Optimization Techniques for Load Balancing and Task Scheduling in Cloud Computing Environment Using Feature Selection: A Review

Idris Mubarak
Department of Computer Science
Ahmadu Bello University,
Nigeria
idrismuby@gmail.com

Mustapha Aminu Bagiwa
Department of Computer Science
Ahmadu Bello University,
Nigeria
mstphaminu@yahoo.com

Muhammad Abdulkarim
Department of Computer Science
Ahmadu Bello University,
Nigeria
amuhd@abu.edu.ng

***Abstract*— With the advent of technology and the generation and accumulation of data on several platforms, cloud computing has become one of the major areas of scientific research. In support of data dependent operations, cloud computing provides high performance, fault tolerance, scalability, reliability, high availability, utilization, ease of use, monitoring capabilities, and economical. In spite of the numerous benefits cloud computing environments provide, they are faced with problems such as difficulty in achieving a high utilization rate of available resources, execution cost, turnaround time for task scheduling, and load balancing optimization. As the number of tasks and computing resources increases, the number of possible solutions and combinations that can give optimal results increase with the magnitude of the challenge which makes optimization of task scheduling and load balancing very difficult and in-fact an NP-hard problem. To address these shortcomings, feature selection techniques have been used by several authors to minimize the problem of load balancing and task scheduling. This paper provides a review of several feature selection techniques for load balancing and task scheduling optimization in cloud computing which will serve as a pool of knowledge for prospective researchers in this fascinating area of study for further research.**

Keywords — Cloud Computing, Load Balancing, Task Scheduling, Optimization

I. INTRODUCTION

Cloud computing is a subscription-based service, in which users can obtain networked storage space and computer resources [1]. It refers to a network or the internet typically available at a remote area and gives benefits over a system or web, or in a remote area. Cloud computing provide its clients with numerous functionalities such as granting access to an extensive number of users, and buying or renting application services online [2]. While a traditional computing setup restricts users to the same location as their data storage devices, cloud computing takes away that restriction allowing users to access their information from anywhere at any time. The cloud environment is a great resource area that provides cloud providers with easy access and gives them control over virtualization machines to accommodate growth in the data centers [3].

In terms of battery life, memory concerns, handling power conditions, memory and network issues, cloud computing addresses the limits of current smart procedures. [4] describes cloud computing as a recent and widely adopted trend in computing, infrastructure use, storage, networking, open access services, and application development across multiple fields. A large pay-per-use model-based, internet-based, scalable, flexible, elastic, utility, and metered computing platform is delivered via cloud. In the architecture of cloud computing, clients submit requests to cloud service providers in the form of tasks. The establishment of a service-level agreement between a cloud vendor and a customer which is frequently based on negotiation often determines how cloud services are used. Physical resources can be abstractly encased in cloud computing to provide consumers outside the cloud with varying levels of services [3]. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are the three categories into which cloud providers might group their main services. IaaS is the model at the lowest level of the cloud computing architecture. This paradigm provides basic computing resources like CPU, storage, memory, and bandwidth [3]. Appropriate allocation of resources made accessible by the cloud architecture improves the cloud's ability to deliver the desired and sustainable performance results from the computations it the virtual machines perform. Cloud technologies are used by large organizations that are regarded as industry leaders in the field of information technology, like Amazon, Google, and Microsoft rely on cloud computing technology for their operations. Due to the fact that every cloud resource must perform the same amount of work any time, load balancing and task scheduling are crucial topics in cloud computing and are used in cloud-based architecture [6]. In order to maintain a balance between requests and solutions for each individual request, specific procedures are employed by load balancing. Cloud computing is becoming more popular at the same time more labor-intensive. Allocating cloud jobs to nodes in the cloud environment properly while maximizing efficiency is a difficult task in cloud settings and is often referred to as load balancing and task distribution [5]. The goal of load balancing is to optimize resource usage in order to reduce response times, prevent overloading of resources, and

maximize the number of tasks completed in a specified timeframe.

