

Reduction of Makespan Using Ant Colony Optimization in Clouds

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

in

Computer Science and Applications

Submitted By

Anviti Pandey

(Roll No. 601303004)

Under the supervision of:

Dr. Rajesh Kumar

Associate Professor, CSED



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

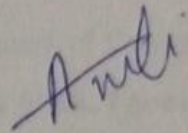
PATIALA – 147004

June 2015

CERTIFICATE

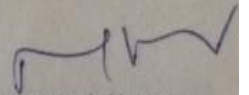
I hereby certify that the work which is being presented in the thesis entitled, "*Reduction of Makespan Using Ant Colony Optimization in Clouds*", in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Applications* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Rajesh Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.



Anviti Pandey

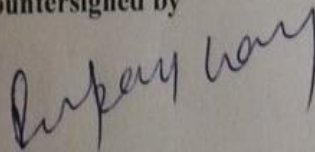
This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.



Dr. Rajesh Kumar

Associate Professor, CSED

Countersigned by



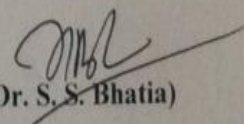
(Dr. Deepak Garg)

Head,

Computer Science and Engineering Department,

Thapar University,

Patiala



(Dr. S. S. Bhatia)

Dean (Academic Affairs),

Thapar University,

Patiala

Acknowledgement

Acknowledgements are passé. They provide us the opportunity to express our gratitude to the people which make such tasks to come out as a success. I take this opportunity to thank **Dr. Rajesh Kumar**, Associate Professor, Department of Computer Science and Engineering, for his immense support and help. His incessant and perpetual cooperation and encouragement helped me overcome hurdles at various stages. I am grateful to him and would remain indebted to him for inspiring me. I thank him for his constant supervision, motivation and suggestions, which have benefited me to accomplish this work.

I am also thankful to **Dr. Deepak Garg**, Head, Department of Computer Science and Engineering and **Dr. S. S. Bhatia**, Dean of Academic Affairs. I give my warm thanks to them for encouraging all of us, for the co-operation and their consent without which it would have been impossible to complete this task.

I would also like to thank my family for providing me resources, support and encouragement; this would not have been possible without you.

Last but definitely not the least I would like to thank my friends for helping me cope with everything. I thank them for standing beside me throughout and for never saying “no” when I needed a hand; thank you for everything.

- Anviti

Abstract

Cloud computing has become a modern age technology. Every other organization is moving its infrastructure to the Cloud. It is now becoming a necessity rather than being a luxury. Cloud computing services are now becoming a household name among the end users. In order to deploy these services without any hindrance it is important to handle task scheduling. Task scheduling on the Cloud not only helps achieve judicious use of resources but also help reduce number of machines involved in execution. The algorithm which has been proposed as a part of this work is based on Ant Colony Optimization algorithm. This heuristic based optimization algorithm has been inspired from the foraging behavior of ants. One of the main features of such algorithms is that with each iteration we get an improved result until we achieve an optimal solution to our problem. The main objective of the proposed algorithm is to reduce the makespan i.e. the total execution time of the tasks at hand. This will be achieved by making an optimal schedule which gives minimum execution time. This algorithm has been compared with other conventional scheduling algorithms like First Come First Serve (FCFS) algorithm and Round Robin (RR) algorithm. The obtained results show that the algorithm based on Ant Colony Optimization gives better results than FCFS and RR algorithms. The algorithm has been implemented using an open source platform called CloudSim. It helps simulate and model Cloud computing scenarios to help in getting better results.

Table of Contents

Certificate	i
Acknowledgement	ii
Abstract	iii
List of Figure	vi
List of Tables	vii
1. Introduction	1
1.1 Service Models in Cloud Computing	4
1.1.1 Infrastructure as a Service	4
1.1.2 Platform as a Service	5
1.1.3 Software as a Service	7
1.2 Deployment Models in Cloud Computing	9
1.2.1 Public Cloud	9
1.2.2 Private Cloud	10
1.2.3 Hybrid Cloud	11
1.2.4 Community Cloud	12
1.3 Virtualization	13
1.4 Resource Scheduling	15
1.5 Ant Colony Optimization	17
2. Literature Review	21
2.1 Cloud Computing	21
2.2 Resource Scheduling Techniques	22

2.3 Evolutionary Algorithms and Optimization Techniques	27
3. Problem Statement	32
4. Implementation	34
4.1 Probabilistic Rule	35
4.2 Updating the Pheromone Trail	36
4.3 Proposed Algorithm	37
4.4 Simulation Tool : CloudSim	38
5. Experimental Results	40
6. Conclusion and Future Scope	44
References	45
Video Reference	48

List of Figures

1.1 Various Advances in Technology Leading to the Development of Cloud Computing	2
1.2 Service Models in Cloud Computing	6
1.3 Deployment Models in Cloud Computing	10
1.4 Cloud Usage Scenario	13
1.5 Behavior of Ants which Inspired the Development of ACO	
(a) Forward Feedback (b) Backward Feedback	18
1.6 Ant Colony Optimization	16
4.1 A Complete Bi-partite Graph between Tasks and Virtual Machines	30
4.2 Layered Cloudsim Architecture	25
5.1 Snapshot for Ant Colony Optimization	37
5.2 (a) Snapshot for Round Robin	37
5.2 (b) Snapshot for First come First Serve	38
5.3 Values of Makespan for Different Number of VMs in Case of ACO, FCFS and RR	38
5.4 Values of Makespan for Different Values of Beta	39
5.5 Values of Makespan for Different Number of Tasks in Case of ACO, FCFS and RR	39

List of Tables

5.1 Setting the parameters for simulation	36
---	----

Chapter 1

Introduction

Over the last few years, many IT professionals, have started talking about a new paradigm in the field of computer science called Cloud computing. Cloud computing has been evolved through many phases and derived many of its computing features from various existing technologies like grid computing, utility computing etc. Grid computing is based on the concept of linking a number of systems to increase the availability and scalability. Utility Computing grows up with the standard web 2.0 for compute services in which resources such as storage, infrastructure, computing power and applications etc. are measured on the pay-per-use basis and utility computing acts as the base of Cloud computing.

Cloud computing was defined differently according to different professionals according to their understanding of what it has to offer. Buyya et al. [1] have defined it as distributed and parallel system of computation which consists of a set of computers connected to each other. These are virtualized computers which are provisioned dynamically. It can be found to have a collaboration of one or more resources which are dependent on the service level agreements (SLAs) established through abstraction between the two parties.

The NIST (National Institute of Standards and Technology) has its own definition of Cloud computing which illustrates it as a framework which works on pay as you go basis, for facilitating easily available, on-demand network access to a shared pool of computing resources which can be configured and provisioned in a short span of time. Some of these resources are networks, storage capacity, applications, services and many more. They require very less effort for managing or service provider interaction for release. [2]0

In other words this type of computing makes use of remotely placed servers kept in data centers, which have high security, for storage of data and its management, so that organizations don't have to buy and maintain their IT solutions on site. It provides a more convenient and easier way to access the storage, databases, servers and a wide collection of application services over the Internet. The Cloud providers

own and maintain the hardware, connected over a network, required for these application services, meanwhile the end users provision and utilize what they require through a web application. Another feature of Cloud computing is that it allows end users and businesses to use the applications without installing and provide access to their personal data at any system which has access to the internet. This technology increases efficiency even more by centralizing bandwidth and processing. It helps access data anywhere, anytime. It maximizes the efficiency of shared resources. It reduces infrastructure costs and makes resource reallocation easy. It provides improved manageability with less maintenance.

As discussed before, Cloud computing also makes use of concepts from utility computing. It utilizes it to provide metrics for the services used. Cloud computing has been advancing by addressing the QoS (quality of service) and issues based on reliability. Cloud computing delivers the technologies and tools to build data and develop intensive parallel applications with much more affordable prices compared to conventional parallel computation techniques. A Cloud should have the following characteristics:

- Payment should be done as per the usage of the resources
- Should be able to scale and should provide the illusion of unlimited resources.
- Should have an interface which makes getting tailor-made services easy
- Should have virtualized resources which are abstract to the consumer

Cloud computing infrastructure offers a very broad range of software services along with raw computing and storage. Apart from these features, Cloud computing provides developers with development tools to create scalable application on these servers without any hindrances. The ultimate goal is allowing customers to run their everyday IT infrastructure “on the Cloud.”

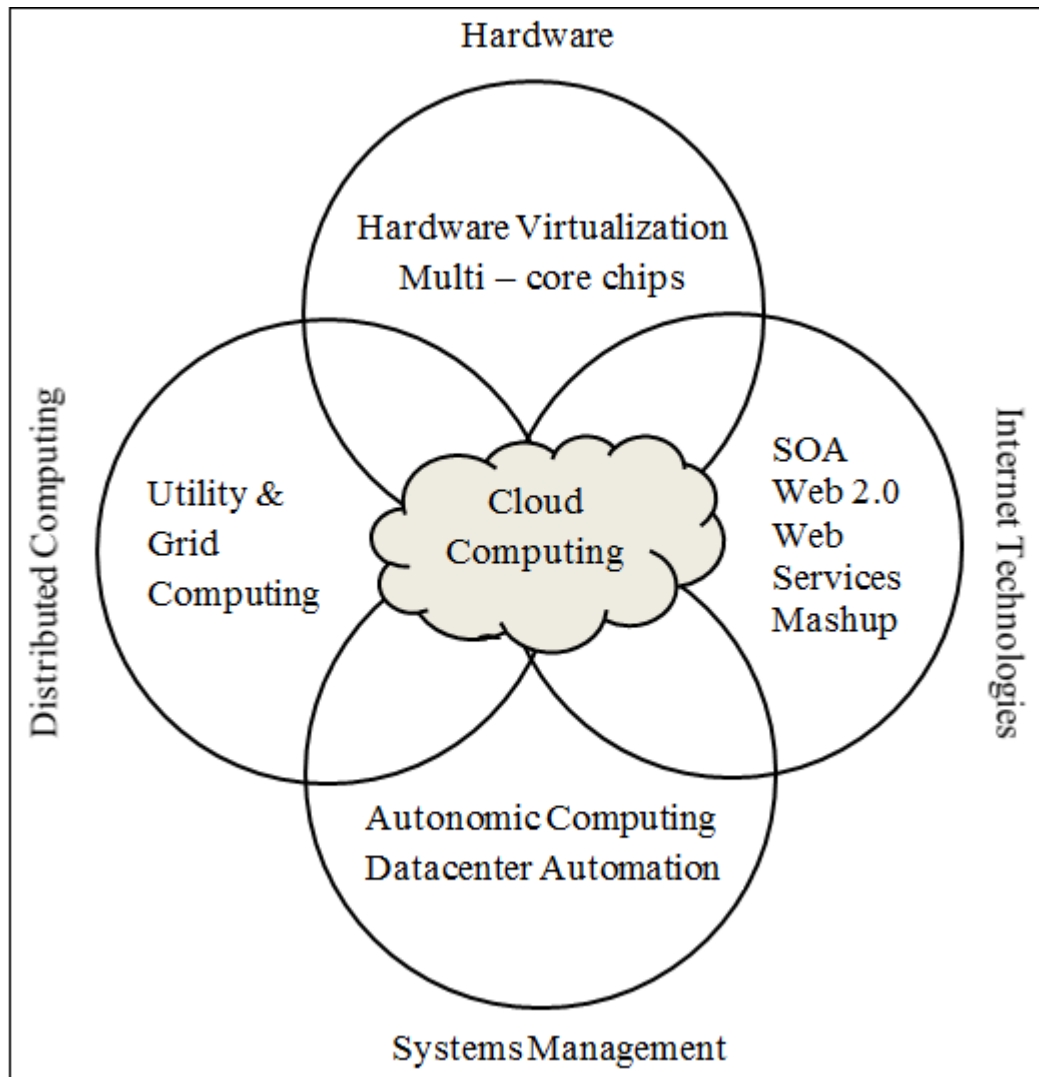


Fig 1.1: Various Advances in Technology Leading to the Development of Cloud Computing [1]

There are several benefits of Cloud computing which are described below in brief:

- Cloud service provider can be accessed with the help of fair internet connection.
- On the user side, by decrease in the investment and running costs and on the provider side, by providing higher productivity in arranging infrastructure services with high survivability and flexibility.
- Users can demand for the computational power, storage and other infrastructure according to their need according to pay per use model.

