IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A REVIEW ON OPTIMIZATION TECHNIQUE IMPLEMENTED ALGORITHMS IN CLOUD COMPUTING

M.Vidhya[1], Dr. R. Devi[2]

[1]Research Scholar, [2]Associate Professor

Department of Computer Science, VISTAS, Chennai, India

ABSTRACT

Cloud Computing emphasis the distributive environment that enhances sharable computing such as applications, software, network, security, storage area, processing power and many other services over the internet. The dynamically demanded requests can be satisfied by virtualized shared resources with internet support. All the features can be attained by various virtualization techniques. The shared resources can be utilized by the customers from resource pooling on pay per use basis. Even more and more services are provided by cloud, still there are some issues arises while resource sharing, security issues at sharing the data, virtual machine migration etc. Those issues can be identified and resolved to assure the better service quality in computing environment. Balancing the workload is major constraint in cloud environment, because it ensures the reliability, efficiency and availability. In this paper, we specify the considered Quality-of-Service metrics, merits and demerits on each optimized algorithms used for load balancing.

KEYWORDS

Cloud Computing, Resource, Optimization, QoS, Load balancing.

1. INTRODUCTION

Cloud computing is an Internet based computing, which allows to access the sharable resources such as hardware, application, memory, databases, storage and network to the customer based on their requirements, which are dynamically changed. Resources availability can be achieved by means of virtualization techniques implemented on various levels [1]. Cloud Service Providers (CSP) supports three categories of services for clients in cloud environment such as SaaS (Software-as-a-Service), PaaS (Platform-as-a-Service), and IaaS (Infrastructure-as-a-Service). CSP receives a request from client; they must establish required number of virtual machines and needed resources to support the customer [2]. Two ways of provisioning can be supported. CSP provides all the services and resources in advance and the client pay the fixed fee or monthly basis payment, this one is called Advance Provisioning. CSP supports the dynamically changed requests of clients and billed on pay-per-use basis is called Dynamic provisioning [3].

Cloud environment have some specialized features such as virtualization, heterogeneity, elasticity, resource pooling and measurable services. Because of those unique features, Cloud faces many challenging issues such as fault tolerance, security, scalability, reliability and scheduling the resources etc. Service Provider's goals are to attain maximum profit and return investment [4]. Similarly the customer also demands cheapest, reliable and more securable services. Resource sharing emphasizes the solution for both side expectations.

Because of concurrent geographic channels, different client requests can be assigned to various cloud providers, sometimes unequal assignment of workload for a node can be occurred. Some of the nodes in the network are overloaded and other nodes may be under loaded [5]. This surely degrades the performance of cloud environment. To handle that problem, appropriate load balancing algorithms is most essential. In this paper, different load balancing approaches are discussed.

2. LOAD BALANCING

Load balancing is a distributive service fully defined and managed by software. It's free from hardware so no need to maintain the physical infrastructure for cloud. Load balancer has predefined functions to distribute the workloads across networks and servers [6]. Load balancer handles the customer requests with the support of many instances of our applications. It monitors the customer demand and workload traffic across the network. To satisfy the client requirements it enhances the multiple server support across the network [7].

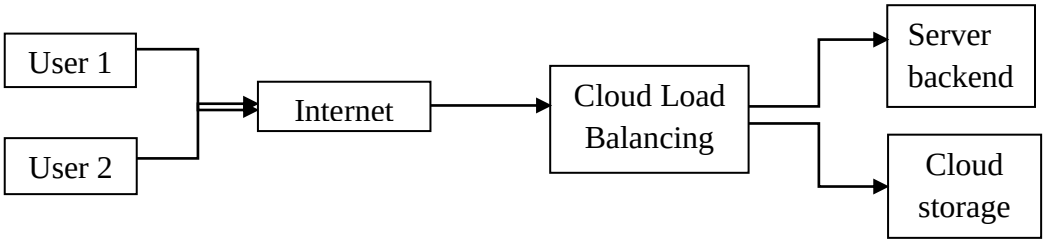

Figure 1. Simple Load Balancing

Load Balancing (LB) can be performed by means of load balancing algorithms. LB algorithms perform the following functions [8].

