CLOUD RESOURCE OPTIMISATION

(Comparison of Probabilistic Optimization Algorithms)

Project Report submitted in fulfillment of requirement of degree

Of

Bachelor of Technology

In

Computer Science and Engineering

By

Harsh Rana(131328)

Under the supervision of

Dr. S.P Ghrera

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **CLOUD RESOURCE OPTIMISATION** (Comparison of Probabilistic Optimization Algorithms) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2016 to May 2017 under the supervision of **Dr. S.P. Ghrera** ,HOD, Department os CSE & IT.

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

(Student Signature)

Harsh Rana(131328)

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

Dr. S.P Ghrera

Head of Department, CSE

Dated:

ACKNOWLEDGEMENT

I owe my profound gratitude to my project supervisor **Dr. S.P Ghrera**, who took keen interest and guided me all along in my project work titled **—Cloud Resource Optimisation(Comparison of Probabilistic Optimization Algorithms**), till the completion of our project by providing all the necessary information for developing the project. The project development helped us in research and i got to know a lot of new things in our domain. I am really very thankful to him.

TABLE OF CONTENT

ABSTRACT

"Cloud Computing" has significantly made its landmark in the field of information technology. A concept which initially stood nebulous is now used as a synonym for internet. It has paved a way to increase capacity as well as add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software."Cloud computing in broad sense can be understood as a connected pool of computing resources (including software and hardware) which can eventually be delivered as a service over a network (in broad sense the Internet)". Resources are available to the user in utility-style infrastructure. However management of resources in a dynamic fashion stands a vast area for researchers. These resources are precisely available at certain fixed times and also for fixed intervals of time. Thus Scheduling of these resources constitutes a major part of resource management. An "optimized" scheduling of resources is required to maintain separation between users of the resources. In this paper, we discuss various resource scheduling strategies that have been and are being implemented in "cloud computing" environments. We also propose a comparison among the characteristics features of these resource scheduling algorithms.

Chapter 1

INTRODUCTION

1.1 Introduction

“Cloud Computing” is a new technology used by industry and in many areas for storing , retrieving and accessing the files and necessary documents where distributed computing is the basis. “Cloud computing” allow the users to systematically and dynamically provision computing resource to meet the IT needs. “Cloud Provider” offers two plans to the customer which are reservation and on-demand. The reservation plan is typically cheaper than on-demand plan. If the actual computing demand is well known in advance reserving the resource would be undemanding.

“Cloud computing” is a technology that uses the internet and central remote servers to maintain data and applications. “Cloud computing” allows consumers to use applications without installation and access their personal files and folders at any computer with internet access a must. It allows for much more productive computing by centralizing storage, memory, processing and bandwidth. According to NIST states “cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. One advantage of the “cloud” paradigm is computation externalisation, where their resource-constraint devices no longer limit the computational power of the cloud customers.

“Cloud computing” has become a crucial part of many developed and also the developing firms/organizations. The major qualities of the cloud comes from its ability to provide flexible and on the run resources mainly in terms of services. “Cloud computing” is claimed the title of specialized computing due to the presence of a unique feature of the cloud computing which is called utility computing. The effectiveness of “cloud computing” environment greatly relies on its ability to not only maintain the resources but also on hyperefficiently scheduling those resources. Resource scheduling inherently contributes to the strength of the “cloud computing”. Just like all the other scheduling algorithms, resource scheduling algorithms mainly focuses on the optimal usage of the available resources. In addition to that the main focus of these algorithms is to schedule the resources in such a manner that satisfies the cloud user’s to a greater possible extent. “Cloud computing” resources stretch over a vast range including infrastructure, software, storage, security and data, etc which are delivered to the user in the form of pay per-use services. Adequate resource scheduling helps the cloud provider and also benefits each person/group associated with the “cloud”. Efficient resource scheduling helps the “cloud provider” in terms of cost effective resource management which in turn provides more resources to allocate without declining any user requests. On the other hand, cloud users bring an advantage in terms of their monetary gains at each front. Optimizing resource scheduling techniques helps upraise these gains in an adequate manner. Some of the influential resource scheduling algorithms comprise of the “Ant Colony algorithm”, Particle swarm technique, genetic algorithm etc. These algorithms are fundamentally optimization algorithms and they are being implemented in the “cloud computing” environment to help optimization of resource scheduling. These algorithms are now being used for quite some time for scheduling of resources, however over the span of time each one of them have been modified in order to make them more and more efficient and inflexible towards their goal. In this report, we present a review of various “optimization” algorithms used for scheduling of resources in cloud and also their evolution over the period of time.

1.2 Problem Statement

Resource allocation “optimization” of algorithms is a typical cloud project scheduling problem that limits a cloud system’s ability to execute and deliver a project as that was earlier planned. The entropy which is a measure of the degree of the disorder in a system, is an indicator of a system’s tendency to progress out of order and into a troublesome condition, and it can hence serve to measure a cloud system's reliability for project scheduling. In this report,the cellular automaton is being used for modeling the complex "cloud" project scheduling system.

The main issues in "Cloud Computing" lies in scheduling the incoming request in a fruitful way with minimum response time and the resources should not be underutilized. The Algorithms such as Round Robin, FCFS and Throttled are good in serving the client request with minimal response time. But the problem of high communication delays overutilization and underutilization of resources are not fulfilled in accessing the client request which leads to the disparity of "cloud" system.

The experience that anything that can go wrong will go wrong and at the worst possible moment is summarized unofficially as "Murphy's Law" . Scheduling systems are not protected to Murphy. In cloud project scheduling system, after an enough power strikes one of the resources, which leads to its yield reduced or collapsed, the whole system gets collapsed. In this scenario, such resource productivity collapse may cause by hardware/software failures, , resources over- or under-provisioning, resources CPU overload or application misbehaviours.

1.3 Objectives

OPTIMIZATION ALGORITHMS

One of the fundamental principles for every object in the universe is to stand in the best possible "optimal" state. "In almost all computing and non-computing fields and also in mathematics, statistics, empirical sciences, or even management science, in nearly every field 'optimization' plays the most essential role. Optimization can briefly be defined as, the selection of best element (pertaining to specified criteria) from an available set of variable alternatives". Optimization can be achieved in various forms and by applying numerous methods; however the goal of each one is to achieve – "maximal output with minimal input". "Optimization" can be classified into wide range of categories based on the type of output desired. It can be a) the need to arrive on a feasible point b) to detect the existence of a particular problem/situation c) to explore the necessary and sufficient conditions etc. However the ‘cloud’ computing environment requires optimization algorithms to optimize the resource usage by providing convenient/optimal schedule. Optimization is widely categorized into 3 broad fields, deterministic, probabilistic and heuristic algorithms. In this paper our primary concern is on the probabilistic algorithms. In ‘cloud’ environment probabilistic algorithms are employed for efficient resource usage. The probabilistic algorithm family involves an especial family of algorithms named “Monte-Carlo algorithms”. Out of the various algorithms in ‘Monte-Carlo’ the most prominent algorithms used in cloud are Ant Colony, Particle Swarm and Genetic algorithms. Though there had been other scheduling algorithms, however they rendered themselves unfruitful in the due course of time. We will discuss the most widely implemented algorithms in the following section.

1.4 Methodology

ACTIVE VM LOAD BALANCER ALGORITHM: The proposed work least ”VM assign algorithm” is compared with Active VM load balancer algorithm. The main aim is to distribute the load to the available “VM” efficiently so that the resources are not over or under utilized.

ANT COLONY LOAD BALANCING ALGORITHM: Individual ants are behaviorally much unsophisticated insects. Acting as a collective however, ants manage to perform a variety of complicated tasks with great reliability and consistency. “The ant colony optimization algorithm is a probabilistic technique for solving computational problems which can be reduced to finding best available paths through graphs.”

GENETIC ALGORITHM:

In the field of ”Artificial Intelligence”, the algorithm working of search heuristics and mimics the natural process of evolution is profoundly termed as Genetic Algorithm. As the name genetic algorithm works widely in a similar fashion to that of natural evolution.

PARTICLE SWARM ALGORITHM:

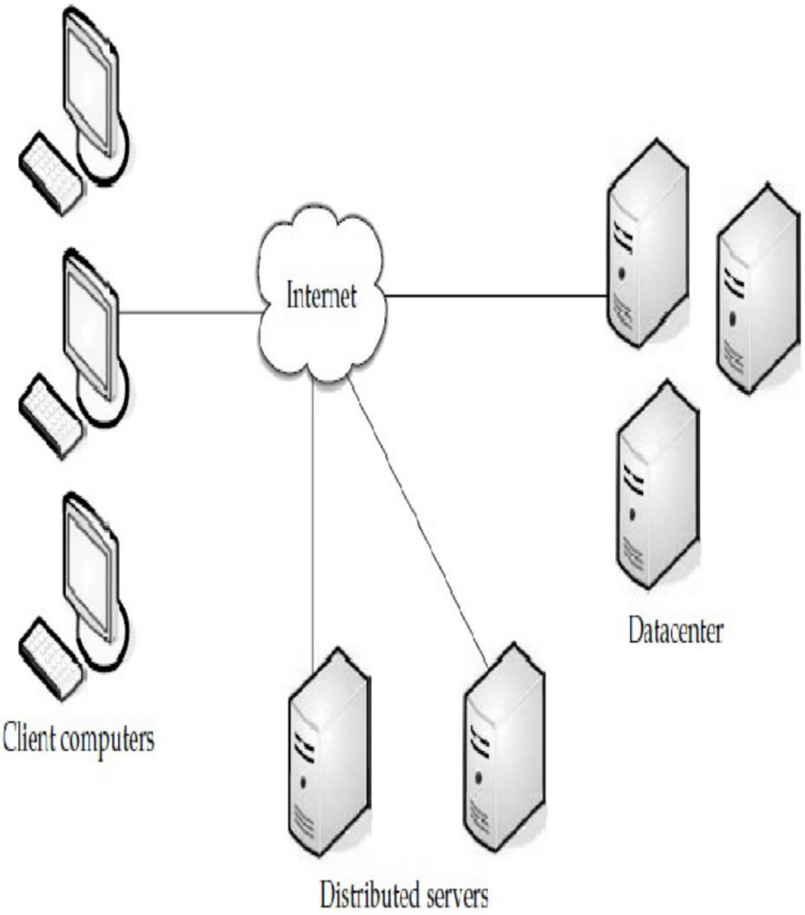
is a computational method which belongs to the super set of “swarm” intelligence. In this algorithm iterative procedure is used to “optimize” a problem. The candidate solutions are iteratively processed with respect to a given measure of quality. In PSO, population of candidate solutions is referred to as ‘particles’. The optimization in “PSO” involves movement of these particles around in the search-space. The movement is guided by simple mathematical formulae over the particle's velocity and position. Each particle’s local best known position influences its movement.

Chapter 2

LITERATURE SURVEY

2.1 Cloud Components :

A Cloud system consists of 3 major components such as clients, datacentre, and distributed servers. Each element has a definite purpose and plays a specific role.



Clients : End users interact with the clients to manage information related to the cloud. Clients generally fall into three categories:

Mobile : Windows Mobile Smartphone, smartphones, like a Blackberry, or an iPhone.

Thin : They don’t do any computation work. They only display the information. Servers do all the works for them. Thin clients don’t have any internal memory.

Thick : These use different browsers like IE or Mozilla Firefox or Google Chrome to connect to the Internet cloud.

Now-a-days thin clients are more popular as compared to other clients because of their low price, security, low consumption of power, less noise, easily replaceable and repairable etc.

Datacentre : Datacentre is nothing but a collection of servers hosting different applications. A end user connects to the datacentre to subscribe different applications. A datacentre may exist at a large distance from the clients.

Now-a-days a concept called virtualisation is used to install software that allows multiple instances of virtual server applications.

Distributed Servers : Distributed servers are the parts of a cloud which are present throughout the Internet hosting different applications. But while using the application from the cloud, the user will feel that he is using this application from its own machine.

2.2 Load balancing in cloud computing : Load Balancing is a method to distribute workload across one or more servers, network interfaces, hard drives, or other computing resources. Typical datacenter implementations rely on large, powerful (and expensive) computing hardware and network infrastructure, which are subject to the usual risks associated with any physical device, including hardware failure, power and/or network interruptions, and resource limitations in times of high demand.

Load balancing in the cloud differs from classical thinking on load-balancing architecture and implementation by using commodity servers to perform the load balancing. This provides for new opportunities and economies-of-scale, as well as presenting its own unique set of challenges.

Load balancing is used to make sure that none of your existing resources are idle while others are being utilized. To balance load distribution, you can migrate the load from the source nodes (which have surplus workload) to the comparatively lightly loaded destination nodes.

When you apply load balancing during runtime, it is called dynamic load balancing — this can be realized both in a direct or iterative manner according to the execution node selection:

- In the iterative methods, the final destination node is determined through several iteration steps.

- In the direct methods, the final destination node is selected in one step.

A another kind of Load Balancing method can be used i.e. the Randomized Hydrodynamic Load Balancing method , a hybrid method that takes advantage of both direct and iterative methods.

Goals of Load balancing :

The goals of load balancing are :

1. To improve the performance substantially.

2. To have a backup plan in case the system fails even partially.

3. To maintain the system stability.

4. To accommodate future modification in the system.

Types of Load balancing algorithms :

Depending on who initiated the process, load balancing algorithms can be of three categorie s as given in:

Sender Initiated : If the load balancing algorithm is initialized by the sender.

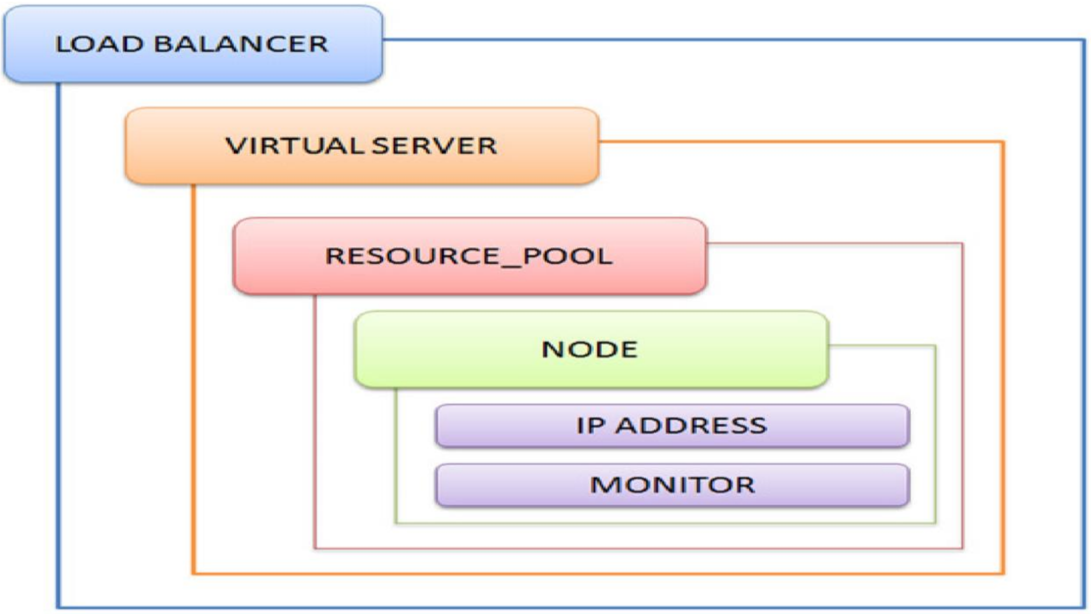
Receiver Initiated : If the load balancing algorithm is initiated by the receiver.

Symmetric : It is the combination of both sender initiated and receiver initiated.

