International Journal of Scientific Research & Engineering Trends
Volume 3, Issue 1, Jan.-2017, ISSN (Online): 2395-566X

Cloud Allocation Strategies for Load Balancing: Review

Shalini Bairagi
Student, Dept. of Computer Science
All Saint's College of Technology, Bhopal
Email: Shalini100b@gmail.com

Prof. Vipin Verma
Dept. of Computer Science
All Saint's College of Technology, Bhopal
Email: vipin.verma26011985@gmail.com

***Abstract* – Nowadays, Cloud computing is an extensiveresearch area and researchers can see it as the future of computing. A lot of effort and time is invested to make it more efficient, scalable, reliable and fault tolerant. Load balancing is one such area that still needs to be explored to attain ultimate performance. This paper will be primarily focusing at surveying various strategies for scheduling of cloudlets to the virtual machines in such a way that no machine is underutilized or overwhelmed. If the load among VMs is balanced, a good speedup and hence improved throughput will be achieved.**

***Keywords* – Cloud Computing, Performance, Reliability, Load Balancing.**

I. INTRODUCTION

Cloud computing has become one of the most evolutionary developments in the field of IT that has completely revolutionized the way computers are used from providing high speed computing to touching the daily lives of people through social media or wearable computing. Cloud has tremendous potential that however comes with certain research challenges. Resource management has always been a matter of interest for researchers, be it at OS level or cloud level. With the growing concern of efficient utilization of resources comes the challenge of efficient task scheduling and balancing the load throughout machines. A lot of survey has already been done in this direction but researchers are everyday developing new techniques and improving our altogether vision to understand this problem.

II. SOME IMPORTANT TERMINOLOGIES

The crux of all the load balancing algorithms is to improve Quality of Service (Qos) for the user in term of response time, make span, waiting time, tardiness, fairness, priority etc. Likewise provider desired Qos include resource utilization, throughput etc. All these parameters act as deciding factors for the efficiency of an algorithm. Some of the terminologies are discussed below:

a. **Cloudlet**: Any user request is submitted as a cloudletwhich may consist of one or more tasks and these tasks

b. **Waiting time**: Total time spent from submission to thebeginning of execution.

c. **Resource utilization**: This defines the utilization rateof resources and should be as high as possible.

d. **Load efficiency**: ratio of minimal average load tomaximal average load on all machines.

e. **Constraints**: Restriction imposed on an algorithm inthe form of some defined criteria such as priority constraints, deadline constraint, budget constraint etc.

III. LOAD SCHEDULING

Apart from just providing for scheduling of tasks, the various algorithms have taken different aspects of cloud into their circumference. Load balancing is an integral part of any scheduling strategy and can be classified based on system state as static and dynamic, process initiator as that initiated by sender, receiver or symmetric. [1]. Different researchers give varied classification based on the characteristics under consideration. It is only during scheduling that effective load balancing can be done by properly distributing the load across machines.

Cloud workload scheduling considers both localized scheduling (local resources) as well as global scheduling (virtualized resources). Designing of efficient algorithms requires knowledge of not only the basic cloud architecture but all other aspects related to it such as resource demand profiling, virtualization, energy aware scheduling, network issues etc[5]. Load balancing and scheduling problems are intractable and thus many heuristic techniques have been into practice to find efficient solution. For e.g. Genetic algorithms, inspired by the human genetic belief 'survival of the fittest', have been in practice for quite a while. Figure 1 shows basic load balancing mechanism.

