International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

TASK SCHEDULING AND DYNAMIC RESOURCE ALLOCATION USING HEURISTICS ALGORITHM IN CLOUD COMPUTING

Mr. J. Noorul Ameen[1], Gowsalya R[2], Pavithra A[2], Pratheesha E[2]

[1]Associate professor, Computer Science And Engineering, E.G.S. Pillay Engineering College, Nagapattinam, Tamilnadu, India.
[2]UG, Student, Computer Science And Engineering, E.G.S. Pillay Engineering College, Nagapattinam, Tamilnadu, India.

ABSTRACT

Cloud computing has emerged as a technology that grease tasks by the dynamic allocation of virtual machines. Users pay for resources based on their demand. A cloud provider has to face many challenges. One out of the essential problem is load balancing, which suffers from many issues like premature convergence, reduced convergence speed, at first chosen random solutions, and stuck in native optima. Cloud computing is a modern paradigm to provide services through the Internet. Load balancing is a key aspect of cloud computing and avoids the situation in which some nodes become overloaded while the others are idle or have little work to do. Load balancing can improve the Quality of Service (QoS) metrics, including response time, cost, throughput, performance and resource utilization. Task Scheduling highly contributes to load balancing, and scheduling tasks much adheres to the requirements of the Service Level Agreement (SLA), a document offered by cloud developers to users. Important SLA parameters such as Deadline are addressed in the LB (load balancer) algorithm. The proposed algorithm is aimed to optimize resources and improve Load Balancing in view of the Quality of Service (QoS) task parameters, the priority of VMs, and resource allocation. To analyze the performance of heuristic-based algorithms, the simulation is carried out and the results are produced efficiently. This heuristic always allocates tasks to virtual machines arbitrarily and then checks for the next available machine. The proposed LB algorithm addresses the stated issues and the current work gap based on the literature's findings. Results showed that the proposed LB (Load Balance) algorithm and heuristic algorithm results in an average of resource utilization compared to the existing Dynamic LBA algorithm. It also achieves good performance in terms of less Execution time and Makespan. To analyze the performance of heuristic-based algorithms, the simulation is carried out and the results are presented in detail. This heuristic always allocates tasks to virtual machines arbitrarily and then checks for the next available machine.

Keyword*: Load Balance, Quality of Service, resource allocation*

1. INTRODUCTION

Cloud computing is a modern technology in the computer field to provide services to clients at any time. Cloud computing has faced many challenges, including security, efficient load balancing, resource scheduling, scaling, QoS management, data center energy consumption, data lock in and service availability, and performance monitoring. Load balancing is one of the main challenges and concerns in cloud environments it is the process of assigning and reassigning the load among available resources in order to maximize throughput, while minimizing the cost and response time, improving performance and resource utilization as well as energy saving. Service Level Agreement (SLA) and user satisfaction could be provided by excellent load balancing techniques. Therefore, providing the efficient load-balancing algorithms and mechanisms is a key to the success of cloud computing environments.

Load balancing in cloud computing provides an efficient solution to various issues residing in cloud computing environment set-up and usage. Load balancing must take into account two major tasks, one is the resource provisioning or resource allocation and other is task scheduling in distributed environment. The cloud computing is an Internet-based computing model that share resources (e.g., networks, servers, storage, applications, and services), software, and information to various devices of the user on demand. The efficient and scalable features of cloud computing can achieve by maintaining proper management of cloud resources. These cloud resources are in the virtual form which is the most important characteristics of the cloud system. The Cloud Service Provider (CSP) provides services to the users in rented basis. The role of CSP to provide the services to the user is a very complex one with the available virtual cloud resources.

The CSP design a trade-off between the financial benefits and user satisfactory through load balancing. The load balancing procedure also taking care of the Service Level Agreements (SLAs), the agreement between the CSP and the cloud users. The load balancing in clouds may be among physical hosts or VMs. This balancing mechanism distributes the dynamic workload evenly among all the nodes (hosts or VMs). The load balancing in the cloud is also referred as load balancing as a service (LBaaS). There are two versions of load balancing algorithms: static and dynamic. The static-based balancing algorithms are mostly fit for stable environments with homogeneous system. Dynamic-based balancing algorithms are more adaptable and effective in both homogeneous and heterogeneous environment. However, the application of static load balancing procedures has less system overhead as compared to the dynamic load balancing procedures. A large number of heuristics has been proposed in the literature to this problem.

