International Journal of Computer Trends and Technology
ISSN: 2231-2803 / https://doi.org/10.14445/2231280453/IJCTT-V67I3P109
Volume 67 Issue 3, 33-40, March 2019

Original Article

Load Balancing of Heuristic Algorithms based on Performance Metrics in Cloud Computing

Shah Faisal[1], Jiong Yu[2]

[1]*School of software Engineering, Xinjiang University, Urumqi 830008, PR China.*
[2]*School of information science and Engineering, Xinjiang University, Urumqi 830046, PR China.*

Abstract *- Allocation of user tasks or scheduling of user requests to the cloud system is a critical job. Overloading and loading is a phenomenon that takes birth when user requests (tasks) are assigned to the cloud system. System failure, the increased execution time for the task, and more energy consumption occur because of cloud infrastructure overloading and under-loading. These load balancings are the main characteristic of task scheduling on virtual machines. Network load, memory load, and computational load are types of load in the cloud network. Detecting overloading and under-loading over the cloud system and then balancing load over the cloud system is the priority of the load balancing mechanism. "As pay-as-you need basis of the client," cloud computing providers insure their clients fulfill their requirements (request, demands of services). Efficient load balancing algorithms are necessary to minimize user requests' execution time and power consumption. Different types of load balancing algorithms were introduced for effective performance by researchers. This paper analyzed different performance parameters like Makespan and energy consumption in multiple load balancing algorithms. CloudSim simulators analyzed the heuristic-based algorithm's performance for the brief results.*

Keywords **-** *cloud computing, load balancing, Makespan, Virtual machine, Energy consumption.*

I. INTRODUCTION

Online dispersion of computing services and resources concerns recent technology known as cloud computing. Services providing systems configuration for end-users are unnecessary for cloud computing because users have to use paid utilities. The cloud system itself handles resource management and system configuration [1]. Dynamic load balancing makes hurdles in cloud computing. Efficiently assigning jobs to cloud nodes is the priority of cloud computing to ensure negligible heterogeneous restraints [2]. Redistribution of a load in

In the distributed and parallel system between the nodes, to improve the performance, a process is used known as load balancing. On the other hand, load balancing can be defined as " overcoming computational time and reasonable resource utilization while allocating different tasks between multiple processes, computers, memory, and other resources; load balancing plays an important role ."Sharing and utilizing resources in a sophisticated way is the core responsibility of load balancing. Load balancing focuses on two main directions: (a) to reduce the user's waiting time for his demanded request by putting a huge amount of simultaneous admittance. (b)by enhancing resource utilization at every node, it reduces overloaded nodes by sharing that load on multiple nodes [3].

Cloud computing provides virtual computing resources to its clients [4]. According to customers' need, in distributed data centers cloud use virtualization technology [5]. Virtualization techniques in the cloud ensure smooth customization of available software and applications [6]. Virtualization containers that are uncoupled from the implicit in physical resources virtualization enable application computation and data legion in the interior of the virtual container. These virtualization-based clouds render a huge substructure to value distant storage, computational, and network resources [7]. In this paper, we appraisal based on different heuristic load balancing algorithms. Different performance parameters are compared among these algorithms to find a reliable one suited to the current environment.

A. Characteristics of cloud computing

Cloud computing characteristics are listed in figure (1), which are further defined in detail below [8], [9].

- ***On-demand services***: Cloud computing provides services to its users for customer demands.
- ***Broad network services***: Ability are available on cloud computing on a network. With the help of

different mechanisms, capabilities can be accessed.

- ***Response pooling*:** Service providers provide their customers' services, pooled by different models. According to customers' needs, all resources are dynamically assigned and reassigned.
- ***Rapid elasticity*:** Quality of services is improved according to the user's needs.
- ***Measured services*:** Utilization of resources are monitored by services providers and users.

