National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Uday Kumar Das
Student ID: 20191022

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Uday Kumar Das
Student ID:	20191022
Programme:	Cloud Computing
Year:	2021
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	16/12/2021
Project Title:	Resource Scheduling for Infrastructure as a Service(IaaS) in cloud computing
Word Count:	XXX
Page Count:	5

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:	Uday Kumar Das
Date:	16th December 2021

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

Uday Kumar Das
20191022

1 Introduction

In order to run CloudSim resource scheduling code, some steps of installation is required for the project, which is available in this manual. In below sections, system configurations, dependencies related to CloudSim resource scheduling code and the implementation steps has been discussed.

1.1 System Configuration

1.1.1 Hardware

Need a computer with minimum dual-core CPU and 4GB of RAM and 100 GB of storage is required to run the CloudSim software simulation toolkit, which is designed in the Java programming language.

1.1.2 Software

- Java Development Kit (Version: 11.0) is required for compilation the JAVA codes. It contains tools for building and testing Java-based applications. Windows, macOS, and Linux are all viable options.
- Eclipse IDE: Eclipse is an integrated development environment. An extensible plug-in system allows you to configure the environment to your needs.
- CloudSim: It is a cloud based simulation tool, which have the ability to simulate, manage services, and model IaaS cloud infrastructure in a single framework.

2 Code Development

2.1 Constant Configuration

In Figure 1, Constant Configuration has been shown. Here, configuring Number of Tasks, Data Centers, VMs etc.

2.2 Changes in CloudSim Code

To create a data center environment in cloudsim, we need to identify the relevant classes where we need to update the methods in the java code, based on the following setup. The following is a list of the three classes in the cloud simulator toolkit that have been modified or created.

```java
package utils;

public class Constants {
    public static final int NO_OF_TASKS = 30; // number of Cloudlets;
    public static final int NO_OF_DATA_CENTERS = 5; // number of Datacenters;
    public static final int NO_OF_VM = 10; // number of Virtual Machines;
    public static final int POPULATION_SIZE = 25; // Number of Particles.
}
```

Figure 1: Constant Configuration.

- **BATscheduler.java** : Added this for for BAT Scheduling for cloud resources in CloudSim.
- **PSOscheduler.java** : Modified this file, so that PSO scheduling can be used for comparative analysis with other schedulers.
- **FCFSscheduler.java** : Modified this file, so that FCFS scheduling can be used for comparative analysis with other schedulers.

2.2.1 Scheduler Code

As shown in Figure 2, some of the BATscheduler.java file codes has been shown.

Console | BATscheduler.java

```java
        double response_time = 0.0;
        double waiting_time = 0.0;
        double makespan = calcMakespan(list);
        dft.setMinimumIntegerDigits(2);
        for (int i = 0; i < size; i++) {
            cloudlet = list.get(i);
            Log.print(indent + dft.format(cloudlet.getCloudletId()) + indent + indent);
            if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {
                Log.print("SUCCESS");

                CPU_time += cloudlet.getActualCPUTime();
                response_time = cloudlet.getFinishTime() - cloudlet.getSubmissionTime();
                waiting_time += cloudlet.getWaitingTime();
                Log.printLine(indent + indent + dft.format(cloudlet.getResourceId()) +
                        indent + indent + indent + dft.format(cloudlet.getVmId()) +
                        indent + indent + dft.format(cloudlet.getActualCPUTime()) +
                        indent + indent + dft.format(cloudlet.getExecStartTime()) +
                        indent + indent + dft.format(cloudlet.getFinishTime()) +
                        indent + indent + (dft.format(cloudlet.getWaitingTime())));
            }
        }

        Log.printLine("Makespan using BAT: " + makespan);
        Log.printLine("Total CPU Time  "+ CPU_time);
        Log.printLine("Total Response Time: " + response_time);
        Log.printLine("Avg. Waiting Time: " + (waiting_time / Constants.NO_OF_TASKS));
        Log.printLine("This simulation is for "+ Constants.NO_OF_VM + " VMs and " + Constants.NO_OF_TASKS + " Tasks.");
    }

    private static double calcMakespan(List<Cloudlet> list) {
        double makespan = 0;
        double[] dcWorkingTime = new double[Constants.NO_OF_DATA_CENTERS];

        for (int i = 0; i < Constants.NO_OF_TASKS; i++) {
            int dcId = list.get(i).getVmId() % Constants.NO_OF_DATA_CENTERS;
            if (dcWorkingTime[dcId] != 0) --dcWorkingTime[dcId];
            dcWorkingTime[dcId] += execMatrix[i][dcId] + commMatrix[i][dcId];
            makespan = Math.max(makespan, dcWorkingTime[dcId]);
        }
        return makespan;
    }

}
```

Figure 2: BATscheduler.java Code.