As a result, load balancing has a substantial impact on performance in the cloud. The time frame between a client request and a response and the balancing of loads distribute a large volume of simultaneous access or information activity to several nodes individually [6]. Load balancing techniques employed in the cloud can be classified into three main categories, adaptive, static, and dynamic. Decisions in adaptive load balancing are not just dependent on the system's current condition, but also on changes in the external environment, as operational strategies may change. The majority of adaptation solutions make use of methods frequently referred to as intelligent approaches. Using deterministic or probabilistic methods that disregard the system's current state, static load balancing is carried out. Decisions are made using the present state of the system in the dynamic method [9] as it significantly affects the environment's performance. Task scheduling is another crucial component of the cloud. It is regarded as a multi-objective optimization problem, and as the number of constraints and factors under consideration increases, so does its complexity. Several techniques have been developed to address the issues with task scheduling and load balancing in a cloud computing environment. To solve task scheduling and load balancing issues, a number of heuristics were initially created, and they offered the best results for relatively small-scale issues. But as the scale of the problem increases, the quality of the solutions produced by the heuristic methods rapidly decline. Instead of employing a single candidate solution as with heuristic techniques, metaheuristic approaches use a more robust approach and are effective in solving a wide range of difficult real-world problems. This is due to the fact that they can use a set of candidate solutions to traverse the solution space. Some metaheuristic algorithms are the well-known Genetic Algorithm (GA) [7], Ant Colony Optimization (ACO) algorithm, Particle Swarm Optimization (PSO) [8], and Binary Crow Search Algorithm (BCSA) [12]. These algorithms have been applied to address the problems of task scheduling and load balancing in cloud environments and have demonstrated good performance. Despite the fact that some of these algorithms have shown improvement in finding the global optimal solution for task scheduling and load balancing problems in the cloud as noted above, they all suffer from premature convergence when faced with complex problems that have a large solution space which frequently results in suboptimal job and task scheduling problems [4]. Task scheduling and load balancing are two crucial factors that improve the efficiency of cloud environments. In the cloud context, load balancing techniques are used to distribute tasks and loads among nodes in order to improve resource allocation, enable flexibility and scalability, and speed up response times. Task scheduling is used to distribute tasks to various nodes in the cloud environment effectively. Therefore, it is crucial to continuously develop and enhance load balancing and task scheduling algorithms in order to increase the performance of cloud environments with the goal of efficiently serving all cloud technology users.

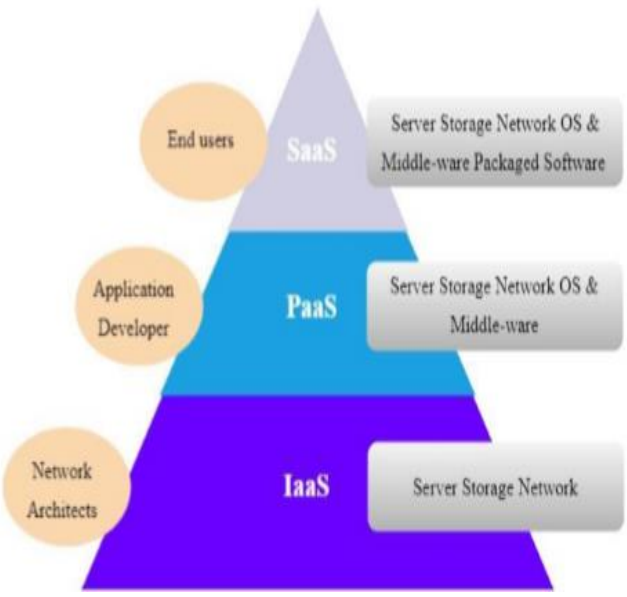

Fig. 1.1 Cloud computing architecture [10]

A. Load Balancing and Task Scheduling Optimization

Load balancing in cloud computing is the process of distributing load over resources and virtual machines in a cloud-based architecture. Cloud load balancers automatically distribute workloads to servers and computing resources to manage online traffic [11]. In a distributed environment, load balancing is the process of distributing work load to various computing nodes in the environment in order to efficiently utilize resources, maximize response time, and improve performance. Without load balancing, resources in a distributed system will be underutilized causing a negative impact to performance and efficiency. Optimization is the process of selecting the best or near best solution that minimizes or maximizes the objective function to a particular problem amongst a given set of competing solutions. Load balancing and task scheduling optimization is the process of distributing load and task amongst cloud computing resources in an efficient manner. There are different techniques for balancing load in the cloud. These techniques include but are not limited to pre-emptive approach, responsive load balancing, centralized and distributive approaches, and static and dynamic methodologies. Task scheduling is the process of allocating different tasks to the available resources in a system. Each task may include a number of variables such as the required data, execution time, and work priority to name a few. The challenge of mapping tasks in the cloud to resources in an efficient manner poses a Non-deterministic Polynomial (NP)-hard problem because of the high frequency and amount of request and operations that occur in cloud environments (Sharma et al., 2022). This is as a result of the number of possible solutions and combinations that increase exponentially with the magnitude of the problem such as the number of tasks and available computing resources leading to an optimization problem.

