AN APPROACH FOR LOAD BALANCING IN CLOUD USING HYBRID OPTIMIZATION ALGORITHM

By Pooja Dhanrajani

-

AN APPROACH FOR LOAD BALANCING IN CLOUD USING HYBRID OPTIMIZATION ALGORITHM

A
THESIS
SUBMITTED TO

GALGOTIAS UNIVERSITY
GREATER NOIDA

IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE AND ENGINEERING

By

Ms. Pooja Dhanrajani
ADMISSION NO- 14SCSE301005

Under the Guidance of Prof. (Dr.) Anurag Dixit
School of Computer Science and Engineering

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
JULY-2020

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis, entitled **"An approach for Load Balancing in Cloud using Hybrid Optimization Algorithm"** in fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science Engineering and submitted in School of Computing Science and Engineering Galgotias University, Greater Noida is an authentic record of my own work carried out during a period from JANUARY, 2015 to July, 2020 under the supervision of Dr. ANURAG DIXIT .

The matter embodied in this thesis has not been submitted by me for the award of any other degree of this or any other University/Institute.

(POOJA DHANRAJANI)

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

(Dr. ANURAG DIXIT)
Supervisor
Professor, School Of Computing Science And Engineering
Galgotias University, Greater Noida

The Ph.D. Viva-Voice examination of Ms. Pooja Dhanrajani Research Scholar has been held on __________.

Sign. Of Supervisor

Sign. Of External Examiner

Dedicated

To

My Family

ACKNOWLEDGEMENT

The Research work presented in this thesis would not been possible without my close association with many peoples. I would like to take this opportunity to extend my sincere thanks and appreciation to all those who made this PhD thesis possible.

First and foremost, praises and thanks to the God, the Almighty, for his showers of blessings throughout my research work to complete the research successfully.

Very special thanks to the **Shri Sunil Galgotia Chancellor and Mr. Dhruv CEO, Galgotias University** for giving me the opportunity to carry out my doctoral research and for their support.

I would like to express special thanks to my supervisor **Dr. Anurag Dixit** for his tremendous support and been tremendous mentor for me. I would like to thank you for introducing me to exciting field and always encouraging, motivating, enthusiasm and allowing me to do what I think and continuous support throughout my PhD, your advice and support have been invaluable.

I express my deep gratitude to my parents for this life. I thanks to God for this lovely family in which my brother and sister are my best friends. I owe my deepest gratitude towards my husband for his eternal support and understanding of my goals and aspirations. His love and support has always been my strength. At the end, my daughter, my little angel, I feel sorry to steal your time to complete the research work.

Thanks God for being with me always and providing all the strength, happiness and good health to carry out this study.

Thank you so much………

POOJA DHANRAJANI

TABLE OF CONTENTS

Page No.

CHAPTER 1: INTRODUCTION

1.1 Problem Overview and motivation

Cloud computing is the new advanced field with the architecture of service-oriented. With the rise in the application of cloud computing which lead to rise in the number of task and workload. This workload makes unbalanced load due to uneven distribution of task on different nodes with different capabilities of each node. This makes some nodes overloaded and some underloaded in cloud and lead to unbalanced load. To make efficient utilization of resources of cloud it becomes necessary to balance this load and improve the fulfilment of client. The load adjusting is the redistribution of task from over-burdened machine to under burdened machine.

As per recent commercial standard, cloud computing got attention from both academic and commercial group. Due to the advancement in cloud computing, most commercial groups and individuals can outsource their large data in cloud. They do not need to maintain their own local data centers. The public cloud also provide many different types of computing services to its user.

NIST declared Cloud computing as a system for providing the service as per user convenience. Cloud computing provide resources on-request of user, The user can use network for shared pool of computing resources. The computer resources can be networks, servers, storage, applications, and services that can be provided speedily with very less endeavour for managing it. The system requires less interaction with cloud provider.

Distributed computing can be viewed as another processing worldview to the extent that it permits the usage of a registering framework at least one degrees of deliberation, as an on-request administration made accessible over the Internet or other PC organize. In view of the suggestions for more noteworthy adaptability and accessibility at minimum price, distributed computing is a technology which is getting a decent arrangement of consideration.

The main task of loa-d balancing (LB) is to unmistakably comprehend the purchaser necessities, information and data can be sent and gotten without taking additional time. LB in CC is one of the serious issues without load adjusting clients could delays, and give tedious framework reactions, the heap can be organize load, memory and CPU loads and so on. LB is the method of increasing the performance of Distributed System (DS). It is the procedure of transferring load between different virtual machine of DS to improve job's reaction time and resource usage where as it also avoiding a condition when few virtual machines are over-loaded where as other virtual machines are idle or doing under loaded work in a given instant of time in the system. LB calculates various terms like minimizing communication delays, minimizing execution time, maximizing throughput and maximizing resource utilization [10].

An average disseminated framework, for example, Expert Cloud incorporates the quantity of appropriated Human Resourcess(HRs). This HRs is associated with one another to accomplish the superior and execute the assignment that one HR can't do it without anyone else. To decrease the errand execution time by the HR, the outstanding task at hand ought to be appropriated dependent on the HRs quality. This makes the heap adjusting fundamental. The primary motivation behind the heap adjusting is that it encourages systems and assets by giving a greatest throughput least reaction time. Separating traffic among clients, and subsequently information can be sent and gotten immediately [13][4]. Load balancing redistributes remaining tasks at hand among cloud with painstakingly planned methodologies that assurance to boost asset usage and protect from over load[14][6]. Load balancing is a technique which offers systems to grow throughput, utilization of resources and execution of structure. As a bit of its organizations, it gives straightforward and versatile system to keep data or reports and make them open for enormous size of customers[9]. On the off chance that an excessive number of assignments are packed on specific node, undertakings could be changed from overwhelming troubled machine to light troubled machine to lessen the holding up time of tasks at machine's wait queue, which is called load adjusting. It is usually expressed that task designation and burden adjusting are vital to the cloud computing [16].

To utilize resources most proficiently in cloud system, there are a few load balancing algorithm. Load balancing algorithm can be ordered into two kinds: static and dynamic, contingent upon whether load balancing choice depends on current load state or not. Static calculations can't adjust to run-time modification of framework when it starts to execute the task. Conversely, run-time calculations settle on adjusting choices as indicated by status of load during run-time and change appropriately to re-allocate outstanding tasks at hand as fundamental. This prompts a wide scope of investigates on powerful calculations for better execution. Diffusion strategies and meta-heuristic based techniques are two principle sorts of dynamic load balancing algorithm. Diffusion strategies [18][6] alludes to the aggregate methodology entrances of certain specialists on others. The communication between neighbouring node inside cloud can be preoccupied as a diffusion procedure [19][6]. In any case, the diffusion strategy can just accomplish locally ideal outcome. Metaheuristic-based methodologies were raised by offering close ideal arrangements in satisfactory time utilizing particle swarm enhancement (PSO), genetic algorithm (GA), Ant Colony optimization (ACO) etc. A meta-heuristic called Gray Wolf Optimizer (GWO) is created in [20] roused by grey wolves (Canis lupus). The GWO algorithm emulates the initiative chain of importance and hunting instrument of grey wolves. Here, four sorts of grey wolves, for example, alpha, beta, delta, and omega are maintained for recreating the administration progressive system. In the interim, another sort of multitude based metaheuristic search strategy, called Elephant Herding Optimization (EHO), is introduced in [21] for providing the solution for the improvization of task. The EHO strategy is motivated because of the crowding conduct of elephant gathering. In Elephant Herding Optimizer, in each group the elephants are refreshed with its present location and matron by group refreshing administrator. In any case, execution of the joining depends exceptionally on the coordinating level between meta heuristic component and framework.

The Cloud Computing is the new advanced field with the architecture of service-oriented. With the rise in demand of the application of cloud computing, that lead to rise in the number of task and workload. This workload makes unbalanced load due to uneven distribution of task on different nodes with different capabilities of each node. This makes some nodes overloaded and some underloaded in cloud and lead to unbalanced load. To make efficient utilization of resources of cloud it becomes necessary to balance this load to satisfy the user. The load adjusting algorithm is vital in the redistribution of task from over burdened virtual machine to under burdened virtual machine.

As there is improvisation introduced in the service-oriented structures which makes distributed computing pulls in the analysts. The enormous amount of task with huge quantity of workload is created by large scale applications formulated in distributed computing[17][6]. The cloud infrastructure, as a superior computing paradigm, makes the utilization of computing infrastructures at every tiers of abstractions. This is provided on demand basis for which request received online.

The Cloud Computing's positive aspects greater interest due to its effortless availability and higher flexibility at low fee [11]. But, the nodes in distributed computing will in general be underutilized while others appear to be over-load because of uneven limits of physical machine, diverse scale of undertakings and unequal division of load [15] [6].Thus, it gets importance for dividing the load between physical machines to get the complete advantages of distributed framework and for improvisation of client's fulfilment [16][6]. As an across the board business worldview, the distributed computing draws in the consideration of clients from both mechanical networks and scholastics. Thus, the advancement of complex distributed computing applications permitted a few endeavors and people for redistributing the critical information to the cloud as opposed to developing and dealing with in-house storage. The clients of cloud who are profited by various kinds of services as there is progression in various sorts of services in the open cloud [12]. The distributed computing is defined as a model by National Institute of Science and Technology (NIST). This empowers advantageous and demand-on-request arrange availability for various resources of processing machines, for example, servers, applications, systems, service, and capacity, that can be immediately provided and delivered with minimum endeavours and collaborations [6].

Because of various services, the distributed computing gives straightforward and adaptable mechanism to keeping information and documents to make them open for large scope clients [9]. On the off chance that a lot of tasks happen on explicit machine, the task is moved to under loaded machine from over-load machines to limit task waiting time of particular machine which is additionally named as Load Balancing (LB). Additionally, in distributing computing, the task's designation and load adjustment are significant [16].The motivation behind balancing the load is to comprehend the prerequisites of users by sending and accepting the information and data without taking a lot of duration. Load balancing is done for distributing again the outstanding tasks at hand into the cloud by planning various systems to augment the usage of resources by maintaining a strategic distance from the overhead [14] [16].The balancing of load has considerable as one major area for significant issues for researchers. The client of cloud can prompt deferral and encourages tedious responses in the absence of balancing the load. In this, the load is of different types such as load of memory, load of processor and load in web. The balancing of load assumes a fundamental job to build the performance of Distributed System [24] [25]. This Distributed System can assist with moving the load within various nodes of Distributed System to improve the task's response time. This also improves the resource's usage by disregarding the situation. In this less machines are over-load whereas many machine are waiting or doing some under loaded work at a particular instance. The balancing of load is utilized in processing various computing that is correspondence delay, execution time, asset use, and throughput [10]. An unique distributed system, similar to Expert Cloud includes diverse dispersed Human Resources (HR's). This current HR's is liable for associating different Human Resources in achieving target of maximum performance and runs the task in a viable way. In the focus of lessening the time for run, the remaining task at hand is probably going to be circulated relying upon the Human Resource's advantage so that it can create the load balancing necessary. The objective for balancing load is that it should give systems and their assets for providing expanded output in minimum time. The final information acquired by separating the traffic between multiple clients is sent and gotten effectively immediately [13] [4]. Likewise, load balancing gives a few techniques to maximize the throughput, resource usage, and system performance.

A few load balancing algorithm are formulated for utilizing resources in cloud framework in an effective way. In this manner, the load balancing algorithm are partitioned into two significant sorts, which include static and dynamic dependent on the choice of load balancing relying upon whether the load is current or not. In this, the static calculations can't alter run time amendments of the framework as when it starts working or running the task [1].The dynamic calculations can offset choices as for the status of load at run time which can modify as needs be for distributing again the necessary workload, that brings in improvement in execution for dynamic algorithm. There are 2 primary components secured by the load balancing algorithm which are based on run time are diffusion techniques and meta-heuristic based techniques. The joined procedure infiltrations of certain operators on others are alluded as a diffusion process [18][6]. The connection between nearest machines inside the cloud may be preoccupied as the diffusion technique [19][6]. Yet, the diffusion techniques are dependable only for accomplishing optimal outcome locally. Metaheuristic based methodologies [28, 29, 30], likeGenetic Algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO, etc, are utilized to provide the optimized solution at particular instance. In [20], a meta-heuristic calculation, named Gray Wolf Optimizer (GWO) is contrived. The Grey Wolf Optimizer is motivated with the conduct of grey wolves. In this manner, the Grey Wolf Optimizer calculation mimics the authority chain of importance and adjusts a hunting technique for grey wolves. In this, there are 4 categories of grey wolves. The categories are alpha, beta, delta, and omega, are adjusted to mimic the initiative chain of command. Besides, an advanced swarm-based meta-heuristic based search technique, named Elephant Herding Optimization (EHO) [21] is concocted to tackle the optimization of tasks. In the Elephant Herding Optimizer which got inspiration from the crowding conduct of elephant. In Elephant Herding Optimizer, the position of elephants in each gathering are refreshed utilizing elephant's present state and matron by family refreshing operator. Be that as it may, the presentation of the consolidation exceptionally relies upon coordinating to the extent that match both system model and meta-heuristic strategies.

1.2 Cloud Computing

Cloud computing is the model based on web is a service provider which provide on-demand services to clients to access the shared group of resources. The resources can be hardware or software. The distributed computing gets consideration from both scholastic and modern networks due to its paradigm that "Everything as a Service". Due to the advancement of cloud computing, many corporatist and scientists/researchers are using it for the storage of large data instead of managing or developing their own local data centres. In today's world, the services on cloud are provided same as utility services like water and electricity. You need to pay as much as you used these services. The cloud users get different types of computing services offered by open cloud [12]. The users now focus on their main objective without worrying about computing resources requirement. With time there is continues increase for the cloud computing resources and it lead to more efficient utilization of computing resources. To provide the hassele free services to its client it is necessary to improve the efficiency which further lead to increase the throughput. Due to the suggestions for more noteworthy adaptability and accessibility at lower cost, distributed computing is a subject that has been getting a decent arrangement of consideration [11].

In Figure 1 there are basically three layers in the cloud computing architecture. Every layer has its own importance. The basic services that the user get from cloud computing architecture are:

1. Software as a Service(SaaS)–In this service end user don't need to install and run different types of applications on their local machine. They can directly use these applications over a cloud network as a service.example is SalesForce.com.
2. Platform as a Service(PaaS)- In this service user can get platform for development and management of Software. In this the software developer can develop and deploy different types application without worrying about tools, languages and API. All these services are provided by cloud service providers. Example is Google App Engine.
3. Infrastructure as a Service(IaaS)-In this service the users are provided storage space in cloud and resources for computation as per their demand and pay as per their usage. Amazon EC2 is an example for the same.

