National College of Ireland

# Configuration Manual

MSc Research Project
Cloud Computing

Satya Venkata Dinesh Kumar Vetcha
Student ID: 20238304

School of Computing
National College of Ireland

Supervisor: Sean Heeney

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** …….Satya Venkata Dinesh Kumar Vetcha

**Student ID:** ……20238304……..……

**Programme:** ……Cloud Computing…… **Year:** ……2019……..

**Module:** ……MSc Research Project…….……

**Lecturer:** ……Sean Heeney…….……

**Submission Due Date:** ……19/09/2022…….……

**Project Title:** ……Computation Based For Improving the Quality of Services

**Word Count:** ……1222…… **Page Count:** ……13 …….……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** ……

**Date:** ……19/09/2022……

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Configuration Manual

Satya Venkata Dinesh Kumar Vetcha
Student ID: 20238304

# 1 Introduction

This configuration manual contains the procedure for implementing a computation-based load balancing algorithm. It also includes the overall setup for installing the appropriate tools for the research project. This configuration manual helps to guide researchers and academic students and will give an deeper understanding of the proposed method carried out in implementing the project.

# 2 Required Software Tools

The Software tools that are used for the implementation of this project:

- Eclipse IDE:- This tool is used for the implementation of java code for this research project.
- IFogSim Simulator:- It is an extended version of the CloudSim simulator and contains the framework.
- Cloud Analyst Tool:- This is also an extended version of the CloudSim simulator. It is a GUI tool. It consists of default algorithms and service broker policies for the configuration of the proposed algorithm.
- JDK 17.0.4 version:- It consists of Java libraries that are required for running the suggested algorithm.

# 3 Hardware Requirements

- Operating system:- Windows/ Mac/ Linux ( We can use any OS).
- RAM:- Minimum 4GB / 8GB or more than needed.
- Processor:- Any Processor from the Intel Core is suitable.

# 4 Step by Step Software Installation

The steps for the installation of software and tools are shown below.

## 4.1 JDK Installation

- Java Development Kit(JDK)
- Download JDK 17.0.4 version from the specified Link [1].

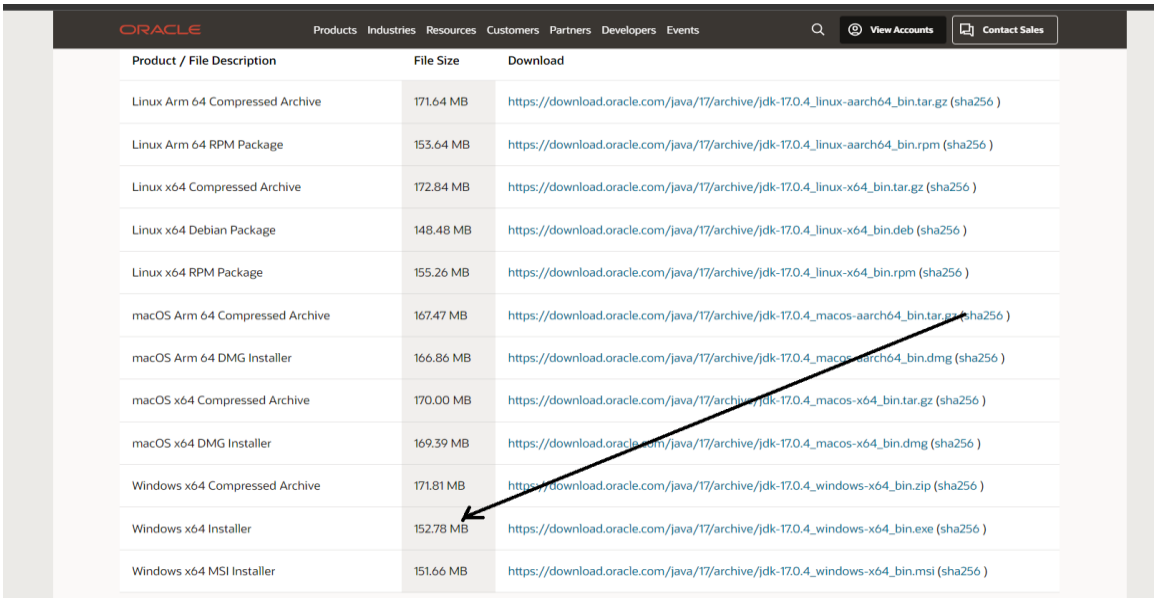



Figure 1:- Download Java Development Kit (JDK)

