Modified novel algorithm based on learning automata to determine the underutilized physical machine(PM) in a data center.

MSc Research Project
Cloud Computing

Bhavya Shukla
Student ID: x17150655

School of Computing
National College of Ireland

Supervisor: Manual tova-izquierdo

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Bhavya Shukla
Student ID:	x17150655
Programme:	Cloud Computing
Year:	2018
Module:	MSc Research Project
Supervisor:	Manual tova-izquierdo
Submission Due Date:	20/12/2018
Project Title:	Modified novel algorithm based on learning automata to determine the underutilized physical machine(PM) in a data center.
Word Count:	XXX
Page Count:	33

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:	28th January 2019

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):	

Modified novel algorithm based on learning automata to determine the underutilized physical machine(PM) in a data center.

Bhavya Shukla
x17150655

Abstract

Cloud computing offers IT solutions all over the world. It works on the pay-as-go model where customers can use services for various purpose such as academic, industrial and personal use. However, cloud data centres consume enormous amount of energy which leads to carbon dioxide emission in the environment. Thus, it leads to the contribution in the greenhouse gases. Furthermore, lot of energy is used to cool the servers in the data centres. Therefore, this paper focuses on determining the underutilized physical machine in the data centres and at the same time focuses on SLA violation. After the successful determination of underloaded host the virtual machine can be shifted from the underloaded to balanced host. The main contribution of the algorithm used is that it helps in determining the balanced host so that the underutilized resources can be run on that. The main problem of the underutilized physical machine is that it consumes energy even though they are not computing the data. Therefore, it is better to switch off these underutilized servers so that the hefty electric bills that the cloud providers pay can be reduced. The designed algorithm is dynamic as it is designed to deal with the heterogenous requests of the customers.

1 Introduction

In recent years, IT infrastructure are continuously growing as there is a need for the computational data. Cloud computing is a large-scale distributed computing in which pool of services are offered to the customers through internet so that users can avail the services(Chaisiri et al.; 2012) . The cloud works on the pay as you go model where customers only pay for the services which they avail. However, a large-scale infrastructure requires lot of energy to work or process the data. For instance, in 2006 the energy consumed by the IT infrastructure was around 4 billion dollar which doubled in the year 2011(Beloglazov and Buyya; 2010a). Another problem apart from the energy consumption is carbon dioxide emission which result in the greenhouse gases. The virtualization technology allows multiple instances to run on the same physical machine. Therefore, by using virtualization the resource usage can be increased and maximum throughput can be achieved.(Beloglazov and Buyya; 2010b). For energy saving dynamic consolidation and switching off the idle is the best practise. As idle load host consumes 70 percent of the power which it consumes in the peak utilization (Beloglazov and Buyya; 2010b).

Thus, proper management of the underutilized host is very important. During the VM live migration the idle need to be turned off to save energy (Menaud et al.; 2010).

As detecting the underutilized physical machine is very important in the data centre. Therefore, this paper deals with determining the underloaded host by using the learning automata algorithm. As the demand of the cloud services is heterogeneous therefore, this paper deals with heterogeneous workload provided by the planet lab data. This data contains the traces of workload in the csv file which is used while performing the simulation. For the simulation cloud sim 3.03(Aazam and Huh; 2014) is used which provides various predefined classes to carry out the simulation. The key contribution of this paper as follows-

1- Determining the underutilized physical machine.

2- Manages the heterogeneous workload.

3- Real environment data set has been used (planet lab).

4- Idle host is also determined.

The remainder of this paper is organized as follows. Section 2 defines the methodology used while performing. Next is implementation where coding details, cloud sim information, data base, screen shot are provided and at last conclusion.

1.1 Motivation

The main motivation behind this project is to minimize the unnecessary energy consumption that happens because of the underutilized physical machine in the cloud data centers. In addition to this another driving force behind doing this project is to minimize the SLA violation as well. As the servers which are not utilized to their threshold limit waste large amount of energy as not only the energy is required to run them but also, they require energy to cool their system which is a complete waste for the data centers. The modified novel algorithm based on the learning automata has been used to determine the overloaded host but not the underutilized host. Therefore, in this paper learning automata algorithm has been used to determine the underloaded host.

1.2 Research Question

Can a modified novel algorithm based on learning automata be used to determine the underutilized physical machine in a data center ?

1.3 Research Objective

To determine the underutilized physical machine in the data center using learning automata from the planet lab workload.

2 Related Work

The low utilization of the server is the major concern in the cloud data centres. As the energy cost is increasing at a rapid pace and more physical machine in the data centres would mean more money spent on their maintenance. Thus, virtualization helps in solving this problem by allowing multiple virtual machine instances to run on the same physical machine so that maximum efficiency can be achieved, and this process is termed as server consolidation. Dynamic resources allocation is commonly done in the virtualized data centre so that unpredicted resources flow can be managed. Thus, to deal with the dynamic work flow novel algorithm has been proposed which proceed in three steps measuring the historical data, forecasting the dynamic demand and then finally allocating virtual machine to physical machine to minimize the energy consumption. This algorithm is used to deal with the dynamic workload as based on the resource history the demand is predicted and then the allocation is performed. Hence, the main advantage of this algorithm is that dynamic load can be handled effectively. But the major disadvantage of this algorithm is that resources types such as I/O or CPU are not considered (Bobroff et al.; 2007)

The dynamic placement of virtual machine on the physical machine based to varying load is a well-studied problem. As the server utilization spike up to 11 to 15 percent during peak demand so there is a major concern to deal with this hiked workload. The proper management of the server is the major concern in the data centres as lot of carbon dioxide is released from the cooling system. The pMapper is an application placement controller that deals with varying workload. This placement algorithm proceeds in three major steps powers manager, performance and migration manager. The pMapper approach is different from the above novel approach (Bobroff et al.; 2007) as in this the cost of migrating the virtual machine is also estimated while dealing the varying workload. But the main similarity among both the algorithm is that it is designed to deal with the dynamic workload.(Verma et al.; 2008).

The virtualization technology like Xen provides the feature of max, min and share. The minimum is the min amount of resources that need to be allocated, maximum resources needed for the allocation and share ensures resources should be available to all devices for the computation process. The data centers presently being used does not focuses on using this virtual machine technology. This min-max approach is used in the power expand min max algorithm which means minimum amount of resources should be allocated and not all the resources should be allocated. This approach helps in minimizing the energy consumption of the data centres up to 40 percent (Cardosa et al.; 2009). The power expand min max algorithm is different from the above approach (Verma et al.; 2008) as it focuses mainly on resource allocation to PM but not much the migration part in the data centers.

The data centers need lot of electrical energy to run the servers. Thus, the bills increase not only this data centers damages the environment by emitting carbon in the atmosphere. Thus, it is very important that the data centers should be managed efficiently. Hence, an adaptive threshold-based approach was introduced to adjust according to the workload. That mean if the CPU utilization is below the threshold limit then VM should be migrated as it means it is over-utilized. If the CPU utilization is below the threshold limit, then it should be migrated as it must underutilized. This mean that the physical machine is unnecessarily consuming more electrical energy. This approach deals with the dynamic workload at the same time SLA violation is prevented (Beloglazov and Buyya; 2010a) . This approach is quite similar to power expand min max(Cardosa et al.; 2009) as CPU utilization is important factor.

Virtualization technology is the key to increase the computation speed so that processing can be achieved at a faster rate. Server consolidation is the main problem in the data centres due the unnecessary migration in data centre the complexity of the operation increases. Thus, it is very important to control these migrations while dealing with heterogenous workload in the data centres. The LP formulation and the heuristic approach helps in reducing this unnecessary migration by prioritizing the virtual machine. Server consolidation is very important in the data centres as it decreases the hefty electric bills. As now in the data centres the there is a need to process the heterogenous workload thus dynamic consolidation plays a critical role in the migration process. LP formulation is designed to deal with the dynamic workload and it is only performed when there is a dire need to change the capacity of the VM (Ferreto et al.; 2011).The pMapper (Verma et al.; 2008) and power expand minmax algorithm (Cardosa et al.; 2009) does not focuses on the need of migration and constraints involved while migrating the virtual machine, but LP formulation focuses on both the aspect.

While carrying the process of server consolidation it very important to take into the account of the virtual machine dependencies running on the server and the context which is running on the top of the virtual machine. Application aware virtual machine solve this issue while carrying out the migration process and it also reduce the network traffic up to 80 percent. (Shrivastava et al.; 2011). All the above existing solution does not focus much on decreasing the traffic so that network congestion. As in the pMapper (Verma et al.; 2008) and LP formulation (Ferreto et al.; 2011).The status of the physical machine is considered whereas in Application aware virtual machine (Shrivastava et al.; 2011). the status of virtual machine and the application running on it is considered.

Virtual machine allows the multiple instances to run on the same physical machine to maximize the profit. In the above paper discussed like pMapper (Verma et al.; 2008)which focuses on the estimating the cost while carrying out the migration process and LP formulationhttps (Ferreto et al.; 2011). which pays emphasis on estimating the constraints involved while migrating. The multi-objective ant-colony algorithm deals with minimizing the resource wastage and minimizing the power wastage. This approach mainly focuses on reducing the wastage of the resource while carrying out the server consolidation process. While carrying out the process of migration it happens that lot of resources are wasted if the resources are utilized to the maximum extent then the optimal solution will be obtained (Gao et al.; 2013).

As there is a dire need to process large amount of data every day by the data centres. So, it is very important that data is processed in the shortest amount of time to increase the computational speed. The energy efficient resources management system keeps track of the current resource utilization, thermal state of the nodes involved in the computation. The multi- objective ant-colony (Gao et al.; 2013) algorithm deals with minimizing the resource usage but doesnt focuses much on meeting the SLA requirement whereas this energy efficient system pays emphasis on SLA part also. The main contribution of this paper is to determine which virtual machine should be migrated so that maximum efficiency is obtained and determine the VM placement according to the current utilization. This system performs the migration process which is like adaptive threshold-based approach (Beloglazov and Buyya; 2010a) as it works monitoring the threshold limit if the CPU utilization goes below the cut-off point then VM should be migrated and machine should be shut off. In the final stage the thermal state of the nodes is observed to prevent the overheating. The state of the node is monitored continuously using sensors and if the nodes are overheated then workload is migrated in order to provide the natural cooling so that cost is reduced (Beloglazov and Buyya; 2010b)

As there is a rapid growth for the computational data in todays world. Therefore, cloud providers focus on minimizing the time to process the data. Virtualization provides solution to this problem as it creates multiple instances on the physical machine. Various consolidation schemes have been introduced but this novel approach is different as it performs migration based on the utilization history of the virtual machine. Another major contribution of this paper is that underutilized physical machine is determined, and It is SLA aware also. As discussed above very few schemes focuses on SLA part. While performing this novel algorithm first there is a detection of the underutilized physical machine the criteria for determining that is if the host has least number of virtual machine instances then it is most likely to be underloaded. This approach is quite like adaptive threshold-based approach as in both underutilized physical machines is determined, and SLA violation is prevented (Horri et al.; 2014)

Cloud computing provides on demand resources and works on pay as you model where user must pay according to the usage. Virtualization is the key for the cloud computing where multiple instances run on single machine to increase throughput. The median based threshold approach is very similar to adaptive threshold approach where the consolidation is performed based on the set median value. If the CPU utilization goes below the median value, then the machine is underutilized, and virtual machine should be migrated. If the CPU utilization goes above the median value, then the machine is overutilized and the resource should be balanced. The main drawback of this algorithm approach is that SLA violation is not prevented as it is done in the above novel approach (Sharma and Saini; 2016)

The virtual machine placement in the data centre is a NP hard problem. Many algorithms have been proposed to reduce the VM migration as consolidation process consumes lot of energy. In this paper energy aware scheme has been proposed which compromises of two algorithm and it is similar to the application aware placement scheme. The main aim of this paper is to reduce the number of idle hosts as the underutilized host consumes lot of energy. This energy efficient scheme performs exact migration which is required using bin-packing extension and best-fit heuristic. All the needed information to run the algorithm and perform the VM migration is extracted from cloud Iaas manager. The simulation result shows using the energy efficient scheme the power reduction can be approximately 46 percent during the encounter of dynamic workload (Ghribi et al.; 2013).

Power consumed by the data centres is the main area of concern in industry as well as academic work. But little emphasis has been paid on reducing the network power which is waisted during the consolidation process. The VMplanner aims on reducing the network traffic so that power consumption can be reduced and takes advantage of the flexibility provided by the heterogenous workload during migration. While carrying out the migration process in the VMplanner the network element and dependencies are considered. As the traffic is known by the VMplanner therefore it becomes easy to carry out the migration process (Fang et al.; 2013).

The virtual machine migration is a major concern in a data centre while dealing with the heterogenous workload. The proposed novel multi-criteria decision making is used to select the virtual-machine and Window moving average for the detection of the underutilized physical machine. The detection of underloaded PM is very important in the datacentre as this host causes lot of energy to waste. The maximum request resource (MRR) policy is used for the limiting the SLA violation. The MRR eliminates hotspot happened by CPU shortage in a physical machine which in turn decrease the SLA violation. All the proposed algorithm in this paper works on the principle on decision tree as it evaluates the history of the usage and it doesnt work on current CPU utilization. This proposed algorithm is some what similar to adaptive threshold based approach as in this also the underutilized physical machine is calculated but in this proposed algorithm maximum request resource policy prevent the SLA violation which is not taken care of in the other algorithm. Furthermore, the resource type is also determined, and it is efficient in handling the heterogenous workload (Arianyan et al.; 2016).

Resource management the data centre is very important in the data centre, but not much work has been done on this. The proposed resource management scheme work focuses on decreasing the energy consumption by determining the VM request beforehand, number of hosts required to cater that request and switching off the unused hosts. This resource management scheme proceed in three steps first is k-mean clustering which maps the virtual machine to the physical machine based on the request it receives. Second is workload prediction which happens through the wiener filter as the resource type are predicted through the observation window. Third is the power management which uses the modified best-fit decreasing to estimate the workload and if the idle host is determined then it is switched off the above approach is similar to this as both of them works on predicting. This approach is different from the above approach because in this virtual machine request is predicted rather than the CPU utilization rate (Dabbagh et al.; 2014).

Quality of service is also a major issue in the cloud computing that means the throughput should be high and job failure should be low. One unique approach based on the learning automata to improve the quality in Iaas which is a cloud type. The experiment result shows that response time and speed up increases because of using the LA and it is self-learning model gains experience from the past so that it can be applied into the environment efficiently. The components in the LA are automation, environment, reward and penalty. The initial probability of all the actions are equal and then with the comparison with threshold value reward and penalty is awarded (Misra et al.; 2014)

A learning automata-based algorithm is used for the resource management and detection of overloaded physical machine in a data centre. After detection of the overutilized host the workload can be migrated. The underloaded physical machine is also determined using this algorithm and after determining it can be switched off. The learning automata uses the reinforcement algorithm as it uses the data which it receives and applies it back to the environment and this algorithm is similar to the above LA algorithm but it is applied to cater different problem in cloud which is quality of service. The overloaded host is determined based on the history of the CPU utilization and then after determining the consolidation of virtual machine can be performed. As the virtual machine migration on the physical machine is a NP hard problem. The learning automata updates itself after receiving the input and the SLA violation is also reduced to a great extent as shown in the figure below LA updates according to each workload provided by the planet lab. The future work of this paper is to determine the underutilized physical machine in a data centre and my research work also focuses on determining the underloaded PM (Ranjbari and Akbari Torkestani; 2018).