Some of the Interfaces, Classes and Methods are shown in Figure 3 :

Cloudlet	Each class that has to perform a task should implement Cloudlet interface.NullPointerExceptions can be avoided by utilizing the NULL object rather than assigning null to Cloudlet variables.
vmList	It is a collection of operations on lists of VMs on cloud.
Datacenter	This class allows the specification of the Datacenter's functional needs, this class also includes methods for configuring VM resource allocation policies.
createVM	It is a method which can create VMs/Cloudlets and send it to a specific broker.
DatacenterCharacteristics	The attributes and rules for a data center are defined by the this Interface.
getCloudletId	This function fetches the ID for a cloudlet running in CloudSim.
getResourceId	This function gets the resource id.
getVmId	This function fetches the VM ID, running in CloudSim.
getCloudletId	This function gets the id for a cloudlet running in CloudSim.
getCloudletId	This function gets the id for a cloudlet running in CloudSim.
getActualCpuTime	Returns the total execution time of the Cloudlet in seconds.
getExecStartTime	Gets the latest execution start time of this Cloudlet.
getFinishTime	Gets the time when this Cloudlet has completed executing in the latest Datacenter.
getWaitingTime	Gets the time the cloudlet had to wait before start executing on a resource.

Figure 3: Details of Interfaces, Classes and Methods Used for BAT, PSO and FCFS Schedulers

3 Test Results

In order to run simulation for BAT scheduler, **BATscheduler** JAVA file should be executed in Eclipse IDE .Means Run as java Application in Eclipse. Similarly for PSO and FCFS run **PSOscheduler** and **FCFSscheduler** JAVA files respectively. The simulation output for all 3 schedulers BAT, PSO and FCFS has been shown in Figures 4, 5 and 6 respectively.

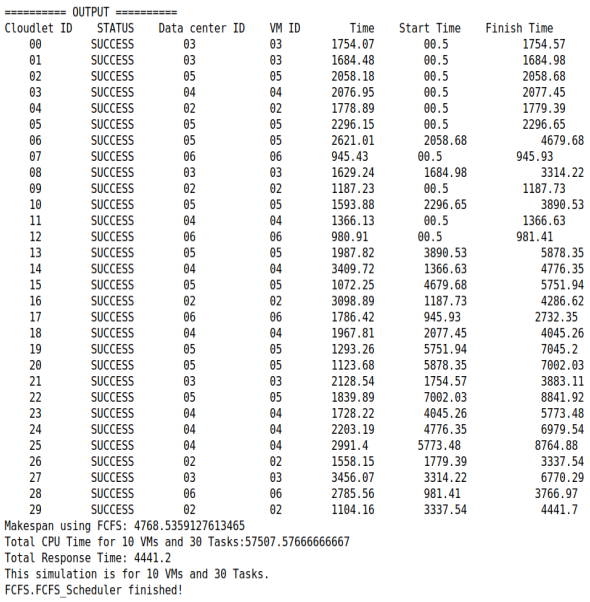
```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID        Time    Start Time    Finish Time
    05         SUCCESS         03            03        815.03       00.1           815.13
    12         SUCCESS         06            06        980.91       00.1           981.01
    00         SUCCESS         02            02       1055.15        00.1           1055.25
    03         SUCCESS         02            02       1310.98        00.1           1311.08
    11         SUCCESS         04            04       1366.13        00.1           1366.23
    01         SUCCESS         05            05       1520.91        00.1           1521.01
    22         SUCCESS         03            03       1035.23       815.13           1850.36
    09         SUCCESS         06            06       1990.93        00.1           1991.03
    02         SUCCESS         05            05       2058.18        00.1           2058.28
    21         SUCCESS         03            03       2128.54        00.1           2128.64
    17         SUCCESS         02            02       1459.39       1311.08           2770.47
    13         SUCCESS         06            06       1812.03        981.01          2793.04
    04         SUCCESS         02            02       1778.89       1055.25           2834.13
    08         SUCCESS         04            04       2840.5         00.1           2840.6
    07         SUCCESS         05            05        934.37       2058.28           2992.64
    23         SUCCESS         03            03       1333.16       1850.36           3183.51
    24         SUCCESS         03            03       1431.66       2128.64           3560.3
    06         SUCCESS         05            05       2621.01       1521.01           4142.02
    26         SUCCESS         02            02       1558.15       2834.13           4392.28
    25         SUCCESS         02            02       1725.63       2770.47           4496.1
    10         SUCCESS         05            05       1593.88       2992.64           4586.52
    14         SUCCESS         04            04       3409.72       1366.23           4775.95
    16         SUCCESS         04            04       2740.68       2840.6           5581.28
    29         SUCCESS         02            02       1104.16       4496.1           5600.26
    15         SUCCESS         06            06       3719.47       1991.03           5710.5
    19         SUCCESS         05            05       1293.26       4586.52           5879.78
    28         SUCCESS         05            05        200.28       5879.78           6080.06
    20         SUCCESS         04            04       1439.09       4775.95           6215.04
    18         SUCCESS         05            05       2089.44       4142.02           6231.46
    27         SUCCESS         02            02       2501.86       4392.28           6894.14
Makespan using BAT: 4258.358085640115
Total CPU Time  51848.593333333345
Total Response Time: 6894.04
This simulation is for 10 VMs and 30 Tasks.
BAT.BATscheduler finished!
```

