Cloud computing relies on sharing of resources to achieve coherence and economies of scale similar to a utility over a network. Cloud Computing is getting popular every day. Cloud service providers provide services to large scale cloud environment with cost benefits. Also, there are some popular large scaled applications like social networking and internet commerce. These applications can provide benefit in terms of minimizing the costs using cloud computing. Cloud computing is considered as internet-based computing service provided by various infrastructure providers based on their need, so that cloud is subject to Quality of Service (QoS), Load Balance (LB) and other factors which have direct effect on user consumption of resources controlled by cloud infrastructure.

Gaurav Dhiman received his Ph.D. in Computer Engineering from Thapar Institute of Engineering & Technology, Patiala. He is currently working as an Assistant Professor with the Department of Computer Science, Government Bikram College of Commerce, Patiala. He has published more than 30 peer-review research papers (indexed in SCI-SCIE).

Enhancement of Dynamic Load Balancing using Ant in Cloud Environment

EDLBACE
Gaurav Dhiman
Jaswinder Singh
Satinder Kaur

Enhancement of Dynamic Load Balancing using Ant in Cloud Environment
Enhancement of Dynamic Load Balancing using Ant in Cloud Environment

EDLBACE

Gaurav Dhiman
Jaswinder Singh
Satinder Kaur

LAP LAMBERT Academic Publishing
Imprint
Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: www.ingimage.com

Publisher:
LAP LAMBERT Academic Publishing
is a trademark of
International Book Market Service Ltd., member of OmniScriptum Publishing Group
17 Meldrum Street, Beau Bassin 71504, Mauritius

Printed at: see last page

Copyright © Gaurav Dhiman, Jaswinder Singh, Satinder Kaur
Copyright © 2019 International Book Market Service Ltd., member of OmniScriptum Publishing Group
Contents

Chapter 1

Introduction 1

Chapter 2

Literature Survey 15

Chapter 3

Problem, Platform & Technology Used 21

Chapter 4

Results And Discussions 35

Chapter 5

Conclusion And Future Work 42

References 43
CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

1.1.1. Cloud

Cloud computing relies on sharing of resources to achieve coherence and economies of scale similar to a utility over a network. Cloud Computing is getting popular every day. Cloud service providers provide services to large scale cloud environment with cost benefits. Also, there are some popular large scaled applications like social networking and internet commerce. These applications can provide benefit in terms of minimizing the costs using cloud computing. Cloud computing is considered as internet-based computing service provided by various infrastructure providers based on their need, so that cloud is subject to Quality of Service (QoS), Load Balance (LB) and other factors which have direct effect on user consumption of resources controlled by cloud infrastructure [1].

In Cloud scheduling process need to achieve several factors. So, it needs to use the effective algorithm for allocating proper task to the proper resources. Various task scheduling algorithms has been proposed, most important task scheduling algorithms are Min-min, Max-min, RASA, etc.

A cloud system is so complex due to its unpredictable environment. It is extremely challenging to obtain accurate information on the state of the system. Moreover, it contains large resources which are shared and require complex policies to manage them. The factors affecting the resource management in cloud are performance, functionality and cost. Resource management in cloud computing is associated with fluctuating workloads which pose a major Challenge to elasticity of cloud computing. The situation for fluctuation can be of two ways. One is a planned spike and the other is an unplanned spike in workloads. For the first case, the situation can be predicted in advance and resource allocation can be done in advance. For the second case, resources have to be allocated on demand and reallocated when needed. This is called Auto-scaling in cloud computing. This shows that the policies for resource management for cloud computing is different from the policies for traditional systems. The general policies to be considered in cloud resource management are Admission control: takes decision whether to admit a job/request to be processed in the cloud, Resource allocation: provisions Virtual
Machines (VMs) onto Physical Machines (PMs) and jobs onto VMs, Quality of Service (QoS): refers to metrics like response time, operational cost, throughput, maximization of profit and so on, Workload balancing: load balancing of jobs between the resources so as to improve its utilization, Energy Management: refers to optimized use of energy in the datacenter [1]. Resource allocation in cloud can be classified into two types:

**First is Mapping of Virtual Machines (VMs) onto Physical Machines:** Resources of cloud include the software and hardware required to execute user workloads. Examples of such resources are memory, CPU, bandwidth, storage and network. Resource allocation is the process of allocating optimal resources to the jobs requested by the user, so these jobs can be processed efficiently. In a cloud environment, resource allocation generally means allocating a Virtual Machine satisfying the configurations specified by the user. The configurations include the operating system, MIPS, network bandwidth, storage and so on. This method of allocation can be referred as mapping of VMs onto Physical Machines [3].

**Second is Mapping of Workloads onto VMs:** There is another situation where the cloud contains a set of existing Virtual Machines and a built environment with predefined memory, CPU and bandwidth. The users submit their workloads which may be time varying and deadline based. These workloads need to be allocated to the optimal resources such that the workloads are processed efficiently. This type of allocation is referred as mapping of workloads onto VMs. This article presents a discussion on various issues and challenges of resource allocation in cloud computing. Research issues include resource provisioning, job scheduling, resource overbooking, scalability, pricing, load balancing, multi-tier applications, availability, overheads in network I/O workloads and Quality of Service (QoS) constraints. Open challenges in resource management for cloud are also listed.
1.2 Features of Cloud

- **Worldwide Access.** Cloud computing increases mobility, as you can access your documents from any device in any part of the world. For businesses, this means that employees can work from home or on business trips, without having to carry around documents. This increases productivity and allows faster exchange of information. Employees can also work on the same document without having to be in the same place.

- **More Storage.** In the past, memory was limited by the particular device in question. If you ran out of memory, you would need a USB drive to backup your current device. Cloud computing provides increased storage, so you won’t have to worry about running out of space on your hard drive.

- **Easy Set-Up.** You can set up a cloud computing service in a matter of minutes. Adjusting your individual settings, such as choosing a password or selecting which devices you want to connect to the network, is similarly simple. After that, you can immediately start using the resources, software, or information in question.

- **Automatic Updates.** The cloud computing provider is responsible for making sure that updates are available — you just have to download them. This saves you time, and furthermore, you don’t need to be an expert to update your device; the cloud computing provider will automatically notify you and provide you with instructions.

- **Reduced Cost.** Cloud computing is often inexpensive. The software is already installed online, so you won’t need to install it yourself. There are numerous cloud computing applications available for free, such as Dropbox, and increasing storage size and memory is affordable. If you need to pay for a cloud computing service, it is paid for incrementally on a monthly or yearly basis. By choosing a plan that has no contract, you can terminate your use of the services at any time; therefore, you only pay for the services when you need them [5].

1.3 Drawbacks

- **Security** When using a cloud computing service, you are essentially handing over your data to a third party. The fact that the entity, as well as users from all over the world, are accessing the same server can cause a security issue. Companies handling confidential information might be particularly concerned about using cloud computing, as data could possibly be...
harmed by viruses and other malware. That said, some servers like Google Cloud Connect come with customizable spam filtering, email encryption, and SSL enforcement for secure HTTPS access, among other security measures.

- **Privacy** Cloud computing comes with the risk that unauthorized users might access your information. To protect against this happening, cloud computing services offer password protection and operate on secure servers with data encryption technology.

- **Loss of Control** Cloud computing entities control the users. This includes not only how much you have to pay to use the service, but also what information you can store, where you can access it from, and many other factors. You depend on the provider for updates and backups. If for some reason, their server ceases to operate, you run the risk of losing all your information.

- **Internet Reliance** While Internet access is increasingly widespread, it is not available everywhere just yet. If the area that you are in doesn’t have Internet access, you won’t be able to open any of the documents you have stored in the cloud [5].

