Configuration Manual

MSc Research Project
Cloud Computing

Sheffy Batra
Student ID: x22121790

School of Computing
National College of Ireland

Supervisor: Dr. Punit Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sheffy Batra
Student ID:	x22121790
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Punit Gupta
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	699
Page Count:	8

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:	13th December 2023

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

Sheffy Batra
x22121790

1 Introduction

This report shows the step-by-step procedure to implement the below experiment steps and the configurations that are used to run the project. The cloudsim environment and its dependencies with various parameters are documented in the below sections for successfully running this project.

2 Prerequisite

The project is implemented on a local laptop with a specification of a 64-bit operating system, x64-based processor, and Windows 11 with 16.0 GB RAM. The Table 1 2 consist of the packages that needs to be downloaded and are used in the project.

Eclipse IDE : It is used as a platform for Cloudsim.

Cloudsim: Simulator is used to run the algorithms and verify the results.

Packages	Version	Resource Links
Eclipse	2023-03	`https://www.eclipse.org/downloads/packages/release/2023-03/r`
Cloudsim Zip	3.0	`https://github.com/Cloudslab/cloudsim/releases`
Java JDK	17.0.4	`https://www.oracle.com/java/technologies/downloads/#java17`
Common Math library	3.2	`https://mvnrepository.com/artifact/org.apache.commons/commons-math3/3.2`

Table 1: Downloadable Pacakges List

3 Cloudsim Parameter Configurations

To simulate different numbers of VMs and hosts, the parameter is set in the Random-Constant.java file.(Figure1)

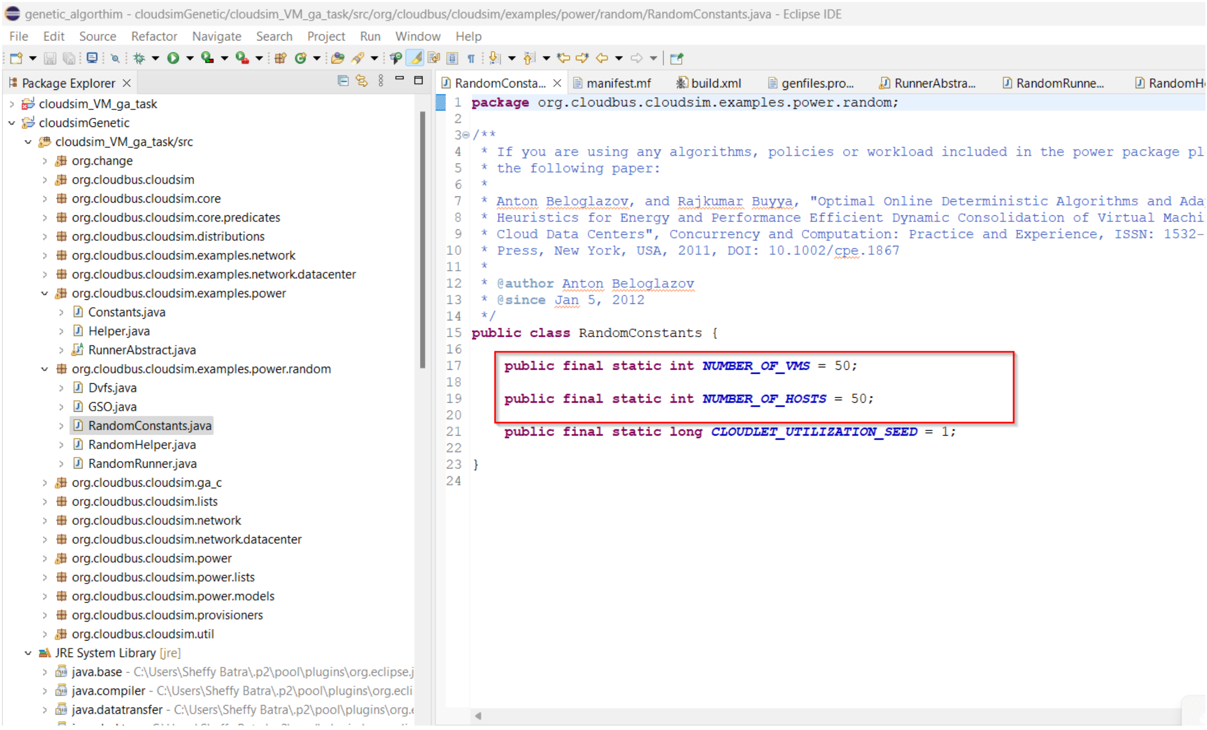

Figure 1: RandomConstants.java

The proposed algorithm has various parameters mutationRate, tournamentSize, and elitism which are set in the GA.java file of the algorithm (Figure(2)

Figure 2: GA.java

The simulation parameters of VM and Host Types and configuration of VM's and Host are stored in Constants.java (Figure 3)

Figure 3: Constants.java

By configuring all these parameters the simulation environment is created and the algorithms are tested for efficiency.

