Effective Resource Utilization in Cloud Environment for Load Balancing using Hybrid Optimization

[1]Aniya Ahuja, [2]Navjeet Kaur
[1,2]Doaba Institute of Engg. And Technology
([1]aniyaahuja94@gmail.com)

Abstract-Cloud computing, a framework for enabling convenient, and on-demand network access to a shared pool of computing resources, is emerging as a new paradigm of large-scale distributed computing. It has widely been adopted by the industry, though there are many existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation, Energy Management, etc. that are not fully addressed. Central to these issues is the issue of load balancing that is a mechanism to distribute the dynamic workload evenly to all the nodes in the whole cloud to achieve a high user satisfaction and resource utilization ratio.

In cloud Computing, Load Balancing is essential for efficient operations in distributed environments. To allocate and balance the load of the resources among the various components and nodes load balancing is required. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource.

In this paper, the proposed technique has been implemented using Java Programming and simulate the algorithms on CloudSim. The Main contribution of CloudSim is to provide a holistic software framework for modeling Cloud computing environments and performance testing application services. And, the proposed hybrid optimization gives better results as compared to existing technique. The results were also analyzed using various performance parameters such as Energy Consumption, Response time and Total Execution time. The value of parameters shows that the hybrid optimization gives the better results for all the proposed parameters when it compared with the existing technique.

Keywords*-cloud computing; load balancing ; cloudsim.*

I. INTRODUCTION

Cloud computing is a new technology and it is becoming popular because of its great features. In this technology almost everything like hardware, software and platform are provided as a service. A cloud provider provides services on the basis of client's requests. An important issue in cloud is, scheduling of users requests, means how to allocate resources to these requests, so that the requested tasks can be completed in a minimum time and the cost incurred in the task should also be minimum. In case of Cloud computing services can be used from diverse and wide spread resources, rather than remote servers or local machines. There is no standard definition of Cloud computing. Generally it consists of a bunch of distributed servers known as masters, providing demanded services and resources to different clients known as clients in a network with scalability and reliability of datacenter. The distributed computers provide on-demand services. Services may be of software resources (e.g. Software as a Service, SaaS) or physical resources (e.g. Platform as a Service, PaaS) or hardware/infrastructure (e.g. Hardware as a Service, HaaS or Infrastructure as a Service, IaaS). AmazonEC2 (Amazon Elastic Compute Cloud) is an example of cloud computing services [2].

A. Characteristics of Cloud Computing

1) Self Healing

Any application or any service running in a cloud computing [3] environment has the property of self-healing. In case of failure of the application, there is always a hot backup of the application ready to take over without disruption. There are multiple copies of the same application - each copy updating itself regularly so that at times of failure there is at least one copy of the application which can take over without even the slightest change in its running state.

2) Multi-tenancy

With cloud computing, any application supports multi-tenancy - that is multiple tenants at the same instant of time. The system allows several customers to share the infrastructure allotted to them without any of them being aware of the sharing. This is done by virtualizing the servers on the available machine pool and then allotting the servers to multiple users. This is done in such a way that the privacy of the users or the security of their data is not compromised.

3) Linearly Scalable

Cloud computing services are linearly scalable. The system is able to break down the workloads into pieces and service it across the infrastructure. An exact idea of linear scalability can be obtained from the fact that if one server is able to process say 1000 transactions per second, then two servers can process 2000 transactions per second [4].

4) Service-oriented

Cloud computing systems are all service oriented - i.e. the systems are such that they are created out of other discrete services. Many such discrete services which are independent of each other are combined together to form this service. This allows re-use of the different services that are available and that are being created. Using the services that were just created, other such services can be created.

5) SLA Driven

Usually businesses [5] have agreements on the amount of services. Scalability and availability issues cause clients to break these agreements. But cloud computing services are SLA driven such that when the system experiences peaks of load, it will automatically adjust itself so as to comply with the service-level agreements. The services will create additional instances of the applications on more servers so that the load can be easily managed.

6) Virtualized

The applications in cloud computing are fully decoupled from the underlying hardware. The cloud computing environment is a fully virtualized environment.

7) Flexible

Another feature of the cloud computing services is that they are flexible. They can be used to serve a large variety of workload types - varying from small loads of a small consumer application to very heavy loads of a commercial application.

B. Cloud Computing Application Architecture

We know that cloud computing is the shift of computing to a host of hardware infrastructure that is distributed in the cloud. The commodity hardware infrastructure consists of the various low cost data servers that are connected to the system and provide their storage and processing and other computing resources to the application. Cloud computing [6] involves running applications on virtual servers that are allocated on this distributed hardware infrastructure available in the cloud. These virtual servers are made in such a way that the different service level agreements and reliability issues are met. There may be multiple instances of the same virtual server accessing the different parts of the hardware infrastructure available. This is to make sure that there are multiple copies of the applications which are ready to take over on another one's failure. The virtual server distributes the processing between the infrastructure and the computing is done and the result returned.

