JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Algorithmic Study of Cloud Computing for Resource Optimization Based On Virtual Machines

Ms. Rachna Pandey[1] Prof.(Dr.) R.K.Bathla[2]

Ph.D Research Scholar, Department of CSA, Desh Bhagat University,Mandi Gobindgarh, Punjab, India[1]
Professor, Department of CSA, Desh Bhagat University,Mandi Gobindgarh,, Punjab, India[2]

Abstract

The expression cloud moves toward from the symbol used to represent internet or network. It is ahead lotof popularity as it provides low-cost access to high computing resources and extremely large storage spaces with high level security. Cloud computing refers to many dissimilar types of services and applications being delivered over the internet cloud. Cloud computing provides a well-organized and reasonably priced means of providing platforms such as IaaS (Infrastructure as a service), SaaS (Software as a service), PaaS (Platform as a service) and much more to the service requesters. A large number of commercial applications are available that provide widerange of services over cloud. Increased and fast access to internet has also led to a boom in this sector. Different services are put within the virtualized resources of a cloud, enabling it to carry out abstractions of its underlying resources. Cloud computing environment provides virtualized resources to applications dynamically. These resources are costly and oftenimpossible for users to own them privately. Users are charged on a pay-per-use basis for theseresources on cloud environment.

Setting up of tasks in cloud environment is a key research area. The problem involves allocation of almost infinite number of tasks on cloud resources that cannot be solved in polynomial time and hence is considered to be a NP-hard problem. The solution provided to close to optimal as no algorithm exists that provides optimal solution in finite time. There areno algorithms which may produce optimal solution within polynomial time to solve these problems. The idea of task scheduling in cloud environment is to optimize the execution timeso that the cost paid by the user is minimal. Task scheduling is the process of distributing workloads across multiple computing resources. Load balancing is an optimization problem and goal of any optimization is to either minimize effort or to maximize benefit. The effort orthe benefit can be usually expressed as a function of certain design variables. Hence,optimization is the process of finding the conditions that give the maximum or the minimumvalue of a function. Load balancing is a problem where you try to minimize value of parameters like Makespan time, Response Time, etc. and increase the utilization of cloud resources.

Keywords: Cloud Computing, Load balancing, IAAS.

1. INTRODUCTION

1.1Progression of Cloud Computing: -

Cloud computing, a term with roots in the 1960s, was popularized through Remote Job Entry (RJE) and implemented by vendors like IBM and DEC. Full time-sharing solutions were available in 1970 on platforms like

Multics and Cambridge CTSS [1]. In the early 1990s, cloud technology emerged with the introduction of virtual private network (VPN) services, enabling better quality services at lower costs and efficient network bandwidth utilization [1].

In the early 2000s, NASA's OpenNebula, the first open-source software for deploying private and hybrid clouds, aimed to improve service quality for real-time interactive applications [1].

Internet access has revolutionized computing resources by enabling easy pay-per-use rental of storage space and computing power for hours or minutes.

1.2 Cloud Computing

Cloud computing refers accessing application or saving information on the internet rather than on local storage devices. Cloud is synonym to internet or services that wecan get over the internet and there is no need to create the infrastructure like storage or high-end processing at local sites. You simply need fast access to the internet so that one feels like accessing the data on local host itself. There are many advantagesthat are provided by cloud computing, as shown in Fig.1.

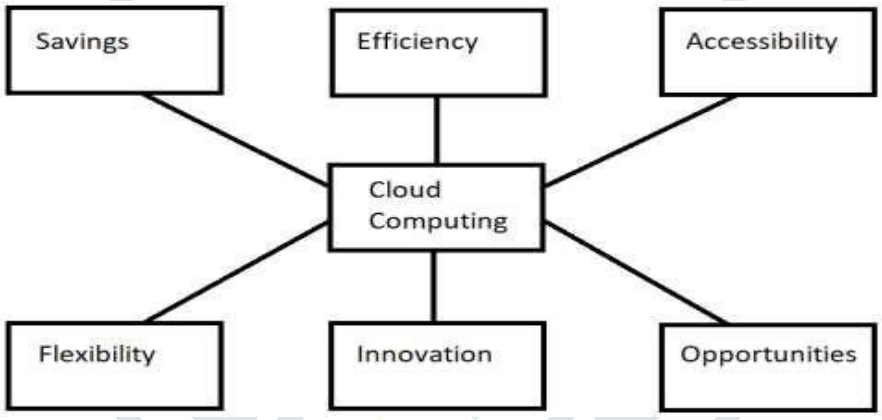


Fig. 1: Cloud Computing

Some other major examples of cloud computing services are:

- **Google Drive:** Google, a major player in cloud services, has seen a significant increase in multimedia data generated by mobile phones. With low memory onboard, Google Drive offers online storage and supports cloud apps like Google Docs, Sheets, and Slides, which use minimal device resources [2]. The google storage services are free of cost and most widely used world-wide.
- **Apple iCloud:** Apple's cloud service provides limited access and is mainly used for backup, online storage, contacts, synchronization of your mail, calendar and more from the apple devices only [2].
- **Amazon Cloud Drive:** Amazon cloud drive offers commercial storage for digital data, including MP3s and videos, accessible through Amazon Prime and other web services, providing large storage space for all digital content [2].

1.3 Service Models

The usage of cloud services after the cloud has been successfully built depends upon the business model of the end-user. What is the requirement of the end-user directs the deployment model to be used.

- **Software as a Service (SaaS)** — SaaS (Software as a Service) allows end users to access cloud-hosted applications or services, like Salesforce.com, through a browser, allowing multiple clients to access the same application over the web [3].
- **Platform as a Service (PaaS)** — PaaS is a web-based application deployment platform for developers, allowing them to rent access to infrastructure software, databases, middleware, and development tools, without managing operating systems or network access [3].
- **Infrastructure as a Service (IaaS)** — The cloud service model allows customers to use resources on demand without long-term commitment, utilizing hardware and software as a service, without owning the cloud infrastructure [3].

Table 1: Major cloud computing vendor

Vendor	IaaS	PaaS	SaaS	Storage
Amazon	EC2 (Elastic Cloud Compute)	Amazon Web Services*	Amazon Web Services*	S3 (Simple Storage Service)
Google	n/a	Google App Engine (Python, Java, Go)	Google Aps	Google Cloud Storage
HP	Enterprise Services Cloud – Compute	Cloud Application Delivery	HP Software as a Service	Enterprise Services Cloud – Compute
IBM	SmartCloud Enterprise	SmartCloud Application Services	SaaS products	SmartCloud Enterprise – object storage
Microsoft	Microsoft Private Cloud	Windows Azure (includes .NET, Node.js, Java, PHP)	MS Office 365	Microsoft Private Cloud
JoyentCloud	Smart Machines	Node.js	n/a	n/a
Rackspace	Cloud Servers	Cloud Sites	Email & Apps	Cloud Files
Salesforce.com	n/a	Force.com	Salesforce.com	n/a

VMware**	VMware vSphere,vCloud	VMware vFabric (Java Spring), vCloud API	n/a	n/a

Google — A private cloud offers various services to end users, including data storage, email, document editing, text translation, and mapping, both free and pay-per-usage [3]. Google provides most of its services free of cost and its storage services are most widely used in the world.