4 Code configuration and simulation Files

The code is developed in java and algorithms that are used is a combination of Genetic and Game theory.Figure 4 represent the pseduo code of the algorithm

Algorithm 1 Game Genetic VM Allocation

Initialize Population:
  Generate an initial population of VM allocations randomly.
for Generation ← 1 to MAX_GENERATIONS **do**
  Evaluate Fitness:
    Calculate the fitness values for each individual in the population based on the payoff function.
  Tournament Selection:
    Perform tournament selection to choose two parents based on their fitness values.
  Nash Equilibrium-inspired Crossover:
    Calculate neighbor payoffs.
    Apply Nash equilibrium-inspired crossover to create offspring.
    Determine crossover points based on payoffs and perform crossover with a certain probability.
  Mutation:
    Apply mutation to the offspring with a certain probability.
    Randomly select a VM and flip its allocation.
  Replace Population:
    Replace the old population with the new population (parents and offspring).
  Apply Nash Equilibrium-inspired Modifications:
    Apply Nash equilibrium-inspired modifications to the population.
    Consider the payoff of neighbors and update each VM allocation based on neighbor payoffs.
  Find Best Allocation:
    Identify the index of the best individual in the population based on fitness values.
  Print Result:
    Output the best VM allocation in the current generation.
end for
End Algorithm =0

Figure 4: Game-Genetic Code Files

The code is basically defined in below 4 Files (Figure 5)

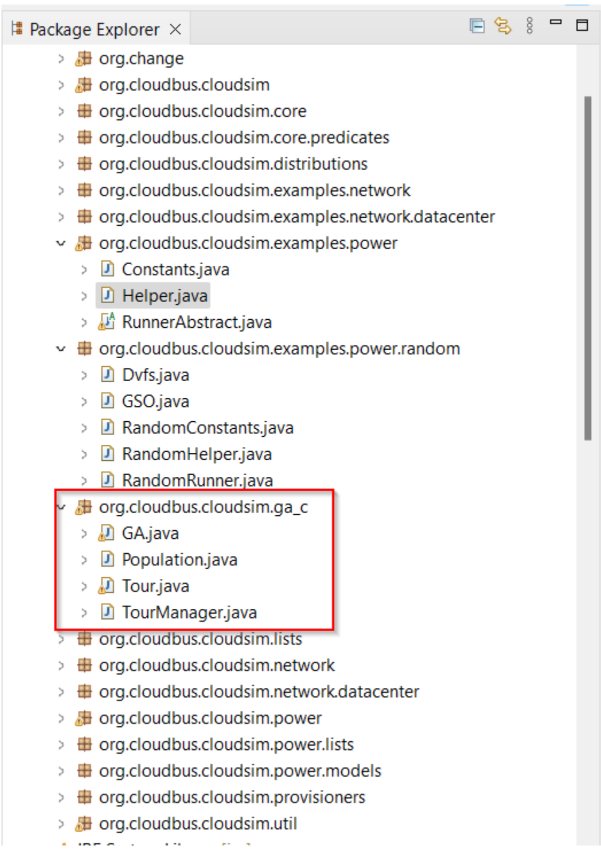

Figure 5: Game-Genetic Code Files

The Simulation is run and main function is in GSO.java file (Figure 6)

