DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5816

Volume 9, No. 2, March-April 2018

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

LOAD BALANCING IN CLOUD COMPUTING: A REVIEW

Sangeeta
Dept. of computer science
DCRUST, Murthal
Sonepat, India

Suman
Dept. of computer science
DCRUST, Murthal
Sonepat, India

Abstract: Cloud computing is rapidly growing due to the enormous benefits it offers over the traditional approach. Earlier, lot of things like buying server, managing traffic and maintenance needs to be managed individually leading to increase in cost and overhead for users. Cloud offers a less expensive and easy way of managing things. With increased number of applications and users, resources are not utilized efficiently This calls for efficient techniques to balance load on cloud. A good load balancing approach is required to distribute load among virtual machines and to provide maximum utilization of resource. A discussion and comparative analysis of some important approaches for balancing load in cloud is presented in this paper.

Key words: Cloud computing, load balancing, static algorithms, dynamic algorithms.

I. INTRODUCTION

Cloud computing is the use of remote servers on the internet to store, manage and process data rather than a local server or on your personal computer. With cloud you can store the data, manage the data using databases or can process data by renting a server which has larger processing capability, by this we can do our work faster. Cloud is popularly being used in many scientific and business applications. Advantages of cloud are: (1) reliability (2) better storage (3) back up (4) pay per use. There are three types of cloud models public cloud, private cloud and community clouds.

Public Cloud: When we want to share our server with host of different people. In this, services are provided by third party over internet. Public cloud may be free or renting pay per use. In an organization every employ can use the same application from any office as long as internet is working.

Private Cloud: unlike public cloud, private cloud services are limited to one company and managed by them only. Security is more in private cloud as compared to public cloud but one may have to pay more for this than private cloud [1].

Community Cloud: these models are used for a specific purpose. In this infrastructure is shared by many organizations for a common purpose, which can be managed by a third party and hosted internally or externally [2, 27].

Hybrid Cloud: hybrid cloud and combination of other clouds (public cloud, private cloud & community cloud). In this type, public and private clouds are as per requirements. Like, companies can use their own infrastructure and when requirements are high public cloud services can be opt [2].

Service Models [1]: There are different services provides by cloud like, IaaS (infrastructure as a service), PaaS (storage as a service) and PaaS (platform as a service). These services are mainly useful in scientific, business and industrial applications.

IaaS: It provides virtualized computing resources over the internet. In IaaS services related to hardware are managed by vender and need to manage application, data, storage, middleware and OS whereas other things i.e. virtualization, server, storage and networking are managed by vender itself. The objective of IaaS is to increase revenue and QoS [4].As all hardware related problems are handled by vender, customer don't need to worry about maintenance.

PaaS: In this hardware and software tools are provided by service provider. Hardware and software are hosted by provider in his own infrastructure. So there is no hassle to install hardware or software for development of application. In PaaS you need to manage only application and data all other services (storage, middleware, OS virtualization, server, storage and networking) are handled by vendor.

SaaS: SaaS advantage is, its reliable alternative server is available in case of failure. But there is a problem of lock-in where, shifting to a new serer is not allowed or you may have to pay extra. Examples of SaaS are SalesForce.com and Google mail [5, 28].

A. Load balancing

In cloud load balancing is a process of distributing the load among VMs in such a way that it increase the number of tasks execution with improving resource utilization. Load balancing allow resources to scale up and down with auto scaling in order to make more tasks successful. As cloud uses are increasing as a result workload and traffic is increasing too. Load can come in any form like memory load, delay load, CPU load [29]. Load balancing is an important requirement in uses of cloud computing. Load balancing approach should consider following: (1) makespan time (2) performance (3) scalability (4) resource utilization (5) execution time (6) user satisfaction [3]. A good balancing approach focuses on increasing resource utilization and decreasing makespan time of tasks. There are different heuristic load balancing algorithms based on load balancing i.e. min-min, min-max, MET (Minimum Execution Time), MCT (Minimum Completion Time), OLB (Opportunistic Load Balancing), Round robin and sufferage. Use of these algorithms has shown different load balancing level. Load balancing algorithms are generally classified into two parts: *Dynamic load balancing* and *static load balancing*.

- **Static load balancing** are used where load variations are less. Prior knowledge of resources is requires before starting the process, information is gathered about system resource and performance of systems. In static approach shifting of task from one resource to another is not allowed [14]. Changes in executing process are not allowed at run time. Static techniques works better when there is slight change in load. But as there are more variations dynamic is opted for load balancing than static.
- **Dynamic Load Balancing,** no prior knowledge is required about resources and works fine when there are unpredictable changes in load. Dynamic load balancing approach is based on the current state of system. Main advantage of this approach is tasks can be shift from one resource to another when load on machine is more and another one is less loaded. Although it is more complex in than static approach but it provides much better results [26].

