Semiconductor Optoelectronics
ISSN: 1001-5868

A SURVEY ON MACHINE LEARNING BASED LOAD BALANCING IN CLOUD ENVIRONMENT

M. Ellakkiya
Research Scholar, #PG and Research Department of Computer Science, Thanthai Periyar Government Arts and Science College (Autonomous), Affiliated To Bharathidasan University,Tiruchirappalli, TamilNadu, India, ellakkiya.researchscholar@gmail.com

T.N.Ravi
Assistant Professor, *PG and Research Department of Computer Science, Thanthai Periyar Government Arts and Science College (Autonomous), (Affiliated To Bharathidasan University),Tiruchirappalli, TamilNadu, India, proftnravi@gmail.com

Abstract-Cloud computing is an on-demand availability of computer system resources for data storage and computing power without active management by user. Cloud providers employ the data center (DC) across globe with number of IT servers. Load balancing is a challenging concept in cloud computing. It is not easy to arrange resources in cloud because the workload in cloud gets changed from time to time. Load balancing distributes the request between diverse machines through performing the task scheduling process. Load balancing minimizes the makespan time while increasing the cloud resource usage. However, the cloud environment experienced from challenges that reduce the performance because of inefficient resource utilization. In order to address above problems, deep learning and machine learning based load balancing methods are introduced to schedule the tasks in virtual machine.

Keywords- Cloud computing, data center, cloud providers, load balancing, resource utilization, task scheduling, makespan, deep learning, machine learning strategy.

I.INTRODUCTION

Cloud is an essential one in information technology in recent years. Cloud computing is the practice of network with the remote servers hosted on internet to store, handle and process data. Cloud computing resources provided the high-speed internet services to users through application. Cloud computing is an on-demand delivery of IT resources over Internet with pay-as-you-go pricing. Cloud computing is the method of delivering the technology to the consumer by Internet servers for processing and data storage. Load balancing is the process of distributing the incoming network traffic across the group of backend servers. Load balancer distributes the client requests or network load effectively across the multiple servers. Load balancer guaranteed high availability and reliability through sending the requests to servers.

This paper is arranged as follows: Section II describes the existing load balancing methods. Section III discusses the experimental settings with comparison between them. Section IV

explains the limitation of existing load balancing techniques. Section V concludes the paper.

II .LOAD BALANCING IN CLOUD ENVIRONMENT

Cloud Computing is described as the method of storing and accessing the data and computing services over Internet. Cloud computing provides the cost-effective and timely disaster recovery options that assist the organizations with speedy data recovery. Cloud load balancing is described as method of splitting the workloads and computing properties in cloud computing. It allowed the enterprise to handle the workload demands through distributing the resources among many computers, networks or servers. Load balancing is the process of distributing collection of tasks over set of resources for making the processing task more efficient. Load balancing optimized the response time and avoided the overloading task.

A. On-demand resource provision based on load estimation and service expenditure in edge cloud environment

An on-demand resource provision model was introduced depending on the service expenditure. A load estimation model was designed depending on ARIMA model and BP neural network. The designed model determined the load consistent with the historical data and minimized the estimation error. The user data on node has migrated to guarantee that user data not get lost. A combined model was introduced to estimate the load depending on the historical data in time during rush hours and avoid the data overload. The load of every area in city was diverse. The edge nodes of area with large load were applied for additional resources to cloud service provider to meet load demand. The area with small load employed the idle resources to cloud service provider due to the resource billing granularity. An on-demand resource provision model was introduced to minimize the service expenditure. A data migration model was introduced depending on the load balancing.

ARIMA model with BP neural network was introduced to address the load estimation issues in edge cloud environment. ARIMA model was employed to determine the linear load sequence. The BP neural network was introduced to manage the nonlinear factors in load sequence. ARIMA model with BP neural network minimized the load estimation error. Based load estimation value, resources in edge cloud environment was provided on demand before load arrives. The explicit cost and hidden cost of resources was introduced through considering the service expenditure. When the load peak period ends, the extra resources are waiting to be released. The user data was migrated to appropriate working nodes consistent with load balancing situation of different nodes, the data migration time and the data transmission cost. The applicable scenarios of on-demand resource provision method depending on load estimation and service expenditure were provided to enhance the system stability.