- Resources are utilized properly because whenever users don't need a resource, they return it to the Cloud provider. So in this way elasticity and flexibility can be increased.
- Cloud computing provides infrastructure-as-a-service on demand for user. So there is no permanent need to purchase the infrastructure for IT. The user can purchase it from Cloud provider whenever needed.
- The consumer generally has no information about the locality of the service provider. Thus the provider serves multiple consumers by assigning resources dynamically and virtually.
- There is no need to provide unnecessary detail about the core business component to the user.
- Non IT user can contact to IT service provider for their business activity needs.

1.1 Service models in Cloud computing

There are basically three main service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Different services are provided by Cloud providers. These services are described in Figure 1.2. In general these services are known as XaaS (Anything as a Service or Everything as a Service). The service provider, provides these services as “black boxes” and only tells about how can these services be offered and how can these services be accessed but the internal working detail of the service is hidden from the end user. There are certain fundamental models according to which Cloud providers give their services. They are as follows:

1.1.1 Infrastructure as a Service

It is defined as providing virtualized resources like storage, communication and computation on demand. A Cloud infrastructure facilitates provisioning of servers on demand which run several types of operating system and provides raw computing. Instead of buying software, servers, data centers, or equipment for networking, clients buy resources as a completely outsourced service on demand. IaaS distributes resources as a service. In other words, IaaS is the delivery of computer infrastructure because of the fluctuations in market demand of hardware and software resources. To fulfill constantly changing demand, infrastructure is provided as a service on Cloud. Customer can order as many resources as they need and return them back to the provider when their goal is achieved. One of the examples for IaaS is Amazon Elastic Compute Cloud (EC2) which provides services for computing power and Amazon Simple Storage Service (S3) for storage capacity. IaaS is used as follows:

- It lowers and matches the power consumption according to demand.
- It reduces cost by providing hardware.
- It gives higher returns in terms of higher resource utilization.
- It can be utilized by upcoming organizations without putting much money for the hardware.
- It can be used when the businesses of an organization are expanding rapidly and therefore scaling the available hardware would be difficult.
- It comes in play when organization is being pressurized into moving to operating expenditure and limiting the capital expenditure.
- It can also be used for meeting specific line of business, temporary and trial infrastructural needs.

- It has a utility model used for pricing as it generates variable cost according to usage.
- It also allows dynamic scaling.
- It generally includes multiple users at a time on a single piece of hardware.
- It is used when demand may vary at any point of time that means it is volatile in nature. In such cases there may be significant spikes and significant reductions in terms of demand on the infrastructure any time.

1.1.2 Platform as a Service

It is another approach to make a Cloud easily programmable by offering a higher level of abstraction to these resources. A Cloud platform provides a development platform on which developers can build and deploy applications without necessarily needing to know how much processing capacity or how much storage is utilized by that application. Also, specialized services like payments, data access and authentication are provided to help develop new applications along with multiple programming models. In other words, PaaS is a platform for computing which allows the quick and easy creation of web applications without worrying about maintenance or buying the software or the infrastructure that is underneath it. PaaS provides a complete set of tools and technologies which are required to develop, test, deploy and operate SaaS applications. This application development is mostly done with the help of web browsers itself. PaaS examples include Microsoft Azure Services Platform, Google App Engine and Amazon S3 etc.

PaaS brings various benefits to the software development world as follows:

- It has a pay per use model for developing, testing and operating SaaS environment.
- Developers just need to focus on application code.

- It is an environment made available over the web, provided for the development of software.
- It provides the facility to test, develop, host, deploy and maintain software in the same integrated development environment.
- It provides all the different types of facilities which are required to complete the development process for the application.
- It helps to create, modify, test and deploy different UI scenarios by providing web based user interface creation tools.
- It provides a multi-tenant architecture in which a single instance of software serves multiple users while running on a single server.
- It has built in capability to scale deployed software for load balancing and also in case of failover.
- It facilitates merging the software to the web services and databases.
- It facilitates a support system for development teams' collaboration.
- Some PaaS solutions help in planning of projects and communication tools for development teams.
- It also provides tools to handle billing and subscription management.
- It is useful especially where external parties need to get into the loop of the development process and also while multiple developers would be working together on a single development project.
- It is useful where developers wish to automate testing and deployment services.

1.1.3 Software as a Service

Applications are found to be on the top of the development stack. These services can be made available to end users via web portals. This is the reason why consumers are shifting from locally installed computer programs to online software services that offer the same functionality. Traditional desktop applications can now be accessed as a service on the Internet. In other words, SaaS can be defined as software being deployed on the Internet. It is a multi-tenant platform. In this, multiple clients get the services from common resources and same infrastructure for both the application and underlying database is provided simultaneously to multiple customers. Individuals don't need to buy the proprietary rights of software they want to use. They can borrow that software from Cloud providers according to their need. Some of the biggest name in market that provide SaaS are Salesforce.com providing its Customer Relationship Management (CRM) application as service, Google web based calendar application, and Microsoft online SharePoint etc. These services are charged as per the usage on monthly basis. SaaS is helpful as follows:

- It eliminates licensing and version compatibility.
- It gives reduced upfront cost and hardware cost.
- This model of delivering applications alleviates the burden of software maintenance for customers and simplifies development and testing for providers.
- With SaaS, a Cloud provider makes an application available to the end-users either as a service on demand, as in a “pay-as-you-go” model.
- SaaS provides web access to commercial software which is managed and delivered according to “one-to-many” model.
- The end users do not have to bother about upgrades and maintenance. APIs allow integration between different parts of the software.

- SaaS has grown as a method of delivering technology and therefore is being adapted by most of the organization.

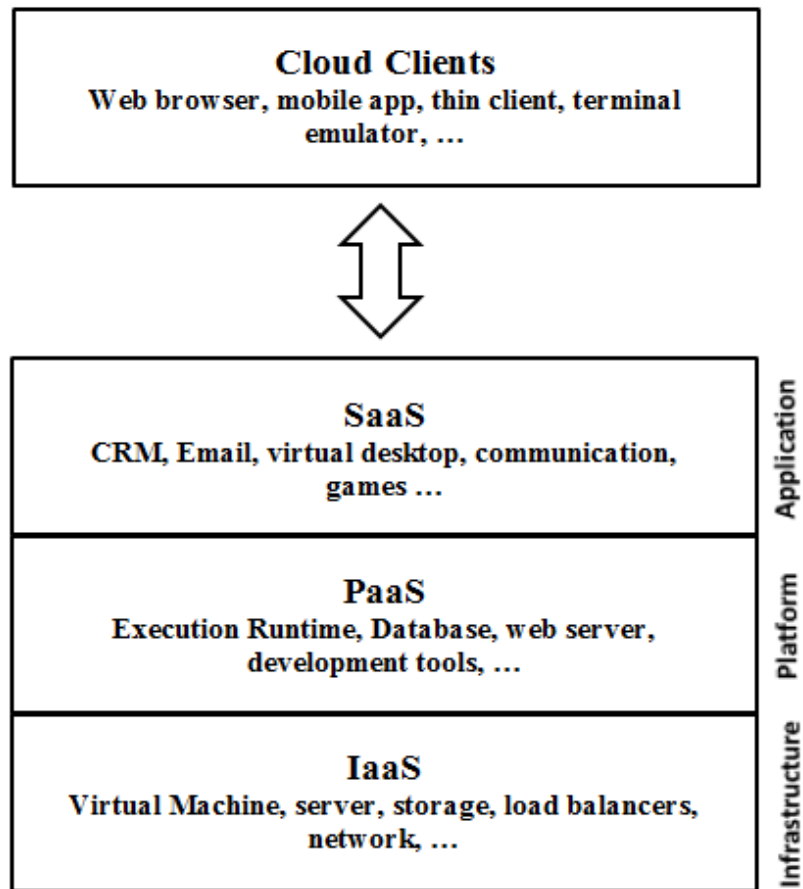


Fig 1.2: Service Models in Cloud Computing [3]

1.2 Deployment Models in Cloud Computing

1.2.1 Public Cloud

It is defined as the service of the Cloud provided to the general public based on a pay as you go model. Public Clouds are created with a primitive objective of offering limitless storage space and improved bandwidth through the Internet to various organizations. Public Clouds are possessed, hosted, operated and managed by service providers from a third-party. A public Cloud looks after all kinds of requirements from small, medium or big sized businesses. A public Cloud is the simplest to setup as it is free from any problems of the hardware, bandwidth expenses and application expenses at the subscribers' end. Organizations pay for only those resources which they are making use of. Consumers pay for the services on a monthly basis. Public Cloud operates on the main fundamental of storage demand scalability, which means it needs no hardware devices.

The actual Cloud hosting scenario is portrayed by the public Cloud deployment model as it provides infrastructure as well as services to various clients. There can be more than one way to provide this service. One is that the Cloud vendor does not charge for the service and the other is that the payment is made on a pay-per-use basis. It would be best to use this deployment model for those business requirements where it is essential to manage issues like load spiking and hosting of SaaS applications. Also when temporary infrastructure is to be utilized for the development and testing of applications public Clouds are the best choice. Public Clouds are used when applications have to be managed especially those which are made in use by many end users and which would otherwise need huge investments in the infrastructure for the businesses. This model helps in reducing operational costs and in bringing down capital expenditure. One of the examples of public Cloud is Google. Other examples of public Cloud include Amazon S3, EC2 and Google App Engine etc. The benefit of public Cloud is that no initial capital investment on infrastructure is required also no data transfer risk involved.

1.2.2 Private Cloud

Private Clouds are operated and managed within a single private organization. Sometimes a third party apart from organization manages the private Cloud in both on-premise and off-premise. Private Cloud is also known as the internal cloud. It is defined as an internal data centre of an organization or a business which is not made available to the general public. Many a times, a private Cloud is created by reframing or restructuring an infrastructure which already exists. This is achieved by virtualizing the resources. This facilitates consumers to communicate with the local data centre while getting all the benefits which a public Cloud has to offer, most significant of these are self-service interface, special access to the servers provisioned virtually, and metering-per-use along with billing.

Although this deployment model does not bring much benefits in terms of cost efficiency. The reason for this is that it is similar to buying, constructing and managing the infrastructure. However, it provides a significant benefit in terms of security. When organizations were initially adapting to the Cloud they were facing challenges and had concerns regarding data security. In this deployment model the hosting is constructed and should undergo maintenance for a particular client; this solves the security issues. The infrastructure may be kept on site or at a third party site. It is needed to host applications therefore it is quite useful. Securely accessible VPN are used to handle security issues.

Furthermore, for mission-critical applications downtime has to be considered in terms of internet availability, quality and performance. Hence, an on-premises private Cloud is preferred for hosting the application. This model has been adopted by various organizations to not only meet security reasons but also when data or applications are required to adapt to various governing standards. This may require data management for the privacy and audits that administer the corporation. For example, moving data to the Cloud may violate the norms in case of the healthcare and pharmaceutical industries. Similarly, for management and handling of data, different countries have different laws and regulations that may hinder the business if the Cloud

service is under different jurisdiction. The advantage of private Cloud is that it can be used to optimize the in-house resource utilization within the organization. It also provides data privacy for the security concerns. Further it needs minimum data transfer cost and is very reliable.

1.2.3 Hybrid Cloud

A hybrid Cloud is the outcome of a private Cloud being complemented with the computational capability of public Clouds. It helps in handling load spikes by using the approach of temporarily renting capacity. This sudden spike in the load is known as “Cloud-bursting”. It is a concept where when the users fall short of the computational entities in the internal data centre, they forward the additional workload to an external Cloud on an on-demand basis. Here the internal computing resource is the private Cloud and the external Cloud is typically a public cloud. The organization gets charged on a pay-per-use basis for utilizing the facility of the public cloud. When a deployment model has the capability to spread the load to the public cloud, it is said to be a Hybrid Cloud. But the functioning of hybrid Clouds is just not limited to Cloud bursting, rather it is much more than that.