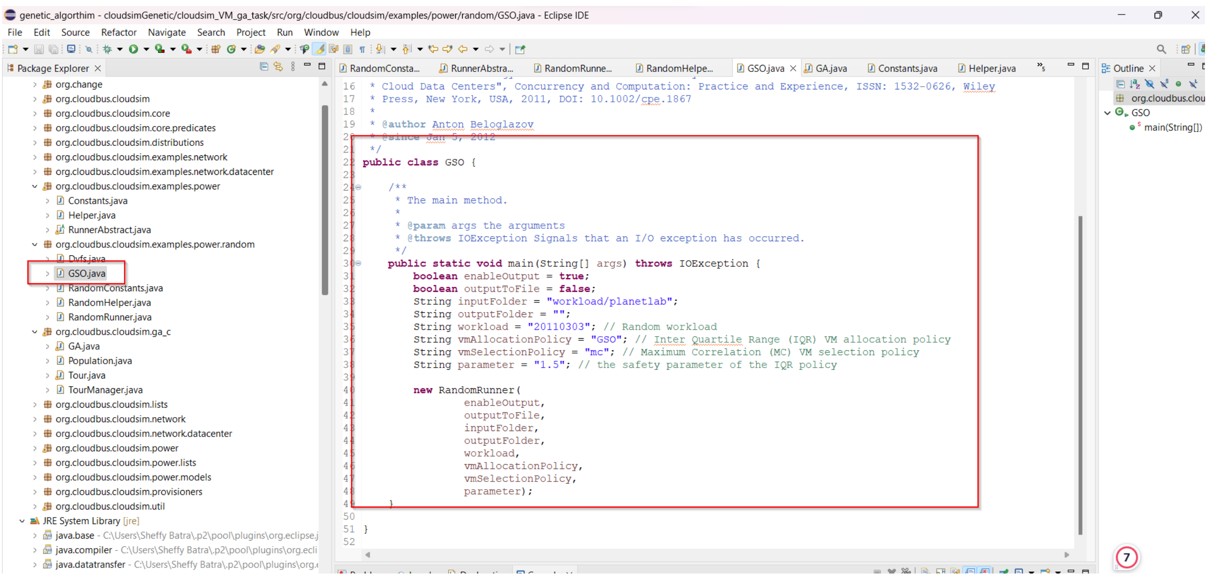

Figure 6: Main Simulation File

5 Test Results

The results are simulated in cloudsim and are shown below

1. Proposed Game Genetic Algorithm -GSO.java (Figure 7)
2. Genetic algorithm -Genetic.java (Figure 8)
3. IQRC algorithm- IQMC.java (Figure 9)
4. MAD algorithm - MADMc.java (Figure 10)

The figures show the simulation for 50 VM and 50 Host. Similalry the algorithms are tested for 100, 200, 300, 400 VM's and the results and graphs are represented in the report.

The Parameters used to evaluate the algorithm are

- No. of VM Migration
- Energy Consumption
- Total Execution Time
- Standard deviation Time

The results show that the proposed Game-Genetic Algorithm is better than other algorithms.