Cloud computing relies on sharing of resources to achieve coherence and economies of scale similar to a utility over a network. Cloud Computing is getting popular every day. Cloud service providers provide services to large scale cloud environment with cost benefits. Also, there are some popular large scaled applications like social networking and internet commerce. These applications can provide benefit in terms of minimizing the costs using cloud computing. Cloud computing is considered as internet-based computing service provided by various infrastructure providers based on their need, so that cloud is subject to Quality of Service (QoS), Load Balance (LB) and other factors which have direct effect on user consumption of resources controlled by cloud infrastructure. In Cloud scheduling process need to achieve several factors. So, it needs to use the effective algorithm for allocating proper task to the proper resources. Various task scheduling algorithms has been proposed, most important task scheduling algorithms are Min-min, Max-min, RASA [1], etc.
1.4 SCHEDULING PROCESS IN CLOUD

The main advantage of job scheduling algorithm is to achieve a high-performance computing and the best system throughput. The available resources should be utilized efficiently without affecting the service parameters of cloud. Scheduling process in cloud can be categorized into three stages they are Resource discovering and filtering, Resource selection, and Task submission [10]. In resource discovery datacenter broker discovers the resources present in the network system and collects status information related to them. During resource selection process target resource is selected based on certain parameters of task and resource. Then during task submission task is submitted to the selected resource.

1.5 MIN-MIN ALGORITHM

Min-min scheduling is based on Minimum Completion Time (MCT) that is used to assign tasks to the resources having minimum expected completion time. It will work in two phases, in the first phase, the expected completion time will be calculated for each task in a meta task list.
During the second phase, the task with the overall minimum expected completion time from meta task list is chosen and assigned to the corresponding machine [1]. Then this task is removed from meta task list and the process is repeated until all tasks in the Meta task list are mapped to the corresponding resources. However, the Min-min algorithm is unable to balance the load well as it usually does the scheduling of small tasks initially.

1. for all submitted tasks in meta-task Ti
2. for all resource Rj
3. compute $C_{ij} = E_{ij} + r_j$
4. While meta-task is not empty
5. find the task $T_k$ consumes minimum completion time.
6. assign task $T_k$ to the resource $R_j$ with minimum execution time.
7. remove the task $T_k$ from meta-tasks set
8. update $r_j$ for selected $R_j$
9. update $C_{ij}$ for all i

In algorithm $r_j$ represents the ready time of the resource $R_j$ to execute a task, $C_{ij}$ and $E_{ij}$ represent the expected completion time and execution time of the tasks.

1.6 MAX-MIN ALGORITHM

Similar to Min-min, a scheduler schedules tasks by expecting the Execution Time of the tasks and allocation of resources. Instead of selecting the minimum MCT, the maximum MCT is selected, that is why it is named Max-min. It focuses on giving priority to large tasks over others small. The Max-min algorithm is typical to the Min-min algorithm, except for being different in; the word “minimum” would be replaced by “maximum” [9]. Officially Max-min algorithm does better than Min-min algorithm in cases when the number of short tasks is more than the longer ones. For example, if there is only one long task, the Max-min algorithm executes many short tasks concurrently with the long one. In order to avoid the main drawbacks of the Max-min and Min-min, the two schedulers can be executed alternatively to each other for assigning tasks to appropriate resources, for eliminating each other drawback. Such methodology, called Resource Awareness Scheduling Algorithm (RASA) which was a new grid task scheduling algorithm

1. for all submitted tasks in meta-task Ti
2. for all resource Rj
3. compute \( C_{ij} = E_{ij} + r_j \)
4. While meta-task is not empty
5. find the task \( T_k \) consumes maximum completion time.
6. assign task \( T_k \) to the resource \( R_j \) with minimum execution time.
7. remove the task \( T_k \) from meta-tasks set
8. update \( r_j \) for selected \( R_j \)
9. update \( C_{ij} \) for all \( i \)

In algorithm, the expected time of resource \( R_j \) is the time to become ready to execute a task after finishing the execution of all tasks assigned to it which is denoted by \( r_j \). Also, \( E_{ij} \) is the estimated execution time of task \( T_i \) on resource \( R_j \) whereas \( C_{ij} \) is the Expected Completion Time that is the estimated execution time and ready time together [1].

**1.7 RASA ALGORITHM**

RASA is a hybrid algorithm of two other ones such as Min min and Max-min. In RASA, an estimation of the completion time of each task on the available resources is calculated then Max-min and Min-min algorithms are applied alternatively to take advantage of both algorithms and avoids their drawbacks. In RASA the Max-min and Min-min scheduling algorithms are applied alternatively [5]. For example, if the first task is assigned to a resource by Min-min strategy, in the next round the task will be assigned by Min-min and so on. Based on experimental results, if the number of available resources in grid system is odd it is highly preferred to start by Min-min algorithm in first round otherwise it will be started by Max min algorithm. For next rounds just assign resources to task using a strategy different from last round it reduces the delays in the execution of small tasks by Max-min algorithm and large tasks by Min-min algorithm.

1. for all tasks \( T_i \) in meta-task
2. for all resources \( R_j \)
3. Compute \( C_{ij} = E_{ij} + r_j \)
4. do until all tasks in metatask are mapped to the resource
5. if the number of resources is even then
6. for each task in meta task find the expected completion time and the resource that make it.
7. find the task which gives the maximum expected completion time.
8. assign that task to the faster resource to get minimum completion time.
9. delete the task from meta task.
10. update ready time \( r_j \)
11. update completion time \( C_{ij} \) for all task \( i \).
12. else
13. for each task in meta task find the expected completion time and the resource that make it.
14. find the task which gives the minimum expected completion time.
15. assign that task to the faster resource to get minimum completion time.
16. delete task from meta task.
17. update ready time \( r_j \)
18. update completion time \( C_{ij} \) for all \( i \).
19. end do.

### 1.8 RESOURCE ALLOCATION ISSUES FOR CLOUD

Though cloud computing was first introduced in early 2000 by Amazon, active research contributions in this area started in 2008 after the emergence of cloud test bed software like Eucalyptus and Open Nebula. This section presents various issues related to resource allocation in cloud computing. The comparison between the existing resource allocation systems.

- **Resource Provisioning:** Hien Nguyen Van et al. have presented resource management architecture for cloud computing environment which deals with the problem of efficient VM provisioning and placement using Constraint Programming [1]. There are three main constraints to be satisfied in cloud resource management problem such as CPU, RAM and Network I/O. The system addresses only CPU and RAM capacity. Moreover, the system gives solution on how to match jobs to available VMs. It does not address the problem of scheduling, which is the order of execution of the applications.

- **Job Scheduling:** The issue of scheduling for virtual machines was first addressed in [2]. The new scheduler software called Haizea that could perform three types of scheduling policies namely best effort, immediate and advanced reservation. Immediate scheduling means allocation of jobs onto resources if available and rejected otherwise. Best effort scheduling means, if resources are unavailable jobs are made to wait in a queue. In
advanced reservation scheduling, resources are allocated with prior time slots and these jobs cannot be preempted. Haizea is a scheduler associated with Open Nebula that implements all these types of scheduling policies. Later deadline-based scheduling was introduced to Haizea in which the deadline of jobs is known in advance. Time constraints are applied to immediate, deadline and advanced reservation policies, so jobs with best effort are preempted often without time constraints. Amit Nathani et al. have identified this pitfall of haizea and have proposed a new algorithm to overcome this problem by introducing time constraint for best effort policies [4]. Job scheduling can also be done in parallel. Some of the available scheduling algorithms are gang scheduling and backfilling. Gang scheduling is used for dynamic job scheduling in a distributed cloud environment [16]. This work also addresses job migration and starvation problems. Improvised back filling algorithm is used in [15] for scheduling parallel jobs in cloud. However, this system does not present any results to evaluate its efficiency.

- **Resource overbooking:** So far, the allocations of VMs to jobs were done in terms of available CPU and memory. Moreover, the method of allocation was static, which leads to resource overbooking. Jin Heo et al. have presented a feedback control system to dynamically re-allocate memory to VMs based on the feedback of the system [3]. The system is implemented in Xen-based environment and allocation is done in real time.

- **Scalability:** The systems presented up to this point have not dealt with scalability, which is considered as a key feature of cloud. Increasing the workload on available resources is called scaling in and increasing the workload by adding resources on demand is called scaling out. The system proposed by Trieu C. Chieu et al. have addressed scalability in cloud computing and also proposed an algorithm to dynamically scale up and down resources on demand in terms of VMs [4]. The algorithm uses a threshold-based indicator which is used to measure the active sessions of web applications in the cloud environment. If the indicator shows that the threshold level has exceeded then additional resources (VMs) are added. Similarly, the indicator also checks if any VM is idle and such VMs are switched off. But this system fails to consider reallocation of resources in
case of load changes. This issue is addressed by Zhen Xiao et al. who have presented a threshold-based algorithm to dynamically reallocate workloads based on load changes. The migration could incur operating cost as the state of the machine has to be saved while migrating. But this cost could be avoided using live migration as done in. The prediction-based algorithm deals with overload avoidance and load balancing. The Exponentially Weighted Moving Average (EWMA) and the skewness concept are used for prediction of the future resources needed by the workloads. Cloud based systems have unpredictable workloads and thus needs dynamic scalable systems. An alternative approach to the prediction-based systems is proposed in [5] which use the queuing information of the system to dynamically scale resources in cloud. Lyapunov Optimization technique is used in this system. Another system has been implemented for dynamic scalability in cloud which predicts the changing patterns in the workloads.

