Cost Efficient Genetic Algorithm for Resource Scheduling in Cloud Infrastructure

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering/Information Technology

By

GIRISHA CHAUDHARY (131255)

Under the supervision of

DR. PUNIT GUPTA

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat,

Solan-173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Cost Efficient Genetic Algorithm for Resource Scheduling in Cloud Infrastructure"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2016 to May 2017 under the supervision of **Dr. Punit Gupta** (Assistant Professor, Department of Information and Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Girisha Chaudhary
131255

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Punit Gupta
Assistant Professor
Department of Information and Technology
Dated:

ACKNOWLEDGEMENT

I owe my profound gratitude to my project supervisor Dr. Punit Gupta, who took keen interest and guided me all along in my project work titled — "Cost Efficient Genetic Algorithm for Resource Scheduling in Cloud Infrastructure" till the completion of my project by providing all the necessary information for developing the project. The project development helped me in research and I got to know a lot of new things in my domain. I am really thankful to him.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
DBaaS	Database as a Service
IT	Information Technology
QoS	Quality of Service
kWh	kilo-Watt hour
SLA	Service Level Agreement
VM	Virtual Machine
I/P	Input
O/P	Output
ID	Identity
CDBMS	Cloud Database Management System
LAN	Local Area Network
DBMS	Database Management System
SQL	Structured Query Language
GA	Genetic Algorithm
i.e.	That is
PE	Processing Element
MIPS	Million Instructions per Second
RAM	Random Access Memory
B/W	Bandwidth

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

ABSTRACT

Cloud computing is a reliable computing platform for large computational intensive or data intensive tasks. This has been accepted by many industrial giants of software industry for their software solutions, companies like Microsoft, Accenture, Ericson etc has adopted cloud computing as their first choice for cheap and reliable computing. But which increase in number of clients adopting this there is requirement of much more cost efficient and high performance computing for more trust and reliability among the client and the service provide to guarantee cheap and more efficient solutions. So the tasks in cloud need to be allocated in an efficient manner to provide high resource utilization and least execution time for high performance, at the same time provide least computational cost. Many resource algorithms are been proposed to improve the performance, but are not cost efficient at same time. Algorithms like genetic, particle swarm and ant colony algorithm are efficient solutions but not cost efficient. So to overcome these issues, we have proposed a learning based cost efficient algorithm for cloud Infrastructure. Proposed algorithm uses genetic algorithm for cost efficient task allocation to minimize cost and high utilization to provide better QoS (Quality of Service) to the client. Proposed strategy has proven to have better performance in term of execution cost, execution time, scheduling time as compared to previously proposed task allocation algorithm

CHAPTER 1 – INTRODUCTION

1.1 Introduction

Cloud computing has been a familiar topic all over the world for quite a lot of time now. It basically acts as a medium using which the users are able to access different pieces of information using a network or a web browser. Therefore, it eliminates the requirement for evolving and maintaining expensive facilities required for computing. The features of a general cloud system are: access of on-demand services, elasticity, drastic reduction of cost, minimum effort required for management, scalability, and independence of the device or the location. As gradually there is an increase in the deployment and adoption of computing in the cloud , it is very essential to evaluate the performance of different cloud environments. Simulation and modeling technologies are generally used for measuring various issues in security and performance. Testing of any software system based on the cloud requires various tools and techniques to help solve different quality concerns based on infrastructure of the cloud environments. The above mentioned tools and technologies may be built upon the cloud platform. This provides us with the advantage of platforms that are virtualized along with the substantial resources as well as simultaneous execution and services.

Cloud Computing used as a service over the Network

Cloud computing which is also generally simply referred to as only “the cloud,” may be defined as the distribution of the required cloud computing resources as per the demand everything starting right from the data centers to the entire applications using the internet network paying only for what you use. It has the following advantages :

- Facility of service to self
    - All kinds of the cloud computing resources one requires with the self access
- Flexibility of the computing resources
    - The users are provided with the full flexibility to scale up or down the demand of resources as per the requirement very easily and immediately
- Facility of measured services

- This allows the users to pay only for the cloud computing services they use

There are three types of cloud services :

Software as a service (SaaS)

SaaS or Software as a service is a cloud based application that runs on computers at a certain distance in the cloud environment. These distant computers are those that are operated and owned by people other than users and these connect to the respective computers of the users through the internet network and is generally a web browser.

The advantages of SaaS are :

- The cloud services are capable of scaling usage needs dynamically.
- Users may sign up and start the usage of the creative business apps immediately.
- Data will not be lost if the user's computer stops, since the data is stored in the cloud.
- The given data and apps may be accessed from any distant computer that is connected.

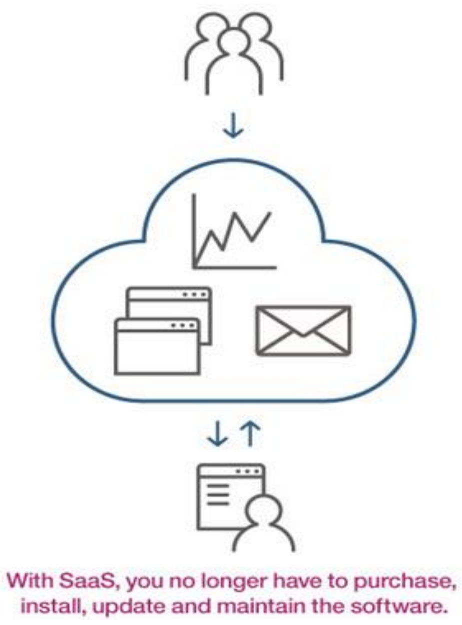

With SaaS, you no longer have to purchase, install, update and maintain the software.

Figure 1.1 SaaS

Platform as a service (PaaS)

PaaS or Platform as a service provides the cloud users with an environment based on the cloud with all computing resources needed to provide support in the complete lifecycle of creating and distributing cloud applications based on the internet without the need of worrying for the complexity and the cost of purchasing and handling the underlying software or hardware, hosting or provisioning.

The advantages of PaaS are as follows:

- Reduced complexity with middleware service
- Developed applications and faster delivery to the market
- New web applications immediately deployed to the respective cloud

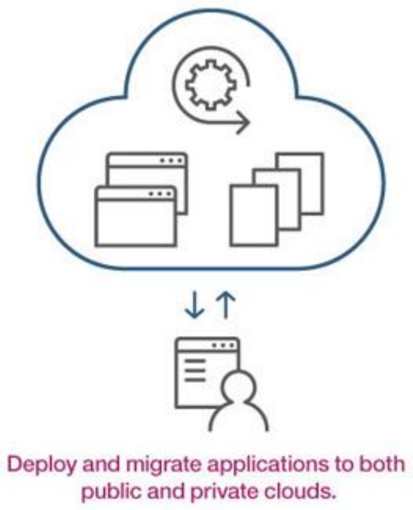

Deploy and migrate applications to both public and private clouds.

Figure 1.2 PaaS

Infrastructure as a service (IaaS)

IaaS or Infrastructure as a service provides the cloud companies with all the requied computing resources which include networking, servers, data center space, and storage paying only for what you use.

The advantages of IaaS are as follows:

- Services that are flexible and innovative are available as per the demand
- Users dont need to invest in their own hardware
- Scalability of on demand infrastructures to support different kinds of dynamic workloads

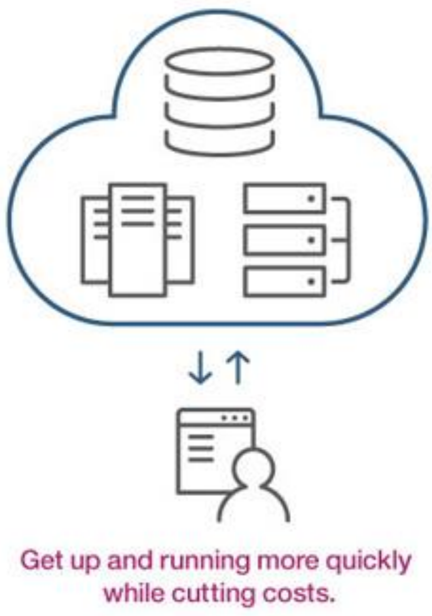


Figure 1.3 IaaS

Types Of Clouds

There are three different types of clouds as follows:

(i) Public Cloud

The public clouds are those clouds that are owned and managed by those companies which offer immediate access to the users to the computing resources that are affordable over a network that is public. With these public cloud services, the cloud users need not to purchase the supporting infrastructure or the hardware or software. These all are owned and operated by the providers themselves.

The key features of a public cloud are :

- Innovative and creative SaaS business apps for various types of applications ranging right from the customer resource management (CRM) to applications like data analytics and transaction management
- Scalable and flexible IaaS for compute services and storage on notice of a moment
- Cloud-based application with a powerful PaaS for the deployment and development environments

Flexibility to access the resources you need, when you need them.

Figure 1.4 Public Cloud

(ii) <u>Private Cloud</u>

A private cloud is an infrastructure that is solely operated for a particular organization, whether handled by a third party or internally and hosted either externally or internally. Private clouds may also take the advantages of efficiencies of the cloud, along with providing facility of steering clear of multi-tenancy and more resource control.

The key features of a private cloud are as follows:

- Sophisticated governance and security designed for some particular requirements of the company.
- Controls services with a self-service interface that allows the IT staff to rapidly arrange, allocate, and distribute the IT resources on-demand.
- Management of pools of resources that is highly automated for everything starting right from computing the capability to analytics, storage as well as the middleware.

The additional level of security you want
with the benefits of cloud.

Figure 1.5 Private Cloud

(iii) <u>Hybrid Cloud</u>

A hybrid cloud is one that uses the foundation for private cloud integrated with the use of public cloud services. The ultimate reality is that a private cloud may not be able to exist in alienation from the remaining IT resources of the company and of course the public cloud. Most of the companies owning private clouds may eventually evolve to handle workloads across various public clouds, data centers as well as private clouds henceforth creating these hybrid clouds.

The key aspects of a hybrid cloud are as follows:

- It allows portability of services, data as well as apps and even more options for the models for deployment.
- It also enables the companies to maintain the sensitive data and the critical applications in a private cloud or a traditional data center environment.
- It also facilitates taking the advantage of different resources of the public cloud including IaaS, for flexible virtual resources and SaaS, for the recent applications.

1.2 PROBLEM DEFINITION

Cloud computing has been a new trend in problem solving and providing reliable computing platform for big and high computational tasks. This technique is used for business industries like banking, trading and many e-commerce businesses to accommodate high request rate, high availability for all time without stopping system and system failure. In case of failure, the requests are migrated to different reliable servers with letting the user knowing about it, providing fault tolerant behaviour of system. Other application zone includes scientific research like computing weather forecasting report, satellite imaging and many more applications which require high computation, which is now possible without creating a private infrastructure. Cloud deals with various kinds of applications which can have different request type and computation servers based on their capability i.e. hardware and software configuration. However, to computer these large requests count data centers consume high power and load over the data centers is also very high which may be storage, computational or network load. This may lead to reduce in QoS (Quality of Service) provided by a data center that may be due to deadline failure or a fault over a datacenter due to various reasons. Survey in 2006 shows power consumption of data center around 4.5 billion kWh, equivalent to 1.5% of total power consumed by USA, which will be increasing 18% yearly [1].

Cloud computing in general has various issues listed below: 1) With adoption of cloud computing techniques by industries and corporate users the user count is increasing rapidly with increase in number of cloud computing services. This increases the datacenter count and power consumption. 2) Resource allocation among data centers i.e. allocation of virtual machine on datacenters to provide high quality resource keeping in mind the behavior and characteristics of datacenters to provide high QoS at resource level. 3) Current task allocation algorithms focus on static load balancing algorithm balance the load when request load increases but does not take into consideration previous behavior of the datacenter under high load which leads us to design efficient learning based algorithms. 4) High loaded data centers has high failure probability to compute requests and with increase in load over data center request completion time increases which is not good for user as well cloud provider. This leads to SLA (Service Level Agreement) failure promised to the client.5) Some request are need to be computed with QoS but due to high load and fault rate they may the QoS promised which is not appropriate to user and will be a critical issue.

Cloud computing is a reliable computing platform for large computational intensive or data intensive tasks. This has been accepted by many industrial giants of software industry for their software solutions, companies like Microsoft, Accenture, Ericson etc has adopted cloud computing as their first choice for cheap and reliable computing. But which increase in number of clients adopting this there is requirement of much more cost efficient and high performance computing for more trust and reliability among the client and the service provide to guarantee cheap and more efficient solutions. So the tasks in cloud need to be allocated in an efficient manner to provide high resource utilization and least execution time for high performance, at the same time provide least computational cost. Many resource algorithms are been proposed to improve the performance, but are not cost efficient at same time. Algorithms like genetic, particle swarm and ant colony algorithm are efficient solutions but not cost efficient. So to overcome these issues, we require a learning based cost efficient algorithm for cloud Infrastructure.