- Install the JDK 17.0.4 Version into the system

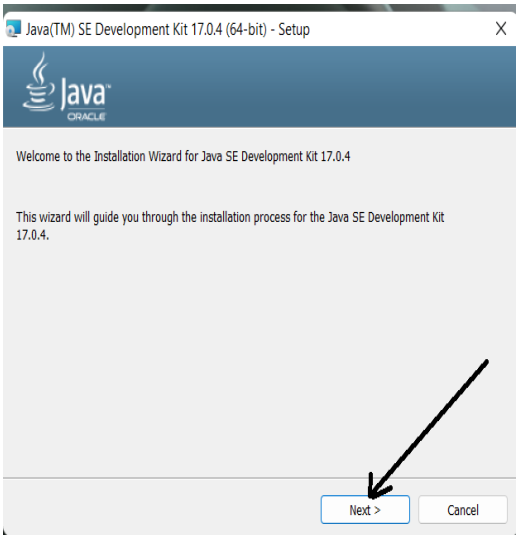



Figure 2: JDK Installation

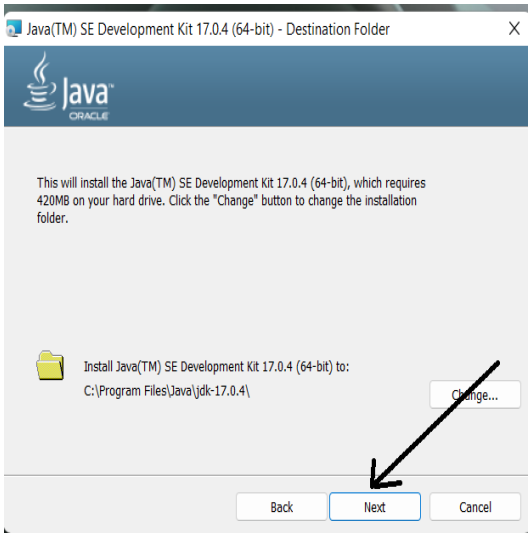



Figure 3:- Installation of JDK

## 4.2 Eclipse IDE Installation

- Download the Eclipse Integrated Development Environment 2022-06 from the given link [2]

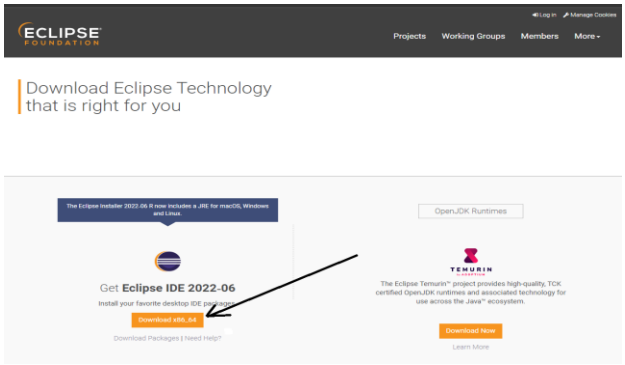



Figure 4:- Download Eclipse

- Install the download Eclipse IDE 2022-06 in system

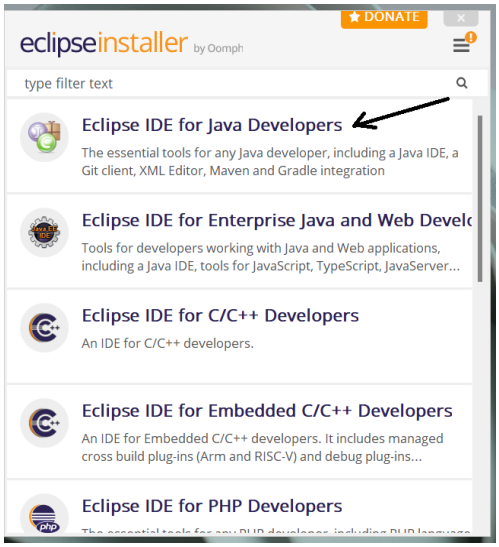



Figure 5:- Install Eclipse


Figure 6:- Final step in Eclipse Installation

- Make Sure the java JDK version before installing Eclipse IDE

- Now open the Eclipse and create a workspace

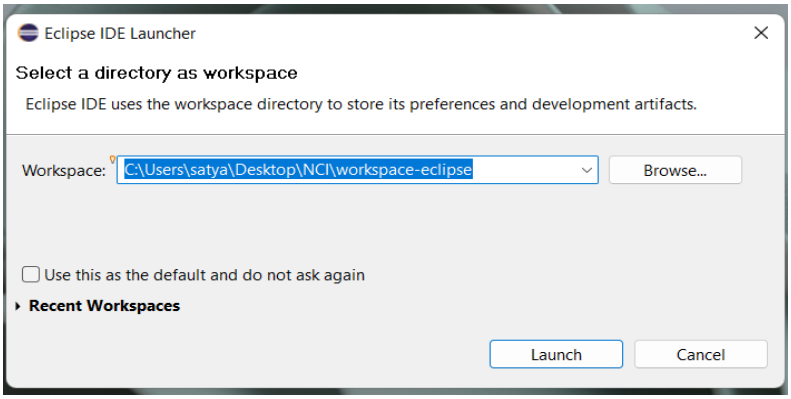



Figure 7:- Workspace Creation for Eclipse IDE

- The created workspace shouldn't be changed. Otherwise, the work done will not be available.
- Now create a new Java project in Eclipse IDE

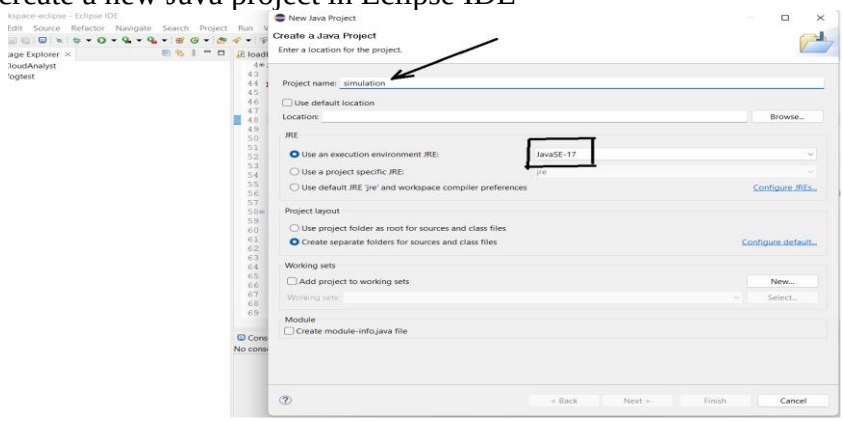



Figure 8:- The new project creation in Eclipse IDE

- Make sure the Java Version and name a project of your interest

## 4.3 Installation of iFogSim Simulator

- Download the . Zip file from the iFogSim GitHub [3].

