National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Aarthy Ramaswamy
Student ID: X20188838

School of Computing
National College of Ireland

Supervisor: Divyaa Manimaran Elango

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Aarthy Ramaswamy
Student ID:	X20188838
Programme:	Cloud Computing **Year:** 2021-22
Module:	Research Project
Supervisor:	Divyaa Manimaran Elango
Submission Due Date:	January 31 2022
Project Title:	Simulation of effective task scheduling algorithm for cloud environments
Word Count:	1300 **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: Aarthy Ramaswamy……

Date: 31/1/2022……

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

Aarthy Ramaswamy
Student ID: X20188838

1 Introduction

1.1 Purpose of the Document

The configuration manual provides clarity on the simulation techniques that are followed in this cloud task scheduling project. It throws light into the steps involved in project set up and application handling.

2 System Requirements

The scheduling algorithms are practically expensive to run on real cloud environments due to which we have chosen simulation method to test the efficiency of FCFS and SJF algorithms. Initially the Cloudsim simulation tool is setup in local machine with the following system configuration.

2.1 Hardware requirements for Cloudsim

The CLoudsim is a simple Java simulation tool which requires basic system requirements only and not any high computing configurations. The system requirements are given below which were followed for the project set up.

Operating system: Windows 10 x64

Processor: Intel core i5

RAM: 140 GB

Hard Drive: 600GB

The above configuration is not the minimum requirements, any computer system with a dual-core processor, 2 GB RAM, and 1 GB storage is good enough to simulate the cloud-based systems using the Cloudsim.

2.2 Software requirements

Cloudsim is a tool built in Java completely, it requires an environmentalk set up to run java files and to design algorithms on it. The applications, IDE and software required to set up the Java IDE and build algorithms on it are mentioned below.

Eclipse

Java Development Toolkit (JDK)

CloudSim 3.0.3

GitHub

3 Setup of simulation environment

The simulator set up involves various steps from IDE installation until running the project to instantiate a simulated cloud environment similar to real time cloud machine. Before starting to working on CloudSim, it needs to be configured correctly. As the tool is developed using java language, we can use any IDE that supports java language development environment like NetBeans, eclipse etc. In this document we will set it up using Eclipse IDE.

3.1 Pre requisite Setup

Step 1: Download Eclipse IDE installation file from the following link and follow the steps to set up the IDE with default settings.
Eclipse Java IDE: https://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/keplersr2

Step 2: To work on Java project requires a JDK which will be set up by following through the steps in the installation kit downloaded above.