While implementing a hybrid Cloud certain parameters have to be kept in mind, like security, performance, making enterprise-class services available, and more. The deployment model used for hybrid Clouds helps organizations to make use of applications which are secure and help in hosting the data on a private cloud. This has to be done while getting cost benefits by keeping shared data and applications on the public Cloud. This model is also deals with a scenario where the existing private Cloud infrastructure is not able to handle load spikes and requires a fall back option to support the load i.e. Cloud bursting. Hence, the Cloud can move around the workload between public hosting and private hosting without causing any hindrances for the end users. Many PaaS implementations have made their APIs as an open source application so that they can be further merged with internal application(s) which are hosted on a private Cloud, while maintaining all the security aspects. Force.com and Microsoft Azure are some examples of this model. In other words, two or more type of Clouds (public, private or community) are

combined together to form this type of Cloud known as hybrid Cloud. To optimize the resources and core competencies hybrid Clouds are used by several organizations as the private Cloud and for business purpose they used is as a public Cloud. The benefit of hybrid Cloud is that it is more flexible than public and private Cloud. It also provides more security over data than public Cloud. It provides on demand service expansion and compaction by provisioning and releasing the resources whenever needed

1.2.4 Community Cloud

Community Cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party, and either hosted internally or externally. The costs are spread over fewer users than a public Cloud (but more than a private Cloud), so only some of the cost savings potential of Cloud computing are realized. In the community deployment model, the Cloud infrastructure is shared by several organizations with the same policy and compliance considerations. This helps to further reduce costs as compared to a private Cloud, as it is shared by larger group. Various state-level government departments requiring access to the same data relating to the local population or information related to infrastructure, such as hospitals, roads, electrical stations, etc., can utilize a community Cloud to manage applications and data. In Cloud computing, investment in any deployment model should be made based on business requirements, the criticality of the application and the level of support required.

Community Clouds are established by sharing the specific resources, infrastructure, requirements and policies of same Cloud to a community or a business sector. These clouds are different from public Clouds, which serves the different needs of multiple users and also different from private Cloud, which serves within a single organization. The benefit of Community Clouds is that the Cloud infrastructure can be hosted by a third party organization and it is economically scalable.

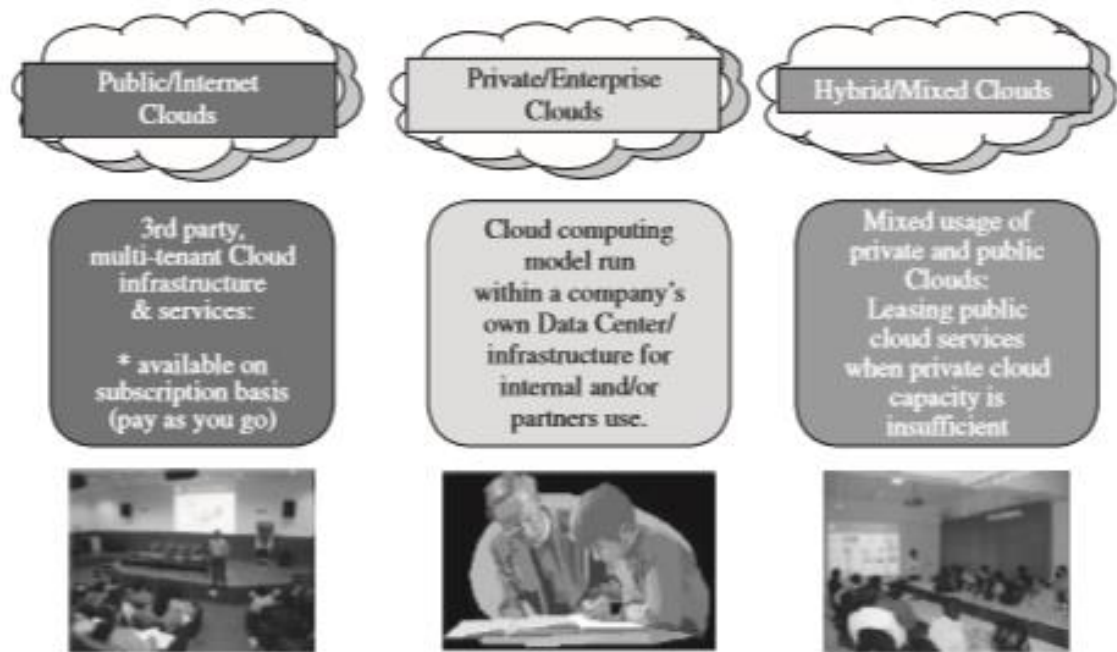


Fig. 1.3: Deployment Models in Cloud computing [1]

1.3 Virtualization

Virtualization works at the heart of Cloud computing as it would be impossible to provide the illusion of unlimited resources without it. If computing is considered fully virtualized that is only because computation has started to involve various components. These distributed components form the processing capability, storing capability and data maintenance for the software resources [1]. Virtualization means creating a virtual version of something. Virtualization is a methodology of division of resources into multiple execution environments. This is done by various techniques, such as:

- (a) Partitioning (hardware or software).
- (b) Time sharing or time slicing.
- (c) Machine simulation (partial or complete).
- (d) Emulation.
- (e) Quality of service, etc.

This helps in utilizing a machine for various purposes along with the illusion of running on a dedicated machine. Typically, a hypervisor provides an abstraction of physical resources (CPU, memory, storage, and local networking) allowing for this illusion [2]. A hypervisor is a thin layer of software that generally provides virtual partitioning capabilities, which runs directly on hardware, but underneath higher-level virtualization services, sometimes referred to as a "bare metal" approach. A VMM (Virtualization Machine Monitor) helps achieve virtualization. The terms hypervisor and virtual machine monitor are generally used interchangeably, but the two are separate conceptually. The VMM can be found in a layer between the hosts' operating system and one or more virtual systems. It provides abstraction to the virtual machines from the guest operating system.

With full virtualization, the virtual machine monitor exports a virtual machine abstraction identical to a physical machine, so that standard operating systems can run just as they would on physical hardware. Virtualization enables organizations to:

- scale the infrastructure such that it can meet the growing needs of any organization.
- avoid dealing with the problem of application isolation, while merging two or more servers.
- relocate critical systems and dynamically provision the resources to increase their availability.
- provide instant provisioning.
- helps in fast scalability.
- facilitate live migration.
- helps in load balancing.
- low downtime for maintenance.
- virtual hardware supports legacy operating systems efficiently.

- facilitates security and fault isolation.

Since more than one instance for the resources can be created virtualization provides flexibility. Also, if physical node is down then temporary migration can be done which provides availability of resources. Proper utilization of hardware is said to be done when virtual machines make use of the hardware resources which are left idle. While utilizing more than one virtual machine it is possible to segregate the services by making each service run on a different virtual machine. The Cloud providers use this technique to deliver resources to users as a substitute for in-house infrastructure. The users provide the requirement and get tailored resources to suit their needs.

1.4 Resource Scheduling

In order to meet consumer needs and to fulfill those needs the resources available at the datacenters have to be managed. This is to be done while virtualizing the resources in such a manner that the resources are put to optimum use and to ensure this various algorithms are used for the scheduling of resources. The primitive objective of scheduling is to expand the resource utilization and minimize the processing time of the tasks. In Cloud computing scenarios, there are two parties involved: Cloud providers and Cloud consumers. On one hand, providers holding large computational resources in their huge datacenters provide their resources to consumers on a pay as you use basis. Meanwhile there are consumers who lease resources from the Cloud provider to run their applications which may have fluctuating loads. The interaction between providers and users occur as shown in Figure 1.4. As illustrated, initially, a request for the resources is sent to the provider. When the provider receives the request, it looks for resources to satisfy the request and assigns the resources to the requesting user, typically as a form of virtual machines (VMs). Then the user uses the assigned resources to run applications and pays for the resources that are used. When the user is done with the resources, they are returned to the provider.

Another interesting aspect of the Cloud computing scenario is that the two parties involved usually have their own interests. Therefore, the Cloud providers focus on getting maximum revenue with minimum investment and in order to achieve

this the providers may squeeze their computational resources, for example, by hosting the virtualized resources so as to use the present virtualized infrastructure to its best. However, if a lot of VMs are allocated to a single host, it may cause interference of values amongst each other. This degrades the results and may lead to unpredictable performance and irritate the consumers. Therefore, to maintain the quality of the service the Cloud provider may reject resource requests or even evict existing virtual systems, all the while ensuring that it does not lead the environment into becoming even more unpredictable. On the other hand, consumers want maximum performance within minimum expenses, in other words, they plan to increase their cost performance as much as possible. This can be achieved by having resources which can satisfy any kind of workload of the application used by the consumer and utilize these resources judiciously. They also have to include unpredictable resources into consideration while they request resources and schedule applications.

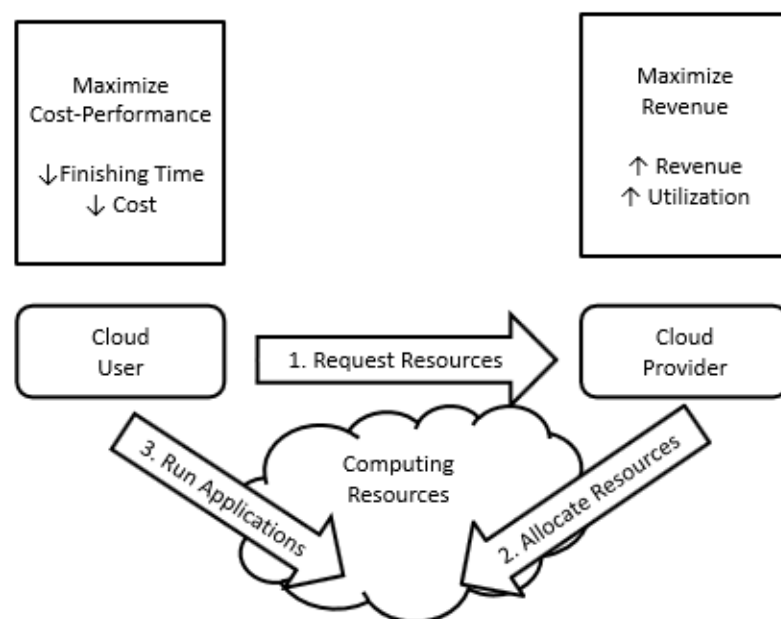


Fig 1.4: Cloud Usage Scenario [4]

However, problem occurs when these two parties are not willing to share information with each other. This makes optimal allocation of resources very difficult to achieve. For example, the Cloud providers are not at all willing to disclose basic information like how many and what kind of machines they have and how they are connected because this information is very critical to the business of the Cloud

provider. In the same manner, consumers are not willing to disclose the details of their workloads and other data like source codes or data sets to the public, including the Cloud providers. This is the reason why the consumers are not able to describe the nature of their resource requests clearly. This is important so as to get assigned resources that are optimal for the requirements of the applications, because they do not know what the Cloud provider has to give in return to serve these requests. Similarly, providers are not able to allocate resources in the best possible manner which would suit the need of the application, as the consumers don't provide the information relating to the workload patterns of the application itself. It is also difficult for the Cloud providers to multiplex the available resources effectively by allocating resources to different applications with complementary resource usage pattern to minimize interference.

There are two fundamental objectives of developing scheduling algorithms. One of the objectives is to enhance the quality of services in job execution and provide the expected output on time. The other objective is the maintenance of fairness of jobs along with efficiency. An effective and well-equipped job scheduling methodology should be resulting in less response time so that the execution of the submitted jobs takes place within a specified frame of time and at the same time there will be an occurrence of within-time resource reallocation. This results in more number of jobs which can be submitted to the Cloud by the consumer which in the end results in improving the business performance of the Cloud system by many folds [5].

1.5 Ant Colony Optimization

In the field of computer science and operations research, the ant colony optimization (ACO) algorithm has been used extensively as a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.

ACO algorithm is one of the algorithms belonging to the ant colony algorithms family consisting of some meta-heuristic optimization techniques. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the objective for the first algorithm was to search for an optimal solution for a path in a graph. Its functionality

was to be inspired by the nature of ants looking for a path between their colony and a source of food. The original idea has met various modifications ever since to overcome a wider range of computational problems. As a result, several problems have emerged, portraying various perspectives to the behavior of ants. Ants have complex social behaviors that has been attracting the attention of human beings ever since. Ant colonies, and more generally social insect societies, are basically distributed systems that, in spite of the simplicity of their individuals, present a highly structured social organization. Ant colonies are able to achieve most intricate tasks which are possible only because of this organizational behavior. These tasks tend to appear to be exceeding beyond the capabilities of individual ants but get done anyway.

The field of “ant algorithms” studies include various models which have been developed observing the behavior of ants. These models are further used as a source of inspiration for the design of new algorithms. These algorithms help obtain solutions for the various optimization and distributed control problems [6]. The basic working principle is that the self-organizing nature which allow the brilliantly coordinated behavior of real ants to be used to dictate coordination among the populations of artificial agents. These artificial agents act as main entities in solving computational problems. There are different perspectives of the behavior of ant colonies which has inspired varied kinds of ant algorithms. Examples are foraging i.e. the search for food, division of labor i.e. categorizing ants according to the tasks given to them, brood sorting, and cooperative transport.

