Proposing a Load Balancing Algorithm For Cloud Computing Applications

Bayan A. Al Amal Murayki Alruwaili, Mamoona Humayun, NZ Jhanjhi
1College of Computer and Information Sciences, Jouf University, Sakakah, Saudi Arabia
2Department of Information systems, College of Computer and Information Sciences, Jouf University, Sakakah, Saudi Arabia
3School of Computer Science and Engineering (SCE), Taylor’s University, Malaysia
Corresponding author: Mamoona Humayun (mahumayun@ju.edu.sa)

Abstract: At the current time, one of the fields that are quickly developing in computing is clouding computing (CC). There are vital services needed by individuals as well as organizations that are offered in CC, namely IaaS, SaaS, and PaaS. Through the creation of the cloud, the development of the application has become easier, as well as providing services to end-users through virtualization over the internet. In CC, we have cloud service providers who deal with large computing structures completely defined on usage as they offer their services in a dependable mode. However, CC is facing the issue of load balancing, which impacts the performance and makes it weak if we overload the system. Developing an effective load balancing algorithm while utilizing CC efficiently is one of the providers’ ultimate goal. Virtualization and scalability are other dynamic techniques which entail developing VMs in CC. Data traffic and Web services provision is growing each day; thus, load balancing is a major challenge in the field of CC and is increasing the importance of task scheduling as well. At this point in time, evolvimento in CC necessitates the greater need for infrastructure and resource development. The load balancing policy guarantees the efficient usage of the resource through the provision of services to the cloud subscribers. Load balancing also prioritizes subscribers by using an appropriate schedule. As a contribution in research, this paper proposes an algorithm for load balancing in CC. Proposed algorithm enhance the performance of Cloud Computing applications

Keywords—Cloud Computing (CC); Load balancing; Machine Learning, Algorithm; Task Scheduling.

1. INTRODUCTION

According to [1], as a result of the recent achievement of the internet in recent years, resources of computing have become available universally, which has enabled CC development. In CC, the supplier is required to have the infrastructure as well as the service provider. According to [2] [3], Infrastructure suppliers arrange the cloud platforms to lease the resources on need-basis for utilization. From infrastructure providers, the service providers take the services to support end-users. Enormous companies like Amazon, Google, and Microsoft have been lured by CC, and they are considered as the biggest companies in the IT world. CC is a facility where infrastructure, platform, and software’s are given out in reference to a client’s need. The rudimentary idea of CC in Virtual Machines is to provide resources as services which are in demand. Passing on an efficient Virtual Machine is being done on the provisions of the load balancing algorithms. The load balancing algorithm plays a noteworthy role in selecting which VM will be assigned to a user on demand. There is also a possibility that there could be a number of requests at a given time. [4] Due to this, some of the requests are retained in the queue, necessitating the likelihood to forward the request to supplementary service providers. Hence provision of the load balancing algorithm, users are capable of deciding whether to remain in the queue or get services from the other providers[5].

Before proceeding further, table 1 list down the acronyms and abbreviations for better understanding of the paper.
Table 1: Table of acronyms and abbreviations

| Acronym | Abbreviations |
|---------|--------------|
| CC      | Cloud Computing |
| VM      | Virtual Machine |
| VMs     | Virtual Machines |
| IT      | Information Technology |
| IaaS    | Platform as a Service |
| IP      | Internet Protocol |
| PaaS    | Platform as a Service |
| SaaS    | Software as a service |
| SHC     | Stochastic Hill Climbing |
| PSO     | Particle Swarm Optimization |
| ML      | Machine Learning |
| LBA     | Load Balance Algorithm |
| RB      | Round robin |
| DCC     | Data Center controller |
| RRLB    | Round robin load balancer |
| ESCE    | Equally spread current execution |
| LB      | Load Balancing |