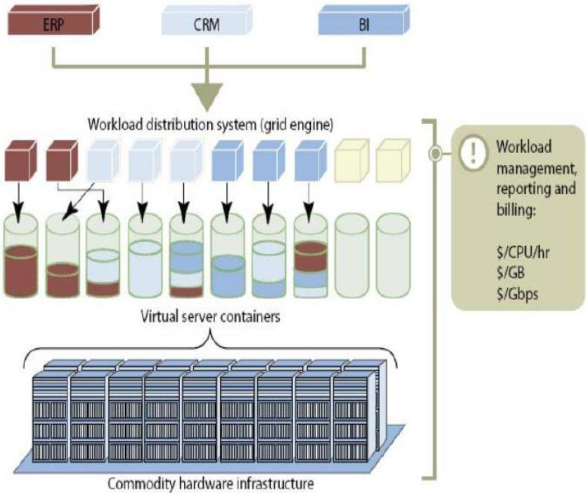

ERP
CRM
BI
Workload distribution system (grid engine)
Workload management, reporting and billing:
$/CPU/hr
$/GB
$/Gbps
Virtual server containers
Commodity hardware infrastructure

Fig.1: Shows the basic Cloud computing application architecture [6]

C. Cloud types

Together with virtualization [17], clouds can be defined as computers that are networked anywhere in the world with the availability of paying the used clouds in a pay-per-use way, meaning that just the resources that are being used will be paid. In the following the types of clouds will be introduced.

1) Public Clouds

A public cloud encompasses the traditional concept of cloud computing, having the opportunity to use computing resources from anywhere in the world. The clouds can be used in a so-called pay-per-use manner, meaning that just the resources that are being used will be paid by transaction fees.

2) Private Clouds

Private clouds are normally datacenters that are used in a private network and can therefore restrict the unwanted public to access the data that is used by the company. It is obvious that this way has a more secure background than the traditional public clouds. However, managers still have to worry about the purchase, building and maintenance of the system.

3) Hybrid Clouds

As the name already reveals, a hybrid cloud is a mixture of both a private and public cloud. This can involve work load being processed by an enterprise data center while other activities are provided by the public cloud. Below an overview of all three clouds computing types is illustrated.

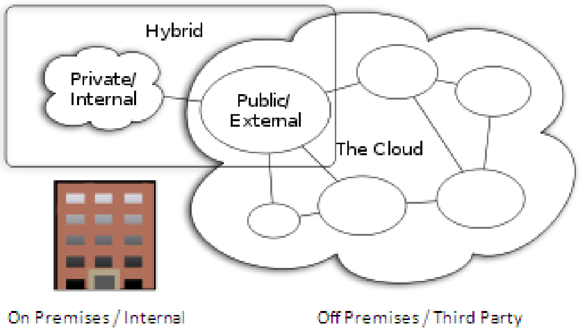


Fig.2: Cloud Computing Types [2]

D. Load Balancing on Cloud Computing

With the increasing popularity of cloud computing, the amount of processing that is being done in the clouds is surging drastically. A cloud [7] is constituted by various nodes which perform computation according to the requests of the clients. As the requests of the clients can be random to the nodes they can vary in quantity and thus the load on each node can also vary. Therefore, every node in a cloud can be unevenly loaded of tasks according to the amount of work requested by the clients. This phenomenon can drastically reduce the working efficiency of the cloud as some nodes which are overloaded will have a higher task completion time compared to the corresponding time taken on an under loaded node in the same cloud. This problem [9] is not only confined only to cloud but is related with every large network like a grid, etc.

Load balancing in large distributed server systems is a complex optimization problem of critical importance in cloud systems and data centers. Load balancing [10] algorithms are classified as static and dynamic algorithms. Static algorithms are mostly suitable for homogeneous and stable environments and can produce very good results in these environments. However, they are usually not flexible and cannot match the dynamic changes to the attributes during the execution time. Dynamic algorithms are more flexible and take into consideration different types of attributes in the system both prior to and during run-time [12]. These algorithms can adapt to changes and provide better results in heterogeneous and dynamic environments. However, as the distribution attributes become more complex and dynamic. As a result some of these algorithms could become inefficient and cause more overhead than necessary resulting in an overall degradation of the services performance.