The functionality which results in these examples is due to a phenomenon called stigmergy. It is a form of indirect communication which is facilitated by modifications of the environment. For example, an ant looking for food, referred to as a foraging ant deposits a chemical on the ground while traversing its path to the food source increases the probability that other fellow ants would follow the same traversed path. Biologists have shown that stigmergic communication based simplified models can be used to explain many colony-level behaviors which can be observed in social insects. In other words, biologists have shown that self-organization in social insects can be achieved by stigmergic, indirect communication. The idea leading to ant algorithms is therefore used to form artificial stigmergy to get coordination among societies of the artificial agents. Ant Colony Optimization is one

of the most successful examples of ant algorithms. ACO is inspired by the foraging behavior of ant colonies, and is used to solve discrete optimization problems. [6].

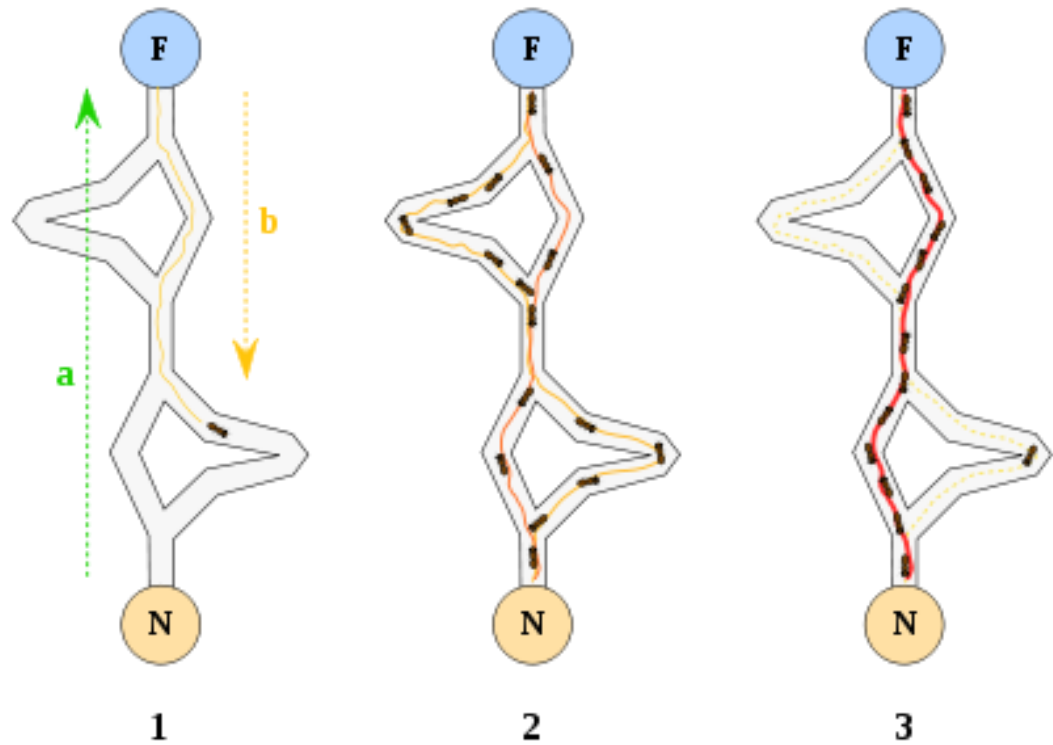


Fig. 1.5: Behavior of Ants which Inspired the Development of ACO (a)
Forward Feedback (b) Backward Feedback

In the real world, ants initially keep on wandering randomly, and when they find a source of food they return to their colony meanwhile laying down pheromone trails. If other ants find this kind of trail, they are not going to continue to travel randomly, but would instead move on to follow the trail, return and reinforce it further if they eventually find the food source. Over time, however, the pheromone trail tends to evaporate which reduces its concentration and thus its attractive strength. If an ant takes a lot of time to traverse down the path and return, the pheromone gets more time to evaporate. Therefore, a short path gets traversed more frequently as compared, and therefore the pheromone concentration becomes higher on shorter paths than longer ones. The evaporation of the pheromone has an advantage. It avoids the procedure to convergence to a locally optimal solution. If there would have been no evaporation at all, the paths which were chosen by the first

ants would tend to become excessively attractive to the following ants. In that case, the exploration of the solution space would become limited. Thus, when one ant finds a good i.e. shorter path from the colony to a food source, other ants are more likely to follow that path, and the positive feedback eventually leads to all the ants following a single path. The idea of the ant colony algorithm is to mimic this behavior with "simulated ants" walking around the graph representing the problem to solve [7].

Ant colony optimization algorithm has been implemented to many combinatorial optimization problems, such as routing vehicles, quadratic assignment, etc. These methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations. It has also been used to solve the travelling salesman problem and produce near-optimal solution. These methods have shown an upper hand over simulated annealing and genetic algorithm approaches while solving similar problems when the graph may change dynamically as the ant colony algorithm can run continuously and adapt to changes in real time. This is of interest in network routing and urban transportation systems.

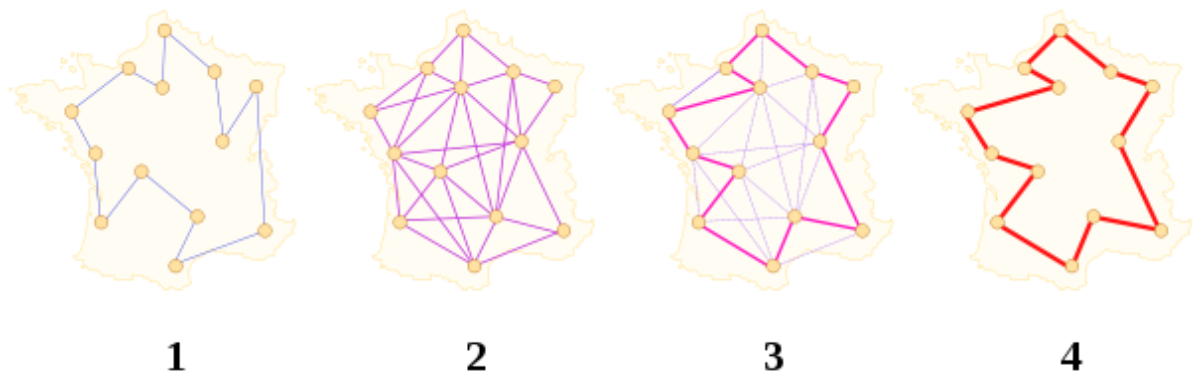


Fig 1.6: Ant Colony Optimization

The first ACO algorithm was called the Ant system and its primitive objective was aimed at solving the travelling salesman problem. In the travelling salesman problem the main objective is to find the shortest round-trip to connect a set of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities.

Chapter 2

Literature Survey

2.1 Cloud Computing

R. Buyya, et al. [1] have described various issues regarding Cloud computing by introducing the Cloud computing paradigm. It provides different scenarios of the various existing computing, storage, and application services provided on the Cloud and illustrates advantages and limits of the existing providers of Cloud computing services. This gives a better understanding of the methodologies needed to utilize Cloud computing in their own respective endeavors. Finally, many open research problems have been detailed that have arisen from the advent of Cloud computing.

Cloud computing architecture can be deployed in various ways. The most famous deployment models are public, private, hybrid and community Cloud. K. Parsi, et al. [8] have described the characteristics of these deployment models. Public Cloud and private Cloud have been compared with each other to give the pros and cons of one over the other. In terms of accessibility, reliability and availability public Cloud is better than private Cloud as it can be accessed from anywhere whereas private clouds is managed by the users in an organization which restricts its access. Public Cloud is more scalable as compared to private Clouds as it is typically larger thus providing seamless on demand service. Moreover data security risks are less in private Clouds. Storage is convenient and less costly in private Clouds. This paper also provides information on which model to choose and when.

A. Burkimsher, et al. [9] have discussed a static, distributed computing platform which has been used for dynamically scheduling parallel and dependent tasks, so as to deliver fairness and quality of service (QoS) to the consumers. The most important QoS requirement is that the responsiveness should be maintained for workloads, with a wide range of execution times even under transient periods of overload. A survey of QoS metrics has been discussed and classified with respect to parameters like responsiveness, fairness and utilization. These metrics are evaluated as to their ability to detect unwanted features which may be present in schedules. The Schedule Length Ratio (SLR) metric is seen to be the most helpful for measuring

responsiveness when dependencies are prevalent. A new list scheduling policy called Projected-SLR is illustrated that delivers good responsiveness and fairness by using the SLR metric in its scheduling decisions. The benefit of using the SLR metric is that small jobs can ‘jump’ the queue to run quickly because their SLRs are more sensitive to the same waiting time. The experimental results obtained in case of Projected-SLR were found to have proved that PSLR performed equally as well in responsiveness, fairness and utilization as the best of the other scheduling policies evaluated like Shortest Remaining Time First (SRTF), using an industrial trace and synthetic workloads. However, Projected-SLR does guarantee a starvation-free behavior, unlike SRTF.

2.2 Resource Scheduling Techniques

Cloud Computing has turned out to be a beneficial and very interesting area in modern distributed computing. It gives a better way of scheduling jobs in restricted conditions. The scheduler maintains the required balance between QoS (Quality of Service) and fairness among the tasks so that the efficiency can be increased. Since the number of Cloud users is growing exponentially day by day and the demand for resources like virtual machines has also increased, therefore the proper scheduling of these tasks is quite a critical issue in Cloud computing. M. Vijayalakshmi, et al. [5] provide a practical comparison of various task scheduling algorithms for the scheduling of virtual machines through an optimization technique which resolves the issues present in the scheduling based algorithms.

H. S. Sidhu, et al. [10] have compared two scheduling policies space-shared and time-shared. This comparison has been made on the basis of parameters like job profit, throughput, net gain and job penalty. In space-shared algorithm the jobs are allocated on a first come first serve basis, whereas in time-shared algorithm the jobs are allocated on a time sharing basis that is similar to the round robin technique. The simulation results prove that space-shared outperforms time-shared policy in case of all of these parameters and otherwise.

Task scheduling is a very effective issue in the field of green Cloud computing. It helps in minimization of energy consumption and also for the minimization of execution time of tasks. To save power use of temperature aware

resource scheduler is made. K. Goyal, et al. [11] have proposed an algorithm which allocates tasks to the scheduler for that machine which is far from its critical temperature as well as which consumes less power. This algorithm is based on a hybrid approach which combines both temperature and power aware resource scheduling. It contains all of those features that satisfy the users and fulfill all the task requirements. Cloud computing utilizes the concept of virtualization which is used for the implementation of resources. This concept is used in the implement of energy efficient programs. Temperature aware resource scheduling can provide enhancement to Green Cloud computing environment.

Cloud computing provides utility-oriented IT services to consumers all around the world. It is based on a “pay-as-you-go” model and enables hosting of widespread applications from various scientific, business and end-user domains. However, data-centers hosting Cloud applications consume huge amounts of electricity, adding to the high operational costs. It also leads to increased carbon footprints to the environment. Therefore, to minimize the operational costs Green Cloud computing solutions are needed which will further reduce the environmental impact. A. Beloglazov [12] has defined an architectural framework and principles for energy-efficient Cloud computing. On the basis of this architecture, open research challenges, and resource provisioning and allocation algorithms for energy-efficient management of Cloud computing environments have been illustrated. It proposes energy-aware allocation heuristics which provision datacenter resources to client applications in such a manner that it improves energy efficiency of the data center, while delivering the negotiated Quality of Service (QoS). A survey has been conducted as a part of research in energy-efficient computing and it provided architectural principles for energy-efficient management of Clouds. Furthermore, it discussed various energy-efficient resource allocation policies and scheduling algorithms which consider QoS expectations and power usage characteristics of the devices. Also a number of open research challenges have been discussed and addressing these challenges can bring substantial benefits to both resource providers and end users. This approach was validated by conducting a performance evaluation on CloudSim toolkit. The results illustrate that Cloud computing model has immense potential as it provides significant cost savings and has high potential for the improvement of energy efficiency under dynamic workload scenarios.