There are 4 categories in cloud infrastructure i.e Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud. The Public cloud is open for all users i.e general pubic and is available all time and user can pay as per usage eg Amazon and Google Cloud. The private cloud is for the particular organization for their use only. This cloud is not available for general public. The Hybrid Cloud combines the features of both public and private cloud. This is used for the need of corporates. The Community cloud is the hybrid form of private cloud i.e. it is used for a particular group of users. A particular community form this type of cloud.

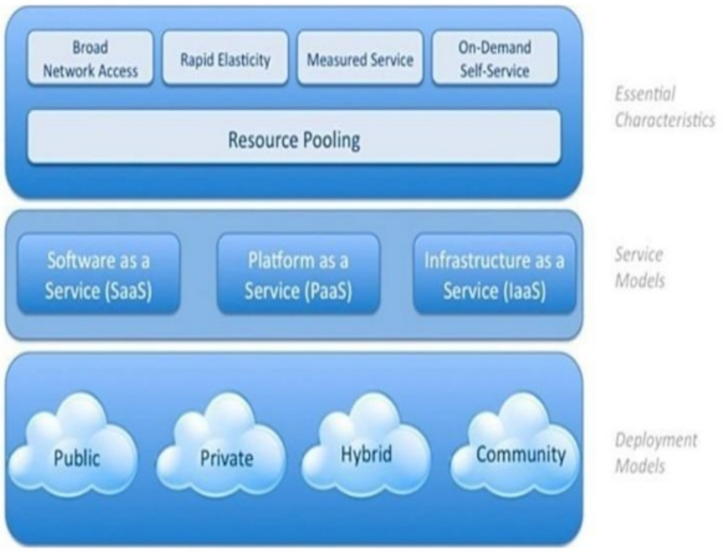


Figure 1.2 : Cloud computing architecture[25]

1.3 Load Balancing in Cloud

The main motive of load balancing is to provide optimize usage of computing resources to reduce the deployment and operational cost for users. The importance of adjusting the load in distributed computing is for providing efficient solution to handle multiple request of multiple client by effective use of computing resources available.

The Balancing of load is one of the important issue in cloud computing. Now a days load balancing becomes an important challenge and concerns. The reasons are:

1. Improvement of performance in cloud environment
2. Maximize the throughput
3. Minimize Response Time
4. Minimize the Cost incurred to user
5. Optimize the resource utilization

6. Save Energy

Without load balancing there can be delays in user response, resources cannot be optimally utilized and resultant is fall in the performance of system. The load of cloud computing is of the type of load on network, memory and CPU etc. The Load adjustment that is the procedure of allocating and reallocating the load among the available nodes in such a manner that no node will remain overloaded in order to improvise the execution in the infrastructure of cloud. Load Balancing calculates various terms like minimizing communication delays, minimizing execution time, maximizing throughput and maximizing resource utilization [10].

1.4 Classification of techniques for load balancing

The algorithm adjusts the load in the distributed system by transferring the load of overloaded machine to under-loaded machine in an efficient manner. The Load Balancing algorithms were categorized on the basis of two factors:

1. Static algorithm
2. Dynamic algorithm

The Static algorithm first needs to get the information of all resources of system. In this algorithm no run-time status of resources are required. It works at the start before execution of task. No changes can be done at the execution time for balancing the load. This type of algorithm is suitable in homogenous process. In this the decision for distributing the load is depend on the prior information of system resources and task. It is used with predicting processing loads in advance.

The Dynamic algorithm works at the run-time. In this the decision for distributing the load is determined by the recent system's state. If there is no overloading of machine at run-time then no redistribution of task will take place. The redistribution decision will take place at run-time only on the basis of current status of system resources. This algorithm gives better performance than static algorithm. It works well for heterogeneous system. It is used with unpredictable processing loads.

1.5 Load balancing Metrics

Throughput (TP):

It is declared as the amount of tasks successfully completed in per unit time by a virtual machine. The system performance value is evaluated by throughput. The System performance is high if throughput value is high. The System performance is low if throughput value is low.

Thrashing (TH):

Due to limited memory or other resources in cloud setup there may cause thrashing. It occurs in the cloud setting when VM is busy in migration instead of processing the task. This is due to incorrect scheduling algorithm.

Therefore, the correct algorithm for load balancing used to manage this aspect.

Reliability (R):

Reliability will perform as per specification of system. If any failure occurs during the runtime of task then that task will be shifted to any other Virtual Machine and thus create a failure free System. The stability of the system depends on reliability of the system.

Accuracy (A):

It evaluates the correctness of result of executed task. It measures the output of executed task and compare it with actual values.

Predictability (PR):

It is the rate for predicting of number of tasks allocated , executed and completed with the available resources in cloud. The value is predicted on the basis of previous pattern of the arrived task, the assigned task and executed task in the cloud system. The balancing of the load improves with improved predicted values of task which further improves makespan.

Makespan (MS):

It means the time needed for completion of all submitted assignments in the cloud system. The time of resource allocation to the assignment is makespan.

Scalability(S):

It means the capability of system using load balancing method to perform in the situation of overloaded task. It means how the system works under unexpected situation. The resources which are available will be rescale from time to time.

Fault Tolerance (FT):

It means ability of the system to work in case of failure. For the provision of uninterrupted service in situation of one or more fault of system elements is the ability of fault-tolerant method. The extra resources and Virtual Machines are needed to handle some types of failure. For this we need to incur extra cost.

Associated Overhead (AO):

The overhead is the additional cost which is incurred for executing the load balancing algorithm. The associated overhead is overhead incurred associated with implementing the algorithm.

Migration time (MT):

It is the duration needed to shift a task or Virtual Machine from one place to another. The shifting of task may take place from one virtual machine to another virtual machine under one or multiple physical machine. In the same way, the shifting of virtual machine will take place from one physical machine to another physical machine within same or different data centres. The more shifting of virtual machines makes increase in shifting time which further result in affecting the makespan and adjusting of load of the system.

Response time (RT):

It is the time required by the framework to react an errand. It is the complete time comprises of transmission time, holding up time, and administration time. Therefore, the framework execution is conversely identified with the reaction time. The base reaction time improves makespan esteem.

1.6 Difficulties of Load Balancing

The difficulties which we observed in different research paper based on the present method for balancing the load are given below:

- Load Balancing is observed as a significant difficulty in distributed infrastructure. The main work of load balancing is to circulate the assignments similarly between various machines to accomplish maximum fulfilment of clients with optimized utilization of resource[8].
- The Response time in distributed system have become a significant issue for Load Balancing algorithm. In load balancing procedure, dynamic method secures more response time, while static resource sets aside reduced effort in producing response[10].
- Cloud computing instigates numerous points of interest for service-based computation. Yet, adjusting the load in multiple class framework is trying because of complexity created in load data fusion [6].
- The significant challenge looked in the distributed computing framework is to maintain a strategic distance from the challenges while, the load will in general be powerfully conveyed by virtual machine [9].
- A few difficulties exist in load balancing methods that fuse security challenges, interoperability challenges, information management challenge, and Quality of Service execution challenge. In Load Balancing calculation, there must be

improvisation for all of these boundaries to redesign the exhibition of complete framework [10].

- In [6], Feature Weight Preferences Fuzzy Clustering- Load Balancing (FWPFC-LB) technique is created for achieving ideal load balancing among the virtual machines. This strategy is capable and having the ability to adjust the load of multiple class, which incorporate storage, network, processors, and resources, yet adjusting the load for multiple class framework is a significant test and starts troubles in load information combination.
- Performance (PR) is very important for any computing environment for overall efficiency of the system. In load balancing algorithm, all parameter are improved then can be improve the overall system performance [10].

CHAPTER 2: LITERATURE REVIEW

2.1 Reviews of existing Load Balancing techniques

Distributed computing is developed as a standard service-oriented system. The huge scope use of distributed computing additionally comes with expanding figure of tasks and flooding quantity of workload [17][6]. Nonetheless, because of various computing limits of nodes and unequal scale of task, few processing nodes inside cloud might be underutilized while others might be over-burden, bringing about uneven load distribution [15][6].Consequently, it is basic to expand the loads among computing machines to exploit distributed computing framework and therefore improve client fulfilment [16][6].

In this segment, the overview of eight existing methods dependent on load balancing is explained alongside their disadvantages.

Shang-Liang Chen et al. [1] built up a dynamic added balance strategy for taking care of the issue brought about by lopsided dissemination of burdens. Here, the load balancing in cloud is mulled over for preparing strength of server and also for machine loading. In this way, the server machine may oversee extraordinary prerequisites of computation. Finally, two calculations in load offsetting are tried with tests for demonstrating the way to deal with be creative. The technique can adjust the load execution when the clients are logged at same time, however the strategy isn't pertinent for various burden strategies.

Qi Liu et al. [2] contrived a versatile plan, 1 named Hadoop-LB utilizing Prediction Model dependent on Kernel work Extreme Learning Machine (PMK-ELM) calculation in accomplishment of the effectiveness of time by providing different types of cloud foundation. The powerful theoretical run time technique which are based on continuous administration for bunch of assets is inferred for upgrading the running season of guide stage and a forecast model is used for quick expectation with least run time of task. A versatile arrangement is concocted for streamlining the exhibition of space-time by joining the expectation representation with a multiple target enhancement calculation, yet the strategy needs additional extra space.

Jia Zhao et al. [3] formulated a propelled approach, named Load Balancing dependent on Bayes and Clustering (LB-BC), for sending mentioned tasks to the server machine which act as a host. LB-BC utilizes restricted requirement for physical machines to achieve an errand organization approach with worldwide quest ability utilizing execution work for figuring the assets. Here, the Bayes hypothesis is incorporated with the grouping procedure for acquiring ideal bunching set with a physical host. Be that as it may, the strategy can't chip away at genuine figuring condition.

Shiva Razzaghzadeh et al. [4] built up a strategy for dispersing the dynamic burden utilizing appropriated lines in cloud foundation. In this strategy, it match the tasks with Human Resource by distributing name to every Human Resource. The load adjusting and planning mehtod are contrived based on Poisson and exponential circulation. This technique allows the assignment portion procedure to execute with high force by adjusting conveyed lines mindful of the administration characteristics. The technique didn't utilize hereditary calculation for deciding the ideal HR by settling the shortcomings.

Ranesh Kumar Naha and Mohamed Othman [5] built up a calculation, named Cost mindful handling and Load mindful facilitating calculation, for adjusting the load in servers and the virtual machine. This calculation limits the general preparing and reaction times as the undertakings are allocated to the accessible physical assets in a successful way. The calculation didn't consider genuine cloud facilitating for the assessment.

Weihua Huang et al. [6] built up a fuzzy bunching strategy utilizing highlight weight inclinations for conquering load adjusting issues in multiclass framework assets and can accomplish ideal answer for load adjusting by load information combination. Here, include weight inclinations are utilized in building up connection within explicit cloud situations and load balancing technique. The technique is not appropriate for advancing the versatile boundary in load information combination and for improvisation of the assembly rate increasing speed in grouping.

Narander Kumar and Diksha Shukla [7] contrived a technique, named fuzzy line punishment strategy, to comprehend the difficulty of load balancing in a distributed framework condition dependent on fuzzy. In this, the fuzzy procedure is utilized in tending to questionable reaction time in fuzzy cloud condition. Here, the fuzzy column punishment strategy is created to point the fair fuzzy load balancing issue and uneven fuzzy load balancing issue in a cloud foundation. The produced outcome is utilized for tackling the load balancing issues dependent on reaction existence complexities for amplifying the exhibition, limiting the versatility, overheads, however outstanding task at hand dispersion is ill-advised.

Santanu Dam et al. [8] used a progressed computational knowledge procedure, named Ant Colony Optimization (ACO), to adjust the load among virtual machine in distributed framework. Here, the Ant Colony Optimizer is utilized in structuring a wise multi-specialist framework by aggregate conduct of ants. Here, the ACO is used for tending to the load balancing issue in distributed framework and to limit the makespan and to limit the virtual machine use, yet adaptation to non-critical failure and dealing with high need work are as yet meant as significant issues.

Mahfooz Alam and Mohammad Shahid [26] introduced a Load Balancing method with Migration cost LBSM to run an autonomous cluster for task on different heterogeneous MINs viz. Here, MetaCube, X-Torus and Folded Crossed Cube having the target of limiting the heap awkwardness on processors. Raza Abbas Haidri et al. [27] built up a Capacity based Deadline Aware Dynamic Load Balancing (CPDALB) calculation for addressing the load adjusting issue. CPDALB centres around the choice of virtual machine in allotment of undertakings to guarantee consumer loyalty regarding complying with time constraint imperatives and rate for executing the app. It utilizes an idea for conveying solicitations to virtual machines dependent on their preparing abilities for limiting irregularity of load with fulfilling cutoff time.