B.Dynamic provisioning of resources based on load balancing and service broker policy in cloud computing

Dynamic resource allocation is an essential objective that encouraged because of number of user service request and increasing network infrastructure complexity. Load balancing and Service Broker Policy were two essential areas for dynamic resource provision to the cloud user for solving the QoS requirement. Multi-agent Deep Reinforcement Learning-

Dynamic Resource Allocation (MADRL-DRA) approach was introduced to present dynamic resource provisioning depending on load balancing and service brokering. MADRL-DRA approach was employed in Local User Agent (LUA) to forecast the environmental user task activities and assign task to the Virtual Machine (VM) depending on priority. Load balancing (LB) was carried out in VM to increase the throughput and to minimize the response time in resource allocation task. Dynamic Optimal Load-Aware Service Broker (DOLASB) was employed in Global User Agent (GUA) for task scheduling.

DOLASB provided the services to users depending on the available cloud brokers (CBs). In global agent, cloud brokers were the mediators between the users and providers. The optimization problem in Global Agent (GA) is formulated by the programming of mixed integers, and Bender decomposition algorithm. Load balancing assigned load over nodes in distributed system and increased the resource utilization as well as response time of task. Resource allocation allocated resources and addressed user expectations with minimal allocation time. MADRL-DRA and DOLASB Method was introduced to improve the optimal multi-cloud service performance and to reduce client request cost. A prediction model was designed to forecast the activity from client request and to design adaptive method for scheduling.

C. A novel cooperative resource provisioning strategy for Multi-Cloud load balancing

Multi-Cloud architecture was introduced to present services in Continuous Writing Application (CWA). A new resource scheduling algorithm was introduced to reduce the system cost. The system model was introduced to manage the resource needs of user on MCP. The study was carried out to recognize the features of different resources for CWA implementation. An optimal user scheduling was carried out depending on Minimum First Derivative Length (MFDL) of system load paths. Multi-Cloud was introduced to guarantee the service availability for satisfying the CWA needs. The service demands addressed the resources requirements for each CWA in terms of storage capacity, bandwidth and CPU cycles. CSP handled the local resources provided by CSP to the users. The resource utility costs get varied for same user service depending on geographical distribution. The same data from single user get stored by two CSPs to guarantee the data availability. Every CSP constructed the virtual Multi-Cloud and arranged the resources flexibly to address the increasing users.

III.PERFORMANCE ANALYSIS OF LOAD BALANCING METHODS IN CLOUD ENVIRONMENT

In order to compare the load balancing methods, number of satellite images is considered as an input to conduct the experiment. Experimental evaluation of three methods namely on-demand resource provision model, Multi-agent Deep Reinforcement Learning-Dynamic Resource Allocation (MADRL-DRA) approach and Multi-Cloud architecture are implemented using Java language. In order to perform efficient load balancing in cloud, Diabetes 130-US hospitals for years 1999-2008 Data Set is collected from the UCI Machine Learning Repository. The URL of the dataset is given as UCI Machine Learning Repository: Diabetes 130-US hospitals for years 1999-2008 Data Set. The input dataset comprises 100000 data instances (i.e., number of user requested tasks) with 55 attributes. The result analysis of

existing load balancing techniques in cloud environment are determined with certain parameters are,

- Load Balancing Efficiency and
- False Positive Rate
- Makespan

4.1 Performance Analysis of Load Balancing Efficiency

Load balancing efficiency is defined as the ratio of number of user requested tasks that are correctly balanced the load to the total number of user requested tasks. Load balancing efficiency is determined as,

$$LBE = \left(\frac{Number\ of\ user\ requested\ tasks\ that\ are\ correctly\ balanced\ the\ load}{n}\right) * 100$$

From (1), LBE'denotes the load balancing efficiency. 'n ' symbolizes the number of user requested tasks. The load balancing efficiency is computed in terms of percentage (%).