1.3 OBJECTIVES

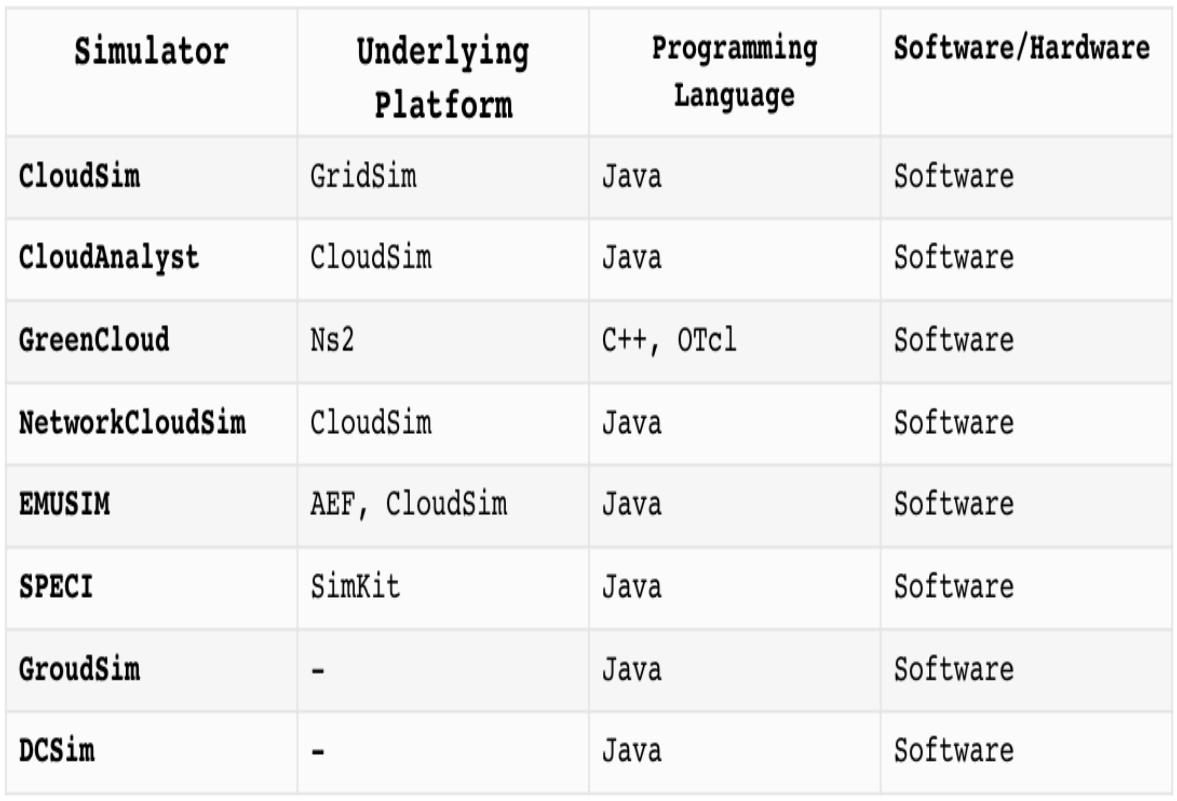
Our main objectives of this project are :

- To build cloud computing simulators.
- To develop a learning based cost efficient scheduling algorithm for cloud infrastructure

1.4 METHODOLOGY

There are a variety of options available when it comes to using a cloud simulator, each with its own specifications, qualities, advantages and disadvantages.

Table 1.1, presents the evaluation and analysis of the various cloud simulators that are based on different software or hardware, underlying platforms or developing languages.



Simulator	Underlying Platform	Programming Language	Software/Hardware
CloudSim	GridSim	Java	Software
CloudAnalyst	CloudSim	Java	Software
GreenCloud	Ns2	C++, OTcl	Software
NetworkCloudSim	CloudSim	Java	Software
EMUSIM	AEF, CloudSim	Java	Software
SPECI	SimKit	Java	Software
GroudSim	-	Java	Software
DCSim	-	Java	Software

Table 1.1: Comparison of various Cloud Simulators

In this project, we will be using CloudSim as our cloud simulator mainly due to the following advantages :

- Using it, the cloud users can evaluate particular system problems, without even considering the details which are usually low level related to the cloud-based infrastructures and services.
- It allows seamless experimentation, modelling and simulation of the application services of the cloud computing environment
- It is also a better alternative in the market than other simulators available mainly due to the fact that the existing simulators in the distributed systems aren't compatible with the cloud computing environment.

Having chosen the cloud simulator, we are now required to modify the simulator and develop scheduling algorithm and run it by incorporating log files. This is necessary since without modifying the original simulator, it cannot accept the real time Google data set. After this modification, our desired simulator is ready to work. We now test its performance by using real time Google data sets of different specifications in different scenarios.

This project focusses on learning based task allocation algorithm that emphasises on hardware capability and least execution time in cloud. Proposed algorithm leads to computation with quality of service in new directions and improved utilization of all resources accordingly by reducing the scheduling time by least. The project proposes a dynamic learning based task allocation algorithm to minimize the computational time and maximize utilization of resource by allocation resource to request by getting global best schedule with least execution time which make it more efficient and reliable computation over cloud environment. Proposed algorithm emphasises on learning base strategy to find a global appropriate solution which cannot be achieved using static algorithm like max-min, min-min and ant colony optimization to improve request failure count, make span and overall reliability of system with increase in QoS and SLA promised to users/clients. Proposed algorithm uses genetic algorithm for cost efficient task allocation to minimize cost and high utilization to provide better QoS (Quality of Service) to the client.   Proposed strategy has proven to have better performance in term of execution cost, execution time, scheduling time as compared to previously proposed task allocation algorithm.

1.5 ORGANISATION

The layered CloudSim architecture is depicted in Figure 1.6.

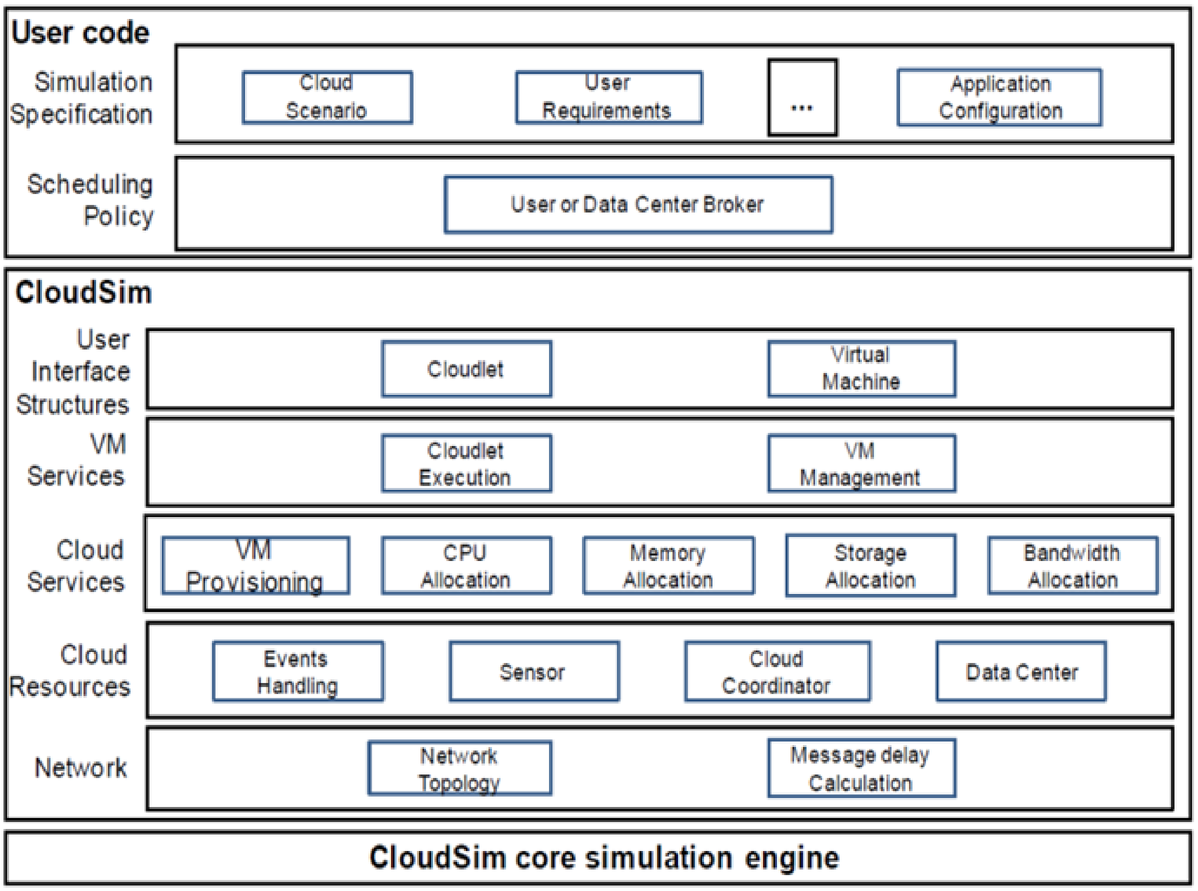


Figure 1.6 : Layered CloudSim architecture

The CloudSim Architecture

The layer of CloudSim has provisions for the support for simulation and modelling of the various cloud environments which include virtual machines, management of the dedicated memory interfaces, storage as well as bandwidth. It also has provisions of dynamic monitoring of the system state, hosts to VMs and other management of application execution. The service provider for cloud can also implement accustomed strategies at the CloudSim layer to analyse the effectiveness of various types of policies in provisioning of VM.

The layer of user code exposes fundamental entities which include the number and type of various machines used, the specifications of these machines, and so on, as well as scheduling policies, VMs, applications, application types and number of

users.
The major parts or components of the framework for CloudSim are as follows:

VM scheduler: The VM scheduler schedules the space or time shared and schedules a policy to distribute to VMs the processor cores.

Regions : The region part models all the different geographical regions where the service providers of cloud distribute resources to respective customers. In cloud evaluation, there are six different regions which correspond to the six continents on earth.

Data centre characteristics: The data centre characteristics model pieces of information related to the configurations of the data centre resource.

Data centres: The data centres model the services of infrastructure provisioned by different providers of cloud services. The data centres also encapsulate a set of servers or computing hosts that are usually either homogeneous or heterogeneous in their nature and also depend on hardware configurations of their own.

The user base: The user base models a users' group that is usually considered as piece of a single unit in the process of simulation. Moreover, its major responsibility is to produce the traffic for the process of simulation.

Hosts: Hosts are used to model physical resources which include storage or compute.

Service broker: It selects ths data centre that should be chosen to arrange the services in response to the service requests from the users.

Cloudlet : The cloudlet specifies the user instructions. It consists of the i/p as well as o/p files, the ID of application, the respective size of the execution commands for request and the respective name of the user foundation which is also the originator where the replies are to be sent back. It schedules the application services based on the cloud. The CloudSim simulator recognises the application complexity in the terms of its respective computational requirements. Each and every application service must have a data transfer overhead that is required to be carried out during its life cycle as well as an instruction length which is pre-assigned.

VMM allocation policy: The VMM allocation policy provides policies on how to provisionally distribute VMs to various hosts.

CHAPTER 2- LITERATURE SURVEY

Heuristic based task allocation in cloud environment is very new and requires a lot of refinement to optimize the system performance and Quality of service provided to the user. Various task allocation algorithms have been proposed to solve and optimize the problem of task allocation. Many of these algorithms are inspired from basic computational algorithm and physical phenomena, animal behaviour or universe evolution concepts. In this section we have discoursed some of those existing algorithms.

R Santhosh[14] came up with an enhancement in the above algorithm through real time scheduling using checkpointing algorithm in 2013.He introduced the check pointing algorithm. The Checkpointing algorithm sensibly migrates aborted tasks and starts execution from where it stopped. In this, checkpoint intervals are allocated for tasks in the queue and ready for execution. If a task misses deadline, it is migrated to other VM and execution starts from where last checkpoint interval was saved.
Its main disadvantages are that it doesn't give the best solution, the tasks chosen randomly for execution so chances of them missing deadlines are comparatively higher.

According to the current IaaS resource allocation policy, the resources are allocated to the user request if available else the user requests are rejected. Amit Nathani[15] came up with an enhancement in the above IaaS resource allocation policy in 2011.His technique is known as Haizea which addresses above issue by complex resource allocation policy. When a new deadline sensitive request submitted to Haizea, it finds a single slot to satisfy request. If no slot available, it reschedules already accommodated requests to make space for new request. It does this in two ways: swapping and backfilling.
Swapping : In this two consecutive requests can be swapped if second request asks lesser resources than the first request and both the requests finish execution in time before swapping.
Backfilling : In this, those requests are found that can be rescheduled on idle resources far from requested time slot of new request and then a time slot is created with idle resources nearer to the requested slot of the new user request.
Its disadvantages are that only consecutive swap is possible, backfilling consumes a lot of time and user requests are rejected if required resources are not available.