II. Related works

[13] proposed an efficient resource allocation and task scheduling mathematical model using Load Balancing Mutation Particle Swarm Optimization [LBMPSO]. The model takes into consideration factors such as reliability, transmission time, execution time, make span, and round-trip time, transmission cost, and load balancing between task and virtual machines. Compared with standard PSO, random algorithm, and Longest Cloudlet to Fastest Processor [LCFP], LBMPSO exhibits minimal execution time, round trip time, and transmission cost.

[14] proposed a work flow scheduling algorithm to schedule large scientific workflows to be executed on IaaS clouds. The algorithm generates a schedule with a task to resource mapping. Catfish Particle Swarm Optimization Algorithm (CPSOA) was used to select the optimal schedule with the least execution and make span cost. The algorithm was simulated on WorkFlowSim simulator and tested for different scientific workflows such as Cyber Shake, Inspiral, and Montage. The experiments revealed that CPSOA demonstrated a slightly better performance for workflows with tasks less than 400 and showed a significant improvement in make span and execution cost on workflows with 400 tasks which highlights the potential of CPSOA performing well on workflows with larger tasks.

[15] proposed a hybridized bat algorithm for allocation of jobs in cloud to optimal virtual machines. The approach is a combination of the bat algorithm and the tabu search technique. The algorithm prevents the same virtual machine from being visible again tabu search is used while balancing the loads. Hybrid BAT algorithm offers better exploration and diversity of search space by avoiding visiting the same solution again.

[16] proposed an efficient load balancing technique. This technique performs classification on the number of files present in the cloud using file type formatting. SVM is used to perform classification considering various file formats present in the cloud. The resultant data class is then fed into the Ant Colony Algorithm using file type formatting to achieve better load balancing in the cloud. Common QoS metrics such as migration time, throughput time, over-head time, and optimization time, are evaluated in cloud environment with comparative analysis performed with a variety of metaheuristic algorithms including Ant Colony Optimization-Particle Swarm Optimization (ACPSO), Chaotic Particle Swarm Optimization, Cat Swarm Optimization (CSO). The proposed algorithm outperforms related techniques in the literature providing good performance and scalability attributes.

Kruekaew & Kimpan (2020) proposed the combination of heuristic search algorithm with artificial bee colony (HABC). The algorithm is proposed to improve virtual machine scheduling solutions for cloud computing within homogeneous and heterogeneous environments to minimize make span and balance loads. The scheduling performance of cloud systems with HABC was compared with other swarm-based algorithms. Cloud sim was used to simulate systems that use different supplementing algorithms for comparing their make span and load balancing capability.

[11] proposed the used of BPSO and ANN for optimal load balancing and task scheduling in cloud computing. A revolutionary concept and calculation for each particle in the model is used to accelerate the research into binary space providing for substantial time savings while delivery the highest quality solution available. Two constraints were presented mainly update and optimization constraints to determine how many fitness solutions can quickly generate an ideal solution. The final step involved reduction of simulation cost to develop a more accurate model of particle position based on load balancing method with an updating constraint.

[17] proposed chaotic social spider optimization algorithm for load balancing and task aware optimization in various heterogenous virtual machines. The technique focused on minimizing overall make span with effective load balancing through the use of the swarm intelligence of social spider with chaotic inertia weight based random selection. The techniques prevent the local convergence at the same time exploring the global intelligence searching in finding the best optimized virtual machine for the user task in the set of virtual machines with minimum make span and balanced resource utilization. Simulation and performance evaluation was performed using cloudsim toolkit with the results compared with other swarm intelligent algorithms such as GA, PSO, and ABC. Experimental results showed that there is a major improvement in minimizing the make span with balanced task distribution.

[17] proposed the Hybrid Max-Min Genetic Algorithm (HMMGA) for effective cloud computing and load balancing. The best VMs to assign the tasks, and migrate the overloaded VMs tasks to the underload VMs tasks are selected by HMMGA. HMMGA significantly avoids the imbalanced workload performance in the cloud environment. The performance of HMMGA is compared to Max-Min Algorithm, Low time complexity, and Low-cost Particle Swarm Optimizer and PSO with Technique Of order Preference by Similarity to Ideal Solution (TOPSIS) algorithm. Experimental results showed that HMMGA on average delivers 1.63 and 3.88 less compared to Max-Min and TOPSIS-PSO algorithm for 5 VMs. In addition, an average enhancement of 10% to 40% was observed for HMMGA over the Max-Min and TOPSIS-PSO algorithm.