Depending on the current state of the system, load balancing algorithms can be divided into 2 categories as given in :

Static : It does not depend on the current state of the system. Prior knowledge of the system i s needed.

Dynamic : Decisions on load balancing are based on current state of the system. No prior kn owledge is needed. So it is better than static approach. Here we will discuss on various dyna mic loadbalancing algorithms for the clouds of different sizes.



Dynamic Load balancing algorithm :

In a distributed system, dynamic load balancing can be done in two different ways:

Distributed and non-distributed.

In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. The interaction among nodes to achieve load balancing can take two forms: cooperative and noncooperative . In the first one, the nodes work side-by-side to achieve a common objective, for example, to improve the overall response time, etc. In the second form, each node works independently toward a goal local to it, for example, to improve the response time of a local task. Dynamic load balancing algorithms of distributed nature, usually generate more messages than the non-distributed ones because, each of the nodes in the system needs to interact with every other node. A benefit, of this is that even if one or more nodes in the system fail, it will not cause the total load balancing process to halt, it instead would affect the system performance to some extent. Distributed dynamic load balancing can introduce immense stress on a system in which each node needs to interchange status information with every other node in the system. It is more advantageous when mostof the nodes act individually with very few interactions with others.

In non-istributed type, either one node or a group of nodes do the task of load balancing. Non distributed dynamic load balancing algorithms can take two forms: centralized and semi distributed. In the first form, the load balancing algorithm is executed only by a single node in the whole system: the central node. This node is solely responsible for load balancing of the whole system. The other nodes interact only with the central node. In semi-distributed form, nodes of the system are partitioned into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in each cluster by appropriate election technique which takes care of load balancing within that cluster.

Hence, the load balancing of the whole system is done via the central nodes of each cluster.

Centralized dynamic load balancing takes fewer messages to reach a decision, as the number of overall interactions in the system decreases drastically as compared to the semi distributed case. However, centralized algorithms can cause a bottleneck in the system at the central node and also the load balancing process is rendered useless once the central node crashes. Therefore, this algorithm is most suited for networks with small size.

Policies or Strategies in dynamic load balancing

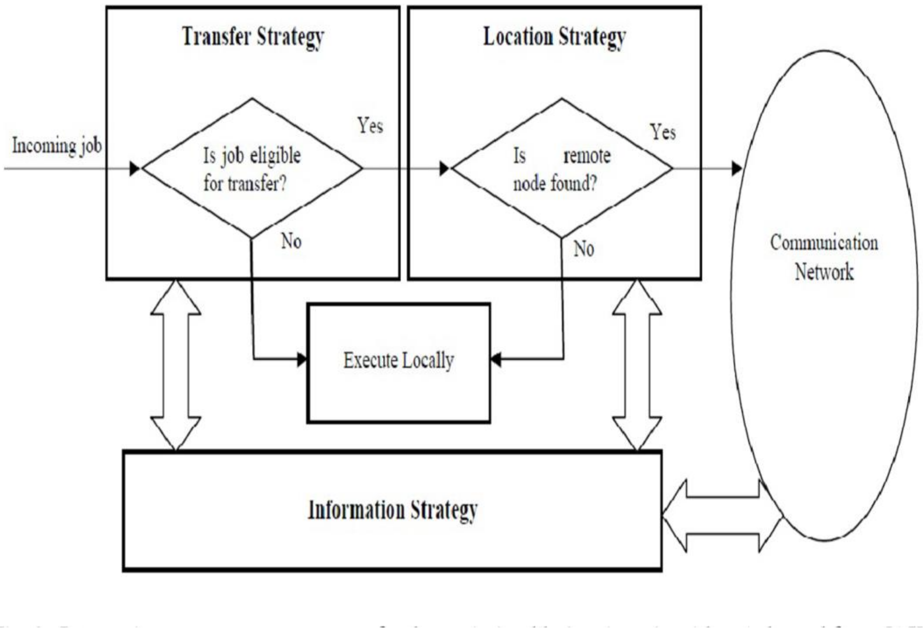
There are 4 policies :

Transfer Policy : The part of the dynamic load balancing algorithm which selects a job for transferring from a local node to a remote node is referred to as Transfer policy or Transfer strategy.

Selection Policy : It specifies the processors involved in the load exchange (processormatching)

Location Policy : The part of the load balancing algorithm which selects a destinationnode for a transferred task is referred to as location policy or Location strategy.

Information Policy : The part of the dynamic load balancing algorithm responsible for collecting information about the nodes in the system is referred to as Informationpolicy or Information strategy.



Interaction among components of dynamic load balancing algorithm

The computational grid is a promising platform that provides large resources for distributed algorithmic processing . Such platforms are much more cost-effective than traditional high-performance computing systems. However, computational grids have different constraints and requirements than those of traditional high-performance computing systems, such as heterogeneous computing resources and consider able communication delays. To fully exploit such grid systems, resource management and s cheduling are key grid services, where issues of task allocation and load balancing represent a common problem for most grid systems. The load-balancing mechanism aims to equally spread the load on each computing node, maximizing their utilization and minimizing the average task execution time.

In general, load-balancing algorithms can be classified as centralized or decentralized and static or dynamic. In the centralized approach , one node in the system acts as a scheduler and makes all the lo ad-balancing decisions. Information is sent from the other nodes to this node. In the decentraliz ed approach , all nodes in the system are involved in the load-balancing decisions. It is therefore very costly for each node to obtain and maintain the dyna mic state information of the whole system. Most decentralized approaches have each node obtaining and maintaining only partial information locally to make suboptimal decisions.

Static load-balancing algorithms assume that all information governing load-balancing decisions that can include the characteristics of the jobs, the computing nodes, and the communication network are known in advance. Load-balancing decisions are made deterministically or probabilistically at compile time and remain constant during runtime. The static algorithms have one major disadvantage—it assumes that the characteristics of the computing resources and communication network are all known in advance and remain constant. Such an assumption may not apply to a grid environment. In contrast, dynamic load balancing algorithms attempt to use the runtime state information to make more informative load-balancing decisions. Undoubtedly, the static approach is easier to implement and has minimal runtime overhead. However, dynamic approaches may result in better performance. One of the major drawbacks of the dynamic algorithms is their sensitivity to inaccuracies in performance prediction information that the algorithm uses for load-balancing purposes. Some dynamic load-balancing algorithms are more sensitive to the inaccuracies and can generate extremely poor results even when the information accuracy is only slightly less than 100 percent; in real grid environments, however, 100 percent accuracy in information is very hard to achieve and maintain.

The so-called hybrid scheduling is another area that has been receiving some attention. In terms of static and dynamic load balancing, a hybrid load balancer attempts to combine the merits of static and dynamic load-balancing algorithms and, by doing so, minimizes their relative inherent disadvantages. Note, however, that the definition between a static and a dynamic load-balancing algorithm in itself is not clear cut, and different authors use slightly different definitions of static and dynamic algorithms. Also, a hybrid algorithm for adaptive load sharing in distributed systems was studied. Few works have been done using game-theoretic approaches and models for load balancing in a grid environment. A study on load balancing in distributed systems, formulating them as a non cooperative game with Wardrop equilibrium as the objective, was discussed before. More recently, a noncooperative load-balancing game for distributed systems was also presented before; using the assumption of exponential service times and Poisson arrival, an algorithm for computing the Nash equilibr ium was derived. However, none of these papers takes into account the communication dela ys in a grid environment that may affect the completion time of tasks.

2.3)GENETIC ALGORITHM

In the field of "Artificial Intelligence", the algorithm working of search heuristics and mimics the natural process of evolution is profoundly termed as Genetic Algorithm. As the name genetic algorithm works widely in a similar fashion to that of natural evolution. Using continual search heuristics, the initial solution is modified over time in order to gain the closest "optimal" solution as desired of problem. The process goes as; initially to begin with we have a population of strings. These are also referred as the "genome" or even genotype in order to provide alias with the algorithm. These populations of strings or (commonly known as "chromosomes") are used to encode the candidate solutions for the problem. The candidate solutions are also termed as "phenotypes" on individual creatures. The candidate solutions encoded in the population string is evolved towards a more desired solution. Usually the population is encoded in a "binary" form using 0 and 1, however other types of encoding are also possible. The evolution begins with population of solutions that are in random generated and goes on towards a better generation. Whenever a new generation is obtained, its "fitness" is checked for each individual in it. After this multiple individuals are selected which are most suitable (in terms of fitness) in the current population which are then modified. "Modification" is done by combining (mutation) various individuals to generate a new population of modified individual. This "generation" of new population is again subjected to the same process, iteratively. After attaining a satisfactory level the algorithm terminates. However if the "algorithm" terminates because of large no. of populations, then the satisfactory level of population might not be achieved. Thus, the entire working of

Genetic algorithm can be summarized into 5 major steps:-

i)Initialisation

ii) Selection

iii) Cross-Over

(iv) Mutation

(v) Termination

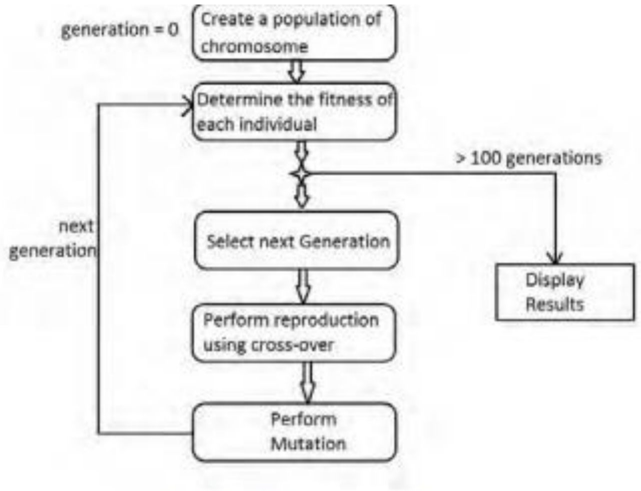

generation = 0
Create a population of chromosome
Determine the fitness of each individual
> 100 generations
next generation
Select next Generation
Display Results
Perform reproduction using cross-over
Perform Mutation

Fig. 5: Flowchart of Genetic Algorithm

“Cloud computing”, which stands for providing resources to all users in the form of service, must have a fast and unique methods to understand the needs of the user and allocate them the resources as soon as possible. “Genetic algorithm” being the most effective in evolution seems to fit perfect for allocating the resources to users and also optimize the resource usage with appropriate scheduling.

OPTIMIZED GENETIC ALGORITHM

“Genetic algorithm” has been an area of wide interest for many researchers. Its evolutionary concept has been studied and being modified to suit the need of the user as well as the resource providers. Some of the most prominent modifications made in genetic algorithm are as follows:- (i) “Robust Genetic Algorithm” for resource scheduling In this method the encoding stage is further explored to consist of the following: Activity-List representation, Priority rule representation, Random Key representation, Shift Key representation. These perform in the effective way to reach an optimal solution . (ii) “Genetic Algorithm”using DSS Using DSS (Decision-Support-System) it was found that it can provide optimal solutions or atleast near-optimal solutions in most of the problems .

(iii) Modified Genetic Algorithm

“Matthew Bartschi Wall” proposed in his work that using modified Genetic algorithm not only provides flexibility but also makes the process more robust. This method used a dual array for encoding purpose which unlike the traditional method is direct and time-based representation.

“Breeder Genetic Algorithm” The selection phase in this method is inspired by the selection used in human breeding. Its model is derived from quantitative genetics. It compares behavior of different mutation schemes, which gives the progress to the desired optimal solution.

“Fuzzy Genetic Algorithm” The central idea of this algorithm is to give a choice to the core/central scheduler. This choice is in reference of the global view of the entire system .

“ Improved Genetic Algorithm” This method is guided by the sole principle of maximum utilization of resources by effective allocation of more and more VM’s. It works by using the idea of shortest genes. It also introduced the concept of Dividend Policy in order to choose an optimal or nearly optimal allocation for VMs.

Optimization of Cloud Resource Service Adaptability using Genetic Algorithm:

“Cloud computing” started a new era in getting variety of information puddle through internet connections by any connective devices. It provides pay and use method for grasping the services by the clients. “Data center” is a sophisticated high definition server, which runs applications virtually in cloud computing. It moves the application, services and data to a centralized large Data centers. Data center provides more service level, which covers maximum of users. So, to find resource service adaptability of a client is a set aside task. This proposed work to find the Cloud Resource Service Adaptability (“CRSA”) of client in cloud computing. In particular, we consider for the most three applications called Community Commercial and Collaboration on behalf of cloud client to verify the adaptation of dynamic services in “Cloud Computing”. The goal is to find adaptation of services in terms of big three factors Bandwidth, Memory and “Central Processing Unit” (CPU) cycle. This work proposes a genetic algorithm for optimizing

Resource Service Adaptability ("RSA") approach for executing cloud applications. In particular, we consider for the most three applications called Business, Collaboration and Productivity applications to optimize adaptation of services.

CLOUD computing is an evolving paradigm to access assortment of data pool via internet using connective devices such as "Personal Digital Assistant" ("PDA"), work station and mobile devices. It is a utility based computing, which has the capability to deliver services over the internet. It provided on - demand access without the need of any human intervention. "Data center is deployed as an individual server room which runs several applications on a single server". "Cloud computing" services were pooled to supply multiple tunings models with many virtual resources. It refers to hold the applications delivered as services over the internet. The hardware and system software in the data center provide those services. The standard deployment object that is used in cloud computing is Virtual Machines ("VM"). "Virtualization" has enabled of many applications to run on a single server or couple of servers. It enhances flexibility and helps to enable a datacenter as dynamic one. It has the ability to hire a server or many servers that can be run in a geophysical modeling application in anywhere. Some important factors of utilization are storage, power, cooling, capacity, response time, and adaptability. It is managing tasks and applications by altering the software, platform, infrastructure and organizing third party data centers known as Cloud Service Providers ("CSP") such that Yahoo!, Amazon, Google and VMware . Cloud computing is able to delivered the services on behalf of resources like Bandwidth, Disk, Random Access Memory "(RAM"), Processor, Floating point systems. It is sharing of resources with enormous services. The "resource" utilization is depends upon the services to be served, it is likely a core cloud computing services: Software as a services (Saas) , Platform as a services(Paas), Infrastructure as a services (Iaas). Each service is considered with their own resources capability. The major difficulty is "digressing" of services along with their resources. How to often those resources? Check the resource adaptation of a particular service. R. B