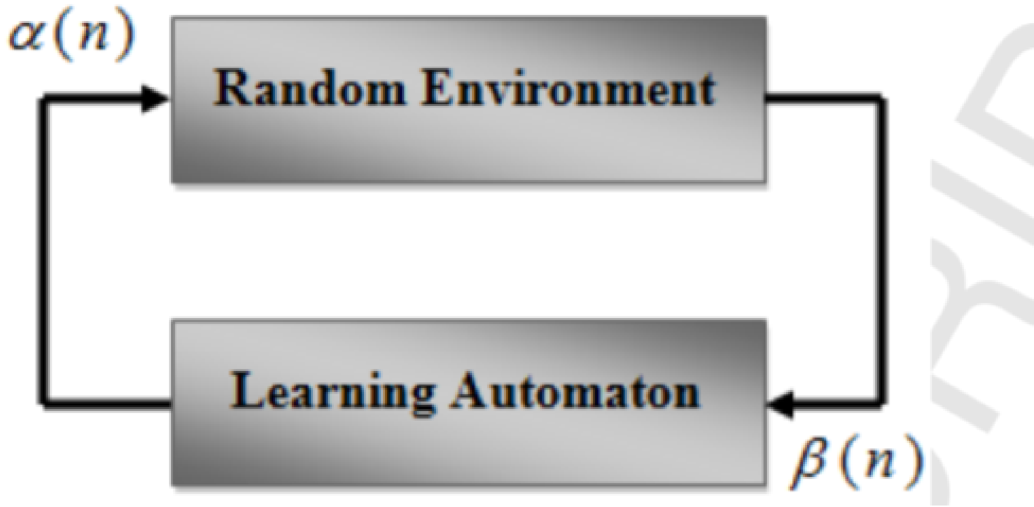


Figure 1: Learning automata algorithm (Ranjbari and Akbari Torkestani; 2018)

Approach	Contribution
Dynami workflow novel algorithm	Dynamic work load management
Pmapper Algorithm	Cost of vm migration
Minmax share	uses minmax share of virtualization
Adaptive threshold algorithm	Migration based on threshold value
LP formulation and heuristic	reduces unnecessary migration
Application aware vm	reduces the dependency while migration
Multi objective ant colony	minimize resource usage and power
Adaptive threshold approach	uses mean value for migration
Energy aware scheme	reduces no. of host
vm planner	reduces network traffic
Maximum request resource	reduces SlA violation while migration
Learning automata	determines the overloaded host

Figure 2: Contribution of previous research work

3 Methodology

The methodology is the steps and method or the approach that has been used to carry out the simulation process. The steps which have been applied are simple to perform. The main aim to carry out the project was to minimize the energy consumption of the data centres as lot of energy is being used to run and cool the system. The major driving force behind this project are-

1. Increasing demand for the computing resources

2. Improper management of the host in the data centers

3. Limited work done on determining the underloaded host

After considering all the above listed points the modified novel algorithm based on learning automata has been developed which focuses on the principle automata where changes the states and move from one state to another. The input for the algorithm is the data set is provided by the user or the cloud sim simulator. For carrying out the simulation process the cloud sim is used. The cloud sim is very popular for setting up the real cloud environment as it has predefined classes for the virtual machine, data centre, cloudlet and host. The code in the cloud sim is written in java and it extends the classes which is provided by the cloud sim 3.0.3. The simulation result is displayed on the swing page by passing the parameter. For the storage of the result MySQL has been used.

4 Design Specification

4.1 Cloud Sim

The cloud sim cloud simulator is a framework which is used for modelling as well as simulating the services. The code in the cloud sim is written in java. The main advantage of cloud is that it can simulate real time cloud entities such as data centre, virtual machine, physical machine, broker. The cloud sim provides an opportunity to concentrate on the algorithm rather than the paying emphasis on the hardware details (Shi et al.; 2011). The entities or predefined classes provided by the cloud sim which is used in the project are as follows.

1. Data center- It set the hardware infrastructure provided by the cloud providers.
2. Host- It set the physical server on which virtual machine instances are set.
3. Virtual machine- It set the VM on the cloud host to deal with the cloudlet.
4. Cloudlet- It set the cloud application-based services.
5. VmAllocation- It is set the provisioning policy for VM to allocate on the host.
6. VmScheduler- It schedule the VM on the host which run

on the data centre (Shi et al.; 2011).

This is the cloud sim simulator architecture and the lower layer contains SimJava discrete event simulation engine that implements the core functionality required to manage the high level concepts such as creating the core components (hosts, virtual machine, data centre and cloudlet) and communication between the above mentioned component. The libraries are implemented using the gridsim toolkit. The cloud sim is implemented programmatically by extending the lower gridsim layer. The cloud sim is the major layer in the cloud sim framework which is used to manage the real computing environment by providing the VM, PM and data centre. The issue is provisioning according to the user request. The topmost layer is the users code layer that writes the configuration such as number of hosts, storage, requirements and specification and so on.

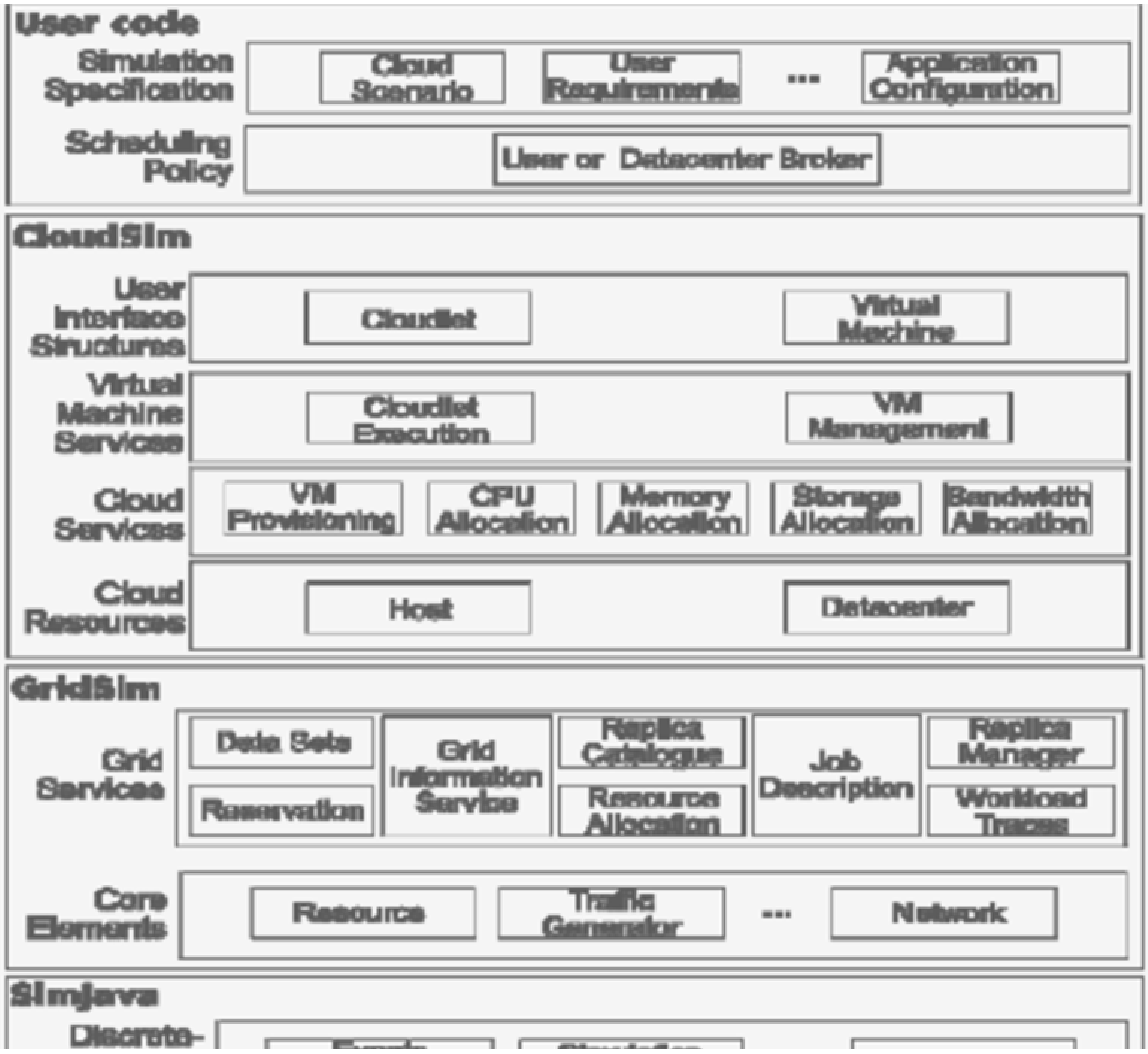


Figure 3: Layered architecture of cloud sim (Buyya et al.; 2009)

4.2 CloudSim Simulation Data Flow Process

This diagram below depicts how the core entities communicate among themselves. Initially each data centre register itself in the cloud Information service (CIS). The CIS is used to match the users requests to the authentic cloud provider (Buyya et al.; 2009). After all this CIS communicates with the broker to execute and perform the relevant operation. The data centre broker communicates regarding execution process with the CIS and count the quality of service. When the user request comes the resources are available to the VM and then the allocation is performed according to the requirement (*Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers - Beloglazov - 2012 - Concurrency and Computation: Practice and Experience - Wiley Online Library*; n.d.).

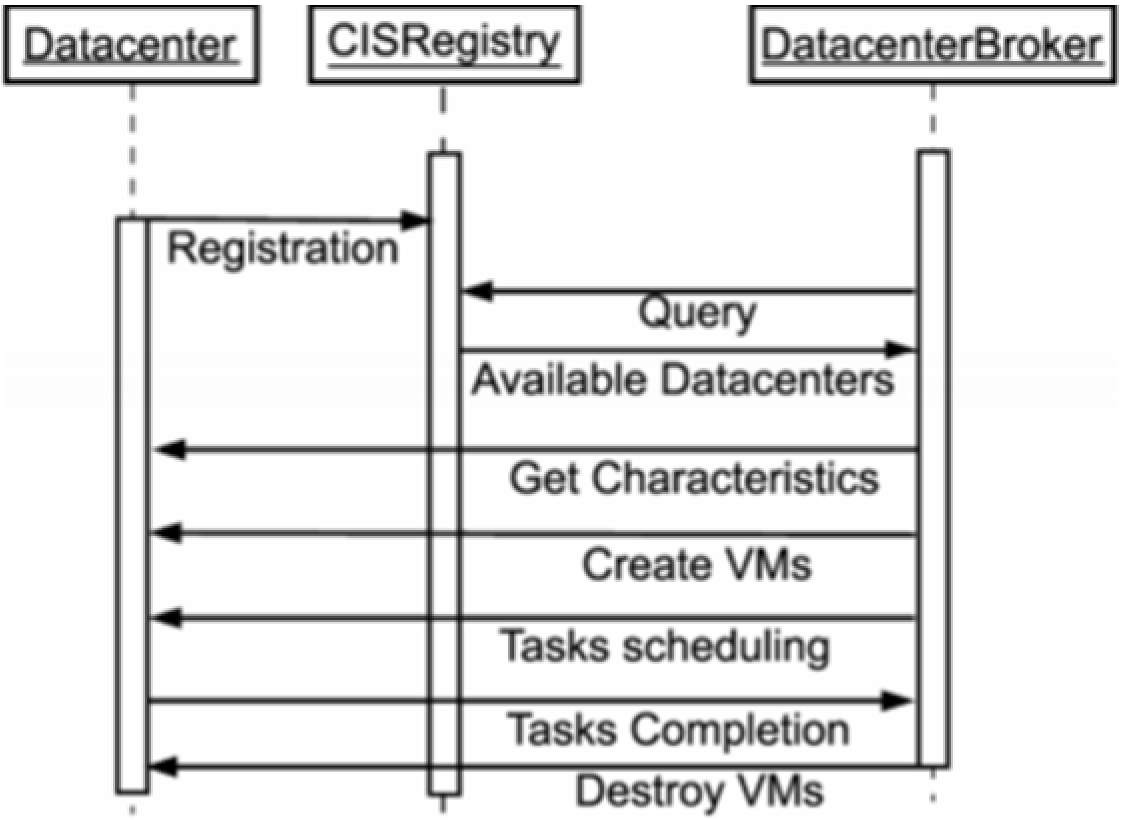


Figure 4: Flow of control in cloud sim (Buyya et al.; 2009)

4.3 Flow Diagram

The data set is selected from the planet lab data set. There are lot of data set in the planet lab workload traces it is up to the user to set the data. After the data set has been set user needs to set up the server. There are two server option available to the user one is HP ProLiant ML110 G4 and other is HP ProLiant ML110 G5. The number of cores, million instruction per second and storage is available dynamically after the selection of respective server. The most important is the selection of host three options are available such as 600,700,800. Then the helper class performs the consolidation process based on the VM, PM and all the mentioned specification. The result is displayed in the LAUD class.

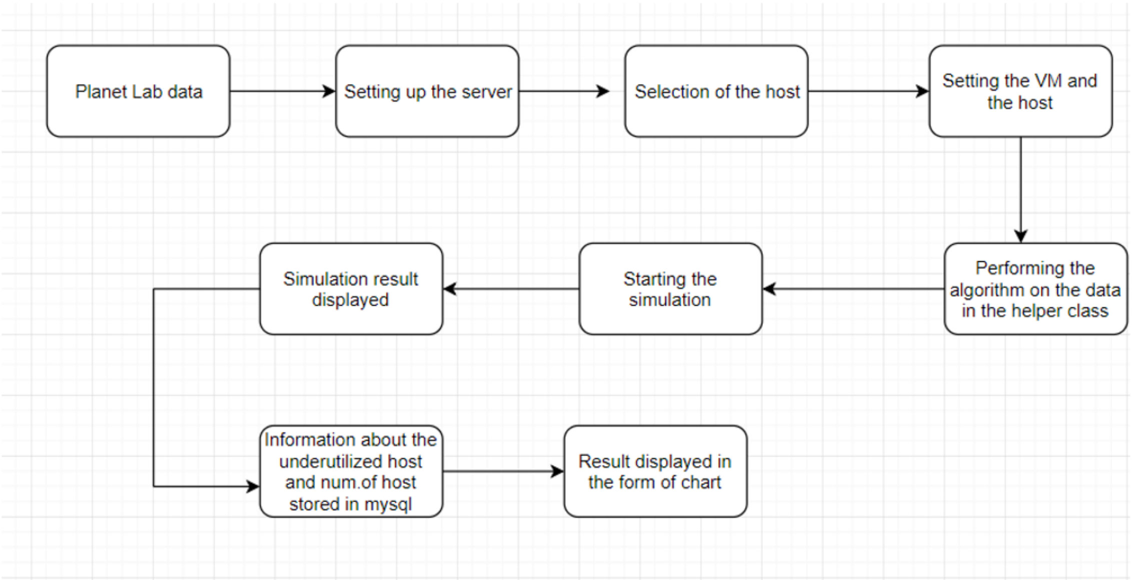


Figure 5: Flow diagram (*draw.io*; n.d.)