Another algorithm is max min algorithm. This algorithm initially estimates execution and completion time of all tasks and assigns them resources based on a decision rule.According to this rule, the task with overall max completion time is chosen and assigned to the resource with min execution time where

Expected execution time Eij of task Ti on resource Rj = required time by Rj to complete Ti provided Rj has no load when assignment occurs

Expected Completion Time Cij of task Ti on Resource Rj = Overall time consumption till finishing any task previously assigned.

ri = begining of execution of task Ti

Cij= ri +Eij

The major disadvantage of max min algorithm is that it is efficient only for homogeneous system where large and small task cannot be clearly differentiated.

Upendra Bhoi[17] proposed an enhancement in the above max-min algorithm through enhanced max-min task scheduling algorithm in 2013.The algorithm allocates the resource with min completion time to the task with maximum execution time. This way the slowest resource is assigned to the largest task. Meanwhile the other high speed resources can execute smaller tasks concurrently.

Huankai Chen[18] later introduced priority in the above enhanced algorithm in 2013 through user-priority guided min-min scheduling algorithm. The algorithm gives highest and lowest priority to the corresponding tasks and treats others as normal priority. It then loads tasks of highest priority group tasks and performs min min algorithm. And consequently it does the same for normal priority and lowest priority group tasks.

Its disadvantages are that a large number of calculations make it complex and it doesn’t provide a load balanced schedule.

To deal with the network delay in scheduling tasks in cloud, we have network aware task scheduler which coordinates computation bound and network bound tasks in a large cluster so that resources are utilized in a more balanced fashion. In 2016,Jingjie Jiang [19] came up with the technique of symbiosis for network-aware task scheduling in data-parallel frameworks. All the tasks that are network bound are always bound to incur a resource imbalance because of the under utilization of the CPU whereas on the other hand, a task that is computation bound with the locality of data mostly underutilizes the provided network links. In the technique of symbiosis one predicts this resource

imbalance and later corrects it. In this technique, all the computation bound tasks (network free) are to be scheduled with computation free tasks (network bound).
The disadvantages of above techniques are that it is not always beneficial to schedule network bound task with computation bound task since network bound tasks not always computation free and that it only decreases network delay and not the overall execution time.

Z Liu [19] has proposed an cost aware scheduling for cloud. Author has proposed an cost based scheduling for task allocation in cloud SaaS model. Where cost in the cost of virtual machine on which the task is executed for a period of time, where cost includes RAM cost, storage cost, processor cost and network bandwidth code per unit of time. Proposed cost model is used by all the cloud providers. But the proposed algorithm suffers from large execution time because most of the tasks are diverted to one virtual machine creating a bottle neck.

Suraj, S [13] proposed an adaptive genetic algorithm for task allocation with least execution time and high resource utilization. Fitness function used in this algorithm is a function of utilization of online request under execution and the execution cost of upcoming request to find he fittest host for requests. So to find a global best solution rather than sticking in local minima and improve the scheduling time
Above discoursed existing algorithms tries to improve the performance of cloud environment in terms of execution time, resource utilization and scheduling delay to find the best solution, but they have not taken the cost function into consideration. So we have proposed a cost and utilization aware genetic algorithm to improve the cost efficiency and provide global best execution time i.e. the best task schedule with least finish time.

R.N. Calheiros[12] suggested that measuring the operation of the Cloud scheduling schemes, application models workload, and resources operation models in a controllable way under differing user and system configurations and needs is tough to achieve. To deal with this type of challenge, he proposed CloudSim. It may be defined as a simulation toolkit which facilitates the simulation and the modeling of the Cloud computing environments and application provisioning systems. This CloudSim toolkit enables both the system and the modeling behaviour of the Cloud system constituents which include resource provisioning schemes, data centers as well as virtual machines or VMs. It even develops provisioning techniques for generic application which may be extended easily and also with limited effort. Presently, it enables the simulation and the modeling of the Cloud computing platforms that consist of both inter-networked i.e. federation of clouds and single clouds. Moreover, it even exposes traditional interfaces for developing policies and provisioning schemes for allotment of virtual machines under the inter-networked scenarios for Cloud computing. Several different researchers and scientists from various organizations, which include HP Labs in U.S.A., are also utilizing the CloudSim in their research and investigation process in the area of energy-efficient management and operation and the Cloud resource provisioning of the data center resources.

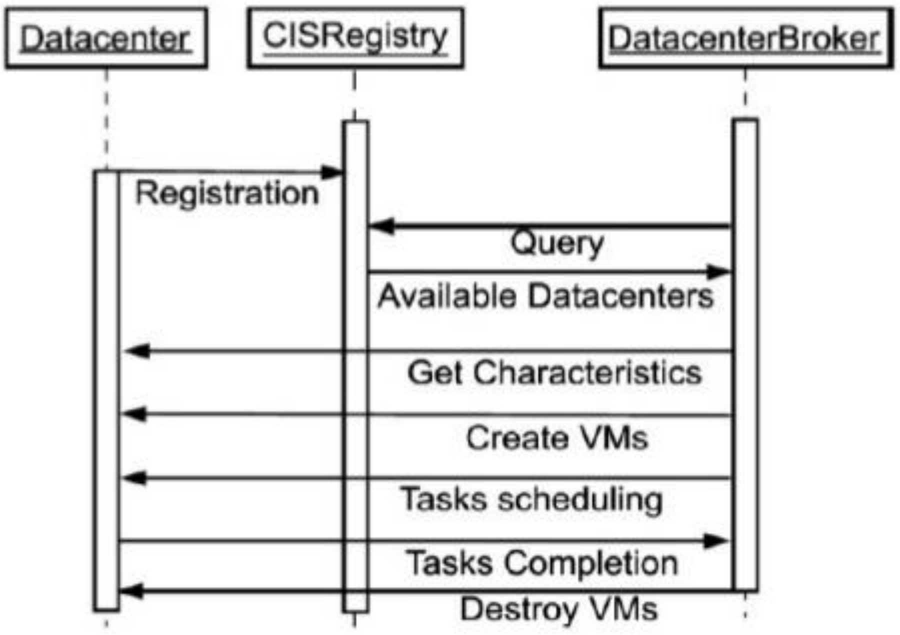


Figure 2.1 Simulation Data Flow

Figure 2.1 describes the communication flow among the various core CloudSim entities. Initially, at the beginning of the process of simulation, each and every Data center entity is required to register with the CIS Registry. This CIS Registry then provides essential information based on the registry-type functionalities, which include services of match-making for mapping the users or brokers, requests to appropriate providers of Cloud. After this, the brokers of the Data center who all are acting on the behalf of the users are required to consult and communicate to the CIS service in order to obtain the list of cloud providers who can offer infrastructure services that match application's QoS, the respective software, as well as hardware requirements. In the case of a match, the Data center broker completely deploys the built application with the suggested CIS cloud. The described communication flow so far depends upon the basic and fundamental flow in a simulated environment. Some of the variations are possible in this flow which tend to depend upon policies. As an example, the messages from the brokers to the Data centers may also require a confirmation of request from all other parts of the corresponding Data center, regarding the execution process of an action, or also about the largest number of virtual machines that a user may create.

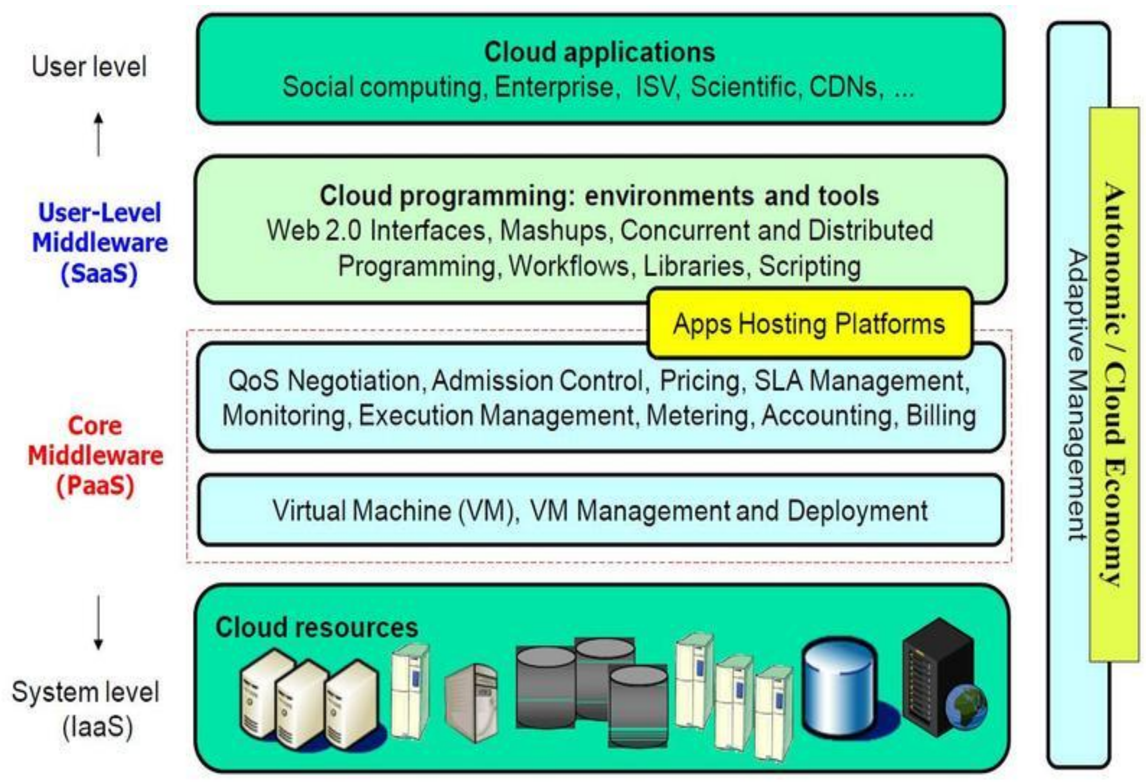

User level
User-Level Middleware (SaaS)
Core Middleware (PaaS)
System level (IaaS)
Cloud applications
Social computing, Enterprise, ISV, Scientific, CDNs, ...
Cloud programming: environments and tools
Web 2.0 Interfaces, Mashups, Concurrent and Distributed Programming, Workflows, Libraries, Scripting
Apps Hosting Platforms
QoS Negotiation, Admission Control, Pricing, SLA Management, Monitoring, Execution Management, Metering, Accounting, Billing
Virtual Machine (VM), VM Management and Deployment
Cloud resources
Autonomic / Cloud Economy
Adaptive Management

Figure 2.2 Layered cloud computing architecture

Figure 2.2 depicts the layered design and description of the Cloud computing architecture. Internal and physical Cloud resources form along with all the core middleware capacities the fundamental basis for the delivering of IaaS and/or PaaS. The operator-level middleware totally aims at provisioning SaaS capacities. The topmost layer completely focuses on the application services of SaaS by way of making the useful utilization of the services which are provided through the medium of lower-layer tag services. PaaS and/or

SaaS services are generally developed, distributed and are provided via the other-party services providers, who are completely different from the conventional IaaS service providers.

Bhathiya Wickremasinghe suggested that there has been a lack or scarcity of the tools that may enable or encourage the developers to analyse or evaluate the requirements of most of the large-scale Cloud applications generally in terms of the geographic distribution of one or both of the computing servers as well as the user workloads. To fulfil this created gap in the form of tools for analysing the evaluation and ofcourse the modeling of th Cloud environments and its applications, he proposed what is called as CloudAnalyst. It was designed and developed in order to simulate all the large-scale of Cloud applications developed with the sole purpose of researching and studying the dynamic behaviour of all such applications and that too under different deployment's configurations. CloudAnalyst also helps the developers coming with insights as in how they can be able to distribute different applications among different Cloud infrastructures and along with it value added services which include such as optimization of its applications and performance and also the providers that are incoming along with the use of all the Service Brokers.

One of the major objectives of CloudAnalyst is definitely to cleanly separate and differentiate between the respective simulation experimentation and exercise starting right from a programming exercise, such that a modeler can totally focus on its simulation complexities without having to spend way too much time only on the specific technicalities of its programming without using what is known as a simulation toolkit. The CloudAnalyst also facilitates a modeler to execute simulations repeatedly and also to conduct a lot of series of simulation of experiments and with slight few parameter variations in a quick, rapid and ofcourse an easy manner.

The major characteristics of the CloudAnalyst include the following:

- **Ease in using Graphical User Interface or GUI**
  CloudAnalyst is modeled with a comparatively easy to operate graphical user interface (Figure 2.3) that facilitates the users to set up and process experiments quickly, rapidly and way too easily.