Table 1 Tabulation for Load Balancing Efficiency

Number of user requested tasks	Load Balancing Efficiency (%)		
	On-Demand Resource Provision Model	MADRL-DRA approach	Multi-Cloud architecture
100	85	78	70
200	87	80	73
300	90	82	75
400	92	85	78
500	91	83	76
600	88	81	75
700	90	82	77
800	93	84	80
900	91	83	78
1000	94	85	81

Table 1 explains the load balancing efficiency with respect to number of cloud user requested data ranging from 100 to 1000. Load balancing efficiency comparison takes place on existing on-demand resource provision model, Multi-agent Deep Reinforcement Learning-Dynamic Resource Allocation (MADRL-DRA) approach and Multi-Cloud architecture. The graphical representation of load balancing efficiency is illustrated in figure 1.

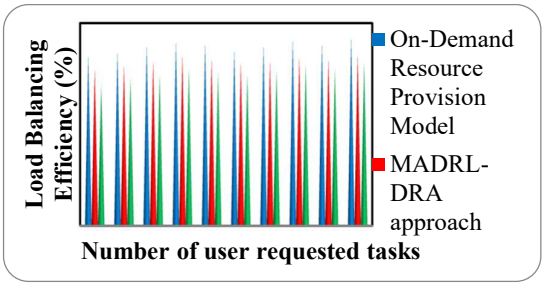

Figure 1 Measurement of Load Balancing Efficiency

As described in figure 1, the load balancing efficiency for different number of user requested tasks is explained. The blue colour cone in figure denotes the load balancing efficiency of on-demand resource provision model. The red colour cone and green colour cone represents the

load balancing efficiency of MADRL-DRA approach and Multi-Cloud architecture respectively. It is clear that load balancing efficiency using on-demand resource provision model is higher when compared to MADRL-DRA approach and Multi-Cloud architecture. This is due to the application of combined model to compute the load depending on historical data in time and avoid data overload. An on-demand resource provision model minimized the service expenditure and increased load balancing efficiency. As a result, the load balancing efficiency of on-demand resource provision model is reduced by 9% when compared to the MADRL-DRA approach and 18% when compared to the Multi-Cloud architecture.

4.2 Performance Analysis on Makespan

Makespan is described as the product of number of user requested tasks and amount of time taken to balance the load of one user requested tasks to resource optimized virtual machines. Makespan is formulated as,

$$Makespan = n * time\ (balance\ the\ load\ of\ one\ user\ requested\ task) \quad (2)$$

From (2), 'n' denotes number of user requested tasks. The makespan is computed in terms of the milliseconds (ms).

Table 2 Tabulation for Makespan

Number of user requested tasks	Makespan (ms)		
	On-Demand Resource Provision Model	MADRL-DRA approach	Multi-Cloud architecture
100	29	25	34
200	31	28	36
300	33	30	39
400	35	32	42
500	38	35	44
600	41	37	47
700	43	40	50
800	46	42	53
900	49	45	56
1000	52	48	59

Table 2 describes the makespan with respect to number of cloud user requested data ranging from 100 to 1000. Makespan comparison takes place on existing on-demand resource provision model, Multi-agent Deep Reinforcement Learning-Dynamic Resource Allocation (MADRL-DRA) approach and Multi-Cloud architecture. The graphical representation of makespan is illustrated in figure 2.