E. Dynamic Load balancing algorithm

In a distributed system, dynamic load balancing can be done in two different ways: distributed and non-distributed. In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. The interaction among nodes to achieve load balancing can take two forms: cooperative and non-cooperative [22]. In the first one, the nodes work side-by-side to achieve a common objective, for example, to improve the overall response time, etc. In the second form, each node works independently toward a goal local to it, for example, to improve the response time of a local task. Dynamic load balancing algorithms of distributed nature, usually generate more messages than the non-distributed ones because, each of the nodes in the system needs to interact with every other node. A benefit, of this is that even if one or more nodes in the system fail, it will not cause the total load balancing process to halt, it instead would effect the system performance to some extent. Distributed dynamic load balancing can introduce immense stress on a system in which each node needs to interchange status information with every other node in the system. It is more advantageous when most of the nodes act individually with very few interactions with others.

In non-distributed type [20], either one node or a group of nodes do the task of load balancing. Non-distributed dynamic load balancing algorithms can take two forms: centralized and semi-distributed. In the first form, the load balancing algorithm is executed only by a single node in the whole system: the central node. This node is solely responsible for load balancing of the whole system. The other nodes interact only with the central node.

In semi-distributed form [22], nodes of the system are partitioned into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in each cluster by appropriate election technique which takes care of load balancing within that cluster. Hence, the load balancing of the whole system is done via the central nodes of each cluster [4]. Centralized dynamic load balancing takes fewer messages to reach a decision, as the number of overall interactions in the system decreases drastically as compared to the semi-distributed case. However, centralized algorithms can cause a bottleneck in the system at the central node and also the load balancing process is rendered useless once the central node crashes. Therefore, this algorithm is most suited for networks with small size.

F. Policies or Strategies in dynamic load balancing

There are 4 policies [14]:

1) Transfer Policy: The part of the dynamic load balancing algorithm which selects a job for transferring from a local node to a remote node is reffered to as Transfer policy or Transfer strategy.

2) Selection Policy: It specifies the processors involved in the load exchange (processor matching)

3) Location Policy: The part of the load balancing algorithm which selects a destination node for a transferred task is referred to as location policy or Location strategy.

4) Information Policy: The part of the dynamic load balancing algorithm responsible for collecting information about the nodes in the system is referred to as Information policy or Information strategy.

G. CloudSim Architecture

The layered implementation of the CloudSim software framework and architectural components. At the lowest layer is the SimJava discrete event simulation engine [16] that implements the core functionalities required for higher-level simulation frameworks such as queuing and processing of events, creation of system components (services, host, data center, broker, virtual machines), communication between components, and management of the simulation clock. Next follows the libraries implementing the GridSim toolkit [19] that support high level software components for modeling multiple Grid infrastructures, including networks and associated traffic profiles, and fundamental Grid components such as the resources, data sets, workload traces, and information services. The CloudSim is implemented at the next level by programmatically extending the core functionalities exposed by the GridSim layer. CloudSim provides novel support for modeling and simulation of virtualized Cloudbased data center environments such as dedicated management interfaces for VMs, memory, storage, and bandwidth. CloudSim layer manages the instantiation and execution of core entities (VMs, hosts, data centers, application) during the simulation period. This layer is capable of concurrently instantiating and transparently managing a large scale Cloud infrastructure consisting of thousands of system components. The fundamental issues such as provisioning of hosts to VMs based on user requests, managing application execution, and dynamic monitoring are handled by this layer. A Cloud provider, who wants to study the efficacy of different policies in allocating its hosts, would need to implement his strategies at this layer by programmatically extending the core VM provisioning functionality. There is a clear distinction a this layer on how a host is allocated to different competing VMs in the Cloud. A Cloud host can be concurrently shared among a number of VMs that execute applications based on user-defined QoS specifications.

II. LITERATURE REVIEW

Load balancing in the cloud computing environment has an important impact on the performance. Good Load balancing makes cloud computing more efficient and improves user satisfaction. There have been many studies of load balancing for the Cloud environment.

Atyaf Dhari et al. 2017) proposed Load Balancing Decision Algorithm (LBDA) to manage and balance the load between the virtual machines in a datacenter along with reducing the completion time (Makespan) and Response time. Findings: The mechanism of LBDA is based on three stages, first calculates the VM capac¬ity and VM load to categorize the VMs' states (Under loaded VM, Balanced VM, High Balance VM, Overloaded). Second, calculate the time required to execute the task in each VM. Finally, makes a decision to distribute the tasks among the VMs based on VM state and task time required. Improvements: We compared the result of our proposed LBDA with Max- Min, Shortest Job Firstand Round Robin.