5 Implementation

Tools Used
Netbeans 8.1
CloudSim 3.0.3
Front end- Java swing
Database-MySQL 5.5.40
Operating system- Windows 10

DataCenter Configurations
MIPS- 1860,2660(Heterogeneous)
RAM-4 GB
Storage- 1000000
Bandwidth- 1500000
Number of hosts-600,700,800

Virtual machine
MIPS- 2500,2000,1000,500(Heterogeneous)
RAM- 870,1740,1740,613
Storage-10000
Bandwidth-2500

5.1 coding

As the main of this paper is to determine the underutilized physical machine is a data center. Therefore, the LA algorithm has been used to determine the underloaded PM and at the same insuring that minimum SLA violation happened. The coding is done using the java as a programming language. The cloud sim 3.03 has been used as it provide the facility of VM selection and VM allocation and it also provides the power model from where the two HP server has been used to perform the simulation. Another important class provided by the cloud sim is the utilization history model which is extended in the learning automata project for carrying out the computation process. The cloud sim also helps in setting up the cloudlet, virtual machine and data center in project.

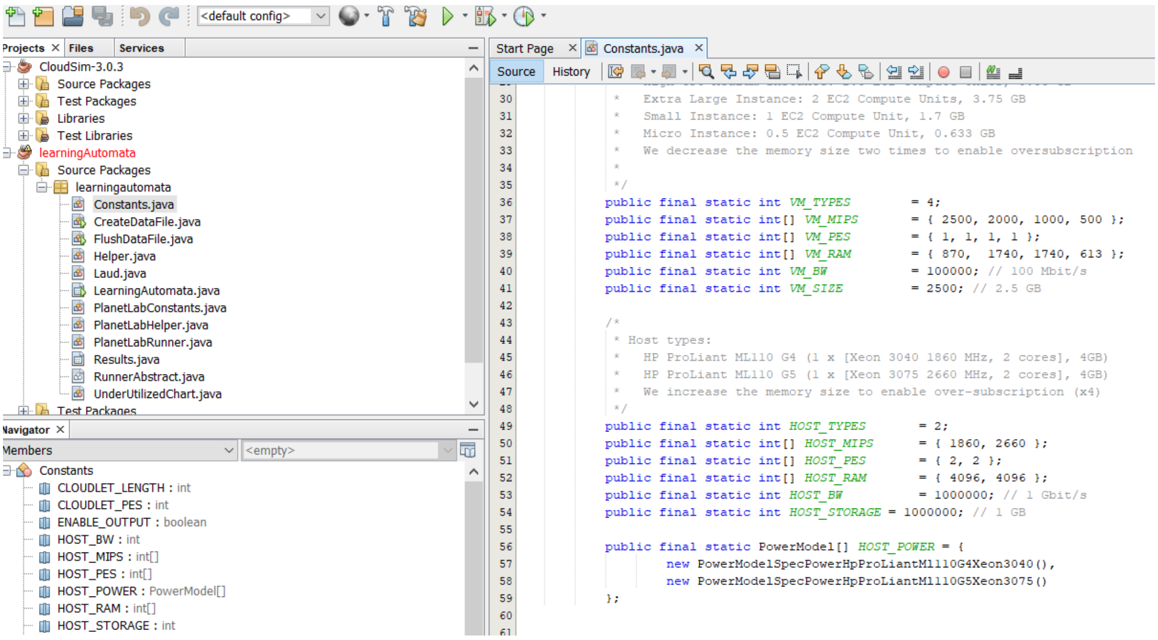


Figure 6: Setting the configuration of VM and PM

As seen in the above screen shot the VM and host is being set up using the power model of the cloud sim. The specification of the VM is being specified such as type, bandwidth, ram and size etc. Furthermore, the host characteristics are also listed, and host storage is also listed which is 1 GB. There are around twelve classes in the project.

5.2 Database

For the database MySQL is being used where the information about the host is being stored three host are used in the project and the number of host is 600, 700 and 800 respectively.

As shown in the screen shot that the host is set up in the class planetLabConstants. The data which is stored in the database consists of two table one is number of host and the other is underutilized host. Below is the code for setting up the database connectivity in the create Datafile class in the project and the username is root and password is blank. The Exception is also handled using try catch. The table name is LAUDhost where it consists of two columns.

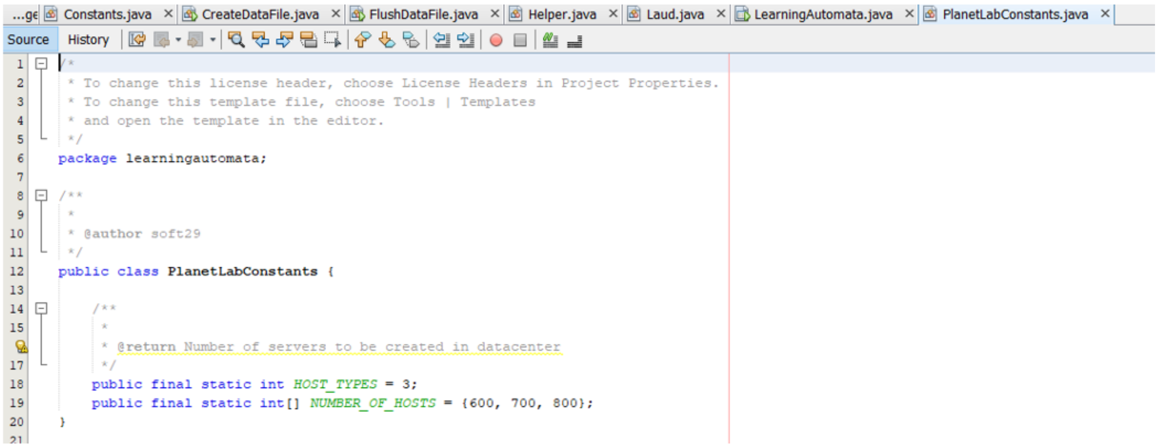


Figure 7: Defining the host number

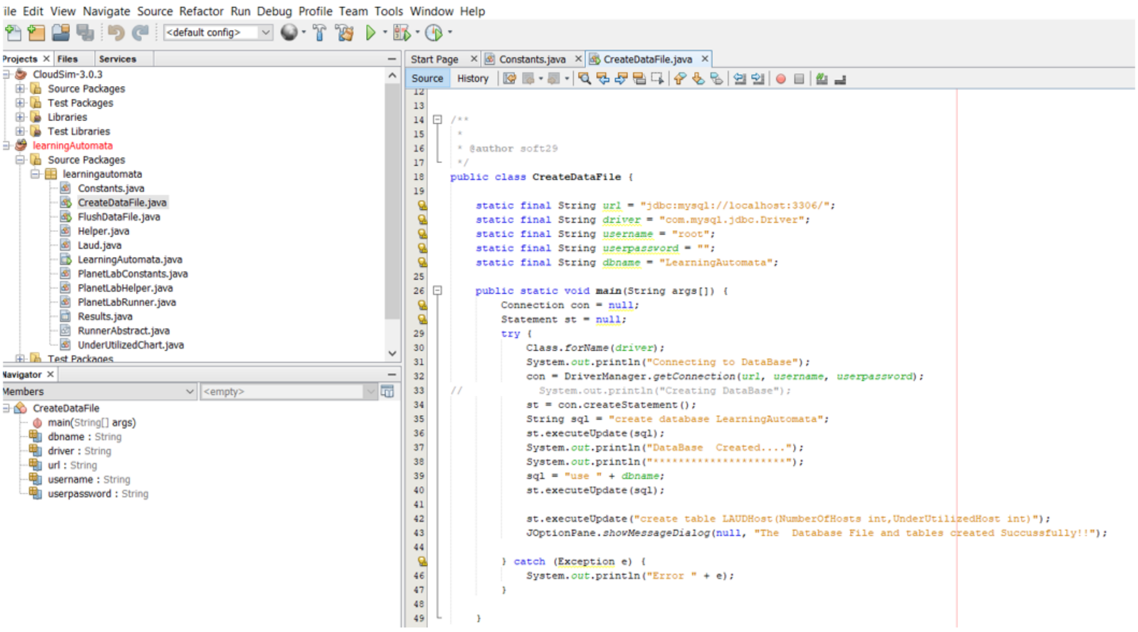


Figure 8: Setting up the database connection

5.3 Consolidation process

The result provided by this algorithm is the number of over-utilized, underutilized, balanced and idle hosts which is computed in the consolidation process only.

As shown in screen shot above in the class helper.java the over-utilized, underutilized, balanced host and the energy consumed or required or required for carrying out the proceed. In addition to this the SLA violation and overall SLA violation is calculated in this class. The helper is required while dealing the VM selection and VM allocation. The cloudlet is also created in the helper class.

As shown in the screen shot above the time before migration and the time after the migration is calculated in this. All the cloud sim classes are imported into this class which will be needed while performing this algorithm such as power model, VM, storage, host and cloudlet etc. The output or the simulation is carried out because of this class only as it imports all the class, performs computation and finally print the output. If the simulation is carried out successfully then success it printed along with the simulation

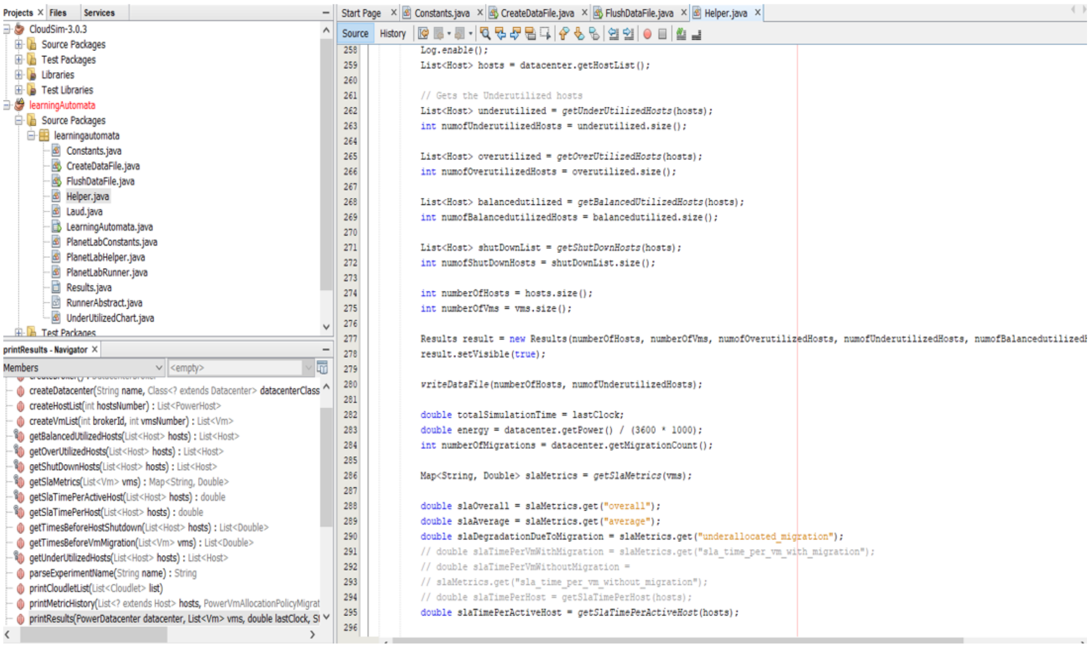


Figure 9: Algorithm code

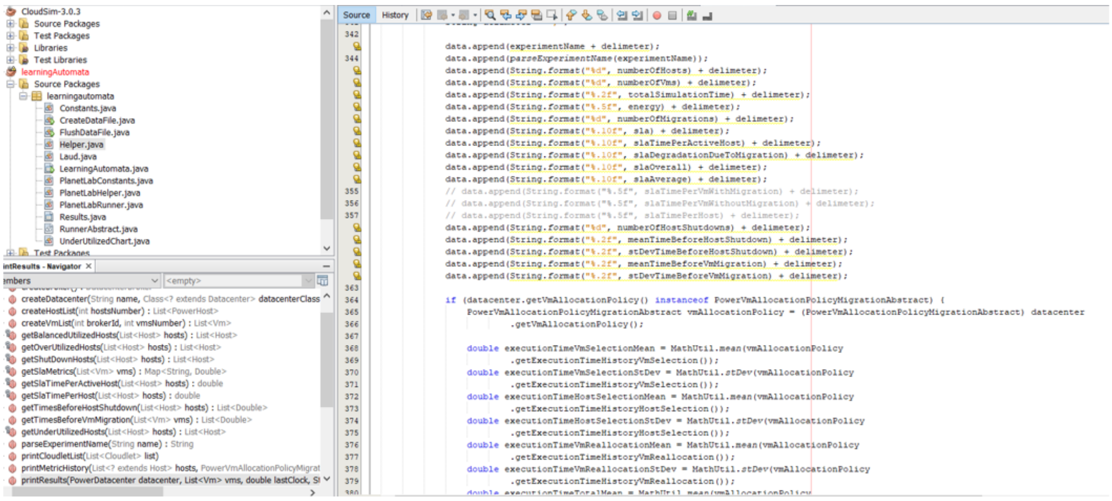


Figure 10: VM allocation and selection

time.

If the simulation process is carried out successfully then the above result will be displayed. The number of hosts can be set up by the user as there are three option for the user and for the workload any of the csv file from the planet lab workload data can be chosen according to the date. Total simulation time and energy consumption which is 19.14 kwh. The overall SLA violation as well average SLA violation is also listed in the result provided by the simulation.

