MONKEY SEARCH ALGORITHM BASED TASK SCHEDULING IN CLOUD IAAS

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

in

Computer Science and Engineering/Information Technology

By

Prateek Tewari (131242)

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

I hereby declare that the work presented in this report entitled **" Monkey Search Algorithm Based Task Scheduling in cloud IAAS"** in partial fulfillment 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/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 Grade II, Department of Information Technology).

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

(Student Signature)

Prateek Tewari, 131242.

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

(Supervisor Signature)

Dr.Punit Gupta

Assistant Professor(Grade II)

Department of Information 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 "Monkey Search Algorithm Based Task Scheduling in Cloud IAAS" 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 really thank him for the constant unstinted support and invaluable guidance throughout the completion of my project.

TABLE OF CONTENT

TOPIC **PAGE NUMBER**

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
SAAS	Software As A Service
PAAS	Platform As A Service
IAAS	Infrastructure As A Service
Qos	Quality of Service
MIPS	Millions of instructions per second
MSA	Monkey Search Algorithm
PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization
VMM	Virtual Machine Monitor
VM	Virtual Machine
SLA	Service Level Agreement

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

ABSTRACT

Monkey search algorithm is a task scheduling algorithm in which we schedule tasks by allocating them to resources.In this project resources are designated as virtual machines(or mountains) and tasks are designated as cloudlets(or monkeys).Earlier many task scheduling algorithms are proposed to improve the performance but are not cost efficient at the same time.PSO and ACO are efficient solutions but not cost efficient.So to overcome the issues I have proposed learning based cost efficient algorithm for cloud infrastructure. In early proven task scheduling algorithms network cost is not included but in my proposed MSA network overhead or cost is taken into consideration which thus improves the efficiency of the algorithm as compared to previous algotithms.I have made comparison of my proposed MSA with FCFS algorithm as well as with ACO. Also I have used cloud simulation which is a framework of modeling and simulation of cloud computing infrastructures and sevices and real time as well as local cloud environment. The poposed MSA works well and is much improved in terms of efficiency and execution cost.

CHAPTER 1 INTRODUCTION

1.1 INTRODUTION

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.

Fig. 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

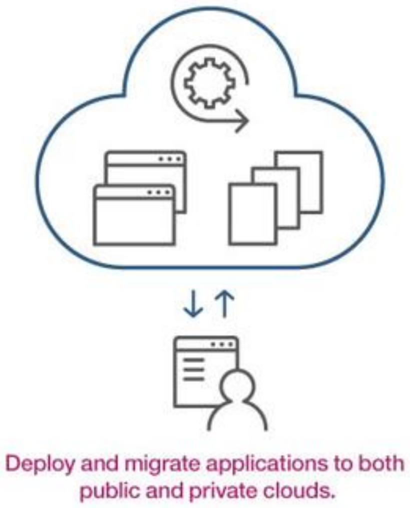


Fig.2: PAAS

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

Fig.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 public cloud are:

- Hosted at a service provider site
- Supports more than one customer
- Often utilises shared infrastructure
- Suited for information that is not highly confidential

Fig.4: Public cloud

(ii) Private Cloud

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 private cloud are:

- Hosted at an enterprise or service provider site
- Does not utilise shared infrastructure
- Supports one customers
- Suited for information that is sensitive

Fig.5: Private Cloud

(iii) Hybrid Cloud

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.

(iv) Community Cloud

The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party.

1.2 PROBLEM STATEMENT

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.

The problem lies in the scheduling and managing resources in cloud IaaS posed by migration of computation to cloud platform. So performance depends on scheduling method and the space of tasks is complex. Therefore the need of task scheduling algorithm like MSA comes into picture as I provides the most efficient solution in terms of least execution cost as compared to other task scheduling algorithms like PSO,ACO ,MaxMin ,MinMin etc.Task scheduling and resourse allocation are major problems involved in cloud as well as grid computing. From the study of related work, we concluded that the existing scheduling strategies in clouds are based on the approaches developed in related areas such as distributed systems and Grids.

1.3 OBJECTIVES

The 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 I have proposed a task scheduling monkey search algorithm(MSA) 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 algorithm 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. To reduce time and increase efficiency of computational performance in cloud computing. Overall to build cloud computing simulators and develop a learning based cost efficient scheduling algorithm for cloud infrastructure

1.4 METHODOLOGY

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

Following Table 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: Comparison of various Cloud Simulators

In this project, I 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.