J. Mair, et al. [13] have proposed three new metrics which will help in selecting the best task schedules. They are Speedup per Watt (SPW), Power per Speedup (PPS) and Energy per Target (EPT). These metrics help task schedulers to achieve energy saving in multicore computers. On the basis of these metrics, they gave new Sharing Policies, the Hare and the Tortoise Policies. These policies take into account the parallelism and Dynamic Voltage Frequency Scaling (DVFS) as a part of their schedules. The obtained experimental results show that on a modern multicore computer the Hare Policy can save energy up to 72% in a system with low utilization. On a busier system the Sharing Policy can make a saving up to 20% of energy over standard scheduling policies

One of the basic problems that Cloud computing deals with is task scheduling. Task scheduling is done at the consumer level as well as on system level. At the user level, scheduling is done to overcome the problems between the consumer and the service provider, whereas at the system level, scheduling has to manage virtualized resources within the datacenter. Y. Chawla and, et al. [14] have made use of conventional scheduling concepts by merging them to provide the solution for better and more efficient task scheduling which would benefit both, the consumer and the service provider. It illustrates the implementation of scheduling algorithm for the Cloud which uses a dynamically optimized cost-based task scheduling algorithm for making proper and efficient mapping of tasks to the available resources in the Cloud. It aims to combine cost based task scheduling which is beneficial to user and dynamically optimized resource allocation methodologies which are beneficial to service provider. It also improves computation or communication ratio and utilization of available resources by grouping the consumer tasks before resource allocation.

With the popularity of Cloud computing, more and more businesses have started operating on the Cloud. Z. Lee, et al. [15] have portrayed the problem of service request scheduling in Cloud computing systems. Considering the three-tiered Cloud architecture, which consists of resource providers, service providers and consumers, the service request scheduling strategies should satisfy the objectives of the service providers and consumers. Therefore a new strategy has been proposed to solve this problem called dynamic priority scheduling algorithm (DPSA). This algorithm is more efficient and optimized than the First Come First Serve (FCFS) and Static Priority Scheduling Algorithm (SPSA).

Cloud computing has emerged as the next generation computing paradigm which facilitates the end users to utilize resources on a pay as you use basis as per requirement. It has been providing a lot of benefits which could not have been realized otherwise. The main objective of Cloud computing is to provide efficient access to remotely placed and geographically distributed resources. A scheduling algorithm is essentially needed to manage the consumers' access to the different Cloud resources. There are various types of resource scheduling technologies in the Cloud computing environment. They are implemented at different levels based on different parameters like cost, performance, resource utilization, time, priority, physical distances, throughput, bandwidth, resource availability etc. C. T. Lin [16] has illustrated, surveyed and analyzed various types of resource allocation scheduling algorithms which provide efficient Cloud services. On the basis of the study of different algorithms, a classification of the scheduling algorithms on the basis of selective features has been presented.

Cloud computing has gained a lot of fame these days as it offers dynamic and flexible resource allocation. It is reliable and provides guaranteed services in pay-as-you-use manner, to Cloud service users. This is the reason that there must be an arrangement that all resources are made available to the consumers who request for them in an efficient manner to satisfy the consumer's need. This resource allocation is done by taking in account the Service Level Agreements (SLA) with the help of parallel processing. The recent implementations consider different strategies with only one SLA parameter. Therefore in order to improve the resource utilization in Cloud it would be better to consider multiple SLA parameters and allocate the resources by preemption mechanism. This mechanism should be for high priority task execution. C. S. Pawar, et al. [17] have discussed an algorithm which considers task execution by preemption and multiple SLA parameters such as memory, network bandwidth, and required CPU time. The experimental results thus obtained show that this algorithm provides better utilization of resources. Therefore it will prove better in situations where resource contention is fierce.

The demand for the Cloud computing infrastructure has been increasing day by day due to its unparalleled benefits; one of them being that different types of computing can be performed on the Cloud environment. Task scheduling is an NP-hard optimization problem, and a lot of meta-heuristic based algorithms have been

implemented and proposed to solve it. A. E. Keshk, et al. [18] have implemented and illustrated a Cloud task scheduling policy based on ant colony optimization algorithm for load balancing. It has been compared with different scheduling algorithms to get better results. Ant Colony Optimization (ACO) is randomized optimization search approach that is used for the allocation of the incoming jobs to the virtual machines. The main benefit of using it is in helping balance the system load meanwhile trying to minimizing the makespan of a given tasks set. The load balancing factor is linked with the job finishing rate and is proposed to make the job finishing rate at different resource being similar and the ability of the load balancing will be improved. The scheduling methodology that has been discussed has used CloudSim for simulation and to obtain the comparison results. The obtained experimental results showed that, the algorithm decreases the degree of imbalance between the available virtual machines and the increase the overall performance.

Cloud computing is still at an early stage in terms of research. It has been providing many cutting-edge features but it still has some shortcomings to be overcome such as relatively high operating cost and environmental hazards like increasing carbon footprints. These environmental hazards can be reduced up to some extent by efficient scheduling of the Cloud resources. To implement this, the temperature on which a machine is currently running can be taken as a significant parameter for Virtual Machine (VM) scheduling. S. Kinger, et al. [19] has discussed a new proactive technique that considers current and maximum threshold temperature of Server Machines (SMs) before making the scheduling decisions. This is done with the help of temperature prediction, so that the maximum temperature is never reached. Different workload scenarios have been taken into account and the experimental results thus obtained show that the proposed technique is better than the existing systems of VM scheduling (which does not consider current temperature of nodes before making scheduling decisions). Thus, reduction in the need of cooling systems for a Cloud environment has been obtained.

When a virtualized server provides an environment for allows computing resources to be shared among various performance-isolated platforms and that too more than one, the concept gives form to something called virtual machines. Virtual Machine (VM) technology has been applied extensively in data center environments due to its important features, including reliability, flexibility and the ease of

management. S. Kinger, et al. [20] has proposed two new scheduling algorithms that provide efficient green enhancements. First algorithm is workload-aware scheduling algorithm which schedules jobs on the basis of current workload and migrates VMs accordingly. Simulation results indicate that this algorithms bring about 7%-12% of energy efficiency. Second algorithm is temperature-aware scheduling algorithm which aims at maintenance of the temperature of Cloud system below critical temperature threshold.

2.3 Evolutionary Algorithms and Optimization Techniques

M. Dorigo, et al. [6] have explained how ants find shortest paths under controlled experimental conditions, and illustrates how the observation of this behavior has been translated into working optimization algorithms. The ACO meta-heuristic is included in the category of combinatorial optimization. It also provides dedicated to the detailed description of all the major ACO algorithms created till now along with its theory. This description, which is developed using the traveling salesman problem as a running example, is completed by a guide to implementing the algorithms. In particular, they proved convergence for a specific class of ACO algorithms and discussed the relationship between ACO and other methods such as stochastic gradient descent, mutual-information-maximizing input clustering, and cross-entropy. They also described how ACO is being used to solve a variety of combinatorial optimization problems. These areas include routing, assignment, scheduling, and subset problems, as well as a number of other problems in such diverse fields as machine learning and bioinformatics.

Cloud computing is the coming up as another field of distributed computing which includes features of parallel computing and grid computing. In other words it is more of a commercial implementation of these computer science concepts. One of the very basic issues in this environment deals with task scheduling. A good job scheduler should be able to adapt its scheduling methodology to the changing environment and the types of tasks. M. Tawfeek, et al. [21] has compared a Cloud task scheduling policy based on the Ant Colony optimization algorithm with other different scheduling algorithms such as First Come First Serve and Round-Robin. Ant colony optimization is random optimization search approach that will be used for allocating the incoming jobs to the virtual machines. The basic motive of these algorithms is

minimizing the makespan of a given tasks set. The algorithms use CloudSim toolkit package for simulation. The obtained experimental results show that Cloud task scheduling based on ant colony optimization outperformed FCFS and Round Robin algorithms.

Cloud Computing has significantly made its mark in the field of information technology. It has paved a way to increase the computational capacity as well as add capabilities on the fly without investing in the new infrastructure, licensing new software or training new personnel. Cloud computing in a broader sense can be understood as a connected pool of computing resources (including software and hardware) which can be delivered eventually as a service over a network. The virtualized resources are available to the user as a utility on pay as and when resources are used basis. However managing these resources in a dynamic fashion is a problem a lot of researchers are studying these days. These resources are made available only at certain fixed times and also only for fixed intervals of time. Thus scheduling of these resources includes a major part of resource management. An optimized scheduling of resources is required to maintain separation between the end users of the resources. M. S. Rana, et al. [22] have discussed various resource scheduling strategies that have been and are being implemented in Cloud computing environments. Also a comparison among the characteristics features of these resource scheduling algorithms has been done.

Cloud computing is developed as a relevant field of distributed computing. It is based on virtualization of resources and grid computing. Job scheduling is quite a critical problem in Cloud computing. While scheduling jobs in Cloud environments, job priority and scheduling should be essentially considered. R. A. Preethima, et al. [23] conducted a detailed survey of algorithms based on meta-heuristics. These algorithms help to obtain an optimal solution in job scheduling. These algorithms are Improved Genetic Algorithm, Ant Colony Optimization, Modified Bee Life Algorithm, Particle Swarm Optimization, Artificial Bee Colony Optimization and Improved Intelligent Water Drops Algorithm. These job scheduling algorithms were evaluated keeping in mind the makespan, data transmission and the computation cost. The existing algorithms used for scheduling do not consider some of the metrics such as energy consumption and availability. Therefore the scheduling algorithm should be

involved in order to improve and achieve the optimal solution for job scheduling in the given Cloud environment.

Cloud Computing after Grid Computing is a new paradigm in the field of distributed computing. Cloud computing seems to be emerging significantly and is more promising in terms of handling requests, flexibility, security and sharing of resources. Its basic and most important objective is to maintain the Quality of Service provided to the end user in terms of processing power, failure rate, space utilization and many such Quality of Service parameters. Resource allocation, resource sharing, and request scheduling are few of the complex and challenging problems in Cloud computing. Cloud maintains the resources and simultaneously schedules the coming requests sent by the consumer. It is a complex problem due to distributed nature of the cloud. A lot of parameter based algorithms have been proposed to help in solving this problem. Some of them are priority based, colony based, cost based algorithms etc. All these algorithms do not necessarily consider the fault rate of datacenters or of the nodes. P. Kumar, et al. [24] have proposed a QoS aware Honey Bee scheduling algorithm for Cloud Infrastructure as a Service (IaaS) service model. Apart from the fault rate, this algorithm has also taken into consideration many additional parameters that are having significant role in Cloud IaaS.

The traveling salesman problem (TSP) in the field of operations research is a classical discrete or in other words combinatorial optimization problem. It is a prominent illustration of a certain class of problems in computational complexity theory which have been classified as NP-hard. Ant colony optimization has been inspired by the co-operative food retrieval instincts of ants and has been widely applied in various combinatorial optimization problems. It has proved to be unexpectedly successful in solving them. L. Li, et al. [25] present an improved ant colony optimization algorithm for solving the traveling salesman problem. This algorithm adopts a new probabilistic selection mechanism by using Held-Karp lower bound to decide the trade-off between the influence of the heuristic information and the pheromone trail. The experimental results show that it can be used to stably generate better solution for the traveling salesman problem. These results were compared with rank-based ant system and max-min ant colony optimization algorithm. The proposed algorithm was found to be performing better.

N. Kumar [26] has handled the problem of load distribution of workloads within nodes of a Cloud. It has been solved by using an algorithm which makes the use of Ant Colony Optimization (ACO). This algorithm has been illustrated as a modified approach of the Ant Colony Optimization which has been applied from the perspective of the Cloud as well as grid network systems with the basic objective of load balancing of nodes. This modified algorithm has a benefit over the original approach in which each ant build its own individual result set and which is later on built into a complete solution. In this approach the ants continuously keep updating a single result set rather than updating their own result set. Further, it is known that a Cloud can support various types of application which can be used by various customers therefore, the system, which is incurring a cost for the end user should operate without any hindrances and should make use of such algorithms that can keep on providing the proper system functioning even at peak usage hours.

Task scheduling and resource management are important factors in deciding the performance of the grid. Grid task scheduling is needed to be achieved various important parameters. Some of these parameters include the performance and economic aspects such as the measure to judge service quality, optimal span, meet load balancing needs, achieve high reliability, achieve maximum resource utilization and many others. Y. Ma, et al. [27] presents a task scheduling methodology based on Chaotic Ant Colony Optimization Algorithm,. It has been obtained by using the randomness periodicity and the regularity of chaotic motion to improve the quality of individual ants, and the premature convergence of Ant Colony Optimization Algorithm and Genetic algorithm. MATLAB is the tool used to conduct simulation experiments for the Chaotic Ant Colony Optimization algorithm and the classic Genetic Algorithm, and the results show that the task scheduling algorithm is capable and effective to decrease the running time.