2. LITERATURE REVIEW

Gomathi and Karthikey [1] introduce a method for assigning tasks in a distributed environment using Hybrid Particle Swarm Optimization algorithm (HybPSO). HybPSO is used to meet the user needs and increase the amount of load balancing with productivity. The goal is to minimize the task completion time among processors and create load balancing. This method assures that each task is assigned to exactly one processor. In this method, each solution is shown as a particle in the population; each particle is a vector with n dimension which is defined for scheduling n independent tasks.

In [2], authors introduce a method based on particle swarm algorithm for tasks' scheduling on distributed environment resources. Their model considers the computational cost and the cost of data transfer. Their proposed algorithm optimizes dynamic mapping tasks to resources using classical particle swarm optimization algorithm and ultimately balances the system loads. This optimization method is composed of two components. One of them is the scheduling operations task and the other one is particle swarm algorithm (PSA) to obtain an optimal mix of the tasks to resources' mapping

3. PROPOSED SYSTEM

Task Scheduling is a process that highly relates to workload balancing. Users send requests, the task is submitted through a cloud broker; this is where researchers should focus on providing an efficient algorithm. The proposed heuristic algorithm should efficiently submit jobs to appropriate VMs following essential parameters such as deadline to maintain a high quality of services and ensuring the requests sent by users are executed and completed within these specific requirements provided in the Service Level Agreement (SLA) document. The user sends requests via the Internet. These requests are stored in Virtual Machines (VMs), and CSP in every delivery model must maintain the QoS by ensuring the users' requests can be executed and completed within a specific deadline. This process depends highly on the scheduling policy's efficiency (Data Broker) which should be programmed to result in a high technique for balancing workload among the machines and servers. Efficient scheduling and utilization of resources can be achieved by designing and developing a dynamic load balancer (LB). Although Task Scheduling is one of the main goals of providing an efficient Load Balancing and improving performance, most researchers focus on one or two aspects. For example, to enhance Load Balancing and considers few Task Scheduling parameters. Thus, only a few metrics are taken into consideration to improve the overall performance. The proposed algorithm is aimed to optimize resources and improve Load Balancing in view of the Quality of Service (QoS) task parameters, the priority of task, and resource allocation. They have performed an experimental evaluation of eminent state-of-the-art static tasks scheduling heuristic algorithm. These algorithms have been critically investigated in terms of resource utilization, task execution time, throughput, and energy consumption. Moreover, individual load-imbalance is computed and compared.

4. METHODOLOGY

The proposed LB (Load Balance) algorithm aims to improve the cloud's performance by considering both aspects of Task Scheduling and Load Balancing. It utilizes all available CPUs in machines and schedules tasks appropriately to reduce Makespan, Execution Time, and maximize resource utilization. Below are the assumptions made in the proposed algorithm :

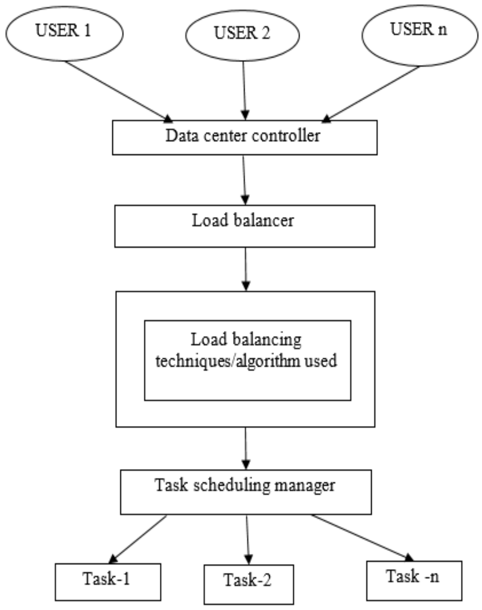
- One-to-many cloudlets (also known as task or user request) per Virtual Machine (VM).
- Cloudlets arrive in a random order (Arrival Time) • Each Cloudlet has a length, a time to complete known as Deadline (included in Service Level Agreement document), a completion time, and finally, the arrival time.
- The proposed algorithm checks the completion time for each workload (a total of cloudlets) against the Deadline.
- If there is any violation, whereby the completion time exceeds the Deadline, then the proposed algorithm will reconfigure the VM's priority based on its CPU. If it is in a successful state, the cloudlets get scheduled else; it will migrate the VM's workload.
- Expected Completion Time is calculated by taking the cloudlet length (also known as Million instruction per second (MIPS)) and dividing it by Virtual Machine MIPS (also known as CPU).
- Initially, all VMs share an equal portion of the available CPU; then, it is reconfigured based on the violation status. The CPU is set to its full utilization in the proposed algorithm.