- Client requests are distributed across multiple servers.
- Ensures reliability and availability of servers to requests send by user.
- Grow or shrink the number of servers based on user's dynamic requirement.

Different approaches of load balancing are

- Static Approach: If the workload is assigned to any one node, we can't change the load to any other node. It is on non-preemptive approach. It concentrates more about the information on the node and negotiates the state of that node [9].
- Dynamic Approach: It majorly concentrates the present state of the node and distributes the load among the nodes. So it is preemptive approach. Single node can handle the state of the system and distribution work then that is stated as Centralized Approach [10].

Main aim of load balancer is enhance the performance of the system. To determine the system performance, various qualitative measures are considered [11]. Some of the most considerable measures are specified as follows.

- Throughput: Number of tasks completed its execution with in specified time. System should attain high throughput.
- Fault Tolerance: Any failure occurs on that system, it could be able to handle the fault and maintain the system in good state.
- Migration Time: Time taken to transfer the task from one node to another node in a system. System attains better performance, if the system utilizes less migration time.
- Response Time: Time taken for first response to task from the system. Minimal response time should attain better performance.
- Scalability: System should support the dynamic requests from customers. System should handle increased workload and required number of resources.
- Makespan: Time taken to finish all the allocated process for that node. By means of efficient load balancer, we can attain less makespan.
- Energy Consumption: Service provider effectively handles all the machine workload and all resources energy consumption.
- Resource Utilization: All resources should be maximally utilized.

3. REVIEW OF DIFFERENT LOAD BALANCING ALGORITHMS

Cloud Computing attains distributed computing by means of various technologies such as virtualization and interoperability. Main goal of cloud is to get better system performance with heterogeneous type of resources. Load balancing algorithms are partitioned into three types such as heuristic approach algorithms, metaheuristic approach algorithms and hybrid approach algorithms [12].

Heuristic Approach Load Balancing Algorithms

Heuristic approach has group of constraints that determines the best solution for any specific problem. Constraints are designed and developed based on the problem and it provides a way to get better solution with minimal time. These algorithms can be of static type or dynamic type.

Metaheuristic Approach Load balancing Algorithms

All the metaheuristic algorithms are dynamic in nature. These algorithms require more time than heuristic load balancing algorithms to execute and find the solution for any problem. Time requires developing the solution based on the problem nature, their requirements and method we prefer to design the solution. All types of swarm intelligence algorithms are belong to this metaheuristic approach. Various metaheuristic approach algorithms are discussed with its considered measures, merits and its limitations.

Table 1. Comparison of Various Optimization techniques implemented Algorithms

Algorithms	References	Environment	Concentrated Measures	Results Compared	Merits	Challenges
Artificial Bee Colony algorithm	[13]	Cloud	Performance Functions such as rate of mutation and crossover.	ES, GA, PSO and DE.	It uses less control parameters for handling the optimization problems such as multimodal as well as multidimensional.	Performance functions are more complex.
Improved Whale Optimization Algorithm	[14]	CloudSim	Time and cost for task scheduling in virtual machine.	Ant Colony and Particle Swarm Algorithm	Cost and efficiency in cloud is improved.	Maximum speed and accuracy is attained in task scheduling.
ABC with Simulated Annealing	[15]	CloudSim	Makespan	MFCFS, LJF and SJF	Validate the request and virtual machine.	More metrics to be considered.
Cuckoo Harmony Search Algorithm	[16]	CloudSim	Expense, storage capacity and energy utilized.	CS, HS and CGSA	Minimize the expense, utilization of memory as well as energy consumption, and get maximum credit.	Implemented on cloud environment.
Energy and Performance Efficient Task Scheduling Algorithm	[17]	Cloud Analyst	good performance and reduce energy usage within the time limits	Genetic Algorithm, AMTS, and E-PAGA	Minimize the energy consumption and enhance the performance by 5%–20%.	Develop more techniques in heterogeneous virtualized cloud environment for efficient task scheduling
Genetic Gray Wolf Optimization	[18]	Cloud Environment	Load utilization, energy requirement, cost and time for migration.	GWO and GA	Improve task scheduling with less computation time, expense for migration, energy requirement and enhance the load utilization.	Consider more scientific workflows.
Slave Ants based Ant Colony Optimization Algorithm	[19]	Cloud Environment	Processing time and Makespan.	Ant Colony Optimization and Improved ACO	Efficiently maximizing the utilization of cloud servers.	Consider heterogeneous clusters.
Hybrid Antlion	[20]	CloudSim	Makespan and Resource	MSDE, Genetic	Attain high degree of	Improve its time