Microsoft — It provides a powerful tool called Share point® which is very powerful cloud service provider. Users can move business intelligence tools and other content over the cloudand use it's popular office applications to edit these document over the cloud [3].

Salesforce.com — Customers can use its application over the cloud. Further, the users can also develop or build customized cloud services by using the platform provided by its productsover force.com and vmforce.com [3].

Amazon — Amazon Web Services (AWS) offers a secure cloud platform for businesses, providing data storage, computing power, and content delivery, enabling users to create scalable and flexible applications for growth.

1.4 Deployment Model

Deployment model determines how an organization uses a cloud platform, either as a private, public, or a mix of both, based on project requirements and scale.

Public Clouds

Public cloud offers off-site internet access to end-users, though more vulnerable than private cloud, but offers greater efficiency in resource sharing. Users prefer public cloud in the following scenarios:

- When the requirement of all users is based on some standardized parameters, like email services.
- Need of SaaS from a service provider with high-end security.
- When the requirement of resources by the user is dynamic. That means lesser requirement of resources at lean time and more requirement of resources at the peak time.

Security and reliability are the major challenges faced while deploying a public. Proper planning must be done to take care of the governance issues and issues related to the securityof data over cloud [5].

Private Cloud

Private cloud offers more reliability than public cloud due to private network maintenance of data and services, but requires company purchase of software and infrastructure. End-use prefers private cloud over public cloud in following scenarios:

- Control and security of enterprise data and applications is highest priority and it cannot be handed over to

third party. Misuse of data is a common practice now adays by major cloud service providers.

- Company deals with a business in which data security is of prime importance andneeds to adhere to strict security protocols.
- Company has enough funds to setup and manage high-end future data centerseffectively.

Considering the benefits and shortcoming of both the public cloud and the private cloud, the companies prefer to hybrid the two to over the shortcomings of each. Some of the services arekept as part of private cloud while other services are migrated to public cloud.

Hybrid Cloud

Hybrid cloud, a concept where multiple providers offer both private and public cloud options, allows better control over services, but challenges include communication across different cloud types. Hybrid cloud is preferred in case of flowing scenarios:

- The end-user wants to use a standardized service like SaaS, but the user is concerned about the security issues. In this case, the service provider can create a VPN enabled private cloud especially for that user in the public cloud space.
- In the second scenario, a company can use a private cloud to store all the data and stillcommunicate with the clients over the tailored made public cloud to over the securityissues.

Most of the hybrid clouds are offering protection to provide security services you can find intraditional dedicated environment like as Intrusion Prevention System (IPS), Web Application Firewalls (WAF), Security Information and File Integrity Monitoring (FIM) [5].

1.5 Challenges in Cloud Computing

Cloud computing offers numerous benefits to individuals and companies, but also faces significant challenges in information management. Planning ahead is crucial for fully utilizing cloud services and overcoming these obstacles.

A complete cloud service model must take care of all aspects before it can be successfully implemented. All the aspects must be taken care of before the model is put in use. These challenges are discussed as follow [5]:

Table 2: Challenges of cloud computing

Sr. No.	Challenges	Description
1.	**Data Management and Resource Allocation**	Resource allocation is vital for the service provider. Resource provision must be done in a manner so as to maximize the profits of the cloud service provider. Resources must be fully utilized to maximize profits. Data management and security is vital for building confidence in customers.
2.	**Load Balancing**	It is related to download performance, utilization of storage & computational resources. This problem mainly occurs in distributed nodes.
3.	**Availability and Scalability**	Performance degradation and oversizing problems occur due to unpredictability in cloud. Making correct estimates about the resources that are required to complete a task in specified time is key.
4.	**Compatibility & Migrationto Clouds**	Cloud migration is a major challenge. Cloud migration is hit by challenges like very little understanding of the cloud structure.
5.	**Interoperability and Communication Between Clouds**	The process of achieving interoperable cloud computing environments are becoming more challengeable due to various shortcomings.

2. LITERATURE SURVEY

Processing elements or machines in cloud computing under static environment installs homogeneous resources. These resources in the cloud are not flexible when environment is made static. In this scenario, the cloud requires prior knowledge of nodes capacity, processing power, memory, performance and statistics of user requirements. These user requirements are not subjected to any change at run-time. Algorithms proposed to achieve load

balancing in static environment cannot adapt to the run time changes in load. Although static environmentis easier to simulate but is not well suited for heterogeneous cloud environment.

FCFS is the easiest static scheduling algorithm in which task that arrives first will be scheduled first of all and resources are allocated to that task as it needs. All tasks for which no resources are available at any particular time are kept in a queue waiting for resources to get free. Once the task is executed, the next task in queue is scheduled next and assigned the free resource. FCFS is a basic scheduling method in which tasks are queued up if resources are busy. The algorithm evaluated arrival time of tasks as well the algorithm is easy to be implemented. There is no optimization function used to map the tasks to the resources and hence it is a pure form of greedy algorithm with no optimization features.

Round Robin is another static scheduling method in which task are executed for a fixedtime slot. The tasks are put in queue at the end and will be taken again when it will reach frontand remaining execution will be carried out. The algorithm calculates expected execution timeas well balances the load. The resources are provisioned to the task on first-cum-first-serve (FCFS- i.e. the task that entered first will be first allocated the resource) basis and scheduledin time sharing manner. It can be considered as greedy (first-fit) round-robin scheduling policy for mapping Virtual Machines and tasks under static environment.

Priority Scheduling Assigning priority to tasks is key to implementing solution to scheduling problem. Priority in cloud computing can be based on multiple attributes. Prioritybased job scheduling algorithm is suitable example of On-line mode heuristic scheduling algorithm. A particular job scheduling algorithm in cloud environments should pay attentionto multi-attribute and multi-criteria properties of jobs. Numerous mathematical functions existto assign weights to tasks so that the tasks can be mapped to resources. These functions are the used to generate a weight for each evaluation criterion according to the decision maker's pairwise comparisons of the criteria. Weights are computed for each task on all resources. Main focus of the priority scheduling algorithm is priority of the tasks and not the optimizationof resources.

Shortest Job First is a simple static priority-based scheduling algorithm in which the tasks are assigned priority on the basis of the task length. The task with the shortest length is assigned resources first. A queue is maintained if all resources are busy.

Improved Cost-Based Algorithm for Task Scheduling [69], is an improved cost-based algorithm so to make efficient mapping of tasks to resources which are available. Sorting of the tasks based on the priority (decided using some criteria) is done and these are further place in 3 different lists namely, high, medium and low priority. The main aim of this task scheduling was to result in minimum total tasks completion time and minimum cost. The algorithm did not address the issue of handling complicated scenario involving other QoS attributes.

Static & dynamic heuristic strategies

Static task scheduling strategies are preferred when it is certain that all tasks will reach at thesame time and will be considered for scheduling simultaneously. Dynamic strategies are particularly useful when the scheduler is not aware of all incoming tasks at the beginning or there is change in available machines dynamically. The tasks are passed to the scheduler as they arrive or in iterations. In certain architecture arrangements, the tasks arrive asynchronously and even some machines go offline at certain intervals. Dynamic heuristics can be used either in on-line mode or batch mode. In online mode, each task is scheduled to aparticular machine as it enters the system or is submitted to the broker. In the batch mode, alltasks are collected as part of a set and scheduling is performed as per the preschedule.