Basic Algorithms used in Loadbalancing :-

A. Static Environment

a. Round Robin and RandomizedAlgorithms

In this algorithm resources are assigned to the task on FCFS basics i.e. the job that arrived first will be first allocated the resources. It is one of the simplest algorithms which pass each new request to the next server. Thus this algorithm does not have status record of each connection. Also allocation order is maintained on each processor locally and it is independent of allocation from remote processor, with identical work load round robin algorithm works properly In this algorithm each node will have equal opportunity, however in public cloud the configuration and performance of each node will not same. Thus there is slightly change in Round Robin algorithm which is called Round Robin based on load degree evaluation. Randomized schemes will work well when process is

more and processor is less. This algorithm will attain the best performance as compared to other load balancing algorithms for particular special purpose application.

b. Central Manager Algorithm

In this algorithm a central processor or channel agent will select the new process. The minimum loaded processor will selected from all the processor, when process is created. Load manager will select the host for new processor and then central load manager will calculate the overall load.

c. Threshold Algorithm

In the threshold algorithm the host is selected locally without sending remote message and each processor must contain the private copy of the system load. In the threshold algorithm the level of the load can be defined as Under loaded-load<Thunder Medium-Thunder ≤ load Tupper Overloaded-load > Tupper

Initially the process will be assumed as under loaded but when the processor will exceed the load limit then it started sending message at remote level and contain copy of all the message and keep updating the entire load of the processor to get the actual status of the load on the processor.

B. Dynamic Environment

In dynamic atmosphere the cloud provider's heterogeneous resources installs and these are flexible resource. In this dynamic environment, in load changes the run time can be easily adapted by proposed algorithm. Dynamic environment can be highly adaptable in cloud computing whereas it is very hard to simulate. Based on Weighted Least connection a load balancing method in dynamic atmosphere is called ESWLC, it allocates resources with the minimum task weight according to its node capability. Based on weight and the capability of node, task is assigned to a node

Load Balancing Based On Spatial Distribution of Nodes. There can be three basic kind of algorithm that distinguishes which node is responsible for cloud computing atmosphere load balancing.

Centralized Load Balancing

In the Centralized Load Balancing technique the central load is the main factor which is responsible for storing knowledge of overall cloud network and then makes decision to apply static and dynamic approach for load balancing. Hence all the decision is made by the central node it reduces the overall time also are free from analyze whole cloud resource but it is no longer fault tolerant and also create overhead for a centralized node, In this technique the chances of failure for the centralized node is very high and the recovery will not easy in case of node failure.

a. Distributed Load Balancing

In the Distributed Load Balancing technique there is no single node responsible for monitoring the cloud network instead of multiple nodes monitors the network to make accurate load balancing decisions. Every node in the network is equally responsible for maintain the knowledge table to ensure efficient distribution of task in static environment and re-distribution in dynamic environment. In distributed scenario, failure intensity of a node is less. Hence the system is reliable and fault tolerant and none of the particular node is overloaded. [8]

b. Hierarchical Load Balancing

Hierarchical Load Balancing involves different levels of the cloud in load balancing decisions. This type of balancing is done by master slave model and it can be modeled as a tree data structure where every node is managed or balanced by its parent node. [9] Based on the information collected by the parent node, allocation and scheduling can be done. In this algorithm the root node is responsible for distributing the load to all its sub nodes.

