Improved Biogeography Based Optimization by Duplicate Replacement during Load Balancing in Cloud Computing

A DISSERATION

SUMMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

Master of Technology

Computer Science Engineering (Software Engineering)

By

Akanksha Gupta

2K14/SWE/03

Under the Guidance of:

Dr. Kapil Sharma

JUNE 2014-2016

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DELHI- 110042, INDIA

DECLARATION

We hereby declare that the **Major Project-II** work entitled **"Improved Biogeography Based Optimization by Duplicate Replacement during Load Balancing in Cloud Computing"** which is being submitted to **Delhi Technological University**, in partial fulfillment of requirements for the **award** of degree of Master Of Technology (Software Engineering) is a benefice report of **Major Project-II** carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Akanksha Gupta (2K14/SWE/03**)**

Department of Computer Science Engineering
Delhi Technological University, Delhi
(Formerly Delhi College of Engineering, University of Delhi)

CERTIFICATE

This is to certify that Major-II Project Report entitled **"Improved Biogeography Based Optimization by Duplicate Replacement during Load Balancing in Cloud Computing"** submitted by **Akanksha Gupta (University Roll no 2K14/SWE/03)** for partial fulfillment of the requirement for the award of degree Master Of Technology (Software Engineering) is a record of the candidate work carried out by her under my supervision.

Guide name

Dr. Kapil Sharma
Department of Computer Science Engineering
Delhi Technological University
(Formerly Delhi College of Engineering, University of Delhi)

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. My greatest thanks are to my parents who best owed ability and strength in me to complete this work.

I own a profound gratitude to my project guide **Dr. Kapil Sharma, Department of Computer Science Engineering, Delhi Technological University, Delhi** who has been a constant source of inspiration to me throughout the period of this project. It was his competent guidance, constant encouragement and critical evaluation that helped me to develop a new i n s i g h t into my project. His calm, collected and professionally impeccable style of handling situations not only steered me through every problem, but also helped me to grow as a matured person.

I am also thankful to him for trusting my capabilities to develop this project under his guidance.

Date : **Akanksha Gupta**

2K14/SWE/03

LIST OF FIGURE

LIST OF TABLE

TABLE OF CONTENT

ABSTRACT

Load balancing on cloud is a vital part of cloud computing algorithm. This is the obligation of cloud computing that we allocate the load continuously at each and every node of cloud. An exceedingly congested provider could plummet to furnish effectual services to its users. So we can state that, appropriate ability for arrangement to balance a load and also increase the throughput of the system. Environment of cloud computing is a combination of parallel and distributed arrangement that encompassing, set of consistent and nearby system. With the rising demand and benefits of cloud computing groundwork, disparate computing can be given on cloud environment. One of the frank subjects in this nature is connected to task scheduling. Cloud burden balancing is an NP-hard optimization setback, and countless meta-heuristic algorithms have been counseled to resolve it. The biogeography-based optimization (BBO) is moderately new and has the supremacy of adapting new good solutions. This BBO adaptive procedure is contrasted to the genetic algorithm and supplementary heuristic algorithms that are reproductive process. We have utilized the average BBO to find the optimal job design of cloud computing additionally encompassing enhanced Chaotic BBO in this paper. The Chaotic BBO fabricates the novel habitats by appealing the immigration and emigration for all particular non-elite habitats from the innumerable most excellent habitats and revises the habitats when novel created habitat is bigger than aged one recognized on chaotic maps. In this work, we counsel a distributed VM Burden balancing way shouted Chaotic BBO alongside elevated reliability and scalability above Cloud Workload Tracing.

CHAPTER 1

INTRODUCTION

"Cloud" in cloud computing hardware, webs, storage, services, and interfaces that aspect of computing as a service to hold to join can be described as a set [1][2]. Internet users set up cloud services on demand multimedia, transportation and storage are the foundation. Four key features are cloud computing: elasticity to scale and skills and the self-service system and automatic de-provisioning, request software design interfaces (APIs), billing and metering capacity of a pay-as-you-go custom Model Figure 1.1 on the web displays a generic cloud period. This flexibility is what people and companies to move to the cloud are appealing. Following a request to inadequate benefits are hosted on the cloud:

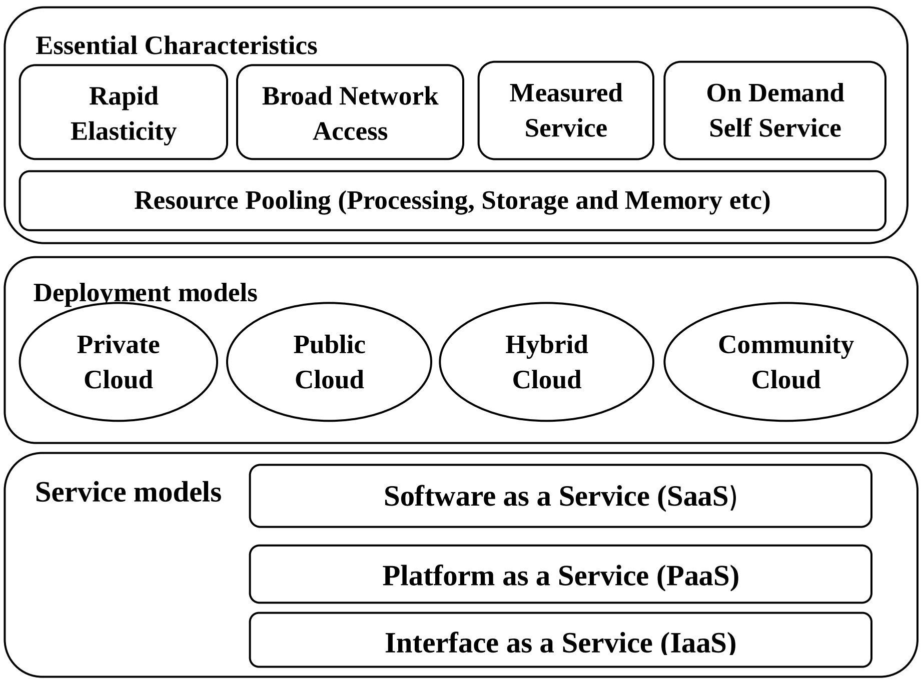


Figure 1 General architecture in cloud computing environment

- This is the perfect cost efficient method to use, sustain and also to upgrade.
- Cloud computing is more inexpensive and also reduces the company's expenditure. Here we can pay only for the cloud space.
- We can get the essential storage according to plans produced by the cloud provider.
- The cloud service supplier who is responsible for IT resources and maintenance.

- It is simple to access information all over the world by utilizing internet connection .we can also reserve the documents to our office staff.
- It also reduces the company's carbon discharge by 35%.

Cloud computing can completely convert the way companies utilize the technology to provide services to customers, partners, and suppliers. Furthermore businesses, such as Google and Amazon, already use most of their IT assets in the cloud. They have originated that it can remove many of the crucial constraints from the conventional computing environment, including space, power, time and cost [2].

Some example of cloud computing is Gmail, Yahoo email or Hotmail etc. Cloud computing is divided into three parts: "application" "connectivity" and "storage". Each and every part serves a different motivate and offers different commodity for businesses and individuals all over the world.

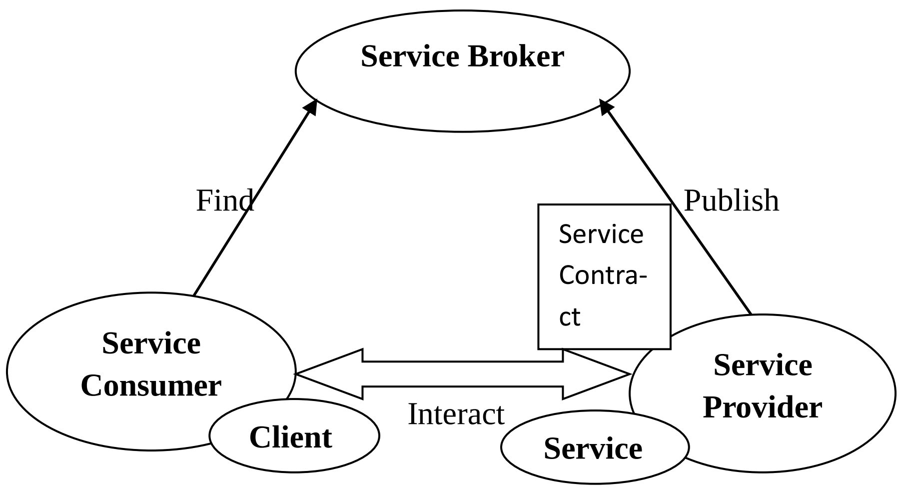


Figure 2 Cloud Platform on the web

1.1 Challenges in Cloud Systems

As cloud computing is in its evolving step, so there are many issues prevalent in cloud computing are following [3]:

1. Ensuring proper retrieval of control (authentication, authorization, and inspection).
2. Network level migration, so we can get minimum cost and time to migrate a job.
3. To provide perfect security to the data in movement and to the data at rest.

4. Data availability problem in cloud.
5. Legal quagmire and transitive trust concern.
6. Data lineage, inadvertent disclosure and data provenance of important information is possible.

1.2 Essential Characteristics:

1.2.1 On Demand Self-service:

A consumer can unilaterally deliver computing capabilities, for example server time and network storage, as required automatically without want human interaction with every service provider.

1.2.2 Broad Network Access:

Capabilities are available across the network and accessed by standard mechanisms that encourage use by heterogeneous thin or thick consumer platforms (e.g. tablets, mobile phones, laptops, and workstations).

1.2.3 Resource pooling

The producer's computing resources are pooled to provide multiple clients using a multi-tenant model, with dissimilar physical and virtual services dynamically assigned and reassigned according to client demand. For example resources include processing storage, memory, and network bandwidth [4].

1.2.4 Rapid elasticity:

Capabilities can be elastically delivered and released, in some cases automatically, to scale quickly outward and inward commensurate with request. To the client, the capabilities available for providing often appear to be infinite and can be appropriated in any amount at any time.

1.2.5 Measured service:

Cloud systems mechanically control and optimize supply use by leveraging a metering ability at some level of abstraction appropriate to the type of service (example processing, storage, bandwidth, and active customer accounts). Resource usage can be monitored, forbidden, and reported, providing transparency for both the supplier and consumer of the utilized service.

1.3 Service Models:

1.3.1 Software as a Service (SAAS):

The capability provided to the customer is to use the producer's applications running on a cloud communications [5]. The applications are easily reached from various client devices through either a thin consumer interface, example a program interface or a web browser. The customer does not manage or organize the underlying cloud communications including servers, network, operating systems, storage, or even person application capabilities, with the possible omission of limited user precise application configuration settings.

1.3.2 Platform as a Service (PAAS):

The capability provided to the customer is to install onto the cloud infrastructure customer created or received applications created using programming [6]. The cloud infrastructure can be outlook as containing both an abstraction layer and a physical layer. The abstraction layer resides of the software deployed crossways the physical layer, which manifests the necessary cloud characteristics. The physical layer resides of the hardware capitals that are necessary to bear the cloud services being produced, and typically includes storage, server and network components. Conceptually the abstraction layer sits upon the physical layer, Languages, services, libraries, and tools supported by the supplier.

1.3.3 Infrastructure as a Service (IAAS):

The customer does not manage or organize the underlying cloud infrastructure but has organize over storage, operating systems and installed applications and possibly limited organize of select networking mechanism (e.g., host firewalls). The data center hardware and software is what we will describe a Cloud. When a Cloud is completely available in a pay-as-you-go manner to the common public, we call it a Public Cloud; the service being retailed is Utility Computing[7][8]. We use the name Private Cloud to refer to internal data centers of a company or other organization, not made available to the common public.

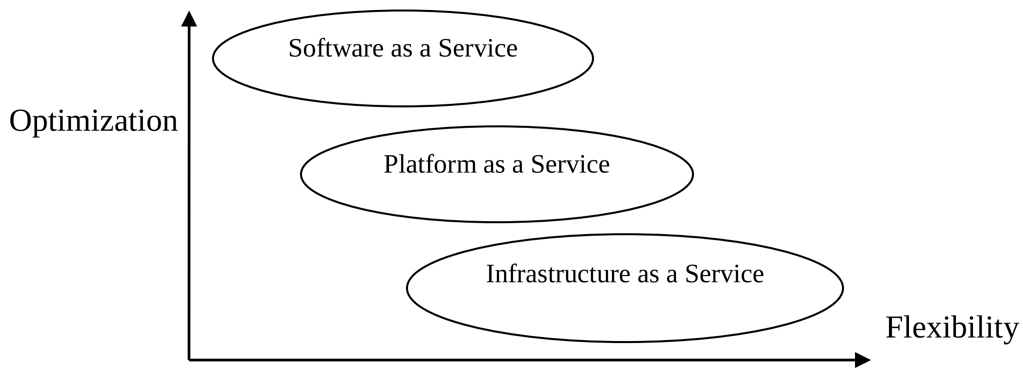


Figure 3 Software, Platform and Infrastructure Services

1.4 Cloud Deployment Models:

1.4.1 Private Cloud:

The cloud infrastructure is delivered for private use by a particular organization comprising multiple customers (e.g., business units). It may be managed, owned, and operated by the business, a third party, or some grouping of them, and it may be present on or off location.

1.4.2 Community Cloud:

The cloud infrastructure is delivered for exclusive use by a precise community of customers from organizations that have mutual concerns (e.g., security requirements, mission, guidelines, and compliance considerations). It may be owned, managed and operated by one or more of the business in the community, a third party, or some grouping of them, and it may be present on or off location.

1.4.3 Public cloud:

The cloud infrastructure is delivered for open use by the common public. It may be owned, managed, and operated by a company, academic, or government organization, or some grouping of them. It exists on the location of the cloud supplier.

1.4.4 Hybrid cloud:

The cloud infrastructure is a grouping of two or more distinct cloud infrastructures (private, community, or public) that have unique unit, but are bound mutually by standardized or proprietary knowledge that enables data and application portability (e.g., cloud satisfied for load Balancing between clouds) [9].

1.5 Cloud Load Balancing Challenges

There are various challenges of Cloud Load Balancing which are explained below [10]:

1. **Spatial Distribution of the Cloud Nodes**

A few algorithms are designed to be proficient only for an intranet or closely situated nodes where communication delays are minor. However, it is confronted to design load balancing algorithms that can effort for spatially distributed nodes. This is because other component must be taken into account for example the speed of the system links among the nodes, the distance between the customer and the job processing nodes, and the distances between the nodes involved in producing the service. There is a requirement to develop a way to organize load balancing method among all the spatial distributed nodes while being capable to effectively bear high delays.

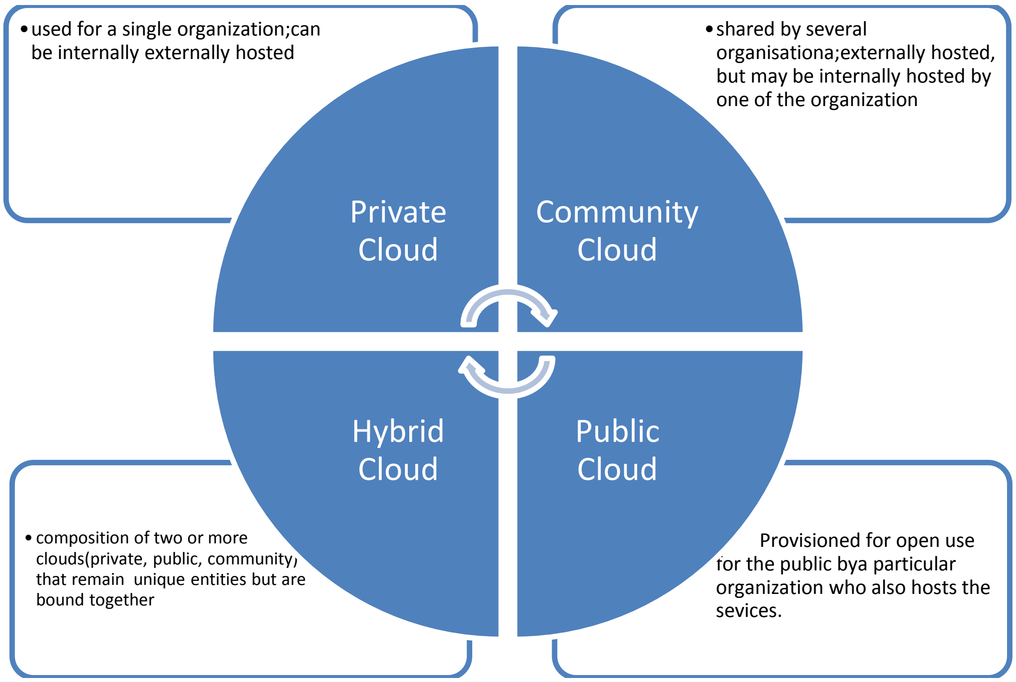


Figure 4 Types of Cloud

2. Storage/ Replication

A complete replication algorithm does not receive resourceful storage utilization into account. This is because the similar data will be reserved in all replication nodes. Full replication algorithms enforce higher costs since large storage is needed. However, partial replication algorithms could keep parts of the data sets in every node based on every node's capabilities for example processing power and capability. This could lead to improved utilization, yet it increases the difficulty of the load balancing algorithms as they effort to take into account the accessibility of the data set's parts transversely the different Cloud nodes.

3. Algorithm Complexity

Load balancing algorithms are chosen to be less complex in terms of execution and operations. The higher execution complexity would lead to an extra complex process which could be a reason of some negative performance issues. In addition, when the algorithms need more information and advanced communication for monitoring and organize, delays would cause an extra problems and the efficiency will fall. Therefore, load balancing algorithms must be considered in the simplest forms.

4. Point of Failure

Managing the load balancing and gathering data about the different nodes must be considered in a way that avoids having a particular point of failure in the algorithm. A few algorithms can produce efficient and effective methods for solving the load balancing in a certain model. However, they have the problem of one controller for the whole organization. In such cases, if the controller crashes, then the whole system would crashes.

5. **Throughput**

It is the overall number of tasks that have finished execution for a certain scale of time. It is necessary to have high through put for better performance of the business [11].