Optimization Algorithm			Utilization.	Algorithm, Particle Swarm optimization, DSOS, MSA, and ALO	Response time, and makespan.	complexity, memory use, and overloads.
Modified Particle Swarm Optimization (MPSO) Algorithm	[21]	CloudSim	Robustness, Scalability, and high Flexibility.	Min-Min, Max-Min, and SPSO.	It enhances the performance in cloud environment.	Applied for all degree of load in real-time cloud environment.
Adaptive PSO-Based Task Scheduling (AdPSO) Algorithm	[22]	CloudSim	Execution time, Throughput and Resource Utilization.	AIW, CRIW, SRIW, and LDIW	Maximum gets 10% improvement in Makespan, 12% improvement in Throughput and 60% in ARUR.	Implement on Real time Environment.
Crow Search Algorithm (CSA)	[23]	CloudSim	VM selection and Makespan.	Min–Min and ACO Algorithms.	Makespan reduction occurred.	Algorithm implemented on dynamically flight length.
Multi-Faceted Optimization Scheduling Framework - PSO Algorithm	[24]	Cloudsim	Cost, Makespan Deadline, and Resource Utilization.	FCFS, Min-Min and PSO.	Provides better output while calculates the cost, violation rate and resource utilization.	Metrics result compared with other optimization methods.
An Improved Evolutionary Algorithm, RAA-PI-NSGAII	[25]	CloudSim	Resource Allocation.	Round Robin, SPEA2, and NSGA-II.	Satisfies the emergent demands in cloud.	Implement on Cloud environment.
Multi Objective Whale Optimization Algorithm-based Differential Evolution (M-WODE)	[26]	CloudSim	Cost and Makespan	Multi Objective GA, Multi Objective ACO and Multi Objective PSO	Provides better makespan and minimize the cost.	Analyze the impact using some control parameters.
Proximity based Task Scheduling Algorithm	[27]	-	Resource Utilization and Power Consumption.	-	Minimize the consumption of energy.	Algorithm implementation.
Enhanced LB Min-Min Algorithm	[28]	-	Makespan	Min-Min, LBMM.	Prefer the task with less completion time and available	Practical implementation of algorithm.

(ElBMM)					resource.	
Improved PSO-based Task Scheduling Algorithm	[29]	CloudSim	Resource Utilization.	GA, SA, ACO, IPSO	Increased resource utilization and task handling.	Task scheduling in dynamic.
Integer-PSO : a discrete PSO Algorithm	[30]	Cloud Environment.	Cost, Makespan.	GA, ACO, ABC, and PSO.	Algorithm finds optimal or near optimal solutions.	Workflow and performance of the algorithm will be explored.
Improved Ant Colony Optimizatio n Algorithm	[31]	CloudSim	Wait time, Load Balance and Cost.	Different Versions of ACO.	High convergence speed with less completion time.	Identify the failure cloud node and its recovery strategy to perform quantitative analysis.
Grey wolf optimization algorithm	[32]	CloudSim	Load balancing and task scheduling.	PSO, ABC and GA.	Better Accessibility, Security, and Scalability and Reliability	Dynamic load balancing will be implemented on dependent tasks.
Hybrid of Firefly and IPSO algorithm	[33]	MATLAB	Response time and resource utilization.	FCFS, Round Robin, Genetic Algorithm, SJF, Improved PSO and firefly.	Rapid convergence is more effective and efficient.	Concentrate on other metrics.

4. CONCLUSION

Cloud environment engaged the requests received by both industries and academics. Major aim of cloud is to satisfy both the clients as well as service providers. Even many merits exists in cloud, some flaws are there in load balancing, migration of VM and server utilization. This paper presents various optimized variants, it's considered metrics, developed environment, merits attained through each algorithm and challenges on cloud environment. Among multiple issues, load balancing seems as major issue. Variety of techniques and algorithms are implemented to attain the better service quality and effective utilization in minimal cost and complexity. In future, we will develop a method for Dynamic optimized load balancing and resource handling in well effective manner.