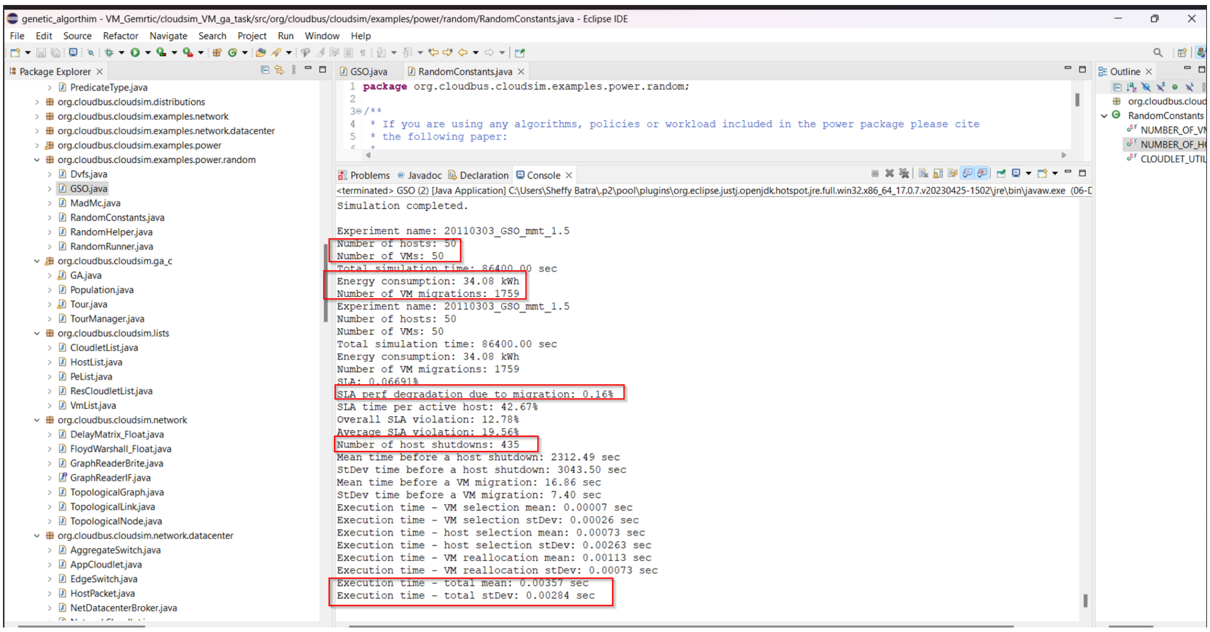

Figure 7: Proposed Game-Genetic Algorthim

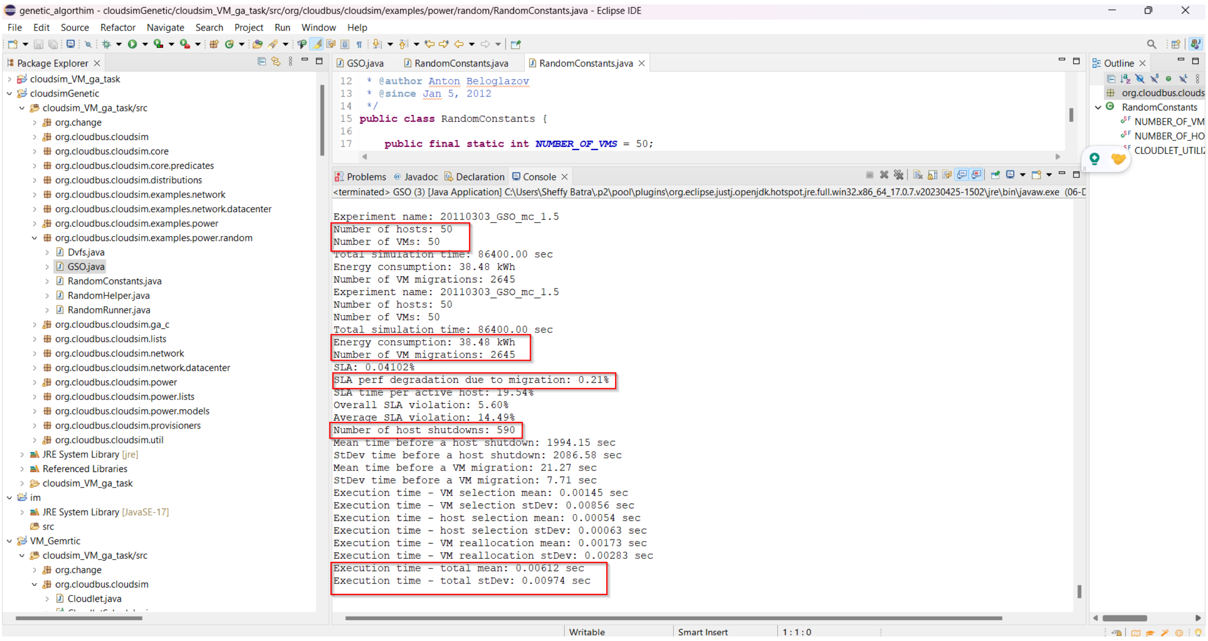

Figure 8: Genetic Algorthim