6. **Associated Overhead**

It defines the amount of overhead during the execution of the load balancing algorithm. It is a work of movement of tasks, inter processor and inter process communication. For load balancing method to work properly, least amount of overhead should be there.

7. **Fault tolerant**

We can characterize it as the ability to execute load balancing by the appropriate algorithm without random link or node crash. Every load balancing algorithm should have high-quality fault tolerance approach.

8. **Migration time**

It is the total of time for a procedure to be transferred from one organization node to another node for implementation. For improved performance of the system this time should be less.

9. **Response time**

In Distributed structure, it is the time taken by a particular load balancing method to respond. This time should be minimized for improved performance.

10. **Resource Utilization**

It is the constraint which gives the data within which present the resource is utilized. For efficient load balancing in organization, optimum resource should be utilized.

11. **Scalability**

It is the capability of load balancing algorithm for a system with any fixed number of central processing unit and machines. This parameter can be improved system performance.

12. **Performance**

It is the total efficiency of the system. If all the parameters are enhanced then the whole system performance can be better.

1.6 Load Balancing in Cloud Computing Environment

Load balancing in cloud computing gives an efficient solution to various problems residing in cloud computing environment set-up and practice. Load balancing must receive into account two major tasks, one is the resource delivery or resource allocation and other is job scheduling in distributed environment. Efficient delivery of resources and scheduling of resources as well as tasks will guarantee [12]:

- Resources are simply offered on demand.
- Resources are resourcefully utilized under situation of high/low load.
- Power is saved in case of fewer loads (i.e. when usage of cloud resources is lower certain threshold).
- Cost of using resources is cheap.

A class diagram of Cloud using following four entities Datacenters, Virtual Machines, Hosts and Application as well as System Software. Datacenters entity has the dependability of producing Infrastructure level Services to the Cloud Clients. They operate as a home to some Host Entities or some instances hosts' entities combined to form a particular Datacenter entity. Hosts in Cloud are Physical Servers that have pre-configured processing ability. Host is accountable for providing Software level service to the Cloud Clients. Hosts have their personal storage and memory. Processing ability of hosts is expressed in MIPS (million instructions per second). They operate as a home to Virtual Machines or some instances of Virtual machine entity combined to form a Host entity. Virtual Machine allows improvement as well as employment of custom application service models. They are mapped to a host that matches their serious characteristics like storage, memory, processing software and accessibility requirements. Thus, alike instances of Virtual Machine are mapped to similar instance of a Host based upon its accessibility. Application and System software are implemented on Virtual Machine on-demand.

Class diagram of Cloud architecture describes relationship between the four essential entities is as follows in

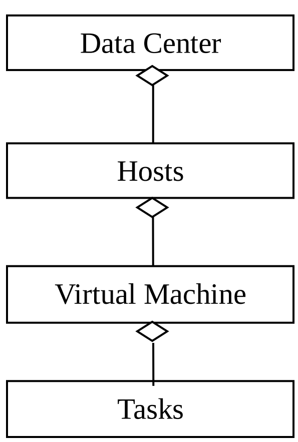


Figure 5 Class Diagram of Cloud

1.6.1 Resource Allocation

Resource delivering is the task of mapping of the resources to different entities of cloud on require basis. Resources must be owed in such a way that no node in the cloud is overloaded and all the accessible resources in the cloud do not suffer any kind of consumption (consumption of bandwidth or memory or processing core etc.). Mapping of resources to cloud entities is completed at two levels:

1.6.1.1 VM Mapping onto the Host

Virtual machines exist in the host (physical servers). More than single instance of Virtual Machine can be mapped onto a particular host subject to its accessibility and ability. Host is dependable for assigning processing cores to Virtual Machine. Delivery policy defines the basis of allocating processing cores to Virtual Machine on order. Allocation policy or algorithm must guarantee that critical characteristics of Host and Virtual Machine do not different.

1.6.1.2 Application or Task Mapping onto VM

Applications or tasks are actually executed on Virtual Machine. All application requests certain amount of processing control for their completion. Virtual Machine must give required processing control to the tasks mapped onto it. All tasks must be mapped onto suitable Virtual Machine based upon its arrangement and accessibility.

Table 1: Comparison between Resource Allocation and Task Scheduling

Task	Sub- Category	Issues Resolved	Provider Oriented	Customer Oriented
Resource Allocation	At host level At VM level	Efficient Utilization Minimize Make span Ensure	Yes	Yes
Task Scheduling	Space-Sharing Time-Sharing	Minimize overall response time	No	Yes

1.6.2 Task Scheduling

Task scheduling is finished after the resources are owed to every cloud entities. Scheduling describe the manner in which dissimilar entities are delivered. Resource provisioning describe which resource will be existing to meet customer requirements whereas task scheduling describe the manner in which the owed resource is existing to the end customer (i.e. whether the resource is completely existing until task completion or is existing on sharing basis). Task development proposes multi-programming abilities in cloud environment.

Task scheduling can be done in two modes:

(a) Space shared

(b) Time shared

Both hosts and Virtual Machine can be delivered to users either in time shared mode or space shared mode. In space sharing mode resources are owed until task does not undergo total execution (i.e. resources are not prevent); whereas in time sharing mode resources are constantly prevent till task undergoes finishing point.

Table 1 gives the comparison between resource allocation and task scheduling and specifies the problem resolved by each procedure of load balancing.

Based on resource delivering and scheduling, four cases can be observed under different performance measure so as to get resourceful load balancing scheme.

Case 1: Hosts and Virtual Machines, both are delivered in space sharing way.

Case 2: Hosts and Virtual Machines, both are delivered to Virtual Machines and tasks correspondingly in time sharing way.

Case 3: Hosts are delivered to Virtual Machines in space sharing way and Virtual Machines are delivered to tasks in time sharing way.

Case 4: Hosts are delivered to Virtual Machines in time sharing way and Virtual Machines are delivered to tasks in space sharing way.

1.7 Load Balancing on the basis of Cloud Environment

Cloud computing can contain either static or dynamic environment depend upon how developer configures the cloud required by the cloud contributor.

1.7.1 Static Environment

In static environment the cloud contributor establish homogeneous resources. Also the resources in the cloud are not elastic when environment is made fixed. The cloud needs prior knowledge of nodes ability, memory, processing power, information of user requirements, performance. These customer requirements are not subjected to any modification at run-time. Algorithms proposed to get load balancing in fixed environment cannot acclimatize to the run time changes in load. Although fixed environment is simple to simulate but is not well appropriate for assorted cloud environment.

Round Robin algorithm provides load balancing in fixed environment. In this the resources are delivered to the task on first-cum-first-serve (FCFS- i.e. the job that entered first will be first owed the resource) basis and scheduled in time sharing way. The resource which is smallest amount loaded (the node with smallest amount number of connections) is allocated to the task.

1.7.2 Dynamic Environment

In dynamic environment the cloud contributor establish heterogeneous resources. The resources are elastic in dynamic environment. The cloud cannot rely on the previous knowledge whereas it takes into account run-time information. The requirements of the customers are granted elasticity (i.e. they may modify at run-time). Algorithm proposed to manage load balancing in dynamic environment can simply adapt to run time alteration in load. Dynamic environment is not easy to be simulated but is extremely adjustable with cloud computing environment.

Based on WLC (weighted least connection) algorithm, Ren describes a load balancing method in dynamic environment called ESWLC. It assigns the resource with smallest amount weight to a task and takes into account node abilities. Based on the weight and abilities of the node, task is allocated to a node. LBMM (Load Balancing Min-Min) algorithm proposed utilizes three level frameworks for resource distribution in dynamic environment. It utilizes OLB (opportunistic load balancing) algorithm as its origin. Since cloud is extremely scalable and independent, dynamic scheduling is improved alternative over static scheduling.

1.7.3 Load Balancing based on Spatial Distribution of Nodes

Nodes in the cloud are extremely distributed. Hence the node that creates the delivery decision also governs the type of algorithm to be used. There are three types of algorithms that identify which node is accountable for load balancing in cloud computing situation.

1.7.4 Centralized Load Balancing

In centralized load balancing method every allocation and scheduling choice are completed by a particular node. This node is accountable for storing information of whole cloud network and can apply fixed or dynamic advancement for load balancing. This procedure reduces the time mandatory to analyze dissimilar cloud resources but creates a huge overhead on the centralized node. Also the system is no longer fault tolerant in this situation as failure intensity of the overloaded centralized node is far above the ground and improvement might not be simple in case of node failure.

1.7.5 Distributed Load Balancing

In distributed load balancing method, no particular node is accountable for making resource delivery or task scheduling result. There is no particular domain accountable for observing the cloud network as a substitute multiple domains observes the network to create correct load balancing decision. All nodes in the network continue with local information to ensure efficient distribution of tasks in fixed environment and redeployment in dynamic environment.

In distributed system, failure intensity of a node is not abandoned. Hence, the system is error free and balanced as well as no particular node is overloaded to create load balancing result.

Difference between fixed and dynamic load balancing algorithms is specified in Table 2. It also evaluates them according to spatial distribution of nodes.

A nature inspired result is obtainable in paper called Honeybee Foraging for load balancing in distributed environment. In Honeybee foraging describes the movement of ant in search of food which forms the basis of distributed load balancing in cloud environment.

1.7.6 Hierarchical Load Balancing

Hierarchical load balancing engage different levels of the load balancing in cloud. Such load balancing method mostly work as master slave mode. These can be represented using tree data structure wherein each node in the tree is evenhanded under the management of its parent node. Master or manager can utilize light weight manager process to get information of slave nodes or child nodes. According to the information collected by the parent node delivering or developing decision is made.

Three-phase hierarchical scheduling projected in article has multiple stages of scheduling. Demand monitor acts as a top of the network and is responsible for examining service manager which in turn examine service nodes. Initial phase uses BTO (Best Task Order) scheduling, next phase uses EOLB (Enhanced Opportunistic Load Balancing) scheduling and last phase uses EMM (Enhanced Min-Min) scheduling.

1.7.7 Load Balancing Based on Task Dependencies

Reliant tasks are those whose implementation is reliant on one or more sub-tasks. They can be implementing only after end of the sub-tasks on which it is reliant. So, scheduling of such task earlier to implementation of sub-tasks is ineffective. Task dependency is modeled using workflow depended algorithms.

Workflow basically utilizes DAG as information to characterize task dependency. Different workflow depended result consider different constraints. Algorithm are considered keeping in mind whether particular or various workflows are to be modeled or particular or various QoS constraints are to be maintained in the organization. Different workflows with or without totally different structure are called as various workflows. Workflows are classified as Data Incentive workflows (size and quantity of data is huge) and Transaction Incentive (various instances of single workflow that have similar structure).

Table 2: Comparison Table of Load Balancing Algorithms in Cloud Computing Environment

Algorithm	Static Environment	Dynamic Environment	Centralize Balancing	Distributed Balancing	Hierarchical Balancing
Round-robin	Yes	No	Yes	No	No

CLBDM[22]	Yes	No	Yes	No	No
Ant Colony[20]	No	Yes	No	Yes	No
Map Reduce[9]	Yes	No	No	Yes	Yes
Particle Swarm Optimization [21]	No	Yes	No	Yes	No
Max Min[22]	Yes	No	Yes	No	No
Min Min[22]	Yes	No	Yes	No	No
Biased Random Sampling	No	Yes	No	Yes	No
Active Clustering[18]	No	Yes	No	Yes	No
LBMM	No	Yes	No	No	Yes
OLB[23]	Yes	No	Yes	No	No
WLC	No	Yes	Yes	No	No
ESWLC	No	Yes	Yes	No	No
Genetic Algorithm[24]	No	Yes	Yes	No	No

Table 3: Comparison of different Types of Load Balancing Scenarios in Cloud Computing Environment

Type of Algorithm	**Knowledge base**	**Issue to be addressed**	**Usage**	**Drawbacks**
Static	Prior knowledge base is acquired about each node statistics and user requirements	Response time Resource utilization Scalability Power consumption and Energy Utilization Makespan Throughput/Performance	Used in homogeneous environment.	Not Flexible Not scalable Is not compatible with changing user requirements as well as load

Dynamic	Run time statistics of each node are monitored To adapt to changing load requirements.	Location of processor to which load is transferred by an overloaded processor. Transfer of task to a remote machine. Information Gathering. Load estimation. Limiting the number of migrations.	Used in heterogeneous Environment.	Complex Time Consuming
Centralized	Single node or server is responsible for maintaining the statistics of entire network And updating it from time to time.	Threshold policies Throughput Failure Intensity Communication between central server and Processors in network. Associated Overhead	Useful in small networks with low Load.	Not fault tolerant Overloaded central decision making node
Distributed	All the processors in the network responsible for load balancing store their own local database (e.g. MIB) to make efficient Balancing decisions.	Selection of processor that take part in load Balancing. Migration time Inter processor communication Information exchange criteria	Useful in large and Heterogeneous environment.	Algorithm complexity Communication overhead
Hierarchical	Nodes at different levels of hierarchy communicate with the nodes below them to get information about the network performance.	Threshold policies Information exchange criteria Selection of nodes at different levels of network Failure intensity Performance	Useful in medium or large size network with heterogeneous	Less fault tolerant Complex

Workflow Dependent	DAG is used to model dependencies of task And can be used to make scheduling decision.	Type of workflow Single workflow Multiple workflow Transaction incentive workflows Data incentive workflows Fault tolerance Execution time Makespan	Used in modeling of task dependencies in any kind of environment (either homogeneous or Heterogeneou	Difficult to model Maintenance of knowledge base is Complex. Higher Complexity

CHAPTER 2

LITERATURE SURVEY

Yubin Yang et al, in "Cloud analysis by modeling the integration of heterogeneous satellite data and imaging" 2006 [13], the authors explain currently in this paper that a computer-aided cloud study process by efficiently modeling the combination of assorted satellite-observed statistics and remote sensing images. Initial, usual cloud recognition and tracking processes are projected to recognize the geo-referenced cloud items in satellite remote sensing pictures. After that, a statistics combination modeling process is designed to gather important assets of those identified clouds by combination the assorted satellite-observed statistics and picturing into combined cloud storage. At last, depend upon the combined global statistics schema, a two-stage data mining method employing the result tree algorithm is executed to examine and estimate the meteorological activities of each the cloud items. Experimental solutions have exposed that the projected data combination model can efficiently extract and produce all the useful information from assorted statistics sources to produce an integrated view of data, on the origin of which the included tendency of clouds can be examined correctly.

Yi Zhao et al, in "Adaptive Distributed Load Balancing Algorithm Based on Live Migration of Virtual Machines in Cloud" 2009 [14], the writer explain EUCALYPTUS, an open basis cloud computing structure, is still lack of weight balancing. This paper describe that it give a kind of execution by included live immigration of virtual machines. They plan and execute an easy model which reduces the immigration instance of VM by collective storage and complete the zero-downtime rearrangement of VM by altering them as Red Hat group services. In immigration procedure, they also remain the inclusion association among VLAN and virtual machines. They suggest a dispersed weight balancing algorithm CONTRAST_AND_BALANCE depend upon sample statistics to reach a balance result. The experimental outcome illustrate that it converges rapidly.

Zehua Zhang et al, in "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation" 2010 [15], the writer explain even however cloud computing is typically documented as a expertise which will has a substantial crash on IT in the outlook. Although, Cloud work out is still in its early days, several crucial troubles need to be determined for the awareness of the well adjacent which hypothetically portray by cloud work out. Weight balancing is one of these troubles; it plays an extremely vital role in considerate of Open Cloud work out Alliance. They projected a load balancing process depends upon composite network idea and ACO in open cloud work out alliance in this study, it exceed many characteristic of the associated ACO which projected to appreciate weight balancing in distributed scheme, In addition, this method obtain the attribute of Complex system into deliberation. At the conclusion the performance of this method is qualitatively estimated, and an example is urbanized to assist the quantitative scrutiny, simulation outcome apparent the scrutiny.

Feller, E. et al, in "Energy-Aware Ant Colony Based Workload Placement in Clouds" 2011 [16], the authors explain with rising nos. of power hungry statistics centers power maintenance has now become a main propose limitation. One conventional process to conserve power in virtualized statistics centers is to execute workload (such as VM) consolidation. Thus, workloads are filled on the smallest amount of corporeal machines and over delivered resources are changeovers into a poorer power situation. Though, a large amount of the workload combination processes applied until currently are partial to a particular resource (such as, CPU) and rely on easy greedy algorithms i.e. First-Fit declining (FFD), which execute resource dispositive workload assignment. Furthermore, they are very federal and called to be solid to allocate. In this job, they replica the workload consolidation trouble as an occasion of the 3-dimensional bin-packing trouble and plan a new, life stimulated workload combination algorithm depend upon the ACO. They appraise the ACO-depended process by contrasting it with one regularly practical greedy algorithm (such as FFD). Our simulation solutions exhibit those ACO outcomes the appraised greedy algorithm as it attains better power gains through improved server consumption and needs smaller number machines. In addition, it calculates results which are almost best. Lastly, the independent nature of the process permits it to be executed in a completely distributed situation.

Kun Li et al, in "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization" 2011 [17], the writer explain the cloud work out is the formation of parallel computing ,disseminated computing and lattice computing, or identified as the marketable execution of these computer discipline ideas. A primary problem in this atmosphere is linked to job scheduling. Cloud job development is an NP rigid optimization issue, and lots of Meta heuristic algorithms have been utilized to resolve it. A good job scheduler should become accustomed its development policy to the varying atmosphere and the kind of jobs. This paper recommends a cloud job development policy based on WBACO (Weight Balancing Ant Colony Optimization algorithm). The major contribution of their work is to equilibrium the whole weight of structure while annoying to reducing the create span of a specified jobs set. The novel development policy was replicated utilizing the CloudSim implement kit enclose. Testing solutions illustrate the projected LBACO algorithm is enhanced than First arrive First supply and the central Ant Colony Optimization.