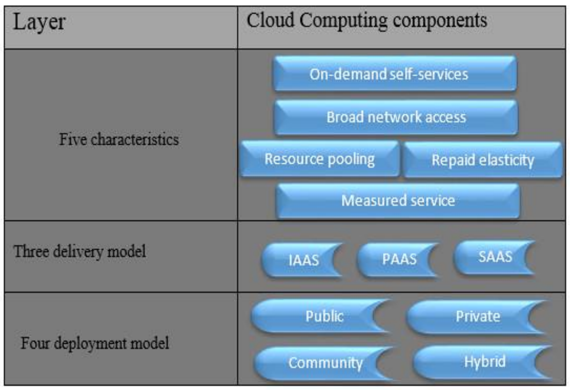

Fig. 1 Cloud computing model

II. CLOUD COMPUTING SERVICES:

Different servers provide different types of applications overcloud. Different services over the cloud are provided for customers, which are defined in detail below [10].

A. Software as a service (SaaS)

SaaS is responsible for providing distinct types of applications to users according to their needs provided by service providers. Applications are mainly run over a cloud foundation. Customers use different interfaces (web browsers) to access different applications. There is no concern about the user controlling the internal functionality of many applications [11], [12]. Customers who cannot develop their applications for their users take advantage of available applications from SaaS.

B. Platform as a service (PaaS)

Our internet Paas provides all resources to customers for application developments. There is no need for customers to download and install any software. Different programming languages and tools are provided to customers to build their applications. There is no need for customers to control operating systems, networks, storage, and servers. Deployed applications are controlled by customers themselves [13].

C. Infrastructure as a service (IaaS)

Users did not change or manage an inherent cloud infrastructure in IaaS. Users can control all deployed applications, storage, memory, and operating systems in infrastructure as a service. The user has limited permission to control networking components [14]. According to customers' demands, virtualization is used to delegate the system against user demands. For the running of their services, users deployed a software stake. Services as on-demand services, providers render networks. Directly customers use these services [15].

III. DEPLOYMENT MODEL LAYER IN CLOUD COMPUTING MODEL

A. public cloud

Users access cloud infrastructure in the public cloud, or the General public or huge organizations access cloud infrastructure in the public cloud. There is no restriction for the user to access any cloud resources in the public cloud.

B. Private Cloud

A single organization or company can use cloud infrastructure in the private cloud. Here third party or organization can manage cloud infrastructures. The general public is restricted from direct access to cloud infrastructures in the private cloud.

C. Community cloud

Many organizations can share cloud infrastructures in the community cloud. A specific shared concerns community supported by community cloud (For example, policy, security, requirements). Third-party or organization maybe manage the community cloud.

D. Hybrid cloud

A combination of two clouds is called a hybrid cloud. It may be community-private, public-private, or public-community clouds. A hybrid cloud may be managed by an organization or third party[16].

IV. ELEMENTS OF CLOUD

The center where the main host exists, incoming tasks, data centers brokers, networks, connections, and virtual machines are all based on cloud architecture. There is some main key in the cloud [22].

A. Brokers

Brokers act as an agent between the cloud and customers. The main responsibility of brokers is to receive service requests from customers and submit that requests to the cloud.

B. Virtual machine (VM)

To process many tasks on one device, multiple virtual machines are created to meet the requirements of processing many tasks on a single physical machine.

C. Datacenter

Datacenters encapsulate storage, memory, core, and capacity and communicate between them to control their components. Each data centers have its pre-defined policies and strategies.

D. Service allocators

Interfaces between users and cloud infrastructures are known as service allocators.

E. Physical machines

A physical machine is the main server in the data center to fulfill processing demands.

F. Cloudlet

The amount of storage and the number of tasks needed to process these tasks is cloudlet.

G. Memory Provisioner

Memory Provisioners are a physical memory allocation policy for virtual machines.

H. Cloud coordinators

Cloud coordinators maintain load balancing internally in data centers and communicate with users and multiple data centers.