5.4 Configuration

The two servers are chosen first one is HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores 1860 MHz, 4 GB), and the second one is HP ProLiant ML110 G5 (Intel Xeon 3075, 2

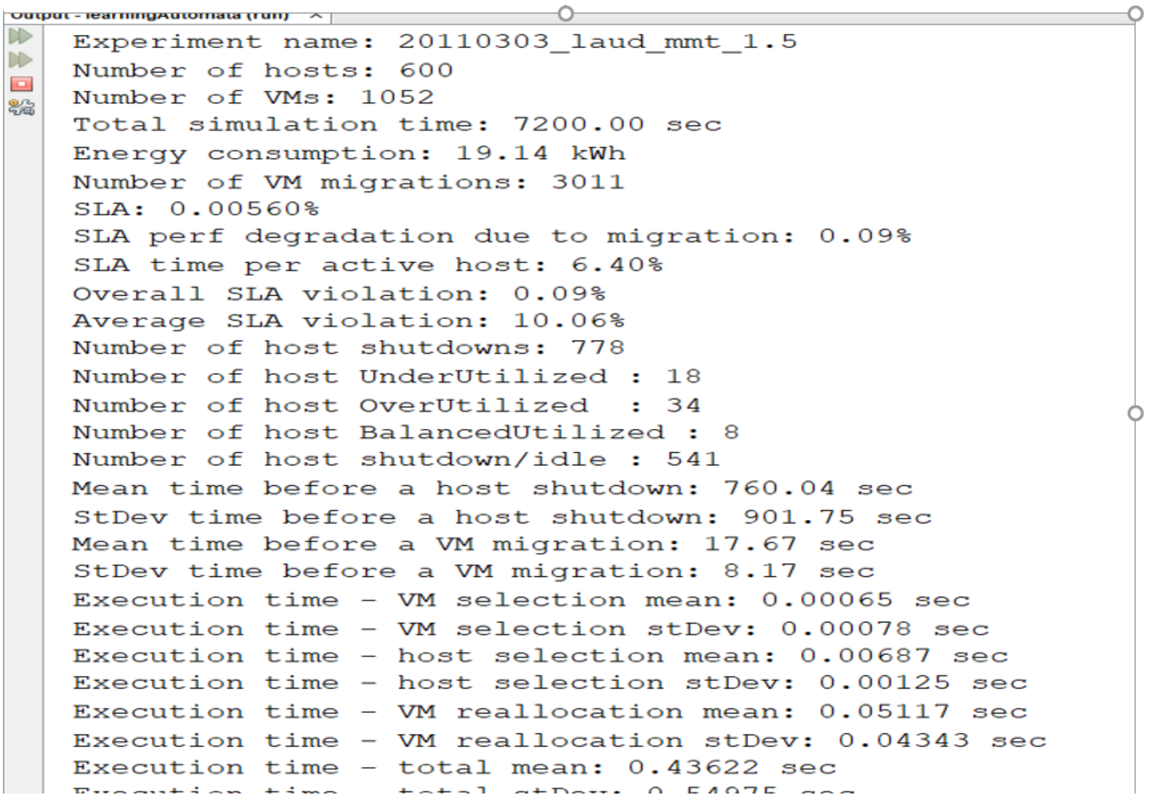


Figure 11: Simulation result

cores 2660 MHz, 4 GB). As shown in the figure below when the user chooses any of the server listed the corresponding cores, million instruction per second and ram gets displayed. The number of host is 600,700 and 800 from where user gets the option to select according to his need.

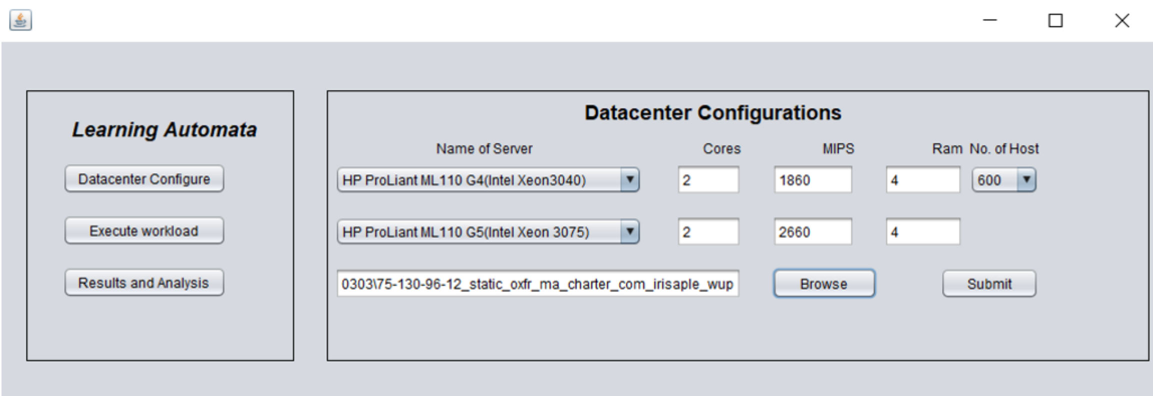


Figure 12: Setting up the data center configuration

5.5 Dataset

The planetlab data of CPU utilization is taken. The data is according to the date and the large data is present in the csv file. The workload data is taken by the user for performing the computation. As shown in the figure below the data of the 3/03/2011 and 06/03/2011 is provided by the cloud sim and these contains the traces of different workload.

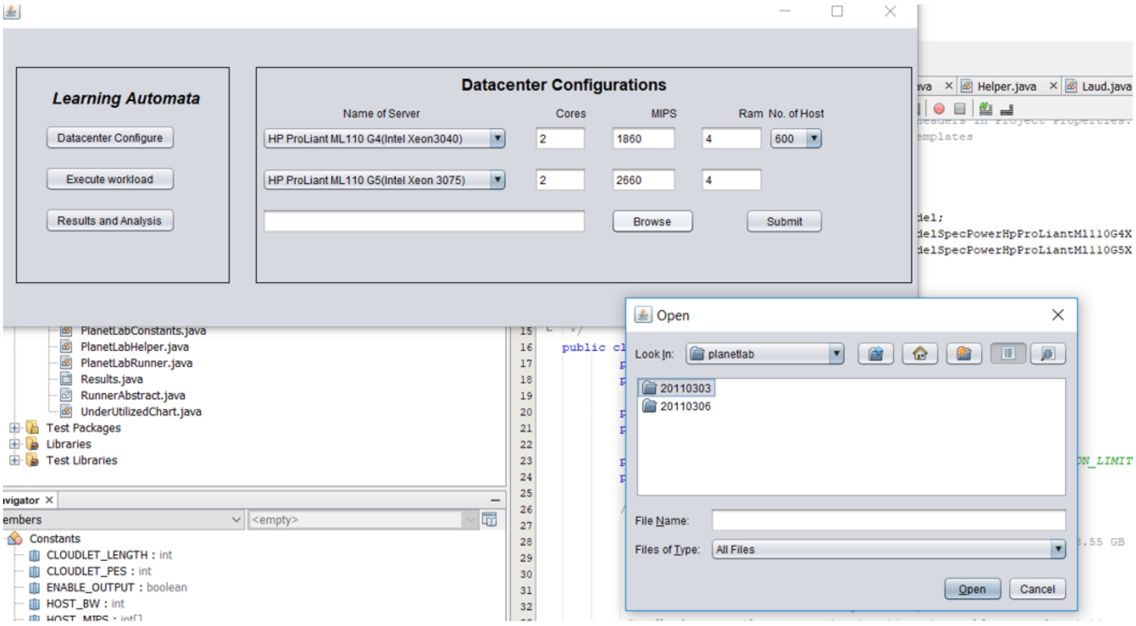


Figure 13: Sellecting the data set

Pseudo code

Pesudo Code

1. Configure the Learning Automaton System environments
2. For i = 1 to n
3. Initialize and assigns an automaton to each user.
4. At first iteration, the user's VM will not be active.
5. Set Actions={ASC, DESC, NONE}
6. If i<>1 select Actions for each VM
7. Determine the utilization of PMs and find PM states Underload/Overload (It describes in **Algorithm-A**).
8. If pmStates="Overload"
9. Select one or more VM(s) to migrate from the PM, the vm selection based on the minimum migration time policy, which selects vms due to the lowest required time for migration.
10. Then a PM is selected to which the VM should be migrated.
11. The PM selection first sorts the list of Pms based on utilization reduction and then selects.
12. Else if pmStates="Underload" then all VM can be migrated to another one.
13. This stage also selects an appropriate value for VMs based on increase in power (the pesudocode describes in **Algorithm-B**).
14. At final, the learning automata is updated for each VM, according to selected action in previous iteration and current CPU utilization (It Describes in **Algorithm – C**).
15. endfor

Figure 14: LA

This is the pseudo code of the learning automata used in the project to determine the underutilized physical machine in a data centre. In the theory of the automata the input is given, and the states changes for the input and then according the specified condition the transition is performed for the value chosen (Narendra and Thathachar; 1974)

The states chosen are overloaded, underloaded. The actions are ASC (average increasing CPU utilization), DESC (average decreasing CPU utilization) and NONE (not increasing CPU utilization) (Ranjbari and Akbari Torkestani; 2018).

Algorithm- A : For detection of PM states.

Input: PM
Output: overload/underload/balanced

```
predictedUtilization  ←  0
utilization  ←  0
pmStates  ←  “balanced“
foreach VMi in a PM
     utilization += utilization of VMi
if selected action of LAi = ASC
     predictedUtilization += Avgi + deviation
if selected action of LAi = DESC
     predictedUtilization -= Avgi   −   deviation
end if
end for
if(utilization > SafetyThreshold && predictedUtilization > (1 - SafetyThreshold))
     pmStates=”overload”;
Else if(utilization between 40 to 60% of SafetyThreshol
     && predictedUtilizaiton<(1 - SafetyThreshold));
     pmStates=”balanced”;
Else if(utilization between 15 to 40% of SafetyThreshold
     && predictedUtilizaiton<(1 - SafetyThreshold));
     pmStates=”underload”;
Else
     pmStates=”idle”;

return pmStates;
```

Figure 15: PM state detection

The initial probability of all the action is set equal and reward is awarded bases on the utilization.

The PM states in the most important thing in this algorithm for detection of underutilized host. If there is an increase in utilization, then ASC and if there is a decrease in the utilization then DESC. The detection of the balanced, overloaded, underutilized and idle host is based on the threshold values. The threshold values are set and accordingly computation is performed (Ranjbari and Akbari Torkestani; 2018)

```
Alogorithm B: Power Aware Best Fit Decreasing(PABFD) VM placement

Input: hostList, vmList
Output: allocation of VMs
vmList.sortDecreasingUtilization()
foreach vm in vmList Do
    minPower ← MAX
    allocatedHost ← Null
foreach host in hostList do
    If host has enough resources for vm then
        Power ← estimatePower(host, vm)
        If power < minPower then
        allocatedHost ← host
        minPower ← power
If allocatedHost <>Null then
    Allocation.add(vm,allocatedHost)
Return allocation
```

Figure 16: VM placement algorithm

The power aware best fit algorithm is used to select the PM to which the VM selected needs to be migrated. As it is clear the main purpose of performing this algorithm to place a VM. The PM is sorted in the decreasing order. The conditions are specified in the algorithm to perform the consolidation (*Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers - Beloglazov - 2012 - Concurrency and Computation: Practice and Experience - Wiley Online Library*; n.d.)

```
Algorithm C:    To update learning automata

Input: List of PM
Output:

foreach PMi in list of PMs do
foreach vmij in PMi do
    avgij ← Get average of CPU utilization
if avgij = CurrentUtil ij && PreviousAction ij is NONE
    reward NONE action of LAij
else if avgij !=CurrentUtil ij && PreviousAction ij is NONE
    penality NONE action of LAij
if avgij < CurrentUtil ij && PreviousAction ij is ASC
    reward ASC action of LAij
else if avgij < CurrentUtil ij && PreviousAction ij is DESC
    penality DESC action of LAij
else if avgij > CurrentUtil ij && PreviousAction ij is ASC
    penality ASC action of LAij
else if avgij > CurrentUtil ij && PreviousAction ij is DESC
    reward DESC action of LAij
end if
    avgij ← calculate average of currentTime utilization
    deviation ij ← calculate deviation of utilization
end for
end for
```

Figure 17: update the LA

This is used to update the learning automata algorithm based on the utilization fetched. If increasing utilization ASC is awarded and if DESC then decreasing utilization.

6 Evaluation

6.1 Simulation

The simulation process allows to run the project in the real environment. The cloud sim is used for the simulation and for this cloud sim jar 3.03 and math jar is used. The cloud sim jar is inserted in the project to carry out the simulation process and the predefined classes of the cloud sim is used in this project the VM allocation, VM selection, power model, utilization model. In addition to this workload is also provided by the cloud sim. As shown in screen shot in source package all the class and workload data set needed as there. To use these classes of cloud sim in the learning automata the classes are extended. The helper is the main class where cloudlet is set and the code for carrying out the consolidation process is written. In the constant.java the simulation time is also defined.

This above screen shot shows how the simulation process is being carried in this the VM allocation is being performed and for every host the VM allocation and selection is performed as shown in the above figure for the host 55,56 and 57 is being performed. The million instructions per second is also displayed and the simulation time depends upon the data set.

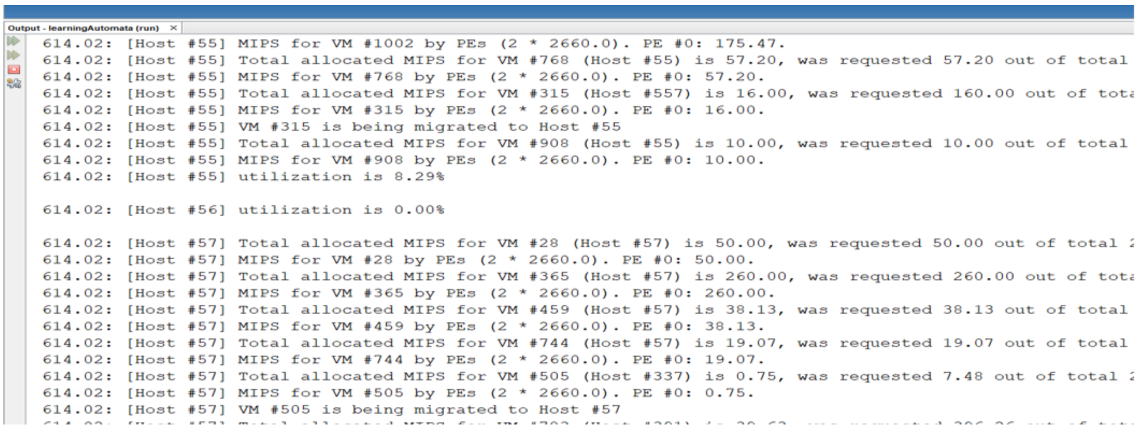


Figure 18: Simulation process

6.2 Result

After submitting the data simulation process gets started where the various VM allocation and VM selection process is carried out. This process is carried of deterring the underutilized, balanced and overloaded host is carried out in the helper class and VM allocation and selection in the abstract runner. The front end is designed using swing. As shown in the screen shot the result is displayed showing the number of hosts in this 600 has been chosen. The number of virtual machines running on that physical machine is 1052. The overutilized host is 34 and it should be shifted. The underutilized host should be shifted so that power consumption be minimized. The balanced host is the best because it doesnt need to shift or switched off. The idle host also need to turn off as it utilizes the energy.

