International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

Load Optimization in Cloud Computing using Clustering: A Survey

Santosh Kumar Upadhyay[1], Amrita Bhattacharya[2], Shweta Arya[3], Tarandeep Singh[4]

[1,2,3,4] *Department of Computer Science and Engineering, Galgotias College of Engineering and Technology, Greater Noida, India.*

Abstract*—The key to existing IT establishments is virtualization. Cloud computing has been a hyped technology which is based on virtualization through which on demand computing resources can be accessed. The resources like computing power, memory, network, etc are the services actually provided by cloud over the internet. Physical servers abstracted as virtual machines forms the base for providing these services. Most concerned problem related to cloud is optimal distribution of load such that none of the VM is overloaded or under loaded. This paper discusses basics of cloud computing and existing approaches (algorithms) to optimize load on cloud servers along with a proposed work based on clustering algorithm.*

Keywords*— Cloud computing, Clustering, Load optimization, Resource allocation, Virtual machine.*

I. INTRODUCTION

Cloud computing provides services like storage, network, infrastructures, programming tools, software, hardware and other resources on demand of its users over the internet. It is a rapidly growing technology which marks its footprints from mobile phones of a common man to business deals of entrepreneurs. Anytime and anywhere services can be taken from cloud. Services provided by cloud is divided into three categories: Infrastructure as a service (IaaS), Platform as a service (PaaS), and Software as a service (SaaS). IaaS provides fundamental resources like network, operating system, servers through virtualization technology where clients can have full control over its resources. Its services are highly scalable and flexible (e.g. Amazon Web Services, Google Compute Engine, etc.). PaaS comes with platform and tools needed for developing softwares (e.g. Apache Stratos, Windows Azure, etc.). Saas is a software distribution model where actual applications are delivered to the client developed by a third party vendor (e.g. Google Apps, Dropbox, etc.).

Cloud computing environment deals with even distribution of load over the VMs whenever a client requests for services. Load in cloud is basically the user requests based on CPU (Virtual Machine) capacity, memory required which would be helpful for completion of user job, network for connection with cloud service broker, etc. Load optimization is a process of ensuring the uniformly distribution of work load on the available set of virtual machines so without burdening or under loading VMs, the running job is completed. Various algorithms have been proposed to achieve load optimization. The basic two variants of load optimization algorithm are static [1] and dynamic [1]. Static algorithms need prior configuration of the system while dynamic algorithms require real time communication with the network. In cloud, load optimization is done by the process called virtual machine migration. For Migration, there are two types of algorithms namely sender and receiver initiated. In sender initiated algorithm, the heavily loaded node or VM will initiate the process and migrates to the least loaded node and vice versa in case of receiver initiated algorithm. Load optimization algorithms are designed to attain the basic goals of cloud like cost reduction, response time enhancement, high throughput, etc.

There are some standard qualitative metrics that load optimization algorithms try to improve-

1) Resource Utilization: Optimum utilization of cloud resources should be done by the algorithm.

2) Throughput: Number of jobs completed per unit of time is throughput. An efficient algorithm gives a higher throughput value.

3) Response Time: Whenever a job is submitted, the time taken by the server to respond for the first time is response time. It should be less to attain user satisfaction.

4) Migration Time: It is the time taken to migrate or transfer load from one VM to another in case of overloading or under loading. Migration time if involved should be less.

5) Performance: Efficiency of an algorithm is justified with high performance.

6) Scalability: High scalability is preferred for load optimizing algorithms as it can easily work with changing number of VMs.

7) Fault Tolerance: In any system faults or errors should least affect the processing. This ability of an algorithm is called fault tolerance. So fault tolerance should be high to get higher performance of the algorithm.

8) Overhead: It is the extra processing or cost needed to implement the algorithm. Overhead involved should be as low as possible.

Three types of deployment models are also defined under cloud as: Public cloud, Private cloud & Hybrid cloud. These strategies help taking use of cloud services in the mode suitable to the client.

II. LITERATURE SURVEY

Load optimization has been an area of research of cloud datacentres, and its main objective is to ensure that every computing resource can effectively handle tasks quickly. Thus ultimately, the overall utilization of resource is improved. Researchers have proposed a series of static, dynamic and hybrid scheduling strategies.

Generally load optimizing strategies are classified into two categories: static load balancing and dynamic load balancing techniques. Static load balancing algorithms are commonly round robin, weight, ant-colony optimization, honey-bee foraging and so on. These static algorithms use static information which is unable to reflect dynamic load changes in the cluster of hosts effectively and also they have poor adaptive ability.