V. LOAD BALANCING

Dispersal of large among small processing nodes to enhance system performance is called load balancing. With the help of load balancing algorithms in cloud computing, dynamic workloads are distributed between multiple nodes [17].

VI. TAXONOMYOF LOAD BALANCING ALGORITHMS

Different load balancing algorithms are classified in figure 2. In the below figure, there are two main categories of load balancing algorithms which is Dynamic load balancing (DLB) and static load balancing (SLB) [18].

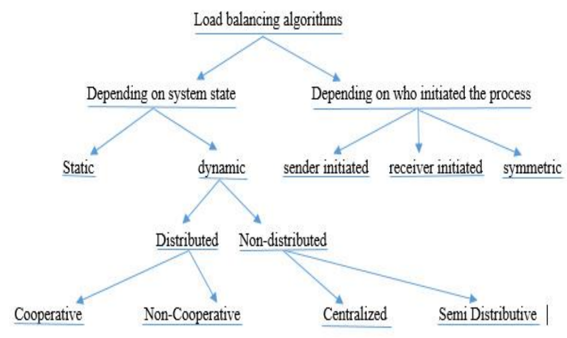

Fig. 2 Taxonomy of load balancing algorithms

A. Static load balancing algorithms

Distribution of traffic equally among all servers are static load balancing algorithms. In the design of the system, static load balancing algorithms are defined.

B. Dynamic load balancing algorithms

Dynamic load balancing algorithms mainly work on the run time of system states during load mobility. For cloud computing, dynamic load balancing is a reliable approach.

Dynamic load balancing is divided into two types, Distributed and non-distributed approaches.

1. Distributed load balancing approach

Every node creates its load vector independently in distributed approach. Vector collects other nodes' load information. With the help of local node vectors, decisions are made locally. Distributed approaches are the best approach for cloud computing.

2. Non-distributed load balancing approach

Unique nodes (single nodes) are responsible for the distributions of load among multiple nodes in a non-distributed approach.

VI. LOAD BALANCING METRICS

A. Throughput

Several tasks executed in specific time intervals are known as the system's throughput. The performance of a system can be calculated from its throughput.

B. Scalability

The algorithm's flexibility is the distribution of a load's infinite range of nodes in a system. For a good system, this metric must be improved.

C. Migration time

Mobility of load between multiple nodes is known as migration time. For better performance of the system, migration time must be minimized.

D. Response time

The time needed for an algorithm to respond to a task in the system is called response time. For better system response performance, the algorithm's time should tell me the minimum.

E. Fault tolerance

quick recovery of the system from any failure is the responsibility of a good fault tolerance algorithm.

VIII. CLOUDSIM TOOLKIT

A. CloudSim simulation toolkit characteristics

Multiple grid simulators are developed, such as GangSim, GridSim, and SimGrid. But these simulators are used in distributed environments for simulations and modeling of grid applications. No one in these simulation toolkits fully answered those questions rising from the cloud computing environment. Hence it should be the core responsibility of the simulation and modeling toolkit to fulfill all necessities of service suppliers and users. So to fulfill all the necessities of service suppliers and users for simulations and modeling of various eventualities, CloudSim simulation, and modeling toolkit are developed. CloudSim Simulator has some features that resemble the GridSim simulation toolkit. For java virtual machines and a data center with a single physical computing node, CloudSim supports the modeling of huge scale cloud computing infrastructure. CloudSim toolkit additionally offers services to change between time-shared and space-shared for virtualized services[23].