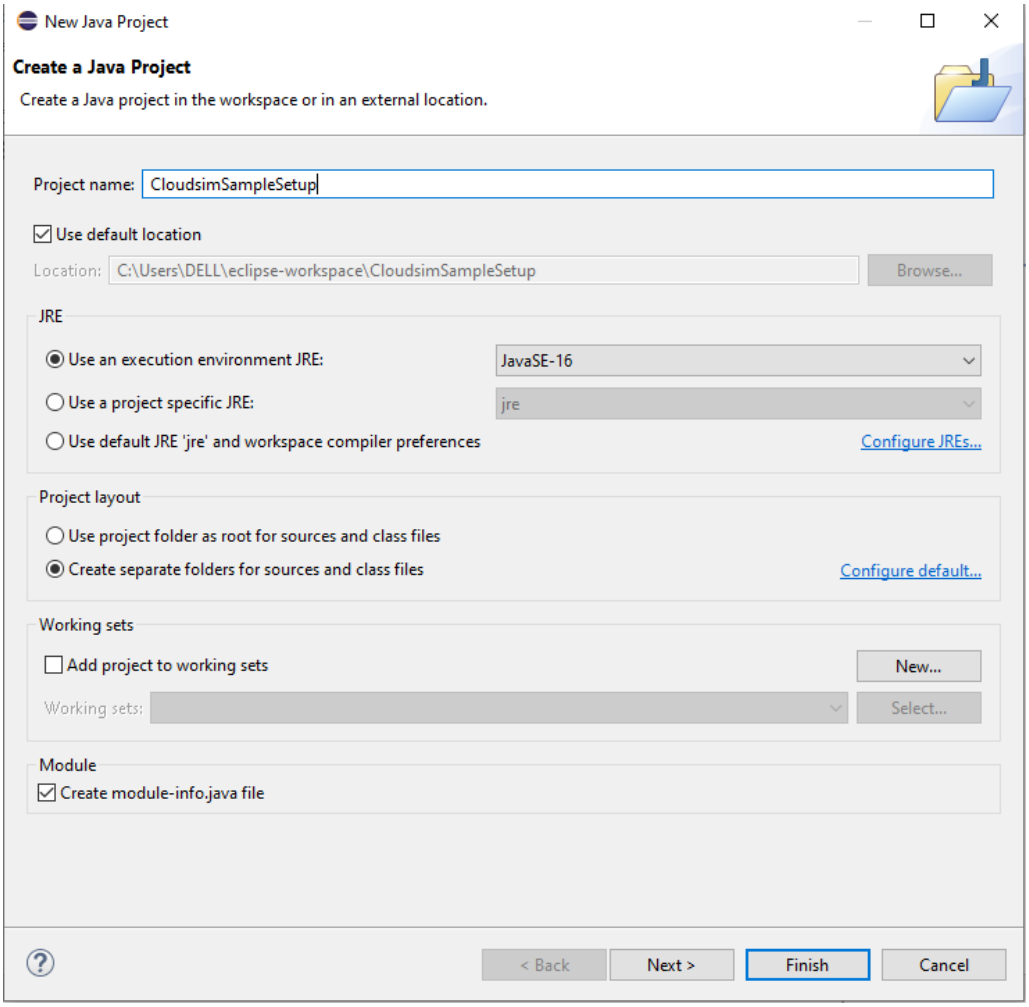
3.2 CloudSim Setup

For this project we have used Cloudsim 3.0.3 version as it is the most stable set up to build algorithms and gives a most similar simulation cloud environment.

Step 1: Download the code files zip from the link provided below. Unpack the downloaded 'CloudSim-3.0.3.tar.gz' or 'CloudSim-3.0.3.zip. The folder structure of the code is as detailed below.

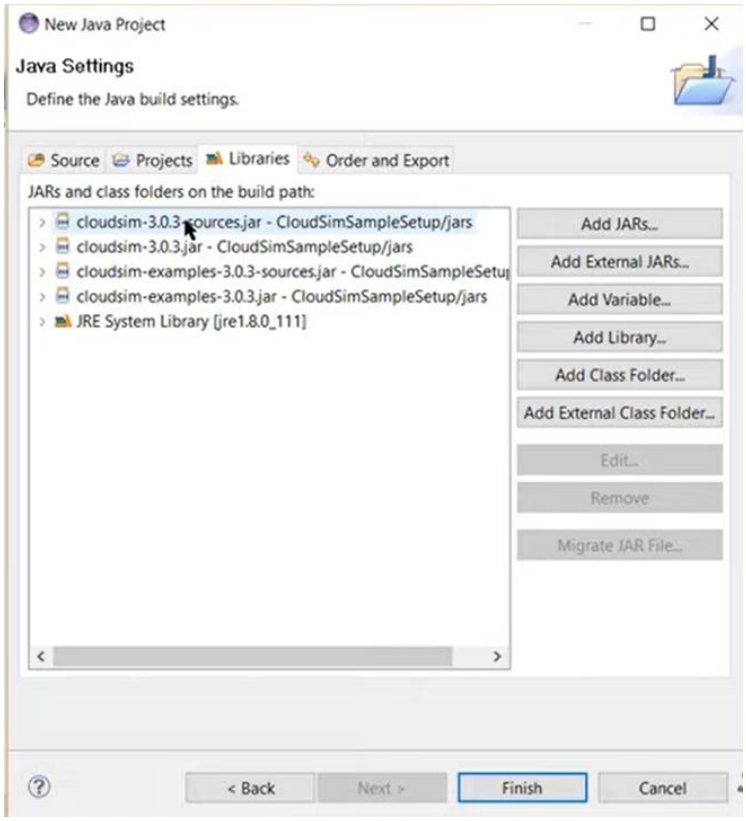
cloudsim/	-- top level CloudSim directory
docs/	-- CloudSim API Documentation
examples/	-- CloudSim examples
jars/	-- CloudSim jar archives
sources/	-- CloudSim source code
tests/	-- CloudSim unit tests

Step 2: Now we shall set up the workspace path for working in IDE. Navigate to **new workspace** in **File** menu of Eclipse. In the path textbox, browse the folder where unzipped code files are placed. Provide a suitable name to the workspace. The figure below shows a similar workspace created for this project.