Zhiliang Zhu et al, in "SLA Based Dynamic Virtualized Resources Provisioning for Shared Cloud Data Centers" 2011 [18], the writer explain cloud work out spotlights on delivery of secure, reliable, sustainable, scalable and dynamic resources stipulation for swarm virtualized requested services in collective cloud statistics hub. For an appropriate delivering method, they developed a novel cloud statistics hub plan depend upon virtualization processes for multi level requests, so as to diminish delivering overheads. Meanwhile, they suggested a new dynamic delivering practice and engaged a plastic cross rowing model to choose the virtualized possessions to deliver to every level of the virtualized request services. They further urbanized meta-heuristic results that are according to abnormal performance necessities of customers from diverse stages. Replication experiment outcomes illustrate that these projected advances can offer suitable way to prudently prerequisite cloud statistics hub assets, mainly for civilizing the generally routine while increasing the global benefit of cloud communications supplier and efficiently decrease the source custom further cost.

Babukarthik, R.G. et al, in "Energy-aware scheduling using Hybrid Algorithm for cloud computing" 2012 [19], the writer explain Cloud computing are becoming an essential platform for scientific applications. With virtualization in cloud computing it is possible to provide a runtime provision of virtualized resources as a service, without any extra waiting time. Scheduling the tasks based on the SLA (service level agreement) reduces the energy utilization, leading to a substantial cut in energy cost. Energy utilization is the blue eye of task scheduling. But all the applications are not based on the SLA, so complexity of the problem arises. Therefore the task scheduling need to be focused by considering the energy utilization and it has to be evaluated based on the value of the schedules. This paper puts forward a hybrid algorithm which spotlights on the reduction of energy utilization. This method is based on the voltage scaling aspect to reduce the energy utilization. The outcome proves that by using this method, energy utilization can be minimized while scheduling the tasks.

Feller, E. et al, in "A case for fully decentralized dynamic VM consolidation in clouds" 2012 [20], the writer explain one way to save power in cloud statistics hub is to conversion of inactive servers into a power reduction status in the time of low consumption. Active virtual machine strengthens algorithms are projected to generate inactive times by sporadically rebounding VMs on the smallest no. of corporeal machines. Presented works typically affect VMC on the above of hierarchical, centralized, or ring depended structure topologies which outcome in reduced flexibility and/or bounding effectiveness with rising no. of VMs and PMs. According to paper, they advise a new entirely decentralized active VMC plan depend on a formless peer-to-peer system of PMs. The recommended schema is authenticated by 3 fine recognized VMC algorithms: Sercon, V-MAN, First Robust Declining and a new immigration cost attentive ACO-depended algorithm. Extensive research's done on the Grid'5000. Outcomes illustrate that once incorporated in their totally decentralized VMC plan; conventional VMC algorithms conquer a universal packing usefulness near to a centralized scheme. Additionally, the scheme remains flexible with rising no. of VMs and PMs. Lastly, the immigration-cost attentive ACO-depended algorithm surpasses Sercon and FFD in the no. of unconfined PMs and necessitates fewer migrations than V-MAN and FFD.

Nishant, K. et al, in "Load Balancing of Nodes in Cloud Using Ant Colony Optimization" 2012 [21], the writer explain in this paper, a load sharing algorithm of workloads between knots of a cloud by utilizing of Ant Colony System. This is a customized process of ant colony system that has been functional from the standpoint of grid network or cloud schemes with the major goal of weight balancing of knots. This customized algorithm has a rim above the new process in which every ant build their personality outcome group and it is presently on constructed into a total result. Nevertheless, in their process the ants regularly modernize a single outcome group rather than modernizing their own outcome group. Additional, as they identify that an obscure is the grouping of several knots, which can bear several types of request that is utilized by the customers which is depend on pay per utilize. Hence, the scheme, which is acquiring a cost for the customer should run effortlessly and must have algorithms, which can preserve the suitable scheme functioning yet at climax practice hours.

Sotiriadis, S. et al, in "Decentralized meta-brokers for inter-cloud: Modeling brokering coordinators for interoperable resource management" 2012 [22], the writer explain the organization of internal resources in large-scale atmosphere is a critical challenge due to the huge number of customers and service requests. In clouds, an efficient resource executive organizes internal resources by assigning brokers to customers for performing on behalf of their clients. This is to map customer needs to cloud datacenters for service execution and sharing. Though, as cloud computing matures, it is crucial to permit the model of inter-clouds, that is to say, enabling the association and thus, the interoperation between several scatter (and highly likely heterogeneous) clouds. To this extend, they initiate the meta-broker model for inter-cloud settings by modeling its formation in a total decentralized model. This is to synchronize various clouds brokers for establishing a service automation and reactive cross-exchange while offering straightforwardness to users. They reproduce an inter-cloud for calculating the performance of an average execution time for various customers that submit concurrent a massive amount of services. The outcome show efficient performance levels when working under meta-brokering solution.

Xiaotang Wen et al, in "Study on Resources Scheduling Based on ACO Algorithm and PSO Algorithm in Cloud Computing" 2012 [23], the authors explain it get better the algorithm because of the inadequacy that the ACO algorithm is simple to fall into local optimal method in the cloud computing resource scheduling. The improved algorithm makes particle optimization inoculated into ant colony algorithm, which first finds out various groups of solutions using ACO algorithm according to the reorganized pheromone, and then gets more efficient solutions using PSO algorithm to do crossover operation and mutation operation so as to keep away from the algorithm prematurely fall into the local optimal solution.

Acharya, S. et al, in "A taxonomy of Live Virtual Machine (VM) Migration mechanisms in cloud computing environment" 2013 [24], the authors explain the resources in a cloud environment are proficiently governed by employing virtualization technology. Virtualization permits multiple operating system instances to run parallel on a single computer. The administration of virtual machines in the datacenter of cloud computing environment is a tough task which requires live migration techniques. Live migration moves running virtual machines in-between different physical machines without yielding control over its network connections and resources for maintaining the load balancing. The main aim of VM migration is to reduce the number of physical machines helping the given task/job with least energy utilization by switching off the idle nodes. In this paper, they explain a detailed review of different live migration techniques. The authors also define the classification of live migration mechanisms based on their utilization in specific cloud environments by bringing out their outstanding features in the respective domains of service.

Ahmad, N. et al, in "Survey on secure live virtual machine (VM) migration in Cloud" 2013 [25], the writer explain the core of cloud computing incorporates virtualization of hardware resources such as physical storage, network and memory provided through virtual machines. The live immigration of these VMs is introduced to get multiple benefits which mainly include high accessibility, hardware maintenance, fault takeover and load balancing. Besides various facilities of the VM immigration, it is tending to severe security hazard during immigration process due to which the industry is uncertain to accept it. The study done so far is on the performance of migration process; but the security aspects in immigration are not fully explored. They have carried out a wide survey to examine the vulnerabilities, threats and possible attacks on the live VM immigration. Moreover, they have known security requirements for secure VM immigration and presented a thorough analysis of existing outcomes on the basis of these security requirements. Lastly, limitations in the current outcomes are represented.

Ajit, M. et al, in "VM level load balancing in cloud environment" 2013 [26], the authors explain as cloud computing is spreading worldwide and number of customers demanding extra cloud services and better outcome are rising rapidly, cloud load balancing become a very interesting and important study area. Usually, cloud is based on dominant datacenters that manage large number of customers, so it must be characterized with load balancer to get consistency which depends on the mode it handles the load. Cloud load balancing helps to get better cloud performance. Several algorithms were suggested for transmission the customers' requests to Cloud resources to give services powerfully. This paper presents the study of three existing algorithms in cloud analyst tool to resolve the crisis of cloud load balancing as a training phase for novel load balancing method. A WSLB (weighted Signature based load balancing) algorithm is suggested to reducing customer's response time. Moreover, this paper also offers the predictable outcome with the execution of the projected algorithm.

Al-Rayis, E. et al, in "Performance Analysis of Load Balancing Architectures in Cloud Computing" 2013 [27], the authors explain the cloud computing is a quickly emerging spread system pattern that suggests a large amount of IT resources as usefulness services at a cheap cost and elastic schemes. The importance of such elasticity is a capable load balancer that suggests better organization and utilization of virtualized essential cloud infrastructures. Though, most of the obtainable load balancers in cloud computing are supported on both centralized and entirely distributed design while the plan of harnessing several load balancers in a hierarchical organization to get better the sever load and task response time is still under deliberate. So, this paper, goals at bridging this space by providing a relative study among the three load balancing design in cloud computing: decentralized, central and hierarchical load balancers. The experimental solutions advise that the hierarchical design for load balancers greatest suits the public cloud atmosphere and call for more research to test whether these solutions can be widespread for other types of clouds.

Chaukwale, R. et al, in "A modified Ant Colony optimization algorithm with load balancing for job shop scheduling" 2013 [28], the authors explain the trouble of powerfully scheduling tasks on many machines is an major consideration when utilizing Job Shop scheduling creation system (JSP). JSP is identified to be a NP-hard crisis and hence techniques that focus on generating an exact result can prove inadequate in finding an optimal result to JSP. So, in such cases, heuristic techniques can be working to find an excellent result within logical time. In this paper, they learn the conservative ACO algorithm and intend a Load Balancing by using ACO algorithm for JSP. They also represent the experimental results, and converse them with reference to the conservative ACO. It is experimental that the projected algorithm gives improved results when compared to conservative ACO.

Raju, R. et al, in "Minimizing the makespan using Hybrid algorithm for cloud computing" 2013 [29], the authors explain cloud computing methods offer major profits to the IT business in terms of flexibility and rapid delivering, pay-as-you-go-model, cheap resources cost, use of unlimited resources, elasticity. Task scheduling is a combinatorial optimization trouble in the areas of computer science where the best tasks are assigned to essential resource at a certain instant of time. In this paper they projected mix algorithm which merge the benefit of ACO and Cuckoo search. The makespan or end time can be cheap with the help of mix algorithm, since the tasks have been implementing within the particular instant of time by allocation of essential resources using the mix algorithm. The acquire solutions shows that mix algorithm execute well than contrast with the ACO algorithm in words of performance of the makespan and algorithm.

Tantar, A.A. et al, in "Computational intelligence for cloud management current trends and opportunities" 2013 [30], the authors explain the expansion of large range data center and cloud computing optimization models lead to a broad range of complex problems like ranging, process cost and energy effectiveness. Various approaches were projected to the end including standard machine learning, resource allocation heuristics or stochastic optimization. No agreement exists but a fashion towards utilizing many-objective stochastic replica became clear over the precedent years. This work analysis in brief a few of the more new research on cloud computing modeling and optimization, and top at notions on constancy, union, description or outcomes that could provide to evaluate, respectively build correct cloud computing models. A extremely brief conversation of simulation structure that contain support for power aware components is also specified.

Tawfeek, M.A. et al, in "Cloud task scheduling based on ant colony optimization" 2013 [31], the authors explain cloud computing is the expansion of distributed computing, similar computing and net computing, or described as the commercial execution of these computer science idea. One of the essential problems in this situation is related to job scheduling. Cloud job scheduling is an NP-hard optimization issue, and several meta-heuristic algorithms have been projected to solve it. A good job scheduler should get used to its scheduling approach to the altering situation and the types of jobs. This paper a cloud job scheduling rule based on ant colony optimization algorithm contrasted with various scheduling algorithms FCFS and round-robin, has been represented. The major aim of these algorithms is reducing the makespan of a specified jobs set. Ant colony optimization is random explore approach that will be utilized for allocating the inward works to the virtual machines. Algorithms have been pretended using Cloudsim toolkit package. Experimental outcome illustrated that the ant colony optimization execute FCFS and round-robin algorithms.

Wei Yuan et al, in "Towards Efficient Deployment of Cloud Applications through Dynamic Reverse Proxy Optimization" 2013 [32], the authors explain with the enhancement of users and the deployment needs the issue of dynamic employment in PaaS becomes important. Different approaches of application employment have been extremely discussed, but the problems like fast reply to a large number of synchronized employment requests are rarely alert. In this work, they expand Nginx as a dynamic reverse proxy to maintain dynamically remote design for better flexibility of cloud applications in PaaS, and then additional optimize it for get better performance under a huge number of synchronized configuration needs. Three optimization approaches are projected: Batch File Processing (BFP), Batch Request Committing (BRC) and In Memory Configuration (IMC). They provide a detailed execution of each technique, and a qualitative analysis on three optimization methods has been made. Lastly, a sequence of experiments is represented to validate the optimization outcome. The experiment outcomes show that the greatest throughput per second has enlarged significantly, and the standard response time of each demand has reduced dramatically.

Weifeng Sun et al, in "PACO: A Period ACO Based Scheduling Algorithm in Cloud Computing" 2013 [33], the authors explain tasks scheduling trouble in cloud computing is NP-hard, and it is hard to attain an optimal result, so they can use intellectual optimization algorithms to estimate the optimal result, for example ant colony optimization algorithm. For organizing to solve the task scheduling trouble in cloud computing, a phase ACO based scheduling algorithm (PACO) has been projected in this paper. PACO utilizes ant colony optimization technique in cloud computing, with the first projected scheduling phase strategy and the development of pheromone strength update strategy. The experiments outcomes show that, PACO has an excellent performance equally in makespan and load balance of the entire cloud cluster.

Xianglin Wei et al, in "Bio-inspired Application Scheduling Algorithm for Mobile Cloud Computing" 2013 [34], the authors explain Mobile Cloud Computing (MCC) allow the mobile devices to relieve of their applications to the cloud, and therefore greatly improved the types of applications on mobile devices and improves the value of service of the applications. Below various conditions, researchers have put onward several MCC architectures. Though, how to decrease the response latency while competently utilizing the idle service abilities of the mobile devices is still a dispute. This paper initially gives a description of MCC. Secondly, by expanding the cloudlet architecture, a Fusion Local Mobile Cloud Model is represented. Thirdly, from two various angles, they originate the application scheduling troubles in FLMCM and bring onward the scheduling algorithms. Lastly, the effectiveness of the proposed algorithms is authenticated by simulation experiments.

Liu, Xiaodong, Lin Liu, and Hongji Yang, in "TARGO: Transition & Reallocation Based Green Optimization for Cloud Virtual Machine" 2013 [35], the authors explain much research has been performed focusing on humanizing resource utilization effectiveness in data centers in the background of Green Cloud Computing (GCC). While effectiveness enables improved resource condition and utilization for different computational resources, diverse approaches are projected based on VM optimizations utilizing either server or workload consolidation methods. However, these results can only be applied within the Cloud. In reality, Infrastructure-as-a-Service (IaaS) customers can hardly proactively get improved VM resource utilization effectiveness, as they usually have no control above any hyper blind or hardware in some Clouds. The problem gets more serious when workloads on VMs change dramatically from point in time. This paper represents a new approach that is conversion & reallocation based Green Optimization for such customers. During fully programmed and intellectual VM optimization perform according to customizable optimization policy.

Abolfazli, S. et al, in "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, & Open Challenges" 2014 [36], the authors explain recently, Cloud-based mobile expansion approaches have increased remarkable ground from university and business. CMA is the state-of-the-art mobile expansion model that utilizes resource-rich clouds to boost, improve, and optimize computing abilities of mobile devices wanting at implementation of resource-intensive mobile applications. Improved mobile devices imagine to perform wide-ranging computations and to accumulate big data outside their intrinsic abilities with slightest footprint and weakness. Researchers use varied cloud-based computing resources (e.g., remote clouds and close by mobile nodes) to gather different computing requirements of mobile customers. Though, utilizing cloud-based computing resources is not a simple panacea. Comprehending serious factors (e.g., present state of mobile customer and distant resources) that impact on expansion process and best choice of cloud-based resource kind are some confront that delay CMA adaptability. This paper expansively surveys the mobile expansion domain and represents classification of CMA approaches. The purpose of this study is to underline the effects of distant resources on the quality and consistency of expansion processes and converse the challenges and chances of employing various cloud-based resources in enhancing mobile devices. They represent expansion definition, inspiration, and classification of expansion types, including conventional and cloud-based. They seriously evaluate the state-of-the-art CMA approaches and categorize them into four set of distant permanent, nearby fixed, proximate mobile, and hybrid to represent a classification. Essential decision making and performance restriction factors that manipulate on the acceptance of CMA approaches are initiated and an excellent decision making flowchart for outlook CMA approaches are represented. Impacts of CMA approaches on mobile computing is conversed and open challenges are represented as the outlook research.

Ahmed, A. et al, in "Cloud computing simulators: A detailed survey and future direction" 2014 [37], the authors explain Cloud computing is one of the rising technologies with its simplicity of access and varied applicability, letting clients attracted to it and thus affectation many challenging problems that need to conquer in this area. As the development of cloud computing: Load balancing, program offloading, power constrains, cost modeling and security problems are the trendy research topic in this area. Organizing actual cloud for testing or for profitable use is extremely costly. Cloud simulator assists to model different kinds of cloud application by producing Virtual Machine, Data Centre and several Utilities which can be further to configure it, thus building it very easy to evaluate. Till now, several cloud simulators have been projected and also existing to utilize. These simulators are constructing for specific reason, and have unreliable features in each of them. This paper they existing a comprehensive learning of main cloud simulators by highlighting their significant features and evaluate their cons and pros. They made a contrast among the simulators by allowing their significant attributes and lastly concluded with their outlook direction.

Ashraf, A. et al, in "Using Ant Colony System to Consolidate Multiple Web Applications in a Cloud Environment" 2014 [38], the authors explain Infrastructure as a Service (IaaS) clouds give virtual machines (VMs) below a pay-per-use company model, which can be used to produce a dynamically scalable group of servers to organize one or more web applications. In compare to the conventional devoted hosting of web applications where every VM is utilized exclusively for a particular web application, the collective hosting of web applications allocate improved VM use by sharing VM resources between synchronized web applications. Though, in a collective hosting atmosphere, dynamic scaling alone does not reduce over-delivering of VMs. This paper, represent a new approach to merge numerous web applications in a cloud-based collective hosting atmosphere. The projected approach utilizes Ant Colony Optimization (ACO) to construct a web application immigration plan, which is then utilized to reduce over-delivering of VMs by merging web applications on underutilized Virtual Machine. The projected approach is established in discrete-event simulations and is estimated in a series of research involving artificial as well as practical load patterns.