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

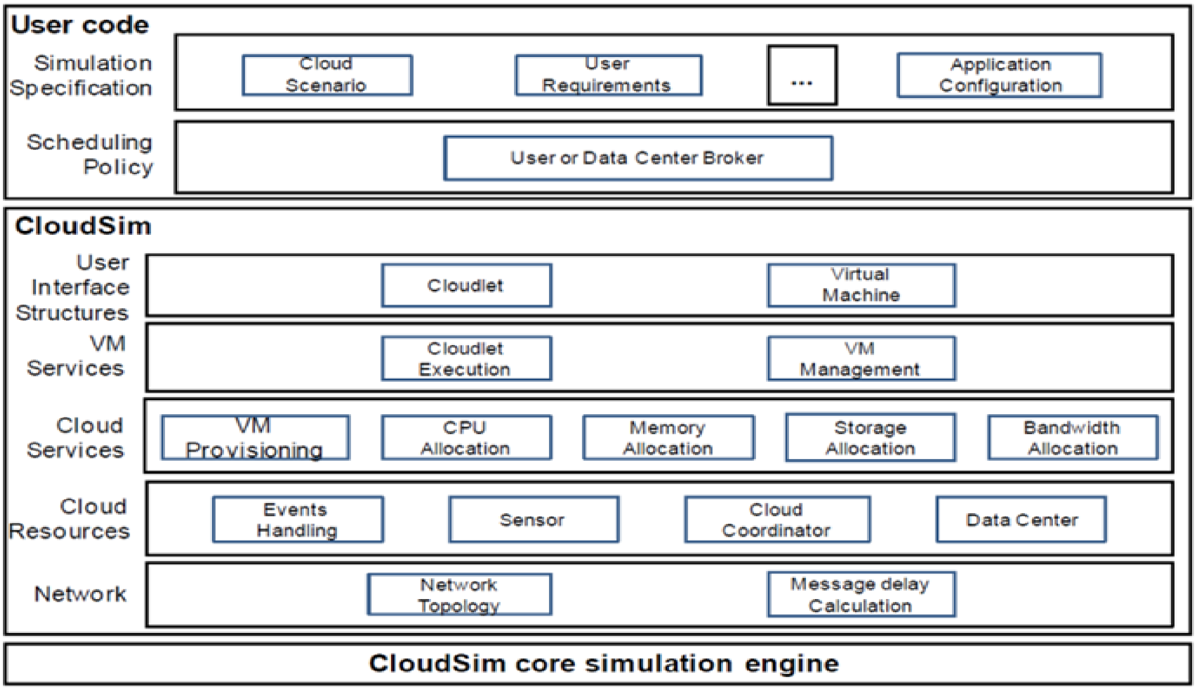


Fig 6 : Layered CloudSim architecture

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 VM's to various hosts.

CHAPTER 2 LITERATURE SURVEY

2.1

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 7 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.

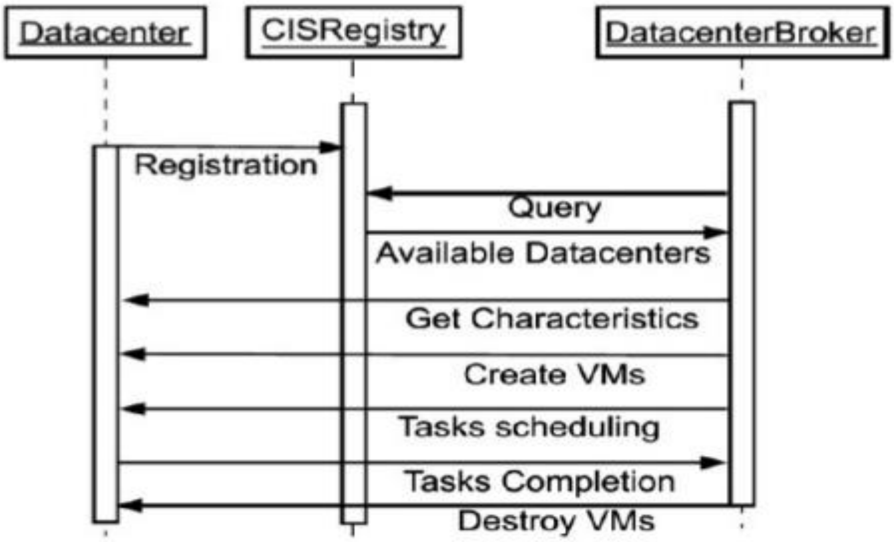


Figure 7 : Simulation Data Flow

Figure 8 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.