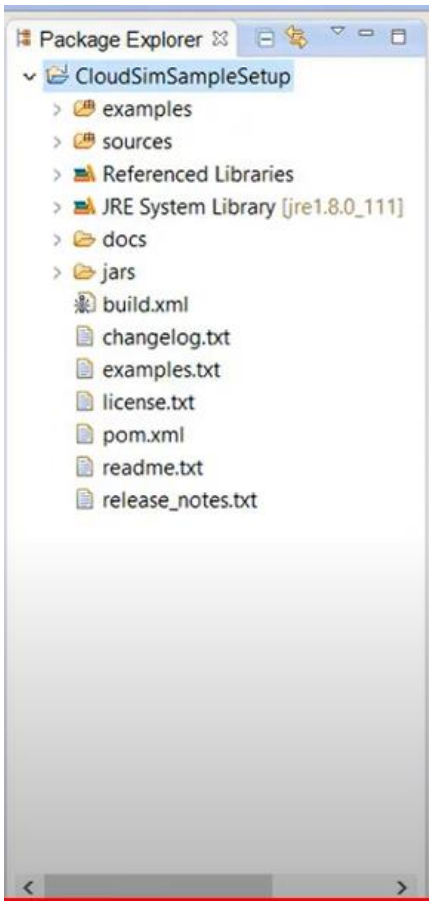


Step 3: Now the libraries required for the running of this toolkit is configured. For configuring the JDK, jar file from this link can be downloaded 'http://commons.apache.org/proper/commons-math/download_math.cgi' and extract 'commons-math3-3.6.1.jar'.

Step 4: In the libraries dialogue box, choose **Add External JARS.** Choose the unzipped jar file from downloaded location. This is shown in below figure.



Step 5: Upon successful set up of workspace, the project files are available in the explorer of IDE. The build is automatically started to make project ready for working. the figure shows project structure.



Now the simulation tool is ready to build algorithms and run the required configuration.

3.3 FCFS Algorithm

Upon successful set up of the Cloudsim tool, the task scheduling algorithm FCFS is built on it with simulated environmental set up to perform task scheduling. The code for running this algorithm is available in the artefacts submitted along with the project files. This code was developed and tested for various scenarios. The files FCFS_Scheduler.java and FCFSDatacenterBroker.java form this algorithm logic.

```
public class FCFS_Scheduler {

    private static List<Cloudlet> cloudletList;
    private static List<Vm> vmList;
    private static Datacenter[] datacenter;
    private static double[][] commMatrix;
    private static double[][] execMatrix;
```

The VM and Cloudlet are configured in the scheduler.java file.

3.4 SJF Algorithm

The artefacts folder contains SJF algorithm design files SJFDatacenterBroker.java and SJF_Scheduler.java. This code files where developed and tested to give the proposed outputs. The datacenter and cloudlet set up are done in these files and SJF algorithm is also run for same environmental setup as the previous algorithm.

```
public class SJF_Scheduler {

    private static List<Cloudlet> cloudletList;
    private static List<Vm> vmList;
    private static Datacenter[] datacenter;
    private static double[][] commMatrix;
    private static double[][] execMatrix;
```