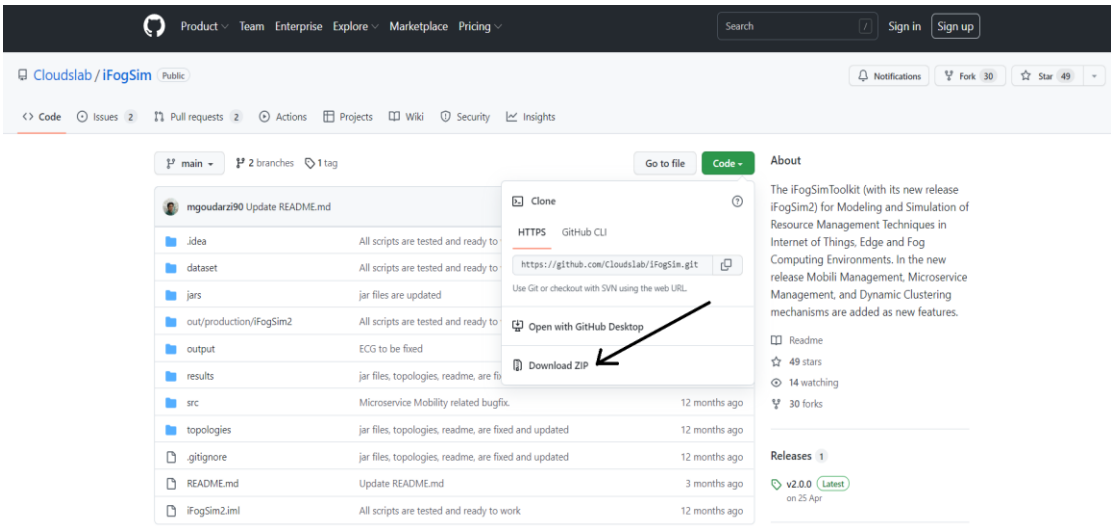



Figure 9:- Download the iFogSim file from GitHub

- Importing the downloaded file into Eclipse.

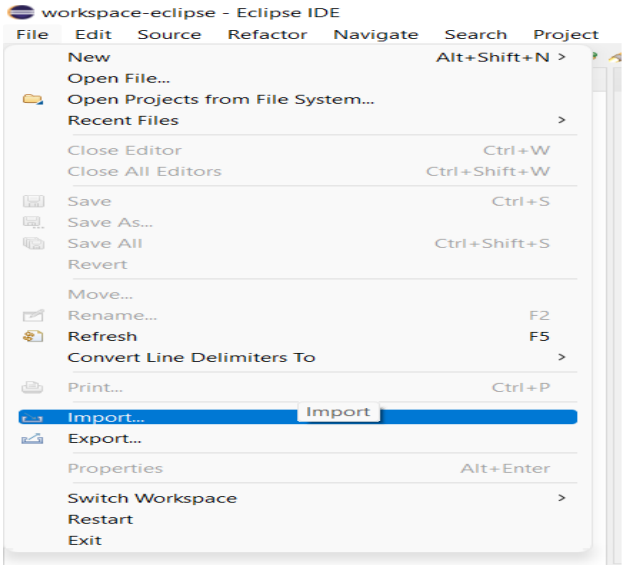



Figure 10:- Import option in Eclipse IDE

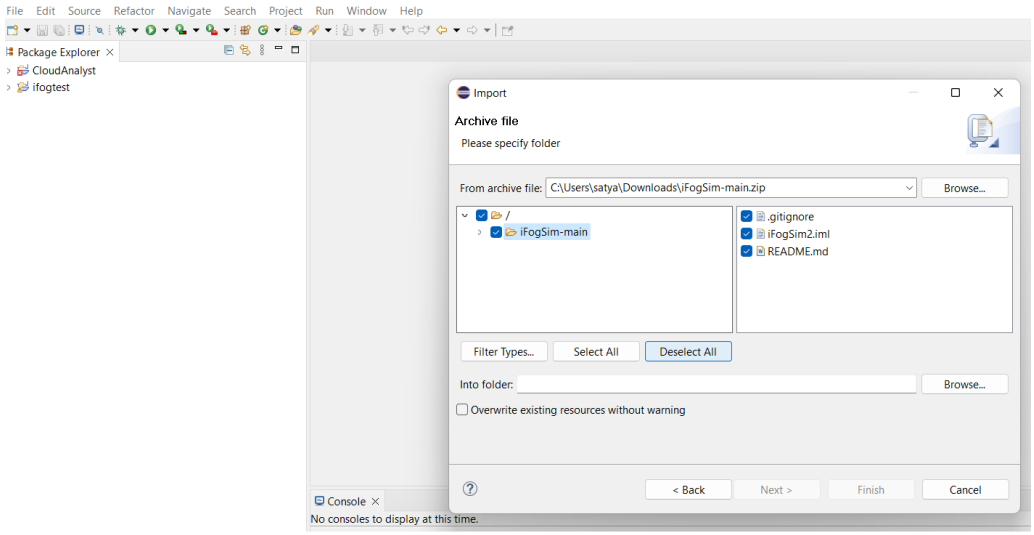



Figure 11:- Importing the iFogSim file

# 5 Development of Proposed Algorithm

The proposed method development and implementation explained in step by step.