Caton, S. et al, in "A Social Compute Cloud: Allocating and Sharing Infrastructure Resources via Social Networks" 2014 [39], the authors explain social network stage have quickly changed the method that people communicate and cooperate. They have allowed the organization of and participation in, digital society as well as the illustration, documentation and investigation of social relationships. They think that as `apps' become extra sophisticated; it will turn into easier for customer to divide their own services, resources and information via shared networks. To verify this, they present a shared compute cloud where the delivering of cloud communications occurs through relationships. In a shared compute cloud, resource owners propose virtualized containers on their private computer(s) or smart appliance(s) to their shared network. However, as customer may have multifarious preference structures regarding with whom they do or do not hope to distribute their resources, they explore, via simulation, how resources can be efficiently allocated within a shared community contributing resources on a best attempt basis. In the assessment of shared resource allocation, they believe benefit, allocation equality, and algorithmic runtime. The main conclusion of this work demonstrate how shared networks can be leveraged in the creation of cloud computing infrastructures & how resources can be owed in the existence of customer sharing preferences.

Chun-Wei Tsai et al, in "A Hyper-Heuristic Scheduling Algorithm for Cloud" 2014 [40], the authors explain rule-based scheduling algorithms have been broadly used on several cloud computing systems as they are effortless and easy to execute. However, there is abundance of room to get better the performance of these algorithms, particularly by utilizing heuristic scheduling. In this paper presents a new heuristic scheduling algorithm, named hyper-heuristic scheduling algorithm (HHSA), to locate better scheduling results for cloud computing organization. The diversity detection and upgrading detection operators are working by the projected algorithm to dynamically establish which low-level heuristic is to be used in result better applicant result. To assess the efficiency of the projected process, this revise compares the projected method with numerous state-of-the-art scheduling algorithms; by contain all of them executed on Hadoop (a real system) and CloudSim (a simulator). The outcome show that HHSA can considerably decrease the make span of job scheduling evaluated with the other scheduling algorithms evaluated, in cooperation CloudSim and Hadoop.

Chun-Wei Tsai et al, in "Meta-heuristic Scheduling for Cloud: A Survey" 2014 [41], the authors explain Cloud computing has develop into an increasingly significant research topic given the tough evolution and migration of several network services to such computational atmosphere. The trouble that arises is related with good organization and utilization of the huge amounts of computing assets. This paper starts with a short retrospect of conventional scheduling, followed by a complete review of meta-heuristic algorithms for resolving the scheduling troubles by placing them in a integrated framework. Equipped with these two expertises, this paper analyzes the new literature about meta-heuristic scheduling results for cloud. In addition to applications via meta-heuristics, a few important issues and questions are represented for the situation of prospect researches on scheduling for cloud.

Hung, P.P. et al, in "Task scheduling for optimizing recovery time in cloud computing" 2014 [42], the authors explain to attain high performance, synchronize tasks to present reliable, thousands of servers in cloud datacenters and extremely available cloud computing services, mainly, in terms of multitasking. Efficient mechanisms are now mandatory to prepare for a crash of such computing nodes. A quantity of studies has been completed to address this trouble, but it cannot always promise an acceptable performance. This paper, represent a scheduling algorithm, based on bandwidth and cost, which formulate efficient recovery probable on heterogeneous computing atmosphere. Our algorithm not only thinks about the network bandwidth, but also obtains into account the economic cost as well. They validate their projected work through general simulations and contrast their work with the accessible studies. The outcome can improve the probable profit of their approach.

Rahman, M. et al, in "Load Balancer as a Service in Cloud Computing" 2014 [43], the authors explain the volatile growth of cloud computing in current years has lead to a enormous increase in both the quantity of traffic and the quantity of service desires to cloud servers. This expansion trend of load poses grim dispute to the cloud load balancer in proficient balancing of the load, by now a daunting work. The cloud load balancing is an extremely researched area where many solutions to balance load have been projected. Regrettably, no research papers give a complete review that focusing on Load Balancer as a Service (LBaaS) model. This paper, they initially know the concepts of load balancing, its significance and desired uniqueness in cloud. Then they give whole review on the accessible load balancing strategies, their power, weakness and a relative study. Finally, they offered load balancer as a service model accepted by the major market company, and their inspection, future requirements and challenges.

Sanaei, Z. et al, in "Heterogeneity in Mobile Cloud Computing: Taxonomy & Open Challenges" 2014 [44], the authors explain the unabated flood of research activities to augment different mobile devices by leveraging various cloud resources has produced a new research field called Mobile Cloud Computing (MCC). In the center of such a non-uniform situation, facilitating portability, interoperability and combination among various platforms is nontrivial. Building such facilitators in MCC wants investigations to recognize heterogeneity and its dispute over the roots. Even though there are several research studies in cloud and mobile computing, convergence of these two region grants further university efforts towards flourishing MCC. In this paper, they describe MCC, give details about major challenges, converse heterogeneity in convergent computing (i.e. mobile & cloud computing) and partition it into two dimensions, namely vertical & horizontal and networking (wireless & wired networks). Heterogeneity roots are examined and taxonomies as platform, hardware, feature, network and API. Multidimensional heterogeneity in MCC solutions in application and code fragmentation problems that impede progress of cross-platform mobile applications which is scientifically illustrate. The collision of heterogeneity in MCC are examine, related opportunities and dispute are identified, and major heterogeneity handling approaches such as service oriented architecture (SOA), potential and middleware are converse. They summarize problems that help in recognizing new study directions in MCC.

Sarbazi-Azad, H. et al, in "Market-Oriented Cloud Computing & the Cloudbus Toolkit" 2014 [45], the authors explain this chapter initiate the fundamental thought of market-oriented Cloud computing systems and represent a orientation model. This model, mutually with the state-of-the-art expertise presented in this chapter, add significantly towards the normal adoption of Cloud computing expertise. Though, any expertise brings with it new dispute and breakthroughs. This chapter center on the main challenges faced by the business when adopting Cloud computing as a normal expertise as part of the distributed computing standard. It represents a utility-oriented Cloud idea that is a generic model for understanding market-oriented

Cloud computing idea. Cloudbus realized this by mounting different tools and platforms that can be utilize independently or mutually as an incorporated solution. The author demonstrates through research that their toolkit can give applications based on time limit, cost efficiency and time of applications, and handle real-world troubles through an incorporated solution.

Wadhwa, B. et al, in "Energy saving approaches for Green Cloud Computing: A review" 2014 [46], the authors explain Cloud Computing is a promising expertise and is being used by more and more IT corporation due to its cost saving profits and simple to use for users. However, it wants to be environment gracious also. Consequently, Green Cloud Computing is the necessity of the today's world. This paper evaluates the efforts made by different researchers to formulate Cloud Computing more energy proficient, to decrease the carbon footprint rate by different approaches and also converse the concept of virtualization and different approaches which utilize virtual machines scheduling and immigration to show how these can assist to formulate the system more energy proficient. The outline of the main features of the projected work of diverse authors that there evaluation is also presented in it.

Yonggang Wen et al, in "Cloud Mobile Media: Reflections & Outlook" 2014 [47], the authors explain this paper surveys the promising standard of cloud mobile media. They begin with two substitute perception for cloud cell phone media system: a back-to-back outlook and a covered outlook. Outline offered study in this region is prepared a/c to the covered service structure: i) cloud source association and manage in communications-as-a-service, ii) cloud-depended medium examine in policy-as-a-service and iii) new cloud-depended scheme and request in software-as-a-service. They further validate their projected design philosophy for cloud-depended cell phone media utilizing an existing case study: a cloud centric media policy urbanized at Nanyang Technological University. Lastly, this paper ends with a point of view of open study troubles for comprehend the apparition of cloud-depended cell phone media.

CHAPTER 3

PROPOSED WORK

3.1 Problem Definition/Need of Study

Load balancing in cloud computing environment is a big challenge. A spread result is required always. Since it is not always practically sufficient or cost efficient to organize one or more inactive services just as to complete the required requests. Jobs cannot be allocated to proper servers and users individually for proficient load balancing as cloud is a extremely complex structure and works are present during a wide spread region. Load balancing algorithms are divided into static and dynamic algorithms. Fixed algorithms are generally suitable for uniform and stable environments and can generate best results in these environments. Nevertheless, they are generally not flexible and cannot equal the dynamic modification to the attributes throughout the execution time. Dynamic algorithms are extra flexible and take into deliberation various types of attributes in the system both previous to and throughout run-time. Load balancing is the method to get better performance of system during a redistribution of load between processor.

3.1.1 Optimization as a Solution to cloud load Balancing

Biogeography based Optimization (BBO) is a newly projected meta-heuristic approach for resolving hard combinatorial optimization issues [48]. Biogeography defines how species travel from one island to another, how new species occur, and how species become dead. A habitat is any island (region) that is geographically cut off from other islands. Regions that are well suited as habitations for biological species are called to have a high habitat suitability index (HSI). Causes that manipulate habitat suitability index contain rainwater, variety of vegetation, topographic elements, land region, and temperature. The elements that describe habitability are known as suitability index variables (SIV). Suitability index variables can be measured as the self-sufficient elements of the habitat, and habitat suitability index can be considered using these elements.This Attribute of BBO algorithm can be fully utilized to tackle the Load balancing problem in Clouds, BBO algorithm provide truly distributed and parallel processing essential to the cloud balancing problem.

3.2 Research Goals/ Objectives

1) To characterize the Energy Model of a cloud system, as the Energy of a cloud system is described to be the group of traffic loads under which the rows in the system can be steady.
2) To Study and Analyze the Biogeography Based Optimization Algorithm and to estimate the possibility of its applicability in Cloud based Systems.

3) To implement the BBO algorithm in MATLAB and provide an Experiment to prove that it is optimal for load balancing.
4) To estimate the findings and solutions via visualization Tools in MATLAB.

3.3 Research Methodology

We want cloud computing results that can not only reduce operational costs but also minimize the environmental effect. This work is focused on the design and implementation of an automated Load balancing that achieves a good balance. To successfully achieve the research goals we will implement the BBO based algorithm in MATLAB and evaluate its performance in the same.

3.4 Tools and Techniques

For the above supposed research work, we will be requiring to contact the several e-journals like ACM, Science Direct and IEEE. For data collection and analysis of the data, I will be requiring various software and Internet facility. The hardware and software necessities for my thesis work are:

Hardware Requirements:

- Processor: Intel(R) Core(TM) i3 or Higher
- RAM: 1 GB or Higher
- Hard Disk: 10 GB or Higher

Software Requirements:

- Platform: Windows 7 Ultimate
- MATLAB R2015b and above
- Microsoft office(Excel and Word)

CHAPTER 4

METHOLOGY

4.1 Cloud Load Balancing

Cloud Load Balancing tries to find out "the best" creation schedule for the datacenters to deliver the required requirement plus transmission losses by reducing the production cost. There is several exploration of load balance on cloud that has been accepted until date, as best results would provide a significant solution for economical profit. Earlier a lot of derivative-based approaches which is based on Lagrangian multiplier technique have been functional to resolve such type of issues. These techniques require that by increasing the cost of curves are monotonically rising in environment. But in practical, the input output distinctiveness of new generating elements is extremely nonlinear due to ramp-rate limits, dynamic loadings, multi-fuel options, etc. Classical dispatch algorithms have to be approximated to meet their characteristics of the requirements. Because of such estimate the results is one of the sub-optimal and hence a large amount of income loss happens over the time. Extremely nonlinear distinctiveness of these requirements for result methods that have no limitations on to the form of the energy cost curves. These types of problems tried to solve by calculus-based methods, which fails. But these techniques suffer from the bother of dimensionality and recreation time which enlarges rapidly with enlarges of system dimension. Furthermore application of artificial intelligence technology has been focused on the interests for result of load balancing issues. Numerous techniques, for example genetic algorithm (GA); artificial neural networks; simulated annealing (SA); Tabu search; evolutionary programming; particle swarm optimization (PSO); ant colony optimization; differential evolution; etc. have been residential and functional effectively to load balancing troubles.

Though, the ANN undergo from very time-consuming convergence, because of implementation of sigmoid function in the model. Genetic Algorithm and Simulated Annealing have been effectively employed to resolve load balancing troubles on cloud. The Simulated Annealing technique is typically slower than the Genetic Algorithm technique as the Genetic Algorithm has equivalent search abilities. Though, study has recognized a few lacks in application to extremely epistemic goal functions where the constraints being optimized are powerfully associated. These types of systems, the genes in the inhabitants, towards the finish of the evolutionary procedure have related organization and their standard strength is high. Offspring crossover and mutation processes cannot protect the large fitness and consequent decline in efficiency is evident. Also, due to the convergence of GA which is prematurely degrades your presentation skills and minimization in the search capabilities.

There are only some parameters to be familiar in PSO, which build PSO more eye-catching. The major advantages of the PSO algorithm are effortless concept, simple implementation, robustness and computational effectiveness. A nearer assessment on the procedure of the algorithm specify that once surrounded by the optimum area, the algorithm developments gradually because of its incapability to correct the velocity step dimension to persist the investigate at a finer particle. Thus elements sometimes fail to achieve global best point, for multi-modal function.

Differential evolution (DE) involves three basic operations, e.g., mutation, crossover, and selection, in order to reach an optimal solution. DE has been originating to provide a better way and faster result, using its different crossover strategies, which fulfills all the restriction, both for single-modal and multi-modal system. DE technique is unable to plot its complete indefinite variables together in an improved way, but when system complication and size enlarges, In DE every variables are altered mutually in the crossover operation. The each variable is not adjusted separately. Thus in early stage, the results travels very rapidly towards the best point but at presently stage when well adjusting operation is essential, DE fails to provide improved performance.

The artificial immune system (AIS) is another adaptation population based or network-based soft computing method in the area of optimization that has been productively executed in different energy system optimization troubles. AIS in every execution, like affinity calculation, hyper-mutation, cloning and selection are performed many operations. During cloning, operation size of population also increases. Convergence velocity of AIS is much lesser than DE or PSO, because of maximizing in number of operations, and bigger size of population.

BBO algorithm based on the two working mechanisms: migration and mutation. BBO has the assets of distribution information between results, while in other biology-based algorithms like GA and PSO this is not possible. In addition, the BBO algorithm has few unique characteristics which conquer numerous disadvantages of the conservative processes are as follows.

BBO and in PSO proposals are to endure forever even if the progress of the optimization process as their characteristics change. But GA, EP, DE, etc. each generation offers of evolutionary algorithms based on the findings of "dead". Due to the presence of foreign procedure established evolutionary algorithms, whose fitness is mainly numerous offers are good from time to time during the process, lose their quality. BBO has no crossover like operation; Proposals for an operation across process goes fine tuned softly as become. Additional excellence operational process algorithm has made in this regard. It provides a range of BBO over methods mentioned above.

In earlier (PSO), results are more probable to mix together in alike groups, whereas in BBO; results do not have the propensity to mix because of its novel type of mutation operation. These are additional merits of BBO in contrast to PSO.

BBO engages smaller number of computational steps per execution in contrasted to AIS. These types of solutions have faster convergence rate.

Worst results admit a lot of novel characteristics from excellent ones which may advance the value of those results in BBO. These are the unique characteristics of BBO algorithm in contrast to other methods. At the similar time these creates constraint fulfillment in simple way, contrast to that in BFA.

4.2 Biogeography-Based Optimization (BBO)

Biogeography explains how species from one island to another, how to create new species to travel, and how species come to extinction. Each island is a habitat (area) Supplementary islands are geographically remote. That extend well as houses are suitable for biological species elevated habitat suitability index (HSI) is called for. Factors that influence habitat suitability indices are rainfall, vegetation, topographic features, the Earth period, and temperature variations. Shouted suitability variables that describe habitability index variable (SIV) are done. Variabless habitat suitability index can be considered as independent variables, and habitat suitability index for employment of these variables can be calculated. Along with a high habitat suitability index will have a vast number of species, with a low habitat suitability index as those of a small number of species. Along with a high habitat suitability index countless species that live adjacent to the trip, obviously a large number of species that they have the quality of the host. Supplement to the habitat of a small species from a habitat migration is recognized as the emigration process. A little beyond the habitat of species of each supplementary enters a habitat, it is recognized as the migration process. Because they are now almost saturated with species habitat suitability indices along with a high rate of migration is a less species. Therefore, high habitat suitability index low habitat suitability index residence residence additional allocation of their species are stable. The same token, a high emigration rate is elevated dwelling habitat suitability index. Along with a low habitat suitability index because of their sparse population is a species of high migration rate.

Low habitat suitability index for new species to inhabit the habitat of the migrating habitat suitability index may rise because of a habitat suitability is proportional to its biodiversity. Here, Fig. A solitary species abundance in habitat 2 shows a model. Let us consider the graph of an image. 2. The maximum potential for habitat migration rate I, species after habitat is zero. A small number of species habitat, supplementary habitat away from the large number of species, the habitat may be so high that the migration rate is. Increases the number of species, the habitat becomes more packed, and subsequently less prosperously habitat for species migration, and migration rate decreases are able to bear. Habitat of species that can resort to the largest possible number Smax * migration rate becomes zero at that point, because no additional species in the habitat that the species can be counted. Nowadays graph consider emigration. A species habitat, the next there is the habitat that can supplement any species habitat change, so emigration rate is zero.

As the number of species increases, the habitat becomes more packs, additional supplementary potential homes to discover the habitat of the species, and are able to depart emigration rate increases. E max emigration rate, the number of species that occurs after Sniax. If the balance is the number of species, migration and emigration rates are equal at that point. In the image. 2 migration and emigration lines, straight lines, as shown in the graphs, but, in the end, they may be more complex curves. However, we have a simple model of the process of migration and emigration over the details. BBO algorithm, the emigration rate and calculate the rate of migration habitat suitability index variabless whose operations will experience an important task to choose is as important as the joy.

Mathematically the concept of emigration and immigration can be represented by a probabilistic model. Let us consider the probability P_s that the habitat contains exactly S species at . P_s changes from time t to time t to t+ Δt as follows:

$P_s(t+ \Delta t)=P_s(t)(1-\lambda_s\Delta t-\mu_s\Delta t)+P_s-1\lambda_s-1\Delta t+P_s+1\mu_s+1\Delta t$

where λ_s and μ_s is are the immigration and emigration rates when there are S species in the habitat. This equation holds because in order to have S species at time (t+ Δt), one of the following conditions must hold:

1) there were S species at time t, and no immigration or emigration occurred between t and t+ Δt:

2) there were (S — 1) species at time t, and one species immigrated;

3) there were (S + 1) species at time t, and one species emigrated.