OLB (Opportunistic Load Balancing) begins by assigning the tasks randomly or in free order to available resources on the cloud. It does so by assigning workload to nodes in free order. The implementation is very easy as no computation is required and it does not consider any constraints while assigning tasks to resources. For instance, it does not considerthe expected execution time of task on different resources. Idea is to ensure that all resourcesor machines get work.

MET (Minimum Execution Time) is also simple and easy to execute. The tasks are assigned to the machines considering that it should take minimum time to execute the task. Itseems very valid. But it does not take into consideration the current load and availability of the machine. It simply assigns the task to the best machine. This strategy can result in poor load balance across various machines. Load balance Min-Min (LBMM) [70] is an example of Minimum Execution Time task scheduling algorithm.

MCT (Minimum Completion Time) is a strategy that works differently than the Minimum Execution Time. Rather than considering the Execution Time, it works on Completion Time of the task. It may take more time to execute, but the task is guaranteed to complete in minimum time as compared to other machines. Each task is assigned arbitrarily to the machines which possesses the minimum completion time to complete this task.However, the strategy fails to ensure that task takes minimum execution time.

MOMCT (Modified Ordered Minimum Completion Time) proposes an algorithm using which it is possible to identify MCT (Minimum Completion Time) that allocates tasks in a random order to the minimum completion time machine. It suggests an ordered approachto the MCT heuristic, which order tasks in accordance to the mean difference of thecompletion time on each machine and the minimum completion time machine.

Min-min is based on Completion Time of all tasks that are still waiting for the resource allocation. Idea is to compute the matrix for minimum completion time of every task which isstill waiting for resource allocation. Task with minimum completion time is scheduled to the respective machine on which its completion time is minimum. The task is then removed fromthe list of tasks that are waiting for resource allocation and the same procedure is followed forall the remaining tasks in the list.

Min-max is also based on Completion Time of tasks and is quite similar to Min-min heuristic on the basis of its implementation. Only difference between min-min and max-min is the selection of corresponding machine where it should execute. It also has a set of all unscheduled tasks. Again, we compute the matrix for minimum completion time of every task which is still waiting for resource allocation. But, rather than selecting the task with overall minimum time, here the task with overall maximum completion time is scheduled therespective machine on which its completion time is maximum. The task is then removed from the list of tasks that are waiting for resource allocation and the same procedure is followed forall the remaining tasks in the list.

In paper a review of different Task Scheduling schemes is discussed. Also, a novel taxonomy is proposed in the paper to solve the problem of task scheduling in cloud environment. Schemes falling under Goal Oriented Task Scheduling (GOTS) schemes give service providers a fair chance to apply specific approach and schedule the tasks and resourcesthat can generate maximum possible economic gains, while using least resource provisioning. Using low resource provision allows providers to use their resources at possible fullest and trading Makespan with marginal increase only.

In paper an Autonomous Agent Based Load Balancing Algorithm (A2LB) is proposed. The objective of this algorithms is to provide dynamic load balancing using ants asthe migration agent. Autonomous agent-based load balancing algorithm (A2LB) focuses on parameters like improving throughput, minimizing response time, dynamic resource scheduling with scalability and reliability. A2LB works by ensuring that all the resources are properly utilized and the resources are further used in a manner that the load remains balanced. A2LB mechanism comprises of three agents: Load agent, Channel gent and Migration Agent.Load and channel agents are static agents whereas migration agent is an ant. Load Agent is responsible to calculate the load on every available virtual machine after allocation of a new job in the data centre. It maintains all such information in table termed as VM_Load_Fitness table. Channel Agent initiates migration agents on receiving the request from load agent. The idea is to search for virtual machines with similar configuration. It maintains the information received from migration agent in table termed as Response Table. Migration Agent communicates with load agents of other datacenters to find a compatible VM whose fitness value is greater than some threshold value. In case any such VM is found, the channel agent migrates the task to that VM. The proposed mechanism has been implemented and found to provide good results. Only problem with this algorithm is the number of migrations. Cloudlets are randomly assigned to Virtual machines and once all Virtual Machines are busy, the algorithm spends much of its time to find a suitable alternative Virtual Machine and fails to ensure that the resource optimization is fully achieved.

3. CLOUD COMPUTING ARCHITECTURE

This chapter discusses the architectural specifications and working of cloud services platform. Cloud computing is all about different technologies and services that put together form a cloudand how these technologies work together to make it a huge success. This chapter explains theworking of each and every module of the cloud platform and the role played by them in cloudspace. It also explain the various benefits and limitations of various cloud components.

3.1 Cloud Architecture

Cloud architecture consists of various cloud components, including front-end and back-end components connected over the internet, enabling access to various services. An cloud computing architecture in general consists of the following [11]:

- ✓ A front-end platform using which a consumer/ end-ser accesses various services over the cloud using various types of devices like mobile, laptop, PC, etc.
- ✓ Back-end platforms to provide services like servers for storage or computing.
- ✓ Cloud-based delivery
- ✓ A network (internet, intranet)

Front-End

User can access the cloud services over desktop or mobile devices called the front-end. Front-end is the side that is visible to the client, customer, or user [11]. The user interface is the main front-end component, allowing users to access services through their computer system and cloud network, and can vary depending on the services used and provider.

Back-End

The front-end of a cloud service refers to the user, while the back-end is the provider, encompassing servers, computer systems, data storage, virtual machines, and programs [11]. The back-end of cloud architecture ensures data and applications security, manages network traffic, and facilitates communication over the network, in addition to providing basic services.

Cloud Based Delivery

Knowingly or Un-knowingly, cloud services are being used by each and every entity who is using internet. It may be individuals or companies. You may be using cloud services on pay-per-use basis from reputed cloud service providers Microsoft Azure, Amazon AWS, etc. or free services by google like google docs [10].

Companies or individuals can purchase subscriptions for SaaS, PaaS, or IaaS services, which enable virtualization on a physical server, dividing it into multiple virtual servers for different tasks or users [11].

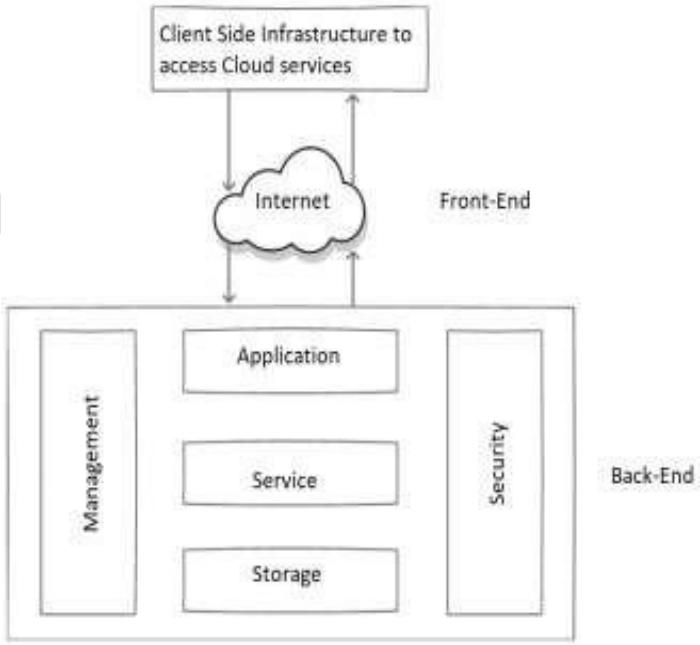
Cloud Services Network

Deployment of the cloud services can be private, public or hybrid, but you need a network using which the services can be delivered to the user [11]. Network quality is crucial for cloud implementation success, especially when combining internet and intranet for hybrid models, especially when datacenters are distant [11].