Mohammad Goudarzi et al. (2017) evaluated the efficiency of the proposed solution using both simulation and testbed experiments. The evaluation study demonstrated that proposal can outperform existing optimal and near-optimal counterparts in terms of weighted execution cost, energy consumption and execution time. Due to nowadays advances of mobile technologies in both hardware and software, mobile devices have become an inseparable part of human life. Along with this progress, mobile devices are expected to perform various types of applications.

Muhammad Baqer Mollah et al. (2017**)** presented the main security and privacy challenges in this field which have grown much interest among the academia and research community. Although, there are many challenges, corresponding security solutions have been proposed and identified in literature by many researchers to counter the challenges. We also present these recent works in short. Furthermore, we compare these works based on different security and privacy requirements, and finally present open issues. The rapid growth of mobile computing is seriously challenged by the resource constrained mobile devices. However, the growth of mobile computing can be enhanced by integrating mobile computing into cloud computing, and hence a new paradigm of computing called mobile cloud computing emerges.

M. Vanitha et al. (2017) proposed, involving a well-organized use of resources, which is known as the dynamic well-organized load balancing (DWOLB) algorithm. This is a powerful algorithm for reducing the energy that is consumed in cloud computing. Cloud computing is used in almost all domains today. Through the use of cloud-based applications, it has become easier for an internet user to make use of the services and re- sources that are widely available. The cloud service provider undertakes to deliver all the subscribers' requirements as per the service level agreement (SLA). These resources must be well-protected since they are used by many subscribers.

Weidong Cai et al. (2016) Presented the proposed load balancing approaches in Map Reduce aim at optimizing task execution time, whereas disk space is not considered. In the research, a new scheme which consists of modified K-ELM and NSGA-II is proposed. Corresponding experiment results have shown that our method can assign tasks evenly, and effectively improve the performance of a cloud system. Map Reduce is a popular programming model widely used in distributed systems. With regard to large-scale applications, e.g. home energy management in a city, online social community etc., load-balancing becomes critical affecting the performance of distributed computing.

Oshin Sharma et al. (2015) provided a clear view of various energy management techniques used for mobile devices and performance analysis of various cloud computing techniques for energy efficient devices. For this, we have given brief introduction and our vision in the field of green computing. In addition to this, they performed performance analysis of various load balancing techniques based on six different cases. For energy efficient mobile devices, cloud computing is very much essential by providing storage and performing computations in the network. With the help of cloud computing many devices can connect over internet and can access the resources at anytime from anywhere.

III. RESEARCH PROBLEM FORMULATION

A. *Problem formulation*

In a cloud environment, there may be any number of host machines and each host machine has different-different load due to virtual machines as per the client's demand. The load of a host machine may be of various types such as CPU load, Memory load, Storage load and Network related load etc. If the load of any host machine exceeds its capacity then it affects its efficiency. In runtime, any client application service may change their resource (CPU, RAM, Storage and Bandwidth etc.) demand and this causes the host system to be imbalanced. If this imbalanced situation occurs due to overloading then system is balanced using load balancing techniques by distributing the extra workload to the whole clouds host heaving light loads. This helps to improve the overall performance of the cloud system.

Multi-objective Load Balancing is defined as a process of making effective resource utilization by reassigning the total load to the individual nodes of the collective system and thereby minimizing the response time of the job. Load Balancing algorithms are classified as Static and Dynamic algorithms. Static algorithms are most suitable for homogenous and stable environments. However, they cannot match the dynamic changes to the attributes during execution time. Dynamic algorithms take into consideration different types of attributes in the system both prior to and during run time. These algorithms can adapt to changes and provide better results in heterogeneous and dynamic environments.

In the past, a number of load balancing algorithms have been developed specifically to suit the dynamic cloud computing environments such as INS (Index Name Server) algorithm[A], WLC (Weighted Least Connection) algorithm[B], DDFTP (Dual direction Downloading algorithm from FTP servers)[C], LBMM (Load Balancing Min-Min) algorithm[D], ACO(Ant Colony Optimization) algorithm[E] and Bee-MMT(Artificial Bee Colony algorithm- Minimal Migration time)[F]. We are going to use the PSO (Particle Swarm Optimization) algorithm for load balancing in dynamic cloud environments as particle swarm has already get better results than genetic and ACO in grid computing[G]. Performance of Particle Swarm Optimization has also been approved better in distributed system [12]. In the proposed research, the bat optimization algorithm for task scheduling and load balancing on cloud computing will be implemented. The proposed algorithm will also compare with the load balancing decision algorithm for evaluation purpose. The results of the proposed work will be analyzed on the basis of Energy Efficiency, execution time and response time.

B. *Objectives*

The key objective of this research work is to optimize the performance of the cloud architecture. Overloaded nodes across the server and storage side often lead to performance degradation and are more vulnerable to various failures. To remove this limitation the load must be migrated from the overloaded resource to an underutilized one without causing harm and disruption to the application workload. Objectives for this research work are:

1) To study and understand the task scheduling and load balancing approach on cloud.