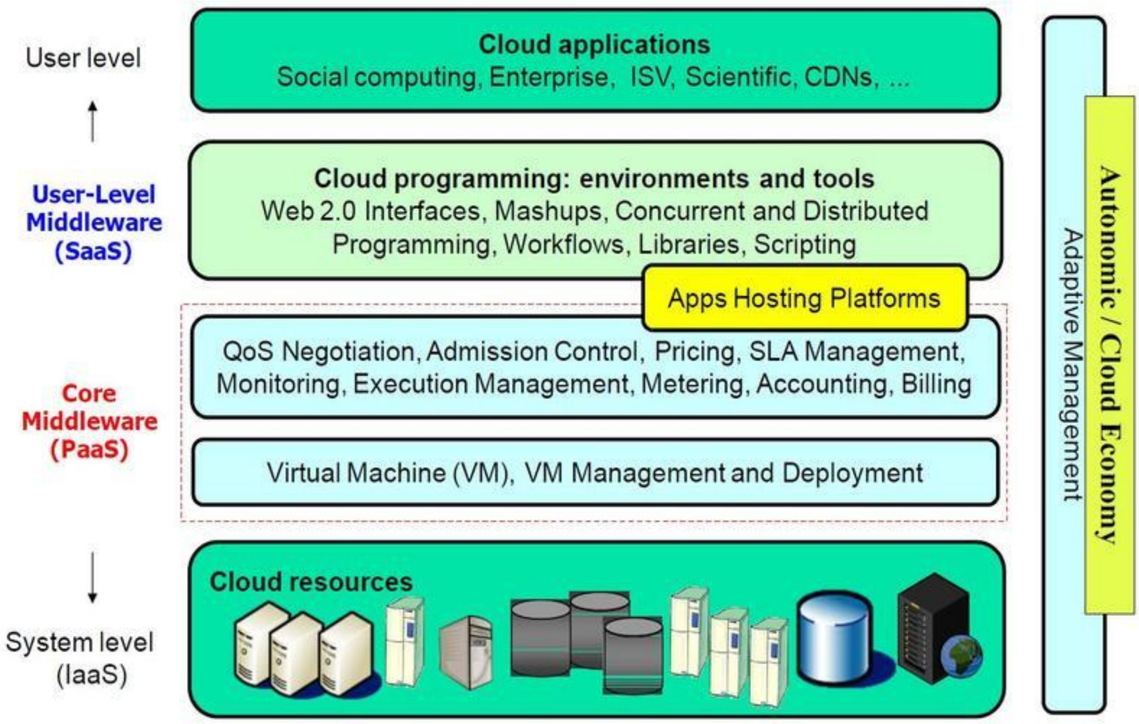


Figure 8 : Layered cloud computing architecture

2.2 A Survey of PSO Based Scheduling Algorithm in Cloud Computing

Mohammad Masdari[1] • Farbod Salehi[1] • Marzie Jalali[1] • Moazam Bidaki[2]

Particle Swarm Optimization is a cooperative, population-based global search swarm intelligence metaheuristic, presented by Kennedy and Eberhart in 1995. Furthermore, it is a powerful optimization technique for solving multimodal continuous optimization problems . In this algorithm, the swarm of particles is randomly generated initially, and each particle position represents a possible solution. Each particle is positioned in the search space and has a fitness value and velocity to determine the speed and direction of its moves. Particles move around in the search space based on the particles' updated position and velocity to get an optimized solution . After the repeated advances, which are also called iteration, particle swarm gradually approaches the optimal location. Then, the optimal solution will be reached .

2.3 Workflow Scheduling in Multi-Tenant Cloud Computing Environments

Bhaskar Prasad Rimal, Student Member, IEEE and Martin Maier, Senior Member, IEEE

Multi-tenancy is one of the key features of cloud computing. In a conventional single-tenant cloud architecture, providers offer a dedicated cloud service (instance of application and underlying infrastructure) to the tenants (customers), where no data is intermingled with other tenants. From the15 service provider's point of view, this model does not provide scalable cloud services and economics of scale. On the other hand, in multi-tenant cloud computing, infrastructures, applications, and database are shared among all tenants. At the downside, tenants may not be able to customize their use of cloud services in order to fit their specific needs. Moreover, multi-tenancy may be seen differently from a cloud service model perspective. For instance, as for Infrastructure-as-a-Service (IaaS), tenants have the ability to provision computing, storing, and networking resources. Even though multi-tenancy allows cloud service providers to better utilize computing resources, supporting the development of more flexible services based on economy of scale, and reducing infrastructural costs. Workflow scheduling is a process of mapping and managing the execution of inter-dependent tasks on distributed resources. It allocates suitable resources to workflow tasks such that the execution can be completed to satisfy objective functions imposed by users. The proper and efficient scheduling can have significant impact on the performance of the workflow system.