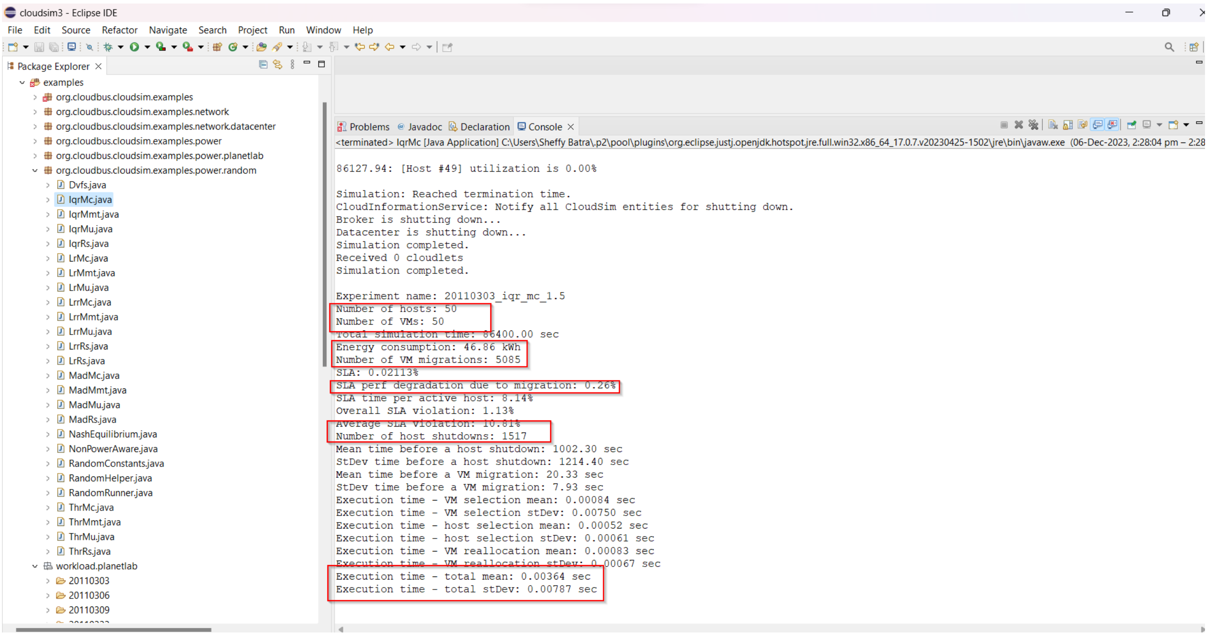

Figure 9: IQMC Algorthim

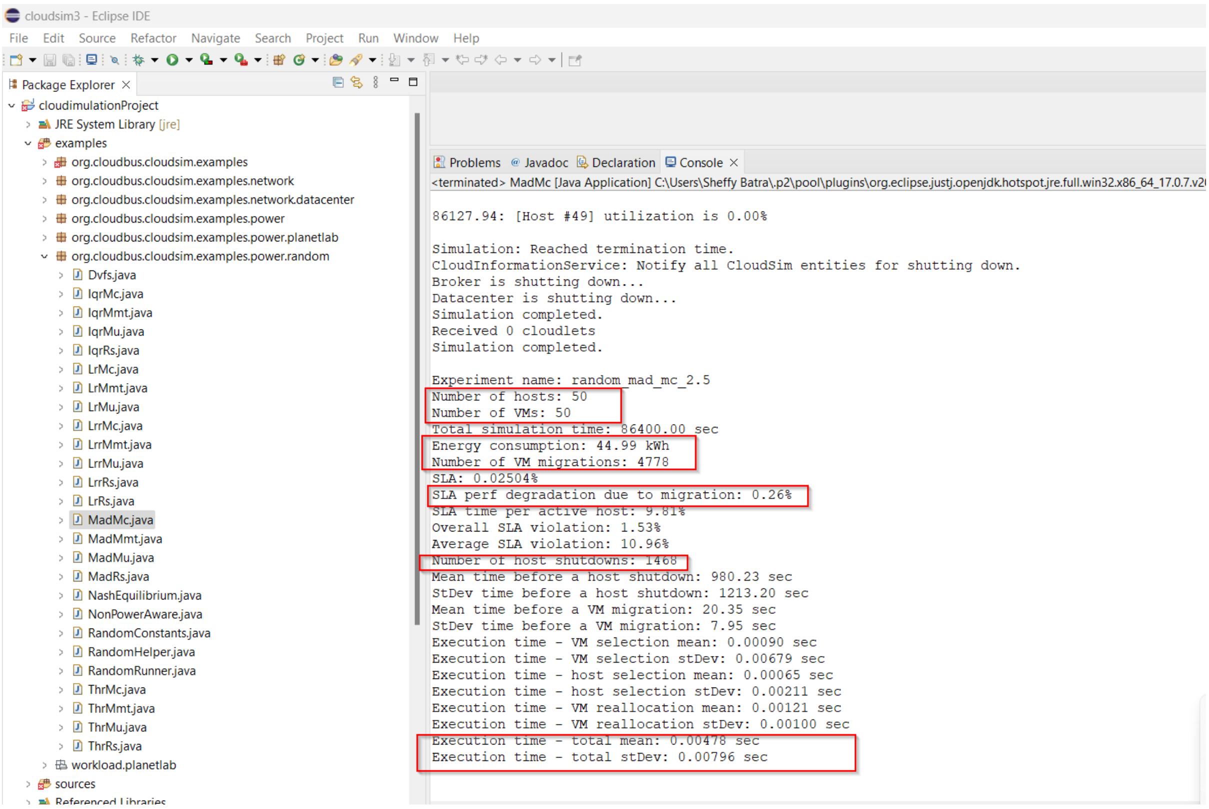

Figure 10: MAD Algorthim