Architecture is the design and construction of a cloud infrastructure, crucial for successful implementation and preventing component failures that could lead to the entire cloud failure. Following are some of the key features of cloud architecture:

- **Multi-Tenant or Multi-User system:** Cloud infrastructure is designed to serve multi-users at the same time. Multi-tenancy is a key feature of operating system which lets multiple usersto pool together for a resource and hence having to pay very less for the use of that resource for a certain period of time [12].
- **Pay as you use:** Customers pay only for the amount of resources they use. There is no need to get into long term contract with the service provider and you can even rent resources for just a few hours [12].
- **Virtualization:** Re-usage of hardware and software can be achieved through the conceptcalled virtualization.

Fig. 2: Cloud Architecture



3.2 Components of Cloud

Following are different components of cloud that are part of the cloud implementation [12].

Subscriber: Cloud basically creates a pool of resources for people who use the cloud servicesin any form such as SaaS, PaaS or IaaS model. These users are either allowed access to theseresources free of cost of they have to pay minor cost for the same.

Brokers: Brokers represent users' interests and combine various requirements into standardized requirements, saving them from cloud complexity. They act as both service consumers and provisioners, providing services like identity management, performance reporting, security, and data integration.

Service conceptualizer: They are responsible for developing the services and presenting the same to users/brokers either free of cost or at a charge of nominal fee. They are also responsible for hosting the services on the cloud environment offered by a cloud service provider.

Resources Allocator: It coordinates between the users and the cloud service provider. This entity ensures the resources management and guarantees that the resources are provided to the end user as per the set service levels.

Cloud Provider: Cloud providers, such as Amazon, Microsoft, IBM, and Google, provide computing infrastructure and services to users through the internet. They offer SaaS (SaaS), PaaS (PaaS), IaaS (IaaS), and other services depending on their role. SaaS providers take ownership of applications and infrastructure, PaaS manages infrastructure, and IaaS hosts virtual machines and provides virtual network interfaces.

Service request examiner and controller: Group of people responsible for allocating and re- allocating of resources based on predefined priority.

VM Monitor: Virtual machine (VM) monitor is responsible for looking after the virtual machines and their availability.

Further, cloud ecosystem consists of users responsible for providing connectivity between the cloud services user devices, accepting service requests and allocation of virtual machines to end-user, comparing the performance levels of services with the agreed service levels.

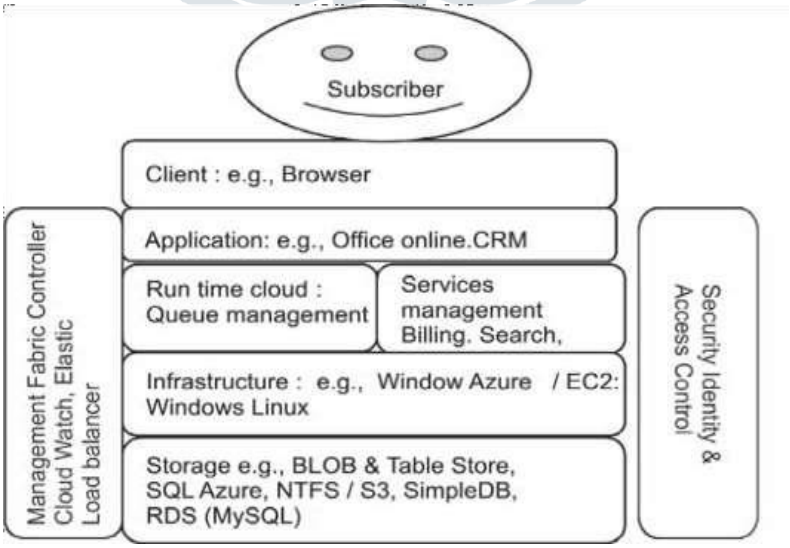


Fig. 3: Cloud architecture components.

3.3 Virtualization

Virtualization revolutionizes IT by reducing equipment usage, energy, and costs, enabling simultaneous application and framework running on a single server, and expanding hardware efficiency and adaptability. [15].

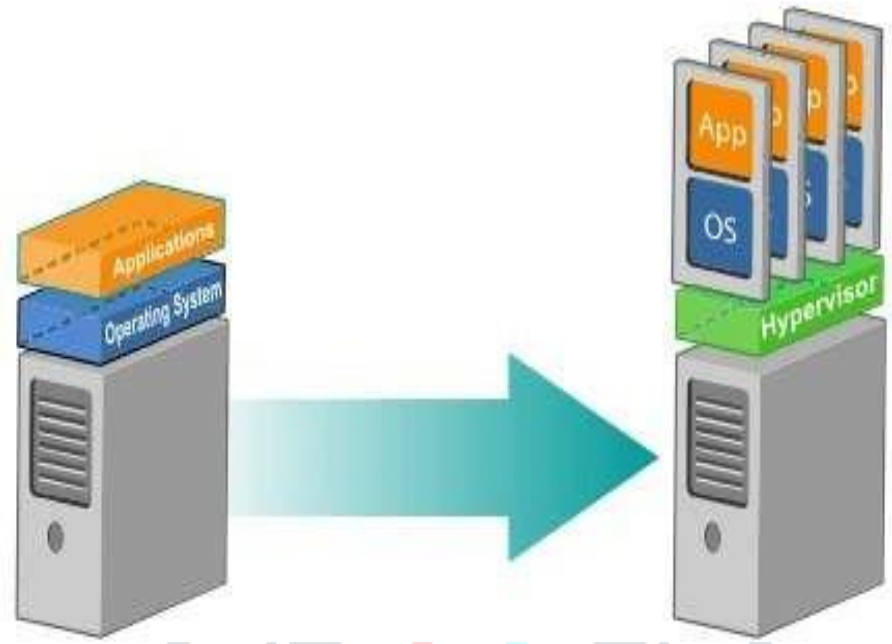


Fig. 4: Virtualization in Cloud Environment

Advantages of Virtualization

Virtualization provides different advantages including saving in time and energy, diminishing expenses and limiting general hazard.

- Gives capacity to oversee resource successfully.
- Increases efficiency, as it gives secure remote access.
- Gives for information damage prevention.

A special program is needed that makes the concept of virtualization possible. This product is known as a Hypervisor also known as virtualization administrator. It provides an interface between the equipment and the working framework. It also defines the rules for an application to gain access to processor and other hardware resources through the working framework.