IX. WORK STYLE OF CLOUDS

In this paper, we will show a basic workflow style of the CloudSim toolkit in figure 3. As we know that different users have different types of demands (Tasks). For example, there is *n* number of users, such is

$User_1, user_2, user_3, user_4, user_5, \ldots\ldots\ldots\ldots, user_n$

m numbers of the independent task, such is

$T_1, T_2, T_3, T_4, T_5, \ldots\ldots\ldots, T_m$

q numbers of virtual machines, such are

$VM_1, VM_2, VM_3, VM_4, VM_5, \ldots\ldots\ldots, VM_q$

And *k* numbers of data centers

$DC_1, DC_2, DC_3, DC_4, DC_5, \ldots\ldots\ldots, DC_k$

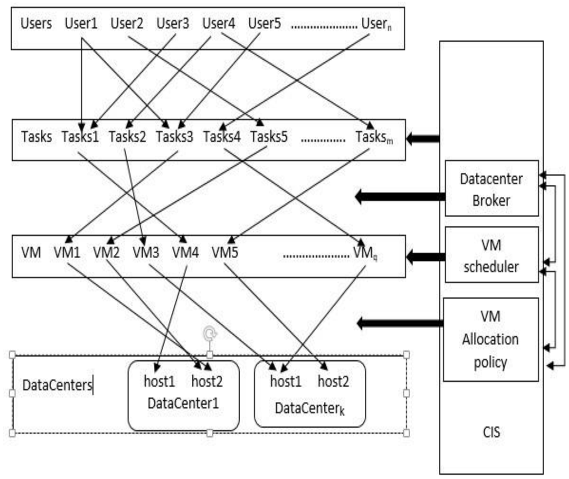

Fig. 3 CloudSim Workflow

In the above figure, cloud information service (CIS) provides a match-making service at the database level. CIS maps cloud providers and users in a suitable way.

X. MIN-MIN ALGORITHM

The Min-Min algorithmic program begins from a list of unscheduled tasks, so the least bit machines and minimum execution time for all tasks are determined. Then at the resultant machine, a task with a minimum completion time is chosen and scheduled [25]. Then from the tasks list, scheduled tasks are unfastened, and until all unscheduled tasks are not successfully scheduled, the process is repeated again and again, which is shown in the pseudo-code of the min-min algorithm below[26],[27],[28].

(1) Pseudo-code for MIN-MIN algorithm

```
for all task ti in MT
    for all VM mj
        CTij = ETij + rj   // M for all unscheduled task
                           // C for task completion time
                           // E for task execution time
                           // r for task i ready time on virtual machine j
do till all task in MT are not successfully mapped
    for each unscheduled task ti in MT
        Find the minimum completion time CTij and virtual machine that obtain it
    end for
        Find the task tq with earliest completion time CTij
        Assign task tq to recourse mj that's gives less completion time
        Delete task tq from all unscheduled task MT
        Update rj that gives minimum completion time
end do
```

XI. MAX-MIN ALGORITHM

Like the min-min algorithmic program shaping completion time for each task at every machine, in the case of the max-min algorithm, those tasks with maximum completion time are scheduled for incompatible machines. Until all tasks are not successfully scheduled, the process is redone [25]. In the min-min algorithm, there is an expectation that a smaller Makespan will be obtained if the execution time is fast and more tasks will be scheduled on the machine. In the case of a few smaller and larger tasks, the mix-min algorithm is used. Pseudo-code of the max-min algorithm is shown below [31].

(2) Pseudo-code for MAX-MIN algorithm

```
for all task ti in MT
    for all VM mj
        CTij = ETij + rj   // M for all unscheduled task
                           // C for task completion time
                           // E for task execution time
                           // r for task i ready time on virtual machine j
do till all task in MT are not successfully mapped
    for each unscheduled task ti in MT
        Find the maximum completion time CTij and virtual machine that obtain it
    end for
        Find the task tq with maximum completion time CTij
        Assign task tq to recourse mj that's gives maximum completion time
        Delete task tq from all unscheduled task MT
        Update rj that gives maximum completion time
end do
```

XII. MINIMUM EXECUTION TIME (MET) ALGORITHM

Without pondering resource availableness and the idea of the best predictable completion time of tasks, the MET algorithmic rule assigns tasks to the virtual machine. On the premise of the minimum execution time of virtual machines or resources, MET has a core plan to assign a task to that resource or virtual machine. Because the assignment is independent of the availability, the result of high load imbalance [29], [30] is shown in the pseudo-code of the MET algorithm below.