2) To implement exising Dynamic Well-Organized Load Balancing algorithm (DWOLB) and proposed Hyrid method (Glowworm Swarm Optimization (GSO), and Cuckoo Search Algorithm (CSA)) on cloud environment.

3) To analyze the behavior of the proposed algorithm on the basis of following parameters:

a) Execution Time
b) Response Time
c) Energy Effieciency

C. *Metrics for Load Balancing In Clouds*

Various metrics will be considered in load balancing techniques in cloud computing are discussed below

1) Throughput is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.

2) Overhead Associated determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a load balancing technique can work efficiently.

3 Fault Tolerance is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique.

4) Response Time is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.

5) Resource Utilization is used to check the utilization of re-sources. It should be optimized for an efficient load balancing.

6) Scalability is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.

7) Performance is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

IV. RESULT AND DISCUSSION

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. *Implementation steps*

1) Setup server_config.xml
2) Initialize the Tomcat Server for project execution
3) Send Request for the execution of the project
4) Then select the specific algorithm for its execution
5) Simulation Results using Tomcat Server
6) CloudSim Results
7) Output Tables
8) Graphical Charts

The implementation steps are eloborated as below:

B. *Setup server_config.xml*

In this research work we are using five servers having their different IDs, names, IP address, speed and RAM. The number of jobs can be increased or wane as per the requirement. As by increasing the number of jobs the speed of server has to be increased so that it cannot affect the overall performance of the system. The parameters of job are id, request Type, arrival Time and length. The arrival time should be in increasing order.

C. *Initialize the Tomcat Server for project execution*

To run the project on server Tomcat Server should be initialized. The Tomcat server is available in different versions. In this Research work Tomcat v6.0 server is configured. The steps for configure Tomcat v6.0 Server on Eclipse interface is following as:

1) Right click on project and then click on run on server

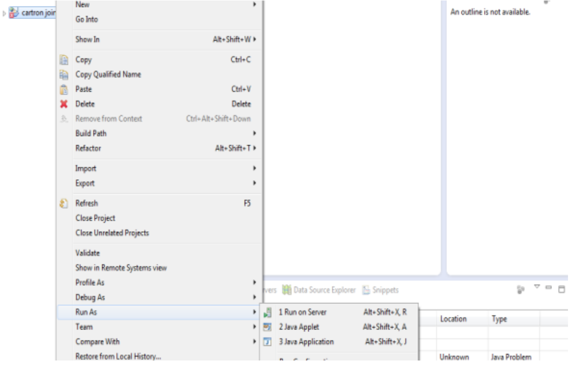
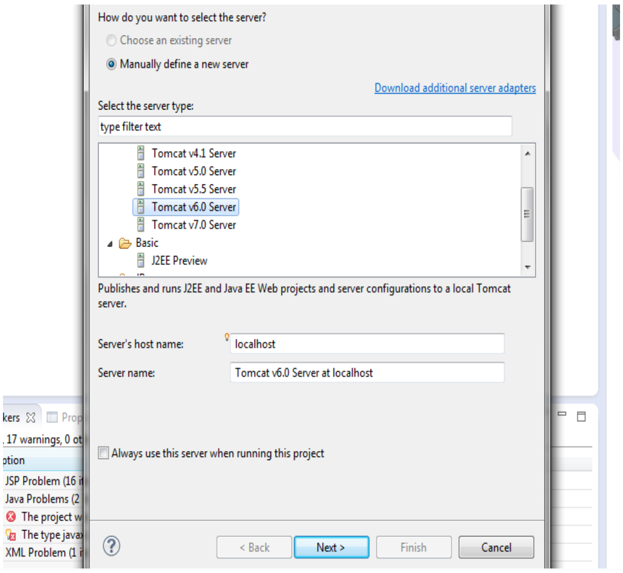


Fig.3: Project execution screen

Under apache select tomcat 6



How do you want to select the server?