## 5.1 Fog environment creation

- By running the fog.gui.java in Eclipse IDE, it helps for the creation of network topology for the proposed methodology.

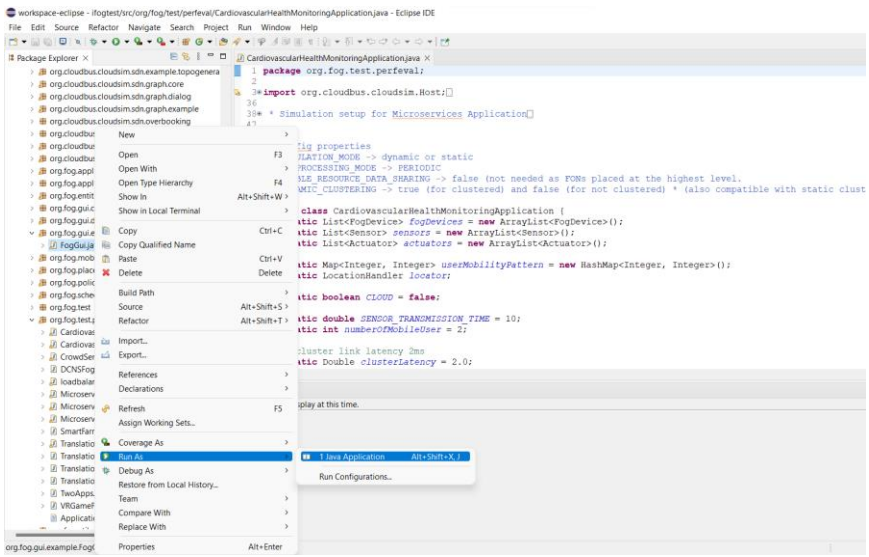



Figure 12:- Creation of the Network topology

- The topology designed

  - C :- Camera (Sensor) of Picture capture
  - Floor-Finder :- Actuator

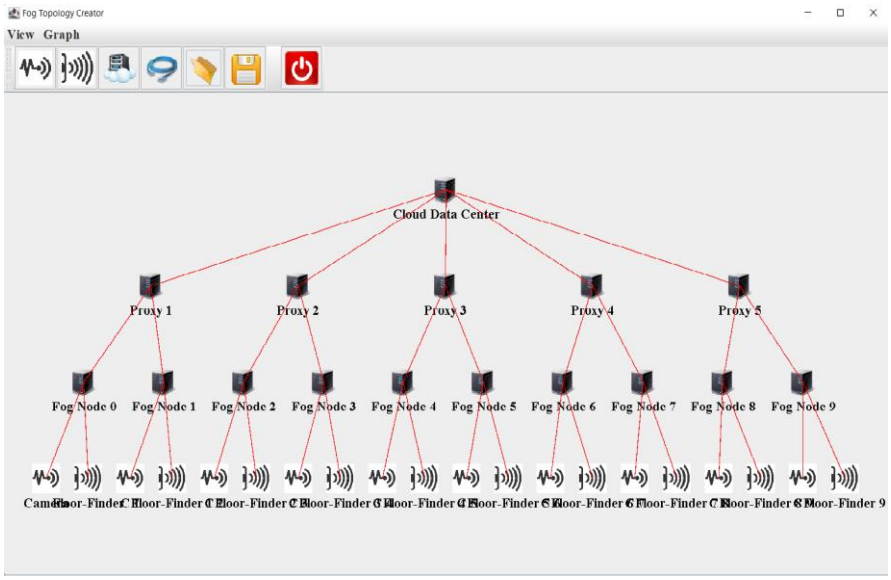
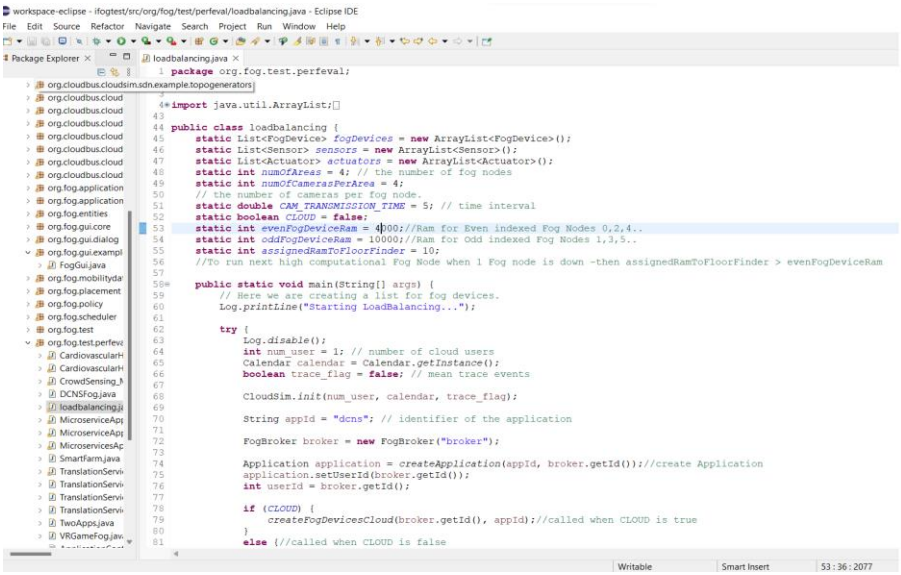



