Ant colony Optimization: A Solution of Load balancing in Cloud

Ratan Mishra[1] and Anant Jaiswal[2]

[1]Amity school of engineering & Technology, Noida,India
mishra.ratan86@gmail.com
[2]Amity school of computer Science, Noida,India
Anantvns20@gmail.com

Abstract

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. In this paper the main concern is load balancing in cloud computing. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. Many methods to resolve this problem has been came into existence like Particle Swarm Optimization, hash method, genetic algorithms and several scheduling based algorithms are there. In this paper we are proposing a method based on Ant Colony optimization to resolve the problem of load balancing in cloud environment.

Keywords:

Cloud computing, Load balance, Ant colony optimization, Swarm intelligence

1. Introduction

1.1 Cloud Computing

Cloud computing has become very popular in recent years as it offers greater flexibility and availability of computing resources at very low cost. The major concern for agencies and organizations considering moving the applications to public cloud computing environments is the emergence of cloud computing facilities to have far-reaching effects on the systems and networks of the organizations. Many of the features that make cloud computing attractive, however, can also be at odds with traditional security models and controls. As with any emerging information technology area, cloud computing should be approached carefully with due consideration to the sensitivity of data. Planning helps to ensure that the computing environment is as secure as possible and is in compliance with all relevant organizational policies and that data privacy is maintained. It also helps to ensure that the agency derives full benefit from information technology spending. The security objectives of an organization are a key factor for decisions about outsourcing information technology services and, in particular, for decisions about transitioning organizational data, applications, and other resources to a public cloud computing environment. To maximize effectiveness and minimize costs, security and privacy must be considered from the initial planning stage at the start of the systems development life cycle. Attempting to address security after implementation and deployment is not only much more difficult and expensive, but also more risky. Cloud providers are generally not aware of a specific organization's security and privacy needs. Adjustments to the cloud computing environment may be warranted to meet an organization's requirements. Organizations should require that any selected public cloud computing solution is configured, deployed, and managed to meet their security and privacy requirements. Every industry standard states that there should be security and privacy policies. Security policies are the top-level set of documents of a company. They document company's decision on the protection, sharing and use of information. A complete and appropriate set of policies will help to avoid liability and compliance problems.Cloud computing has been defined by NIST as 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 cloud provider interaction[1]. Cloud computing technologies can be implemented in a wide variety of architectures, under different service and deployment models, and can coexist with other technologies and software design approaches. The security and privacy challenges cloud computing presents, however, are formidable, especially for public clouds whose infrastructure and computational resources are owned by an outside party that sells those services to the general public.

1.2 Cloud Service Models

Cloud service delivery is divided into three models. The three service models are :

1.2.1 Cloud Software as a service (Saas)

The capability provided to the consumer is to use the providers applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser. The consumer does not manage the underlying cloud infrastructure.

1.2.2 Cloud Platform as a Service (Paas)

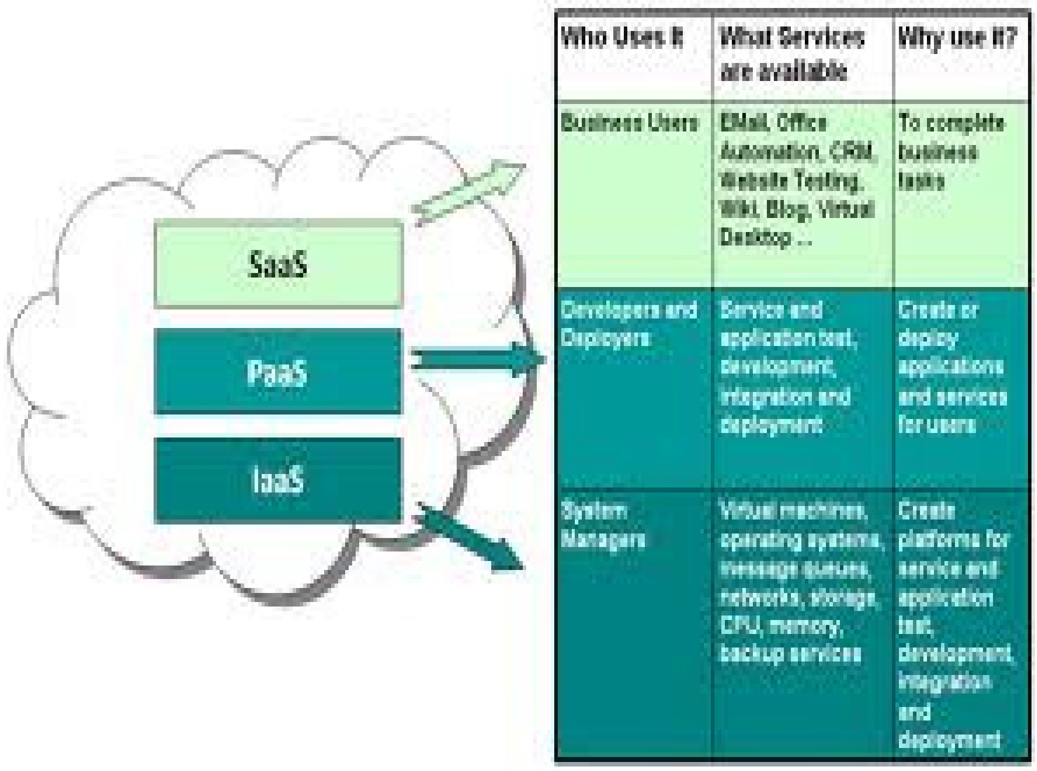
The capability provided to the consumer is to deploy onto the cloud infrastructure consumer created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure, but has control over the deployed applications and possibly application hosting environment configurations.