2.2 Analysis of different load balancing algorithm

S.No.	Authors	Methods	Advantages	A. Disadvantages	B. Parameters	C. Experimental /Simulaion Tool used
1	Shang-Liang Chen et al. [1]	Cloud Load Balance (CLB) algorithm	balance the loading performance when users logged in at the same time	**D.** The method failed to consider different load approach	**E.** Online users, Time-cost, Priority Service(PS) value	**F.** testing server uses OS Windows7; the Programming Language is MS Visual Studio 2010 C# with the MS SQL Server 2005 database. The Server is Internet Information Services 2.0 with a RAM of 4,096 MB and an Intel T2390/1.86 GHz CPU
2	Qi Liu et al. [2]	Hadoop-LB with prediction model based on K-ELM	reduce the running time, high accuracy	**G.** waste more storage space	**H.** execution time	**I.** The server is equipped with 288 GB of **J.** memory and a 10 TB hard

		(PMK-ELM)				driver on **K.** Hadoop 2.6.0
3	Jia Zhao et al. [3]	Load Balancing based on Bayes and Clustering (LB-BC)	improved throughput	The method failed to apply LAN	**L.** MakeSpan, Throughput	**M.** CloudSim platform
4	Shiva Razzaghzadehet al. [4]	load balancing strategy	improves the makespan and cost	The method failed to extend load balancing for dependent tasks and failed to consider more factors such as fault tolerance for HR label	**N.** Makespan, execution time	**O.** Cloudsim and validate it in Amazon EC2
5	Ranesh Kumar Naha and Mohamed	Cost aware brokering and Load aware brokering	minimized the cost	The method failed to consider efficient load	**P.** virtual machine cost, Response	**Q.** CloudAnalyst

	Othman [5]	algorithm		balancing algorithm for minimizing the execution time	time and processing time	
6	Weihua Huang et al. [6]	Fuzzy clustering method with Feature Weight Preferences for Load Balancing (FWPFC-LB)	achieve better load balancing performance	The method failed to consider adaptive parameter optimization for data fusion and convergence rate acceleration in clustering	**R.** Through puts, Makespans, Migration times	**S.**
7	Narander Kumar and Diksha Shukla [7]	fuzzy row penalty method	increase performance and scalability, minimize associated overheads, takes less	workload is not distributed properly	**T.** Execution Time, response time and space complexity	**U.** CloudSim

			execution time			
8	Santanu Dam et al. [8]	Ant-Colony-Based Meta-Heuristic Approach	minimize the make span as well as the number of Virtual Machine (VM) is also reduced	The method failed to include fault tolerance and priority of the job	**V.** response time	**W.** CLOUD ANALYST

CHAPTER 3:RESEARCH METHODOLOGY

3.1 Definition of Problem

The headways in our distributed computing has picked up the consideration of a few scientists to give on-request access to network for clients with resource sharing. Distributed framework is significant an exploration heading that can give platforms and software to customers utilizing internet. However, dealing with tremendous number of task in cloud framework is a muddled assignment. Along these lines, it needs balancing for load technique for allotting undertakings to virtual machines in the absence of impacting execution of system. We proposes a load balancing technique, named Hybrid alogorithm. The hybrid algorithm is also named as 1 Elephant Herd Grey Wolf Optimization (EHGWO) which do balance the load in cloud infrastructure considering all factors. The introduced Hybrid algorithm is designed by integrating 1 Elephant Herding Optimization (EHO) in Grey Wolf Optimizer (GWO) for selecting the optimal virtual machines for allocating again on the basis of newly devised fitness procedure. The proposed load balancing technique considers different parameters of virtual machines and physical machines for selecting the tasks to initiate the reallocation for load adjusting. In this, 2 pitch factors, named Task Pitch Factor and Virtual Machine Pitch Factor, are designed for allocating the assignments to balance the loads.

Distributed framework has developed as a standard service-oriented architecture. The huge scope utilization in distributed framework likewise results in expanding task's quantity and flooding measure for outstanding tasks at hand [17][6]. Be that as it may, because of various figuring limits of machines and uneven scale of task, some machines inside the cloud might be underutilized whereas others might be over-burden, bringing about uneven load circulation [15][6].Therefore, it is basic to spread the load across processing nodes to exploit distributed computing framework and thus improve client fulfilment [16][6].As another pervasive business worldview, distributed computing has stood out from both scholarly and commercial organization. Because of the propelled advancement of distributed computing, numerous organizations and people are permitted to redistribute the significant measure of information to cloud as opposed to building and keeping up local data centers. The cloud clients may likewise appreciate different sorts of processing services offered by open cloud.[12].

3.2 Objective

The major purpose in our research is that we want to design and implement a technique for load balancing to improve the efficiency of the distributed framework which is cloud system. Here, load balancing is performed to expel the tasks from over loaded virtual machines and relegating them to under loaded virtual machines without affecting the system performance. Accordingly, Elephant Herding-based Grey Wolf Optimizer (EHGWO) will be newly developed method in adusting the load. Initially, the capacity and loads of the virtual machine will be found based on the executed tasks, then, the balance of the cloud system will be checked. When the load is found unbalanced, the limit with load will be checked to take the selection whether load balancing can be done or not. In the other case, find the tasks to be removed by checking the two constraints like, load of the task and cost of the load balancing. Then, the removed tasks will be added in other VMs by optimally finding the VMs for the task execution. The optimal finding of VMS for executing of the removed task will be found out using the proposed EHGWO, which will be designed by combining Elephant Herding Optimizer[21] and Grey Wolf Optimizer[20]. The implementation of the proposed approach will be in JAVA with Cloudsim tool. The achievement of the proposed technique for load adjusting will be evaluated in distinct cloud set up for makespan, and the results attained will be compared with that of existing works [1] [2] and [3].

3.3 Simulator

3.3.1 CloudSim

The CloudSim is the simulator which is used to give a normal cloud environment and framework for simulation. This makes it possible to model, simulate and experiment on distributed computing architectures and application services. Its allow us to work on particular design of system. We do not need to focus on low level details of cloud computing infrastructure and its services.

Features of CloudSim

1. It is used to model and simulate the issue in large scale Cloud computing data centers.

2. It is used to model and simulate the energyaware computational resources.

3. It is used to model and simulate the message passing application's data center and its network topologies.

4. It is used to model and simulate the united clouds.

5. It is used to simulate dynamically the addition of segment, end and again start the simulation.

6. It is used to model and simulate the defined algorithm for allotment of virtual machines and planning to give resources of physical machine to the result of virtual machine.

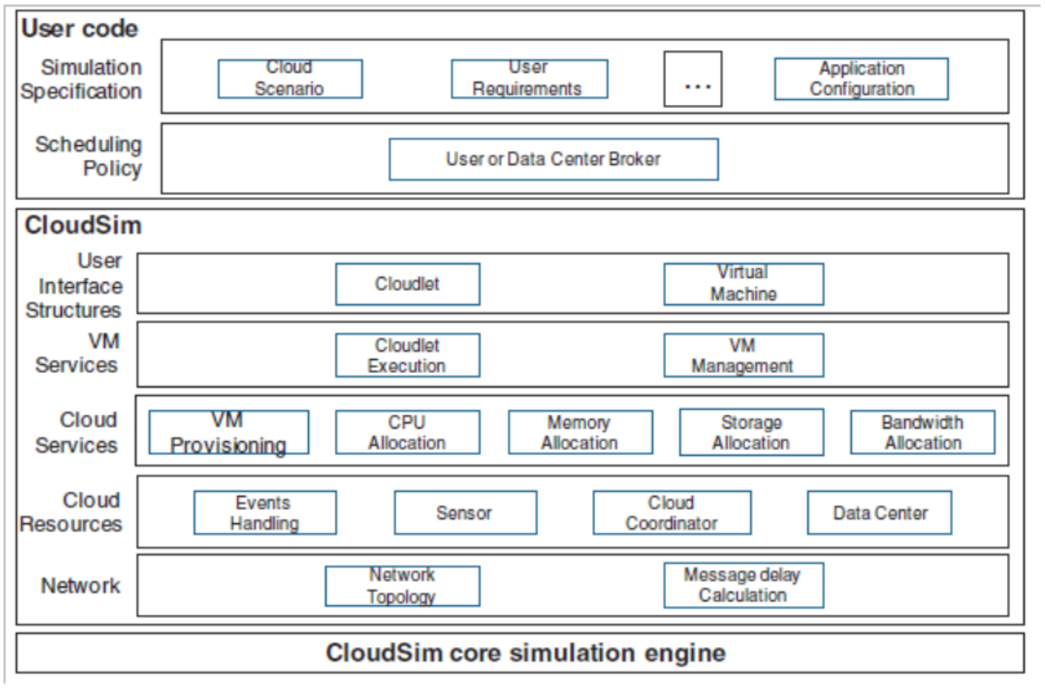


Figure 3.3.1 CloudSim Core Simulation Enginer

Components of CloudSim

The 4 main components in the CloudSim are

1. Datacenter

2. Virtual Machine

3. Broker

4. Cloudlet

The datacenters in CloudSim have the resources for provision of its users. The task of virtual machines is to allocate the resources to the users. The role of broker is to provide the facility of available resources to the users. The task of cloudlets is to customize the job of the cloud providers.

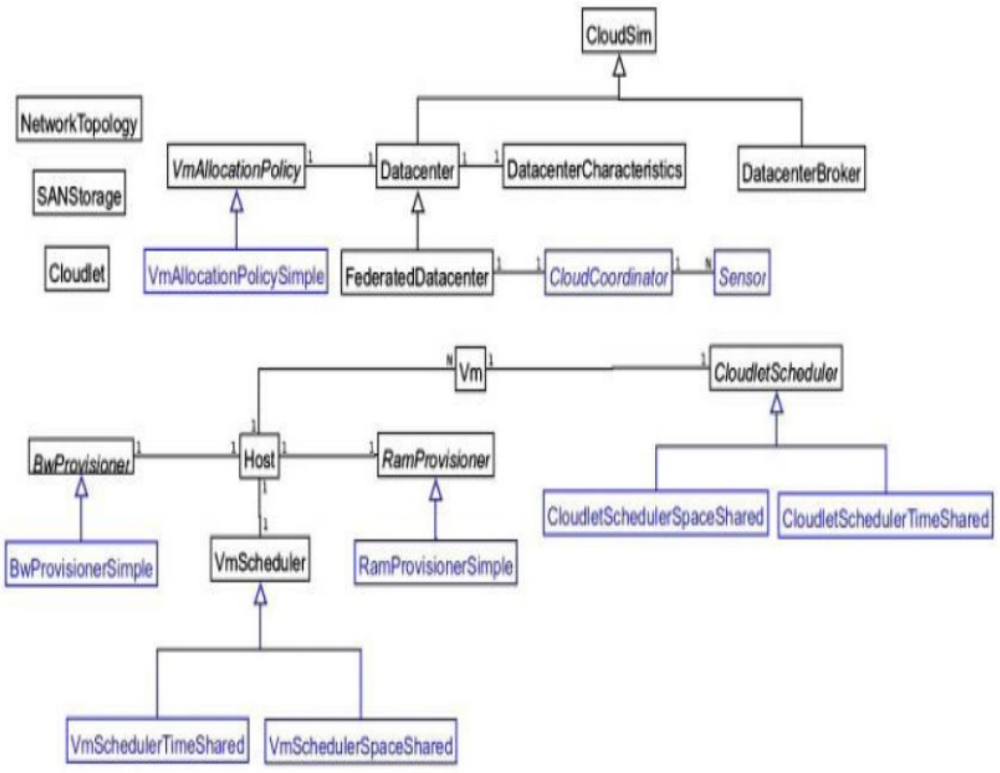


Figure 3.3.2 : CloudSim class design diagram

In Figure 3.3.2 , there are basic classes of CloudSim. They are also called as main components of the simulator.

2 *BwProvisioner*: This is an abstract class which is the blueprint of the arrangement for providing of transmission capacity to virtual machines. The fundamental job in this segment is to embrace the allotment of system's transfer speeds to a lot of contending virtual machines that are conveyed over the data center. The designers and specialists of cloud system can expand 2 this class with their own strategies (need, QoS) to mirror the necessities of their applications. The BwProvisioningSimple permits a virtual machine to save data transfer

capacity, whereas, this is compelled by the all out accessible transmission capacity of the physical machine.

CloudCoordinator: This is an abstract class that expands a server on cloud as information centre in an organization. This is liable for intermittently observing inside condition of resources of data center. This is dependent on run time load shredding decision. The concrete implementation of cloud corrdinator segment incorporates particular sensors. The approach that ought go along with load shredding's time.

Checking resources of information centre is executed 2 by the updateDatacenter() procedure by inquiries the sensors. The Service Discovery is acknowledged in the setDatacenter() abstract procedure that stretched out for executing conventions and systems (multicast, broadcast, shared). This segment can reached out for reenacting 2 Cloud-based services, for example, the Amazon EC2 Load Balancer. Engineers intending to implement their application services over numerous clouds that can inherit this class for executing its inter cloud provisioning strategies.

Cloudlet: This class which is a blueprint of the cloud-based application administrations, (for example, 2 content delivery, social networking and business work process). CloudSim coordinates multifaceted nature of an application as far as its calculation necessities. Each application's 2 service has a pre-doled out guidance length and information move (both pre and post gets) overhead which is required to embrace during its lifetime. This is a clas which will likewise be reached out for help demonstrating for other execution and creation measurements for applications, for example, transaction 3 in database oriented applications.

CloudletScheduler: This is an abstract class which have been inherited in the execution of various strategies which decide the portion of handling 2 power among Cloudlets in a virtual machine. As It is portrayed lastly, two kinds 7 of provisioning strategies are offered: space-shared (Cloudet Scheduler Space Shared) and time-shared (Cloudlet Scheduler Time Shared).

Datacenter: This class which models the center framework level services (hardware) which are supplied by Cloud suppliers (Amazon, Azure, and App Engine). It exemplifies a lot of physical machines that compute either be homogeneous or heterogeneous regarding their configurations of hardware (memory, centers, limit, and capacity). Moreover, every Datacenter part starts up a summed up application provisioning segment that executes a lot of strategies for allotting transmission capacity, memory, and storage mechanism to Physical machines and virtual machines.

Data center Broker or Cloud Broker: This class models a representative, which is liable for intervening arrangements among SaaS and Cloud suppliers; and such exchanges are driven by QoS prerequisites. The dealer follows up in the interest of SaaS suppliers. It finds appropriate Cloud specialist co-ops by questioning the CIS and attempts online exchanges for distribution of resources/servicess that can meet the application's QoS needs. Analysts and system engineers must inherit this class for assessing and testing custom facilitating rules. The distinction within the representative and the CloudCoordinator is that the previous speaks to the client (for example choices of these parts are made so as to expand client related performance measurements), while the last follows up for the benefit of the data center, for example it attempts to boost the general execution of the information centre, without thinking about the requirements of explicit clients.

DatacenterCharacteristics: This class contains set-up data for information centre resources.

Host: This is a class which models a physical resource, for example, a process or storage server. It typifies significant data, for example, the measure of memory and capacity, a list and sort of processing centers (to speak to a multi-center machine), a distribution of strategy for sharing the handling power among virtual machines, and strategies for provisioning memory and transfer speed to the virtual machines.

Network Topology: This class contains the data for instigating system conduct (latencies) in the reenactment. It stores the topology's data, which is created utilizing the BRITE topology generator.

RamProvisioner: This is a abstract class that speaks to the provisioning strategy for dispensing essential memory (RAM) to the virtual machines. The run time and arrangement of virtual machine on a physical machine is practical just if the RamProvisioner part endorses that the physical machine has the necessary measure of free memory. The RamProvisionerSimple doesn't uphold any impediment on the measure of memory that a virtual machine may ask for. Be that as it may, on the off chance that the solicitation is past the accessible memory limit, at that point it is basically dismissed.

SanStorage: This is a class which models a capacity area network that is ordinarily encompassing in Cloud-based server as data centers for putting away huge pieces of information, (for example, Amazon S3, Azure blob storage). SanStorage executes a straightforward interface that can be utilized to reproduce capacity and recovery of any size of information, subject to the accessibility of system data transmission. Getting to records in a SAN at run time causes extra deferrals for task unit execution; this is because of the extra lags that are acquired in moving the information documents within the internal network of data center.