Figure 13:- Network Topology for the Proposed Method

## 5.2 Load Balancing Code Implementation and Creation

The proposed computation based load balancing algorithm code is shown in below figure 14.



```
package org.fog.test.perfeval;

import java.util.ArrayList;

public class loadbalancing {
    static List<FogDevice> fogDevices = new ArrayList<FogDevice>();
    static List<Sensor> sensors = new ArrayList<Sensor>();
    static List<Actuator> actuators = new ArrayList<Actuator>();
    static int numOfAreas = 4; // the number of fog nodes
    static int numOfCamerasPerArea = 4;
    // the number of cameras per fog node.
    static double CAM_TRANSMISSION_TIME = 5; // time interval
    static boolean CLOUD = false;
    static int evenFogDeviceRam = 4000;//Ram for Even indexed Fog Nodes 0,2,4..
    static int oddFogDeviceRam = 10000;//Ram for Odd indexed Fog Nodes 1,3,5..
    static int assignedRamToFloorFinder = 10;
    //To run next high computational Fog Node when 1 Fog node is down -then assignedRamToFloorFinder > evenFogDeviceRam

    public static void main(String[] args) {
        // Here we are creating a list for fog devices.
        Log.printLine("Starting LoadBalancing...");

        try {
            Log.disable();
            int num_user = 1; // number of cloud users
            Calendar calendar = Calendar.getInstance();
            boolean trace_flag = false; // mean trace events

            CloudSim.init(num_user, calendar, trace_flag);

            String appId = "dcns"; // identifier of the application

            FogBroker broker = new FogBroker("broker");

            Application application = createApplication(appId, broker.getId());//create Application
            application.setUserId(broker.getId());
            int userId = broker.getId();

            if (CLOUD) {
                createFogDevicesCloud(broker.getId(), appId);//called when CLOUD is true
            }
            else {//called when CLOUD is false
```

Figure 14:- Proposed Java Code Implementation

## 5.3 Network topology Simulator Output

The below Figure 15 is the simulated network topology output.

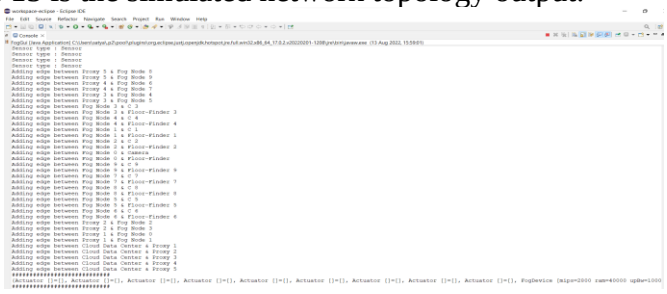


Figure 15:- Implemented Topology Output

## 5.4 Proposed Method Testing and Running

- Run the implemented code

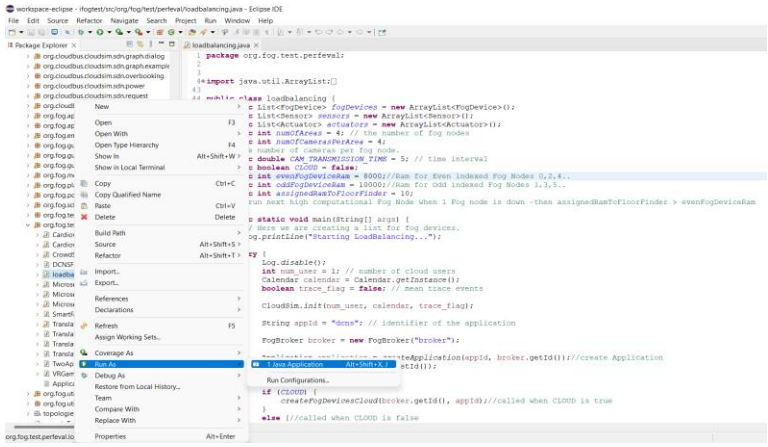


Figure 16:- Running the Implemented Code

- After running the simulation, the outputs are showed in below figure 17 and 18. The task assigned based on the computational capacity of the fog devices.