**Pricing:** The primary attribute of cloud computing inspired from utility computing is its feature of pay-per-use model. That is, services rendered through cloud are charged based on service consumption. The services may be hardware or software appliances. This leads to the need of a pricing model whose objective is profit maximization for the service providers and quality of service for the consumers. But there is a tradeoff between these two objectives along with service request fluctuation in cloud. A pricing model addressing these issues for cloud computing is presented in [3]. The model uses processor sharing technique while addressing cost effectiveness for both service providers and users. Quality of service constraints in cloud includes cost and response time. While the proposed model considers only cost it fails to address response time. There are two ways of designing the pricing model. One way the price is calculated as per the requirements of the user and the second way is allowing user to choose from services already set with predefined price [2].

**Load balancing:** When resources are wasted, it is referred as underutilization. For example, there may be physical machine or virtual machine without any application running on it. Over utilization refers to accumulation of large number of processing on the same physical or virtual machine. This leads to slower response times of the
applications. To tackle this problem, migration of VMs is one solution which is called as autonomic management of resources in cloud. Such a dynamic resource management system for cloud through migration is presented in [2]. So far, the systems discussed in this article were all centralized systems while in [2] it is a distributed data center each having a node agent to keep track of the utilization rates of the local data center. The local agent communicates with a global agent to decide upon migration between the data centers. The system uses PROMETHEE method to make out decisions in parallel which enhances scalability through migrations. But migration incurs cost of transferring VM from one location to another. Hence the problem is how to avoid migration or how to minimize migration. The load balancing system in [3] solves this problem using genetic algorithm based on the current state and historical data of the system.

- **Multi-tier applications:** The resources management systems presented until this point have only applications with single tier. As cloud is a server-based technology which uses internet as a communication medium, multi-tier applications are obviously need to be included. The M/M/C and M/M/1 queuing models are used to develop a multi-tier cloud. But it still needs to incorporate other constraints like load balancing, Quality of service and so on [2].

- **Availability:** The system becomes unavailable due to reasons like hardware or software failures, load fluctuations, long waiting time of jobs and so on. The traditional method to provide high availability is providing extra idle resources to be used in case of failures hence wasting resources [4]. This could be avoided by replicating resources like running multiple instances of the same applications as done in [4]. But the issue of how many replications and load balancing in such situations is not addressed in the system.

- **Overheads in Network I/O Workloads:** Workloads to be processed on virtualized cloud platforms may be CPU intensive or network I/O intensive. The issues related to these types of workloads especially I/O workloads were addressed in like overheads of high context switching, contention for CPU due to frequent exchange of memory pages. The authors conclude that in-depth understanding of these overheads is necessary for effective
resource management in cloud. Not much work has been done in network bandwidth as constraint in solving resource allocation problems besides CPU [4].

- **Quality of Service (QoS) constraints**: This includes various parameters like cost, response time, throughput and so on. A cloud system providing services to its customers should take care that it satisfies the customer requirements by fulfilling their quality of service requirements. Linlin Wu et al. have presented two resource allocation algorithms to maximize profit and minimize cost by allocating available VMs to user requests [4]. This system is for SaaS and did not address any scalability issues. Another system proposed by Hadi Goudarzi and Massoud Pedram deals with profit maximization in a distributed cloud environment and it uses force-directed technique to optimize the resource allocation.

### 1.9 Introduction to Healthcare

Healthcare is defined as the service delivered to individuals or populations by healthcare service providers to promote, maintain, monitor or restore health [4]. Internationally, the ultimatum of healthcare is well predictable as it is expected to endlessly expand in future due to tangible reasons such as projected demographic shifts among the aging population, life expectancy, and lifestyle diseases.

The growing progress of healthcare coupled with local as well as foreign patients will definitely attain huge aggregate of patient data, variety, accuracy and leads to the requirement of perfect system in order to sustain and further up the good quality status of healthcare service provider. Generally, this situation creates two main problems for a healthcare system that includes complexity and upsurges the needs of the IT experts deadly.

Moving of healthcare sector to cloud computing system in order to overcome this offending will be definitely a good idea of solution of all the problems individually [5]. Cloud computing is highly rated as the sole representative from current IT trends of efficiency, business agility, and cost reductions.
1.10 Cloud Computing in Healthcare Sector

Cloud computing in healthcare is growing by day to day and plays a major role in the field of healthcare. Traditionally, healthcare sector under-utilized technology especially in improving the delivery of patient care. Healthcare has entered sixteen years after millennium, but hence in healthcare, the number of systems operate manually, generally or relaying on paper, such as medical records to notify and make decisions in most of the conditions still significantly high. Healthcare industry differs greatly from other industries, and the key differences of the healthcare industry with other industry can be categorized into three segments. Firstly, this sector is highly regulated by governed law including regulations to safeguard patients. Secondly cost of high-risk errors to occur in healthcare are costlier than in other industry, and finally, this sector consists of numerous number units such as hospital administration staff, labs, and patients.

Exceptionality privacy of healthcare and security of patients' data makes the data itself sensitive and any criteria misleading will cause severe impact and may lead to life or death at times. Hence, the sensitivity of data handling can result to be unhurried by the adoption of new technologies. Universally, healthcare is reorganized, and reform causes the healthcare information technologies [HIT] to be modernized and as a pathway for this route or center of this transformation is definitely cloud computing without hesitation.

Adoption of cloud computing in healthcare can ominously enrich the healthcare system especially in the comfort zone of efficiency, effectiveness, and reliability. Cloud computing offers an infrastructure that permits hospitals, medical practices, and insurance companies including research facilities which use computing resources at lesser of initial capital costs [6]. By implementing this cloud computing in healthcare, access costs which will usually be in millions of dollars each year, especially in duplication and waste, can be overcome thoroughly.

There are various suitable reasons to use cloud computing in healthcare and numerous problems can be solved such as limitation capacity of storage, high operating cost, and optimizing resource. Migration of healthcare sector towards cloud computing brings some risk which strongly interrelated to the privacy and security although benefits from this decision is attainable. Consequently, it is necessary to maintain, upgrade, and monitor the hardware and software which consists of healthcare data and are crucial in order to avoid negative consequences [7]. The
architecture of cloud computing has the capability to assemble, integrate, analyze the data from various sources either in real time and permits doctors to access patient records without any barrier of place and time. One of the offers that important in healthcare that could be done by cloud computing is the capability of recovery of data in an emergency state such as disaster recovery, and backup data redundancy as it reproduces the data in numerous locations for more heftiness and accessibility [8].

1.11 Current State of Cloud Computing in Healthcare

Over the last few years, cloud computing technology has gradually gained attention in research and numbers of implementations have increased in public and private sectors as well. According to Economics Commerce and Management of United Kingdom, major businesses was expected to invest over $150 billion on cloud computing by 2014 but however the final results shows the final amount is far higher than predicted earlier. Besides, studies on global healthcare IT trends, expected worldwide healthcare cloud computing sector’s revenue in 2017 will unquestionably boost up to $5.4 billion due to upwelling of this sector. Furthermore, the prime contributor, North America prophesied to impact the market share of this sector up to $6.5 billion in 2018 from 1.7 billion in 2013. Canada is the country that recognized diagnostic imaging repositories across the country in order to aid in patients care and as well as cost saving. Based on study, statistic shows that 37% of healthcare service providers have strategic plans with adoption of cloud, 22% are in the planning stages while 25% are in the midst of executing in which this can definitely drive the respective industry.
CHAPTER 2
LITERATURE SURVEY

Shuibing He, Yang Wang (2016) et al: In this paper, they have considered to improve scientific workflows in cloud environments where data transfers between tasks are performed via provisioned in-memory caching as a service, instead of relying entirely on slower disk-based file systems. However, this improvement is not free since services in the cloud are usually charged in a “pay-as-you-go” model. To further show the values of this concept, we also implement these algorithms and apply them, through a simulation study, to improve deadlock resolutions in workflow-based workloads when memory resources are constrained [1].