4 Task Scheduling and Validation

Following the completion of set up and algorithm design, the environment variables are to be configured and simulation is run to view results. For this we have used various environmental variable values for both the algorithms and validated the respective outputs for evaluation. The Cloudsim scheduler is run to perform task scheduling using the SJF algorithm and FCFS for static activities with a pre-determined processing time. The workloads, cloudlet length etc are pre defined and scheduler is run for these values. The cloudlet length is not taken into account by the FCFS algorithm. It starts with the first task (cloudlet) and then moves on to the next. However, the SJF algorithm sorts all cloudlets sent to the dc broker first. The experiment is carried out with 5 datacenters, and minimum of 30 cloudlets set up. The experiment is repeated several times to derive at a consistent result. The efficiency is checked with the make span time output from the simulation task.

```
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID        Time    Start Time    Finish Time
    00        SUCCESS        04            04        117.86       00.5          118.36
    01        SUCCESS        03            03        461.88       00.5          462.38
    02        SUCCESS        03            03        470.97     462.38          933.35
    03        SUCCESS        04            04        809.03     118.36          927.38
    04        SUCCESS        06            06        247.91       00.5          248.41
    05        SUCCESS        05            05        238.49       00.5          238.99
    06        SUCCESS        02            02        496.28       00.5          496.78
    07        SUCCESS        06            06        319.72     248.41          568.13
    08        SUCCESS        02            02        574.16     496.78         1070.94
    09        SUCCESS        02            02        993.52    1070.94          2064.47
    10        SUCCESS        06            06        218.04     568.13          786.16
    11        SUCCESS        02            02        752.31    2064.47          2816.78
    12        SUCCESS        03            03        575.73     933.35         1509.08
    13        SUCCESS        04            04        709.84     927.38         1637.22
    14        SUCCESS        05            05        575.49     238.99          814.48
    15        SUCCESS        02            02        973.1     2816.78         3789.87
    16        SUCCESS        04            04        374.32    1637.22          2011.54
    17        SUCCESS        05            05        861.92     814.48         1676.4
    18        SUCCESS        06            06        203.89     786.16          990.06
    19        SUCCESS        05            05        530.81     1676.4         2207.21
    20        SUCCESS        06            06        603.67     990.06         1593.73
    21        SUCCESS        03            03        91.03     1509.08         1600.11
    22        SUCCESS        05            05        702.1     2207.21         2909.31
    23        SUCCESS        05            05        649.85     2909.31          3559.16
    24        SUCCESS        02            02        398.72     3789.87          4188.6
    25        SUCCESS        04            04        358.21     2011.54          2369.76
    26        SUCCESS        03            03        641.19     1600.11          2241.3
    27        SUCCESS        03            03        345.91     2241.3         2587.21
    28        SUCCESS        02            02        646.58     4188.6         4835.18
    29        SUCCESS        06            06        750.72     1593.73          2344.45
Makespan using FCFS: 3996.083402984705
FCFS.FCFS_Scheduler finished!
```

FCFS Scheduler results

Repeating the same parameters for SJF scheduler provides a make span time of 6329 seconds as shown in figure below. Which is far beyond the FCFS scheduler.

```
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID        Time    Start Time    Finish Time
    01        SUCCESS        05            05        368.67       00.1          368.77
    02        SUCCESS        04            04        647.76       00.1          647.86
    03        SUCCESS        02            02        678.17       00.1          678.27
    06        SUCCESS        06            06        702.03       00.1          702.13
    00        SUCCESS        03            03        745.58       00.1          745.68
    07        SUCCESS        06            06        319.72     702.13          1021.84
    18        SUCCESS        06            06        203.89    1021.84          1225.74
    04        SUCCESS        03            03        535       745.68          1280.68
    16        SUCCESS        05            05        960.27     368.77         1329.03
    09        SUCCESS        04            04        748.46     647.86         1396.32
    11        SUCCESS        02            02        752.31     678.27         1430.58
    10        SUCCESS        04            04        325.12    1396.32          1721.43
    05        SUCCESS        03            03        474.13    1280.68          1754.81
    19        SUCCESS        05            05        530.81    1329.03          1859.84
    20        SUCCESS        05            05        141.65    1859.84          2001.49
    12        SUCCESS        02            02        807.91    1430.58          2238.49
    15        SUCCESS        04            04        636.27    1721.43          2357.7
    08        SUCCESS        03            03        727.34    1754.81          2482.14
    22        SUCCESS        05            05        702.1     2001.49         2703.6
    13        SUCCESS        03            03        396.17     2482.14          2878.31
    21        SUCCESS        03            03        91.03     2878.31         2969.34
    25        SUCCESS        05            05        327.74     2703.6         3031.33
    14        SUCCESS        02            02        873.55     2238.49          3112.04
    26        SUCCESS        05            05        247.6     3031.33         3278.93
    17        SUCCESS        04            04        1037.09      2357.7          3394.79
    23        SUCCESS        02            02        833.26     3112.04          3945.3
    24        SUCCESS        04            04        943.63     3394.79          4338.42
    27        SUCCESS        04            04        917.57     4338.42          5256
    28        SUCCESS        04            04        428.55     5256           5684.55
    29        SUCCESS        04            04        739.39     5684.55          6423.94
Makespan using SJF: 6329.79919328607
```

SJF Scheduler results.