1.2.3 Cloud Infrastructure as a Service (Iaas)

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage

or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components.
Features and parts of IaaS include:

- Utility computing service and billing model.
- Automation of administrative tasks.
- Dynamic scaling.
- Desktop virtualization.
- Policy-based services.



Who Uses It	What Services are available	Why use it?
Business Users	EMail, Office Automation, CRM, Website Testing, Wiki, Blog, Virtual Desktop ...	To complete business tasks
Developers and Deployers	Service and application test, development, integration and deployment	Create or deploy applications and services for users
System Managers	Virtual machines, operating systems, message queues, networks, storage, CPU, memory, backup services	Create platforms for service and application test, development, integration and deployment

Fig 1: cloud Service Model

1.3Virtualization

It is a very useful concept in context of cloud systems. Virtualisation means ”something which isn’t real”, but gives all the facilities of a real. It is the software implementation of a computer which will execute different programs like a real machine. Virtualisation is related to cloud, because using virtualisation an end user can use differentservices of a cloud. The remote datacentre will provide different services in a fully or partial virtualised manner.

Two types of virtualization are found in case of clouds as given in [19] :

Full virtualization
Para virtualization

1.3.1 Full Virtualization

In case of full virtualisation a complete installation of one machine is done on the another machine. It will result in a virtual machine which will have all the software that are present in the actual server.

Here the remote datacentre delivers the services in a fully virtualised manner. Full virtualization has been successful for several purposes as pointed out in [19]:

a. Sharing a computer system among multiple users
b. Isolating users from each other and from the control program
c. Emulating hardware on another machine

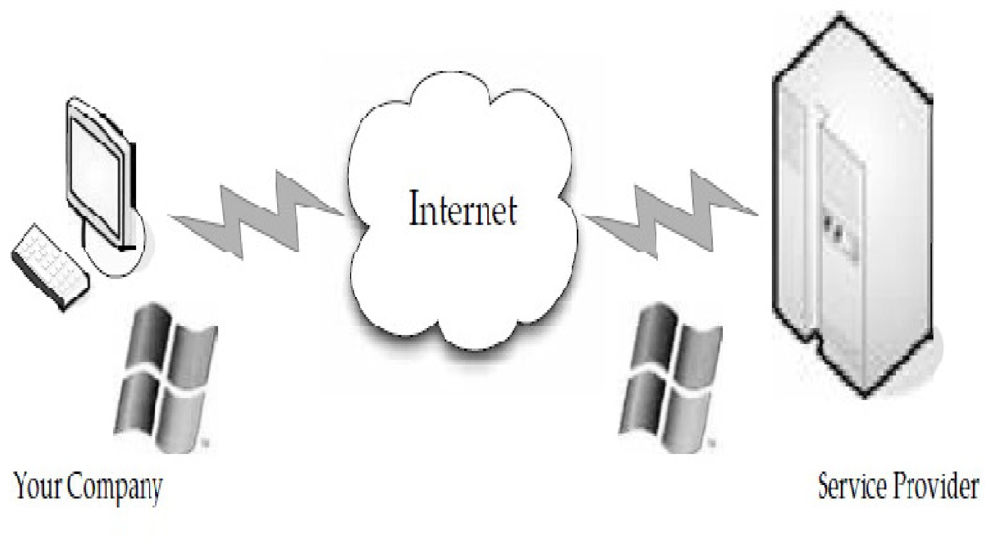


Fig2; Full Virtualization (adopted from [19]).

1.3.2 Para virtualization

In paravitualisation, the hardware allows multiple operating systems to run on singlemachine by efficient use of system resources such as memory and processor. e.g. VMwaresoftware. Here all the services are not fully available, rather the services are providedpartially.