○ Choose an existing server

● Manually define a new server

Select the server type:

type filter text

Tomcat v4.1 Server
Tomcat v5.0 Server
Tomcat v5.5 Server
Tomcat v6.0 Server
Tomcat v7.0 Server
Basic
J2EE Preview

Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

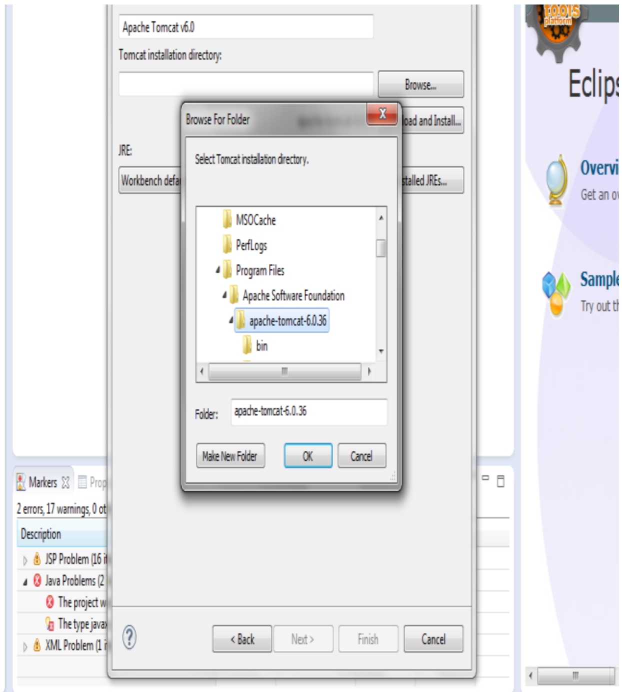
Server's host name: localhost

Server name: Tomcat v6.0 Server at localhost

□ Always use this server when running this project

Fig.4: Selection of tomcat v6.0 server

Select apache install directory



Apache Tomcat v6.0

Tomcat installation directory:

Browse For Folder

Select Tomcat installation directory.

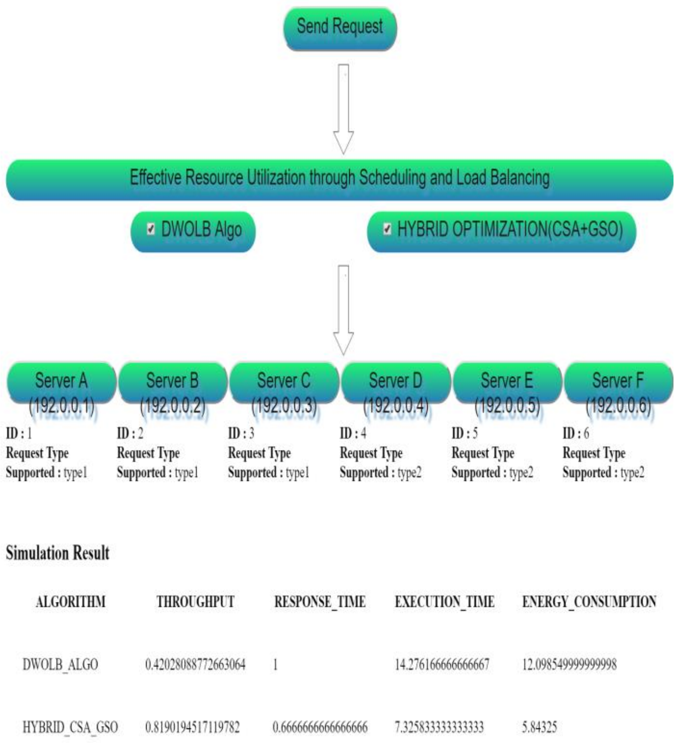
MSOCache
PerfLogs
Program Files
Apache Software Foundation
apache-tomcat-6.0.36
bin

Folder: apache-tomcat-6.0.36

Fig.5: Installation of apache-tomcat server from directory

D. Request for the execution of the project

After the program execution, URL is generated automaticallylikehttp://localhost:8080/Effectiveresourceutilization and fill this on crome browser to get the desired results. It is shown in figure 6.



ID : 1 Request Type Supported : type1

ID : 2 Request Type Supported : type1

ID : 3 Request Type Supported : type1

ID : 4 Request Type Supported : type2

ID : 5 Request Type Supported : type2

ID : 6 Request Type Supported : type2

Simulation Result

ALGORITHM	THROUGHPUT	RESPONSE_TIME	EXECUTION_TIME	ENERGY_CONSUMPTION
DWOLB_ALGO	0.42028088772663064	1	14.276166666666667	12.098549999999998
HYBRID_CSA_GSO	0.8190194517119782	0.6666666666666666	7.325833333333333	5.84325

Fig.6: execution of the project

E. Output Tables

The output tables shows the values for different parameters like throughput, response time, execution time and energy consumption when a particular technique is used for Multi-objective Load Balancing with the distinct number of jobs.