If time At is small enough so that the probability of more than one immigration or emigration can be ignored then taking the limit of (16) as $\Delta t \to 0$ gives (1) at the bottom of the page. From the straight-line graph of Fig. 2, the equation for emigration rate μ_k and immigration rate λ_k for k number of species can be written as per the following way:

$$\mu_k = \frac{EK}{n}$$

$$\lambda_k = I\left(1 - \frac{k}{n}\right)$$

When value of E = I, then combining (2) and (3)

$$\lambda_k+\mu_k=E$$

4.3 Biogeography-Based Optimization (BBO) techniques:

This section describes development of biogeography-based optimization technique and the different steps involved therein.

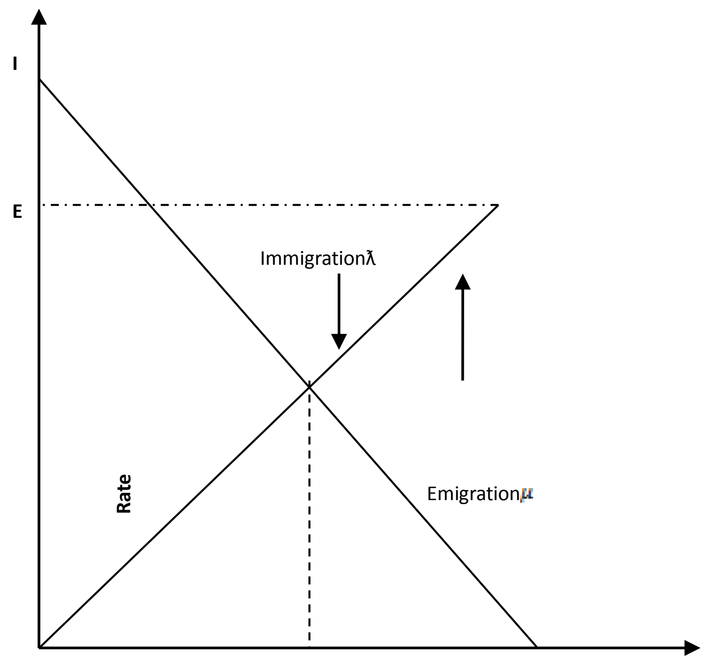


Figure 6 Species replica of an only habitat

Methodology of application of BBO technique to different types of load balancing problems has also been presented in this section. BBO concept is based on the two major steps, e.g., migration and mutation as discussed below.

A. Migration

BBO algorithm candidate in the actual number of the population of the resolution can be embodied as vectors. Each one is a real number in the array (suitability index variable) is considered as. The suitability index variable employment, candidate resolution, namely the fitness of each set, the habitat suitability index is worth can be assessed. Habitat suitability index a dizzying blow to adapt proposals embodies great quality resolution, and low habitat suitability index resolutions embody an inferior solution.

Emigration and migration rates for each resolution would probably be used to allocate the data. With the possibility, habitat modification is recognized as a possibility; supplementary solution installed on each resolution can be adjusted. Select variable suitability index for the amendment, supplement of proposals to choose emigration rates set between the habitat suitability index variables proposals selected will travel randomly selected resolution are used. Migration process in order to prevent proposals from being corrupted by the best, excellence BBO algorithm is maintained in a small way. Here, set the best habitat, ie, those whose houses are the best habitat suitability index, as it is a walk in the migration process is kept short. This process is recognized as process excellence.

B. Mutation

It is well recognized that a little common disasters or events complement the usual habitat suitability index can be modified due to the sudden. BBO suitability index variables such an event is embodied by mutation and species counts are used to determine the potential mutation rates. All the possibilities of a species count (1) can be calculated employing differential equations. An associate every habitat likely possibility is that a resolution to the problem as it exists indicates. If a given resolution is unlikely, next to a smaller resolution as supplementary solution is possible to change. Have raised the possibility of a relatively small supplementary resolution, the resolution is next to mutate very slight chance. Therefore it can be said that the very high resolution and extremely low habitat suitability index habitat suitability indices proposals at a later stage are less likely to hire additional enhanced suitability index variable. But moderate habitat suitability index resolutions mutation subsequent operation is far larger chance to craft proposals. Each mutation rate of resolution of a set number of species likely to pursue employment in the words of the equation can be calculated:

$$\boldsymbol{m(S) = m_{max}\left(\frac{1-Ps}{Pmax}\right)}$$

where m_{max} is a user-defined parameter. The mutation plan to increase diversity between habitats inclines. In the absence of this amendment, additional major habitat highly probable proposals will want to be finished. This mutation habitat suitability index both low and high resolution as possible to change the value of their previous proposals in analogy gives an opportunity to increase both types of builds. Under a resolution of excellence so as inadequate later, the resolution becomes mutation process (mutation that is already set to address) beside predecessor, features a resolution back to save the place again if it can be handed back to the mutation process is maintained. Shock load balancing, if a resolution is selected for mutation next is replaced by a randomly generated set of new resolutions. The supplement, each supplementary plan mutation that has been requested for gas can be requested for BBO.

C. BBO Algorithm

The BBO algorithm can be described in the following way.

Step 1) Initialize the BBO parameters like habitat modification probability Pmod, mutation probability, maximum mutation rate r_{max}, max immigration rate ƛ, max emigration rate E, lower bound and upper bound for immigration probability per gene, step size for numerical integration dt, number of habitat N, number of suitability index variables m, elitism parameter “p” which indicates the number of best habitats to be retained in the habitat matrix as it is, from one generation to the next without performing migration operations on them, etc. Set maximum number of iteration. Generate the SIVs of the given problem within their feasible region using random number. A complete solution consisting of suitability index variables is known as one habitat H. There are several numbers of habitats to search the optimum result.

Step 2) Suppose we are minimizing a function f(x)$\sum_{i=1}^{m} x_1^2 and 0 < xi < z$.Initialize several numbers of habitats depending upon the habitat size within feasible region. Each habitat represents a potential solution to the given problem. So total habitat in matrix form is written in the following forms:

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{1m} \ldots \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{2m} \ldots \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{3m} \ldots \\ x_{N1} & x_{N2} & x_{N3} & x_{N4} & x_{N5} & x_{Nm} \ldots \end{pmatrix}$$

Step 3) Calculate the HSI value for each habitat of the population set for given emigration rate E , immigration rate ƛ. For the function f(x) =$\sum_{i=1}^{m} x_i^2$, habitat suitability index of all *N suitability index variables* sets is calculated as per the following way:

Habitat suitability index$_1$= $[x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 + x_{15}^2 + \ldots\ldots\ldots x_{1m}^2]$

Habitat suitability index$_2$=$[x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 + x_{15}^2 + \ldots\ldots\ldots x_{1m}^2]$

Habitat suitability index$_3$=$[x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 + x_{15}^2 + \ldots\ldots\ldots x_{1m}^2]$

……………………………………………………….

Habitat suitability index$_m$=$[x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 + x_{15}^2 + \ldots\ldots\ldots x_{1m}^2]$

Calculate the number of valid species out of all habitats using their habitat suitability index values. Those habitats, whose fitness values, i.e., habitat suitability index values, are finite, are considered as valid species S.

Step 4) Based on the optimum habitat suitability index value, elite habitats are identified.

Step 5) probabilistically immigration rate and emigration rate are used to modify each non-elite habitat using migration operation. The probability that a habitat H_i is modified is proportional to its immigration rate A; and the probability that the source of the modification comes from a habitat Hj is proportional to the emigration rate /ij. Habitat modification using migration operation can be described as follows.

Select a habitat H_t with probability proportional to ,

If *Hi* is selected

For *j* = 1 to *N*

Select another habitat *Hj* with probability proportional to //., If Hj is selected

Randomly select a $suitability\ index\ variable$ from habitat Hj

Replace a random $suitability\ index\ variable$ in Hi with that selected $suitability\ index\ variables$ of Hj

End

End

End

From this algorithm, we note that elitism can be implemented by setting $\lambda = 0$ for the best p habitats. After each habitat is modified, its feasibility as a problem solution should be verified. If it does not represent a feasible solution, then the above procedure is ignored and the same procedure is performed again in order to map it to the set of feasible solutions. After modification of each non-elite habitat using migration operation, each habitat suitability index is recomputed.

Step 6) for each habitat, the species count probability is updated using (17). Mutation operation is performed on each non-elite habitat as discussed in Section IV-B and habitat suitability index value of each habitat is computed again. Mutation operation can be described as follows.

For $i = 1$ to N

For $j = 1$ to m

Use μi and μi to compute the probability P_t using (17)

Select a *suitability index variables* $H_i(j)$ with probability proportional to Pi

If $Hi(j)$ is selected

Replace with a randomly generated *suitability index variables* within its feasible region

End

End

End

As with habitat modification, elitism can be implemented by setting the probability of mutation selection Pi to zero for the best p habitats. After each habitat is modified, its feasibility as a problem solution should be verified. If it does not represent a feasible solution, then the above step is ignored and the above-mentioned method is applied again in order to map it to the set of feasible solutions.

Step 7) Go to (**step 3)** for the next iteration. This loop can be terminated after a predefined number of iterations have been found.

4.4 BBO for Cloud Load Balancing

In this section, a new approach to implement the BBO algorithm will be described for solving the cloud load problems [49]. Especially, a suggestion will be given on how to deal with the equality and inequality constraints of the cloud load problems when modifying each individual's search point in the BBO algorithm. The process of the BBO algorithm can be summarized as follows.

1) Representation of the SIV: Since the decision variables for the cloud load problems are real virtual machines, they are used to represent individual habitat. The capacity of all virtual machines is represented as the SIV in a habitat. For initialization, choose number of SIV of BBO algorithm m, number of habitat N.

The complete habitat set is represented in the form of the following matrix:

$H= [H^1, H2, H3…H^i…H^N]$

i.e., see the equation at the bottom of the next page, where i = 1, 2...N and j = 1, 2, 3.... M.

Here H^i is the position vector of the habitat i. Each habitat is one of the possible solutions for the cloud load problem. Size of the habitat is equivalent to the population size of GA. The element H^{ij} of H^i is the j^{th} position component of habitat i or in other words $H^{i\ j}$is the jth SIV of the ith habitat. SIV^{ij} represents the capacity of the virtual machine jth of the ith habitat set H^i.

2) Initialization of the SIV: Each element of the Habitat matrix, i.e., each SIV of a given habitat set H, is initialized randomly within the effective real capacity. The initialization is based on (4) for VMs without rate limits and based on (4), (7) for VMs with ramp rate limits.

Now the steps of algorithm to solve cloud load problem are given below.

Step 1) For initialization, choose the number of generator units, i.e., number of SIV is m, number of habitat is N. Specify maximum and minimum capacity of each VM, user demand, B-coefficients matrix for calculation of transmission loss. Also initialize the BBO parameters like habitat modification probability P_{mod}, mutation probability, maximum mutation $rate_{max}$, max immigration rate E, max emigration rate E, lower bound for immigration probability per gene, upper bound for immigration probability per gene, step size for numerical integration dt, elitism parameter "p", etc. Set maximum number of iteration.

Step 2) Each SIV of a given habitat of H matrix is initialized using the concept mentioned in "Initialization of the SIV". Each habitat set of H matrix should satisfy equality constraint (2) using the concept of slack VM as mentioned in Section II-E. Each habitat represents a potential solution to the given problem.

Step 3) Calculate the HSI for each habitat set of the total habitat set for given emigration rate μ, immigration rate ƛ. HSI represent the running cost of the VMs in the datacenter for a particular Task demand. Here, HSI^i indicates the total cost due to the ith set of generation value (i.e., ith set of habitat matrix H) in $/h. If there are m units that must be operated to provide VM to tasks, then the ith individual SIV can be defined as follows:

$$H^i = SIV^{ij} = [SIV^{i1}, SIV^{i2}, \dots SIV^{im}]$$

$$= [PG^{i1}, PG^{i2}, \dots PG^{im}]$$

$$i = 1,2, \dots N, \dots S; j = 1,2,3, \dots m$$

The dimension of the habitat matrix is S x m. All these components in each individual are represented as real values. The matrix represents the total habitat set.

Step 4) Based on the HSI (cost in case of cloud load problem), value elite habitats are identified. Here elite term is used to indicate those habitat sets of VM capacity, which give best cost. Top "p" habitat sets are kept as it is after individual iteration without making any modification on it. Identification of valid species is a little interesting. Those habitats, whose fitness values, i.e., HSI values, are finite, are considered as valid species S in cloud load problem.

Step 5) probabilistically performs migration operation on those SIVs of each non-elite habitat, selected for migration. How to select any SIV for migration operation is described below.

1) First select lower and upper value of immigration rate A_{lower} and A_{upper}, respectively Next calculate Species count:-

For i =1 to N

if cost of habitat set i, HSI_i<inf

SpeciesCount of habitat i = N — r,

Else

SpeciesCount of habitat i = 0;

end

end

2) Then calculate value of and for each habitat set:-

for i = 1 to N

$\lambda(i)$ = I* (1- VMSize(j).SpeciesCount -1)/Population(P);

$\mu(i)$ = E*(VMSize(j).SpeciesCount +1)/Population(P);

end

λ_{MIN}, = minimum value of lambda;

λ_{Max} =maximum value of lambda;

$$\text{Fitness value} = \sum \min(P_G^i)$$

$$VM(P_G^i) = P_G^i > P_{GMax}^i \qquad P_{GMax}^i = P_G^i - P_{GMAX}^I$$

$$VM(P_G^i) = P_G^i < P_{GMin}^i \qquad P_G^i = P_{GMin}^i$$

$$VM(P_G^i) = P_{GMax}^i < P_G^i < P_{GMin}^i \qquad P_G^i = 0$$

3) Next calculate from which habitat and which SIV to be selected for newly generated habitat after migration for

k = 1 to N

If randomly generated number is less than habitat modification probability $P_{'mod}$. Then following operations are calculated

Normalize the immigration rate using the following formula:-.

$$\lambda_{Scale} = \lambda_{lower} + (\lambda_{Upper} - \lambda_{lower}) * (\lambda_{(k)} - \lambda_{MIN}) / (\lambda_{Max} - \lambda_{MIN})$$

$$H = \begin{bmatrix} SIV^{11} & SIV^{12} & SIV^{13} & \ldots & SIV^{1j} & \ldots\ldots & SIV^{1m} \\ SIV^{21} & SIV^{22} & SIV^{23} & \ldots & SIV^{2j} & \ldots\ldots & SIV^{2m} \\ SIV^{31} & SIV^{32} & SIV^{33} & \ldots & SIV^{3j} & \ldots\ldots & SIV^{3m} \\ SIV^{41} & SIV^{42} & SIV^{43} & \ldots & SIV^{4j} & \ldots\ldots & SIV^{4m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\ \vdots & \vdots \\ SIV^{N1} & SIV^{N2} & SIV^{N3} & \ldots & SIV^{Nj} & \ldots\ldots & SIV^{Nm} \end{bmatrix}$$

Pick up a habitat from which to obtain a feature

for j = 1 to m

if a randomly generated number $< \lambda_{Scale}$

RandomNum = rand * sum (μ);

Select = μ (l);

SelectIndex = 1;

while (RandomNum> Select) and (SelectIndex<N)

SelectIndex = SelectIndex + 1:

Select = Select + μ (SelectIndex);

End

Newly generated habitat (k , j)

Old habitats (SelectIndex, j);

Else

Newly generated habitat (k , j) = Old habitat (k . j) ;

End

End

End

After migration operation, new habitat set is generated. In cloud load problems, these represent new modified generation values of VMs (P_G).

(4) Operating Limit Constraint Is Satisfied in the Following Manner:-

If output of ith VM, P_G^i> maximum capacity of s VM, P_{GMax}^i

= $P^i_G - P^I_{GMAX}$

end

If output of ith VMP_G^i< minimum capacity of ith VM (P_{GMIN}^i)

$P^i_G = P^i_{GMIN}$

end

If generation value of ith VM P_{Gi} is within its maximum and minimum generation capacity

$P^i_G = P^i_G$

End

Equality constraint (2) is satisfied using concept of slack VM.

Step 6) Species count probability of each habitat is updated using equation. Mutation operation is performed on the non-elite habitat. If mutation rate as calculated using equation of any habitat is greater than a randomly generated number, then that habitat is selected for mutation. In mutation operation, that habitat set which is selected for mutation is simply replaced by another randomly generated new habitat set that satisfies both equality constraint and inequality constraints of cloud load problems. HSI value of each new habitat set is recomputed

Step 7) Go to (**step 3)** for the next iteration. This loop can be terminated after a predefined number of iterations. After each habitat is modified (steps 5 and 6), its feasibility as a problem solution should be verified, i.e., each SIV should satisfy different operational constraints of VM as mentioned in the specific problem. Equality constraints should also be satisfied.

4.5 Chaos Theory

Chaos theory is a division of mathematics that deals with nonlinear dynamical systems. The only component of the system is a set of negotiations to form a larger whole. Nonlinear way that reaction between components or terminated due to multiple outcomes just becomes something greater than the individual parts add up. Lastly, dynamical systems over the duration adjustment path set your current position. In pursuit of the piece, we cracked a little of the jargon, the interesting characteristics of the disorder is imagination, and vision and to predict their impact on the debate.

Chaotic system of nonlinear dynamical systems is a simple subtype. They can encompass interacting parts extremely inadequate and could pursue these extremely simple laws, but these arrangements are all in their early reliance on the terms of a highly sensitive. However deterministic their simplicity, these arrangements over the period, entirely unexpected and wildly deviating (aka, chaotic) behavior can produce. Edward Lorenz, the father of the theory of disorder portrayed as "determines the current forthcoming, but the estimated current detection concerning future is not."