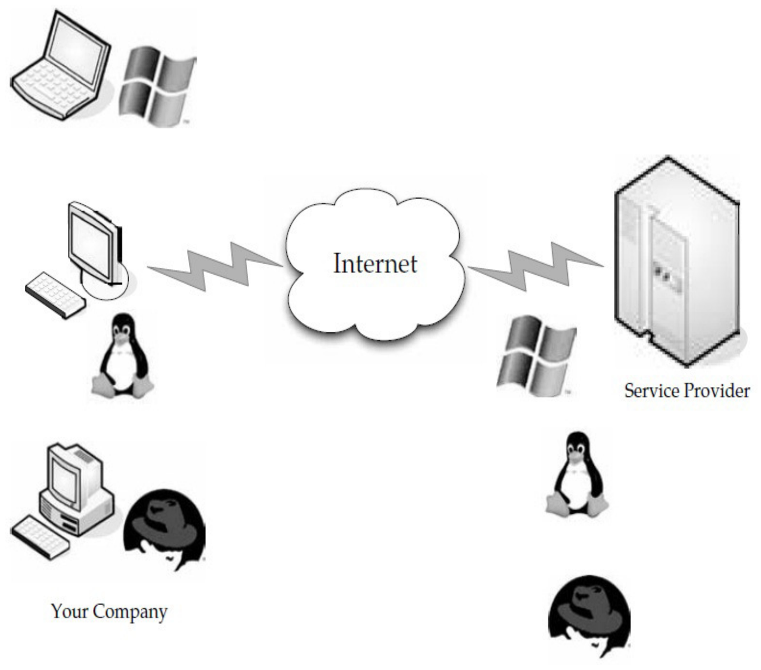


Fig3: Para virtualization (adopted from [19]).

Para virtualization has the following advantages as given in [19]:

Disaster recovery: In the event of a system failure, guest instances are moved toanother hardware until the machine is repaired or replaced.

Migration: As the hardware can be replaced easily, hence migrating or moving thedifferent parts of a new machine is faster and easier.

Capacity management: In a virtualized environment, it is easier and faster to addmore hard drive capacity and processing power. As the system parts or hardware can be moved or replaced or repaired easily, capacity management is simple and easier.

1.4 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.

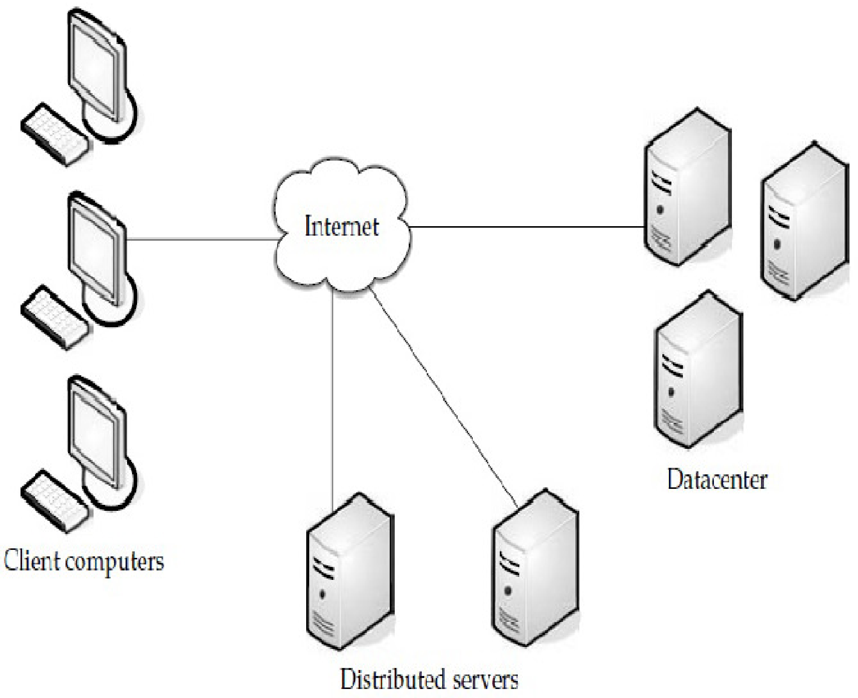


Fig4. Three components make up a cloud computing solution

1.4.1 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.

1.4.2 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 allowsmultiple instances of virtual server applications.

1.4.3 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.

1.5 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.

1.5.1 Goals of Load balancing

As given in [4], 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.

1.5.2 Types of Load balancing algorithms

Depending on who initiated the process, load balancing algorithms can be of three categories as given in [15]:

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 [15]:

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

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

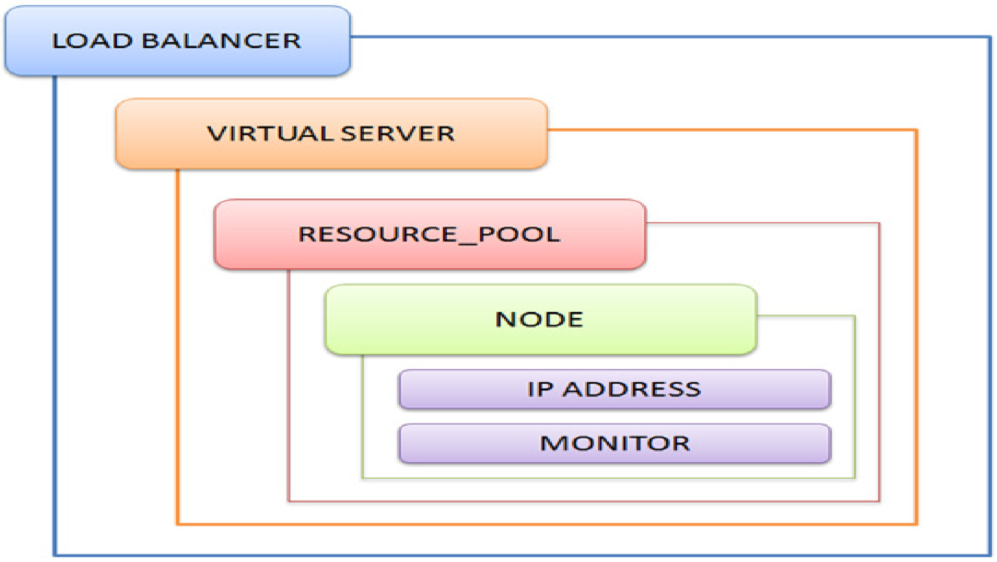


Fig 5: Load Balancer

1.6 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 non-cooperative [4]. 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.

1.6.1 Policies or Strategies in dynamic load balancing

In non-distributed type, either one node or a group of nodes do the task of loadbalancing. 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 [15].

Centralized dynamic load balancing takes fewer messages to reach a decision, as thenumber 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.

1.6.2 Policies or Strategies in dynamic load balancing

There are 4 policies [15]:

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.

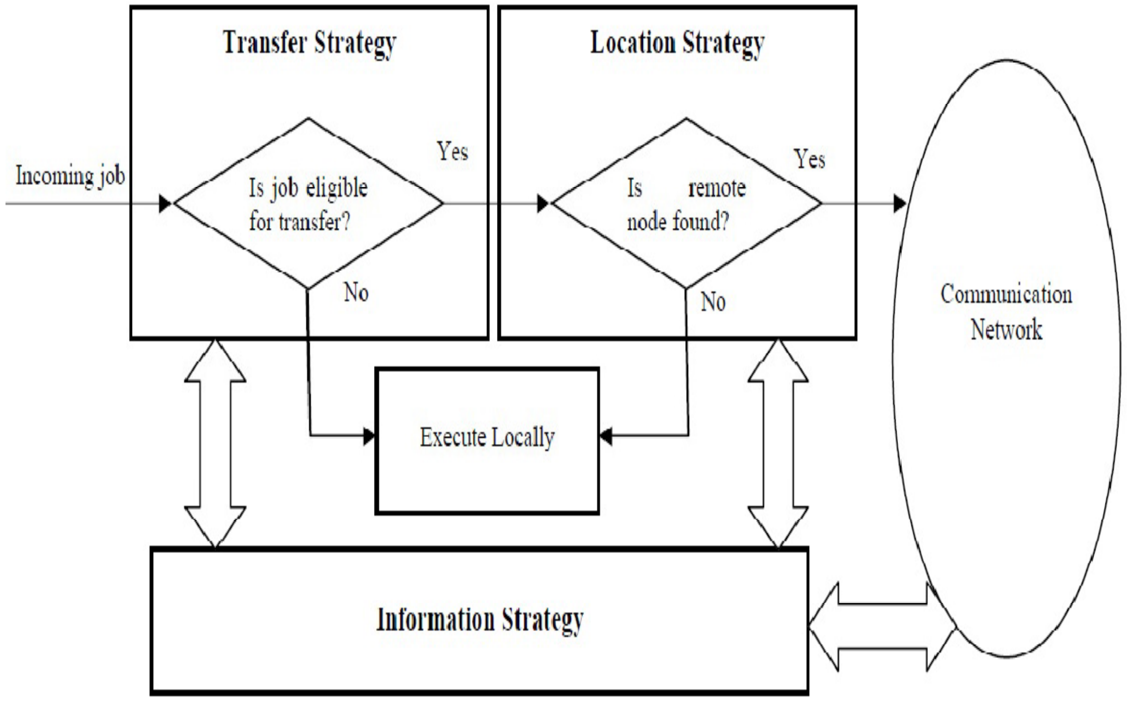


Fig 6: Interaction among components of a dynamic load balancing algorithm (adopted from [15])

1.7 Ant Colony Optimization

Individual ants are behaviorally much unsophisticated insects. They have a very limited memory and exhibit individual behavior that appears to have a large random component. Acting as a collective however, ants manage to perform a variety of complicated tasks with great reliability and consistency.

Although this is essentially self-organization rather than learning, ants have to cope with a phenomenon that looks very much like overtraining in reinforcement learning techniques.