- **Capability to describe a simulation process with a configurability and a flexibility of a high degree**

  Simulation of very complex systems which include the Internet applications totally depends on a lot many parameters. Conventionally, the values corresponding to those parameters are required to be arbitrarily pre assumed and/or determined through using a process of complete trial along with error. CloudAnalyst is bound to provide its modelers with a relatively high degree of repeatable control over the whole of experiment, by way of modeling of the entities and also the configuration options which include such as: a Data Center, with its hardware configuration described in terms of its physical machines constituted of large processors, many storage devices, internal and external memory and at last the internal bandwidth; Data Center with virtual machine specification in terms of its corresponding memory, storage as well as the bandwidth quota; The resource allocation policies for the Data Centers which include time-shared vs. space-shared; All the users of these application in terms as groups and their corresponding distribution and that too both geographically and temporally; Internet and its dynamics along with various configuration options which are for the network delays and also the available bandwidth; The Service Broker Policies that are bound to control that which segment of the respective total user base is totally serviced by which corresponding Data Center at a given point of time; and last but not the least the simulation duration in minutes, hours or probably days.

- **Repeatable nature of experiments.**

  CloudAnalyst facilitates modelers to help save the simulation experiments' input parameters and the corresponding results in form of XML files such that the experiments can be repeated or reused. This underlying CloudSim simulation framework makes sure that even the repeated experiments yield only identical results.

- **Output in graphical format.**

  CloudAnalyst has tha capability of generating automatic graphical output for the simulation of the results in form of charts, and tables and this is also desirable for summarizing a large amount of stats that are collected during the process of simulation. This type of efficient presentation also helps in recognising the essential patterns of all the output parameters and also helps and supports in the process of comparisons between all the related parameters. In this present version of CloudAnalyst, the statistical metrics given as follows are created as the output of the corresponding simulation: The respective response time of simulated application; overall maximum, average, and minimum response time of all the different user requests that have been simulated; The response time that has been arranged by the user groups that are located within the limited geographical regions; the response time that has been arranged by the time that has been showing the pattern of all the changes that have occurred in the application consumption throughout the day; consumption patterns of the running application; the number of operators that have been arranged by the corresponding time or the regions of the world, and also along with the overall effect of that consumption on the data centers that have hosted the application; the time taken by all the data centers for the purpose of servicing a user request; the overall time required for the purpose of processing of the request for the entire process of simulation; maximum, average, and minimum request processing time taken by each of the data centers; the response time variation pattern that has occurred during the day throughout as the load modifies; and at last the details of costs of the operation.

- **Utilization of integrated technology and easy extension**.

  CloudAnalyst is dependent on a modular design that is much easier to extend. It has been developed bu utilizing the following technologies:

  Java since the simulator is designed and developed to 100% upon Java platform, by method of using Java SE 1.6; Java Swing due to the fact that the GUI component has been built by method of using Swing components; CloudSim as CloudSim characteristics for the purpose of modeling data centers are used in CloudAnalyst; and ofcourse, SimJava.

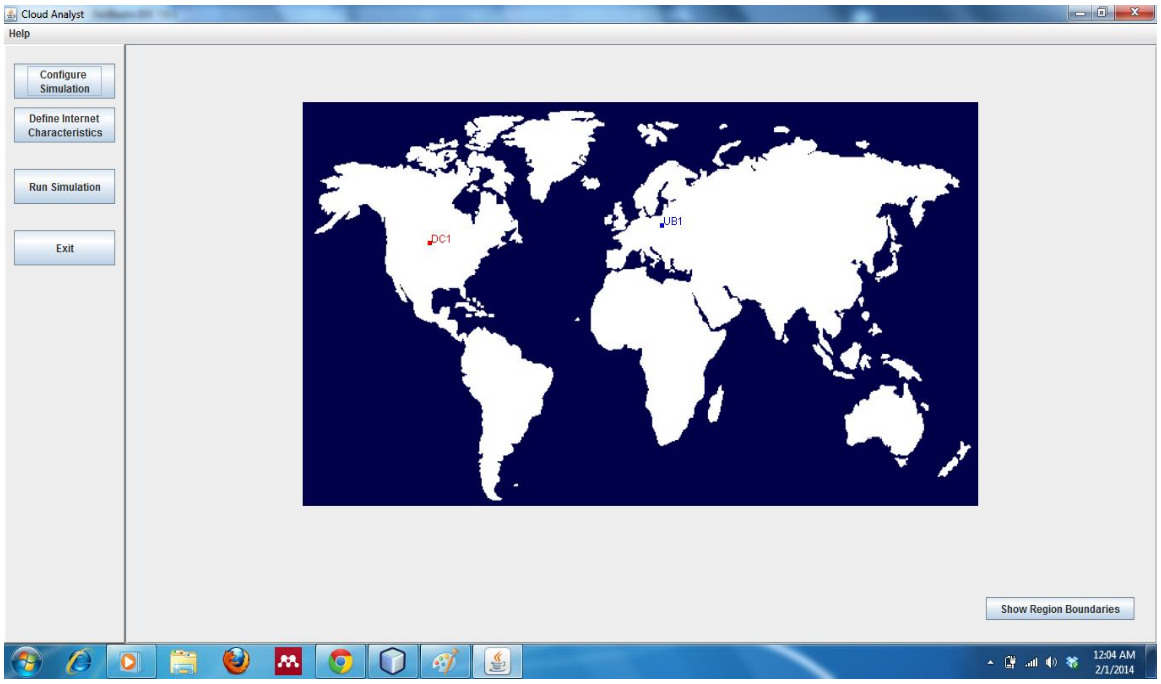


Figure 2.3 Cloud Analyst GUI

CHAPTER 3- SYSTEM DEVELOPMENT

3.1 DATA SET REQUIREMENTS

Real Time Google Data Set

The real-time Google data set allows the cloud users to inspect closely the activities as soon as they occur on the user's respective app or site. The further reports are later upgraded continuously and usually each and every hit is immediately reported within seconds of its occurrence. For instance, the users are able to see exactly the number of people that are on their site at the respective moment, which particular events or pages those people are interacting with, and the occurrences of goal conversions.

Google has recently launched a processing engine for data in real-time which is called as Google Cloud Dataflow and it has been recently announced about a year before. It even has inserted more new characteristics to its analysis tool known as BigQuery, which was introduced first in 2010. Both of the above mentioned cloud services can together be used to allow the processing in real-time of huge chunks of data together.

Google Cloud Dataflow is now also available as a beta which provisions the capability to evaluate data the moment it enters from a line of live updates. Google also has taken care of almost all of the configuration of software and provisioning of hardware, allowing the real-time users to completely ramp up the specific services without having to worry about the infrastructure underlying. The service is also able to evaluate data that has been already captured on the disk, in batch mode, facilitating the firm to mix and match current and historical analysis done in same workflows.

The service allows for a way for each and every Python or Java coder to write large applications having used the big data. It also makes it easier to compile and run all the end-to-end requests across varying and highly complex sets of data.

Several ways to use Real-Time

In Real-Time, users can continuously and rapidly monitor all the effects that changes in site and the recent campaigns are likely to have on the users' traffic. Following are some of the ways by which one may be able to use the Real-Time:

- Inspect completions of goals as you tend test the changes made to the site.

- Observe if a single day promotion is encouraging the traffic to the respective app or site, and what all pages are being viewed by the users.
- Analyse if modified and latest content on the site has been viewed.
- Confirmation of the working of the tracking code on the respective app or site.
- Evaluate the quick after-effects on the data traffic from a social network post or tweet or blog.

Advantages of using Real Time Data

- **Repeatable and controllable environment** : Different combinations of Real Time Google Data set can be used in different scenarios to test the performance of simulator repeatedly and controllably.
- **Tune system bottlenecks before deployment** : Bottlenecks can be avoided timely before deployment by correcting any faults or defects in the system simulator.
- **Time effectiveness** : Time complexity can be reduced on the real system by reducing the time complexity on the simulator of the system.
- **Standard data set** : Real Google Data set is the standard data set the use of which gives realistic results to the users before they deploy the actual working system and this can help in increasing effectiveness of the real system.
- **Easy analysis** : This data is easily made available by Google helps in easy analysis of the results.

3.2 SOFTWARE REQUIREMENTS

The software requirements of the project include:
(i) Java
(ii) Cloudsim

Java

Cloud computing explains the latest delivery model, supplement as well as consumption for computing services depending upon the Internet network. Java has now been the latest programming language for quite a long time which provisions the structure for applications related to Web, and recently development of Java has even reached the applications of cloud.

The rapid growth in development of web services has been a popular trend that has completely been changing the field of playing, as adoption movement and cloud conversion from Java EE 7 to Java EE 8 are growing. Some self-describing “late adopters,” and firms, have informed stability and security issues as critical risks in transporting the platform of their development. To tackle the fear of this kind, there has been an era of pouring out of positive comments from the Java community after the release of EE 8, which highly increases features of code simplification by using Lambda Expressions. The recent release of the the open source tool NetBeans 8.0.2 as well as official Java IDE are revamping faster the discussions of migration. With even more easier machine learning, about 30% less programming code required with Java 8 (no need to describe more effective code), and lessened code complexity for Java and Spark programmers who might not have specific professionalism in Big Data earlier, may ultimately create machine learning in approximately half of the code lines with almost identical programs of Hadoop.

CloudSim

CloudSim is the tool for simulation which facilitates the cloud program developers to check the operation of all their policies for provisioning in a controllable and repeatable surrounding, for free of any expenditure. It also helps keep a check on the bottlenecks before deployment in real time for the real world. Its a simulator; therefore, it does not compile and then run any original software. In nutshell, it may be described as ‘compiling and running a module of a surrounding in a module of hardware’, where technical details are encapsulated.

CloudSim is basically a library for various cloud scenario simulations. Moreover, it provides critical classes for defining the computational resources, data centres, applications, users, virtual machines, and schemes for the handling of different constituents of the system including provisioning and scheduling. Running these constituents, it is easier to analyse latest introduced strategies monitoring the working of clouds, as also considering policies of load balancing, schemes, scheduling algorithms, ,etc. It can further be used for assessing the complexity of different

strategies from typical perspectives including execution time of application, cost, etc. It even helps in the evaluation process of the Green IT schemes. It may also be used to work as a fundamental base for cloud environment that has to be simulated and may even add latest schemes for new scenarios, scheduling, and load balancing. It is scalable to work as a library which facilitates the user for adding a scenario desired by programming in Java.
Using CloudSim, firms, industry-based programmers and R&D departments may also check the operation of a latest programmed application in an easy set-up as well as controlled environment.
The major characteristics possessed by CloudSim simulator are shown in Figure 3.1.

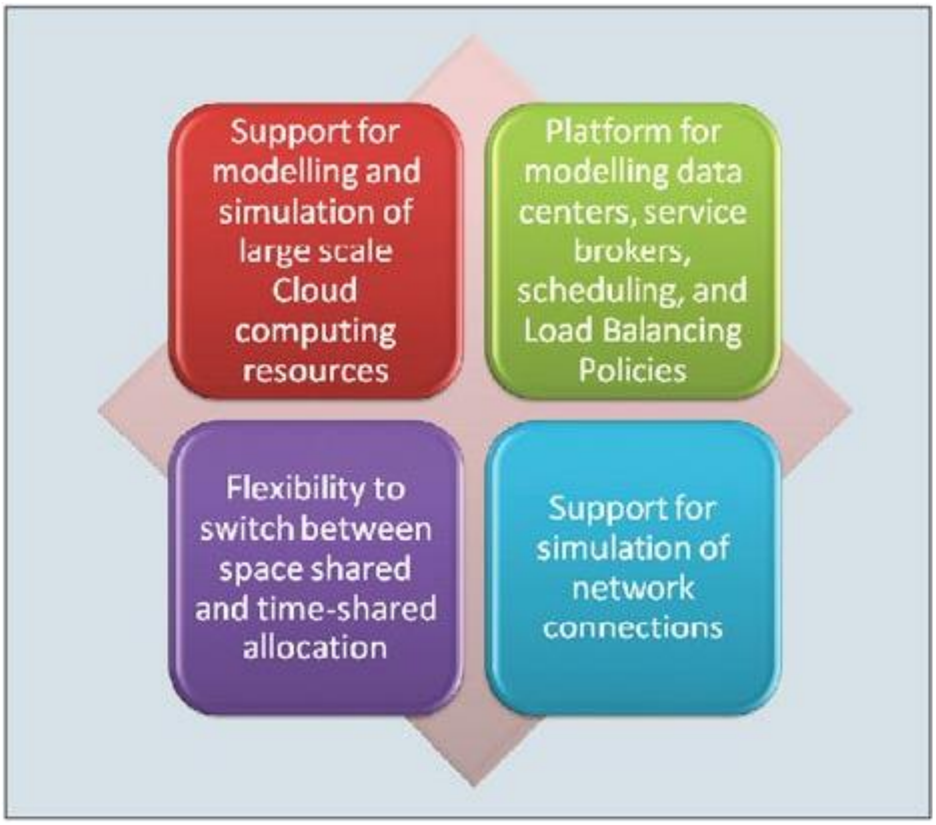

Support for modelling and simulation of large scale Cloud computing resources
Platform for modelling data centers, service brokers, scheduling, and Load Balancing Policies
Flexibility to switch between space shared and time-shared allocation
Support for simulation of network connections

Figure 3.1 Characteristics of CloudSim

3.3 STORAGE REQUIREMENTS

The storage requirements for this project are fulfilled by any database server.The server chosen by us is SQL.