In Round robin algorithm [1], a fixed quantum of time is allocated to the job is an example of static algorithm. In case of larger jobs, it takes longer time for completion. Also some virtual machines remain overloaded and some under-loaded. In weighted round robin each node is allowed to receive specific number of requests according to the assigned weight. Virtual machines are now assigned job checking weights so they are not overloaded or under-loaded. Response time is high but still dynamic optimization is curbed. Throttled algorithm depends upon the theory of suitable search of virtual machine. The task manager makes a list of virtual machines, using the list, client request allotted to the relevant machine. Jobs have to wait for their chance here and also the throughput is low.

Weighted Round Robin Algorithm [11] is an improvement over the basic Round Robin algorithm. Here each node is allowed to receive specific number of requests according to the assigned weights to each VM. This algorithm is similar to round robin in terms of time division in circular fashion but VMs are now assigned job based on a constraint i.e.; checking of weights so that VMs are not overloaded or under loaded. Response time is high (less numerically) but still dynamic balancing is curbed.

Throttled algorithm [12] is based on the theory of suitable search. Here the search is for Virtual Machine. The task manager makes a list of all the VMs in the system along with their states as busy or ideal. Whenever a client places a request the list is searched by the load balancer for the first available and compatible VM. Based on the search result the VM is allocated for performing the task and index table is updated. If none of the VMs is available load balancer returns -1 to the datacentre. Here the jobs have to wait for their chance so starvation may occur. Throughput is low in this case as number of jobs completed in a unit time depends on search.

Efficient Throttled algorithm [11], is a generalized model which includes 3 algorithms Round Robin, ESCE (Equally Spread Current Execution algorithm) and Throttled algorithm. It is an advancement to the original Throttled algorithm in terms of data structure used to save information about the VMs. Here a Hash Map index is used instead of a simple list. So the hash map is searched for selection of a VM to allocate a job. Searching is faster than the Throttled algorithm in this case.

DCBT algorithm [19] is a hybrid approach formed by combining the methodology of Divide-and-Conquer and Throttled algorithms referred to as DCBT. This is the first approach that minimizes the total execution time of the tasks. This algorithm also considers the priority of the requests while allocating VMs. DCBT algorithm considers independent tasks and divides them on VMs and accordingly updates the table. Thus its distribution is equal on all VMs and this algorithm is 9.972% faster than throttled. The major problem here is distribution of dependent tasks and deadline constraints are not properly handled.

In Dynamic load management algorithm [16], firstly load balancer manages a symbol table for all present VMs and their status as (busy/available). It takes set of available virtual machines in a group. When a new request comes we check for best suited VM. Once the request is allocated, we remove that VM index from group of available VMs and is not considered for further requests until its assigned work is completed, so dynamic list of available VMs is searched instead of searching whole list. Its response time is better but it considers dynamic loads only. Parallel allocation and static loads mixed with dynamic loads needs to be considered to improve the algorithm.

In Genetic Algorithm [13] for load optimization the VMs are represented as unit vector *PUV* and jobs to be allocated as *JUV*. Here the basic genetic algorithm is applied. First step is population generation possible solutions are encoded into binary form called chromosomes. Now based on fitness function best fit pairs of chromosomes are selected and then mutation is done to find the best optimum solution based on fitness of the offspring. GA can give global optimum results without being trapped into local optima. Population generated is randomly chosen to crossover which is a disadvantage here, as time can be wasted over unfit chromosomes, also this algorithm can be further enhanced using better crossover and selection techniques.

Improved GA [14] using population reduction is a variant of Genetic algorithm. To select VM in order to satisfy the jobs Genetic algorithm is applied but before that population reduction is done by tournament selection method in order to find the finest resources. Further chromosomes are competed to get optimal solution. It also helps in identifying overwhelmed VMs and their replacements to selected VMs. This algorithm is better than GA as it doesn't waste its computation on randomly selected chromosomes. Selection techniques need to be taken care of to improve this algorithm.

In Honey Bee algorithm [4], scheduling and load optimization of non-preemptive independent tasks is inspired by honey bee behaviour. It is based on the intelligent foraging behaviour of honey bee swarm. The algorithm imitates the food foraging behaviour of swarms of honey bees. It performs a neighbourhood search combined with random search and can be used for both combinatorial optimization and functional optimization. Honey bees have developed the ability to collectively choose between nectar sources by selecting the optimal one. This source provides a maximum ratio of gain compared to costs. The whole decentralized decision process is based on competition among dancing bees, which guide new (naive) bees to their foraging targets. It uses task-level load optimization. It uses a distributed strategy with local decision making.

Novel Honey Bee Inspired Algorithm [15] was proposed as an improvement over Traditional honey bee algorithm. The traditional algorithm has limitations like uncertainty in quality parameters and also there is no improvement in throughput as tasks are assigned to VMs until it becomes overloaded along with it is non-pre-emptive. This algorithm overcomes its problem by selecting optimal VMs considering more than one constraints using pare to optimal solution .It also considers priorities of the tasks by working in pre-emptive manner. Pre-emptive nature of this algorithm may sometimes lead to starvation. This algorithm does not work for dependent tasks.