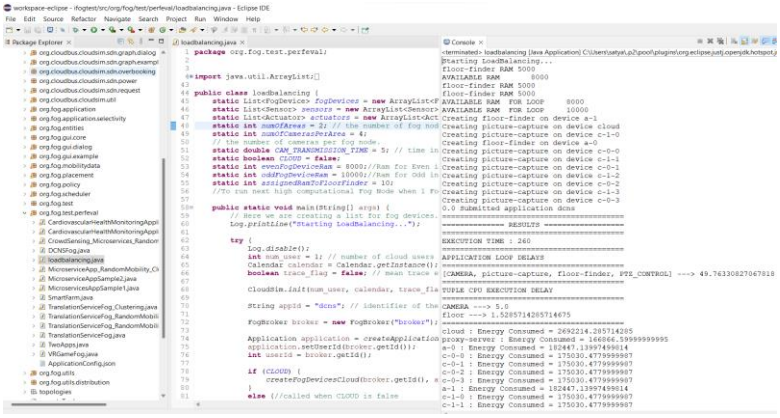


Figure 17:- The tasks assigned to nearest fog node

- In the above-plotted results in figure 17, the load on the fog node is less, and also the fog device has the capacity to perform the operation.

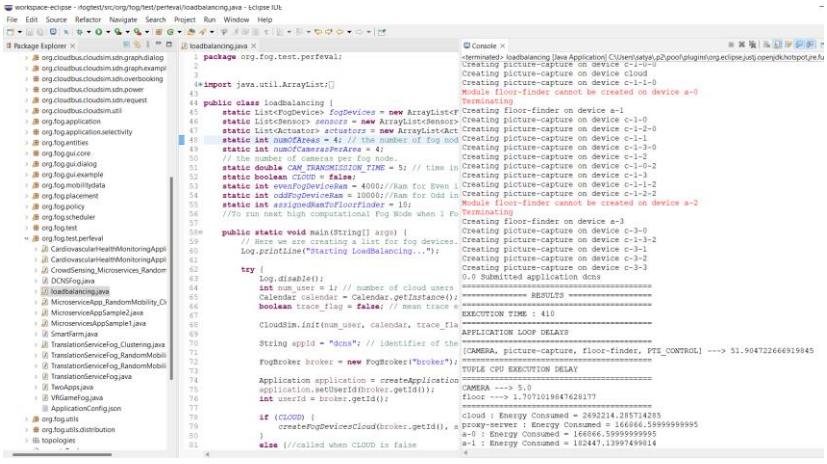


Figure 18:- The tasks are assigned to next nearest fog node

- From the above Figure 18, the nearest fog node capacity is not enough for the performing operations. Then tasks are assigned to the next high capability of the fog node.

# 6 Cloud Analyst Tool

## 6.1 Download Cloud Analyst tool

- You can download the cloud analyst tool . Zip file from the given link [4].

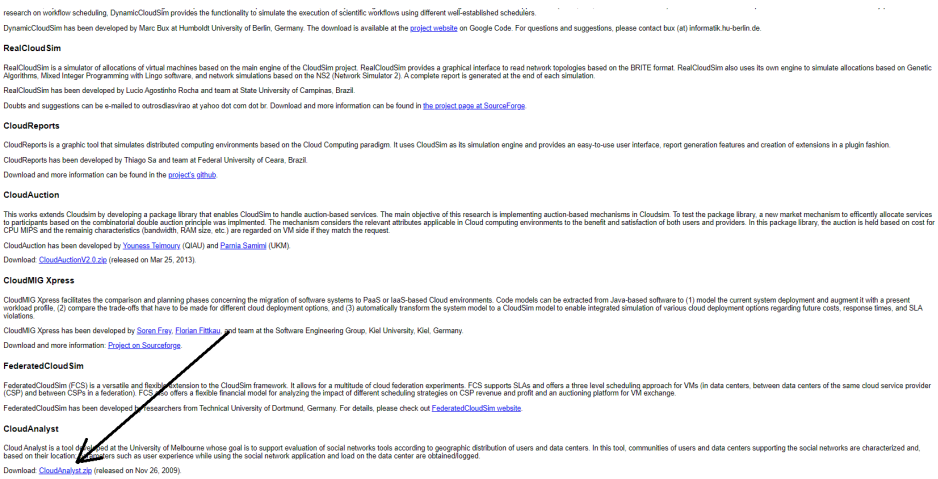



Figure 19: - Download the Cloud Analyst tool

- Extract the . Zip file and Open that file. Then run the file which was shown in below figure 20.

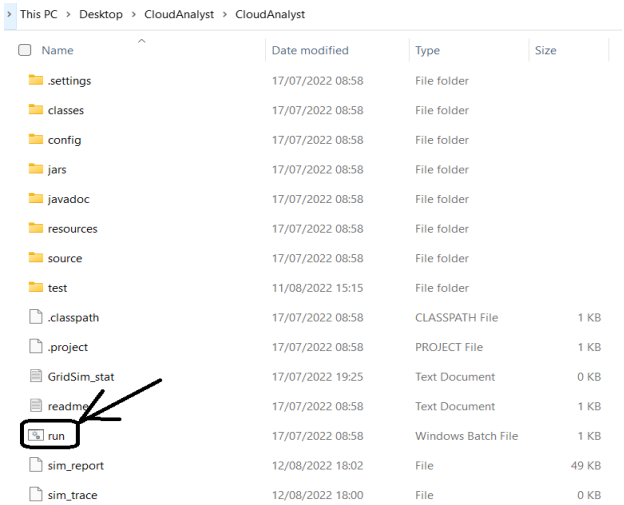



Figure 20:- Run the Tool

## 6.2 Regions in Cloud Analyst tool

- The division of regions in the cloud analyst toolkit is shown in below figure 21.