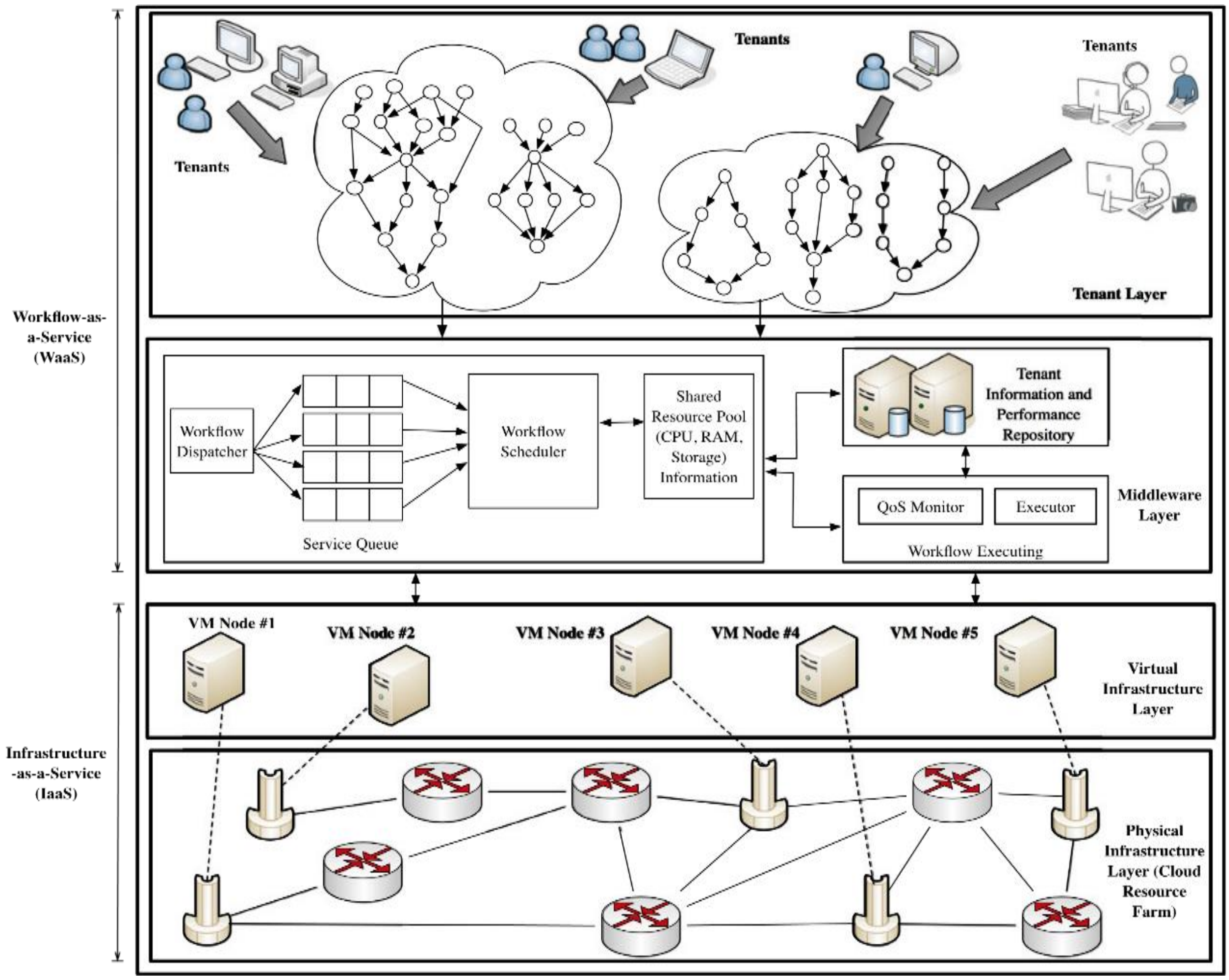


Fig.9: Architecture of workflow scheduling in multi-tenant cloud environment

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:

- JAVA
- 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.

Proposed MSA for task allocation is divided into 4 phases as follows.

I. Initialization

II. Climb

III. Watch and Jump

IV Somersault

I. Initialization

In this phase we have a set of tasks (T1, T2, T3, T4, T5, T6…. T n) in terms of cloudlets and a set of resources in term of virtual machine (VM1, VM2, VM3, VM4, VM5…. VM m) which are pre allocated on hosts in distributed datacenters and we initialize asset of sequences or schedules allocated randomly.

II. Climb

In this phase monkey designated as cloudlet climbs/schedule on the mountain designated as vm which ever comes first in its path.

III. Watch and Jump

In this phase monkey searches in its local domain to find most optimal solution. For each monkey, when it gets on the top of the mountain, it is natural to have a look and to find out whether there are other mountains around it higher than its present positions. If yes, it will jump to some place on the mountain watched by it from the current position (this action is called the 'watch–jump process') and then repeat the climb process until it reaches the top of the mountain.

Where

VM_MIPS_i : MIPS of i^{th} virtual machine

T_Leng_i : Length of i^{th} Task

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

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

$$Total\_time = \sum_{i=1-n} \frac{\text{T\_Lenght i}}{VM\_MIPSi} \quad (5)$$

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.

Start : cost of allocating task to resource

Cost : cost of execution time

Cost1 : network cost involved

T_cost : The total cost of allocating the task

Cost = (Cost +Cost1 + vm1.start);

$$Total\_cost = \sum_{i=1-n} Cost_i$$

The fitness function is given as:

Fitness_function = Cost + Cost 1 + $vm_{i.}$ .start (6)

Based on the fitness function value the least network cost path is chosen and proceed for final fourth phase.

IV. Somersault

In this phase after repetitions of the climb process and the watch–jump process, each monkey will find a locally maximal mountaintop around its initial point the most optimal solution i.e least execution cost path is found and monkey rolls down from the mountain.

Following three algorithms discourse the proposed algorithm:

Algorithm : Proposed Task Scheduling Algorithm

---

1. Proposed(VM_n , T_n)

Input: Number of VM's VM_n , Number of Tasks T_n

2. VM_n <- Number of VM's

3. T_n <- Number of tasks

4. Path<- Find(VM_n ,T_n)

5. Allocate Tasks

Ouptut: All tasks are allocated/scheduled.

Proposed algorithm (a)

Algorithm: To get least execution cost path

---

1. Find(VM_n , T_n)

Input: Number of VM's VM_n , Number of Tasks T_n ,Start_time start

2. climb the first mountain in the domain

3. evolve new mountains in the search domain higher than current one

   and with least network cost i.e call Evolve()

4. Jump on the new searched mountain with desired characteristics

5. Repeat step 3 and 4 until highest mountain in search domain is found

Output: Least Execution cost path found

Proposed algorithm (b)

Algorithm: Evaluation

```
1. Evolve()
2. for each i 0 to getVmsCreatedList().size()
3.     vm1=getVmsCreatedList().get(i)
4.     double cost=cloudlet.getCloudletLength()/vm1.getMips()
5.     double cost1=NetworkTopology.getDelay(vm1.getHost().getid(),
                 getVmsCreatedList().get(i).getUserId())
6.     cost=cost+cost1+vm1.start
7.       if cost<fcost
8.        vm=vm1
9.        fcost=cost
10.     End if
11. End for
```