Sensor: This is an interface that must be actualized to start up a sensor segment which can be utilized by a Cloud Coordinator for checking explicit execution boundaries (energy utilization, resource usage). Review that, Cloud Coordinator uses the run time execution data for undertaking decision of load balancing. The techniques characterized by this interface are:

(i) initialize the base and most extreme edges for performance parameter

(ii) Occasionally calculate the estimation.

It can be utilized to display this present reality administrations offered by driving Cloud suppliers, for example, Amazon's CloudWatch and Microsoft Azure's FabricController. One data center may launch at least one Sensors, every one liable for observing performance metric of a particular data center.

Vm: This is a class which represents a virtual machine, which is overseen and facilitated by a cloud host segment. Each virtual machine part approaches a segment that stores the accompanying attributes identified with a virtual machine - open memory, processor, storage size, and the virtual machine's internal provisioning strategy that is reached out from abstract segment called the CloudletScheduler.

VmmAllocationPolicy: This is an abstract class that speaks to a provisioning strategy that a virtual machine's Monitor uses for apportioning virtual machines to host. The main usefulness of the Vmm Allocation Policy is to choose the accessible host in a data center that meets the memory, storage, and accessibility prerequisite for a virtual machine deployement.

VmScheduler: This is a abstract class which actualized by a Host segment that models the approaches (space-shared, time-shared) required for apportioning processor centers to virtual machines. The functionalities of this class can without much of a stretch be superseded to oblige application-explicit processor sharing approaches.

3.3.2 CloudAnalyst

The CloudAnalyst can be utilized to display a reproduced situation to consider the conduct of a huge scaled application on the Internet. In any case, it became clear that having a simple to utilize device with a degree of perception ability is far superior to only a toolbox. Such a tool isolates the re-enactment experiment set up practice from a programming exercise and empowers a modeller to focus on the recreation boundaries instead of the details of programming. It likewise empowers the modeller to execute recreations consistently with alterations to the parameters rapidly and without any problem. A graphical yield of the simulation results empowers the outcomes to be investigated all the more effectively and all the more productively and it might likewise help in rapidly featuring any issues with the exhibition and exactness of the simulation logic. In this manner we chose to build up a simulation tool before beginning the investigation.

3.3.2.1 Features of the Simulator

There are a few exceptionally attractive highlights of a device like the one depicted in the above area.

1 Ease of utilization Ease of setting up and executing a simulation test is the central matter of having a simulation device. The test system needs to give a simple to utilize graphical UI which is natural yet extensive.

2 Ability to characterize a reproduction with a serious extent of configurability and adaptability. Perhaps the most significant element is the degree of configurability the tool can give. A simulation, particularly of the idea of demonstrating something as perplexing as an Internet Application relies upon numerous parameters and more often than not the qualities for those parameters should be expected. In this way it is imperative to have the option to enter and change those parameters rapidly and effectively and rehash simulation.

3 Graphical yield An image is supposed to merit a thousand words. Graphical yield as tables and outlines is profoundly attractive to sum up the conceivably huge measure of insights that is gathered during the simulation. Such successful introduction helps in recognizing the significant examples of the yield boundaries and aides in correlations between related boundaries.

4 Repeatability of trials is a significant prerequisite of a test system. A similar test with similar boundaries should create comparative outcomes each time the recreation is executed. In any case the simulation turns out to be only an irregular arrangement of events as opposed to a controlled test. It is likewise useful to have the option to spare a trial (the arrangement of

input parameters) as a document and furthermore have the option to spare the consequences of an analysis as a record.

5 Ease of augmentation as of now referenced mimicking something like the Internet is a perplexing assignment and it is 9 far-fetched a 100% reasonable simulation system and a lot of info boundaries can be accomplished in a couple of endeavours. In this way the test system is relied upon to develop ceaselessly instead of a program that is composed unequivocally and afterward utilized persistently. Thusly the test system design should bolster expansions with insignificant exertion with reasonable structures.

Output of Simulation

Following are the factual measures created as yield of the simulation in the underlying variant of the test system.

1. Reaction time of the simulated application
    - Overall normal, least and most extreme reaction season of all client demands mimicked
    - The reaction time separated by client gatherings, situated inside geological districts
    - The reaction time additionally separated when indicating the example of progress over the term of a day.
2. The use examples of the application
    - How numerous clients utilize the application at what time from various districts of the world, and the general impact of that utilization on the server facilitating the application.
3. The time taken by server to support a client demand
    - The by and large solicitation preparing time for the whole simulation.
    - The normal, least and most extreme solicitation preparing time by every server.

- The reaction time variety design during the day as the load changes

4. The expense of activity

CloudAnalyst Design

The CloudAnalyst is based on head of CloudSim toolbox, by expanding CloudSim usefulness with the presentation of ideas that model web and web Application practices.

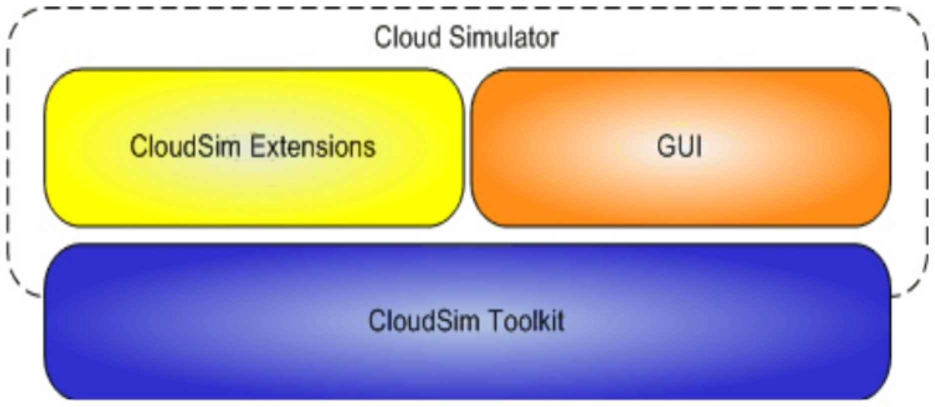


Figure 3.3.1 CloudAnalyst built on top of CloudSim toolkit

CHAPTER 4: PROPOSED METHODOLOGY

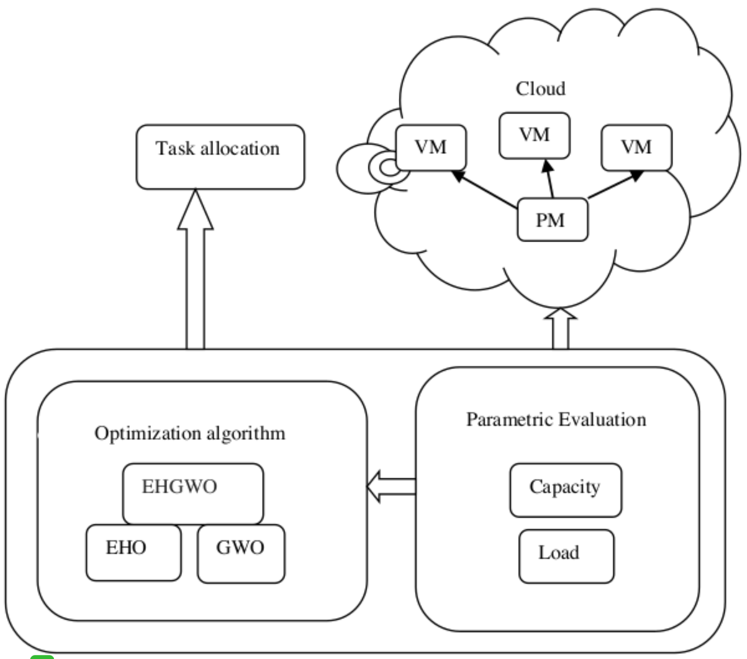
4.1 The Proposed Hybrid Method

An optimized hybrid method, named EHGWO, is proposed in the distributed computing model for load balancing by deciding the ideal VM's for running jobs which are allocated again. The introduced Elephant Herding Grey Wolf Optimizer is inferred by fusing Elephant Herding Optimizer in Grey Wolf Optimizer to achieve the end goal that the assignments are allotted to the virtual machine by disposing of the task from over-loaded virtual machine by keeping up the system performance. In this the physical machine’s load, limit, and virtual machine’s load is registered for choosing to go for the load balancing or not. In addition, 2 factors for picking the virtual machine and task, in particular Task Pitch Factor and Virtual machine Pitch Factor, are in consideration for picking the task for relocate them from over-burden virtual machine to under-loaded virtual machine. The proposed Hybrid algorithm chooses the task to be designated in the virtual machine dependent on the recently determined procedure.

The significant commitments for load adjusting method are as per the following,

- Designing an optimized Hybrid method, named Elephant Herding Grey Wolf Optimizer, by coordinating EHO into GWO, for choosing the ideal virtual machines for re-allocation of the assignments to adjust the load.
- Developing two choice possibilities, Task Pitch factor and Virtual machine pitch factor, in view of specific parameters, for example, need level, running time, correspondence cost, limit and virtual machine’s load. Thus, the task can be appointed to virtual machine adequately utilizing the choice possibilities planned.
- Originating a new function utilizing 5 variables, to be specific virtual machine’s load, physical machine’s load, vitual machine’s limit, correspondence cost of task, and makespan. In this manner, the ideal virtual machine is picked utilizing the proposed

Elephant Herding Grey Wolf Optimizer for taking care of explicit task based on fitness function.



-

Figure 4.1. Block diagram of the introduced Hybrid Method for balancing load

4.2 Algorithm for optimized Hybrid method for load balancing in cloud computing

The pseudocode of the load balancing algorithm using the proposed EHGWO algorithm is presented in Algorithm 4.2.

Algorithm 4.2. An Hybrid based Load balancing Algorithm	
1	Initialize the tasks, $Y_s = 100$
2	For each task
3	Assign the task to VMs based on round robin scheduling
4	Compute $\ell(M_g^P)$ using equation (8).
5	If $\ell(M_g^P) > 0.8$

6	Call load balancing algorithm using the proposed EHGWO algorithm
7	Else
8	No load balancing
9	End if
10	End for
11	Repeat

In this section, the newly devised Hybrid algorithm for adjusting the load for improvisation in the productivity of the distributed computing framework is clarified. The introduced algorithm is structured by coordinating EHO [21] and GWO [20] for choosing the ideal virtual machine. The objective is to adjust the heaps present in the virtual machine so that no machine is over-burden by expelling the undertakings from over-burden virtual machines and appoint them to under loaded virtual machines. In this way, it is fundamental to pick the task, which should be reallocated in a viable way. At first, the limit of burdens involved by the virtual machine is resolved utilizing the executed undertakings and afterward, the balance is assessed. At whatever point the virtual machine's load is lopsided, the limit with load is assessed to take the choice for the load adjusting. The Elephant Herding-based Grey Wolf Optimizer is adjusted for reallocating the undertakings by adjusting the load, at whatever point the virtual machine is over-burden thinking about specific variables, which include data transfer capacity, running time, need and correspondence cost. When the assignments are evacuated, it is added to different virtual machines for running the task.

The proposed model plays out the reallocation of undertakings for adjusting the load by apportioning the assignments from under loaded virtual machine to over-burden virtual machine by upgrading the cloud framework productivity.

Here, the status of load is checked once the undertakings are allotted to the virtual machines and afterward, the reallocation is finished. Along these lines, the physical machine's load and virtual machine load, and limit of virtual machine are assessed for apportioning the task. These assessments are used for relocating the assignments from the over-burden virtual machine and the virtual machine that can deal with the load are resolved utilizing the load adjusting calculation.

Here, two pitch factors, specifically Task pitch factor (TPF) and virtual machine pitch factor (VPF) are planned, which choose the determination of tasks. These components decide the appropriate under stacked virtual machines for reallocating the undertakings that are taken from the over-burden virtual machine. In the interim, the virtual machine pitch factor is registered utilizing the limit and the virtual machine's load, for finding the under stacked virtual machines and can be used for reallocating the assignments.

4.3 System model

The framework design contrived for adjusting the loads in distributed computing condition is appeared in Fig. 4.3. The objective of the contrived model is to locate the quantity of undertakings which are over-burden and allot those assignments to the virtual machine so that the load is adjusted. Expect a cloud situation that comprises of all h quantity of physical machines, and their portrayal is $PM^P = \{PM_1^P, PM_2^P, \cdots, PM_g^P, \cdots PM_h^P\}$; $1 \le g \le h$, and every Physical Machine contains various virtual machines. Consider the virtual machines contained in the g^{th} physical machine communicated as $VN = \{VN_1, VN_2, \cdots, VN_p, \cdots, VN_q\}$; $1 \le p \le q$, where q is sum of all virtual machines in the g^{th} physical machine. Here, the task mentioned by every client is relegated to the virtual machine in cooperative way and is spoken as $Y = \{Y_1, Y_2, \ldots, Y_s, \ldots, Y_t\}$. ,where t is total quantity of tasks allocated to the virtual machine. At whatever point the status of load of virtual machine is in ordinary express, the undertakings are handled typically, and when the framework faculties over-burden express, a load adjusting calculation is adjusted for allocating the assignments from over-burden virtual machine to under loaded virtual machine.