2. CLOUD COMPUTING OVERVIEW

In CC, the major technique used is that of the internet and a central remote server for applications and data management. Further, in CC, users and entities use applications without installing personal files or without access to a computer as long as the internet access. This technique is ineffective because it centralizes processing, storage, bandwidth, and memory. CC, by definition, is a prototype of a computing network in which a given application or a program is executed on a linked server. In the traditional client-server model, clients have to link with a server to execute any given job. Variance in CC occurs since the computing process can function on a numerous number of linked computers using data virtualization. [6] Virtualization is where several physical servers are configured, subdivided into some number of unconnected virtual servers which work autonomously, but they appear to users as a single machine. Due to the evolving nature of computing resources, there is a need to develop a user friendly application on devices, allowing resource pooling, creating service on demand and reviewing capability, and giving fast elasticity. This technique is done by using distributed computers concentrating on a wide scope of users who have access to distributed virtual hardware and software infrastructures on the internet. This is inclusive of web software, networking, web services, and distributed computing virtualization. The main concepts of CC are having the user interest concentrate towards distributed, parallel, and virtualization of computing systems. Through virtualization, CC is capable of having an enormous client base with innumerable physical computing infrastructure. Security is a major concern in the world of developing CC. [7][8]

CC has four cloud deployment models namely, Public cloud – it exists for public use and as an alternative to the giant industry. In this type of cloud, the user has no control access and visibility computing infrastructure the user hosted on. Private cloud- maintenance of this cloud is done by an organization or third party, which makes them more secure and more costly as compared to the public clouds. Single organizations use externally hosted private clouds of a third party focusing on cloud infrastructure. Additionally, On-premise private clouds are cost demanding compared to externally hosted private clouds. Hybrid cloud- this type of cloud integrates several clouds where the clouds are used to retain their specific identities and are bonded together as a single unit. Cloud bursting is where organization utilize their own computing infrastructure to access the cloud in order to get high load requests. Community cloud is organized to support a mutual function or objective. An example is a cloud for one organization or several organizations, sharing the same concerns as their policies, mission, security, and regulatory compliance requirements [9]. These four deployment models are presented in four parts of Figure 1
3. CLASSIFICATION OF CLOUDS BASED ON THE SERVICE PROVIDER

The clouds are mainly divided into three forms based on the various services offered by them. These three modes of cloud services include: IaaS, which entails the provisions of hardware associated services through utilization of CC principles. It offers virtual-machine, virtual storage, file or object storage, virtual infrastructure, IP addresses, disk image library, raw block storage, virtual local area networks load balancer, firewalls, and software. Providers supplies them upon demand from the large tarns installed in the data center. PaaS models involves the cloud provider providing a computing platform, which is inclusive of database, web server, operating system, and programming language execution atmosphere. Application developers operate their software solutions on the cloud platform without necessitating the absorption of software and hardware layers or the cost of buying and handling them. SaaS—entails providing the entire software in the cloud. Users on pay per use basis are granted access to the software application, which is hosted by the cloud providers.

4. LOAD BALANCING

Load balancing is a technology of distributing loads to some resources in a given system. Thus, load-distributing in cloud-based architecture is very crucial in that every resource needs to do the same quantity of work at any given time. The most important procedure is to give some methods so as there is a balance between requests and solutions for any given request. Online traffic in cloud load balancers is maintained automatically through distributing of loads amongst multiple servers and resources. This had an advantage in that it increases output while avoiding overload as well as decreasing response time. In paper [12], the load balancing algorithm for optimization of CC application is discussed, and a sum up is given inform of an overview. Different algorithms are designed for different purpose e.g. some algorithms are intended to attain maximum throughput; others intend to have the least response time, while others intend to have a maximum usage of resources while others target to have a trade-off in all the system measurements. Figure 2 illustrates a context under which some load balancing algorithms operate in an atmosphere of CC.
5. A COMPARATIVE STUDY

Below we provide the existing commonly used load balancing algorithm for detailed analysis and to provide state-of-the-art picture

5.1 Round robin (RR) load balancer algorithm

RR divides time into several slices and each given a specific time quantum within which it performs its operations. DCC in this algorithm assigns requests on a rotating basis to a list of VM’s. A random VM is selected from the group and assigned the first request by the DCC, which subsequently assigns the other requests in a circular manner. The Weighted Round Robin Allocation is a better RRLB which assigns weight to all available VMs so that if a VM can handle twice as much load as other VMs, it is given a weight of 2, and it will be allocated two requests to the powerful VM. RRA is very simple, however, it takes longer average waiting time, also it has higher context switches, higher turnaround time, and low throughput [13]. Figure 3 shows the RR balancer algorithm working