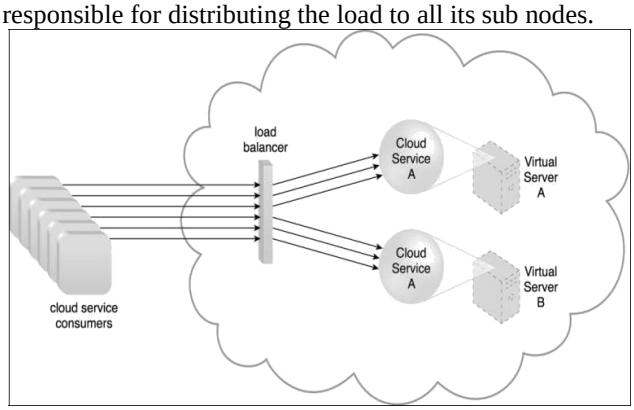

Fig.1. Basic load balancing mechanism

IV. RELATED WORK

Various surveys about load balancing have been found in literature with some differentiating algorithms into static and dynamic[13][16] and others describing few famous algorithms[17]. A complete survey of metaheuristic scheduling techniques is done by M.Kalra in [3]. It is a well known fact that scheduling is an NP-hard problem and hence finding optimal solution is very difficult. Various techniques including ACO, GA, PSO etc. have been explored and related research done in respective fields has been elaborated. Different algorithms opt for different optimization criteria but the overall goal remains the same i.e. performance improvement and getting the best possible solution. Banerjee et.al.[1] have proposed a cloudlet allocation strategy that provides scheduling of cloudlets along with load balancing but the methodology used to balance the load is straightforward using one to one mapping of cloudlets with VMs. Thamarai et.al. have proposed a load balancing mechanism that is adaptive in nature and provides effective resource allocation and release in cloud through a cloud broker. The architecture proposed comprises different entities including a request handler, controller, load balancer, scheduler, information gatherer and other monitoring units. This has certainly improved availability and scalability in cloud and results have been simulated on Eucalyptus cloud infrastructure[2].

This work[9] comprises of developing an adaptive scheduling algorithm in the form of a variation to the famous knapsack problem for hybrid environments.

Objective of this algorithm is to maximize resource utilization in private cloud and minimize rent in public cloud depending on the deadline given by consumer. Near optimality is achieved by dividing the tasks among public and private clouds based on a few steps. The foremost step is sharing of resources according to size of workload and deadline imposed by tasks. Following this, quick scheduling strategy is applied to this collection of resources. And finally the tasks are migrated among private and public clouds depending on the computational demands or deadline constraints. Another resource allocation model is given by [15] to provide mapping of tasks with nodes through a two step approach. In the first step, physical machines are assigned to VMs by using GA followed by calculation of load parameters and final allocation of tasks to

VMs in the second step. This work also doesn't consider priority of tasks in allocation model. However, it collaborates both consumer and provider needs to form indexes that affects the allocation process. This particular framework give by Sukhpal et.al.[8] is a predecessor to the work done in [14]. K-means clustering algorithm has been used to analyse and cluster workloads for assigning weights and Qos parameters. Accordingly scheduling policies are chosen for various categories of workloads. In a cloud environment, predicting customer needs poses a challenge to effectively perform scheduling process. The distinguishing work done in this paper is creation of workload analyzer framework used to categorize different types of workloads and then schedule them. Clustering of workloads is then done by k-means based clustering algorithm. Special emphasis is laid on Qos parameters to take care of consumer's requirements and accordingly resource is chosen for efficient provisioning. As a result effective resource management and scheduling is done.[14]

Kousik et.al.[10] have proposed a load balancing mechanism using GA optimizing strategy considering the fact that cloud environment and tasks keep on changing. Both tasks and machines are characterized with parameters such as number of instructions, arrival time etc for tasks and cost, MIPS for machines respectively. However this work assumes all tasks to have the same priority which may not be the case.

Instead of using single fitness function, Tingting[6] advocates the use of double fitness function that prevents premature convergence of basic GA. Load balancing is efficiently achieved along with reduced makespan on comparison with adaptive GA.[11] Implementing load balancing is always difficult particularly when done with scheduling. Multi Agent GA is another variation of GA that establishes a load balance model by analysing memory and CPU consumption of VMs. [18] has given various metrics for load balanced scheduling such as capacity makespan, load efficiency, imbalance level etc. and proposed a new algorithm OLRSA with improved results over certain famous algorithms. This work [20] basically deals with the preference given to customer on the basis of his capability to afford expense of resources. Undoubtedly, users pay for as much quantity and duration as they used the resource. Auctioning of cloud resources is done by providers. The process comprises bidding, revision of bids and support for multiple payment criteria. This helps provider in understanding the demand for resources in market. In [4], splitting and shifting of tasks to balance load among cores of the same machine. Comparing the two, shifting takes lesser time but splitting provides better load balancing. Nature based algorithms have been into practice since long. In [12], the author has studied behavior of honey bees to search and reap food which inspires load balancing process. Analogous relationship of VMs and honey bees has been identified to find overloaded VMs and distribute that load among under-utilized machines. Task migration among VMs adds to the efficiency of this algorithm and makes it even more dynamic. Resultantly throughput and waiting time are improved. To minimize waiting time of tasks, priorities have been assigned. Li et.al.[19] have studied the behavior of ants and implemented it in scheduling strategy to balance load across all the machines. Degree of imbalance has been used as the measuring criteria, showing great improvement over FCFS algorithm.