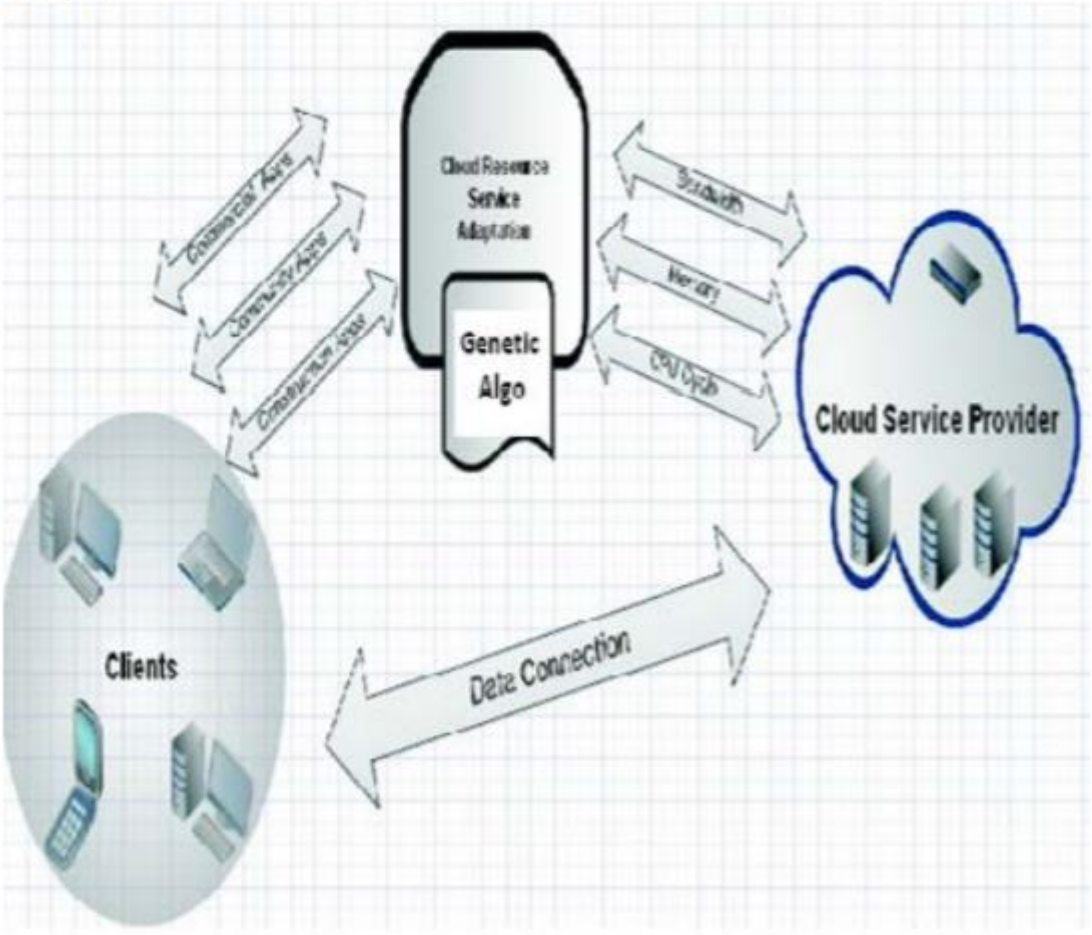
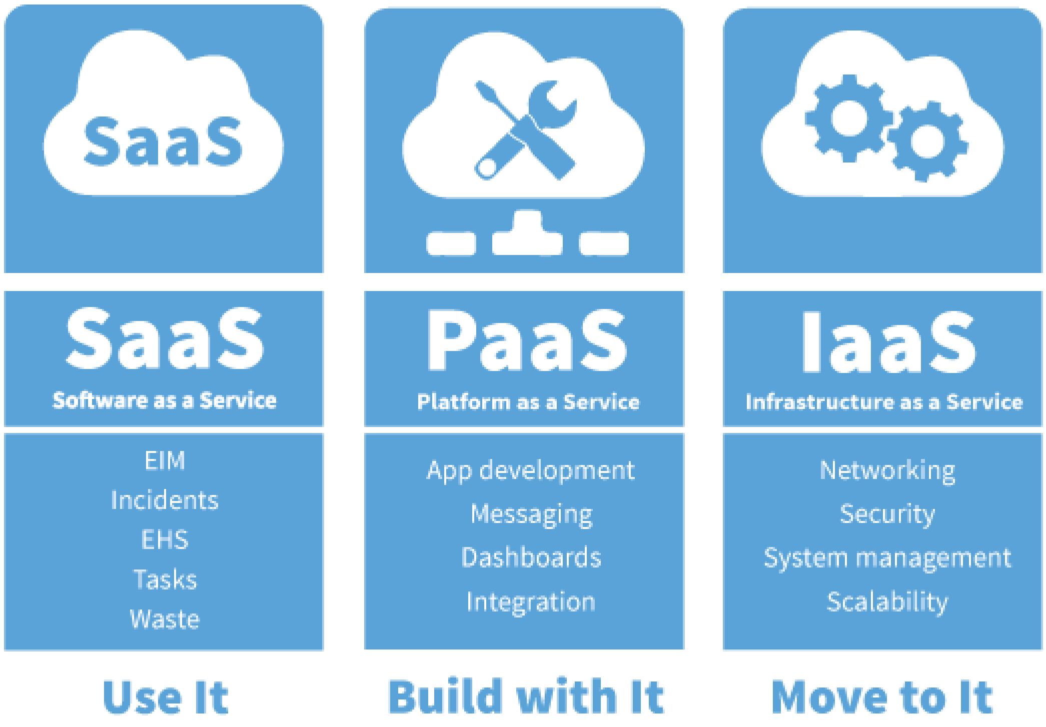


Fig. 1 System architecture



2.5) ANT COLONY ALGORITHM

“The ant colony optimization algorithm is a probabilistic technique for solving computational problems which can be reduced to finding best available paths through graphs.”. Ant Colony Algorithm is essentially a category of swarm intelligence. [5]. “Ant Colony algorithm” is based on the characteristic feature of an ant. Ants, in spite of their absence of eyes, could easily find their food and also trace back there path to their homes. Ants use ‘pheromone deposition’ in order to accomplish these tasks. Initially they deposit pheromones on the ground to mark the path used by them to discover the way to the “food”. Secondly, they deposit the “pheromone” indicating the optimized route between the food source and the home

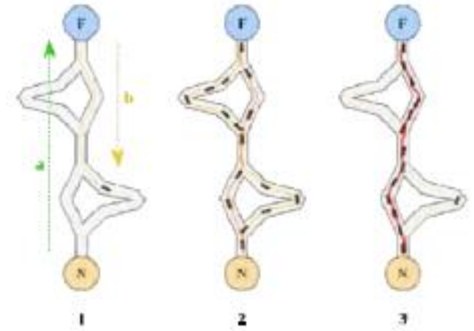


Fig. 1. Bridge experiment depicting the pheromone principle for Ant Colony Algorithm

Let us discuss where this "algorithm" does finds its application and how does it work. Consider a fully functional cloud computing environment; A host/cloud provider with various resources and virtualised distribution of resources on different Virtual Machine's ("VM's"). On the other hand we have many cloud users connected to the cloud provider on the pay-as-you-go basis, as shown in figure. The end user can be any single person, group of individuals, a company, an organisation, of other giant "firms". Now each cloud user submits its job to the cloud provider specifying "SLA" (Service Level Objectives) which includes resource requirements, time to accomplish the task, etc and other relevant details. The user pays to the "cloud provider" while he is accessing the resources allocated to him by the cloud provider. Similarly, all the cloud users submit their tasks to the cloud provider specifying all the essential details. The cloud provider starts by gathering all the requests (in a particular time stamp) and then allocation is done for suitable resources to each of them. At this point, "Ant Colony algorithm" plays a helpful role. The cloud provider needs to allocate jobs to various "Virtual Machines". Job scheduling is of vital importance in this scenario. Here all the virtual machines are analogous to the nodes, ants are analogous to the mobile agent, and resources are similar to the food source in case of "ant colony". Ant traverses amongst the nodes to allocate the jobs to various resources in an optimized manner using "Ant Colony Algorithm".

OPTIMIZED ANT COLONY ALGORITHM

Ant colony algorithm though being an efficient way for resource scheduling does suffer from some of inherent drawbacks like the table updating strategy may sound perfect however in case of "asymmetric" network of nodes it may fail to deliver best results in cloud environment. Also in the case where more VM’s are involved the ant colony algorithm would develop colonies, so an ant is less likely to follow a “pheromone” trail from another colony.

In order to avoid such short comings there were several reforms were made to the traditional Ant colony algorithm to achieve an optimized version of “Ant Colony algorithm”. Over the period of time there were several changes made to the ant colony algorithm.

The major ones are listed below:Creating gateways for each set of VM’s; each group of mobile agents corresponds to a colony of ants, and the routing table of each group corresponds to a pheromone table of each “colony”. In this way the ant/mobile agent in spite of having its own routing preferences will take into consideration the one encountered.

“The mobile agents/ants will automatically update the information from local knowledge table to Global knowledge table.”

“VM” is also taken into consideration.

“Classifying the pheromone for the forward and backward movements. As in backward movement; when an ant encounters an overloaded node after it has encountered an overloaded node then it travels in the backward direction leaving a trailing pheromone. And vice versa in case of forward movement leaving a foraging pheromone.”

2.5) PARTICLE SWARM OPTIMIZATION

It is a computational method which belongs to the super set of “swarm intelligence”. In this algorithm iterative procedure is used to optimize a problem. The candidate solutions are iteratively processed with respect to a given measure of quality. In “PSO”, population of candidate solutions is referred to as ‘particles’. The optimization in “PSO” involves movement of these particles around in the search-space. The movement is guided by simple mathematical formulae over the particle's velocity and position. Each particle’s local best known position influences its movement. In addition to this it is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. The swarm is supposed to converge towards the optimal solution by these particle movements. Like Ant colony algorithm, Particle swarm algorithm is also a Metaheuristic algorithm. A simple form of algorithm works by starting a population of the candidate solutions (the Swarm). Each candidate solution is analogous to a particle in the system. The system is supposed to be the solution space. These particles are then moved around in the search-space on basis of some mathematical formulae with the aim to converge to an optimum solution.

Consider a simple cloud computing environment where various users are submitting their jobs with request for specific resources to the cloud provider. Now at any time instant the cloud provider need to schedule the jobs to the available resources in the best possible manner. In this scenario, Particle Swarm Algorithm serves to be of great use. Each mobile agent in the cloud environment could serve as a potential candidate or so called particle. [23]The particle moves with a specific velocity with defined magnitude and in defined direction. This movement is associated with the best solution attained so far. Also a fitness value is used to indicate the performance of the particle. It keeps a track of each position at each time instant, as shown in Fig. 3 and Fig. 4. The movement is guided towards attaining a better solution by varying the acceleration at each time step. Ultimately when the particle covers the entire search space making them its topological neighbor the final best solution is generated, as shown.

ution is generated, as shown in Fig. 4 [23].

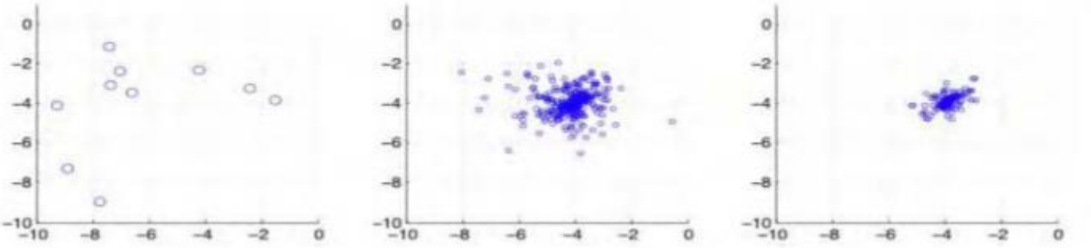


Fig. 1. The movement of particles to attain the best solution

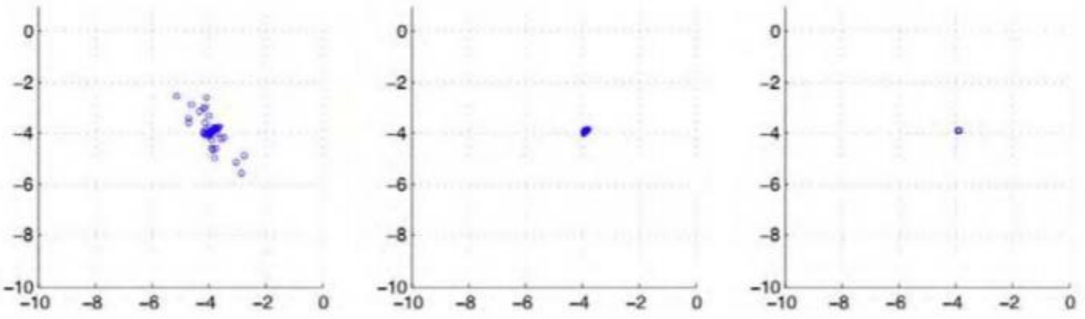


Fig. 2. Attaining the best solution after varying the acceleration at each time step

VI. OPTIMIZED PARTICLE SWARM ALGORITHM

OPTIMIZED PARTICLE SWARM ALGORITHM

As said Particle Swarm Algorithm is a metaheuristic approach, thus it needs to make some certain assumptions to attain the best possible solution. Also the solution space or the search space can end up being very large and complicated. Hueristic design aproach such as PS may not even be able to guarantee an optimal solution is ever found. In terms of mathematical form also Particle swarm algorithm doesn’t considers the inclination of the problem which is being optimized which means that problem may not be differentiable. However over the years of research done in the field of PSA some elementary modifications are made to the algorithm, to make it more effective and influential. Some of them are listed below:-

(i) “Dynamic Hierarchical PSO” In this the next population, after the initial start is allowed to explore the patent of the earlier population. This gives an advantage because it is no longer required to predetermine the no. of clusters to be found in the solution space.

(ii) “Revised Discrete PSO” It emphasises on calculating more accurately the position and velocity of the particle. This is done by defining some operation rules and also defining their equation of motion in accordance with few discrete variables.

(iii) “Modified PSO” In this method the rate at which the position of the particle changes is given by a modified velocity equation an mentioned below

$$V_{id} = V_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * Rand() * (p_{gd} - x_{id})$$

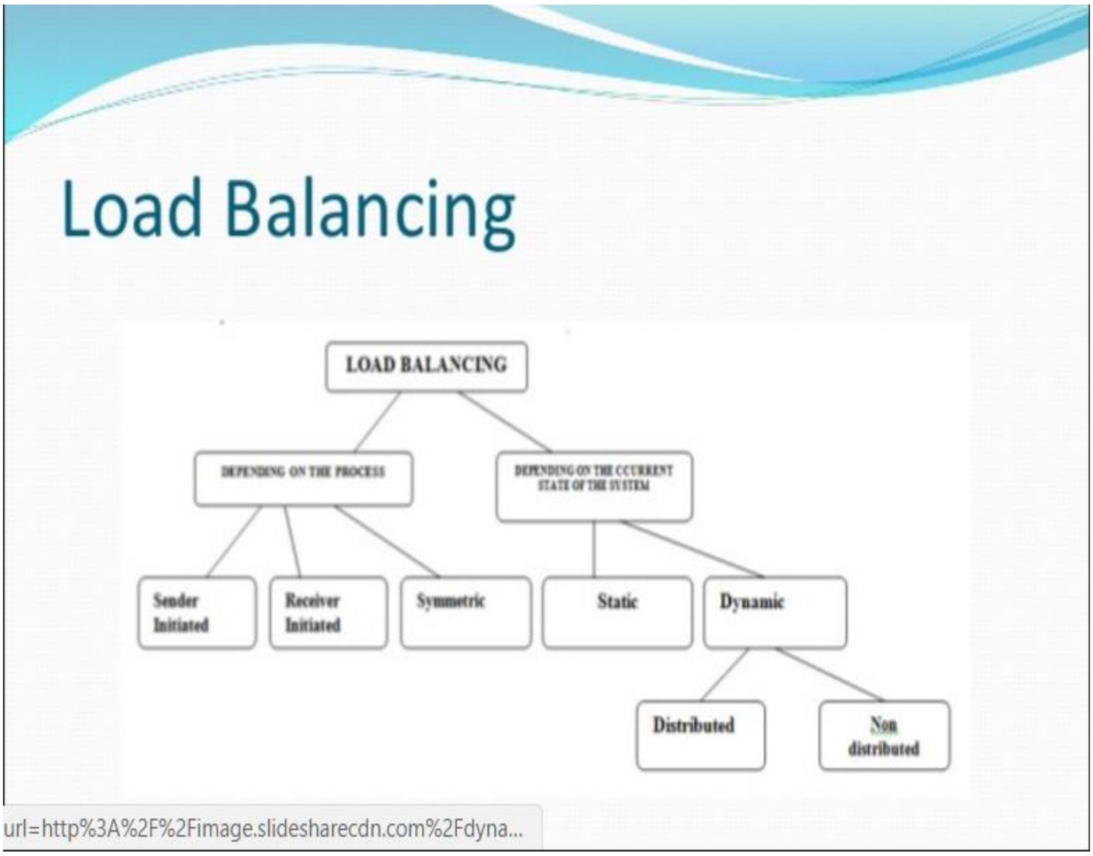
$$\text{Where} \qquad x_{id} = x_{id} + v_{id}$$

2.6) Optimization Of Resources in Cloud Computing Using Effective Load Balancing Algorithms

Load balancing is essential for optimization of resources in distributed environments. The major goal of the cloud computing service providers is to use cloud computing resources efficiently to enhance the overall performance and improve efficiency. Load balancing in cloud computing environment is a methodology to distribute workload across multiple computers to achieve optimal resource utilization with minimum response time and improved efficiency. The proposed system pave the way for the green computing by allocating the virtual machine based on the load its processing for the “optimization” of number of servers in use. The performance of the algorithm is analyzed using Cloudsim simulator .The simulation result ensures that all the processors in the system as well as in the network does coherently equal amount of work at any instant of time by comparing with other algorithms by its burlesque abilities.