(3) Pseudo-code for MET algorithm

```
for all task ti in MT
    for all VM mj
        CTij = ETij + rj   // M for all unscheduled task
                           // C for task completion time
                           // E for task execution time
                           // r for task i ready time on virtual machine j
do till all task in MT are not successfully mapped
    for each unscheduled task ti in MT
        Find the best completion time CTij and virtual machine that obtain it
    end for
        Find the task tq with best completion time CTij
        Assign task tq to recourse mj that's gives best completion time
        Delete task tq from all unscheduled task MT
        Update rj that gives best completion time
end do
```

XIII. RELATED WORK

A.M.Nikai et al. [19] used LB for internet cloud services in cloud servers' environments. Their main description was to avoid overloading remote servers by limiting the redirection rate using a protocol. That protocol was supported by middleware. In their work, they find avoiding overloading of servers using redirecting the request to the nearest servers to reduce server response time. The mean response time is 31% smaller than SL (smallest latency) and 29% smaller than RR (Round Robin).

Y.Lua et al. [20] present join-idle-queue techniques in a cloud data centers environment. The main description of their work is the assignment of idle processors to all or any dispatchers to ensure the supply of idle processors at every dispatcher. Then assignments of the task to each processor to reduce the queue size of each task at each processor. Their main finding is to reduce system load effectively and at the time of job arrival to ensure no communication overhead. They also work on the reduction of actual response time.

J.Hu et al. [21] presented a scheduling strategy for load balancing of virtual machine resources exploitation cloud computing atmosphere. The main description of their work was using genetic algorithm, historical data, and the current state for the reduction of dynamic migration and achievement of best load balancing. Their finding was the solution to high migration costs and load balancing problems.

SaeedParsa et al. [24] proposed a new algorithm known as RASA for task scheduling. RASA algorithm has advantages over the Max-Min algorithm and Min-Min algorithm. This algorithmic program has the power to use the Min-Min algorithm strategy to execute tiny tasks before the execution of huge tasks and to avoid delay in the execution of large task exploitation Max-Min algorithm strategy, between the execution of the small task and large tasks RASA support dissension.

XIV. THE FIRST SCENARIO IS BASED ON THE NUMBER OF TASKS

This Scenario will analyze the performance metrics of some heuristics algorithms such as Max-Min, MET, and Min-Min through a simulation toolkit CloudSim 3.0.3. Intel Core i7 4^{th} generation processor, 8 GB RAM, Window 8 operating system, and 3.4 GHz CPU are used in table (1). Here we will analyze these three heuristic algorithms' performance metrics like Makespan and energy consumption.

Table 1. system specifications for the first Scenario

CloudSim	3.0.3
Version of system	Intel core i7
System generation	4th
System processor	3.4 GHz
System RAM	8 GB
Operating system	Microsoft Windows 8
Numbers of tasks	500
Numbers of VMs	20 - 200
intervals	20

In the first scenario, the total numbers of tasks are 500. With the intervals of 20, the total number of virtual machines varies from 20 to 200, as shown in table (1). The resultant report of comparison is shown in Figures 4 and 5. Figures 4 and 5 show Makespan and energy consumption based on tasks of these three heuristic algorithms Max-Min, MET, and Min-Min. From these two graphs in Figures 4 and 5, it is clear that the Makespan and energy consumption of the MET algorithm is minimum among these three algorithms.