.

2. RELATED WORK

Min-min algorithm assigns tasks to resources which have best expected execution time while arbitrary fashion is used for assigning. Before assigning tasks it doesn’t check availability of resources. Uses of min-min have shown in [5, 7-12]. In [5] min-min algorithm is used for load balancing, where some changes are made on traditional min-min algorithms and focus on makespan time, user priority and load balancing. In results they have shown decrement in completion time and improved load balancing level compare to min-min algorithm. In [7] comparison of several algorithms has been shown where min-min resulted as having minimum number of failed cloudlets (mobility-enhanced small-scale cloud datacenter).

A comparison between two algorithms is performed in [8] where min-min and max-min are used. Comparison is done in two manners, space shared manner and time shared manner which is done using simulator CloudSim. . In this comparison max –min algorithm has shown better results than min-min.

Considering makespan time and resource utilization an algorithm based on min-min called LBMM (Load Balanced Min-Max) is presented in [9]. Algorithm used secondary scheduling approach where in first scheduling, a greedy algorithm is applied. It combines largest and smallest task. together and assign them to resource with strong computing capacity. And whenever there is overloading, task are again reschedule and assign to under load resource. For independent tasks an improved min-min algorithm is proposed in [10] which focus in minimizing execution time of tasks. Comparison is done with min-min and sufferage algorithm which is showing better results with improved min-min algorithms.

Unlike min-min, max-min chose tasks which have larger execution time and schedule them first as shown in [11, 14]. Ability of a cloud service to serve on demand offerings when demands go up and down is called elasticity in cloud. An elastic cloud task-scheduling algorithm on min-max is proposed in [11]. An analysis of task list is done to estimate the number of task and their execution time. Task information is collected by a client and execution time is updated by load balancer according to task information.

A load balancing algorithm with combination of weighted Round robin and max-min algorithm is presented in [14]. Algorithm called WeightedMaxMin, which focuses on constraints as waiting time and response time. To prevent any task waiting from so long task is again scheduled by scheduler. Then scheduler can again schedule to an appropriate virtual machine. This algorithm is suited for static environment.

A dynamic load balanced algorithm is proposed in [3] with constraints like elasticity and deadline in cloud. Aim of this dynamic algorithm is to minimize makespan time and improve the number of task meet the deadline specified by client. To make this happen task are first been sorted on the basis of deadline. In each interval number of tasks which are not meeting deadline numbers of virtual machine increased. Increment or decrement depends on overload and under load situation of resources. The results are compared with min-min, FCFS and SJF algorithm where proposed algorithm showing better results.

A hybrid of two algorithms SLA aware decentralized and JIQ algorithm is proposed in [15]. This algorithm focuses on balancing load between virtual machines. Through iterations response time of virtual machines is calculated and a threshold value is created using user request and number of VMs. SLA is created by response time and comparison is performed on RSA response time and VMs. If response time of VM is less than RSA then a compatible list is created. Task is assigned to resources with the use of JIQ basic on their availability. If VMs are not available they further balanced on the basis of response time.

For heterogeneous environment, a heuristic task scheduling algorithm named HABC is proposed (heuristic Artificial Bee Colony Algorithm) in [16]. In this algorithm large tasks are given priority over small tasks which have shown better use of resources. Data is distributive in two ways, normal distribution and data distribution. This algorithm has shown better results even if number of tasks increased.

A honey bee inspired algorithm for load balancing is proposed in [24] which focus on load of VMs. Rescheduling is performed when there is situation of underload. Honey bees behavior is followed to balance load in cloud computing. Honey and food sources in honey bee algorithm are conceptualized as resources and load. Under loaded VMs are paid more attention in this particular algorithm than overloaded resources.

Genetic algorithm works on natural selection approach. Number of tasks performed in GA is selection, crossover, and mutation. Several genetic algorithms are discussed in [19-21]. A combination of genetic algorithm with double fitness adaptive algorithm is proposed called JLGA. This algorithm takes short jabs first for scheduling. For population analysis greedy algorithm is used [19]. A comparison of genetic algorithm and JLGA is also performed through simulation. But priority is not set with this algorithm. Genetic algorithm with time as priority is used in [20] and population initialization is also done based on time. Time calculation is based on the length of the task. An enhanced generic algorithm is proposed in [21] which focus on makespan time. Load variation is comparatively less as fitness function is used for allocation of resources. Results are compared with ACO, PSO where GA shown better results.