REFERENCES

[1] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 2. King Saud bin Abdulaziz University, pp. 149–158, Feb. 01, 2020. doi: 10.1016/j.jksuci.2018.01.003.

[2] S. Swarnakar, R. Kumar, S. Krishn, and C. Banerjee, "Improved Dynamic Load Balancing Approach in Cloud Computing," *2020 IEEE Int. Conf. Converg. Eng. ICCE 2020 - Proc.*, pp. 195–199, 2020, doi: 10.1109/ICCE50343.2020.9290602.

[3] S. Kaur and T. Sharma, "Efficient load balancing using improved central load balancing technique," *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 1–5, 2018, doi: 10.1109/ICISC.2018.8398857.

[4] H. Rai, S. K. Ojha, and A. Nazarov, "Cloud Load Balancing Algorithm," *Proc. - IEEE 2020 2nd Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2020*, pp. 861–865, 2020, doi: 10.1109/ICACCCN51052.2020.9362810.

[5] T. Mohmmed and N. Abdalrahman, "A Load Balancing with Fault Tolerance Algorithm for Cloud Computing," *Proc. 2020 Int. Conf. Comput. Control. Electr. Electron. Eng. ICCCEEE 2020*, 2021, doi: 10.1109/ICCCEEE49695.2021.9429597.

[6] S. Shiju George and R. Suji Pramila, "A review of different techniques in cloud computing," *Mater. Today Proc.*, vol. 46, no. xxxx, pp. 8002–8008, 2021, doi: 10.1016/j.matpr.2021.02.748.

[7] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.02.007.

[8] F. Ebadifard, S. M. Babamir, and S. Barani, "A Dynamic Task Scheduling Algorithm Improved by Load Balancing in Cloud Computing," *2020 6th Int. Conf. Web Res. ICWR 2020*, pp. 177–183, 2020, doi: 10.1109/ICWR49608.2020.9122287.

[9] S. Ghosh and C. Banerjee, "Dynamic time quantum priority based round robin for load balancing in cloud environment," *Proc. - 2018 4th IEEE Int. Conf. Res. Comput. Intell. Commun. Networks, ICRCICN 2018*, pp. 33–37, 2018, doi: 10.1109/ICRCICN.2018.8718694.

[10] K. Y. Wang, S. J. Kao, and M. T. Kao, "An efficient load adjustment for balancing multiple controllers in reliable SDN systems," *Proc. 4th IEEE Int. Conf. Appl. Syst. Innov. 2018, ICASI 2018*, pp. 593–596, 2018, doi: 10.1109/ICASI.2018.8394323.

[11] M. Kumar and S. C. Sharma, "Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment," *Comput. Electr. Eng.*, vol. 69, pp. 395–411, 2018, doi: 10.1016/j.compeleceng.2017.11.018.

[12] A. Thomas, G. Krishnalal, and V. P. Jagathy Raj, "Credit based scheduling algorithm in cloud computing environment," *Procedia Comput. Sci.*, vol. 46, no. Icict 2014, pp. 913–920, 2015, doi: 10.1016/j.procs.2015.02.162.

[13] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009, doi: 10.1016/j.amc.2009.03.090.

[14] L. W. Jia, K. Li, and X. Shi, "Cloud Computing Task Scheduling Model Based on Improved Whale Optimization Algorithm," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/4888154.

[15] B. Muthulakshmi and K. Somasundaram, "A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment," *Cluster Comput.*, vol. 22, pp. 10769–10777, 2019, doi: 10.1007/s10586-017-1174-z.

[16] K. Pradeep and T. Prem Jacob, "A Hybrid Approach for Task Scheduling Using the Cuckoo and Harmony Search in Cloud Computing Environment," *Wirel. Pers. Commun.*, vol. 101, no. 4, pp. 2287–2311, 2018, doi: 10.1007/s11277-018-5816-0.