4.6 Chaos Vs Randomness

These two-dimensional state spaces delineate Poincare plots: An imaginary space embellishment uses its arrangement Vs dimensions variables. Every single point in the state space of a state probably arrangement, and these supplementary words are a set of variable values. The Poincare plots instructive bizarre attractor’s period for functional sequence data (written cradling produced bee logistics map), this was one-dimensional statistics that they are embed into two and three-dimensional state space and bricks.

Ended precision if it can be hard then notify in definite time periods and just random sequence after you fully to note comprehend their underlying dynamics size. There are some examples:

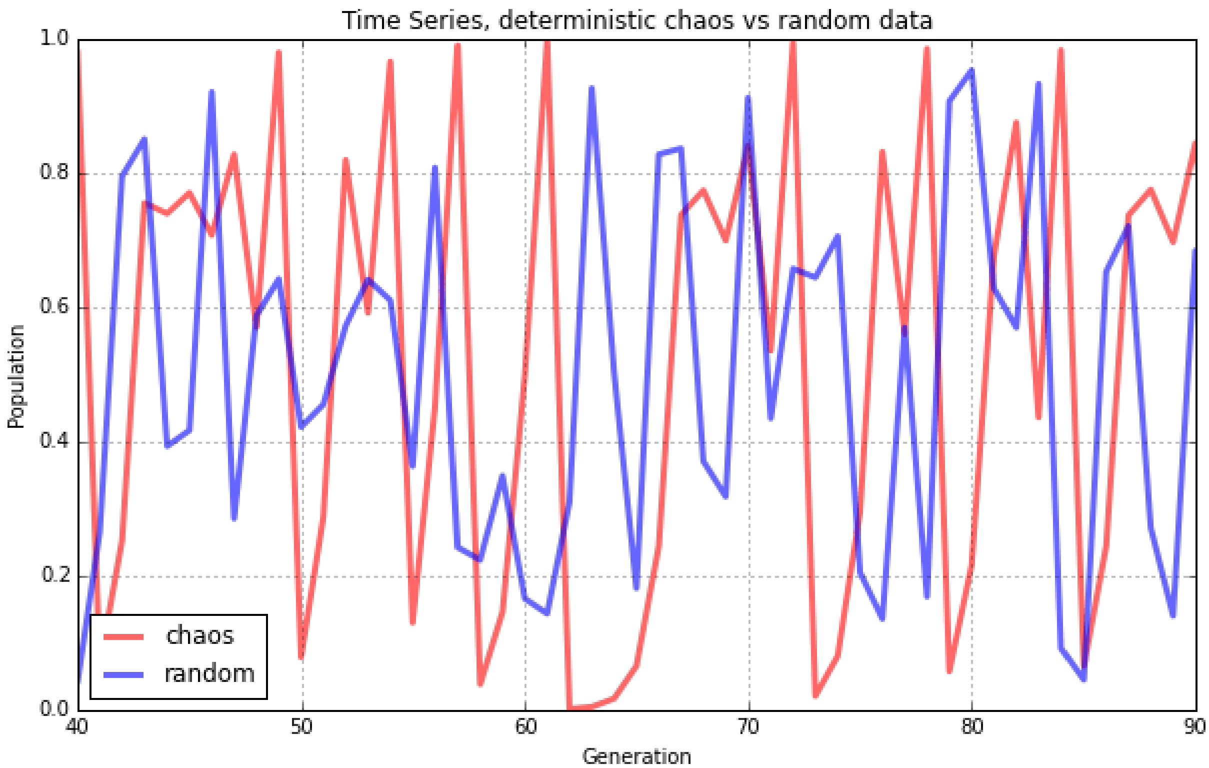


Figure 7 Chaos Vs Randomness

Both of the lines seem to hop concerning randomly. The blue line does depict random data, but the red line comes from our logistic ideal after the development rate is set to 3.99. This is deterministic disorder, but it's hard to differentiate it from randomness. So, let's visualize these alike two data sets alongside Poincare plots instead of line charts:

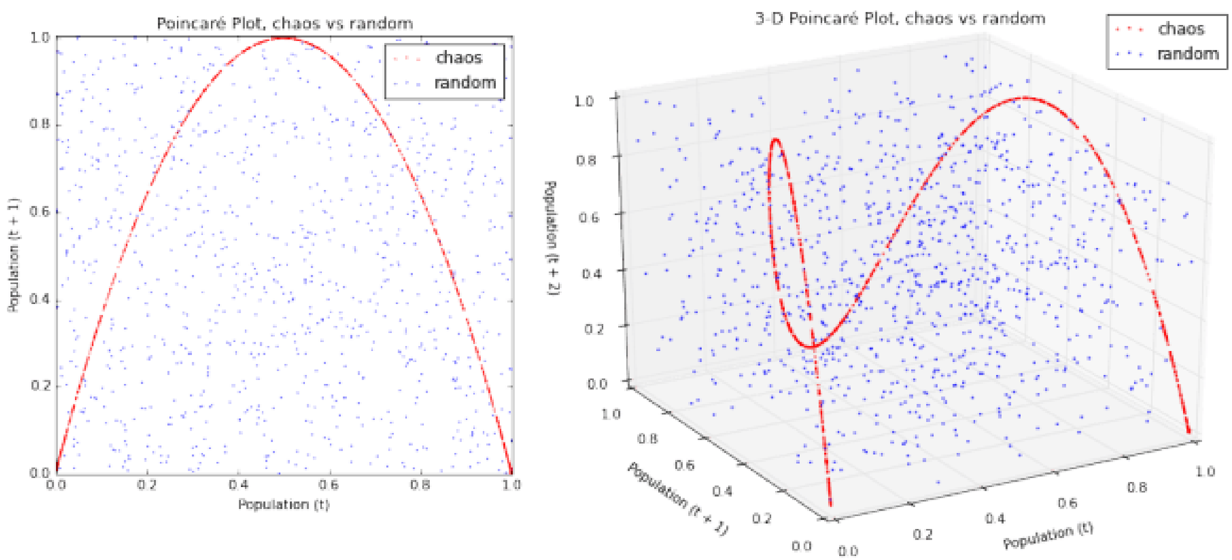


Figure 8 Poincare plots

Currently we can notice our chaotic structure (in red, over) embarrassed by its bizarre attractor. Contrast, the arbitrary data (in blue, over) just look similar to noise. This is yet extra persuasive in the 3-D Poincare design that inserts our time periods into a 3-D state space by portraying the population cost at creation *t + 2* Vs the value at creation *t + 1* Vs the value at *t*.

4.7 Chaotic maps for BBO

At this part, we represent the chaotic maps utilized and illustrate the better performance of BBO by using chaotic methods. Now we are taken ten dissimilar chaotic maps as representing in Table 1, Fig. 4.3.

This is merit, which are mentioning below i.e. Fig. 4.3 describe deterministic arrangement are available as follows:

No.	Name	Chaotic	Range
1	Chebyshev	$x_{i+1} = \cos(icos^{-1}(x_i))$	(0,1)
2	Circle	$x_{i+1} = \text{mod}\left(x_i + b - \left(\frac{a}{2\pi}\right)\sin(2\pi x_k), 1\right),$ $a = 0.5\ and\ b = 0.2$	(0,1)
3	Gauss/mouse	$x_{i+1} = \begin{cases} 1 & x_i = 0 \\ \frac{1}{mod(x_i, 1)} & otherwise \end{cases}$	(0,1)
4	Iterative	$x_{i+1} = sin\left(\frac{a}{2\pi}\right), a = 0.7$	(-1,1)
5	Logistic	$x_{i+1} = ax_i(1 - x_i), a = 4$	(0,1)
6	Piecewise	$x_{i+1} = \begin{cases} \frac{x_i}{P} & 0 \leq x_i \leq P \\ \frac{x_i - P}{0.5 - P} & P \leq x_i \leq 0.5, P = 0.4 \\ \frac{1 - P - x_i}{0.5 - P} & 0.5 \leq x_i < 1 - P \\ \frac{1 - x_i}{P} & 1 - P \leq x_i < 1 \end{cases}$	(0,1)
7	Sine	$x_{i+1} = \frac{a}{4}\sin(\pi x_i), a = 4$	(0,1)
8	Singer	$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4), \mu = 1.07$	(0,1)

9	Sinusoidal	$x_{i+1} = ax_i^2 \sin(\pi x_i)\,, a = 2.3$	(0,1)
10	Tent	$x_{i+1} = \begin{cases} \frac{x_i}{0.7} & x_i < 0.7 \\ \frac{10}{3}(1 - x_i) & x_i \geq 0.7 \end{cases}$	(0,1)

Table 1 Chaotic maps

The table above is containing random component, but the equations are fairly clear image of chaotic behavior. 3. This set of chaotic chart, has been selected with unequal treatment as early point is 0.7 for all. Starting point 0 and 1 (or -1 and 1, depending on the scope of chaotic maps) between each number can be selected.

We BBO algorithm selection, migration, and mutation to affect operators retain chaotic chart. Selection and chaotic emigration operators, enhancing discovery, exploitation increases the chaotic mutation. BBO chaotic chart a different algorithm is used for ten variants. Each one of the main operators of the chart chaotic request is tested alone and in combination on the next.

4.7.1 Chaotic maps for selection

Since we can observe in Fig. 4.2, the probability of choosing of a habitat for migration is denoted by k. By utilizing the chaotic map, we can describe this probability. The final outcomes through the chaotic map must lie in the period [0, 1], thus we standardize those in [-1, 1]. According to this study, the rand outcomes (underlined in Fig. 4.2) are replaced with the help of outcomes of the chaotic map to give chaotic behaviors for the choosing operator are given as:

If C (t) $<\lambda_i$ after that

Emigrates habitants (H_i to H_j) selected through probability comparative to μ_i

End if

Where C (t) is the outcome through the chaotic map in the t-th execution and H_i explains i-th habitat of the procedure. It is to be note that incorporating the chaotic maps methods into the BBO assortment phase is like to that. This is the secondary thing from Eq. (4) that the chaotic maps are dependable for selecting the cause of immigration in the projected chaotic selection operator.

4.7.2 Chaotic maps for emigration

Though mentioned in Fig. 2, emigration E is achieve with probability proportional to l after choosing a habitat. We utilized the chaotic map to compute this probability in the following way:

If C (t) <λ_i then

Select rand outcomes of habitant in x_i and replace it with x_j

End if

Where C (t) is the outcome through the chaotic map in the t-th execution and xi describes i-th habitant. Equation (5) describes that the chaotic maps are permissible to classify the probability of emigration and therefore chaotic migration behaviors. Note that we again normalize those chaotic maps that lie in the interval [-1, 1]

4.7.3 Chaotic maps for mutation

The probability of mutation is described straightly by the chaotic map in a given ways:

For i = 1 to no. of habitants at k-th habitat

If C (t) < Mutation_ rate (k) after that

Mutate i-th habitants

End if

End for

Where C (t)-th execution and Mutation rate (k) is the k-th outcome of the chaotic map habitat mutation rate, which (3) or some further mutation is probable to be described by the creator explain. It is noted that the mutation time [0, 1] is widespread to the chain.

After that next part of this described as, we give a relative learning by utilizing these novel chaotic operators and rebellious it called BBO (CBBO). Besides, this type of chaotic BBO operators (selection / emigration and select / emigration / mutation with BBO) is a mixture of urbanized and standard.

The following explanation describes how the proposed method can aid in efficient operators theoretically either individually or together:

- Chaotic BBO utilized the chaotic selection operator to pick habitats excitedly, which recovers study.
- The chaotic emigration operator allows Chaotic BBO to perform emigration with a chaotic pattern that again emphasizes exploration.
- Chaotic BBO utilized the chaotic mutation operator to develop the explore space which is superior to BBO because there should be various outcomes for mutation possibility.
- Various chaotic maps for selection, mutation and emigration give dissimilar investigation and development outline for the Chaotic BBO.

- Given that chaotic maps demonstrate chaotic activities, a creation of Chaotic BBO should underline either investigation or development.
- At this phase, trap in limited optima, the chaotic variety and migration operators collectively help Chaotic BBO to leave from them.
- Discovering a capable area(s) of search space in that case is crucial, the chaotic alteration operator assists Chaotic BBO to excitedly develop the region.

CHAPTER 5

RESULTS AND ANALYSIS

MATLAB

MATLAB technical computing is an impressive speech. Period MATLAB stands for matrix laboratory, because its open data agent a matrix (array) is. The math computations, modeling and simulation, data scrutiny and processing, visualization and graphics, and high-level software design to the development of algorithm 4 is the creation of speech. It allows matrix manipulations, objectives and data, implementation of algorithms, user interface, C, C ++, Java, FORTRAN creation of plotting with the creation of the plan: the data analysis algorithm, makes and models and applications. MATLAB is case sensitive: the difference between capital and small letters. For example, BB, Bb, bB, and bb would refer to four different variables.

Features:

- It gives built in graphics for Visualizing data & tools.
- Custom plots, improve code quality, maintainability & maximizing performance.
- Building application with custom graphical interface.
- It provides functions for integrating MATLAB based algorithm.

Applications:

1) **Control system:** The discovery and design of a manipulation arrangement includes assorted steps such as modeling, simulation, scrutiny and optimization. The reply of a manipulation arrangement is learned by subjecting the ideal to average input signals. If the reply is unsatisfactory, arrangement is enhanced and optimized by adjustment of arrangement parameters or incorporating suitable compensation devices.
   Mathematical equations and manipulate individual components of the system input-output characteristics of the works featured. There are two ways to design scrutiny and manipulation systems. Classical or frequency domain approach is the way to quickly shouted. In this way, the difference or differential equations, Laplace or Z inputs and outputs plus the transfer of employment to changes in the objectives are modified. Such linear time-invariant systems are applied. After the way the state-variable or current perturbation theory in a way that is time-domain approach. It's such a rush and non-linear multi-input multi-output system in the form of manipulation of the system is applicable to a vast scope, in terms of the method needs additional Laplace Transformation.

2) **Test Measurements:** The Test & Measurement Tool (TM tool) enables us to configure and control resources (instruments, sequential devices, interfaces, drivers etc.) available through the toolbox without having to write down the MATLAB script.

We can utilize the Test & Measurement Tool to control our session with the toolbox. This tool allows us to do the following:

- Detect existing hardware and drivers.
- Connect to a tool or device.
- Configure tool or device settings.
- Read and write information.
- Automatically produce the MATLAB script.
- Visualize acquired information
- Export acquired information to the MATLAB workspace.

3) **Computational Biology:** Computational biology, mathematical and statistical researcher's multimedia, computer modeling, and computational and engineering methods by adopting the supplementary test are addressed. Computational biology that is flexible multimedia needs, supports various requests, scalable, fast, heavy data sets can understand and placement capabilities. It is amazing that the result of countless scrutiny pharmacokinetic modeling and statistical functions for deployment at the request of a spectrum of MATLAB is adopted.

   Computational biologists are presently working to incorporate analysis techniques, such as microarray and mass spectrometry scrutiny, to provide them various views into genomic and proteomic statistics sets and get better perceptive of diseases and medical circumstances. They are also discovering ways to series of the human genome that will ultimately enable doctors to expand treatments accurately to regulate individual patient genomes. Jobs such as this would not be possible without the most recent advances in h/w and s/w. They must work personally with life scientists who know molecular biology or chemistry but they are not encoding or math specialists. As a solution, computational biology has developed into the latest engineering control. Computational biologists normally arrive from computer science, mathematics, or engineering.

5.1 Cloud Workload traces

Numerous associations depend on MapReduce to hold their large-scale data processing requirements. As business across various industries accept MapReduce beside parallel databases, novel MapReduce workloads have appeared that characteristic several small, tiny, and more and more interactive jobs. These large-scale interactive query processing workloads, RDBMS community with the expertise of a particular batch computation period, and shares common sense will depart from the terms of use MapReduce target early. Consequently, MapReduce and MapReduce queries for detail like software design methods to optimize the query to present studies on the request to hold the substantial gains are possible. However, incorporating these ideas into business-critical system configuration and performance tuning realize MapReduce workloads generating resist benchmarking needs. Such workloads in terms of current knowledge are manipulated to a handful of companies. A cross workload analogy is missing so far, and use cases beyond the knowledge industry have not been described. Operators MapReduce workloads increasing variety of manufacturing companies and industries to illustrate a pressing demand Crafts.

Hypotheses on Workload Behavior

We can expand hypotheses about workload performance founded on earlier work done. Below are various key questions to put about MapReduce workload.

1. **For optimizing the fundamental storage system:**

- How consistent or skewed are the information accesses?
- How much sequential locality is present?

2. **For workload-level delivering and load determining:**

- How standard or random is the cluster load?
- How huge are disintegrate in the workload?

3. **For job-level scheduling and implementation planning:**

- What are the ordinary task types?
- What are the size, form, and time of these tasks?
- How repeatedly does each task type appear?