Cloud Computing is a new technology used by industry and in many areas for storing, retrieving and accessing the files and necessary documents where distributed computing forms the basis. The main issues in Cloud Computing lies in scheduling the incoming request in an efficient way with minimum response time and also the resources should not be underutilized. Algorithms like “Round Robin”, FCFS and Throttled are good in serving the client request with minimum response time. But the problem of high communication delays and underutilization of resources are not fulfilled in executing the client request which leads to imbalance of cloud system. “Load balancing” is very much essential for increasing the throughput and minimizing the response time. Therefore every virtual machine in cloud system should do the same amount of work throughout in processing the request. The “load balancing” can be done by dynamically forwarding the incoming client request to remote nodes or machines which are less utilized. This load balancing improves by maximizing the user satisfaction, minimizing response time, increasing resource utilization, reducing number of job rejections and thus overall performance of the system is enhanced. Dynamic resource management can be efficiently done in cloud system by virtualization technology. Therefore power efficiency can be improved by assigning multiple virtual machines to a single physical server. Consequently power consumption can be lowered by turning off some of the servers or putting them in sleep mode. In this paper, we present a novel “VM” assign algorithm which allocates incoming client request to available virtual machines depending on the load i.e. VM with least work load is found and then new request is allocated.

In this area stack adjusting calculations utilized as a part of distributed computing condition are quickly condensed. The “Active load adjusting” calculation keeps up all the present data about the virtual machines and number of solicitations as of now allotted. At first when a demand arrives, it chooses the VM in view of which machine is minimum stacked and returns id of the “VM” to the server farm controller. At the point when there are more than one VM is found, the principal distinguished is returned. Server farm controller advises the Active VM stack balancer of the new distribution. The distribution utilizing Throttled calculation is totally in view of the virtual machine. The customer initially sends adjuration to the heap balancer to check for the accessibility of VM which is equipped for dealing with and preparing the demand. The “Active Monitoring Load” Balancer keeps up data about each VM's and the quantity of demand as of now distributed to which VM when a demand is dispense another VM arrives. In the event that there are more than one VM, the principal distinguished is chosen AMLB gives back the “VM” id to the server farms controller. The server farms controller send the demand to the VM recognized by that id. The server farm controller advises the AMLB to new distribution and cloudlets is sent to it.



Load Balancing

url=http%3A%2F%2Fimage.slidesharecdn.com%2Fdyna...

Optimization of a Cloud Resource Management Problem from a Consumer Perspective

Distributed computing is a conveyed registering worldview in which processing assets are accessible to clients by means of Internet. In spite of the fact that there are many takes a shot at asset administration in related writing, few of them handle the issue from the point of view of business cloud buyers. In this paper, the proposed asset administration issue chooses cloud assets expecting to lessen the installment cost and the execution time of client applications. With a specific end goal to take care of this issue, a whole number programming plan and a heuristic in view of Greedy Randomized Adaptive Search Procedure (GRASP) are additionally presented. The model and the calculation were tried over an arrangement of examples developed from prerequisites of genuine applications consolidated with sets of assets offered by business mists. The acquired outcomes demonstrate that the exhibited strategies can be an imperative choice bolster instrument for cloud buyers.

2.7) Optimizing the Cost for Resource Subscription Policy in IaaS Cloud

Distributed computing is an innovation that uses the web and focal remote servers to keep up information and applications. Distributed computing enables buyers and business to utilize applications without establishment and get to their own documents at any PC with web get to. It takes into account a great deal more proficient registering by bringing together capacity, memory, preparing and data transmission. As indicated by NIST states "distributed computing is a model for empowering helpful, on-request organize access to a mutual pool of configurable figuring assets (e.g., systems, servers, stockpiling, applications, and administrations) that can be quickly provisioned and discharged with insignificant administration exertion or specialist organization collaboration". One essential preferred standpoint of the cloud worldview is calculation outsourcing, where their asset imperative gadgets no longer farthest point the computational energy of cloud clients. By outsourcing the workloads into the cloud, clients could appreciate the truly boundless processing assets in a compensation for every utilization way without submitting any vast capital expenses in the buy of both equipment and

programming as well as the operational overhead in that. Every supplier serves a particular capacity, giving clients pretty much control over their cloud contingent upon the sort. At the point when customers pick a supplier, they contrast their requirements with the cloud administrations accessible. Cloud purchaser needs will differ contingent upon how they mean to utilize the space and assets related with the cloud. Remember that cloud supplier will be pay-as-you-go, implying that if mechanical requirements change anytime buyer can buy more storage room (or less besides) from cloud supplier. The distributed computing model is included a front end and a back end. These two components are associated through a system. The front end is the vehicle by which the client cooperates with the framework and the back end is simply the cloud. The front end is made out of a customer PC, or the PC system of an undertaking, and the applications used to get to the cloud. The back end gives the applications, PCs, servers, and information stockpiling that makes the billow of administrations. Distributed computing depicts a sort of outsourcing of PC administrations, like the route in which the supply of power is outsourced. Clients can essentially utilize it. They don't have to stress where the power is from, how it is created, or transported.

Trust Based Resource Selection in Cloud Computing Using Hybrid Algorithm

Distributed computing is encountering fast progression in the scholarly community and industry. This innovation offers circulated, virtualized and versatile assets as utilities for end clients and can bolster full acknowledgment of "figuring as an utility" later on. Booking conveys assets among gatherings which all the while and nonconcurrently look for it.

Planning calculations are implied for booking and they diminish asset starvation guaranteeing decency among those utilizing the assets. Most Task-planning distributed computing methodology consider assignment, asset prerequisites for CPU and memory, and not data transfer capacity. This review recommends upgrading planning with BAT-Harmony look half and half calculation

To guarantee that cloud administrations are successful as supplier framework, undertaking booking calculation is a noteworthy prerequisite. They are in charge of guide employments presented a cloud domain to assets so that aggregate reaction time in makespan, is decreased [3]. Errand booking is dispensing at least one time interims to at least one assets. Booking is an issue of planning submitted errands set for various clients of figuring assets to lessen a particular occupation fruition time or framework makespan. Numerous parameters are variables for planning issues like framework throughput, stack adjusting, benefit cost, benefit unwavering quality, and framework utilize.

Accessible assets must be utilized proficiently without influencing administration parameters of cloud. Cloud planning process benefit parameters are summed up in 3 organizes in particular ,

1. Resource discovering and filtering - Datacenter Broker notices resources in network and gets their status information.

2. Resource selection - Target resource selection is based on task and resource parameters. This is a deciding stage

3. Task submissions - Tasks are submitted to selected resources.

2.8) BEE COLONY OPTIMSATION LITERATURE

Since errands planning is dynamic in distributed computing condition, it is hard to acquire an appropriate arrangement that can limit undertakings usage, the season of executing info assignments and can consider stack adjust between assets.

The quantity of demand that has been entered to the framework is more and higher than the quantity of asset, and it doesn't stop . * The properties of solicitations that house been entered to nature are indistinct, and they consistently change, for example, entering time, executing time and required memory.

* Cloud figuring condition is an arrangement of heterogynous assets. * Resources home some equipment and programming properties, for example, stack volume, the measure of free memory in a framework, it has properties of correspondence system, for example, data

transfer capacity, movement and so forth. Kanlee and his partners (2011) proposed booking of cloud errands on the premise of load adjusting subterranean insect province enhancement calculation (LBACO)[17].

The principle strategy of this technique is load adjust of entire framework. Likewise, it has been endeavored to limit the season of Make traverse. In this article, the properties of subterranean insect calculation has been considered to plan the undertakings. Sourav Banerjee and his associates (2002) displayed booking calculation in light of hereditary algorithm[11]. Kennedy and his associates (2001) proposed a calculation that has been motivated from the conduct of feathered creatures and fish , and it is in unline discrete defeats utilized as a part of ants settlement calculation. This calculation called particles swarm calculation looks the arrangement space by altering different elements. Babu and his associates (2003) utilized honey bee metaheuristic calculation to achieve stack adjust in machines. The proposed calculation gives assignments need in machines line on the premise of the slightest holding up time. They discovered significant change in execution time and diminishing holding up time in line.

Chapter-3

SYSTEM DEVELOPMENT

3.1) TOOLS FOR CLOUD SIMULATION:

A. Eucalyptus (Eucalyptus 3.2.1) It is an open source cloud computing system. “Eucalyptus” provides a framework for software systems in cloud computing environment, especially for IaaS (Infrastructure as a Service). “Eucalyptus” not only provides some software as a service, it provides an entire virtual computing environment to the user as a service. It consists of a very simple architecture, in which all components interact with each other in a hierarchical manner. It provides user a familiar interface, which enables them to use tools in the same manner as they would be using on “Amazons EC2”. Operating systems and hypervisors for Eucalyptus are:-

a) All versions of Ubuntu

b) CentOS

c) RHEL

d) VMWare

(x86_64 architecture)

B. CloudSim (CloudSim Toolkit 4.0) This stands for Cloud “simulation”, is a framework for modelling as well as simulating various services and infrastructures on cloud. “CloudSim” can provide virtualised environment to users even on the fly while maintain the quality of service provided to the end user. It leverages virtualised environments based on the user requirement such as workload which varies with time. It provides user both the facility to either model the cloud or simulate various services to the intended user and provide better solutions. Another attractive feature of “CloudSim” is that, it provides user with a custom interface which allows user to decide implementing policies for the VM’s. It provides nearly all types of major services like IaaS, PaaS and “SaaS”. Some of its main features are illustrated below:-

Modelling and simulation support for

a) Virtualized servers

b) Resources with energy aware computing

c) Topology with data centred network

d) Cloud federation

C. Aneka (Aneka 3.0) For developing cloud applications “aneka” provides the essential platform. “Aneka” proves to be an extensible, flexible and to a great extent a market oriented cloud deployment and cloud application development solution using the .NET framework. Another important feature of “aneka” is that it equips the user with a SDK which can support multiple instances of programming model. It is essentially regarded as the market oriented cloud development as well as management framework. This title is owing to the fact that “aneka” allows user to create, schedule and monitor results using parameters like accounting, pricing and QoS or “SLA” (Service Level Agreement) for services on leased environments. Some of the lucrative feature of aneka are enlisted below :-

a) Rapid development and deployment of tools and framework.

b) SLA/QoS based provisioning ability.

c) Ability to run multiple programming instances and applications environment.

d) In order to obtain application result quickly and effectively ,it can gain multiple physical or virtual machines.

D. D. Haizea (haizea 1.3.tar.gz) In simple terms, “haizea” can be understood as a resource manager. Precisely, haizea is software which is employed to maintain a set of resources. These set of resources (often termed clusters) are managed by “haizea” and allocated to user as per requirement. It is obvious that an end user often asks for more than one resource at a time, in such scenario’s “haizea” comes to play and allocates the user a cluster or a part of it pertaining to the needs and the priority. It also termed as lease manager as mostly the resources are allocated to the user for a fixed time instant and for a fixed value of money, as per the contract of agreement. It alsoconsidered an appropriate replacement for “OPENNEBULA”. Some of the noticeable features of “haizea” are discussed below:-

a) “Allows requests for one and those with more no. of VM’s to run in parallel.

b) Provides a token for advance reservation of leases.

c) Leases marked as immediate are allowed to run instantly owing to priority issues without arising conflicts.

d) Maintains most users as cluster; facilitates fats allocation.”

3.2)“Optimization of Cloud Resource Service Adaptability using Genetic Algorithms”

We propose a novel method to minimize the cloud resource adaptation capability in cloud computing. The proposed model is formulated as liner programming model. Estimating the adaptation capability are cloud resources among services.

Minimize F(x, y) = ∑= n i 1 ai bi xi yi (1)

g1(x, y) = ∑= n i 1 bi xi yi <= Maximum executable instructions.

g2 (x, y) = ∀ yi <= Max. of Resource

g 3(x, y) = ∑= n i 1 xi <= MaxNo. CloudServices

“Where: o ai stands for computing capacity called CPU of Cloud resources of non-functional type i, expressed in millions of instructions per second (MIPS). o xi stands for number of Cloud resources of nonfunctional type i to allocate. o yi stands for number of hours of non-functional Cloud resource type i to allocate. o Cloud Resource Types stands for number of available non-functional Cloud resource types in a given Cloud environment. o Total_Instructions stands for total amount of instructions of a given application to be executed. o Deadline stands for deadline to execute a given application and it is expressed in complete hours (i.e., 60 m). o MaxNoResources stands for the overall maximum number of Cloud resources that can be allocated from Cloud providers, which is the summation of all the Cloud providers' limits with respect to the number of Cloud resources that can be allocated by Cloud consumers. o g1, g2, and g3 represent constraint functions. By minimizing F(x, y), the following can be determined: (a) the minimum number of Cloud resources (xi) to allocate and run in parallel, (b) the type (regarding price (ai) and computing capacity (bi) of Cloud resources to allocate, and (c) the minimum number of allocated hours (yi) of Cloud resources for executing applications while staying within budget and meeting deadlines. We propose a novel approach to predict the Cloud Resource Service Adaptability (CRSA) of client in cloud computing. Genetic Algorithm (GA) a stochastic global search method that mimics the process of natural evolution. The genetic algorithm starts with no knowledge of the correct solution and depends entirely on responses from its environment and evolution operators (i.e. reproduction, crossover and mutation) to arrive at the best solution. By starting at several independent points and searching in parallel, the algorithm avoids local minima and converging to sub optimal solutions. In this way, GAs have been shown to be capable of locating high performance areas in complex domains without experiencing the difficulties associated with high dimensionality, as may occur with gradient decent techniques or methods that rely on derivative information. In this, we consider the provision of resource adaptation for three basic applications called Business, Collaborative and Productivity. The goal is to find adaptation in terms of resources named, Bandwidth, Memory and Central Processing Unit (CPU) Cycle. Here, finding service adaptability is an uncertainty in nature, so, we are including a stochastic uncertainty with reasoning using GA."

"The informal GA has been used to identify the adaptability of clients in cloud computing. The proposed work is formulated as problem solving control system methodology. The control actions- the technique of GA is used to efficiently predict the cloud service adaptability in resource provision IV."