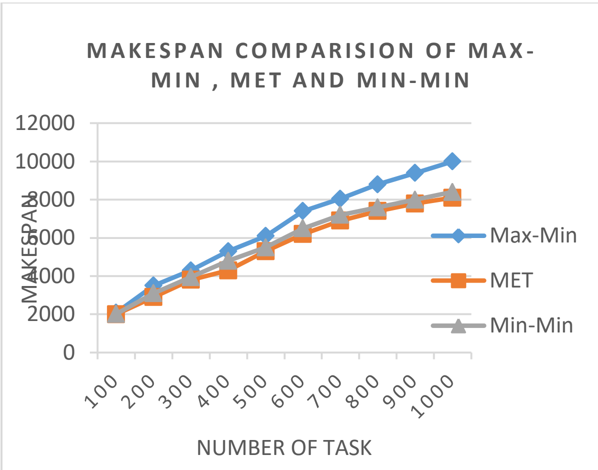

Fig. 4 Makespan comparison for Scenario first

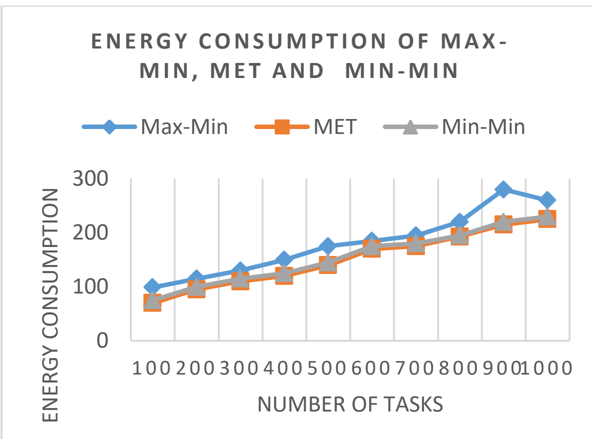

Fig. 5 Energy consumption comparison for Scenario first

The second scenario is based on the number of Virtual machines

This Scenario will analyze the performance metrics of some heuristics algorithms such as Max-Min, MET, and Min-Min through a simulation toolkit CloudSim 3.0.3. Intel Core i7 4th generation processor, 8 GB RAM, Window 8 operating system, and 3.4 GHz CPU are used in table (2). Here we will analyze these three heuristic algorithms' performance metrics such as Makespan and energy consumption.

Table 2. system specifications for the second Scenario

CloudSim	3.0.3
Version of system	Intel core i7
System generation	4th
System processor	3.4 GHz
System RAM	8 GB
Operating system	Microsoft Windows 8
Numbers of tasks	100 - 1000
Numbers of VMs	100
intervals	100

In the second scenario, the total numbers of Virtual machines are 100. The intervals of 100total variety of input tasks vary from 100 to 1000, as shown in table (2). The resultant report of comparison is shown in Figures 6 and 7. Figures 4 and 5 show Makespan and energy consumption based on tasks of these three heuristic algorithms Max-Min, MET, and Min-Min. From these two graphs in Figures 6 and 7, it is clear that the Makespan and energy consumption of the MET algorithm is minimum among these three algorithms.