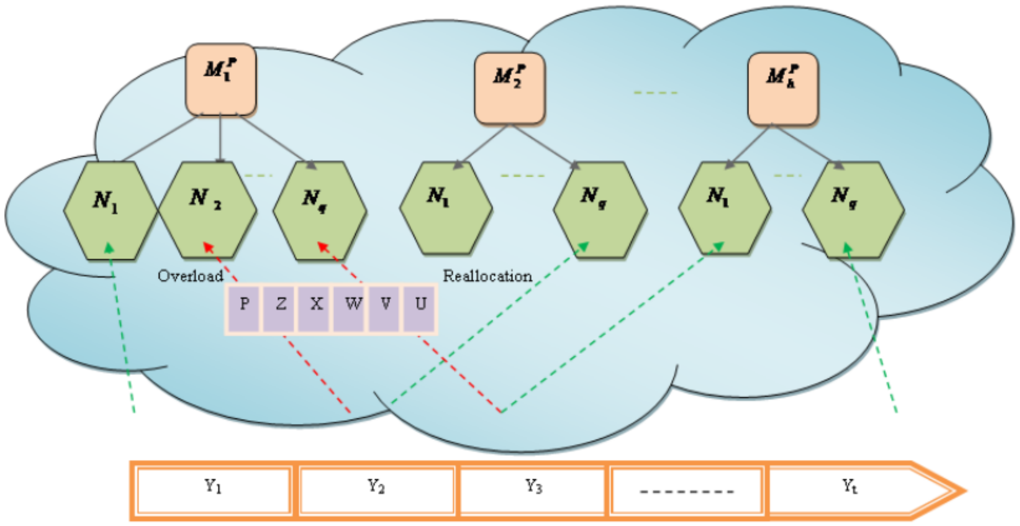


Figure 1.1 System model

Each virtual machine chose for adjusting the load is arranged dependent on certain boundaries, similar to data transfer capacity, processors, storage, Million Instructions Per Second (MIPS), movement expense, and recurrence and are communicated as

$$N_{g,p} = \{P_{g,p}, Z_{g,p}, X_{g,p}, W_{g,p}^{B}, V_{g,p}^{C}, U_{g,p}\} \tag{E1}$$

where, $P_{g,p}$ represents the memory of p^{th} VM in g^{th} PM and, $Z_{g,p}$ denotes the number of processors of p^{th} VM in g^{th} PM, $X_{g,p}$ specifies number of MIPS in p^{th} VM in g^{th} PM, $W_{g,p}^{B}$ is the bandwidth in p^{th} VM in g^{th} PM, $V_{g,p}^{C}$ specify the migration cost of task and $U_{g,p}$ specify the frequency. The parameters $P_{g,p}, Z_{g,p}, X_{g,p}, W_{g,p}^{B}, V_{g,p}^{C}$, and $U_{g,p}$ takes a floating point number that ranging from 1 to d where d is the constant , with the end goal that d=10 . The virtual machine is over-burden if the estimation of above characterized boundaries surpasses the predefined go. Along these lines, the undertaking distributed to the over-burden virtual machine should be doled out on another virtual machine.

Each undertaking Y_s thinks about the boundaries, similar to transfer speed, correspondence cost, need and run-time for the assessment.

$$Y_s = \{W_s^x, T_s, R_s, V_s^L\} \tag{E2}$$

here, W_s^x is the bandwidth, T_s which shows the run-time, R_s demonstrates the need level and alludes the correspondence cost of undertakings, in this the estimations of their boundaries lie somewhere in the range of 1 and 10. The tasks picked for the reallocation are resolved utilizing these boundaries.

4.4 Expected Outcome

In [3], the performance of the LB-BC method is evaluated using makespan as one of the metrics, wherein the minimum makespan is approximately 550 sec. However, the proposed load balancing method will be implemented in such a way to reduce the makespan than that in [3].

CHAPTER 5 : A Hybrid algorithm for load balancing in cloud computing

5.1 The hybrid algorithm based on Elephant Herding optimizer and Grey Wolf Optimizer for load balancing

In this segment, the designed Hybrid algorithm based on Elephant Herding optimizer and Gray Wolf Optimizer introduced for adjusting the load to improve the proficiency of the distributed computing framework is clarified. The proposed Hybrid is structured by coordinating EHO [21] and GWO [20] for choosing the ideal virtual machine. The objective is to adjust the loads present in the VM so that no machine is over-burden by expelling the undertakings from over-burden virtual machines and dole out them to under loaded virtual machines. Therefore, it is fundamental to pick the undertakings, which should be reallocated in a powerful way. At first, the limit of burdens involved by the virtual machine is resolved utilizing the executed undertakings and afterward, the equalization is assessed. At whatever point the load of virtual machine is lopsided, the limit with load is assessed to take the choice for the load adjusting. The introduced algorithm is adjusted for reallocation of the assignments by adjusting the load, at whatever point the virtual machine is over-burden thinking about specific elements, which include data transfer capacity, running time, need and correspondence cost. When the assignments are evacuated, it is added to different virtual machine for completing the task. Figure 5.1 displays the model for load adjusting utilizing the proposed hybrid algorithm calculation.

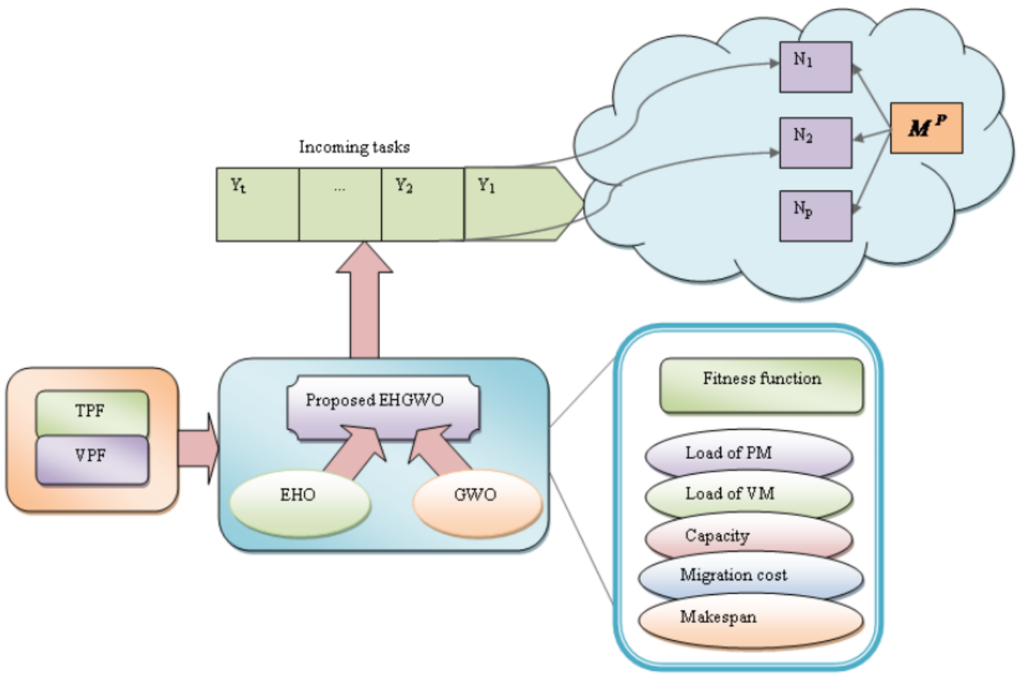


Figure 5.1 A Hybrid Load Balancing Algorithm's Schematic Diagram

5.2 A Hybrid Load Balancing Algorithm's Model

The introduced model plays out the re-allocation of undertakings for adjusting the load by distributing the tasks from over-burden virtual machine to under-burden virtual machine by upgrading the cloud framework effectiveness.

Here, the status for load is checked once the undertakings are appointed to the virtual machines and afterward, the reallocation is finished. Along these lines, the physical machine's load and virtual machine's load, and limit of virtual machine are assessed for designating the undertakings. These assessments are made for relocating the task from the over-burden virtual machine and the virtual machine that can deal with the loads are resolved utilizing the load adjusting calculation.

a) Virtual machine's capacity: Consider the limit of burden be spoken to as $D(N_p)$, which is assessed dependent 1 on the quantity of hardware, cost, and recurrence and is spoken to as,

$$D(N_p) = \left[F_p + G_p * H_p\right] * \frac{1}{3\times 10} \tag{E3}$$

In this, F_p indicates number of hardwares in p^{th} VM, G_p is the cost of p^{th} VM, and H_p is the frequency of p^{th} VM.

The number of hardwares required for computing the capacity is represented as,

$$F_p = \left[\frac{P_p + Z_p}{2}\right] \tag{E4}$$

The expense of the registering gadget is spoken to as far as data transfer capacity and relocation expense as,

$$G_p = \left[\frac{W_p^B + [1 - V_p^C]}{2}\right] \tag{E5}$$

The recurrence for processing the limit is dictated by

$$H_p = \left[\frac{X_p + U_p}{2}\right] \tag{E6}$$

b) Virtual Machine's load: The load of virtual machines is processed dependent on the task's running time, data transfer capacity and relocation expense of the virtual machine and is spoken to as

$$\ell(N_p) = \frac{\frac{1}{|Y_b|}\sum_{\forall Y_o on\, N_p} T_s}{K_f} + \left[\frac{1}{|Y_b|}\sum_{|Y_o|}\left(W^x + V^C\right)\right] * \frac{1}{2*10} \tag{E7}$$

In this T_s determines the running time of s^{th} task, K_f speaks to standardization parameter, and the quantity of undertakings appointed is meant as Y_o, and the quantity of undertakings stayed for the task is demonstrated as Y_b.

c)Physical Machine's Load : The physical machine's load relies upon that of virtual machine, and is determined utilizing the loads of virtual machine and is shown as,

$$\ell\left(M_g^P\right)=\frac{1}{q}\sum_{p=1}^{q}\ell\left(N_p\right) \tag{E8}$$

where, $\ell\left(N_p\right)$ means the virtual machine's load and q is the complete number of virtual machines in the cloud.

5.3 Devising pitch factors

In this, there are 2 factors for picking virtual machine and tasks, to be specific Task pitch factor (TPF) and virtual machine pitch factor (VPF) are structured, which choose the determination of assignments. These components decide the appropriate under stacked virtual machines for reallocating the assignments that are taken from the over-burden virtual machine. In the mean time, the VPF is registered utilizing the limit and the virtual machine's load, for finding the under-stacked virtual machines and can be used for allocating again the tasks.

a) Task Pitch factor: The undertakings to be reallocated are picked based on Task pitch factor for performing load adjusting from the over-burden virtual machine. Here, four boundaries, which incorporate data transmission, need, correspondence expense, and running time, are thought of. These boundaries are utilized for registering the Task Pitch factor and is given by

$$E_p^S=\frac{1}{4\times 10}\left[W_s^{x}T_sR_sV_s^{L}\right] \tag{E9}$$

Here, W_s^Q means bandwidth, R_s speaks to the need level, T_s shows the runtime, and V_s^L means the correspondence expense of the undertaking. The estimation of Task Pitch Factor ranging somewhere in the range of 0 and 1.

b) Virtual Machine Pitch factor: Virtual Machine Pitch factor decides the under loaded virtual machines feasible for reallocating the tasks and contains two angles, in particular limit of virtual machines and load of virtual machines, and is given by,

$$E_p^I = \frac{1}{3} * \left[\left[1 - \frac{Y_o}{K_f} \right] + D(N_p) + \left(1 - \ell(N_p)\right) \right] \qquad \text{(E10)}$$

An under loaded virtual machine is chosen so that its ability is greatest, and the load is least and the estimation of virtual machine pitch factor ranges somewhere in the range of 0 and 1. In this way, the undertaking with high task pitch factor is wiped out from the over-burden virtual machine and be reallocated to the virtual machine with greatest virtual machine pitch factor.

5.4 Proposed Hybrid Algorithm for load balancing

This area explains the introduced load adjusting calculation formulated based on Elephant Herding Grey Wolf Optimizer for improving the load adjusting in cloud model. The introduced Hybrid algorithm based on Elephant Herding and Grey Wolf Optimizer calculation decides the ideal virtual machine for executing the reallocated undertakings, acquired due to over-burdening. Elephant Herding Grey Wolf is structured by joining Elephant Herding Optimizer into the position update procedure of Grey Wolf Optimizer. Grey Wolf Optimizer [20] is a meta-heuristic procedure motivated from the grey wolves chasing conduct for taking care of the old style building issues for indicating elite for un-obliged issues. Elephant Herding Optimizer [21] is a swarm based meta-heuristic strategy utilized for taking care of worldwide streamlining issues. Coordinating Elephant Herding Optimizer and Grey Wolf Optimizer, the presentation of the Elephant Herding Grey Wolf Optimizer can be upgraded by improvisation in the intermingling speed.

5.4.1 Solution Encoding

The encoding of arrangements is significant in resulting the ideal answer for the enhancement issues. In this, the introduced Hybrid algorithm is utilized for choosing the suitable arrangement, in which the undertakings and the virtual machines are chosen from the accessible arrangements. Take $Y = \{1, 2, \ldots, r\}$ indicates the arrangement speaking to the assignments to be picked, in this r alludes the complete quantity of assignments to be browsed the over-burden virtual machines. These undertakings are reallocated to the under stacked virtual machine. Accordingly, the under loaded virtual machines are chosen for the errand reallocation dependent on the wellness work. The objective is to reallocate undertakings to the under stacked virtual machines utilizing the proposed Hybrid Algorithm. The arrangement encoding of the introduced Hybrid calculation for adjusting the load is delineated in Fig. 5.4.1.

In this there are r task which are fetched by TPF which needs to be reallocated and there are l virtual machines fetched by VPF which are under-burden. So we encode the solution in such a way with minimum cost we shift the task having higher priority to the virtual machine which are under burden from the list of l. In this way we get the optimal virtual machine selected using the introduced method for re-allocating the task.

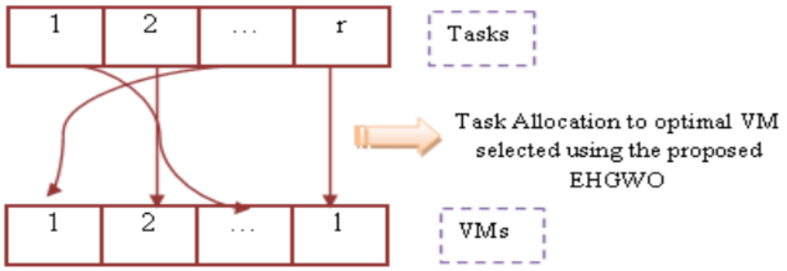

1
2
...
r
Tasks
Task Allocation to optimal VM selected using the proposed EHGWO
1
2
...
l
VMs

Figure 5.4.1 Solution Encoding

5.4.2 Multi-objective Fitness Function

The fitness function is assessed for resulting the ideal outcome from an answer set. The fitness function defined for introduced Elephant Herding Grey Wolf Optimizer depends on five boundaries, which include Physical machine's load, VM's load, Virtual machine's limit, task movement expense, and makespan. The fitness of the outcome is taken as an expansion work with the goal that an ideal virtual machine can be chosen for the run. The wellness capacity of the introduced Hybrid algorithm is spoken to as

$$F_j = \left[1 - \ell(M_g^p)\right] * \sum_{p=1}^{q} \left\{\left[1 - \ell\left(N_p\right)^s\right] + D(N_p)^s + \left[1 - (V^C)^s\right]\right\} \times \frac{1}{3} + \sum_{p=1}^{q} \left[\frac{1 - M_w}{K_f}\right] \qquad \text{(E11)}$$

where, $\ell\left(N_p\right)^s$ is the virtual machine's load for the s^{th} task, $D(N_p)$ indicates the limit of virtual machine for assignment, speaks to the undertaking movement cost, and speaks to the makespan.