S.Devipriya (2013) et al: Cloud computing is the use of computing resources that are delivered as a service over a network. It supplies a high-performance computing based on protocols which allows shared computation and storage over long distances. In cloud computing, many tasks need to execute at a time by the available resources in order to achieve better performance, minimum completion time, shortest response time, resource utilization etc. [2].

Hitoshi Matsumoto (2011) et al: Two mechanisms of cooperative PSO and CPSO are analyzed and the load-balance requirement of equipotent CPSO mechanism was discussed. Then the CPSO load-balance architecture was set up, control parameters were chosen, and the criterion of PSO convergence degree was established. Finally, the control strategy for CPSO’ load-balance was proposed. Two tests show that the proposed technique improved the CPSO in precision and efficiency [3].

Pankaj Arora (2012) et al. proposed a Set-Based PSO approach. It tackles a cloud workflow scheduling problem which enables users to define various QoS constraints like deadline constraint, budget constraint and reliability constraint in. It enables users to specify one preferred Qos parameter as the optimization objective. Defined penalty-based fitness functions to address multiple Qos constraints and integrate S-PSO with seven heuristics. A discrete version of Comprehensive Learning PSO algorithm based on S-PSO is implemented [4].
Shaminder Kaur (2012) et al. Discussed cloud computing is the use of computing resources that are delivered as a service over a network. In cloud computing, many tasks need to execute at a time by the available resources to achieve better performance, minimum completion time, shortest response time, resource utilization etc. Because of these different factors, an Improved Max-min algorithm is designed to outperform scheduling process of RASA in case of total complete time for all submitted jobs. So, the scheduling tasks within cloud environment using Improved Max-min can achieve lower make span rather than original Max-min [5].

Rajesh Piplode (2012) et al: An optimal power flow model was established for Available Transfer Capability (ATC) under the static security constraints. The maximum active power of all load nodes in receiving area was taken as objective function. To aim at the low accuracy and premature convergent in ATC optimization algorithms, the chaos cloud particle swarm algorithm based on golden section evaluation criteria (CCGPSO) was proposed. This method divided the particle swarm into standard particle, chaos cloud particle and cloud particle, which used the golden section judge principle according to fitness level. Every sub-swarm particle had respective different algorithm operations [6].

Vignesh V (2013) et al. Discussed resource management is the primary issue as the demand grows for provisioning resources and computation in cloud systems. It presents various research issues pertaining to the management of cloud resources while a comparison is made between existing resource allocation systems. The issues and challenges discussed are resource provisioning, job scheduling, load balancing, scalability, pricing, energy management and availability [7].

Hongsheng Su (2013) et al. Literature meaning of cloud computing is distributed computing, storing, sharing and accessing data over the Internet. It provides a pool of shared resources to the users available on the basis of pay as you go service, means users pay only for those services which are used by him according to their access times. The data processing and storage amount is increasing quickly day by day in cloud environment. This leads to an uneven distribution of overall work on cloud resources. So, a proper balance of overall load over the available resources is a major issue in cloud computing paradigm. Load balancing
ensures that no single node will be overloaded and used to distribute workload among multiple nodes. It helps to improve system performance and proper utilization of resources. It also minimizes the time and cost involved in such big computing models. Load balancing and better resource utilization is provided by many existing algorithms [8].

Yi Zhang (2015) et al: Based on the study of traditional min-min scheduling algorithm, the paper proposed a min-min task scheduling algorithm based on QOS constraints in cloud computing. According to the vector which is generated by QOS parameters, the algorithm processes the matching of resources and tasks, and then provides users with resources which meet their requirements. Experimental results show that the min-min task scheduling algorithm for cloud computing has better performance in such aspects as task execution time, rate of discarding task and QOS satisfaction compared with traditional min-min scheduling algorithm [9].

Anterpreet Kaur (2015) et al: Over the years, distributed environments have evolved from shared community platforms to utility-based models; the latest of these being Cloud computing. This technology enables the delivery of IT resources over the Internet and follows a pay-as-you-go model where users are charged based on their usage. There are various types of Cloud providers each of which has different product offerings. They are classified into a hierarchy of as-a-service terms: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). There is a mass of researches on the issue of scheduling in cloud computing, most of them, however, are about workflow and job scheduling. A cloud workflow system is a type of platform service which facilitates the automation of distributed applications based on the novel cloud infrastructure. Many scheduling policies have been proposed till now which aim to maximize the amount of work completed while meeting QoS constraints such as deadline and budget. However, many of them are not optimal to incorporate some basic principles of Cloud Computing such as the elasticity and heterogeneity of the computing resources. Therefore, there work focuses on studying various problems and issues related to workflow scheduling [10].

Nima Jafari Navimipour (2015) et al: Cloud computing is the latest emerging trend in distributed computing, where shared resources are provided to end-users in an on-demand fashion that brings many advantages, including data ubiquity, flexibility of access, high
availability of resources, and flexibility. The task scheduling problem in Cloud computing is an NP-hard problem. Therefore, many heuristics have been proposed, from low level execution of tasks in multiple processors to high level execution of tasks. In this paper, a new evolutionary algorithm is proposed which named CSA to schedule the tasks in Cloud computing. CSA algorithm is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. The simulation results demonstrated that when the value of Pa is low, the speed and coverage of the algorithm become very high [11].

A performance analysis of load balancing algorithms in Cloud environment Computer Communication and Informatics (ICCCI), 2015 International Conference on 8-10 Jan 2015

Load Balancing is essential for efficient operations in distributed environments. As Cloud Computing is growing rapidly and clients are demanding more services and better results, load balancing and task scheduling for the Cloud has become a very interesting research area. Here in this paper, it was investigated the different algorithms proposed to resolve the issue of load balancing and task scheduling in Cloud Computing. The goal is to help in developing a new algorithm after studying almost all available algorithms [12].

Task Scheduling on the Cloud with Hard Constraints Services (SERVICES), 2015 IEEE World Congress on 27 June to 2 July 2015

Scheduling Bag-of-Tasks (BoT) applications on the cloud can be more challenging in order to keep the overall execution costs low. The research in this paper is motivated to investigate task scheduling on the cloud, given two hard constraints based on a user-defined budget and a deadline. A heuristic algorithm is proposed and implemented to satisfy the hard constraints for executing the BoT application in a cost-effective manner. The proposed algorithm is evaluated using four scenarios that are based on the trade-off between performance and the cost of using different cloud resource types. The experimental evaluation confirms the feasibility of the algorithm in satisfying the constraints. The key observation is that multiple resource types can be a better alternative to using a single type of resource [13].

Resource allocation issues and challenges in cloud computing Recent Trends in Information Technology (ICRTIT), 2014 International Conference on 10-12 April 2014
Resource management is the primary issue as the demand grows for provisioning resources and computation in cloud systems. This article presents various research issues pertaining to the management of cloud resources while a comparison is made between existing resource allocation systems. The issues and challenges discussed in this paper are resource provisioning, job scheduling, load balancing, scalability, pricing, energy management and availability [14].

**Improved Max-min heuristic model for task scheduling in cloud**

*Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference on 12-14 Dec 2013* Cloud computing is the use of computing resources that are delivered as a service over a network. In cloud computing, many tasks need to execute at a time by the available resources in order to achieve better performance, minimum completion time, shortest response time, resource utilization etc [4]. Because of these different factors, an Improved Max-min algorithm is designed to outperform scheduling process of RASA in case of total complete time for all submitted jobs. So, the scheduling tasks within cloud environment using Improved Max-min can achieve lower make span rather than original Max-min [15].

**Task Scheduling in cloud computing**

*International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), June 2015* Cloud computing system is a virtual pool of resources which are provided to users via Internet. It gives users virtually unlimited pay-per-use computing resources without the burden of managing the underlying infrastructure. One of the goals is to use there sources efficiently and gain maximum profit. Scheduling is a critical problem in Cloud computing, because a cloud provider has to serve many users in Cloud computing system. So scheduling is the major issue in establishing Cloud computing systems. The scheduling algorithms should order the jobs in a way where balance between improving the performance and quality of service and at the same time maintaining the efficiency and fairness among the jobs. This paper aims at studying various scheduling methods [16].