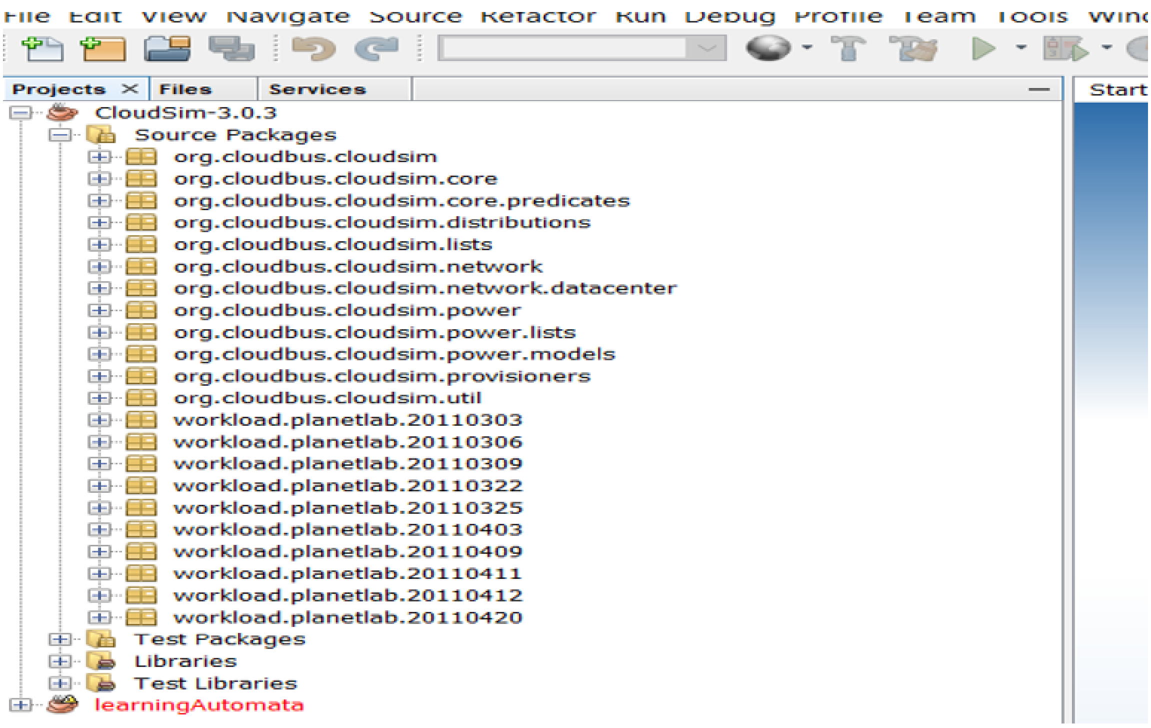


Figure 19: Cloudsim classes and workload

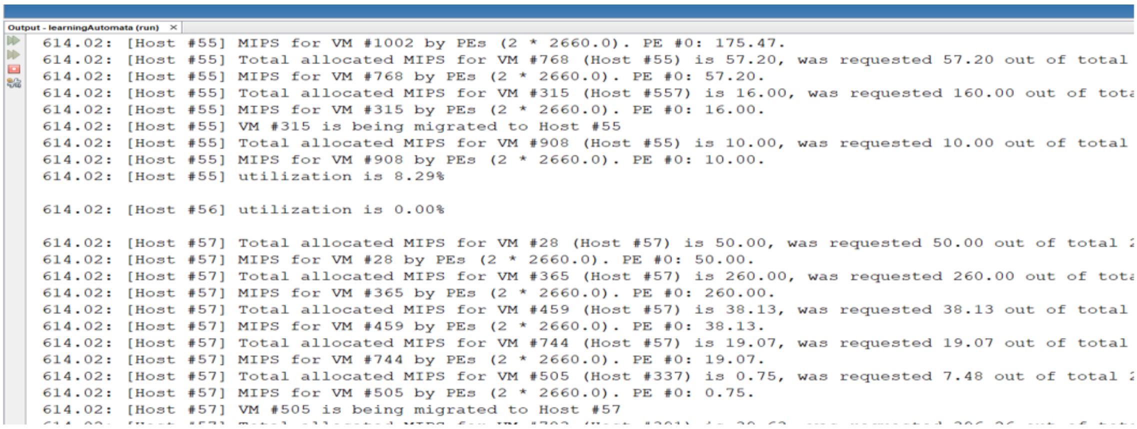


Figure 20: Simulation process

6.3 Chart

After the showing the result chart will be displayed next. For displaying the chart, the button on the result need to be pressed. The chart displays the number of hosts and the underutilized host. In the flushdatafile.java all the other data is deleted and only the information about the number of host and underutilized host is stored. The table name is LAUDhost where the two column is created.

As shown in the screen shot above the number of underutilized hosts for 600 in the initial phase is 14 and for 800 host it is also 14. The underutilized host was minimum for 700 host. The chart value is displayed using swing and these are stored in flushdatafile.java and the value from this is passed to underutilizedchart.java.

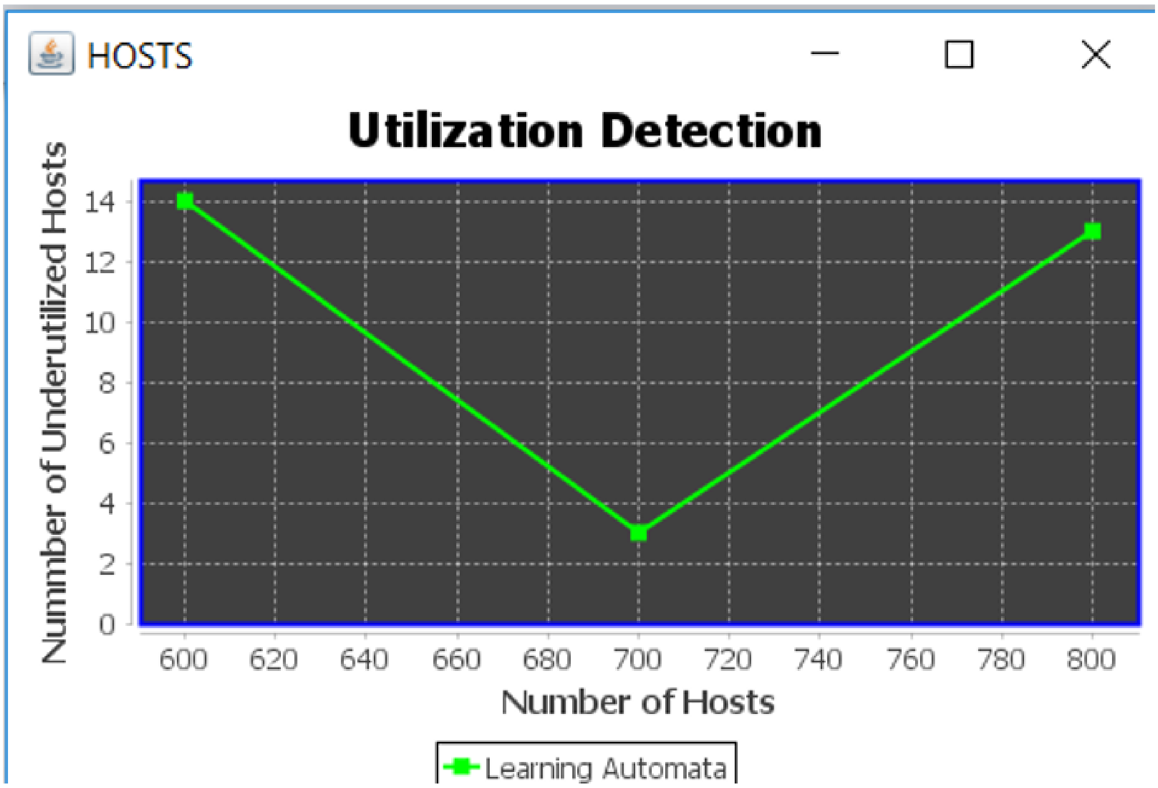


Figure 21: Number of host vs underutilized hosts

6.4 Discussion

The novel algorithm which has be proposed successfully helps in determining the underutilized physical machine in a data centre. The energy consumed by the data centre and by the application of the learning automata algorithm for determining the underloaded host and power aware best fit algorithm for placing the virtual machine in an appropriate data centre. The power consumption can be minimized to a great extent. The cloud sim algorithm helps in the setting up the necessary hardware required to run the algorithm. The result obtained shows the computed data which contains the information about the underutilized, overloaded, balanced and idle hosts. The percentage of SLA violation is also computed at the end of the simulation. A final chart will be displayed which contains information about the number of host and underutilized host.

7 Conclusion and Future Work

This study proposed the novel algorithm based on learning automata to determine the underutilized physical machine in the data center corresponding to the number of hosts used. The power aware best fit algorithm is used place the VM after detection of the underloaded host. The energy consumption of the data center is the major concern for the cloud provider. The hefty bills form the data center is not only the one major issue another problem which arises due to the underutilized physical machine is the emission of the green house gases from the data center which pollutes the environment. For the detection of the balanced and underloaded host the threshold values are used and if the value falls within the range the state of the host is determined. The learning automata works on the principle of the automata where the input is applied, and the transition of the input is from state to state to determine the output.

The future of this paper might include determining of underutilized based on the dynamic values other than the threshold values. As in the algorithm used the fixed values are used for determining the host. In addition, to this the topology of the network, cooling systems these all issues need to be considered.

8 Acknowledgement

I am very thankful to my supervisor Manual Tova-Izquierdo for his continuous support. Sir helped me and guided me throughout the development of my project. Without his support and tremendous believe on me. I wouldn't be able to develop the project.

References

Aazam, M. and Huh, E. (2014). Broker as a Service (BaaS) Pricing and Resource Estimation Model, *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pp. 463–468.

Arianyan, E., Taheri, H. and Sharifian, S. (2016). Novel heuristics for consolidation of virtual machines in cloud data centers using multi-criteria resource management solutions, *The Journal of Supercomputing* **72**(2): 688–717.
URL: *https://doi.org/10.1007/s11227-015-1603-9*

Beloglazov, A. and Buyya, R. (2010a). Energy Efficient Allocation of Virtual Machines in Cloud Data Centers, *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, IEEE, Melbourne, Australia, pp. 577–578.
URL: *http://ieeexplore.ieee.org/document/5493430/*

Beloglazov, A. and Buyya, R. (2010b). Energy Efficient Resource Management in Virtualized Cloud Data Centers, *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, IEEE, Melbourne, Australia, pp. 826–831.
URL: *http://ieeexplore.ieee.org/document/5493373/*

Bobroff, N., Kochut, A. and Beaty, K. (2007). Dynamic Placement of Virtual Machines for Managing SLA Violations, *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, pp. 119–128.

Buyya, R., Ranjan, R. and Calheiros, R. N. (2009). Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities, *2009 International Conference on High Performance Computing & Simulation*, IEEE, Leipzig, Germany, pp. 1–11.
URL: *http://ieeexplore.ieee.org/document/5192685/*

Cardosa, M., Korupolu, M. R. and Singh, A. (2009). Shares and utilities based power consolidation in virtualized server environments, *2009 IFIP/IEEE International Symposium on Integrated Network Management*, pp. 327–334.

Chaisiri, S., Lee, B. and Niyato, D. (2012). Optimization of Resource Provisioning Cost in Cloud Computing, *IEEE Transactions on Services Computing* **5**(2): 164–177.

Dabbagh, M., Hamdaoui, B., Guizani, M. and Rayes, A. (2014). Energy-efficient cloud resource management, *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, Toronto, ON, Canada, pp. 386–391.
URL: *http://ieeexplore.ieee.org/document/6849263/*

draw.io (n.d.).
URL: *https://www.draw.io/*

Fang, W., Liang, X., Li, S., Chiaraviglio, L. and Xiong, N. (2013). VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers, *Computer Networks* **57**(1): 179–196.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1389128612003301*

Ferreto, T. C., Netto, M. A., Calheiros, R. N. and De Rose, C. A. (2011). Server consolidation with migration control for virtualized data centers, *Future Generation Computer Systems* **27**(8): 1027–1034.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0167739X11000677*

Gao, Y., Guan, H., Qi, Z., Hou, Y. and Liu, L. (2013). A multi-objective ant colony system algorithm for virtual machine placement in cloud computing, *Journal of Computer and System Sciences* **79**(8): 1230–1242.
URL: *http://www.sciencedirect.com/science/article/pii/S0022000013000627*

Ghribi, C., Hadji, M. and Zeghlache, D. (2013). Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms, *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, pp. 671–678.

Horri, A., Mozafari, M. S. and Dastghaibyfard, G. (2014). Novel resource allocation algorithms to performance and energy efficiency in cloud computing, *The Journal of Supercomputing* **69**(3): 1445–1461.
URL: *https://doi.org/10.1007/s11227-014-1224-8*

Java SE Development Kit 10- - Downloads (n.d.).
URL: *https://www.oracle.com/technetwork/java/javase/downloads/jdk10-downloads-4416644.html*

Menaud, J., Van, H. N. and Tran, F. D. (2010). Performance and Power Management for Cloud Infrastructures, *2010 IEEE 3rd International Conference on Cloud Computing(CLOUD)*, pp. 329–336.
URL: *doi.ieeecomputersociety.org/10.1109/CLOUD.2010.25*

Misra, S., Krishna, P. V., Kalaiselvan, K., Saritha, V. and Obaidat, M. S. (2014). Learning Automata-Based QoS Framework for Cloud IaaS, *IEEE Transactions on Network and Service Management* **11**(1): 15–24.

Narendra, K. S. and Thathachar, M. A. L. (1974). Learning Automata - A Survey, *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-4**(4): 323–334.

NetBeans IDE Download (n.d.).
URL: *https://netbeans.org/downloads/*

Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers - Beloglazov - 2012 - Concurrency and Computation: Practice and Experience - Wiley Online Library (n.d.).
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.1867*

Ranjbari, M. and Akbari Torkestani, J. (2018). A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers, *Journal of Parallel and Distributed Computing* **113**: 55–62.
URL: *http://www.sciencedirect.com/science/article/pii/S074373151730285X*

Sharma, O. and Saini, H. (2016). VM Consolidation for Cloud Data Center Using Median Based Threshold Approach, *Procedia Computer Science* **89**: 27–33.
URL: *http://www.sciencedirect.com/science/article/pii/S1877050916310705*

Shi, Y., Jiang, X. and Ye, K. (2011). An Energy-Efficient Scheme for Cloud Resource Provisioning Based on CloudSim, *2011 IEEE International Conference on Cluster Computing*, pp. 595–599.

Shrivastava, V., Zerfos, P., Lee, K., Jamjoom, H., Liu, Y. and Banerjee, S. (2011). Application-aware virtual machine migration in data centers, *2011 Proceedings IEEE INFOCOM*, pp. 66–70.