5.4.3 Hybrid Algorithm's Steps

The proposed Hybrid algorithm for playing out the load adjusting is represented in this area. Elephant Herding Optimizer [21] is enlivened from the elephants crowding conduct, which is used for taking care of the worldwide advancement issues and can yield best answers for the vast majority of the standard issues though Grey Wolf Optimizer [20] is inspired from the gray wolves, which is liable for giving elite in obscure pursuit spaces and can take care of obliged and unconstrained issues. The proposed Hybrid algorithm joins Elephant Herding Optimizer and Grey Wolf Optimizer, for assessing the update condition. The means engaged with the proposed calculation are summed up as follows,

The number of inhabitants in the grey wolf is introduced haphazardly with a documentation $S_k \;\; ;1 \le k \le m$. The portrayal of the introduced populace is,

$$S_k = \left\{S_k^1, S_k^2, \cdots, S_k^m\right\} \qquad \text{(E12)}$$

where, m is the quantity of arrangements required.

ii) Fitness evaluation

The wellness of the outcome in the underlying populace is registered utilizing the wellness inferred in condition (E11) under the segment 4.2.2. The necessity of the calculation is to recognize the arrangement having most extreme wellness esteem as the best fit arrangement. The initial 3 arrangements with best wellness esteems are named as 1st best wellness operator, 2nd best wellness specialist and 3rd best wellness specialist, individually.

iii) Evaluation of Position Update

For improvisation in the search space, the introduced calculation utilizes the position update regulation of Grey Wolf Optimizer. As indicated by Grey Wolf Optimizer, the position update regulation put in over the calculation is spoken to as,

$$S(y+1) = \frac{S_1 + S_2 + S_3}{3} \qquad \text{(E13)}$$

In this, S_1, S_2, and S_3 are the positions at a time y, and are spoken as,

$$S_1 = S_\alpha - Q_1 A_\alpha \qquad \text{(E14)}$$

$$S_2 = S_\beta - Q_2 A_\beta \qquad \text{(E15)}$$

$$S_3 = S_\delta - Q_3 A_\delta \qquad \text{(E16)}$$

where, S_α, S_β, and S_δ are the principal best inquiry specialist, second best hunt operator and third best pursuit specialist, Q_1, Q_2 and Q_3 speak to the coefficient vector , and A_α, A_β and A_δ are the separations between the pursuit specialists, alpha, beta, and delta and the grey wolf position. Subbing conditions (E14), (E15), and (E16) in equation (E13),

$$S(y+1) = \frac{S_\alpha - Q_1 A_\alpha + S_\beta - Q_2 A_\beta + S_\delta - Q_3 A_\delta}{3} \qquad \text{(E17)}$$

The distance boundaries are processed utilizing a coefficient vector increased by the relating search specialists, which is then deducted from the arrangement at once y as

$$A_\alpha = | I_1 S_\alpha - S(y) | \qquad \text{(E18)}$$

$$A_\beta = | I_2 S_\beta - S(y) | \quad \text{(E19)}$$

$$A_\delta = | I_3 S_\delta - S(y) | \quad \text{(E20)}$$

Subbing conditions (E18), (E19), and (E20) in equation (E17),

$$S(y+1) = \frac{S_\alpha - Q_1 | I_1 S_\alpha - S(y) | + S_\beta - Q_2 | I_2 S_\beta - S(y) | + S_\delta - Q_3 | I_3 S_\delta - S(y) |}{3} \quad \text{(E21)}$$

where, I signifies coefficient vector. When the best three pursuit operators are resolved, the refreshing principle is applied over the calculation by accepting S_α, S_β and S_δ more noteworthy than $S(y)$ and is spoken to as,

$$S(y+1) = \frac{S_\alpha + S_\beta + S_\delta - Q_1(I_1 S_\alpha - S(y)) - Q_2(I_2 S_\beta - S(y)) - Q_3(I_3 S_\delta - S(y))}{3} \quad \text{(E22)}$$

The position update of the pursuit operators is given by

$$S(y+1) = \frac{S_\alpha + S_\beta + S_\delta - Q_1 I_1 S_\alpha - Q_1 S(y)) - Q_2 I_2 S_\beta - Q_2 S(y)) - Q_3 I_3 S_\delta - Q_3 S(y))}{3} \quad \text{(E23)}$$

After rearranging the above equation,

$$S(y+1) = \frac{S_\alpha + S_\beta + S_\delta - Q_1 I_1 S_\alpha - Q_2 I_2 S_\beta - Q_3 I_3 S_\delta + S(y)[Q_1 + Q_2 + Q_3]}{3} \quad \text{(E24)}$$

For improving the position, the calculation refreshes the situation by receiving the Elephant Herding Optimizer calculation. As indicated by Elephant Herding Optimizer, the position update is given by,

$$S(y+1) = S(y) + \mu \times (S^* - S(y)) \times a \quad \text{(E25)}$$

where, $S(y+1)$ and $S(y)$ mean situation of elephant at next and existing emphasis, μ indicate scale factor, S^* indicate the best position, and a denotes a random value range between 0 and 1.

Throughout the following emphasis, the position vectors are refreshed according to the refreshed boundaries and is given by,

$$S(y+1) = S(y) + \mu a S^{*} - \mu a S(y)) \tag{E26}$$

Subsequent to reworking the conditions, the position update is given by,

$$S(y)[1-\mu a] = S(y+1) - \mu a S^{*} \tag{E27}$$

The last position update of Elephant Herding Optimizer for acquiring superior situation by considering the situation of recent emphasis is given by,

$$S(y) = \frac{S(y+1) - \mu a S^{*}}{1-\mu a} \tag{E28}$$

Subbing condition (E28) in equation (E24),

$$S(y+1) = \frac{1}{3}\left[\begin{array}{l} S_{\alpha} + S_{\beta} + S_{\delta} - Q_1 I_1 S_{\alpha} - Q_2 I_2 S_{\beta} - Q_3 I_3 S_{\delta} + \\ [Q_1 + Q_2 + Q_3]\dfrac{S(y+1) - \mu a S^{*}}{1-\mu a} \end{array}\right] \tag{E29}$$

In the wake of reworking the above condition,

$$S(y+1) = \frac{1}{3}\left[\begin{array}{l} S_{\alpha} + S_{\beta} + S_{\delta} - Q_1 I_1 S_{\alpha} - Q_2 I_2 S_{\beta} - Q_3 I_3 S_{\delta} \\ +[Q_1 + Q_2 + Q_3]\dfrac{S(y+1)}{1-\mu a} - [Q_1 + Q_2 + Q_3]\dfrac{\mu a S^{*}}{1-\mu a} \end{array}\right] \tag{E30}$$

$$S(y+1) - \frac{1}{3}[Q_1 + Q_2 + Q_3]\frac{S(y+1)}{1-\mu a} = \frac{1}{3}\left[\begin{array}{l} S_{\alpha} + S_{\beta} + S_{\delta} - \\ Q_1 I_1 S_{\alpha} - Q_2 I_2 S_{\beta} - Q_3 I_3 S_{\delta} - [Q_1 + Q_2 + Q_3]\dfrac{\mu a S^{*}}{1-\mu a} \end{array}\right] \tag{E31}$$

$$\frac{3(1-\mu a)S(y+1) - [Q_1 + Q_2 + Q_3]S(y+1)}{3(1-\mu a)} = \frac{1}{3}\left[\begin{array}{l} S_{\alpha} + S_{\beta} + S_{\delta} - Q_1 I_1 S_{\alpha} \\ - Q_2 I_2 S_{\beta} - Q_3 I_3 S_{\delta} - [Q_1 + Q_2 + Q_3]\dfrac{\mu a S^{*}}{1-\mu a} \end{array}\right] \tag{E32}$$

$$\frac{S(y+1)[3(1-\mu a) - (Q_1 + Q_2 + Q_3)]}{3(1-\mu a)} = \frac{1}{3}\left[\begin{array}{l} S_{\alpha} + S_{\beta} + S_{\delta} - Q_1 I_1 S_{\alpha} \\ - Q_2 I_2 S_{\beta} - Q_3 I_3 S_{\delta} - [Q_1 + Q_2 + Q_3]\dfrac{\mu a S^{*}}{1-\mu a} \end{array}\right] \tag{E33}$$

The situation of the arrangements subsequent to coordinating Elephant Herding Optimizer and Grey Wolf Optimizer calculation is refreshed as

$$S(y+1)=\frac{(1-\mu a)}{3(1-\mu a)-(Q_1+Q_2+Q_3)}\begin{bmatrix} S_\alpha+S_\beta+S_\delta-Q_1 I_1 S_\alpha \\ -Q_2 I_2 S_\beta-Q_3 I_3 S_\delta-[Q_1+Q_2+Q_3]\frac{\mu a S^*}{1-\mu a}\end{bmatrix} \qquad \text{(E34)}$$

As the best arrangement in Elephant Herding Optimizer and the principal fittest arrangement in Grey Wolf Optimizer are proportionate, the last position update condition becomes

$$S(y+1)=\frac{(1-\mu a)}{3(1-\mu a)-(Q_1+Q_2+Q_3)}\begin{bmatrix} S_\beta+S_\delta-Q_1 I_1 S_\alpha \\ -Q_2 I_2 S_\beta-Q_3 I_3 S_\delta-\left([Q_1+Q_2+Q_3]\frac{\mu a}{1-\mu a}-1\right)S_\alpha\end{bmatrix} \qquad \text{(E35)}$$

iv) Assurance of feasible arrangement

The plausible arrangements are gotten dependent on the wellness and the arrangement that positioned the most noteworthy, structures the best arrangement S_α.

v) End

The cycle is proceeded for the most extreme number of emphases and ends upon the age of the worldwide ideal arrangement.

CHAPTER 6: EXPERIMENTAL RESULTS AND ANALYSIS

6.1 Results and Discussion

In this segment, the experimentation of introduced Hybrid Algorithm based on EHO and GWO as far as burden and makespan is explained.

6.1.1 Experimental Set-up

The experiment is conducted on a Computer with Windows 10 operating system; RAM 8GB I7 Processor and is actualized in java language. In this, 2 cloud arrangements are utilized for the execution.

I) Setup 1: In setup1, the complete tasks allocated to the virtual machine are 100, and the load adjusting is conveyed by changing the quantity of physical machines as 10, 20, and 30, individually.

ii) Setup 2: In setup2, the all task doled out to the virtual machine are 200, and the load adjusting is conveyed by changing the quantity of PMs as 10, 20, and 30, individually.

6.1.2 Evaluation Metrics

The measurements utilized for the investigation incorporate burden and makespan and they are detailed as,

a) Load: It is characterized as the quantity of undertakings relegated to the virtual machine and is registered utilizing condition (E7).

b) Makespan: The makespan is the greatest burden on any virtual machine.

6.1.3 Comparative Methods

This area delineates the near strategies thought about, which includes Constraint Measure based LB(CMLB)[22] which is denoted by C-22, Fractional Dragonfly based LB Algorithm (FDLA)[23] which is denoted by F-23, Elephant Herding Optimizer(Applied EHO [21] rather than Elephant Herding Grey Wolf Optimizer in the proposed method), and Grey Wolf

Optimizer(Applied Grey Wolf Optimizer[20] rather than Elephant Herding Grey Wolf Optimizer in the proposed strategy).

6.1.4 Comparative Analysis

This area delineates the near examination of the current and proposed techniques regarding load and makespan utilizing two arrangements.

6.1.4.1 Analysis with PM=10

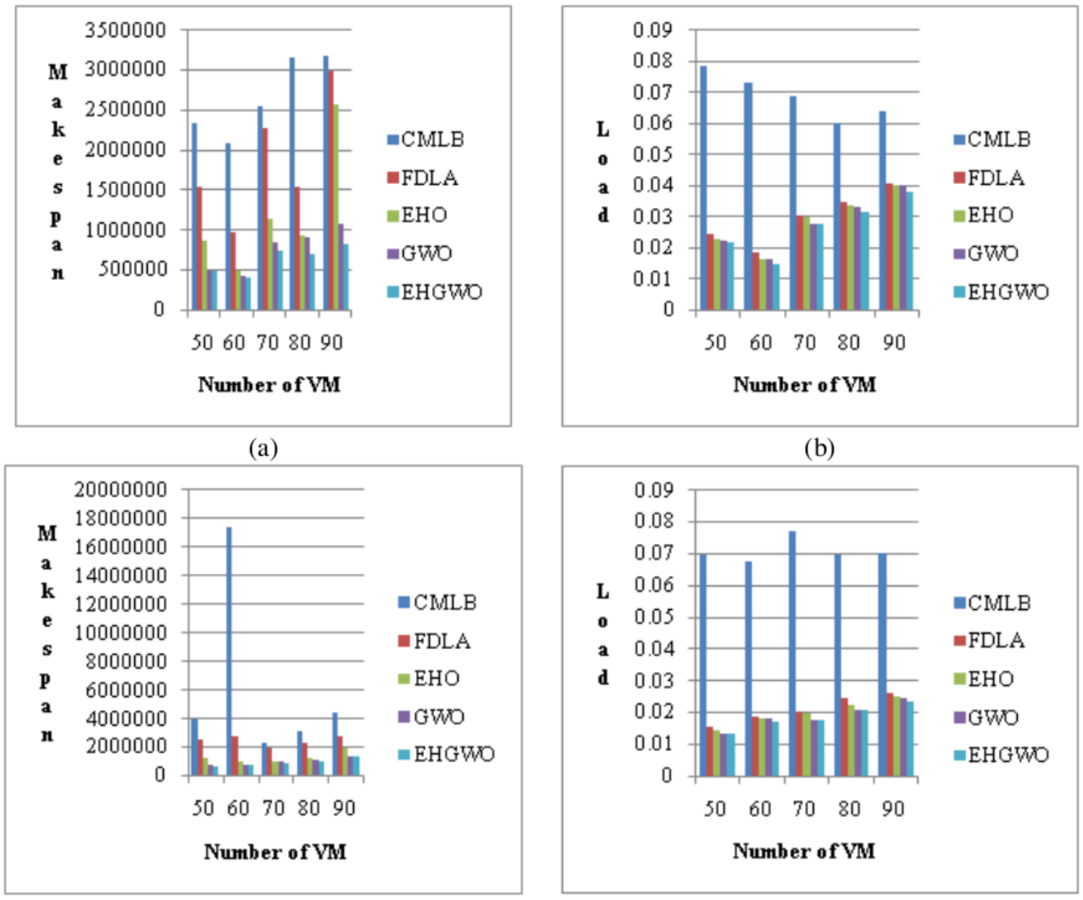


Figure 6.1.4.1 Investigation thinking about 10 PMs for estimating a) Makespan utilizing cloud set up 1 b) Load utilizing cloud set up 1 c) Makespan utilizing cloud set up 2 d) Load utilizing cloud arrangement 2.