Output: total cost found

Proposed Algorithm(c)

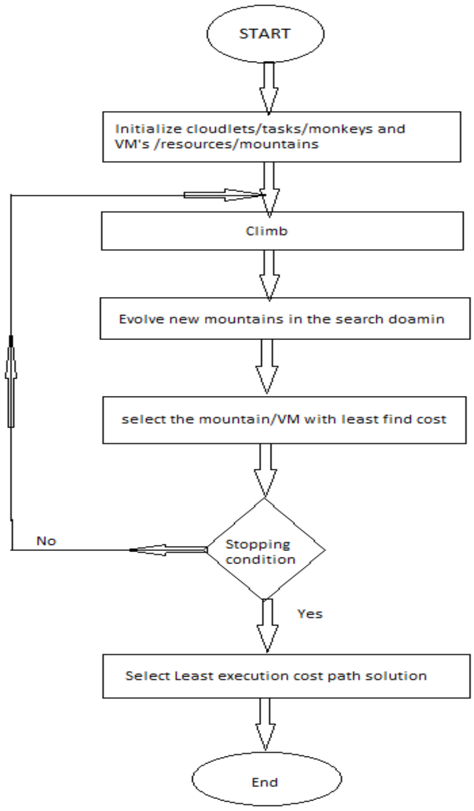


Fig. 10: Flow Diagram

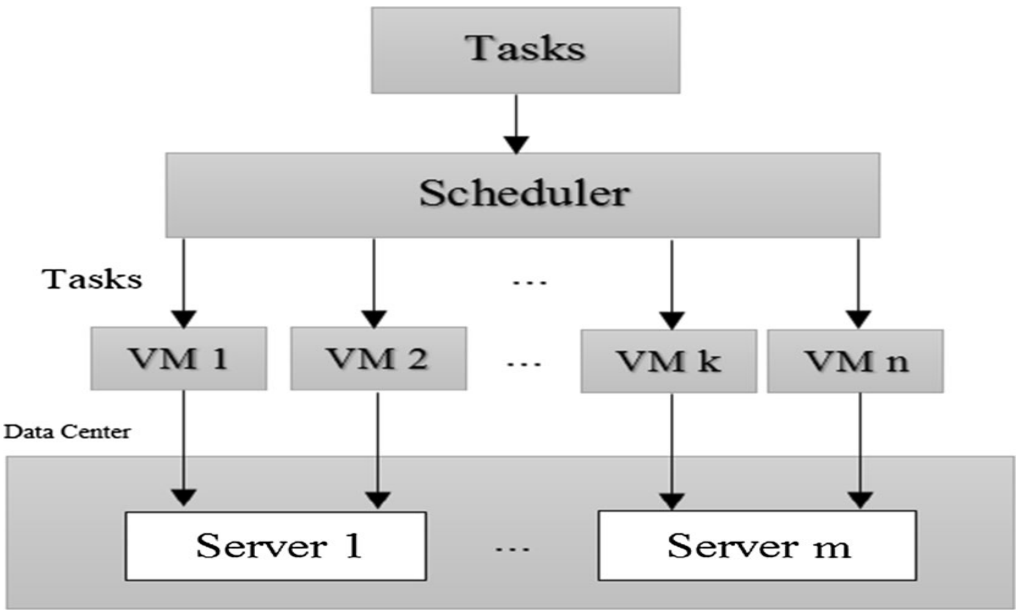


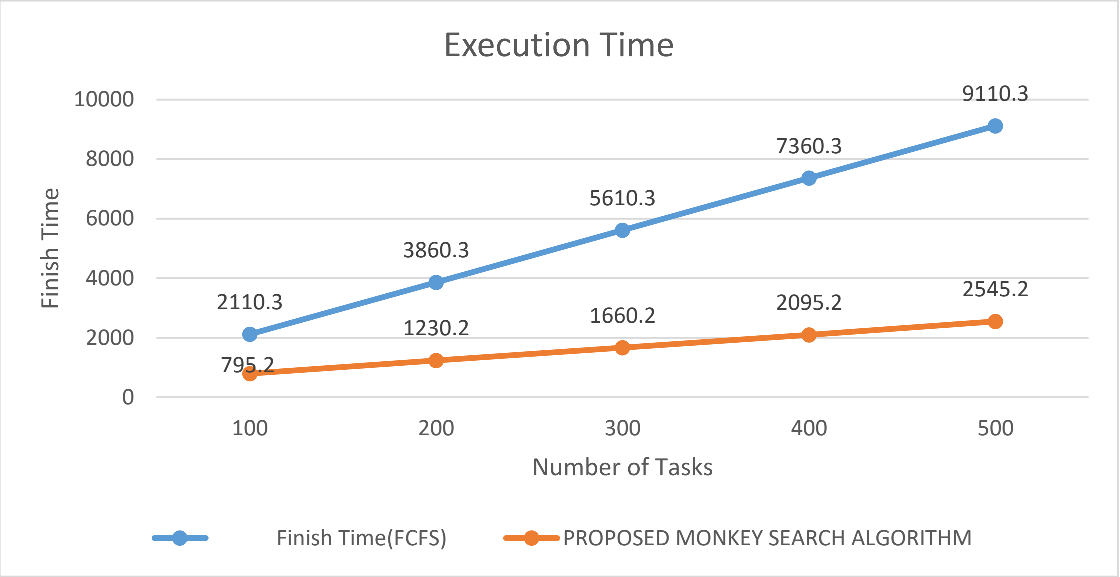
Fig.11: Task scheduling in cloud environment