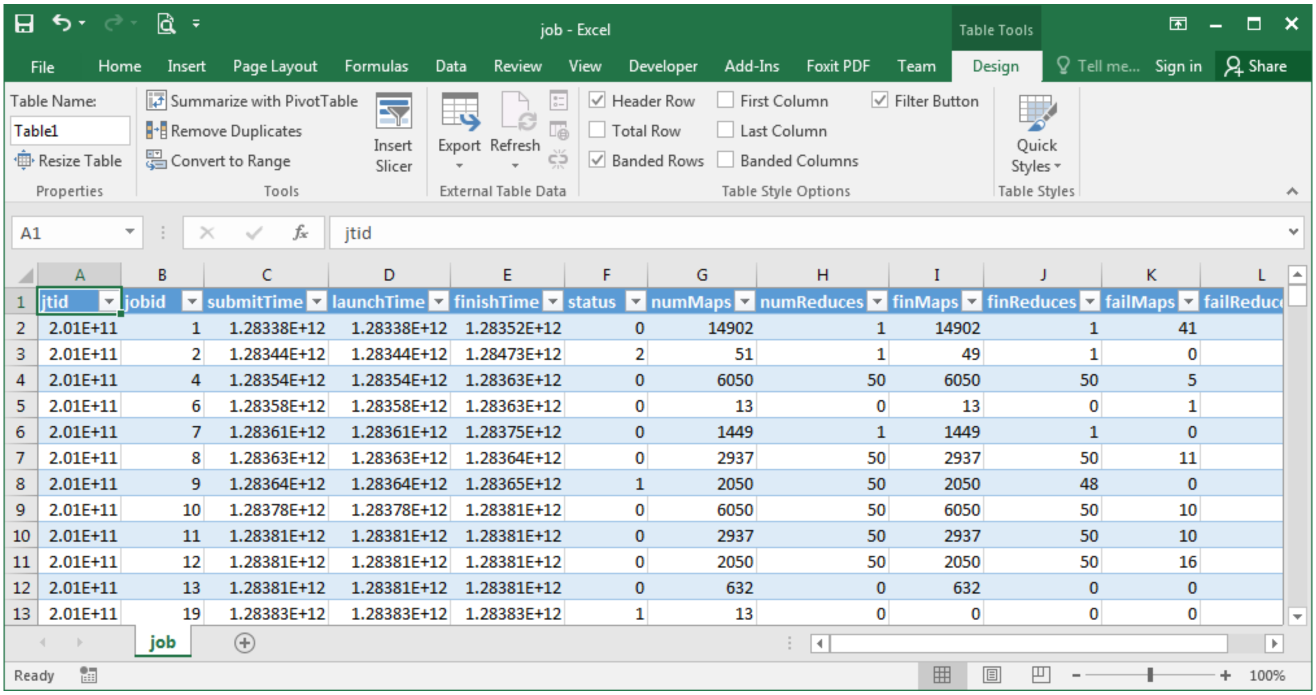
4. **For optimizing query like encoding structures:**

- What percentage of cluster load arrives from these structures?
- What are the ordinary uses of every structure?

5. **For presentation evaluation between systems:**

- How much difference subsists between workloads?
- Can we distill characteristics of a preventative workload?

In figure 9 data is detained from a Hadoop cluster grasped by CMP's Parallel Data Lab, these traces provide tremendously methodical visions in the workload of a cluster utilized for logical workloads for a 20-month period. That includes timestamps, slot counts, and extra.



	jtid	jobid	submitTime	launchTime	finishTime	status	numMaps	numReduces	finMaps	finReduces	failMaps	failReduc
2	2.01E+11	1	1.28338E+12	1.28338E+12	1.28352E+12	0	14902	1	14902	1	41	
3	2.01E+11	2	1.28344E+12	1.28344E+12	1.28473E+12	2	51	1	49	1	0	
4	2.01E+11	4	1.28354E+12	1.28354E+12	1.28363E+12	0	6050	50	6050	50	5	
5	2.01E+11	6	1.28358E+12	1.28358E+12	1.28363E+12	0	13	0	13	0	1	
6	2.01E+11	7	1.28361E+12	1.28361E+12	1.28375E+12	0	1449	1	1449	1	0	
7	2.01E+11	8	1.28363E+12	1.28363E+12	1.28364E+12	0	2937	50	2937	50	11	
8	2.01E+11	9	1.28364E+12	1.28364E+12	1.28365E+12	1	2050	50	2050	48	0	
9	2.01E+11	10	1.28378E+12	1.28378E+12	1.28381E+12	0	6050	50	6050	50	10	
10	2.01E+11	11	1.28381E+12	1.28381E+12	1.28381E+12	0	2937	50	2937	50	10	
11	2.01E+11	12	1.28381E+12	1.28381E+12	1.28381E+12	0	2050	50	2050	50	16	
12	2.01E+11	13	1.28381E+12	1.28381E+12	1.28381E+12	0	632	0	632	0	0	
13	2.01E+11	19	1.28383E+12	1.28383E+12	1.28383E+12	1	13	0	0	0	0	

Figure 9 Sample Map reduce Job Trace format taken from http://ftp.pdl.cmu.edu/pub/datasets/hla/