The proposed algorithm gives better output in terms of energy efficiency, cost and also all the VMs are allocated tasks. The cost is improved by making the following changes in the cost function:

Cost=w1*ProcessingCost()*(FreePes() /NumberOfPes())+w2*Delaycost()

w1 and w2 are predefined weights, pes: processing elements.

5. ARCHITECTURE



6. CONCLUSION

The primary objective of the proposed algorithm is to minimize MakeSpan and improve fitness function. Improving the Load Balancing process through Task Scheduling can result in efficient utilization of cloud resources. The objective of this proposed work was to provide an enhanced heuristic algorithm. Results proved that our algorithm reduces Makespan and provide efficient resource utilization of compared to existing Dynamic LBA (Load Balancing Algorithm). It also shows that the proposed algorithm can function in a dynamic cloud environment where user requests arrive in random order and where there are many changes in the length of the user requests. The algorithm is also able to handle large size requests compared to the existing approach.

REFERENCE

[1] B. Gomathi and K. Karthikeyan: Task Scheduling Algorithm Based on Hybrid Particle Swarm Optimization in Cloud Computing Environment, Journal of Theoretical and Applied Information Technology, Vol. 89, 2013, pp. 33-38

[2] E. Emary, Hossam M. Zawbaa, Crina Grosan, and Abul Ella Hassenian: Feature Subset Selection Approach by Grey-Wolf Optimization. Industrial Advancement, Springer International Publishing, Vol. 63. 2015 pp. 1-13

[3] N. Zanoon, ‘‘Toward cloud computing: Security and performance,’’ Int. J. Cloud Comput.: Services Archit., vol. 5, no. vol. 5, nos. 5–6, pp. 17–26, Dec. 2015, doi: 10.5121/ijccsa.2015.5602.

[4] C. T. S. Xue and F. T. W. Xin, ‘‘Benefits and challenges of the adoption of cloud computing in business,’’ Int. J. Cloud Comput.: Services Archit., vol. 6, no. 6, pp. 1–15, Dec. 2016, doi: 10.5121/ijccsa.2016.6601.

[5] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, ‘‘Proposing a load balancing algorithm for the optimization of cloud computing applications,’’ in Proc. 13th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS), Dec. 2019, pp. 1–6, doi: 10.1109/MACS48846.2019.9024785.

[6] S. K. Mishra, B. Sahoo, and P. P. Parida, ‘‘Load balancing in cloud computing: A big picture,’’ J. King Saud Univ.–Comput. Inf. Sci., vol. 32, no. 2, pp. 149–158, 2020, doi: 10.1016/j.jksuci.2018.01.003.

[7] I. Odun-Ayo, M. Ananya, F. Agono, and R. Goddy-Worlu, ‘‘Cloud computing architecture: A critical analysis,’’ in Proc. 18th Int. Conf. Comput. Sci. Appl. (ICCSA), Jul. 2018, pp. 1–7, doi: 10.1109/ICCSA.2018.8439638.

[8] A. Jyoti, M. Shrimali, and R. Mishra, ‘‘Cloud computing and load balancing in cloud computing -survey,’’ in Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), Jan. 2019, pp. 51–55, doi: 10.1109/confluence.2019.8776948.