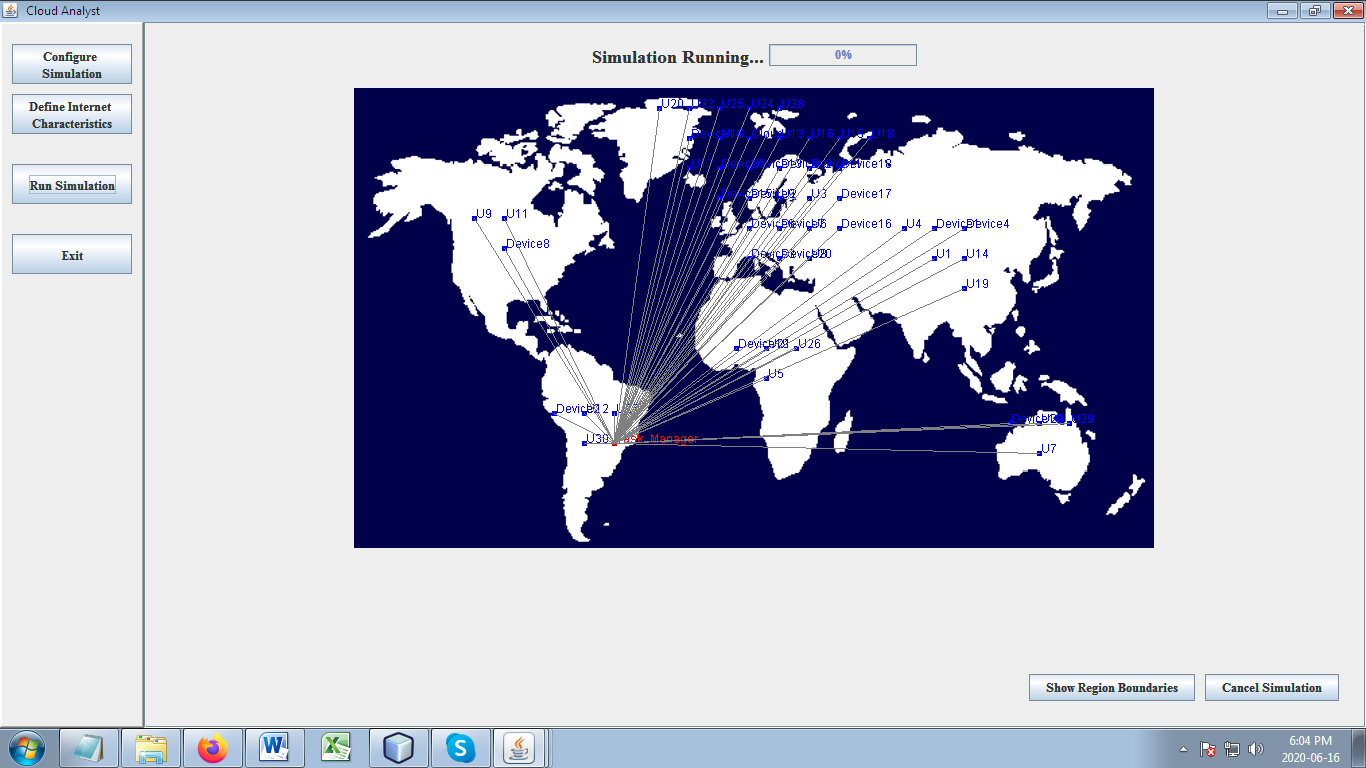




Configured network



Run the simulation





The overall system is evaluated in terms of following performance metrics, Number of Tasks with respect to Response time, Number of Tasks with respect to Success rate, Number of Tasks with respect to Satisfaction rate, Number of Tasks with respect to Service latency, Number of fogs with respect to Network Usage and Number of Requests with respect to Accuracy in retrieval,..