The complex social behaviors of ants have been much studied by science, and computer scientists are now finding that these behavior patterns can provide models for solving difficult combinatorial optimization problems. The attempt to develop algorithms inspired by one aspect of ant behavior, the ability to find what computer scientists would call shortest paths, has become the field of ant colony optimization (ACO), the most successful and widely recognized algorithmic technique based on ant behavior.

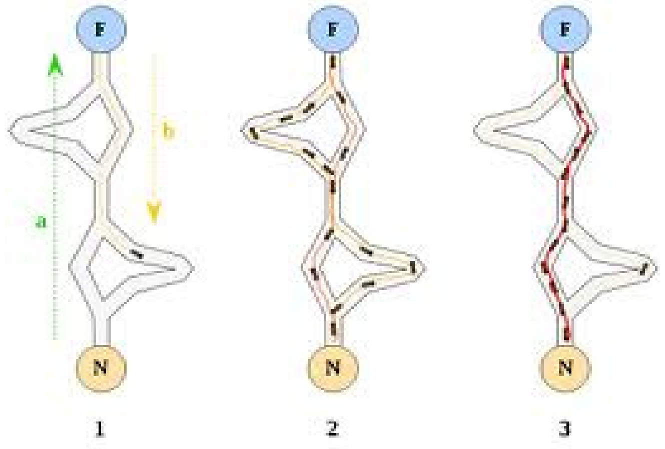


Fig 7: Ant colony Optimization

1.7.1. Basic principles of trail laying

Depending on the species, ants may lay pheromone trails when travelling from the nest to food, or from food to the nest, or when travelling in either direction. They also follow these trails with a fidelity which is a function of the trail strength, among other variables. Ants drop pheromones as they walk by stopping briefly and touching their gesture, which carries the pheromone secreting gland, on the ground. The strength of the trail they lay is a function of the rate at which they make deposits, and the amount per deposit. Since pheromones evaporate and diffuse away, the strength of the trail when it is encountered by another ant is a function of the original strength, and the time since the trail was laid. Most trails consist of several superimposed trails from many different ants, which may have been laid at different times; it is the composite trail strength which is sensed by the ants. The principles applied by ants in their search for food are best explained by an example as given in [7].

Related Work

Many researchers have invented different techniques for load balancing and distributing the workload in a cloud environment. Many tools has been developed like c-meter, Amizone EC-2, GrenchMark etc.[18]

Many methods likehoneybee Foraging system,had been took place in the area of research. These methods also inspired by swarm intelligence. This algorithm is derived from the behaviour of honey bees for finding and reaping food. There is a class of bees called the forager bees which forage for food sources, uponfinding one, they come back to the beehive to advertise this using a dance called waggle dance. The display of this dance, gives the idea of the quality or quantity of food and also itsdistance from the beehive. Scout bees then follow the foragers to the location of food and then began to reap it. They then return to the beehive and do a waggle dance, which gives an idea of how muchfood is left and hence results in more exploitation or abandonment of the food source. In case of load balancing, as the webservers demand increases or decreases, the services are assigned dynamically to regulate the changing demands of the user. The servers are grouped under virtual servers (VS), each VS having its own virtual service queues. Each server processing a request from its queue calculates a profit or reward, which is analogous to the quality that the bees show in their waggle dance. One measure of this reward can be the amount of time that the CPU spends on the processing of a request. The dance floor in case of honey bees is analogous to an advert board here. This board is also used to advertise the profit of the entire colony. Each of the servers takes the role of either a forager or a scout. The server after processing a request can post their profit on the advert boards with a probability.[16]

C-Meter is a portable, extensible and easy-to-use framework for generating and submitting both real and synthetic workloads to analyse the performance of cloud computing environments. It is designed as an extension to GrenchMark. It is portable in the sense that it is implemented in Python, which is a platform-independent programming language. It is extensible in the sense that it can be extended to interact with many cloud computing environments and it can also be extended with different scheduling algorithms.

GrenchMark is a framework for generating and submitting synthetic or real workloads to grid computing environments. Over the seven three years, GrenchMark has been used in over 25 testing scenarios in grids (e.g., Globus based), in peer-to-peer systems (e.g., BitTorrent-based), and in heterogeneous computing environments (e.g., Condor-based).

Amazon EC2 (Elastic Compute Cloud) is a web service that opens Amazon's cloud computing infrastructure to its users [5]. It is elastic in the sense that it enables the applications to adapt themselves to their computational requirements either by launching new virtual machines or by terminating virtual machines which are running. EC2 uses its own image format called AMI (Amazon Machine Image) for virtual machine images allowing the users to create their own virtual computing environments containing their software, libraries and other configuration items.

Proposed Work

Ant based control system was designed to solve the load balancing in cloud environment.Each node in the network was configured with

1) Capacity that accommodates a certain.
2) Probability of being a destination.
3) Pheromone (or probabilistic routing) table.

Each row in the pheromone table represents the routing preference for each destination, and each column represents the probability of choosing a neighbor as the next hop. Ants are launched from a node with a random destination.