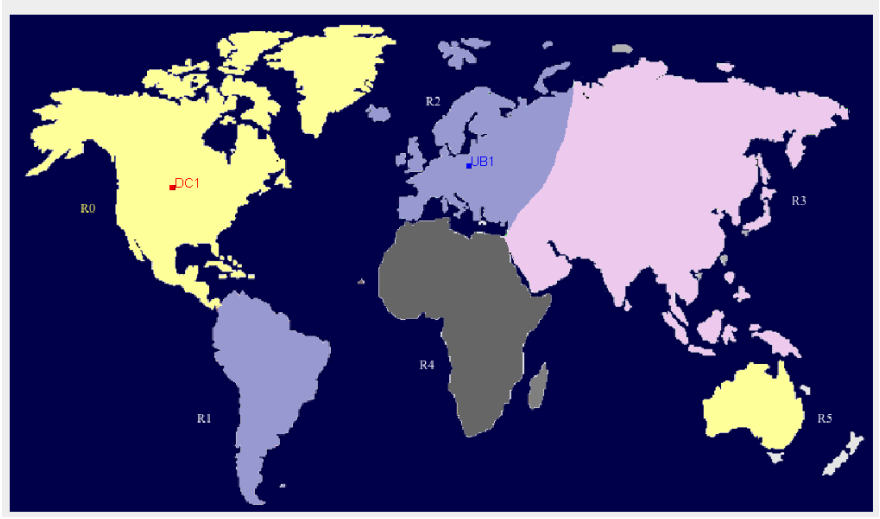



Figure 21:- The Region division in Cloud analyst tool

## 6.3 Configuration of User bases

- Userbases are used for the generation of traffic to the data center. The world map is divided into 6 regions and data centers are placed at different locations. Below Figure 22 shows the user base configuration.

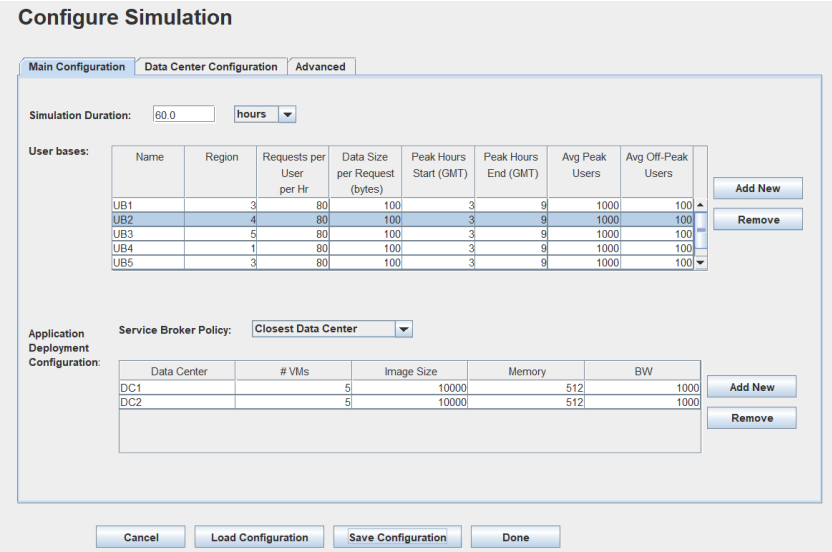



Figure 22:- Userbases Configuration

- In the user base, the service broker policies can be configured from the drop down list.

## 6.4 Data Center Configuration

- Data Centers are able to perform the tasks that are generated from the userbases. Below is figure 23.

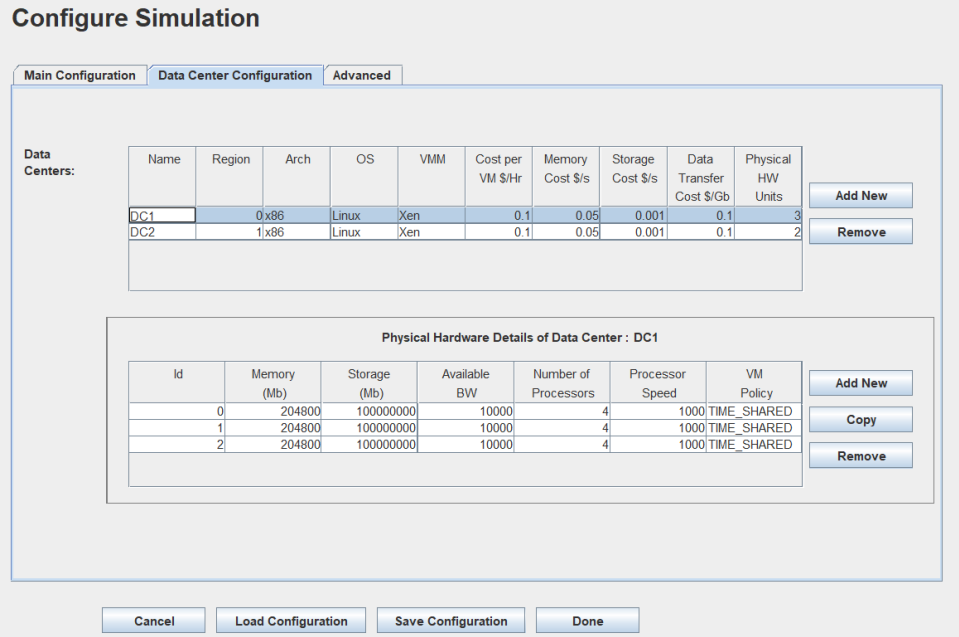



Figure 23:- Data Center Configuration

- After configuration of userbases and data centers configuration in different regions on world map. The figure is shown in below figure 23.