**Efficient Resource Management for cloud computing environments**

*Pervasive Technology Institute Indiana University Bloomington, IN USA And Rochester Institute of Technology Rochester, NY USA* Many advanced features of Cloud computing like reshaping the field of distributed systems and fundamentally changing how businesses utilize computing today it still has some short comings such as the relatively high operating cost for both public and private
Clouds. The area of Green computing is also becoming increasingly important in a world with limited energy resources and an ever-rising demand for more computational power. In this paper a new framework is presented that provides efficient green enhancements within a scalable Cloud computing architecture. Using power aware scheduling techniques, variable resource management, live migration, and a minimal virtual machine design, overall system efficiency will be vastly improved in a data center-based Cloud with minimal performance overhead [17].
3.1 Problem definition

In current paper that have taken min-min and max-min algorithm. Based on these they have developed the new enhanced algorithm as RASA (resource awareness scheduling algorithm). In this process they have awareness of the resource completion speed. Such that large tasks that requires larger execution time will be put to the slower resources. On the other hand, smaller tasks are allocated to the faster resources. so that over all waiting time can be reduced. In current research they have compared this to the existing max-min and min-min algorithms. such that the performance can be enhanced. As it is very difficult to know the execution speed of the resource for given process.

In our research we will be using genetic algorithm. Which can be further used for identifying the optimal mechanism for execution of the processes. So that performance can be further enhanced. And compare this to the RASA (Resource Awareness Scheduling Algorithm).

3.2 Platform used

Cloud computing is a model for enabling convenient and on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal service provider interaction [1]. Cloud computing can be viewed from two different perspectives: cloud application and cloud infrastructure as the building block for the cloud application. Now a days, most organizations focuses on adopting cloud computing model so that they can cut capital expenditure, efforts and control operating costs. These reasons trigger aggressive growth for cloud adoption in business [2]. Some of the traditional Cloud-based application services include social networking, web hosting, content delivery and real time instrumented data processing, which has different composition, configuration, and deployment requirements. Quantifying the performance of scheduling and allocation policies in a real Cloud computing environment for different application models is extremely challenging.
The use of real infrastructures for benchmarking the application performance under variable conditions is often constrained by the rigidity of the infrastructure. Thus, it is not possible to perform benchmarking experiments in repeatable, dependable, and scalable environments using real world Cloud environments [2, 3]. A more viable alternative is the use of cloud simulation tools. Cloud simulators are required for cloud system testing to decrease the complexity and separate quality concerns. They enable performance analysts to analyse system behavior by focusing on quality issues of specific component under different scenarios. These tools open up the possibility of evaluating the hypothesis in a controlled environment where one can easily reproduce results. Simulation-based approaches offer significant benefits to IT companies by allowing them to test their services in repeatable and controllable environment and experiment with different workload mix and resource performance scenarios on simulated infrastructures for developing and testing adaptive application provisioning techniques [3].

None of the current distributed system simulators offer the environment that can be directly used for modelling Cloud computing environments but CloudSim which is generalized and extensible simulation framework that allows seamless modelling, simulation, and experimentation of emerging Cloud computing infrastructures and application services. This paper first gives background about various Simulators available. Section 3 define and explores various Cloud simulators such as CloudSim, CDOSIM, TeachCloud, icancloud, SPECI and DCSIM. In the section 4, it compares all Cloud Simulators with respect to networking, platform and language.

There have been many studies using simulation techniques to investigate behavior of large-scale distributed systems and tools to support such research. Some of these simulators are GridSim [4], MicroGrid, GangSim, OptorSim, SimGrid [4] and CloudSim [5]. While the first three focuses on Grid computing systems. CloudSim is the only simulation framework for studying Cloud computing systems. However, grid simulators have been used to evaluate costs of executing distributed applications in Cloud infrastructures. GridSim is a java-based event driven simulation toolkit and was developed to address the problem of performance evaluation of real large scaled distributed environments and heterogeneous Grid systems in a repeatable and controlled manner. CloudSim enables seamless modelling, simulation and experimenting on Cloud computing infrastructures.
It is a self-contained platform that can be used to model data centres, service brokers, and scheduling and allocation policies of large-scale Cloud platforms. CloudSim framework is built on top of GridSim toolkit. SimGrid is a generic framework for simulation of distributed applications in Grid platforms. GangSim is a Grid simulation toolkit that provides support for modeling of Grid-based virtual organizations and resources. In particular, there is no support in existing Grid simulation toolkits for modeling of on-demand virtualization enabled resource and application management. Further, Cloud infrastructure modeling and simulation toolkits must provide support for economic entities such as Cloud brokers and Cloud exchange for enabling real-time trading of services.

Among the currently available simulators discussed, only GridSim offers support for economic-driven resource management and application scheduling simulation.

### 3.3 CLOUD SIMULATORS

While grid computing simulators have good but they cannot sufficiently model the cloud infrastructure. There are still only a few options for simulating cloud architecture, possibly because virtualization has enabled the deployment of virtual private clouds on small scale physical test beds. However, there have been some notable proposals for software simulation of clouds of very large scale. The CloudSim simulation framework is based on the SimJava discrete event simulation engine at the lowest layer, while the higher layers implement the GridSim toolkit for the modeling of the cluster, including networks, traffic profiles, resources, etc. CloudSim effectively extends the GridSim core functionalities by modeling storage, application services, resource provisioning between virtual machines, and data centre brokerage, and can even simulate federated clouds.

**A. CloudSim** CloudSim is a new, generalized and extensible simulation toolkit and application which enables seamless modeling, simulation, and experimentation of emerging cloud computing system, infrastructures and application environments for single and internetworked clouds [2, 5, 6]. The existing distributed system simulators were not applicable to the cloud computing environment due to evaluating the performance of cloud provisioning policies, services, application workload, models and resources under varying system, user configurations
and requirements [1]. To overcome this challenge, CloudSim can be used. In simple words, CloudSim is a development toolkit for simulation of Cloud scenarios.

CloudSim is not a framework as it does not provide a ready to use environment for execution of a complete scenario with a specific input. Instead, users of CloudSim have to develop the Cloud scenario it wishes to evaluate, define the required output, and provide the input parameters.

CloudSim is invented as CloudBus Project at the University of Melbourne, Australia and supports system and behavior modeling of cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited efforts. CloudSim helps the researchers to focus on specific system design issues without getting concerned about the low-level details related to cloud-based infrastructures and services [7].

CloudSim is an open source web application that launches preconfigured machines designed to run common open source robotic tools, robotics simulator Gazebo. SimJava is a toolkit for building working models of complex systems. It is based around a discrete event simulation kernel at the lowest level of CloudSim. It includes facilities for representing simulation objects as animated icons on screen [7,8].

**B. CDOSim**

CDOSim is a cloud deployment option (CDO) Simulator which can simulate the response times, SLA violations and costs of a CDO. A CDO is a decision concerning simulator which takes decision about the selection of a cloud provider, specific runtime adaptation strategies, components deployment of virtual machine and its instances configuration. Component deployment to virtual machine instances includes the possibility of forming new components of already existing components. Virtual machine instance’s configuration, refer to the instance type of virtual machine instances. CDOSim can simulate cloud deployments of software systems that were reverse engineered to KDM models. CDOSim has ability to represent the user’s rather than the provider’s perspective. CDOSim is a simulator that allows the integration of fine-grained models. CDOSim is best example for comparing runtime reconfiguration plans or for determining the trade-off between costs and performance [16]. CDOSim is designed to address the major shortcomings of other existing cloud simulators such as 1. Consequently, oriented towards the cloud user perspective instead of exposing fine-grained internals of a cloud platform. 2. Mitigates the cloud user’s lack of knowledge and control.
concerning a cloud platform structure. 3. Simulation is independent of concrete programming languages in the case appropriate KDM extractors exist for a particular language. 4. Workload profiles from production monitoring data can be used to replay actual user behavior for simulating CDOs.