In this approach, incoming ants update the entries the pheromone table of a node. For instance, an ant traveling from (source) to (destination) will update the corresponding entry in the pheromone table in. Consequently, the updated routing information in can only influences the routing ants and calls that have as their destination. However, for asymmetric networks, the costs from to and from to may be different. Hence, In thisapproach for updating pheromone is only appropriate for routing in symmetric networks.

If an ant is at a choice point when there is no pheromone, it makes a random decision However, when only pheromone from its own colony is present there is a *higher* probability that it will choose the path with the higher concentration of its own pheromone type. In addition, due to repulsion, an ant is *lesslikely* to prefer paths with (higher concentration of) pheromone from other colonies. Moreover, it is reminded that the degrees of attraction and repulsion are determined by two weighting parameters.

Procedure ACO Optimization

```
Initialize Variables;
Initialize Pheromone on the trail selected by
GJAP;
While (Value of Timer<$T_l$) do
Ants Construct Solutions;
Xnew=min{$f_{obj}(P_k)$|k=1,2,…K};
If Xnew<X then X=Xnew;
            Pheromone Update;
End
End
```

Fig 8: Psedo Code of ACO[17]

Adopting the problem-solving paradigm of ACO, this example illustrates the use of two sets of mobile agents (that act as routing packets) for establishing call connections in a circuit- switched network.To establish connections between gateways 1 and 3, the two groups of mobile agents construct, manipulate and consult their own routing tables. In ACO, 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. Even though the two groups of mobile agents (MAG1 and MAG2) may have their own routing preferences, they also take into consideration the routing preferences of the other group. (While the routing preferences of ants are recorded in their pheromone tables, the routing preferences of mobile agents are stored in their routing tables).

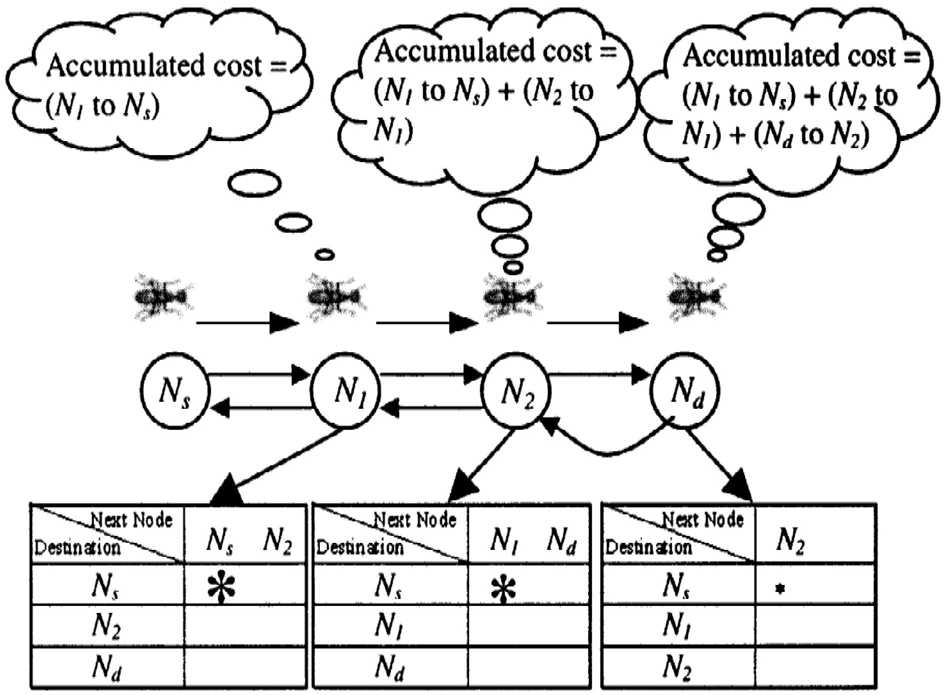

Accumulated cost = (N_l to N_s)
Accumulated cost = (N_l to N_s) + (N_2 to N_l)
Accumulated cost = (N_l to N_s) + (N_2 to N_l) + (N_d to N_2)
N_s
N_l
N_2
N_d
Next Node
Destination

Fig 9: Ant Making Decisions

In constructing its routing table,MAG1 (respectively,MAG2) consults the routing table of MAG2 (respectively, MAG1) so as toavoid routing packets to those paths that are highly preferred by the other group. Doing so *increases the likelihood* that two different connections between gateways 1 and 3 may be established.

This increases the chance of distributing data traffic between gateways 1 and 3 between the two connections and. In Fig. 16, for the same destination Gateway3, MAG1 is *more likely* to move along whereas MAG2 is *morelikely* to move along . By adopting the MACO approach, it may be possible to *reduce the likelihood* that all mobile agents establish connections using *only* the optimal path. If MAG2

selects the optimal path , the idea of repulsion may *increasethe probability* that MAG1 will select an alternative to. The advantage of using MACO in circuit-switched routing is that it is *more likely* to establish connections through multiple paths to help balance the load but does not increase the routing overhead. An on-going work implements *some* of the ideas of ACO (first proposed in [48]) in a test bed, and preliminary empirical results seem to suggest that using the same number of mobile agents (routing packets), it is *more likely* that ACO can establish connections through multiple paths while traditional

ACO is *more likely* to establish connections through the optimal path.