CHAPTER 4 PERFORMANCE ANALYSIS

For simulation we CloudSim 3.0 power module is used. CloudSim 3.0 provides cloud simulation and predefined power model simulation. Initial task is task scheduling in local environment which has been done using two algorithms, namely FCFS (First Come First Serve) and Proposed MSA(Monkey Search Algorithm). These algorithms are run on the CloudSim simulator for number of VMs = 30.

NO.OF CLOUDLETS	PROPOSED MSA	FCFS
100	795.2	2110.3
200	1230.2	3860.3
300	1660.2	5610.3
400	2095.2	7360.3
500	2545.2	9110.3

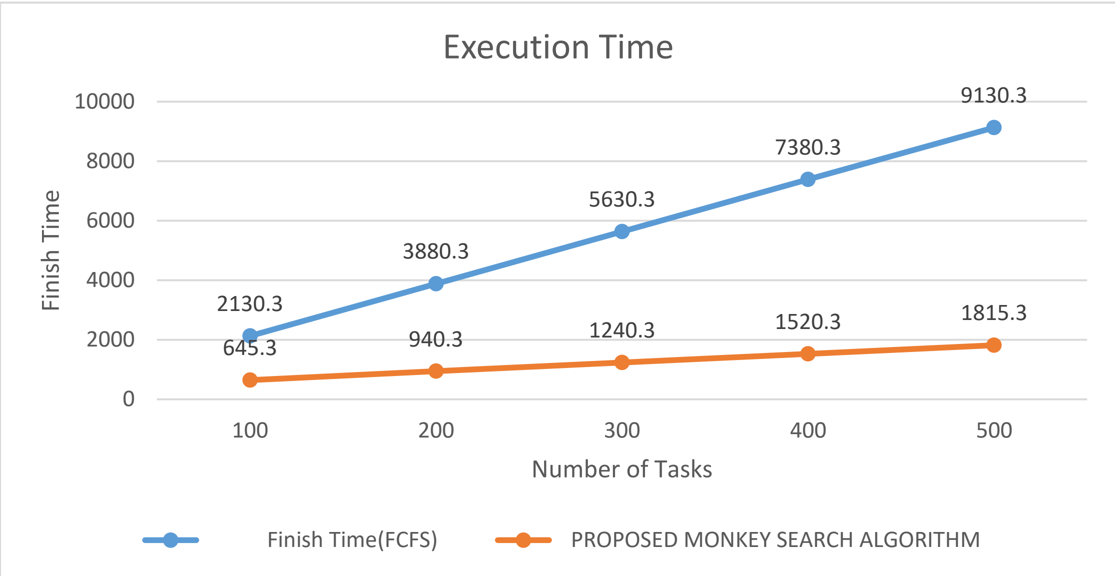
Table 2 : Least Execution Cost of Proposed MSA and FCFS with two datacenters



Graph 1: Least Execution Cost of Proposed MSA and FCFS with two datacenters

NO.OF CLOUDLETS	PROPOSED MSA	FCFS
100	645.3	2130.3
200	940.3	3880.3
300	1240.3	5630.3
400	1520.3	7380.3
500	1815.3	9130.3