Figure 4: Simulation output for BAT Scheduler

```
************ OUTPUT *************
CloudletId    DC Id    VM Id    Status    Time    StartTime    FinishTime
    05         06        06    SUCCESS    815.03    00.1         815.13
    09         04        04    SUCCESS    900.47    00.1         900.57
    07         05        05    SUCCESS    945.43    00.1         945.53
    12         03        03    SUCCESS    1041.11    00.1         1041.21
    02         04        04    SUCCESS    1047.9    00.1         1048
    00         02        02    SUCCESS    1055.15    00.1         1055.25
    01         03        03    SUCCESS    1520.91    00.1         1521.01
    06         06        06    SUCCESS    913.36    815.13         1728.49
    10         05        05    SUCCESS    967.28    945.53         1912.81
    03         05        05    SUCCESS    1953.51    00.1         1953.61
    04         06        06    SUCCESS    1964.35    00.1         1964.45
    08         02        02    SUCCESS    1984.49    00.1         1984.59
    16         06        06    SUCCESS    549.08    1728.49         2277.57
    20         05        05    SUCCESS    369.38    1912.81         2282.19
    15         02        02    SUCCESS    807.67    1984.59         2792.26
    21         05        05    SUCCESS    861.38    1953.61         2814.99
    13         03        03    SUCCESS    1987.82    1041.21         3029.03
    14         03        03    SUCCESS    1508.65    1521.01         3029.66
    24         05        05    SUCCESS    982.95    2282.19         3265.14
    19         06        06    SUCCESS    1305.02    1964.45         3269.47
    22         06        06    SUCCESS    1035.23    2277.57         3312.8
    27         05        05    SUCCESS    407.14    3265.14         3672.28
    11         02        02    SUCCESS    2683.59    1055.25         3738.84
    17         03        03    SUCCESS    783.12    3029.03         3812.15
    25         03        03    SUCCESS    935.5    3029.66         3965.16
    28         03        03    SUCCESS    200.28    3812.15         4012.43
    26         05        05    SUCCESS    1489.92    2814.99         4304.91
    18         02        02    SUCCESS    1672.24    2792.26         4464.49
    23         02        02    SUCCESS    1163.91    3738.84         4902.75
    29         03        03    SUCCESS    1370.75    3965.16         5335.91

Best fitness value: 5133.618724366767
Best makespan: 2805.114631783332
Total CPU Time  35222.603333333325
Total Response Time: 5335.8099999999995
This simulation is for 10 VMs and 30 Tasks.
PSO.PSO_Scheduler finished!
```

Figure 5: Simulation output for PSO Scheduler



```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID        Time    Start Time    Finish Time
    00         SUCCESS         03             03        1754.07        00.5           1754.57
    01         SUCCESS         03             03        1684.48        00.5           1684.98
    02         SUCCESS         05             05        2058.18        00.5           2058.68
    03         SUCCESS         04             04        2076.95        00.5           2077.45
    04         SUCCESS         02             02        1778.89        00.5           1779.39
    05         SUCCESS         05             05        2296.15        00.5           2296.65
    06         SUCCESS         05             05        2621.01        2058.68           4679.68
    07         SUCCESS         06             06        945.43        00.5           945.93
    08         SUCCESS         03             03        1629.24        1684.98           3314.22
    09         SUCCESS         02             02        1187.23        00.5           1187.73
    10         SUCCESS         05             05        1593.88        2296.65           3890.53
    11         SUCCESS         04             04        1366.13        00.5           1366.63
    12         SUCCESS         06             06        980.91        00.5           981.41
    13         SUCCESS         05             05        1987.82        3890.53           5878.35
    14         SUCCESS         04             04        3409.72        1366.63           4776.35
    15         SUCCESS         05             05        1072.25        4679.68           5751.94
    16         SUCCESS         02             02        3098.89        1187.73           4286.62
    17         SUCCESS         06             06        1786.42        945.93           2732.35
    18         SUCCESS         04             04        1967.81        2077.45           4045.26
    19         SUCCESS         05             05        1293.26        5751.94           7045.2
    20         SUCCESS         05             05        1123.68        5878.35           7002.03
    21         SUCCESS         03             03        2128.54        1754.57           3883.11
    22         SUCCESS         05             05        1839.89        7002.03           8841.92
    23         SUCCESS         04             04        1728.22        4045.26           5773.48
    24         SUCCESS         04             04        2203.19        4776.35           6979.54
    25         SUCCESS         04             04        2991.4        5773.48           8764.88
    26         SUCCESS         02             02        1558.15        1779.39           3337.54
    27         SUCCESS         03             03        3456.07        3314.22           6770.29
    28         SUCCESS         06             06        2785.56        981.41           3766.97
    29         SUCCESS         02             02        1104.16        3337.54           4441.7
Makespan using FCFS: 4768.5359127613465
Total CPU Time for 10 VMs and 30 Tasks:57507.57666666667
Total Response Time: 4441.2
This simulation is for 10 VMs and 30 Tasks.
FCFS.FCFS_Scheduler finished!
```

Figure 6: Simulation output for FCFS Scheduler.