The Probabilistic Traveling Salesman Problem is a variation of the classic Traveling Salesman Problem. It is one of the most important stochastic routing problems. Y. Marinakis, et al. [28] proposed a nature inspired approach based on Honey Bees Mating Optimization (HBMO), Expanding Neighborhood Search Strategy (ENS) and Greedy Randomized Adaptive Search Procedure (GRASP) for the solution of the Probabilistic Traveling Salesman Problem. This new hybrid algorithm has two additional innovative features as compared to the other Honey Bees Mating

Optimization algorithms which concerns the workers and the crossover operator. This proposed algorithm was tested on various benchmark problems from TSPLIB with quite satisfactory results. TSPLIB is a library of sample instances for the TSP (and other related problems) from various sources and of various types. Comparisons with the classic GRASP algorithm, the Particle Swarm Optimization (PSO) algorithm and along with a Tabu Search algorithm have been done and illustrated. Also, a comparison is performed with the results of quite a number of implementations of the Ant Colony Optimization algorithm from the work done in the literature and in 6 out of 10 cases the proposed algorithm gives a better solution.

Cloud computing is a type of parallel and distributed computing which can handle and maintain data in huge quantities. The resources on Cloud have to be assigned properly and that is the main problem to be handled. Researchers have suggested algorithms to get this problem solved like greedy algorithms, first come first serve and many others. However, these methods provide just a local optimal solution and therefore there is no global solution to solve the problem of allocation of resources in the Cloud computing. In order to solve this problem, inspired from the ant foraging behavior and integrated multi-goal match of the secondary distribution, an ant colony optimization algorithm has been illustrated by H. Chen, et al. [29]. The improved ant colony algorithm combines the traditional ant colony algorithm and the advantage provided by greedy algorithm. The Cloud computing simulation tool CloudSim was used to implement this algorithm also. The simulation experiments were conducted on it by embedding these scheduling algorithms onto a visualization system which uses Java. The simulation results were compared and it was seen that the improved ant colony algorithm reduced the scheduling time and it also implemented system load balancing at the same time.

When a large number of tasks are submitted by the end users, the Cloud providers have to equip the data centers in order to get these massive tasks finished, meanwhile keeping in mind the end user's service demand. S. Xue, et al. [30] implemented an ACO-LB (Load balancing optimization algorithm based on ant colony algorithm) algorithm to overcome the problem of load imbalance of virtual machine while the procedure of task scheduling. The ACO-LB algorithm can very efficiently adapt to the dynamic Cloud environment. It is used to reduce the makespan of task scheduling, while maintaining the load balance of virtual machines in the data

center. The simulation of this workflow scheduling is done in CloudSim. The experimental results show that the proposed ACO-LB algorithm has better performance and load balancing ability.

A job scheduler should be able to adapt its scheduling methodology to the changing environment and the types of tasks. K. Li, et al. [31] proposed and illustrated a Cloud job scheduling technique based on Load Balancing Ant Colony Optimization (LBACO) algorithm. The objective of this algorithm is to balance the entire system load and at the same time try to minimize the execution time of the given tasks. This technique was implemented and tested on CloudSim. Experimental results showed the proposed LBACO algorithm performs better than FCFS (First Come First Serve) and the basic ACO (Ant Colony Optimization)

R. Thakur, et al. [32] proposed an approach for solving the scheduling problem quite efficiently. The manner in which colonies of real ants behave has led to the emergence of a new procedure in computing, which is called Ant colony optimization technique. The problem discussed in this paper is a single processor-scheduling problem, where the total of values of all jobs is maximized. It also gives a new pheromone updating methodology which can be used to optimize ACO (Ant Colony Optimization) technique used in solving the Traveling Salesman Problem. The value of a job is described by a stepwise increasing function with one or more moments at which the changes in job value are done. Further it proves that the special scenario of this problem, with a single moment of change of job values, is same as to the well-known NP-hard problem of minimizing weighted number of late jobs. The programming language used is JAVA to solve the problem. Also a number of heuristic algorithms were implemented and tested to solve the above problem.

Chapter 3

Problem Statement

From the various problems which have been studied it was noted that Cloud computing is a highly virtualized platform which captivates the attention of varied users to meet their demands for resources. It has made availability of resources very convenient which makes it even easier to implement services on these platforms. It undertakes execution of tasks or jobs to meet the requests for these services.

The Cloud provider should make use of a scheduler which is responsible to arrange, allocate and get these tasks executed in the best possible manner so as to make judicious use of the available resources. Cloud providers have to deal with a humongous number of tasks as they host different types of services for different organizations.

The sequence in which tasks are executed is called a schedule. In Cloud computing the schedule should be such that it meets the deadlines, should properly allocate tasks to virtual machines and should execute within stipulated time.

The algorithm proposed in this work makes use of a famous evolutionary algorithm called Ant Colony Optimization. The reason for choosing an evolutionary algorithm for scheduling is that these type of algorithms keep improving the processed results until an optimal solution is reached. The ant colony optimization algorithm is based on the foraging behavior of ants.

The virtual machines (VMs) and the tasks form the nodes of a bipartite graph whose edges connect the VMs and the tasks to each other. During an iteration ants move from one VM to another for the next task until they complete a tour (all tasks have been allocated), which helps to build solutions to the scheduling problem. This concept is then used to achieve a task schedule which would give the least makespan among all the schedules, including those delivered by other algorithms. Makespan is defined as the execution time taken by all the jobs to finish. The optimal schedule is compared with other schedules and the optimization algorithm improves the result with each iteration.

Chapter 4

Implementation

The allocation of tasks to resources can be defined as a graph problem. In the scenario of the proposed algorithm the set of nodes, V , of the graph $G = (V, E)$ are the VMs and the tasks to be executed. The set of edges E is obtained by connecting the VMs to the tasks which can be illustrated as follows:

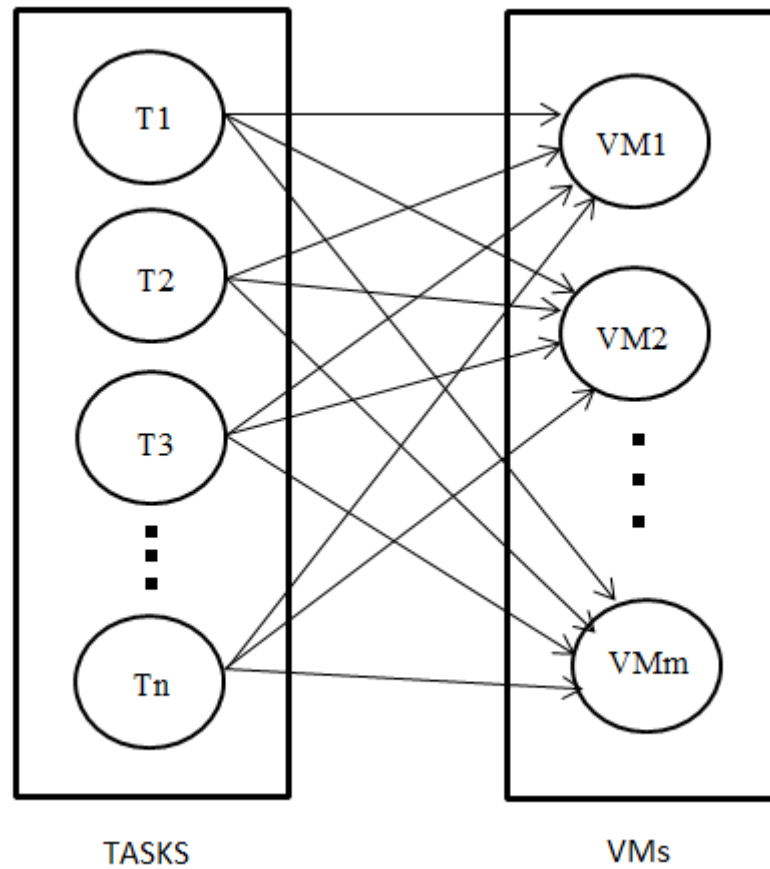


Fig 4.1: A Complete Bi-partite Graph between Tasks and Virtual Machines

Ants placed on the VM traverse the edges to build solutions or in this case schedules, until all the tasks have been allocated to the virtual machines. During the traversal on the edges ants deposit a chemical called pheromone. When one ant deposits pheromone on a particular edge another ant is likely to follow the pheromone trail. This mechanism is followed by ants when they go out to look for food. At times, some of the ants may take different routes to the food source. This is because of

occurrence of pheromone evaporation which reduces pheromone concentration on the path to the food source. Once an ant has traversed all the task nodes the edges traversed are updated with pheromone so that in the next iteration the ant should traverse the edges with higher pheromone content.

4.1 Probabilistic Rule

In order to decide which nodes to traverse the ants make use of a probabilistic rule. This rule was given by M. Dorigo and T. Stutzle [6]. While describing the Simple-ACO algorithm, which is inspired by the behavior of real ants, they provided the rule and it defined the path for ants' searching behavior. The ant continuously moves from one node to the other using this decision policy until it finally reaches the destination node. Initially a constant amount of pheromone is deposited, or in this case assigned to all the edges. When an ant k is at node i , it uses the pheromone trail τ_{ij} to calculate the probability of choosing the next node j :

$$\tau_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{l \in N_i^k} \tau_{il}^\alpha}, & \text{if } j \in N_i^k; \\ 0, & \text{otherwise;} \end{cases}$$

(1)

here, N_i^k is the neighborhood of ant k when it is on node i . In S-ACO the neighborhood of a node i is defined as the nodes connected directly to i in the graph, except the predecessor of i . The same concept has been applied to the existing problem of task scheduling. The nodes in this case are different so the probabilistic rule is defined as follows:

$$\tau_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}]^\beta}{\sum_{s \in VMS} [\tau_{is}(t)]^\alpha * [\eta_{is}]^\beta}, & \text{if } ij \in E; \\ 0, & \text{otherwise;} \end{cases}$$

(2)

here, $\tau_{ij}(t)$ is the pheromone concentration at time t on the edge ij which is from Task i to VM j . VMS is the set m of virtual machines represented as $VMS = \{VM_1, VM_2, \dots, VM_m\}$. It is a subset of set of vertices V . η_{ij} is the visibility calculated as:

$$\eta_{ij} = 1/d_{ij},$$

(3)

here, d_{ij} is the execution time and transfer time of the task i on VM j and it can be calculated as:

$$d_{ij} = \frac{Total_Length_i}{Pe_Num_j * Pe_mips_j} + \frac{Filesize_i}{BW_j}$$

(4)

Here, $Total_Length_i$ is the total length of the task i submitted to VM j . Pe_Num_j is the number of processors in VM j . Pe_mips_j is the MIPS (million instructions per second) of each processor in VM j . $Filesize_i$ is the input file size of task i . BW_j is the bandwidth of VM j . The parameters α and β in Eq. (2) are used to control the weight of pheromone trail and visibility respectively.

4.2 Updating the Pheromone Trail

As discussed before the edges between the tasks and the VMs is initialized by a certain amount on pheromone before the ants start the traversal. After the first traversal has been made the tour of the ant is stored in the memory. This information is used to update the pheromone concentration on the edges which were a part of the tour. This is done so that preference to those edges is given in the next iteration. The updation of the pheromone trail is done by the ant depositing a particular amount of pheromone, $\Delta\tau_{ij}^k(t)$ on the edge. The quantity of pheromone is decided as follows:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)}, & \text{if } (i,j) \in T^k(t); \\ 0, & \text{otherwise;} \end{cases}$$

(5)

Here, $T^k(t)$ is the tour done by ant k at iteration t . $L^k(t)$ is the length of the tour i.e. the makespan of this tour which is computed by Eq. (6). Q is an adaptive parameter.

$$L^k(t) = \sum_{ij \in T^k(t)} d_{ij}$$

(6)

So, the updation of the pheromone trail is done according to the following equation:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

(7)

Here, ρ is the evaporation rate of the pheromone, also known as trail decay. Its value is defined as $0 < \rho < 1$.

4.3 Proposed Algorithm

The proposed algorithm is as follows:

Step 1: Initialize the task list, the VM list and min = 99999.0

Step 2: for each edge between the tasks and the VMs initialize the pheromone trail and the visibility

Step 3: if current_iteration < max_iteration then goto step 4 else goto step 8

Step 4: for every task i do step 5

Step 5: calculate the VM for the current task i according to Eq. (2) and bind them together for execution

Step 7: calculate tour length by summing up the execution time of task i on the corresponding VM

Step 8: if all tasks allocated then goto step 9 else goto step 4

Step 9: if execution_time of schedule < min then min = execution_time

else current_iteration = current_iteration + 1 and goto step 3

Step 10: makespan of schedule = min

Step 11: end

4.4 Simulation Tool : CloudSim

In order to implement the above methodology CloudSim was made in use. CloudSim is a toolkit which helps model and simulate Cloud computing scenarios. The application services hosted under Cloud computing model have complex composition, provisioning, configuration, and deployment needs. Evaluation of the performance of Cloud provisioning policies, application workload models, and resources performance models in a repeatable manner under varying system and user configurations and requirements is difficult to achieve. CloudSim supports system as well as behavioral modeling of Cloud system components like virtual machines, data centers, and resource scheduling techniques. It is used to implement generic application that can be extended with ease and limited effort. Currently, it helps in modeling provisioning methods and simulating Cloud computing environments consisting of single as well as inter-networked Clouds. It helps in defining interfaces for implementation of policies and allocation techniques of VMs under inter-networked Cloud computing scenarios.