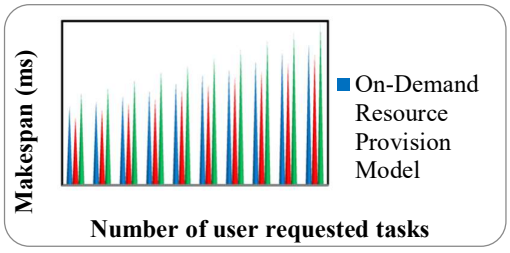

Figure 2 Measurement of Makespan

From the figure 2, the makespan depending on different number of user requested tasks is illustrated. The blue colour cone in figure represents the makespan of on-demand resource provision model. The red colour cone and green colour cone represents the makespan of MADRL-DRA approach and Multi-Cloud architecture respectively. It is clear that makespan

using MADRL-DRA approach is lesser when compared to machine on-demand resource provision model and Multi-Cloud architecture. This is due to the application of Local User Agent (LUA) in MADRL-DRA approach to predict the environmental user task activities and allocate the task to Virtual Machine (VM) based on their priority. Load balancing (LB) increased the throughput and reduced the makespan during the resource allocation. As a result, the makespan of MADRL-DRA approach is reduced by 9% when compared to the on-demand resource provision model and 22% when compared to the Multi-Cloud architecture.

4.3 Performance Analysis of False positive rate:

False positive rate is the described as the number of user requested tasks that are incorrectly scheduled to the virtual machine in the cloud server to the total number of user requested tasks. The false positive rate is computed as given below,

$$FPR = \left(\frac{Number\ of\ incoming\ tasks\ that\ are\ incorrectly\ scheduled}{n}\right) * 100 \quad (3)$$

From (3), 'FPR ' denotes the false positive rate. 'n ' represent the number of user requested tasks. The false positive rate is determined in terms of percentage (%).

Table 3 Tabulation for False Positive Rate

Number of user requested tasks	False Positive Rate (%)		
	On-Demand Resource Provision Model	MADRL-DRA approach	Multi-Cloud architecture
100	25	18	11
200	28	20	15
300	30	23	18
400	27	21	16
500	24	19	14
600	26	22	17
700	29	24	20
800	31	28	22
900	33	32	25
1000	36	35	27

Table 3 explains the false positive rate with respect to number of cloud user requested data ranging from 100 to 1000. False positive rate comparison takes place on existing on-demand resource provision model, Multi-agent Deep Reinforcement Learning-Dynamic Resource Allocation (MADRL-DRA) approach and Multi-Cloud architecture. The graphical representation of false positive rate is illustrated in figure 3.

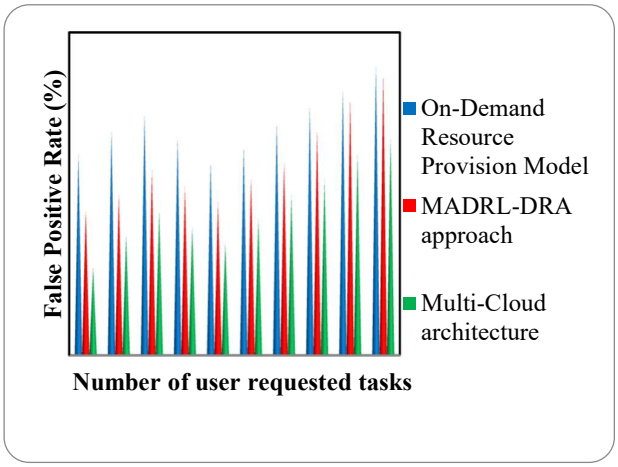

Figure 3 Measurement of False Positive Rate

As illustrated in figure 3, the false positive rate depending on different number of user requested tasks is described. The blue colour cone in figure represents the false positive rate of on-demand resource provision model. The red colour cone and green colour cone represents the false positive rate of MADRL-DRA approach and Multi-Cloud architecture respectively. It is observed that false positive rate using Multi-Cloud architecture is lesser when compared to machine on-demand resource provision model and MADRL-DRA approach. This is because of performing the optimal user scheduling depending on Minimum First Derivative Length (MFDL). Multi-Cloud guaranteed the service availability for satisfying the CWA needs with minimal false positive rate. Accordingly, the false positive rate of Multi-Cloud architecture is reduced by 37% when compared to the on-demand resource provision model and 24% when compared to the MADRL-DRA approach.