COMPUTE CLOUD RESOURCE SERVICE ADAPTABILITY (CRSA)

"Cloud computing" services incremented rapidly, so it is essential to adapt each client with their resources is a vital one. In order to provide these adaptability our system required bandwidth, memory, CPU cycle as a major resources. We analyze our proposed scheme in terms of three basic applications namely, community application, commercial application, construction application of optimization in MATLAB using GA and Performance evaluation using "CLOUDSIM". A. Resource allocation in Cloud using Genetic Algorithm In order to solve using genetic algorithms, we use binary coding. 10-bits are chosen for each variable, thereby making the total string length equal to 20. "Genetic Algorithm" has predefined steps in implementing the maximization function and finding the best fitness values. 1. In order to solve problem using Genetic algorithm, the variables are first

coded in binary coded string structures. The binary strings are made to point a value in the search space using the following mapping rule.)(/( )12)()()(−+= −i L l i U i L ii xxxx 2. Genetic algorithms mimic the survival of the fittest principle of nature to make a search process. So GA's are naturally suitable for solving maximization problems. For maximization problems, the "fitness function" can be considered the same as the objective function. F(x) =f(x): F (x) = [1/1+f(x)] 3. Now the strings in the mating pool undergo crossover operation. In Single point crossover, two strings with the highest fitness values are selected and crossover is done based on the probability 0.8 to find whether crossover is desired or not. If the coin flipping becomes true, crossover is done; otherwise strings are directly placed in the next generation. The crossover point is a random number selected between 0 and l-1 . The Strings undergo crossover at the selected crossover point. The complete population at the end of crossover is passed over the next generation.

Optimization Of Resources in Cloud Computing Using Effective Load Balancing Algorithms

“The proposed work least VM assign algorithm is compared with Active VM load balancer algorithm. The main aim is to distribute the load to the available VM efficiently so that the resources are not over or under utilized. Initially all the VM are assigned to zero. If the VM is used already used then its value is incremented.”

Then the VM having least value is assigned the load.

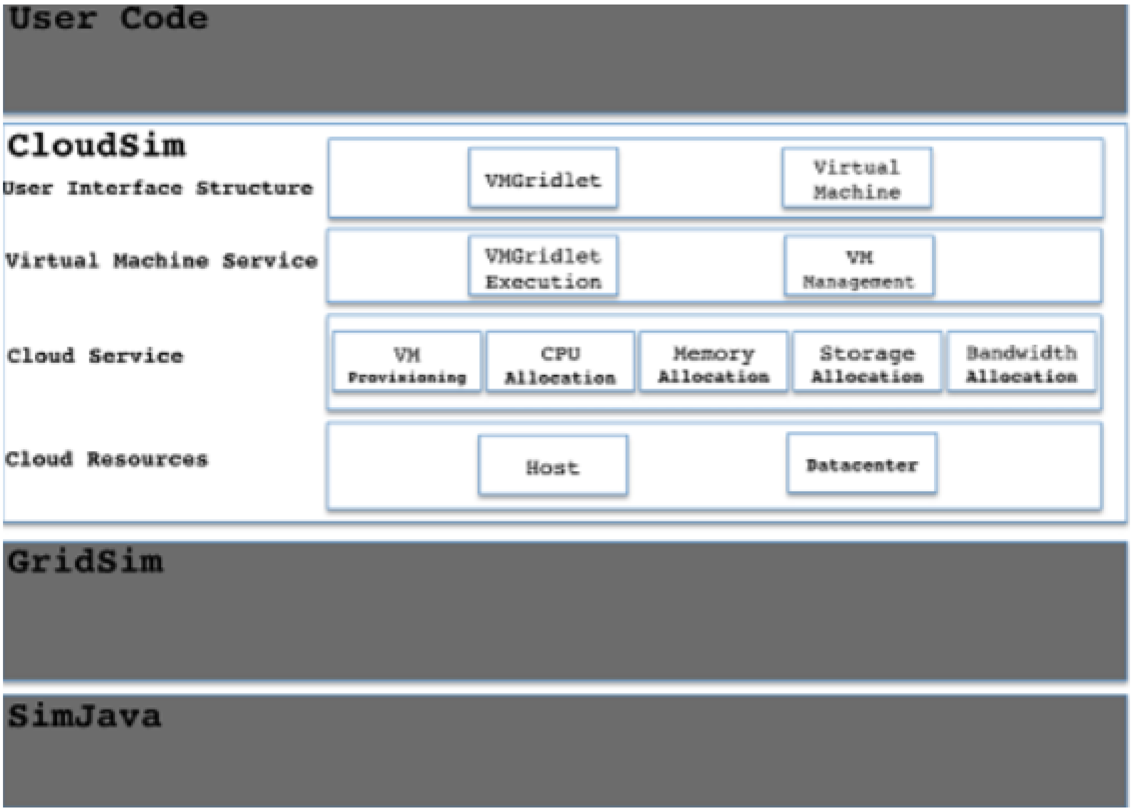
If the selected VM is not free then it is excluded from the VM list.

ALGORITHM

```
Least VM assign algorithm()
{ Inputs: Job= X1,X2,…………Xn
Initialize all the available VM to zero
Old VM OVM1,OVM2,………………..,OVMn=0;
 New VM NVM1,NVM2,…………………,NVMN=0;
 If (OVM1==1)
 { NVM1+=1; }
Else if { NVM1+=1; }
SelectedVM=LeastVMof(NVM1,NVM2,………………… ,NVMN)
 If(Selected VM is free)
 { Selected VM=req(X1,X2,…………Xn) }
Else { Least VM =exclude(selected VM) }
goto selected VM
```

SIMULATORS

The main aim of simulator is to test the implementation work in the absence of the required environment. Thus in the cloud environment two simulator are used “CloudSim” and Vcloud. CloudSim is the open source. Some “simulators” available for the distributed field such as SimGrid, GridSim, etc such simulators are not valid for the cloud computing as the cloud environment having multiple layers while “SimGrid” and “GridSim” are made for the single layer environment.

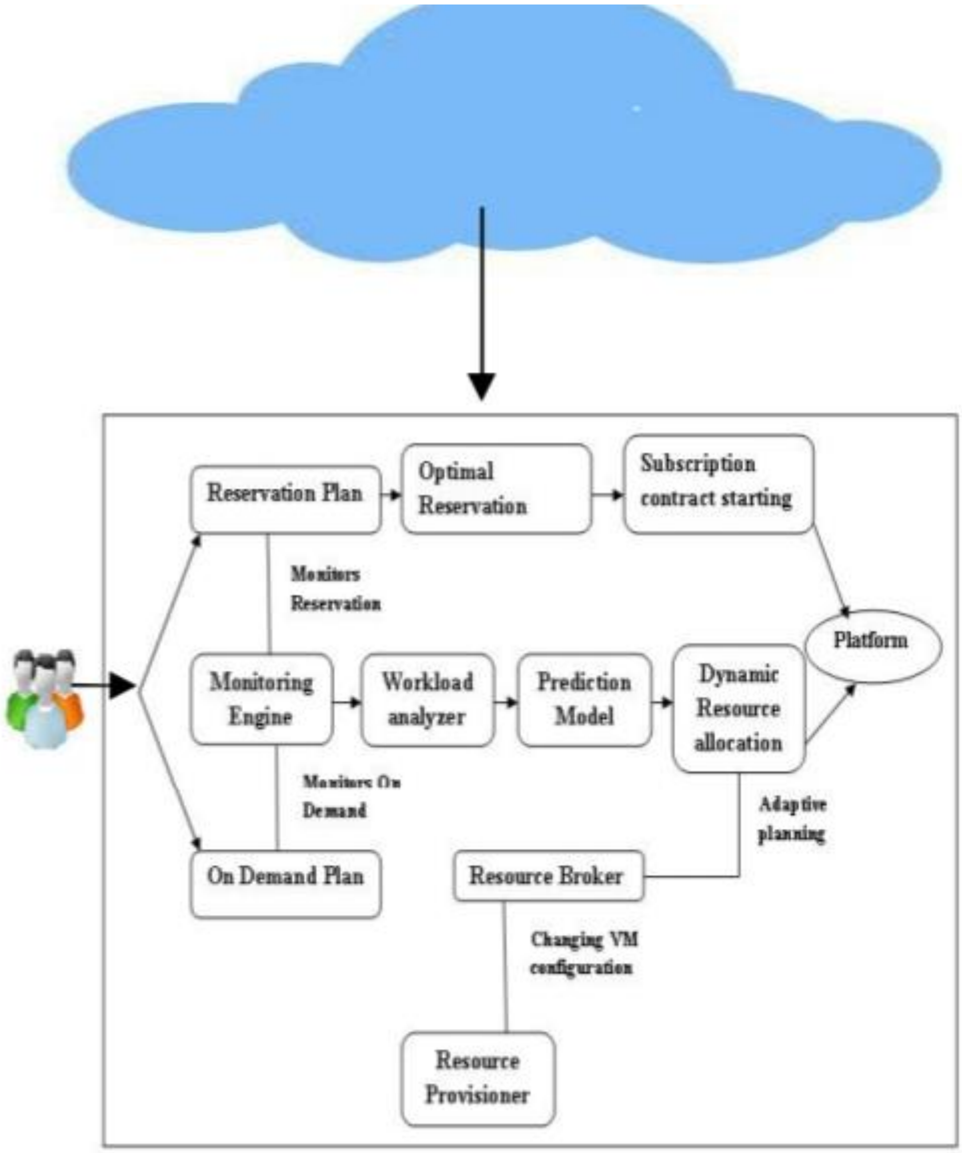


Optimizing the Cost for Resource Subscription Policy in IaaS Cloud

For long-term provisioning, we aim to determine the optimal number of “VMs” needs to be reserved. For short-term provisioning, we aim to determine how to configure “VMs” to provide sufficient service capacity for time varying workload. We assume that an “IaaS” provider offers multiple VM types. Each type features different hardware specifications. Besides, there are three different rental costs, including an upfront fee for long term “reservation”, a usage charge of reserved resources, and an on-demand cost on dynamically allocated resources. These costs are normalized to per short-term time interval hereafter for ease of cost calculation. Let V = {V1, V2, …, VM} denote the set of VM types and M be the total number of “VM” types supported by the IaaS provider. Each “VM” type has its own hardware specification and service capacity. Let Ci denote the capacity of Vi which corresponds to the maximum number of concurrent users or the service request rate that can be supported by an instance of Vi without violating the quality of service guarantee.

SYSTEM ARCHITECTURE

b



The optimization process has two phases, and their main functionalities are long-term resource reservation optimization and effective short-term resource allocation. In this section, we describe the proposed two-phase planning algorithms. A. Long term Resource Reservation In the long-term resource reservation,

1. “Given set of demands we have to calculate the provisioning cost which includes the upfront fee, the usage charge for launching reserved VMs when demand is less than reservation capacity, the usage charge of launching all the reserved capacity when the

demand exceeds the reserved capacity and the cost for on-demand allocated VMs to serve the exceeded demand.”

2. “The objective is to minimize the provisioning cost and to derive the optimal amount of long-term reserved resource with a model where the demand is a discrete random variable and only one single type of VM is considered.”

3.Assume that r* is the optimal number of VMs to be reserved for long term planning and calculate an upper and lower bound of the optimal number of reserved VMs. 4. To show how to use the result of single type VM solution for the original problem with multiple VM types. (i)Selecting the VM that has the best capacity/price (CP) ratio (ii)Under constraint (i), the workload demand is transformed to demand of the number of best CP ratio VMs (iii)Based on upper and lower bound we will obtain the optimal reservation (iv)We could do a search for the best combination of multiple VM types with capacity falls between the capacity of (r*-1).Cbestcp and r*.Cbestcp. Cbestcp is the capacity of the VM with the best CP ratio. (v)Each value in the range is regarded as the reserved demand of the Integer Linear Programming formulation. Minimize ∑i M ni * pi R (1) Subject to ∑i M ni * Ci ≥ Reserved Demand (2) ni € No i€M (3) (1)→to minimize the upfront cost of reserved VMs ni→number of type i VM that is subscribed in the long term lease contract. B. On Demand Resource Allocation A straight forward way to configure VMs for next shortterm planning interval is based on the measured demand of current I0 configuring VMs based on some prediction

mechanism will significantly reduce the operational cost. Our prediction mechanism is based on kalman filter because it has low computation complexity. SHORT-TERM PLANNING ALGORITHM (“SPA”) In the following,describe the proposed short-term planning algorithm (SPA). Depending on the values of rp, rc, and rr, the SPA classifies the resource planning scenarios into three cases which are illustrated as follows: Input: rm ,rp , rc , rr , Ic , Ir rm->VM Capacity requirement for current demand rp->Predictive “VM” Capacity rc->Current launched VM Capacity rr ->Overall VM Capacity of reserved resources Ic◊Current launched VM Configuration Ir->VM Configuration in reservation contract Output: The updated Ic, which is used for adaptive planning Initialization: IO :={0}//VM Configuration subscribed via ondemand plan is empty IO -> The “VM” configuration subscribed via on-demand plan

Procedure: 1 if rr < rp then

```
 2 Io ⇓ILP1_OnDemand (rp - rr ,Δ)
 3 Ic= Ir + Io
 4 else if rc<rp then
5Ic ←ILP2_AdjustVMConfiguration (Ir ,Δ,Ic ,rp)
 6 else if rc>rp then
7 Ic ←ShutDownSpareVMs (Ic, rp)
8 end if
 9 UpdatePredicationModel (rm )
10 return Ic
 End procedure
```

Depending on the values of rp, rc, and rr, the SPA” classifies the resource planning scenarios into three cases which are illustrated as follows:

Scenario 1 (lines 1-3): On Demand Resource

The predicted demand (“rp”) exceeds the capacity of all reserved VMs (rr), thus the Resource Broker must operate the on-demand option to subscribe more VMs.

Scenario 2(lines 4-5): Adjust VM configuration The predicted demand can be served by reserved VMs, but it exceeds the capacity of current VM configuration (“rc”). Therefore, reconfiguring launched VMs from the reserved VM pool (Ir) is necessary.

Scenario 3 (lines 6-7): Shutdown spare VMs The predicted demand is less than the currently configured VM capacity. Therefore, the corresponding action is to shut down some launched “VMs”, which had nearly a full hour of operation first, until the provisioning capacity is just above the predicted demand.