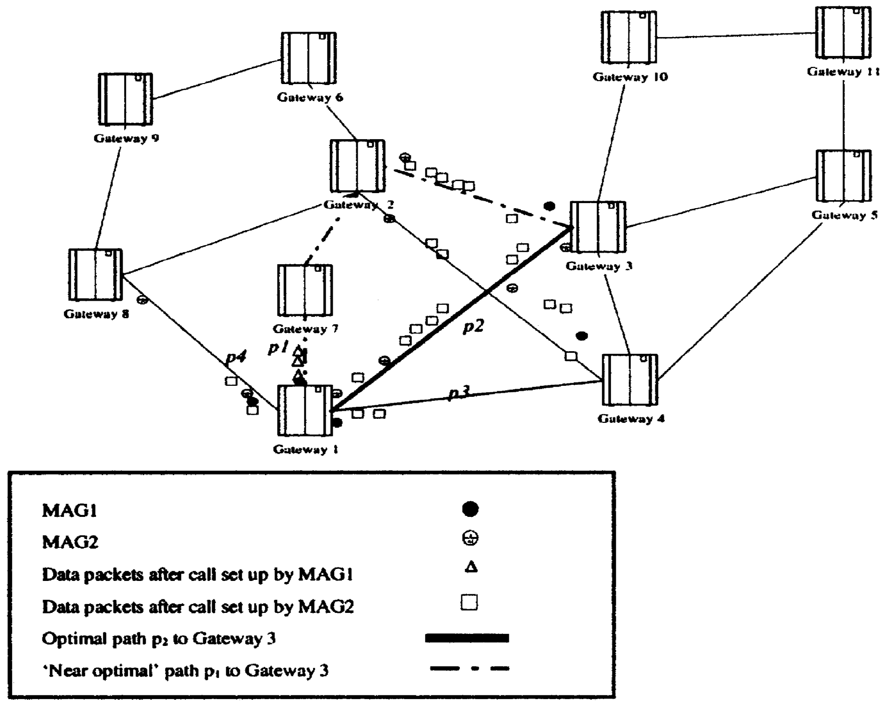

Gateway 9
Gateway 6
Gateway 10
Gateway 11
Gateway 2
Gateway 3
Gateway 5
Gateway 8
Gateway 7
Gateway 4
Gateway 1
p1
p2
p3
p4
MAG1 ●
MAG2 ⊜
Data packets after call set up by MAG1 △
Data packets after call set up by MAG2 □
Optimal path p_2 to Gateway 3
'Near optimal' path p_1 to Gateway 3

Fig 10: Connection Establishment

In the case of load balancing in cloud environment, as the web server demands increases or decreases, the Services are assigned dynamically to regulate the changing demand of the user. The servers are grouped under Virtual Server(VS),each VS having its own virtual service queues. Each server processing a request from its queue calculates a profit or reward, which is analogous to the amount of pheromone which is to be laid by the ants. Each individual server is an analogy to individual mobile agents. The server after processing a request can post their profit in the pheromone table, which is to be maintained over each server. A server can choose a queue of a virtual server by a probability of px. A server serving a request, calculating its profit and compare it with the colony profit and then set the px. If the profit is high, then the server stays at the current server, posting an advertisement for it by probability pr. If it was low, then the server returns to the pheromone table.

Pheromone tables

We replaced the routing tables in the network nodes by tables of probabilities, which we will call 'pheromone tables', as the pheromone strengths are represented by these probabilities. Every node has a pheromone table for every possible destination in the network, and each table has an entry for every neighbour. For example, a node with four neighbours in a 30-node network has 29 pheromone tables with four entries each. One could say that an *n*-nodenetwork uses *n* different kinds of pheromones. The entries in the tables are the probabilities which influence the ants′ selection of the next node on theway totheir destination node. Figure 4 shows a possible network configuration and apheromone table. For example, ants travelling from node 1 to node 3 have a 0.49probability of choosing node 2 as their next node, and 0.51 of choosing node 4 'Pheromone laying' is represented by 'updating probabilities'.

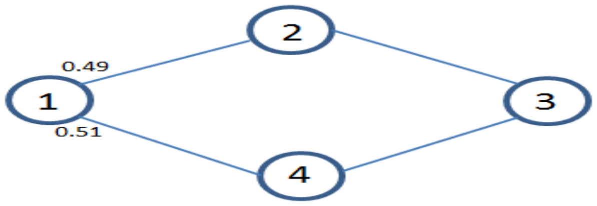


Figure 11Using ants for network management.