It is less demanding to comprehend virtualization once we think about various types of virtualization, which are discussed as below:

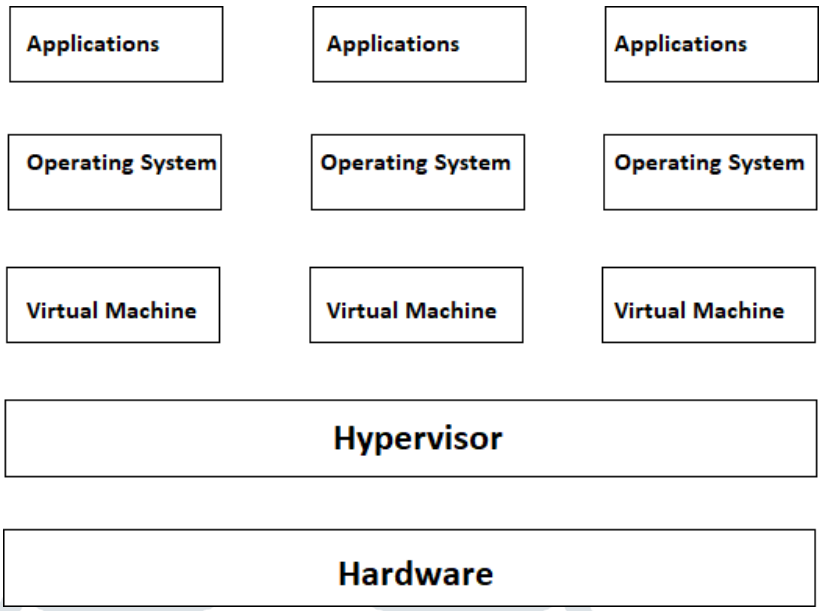


Fig.5 Structure of Hypervisor

4. METHODOLOGY

4.1.1 Problem Formulation

The huge amount of computations a Cloud can fulfil in a specific amount of time cannot be performed by the best supercomputer. However, Cloud performance can still beimproved by making sure all the resources available in the Cloud are utilized optimally using a good load balancing algorithm. Statistical results of various research papers in the literature review indicate that intensive local search improves the quality of solutions found in such dynamic load balancing algorithms. Moreover, the multi-population approaches obtain better quality with less computational effort. In the existing algorithm, Loadbalancing is done by using ant colony optimization technique in cloud computing by considering autonomous agent approach in order to reduce the issues like migration of task to the available resources in case the load of nodes reaches the threshold level. In the existing algorithm, only the load of machines is considered and it does not work on proper resource utilization. **In the proposed work the execution time for balancing the load and improving the resource utilization will be minimized**

4.1.2 Research Gap

- In the previous study, the load balancing has not been studied separately from theoptimization of resources.
- The existing study fails to efficiently implement the aggregation of two or more loadbalancing techniques.
- The existing study lacks in optimizing both the Data Centre Service Time andTransfer Cost.

4.1.3 Objective

- To study and analyze various meta heuristic load balancing algorithms.
- To design efficient algorithm for providing system load balancing using Novel Hybrid(ACO & PSO) based technique.
- To design efficient algorithm for providing the system load balancing using Novel GAtechnique.

- To compare the results of A2LB Algorithm with the proposed Hybrid (ACO & PSO)based technique and Novel GA based technique for load balancing on the basis of:
  - Over all response time
  - Data Center Service Time
  - Transfer Cost.
  - To minimize the execution time for improving the resource utilization of the balanced machines by using Novel Resource Aware Scheduling Algorithm.
  - To compare the results of existing resource aware algorithm with the proposed Novel Resource Aware Scheduling algorithm on the basis of:
    - Over all response time
    - Data Center Service Time
    - Transfer Cost.

4.1.4 Introduction to CloudSim

CloudSim is an open-source software developed by the University of Melbourne's CLOUDS Laboratory. It allows users to test applications in controlled environments, identify system bottlenecks, and develop adaptive provisioning techniques. CloudSim provides a generalized and extensible simulation framework for app performance modeling, allowing developers to focus on specific system design issues without worrying about cloud-based infrastructures and services. Cloud computing is ideal for applications with heterogeneous, dynamic, and competing QoS requirements, as it creates complex provisioning, deployment, and configuration requirements.

4.1.5 Architecture

The CloudSim model of cloud computing architecture comprises three layers: system layer, core middleware, and user-level middleware shown in figure 20, representing the top layers of Iaas, Paas, and SaaS respectively.

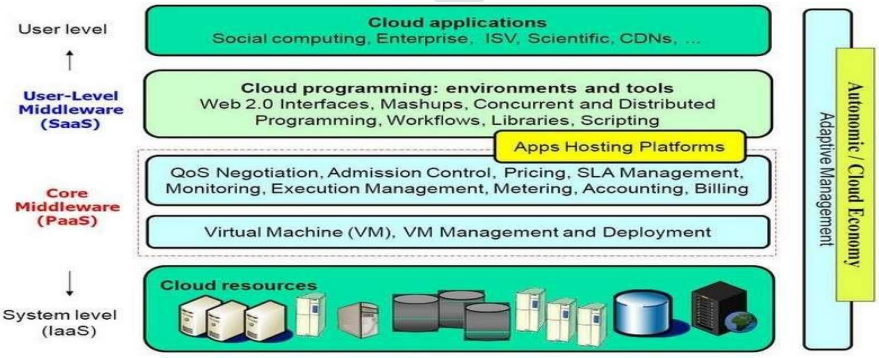


Fig.6 Cloud-simulation-framework

CloudSim Toolkit

CloudSim is an extensible simulation toolkit or framework that enables modelling, simulation and experimentation of Cloud computing systems and application providing environments. The CloudSim toolkit supports both system and behaviour modelling of Cloud systemcomponents.

Basic components of CloudSim

- **Datacenter:** The system-level cloud infrastructure model involves hosts managing virtual machines for low-level processing, requiring at least one datacenter to begin simulation.
- **Host:** This component assigns processing capabilities, memory, and scheduling policy to multiple virtual machines managed by the host, ensuring efficient allocation of processing cores.
- **Virtual Machines:** This component allocates virtual machines to hosts, scheduling processing cores based on application, with a default policy of "first-come, first-serve".
- **Datacenter Broker:** A broker negotiates between users and service providers based on user requirements, identifying suitable providers based on Cloud Information Service data. Users must extend this class for experimentation requirements.
- **Cloudlet:** This component represents the application service whose complexity is modelled in CloudSim in terms of the computational requirements.
- **CloudCoordinator:** This component manages the communication between other Cloud Coordinator services and brokers, and also monitor the internal state of a datacenter which will be done periodically in terms of the simulation time.

Introduction to NetBeans IDE

Net Beans IDE is a free, open-source development environment that supports desktop, mobile, and web application development in Java, HTML5, PHP, and C++. It runs on Windows, Linux, Mac OS X, and UNIX-based systems, and supports JDK 7 technologies, Java EE 7, and JavaFX 2, including GlassFish Server.

Tools for Java 8 Technologies

Anyone interested in getting started with lambdas, method references, streams, and profiles in Java 8 can do so immediately by downloading NetBeans IDE 8. Java hints and code analyzers help you upgrade anonymous inner classes to lambdas, right acrossall your code bases, all in one go.

Tools for Java EE Developers:

The code generators for which NetBeans IDE is well known have been beefed up significantly. Where before you could create bits and pieces of code for various popular Java EE component libraries, you can now generate complete PrimeFaces applications, from scratch, including CRUD functionality and database connections.