Verma, A., Ahuja, P. and Neogi, A. (2008). pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems, *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, Middleware '08, Springer-Verlag New York, Inc., New York, NY, USA, pp. 243–264.
URL: *http://dl.acm.org/citation.cfm?id=1496950.1496966*

9 Appendix - Configuration Manual

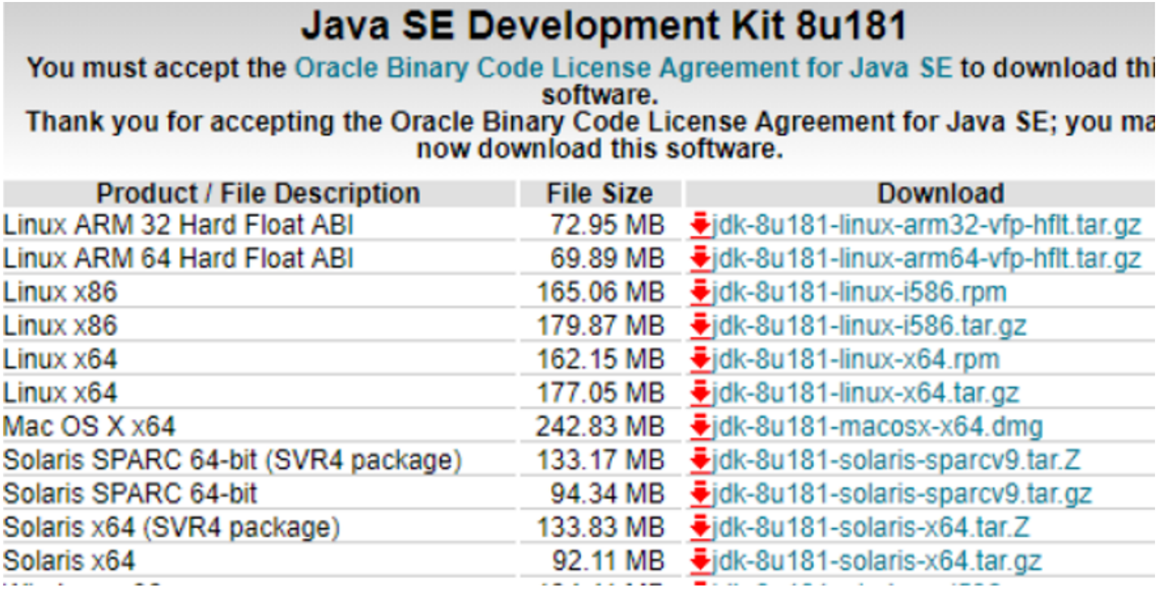
The configuration manual provides the detailed explanation of how to install the java, NetBeans IDE and cloud sim 3.0.3. Each section below provides the detailed description of how to install and run all the functionalities to run the project on the system.

9.1 Download Software

To run the project the main thing is to install the necessary software. **Step 1**- 1.1 Java Development Kit: The JDK is required to run the java applications on the system. Without the installation of the JDK java applications cant be run on the system (*Java SE Development Kit 10- - Downloads*; n.d.).

The windows X64 needs to be downloaded. Henceclick on the option which is parallel to 64 it that is jdk-8u181-windows-x64 exe.

Step 2- 1.2 Download the Netbeans IDE - The netbeans is an integrated environment developed to run the java application on the system. The java code is written in netbeans and it is popular because it is much lighter than eclipse.



Java SE Development Kit 8u181

You must accept the Oracle Binary Code License Agreement for Java SE to download thi software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you ma now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.95 MB	jdk-8u181-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.89 MB	jdk-8u181-linux-arm64-vfp-hflt.tar.gz
Linux x86	165.06 MB	jdk-8u181-linux-i586.rpm
Linux x86	179.87 MB	jdk-8u181-linux-i586.tar.gz
Linux x64	162.15 MB	jdk-8u181-linux-x64.rpm
Linux x64	177.05 MB	jdk-8u181-linux-x64.tar.gz
Mac OS X x64	242.83 MB	jdk-8u181-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	133.17 MB	jdk-8u181-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.34 MB	jdk-8u181-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.83 MB	jdk-8u181-solaris-x64.tar.Z
Solaris x64	92.11 MB	jdk-8u181-solaris-x64.tar.gz

Figure 22: JDK download

Figure 23: Net beans IDE download (*NetBeans IDE Download*; n.d.)

Step 3- 1.3 Downloading of cloud sim environment - The cloud sim jar is available to set up the environment for the cloud sim. The cloud sim provides the jar that can be put into the project so that the environment can be set up.

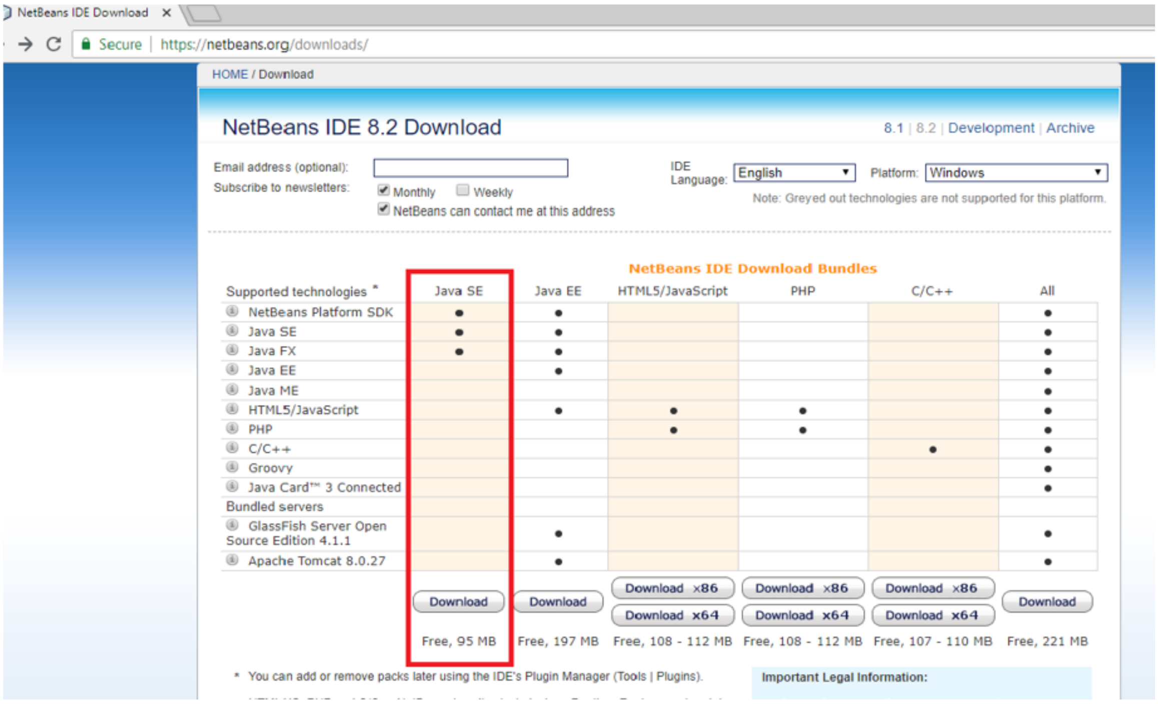


Figure 24: Downloading the configuration

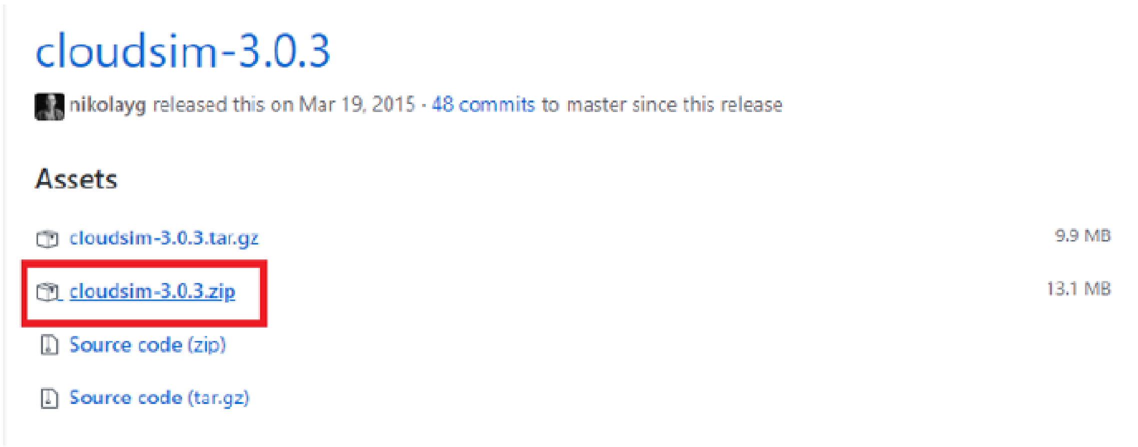


Figure 25: Download cloud sim 3.0.3

9.2 Software installation procedure

The software installation procedure provides the detailed description about how to download the software.
2.1 Java JDK installation
After the installation of the JDK the environment needs to be created.
Control Panel -System and Security - System
Advanced System Settings Environment - Variables
System Variables -Path New
After following the described path give the installation path so that installation can be done and click OK.
2.2 Install NetBeans

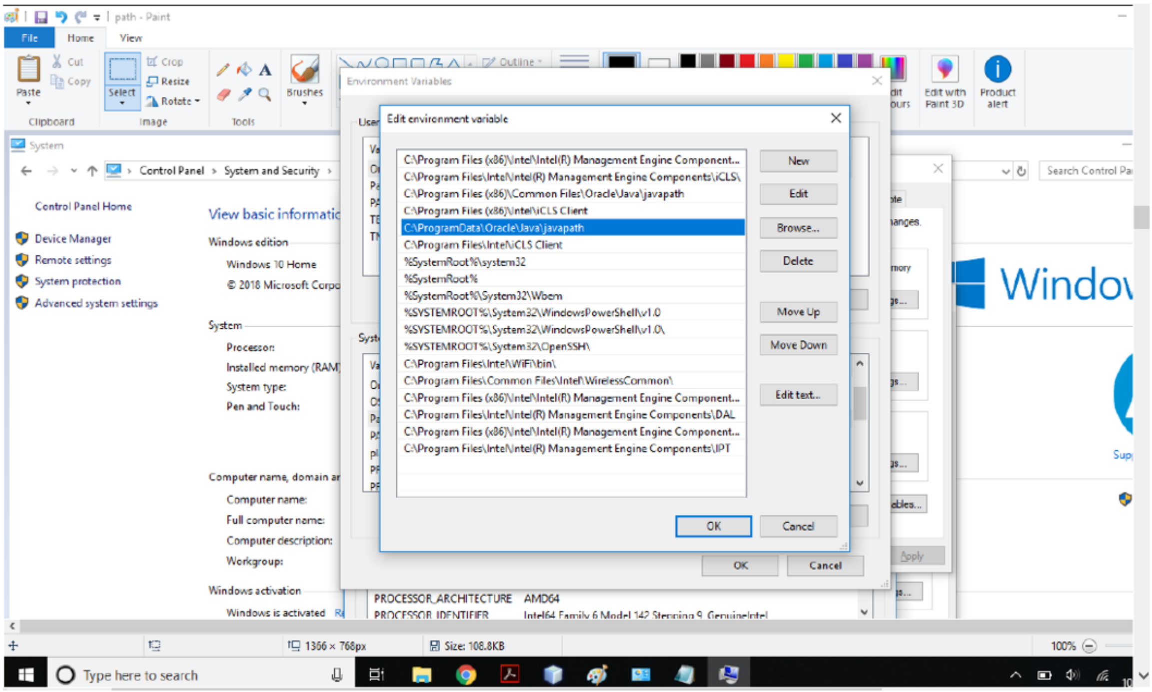


Figure 26: Setting up the configuration

The NetBeans is getting installed and just need to click on the agree button. Finally click on the finish button.

2.3 Integration of cloud Sim
The cloud Sim needs to be imported into the project and below is the procedure to import the project. Right click on project -Properties - Libraries -Add Jar/Folder

9.3 Implementation

This procedure shows how to import the project and run it to obtain the result.
3.1- Importing project
File - Import Project - From Zip
After opening the NetBeans import the zip file.

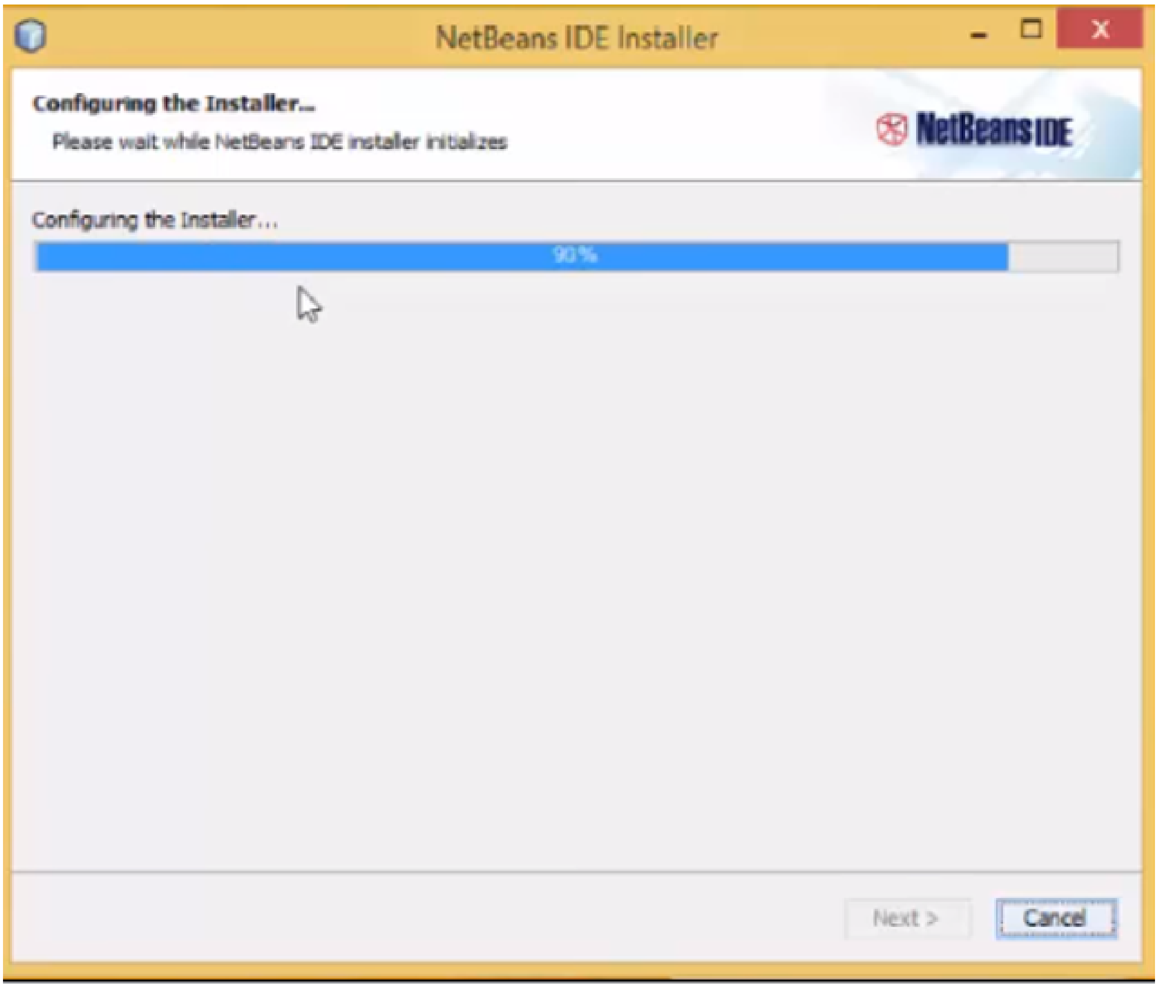


Figure 27: Downloading

9.4 Output

The main class is inside the learning automata all the jars are already added in this project. The package name is learning automata and all the classes are written inside it. The main class is helper.java where the algorith has been written. In this figure the proceduce to run the project is illustrated. For runnig the project right click on the learning automata package and run. The learning automata contains different classes where the code to run the project. The output of the simulation is displayed in the figure.