C. TeachCloud TeachCloud is a cloud simulator which is specially made for education purposes. TeachCloud provides a simple graphical interface through which students and scholars can modify a cloud’s configuration and perform simple experiments [17]. TeachCloud uses CloudSim as the basic design platform and introduces many new enhancements on top of it such as:

1. Developing a GUI toolkit.

2. Adding the cloud workload generator to the CloudSim simulator.

3. Adding new modules related to SLA and BPM.

4. Adding new cloud network models such as VL2, BCube, Portland and DCell.

5. Introducing a monitoring outlet for most of the cloud system components.

6. Adding an action module that enables students to reconfigure the cloud system and study the impact of such changes on the total system performance.

D. iCanCloud iCanCloud is a cloud simulator which is based on SIMCAN. In simple words, iCanCloud is a software simulation framework for large storage networks. iCanCloud can predict the trade-off between costs and performance of a particular application in a specific hardware in order to inform the users about the costs involved. It focuses on policies which charge users in a pay-as-you-go manner [18]. iCanCloud has a full graphical user interface from which experiments can be designed and run, but existing software systems can only be modeled manually. It also allows parallel execution of one experiment over several machines [19].

E. SPECI Simulation Program for Elastic Cloud Infrastructures (SPECI) is a simulation tool which allows analyzing and exploration of scaling properties of large data center behavior under the size and design policy of the middleware as inputs. SPECI is a simulation tool which allows exploration of aspects of scaling as well as performance properties of future Data Centers. The
The aim of SPECI is to simulate the performance and behavior of data centers, given the size and middleware design policy as input. Discrete event simulations (DES) are a type of simulation where events are ordered in time maintained in a queue of events by the simulator and each processed at given simulation time [8, 9]. SPECI uses an existing package for DES in Java. SPECI is intended to give us insights into the expected performance of DCs when they are designed, and before they are built. The size of data centers that provide cloud computing services is increasing, and some middleware properties that manage these data centers will not scale linearly with the number of components. SPECI is composed of two packages: data center layout and topology, and the components for experiment execution and measuring. The experiment part of the simulator builds upon SimKit, which offers event scheduling as well as random distribution drawing [9, 10, 12].

Figure 3.1 Types of CloudSim

F. GroudSim GroudSim is an event-based simulator that needs one simulation thread for scientific applications on grid and cloud environments based on a scalable simulation independent discrete-event core. It is mainly concentrated on the IaaS, but it is easily extendable to support additional models such as PaaS, DaaS and TaaS. The user to simulate their experiments from the same environment used for real applications by integrating GroudSim into the ASKALON environment [11].

GroudSim provides a comprehensive set of features for complex simulation scenarios such as simple job executions on leased computing resources, calculation of costs, and background load on resources. Simulations can be parameterized and are easily extendable by probability distribution packages for failures which normally occur in complex environments. Experimental
results demonstrate the improved scalability of GroudSim compared to a related process-based approach [13].

**G. DCSim** DataCenter Simulator is concentrated on virtualized data center which offers IaaS to Multiple tenants, in order to achieve a simulator to evaluate and develop data center management techniques. Data centers are becoming increasingly popular for the provisioning of computing resources. The cost and operational expenses of data centers have skyrocketed with the increase in computing capacity.

Cloud Computing provides environments to enable resource sharing in terms of scalable infrastructures, middleware and application development platforms, and value-added business applications.

- Infrastructure resources

- Software resources

- Application resources

- Business processes

**Two key enabling technologies:**

- SOA (Service-Oriented Architecture)
  - It is the evolution of a system or software architecture for addressing componentization, reusability, extensibility, and flexibility.

- Virtualization technology
  - Handle how images of the operating systems, middleware, and applications are pro-created and allocated to the right physical machines.
3.4 Previous RASA Algorithm

In both MIN-MIN and MAX-MIN the scheduling techniques will be failed when there are multiple smaller tasks and multiple larger tasks. In case of multiple larger tasks MAX-MIN will be failed. Because multiple larger tasks cannot be allocated to multiple large completion time tasks. Similarly, when there are large number of smaller tasks and it is not possible to allocate all the tasks to be allocated to multiple smaller resources. so, in RASA MAX-MIN and MIN-MIN is done simultaneously. So that Resource awareness is performed. When there is large no. of larger tasks then max. completion time will be selected and when there is large no. of smaller tasks then MIN-MIN is used.

1. for all tasks Ti in meta-task
2. for all resources Rj
3. Compute $C_{ij} = E_{ij} + r_j$
4. do until all tasks in metatask are mapped to the resource
5. if the number of resources is even then
6. for each task in meta task find the expected completion time and the resource that make it.
7. find the task which gives the maximum expected completion time.
8. assign that task to the faster resource to get minimum completion time.
9. delete the task from meta task.
10. update ready time $r_j$
11. update completion time $C_{ij}$ for all task i.
12. else
13. for each task in meta task find the expected completion time and the resource that make it.
14. find the task which gives the minimum expected completion time.
15. assign that task to the faster resource to get minimum completion time.
16. delete task from meta task.
17. update ready time $r_j$
18. update completion time $C_{ij}$ for all i.
19. end do.
3.5 Previous RASA Flowchart

In flow chart it represents how, data is flow in all the steps of the algorithm is shown. In this all the parts of flowchart are represented its flow with the help of many different shapes. i.e. square box, diamond box, arrows etc. In RASA flowchart existing flow is represented.

![Previous flow chart of RASA algorithm](image)

Figure 3.2: Previous flow chart of RASA algorithm
3.6 Research Methodology

In this flowchart research methodology of our work is represented with help of symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

![Flowchart]

The starting point for the analysis process identified and analysed the problem to provide a clear understanding and suggest a suitable technique or method to use. An initial feasibility study was sought to consider whether the plans for implementing Cloud computing. The system analysis was done differently because of the changes in the focus, particularly how the design was analyzed and implemented. the process of implementation required decision making processes in choosing the best option for which Cloud service provider to use. This phase defined how the
Cloud Based Patient Information Management System [CBPIMS] was implemented, ensuring that it was operational and user friendly. Testing and validation of the proof-of-concept was designed as the last stage of the research process after the system implementation. The engineering method required testing and validation to be undertaken at the end of the process.

3.6.1 EXPLANATION

Step 1: In step 1 we will implement the cloud simulator in net beans. And put various number of resources and virtual machines.

Step 2: In second step we will be having plenty of tasks requests from various types of clients. So that scarcity of the resources can be shown.

Step 3: In this step we will schedule the resources using genetic based algorithm known as PSO.

Step 4: Identify the various performance parameters like energy and time.

3.7 Algorithm for Proposed

Step 1: Set the algorithm parameters: size of population and dimensions, maximum number of Iterations, Tmax or the expected detected entropy, the inertia weight Wmax, Wmix and the optimal solution set.

Step 2: Initialize each population, calculate the fitness value of each particle Fij (x1, x2, x3).

Step 3: Particles conduct migration in the system in accordance with each particle’s value Fi (x1, x2, x3) in which following the principle: follow the migration process, the particles move to immigration node with the minimal cost (Fi (x1, x2, x3) value is the smallest node). The fitness value of each particle compare with Pbest, if the current position is better, it will be the best position Pbest;

Step 4: For each particle, compare its fitness value and the best position it passed Pbest. If the former is better, take it as current best position.

Step 5: Obtain particle’s new speed and position based on the formula of iterations. When the load value of immigration node is equal to the total load value /number of nodes of the threshold value of system, then the node is automatically removed.

Step 6: The termination condition: the number of iterations reaches the maximum number of iterations set or achieve set detection value (entropy) of best resource for allocation.
3.8 Flowchart of Proposed work

In this the proposed work is represented with help of symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

Figure 3.4: Proposed flowchart of work
3.9 GENETIC ALGORITHM

3.9.1 Particle Swarm Algorithm Technique
PSO is a dynamic load balancing technique. It can be used in decentralized load balancing. It is used for dynamic decentralized optimum load balancing at broker level.
PSO is a swarm based heuristic optimization technique. It is developed by observing the social and biological behavior of swarm intelligence i.e. movement behavior of bird flocks and fish flock. The birds are scattered throughout the searching process for food. While the birds are searching for food from one source to other one, there is always a bird from the bird flock that can sense the food source very well. That bird is detectable of the lay where the in the flock are transmitting the information about their location for the food, it helps them to move nearer towards the food source.