Tools for Maven:

A key strength of NetBeans IDE, and a reason why many developers have started using it over the past years, is it's out of the box support for Maven. No need to install a Maven plugin as it's a standard part of the IDE. No need to deal with IDE-specific files, since the POM provides the project structure.

Tools for JavaScript

Thanks to powerful new JavaScript libraries and frameworks over the years, JavaScript as a whole has become a lot more attractive for many developers. For some releases already, NetBeans IDE has been available as a pure frontend environment.

Best support for Latest Java Technology:

NetBeans IDE is the official IDE for Java 8. With its editors, code analyzers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs [48].

Write Bug Free Code:

The cost of buggy code increases the longer it remains unfixed. NetBeans provides static analysis tools, especially integration with the widely used FindBugs tool, for identifying and fixing common problems in Java code. In addition, the NetBeans Debugger lets you place breakpoints in your source code, add field watches, step through your code, run into methods, take snapshots and monitor execution as it occurs[48].

Working with CloudSim in NetBeans:

- Download Net Beans Latest version. Install it in your device
- Hit Google and download CloudSim 3.0.
- Extract the CloudSim 3.0 folder using winrar or any other software.
- Open up your net beans.
- Go to "File" in the menu bar. Click new project.
- Click Java and then "java application".
- Click Next and Name your project as "cloudsimproject1".
- Click next and then you will see "cloudsimproject1" will be created. Now expand cloudsimproject1 and you will see two folders. Source packages and Libraries.
- As you can see above initially you will have to add jar files into the libraries folder. Now click add JAR/FOLDER.
- Select all those jar files and click Open. These all would get added to the Libraries folder.
- Now go to the CloudSim 3.0.3 extracted folder into your directory. Go to the path as given below: cloudsim3.0.3 -> examples->org. Copy the "org" sub folder which is inside examples as its parent folder. Once you copy it, now go to netbeans and clickpaste into the source packages.

- You will see the CloudSim examples get installed.
- Click to open any example and run it.
- You will get the results.

PSO Task Scheduling Algorithm

1. Set particle dimension as equal to the size of ready tasks T.
2. Initialize particles position randomly from PC = 1…..j and velocity vi randomly.
3. For each particle, calculate its fitness value.
4. If the fitness value is better than the previous best pbest, set the current fitness value asthe new pbest.
5. Perform Steps 3 and 4 for all particles and select the best particle as gbest.
6. For all particles, calculate velocity and update their positions.
7. If the stopping criteria or maximum iteration is not satisfied, repeat from Step 3 & 4.

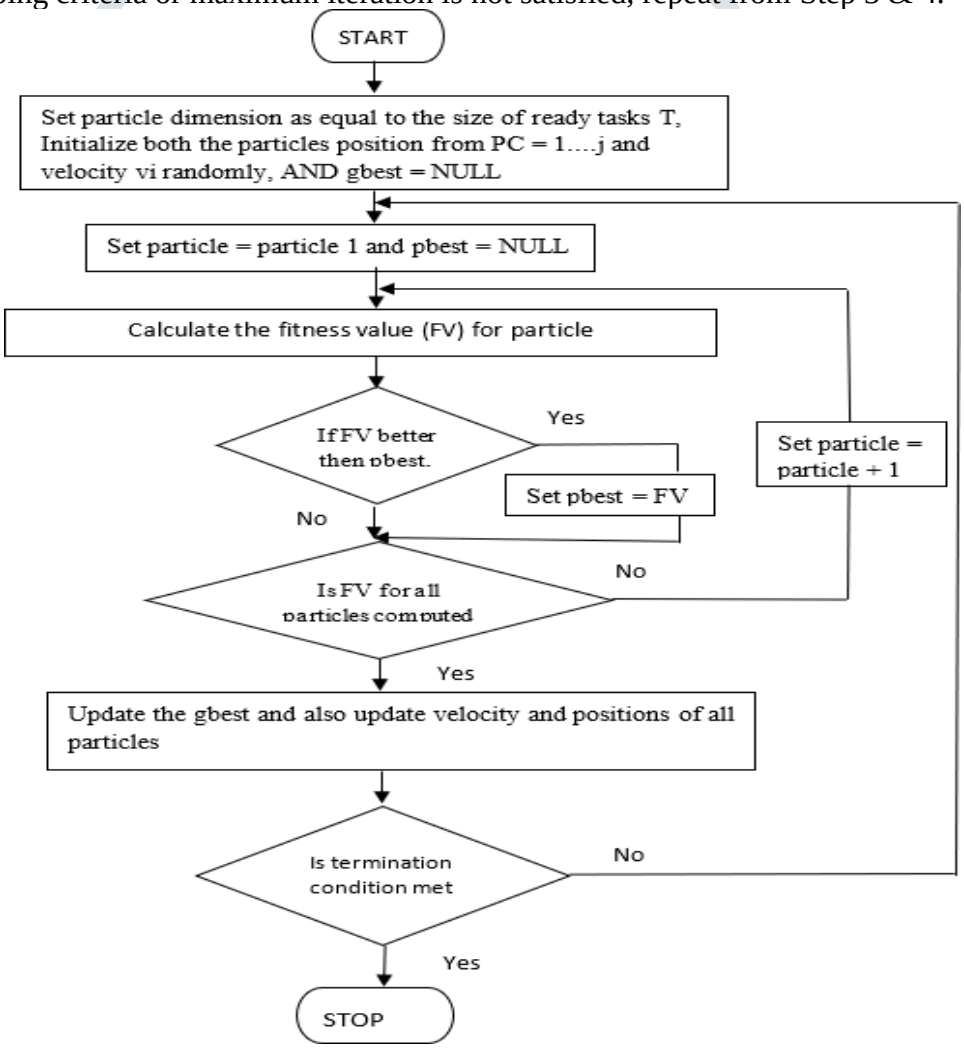


Fig.7 Flow Diagram of Particle Swarm Optimization Scheduling algorithm

Proposed Hybrid Meta-heuristic task scheduling algorithm Algorithm

Input:

CloudletList: List of all clouldlets received.VmList: List of all VM's

Main_Function Hybrid_TS

// initially schedule a small no. of incoming tasks using PSO [number of tasks can befixed or decided on the basis of percentage of total tasks]. In the proposed work, cloudlets

= 2 times the size of VM list are initially passed to PSO scheduling algorithm

PSOCloudletList = CloudletList [1: Sizeof (VmList)* 2]

1. Call PSO_TS (PSOCloudletList, VmList)
2. ACOCloudletList = CloudletList – PSOCloudletList
   // Scheduling based ACO algorithm
4. temp_List_of_Cloudlet = null, temp_ACO_List_of_Cloudlet = ACOCloudletList and n=size of VMs list
5. while temp_ACO_List_of_Cloudlet not empty

   if (size of temp_ACO_List_of_Cloudlet greater than n)

   Transfer the first arrived n Cloudlets from temp_List_of_Cloudlet and put them ontemp_List_of_Cloudlet

   else

   Transfer all Cloudlets from temp_List_of_Cloudlet and put them on temp_List_of_Cloudletend If
6. Call ACO_TS (temp_ACO_List_of_Cloudlet, VmList)
7. temp_ACO_List_of_Cloudlet =temp_ACO_List_of_Cloudlet-temp_List_of_Cloudlet
8. Compute the degree of imbalance factor between the VM's.
9. If the degree of imbalance factor is greater than threshold value then

a. For each VM in the VM List, compute the length of all cloudlets assigned toeach VM in the VM List.

b. Sort of VM's on the basis of length and create a list UL_VmList of allunderloaded VM's and List of Cloudlets OL_VM_CloudletList of Cloudlets in the execution list of overloaded VM's

c. Call PSO_TS (OL_VM_CloudletList, UL_VmList) and transfer Cloudletsfrom overloaded loaded VM to Least loaded VM in the sorted list.