Table 3 : Least Execution Cost of Proposed MSA and FCFS with three datacenters

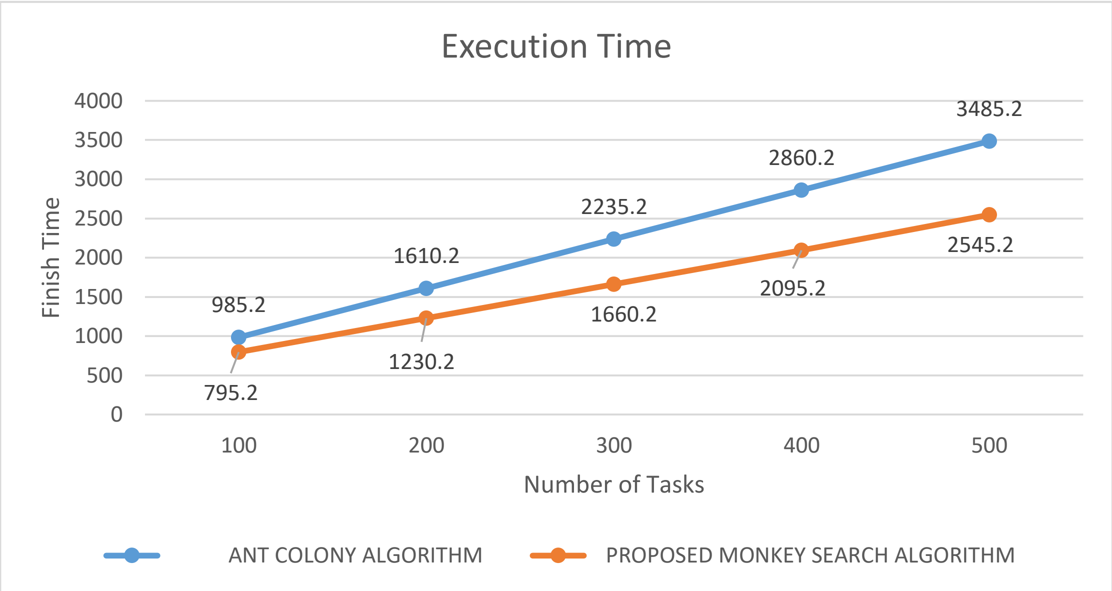


Graph 2 : Least Execution Cost of Proposed MSA and FCFS with three datacenters

Comparison of proposed MSA with ACO:

NO.OF CLOUDLETS	PROPOSED MSA	ACO
100	795.2	985.2
200	1230.2	1610.2
300	1660.2	2235.2
400	2095.2	2860.2
500	2545.2	3485.2

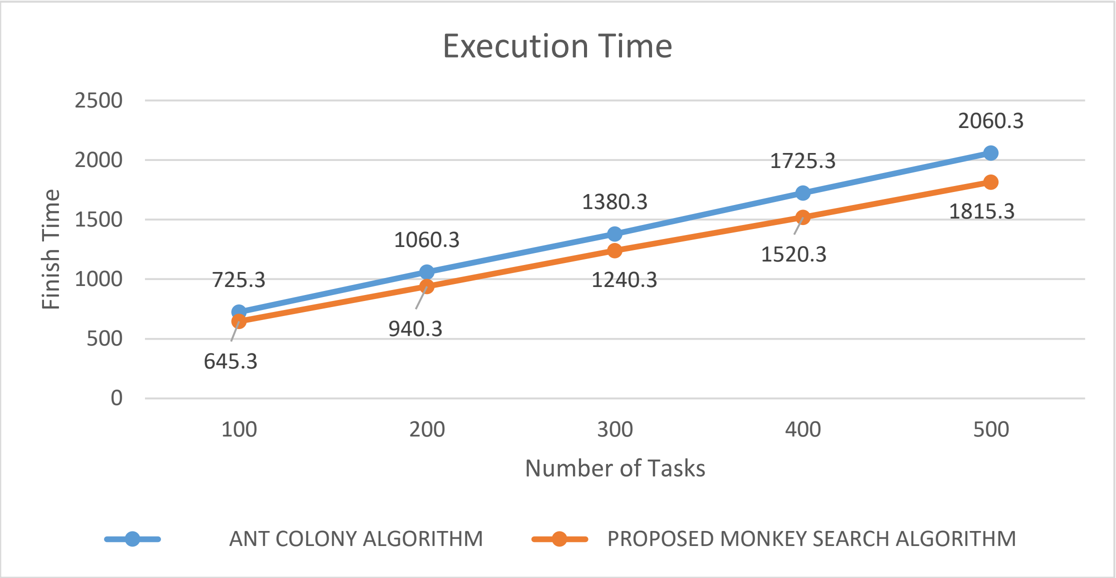
Table 4 : Least Execution Cost of Proposed MSA and ACO with two datacenters



Graph 3: Least Execution Cost of Proposed MSA and ACO with two datacenters

NO.OF CLOUDLETS	PROPOSED MSA	ACO
100	645.3	725.3
200	940.3	1060.3
300	1240.3	1380.3
400	1520.3	1725.3
500	1815.3	2060.3

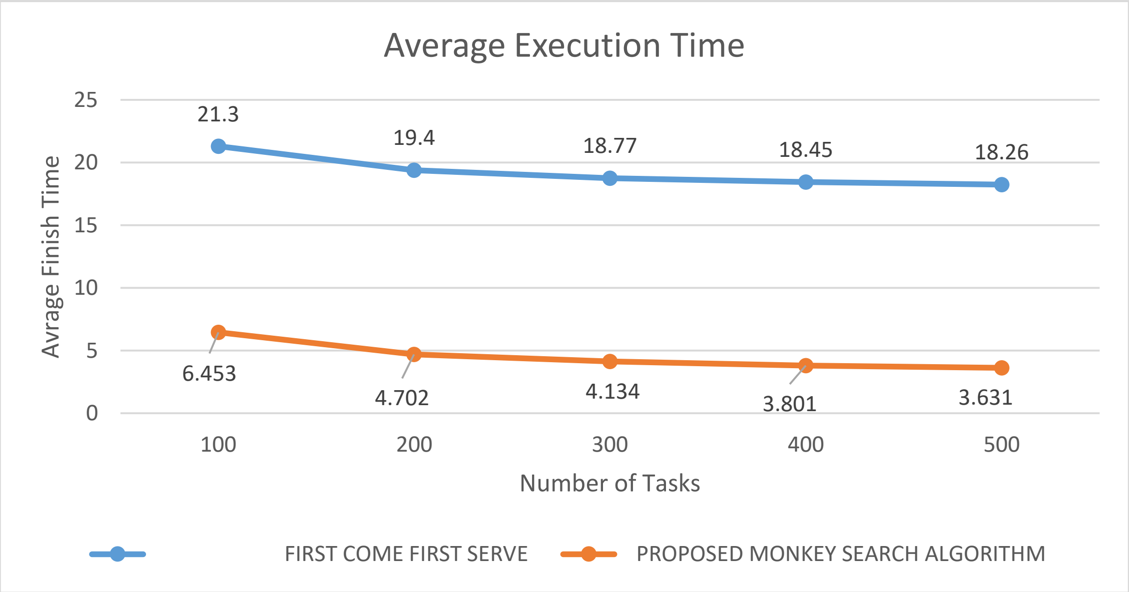
Table 5 : Least Execution Cost of Proposed MSA and ACO with three datacenters