## 6.5 After the Configuration

- Configuration of user base and data centers are completed

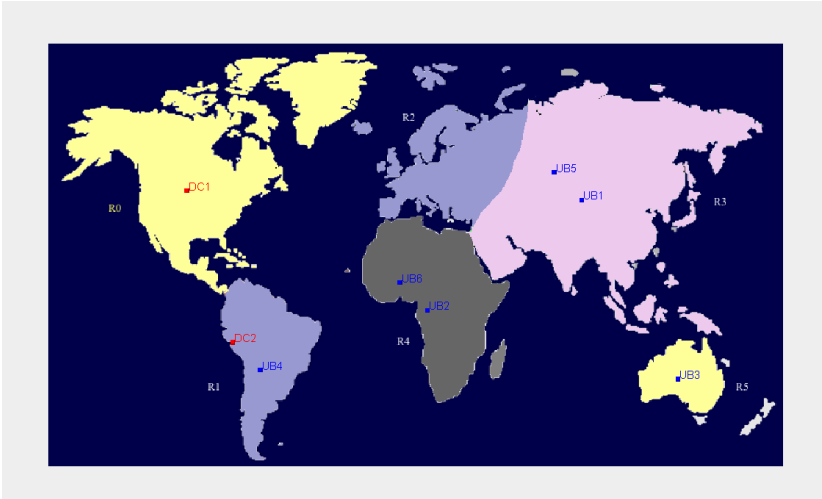



Figure 24:- Configured the Userbase and Data centers

## 6.6 Simulation Results

- By running the simulation results with different service broker policies and algorithms. The outputs will be shown in below figure 25 and figure 26.

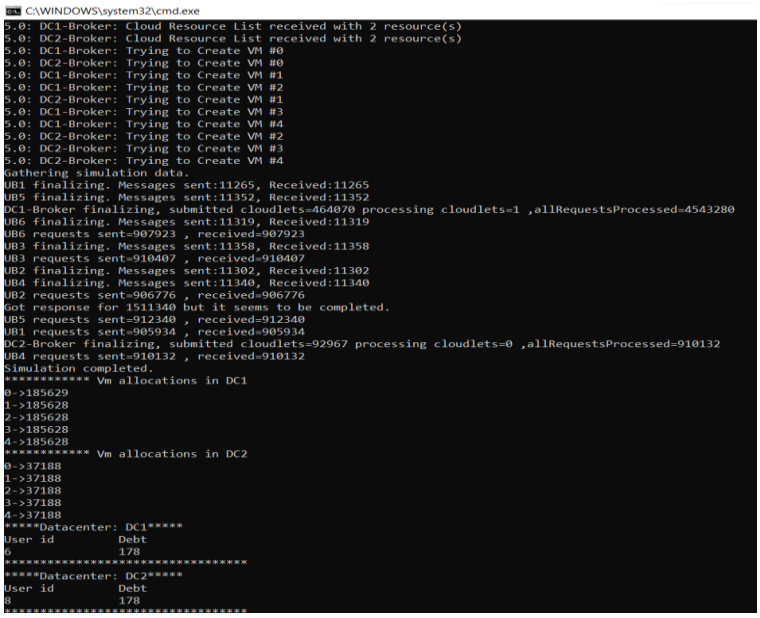



Figure 25:- The outputs in command window

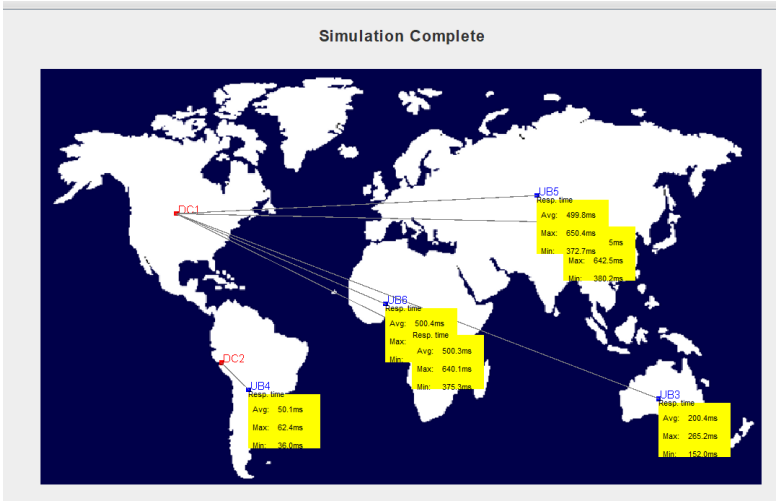



Figure 26:- The results of Cloud analyst in tool

# References

[1] “Oracle,” [Online]. Available: https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html. [Accessed August 2022].

[2] “Eclipse Foundation,” Eclipse, [Online]. Available: https://www.eclipse.org/downloads/aca.

[3] “Cloudslab/IfogSIM,” GitHub, [Online]. Available: https://github.com/Cloudslab/iFogSim.

[4] “CloudSim: A Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services,” The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, [Online]. Available: http://www.cloudbus.org/cloudsim/. [Accessed June 2022].