The different tables are drawn for two dynamic load balancing techniques and then for a particular number of jobs these are compared and the experimental results show that our proposed model gives the best results in terms of energy consumption, execution time, response time and throughput. The tables are shown as below:

1) Output Table of Dynamic Well-Organized Load Balancing algorithm:

This algorithm used for web services and systems called as Dynamic Well-Organized Load Balancing algorithm. It facilitates large scale load balancing with distributed dispatchers. In each dispatch firstly load balancing algorithm idles the processors for the availability and then does allotment of the task to processors in such a way that reduces the queue length at each server. This algorithm remove the load balancing work from critical path of request processing which helps in effective reduction of the system load. Join idle queue is the technique for load balancing which unveil the information about different parameters like Throughput, Response time, Execution time, Energy consumption with distinct numbers of jobs as shown in table 1.

Table.1: Simulation results Dynamic Well-Organized Load Balancing algorithm for different number of jobs

Sr. No.	No. of Jobs	Parameters Name			
		Throughput	Response Time	Execution Time	Energy Consumption
1.	5	0.4051	0.8571	12.45	10.56
2.	6	0.42	1.00	14.27	12.09
3.	7	0.423	1.2	16.53	13.97
4.	8	0.4829	1.3	16.62	14.51
5.	9	0.5444	1.4	16.87	15.21

2) Hybrid Optimization:

This load balancing algorithm works on the principle of grouping similar one’s and working on them group wise. The performance of the system is enhanced with high resources thereby increasing the parameter outcome using the algorithm. This algorithm is degraded with an increase in system diversity. A node initiates the process and selects another node called the matchmaker node from its neighbors, satisfying the criteria that it should be a different type than the former one.

Table.2: Simulation results for Hybrid Optimization for different number of jobs

Sr. No.	No. of Jobs	Parameters Name			
		Throughput	Response Time	Execution Time	Energy Consumption
1.	5	0.7559	0.7023	6.6143	5.3100
2.	6	0.819	0.6667	7.3258	5.8432
3.	7	0.7236	0.5892	6.3527	6.8211
4.	8	1.3330	0.5012	6.001	7.1027
5.	9	1.4521	0.5032	5.021	7.4521

3) Comparison Tables:

The Algorithms are implemented and compared on CloudSim tool for energy efficiency and load balancing. Table 3 depicts the result of different parameters for five jobs. During the comparison, Vector dot technique is counted as best model for producing the good results according to the user requirements.

Table.3: Different algorithms are compared for 5 Jobs

Algorithms	No. of Jobs	Parameters Name			
		Throughput	Response Time	Execution Time	Energy Consumption
Dynamic Well-Organized Load Balancing algorithm	#5	0.4051	0.8571	12.45	10.56
Hybrid Optimization	#5	0.7559	0.7023	6.6143	5.3100

Table.4: Different algorithms are compared for 06 Jobs

Algorithms	No. of Jobs	Parameters Name			
		Throughput	Response Time	Execution Time	Energy Consumption
Dynamic Well-Organized Load Balancing algorithm	#6	0.42	1.00	14.27	12.09
Hybrid Optimization	#6	0.819	0.6667	7.3258	5.8432

Table.6: Different algorithms are compared for 06 Jobs

Algorithms	No. of Jobs	Parameters Name			
		Throughput	Response Time	Execution Time	Energy Consumption
Load Balancing Decision Algorithm	#9	0.5444	1.4	16.87	15.21
Bat Optimization	#9	1.4521	0.5032	5.021	7.4521

All two algorithms are compared for energy efficiency and load balancing. The results show that Hybrid algorithm is better because which consumes less energy and all the tasks are executed in less time with no delay. It concluded that Hybrid technique is best in the energy efficient technique in cloud computing.

V. CONLUSIONS AND FUTURE SCOPE

In recent years, energy efficiency has emerged as one of the most important design requirements for modern computing systems, ranging from single servers to data centers and Clouds, as they continue to consume enormous amounts of electrical power. Apart from high operating costs incurred by computing resources, this leads to significant emissions of CO2 into the environment. For example, currently, IT infrastructures contribute about 2% of the total CO2 footprints. Unless energy-efficient techniques and algorithms to manage computing resources are developed, its contribution in the world's energy consumption and CO2 emissions is expected to rapidly grow. It has been shown that proper load balancing of computing resources can lead to a significant reduction of the energy consumption by a system, while still meeting the performance requirements. A relaxation of the performance constraints usually results in a further decrease of the energy consumption. Load balancing that is required to distribute the excess dynamic local workload evenly to all the nodes in the whole Cloud to achieve a high user satisfaction and resource utilization ratio.

In this paper we have proposed and implemented a Hybrid optimization on cloud environment using CloudSim Toolkit. And compared it with the Dynamic Well-Organized Load Balancing algorithm. The results show that proposed technique is much better than the existing load balancing methods in terms of Response time, Execution Time, and Throughput. We also concluded that Bat optimization technique consumes less energy than Central Load Balancer.