Graph 4 : Least Execution Cost of Proposed MSA and ACO with three datacenters

NO.OF CLOUDLETS	PROPOSED MSA	FCFS
100	6.453	21.30
200	4.702	19.40
300	4.134	18.77
400	3.801	18.45
500	3.631	18.26

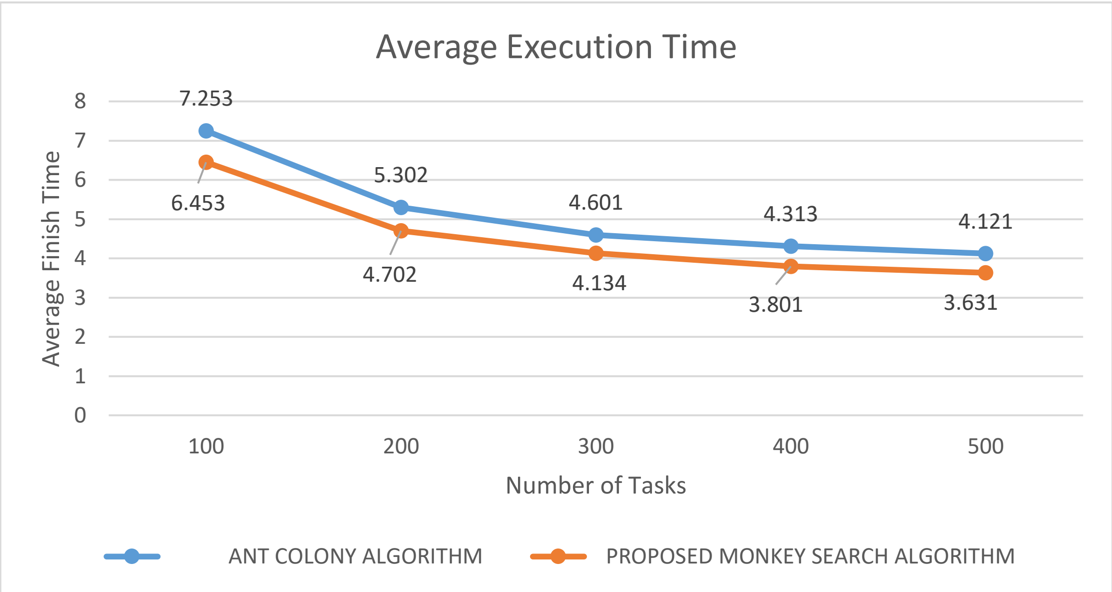
Table 6: Average Execution Cost of Proposed MSA and FCFS with three datacenters



Graph 5 :Average Execution Cost of Proposed MSA and FCFS with three datacenters

NO.OF CLOUDLETS	PROPOSED MSA	ACO
100	6.453	7.253
200	4.702	5.302
300	4.134	4.601
400	3.801	4.313
500	3.631	4.121

Table 7: Average Execution Cost of Proposed MSA and ACO with three datacenters



Graph 6 :Average Execution Cost of Proposed MSA and ACO with three datacenters

CHAPTER 5 CONCLUSIONS

5.1 CONCLUSION

From experimental result section, it is clear that proposed monkey search task scheduling algorithm provides better task allocation and cost efficiency as compared to previous proposed task scheduling algorithm. The main idea of this algorithm in cloud computing is to complete maximum number of requests with least execution time and cost, proposed algorithm shown that it can provide better execution time over large requests and least cost as MSA task scheduling algorithm are known to find best global solution that cannot be achieved by any static or dynamic resource allocation algorithms.

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 data set.

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, 340-347.

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. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in Engineering Software. 2014 Mar 31;69:46-61.

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. Elzeki OM, Reshad MZ, Elsoud MA. Improved max-min algorithm in cloud computing. International Journal of Computer Applications. 2012 Jan 1;50(12).

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. 1-9). 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.

21. Monkey algorithm for global numerical optimization J. Uncertain [Zhao R Q and Tang W S 2008].

22. A modified monkey algorithm for optimal sensor placement in structural health

23. Sensor placement on Canton Tower for health monitoring using asynchronous-climb monkey algorithm [Ting-Hua Yi, Hong-Nan Li and Xu-Dong Zhang, 2008].

24. Human activity classification approach on smartphone using monkey search algorithm (2015 Seventh international conference on advance communication and networking) [Mustafa Zain, Amr Badr, Tai-Hoo Kim].

25. A Survey of PSO-Based Scheduling Algorithms in Cloud Computing by Mohammad Masdari [1] , Farbod Salehi[1] , Marzie Jalali[1] , Moazam Bidaki[2]