The investigation of C-22, F-23, Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid i.e Elephant Herding Grey Wolf Optimizer as far as burden and makespan is portrayed in Figure 6.1.4.1. The investigation is finished by differing the quantity of virtual machine. The investigation of makespan boundary utilizing cloud setup1 with 100 quantities of tasks is portrayed in figure 6.1.4.1 a. At the point when the quantity of virtual machines is 90, the makespan values estimated by existing C-22 , F-23, Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 3186896ns, 3000789ns, 2562055ns, 1073863ns, and 814264ns, separately. Essentially, where the quantity of assignments is fifty, the relating makespan values estimated by current C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 2344096ns, 1543197ns, 867727ns, 488867ns, and 479100ns. The examination as far as burden for arrangement 1 with 100 undertakings is portrayed in figure 6.1.4.1b. At the point when the quantity of virtual machines is 90, the heap esteems estimated by existing C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20] are 0.064035, 0.0406,0.0403, and 0.04. In the mean time, the heap esteem estimated by existing Hybrid is 0.0379, individually. Also, when the quantity of assignments is 50, the comparing load esteems estimated by C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20] and proposed Hybrid are 0.078438, 0.0243, 0.0226, 0.0223, and 0.0216. The examination regarding 1 makespan for arrangement 2 with 200 undertakings is delineated in Fig. 6.1.4.1 c. For 90 virtual machines, the makespan values estimated by C-22, F-23, Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 4348914ns, 2763051ns, 1891727ns, 1307759ns, and 1245558ns, individually. So also, when the quantity of undertakings is 50, the relating makespan values estimated by C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 3905798ns, 2465413ns, 1202377ns, 663646ns, and 593221ns.The examination dependent on load boundary utilizing arrangement 2 with 200 tasks is delineated in Fig. 6.1.4.1 d. At the point when the quantity of virtual machine is 90, the relating load esteems estimated by C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20] and proposed Hybrid are 0.0705, 0.026, 0.0248, 0.0247, and 0.02365, separately. So also, when the quantity of undertakings is 50, the comparing load esteems estimated by C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 0.069, 0.015, 0.014, 0.013, and 0.013, separately.

6.1.4.2 Analysis with PM=20

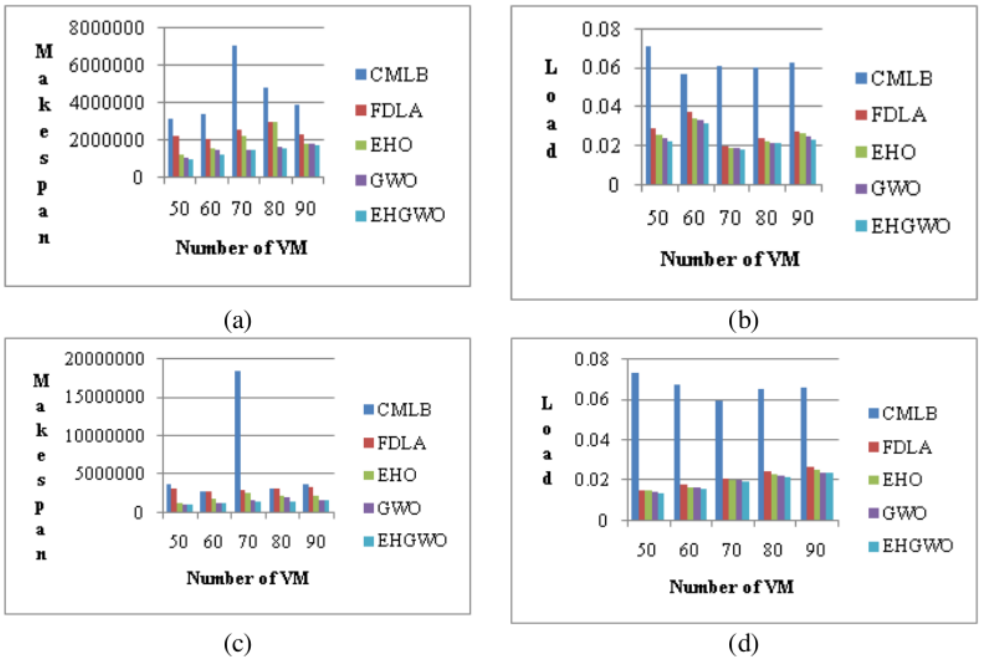
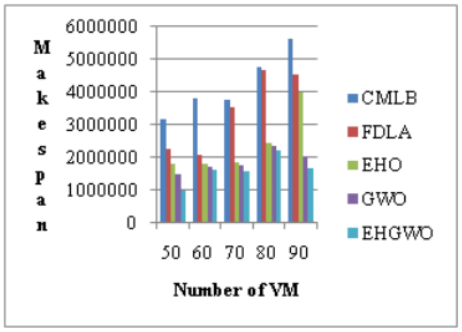


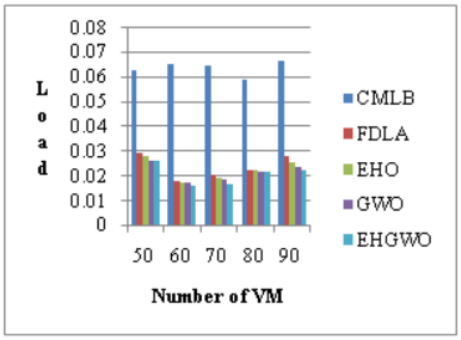
Figure 6.1.4.2 Investigation thinking about 20 physical machines for estimating a) Makespan utilizing cloud set up 1 b) Load utilizing cloud set up 1 c) Makespan utilizing cloud set up 2 d) Load utilizing cloud arrangement 2.

Figure 6.1.4.2 delineates the examination of C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid dependent on load and makespan by changing the quantity of virtual machines. Fig. 6.1.4.2 a delineates the investigation of makespan boundary utilizing cloud setup1 with 100 quantities of tasks with 20 physical machine. At the point when the quantity of virtual machine is 90, the makespan values estimated by existing Constraint Measure LB is 3887806ns, Fractional Dragonfly based LB algorithm is 2309140ns, Elephant Herding Optimizer is 1804851ns, Grey Wolf Optimizer is 1802280ns, and proposed Hybrid is1755501ns, respectively. Likewise, when the quantity of undertakings is 50, the relating makespan values estimated by C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 3133686ns, 273156ns, 1218313ns, 1102137ns, and 992642ns, individually. The examination dependent on load utilizing arrangement 1 with 100 undertakings is delineated in figure 6.1.4.2 b. At the point when the quantity of virtual machines is 90, the load esteems estimated by Constraint Measure LB is 0.0627, Fractional Dragonfly based LB algorithm is 0.0275, Elephant Herding Optimizer is 0.0268, and Grey Wolf Optimizer is 0.0255, individually. Then, the load esteem estimated by proposed Hybrid is 0.0236. Essentially, where the quantity of undertakings is 50, the relating load esteems estimated by existing C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and introduced Hybrid are 0.071, 0.0297, 0.026, 0.0248, and 0.023,respectively. The examination as far as makespan for arrangement 2 with 200 assignments is delineated in figure 6.1.4.2 c. For 90 VMs, the makespan values estimated by current Constraint Measure based LB is 3819437ns, Fractional Dragonfly based LB algorithm is 3456513ns, Elephant Herding Optimizer is 2181140ns, Grey Wolf Optimizer is 1766297ns, and proposed Hybrid is 1726200ns, separately. Additionally, when the quantity of undertakings is 50, the relating makespan values estimated by C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and introduced Hybrid are 3685268ns, 3286874ns, 1228594ns, 1173591ns, and 1051244ns, separately. The investigation dependent on load boundary utilizing arrangement 2 with 200 tasksis portrayed in figure 6.1.4.2 d. At the point when the quantity of virtual machine is 90, the relating load esteems estimated by Constraint Measure based LB is 0.065, Fractional Dragonfly based LB algorithm is 0.0266, Elephant Herding Optimizer is 0.0251, Grey Wolf Optimizer is 0.0243, and introduced Hybrid is 0.0239, respectively. Thus, when the quantity of assignments is 50, the comparing load esteems estimated by C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 0.0731, 0.015, 0.01495, 0.0145, and 0.01355, respectively.

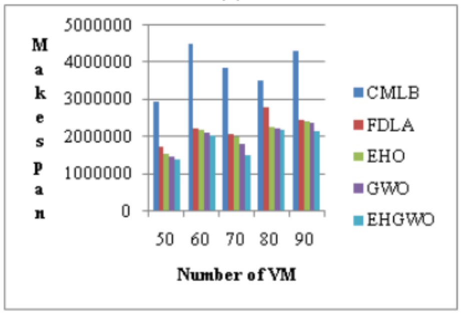
6.1.4.3 Analysis with PM=30



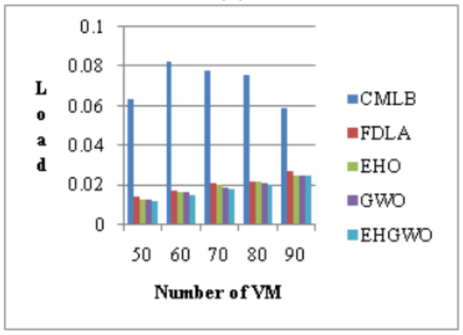
(a)



(b)



(c)



(d)

Figure 6.1.4.3 Investigation thinking about 30 physical machines for estimating a) Load utilizing cloud set up 1 b) Makespan utilizing cloud set up 2 c) Load utilizing cloud set up 2 d) Makespan utilizing cloud arrangement 2.

The similar examination of C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid regarding load and makespan inspecting thirty physical machine with thirty physical machines portrayed in Fig. 6.1.4.3. Fig. 6.1.4.3 a shows the examination of makespan boundary utilizing cloud setup 1 with hundred quantities of assignments. At the point when the quantity of virtual machines is 90, the makespan values estimated by C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 5652561ns, 4537059ns, 3994215ns, 2067534ns, and 1669140ns,respectively. The examination as far as burden utilizing arrangement 1 with 100 undertakings is delineated in figure 6.1.4.3 b. At the point when the quantity of virtual machines is 90, the load esteems estimated by C-22, F-23, Elephant Herding Optimizer[21], Grey Wolf Optimizer[20] are 0.0668, 0.0276, 0.0251, 0.0234 respectively. In the interim, the load esteem estimated by proposed Hybrid is 0.0221. The examination as far as makespan for arrangement two with 200 errands is delineated in figure 6.1.4.3 c. For 90 virtual machines, the makespan results estimated by C-22, F-23, Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 4325782ns, 2453590ns, 2416578ns, 2404755ns, 2148755ns, respectively. The examination dependent on load boundary utilizing arrangement 2 with 200 tasks is delineated in Fig. 6.1.4.3 d. At the point when the quantity of virtual machines is 90, the relating load esteems estimated by C-22, F-23, Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid are 0.0587, 0.02735, 0.0253, 0.0251, and 0.025, separately. In future, we will investigate the degree of burden adjusting of the proposed Hybrid.

6.1.5 Comparative Study

The relative conversation of the strategies is shown in this segment and delineated in table 6.1.5. Here, the conversation is done by making the consideration of the greatest exhibition dependent on load and makespan utilizing two cloud arrangements by changing quantity of physical machines as ten, twenty and thirty.

Table 6.1.5 Comparative Study

Comparative Methods	***Makespan (ns)***	***Load***
CMLB	3186896	0.0587
FDLA	2309140	0.026
EHO	1804851	0.0248
GWO	1073863	0.0234

Proposed EHGWO	**814264**	**0.0221**

From table 6.1.5, the base burden and makespan got in the current techniques, C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20], and proposed Hybrid algorithm for the greatest number of virtual machines is delineated in this area. The designed Hybrid algorithm accomplished least makespan with esteem 814264ns and least burden with esteem 0.0221, individually. In the interim, the makespan values accomplished by existing C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20] are 3186896ns, 2309140ns, 1804851ns, and 1073863ns, separately. The base burden esteems figured by existing strategies, C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20] are 0.0587, 0.026, 0.0248, and 0.0234. Then again, the proposed Hybrid with least burden esteem is 0.0221. Subsequently, the proposed Hybrid achieves most extreme execution when contrasted with the current strategy.

The complexity of computation of the Elephant Herding Optimizer is $O(m)$and the complexity of computation of the Grey Wolf Optimizer is $O(mt)$. Thus, the computational intricacy of the proposed Hybrid is $O(mt)$ + , $O(m)$where, m is the quantity of arrangements and t is the greatest amount of cycles.

CHAPTER 7 : CONCLUSION AND FUTURE WORK

7.1 Conclusion

The introduced load adjusting calculation utilizing hybrid algorithm based on Elephant Herding Optimizer and Grey Wolf optimizer in a distributed computing framework utilizing two pitch factors, named Task Pitch Factor and Virtual machine Pitch Factor. For starting burden adjusting, the tasks doled out to the over-burden VM are allocated to under-stacked virtual machines. Here, the proposed load adjusting calculation adjusts limit and loads for the reallocation. In view of Task Pitch Factor and Virtual machine Pitch Factor, the undertakings are reallocated from virtual machines utilizing the proposed hybrid algorithm. The proposed hybrid algorithm is created by coordinating Elephant Herding Optimizer and Grey Wolf Optimizer calculation utilizing another wellness work detailed by load of virtual machine, relocation cost,virtual machine's load, virtual machine's limit, and makespan. The proposed Hybrid algorithm is broke down dependent on load and makespan. The presentation of proposed algorithm is contrasted and the current procedures, as C-22, F-23,Elephant Herding Optimizer[21], Grey Wolf Optimizer[20] in which the proposed Hybrid algorithm accomplishes least burden and least makespan with values 0.0221 and 814264ns, individually.