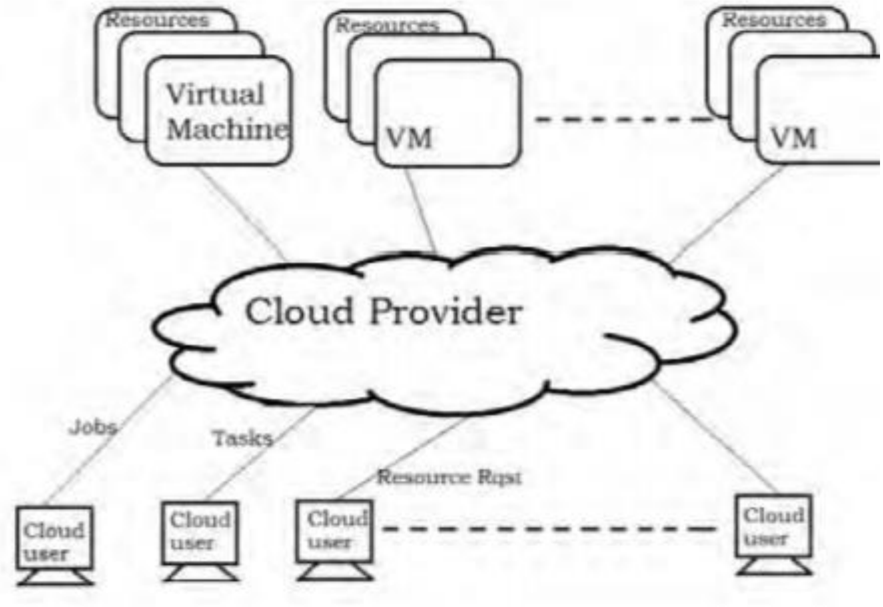


Fig. 2. Architecture of basic cloud resource allocation schema

THE BEE ALGORITHM

For example, as it has been shown in figure1, the cloud system involves scheduling of S and three cloud nodes of VM={VM1,VM2,VM3}. In "cloud" system, we consider S= {VM1, VM2, VM3}. As it has been already said, the number of cloud node is equal to the number of schedulers. "Bees" involve three groups in bee colony algorithm. They involve employed bees, supervision. They involve employed bees, supervisory bees and scout bees. The "bee" staying in dancing area to make decision and select the food source is called explorer bee. The "bee" going toward predetermined food source is called employed bee, while the bee performing random search is called scout bee. The main steps of algorithms are as follow: • Initial initialization • Repetition

A. "The location of employed bees in food sources in the memory"

B. “The location of search bees in food sources in the memory”

C. “Sending scout bees to search new food sources”

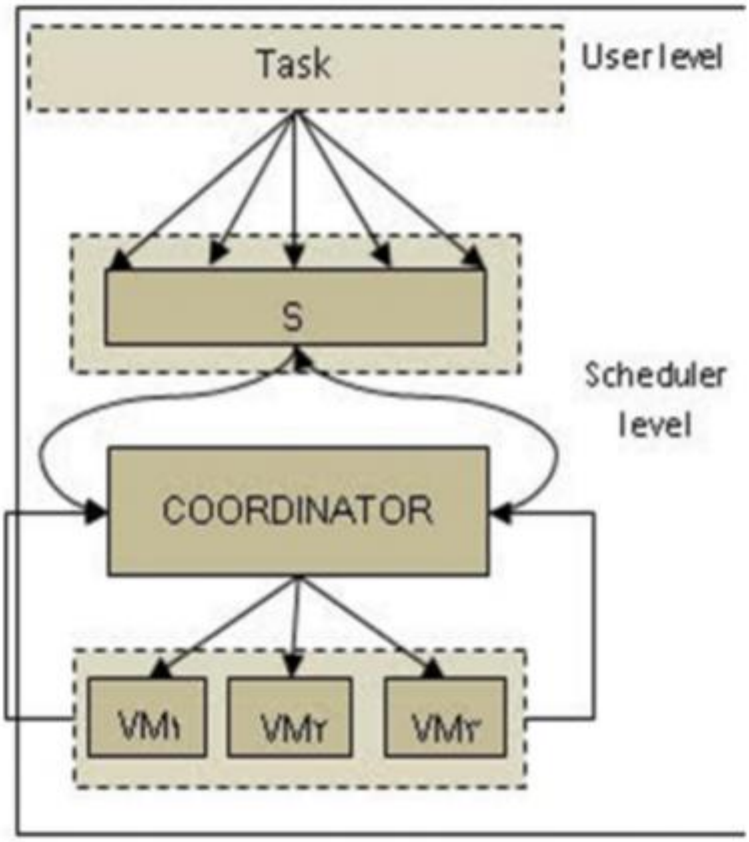


Fig. 1. the assumptions of cloud system for proposed algorithm.

VM= {VM1, VM2,...., VM3} Task= {T1, T2,...Tn} Each bee is the answer of a problem, so the bee is shown as follows: Bee=[X11, X21... Xij] Xij shows the ith task processing in jth processor. For instance, X12 shows that the first task is processed in the second processor. Therefore, in order to simplify it, the bee has been shown as follows: B=2 3 1 2 1 3

Generally, the stages are as follows:

1. Creating the initial population

2. Computing the fitness function

3. Ordering the population based on fitness function.

4. Defining the “threshold” limit

- If the random number is lesser than “threshold” limit, then some locations are randomly selected. Random numbers are created by a uniform distribution between [-1, 1]. The selected locations created by random numbers are summed up.

- If the random number is lesser than “threshold” limit, nodes information is extracted from coordinator. A specified percentage of nodes having higher volume of processing is determined. A specified percentage of the nodes having the least volume of processing is determined. The load of nodes in moved.

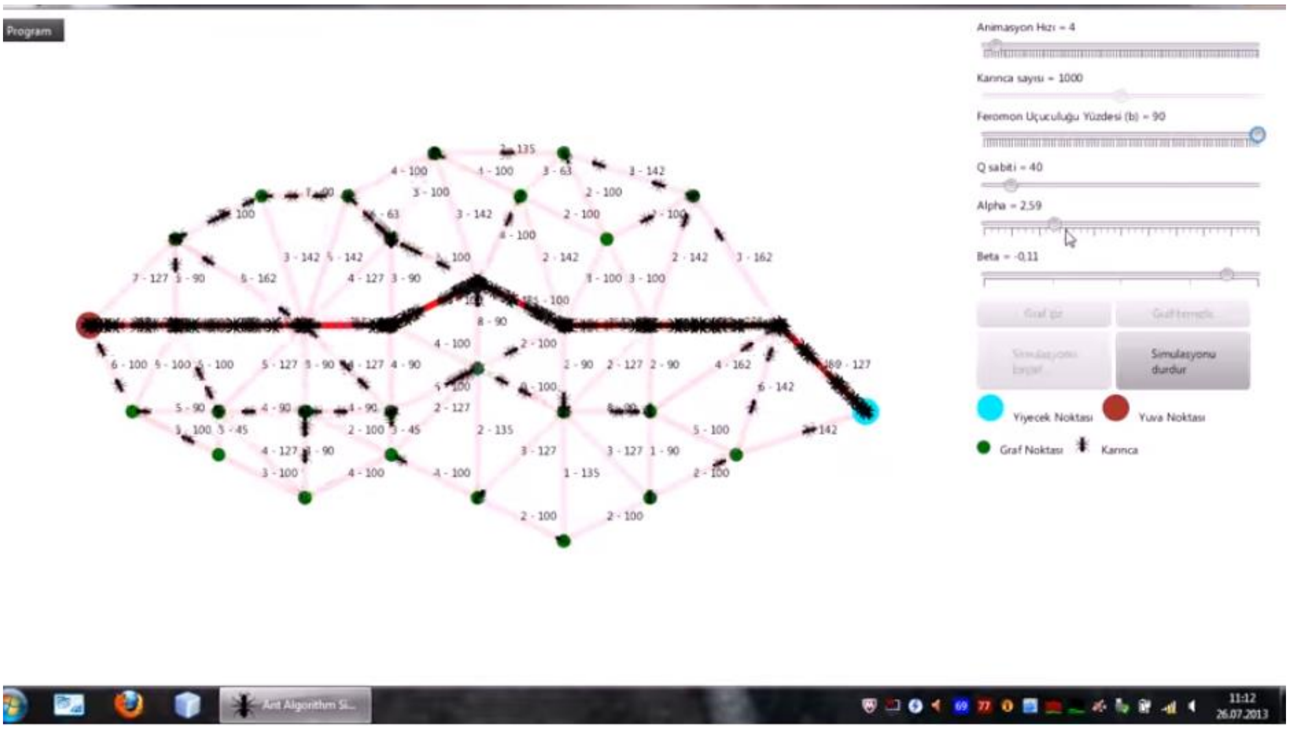
5. Computing fitness function

6. Selecting the best bees

7. Producing the new initial population 8. If the number of replications is enough, it ends; otherwise, return to stage 3.

Chapter-4

PERFORMANCE ANALYSIS



Animasyon Hızı = 4

Karınca sayısı = 1000

Feromon Uçuculuğu Yüzdesi (b) = 90

Q sabiti = 40

Alpha = 2,59

Beta = -0,11

ANT ALGO SIMULATION

The proposed cloud model has to be evaluated for its performance. The real time “CloudSim” tool helps the cloud users and the cloud service providers to evaluate the running cloud environment. The CloudReports tool, a GUI tool built over the “CloudSim” tool makes the evaluation easier. The virtual environment of the proposed model can be visualized and the cloudlets running and profit attained can be viewed. C Cloud Reports is a graphic tool that simulates distributed computing environments based on the “Cloud” Computing paradigm. It uses “CloudSim” as its simulation engine and provides an easy-to-use user interface, report generation features and creation of extensions in a plug-in fashion. The application simulates software as a Service (“SaaS”) provider with an arbitrary number of datacenters. Each datacenter is entirely customizable. The user can easily set the amount of computational nodes (hosts) and their resource configuration, which includes processing capacity, amount of “RAM”, available bandwidth, resource utilization and execution time. The user can set the number of “vm” each customer owns, a broker responsible for allocating these virtual machines and resource consumption algorithms. Each virtual machine has its own configuration that consists of its hyper visor, image size, scheduling algorithms for tasks (here known as cloudlets) and required processing capacity, “RAM” and bandwidth. Additionally, Cloud Reports generates HTML reports of each simulation and raw data files that can be easily imported by third-party applications such as “Octave” or MATLAB. By using “Cloud” Reports, the cloud service providers are able to evaluate their cloud environment before leasing the services to the users.

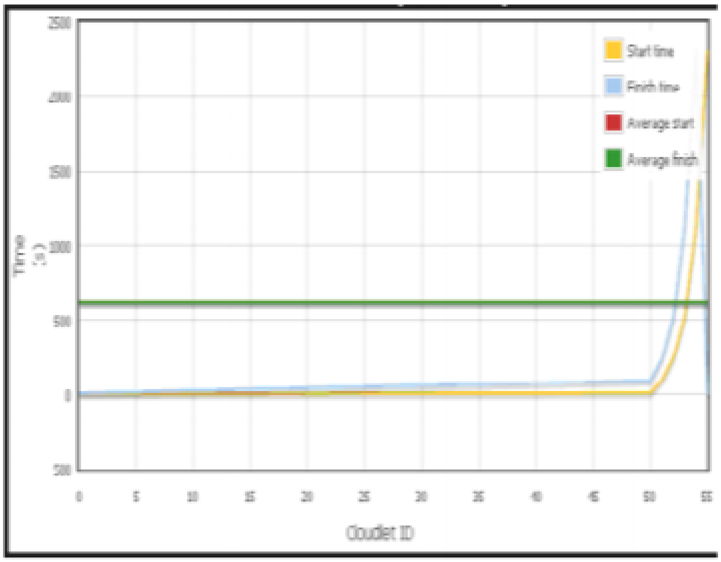


Fig. 5 Execution views of the cloud services.

The execution time of the different “cloudlets” based on resource allocation in the proposed model is simulated. The adaptability of “cloudlets” to be executed is got from the optimization tool. In Figure depicts the service adaptability and execution time of a particular client. It includes an average start time and finish time are showed. The resource utilization of each client, adaptability displayed in Figure . The resources utilized by the services are “simulated”. The three major types of resources are considered such as CPU, RAM and bandwidth. The memory and bandwidth usage is almost constant whereas the “CPU” usage depends on the type of priority service executing

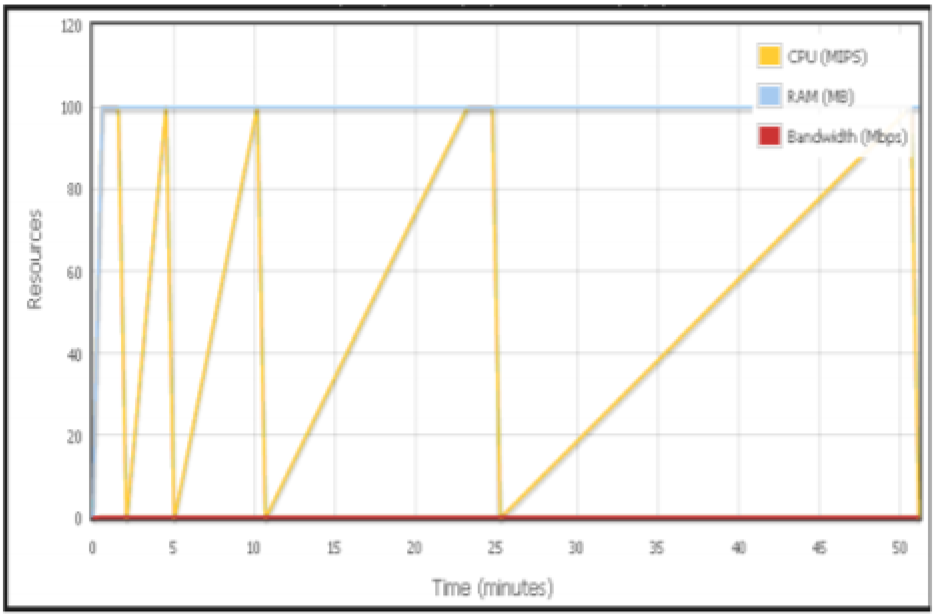


Fig. 6 Resource utilization in the Datacenter

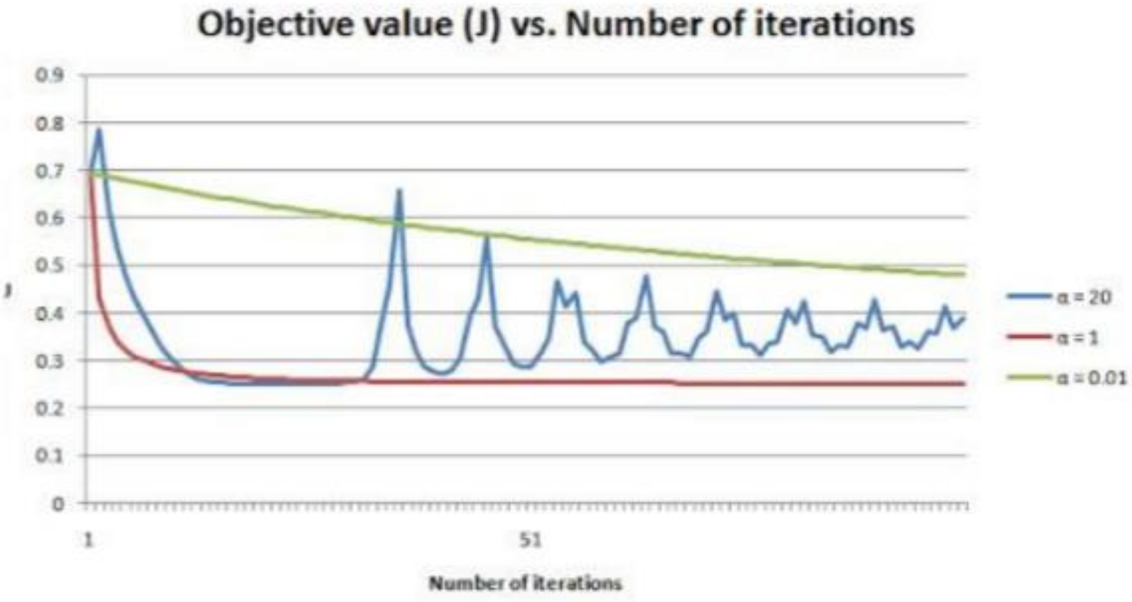


Figure 3. Number of Iterations vs. Objective Value.

In this report “CloudSim” platform is chosen for the test, and the CPU is core i3 and 8GDDR3 while the operation system is Windows Xp. In this paper, distribution of resource in cloud computing is carried out from three aspects. (1) Comparison with Basic “Cuckoo Algorithm” (CS) Set there are 800 virtual tasks and 50 virtual nodes.