   SO_TS (_CloudletList, UL_VmList) and transfer Cloudletsfrom overloaded loaded VM to Least loaded VM in the sorted list

Function PSO_TS (CloudletList, VmList)

1. Set particle dimension as equal to the size of ready tasks T.

2. Initialize particles position randomly from PC = 1… j and velocity vi randomly.
3. For each particle, calculate its fitness value.
4. If the fitness value is better than the previous best pbest, set the current fitness valueas the new pbest.
5. Perform Steps 3 and 4 for all particles and select the best particle as gbest.
6. For all particles, calculate velocity and update their positions.
7. If the stopping criteria or maximum iteration is not satisfied, repeat from Step 3 & 4.

Function ACO_TS (CloudletList, VmList)

1. Initialize pheromone value for each path between tasks and resources, set optimal solution to NULL and place m ants on random resources.
2. Repeat for each ant

a. Put the starting resource of first task in tabu list and all other tasks in allowed list.

b. Based on the probability function or transition rule, select the resource for allremaining tasks in the allowed list.

3. Compute fitness of all ants which in this case is Makespan time.
4. Replace the optimal solution with the ant's solution having best fitness value if itsvalue of better than previous optimal solution.
5. Update both local and global pheromone.
6. Stop when the termination condition is met and print the optimal solution.

Degree of imbalance (DI) can be computed using equation 1 & 2. It measures the imbalancebetween VMs [63].

$$T_i = \frac{TL\_Tasks}{pe\_num_j * Pe\_mips_j} \quad (1)$$

$$DI = \frac{T_{max} - T_{min}}{T_{avg}} \quad (2)$$

TL_Tasks refers to the length of all tasks assigned to VMi. Tmax, Tmin and Tavg refer to the average of Ti among all VMs

Proposed Improved Resource Aware Hybrid Meta-heuristic algorithm for Load Balancing

The pheromone value for all paths between each task and resource is set to a constant value at the start of each iteration of the ACO algorithm. It somehow fails to take into account the current load of each resource into consideration for the tasks previously assigned to these resources. In the proposed method, the pheromone value for each path is set based on the degree of imbalance between all resources based on already assigned tasks to each resource. Initially the tasks are scheduled using the Particle Swarm

Optimization. Even though the scheduling is done with the help of PSO, the pheromone values are updated as per the scheduleprepared using PSO. Once the set number of tasks are scheduled using PSO, remaining tasksare scheduled using PSO [72]. With this the ACO is expected to produce better results as it based on resource awareness. Pheromone value for each path at the start of iterations is set asfollows:

For all VM's, i= 0 to sizeof(VmList)-1, in the sorted VM list,

Compute the Degree of imbalance $DI = \frac{T_i - T_0}{T_{avg}}$ [63] (3)

Set pheromone value for paths between each task and resource $VM_i = c - DI$, where $c$is some constant value.

All VM's are sorted on the basis of length of all cloudlets assigned to them. T_0 has minimumlength and pheromone value for let's say T_4 depends on T_0, T_1, T_2, T_3 & T_4.

Also, the probability function in ACO is computed as follows,

$$P^k(t) = \frac{[c_{ij}(t)]^{\alpha} * [n_{ij}(t)]^{\beta}}{\sum s \in allowed_k [c_{is}(t)]^{\alpha} * [n_{is}(t)]^{\beta}}$$ if jε allowed k [63]

0 Otherwise

Where $\lceil n_{ij} \rceil^{\beta} = 1/d_{ij}$

d_{ij} is the total time needs by task I to finish, which is sum of expected execution time of a task$_i$ on resource$_j$ & the transfer time. It is expressed as follows:

$$d_{ij} = \frac{TL\_Task_i}{Pe\_num_j * Pe\_mips_j} + \frac{inputFileSize}{VM\_bw_j}$$ [63]

In the equation above, **Transfer cost** = $\frac{inputFileSize}{VM\_bw_j}$

where Input File Size is the total length of the file & VM_bw_j stands for band width ofresource j

To reduce the transfer cost along with the overall execution time, it is important to sperate the expected execution and the transfer cost as follows:

$$d1_{ij} = \frac{TL\_Task_i}{Pe\_num_j * Pe\_mips_j}$$

$$d2_{ij} = \frac{inputFileSize}{VM\_bw_j}$$

Now $\left[n_{ij}\right]^{\beta}$ can be represented as = $[(1/ d1ij)^{\beta 1} + (1/ d1ij)^{\beta 2}]$and hence, $\beta = \beta 1 + \beta 2$

Algorithm

Input:

CloudletList: List of all clouldlets received.VmList: List of all VM's

Main_Function Hybrid_TS

// initially schedule a small no. of incoming tasks using PSO [number of tasks can befixed or decided on the basis of percentage of total tasks]. In the proposed work, cloudlets

= 2 times the size of VM list are initially passed to PSO scheduling algorithm

1. PSOCloudletList = CloudletList [1: Sizeof (VmList)* 2]
2. Call PSO_TS (PSOCloudletList, VmList)
3. ACOCloudletList = CloudletList – PSOCloudletList

// Scheduling based ACO algorithm

4. temp_List_of_Cloudlet = null, temp_ACO_List_of_Cloudlet = ACOCloudletList and n=size of VMs list
5. while temp_ACO_List_of_Cloudlet not empty
   if (size of temp_ACO_List_of_Cloudlet greater than n)
   Transfer the first arrived n Cloudlets from temp_List_of_Cloudlet and put them on temp_List_of_Cloudlet
   Else
   Transfer all Cloudlets from temp_List_of_Cloudlet and put them ontemp_List_of_Cloudlet
   End if
6. call ACO_TS (temp_ACO_List_of_Cloudlet, VMList)
7. temp_ACO_List_of_cloudlet = temp_ACO_List_of_Cloudlet- temp_List_of_Cloudlet
8. Compute the resource utilization (VM utilization), and if it is less than 80%

// imbalance factor is used to determine is migration is needed

   a. **Compute the degree of imbalance factor between the VM's.**
      i. For each VM in the VM List, compute the length of all cloudlets assigned to each VM in the VM List
      // sorting is done to find the underloaded and overloaded VMs.

ii. Sort the VM's on the basis of length and create a list UL_VmList of allunderloaded VM's and List of Cloudlets OL_VM_CloudletList ofCloudlets in the execution list of overloaded VM's

iii. Call PSO_TS (OL_VM_CloudletList, UL_VmList) and transferCloudlets from overloaded loaded VM to Least loaded VM in the sorted list.