5.2 Equally spread current execution (ESCE)
ESCE maintains the index of Virtual machines and the number of requests currently allocated to the Virtual machines. At the initial stage, all Virtual machines have 0 allocations. As a request arrives from the DCC to be allocated to a new virtual machine, it is parsed with an index and the least loaded virtual machine identified. If it’s more than one virtual machine, one to be first identified parse the load [12]. ESCE returns the ID of the virtual machine to the DCC upon which the virtual machine identified by that ID is given the request, and the LB is notified of the new allocation, which in turn updates the allocation table resulting in an incrementation of allocation count for that specific virtual machine. Upon completion of the processing of the request, the DCC receives the response cloudlet and notifies the LB for virtual machine de-allocation triggering decrement of the allocation count for the virtual machine [13]. Figure 4 elaborate ESCE

5.3 Active monitoring load balancer

Manages the number of requests handed over to every VM and the information about them. Upon the arrival of a new request, it determines the least load VM and assigns it that request. The first determined VM is chosen when there more than least loaded. The active VM id is returned to the data center controller, and the data center controller assigns a request using that id. Figure 5 elaborate active monitoring load balancer

5.4 Throttled Load Balancer
This type of load balancer allows clients to initiate a request to the load balancer to check the appropriate Virtual Machine, which will complete the operations of the load easily upon provision by the client. Figure 6 shows the working of Throttled load balancer.

![Throttled Load Balancer](image)

The above discussion shows that there exist various load balancing algorithms but still the challenge of load balancing in CC persist. To contribute in the research, this paper aimed to enhance the performance of load balancing in cloud computing by proposing a priority-based load balancing algorithm. The first section of the paper includes the introduction and the common load balancing algorithms. The second section discusses the related work with a comparative summary of the discussed papers. The proposed algorithm is elaborated in the third section with the implementation of a priority-based load balancing algorithm. Next, the paper presents the discussion and conclusion followed by future work at the end.

6. RELATED WORK
This section provide a brief discussion on some Load Balancing methods employed in cloud architectures. Stochastic Hill Climbing this paper introduced a computing technique to load balancing whereby requests of incoming jobs to the VMs utilizes a local optimization technique of Stochastic Hill climbing. It has a family of two significant processes for resolving an optimization issue. One of the families of processes known as complete methods determines if there is a binding allocation of values to the variables or if there exist no such allocation. The advantage of these techniques is good performance as it ensures an optimal and accurate response to all inputs. Unfortunately, more time is a need in the worst instance, and this is a disadvantage in the CC domain. A Hill Climbing Algorithm Stochastic Hill Climbing (SHC) version is one of the inadequate procedures that can be used to solve optimization issues [13]. Particle Swarm Optimization (PSO) Algorithm is a universal search-based optimization technique which is self-adaptive, and it was proposed by Kennedy and Eberhart. The PSO has its main concentration capped at reducing the cost of computation of the workflow of an application. Additionally, PSO distributes tasks to existing resources [17]. Genetic Algorithms are made up of selection, genetic operation, and replacement. The main advantage of this scheme is that it manages huge exploration space which suits complex objective functions while avoiding ensnaring into local optimum solution. Genetic Algorithms are randomized though they are not random simple walks [18]. Honeybee Foraging Behavior Algorithm is search procedures with the aim of coming up with an optimal resolution to a problem mainly to satisfy more objective functions subjected to a cluster of restraints. Scrutiny of insects and animals as social occasioned several computational models of flight of intelligence. There are several load balancing methods, and the honeybee is one of the approaches which is nature inspired algorithms which have capability of self-organizing. This shows better performance with the increase in diversity. However, throughput is not increased as of performance [19].