Cloud Database Management System (CDBMS)

A CDBMS or a cloud database management system is a database
system management which is run by the service provider of some other party on a remote server and later addressed over the network.

A typical database system is setup on a server system at a firm's site and the related data is stored as well as accessed either directly or through a LAN orlocal area network. On the contrary, a cloud database management system, operates on the platform of a cloud provider and the related data might be accessed or stored whenever there is a network connection.

A cloud DBMS may be operated in three ways:
The first one is as an image of a VM or a virtual machine. In this model of deployment, the cloud owner sells instances of VM or virtual machine over which the database management system is run. The owner is accountable for the cloud infrastructure supporting the virtual machine and the customer is accountable for purchasing and/or uploading of the DBMS, maintaining the databases supported by it and confirming the concerned DBMS is managed properly.
The second deployment model contains the cloud provider who is accountable for the maintenance and supply of the DBMS. The customer himself is accountable for payment for compute resources and storage and for management of the databases supported by the DBMS. This form of deployment is known as DBaaS or Database as a service.
The third deployment model consists of the whole database implementation installed, maintained and managed by the cloud provider. This type of approach, known as managed hosting, may also entitle a small organization with the advantages provided by a database without the administrative accountabilities and IT overhead required typically from usage of DBMS.

Data Model

The development and deployment of traditional systems utilize relational databases and data management as their basic building blocks. More advanced queries that are defined in SQL operate well with the firm relationships which are superimposed on info by relational databases. But, relational database technology initially wasn’t designed and developed for operation over the distributed systems. This drawback had been addressed by adding the clustering enhancements to the relational databases, even though some fundamental tasks need expensive and complex protocols, including data.

SQL databases

Databases including EDB Postgres Advanced Server, PostgreSQL, Microsoft SQL Server , NuoDB, MariaDB, MySQL, and Oracle Database, are the kind of databases that may operate in the cloud, either as a service or in a virtual machine, solely depending on the respective vendor. Although SQL databases are easily scalable vertically, horizontal scalability imposes a great challenge, that the SQL bases cloud database services have begun to address**.**

3.4 PROPOSED MODEL

Above discoursed existing task allocation algorithm for cloud infrastructure aims to improve the performance inter mod scheduling delay and average utilization of the system or improving cost efficiency of the system, but the existing algorithm are static or dynamic in nature, and they may suffer from local minima solution considering that as the best solution where a better solution still cannot be found. So in this section we have proposed a cost based genetic algorithm (GA) for global best schedule to fine best execution time and with lease execution cost which provided better QoS then existing static, dynamic and learning based algorithms.

Proposed algorithms aims to find least cost and least execution time to finish the task with least finish time (Time taken to complete a set of requests) and at the same time providing the global i.e. the least finish time a system can achieve.

Proposed GA for task allocation is divided into 4 phases as follows:

I. Initialization

II. Evaluation and selection

III. Crossover

IV. Mutation

I. Initialization

In this phase we have a set of tasks (T1, T2, T3, T4, T5, T6…. T n) and a set of resources in term of virtual machine (VM1, VM2, VM3, VM4, VM5…. VM m) are pre allocated on hosts in distributed datacenters. Here we initialize asset of sequences or schedules allocated randomly, each sequences act a chromosomes for genetic algorithm. The complete set of chromosomes is said to be a population, acting as a input for algorithm.

II. Evaluation and selection

In this phase we evaluate the fitness value for each set of sequence or chromosome, which depends up on the computing capability, total time taken to complete the schedule and the network delay of complete schedule.

Where

VM_MIPS i : MIPS of ith virtual machine

T_Length i : Length of ith Task

Then the predicted time to complete a task Ti is defined:

$$\text{T\_Exei} = \left(\frac{\text{T\_length i}}{\text{VM\_MIPSi}}\right) \tag{4}$$

$$\text{Total\_time} = \sum_{i=1}^{n} \frac{\text{T\_length}_i}{\text{VM\_MIPS}_i} \tag{5}$$

Computational cost for a task over a virtual machine can be defined as the cost of resources used by a virtual machine for execution of task. There are various different costs involved in evaluation of final cost listed as follows.

costPerPE	:	The cost of using processing in this resource
costPerMIPS	:	The cost of using MIPS () in this resource
costPerRAM	:	The cost of using memory in this resource
costPerStorage	:	The cost of using storage in this resource
costPerBw	:	The cost of using bandwidth in this resource
T_cost i	:	The total cost of executing the task

$$\mathrm{Cost_i} = (\mathrm{T_{Exei_i}} * (\mathrm{V_i.costPerRAM} + \mathrm{V_i.costPerStorage} + \mathrm{V_i.costPerBW} + \mathrm{V_i.costPerPE}))$$

Total_Cost = $\sum_{i=1}^{n} \mathrm{Cost_i}$

The fitness value for a chromosome is defined by the fitness function gives as:

$$\mathrm{Fitness_{Chromosome_i}} = \frac{1}{\alpha\left(\mathrm{Total_{time_i}}\right)+\beta\left(\mathrm{Total_{time_i}}\right)} \tag{6}$$

Where

$$\alpha + \beta = 1 \tag{7}$$

Based on the fitness value of chromosome the fittest one is selected having least fitness value. The population is sorted based on the fitness value and best two are selected for next phase.

III. Crossover

In this step two fittest solutions based on least make span and cost is selected. We have used multi point crossover to generate new fittest sequences/ chromosome. Steps to generate crossover are as follows:

1. The two fittest chromosomes are selected
2. A new fittest chromosome is generated using multi point cross over by interchanging the set of schedules between two chromosomes.
3. The new chromosome replaces the chromosome with highest fitness value.

IV. Mutation

In this phase new merging the new offspring, which can be better solution with remaining which, regenerates population keeps the total population size constant after each iteration. After specific count of iteration predefined as an input to genetic algorithm, best chromosome is selected i.e. the chromosome with least fitness value is selected for schedule. Following algorithm discourses the proposed algorithm:

Algorithm: Cost Based Genetic Algorithm Task Allocation

```
Input: VM list $VM_i$, Task list $T_i$, Population size Po, Iteration Itr
1    CGATA ($VM_i$, $T_i$, Po, Itr)
2    $VM_i \leftarrow$ VM_List ( )
3    $i \leftarrow$ No. of VM
4    $T_i \leftarrow$ Task_List ( )
5    $C \leftarrow$ Genetic_algo ($VM_i$, $T_i$, Po, Itr)
6    Allocate_Resource (C) //processing the client request
Output: All requests have been scheduled
```

Algorithm: Genetic Algorithm

```
Input: VM list $VM_i$, Task list $T_i$, Population size Po, Iteration Itr
1    CGATA ($VM_i$, $T_i$, Po, Itr)
2    Po $\leftarrow$ Initiate_Population( $T_i$ )
3    Evaluation ( )
4    CenterMass ( ) //Find mean of all fitness values
5    C1 $\leftarrow$ GetFittest1 ( )
6    C2 $\leftarrow$ GetFittest2 ( )
7    Mutation (C1, C2)
8    Crossover (Po, C1, C2)
9    Return (GetFittest ( ))
Output: Server with minimum fitness value
```

Algorithm: Evaluation

```
1    Evaluation ( )
2    for each $C_i$ 0 to Po do
3        for each $T_i$
4            exec = $\alpha$ ($T_i/VM_i$) + $\beta$ ($FP_i$)
```

```
5                   $Fitness_i = Fitness_i + exec$
6             end for
7       end for
```

Output: To evaluate the fitness of all hosts

Algorithm: Allocate_Resource

Input: Chromosome List C

```
1       Allocate_Resource (C)
2       $C_i \leftarrow$ GetChromosomes ( )
3       for each $C_i$
4             Allocate ($C_i$)
5       end
```

Output: To allocate the best schedule

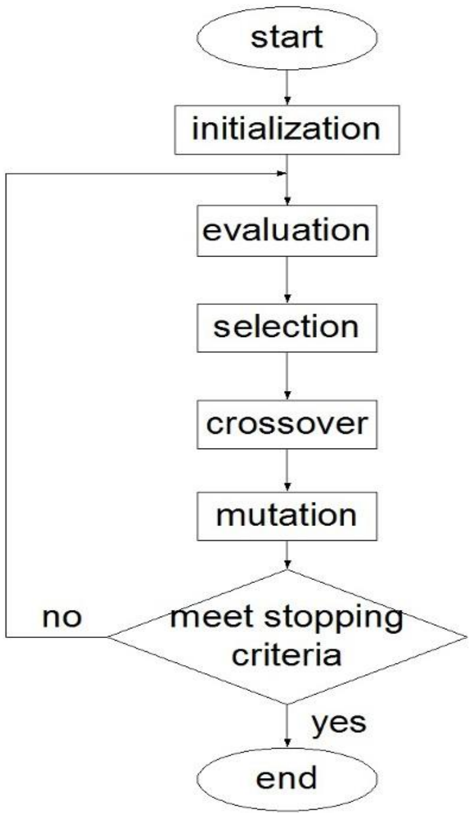


Figure 3.1 Genetic Algorithm

Algorithm 1 discourses basic system initialization.

Algorithm 2 shows the pseudo code for proposed algorithm and its evaluation, crossover and mutation phase respectively.

Algorithm 3 defines the evolution phase including the calculation of fitness value.

Algorithm 4 shows the final allocation phase according to proposed genetic algorithm.

Proposed algorithm provides a benefit over existing static scheduling algorithm, that it can search for best global solution rather than assuming the local best solution as the best solution. Moreover, the proposed algorithm takes into consideration the faulty behavior of cloud, which helps in find a solution with similar high utilization and least cost.

Figure 3.1 shows the flow diagram of proposed algorithm and the stopping condition of proposed algorithm is the number of evolutions, when number of evolution equals to zero stop iteration and we have found the best solution.

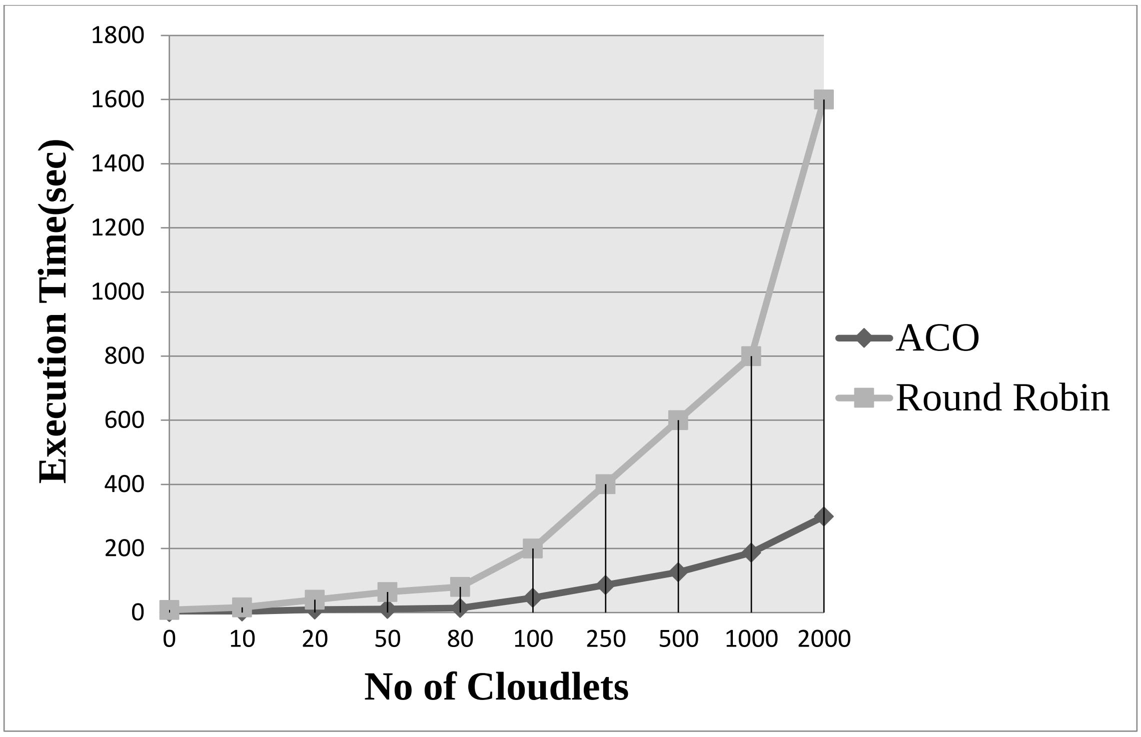
CHAPTER-4 PERFORMANCE ANALYSIS

4.1 EXPERIMENTAL RESULTS

For simulation we CloudSim 3.0 power module is used. CloudSim 3.0 provides cloud simulation and predefined power model simulation. Initial task is Cloud workflow simulation using Google data set which has been done using two algorithms, namely Round Robin (RR) and Ant Colony Optimization (ACO). These algorithms are run on the CloudSim simulator for number of VMs = 5. Their comparative analysis of performance in terms of execution time is given in table 4.1 and figure 4.1 as follows:

No of cloudlets	**ACO**	**Round Robin**
10	1.64	8.2
20	3.7	16.2
50	8.84	40.2
80	11.41	64.2
100	13.98	80.2
250	46.37	200.2
500	86.47	400.2
750	125.54	600.2
1000	187.23	800.2
2000	299.81	1600.2

Table 4.1 Comparative analysis of ACO and Round Robin algorithms



Graph 4.1 Comparative analysis of ACO and Round Robin algorithms

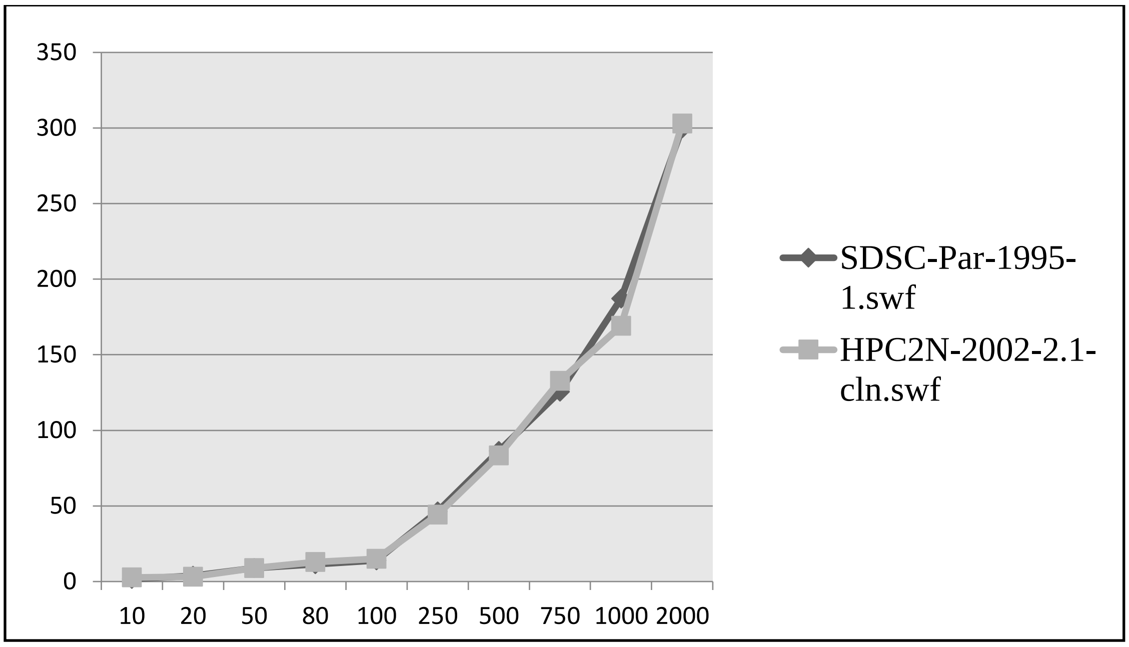
Initially the simulation is done using manual requests. Now we use automatic requests in real-time using Google data set. We have used 2 files of Google data set, namely SDSC-Par-1995-1.swf released by Google in the year 1995 and HPC2N-2002-2.1-cln.swf released by Google in the year 2002.Their comparison in terms of execution time is given as follows in table 4.2 and figure 4.2. :

Number of VMs used = 5

No. of cloudlets	SDSC-Par-19951.swf	HPC2N-2002-2.1-cln.swf
10	1.64	2.67
20	3.7	3.18
50	8.84	8.84
80	11.41	12.95
100	13.98	15.01

250	46.37	44.31
500	86.47	83.38
750	125.54	132.73
1000	187.23	169.23
2000	299.81	302.9

Table 4.2 Comparative analysis of Google data set log files



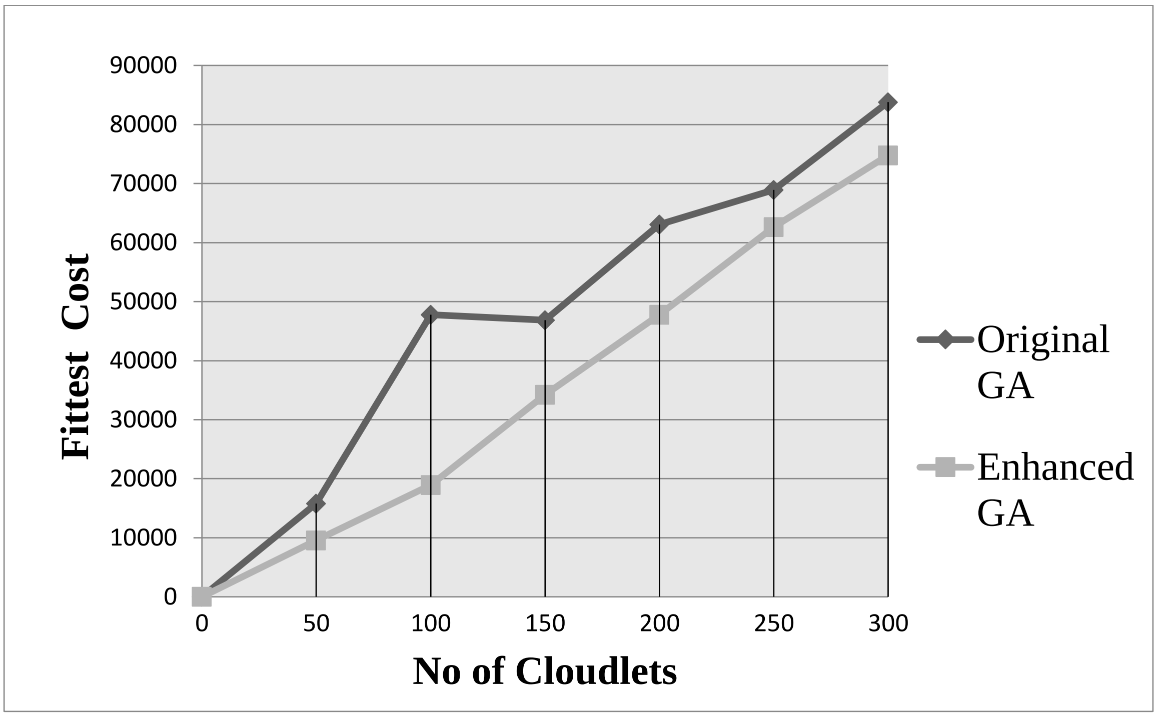
Graph 4.2 Comparative analysis of Google data set log files

Proceeding from simulation, we have proposed cost aware Genetic task allocation algorithm in CloudSim. Proposed algorithm is being tested over various test cases. We have compared original genetic algorithm with our proposed genetic algorithm with enhancement i.e. enhanced genetic algorithm on the basis of different parameters such as no of cloudlets, no of VMs and varying unit costs. The performance is measured in terms of the execution time and the final output of the algorithm in terms of the fittest cost. We observe that enhanced algorithm has given lower fittest cost than the original genetic algorithm for all the parameters under all the different cases and execution time is almost same for both the algorithms. So, in nutshell, the enhanced genetic algorithm gives lower fittest cost than original algorithm without compromising in the execution time. The experimental results are shown below:

Initially we vary cloudlets and keep other things constant (given below the diagrams):

No of cloudlets	Original GA	Enhanced GA
0	0	0
50	15778	9573
100	47765	18921
150	46858	34238
200	63070	47753
250	68927	62620
300	83793	74783

Table 4.3 Fittest Cost of OGA and EGA for different no of cloudlets



Graph 4.3: Fittest Cost of OGA and EGA for different no of cloudlets

In the above graph 4.3 and table 4.3 apart from the no of cloudlets, other specifications are as follows:

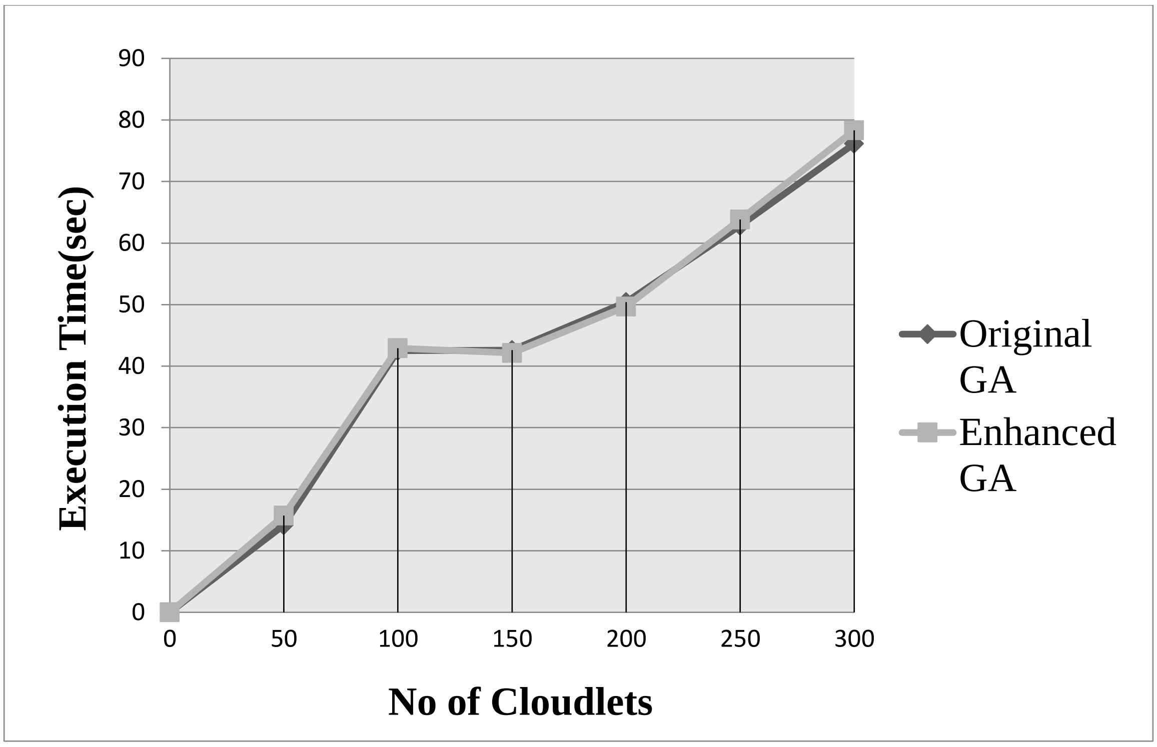
No of VMs = 5

Cost per Memory	=	0.45 units
Cost per Storage	=	0.9 units
Cost per B/W	=	0.9 units

Now we measure performance in terms of execution time for different no of cloudlets :

No of cloudlets	Original GA	Enhanced GA
0	0	0
50	14.15	15.7
100	42.5	42.9
150	42.6	42.15
200	50.4	49.69
250	62.83	63.8
300	76.14	78.32