Cloud Computing is a vast concept and energy efficiency plays a very important role in case of Clouds. There is a huge scope of improvement in this area. We have implemented only two dynamic load balancing algorithms. But there are still other approaches that can be applied to balance the load and energy consumption in clouds. The performance of the given algorithms can also be increased by varying different parameters. We can also move our research work on any Private Cloud for the Security and further enhancements.

REFERENCES

[1] Atyaf Dhari and Khaldun I. Arif , "An Efficient Load Balancing Scheme for Cloud Computing" in the Indian Journal of Science and Technology, Vol 10(11), , March 201

[2] Amir Nahir , Ariel Orda, Danny Raz "Schedule First, Manage Later: Network-Aware Load Balancing" in the Proceedings IEEE INFOCOM, 2013

[3] H.Jamal, A.Nasir, K.Ruhana, K.Mahamud and A.M. Din, "Load Balancing Using Enhanced Ant Algorithm in Grid Computing", Proceedings of the Second International Conference on Computational Intelligence, Modelling and Simulation, pp. 160-165,2010.

[4] Isam Azawi Mohialdeen,2013, Comparative Study Of Scheduling Algorithms In Cloud Computing Environment .Journal of Computer Science, 9 (2): 252-263, 2013 ISSN 1549-3636 © 2013 Science Publications.

[5] Jianzhe Tai Juemin Zhang Jun Li Waleed Meleis Ningfang Mi "ARA: Adaptive Resource Allocation for Cloud" in the proceeding IEEE 2011

[6] Kumar Nishant, Pratik Sharma, Vishal Krishna, Nitin and Ravi Rastogi "Load Balancing of Nodes in Cloud Using Ant Colony Optimization" in the 14th International Conference on Modeling and Simulation, IEEE 2012

[7] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" in the proceeding IEEE 2012.

[8] Mohammad Goudarzi, Mehran Zamani, Abolfazl Toroghi Haghighat, "A fast hybrid multi-site computation offloading for mobile cloud computing" in the Journal of Network and Computer Applications, Elsevier 2017.

[9] Muhammad Baqer Mollah, Md. Abul Kalam Azad, Athanasios Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way Ahead" in the Journal of Network and Computer Applications, Elsevier 2017.

[10] Mohamed Abu Sharkh, Abdelkader Ouda, and Abdallah Shami, "AResource Scheduling Model for Cloud Computing Data centers" in the proceeding IEEE 2013.

[11] M.sudha, M.Monica:"Investigation on Efficient Management of Workflows in Cloud Computing Environment", International Journal of Computer Science and Engineering (IJCSE), Volume 02, Number 05, August 2010, Pages 1841- 1845.

[12] Parveen Patel ,Deepak Bansal, Lihua Yuan, Ashwin Murthy, Albert Greenberg " Ananta: Cloud Scale Load Balancing" in SIGCOMM, August12–16,2013, HongKong,China.

[13] Prof. Dr. Jayant. S. Umale, Miss. Priyanka A. Chaudhari, "Survey on Job Scheduling Algorithms of Cloud Computing" International Journal of Computer Science and Management Research , 2013.

[14] Parin.V.Patel ,Hitesh.D.Patel, Pinal.J.Patel, "A Survey Of Load Balancing In Cloud Computing" IJERT, Vol.1,Issue 9,November 2012.

[15] P.Salot, "A Survey of various Scheduling algorithm in cloud computing Environment" ,IJRET ,Volume:2,Issue:2,Feb 2012.

[16] Shiva Razzaghzadeh, Ahmad Habibizad Navin, Amir Masoud Rahmani, Mehdi Hosseinzadeh, " Probabilistic modeling to achieve load balancing in Expert Clouds" ElsevierB.V 2017.

[17] S.Maguluri, R.Srikant, and L.Ying, "Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters," IEEE INFOCOM 2012 Proceedings.pp.702- 710,25-30Mar,2012

[18] Vanitha, P. Marikkannu (2017). Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines. Elsevier Ltd.

[19] Saeed Javanmardi , Mohammad Shojafar, Danilo Amendola, Nicola Cordeschi "Hybrid Job scheduling Algorithm for Cloud computing Environment" published in SpringerVerlag Berlin Heidelberg 2014.

[20] Suriya Begum, Dr. Prashanth C.S.R, "Review of load balancing in cloud Computing". 10, Issue 1, No 2, January 2013.

[21] Rajesh Gorge Rajan and V.Jeyakrishnan "A Survey on Load Balancing in Cloud Computing Environments"Vol-2.Issue-12,December 2013.

[22] Tom Guérout, Samir Medjiah, Georges Da Costa, Thierry Monteil (2014)," Quality of service modeling for green scheduling in Clouds", In Elsevier.