It can be found by comparing energy consumption of two algorithms that algorithm in this paper consumes more energy in the beginning, mainly because “Gaussian mutation” and self-adaptive factors are adopted to make the algorithm have larger vibration at first and then tend to be stable later. Compared with the energy vibration during the process of “CS algorithm”, algorithm in this paper has lower consumption of energy

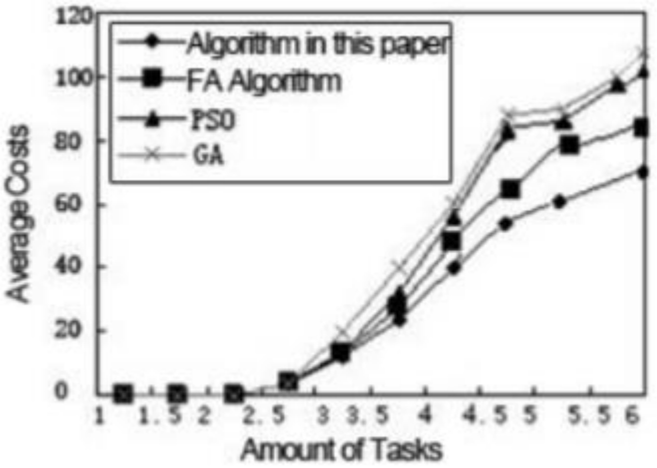
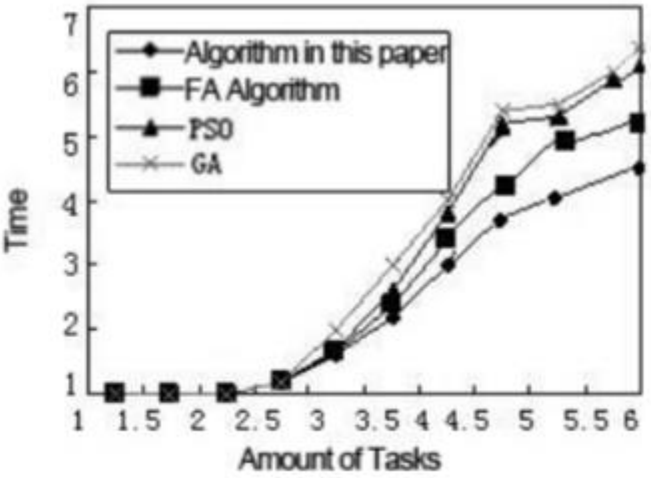


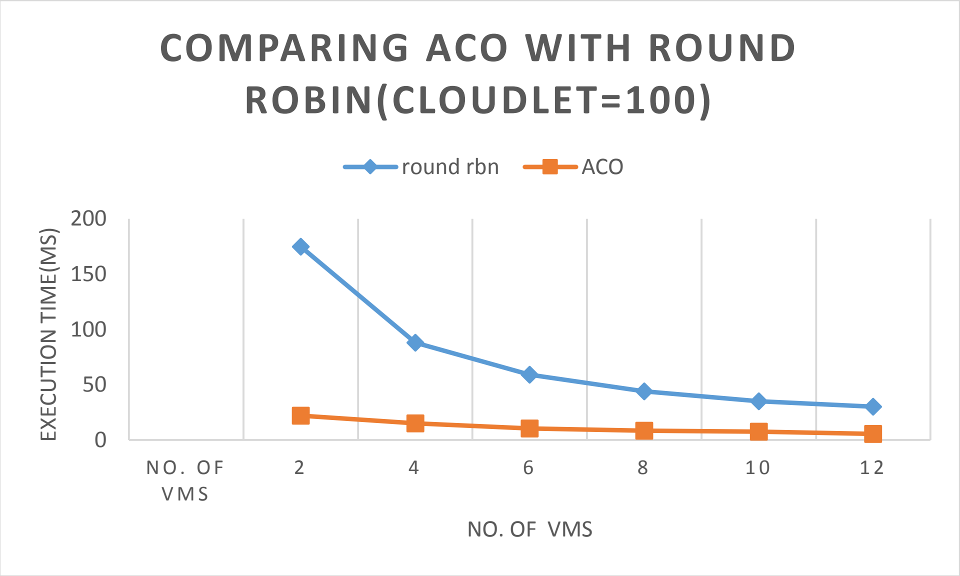
Figure 4. Comparison of Costs to Complete Tasks

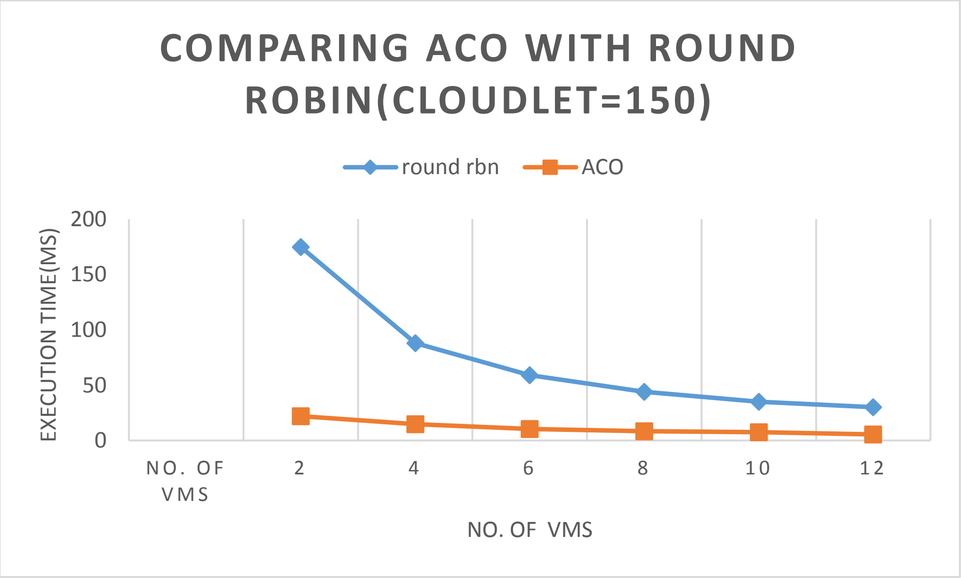
It can be found from Figure that this “algorithm” in terms of resource load time is lower than the other three reference algorithm and is increasing along with the number of tasks and time consuming small range, this algorithm is relatively stable, resource load balancing can better. From the modified algorithm is found in Figure 3 in a “cloud” computing environment with virtual node network consumption in more energy than the other three references algorithm. Introduction from the above algorithms can be found in the fitness function can effectively balance time and the relationship between costs, so that they can better meet the requirements of “cloud” computing resource scheduling. (3) Comparison with Other Intelligent Algorithms Select 500 tasks run on a platform in the cloud computing resources is 50, using “GA”, “PSO”, FA and the algorithm to distribute the tasks and resources, this algorithm can be found in Figure 4 is different from the other three intelligent algorithm, derived cloud computing resource scheduling performance cost of this task is better solution. From Figure , you can find this algorithm differs from the other three algorithms in terms of task completion time for cloud computing resource scheduling performance better programs.

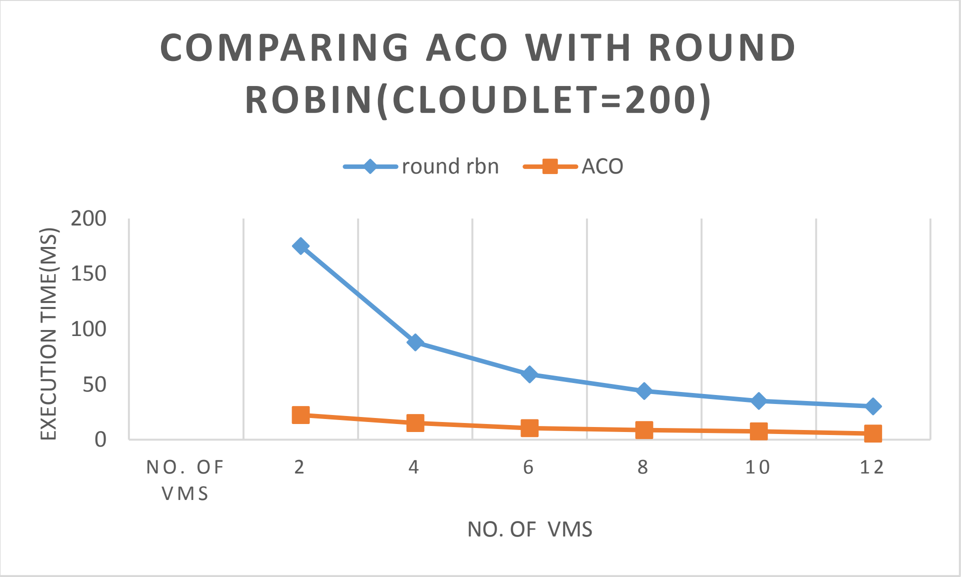


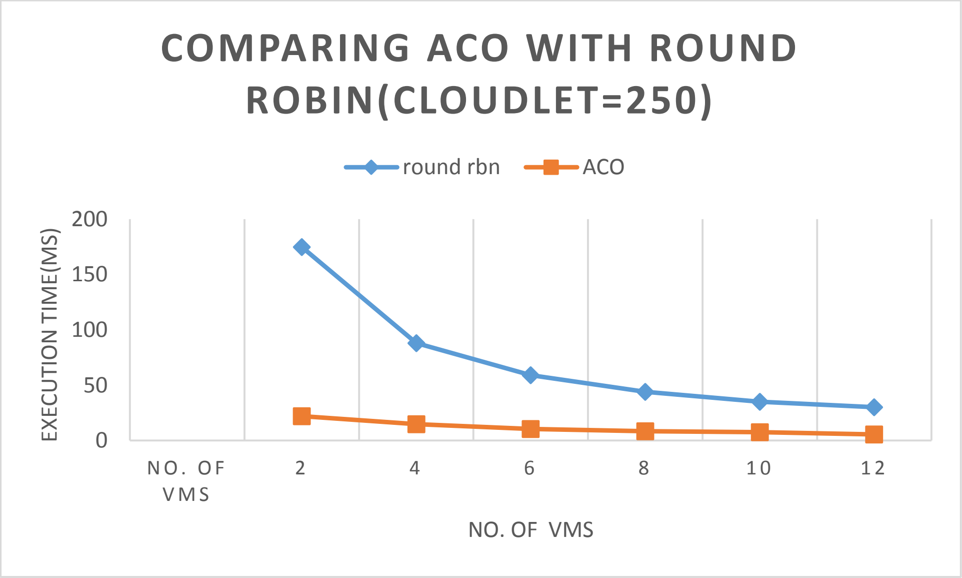
igure 5. Comparison of Time to Complete Tasks

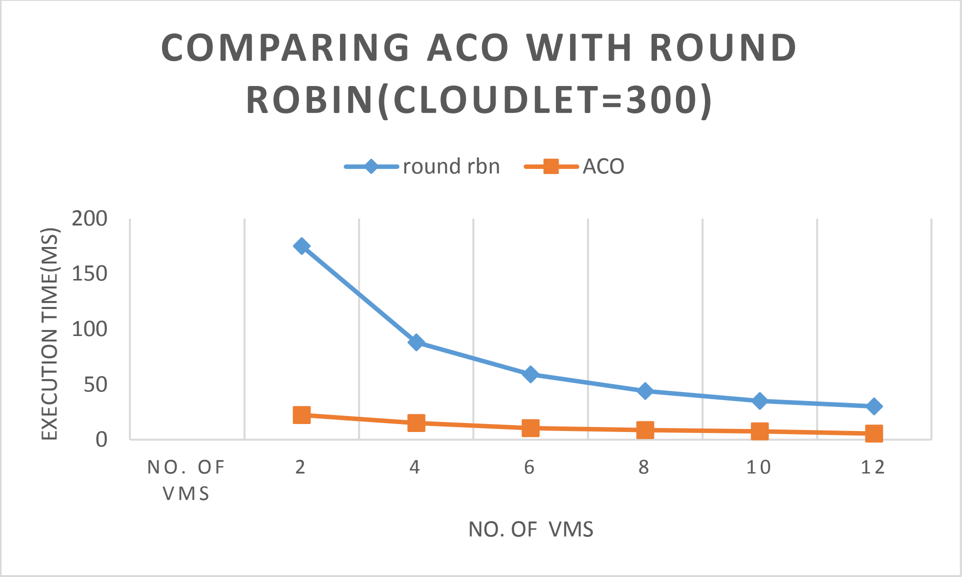
Results:

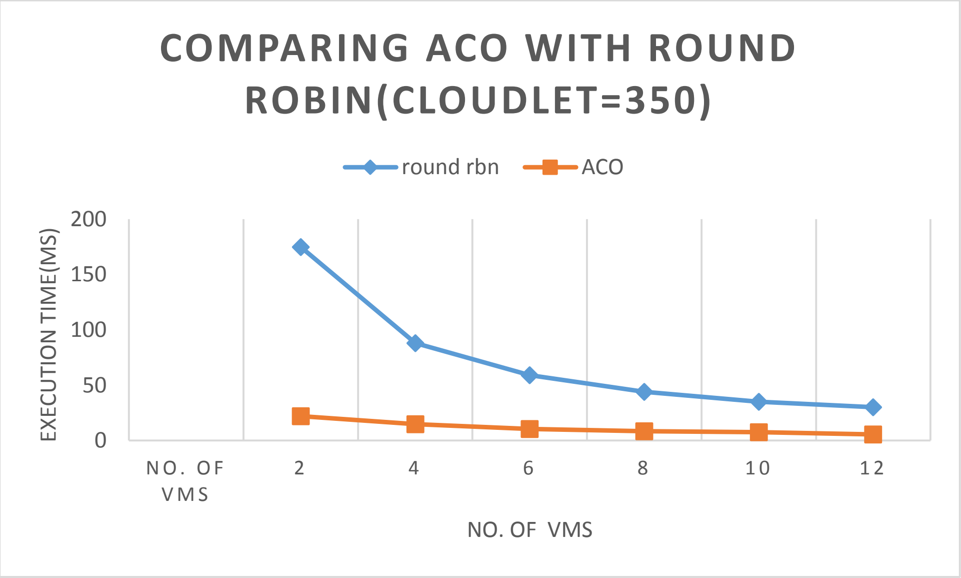



COMPARING ACO WITH ROUND ROBIN(CLOUDLET=150)
round rbn
ACO
EXECUTION TIME(MS)
200
150
100
50
0
NO. OF VMS
2
4
6
8
10
12
NO. OF VMS


COMPARING ACO WITH ROUND ROBIN(CLOUDLET=200)
round rbn
ACO
EXECUTION TIME(MS)
200
150
100
50
0
NO. OF VMS
2
4
6
8
10
12
NO. OF VMS