CloudSim was proposed by R. Buyya [33] which extends the features of GridSim that allows modeling and simulating the environment of Cloud, datacenters, virtual machines, and cloudlets etc. CloudSim deals only with single work load but it is not suitable for workflow scheduling as multiple tasks need to be scheduled together. CloudSim deals with task execution, but it doesn't consider dependencies between tasks and task clustering. It also doesn't deal with overhead due to failures.

CloudSim is a novel, extensible and generalized simulation framework that helps in modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services without any hitch. By using CloudSim, industry-based developers and researchers test the efficiency of a newly developed application in a controlled environment as it is easy to set up according to the requirements. Further, the results obtained by running the simulation on CloudSim can be used to alter the performance of the application as required. The advantages of using CloudSim to test the initial performance include: (i) it takes very less effort to implement Cloud based application provisioning scenario for testing and this is done within a very short span of time and (ii) applications can be tested with very less

deployment and programming effort by the developers who model these applications. They can be tested for performance in heterogeneous Cloud scenarios.

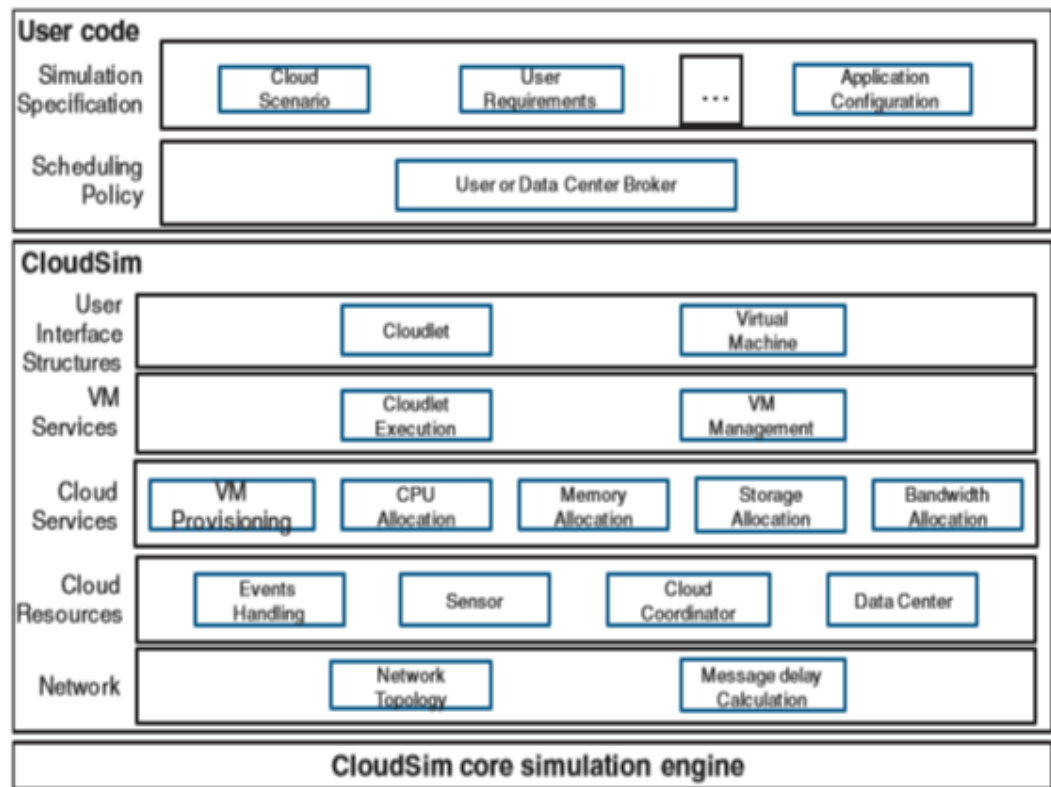


Fig 4.2 Layered CloudSim Architecture [33]

Chapter 5

Experimental Results

The experiment was conducted with 20 virtual machines and 60 tasks. The virtual machines had MIPS ranging from 1000 to 2100. They utilized 1-3 processing elements and bandwidth ranging from 1000 to 2000. The tasks or cloudlets had length ranging from 9000 to 40000 and the input file size ranged from 50 to 400. Initial pheromone deposition is 0.8, Q is taken to be 100, the value of α is 0.3 and β is 2. But it was changed to see the effect on the makespan and these values were found best to use for experimentation.

Entity Type	Parameters	Value
Datacenter	Architecture	x86
	Operating System	Linux
	Virtual Machine Monitor	Xen
	Number of Hosts	1
Virtual Machine	Million Instructions per second	650-2100
	Number of virtual machines	20
	Bandwidth	1000-2000
	Number of Processors	1-3
	RAM	512
Tasks (Cloudlet)	Million Instructions	9000-40000
	Number of Tasks	60
	Input File Size	50-400
	Output File size	300
	Number of Processors	1

Table 5.1: Setting the parameters for simulation

The results obtained for further comparison was that Ant Colony Optimization is more efficient in providing a better schedule with reduced makespan. The following snapshots illustrate the execution time obtained in each of the algorithms:

```

<terminated> MainAnt [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Jun 30, 2015, 3:
The algo gives minimum executin time as:960.0
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.0: Broker: Trying to Create VM #4 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
0.0: Broker: Trying to Create VM #6 in Datacenter_0
0.0: Broker: Trying to Create VM #7 in Datacenter_0
0.0: Broker: Trying to Create VM #8 in Datacenter_0
0.0: Broker: Trying to Create VM #9 in Datacenter_0
0.0: Broker: Trying to Create VM #10 in Datacenter_0
0.0: Broker: Trying to Create VM #11 in Datacenter_0
0.0: Broker: Trying to Create VM #12 in Datacenter_0
0.0: Broker: Trying to Create VM #13 in Datacenter_0
<

```

Fig 5.1: Snapshot for Ant Colony Optimization

```

<terminated> MainTimeShared [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Jun 30, 2
107.09899999999999: Broker: Destroying VM #10
107.09899999999999: Broker: Destroying VM #11
107.09899999999999: Broker: Destroying VM #12
107.09899999999999: Broker: Destroying VM #13
107.09899999999999: Broker: Destroying VM #14
107.09899999999999: Broker: Destroying VM #15
107.09899999999999: Broker: Destroying VM #16
107.09899999999999: Broker: Destroying VM #17
107.09899999999999: Broker: Destroying VM #18
107.09899999999999: Broker: Destroying VM #19
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
Schedule takes 1158.1553354691284to complete
Simulation takes 68 millisecs to complete
<

```

(a)

```

Problems @ Javadoc Declaration Search Console Progress Call Hiera
<terminated> MainSpaceShared [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Jun 30, 2
107.09898831398488: Broker: Destroying VM #10
107.09898831398488: Broker: Destroying VM #11
107.09898831398488: Broker: Destroying VM #12
107.09898831398488: Broker: Destroying VM #13
107.09898831398488: Broker: Destroying VM #14
107.09898831398488: Broker: Destroying VM #15
107.09898831398488: Broker: Destroying VM #16
107.09898831398488: Broker: Destroying VM #17
107.09898831398488: Broker: Destroying VM #18
107.09898831398488: Broker: Destroying VM #19
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
Schedule takes 1158.1553354691284to complete
Simulation takes 50 millisecs to complete

```

(b)

Fig 5.2: Snapshot for (a) Round Robin (b) First Come First Serve

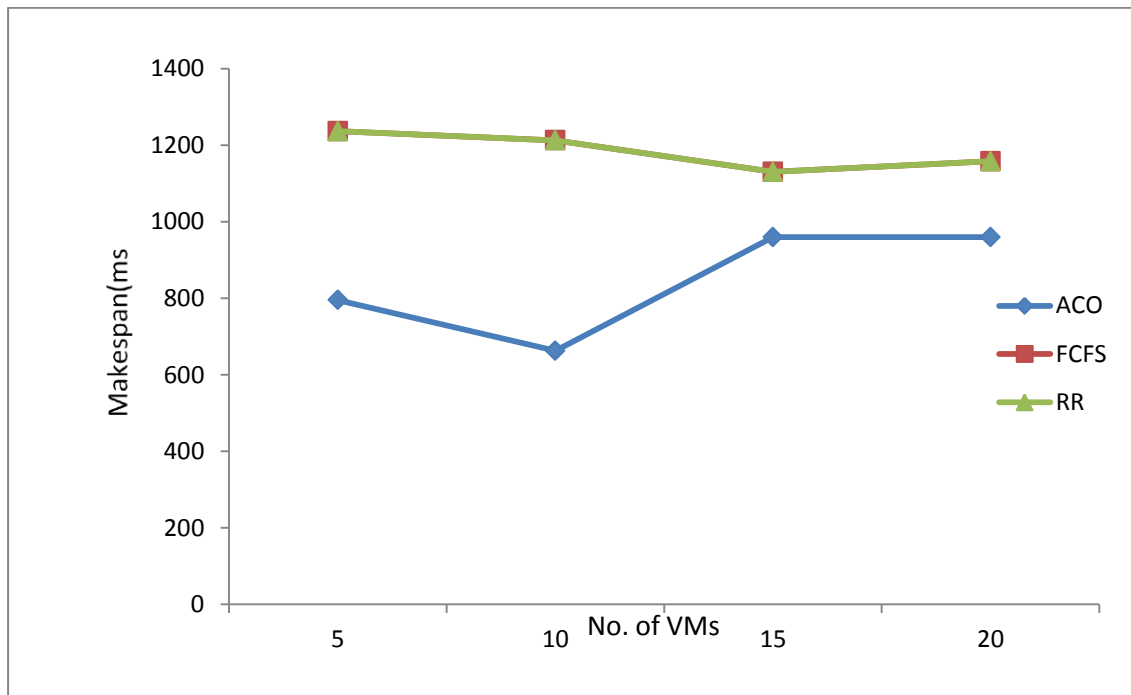


Fig 5.3: Values of Makespan for Different Number of VMs in Case of ACO, FCFS and RR