We have several types of load balancing approaches in CC namely: static, geographical distribution, and dynamic. **Static load balancing algorithms;** in this approach, execution is not based on the present state of the system because executing processes are determined at the starting point. The main objective of this approach is to reduce the execution time of a synchronous program while aiming to decrease communication delays. Round Robin algorithm, Randomized algorithm, and Threshold algorithm form some instances of the static load balancing algorithms [13][20].
Geographical distribution: in this distribution, the way nodes are distributed, is crucial to the performance of CC systems, in particular for bulky applications such as Twitter and Facebook. A system that has well-distributed nodes has a better fault tolerance while it is easy to manage system effectiveness [21].

Dynamic load balancing algorithm, in this approach, the verdicts of load balancing are dependent on the present state of the system in the essence that no preceding information is required for execution. One of the greatest advantages of dynamic load balancing is that if one node fails, it does not cause the failure of the system, but it has a ripple effect in that it influences the way the system performs. Transfer, information, selection, and location policy selection are some of the schemes of this approach. Distributed nodes share the load-balancing tasks. Distributed and non-distributed dynamic load balancing can be classified as dynamic load balancing [22].

Distributed Dynamic Load Balancing Algorithm: in this algorithm, the execution is done by all nodes present in the system, and the scheduled task is shared amongst them. The communication between nodes is either cooperative or non-cooperative. Non-Distributed Load Balancing Algorithm; in this type, each node executes separately but instigate a shared goal. Semi-distributed Dynamic Load Balancing, it has nodes that are segregated in groups such that the load balancing of every group is centralized. The most suitable method of taking into account load balancing in each group is used in selecting the central node used to perform all load balancing in the system [18],[23].

Centralized Dynamic Load balancing: in this algorithm the execution is done by a single node in the whole system. The work of this node is load balancing while maintaining communication with all other nodes.

Table 2 Comparative Summary

| Paper | Algorithm used | Advantages | Disadvantages | Future Work |
|-------|----------------|------------|---------------|-------------|
| [1]   | Dynamic Load balancing algorithm | Balance workload among Virtual machines while increasing resources ratio utilization | It does not consider the priorities of a task and is more time-consuming | Put into consideration other parameters of the task such as delivery time |
| [2]   | Resource-Aware Min-Min | It has less completion time as well as producing the best result on small tasks. | the capacity of Virtual Machines and task priorities are not put into consideration | Extension of a multi-CC environment. |
| [3]   | Priority Based Shortest Job First | It has an effective method of utilizing resources and has a minimal response time | Consideration of tasks deadline is not taken care of. | Not Available. |
| [4]   | Honeybee Based Load Balancing | The remaining tasks are informed and updated on the status of allocation. | Time take to transfer, and complete tasks are not taken into consideration | Consider implementation Of task migration on dependent tasks. |
| [5]   | Clustering Algorithms | It has increased throughput, and it considers the priorities of a task. | Inadequacies of load balancing do exist. | Adopt an effective LBA. |
| [6]   | Classification ML Of Tasks and VMs | Has maximum utilization of resources and makes use of log files from the users. | It has huge task latency, and task priorities are taken into account. | Consider performance priority on tasks. |
| [7]   | Virtual machine APC | The utilization of resources is higher as compared to the matching traditional algorithm. | It has a limited number of tasks that are used for evaluation. | Consideration of pre-emptive scheduling of tasks should be enforced. |
| [8]   | Categorization of task on Basis of Global and Local queues | It prevents over and under usage of resources. | It has a non-pre-emptive scheduler. | Not Available. |
7. PROPOSED SCHEDULING ALGORITHM

To enhance the performances of cloud computing applications we proposed an algorithm for handling load balancing in CC. The pseudo-code algorithm used for scheduling tasks to the virtual machine is detailed as follows:

**Proposed Algorithm:**

1. Sorting of tasks based on the deadline ascendingly
   - Tasks if have the same deadline, then Pick the task with the earliest arrival time
   - **Else**
     - Priorities based on (deadline)
   - For each virtual machine
     2. Utilization computation
     3. Sort virtual machines according to their Utilization

   **Repeat**
   - If a virtual machine is available and a task is allocated to a heavy virtual machine.
     - then
     - Migrate task to the less utilized virtual machine
   - **Else**
     - Start scheduling
   - Till all tasks allocated to a virtual machine

**End**

Our proposed algorithm helps to maintain the load balancing using the priority,

- Task time of arrival, represents that task is ready and in queued, which can be measured in milliseconds [2]
- Task type can be identified based on the length of the task and further it can be divided in heavy, medium, and light.
- Task deadline time also measured in milliseconds and it represents the maximum time for the outcome of the task.