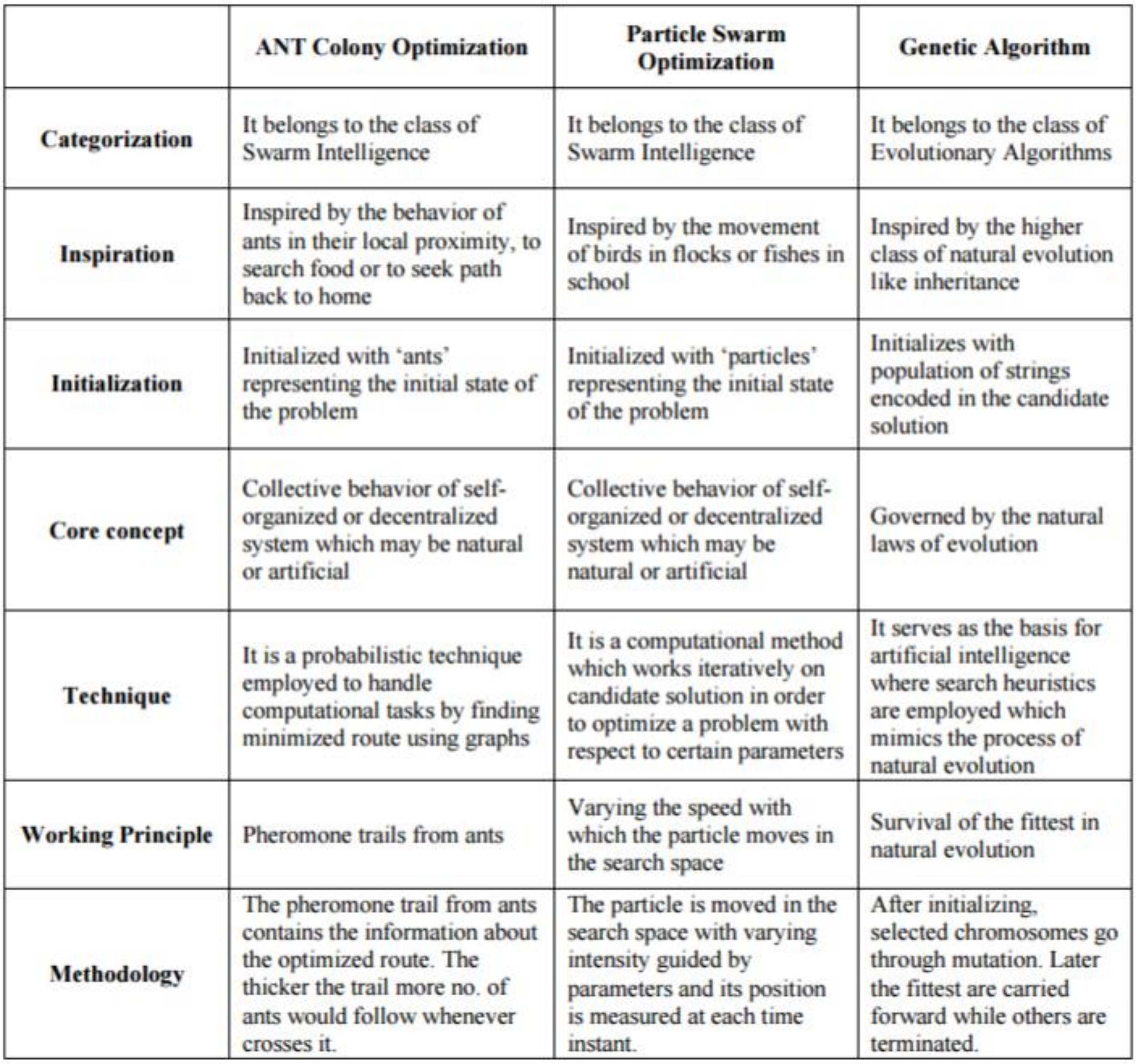

COMPARING ACO WITH ROUND ROBIN(CLOUDLET=250)
round rbn
ACO
EXECUTION TIME(MS)
200
150
100
50
0
NO. OF VMS
2
4
6
8
10
12
NO. OF VMS


COMPARING ACO WITH ROUND ROBIN(CLOUDLET=300)
round rbn
ACO
EXECUTION TIME(MS)
200
150
100
50
0
NO. OF VMS
2
4
6
8
10
12
NO. OF VMS



As we can see from the line chart in every case, the execution time for ant colony optimisation is less than the execution time for round robin algorithm with varying cloudlets.

TABLE I
Comparative analysis of various Resource optimization algorithms



	ANT Colony Optimization	Particle Swarm Optimization	Genetic Algorithm
Categorization	It belongs to the class of Swarm Intelligence	It belongs to the class of Swarm Intelligence	It belongs to the class of Evolutionary Algorithms
Inspiration	Inspired by the behavior of ants in their local proximity, to search food or to seek path back to home	Inspired by the movement of birds in flocks or fishes in school	Inspired by the higher class of natural evolution like inheritance
Initialization	Initialized with 'ants' representing the initial state of the problem	Initialized with 'particles' representing the initial state of the problem	Initializes with population of strings encoded in the candidate solution
Core concept	Collective behavior of self-organized or decentralized system which may be natural or artificial	Collective behavior of self-organized or decentralized system which may be natural or artificial	Governed by the natural laws of evolution
Technique	It is a probabilistic technique employed to handle computational tasks by finding minimized route using graphs	It is a computational method which works iteratively on candidate solution in order to optimize a problem with respect to certain parameters	It serves as the basis for artificial intelligence where search heuristics are employed which mimics the process of natural evolution
Working Principle	Pheromone trails from ants	Varying the speed with which the particle moves in the search space	Survival of the fittest in natural evolution
Methodology	The pheromone trail from ants contains the information about the optimized route. The thicker the trail more no. of ants would follow whenever crosses it.	The particle is moved in the search space with varying intensity guided by parameters and its position is measured at each time instant.	After initializing, selected chromosomes go through mutation. Later the fittest are carried forward while others are terminated.

Chapter-5

CONCLUSIONS

As the “cloud computing” is a new style of computing over internet. It has many advantages along with some crucial issues to be resolved in order to improve reliability of “cloud” environment. These issues are related with the load management, fault tolerance and different security issues in “cloud” environment. The load can be CPU load, memory capacity, delay or network load. In “cloud” computing, cloud providers can offer cloud consumers two provisioning plans for computing resources, namely reservation and on-demand plans. In general, cost of utilizing “computing” resources provisioned by reservation plan is cheaper than that provisioned by on-demand plan The load management is done using “round robin” method. From the results, the algorithm can optimally adjust the trade off between reservation of resources and allocation of on-demand resources.

According to the client conception our proposed system provides resources to them in optimized manner by using genetic algorithm. It allows to finding out the durability of the clients, which makes to maintain the consistency of them to provide services and resources. The execution time influences the client to access the services in best way for the efficiency concern. It is based on worldwide regions, in order to strike a good resource balance between client and cloud service resource provider.

Hence, the comparison is made among various algorithms and there execution time is checked,and the most optimal algorithm is analysed.

The proposed algorithm least “VM” assign method distribute workload across multiple computers to achieve optimal resource utilization with minimum response time. Thus problems in existing algorithms are overcome in proposed method thus achieving increased “resource utilization”, minimum response time and maximum user satisfaction.

Future scopes:

In this rapidly evolving “computing” world, resources are of utter importance. Therefore it becomes highly essential and mandatory to effectively manage these resources and a major part of it is dominated by effective scheduling of jobs on the resources. Thus, we need to find the most effective “algorithm” to schedule the jobs for the resources.

As discussed earlier, there are several algorithms which have been employed for effective “scheduling” of resources. However, none of them is perfect. Each one suffers from one or more of their little flaws. These algorithms were designed to get the optimal or atleast nearly “optimal” solution of the problem. However, the current scenario demands of much more accurate algorithm for resource scheduling and resource allocation.

The biggest research challenge is to discover such an algorithm. There are various parameters which need to be considered, evaluated and modified in order to attain the goal, to develop a perfect job scheduling “optimization” algorithm. Some of the highly discussed research concerns are enlisted below:- A. Un-interrupted connection throughout the scheduling The “cloud” services are provided to the end user extend over large physical areas. Some of them are covered by wired links while others by wireless communication. In such scenarios, while we develop an efficient “algorithm” we also need to consider the challenges faced by wireless communication such as fading, interference, attenuation, noise etc. While mobile agents are communicating to design an efficient strategy for resource scheduling we should be prepared that none these problems hinder the process of scheduling.

The scheduling algorithm should run while keeping a backtrack of the progress so far covered. So that in case of any hindrance the scheduling can resume from most effective point . B. “Overhead” of scheduling Another most important thing to be considered is the main goal of cloud computing. The main goal is to “provide optimal services that too at a minimal expense to the users”. In order to achieve this it must be ensured that the resource scheduling algorithm undertaken not only benefits the cloud provider but also proceed in direction to provide optimal economic benefits to the end users also. C. Robust and flexible

The resource scheduling technique must be robust and flexible at the same time. In case of any failure of VM’s the jobs submitted by user must be re-scheduled on other “VM’s” without disturbing the execution of other jobs.

The scheduling algorithm should be such that it provides complete flexibility to the user. The user can anytime withdraw his job from the “scheduling”, however when he again submits task it must be re-initiated from the same point, where it was left . “Scheduling” of jobs is a classical problem. All the possible options are considered while the others are explored. Scheduling meta-jobs (independent jobs) has been considered.

Dynamic Acyclic Graphs, commonly referred as “DAG’s” have been used to represent application tasks. Dynamic approach in terms of techniques like hybrid mapper and algorithms like generational algorithms are being implemented. All the effort and research is directed with one notion, to achieve an algorithm which can provide “optimal” usage of the resources by effectively scheduling and maintaining them.

REFERENCES

[1]Amandeep Kaur sidhu1and Supriya Kinger2, “Analysis of Load Balancing Techniques in Cloud Computing”, International Journal of Computers & Technology, volume 4, No. 2, March- April 2013, pg 737- 741.

[2]Shridhar G. Domanal and G. Ram Mohana Reddy," Load Balancing in Cloud Computing Using Modified Throttled Algorithm"IEEE, International conference. CCEM 2013. In press. [3]

[3]. Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela AI-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", in Second Symposium on Network Cloud Computing and Applications, 978-0-7695-4943-9/12, IEEE 2012.

[4]. Mr.Manan D. Shah, "Allocation Of Virtual Machines In Cloud Computing Using Load Balancing Algorithm" in International Journal of Computer Science and Information Technology & Security (lJCSlTS), ISSN: 2249-9555 Vol. 3, No.1, February 2013.

[5]. Pooja Samal1and Pranati Mishra2, “Analysis of Variants in Round Robin Algorithms for Load Balancing in Cloud Computing”, (IJCSIT) International Journals of Computer Science and Information Technologies, Volume 4 (3), 2013, pg. no. 416- 419.

[6]. Sonika Matele1, Dr, K James2 and Navneet Singh3, “A Study of Load Balancing Issue Among Multifarious Issues of Cloud Computing Environment”, International Journals of Emerging Technolog Computational and Applied Science (IJETCAS), volume 13- 142, 2013, pg. 236- 241.

[7]. Prof Meenakshi Sharma1 and Pankaj Sharma2, “Performance Evaluation of Adaptive Virtual Machine Load Balancing Algorithm”, International Journal of Advanced Computer Science and Applications, volume 3, no. 2, 2012, pp. 86-88.

[8]. Martin Randles, Enas Odat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab,”A Comparative Experiment in Distributed Load Balancing”, 2009 Second Interna-tional Conference on Developments in eSystems Engineering.

[9]. Hemant S. Mahalle, Parag R. Kaveri and Vinay Chavan, "Load Balancing On Cloud Data Centres" in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, issue 1, January 2013.

[10]. Hung-Chang Hsiao, Hsueh-Yi Chung, Haiying Shen, and Yu-Chang Chao, "Load Rebalancing for Distributed File Systems in Clouds" IEEE transactions on parallel and distributed systems, vol. 24, no. 5, may 2013.

[11] C. Kang and Z. Wei-Min, “Cloud Computing: System Instances and Current Research”, Journal of Software , vol. 20, no. 5, (2009), pp.1337-1348.

[12] L. Di, “Enhanced fairness-based multi-resource allocation algorithm for cloud computing[J]”.Journal of Xidian University, vol. 41, no. 3, (2014), pp.175-181.

[13] Y. Zhao-feng， W. Qi-ming， L Hai-lian, “Research on Resource Scheduling Algorithm of Cloud Computing Based on Improved DAG Diagram and Task Delay[J]”, Computer Measurement & Control , vol. 22, no. 2, (2014), pp. 499-502.

[14] L Bo， L Qing-feng, “Resource Scheduling Model in Cloud Computing Based on QoS and Utility[J]”, Computer Measurement & Control , vol. 22, no. 3, (2014), pp. 826-828.

[15] A.H. Alhusaini and V.K. Prasanna, C.S. Raghavendra, “A Unified Resource Scheduling Framework for Heterogeneous Computing Environments”, HCW '99 Proceedings of the Eighth Heterogeneous Computing Workshop, 1999, pp 156 – 165.

[2] C.D. Chapman, M.J. Jakiela, “Genetic algorithm-based structural topology design with compliance and topology simplification considerations”. M.I.T, Journal Of Mechanical Design: vol. 118, n1: pp. 89-98 (24 ref.) June 01, 1994. doi:10.1115/1.2826862

[3] Cloud Book [2012] “Cloud Federation”, [Online]. Available: http://www.cloudbook.net/directories/research project.php?id=100004

[4] Cloudsim [2013] “Infrastructures, Services and features” [Online]. Available: http://code.google.com/p/cloudsim/

[5] Eberhart R.C., Shi Y., Kennedy J., “Swarm Intelligence”. 2001

[6] H. Fattah and C. Leung, “An Overview of Scheduling Algorithms in Wireless Multimedia Networks”, IEEE Wireless Communications, October 2002, 9(4), pp: 76 – 83

[7] H. Mühlenbein and D. Schlierkamp-Voosen, “Evolutionary Computation”. Journal of Evolutionary Computation, Spring 1993 by the Massachusetts Institute of Technogy: , Vol. 1, No. 1, December 10, 2007. doi>10.1162/evco.1993.1.1.25

[8] H.Zhong, K. Tao, X. Zhang, “An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems”, The Fifth Annual ChinaGrid Conference, July 2010. pp. 124 – 129. do10.1109/ChinaGrid.2010.37

[9] Ian Kenny, “Dynamic, Hierarchical Particle Swarm Optimization”, Technical report, 2008-2009.

[10] J.Alcaraz and C.Maroto “A Robust Genetic Algorithm for Resource Allocation in Project Scheduling”. Annals of Operations Research 102: pp 83-109, February 2001, Volume 102, Issue 1-4

[11] K. Li, G. Xu, G. Zhao, Y. Dong, D. Wang, “Cloud Task scheduling based on Load Balancing Ant Colony Optimization”, Sixth Annual ChinaGrid Conference, 22-23 Aug. 2011, pp 3-9. doi: 10.1109/ChinaGrid.2011.17

[12] K.Nishant, P. Sharma, V. Krishna, C. Gupta and K.P. Singh, Nitin and R.Rastogi, “Load Balancing of Nodes in Cloud Using Ant Colony Optimization”, 14th International Conference on Modelling and Simulation, March 2012, pp.3-8

[13] M. Iverson and F. Ozguner, “Dynamic, competitive scheduling of multiple DAGs in a distributed heterogeneous environment,” 7th Heterogeneous Computing Workshop (HCW’98), , March 1998, pp. 70-78. doi: 10.1109/HCW.1998.666540

[14] M. Maheswaran and H. J. Siegel, “A Dynamic matching and scheduling algorithm for heterogeneous computing systems,” 7th Heterogeneous Computing Workshop (HCW’98), March 1998, pp. 57-69.

[15] M.B. Wall, “A Genetic Algorithm for Resource-Constrained Scheduling” – Ph.D Thesis, M.I.T., June 1996

[16] Q. Zhang, L. Cheng, R.Boutaba, “Cloud computing: state-of-the-art and research challenges”, Springer, May 2010, Volume 1, Issue 1, pp 7-18

[17] R. Mishra and A. Jaiswal, “Ant colony Optimization: A Solution of Load balancing in Cloud”, International Journal of Web & Semantic Technology(IJWesT), April 2012, Vol.3, No.2, pp. 33 – 50

[18] S. Banerjee, I. Mukherjee, and P.K. Mahanti, “Cloud Computing Initiative using Modified Ant Colony Framework”, World Academy of Science, Engineering and Technology (56) 2009, pp 221 – 224

[19] S. Hartmann, “A competitive genetic algorithm for resource-constrained project scheduling”, Naval Research Logistics, 7 DEC 1998, Volume 45, Issue 7 pp 733–750. doi: 10.1002/(SICI)1520-6750(199810)45:73.0.CO;2-C

[20] S. Pandey, L. Wu, S.M.Guru, R. Buyya, “A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments”, 24th IEEE International Conference on Advanced Information Networking and Applications, 20- 23 April 2012, pp 400 – 407. doi: 10.1109/AINA.2010.31