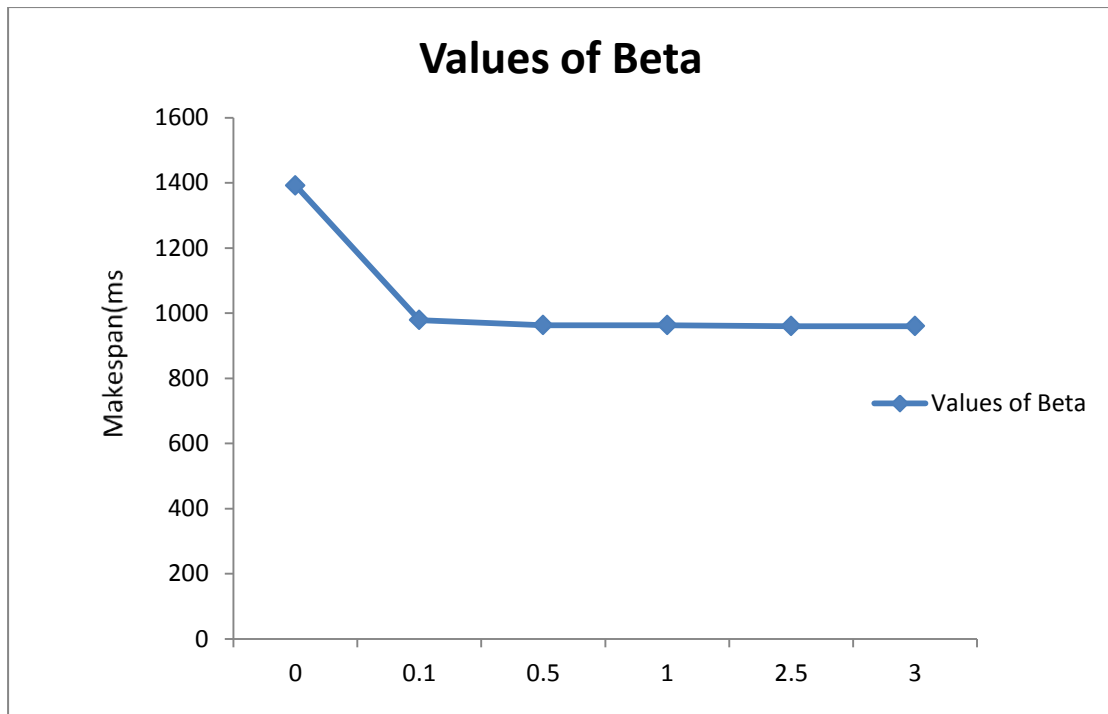


Fig 5.4: Values of Makespan for Different Values of Beta

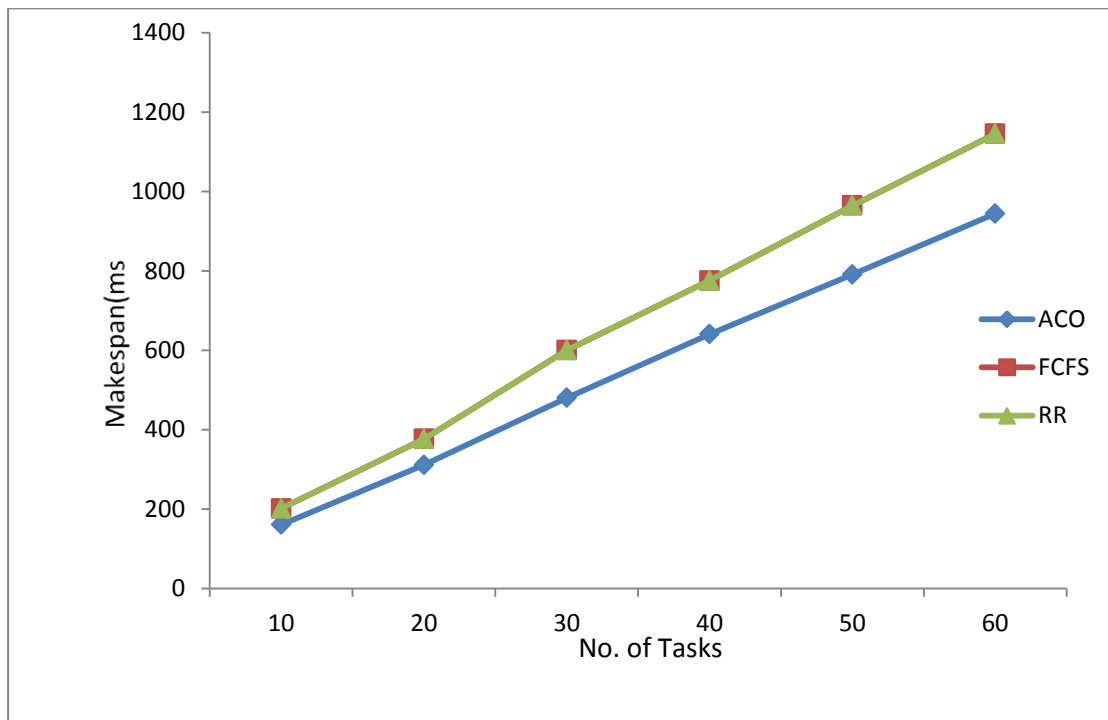


Fig 5.5: Values of Makespan for Different Number of Tasks in Case of ACO, FCFS and RR

Chapter 6

Conclusion and Future Scope

This report gives a brief introduction about Cloud computing and how it is implemented using CloudSim for improving performance of Cloud application, data centers etc. It further explains how scheduling plays an important role in the efficient utilization of resources on the data center. It gives an insight into the state-of-the-art research done in the field of Cloud computing. Along with that various resource scheduling techniques from various fields were studied to get a better understanding of the concept. It explains further about CloudSim and its architecture, which offers the environment which can be directly used for modeling Cloud computing environments. This report illustrates how it can be used to simulate the working of Clouds by creating data centers, VMs, brokers etc. It gives details about scheduling strategies implemented on CloudSim to compare how various entities interact among themselves. The proposed algorithm in this report is based on a heuristic based algorithm called the Ant Colony Optimization algorithm. The main objective of implementing this algorithm was to reduce the makespan of schedules. The experimental results show that on comparing with other traditional algorithms the proposed algorithm gives less makespan time thus proving to be superior and efficient than others.

In further prospects we will focus on merging two or more evolutionary algorithms to get better schedules. This concept may be used for energy minimization to implement green Clouds by using efficient scheduling techniques. This would be achieved by using algorithms based on various parameters to provide optimum schedules. Power consumption at data centers is to be reduced with the help of VM migration and consolidation for which resource scheduling in green Clouds will be needed. These scheduling techniques would be implemented through simulation to come up with a better solution in the form of scheduling to solve the problem of power consumption. These techniques would be based on evolutionary algorithms to get better results.

References

- [1] R. Buyya, J. Broberg and A. Gościński, *Cloud computing*, Hoboken, N.J.: John Wiley & Sons, 2011.
- [2] National Institute of Standards and Technology U.S. department of Commerce, “The NIST Definition of Cloud Computing”, NIST Special Publication 800-145, 2011.
- [3] Wikipedia, “Cloud computing”, 2015. [Online]. Available: https://en.wikipedia.org/wiki/Cloud_computing.
- [4] G. Lee, “Resource Allocation and Scheduling in Heterogeneous Cloud Environments”, Doctor of Philosophy in Computer Science in the Graduate Division, University Of California, Berkeley, 2012.
- [5] M. Vijayalakshmi and V. V. Kumar, “Investigations on Job Scheduling Algorithms in Cloud Computing”, *International Journal of Advanced Research in Computer Science & Technology*, vol. 2, no. 1, pp. 157-161, 2014.
- [6] M. Dorigo and T. Stutzle, *Ant colony optimization*. Cambridge, Mass.: MIT Press, 2004.
- [7] Wikipedia, “Ant colony optimization algorithms”, 2015. [Online]. Available: https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms.
- [8] K. Parsi and M. Laharika, “A Comparative Study of Different Deployment Models in a Cloud”, *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 5, pp. 512-515, May 2013.
- [9] A. Burkimsher, I. Bate and L. S. Indrusiak, “A Survey of Scheduling Metrics and an Improved Ordering Policy for List Schedulers Operating on Workloads with Dependencies and a Wide Variation in Execution Times”, *Future Generation Computer Systems*, vol. 29, no. 8, pp. 2009–2025, October 2013.

- [10] H. S. Sidhu and Himani, "Comparative Analysis of Scheduling Algorithms of Cloudsim in Cloud Computing", *International Journal of Computer Applications*, vol. 97, no. 16, pp. 29-33, July 2014.
- [11] K. Goyal and S. Kinger, "Hybrid Approach for Resource Scheduling in Green Clouds", *International Journal of Computer Applications*, vol. 74, no. 17, 2013.
- [12] A. Beloglazov, "Energy Aware Resource Allocation Heuristics for Efficient Management of Data centres for Cloud Computing", *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755-768, 2012.
- [13] J. Mair, K. Leung and Z. Huang, "Metrics and Task Scheduling Policies for Energy Saving in Multicore Computers", *Proceedings of Grid Computing (GRID) 2010: 11th IEEE/ACM International Conference*, Brussels, pp. 266 – 273, 25-28 October 2010,
- [14] Y. Chawla and M. Bhonsle, "Dynamically Optimized Cost Based Task Scheduling in Cloud Computing", *International Journal of Emerging Trends and Technology in Computer Science*, vol. 2, no. 3, pp. 38-42, June 2013.
- [15] Z. Lee, Y. Wang and W. Zhou, "A Dynamic Priority Scheduling Algorithm on Service Request Scheduling in Cloud Computing", *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference*, Harbin, Heilongjiang, vol. 9, pp. 4665 – 4669, 12-14 Aug. 2011.
- [16] C. T. Lin, "Comparative Based Analysis of Scheduling Algorithms for Resource Management in Cloud Computing Environment", *International Journal of Computer Science and Engineering*, vol. 1, no. 1, pp. 17-23, July 2013.
- [17] C. S. Pawar, and R. B. Wagh, "Priority Based Dynamic resource allocation in Cloud Computing", *Cloud and Services Computing (ISCOS: 2012 International Symposium)*, Gujarat, pp. 311 – 316, 1-2 March 2013.

- [18] A. E. Keshk, A. El-Sisi, M. A. Tawfeek, and F. A. Torkey, "Intelligent Strategy of Task Scheduling in Cloud Computing for Load Balancing", *International Journal of Emerging Trends and Technology in Computer Science*, vol. 2, no. 6, pp. 12-22 December 2013.
- [19] S. Kinger, R. Kumar and A. Sharma, "Prediction Based Proactive Thermal Virtual Machine Scheduling in Green Clouds", *The Scientific World Journal*, vol. 2014, pp. 1-12, March 2014.
- [20] S. Kinger, R. Kumar and A. Sharma, "Workload and Temperature Aware Virtual Machine Scheduling and Migration in Green Clouds", *2nd International Conference on Advances in Computer Science and Application*, Lucknow, pp. 591-598, 21-22 June 2013.
- [21] M. Tawfeek, A. El-Sisi, A. Keshk and F. Torkey, "Cloud Task Scheduling Based on Ant Colony Optimization", *Computer Engineering & Systems (ICCES), 2013 8th International Conference*, Cairo, pp. 64 – 69, 26-28 November 2013.
- [22] M. S. Rana, S. Kumar KS and Jaisankar N, "Comparison of Probabilistic Optimization Algorithms for Resource Scheduling in Cloud Computing Environment", *International Journal of Engineering and Technology*, vol. 5, no. 2, pp. 1419-1427, May 2013.
- [23] R. A. Preethima and M. Johnson, "Survey on Optimization Techniques for Task Scheduling in Cloud Environment", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 12, pp. 413-415, December 2013.
- [24] P. Kumar, K. Gopal and J. P. Gupta, "Fault Aware Honey Bee Scheduling Algorithm for Cloud Infrastructure", *Confluence: The Next Generation Information Technology Summit (4th International Conference)*, Noida, pp. 133-137, 26-27 September 2013.
- [25] L. Li, S. Ju and Y. Zhang, "Improved Ant Colony Optimization for the Traveling Salesman Problem", *International Conference on Intelligent*

Computation Technology and Automation (ICICTA), Hunan, vol. 1, pp. 76 – 80, 20-22 October 2008.

- [26] N. Kumar, “Load Balancing of Nodes in Cloud Using Ant Colony Optimization”, *Computer Modelling and Simulation (UKSim): 2012 UKSim 14th International Conference*, Cambridge, pp. 3 - 8, 28-30 March 2012.
- [27] Y. Ma and Y. Wang, “Grid Task Scheduling Based on Chaotic Ant Colony Optimization Algorithm”, *Computer Science and Network Technology (ICCSNT): 2012 2nd International Conference*, Changchun, pp. 469 – 472, 2012.
- [28] Y. Marinakis and M. Marinaki, “A hybrid honey bees mating optimization algorithm for the probabilistic traveling salesman problem”, *CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, pp. 1762-1769, 2009.
- [29] H. Chen, L. Xiong and C. Wang, “Cloud Task Scheduling Simulation via Improved Ant Colony Optimization Algorithm”, *Journal of Convergence Information Technology*, vol. 8, no. 7, pp. 1139 - 1147, 2013.
- [30] S. Xue, M. Li, X. Xu, and J. Chen , “An ACO-LB Algorithm for Task Scheduling in the Cloud Environment”, *Journal of Software, Academic Publisher*, vol. 9, no. 2, pp. 466-473, February 2014.
- [31] K. Li, G. Xu, G. Zhao, Y. Dong and D. Wang, “Cloud Task scheduling based on Load Balancing Ant Colony Optimization”, *Chinagrid Conference (ChinaGrid): 2011 Sixth Annual*, pp. 3-9, 2011.
- [32] R. Thakur and A. Yadav, “Solving the Single Processor Scheduling Problem by Using Ant Colony Optimization Technique”, *Communication Systems and Network Technologies (CSNT): 2012 International Conference*, Rajkot, pp. 759 – 763, 2012.
- [33] R. Calheiros, R. Ranjan and A. Beloglazov, C. De Rose and R. Buyya, “CloudSim: a toolkit for modeling and simulation of Cloud computing

environments and evaluation of resource provisioning algorithms”,
Software: Practice and Experience, vol. 41, no. 1, pp. 23-50, 2010.

Video Reference

The video for this thesis work is available at:

<https://www.youtube.com/watch?v=XsRUHNIkw4Q&feature=youtu.be>