This figure shows the result after the application of the algorithm.

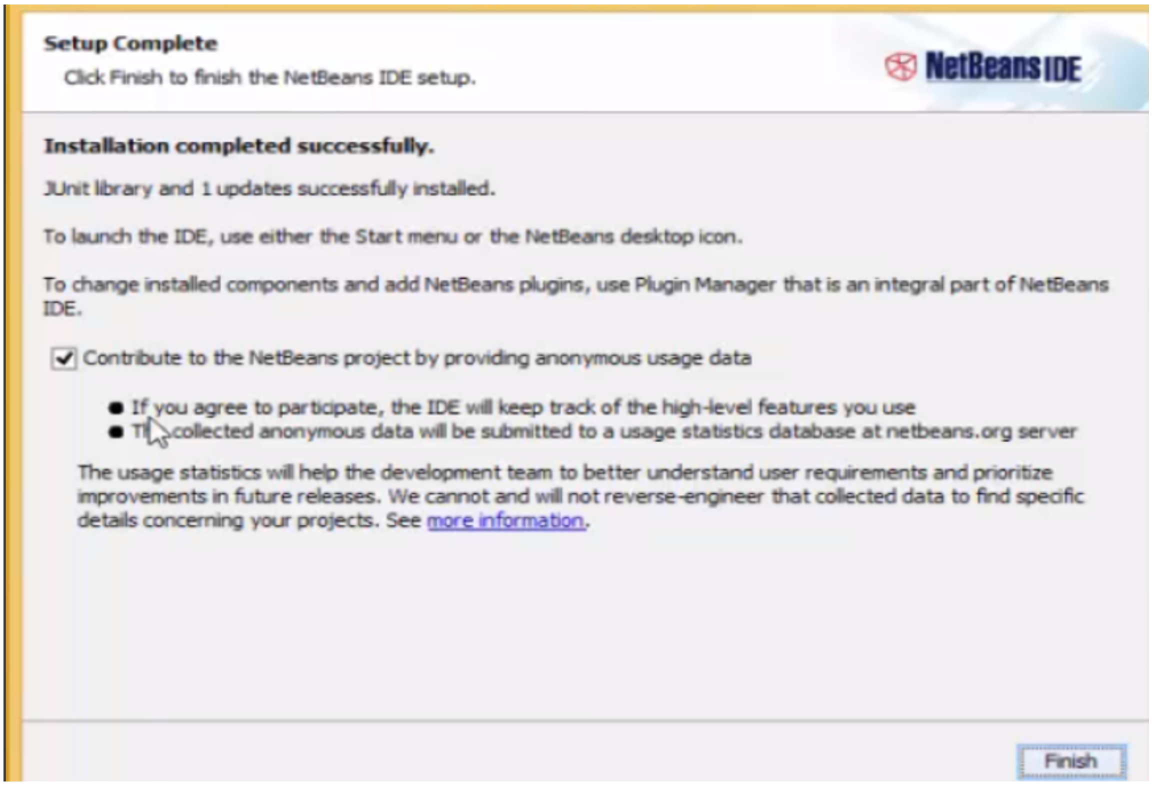


Figure 28: Final step to download

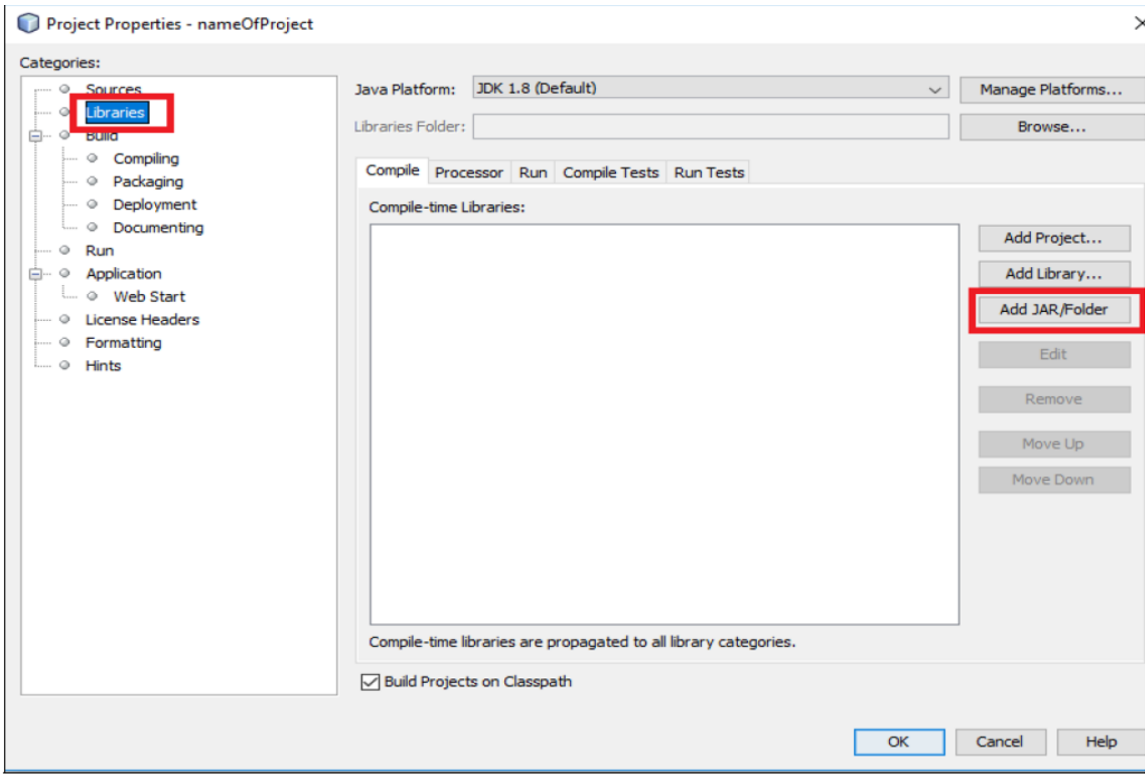


Figure 29: Adding the necessary jar

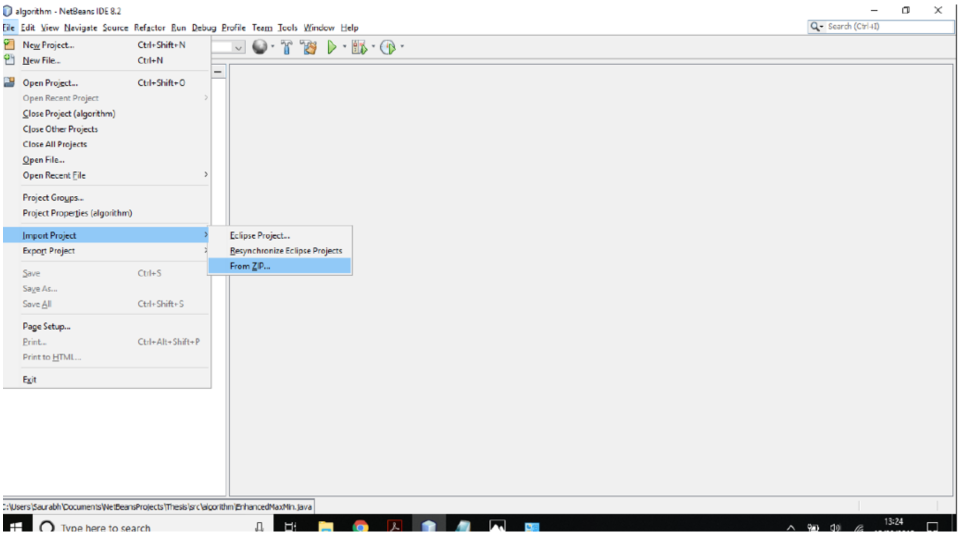


Figure 30: Necessary steps

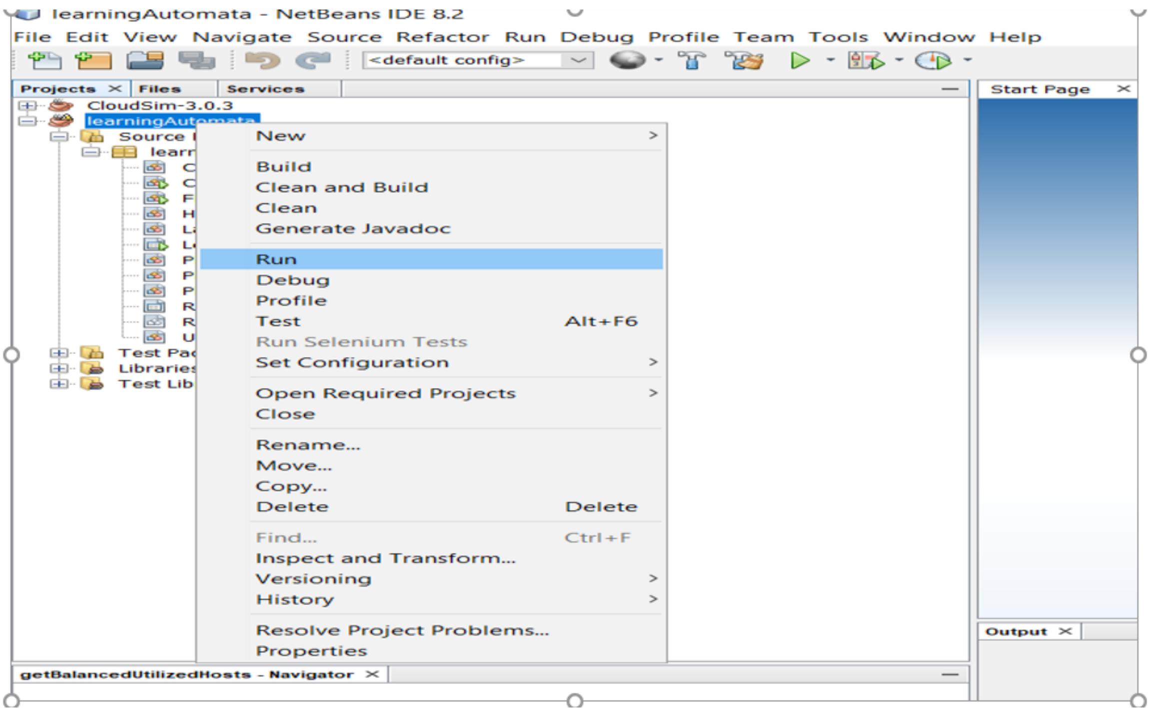


Figure 31: Run the file

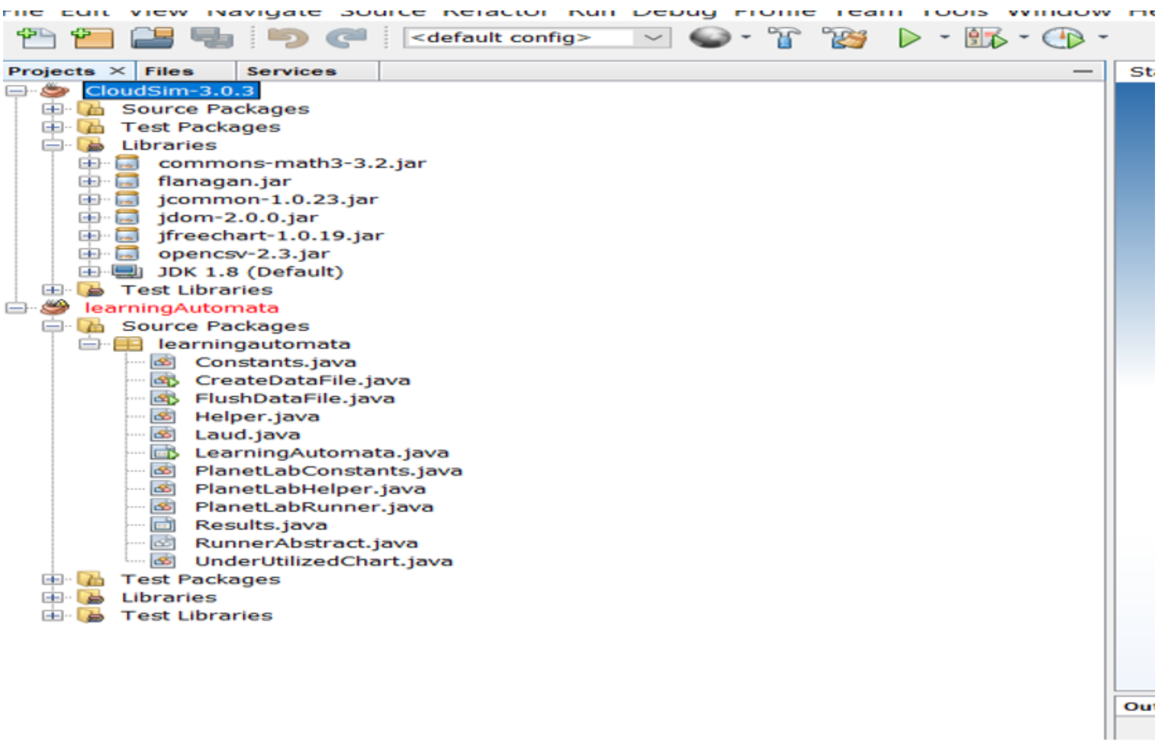


Figure 32: Structure of project

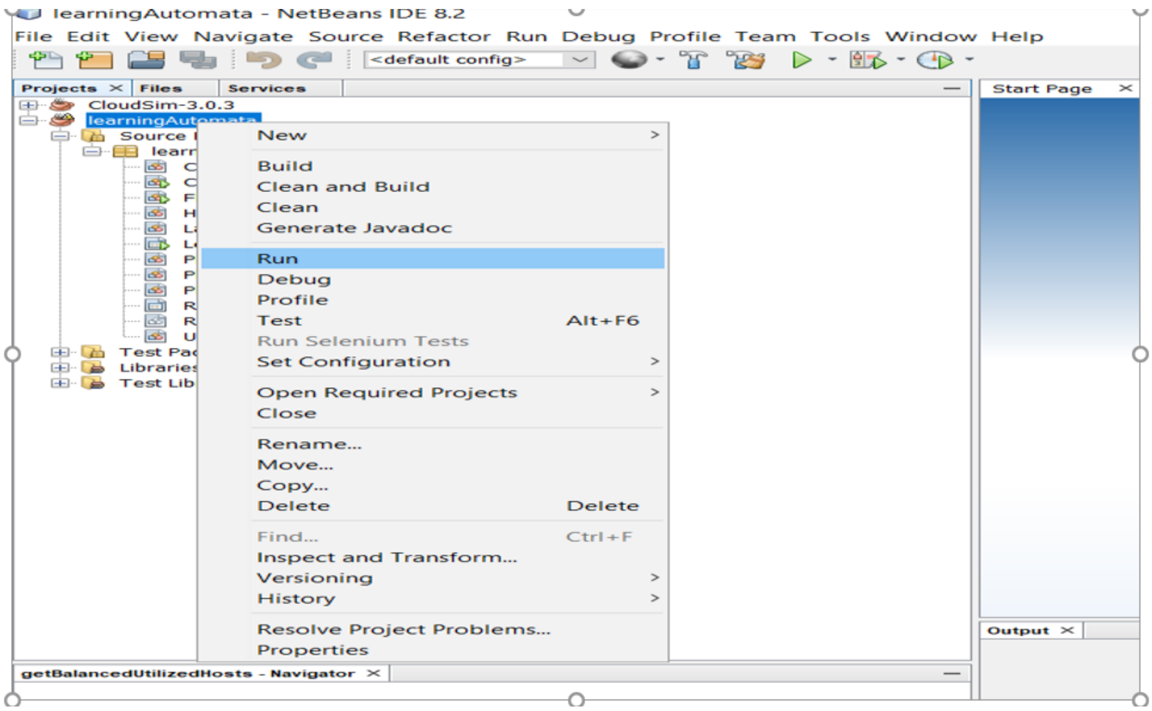


Figure 33: Run the file

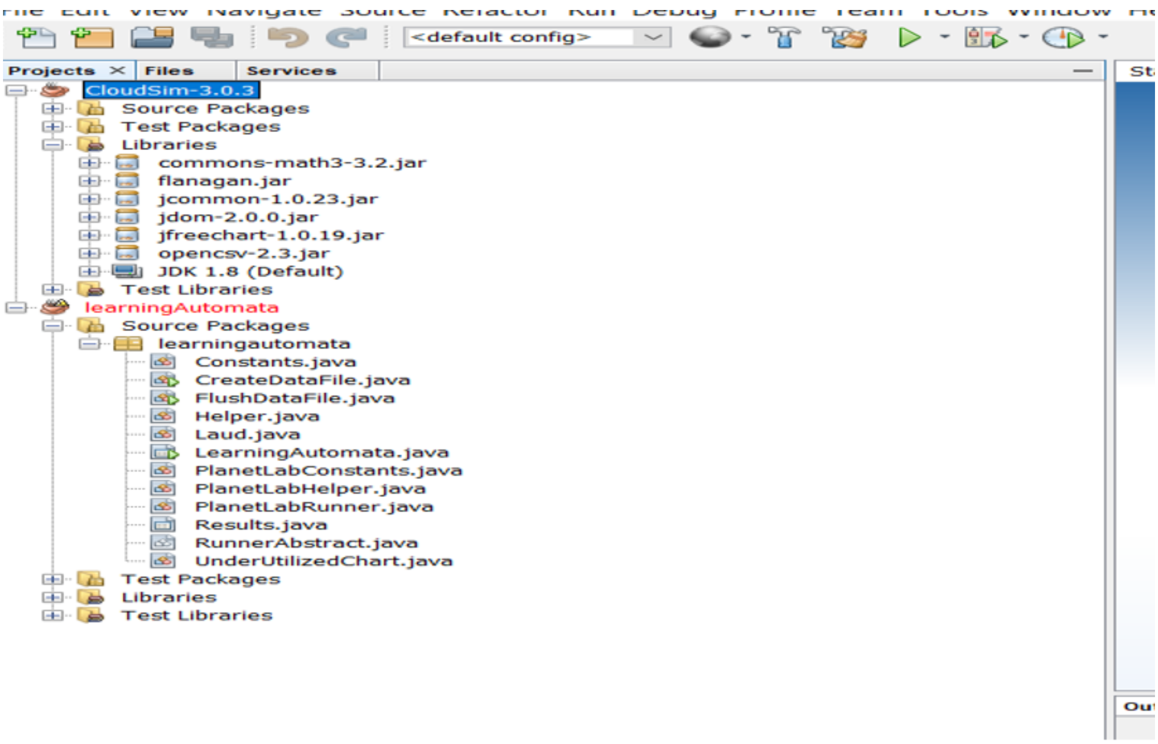


Figure 34: Structure of project

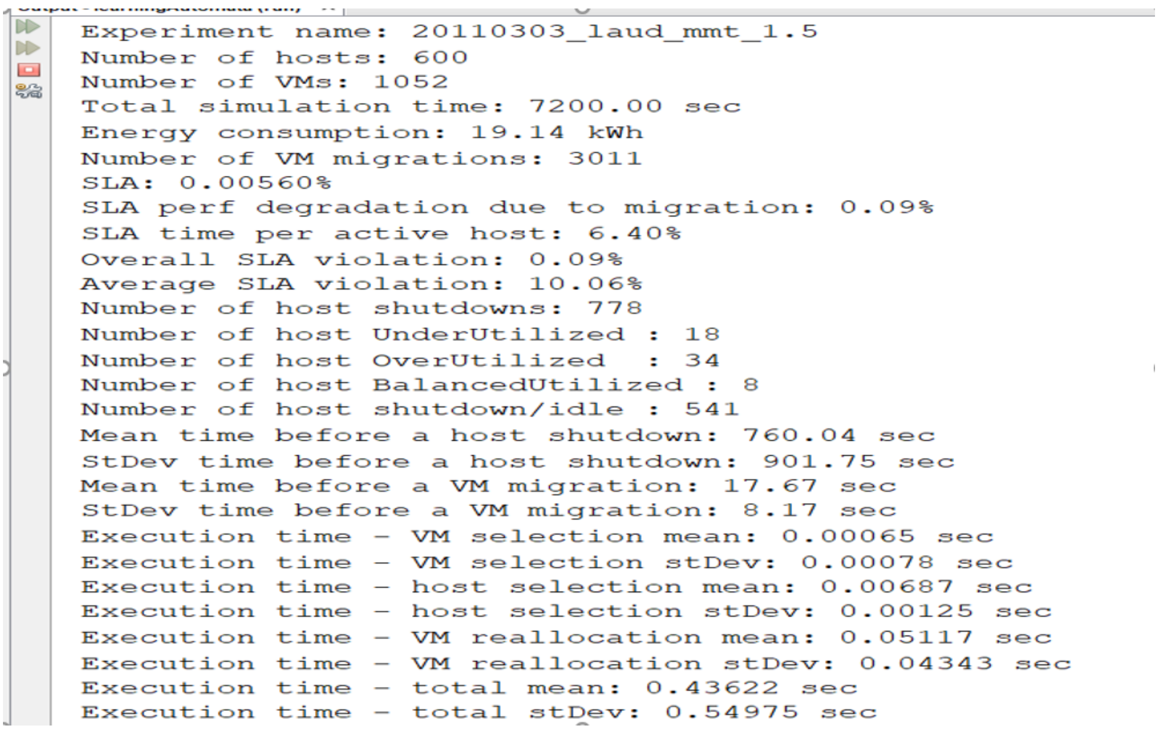

```
Experiment name: 20110303_laud_mmt_1.5
Number of hosts: 600
Number of VMs: 1052
Total simulation time: 7200.00 sec
Energy consumption: 19.14 kWh
Number of VM migrations: 3011
SLA: 0.00560%
SLA perf degradation due to migration: 0.09%
SLA time per active host: 6.40%
Overall SLA violation: 0.09%