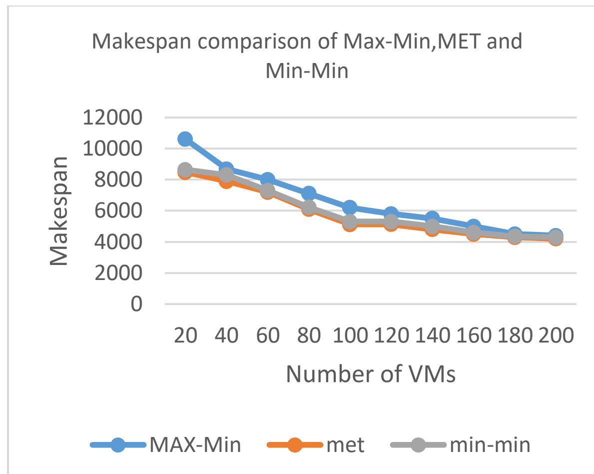

Fig. 6 Makespan comparison for scenario second

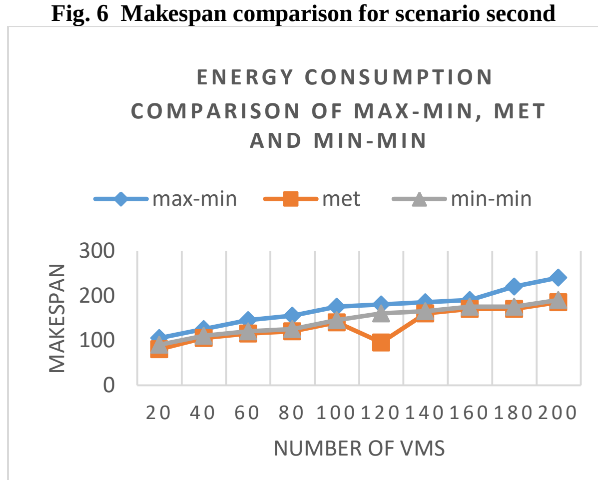

Fig. 7 Energy consumption comparison for scenario second

XV. CONCLUSION

This paper analyzed different heuristic algorithms such as Max-Min, MET, and Min-Min in a cloud computing environment based on different performance parameters like Makespan and energy consumption using the simulation toolkit CloudSim. A comparative study of these heuristic algorithms in a cloud environment shows that the MET algorithm gives good results among these algorithms. Future work includes enhancing these heuristic algorithms in a cloud computing environment based on their performance parameters and practical implementation.

REFERENCES

[1] Wang, R., and Rao, B. (2015) 'Research on the Cloud Computing Load Balance Degree of Priority Scheduling Algorithm based on Convex Optimization Theory,' pp. 156–160.

[2] Chaisiri, S., Lee, B. S., and Niyato, D. (2012) 'Optimization of resource provisioning cost in cloud computing, IEEE Transactions on Services Computing, 5(2), pp. 164–177.

[3] Dam, S., Mandal, G., Dasgupta, K., and Dutta, P. (2015, February) 'Genetic algorithm and gravitational emulation based hybrid load balancing strategy in cloud computing,' Third IEEE International Conference on Computer, Communication, Control and Information Technology (C3IT), pp. 1–7.

[4] Wang, R., and Rao, B. (2015) 'Research on the Cloud Computing Load Balance Degree of Priority Scheduling Algorithm based on Convex Optimization Theory,' pp. 156–160.

[5] Jebalia, M., Ben Letafa, A., Hamdi, M., and Tabbane, S. (2015) 'An overview on coalitional game-theoretic approaches for resource allocation in cloud computing architectures, International Journal of Cloud Computing, 22, 4(1), pp. 63-77.

[6] Singh, A., Juneja, D., and Malhotra, M. (2015) 'Autonomous agent-based load balancing algorithm in cloud computing,' Procedia Computer Science, 45, pp. 832–841.

[7] Patni, J. C., Aswal, M. S., Agarwal, A., and Rastogi, P. (2015) 'A Dynamic and Optimal Approach of Load Balancing in Heterogeneous Grid Computing Environment,' In Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI, Springer, Vol. 2, pp. 447–455.

[8] http://blog.nexright.com/?cat=6"Cloud Computing « Nexright Blog."

[9] A.Sidhu, S. Kinder, "Analysis of load balancing techniques in cloud computing," INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY 4 (2) (2013) pages737–741.

[10] Chaudhari, Anand, and Kapadia, Anushka, " Load Balancing Algorithm for Azure Virtualization with Specialized VM," 2013, algorithms, vol 1, page 2, Chaudhari.

[11] http://letslearncloud.wordpress.com/"Cloud Computing | Learn Cloud and its tips."

[12] Sahu, Yatendra and Pateriya, RK, "Cloud Computing Overview with Load Balancing Techniques," International Journal of Computer Applications, 2013, vol. 65, Sahu2013.