[17] proposed a new load balancing algorithm based on hybrid of Firefly and Improved Multi-Objective Particle Swarm Optimization (FIMPSO) algorithm. The technique employs the use of Firefly algorithm to minimize the search space while Multi-Objective Particle Swarm Optimization (IMPSO) was used to identify the enhanced response. IMPSO works by selecting the global best using a small distance of point to a line and through the application of minimum distance to point to a line, the global best candidates could be elected. FIMPSO algorithm achieved effective average load and enhanced performance measures such as proper resource usage and response time of tasks. Experimental results revealed that FIMPSO executed an effective performance as compared to related techniques with an average response time of 13.58 ms, maximum CPU utilization of 98%, memory utilization of 93%,

reliability of 67%, and throughput of 72%, and a make span of 148.

III. Conclusion

Through cloud computing technology computing resources are delivered over the internet offering cost savings, scalability, high performance, and economies of scale. Cloud users can typically access computational assets wherever, and whenever they need it in the required amount without necessarily maintaining and purchasing physical hardware resources providing flexible resources, and faster innovation. Load balancing and task scheduling are important concepts in cloud environments as cloud resources need to be adequately allocated to VMs in an efficient manner for effective resource utilization. Due to the complexity of effectively balancing load and allocating task in cloud environment, feature selection algorithms have been widely adopted to solve task scheduling and load balancing as an optimization problem. Thus, in this paper several techniques for optimization of load balancing and task scheduling using feature selection were reviewed as a basis for further research.

REFERENCES

[1] Huth A, Cebula J. The Basics of Cloud Computing. 2011;1–4.

[2] Malik MI. CLOUD COMPUTING-TECHNOLOGIES. Int J Adv Res Comput Sci. 2018;9[2]:379–84.

[3] Ghafari R, Mansouri N. An Efficient Task Scheduling Based on Seagull Optimization Algorithm for Heterogeneous Cloud Computing Platforms. 2022;35[02]:433–50.

[4] Attiya I, Elaziz MA, Xiong S. Job Scheduling in Cloud Computing Using a Modified Harris Hawks Optimization and Simulated Annealing Algorithm. 2020;2020.

[5] Chourasia U. Optimal Load Balancing in Cloud using ANFIS and MGWO based Polynomial Neural Network. 2021;

[6] Mishra K, Majhi SK. A binary Bird Swarm Optimization based load balancing algorithm for cloud computing environment. 2021;146–60.

[7] Sambangi S, Gondi L. DLMNN : A Deep Learning Modified Neural Network for Balancing the Load of Cloudlets on Cloud. 2019;8958[1]:6524–32.

[8] Hassanien AE. Particle Swarm Optimization from Theory to Applications. 2018;[December].

[9] Laabadi S, Naimi M, Amri H El, Achchab B. ScienceDirect ScienceDirect A Binary Crow Search Algorithm for Solving Two-dimensional Bin A Binary Crow Search Algorithm for Solving Two-dimensional Bin Packing Problem with Fixed Orientation Packing Problem with Fixed Orientation. Procedia Comput Sci. 2020;167[2019]:809–18.

[10] Jamal F, Khan RZ. Emerging Technologies and Developments in Cloud Computing : A Systematic Review International Journal of Emerging Trends in Engineering Research Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter46832020.pdf Emerging Technologies and D. 2020;[December].

[11] Alghamdi MI. Optimization of Load Balancing and Task Scheduling in Cloud Computing Environments Using Artificial Neural Networks-Based Binary Particle Swarm Optimization [BPSO]. 2022;

[12] Sharma I, Gupta R, Singh P. A Survey on Task Scheduling Approaches in the Cloud. 2022;[November].

[13] Awad AI, El-Hefnawy NA, Abdel-Kader HM. Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments. Procedia Comput Sci. 2015;65[January 2016]:920–9.

[14] Nirmala SJ, Bhanu SMS. Catfish-PSO based scheduling of scientific workflows in IaaS cloud. Computing. 2016;98[11]:1091–109.

[15] Sharma S, Verma S, Jyoti K, Kavita. Hybrid bat algorithm for balancing load in cloud computing. Int J Eng Technol. 2018;7[4.12 Special Issue 12]:26–9.

[16] Junaid M, Sohail A, Ahmed A, Baz A, Khan IA, Alhakami H. A Hybrid Model for Load Balancing in Cloud Using File Type Formatting. IEEE Access. 2020;8:118135–55.

[17] Comput JPD, Saviour AF, Elhoseny M, Dhanasekaran S, Lydia EL. Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments. J Parallel Distrib Comput [Internet]. 2020;142:36–45. Available from: https://doi.org/10.1016/j.jpdc.2020.03.022