3.9.2 PSO Steps
Step1: Set the algorithm parameters: size of population and dimensions, maximum number of Iterations, Tmax or the expected detected entropy, the inertia weight Wmax, Wmix and the optimal solution set.
Step2: Initialize each population, calculate the fitness value of each particle Fij (x1, x2, x3).
Step3: Particles conduct migration in the system in accordance with each particle’s value Fi (x1, x2, x3) in which following the principle: follow the migration process, the particles move to immigration node with the minimal cost (Fi (x1, x2, x3) value is the smallest node) .The fitness value of each particle compare with Pb est, if the current position is better, it will be the best position Pbest;
Step4: For each particle, compare its fitness value and the best position it passed Pbest .If the former is better, take it as current best position.
Step5: Obtain particle’s new speed and position based on the formula of iterations. When the load value of immigration node is equal to the total load value /number of nodes of the threshold value of system, then the node is automatically removed.
Step6: the termination condition: the number of iterations reaches the maximum number of iterations set or achieve set detection value (entropy) of load balancing.
3.10 Objectives
1. To build a cloud with variable speed of resource.
2. To build a cloud with variable execution time tasks.
3. To implement RASA (Resource Awareness Scheduling Algorithm).
4. To implement genetics-based algorithm.
5. Compare the performance of both the algorithms.

3.11 Parameters Taken
1. Energy Consumed
2. Throughput.
3. Total Time Taken
CHAPTER 4

RESULTS AND DISCUSSIONS

In current research the improvement over to the RASA algorithm has been done. In previous research resource awareness is performed. This resource awareness is to know the resource current status before being allocated to the process. In current research based on the PSO optimal resource is identified. Which is better ways of making cloud scheduling efficient.

4.1 RASA Implementation

<table>
<thead>
<tr>
<th>Start Time</th>
<th>Finish Time(Ms)</th>
<th>Power Consumption(Joule)</th>
<th>Throughput(per second)</th>
<th>processor utilization(Ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>297</td>
<td>17.888</td>
<td>13.88854356</td>
<td>287</td>
</tr>
<tr>
<td>0</td>
<td>282</td>
<td>17.888</td>
<td>13.88854356</td>
<td>314</td>
</tr>
<tr>
<td>0</td>
<td>289</td>
<td>17.888</td>
<td>13.88854356</td>
<td>222</td>
</tr>
<tr>
<td>0</td>
<td>280</td>
<td>17.888</td>
<td>13.88854356</td>
<td>291</td>
</tr>
<tr>
<td>0</td>
<td>262</td>
<td>17.888</td>
<td>16.88854356</td>
<td>225</td>
</tr>
<tr>
<td>0</td>
<td>246</td>
<td>17.888</td>
<td>13.88854356</td>
<td>316</td>
</tr>
<tr>
<td>0</td>
<td>251</td>
<td>17.888</td>
<td>15.88854356</td>
<td>311</td>
</tr>
<tr>
<td>0</td>
<td>228</td>
<td>17.888</td>
<td>16.88854356</td>
<td>287</td>
</tr>
<tr>
<td>0</td>
<td>221</td>
<td>17.888</td>
<td>15.88854356</td>
<td>221</td>
</tr>
<tr>
<td>0</td>
<td>309</td>
<td>17.888</td>
<td>15.88854356</td>
<td>303</td>
</tr>
<tr>
<td>200</td>
<td>457</td>
<td>17.888</td>
<td>14.88854356</td>
<td>231</td>
</tr>
<tr>
<td>200</td>
<td>438</td>
<td>17.888</td>
<td>16.88854356</td>
<td>264</td>
</tr>
<tr>
<td>200</td>
<td>434</td>
<td>17.888</td>
<td>13.88854356</td>
<td>267</td>
</tr>
<tr>
<td>200</td>
<td>484</td>
<td>17.888</td>
<td>15.88854356</td>
<td>259</td>
</tr>
<tr>
<td>200</td>
<td>490</td>
<td>17.888</td>
<td>14.88854356</td>
<td>245</td>
</tr>
<tr>
<td>200</td>
<td>458</td>
<td>17.888</td>
<td>15.88854356</td>
<td>274</td>
</tr>
<tr>
<td>200</td>
<td>446</td>
<td>17.888</td>
<td>13.88854356</td>
<td>299</td>
</tr>
<tr>
<td>200</td>
<td>462</td>
<td>17.888</td>
<td>13.88854356</td>
<td>290</td>
</tr>
<tr>
<td>200</td>
<td>465</td>
<td>17.888</td>
<td>17.88854356</td>
<td>271</td>
</tr>
<tr>
<td>200</td>
<td>513</td>
<td>17.888</td>
<td>13.88854356</td>
<td>280</td>
</tr>
<tr>
<td>0</td>
<td>314</td>
<td>17.888</td>
<td>17.88854356</td>
<td>286</td>
</tr>
<tr>
<td>0</td>
<td>228</td>
<td>17.888</td>
<td>17.88854356</td>
<td>317</td>
</tr>
<tr>
<td>0</td>
<td>254</td>
<td>17.888</td>
<td>14.88854356</td>
<td>282</td>
</tr>
<tr>
<td>0</td>
<td>289</td>
<td>17.888</td>
<td>14.88854356</td>
<td>253</td>
</tr>
<tr>
<td>0</td>
<td>292</td>
<td>17.888</td>
<td>14.88854356</td>
<td>315</td>
</tr>
<tr>
<td>0</td>
<td>239</td>
<td>17.888</td>
<td>13.88854356</td>
<td>245</td>
</tr>
<tr>
<td>0</td>
<td>271</td>
<td>17.888</td>
<td>16.88854356</td>
<td>248</td>
</tr>
</tbody>
</table>
This table shows the cloud implementation performance parameters. These performance parameters include throughput, Power Consumption, Throughput and processor utilization. It is based on RASA algorithm.

### 4.2 PSO Implementation

<table>
<thead>
<tr>
<th>Start Time(Ms)</th>
<th>Finish Time(Ms)</th>
<th>Power Consumption(Joule)</th>
<th>Throughput(per second)</th>
<th>processor utilization(Ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>302</td>
<td>16.124516</td>
<td>17.84555749</td>
<td>299</td>
</tr>
<tr>
<td>0</td>
<td>219</td>
<td>17.66352</td>
<td>18.11643349</td>
<td>233</td>
</tr>
<tr>
<td>0</td>
<td>250</td>
<td>16.462078</td>
<td>19.43861511</td>
<td>219</td>
</tr>
<tr>
<td>0</td>
<td>266</td>
<td>16.124516</td>
<td>18.84555749</td>
<td>218</td>
</tr>
<tr>
<td>0</td>
<td>246</td>
<td>15.652476</td>
<td>19.44404947</td>
<td>286</td>
</tr>
<tr>
<td>0</td>
<td>242</td>
<td>16.093477</td>
<td>18.88383213</td>
<td>277</td>
</tr>
<tr>
<td>0</td>
<td>239</td>
<td>15.394804</td>
<td>20.78623411</td>
<td>268</td>
</tr>
<tr>
<td>0</td>
<td>222</td>
<td>16.852299</td>
<td>16.98850744</td>
<td>265</td>
</tr>
<tr>
<td>0</td>
<td>213</td>
<td>16.703293</td>
<td>17.15789916</td>
<td>296</td>
</tr>
<tr>
<td>0</td>
<td>214</td>
<td>17.117243</td>
<td>18.69459956</td>
<td>287</td>
</tr>
<tr>
<td>200</td>
<td>471</td>
<td>17.20465</td>
<td>17.59962183</td>
<td>201</td>
</tr>
<tr>
<td>200</td>
<td>453</td>
<td>15.652476</td>
<td>19.44404946</td>
<td>309</td>
</tr>
<tr>
<td>200</td>
<td>420</td>
<td>17.464249</td>
<td>17.32314726</td>
<td>217</td>
</tr>
<tr>
<td>200</td>
<td>489</td>
<td>16.763054</td>
<td>18.08960038</td>
<td>251</td>
</tr>
<tr>
<td>200</td>
<td>402</td>
<td>16.911535</td>
<td>17.92199584</td>
<td>222</td>
</tr>
<tr>
<td>200</td>
<td>448</td>
<td>15.779734</td>
<td>20.27917624</td>
<td>270</td>
</tr>
<tr>
<td>200</td>
<td>446</td>
<td>16.673332</td>
<td>19.19232435</td>
<td>308</td>
</tr>
<tr>
<td>200</td>
<td>463</td>
<td>17.14643</td>
<td>18.66277805</td>
<td>285</td>
</tr>
<tr>
<td>200</td>
<td>433</td>
<td>15.905973</td>
<td>18.11822799</td>
<td>312</td>
</tr>
</tbody>
</table>
This table shows the various performance parameters reading like throughput, power consumption, Processor utilization under PSO.