[13] Liang, B., Yanli, H., Songyang, L., Weiming, Z., "Task Scheduling with Load Balancing using Multiple Ant Colonies Optimization in Grid Computing" in 2010 Sixth International Conference on Natural Computation (ICNC 2010), DIO: 10.1109/ICNC.2010.5582599, pp.2715-2719, 2010.

[14] U.Bhoi, P. N. Ramanuj, " Enhanced max-min task scheduling algorithm in cloud computing," International Journal of Application or Innovation in Engineering and Management (IJAIEM), ISSN,2013, pages 2319—4847.

[15] S.Kabiraj, V. Topka, and R. C. Walke, "Going Green: A Holistic Approach to Transform Business," International Journal of Managing Information Technology (IJMIT), Vol. 2, No. 3, August 2010, pages 22-31.

[16] S.Ray and A.D. Sarkar (2012, October). Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA). 2(5), pp. 1-13

[17] Chaudhari, Anand, and Kapadia, Anushka, " Load Balancing Algorithm for Azure Virtualization with Specialized VM," 2013, algorithms, vol 1, page 2, Chaudhari.

[18] Rajwinder Kaur1 and Pawan Luthra2 "Load Balancing in Cloud Computing," 2014, vol 5, page 8.

[19] A.M.Nakai, E. Madeira, L. E. Buzato, "Load Balancing for Internet Cloud Services Using Limited Redirection Rates," 5th IEEE Latin-American Symposium on Dependable Computing (LADC), pp. 156-165, 2011.

[20] Y.Lua, Q. Xia, G. Kliotb, A. Geller, J. R. Larus, A. Greenberg, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services," An International Journal on Performance evaluation, In Press, Accepted Manuscript, Available online 3 August 2011.

[21] J.Hu, J. Gu, G. Sun, T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment," Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), pp. 89-96, 2010.

[22] Manasa, HB & Basu, A., 2013. Energy-Aware Resource Allocation in Cloud Datacenter. , International Journal of Engineering and Advanced Technology (IJEAT), 2(5), pp.277–281.

[23] Calheiros, R.N., Ranjan, R., De Rose, C.A.F., Buyya, R., " CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services" in Technical Report, GRIDS-TR-2009-1, Grid Computing, and Distributed Systems Laboratory, The University of Melbourne, Australia, 2009.

[24] SaeedParsa and Reza Entezari-Maleki, "RASA: A New Grid Task Scheduling Algorithm," International Journal of Digital Content Technology and its Applications, Vol. 3, pp. 91-99, 2009

[25] Aisi H, Bazan C, Vanderpooten D. Complexity of the min-max and min-max regret assignment problems. Operations research letters. 2005; 33(6):634–40.

[26] Braun TD, Siegel HJ, Beck N, Bo¨lo¨ni LL, Maheswaran M, Reuther AI, et al. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed computing. 2001; 61(6):810–37

[27] Ibarra OH, Kim CE. Heuristic algorithms for independent scheduling tasks on nonidentical processors. Journal of the ACM (JACM). 1977; 24(2):280–9.

[28] Wu M-Y, Shu W, Zhang H. Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems. hcw; 2000: IEEE

[29] He X, Sun X, Von Laszewski G. QoS guided min-min heuristic for grid task scheduling. Journal of Computer Science and Technology. 2003; 18(4):442–51.

[30] Maheswaran M, Ali S, Siegel HJ, Hensgen D, Freund RF. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. Journal of parallel and distributed computing. 1999; 59 (2):107–31

[31] LK.Wiseling Lazarus, Dr.P.Venkadesh, J.P.Jayan "Joint Power Allocation and Beam Forming for Energy-Efficient Multi-Way Multi-Antenna Relay Networks ."International Journal of Computer Trends and Technology (IJCTT) V60(2):111-114 June 2018.