1. DISCUSSION ON LIMITATIONS OF LOAD BALANCING TECHNIQUES IN CLOUD ENVIRONMENT

An on-demand resource provision model was performed depending on the service expenditure. The resource demands were computed with load estimation model depending on ARIMA model and BP neural network. The resource demand needs to be estimated in advance. But, an on-demand resource provision model failed to minimize the estimation error. In addition, throughput level was not increased by on-demand resource provision model. MADRL-DRA was employed with Local User Agent (LUA) to predict the environmental activities of user task and to allocate the task to Virtual Machine (VM) depending on their priority. Load balancing (LB) was carried out in VM to increase throughput and to reduce the network infrastructure complexity. However, the load balancing efficiency was not improved by designed approach. Multi-Cloud architecture was performed to provide the services to Continuous Writing Applications (CWA). A resource scheduling algorithm reduced the system cost. Multi-Cloud architecture increased the data accessibility with low cost. However, the load balancing efficiency was not increased by Multi-Cloud architecture.

5.1 Future Direction:

The future direction of load balancing techniques can be carried out using machine leaning and deep learning techniques for improving the load balancing performance with higher efficiency and lesser makespan.

6. CONCLUSION

A comparison of different existing load balancing methods was discussed. From the study, it is observed existing load balancing techniques failed to increase the efficiency in cloud environment. The survival review shows that on-demand resource provision model failed to minimize the estimation error. In addition, makespan was not reduced. The wide range of experiments on many existing load balancing techniques determines the performance with its limitations. Finally, from the result, the research work can be carried out using machine learning techniques and deep learning methods for increasing the load balancing performance.

REFERENCES

[1] Jingjing Guo, Chunlin Li, Chen Yi and Youlong Luo, "On-Demand Resource Provision Based on Load Estimation and Service Expenditure in Edge Cloud Environment", Journal of Network and Computer Applications, Elsevier, Volume 151, February 2020, Pages 1-31

[2] Amrita Jyoti and Manish Shrimali, "Dynamic provisioning of resources based on load balancing and service broker policy in cloud computing", Cluster Computing, Springer, April 2019, Pages 1-9

[3] Bo Zhang, Zeng Zeng, Xiupeng Shi, Jianxi Yang, Bharadwaj Veeravallid and Keqin Li, "A novel cooperative resource provisioning strategy for Multi-Cloud load balancing", Journal of Parallel and Distributed Computing, Elsevier, Volume 152, June 2021, Pages 98-107

[4] Yogesh Gupta, "Novel distributed load balancing algorithms in cloud storage", Expert Systems with Applications, Elsevier, Volume 186, December 2021, Pages 1-20

[5] Venkateshwarlu Velde, Kiran Enumala and Krishna Bandi, "Optimized Adaptive load balancing algorithm in cloud computing", Materials Today: Proceedings, Elsevier, March 2021, Pages 1-15

[6] Chunlin Li, Jianhang Tang, Tao Ma, Xihao Yang and Youlong Luo, "Load balance based workflow job scheduling algorithm in distributed cloud", Journal of Network and Computer Applications, Elsevier, Volume 152, February 2020, Pages 1-18

[7] A. Francis Saviour Devaraj, Mohamed Elhoseny, S. Dhanasekarana, E. Laxmi Lydia and K. Shankar, "Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments", Journal of Parallel and Distributed Computing, Elsevier, Volume 142, August 2020, Pages 36-45

[8] Zhao Tong, Xiaomei Deng, Hongjian Chen and Jing Mei, "DDMTS: A novel dynamic load balancing scheduling scheme under SLA constraints in cloud computing", Journal of Parallel and Distributed Computing, Elsevier, Volume 149, March 2021, Pages 138-148

[9] Zhang Miao, Peng Yong, Yang Mei, Yin Quanjun and Xie Xu, "A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment", Future Generation Computer Systems, Elsevier, Volume 115, February 2021, Pages 497-516