8. IMPLEMENTATION OF PRIORITY-BASED LOAD BALANCING ALGORITHM

For computation and assigning priority to every request based on its threshold value, and allocation of service to the requests is required [25][26]. Below we provide the step by step working of proposed algorithm and is also shown in Figure 7

**Stage 1:** Reading of the users' request data such as importance, time, node, price, and requested server name is done while inserting all the values to the list linked.

**Stage 2:** Assigning a PV (priority value) to every task on the user request list.

**Stage 3:** Based on predefined conditions for every request and the task to be performed, find the node PV (priority value).

**Stage 4:** For each input of data from the user, check whether it is within the threshold value or not

**Stage 5:** User's request is ready to execute.

| [9] | Priority Based Supervised Classification ML | It has improved scheduling and response time. | Load balancing and task migration are not into considerations. | Consideration of migrating tasks should be enforced. |
|-----|---------------------------------------------|-----------------------------------------------|----------------------------------------------------------|------------------------------------------------------|
| [10]| Using Queues Prioritize Task Scheduling     | Low priority task response time is reduced.    | Load balancing and task migration are not put into considerations. | Consider implementation Of task migration on dependent tasks. |
9. DISCUSSION
Due to the rapid and significant development that cloud computing is witnessing, there is an urgent need for efficient and high performance that meets users' renewable needs. One of the factors that significantly affect the performance of CC is load balancing. This paper worked on finding a solution to increase the efficiency load balancing process. To do this, we proposed a priority-based algorithm. This algorithm studies the tasks and deadlines for each task and rearranges them based on the deadline. If there are two tasks with the same deadline, it will prioritize the task that has the earliest arrival time. This algorithm effectively helps in recruiting tasks easily and quickly, thus improving overall load balancing performance. Different cloud-based applications have direct impact over the load balancing [27-28]. Load balancing helps the smooth run of different application with optimized resources.

10. CONCLUSION
Cloud computing is of great importance in technology and business, as they give everything to the users as services entailing applications, platforms, and infrastructure, all as services. So it is in continuous development and growth to meet the current needs of users. However, the increase of the provided services led to many challenges in the CC and one of these challenges is load balancing. Therefore, this paper proposed the priority-based algorithm with the aim of improving performance and load balancing by studying the priority of each task and arranging them based on the deadline to enhance and improve the response time. Proposed algorithm is best suitable for increasing performance of CC and efficient usage of resources.

11. FUTURE WORK
In future work, we will work to measure the algorithm's effectiveness and its effect on load balancing and will apply it on a real life case to obtain better results and performance.
REFERENCE

[1] F. Jauro, H. Chiroma, A. Y. Gital, M. Almutairi, S. M. Abdulhamid, and J. H. Abawajy, “Deep learning architectures in emerging cloud computing architectures: Recent development, challenges, and next research trend,” *Appl. Soft Comput. J.*, vol. 96, pp. 1–91, 2020, doi: 10.1016/j.asoc.2020.106582.

[2] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, “Proposing A Load Balancing Algorithm for the Optimization of Cloud Computing Applications,” *MACS 2019 - 13th Int. Conf. Math. Actuar. Sci. Comput. Sci. Stat. Proc.*, 2019, doi: 10.1109/MACS48846.2019.9024785.

[3] R. Buyya et al., “A manifesto for future generation cloud computing: Research directions for the next decade,” *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–51, 2019, doi: 10.11145241737.

[4] I. Pietri and R. Sakellariou, “Mapping virtual machines onto physical machines in cloud computing: A survey,” *ACM Comput. Surv.*, vol. 49, no. 3, 2016, doi: 10.11145/2985375.