Wu et.al. have devised a scheduling algorithm based on the priority as well as completion time of tasks. Instead of directly assigning tasks to a resource, they are collectively stored in a set which is further analyzed for scheduling. The tasks are described by task length, resource required, pending time, user privilege etc. Qos and load balancing is achieved through this efficient priority based scheduling.[7]

V. COMPARATIVE ANALYSIS

The table 1 shows comparison between various Scheduling Techniques.

Table 1: Comparison of different Scheduling Techniques

Paper Ref. No	Main consideration	Tool used	Description	Limitation
[1]	Allocating cloudlets with load balancing	Cloudsim	Load balancing done by sorting the tasks and machines and calculating their load capacity	Workload has not been classified
[2]	Adaptive load Balancing mechanism	Not mentioned	Request handler, controller, scheduler, load balancer, monitor and provisioner work in integration to provide best possible solution	No support for session affinity among users requests
[6]	Double fitness function used, internode balancing	Not mentioned	Hybrid version of GA known as JLGA	Priority not assumed
[7]	Priority of tasks	Cloudsim	Queue of tasks based on priority and then scheduled with comparing makespan, average latency, load balancing	Increased complexity due to priority calculation
[8]	Qos and user requirements	Cloudsim	Analysis of workload with machine learning and then scheduling based on time, cost etc.	Very complex, only for large scale clouds
[10]	Both machine and task attributes are used	Cloud analyst	Load balancing done using famous GA	All jobs are assumed to have same priority
[11]	Neighbourhood competition, self learning using multiple agents	Not mentioned	Hybrid version of GA known as MAGA	Parameters need to be adjusted to obtain efficiency
[12]	Reduction in waiting time of tasks, dynamic task migration	Cloudsim	Load balancing using honey bee behaviour foraging methodology	Not meant for dependent tasks, priority is the only parameter
[15]	Qos improvement and efficient load balancing	Cloudsim	Mapping of VMs to PMs using GA and task allocation to VMs	Response time gets increased when number of data centres is large, priority not taken into consideration
[20]	Payment capacity of customer	Cloudsim	Pre auction and market driven open auction of resources followed by preference driven payment	Resource allocation based solely on payment capacity

VI. CONCLUSION

The paper reviews various load balancing algorithms proposed by different researchers either using meta-heuristics, nature inspired, novel techniques or hybrid versions. However it is difficult to identify one algorithm which is complete in itself but certainly some researchers have attempted to achieve excellence in their work and the results of their simulation give all the proof.

REFERENCES

[1] Banerjee S., Adhikari M., Kar S., & Biswas U. (2015). Development and Analysis of a New Cloudlet Allocation Strategy for QoS Improvement in Cloud.Arabian Journal for Science and Engineering, Springer, 40(5), 1409-1425.

[2] Somasundaram T. S., Govindarajan K., Rajagopalan M. R., & Rao S. M. (2012, January). A broker based architecture for adaptive load balancing and elastic resource provisioning and deprovisioning in multi-tenant based cloud environments. In Proceedings of International Conference on Advances in Computing (pp. 561-573). Springer India. http://dx.doi.org/10.1007/978-81-322-0740-5_67.