Deepak Mahapatra at el. [22] proposed a heuristic based ant colony optimization algorithm which focuses on delay, network load and CPU load. The pheromone is updated by incoming ants travelling from source to destination. However fault tolerance factor is not considered in this algorithm. An improved ACO algorithm is proposed in [23], which apart from original algorithm taking cost and time of tasks execution as main factors. Pheromone and inspired factor are improved in proposed algorithm with improving time, resource utilization and less cost

Yongfei Zhu et al. [25] proposed am algorithm based on particles swarms optimization. This is used along with red black tree for load balancing. New improved algorithm shows better results in terms of tasks solving and time than PSO. The literature survey is summarized in Table 1.

Table 1 Comparison of Load Balancing Algorithms

Algorithm	***Type***	***Focus on***	***Advantage***	***Disadvantage***
MET	Static	Execution time	Good for independent task assignment	Not suitable for grid environment
MCT	Static	completion time	Better over OLB and MET	Resource selection is poor
Max-Min, Min-Min[6-13]	Static	Expected Completion time	Better makespan than others	Starvation, QoS is not considered
OLB[7]	Static	Arriving time	Easy to understand	Poor makespan
Sufferage[10]	Dynamic	Suferrage value, completion time	Fast, less makespan, more success tasks	Not suitable for cluster type resource
JIQ+SLA[15]	Dynamic	Threshold value, response time	improved response time, waiting time and makespan time	Overloading of host
Genetic Algorithm[20]	Dynamic	Makespan, optimization	Efficient in term of makespan	No guarantee of optimal solution, complex
JLGA[19]	Dynamic	Short jobs	Good makespan	No priority set
Enhanced genetic Algorithm[21]	Dynamic	Makespan time	Load variance are less as fitness function used	More energy consumption
Enhanced bee colony algorithm [24]	Dynamic	Behavior of bees	Low VM migration	Scalability
Heuristic Artificial Honey Bee[16]	Dynamic	Makespan, file length	Low makespan even if tasks increased	Result are not stable always
Heuristic Ant colony optimization	Dynamic	Delay, network load	Efficient use of Resources	Fault tolerance factor not considered
IACO[23]	Dynamic	Cost, time, resource utilization	Improved cost and resource utilization	No priority set in selection
Improved PSO[25]	Dynamic	Efficiency, speed of task	Time complexity is better than PSO	Focuses on initial set of particles only

.

3. CONCLUSION

Load balancing in cloud computing is a critical yet important thing to manage as it ensuring efficient use of resources. Several static and dynamic algorithms are explained in this paper and it has been observed that dynamic algorithms are more efficient as compared to static algorithms. Dynamic algorithms work on current state of system whereas static algorithms require system information before starting the process. Although dynamic algorithms are more complex than static but provides better results.

REFRENCES

[1] Gupta, S., & Sanghwan, S. (2015). Load balancing in cloud computing: A review. International Journal of Science, Engineering and Technology Research (IJSETR), 4(6).

[2] Belbergui, C., Elkamoun, N., & Hilal, R. (2017, October). Cloud computing: Overview and risk identification based on classification by type. In Cloud Computing Technologies and Applications (CloudTech), 2017 3rd International Conference of (pp. 1-8). IEEE.

[3] Kumar, M., & Sharma, S. C. (2017). Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment. Computers & Electrical Engineering.

[4] Adhikari, M., & Amgoth, T. (2018). Heuristic-based load-balancing algorithm for IaaS cloud. Future Generation Computer Systems, 81, 156-165.

[5] Chen, H., Wang, F., Helian, N., & Akanmu, G. (2013, February). User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In Parallel computing technologies (PARCOMPTECH), 2013 national conference on (pp. 1-8). IEEE.

[6] Kalita, R., & Patnaik, H. (2014, May). A novel heuristic resolving deadline-oriented task scheduling in cloud. In Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on (pp. 1137-1142). IEEE.

[7] Kushwah, V. S., & Goyal, S. K. (2017, June). Performance and analysis of various fault-tolerant algorithms for cloud computing under CloudSim. In Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on (pp. 1171-1175). IEEE.

[8] Kumar, S., & Mishra, A. (2015). Application of Min-Min and Max-Min Algorithm for Task Scheduling in Cloud Environment Under Time Shared and Space Shared VM Models. International Journal of Computing Academic Research (IJCAR), 4(6), 182-190.