Table 4.4: Execution Time of OGA and EGA for different no of cloudlet



Graph 4.4: Execution Time of OGA and EGA for different no of cloudlets

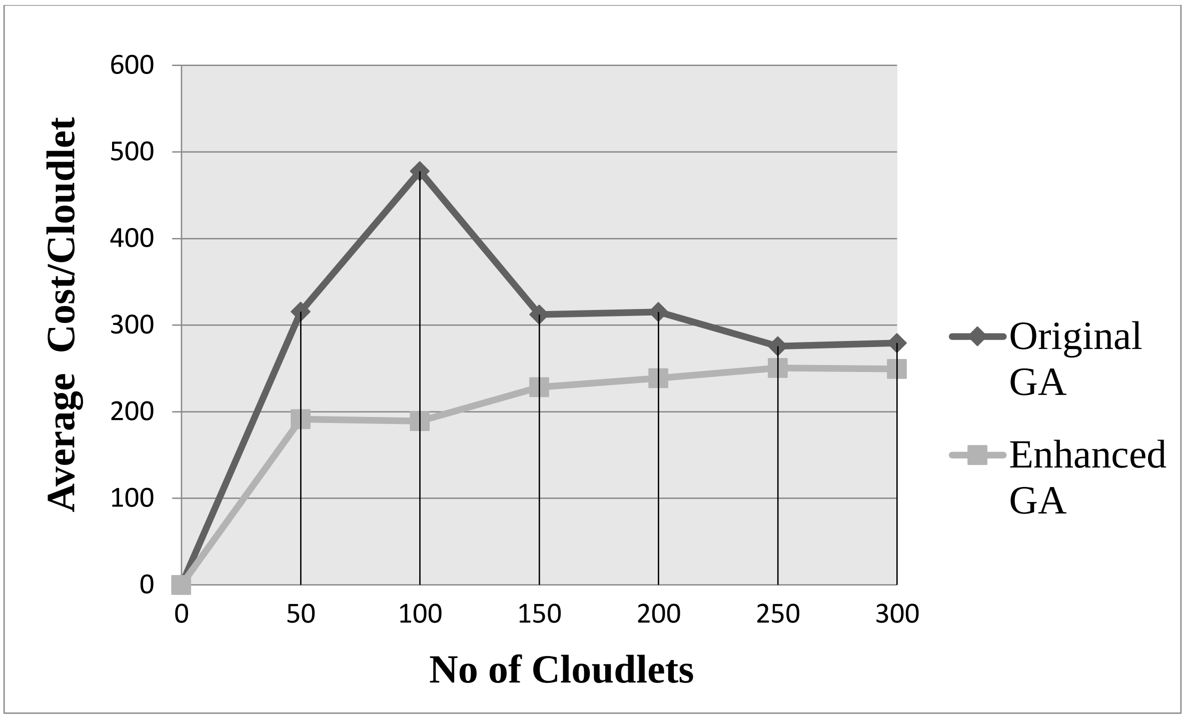
In the above graph 4.4 and table 4.4, apart from the no of cloudlets, other specifications are as follows:

No of VMs	=	5
Cost per Memory	=	0.45 units
Cost per Storage	=	0.9 units
Cost per B/W	=	0.9 units

Now, we measure performance in terms of average cost per cloud for different no of cloudlets:

No of cloudlets	Original GA	Enhanced GA
0	0	0
50	315.56	191.41
100	477.65	189.21
150	312.39	228.25
200	315.35	238.77
250	275.71	250.48
300	279.31	249.27

Table 4.5: Average cost/cloudlet of OGA and EGA for different no of cloudlets



Graph 4.5: Average cost/cloudlet of OGA and EGA for different no of cloudlets

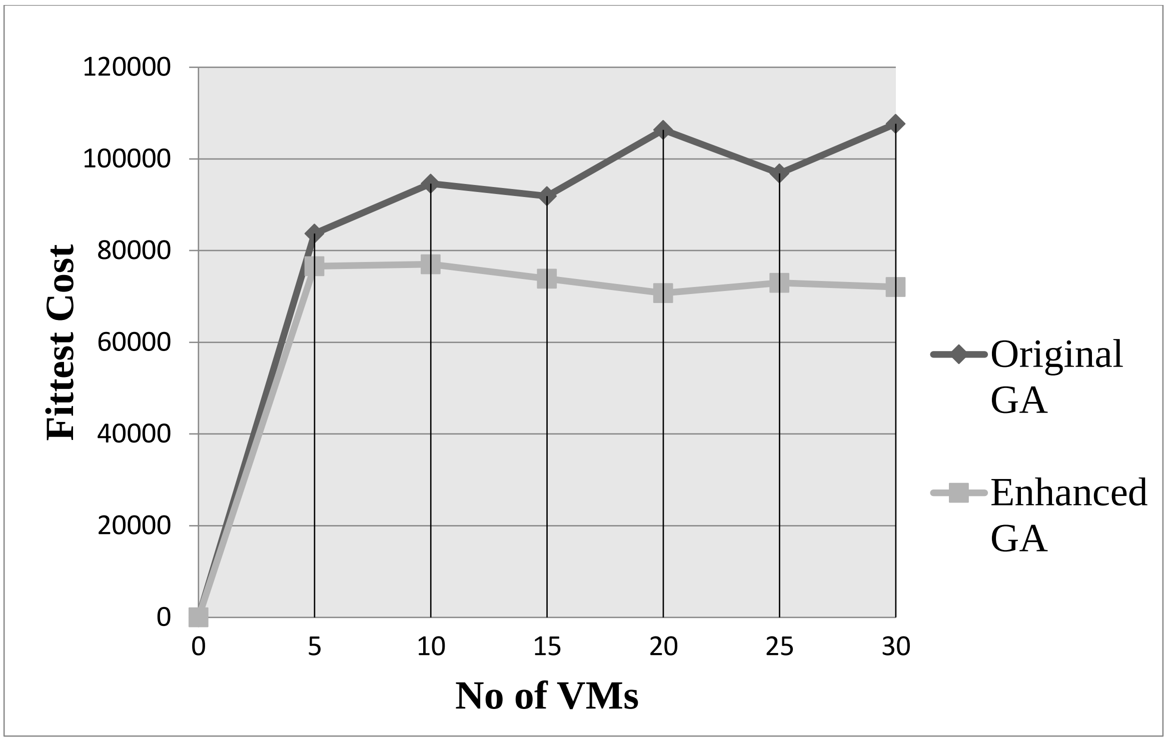
In the above graph 4.5 and table 4.5, apart from the no of cloudlets, other specifications are as follows:

No of VMs	=	5
Cost per Memory	=	0.45 units
Cost per Storage	=	0.9 units
Cost per B/W	=	0.9 units

Now, we measure performance in terms of fittest cost for different number of VMs:

No of VMs	Original GA	Enhanced GA
0	0	0
50	83793	76585
100	94605	77036
150	91902	73882
200	106330	70729
250	96858	72981
300	107682	72080

Table 4.6: Fittest cost of OGA and EGA for different no of VMs



Graph 4.6: Fittest cost of OGA and EGA for different no of VMs

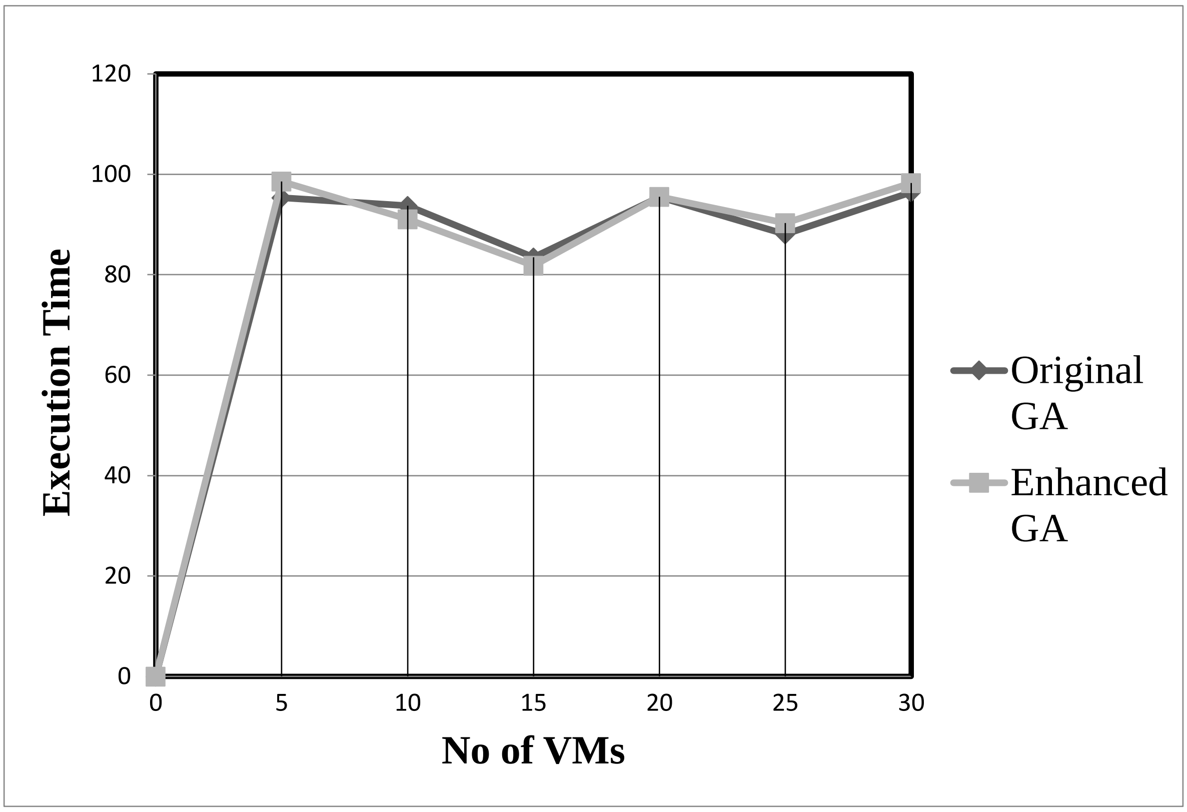
In the above graph 4.6 and table 4.6, apart from the no of VMs, other specifications are as follows:

No of cloudlets	=	300
Cost per Memory	=	0.45 units
Cost per Storage	=	0.9 units
Cost per B/W	=	0.9 units

Now, we measure performance in terms of execution time for different number of VMs:

No of VMs	Original GA	Enhanced GA
0	0	0
50	95.35	98.58
100	93.74	91.06
150	83.45	81.8
200	95.46	95.51
250	88.04	90.31
300	96.43	98.26

Table 4.7: Execution time of OGA and EGA for different no of VMs



Graph 4.7: Execution time of OGA and EGA for different no of VMs

In the above graph 4.7 and table 4.7, apart from the no of VMs, other specifications are as follows:

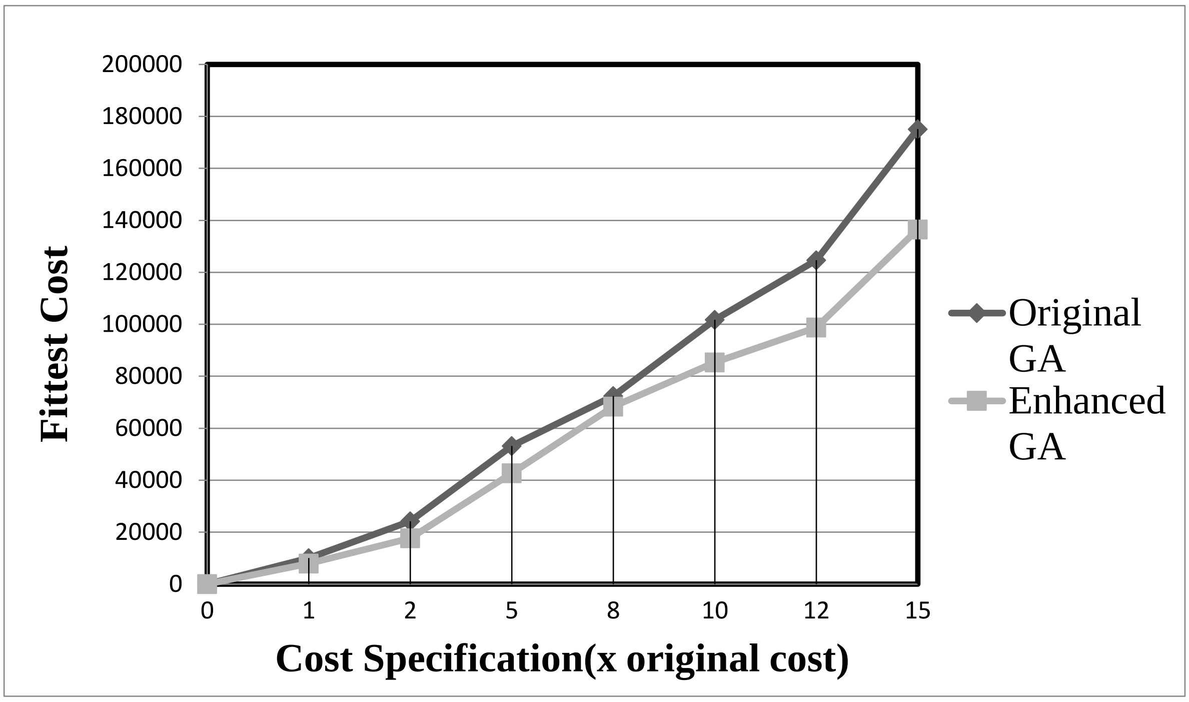
No of cloudlets	=	300
Cost per Memory	=	0.45 units
Cost per Storage	=	0.9 units
Cost per B/W	=	0.9 units

Now, we measure performance in terms of fittest cost for different cost specifications:

Cost Specification (x original cost)	Original GA	Enhanced GA
0	0	0
1	10048	7956
2	24212	17630
5	53199	42662
8	72424	68309

10	101772	85324
12	124639	98720
15	175130	136555

Table 4.8: Fittest Cost of OGA and EGA for different cost specifications



Graph 4.8: Fittest Cost of OGA and EGA for different cost specifications

In the above graph 4.8 and table 4.8, different specifications are as follows:

No of cloudlets	=	300
No of VMs	=	5

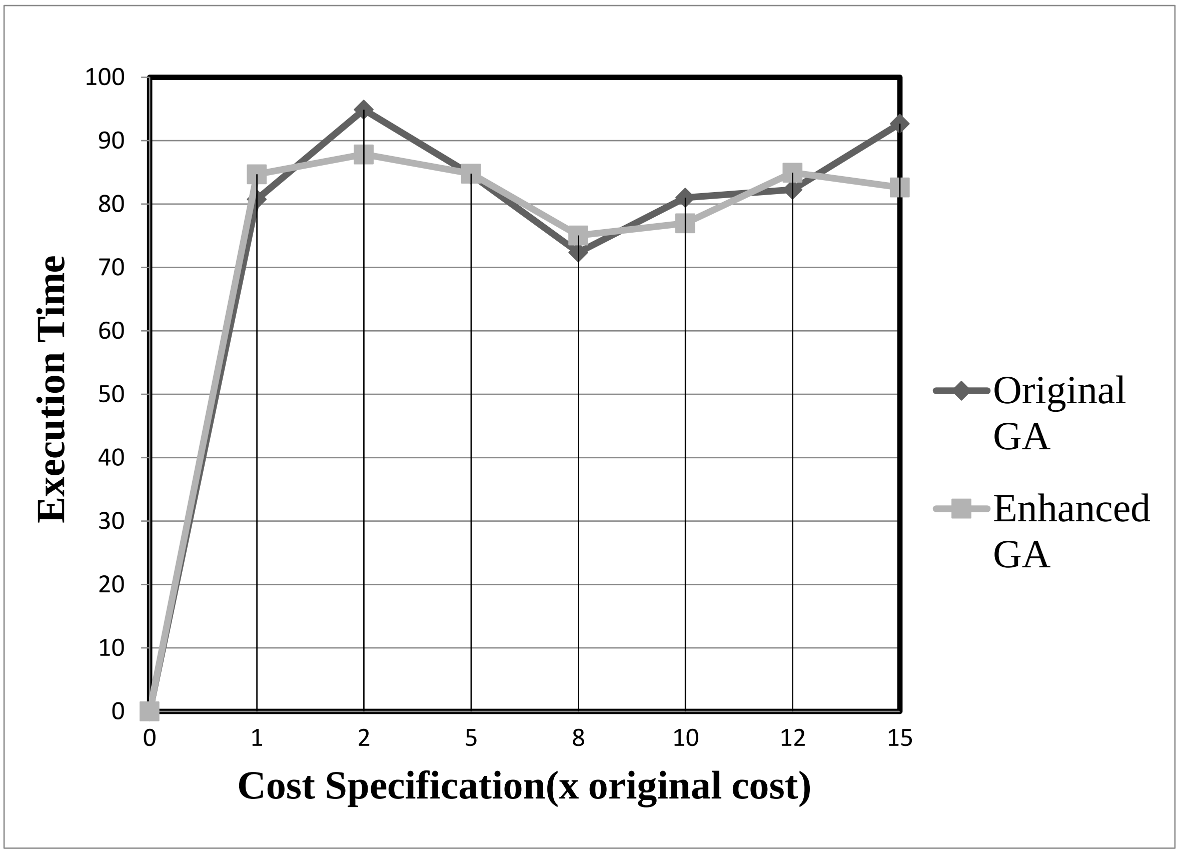
Original Cost Specification:

Cost per Memory	=	0.05 units
Cost per Storage	=	0.1 units
Cost per B/W	=	0.1 units

Now, we measure performance in terms of Execution Time for different cost specifications:

Cost Specification (x original cost)	Original GA	Enhanced GA
0	0	0
1	80.78	84.72
2	94.89	87.86
5	84.66	84.83
8	72.37	75.06
10	81	76.98
12	82.3	84.95
15	92.67	82.66

Table 4.9: Execution Time of OGA and EGA for different cost specifications



Graph 4.9: Execution Time of OGA and EGA for different cost specifications

In the above graph 4.9 and table 4.9, different specifications are as follows:

No of cloudlets	=	300
No of VMs	=	5

Original Cost Specification:

Cost per Memory	=	0.05 units
Cost per Storage	=	0.1 units
Cost per B/W	=	0.1 units

CHAPTER-5

CONCLUSION

5.1 CONCLUSION

From experimental result section, it is clear that proposed grey wolf optimization algorithm provides better QoS (Quality of service) as compared to previous proposed GA algorithm. The main idea of this algorithm in cloud computing is to complete maximum number of requests with least execution time, proposed algorithm shown that it can provide better execution time over large requests with reduces average start time and average finish time over the system. Proposed algorithm reduces the number of iteration required to achieve a global best solution with least scheduling time. This strategy has proven that it provides better QoS in term of high reliability with increase in number of requests and resources with least scheduling time with decrease in execution time with increase in population size and number of requests. Proposed algorithm insures the schedule achieve is global best solution

Cloud computing till date is one of the most rapid evolving parts in the IT market. Methodologies based on simulation are becoming popular in the market and the well educated society to analyse frameworks for cloud computing, application behaviours and their safety. A very little simulators are being particularly evolved for performance evaluation of environments of cloud computing including Greencloud, Cloudsim, Cloudanalyst, Networkcloudsim, Mdcsim as well as EMUSIM although the simulation environments for data centres of cloud computing made available for use of public is limited. The Cloudsim simulator is pre assumedly the most purified among the different reviewed simulators.

Nevertheless there have been several cloud simulators made available, we may assume that opting a simulator depends a lot on the kind of problem in hand as there are different types of simulators which are usually geared up for specific kinds of research problems in hand. CloudSim being a solely general purpose simulator, is suggested depending upon its popularity and characteristics in the research society.

Hence, we have evolved the CloudSim toolkit for simulating and modeling extensible Clouds. The research papers that are published by different authors relating to the Cloud are read thoroughly and different log files available on the internet of various sizes are downloaded and incorporated in the project. A graphical user interface is designed wherein freedom to input the number of instructions he/she wants to work upon is given. The final project was then tested by incorporating Google dataset i.e. real time dataset.

5.2 FUTURE SCOPE

In the future the project can be enhanced by implementing a GUI or a graphical user interface in which the user may input the number of instructions he/she wants to execute from the log file at a particular time rather than executing the whole lot of instructions. Furthermore, when the project is completed using the CloudSim toolkit it is tested on Google dataset i.e. the real time dataset. Rather than finding dummy results on dummy datasets we work on the real-time dataset.

Moreover, the latest studies have shown that the data centers intake an unprecedented amount of electrical power and as a result of which, they incur huge amounts of capital expenditure for day-to-day management, handling and operation. As an example, we have that a Google data center intakes power approximately equivalent to that used by an entire city like San Francisco. The natural conditions and the socio-economic factors of the geographical region where a data center is located together directly affect the total sum of power bills incurred. As an example, a data center at a location where the cost of power is low and has lesser hostile weather conditions comparatively tends to incur lesser amount of power bills. To get simulation of the above mentioned Cloud computing infrastructure, a lot of our future work will research latest techniques and models for the allotment of the services to new applications which depend upon the cost of service providers and also the energy effectiveness.

Our work will be the first attempt towards innovating and creating an approach and later the tool for examining large scale of behaviour of distributed applications through simulation of the Cloud computing environments. Hence, the tool will develop and evolve over the time, and the process resulting into the improvement of the quality of the model and of the evaluation supported by it. In the long run, this type of simulation experiment will result into a huge potential to support the testers to recognise new issues and characteristics, model them, and later develop and analyse more new algorithms and mechanisms for the purpose of resource management, this way enhancing the operation of the evolving Cloud applications.

5.3 APPLICATIONS AND CONTRIBUTIONS

- First of all efforts are needed for designing the software at different levels like compiler, OS, application and algorithm which allow the system to attain wide energy effectiveness. Although the providers of SaaS can still reuse the already deployed software, they require evaluating the dynamic behaviour of the applications at run time. The collected empirical data may then utilised in energy efficient resource provisioning and scheduling. The operating systems and the compiler are required to be designed in a manner such that the resources may be allotted to the application based on the respective requirement of the level of operation, and hence the energy consumption versus performance trade-off may be managed easily.

- To encourage the green Cloud data centers, the providers of the Cloud are required to measure and understand ongoing data center energy and anti heating designs, power requirements of the servers along with their cooling power consumption, and also the equipment of resource utilization in order to achieve the greatest efficiency. In addition to these, the modeling tools are also required to analyse the energy consumption of almost all the constituents and services of the Cloud, from the data center to the user PC where the hosting of the Cloud services takes place.

- For developing the design of the holistic solutions in the resource provisioning and scheduling of applications existing within the same data center, all the features including the Central Processing Unit, cooling, memory as well as network, must manadatorily be considered. As an example, the integration of VMs nevertheless the efficient technique to reduce the overall power consumption of the data center, addresses the problem regarding the urgent required redundancy as well as the placement geo-diversity which are needed to be evolved to meet the demands of SLAs with users.

- At the last of all, the accountability even goes to both the customers and the providers to ensure that evolving technologies and techniques never lead to irreversible modifications that may eventually lead to threat for the health of the human society as a whole. The manner in which the end users do an interaction with the application has also a much real impact as well as cost. As an example, we have merging of unsolicited emails may eradicate the energy wasted in the network and storage. In a similar way, if Cloud providers desire to provision a truly renewable and green Cloud, they are required to host their data centers near to the location of the renewable sources of energy as well as increase the consumption if the Green energy in their data centers established previously. Also Before the addition of the new and recent technologies which include virtualization, a proper evaluation of the overhead requires to be done to obtain the real advantage in terms of energy effectiveness.

REFERENCES

1. B. Richard. "Report to congress on server and data center energy efficiency: Public law 109-431.", Lawrence Berkeley National Laboratory , pp:0-10, 2008.

2. Nathani, Amit, Sanjay Chaudhary, and Gaurav Somani. "Policy based resource allocation in IaaS cloud." Future Generation Computer Systems28.1 (2012): 94-103.

3. Gao, Y., Guan, H., Qi, Z., Hou, Y., & 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.

4. Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A genetic algorithm (GA) based load balancing strategy for cloud computing.Procedia Technology, 10, 340347.

5. Quang-Hung, N., Nien, P. D., Nam, N. H., Tuong, N. H., & Thoai, N. (2013). A genetic algorithm for power-aware virtual machine allocation in private cloud. In Information and Communication Technology (pp. 183-191). Springer Berlin Heidelberg.

6. Jena, R. K. (2015). Multi Objective Task Scheduling in Cloud Environment Using Nested PSO Framework. Procedia Computer Science, 57, 1219-1227.

7. Abrishami, S., & Naghibzadeh, M. (2012). Deadline-constrained workflow scheduling in software as a service cloud. Scientia Iranica, 19(3), 680-689.

8. Bhoi, U., & Ramanuj, P. N. (2013). Enhanced max-min task scheduling algorithm in cloud computing. International Journal of Application or Innovation in Engineering and Management, 2(4), 259-264.

9. Gupta, P., & Ghrera, S. P. (2015). Load and Fault Aware Honey Bee Scheduling Algorithm for Cloud Infrastructure. In Proceedings of the 3rd International Conference on

Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014 (pp. 135-143). Springer International Publishing.

10. Komaki, G. M., & Kayvanfar, V. (2015). Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. Journal of Computational Science, 8, 109-120.

11. Wickremasinghe B, Calheiros RN, Buyya R. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. InAdvanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on 2010 Apr 20 (pp. 446-452). IEEE.

12. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1), 23-50.

13. Suraj, S. R., and R. Natchadalingam. "Adaptive Genetic Algorithm for Efficient Resource Management in Cloud Computing." International Journal of Emerging Technology and Advanced Engineering 4.2 (2014): 350-356.

14 Santhosh R, Ravichandran T. Non-Preemptive Real Time Scheduling using Checkpointing Algorithm for Cloud Computing. International Journal of Computer Applications. 2013 Jan 1;80(9).

15 Nathani A, Chaudhary S, Somani G. Policy based resource allocation in IaaS cloud. Future Generation Computer Systems. 2012 Jan 31;28(1):94-103.

16 Wickremasinghe B, Calheiros RN, Buyya R. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. InAdvanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on 2010 Apr 20 (pp. 446-452). IEEE.

17 Bhoi U, Ramanuj PN. Enhanced max-min task scheduling algorithm in cloud computing. International Journal of Application or Innovation in Engineering and Management. 2013 Apr;2(4):259-64.

18 Chen H, Wang F, Helian N, Akanmu G. User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. InParallel Computing Technologies (PARCOMPTECH), 2013 National Conference on 2013 Feb 21 (pp. 1-8). IEEE.

19 Jiang J, Ma S, Li B, Li B. Symbiosis: Network-aware task scheduling in data-parallel frameworks. InComputer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on 2016 Apr 10 (pp. 19). IEEE.

20 Liu Z, Wang S, Sun Q, Zou H, Yang F. Cost-aware cloud service request scheduling for SaaS providers. The Computer Journal. 2013 Feb 5:009.