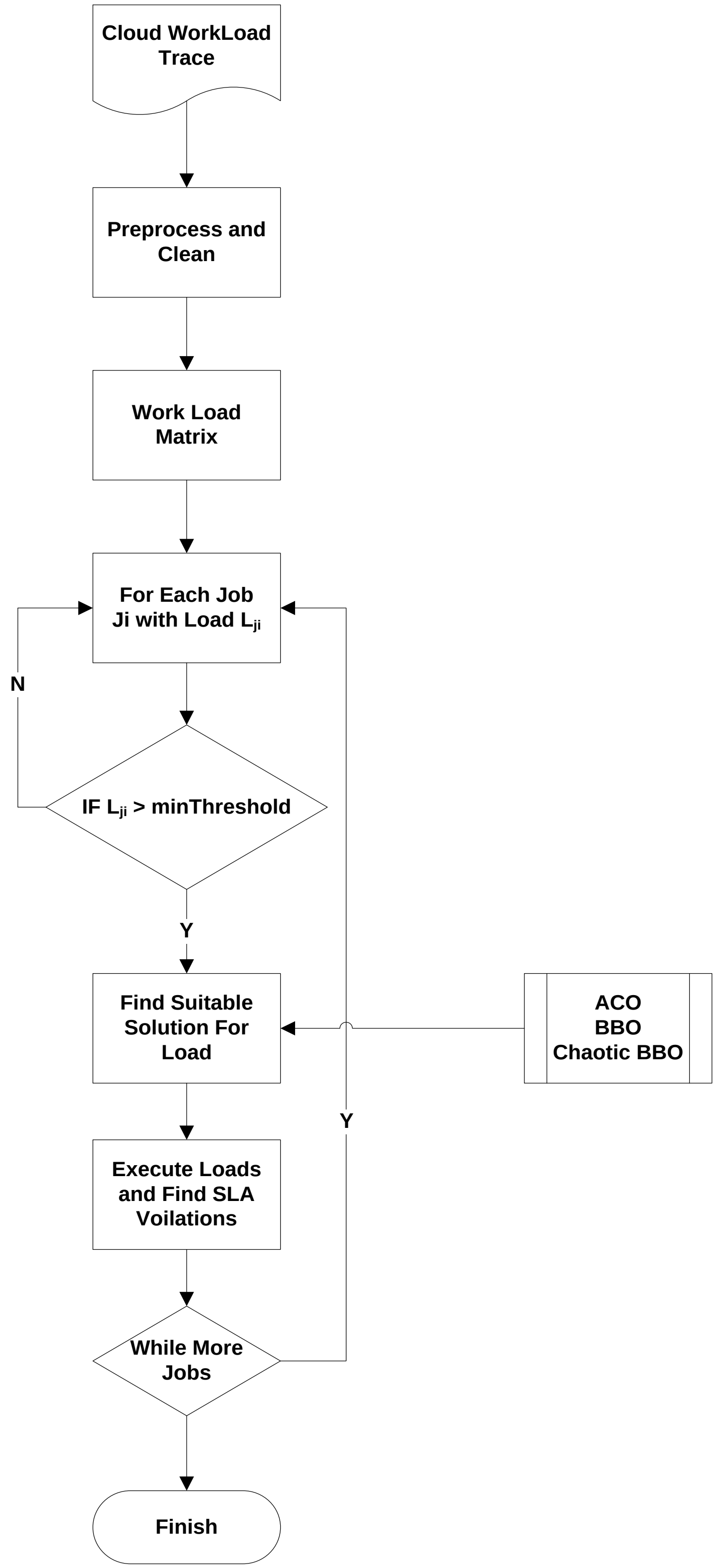


Figure 10 Proposed Workflow for Chaotic BBO based Load Balancing Algorithm

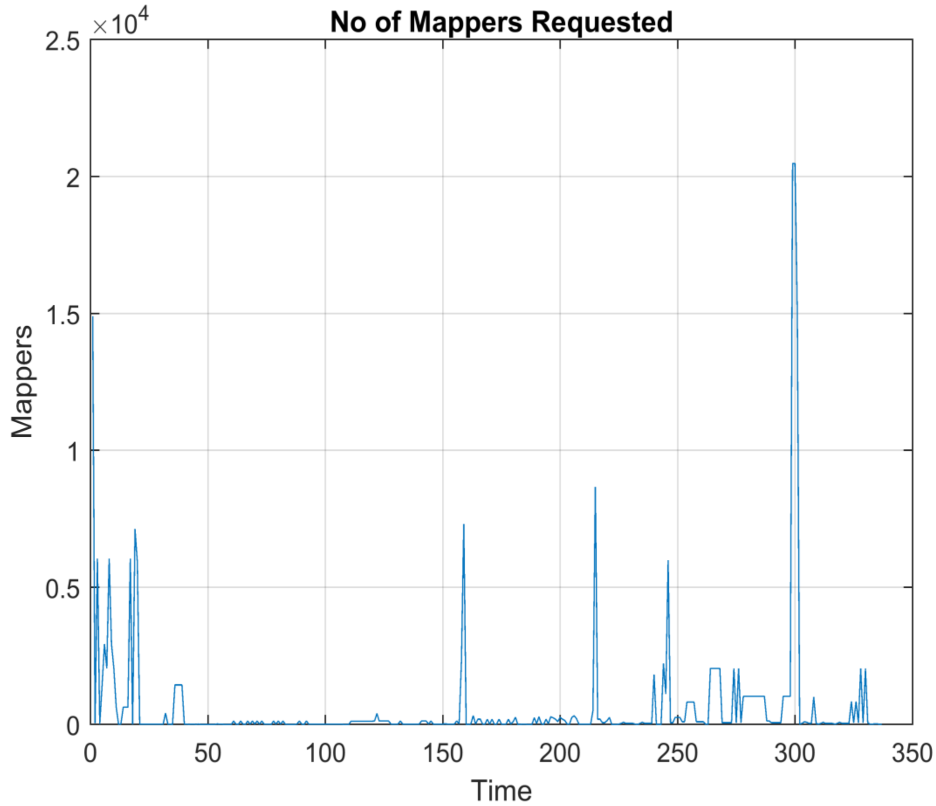


Figure 11 No of Mappers requested at cloud with respect to time

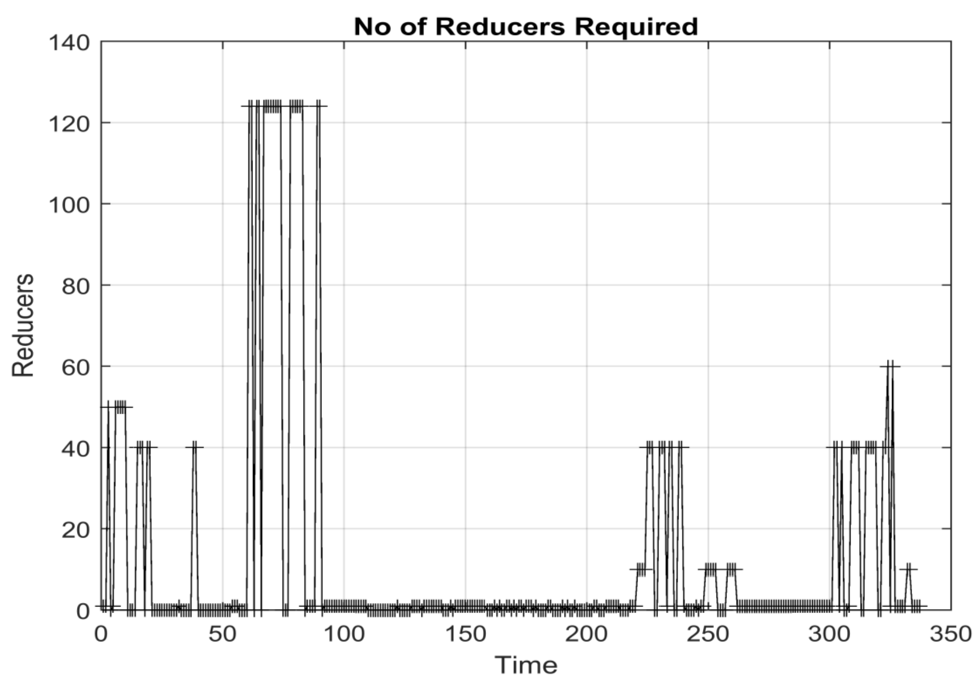


Figure 12 No of Reducers required completing cloudlet tasks

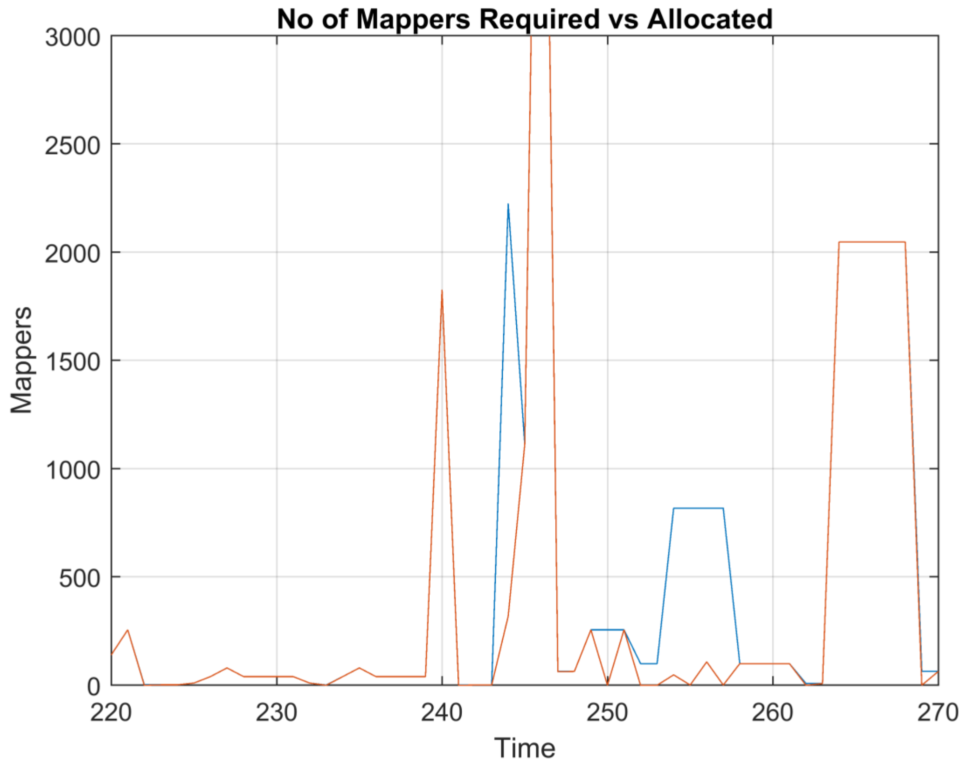


Figure 13 Final Allocation of Mappers Requested and Allocated over time

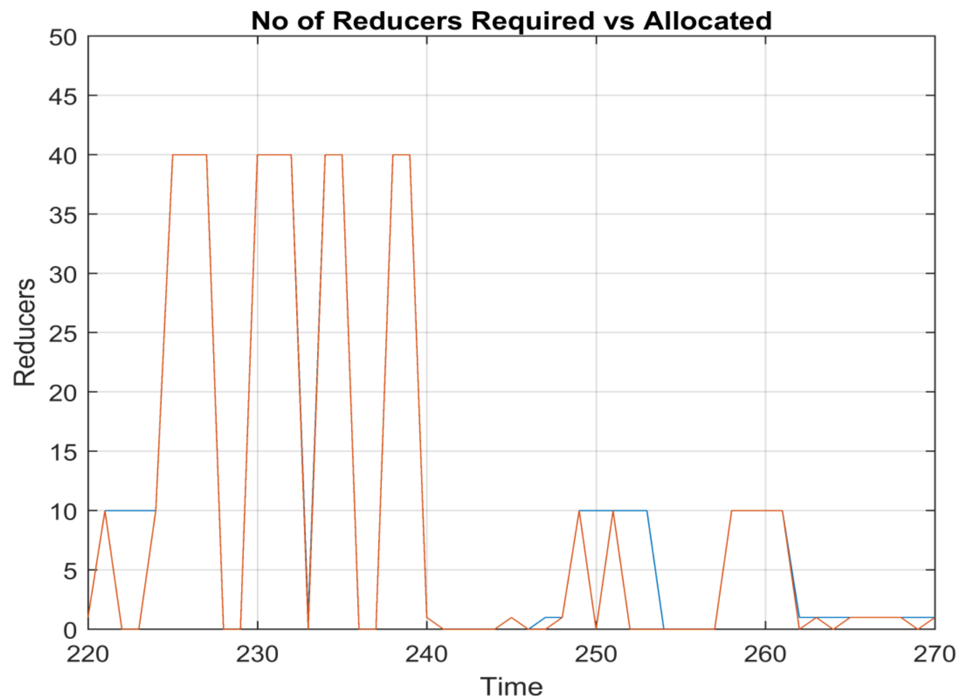


Figure 14 Final Allocation of Mappers Requested and Allocated over time

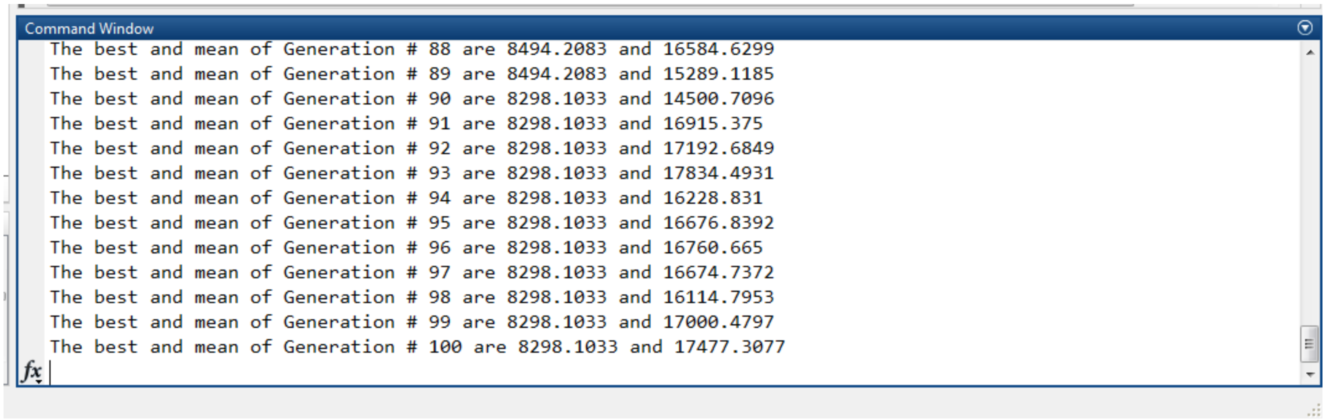


Figure 15 the best and mean of Generation for ACO, BBO and Chaotic BBO

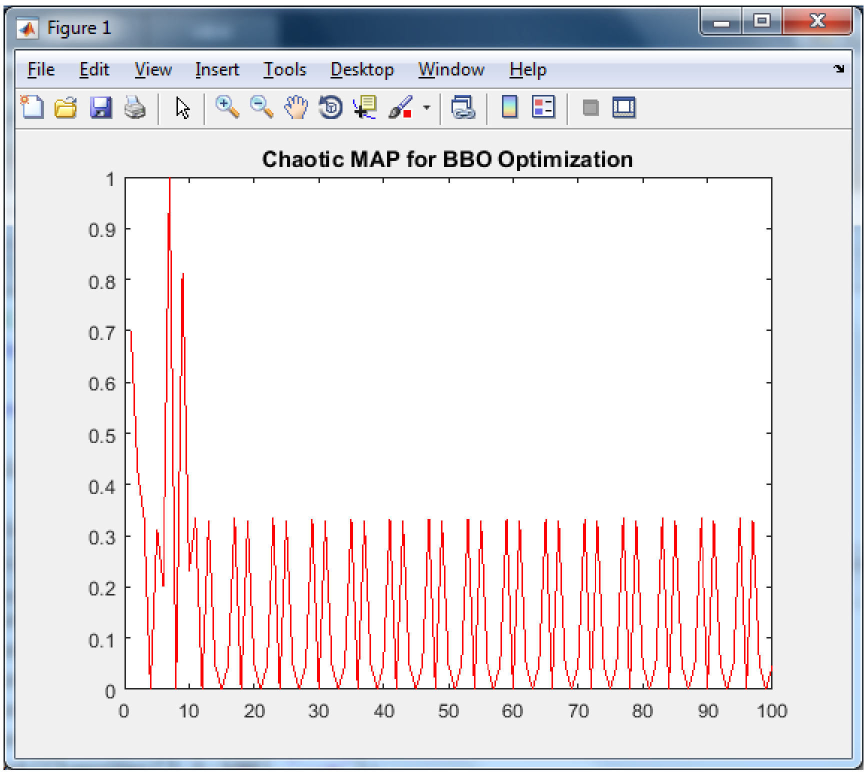


Figure 16 Chebyshev map Chaotic MAP for BBO Optimization

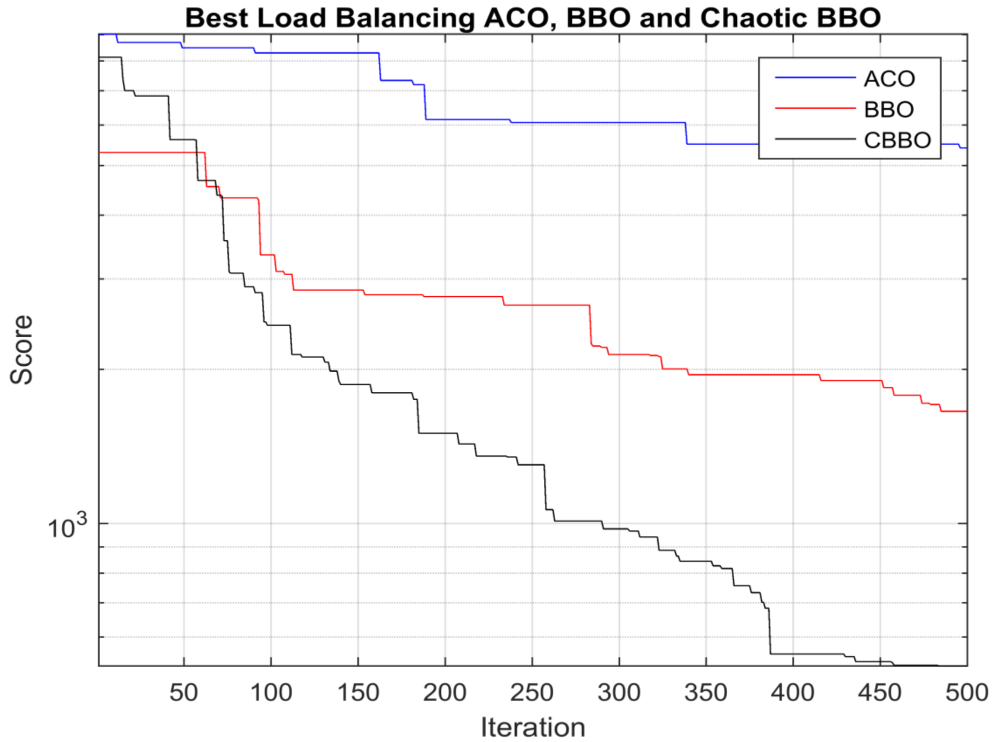


Figure 17 Best Load Balancing Score for Various Algorithms

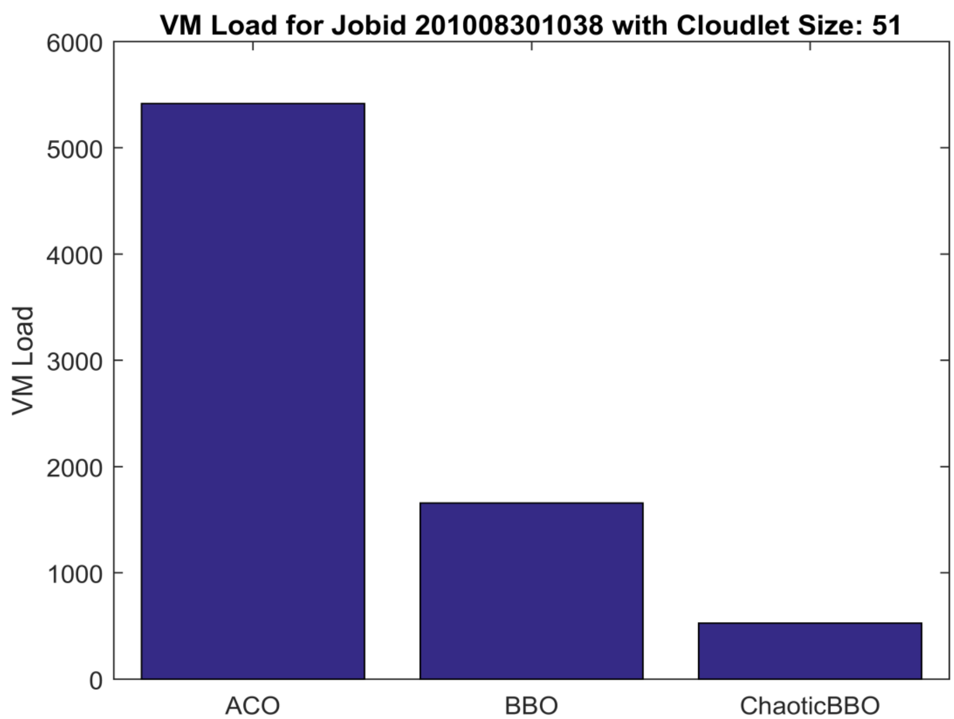


Figure 18 Minimum Virtual Machine Load For a given Cloud let size

CHAPTER 6

CONCLUSION

Cloud computing is a combination of parallel and distributed system that contains a set of interrelated and virtual computers. With the rising requirement and advantages of cloud computing infrastructure, various computing can be completed on cloud atmosphere. One of the primary problems in this atmosphere is related to job scheduling. Cloud job scheduling is an NP-hard optimization trouble, and several meta-heuristic algorithms have been projected to resolve it. A high-quality job scheduler should get used to its scheduling plan to the altering atmosphere and the types of jobs. A cloud job load algorithm for load balancing is compared with various scheduling algorithms has been projected in this work for example ACO (Ant Colony Optimization) and biogeography-based optimization (BBO).

The biogeography-based optimization (BBO) is comparatively novel and has the benefit of get used to new better results. This BBO adaptive procedure is contrasts to the Genetic Algorithm and many other heuristic algorithms that can be a reproductive process. We have to utilize the ordinary BBO to find the best task schedule of cloud computing that also including better Chaotic BBO in this research paper. The Chaotic BBO creates the novel habitats by pertaining the migration and emigration for every non-elite habitat from the many finest habitats and modernizes the habitats if novel produced habitat is improved than old one which is based on Chaotic Maps. In this paper, we recommend a distributed VM Load balancing algorithm called Chaotic BBO with extreme consistency and scalability over Cloud Workload Tracing. While finding the finest mapping relationship is severely NP hard, the Chaotic BBO approach resolves the issue of overload stability in data centers by utilizing Bio Geographic Optimization approach to find out a best-optimal result. When we contrasted to the presented migration approaches, Chaotic BBO not only minimize the number of migration, but also reduce make span. In addition, it attains load balance in a data center which is a most important.

CHAPTER 7

FUTURE WORK

Load balancing is playing a vital role in cloud computing challenges. It is necessary to distribute the load uniformly at every node of the cloud. An extremely congested supplier may fall to make available efficient services to its clients. Therefore, with appropriate load balancing algorithm organization service and throughput can be greater than before.

For a future work, we will provide an arrangement to find better results of the issues of overload balance by utilizing more enhanced Optimization algorithm; we will also provide an arrangement to expand our model to a condition that there are numerous data centers available. In addition, the current researchers have discovered that data centers use a unique amount of electrical power; therefore, we plan to get better task-scheduling optimization and create optimization strategy to optimize not only the effectiveness but also the power. On the other hand, when an unusual application model is used into account, optimizing the complete number of needs served with a back up content for another possible objective that should be examined in future work.

REFERENCES

1) Thiyagarajan, B., and R. Kamalakannan. "Data integrity and security in cloud environment using AES algorithm." Information Communication and Embedded Systems (ICICES), 2014 International Conference on. IEEE, 2014.
2) Dinh, Hoang T., et al. "A survey of mobile cloud computing: architecture, applications, and approaches." Wireless communications and mobile computing 13.18 (2013): 1587-1611.
3) Sran, Nayandeep, and Navdeep Kaur. "Comparative analysis of existing load balancing techniques in cloud computing." International Journal of Engineering Science Invention 2.1 (2013): 60-3.
4) Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011): 20-23.
Wen, Wei-Tao, et al. "An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment". Frontier of Computer Science and Technology (FCST), 2015 Ninth International Conference on IEEE, 2015.
5) Zissis, Dimitrios, and Dimitrios Lekkas. "Addressing cloud computing security issues." Future Generation computer systems 28.3 (2012): 583-592.
6) Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011): 20-23.
7) Singh, Swarnpreet, and Tarun Jangwal. "Cost breakdown of public cloud computing and private cloud computing and security issues." International Journal of Computer Science & Information Technology 4.2 (2012): 17.
8) Bohli, J-M., et al. "Security and privacy-enhancing multicloud architectures."Dependable and Secure Computing, IEEE Transactions on 10.4 (2013): 212-224.
9) Dillon, Tharam, Chen Wu, and Elizabeth Chang. "Cloud computing: issues and challenges." Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on IEEE, 2010.
10) Nuaimi, Klaithem Al, et al. "A survey of load balancing in cloud computing: Challenges and algorithms." Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on. IEEE, 2012.
11) Begum, Suriya, and Dr Prashanth CSR. "Review of load balancing in cloud computing." IJCSI International Journal of Computer Science Issues 10.1 (2013): 1694-0784.
12) Kaur, Sukhvir, and Supriya Kinger. "Minimizing Virtual Machine Migration for Efficient Resource Management in Green Clouds."
13) Yang, Yubin, Hui Lin, and Jixi Jiang. "Cloud analysis by modeling the integration of heterogeneous satellite data and imaging." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 36.1 (2006): 162-172.
14) Zhao, Yi, and Wenlong Huang. "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud." INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on. IEEE, 2009.

15) Zhang, Zehua, and Xuejie Zhang. "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation."Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on. Vol. 2. IEEE, 2010.
16) Feller, Eugen, Louis Rilling, and Christine Morin. "Energy-aware ant colony based workload placement in clouds." Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing. IEEE Computer Society, 2011.
17) Li, Kun, et al. "Cloud task scheduling based on load balancing ant colony optimization." Chinagrid Conference (ChinaGrid), 2011 Sixth Annual. IEEE, 2011.
18) Zhu, Zhiliang, et al. "SLA based dynamic virtualized resources provisioning for shared cloud data centers." Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011.
19) Babukarthik, R. G., R. Raju, and P. Dhavachelvan. "Energy-aware scheduling using Hybrid Algorithm for cloud computing." Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on. IEEE, 2012.
20) Feller, Eugen, Christine Morin, and Armel Esnault. "A case for fully decentralized dynamic VM consolidation in clouds." Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on. IEEE, 2012.
21) Nishant, Kumar, et al. "Load balancing of nodes in cloud using ant colony optimization." Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on. IEEE, 2012.
22) Sotiriadis, Stelios, Nik Bessis, and Nick Antonpoulos. "Decentralized meta-brokers for inter-cloud: Modeling brokering coordinators for interoperable resource management." Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on. IEEE, 2012.
23) Wen, Xiaotang, Minghe Huang, and Jianhua Shi. "Study on resources scheduling based on ACO allgorithm and PSO algorithm in cloud computing." Distributed Computing and Applications to Business, Engineering & Science (DCABES), 2012 11th International Symposium on. IEEE, 2012.
24) Acharya, Sanjeev, and Demian Antony D'Mello. "A taxonomy of Live Virtual Machine (VM) Migration mechanisms in cloud computing environment."Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference on. IEEE, 2013.
25) Ahmad, Nafees, Ayesha Kanwal, and Muhammad Awais Shibli. "Survey on secure live virtual machine (VM) migration in Cloud." Information Assurance (NCIA), 2013 2nd National Conference on. IEEE, 2013.
26) Ajit, M., and G. Vidya. "VM level load balancing in cloud environment."Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on. IEEE, 2013.

27) Al-Rayis, Ektemal, and Heba Kurdi. "Performance analysis of load balancing architectures in cloud computing." Modelling Symposium (EMS), 2013 European. IEEE, 2013.
28) Chaukwale, Rajesh, and S. Sowmya Kamath. "A modified ant colony optimization algorithm with load balancing for job shop scheduling."Advanced Computing Technologies (ICACT), 2013 15th International Conference on. IEEE, 2013.
29) Raju, R., et al. "Minimizing the makespan using Hybrid algorithm for cloud computing." Advance Computing Conference (IACC), 2013 IEEE 3rd International. IEEE, 2013.
30) Tantar, Alexandru-Adrian, et al. "Computational intelligence for cloud management current trends and opportunities." Evolutionary Computation (CEC), 2013 IEEE Congress on. IEEE, 2013.
31) Tawfeek, Medhat A., et al. "Cloud task scheduling based on ant colony optimization." Computer Engineering & Systems (ICCES), 2013 8th International Conference on. IEEE, 2013.
32) Yuan, Wei, et al. "Towards efficient deployment of cloud applications through dynamic reverse proxy optimization." High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on. IEEE, 2013.
33) Sun, Weifeng, et al. "PACO: A period ACO based scheduling algorithm in cloud computing." Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on. IEEE, 2013.
34) Wei, Xianglin, et al. "Bio-inspired application scheduling algorithm for mobile cloud computing." Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on. IEEE, 2013.
35) Liu, Xiaodong, Lin Liu, and Hongji Yang. "TARGO: Transition and reallocation based green optimization for cloud VMs." Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing. IEEE, 2013.
36) Abolfazli, Saeid, et al. "Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges." Communications Surveys & Tutorials, IEEE 16.1 (2014): 337-368.
37) Ahmed, Arif, and Abadhan Saumya Sabyasachi. "Cloud computing simulators: A detailed survey and future direction." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.
38) Ashraf, Adnan, and Ivan Porres. "Using ant colony system to consolidate multiple web applications in a cloud environment." Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on. IEEE, 2014.

39) Caton, Simon, et al. "A social compute cloud: allocating and sharing infrastructure resources via social networks." Services Computing, IEEE Transactions on 7.3 (2014): 359-372.
40) Tsai, Chun-Wei, et al. "A hyper-heuristic scheduling algorithm for cloud."Cloud Computing, IEEE Transactions on 2.2 (2014): 236-250.
41) Tsai, Chun-Wei, and Joel JPC Rodrigues. "Metaheuristic scheduling for cloud: A survey." Systems Journal, IEEE 8.1 (2014): 279-291.
42) Hung, Pham Phuoc, et al. "Task scheduling for optimizing recovery time in cloud computing." Computing, Management and Telecommunications (ComManTel), 2014 International Conference on. IEEE, 2014.
43) Rahman, Mosaddequr, Sajid Iqbal, and Jerry Gao. "Load balancer as a service in cloud computing." Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on. IEEE, 2014.
44) Sanaei, Zohreh, et al. "Heterogeneity in mobile cloud computing: taxonomy and open challenges." Communications Surveys & Tutorials, IEEE 16.1 (2014): 369-392.
45) Buyya, Rajkumar, Suraj Pandey, and Christian Vecchiola. "Market-Oriented Cloud Computing and The Cloudbus Toolkit." Large Scale Network-Centric Distributed Systems (2013): 319-358. Wadhwa, Bimlesh, and Amandeep Verma. "Energy saving approaches for green cloud computing: A review." Engineering and Computational Sciences (RAECS), 2014 Recent Advances in. IEEE, 2014.
46) Wen, Yonggang, et al. "Cloud mobile media: Reflections and outlook."Multimedia, IEEE Transactions on 16.4 (2014): 885-902.
47) Mathur, Ms Divya. "A New Methodology for Solving Different Economic Dispatch Problems.
48) Simon, Dan. "Biogeography-based optimization." Evolutionary Computation, IEEE Transactions on 12.6 (2008): 702-713.
49) Mathur, Divya. "Biogeography Based Optimization of Different Economic Dispatch Problems."