7.2 Future Work

The main focus of this research is to design and implement a load balancing technique that would rather improve the efficiency of the distributed computing system. Here, load balancing is performed to remove the tasks from over-loaded VMs and allocate them to under loaded VMs without affecting the system performance. Based on this concept, our research dealt with the Elephant Herd Grey Wolf Optimizer (EHGWO) procedure for stabilize the load in the cloud based on the cost and load factors. The extension of the research is on the merging of the self-adaptive principle in the EHGWO algorithm with the inclusion of the energy and resource utilization factors for efficient load balancing in the cloud. Accordingly, a self-adaptive Elephant Herd optimization-based Grey Wolf Optimizer (self-adaptive EHGWO) will be newly developed algorithm to perform the load balancing. Initially, the energy, resource utilization, capacity and loads of the virtual machine will be found based on the number of tasks already executed, then, the balance factor of the cloud system will be checked. When the load is not found balanced, the volume with load will be checked to make the choice whether load balancing can be done or not. In the other case, find the tasks to be removed by checking two constraints, like load of the physical machines, load of virtual machines, cost of migrating task, and resource utilization. Then, the removed tasks will be added in other VMs by optimally finding the VMs for the task execution. The optimal finding of VMs for executing of the removed task will be found out using the proposed self-adaptive EHGWO, which will be designed by combining the self-adaptive characteristics in the hybridization of Elephant Herding Optimization and Grey Wolf Optimizer. The implementation of the proposed approach will be done in JAVA with Cloudsim tool. The efficiency of the proposed technique for load balancing will be evaluated with different cloud set up for makespan, load, resource utilization rate, and energy, and the results attained will be contrast with the current works.

References

[1]Shang-Liang Chen, Yun-Yao Chen ,Suang-Hong Kuo,"CLB: A novel load balancing architecture and algorithm for cloud services," Computers & Electrical Engineering,vol.58, pp.154-160, February 2017.

[2] Qi Liu, WeidongCai, JianShen, Xiaodong Liu, Nigel Linge, "An Adaptive Approach to Better Load Balancing in a Consumer-centric Cloud Environment," IEEE Transactions on Consumer Electronics, vol.62, no.3, pp.243 - 250, August 2016.

[3] Jia Zhao, Kun Yang, Xiaohui Wei, Yan Ding, Liang Hu, and Gaochao Xu, "A Heuristic Clustering-based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment", IEEE Transactions on Parallel and Distributed Systems, vol.27, no.2, pp.305-316, February 2016.

[4] Shiva Razzaghzadeh, Ahmad HabibizadNavin, Amir MasoudRahmani, and Mehdi Hosseinzadeh, "Probabilistic Modeling to Achieve Load balancing in Expert Clouds", Ad Hoc Networks, January 2017.

[5]Ranesh Kumar Naha and Mohamed Othman,"Cost aware service brokering and performance sentient load balancing algorithms in the cloud", Journal of Network and Computer Applications, vol.75, pp.47-57, November 2016.

[6]Weihua Huang, Zhong Ma, Xinfa Dai, MingdiXu and Yi Gao, "Fuzzy Clustering with Feature Weight Preferencesfor Load Balancing in Cloud, "International Journal of Software Engineering and Knowledge Engineering, vol.28, no.5, pp.593–617, 2018.

[7]Narander Kumar and DikshaShukla, “Load Balancing Mechanism Using Fuzzy Row Penalty Method in Cloud Computing Environment,”Information and Communication Technology for Sustainable Development, pp.365-373, 2017.

[8]Santanu Dam, GopaMandal, KousikDasgupta and ParmarthaDutta,”An Ant-Colony-Based Meta-Heuristic Approach for Load Balancing in Cloud Computing”, Applied Computational Intelligence and Soft Computing in Engineering, pp.29, 2018.

[9]Shalini Joshi and Uma Kumari,"Load Balancing in Cloud Computing Challenges & Issues," In proceedings of 2nd International Conference on Contemporary Computing and Informatics, December 2016.

[10]MahfoozAlam and Zaki Ahmad Khan, "Issues and Challenges of Load Balancing Algorithmin Cloud Computing Environment",Indian Journal of Science and Technology, Vol.10, no.25, July 2017.

[11]Wayne Jansen and Timothy Grance, “Guidelines on security and privacy in public cloud computing”, pp.800-144, 2011.

[12] JiantingNing, Zhenfu Cao, Xiaolei Dong, Kaitai Liang, Hui Ma and Lifei Wei," Auditable Time Outsourced Attribute-Based Encryption for Access Control in Cloud Computing", IEEE Transactions on Information Forensics And Security, 2017.

[13]ZenonChaczko, VenkateshMahadevan, ShahrzadAslanzadeh and Christopher Mcdermid," Availability and Load Balancing in Cloud Computing," In proceedings of International Conference in Computer and Software Modeling, IPCSIT, vol.14, 2011.

[14]Daraghmi, E.Y. and Yuan, S.M.,”A small world based overlay network for improving dynamic load-balancing”, Journal of Systems and Software, vol.107, pp.187-203, 2015.

[15]A. S. Milani and N. J. Navimipour,” Load balancing mechanisms and techniques in thecloud environments: Systematic literature review and future trends”, Journal Networking Computer Application, vol.71, no.1, pp.86–98, 2016.

[16]Y. C. Jiang, “A survey of task allocation and load balancing in distributed systems”, IEEE Trans. Parallel Distrib. Syst., vol.27, no.2, pp.585–599, 2016.

[17]G. Mateusz, G. Alicja and B. Pascal,” Cloud brokering: Current practices and upcomingchallenges”, IEEE Cloud Comput., vol.2, no.2, pp.40–47, 2015.

[18]I. D. Falco, E. Laskowski, R. Olejnik, U. Scafuri, E. Tarantino and M. Tudruj, “Extremal optimization applied to load balancing in execution of distributed programs”, Appl. Soft Comput.,vol.30, no.3, pp.501–513, 2015.

[19]. Y. Jiang, “Concurrent collective strategy diffusion of multiagents: The spatial model andcase study”, Journal Networking Computer Application, vol.39, no.4, pp.448–458, 2009.

[20]Mirjalili, S., Mirjalili, S.M. and Lewis, A.,”Grey wolf optimizer” Advances in engineering software, vol.69, pp.46-61, 2014.

[21]G. Wang, S. Deb and L. d. S. Coelho, "Elephant Herding Optimization, "In proceedings of 3rd International Symposium on Computational and Business Intelligence, pp.1-5, 2015.

[22]Polepally, V. and Chatrapati, K.S., "Dragonfly optimization and constraint measure-based load balancing in cloud computing," Cluster Computing, pp.1-13, 2017.

[23]Kumar, C.A., Vimala, R., Britto, K.A. and Devi, S.S., “FDLA: Fractional Dragonfly based Load balancing Algorithm in cluster cloud model,” Cluster Computing, pp.1-14, 2018.

[24] N. S. Ghumman and R. Kaur, "Dynamic combination of improved max-min and ant colony algorithm for load balancing in cloud system," in proceedings of 6th International Conference on Computing, Communication and Networking Technologies, pp. 1-5, 2015.

[25] Desai, Tushar, and Jignesh Prajapati., "A survey of various load balancing techniques and challenges in cloud computing.", International Journal of Scientific & Technology Research, vol. 2, no.11 pp.158-161, 2013.

[26] Mahfooz Alam and Mohammad Shahid, "A Load Balancing Strategy with Migration Cost for Independent Batch of Tasks BoT on Heterogeneous Multiprocessor Interconnection Networks," International Journal of Applied Evolutionary Computation, vol.8, no. 3, pp. 74-92, July 2017.

[27] Raza Abbas Haidri, Chittaranjan Padmanabh Katti, and Prem Chandra Saxena, "Capacity based deadline aware dynamic load balancing (CPDALB) model in cloud computing environment," International Journal of Computers and Applications, pp. 1-15, 2019.

[28] Pramod P Jadhav and SD Joshi, "ACADF: Ant Colony Unified with Adaptive Dragonfly Algorithm Enabled with Fitness Function for Model Transformation," ICCCE 2019, pp. 101-109, 2020.

[29] D Menaga and S Revathi, "Least lion optimisation algorithm (LLOA) based secret key generation for privacy preserving association rule hiding," IET Information Security, vol. 12, no. 4, pp. 332-340, 2018.

[30] Amolkumar Narayan Jadhav and Gomathi N, "DIGWO: Hybridization of Dragonfly Algorithm with Improved Grey Wolf Optimization Algorithm for Data Clustering," Multimedia Research, vol. 2, no. 3, pp. 1-11, 2019.

[31] Priya, V., Kumar, C.S. and Kannan, R., "Resource scheduling algorithm with load balancing for cloud service provisioning," Applied Soft Computing, vol.76, pp.416-424, 2019.

[32] Yang, C.T., Chen, S.T., Liu, J.C., Su, Y.W., Puthal, D. and Ranjan, R., "A predictive load balancing technique for software defined networked cloud services," Computing, vol.101, no.3, pp.211-235, 2019.

[33] Li, M., Zhang, J., Wan, J., Ren, Y., Zhou, L., Wu, B., Yang, R. and Wang, J., "Distributed machine learning load balancing strategy in cloud computing services," Wireless Networks, pp.1-17, 2019.

[34] Hussain, A., Aleem, M., Iqbal, M.A. and Islam, M.A., "SLA-RALBA: cost-efficient and resource-aware load balancing algorithm for cloud computing," The Journal of Supercomputing, pp.1-27, 2019.

[35] Qi Liu, WeidongCai, JianShen, Xiaodong Liu, Nigel Linge, "An Adaptive Approach to Better Load Balancing in a Consumer-centric Cloud Environment," IEEE Transactions on Consumer Electronics, vol.62, no.3, pp.243 - 250, August 2016.

[36]Weihua Huang, Zhong Ma, Xinfa Dai, MingdiXu and Yi Gao, "Fuzzy Clustering with Feature Weight Preferencesfor Load Balancing in Cloud, "International Journal of Software Engineering and Knowledge Engineering, vol.28, no.5, pp.593–617, 2018.

[37]Narander Kumar and DikshaShukla, “Load Balancing Mechanism Using Fuzzy Row Penalty Method in Cloud Computing Environment,”Information and Communication Technology for Sustainable Development, pp.365-373, 2017.

[38]Shang-Liang Chen, Yun-Yao Chen ,Suang-Hong Kuo,”CLB: A novel load balancing architecture and algorithm for cloud services,” Computers & Electrical Engineering,vol.58, pp.154-160, February 2017.

[39]Randles, M., Lamb, D. and Taleb-Bendiab, A., "A comparative study into distributed load balancing algorithms for cloud computing," in proceedings of 24th International Conference on Advanced Information Networking and Applications Workshops, IEEE, pp. 551-556, 2010.

[40]Al Nuaimi, K., Mohamed, N., Al Nuaimi, M. and Al-Jaroodi, J., "A survey of load balancing in cloud computing: Challenges and algorithms," In proceedings of Second Symposium on Network Cloud Computing and Applications, pp. 137-142, 2012.

[41]Hu, J., Gu, J., Sun, G. and Zhao, T., "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," In proceedings of 3rd International symposium on parallel architectures, algorithms and programming, pp. 89-96, 2010.

[42] LD, D.B. and Krishna, P.V., "Honey bee behavior inspired load balancing of tasks in cloud computing environments," Applied Soft Computing, vol.13, no.5, pp.2292-2303, 2013.

[43] Li, K., Xu, G., Zhao, G., Dong, Y., Wang, D., “Cloud Task Scheduling based on Load Balancing Ant Colony Optimization”, in proceedings of Sixth Annual Chinagrid Conference, pp. 3-9, 2011.

[44] Chen, H., Wang, F., Helian, N., Akanmu, G. “User-priority Guided Min-Min Scheduling Algorithm for Load Balancing in Cloud Computing”, 2013 National Conference on Parallel Computing Technologies, IEEE, pp. 1-8, 2013.

[45]Panwar, R., Mallick, B., "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm," In: Proceedings of IEEE International Conference on Green Computing and Internet of Things, pp. 773–778, 2015.

[46] Xue, S., Li, M., Xu, X., Chen, J. and Xue, S., "An ACO-LB algorithm for task scheduling in the cloud environment," Journal of Software, vol.9, no.2, pp.466-473, 2014.

[47]Guo, Q., "Task scheduling based on ant colony optimization in cloud environment," In proceedings of AIP Conference Proceedings, vol. 1834, no.1, pp.040039, 2017.

[48]Katyal, M. and Mishra, A., "A comparative study of load balancing algorithms in cloud computing environment," arXiv preprint arXiv:1403.6918, 2014.

[49]RajwinderKaur and PawanLuthra, "Load Balancing in Cloud Computing ," on Recent Trends in Information, Telecommunication and Computing, 2014.

[50]Soni, G. and Kalra, M., "A novel approach for load balancing in cloud data center," In proceedings of IEEE international advance computing conference, IEEE, pp. 807-812, 2014.

[51] Lei Feng and Wei Wei, "Adaptive Particle Swarm Optimization Algorithm and its Application," Journal of Software Engineering, vol.6, no.3, pp.41-48, 2012.

[52]Mirjalili, S., Mirjalili, S.M. and Lewis, A.,"Grey wolf optimizer" Advances in engineering software, vol.69, pp.46-61, 2014.

[53]G. Wang, S. Deb and L. d. S. Coelho, "Elephant Herding Optimization, "In proceedings of 3rd International Symposium on Computational and Business Intelligence, pp.1-5, 2015.

[54]Polepally, V. and Chatrapati, K.S., "Dragonfly optimization and constraint measure-based load balancing in cloud computing," Cluster Computing, pp.1-13, 2017.

[55]Kumar, C.A., Vimala, R., Britto, K.A. and Devi, S.S., "FDLA: Fractional Dragonfly based Load balancing Algorithm in cluster cloud model," Cluster Computing, pp.1-14, 2018.

AN APPROACH FOR LOAD BALANCING IN CLOUD USING HYBRID OPTIMIZATION ALGORITHM

ORIGINALITY REPORT

10%

SIMILARITY INDEX

PRIMARY SOURCES

1	Pooja Arora, Anurag Dixit. "An elephant herd grey wolf optimization (EHGWO) algorithm for load balancing in cloud", International Journal of Pervasive Computing and Communications, 2020 Crossref	909 words — 6%
2	www.buyya.com Internet	343 words — 2%
3	onlinelibrary.wiley.com Internet	119 words — 1%
4	baadalsg.inflibnet.ac.in Internet	93 words — 1%
5	C. Ashok Kumar, R. Vimala, K. R. Aravind Britto, S. Sathya Devi. "FDLA: Fractional Dragonfly based Load balancing Algorithm in cluster cloud model", Cluster Computing, 2018 Crossref	42 words — < 1%
6	www.slideshare.net Internet	39 words — < 1%
7	Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Software: Practice and Experience, 2011 Crossref	28 words — < 1%

8	www.coursehero.com Internet	14 words — < 1%

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 1%
EXCLUDE BIBLIOGRAPHY	ON		