The three pheromone tables for node 1:

	Next Node 2	Next Node 4
Destination node 2	0.95	0.05
3	0.49	0.52
4	0.05	0.95

At every time step during the simulation, ants can be launched from any node inthe network. Each ant has a random destination node. Ants move from node tonode, selecting the next node to move to according to the probabilities in thepheromone tables for their destination node. Arriving at a node, they update theprobabilities of that node's pheromone table entries corresponding to their *source*node i.e. ants lay the kind of pheromone associated with the node they werelaunched from. They alter the table to increase the probability pointing to theirprevious node.

The method used to update the probabilities is quite simple: when an ant arrivesat a node, the entry in the pheromone table corresponding to the node fromwhich the ant has just come is increased according to the formula:

$$P_{new} = \frac{(P_{old} + \Delta P)}{(1+\Delta P)}$$

Here p is the new probability and ΔP is the probability(or pheromone) increases.
The other entries in the table of this nodes are decreased according to:

$$P = \frac{P_{old}}{(1+\Delta P)}$$

Since the new values sum to 1, they can again be interpreted as probabilities.

Note that a probability can be reduced only by the operation of normalisation following an increase in another cell in the table; since the reduction is achievedby multiplying by a factor less than one, the probability can approach zero if the other cell or cells are increased many times, but will never reach it.

Active Clustering

Active Clustering works on the principle of grouping similar nodes together and working on these groups. The process involved is:

1. A node initiates the process and selects another node called the matchmaker node from its neighbours satisfying the criteria that it should be of a different type than the former one.

2.The so called matchmaker node then forms a connection between a neighbour of it which is of the same type as the initial node.

3. The matchmaker node then detaches the connection between itself and the initial node.

Conclusion And Future Scope

Till now we have discussed on basic concepts of Cloud Computing and Load balancing.In addition to that, the load balancing technique that is based on Swarm intelligence has been discussed. We have discussed how the mobile agents can balance the load of a cloud using the concept of Ant colony Optimization. The limitation of this technique is that it will be more efficient if we form cluster in our cloud.So, the research work can be proceeded to implement the total solution of load balancing in a complete cloud environment.

Our objective for this paper is to develop an effective load balancing algorithm using Ant colony optimization technique to maximize or minimize different performance parameters like CPU load, Memory capacity, Delay or network load for the clouds of different sizes.

In this paper, a heuristic algorithm based on ant colony optimization has been proposed to initiate the service load distribution under cloud computing architecture. The pheromone update mechanism has been proved as a efficient and effective tool to balance the load. This modification supports to minimize the make span of the cloud computing based services and portability of servicing the request also has beenconverged using the ant colony optimization technique. This technique does not consider the fault tolerance issues. Researchers can proceed to include the fault tolerance issues in their future researches.

References

[1] Wayne Jansen, Timothy Grance, “Guidelines on Security and Privacy in Public Cloud Computing”, National Institute of Standards and Technology Gaithersburg, January 2011.

[2] Jeep Ruiter, MartijnWarnier, “Privacy Regulations for Cloud Computing”, Faculty of Sciences, VU University Amsterdam

[3] DanchoDanchev,"Building and Implementing a successful Information Security Policy" windowsecurity.com- Windows Security Resources for IT admins.

[4] David Escalante and Andrew J. Korty, Cloud Services: Policy and Assessment, *EDUCAUSE Review,* vol. 46, no. 4 (July/August 2011)

[5] Richard N. Katz, "Looking at Clouds from All Sides Now", *EDUCAUSE Review,* vol. 45, no. 3 (May/June 2010): 32-45

[6] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition 2010.

[7] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[8] Mladen A. Vouk, Cloud Computing Issues, Research and Implementations, Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, 2008, June 23-26.

[9] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[10] ibm.com/press/us/en/pressrelease/22613.wss

[11] http://www.amazon.com/gp/browse.html?node=201590011

[12] Martin Randles, EnasOdat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab, "A Comparative Experiment in Distributed Load Balancing", 2009 Second International Conference on Developments in eSystems Engineering.

[13] Peter S. Pacheco, "Parallel Programming with MPI", Morgan Kaufmann Publishers Edition 2008

[14] MequanintMoges, Thomas G.Robertazzi, "Wireless Sensor Networks: Scheduling for Measurement and Data Reporting", August 31, 2005

[15] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security,VOL.10 No.6, June 2010.

[16] Martin Randles, EnasOdat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab, "A Comparative Experiment in Distributed Load Balancing", 2009 Second International Conference on Developments in eSystems Engineering.

[17] Fourth International Conference on Semantics, Knowledge and Grid" Load Balancing in Non-dedicated Grids Using Ant Colony Optimization".

[18] 9th IEEE/ACM International Symposium on Cluster Computing and the Grid.,2011

[19] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A PracticalApproach, TATA McGRAW-HILL Edition 2010.