[5] H. Noura, O. Salman, A. Chehab, and R. Couturier, “Preserving data security in distributed fog computing,” *Ad Hoc Networks*, vol. 94, no. 16, 2019, doi: 10.1016/j.adhoc.2019.101937.

[6] M. Durairaj and P. Kannan, “A Study On Virtualization Techniques And Challenges In Cloud Computing,” *Int. J. Sci. Technol. Res.*, vol. 3, no. 11, pp. 147–151, 2014.

[7] H. Tabrizchi and B. Uma Maheswari, “Research on enterprise public and private cloud service,” *Int. J. Innov. Technol. Explore. Eng.*, vol. 8, no. 6 Special Issue 4, pp. 1453–1459, 2019, doi: 10.35940/ijitee.F1296.0486S419.

[8] J. Govindasamy, “Cloud Computing : Cloud Delivery Models, Cloud Deployment Models, and Basic Security Challenges,” vol. IX, no. VI, pp. 2287–2289, 2020.

[9] S. G. Domanal, R. M. R. Guddeti, and R. Buyya, “A Hybrid Bio-Inspired Algorithm for Scheduling and Resource Management in Cloud Environment,” *IEEE Trans. Serv. Comput.*, vol. 13, no. 1, pp. 3–15, 2020,
[24] S. Lamba and D. Kumar, “A Comparative Study on Load Balancing Algorithms with Different Service Broker Policies in Cloud Computing,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 5671–5677, 2014.

[25] Humayun, M., Role of Emerging IoT Big Data and Cloud Computing for Real Time Application. International Journal of Advanced Computer Science and Applications (IJACSA), 2020. 11(4).

[26] R. Sundar Rajan and V. Vasudevan, “Enhanced priority based load balance scheduling of parallel work load in cloud computing,” *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 4, pp. 230–241, 2020, doi: 10.34218/IJARET.11.4.2020.023.

[27] Almusaylim, Z. A., & Jhanjhi, N. Z. (2020). Comprehensive Review: Privacy Protection of User in Location-Aware Services of Mobile Cloud Computing. Wireless Personal Communications, 111(1), 541-564.

[28] Z. A. Almusaylim, N. Zaman and L. T. Jung, "Proposing A Data Privacy Aware Protocol for Roadside Accident Video Reporting Service Using 5G In Vehicular Cloud Networks Environment," 2018 4th International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2018, pp. 1-5, doi: 10.1109/ICCOINS.2018.8510588.

[29] Natarajan, B., Obaidat, M.S., Sadoun, B., Manoharan, R., Ramachandran, S. and Velusamy, N., 2020. New Clustering-Based Semantic Service Selection and User Preferential Model. IEEE Systems Journal. DOI: 10.1109/JSYST.2020.3025407.

[30] Nataraj, S.K., Al-Turjman, F., Adom, A.H., Sitharthan, R., Rajesh, M. and Kumar, R., 2020. Intelligent Robotic Chair with Thought Control and Communication Aid Using Higher Order Spectra Band Features. IEEE Sensors Journal, DOI: 10.1109/JSEN.2020.3020971.

[31] Babu, R.G., Obaidat, M.S., Amudha, V., Manoharan, R. and Sitharthan, R., 2020. Comparative analysis of distributive linear and non-linear optimised spectrum sensing clustering techniques in cognitive radio network systems. IET Networks, DOI: 10.1049/iet-net.2020.0122.

[32] Sitharthan, R., Yuvaraj, S., Padmanabhan, S., Holm-Nielsen, J.B., Sujith, M., Rajesh, M., Prabaharan, N. and Vengatesan, K., 2021. Piezoelectric energy harvester converting wind aerodynamic energy into electrical energy for microelectronic application. IET Renewable Power Generation, DOI: 10.1049/rg.2.12119.

[33] Sitharthan, R., Sujatha Krishnamoorthy, Padmanaban Sanjeevikumar, Jens Bo Holm-Nielsen, R. Raja Singh, and M. Rajesh. "Torque ripple minimization of PMSM using an adaptive Elman neural network-controlled feedback linearization-based direct torque control strategy." International Transactions on Electrical Energy Systems 31, no. 1 (2021): e12685. DOI: 10.1002/2050-7038.12685.