[17] M. Hussain, L. F. Wei, A. Lakhan, S. Wali, S. Ali, and A. Hussain, "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing," *Sustain. Comput. Informatics Syst.*, vol. 30, no. January, p. 100517, 2021, doi: 10.1016/j.suscom.2021.100517.

[18] N. Gobalakrishnan and C. Arun, "A new multi-objective optimal programming model for task scheduling using genetic gray Wolf optimization in cloud computing," *Comput. J.*, vol. 61, no. 10, pp. 1523–1536, 2018, doi: 10.1093/comjnl/bxy009.

[19] Y. J. Moon, H. C. Yu, J. M. Gil, and J. B. Lim, "A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments," *Human-centric Comput. Inf. Sci.*, vol. 7, no. 1, 2017, doi: 10.1186/s13673-017-0109-2.

[20] L. Abualigah and A. Diabat, "A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments," *Cluster Comput.*, vol. 24, no. 1, pp. 205–223, 2021, doi: 10.1007/s10586-020-03075-5.

[21] B. Jana, M. Chakraborty, and T. Mandal, *A task scheduling technique based on particle swarm optimization algorithm in cloud environment*, vol. 742. Springer Singapore, 2019. doi: 10.1007/978-981-13-0589-4_49.

[22] S. Nabi, M. Ahmad, M. Ibrahim, and H. Hamam, "AdPSO: Adaptive PSO-Based Task Scheduling Approach for Cloud Computing," *Sensors*, vol. 22, no. 3, pp. 1–22, 2022, doi: 10.3390/s22030920.

[23] K. R. Prasanna Kumar and K. Kousalya, "Amelioration of task scheduling in cloud computing using crow search algorithm," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 5901–5907, 2020, doi: 10.1007/s00521-019-04067-2.

[24] M. Bansal and S. K. Malik, "A multi-faceted optimization scheduling framework based on the particle swarm optimization algorithm in cloud computing," *Sustain. Comput. Informatics Syst.*, vol. 28, no. September, p. 100429, 2020, doi: 10.1016/j.suscom.2020.100429.

[25] J. Chen, T. Du, and G. Xiao, "A multi-objective optimization for resource allocation of emergent demands in cloud computing," *J. Cloud Comput.*, vol. 10, no. 1, 2021, doi: 10.1186/s13677-021-00237-7.

[26] N. Rana, M. S. Abd Latiff, S. M. Abdulhamid, and S. Misra, "A hybrid whale optimization algorithm with differential evolution optimization for multi-objective virtual machine scheduling in cloud computing," *Eng. Optim.*, vol. 54, no. 12, pp. 1999–2016, 2022, doi: 10.1080/0305215X.2021.1969560.

[27] D. K. Sinha and G. Sinha, "Designing Energy Management Aware Task Scheduling Algorithm And Model For Cloud Data Centers Turkish Journal of Computer and Mathematics Education Research Article," vol. 12, no. 12, pp. 1008–1016, 2021.

[28] G. Patel, R. Mehta, and U. Bhoi, "Enhanced Load Balanced Min-min Algorithm for Static Meta Task Scheduling in Cloud Computing," *Procedia Comput. Sci.*, vol. 57, pp. 545–553, 2015, doi: 10.1016/j.procs.2015.07.385.

[29] S. Zhan and H. Huo, "Improved PSO-based task scheduling algorithm in cloud computing," *J. Inf. Comput. Sci.*, vol. 9, no. 13, pp. 3821–3829, 2012.

[30] A. S. A. Beegom and M. S. Rajasree, "Integer-PSO: a discrete PSO algorithm for task scheduling in cloud computing systems," *Evol. Intell.*, vol. 12, no. 2, pp. 227–239, 2019, doi: 10.1007/s12065-019-00216-7.

[31] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2020, doi: 10.1007/s12652-020-02614-7.

[32] S. S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation," *J. Supercomput.*, vol. 78, no. 1, pp. 18–42, 2022, doi: 10.1007/s11227-021-03810-8.

[33] M. M. Golchi, S. Saraeian, and M. Heydari, "A hybrid of firefly and improved particle swarm optimization algorithms for load balancing in cloud environments: Performance evaluation," *Comput. Networks*, vol. 162, p. 106860, 2019, doi: 10.1016/j.comnet.2019.106860.