4.3 Comparison of Power for RASA and PSO

<table>
<thead>
<tr>
<th>Power Consumption of PSO(J)</th>
<th>Power Consumption Of RASA(J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.124516</td>
<td>17.888</td>
</tr>
<tr>
<td>17.66352</td>
<td>17.888</td>
</tr>
<tr>
<td>16.462078</td>
<td>17.888</td>
</tr>
<tr>
<td>16.124516</td>
<td>17.888</td>
</tr>
<tr>
<td>15.652476</td>
<td>17.888</td>
</tr>
<tr>
<td>16.093477</td>
<td>17.888</td>
</tr>
<tr>
<td>15.394804</td>
<td>17.888</td>
</tr>
<tr>
<td>16.852299</td>
<td>17.888</td>
</tr>
<tr>
<td>16.703293</td>
<td>17.888</td>
</tr>
<tr>
<td>17.117243</td>
<td>17.888</td>
</tr>
<tr>
<td>17.20465</td>
<td>17.888</td>
</tr>
<tr>
<td>15.652476</td>
<td>17.888</td>
</tr>
<tr>
<td>17.464249</td>
<td>17.888</td>
</tr>
</tbody>
</table>
Table 4.3 Comparison of Power for RASA and PSO

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16.763054</td>
<td>17.888</td>
</tr>
<tr>
<td>16.911535</td>
<td>17.888</td>
</tr>
<tr>
<td>15.779734</td>
<td>17.888</td>
</tr>
<tr>
<td>16.673332</td>
<td>17.888</td>
</tr>
<tr>
<td>17.14643</td>
<td>17.888</td>
</tr>
<tr>
<td>15.905973</td>
<td>17.888</td>
</tr>
<tr>
<td>15.996685</td>
<td>17.888</td>
</tr>
<tr>
<td>15.066519</td>
<td>17.888</td>
</tr>
<tr>
<td>17.262676</td>
<td>17.888</td>
</tr>
<tr>
<td>17.776388</td>
<td>17.888</td>
</tr>
<tr>
<td>16.852299</td>
<td>17.888</td>
</tr>
<tr>
<td>15.556349</td>
<td>17.888</td>
</tr>
<tr>
<td>17.406895</td>
<td>17.888</td>
</tr>
<tr>
<td>15.491934</td>
<td>17.888</td>
</tr>
<tr>
<td>16.492422</td>
<td>17.888</td>
</tr>
<tr>
<td>17.720045</td>
<td>17.888</td>
</tr>
<tr>
<td>15.264338</td>
<td>17.888</td>
</tr>
<tr>
<td>17.175564</td>
<td>17.888</td>
</tr>
<tr>
<td>17.521416</td>
<td>17.888</td>
</tr>
<tr>
<td>15.459625</td>
<td>17.888</td>
</tr>
<tr>
<td>17.58722</td>
<td>17.888</td>
</tr>
<tr>
<td>15.6205</td>
<td>17.888</td>
</tr>
<tr>
<td>15.9687195</td>
<td>17.888</td>
</tr>
<tr>
<td>16.822603</td>
<td>17.888</td>
</tr>
<tr>
<td>15.874508</td>
<td>17.888</td>
</tr>
<tr>
<td>17.117243</td>
<td>17.888</td>
</tr>
<tr>
<td>17.888544</td>
<td>17.888</td>
</tr>
</tbody>
</table>

This table shows the comparisons for power consumption for both RASA and PSO.

4.4 Comparison of average for power for RASA and PSO

<table>
<thead>
<tr>
<th>Average Power consumption</th>
<th>Average power consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO(J)</td>
<td>RASA(J)</td>
</tr>
<tr>
<td>16.54030369</td>
<td>17.888</td>
</tr>
</tbody>
</table>

Table 4.4 average for power for RASA and PSO

This table shows the average power consumption for both PSO and RASA.
4.5 Comparison Graph for Power of RASA and PSO

![Comparison Graph for Power of RASA and PSO](image)

This graph shows the comparison for power consumption of PSO and RASA. Clearly it is depicted PSO is more efficient compared to the RASA. Such that less energy is required to schedule the resources for the process for the efficiency of the process. It is the total power consumed while executing the task from a given resource. In case of PSO the power consumption is less compared to the RASA. That means in context to the power the PSO performance has improved to 16.53%.

4.6 Comparison of Throughput for PSO and RASA

<table>
<thead>
<tr>
<th>Throughput of PSO (per second)</th>
<th>Throughput of RASA (per second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.84555749</td>
<td>13.88854356</td>
</tr>
<tr>
<td>18.11643349</td>
<td>17.88854356</td>
</tr>
<tr>
<td>19.43861511</td>
<td>13.88854356</td>
</tr>
<tr>
<td>18.84555749</td>
<td>13.88854356</td>
</tr>
<tr>
<td>19.44404947</td>
<td>16.88854356</td>
</tr>
<tr>
<td>18.88383213</td>
<td>13.88854356</td>
</tr>
<tr>
<td>20.78623411</td>
<td>15.88854356</td>
</tr>
<tr>
<td>16.98850744</td>
<td>16.88854356</td>
</tr>
<tr>
<td>17.15789916</td>
<td>15.88854356</td>
</tr>
<tr>
<td>18.69459956</td>
<td>15.88854356</td>
</tr>
<tr>
<td>17.59962183</td>
<td>14.88854356</td>
</tr>
<tr>
<td>19.44404946</td>
<td>16.88854356</td>
</tr>
<tr>
<td>17.32314726</td>
<td>13.88854356</td>
</tr>
</tbody>
</table>
4.7 Comparison of Average Throughput for RASA and PSO

<table>
<thead>
<tr>
<th>Average Throughput for PSO</th>
<th>Average Throughput for RASA</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.5866685</td>
<td>15.51354356</td>
</tr>
</tbody>
</table>

Table 4.7 Average Throughput for RASA and PSO

This table shows the comparison for Average Throughput for both RASA and PSO.
4.8 Comparison Graph for Throughput of RASA and PSO

This graph depicts the throughput comparison for PSO and RASA. Throughput has improved over to the RASA. It is the performance parameter in terms to number of processes per unit interval of time. In case of PSO the throughput has improved to 8.11%.
CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

From the current research it is clear that cloud efficiency will depend upon this issue that how well cloud schedules the resources amongst different processes. MAX-MIN and MIN-MIN individually are not so efficient because there may be various longer or even shorter tasks. For optimization of the selection process in current research we have used PSO. This technique identifies the best possible resources amongst the multiple available resources. In previous research the RASA based technique was used. Performance parameters like throughput, power consumption has been used to compare the performance of previous and current research. PSO has improved upon the power consumption and throughput. Power consumption has improved 16.53%, and throughput has improved upon 8.11%.

5.2 FUTURE WORK

In future another genetic based approach can be tested and compared with PSO. So that best optimization technique can be identified.
REFERENCES

[1] Shuibing He, Yang Wang, Xian-He Sun IEEE Fellow, and Chengzhong Xu,” Using MinMax-Memory Claims to Improve In-Memory Workflow Computations in the Cloud”, 1045-9219 (c) 2016 IEEE.


[16] Shuibing He, Yang Wang, Xian-He Sun IEEE Fellow and Chengzhong Xu, “Using Min-Max-Memory Claims to Improve In-Memory Workflow Computations in the Cloud”, 1045-9219 (c) 2016 IEEE.


Buy your books fast and straightforward online - at one of world’s fastest growing online book stores! Environmentally sound due to Print-on-Demand technologies.

Buy your books online at www.morebooks.shop

Kaufen Sie Ihre Bücher schnell und unkompliziert online – auf einer der am schnellsten wachsenden Buchhandelsplattformen weltweit! Dank Print-On-Demand umwelt- und ressourcenschonend produziert.

Bücher schneller online kaufen
www.morebooks.shop