[9] Chen, H., Liu, Q., & Ai, Q. (2016, August). A New Heuristic Scheduling Strategy LBMM in Cloud Computing. In Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016 8th International Conference on (Vol. 1, pp. 314-317). IEEE.

[10] Bey, K. B., Benhammadi, F., & Benaissa, R. (2015, April). Balancing heuristic for independent task scheduling in cloud computing. In Programming and Systems (ISPS), 2015 12th International Symposium on (pp. 1-6). IEEE.

[11] Li, X., Mao, Y., Xiao, X., & Zhuang, Y. (2014, June). An improved max-min task-scheduling algorithm for elastic cl

[12] Ghumman, N. S., & Kaur, R. (2015, July). Dynamic combination of improved max-min and ant colony algorithm for load balancing in cloud system. In Computing, Communication and Networking Technologies (ICCCNT), 2015 6th International Conference on (pp. 1-5). IEEE.

[13] Khatavkar, B., & Boopathy, P. (2017, April). Efficient WMaxMin static algorithm for load balancing in cloud computation. In Power and Advanced Computing Technologies (i-PACT), 2017 Innovations in (pp. 1-6). IEEE.

[14] Shah, J. M., Kotecha, K., Pandya, S., Choksi, D. B., & Joshi, N. (2017, May). Load balancing in cloud computing: Methodological survey on different types of algorithm. In Trends in Electronics and Informatics (ICEI), 2017 International Conference on (pp. 100-107). IEEE.

[15] Choudhary, M., Chandra, D., & Gupta, D. (2017, May). Load balancing algorithm using JIQ methodology for virtual machines. In Computing, Communication and Automation (ICCCA), 2017 International Conference on (pp. 730-735). IEEE.

[16] Kimpan, W., & Kruekaew, B. (2016, August). Heuristic Task Scheduling with Artificial Bee Colony Algorithm for Virtual Machines. In Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems, 2016 Joint 8th International Conference on (pp. 281-286). IEEE.

[17] Madni, S. H. H., Latiff, M. S. A., Abdullahi, M., & Usman, M. J. (2017). Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. PloS one, 12(5), e0176321.

[18] Han, H., Deyui, Q., Zheng, W., & Bin, F. (2013, September). A Qos Guided task Scheduling Model in cloud computing environment. In Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on (pp. 72-76). IEEE.

[19] Wang, T., Liu, Z., Chen, Y., Xu, Y., & Dai, X. (2014, August). Load balancing task scheduling based on genetic algorithm in cloud computing. In Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on (pp. 146-152). IEEE.

[20] Makasarwala, H. A., & Hazari, P. (2016, June). Using genetic algorithm for load balancing in cloud computing. In Electronics, Computers and Artificial Intelligence (ECAI), 2016 8th International Conference on (pp. 1-6). IEEE.

[21] Sharma, Harshdeep, and Gianetan Singh Sekhon. "Load Balancing in Cloud Using Enhanced Genetic Algorithm." (2017).

[22] Mahapatra, D., Saini, G. K., Goyal, H., & Bhati, A. ANT COLONY OPTIMIZATION: A SOLUTION OF LOAD BALANCING IN CLOUD.2016.

[23] Qingbin, N., & Pinghua, L. (2016, October). An Improved Ant Colony Optimization Algorithm for Improving Cloud Resource Utilization. In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016 International Conference on (pp. 311-314). IEEE.

[24] Babu, K. R., & Samuel, P. (2016). Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud. In Innovations in bio-inspired computing and applications (pp. 67-78). Springer, Cham.

[25] Zhu, Y., Zhao, D., Wang, W., & He, H. (2016, January). A Novel Load Balancing Algorithm Based on Improved Particle Swarm Optimization in Cloud Computing Environment. In International Conference on Human Centered Computing (pp. 634-645). Springer, Cham.

[26] Suman, P. S., & Patel, R. B. (2013). User Specific Algorithm for Vertical Handoff in Heterogeneous Wireless Networks. *International Journal of Scientific & Engineering Research*, *4*(5), 676-680.

[27] Anuradha, D., & Sangwan, S. (2016). Implementing Multiple Security in the Cloud Environment.International journal of Advance research. Ideas and innovation in technology.

[28] Sangwan, S., Singh, P., & Patel, R. B. (2012). Adaptive vertical handoff in heterogeneous wireless networks. *International Journal of Data & Network Security*, *1*(3), 98-100.

[29] M Kumar, Suman & S Singh. (2018). A Survey on Virtual Machines in Cloud Computing. International Journal of Computer Science 6(3), 485-490.