End do

Function PSO_TS (CloudletList, VmList)

// initialize the particle positions

1. Set particle dimension as equal to the size of ready tasks T.
2. Initialize particles position randomly from PC = 1….j and velocity vi randomly.
3. For each particle, calculate its fitness value.
4. If the fitness value is better than the previous best pbest, set the current fitness valueas the new pbest.
5. Perform Steps 3 and 4 for all particles and select the best particle as gbest.

//update the velocity and positions

6. For all particles, calculate velocity and update their positions.

//terminate if the stopping criteria is met

If the stopping criteria or maximum iteration is not satisfied, repeat from Step 3 & 4.

Function ACO_TS (CloudletList, VmList)

1. Sort the VM's on the basis of length. i.e. length of cloudlets assigned to each VM**.**
2. Initialize pheromone value for each path between tasks and resources as follows:
3. For all VM's, i= 0 to sizeof(VmList)-1, in the sorted VM list,

   Compute the Degree of Imbalance $$DI = \frac{T_i - T_0}{T_{avg}}$$

   Set pheromone value for paths between each task and resource $VM_i = c - DI$, where c is some constant value.
4. Set optimal solution to NULL and place m ants on random resources.
5. Repeat for each ant Put the starting resource of first task in tabu list and all other tasks in allowed list. Based on the probability function or transition rule, select the resource for all remaining tasks in the allowed list.
6. Compute fitness of all ants which in this case is Makespan time.
7. Replace the optimal solution with the ant's solution having best fitness value if itsvalue of better than

previous optimal solution.

8. Update both local and global pheromone.
9. Stop when the termination condition is met and print the optimal solution.

Degree of imbalance (DI) can be computed using equation 3 & 4. It measures the imbalance between VMs [63].

$$T_i = \frac{TL\_Tasks}{Pe\_num_j * Pe\_misps_j} \quad (3)$$

$$DI = \frac{T_{max} - T_{min}}{T_{avg}} \quad (4)$$

TL_Tasks refers to the length of all tasks assigned to VMi. Tmax, Tmin and Tavg refer to theaverage of Ti among all VMs [63]

Resource/VM utilization can be measured using equation below [43]:

$$\text{Average Resource Utilization} = \frac{\sum_{i=1}^{n} TimeTakenBy\,\text{Re}source"i"toFinishAllJobs}{makspanXn}$$

Where n is total numbers of resources or VMs.

REFERENCES

[1] Buyya, R., Vecchiol, C., Selvi, S. T. (2013). *Mastering Cloud Computing*. Chennai, Chennai: Mc Graw Hill Education (India) Private Limited.

[2] Griffith, E. (2016, May 3). *What is Cloud Computing*. Retrieved from https://in.pcmag.com/networking-communications-software/38970/what-is-cloud- computing.

[3] Lau, W. (2012, May 16). *An Introduction to Cloud Computing Characteristics and Service/Deployment Models*. Retrieved from https://dzone.com/articles/introduction- cloud-computing.

[4] Berry, M. (2012, April 4). *Major Cloud Computing Vendors*. Retrieved from http://www.itmanagerdaily.com/cloud-computing-vendors.

[5] Moghaddam, F. F., Ahmadi, M., Sarvari, S., Eslami, M., & Golkar, A. (2015). *CloudComputing Challenges and Opportunities: A Survey*, presented at 1st International conference on Telematics and Future Generation Networks (TAFGEN), Kuala Lumpur, May 26-28, Kuala Lampur: IEEE.

[6] Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud Computing: State-of-the-Art andResearch Challenges. *Journal of Internet Services and Application*, 1(1), 7–18.

[7] Reddy, V., Rao, B., Reddy, L., & Kiran, L. (2011). Research Issues in Cloud Computing. *Global Journal of Computer Science and Technology*, 11(11), 1-8.

[8] Senkul, P., & Toroslu, I. H. (2005). An architecture for Workflow Scheduling under Resource Allocation

Constraints, *Journal of Information Systems*, 30(5), 399-422.

[9] Chenley. (2011, February 9). *Hypervisors*. Retrieved from https://blogs.technet.microsoft.com/chenley/2011/02/09/hypervisors.

[10] Barr, J. (2006, August 24). *Amazon EC2*, Retrieved from https://aws.amazon.com/blogs/aws/amazon_ec2_beta.

[11] Cloud Computing Architecture. (2018, September 12). Retrieved from https://www.simplilearn.com/cloud-computing-architecture-article.

[12] Srinivasan, S., & Suresh, J. (2014). *Cloud Computing: A Practical Approach for Learning & Implementation*. Chennai, Chennai: Pearson.

[13] Cloud Computing Architecture. (2016, May 23). Retrieved from https://www.w3schools.in/cloud-computing/cloud-computing-architecture.

[14] Bhupender. (2016, June 17). *An Overview of the types of Visualization in Cloud Computing.* Retrieved from https://www.znetlive.com/blog/virtualization-in-cloud-computing.

[15] Pearce, M., Zeadally, S., & Hunt, R. (2013). Virtualization: Issues, security threats, and solutions. *ACM Computing Surveys (CSUR), New York, USA*, 45(2), 17:1-17:39.

[16] Deol, G. (2015, January 11), *Step by step installation of cloud sim into net beans* Retrieved from https://researchcloudcomputing.wordpress.com/2015/01/11/step-by- step-ui-installation-of-cloud-sim-into-net-beans.

[17] Talbi, E. G. (2017). *Metaheuristics: From Design to Implementation*. Hoboken, New Jersey: John Wiley & Sons, Inc.

[18] Yang, X. S. (2010). *Engineering optimization: An Introduction with Metaheuristic Application*, New York, NY: Wiley.

[19] Lin, M. H., Tsai, J. F., & Yu, C. S. (2012). A Review of Deterministic Optimization Methods in Engineering and Management, Journal of *Mathematical Problems in Engineering*, 2012, 1-15.

[20] Siddique, N., & Adeli, H. (2015). Nature Inspired Computing: An Overview and some future Directions, *journal of Congnitive Computation,* 7(6), 706-714

[21] Fister, I. Jr, Yang, X. S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization, Elektrotehniski Vestnik, 80(3), 1-7.

[22] Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del-Ser, J., Bilbao, M. N., Salcedo- Sanz, S., & Geem, Z. W. (2013), A Survey on Applications of the Harmony Search Algorithm. *Engineering Applications of Artificial Intelligence*, 26(8), 1818-1831.

[23] Szeto, W. Y., Wang, Y., & Wong, S. C. (2014). The Chemical Reaction Optimization Approach to Solving the Environmentally Sustainable Network Design Problem. *Computer-Aided Civil Infrastructure Engineering*, 29(2), 140-158.

[24] Badawy, R., Yassine, A., Hebler, A., Hirsch, B., & Albayrak, S. (2013). A Novel Multi-Agent System Utilizing Quantum-Inspired Evolution for Demand Side /Management in the Future Smart Grid, *Integrated Computer-Aided Engineering*, 20(2), 127-141.

[25] Campomanes, A., lvareza, B. R., Cordon, O., & Damasa, S. (2013). Evolutionary multi-objective optimization for mesh simplification of 3D open models, *Integrated Computer-Aided Engineering*, 20(4), 375-390.