[3] Mala Kalra, Sarbjeet Singh, A review of metaheuristic scheduling techniques in cloud computing, Egyptian Informatics Journal, Available online 18 August 2015, ISSN1110-8665. http://dx.doi.org/10.1016/j.eij.2015.07.001.

[4] Hussain H., Shoaib M., Qureshi M. B., & Shah S. (2013, September). Load balancing through task shifting and task splitting strategies in multi-core environment. In

Digital Information Management (ICDIM), 2013 Eighth International Conference on (pp. 385-390). IEEE. doi: 10.1109/ICDIM.2013.6694040

[5] Jennings, B., & Stadler, R. (2014). Resource management in clouds: Survey and research challenges. Journal of Network and Systems Management, 1-53.

[6] 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.

[7] Wu X., Deng M., Zhang R., Zeng B., & Zhou, S. (2013). A task scheduling algorithm based on QoS-driven in Cloud Computing. Procedia Computer Science, 17, 1162-1169., ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2013.05.148

[8] Singh S., & Chana I. (2015). QRSF: QoS-aware resource scheduling framework in cloud computing. The Journal of Supercomputing, 71(1), 241-292. http://dx.doi.org/10.1007/s11227-014-1295-6

[9] Wang W. J., Chang Y. S., Lo W. T., & Lee Y. K. (2013). Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments. The Journal of Supercomputing, 66(2), 783-811. http://dx.doi.org/10.1007/s11227-013-0890-2

[10] Dasgupta K., Mandal B., Dutta P., Mandal J. K., & Dam S. (2013). A genetic algorithm (ga) based load balancing strategy for cloud computing.Procedia Technology, 10, 340-347

[11] Zhu, K., Song, H., Liu, L., Gao, J., & Cheng, G. (2011, December). Hybrid genetic algorithm for cloud computing applications. In Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific (pp. 182-187). IEEE.

[12] Krishna P. V. (2013). Honey bee behavior inspired load balancing of environments tasks in cloud computing Applied Computing, 13(5), 2292-2303., ISSN http://dx.doi.org/10.1016/j.asoc.2013.01.025.

[13] Nuaimi, K. A., Mohamed, N., Nuaimi, M. A., & Al-Jaroodi, J. (2012, December). A survey of load balancing in cloud computing: challenges and algorithms. In Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on (pp. 137-142). IEEE.doi: 10.1109/NCCA.2012.29

[14] Singh S., & Chana I. (2015). Q-aware: Quality of service based cloud resource provisioning. Computers & Electrical Engineering. Available online 25 February 2015, ISSN 0045-7906, http://dx.doi.org/10.1016/j.compeleceng.2015.02.003

[15] Liu L., Mei H., & Xie B. (2015). Towards a multi-QoS human- centric cloud computing load balance resource allocation method. The Journal of Supercomputing, 1-14. http://dx.doi.org/10.1007/s11227-015-1472-2

[16] Shoja, H., Nahid, H., & Azizi, R. (2014, July). A comparative survey on load balancing algorithms in cloud computing. In Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on (pp. 1-5). IEEE.

[17] Chaudhary, D., & Kumar, B. (2014, December). Analytical study of load scheduling algorithms in cloud computing. In Parallel, Distributed and Grid Computing (PDGC), 2014 International Conference on (pp. 7-12). IEEE.

[18] Tian, W. D., & Zhao, Y. D. (2014). Optimized Cloud Resource Management and Scheduling: Theories and Practices. Morgan Kaufmann.

[19] Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011, August). Cloud task scheduling based on load balancing ant colony optimization. In Chinagrid Conference (ChinaGrid), 2011 Sixth Annual (pp. 3-9). IEEE.

[20] Kumar N., & Saxena S. (2015). A Preference-based Resource Allocation in Cloud Computing Systems. Procedia Computer Science, 57, 104-111., ISSN 1877-0509. http://dx.doi.org/10.1016/j.procs.2015.07.375