Average SLA violation: 10.06%
Number of host shutdowns: 778
Number of host UnderUtilized : 18
Number of host OverUtilized  : 34
Number of host BalancedUtilized : 8
Number of host shutdown/idle : 541
Mean time before a host shutdown: 760.04 sec
StDev time before a host shutdown: 901.75 sec
Mean time before a VM migration: 17.67 sec
StDev time before a VM migration: 8.17 sec
Execution time - VM selection mean: 0.00065 sec
Execution time - VM selection stDev: 0.00078 sec
Execution time - host selection mean: 0.00687 sec
Execution time - host selection stDev: 0.00125 sec
Execution time - VM reallocation mean: 0.05117 sec
Execution time - VM reallocation stDev: 0.04343 sec
Execution time - total mean: 0.43622 sec
Execution time - total stDev: 0.54975 sec
```

Figure 35: Result of simulation

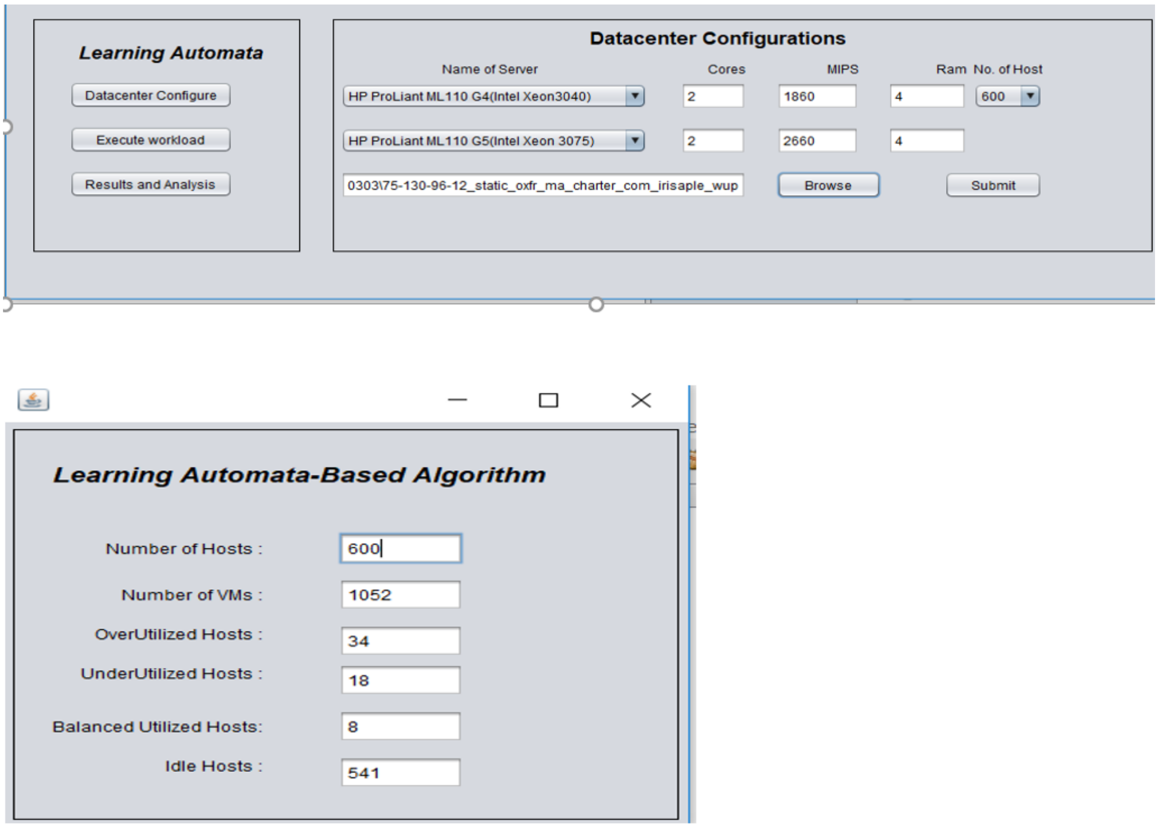


Figure 36: Final result

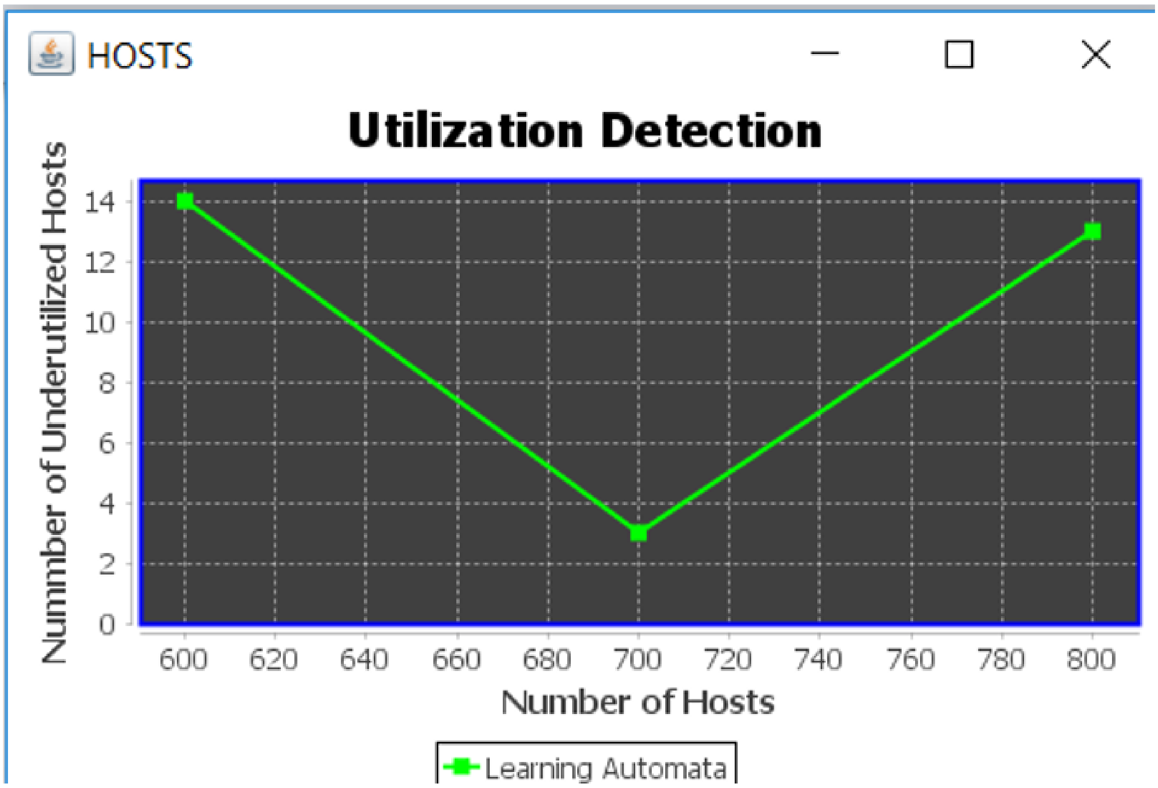


Figure 37: Result displaying in form of graph