Stochastic Hill Climbing [2] is an optimization approach that is used for allocation of incoming jobs to the servers or virtual machines. It is simply a loop that continuously moves in the direction of increasing value, which is uphill. It stops when it reaches a peak value where no adjacent neighbour has a higher value. This variant chooses at random from among the uphill moves and the probability of selection can vary with the steepness of the uphill move.

Ant Colony Optimization [7] uses master-slave architecture with a single job tracker and several slave servers, which has been widely used in cloud computing like Google's MapReduce and Hadoop. The type of network topology is based on the master-slave architecture and the cloud platform. In master-slave architecture, a job is first submitted to a master node by the user. Then it is divided into several sub-tasks in the master node that are executable in nature. And then the generated tasks are distributed to different slave nodes. Then the tasks are executed in the slave nodes separately in coordination with the master node, and the results are then returned. Ultimately, the distributed results are compiled in the master node and sent to the user. Further, the master node is responsible for overall monitoring of all the steps and re-executing the failed tasks. It is also possible that during this process, the uneven distribution of tasks may cause some slave nodes in less loaded conditions while others are in heavy loaded conditions. In this case, load optimization operation should to be carried out dynamically for the cloud platform in order to keep the platform stable and operating efficiently.

Particle Swarm Optimization [5] uses live virtual machine migration. It is a technique for achieving system load optimization in a cloud environment by transferring an active virtual machine from one physical host to another. It has been proposed to reduce the downtime for migrating overloaded machines. It is achieved by only transferring extra tasks from an overloaded virtual machine instead of migrating the entire overloaded virtual machine.

LB-BC algorithm [17] involves optimization of load using Bayes and clustering. This algorithm tries to overcome problems in VM migration algorithm of time wasted in transferring loads and downtime of the VMs. Here set of physical hosts and task requests is maintained in terms of CPU resource and memory they possess. Now the task request with maximum demands is considered as performance constraint based on which new set of optimized physical hosts are filtered. Posterior probability using bayes theorem is evaluated of each optimized physical host. Now clustering of the hosts is done taking three attributes CPU resource, memory and posterior probability. The maximum posterior probability is used to measure the similarity degree between the optimized set of physical hosts to form cluster. Now the task requests are deployed on the clustered hosts. It is a heuristic approach which gives long term optimization of the system in contrast load optimization in a single cycle.

In Cluster based load balancing algorithm [18] the system is divided in master and slave nodes. Here the network is divided into clusters. Initially a cluster consists of a single node. Whenever a new node (VM) is added to the system, it is either added to one of the existing cluster or a new cluster is defined for it. Every cluster consists of a master node called the ICC (Inter Cluster Communication node) which maintains the load distribution information among the slave nodes in its cluster. The slave nodes are the actual computing elements which are connected to only one master. Whenever requests arrive they are first distributed among the masters then further masters distribute the load among its slaves. The load distribution and optimization is done based on the parameter called performance factor. This algorithm uses round robin algorithm for distribution of load among the slaves.

K-Means clustering [3] of virtual machines is also done in the cloud environment. All the cloudlets given by the user are divided into clusters depending upon client's priority, cost and instruction length of the cloudlet. The virtual machines inside the hosts are also classified into multiple clusters depending upon the characteristics. Compared with the other load optimization algorithms, it has outperformed them according to the experimental results.

III. COMPARATIVE STUDY

TABLE I Comparison of Algorithms

Scheduling Algorithm	Parameters/Metrics				
	Type	Over head	Scalability	Throughput	Response Time
Round Robin	Static	High	High	High	Yes
Ant Colony	Static	Low	High	Low	No
Honey Bee	Static	Low	Low	Low	No
K-mean	Dynamic	High	Low	High	No
Genetic	Dynamic	Low	High	Low	No
Min-Max	Static	High	Low	High	Yes
Opportunistic Load Balancing	Static	Low	Low	Low	No

TABLE II Features of Cloud Simulators

Stimulator	Features			
	Language	Type	GUI Support	Stimulation Time
CloudSim	Java	Open Source	Limited	Second
GreenCloud	C++	Open Source	Limited (Support via Nam)	Minute
iCanCloud	C++	Open Source	Not Limited	Second

IV. PROPOSED WORK

We propose a system of K-mean clustering which is used to create the clusters for effective allocation of tasks. For all clusters centroids are calculated based on multi-objectives. A list is maintained for all clusters stating the maximum and minimum resource capacity of the each cluster. The step-wise approach for the proposed work is given as follows:

STEP 1: Initialize all VMs with their specific resource types, capacities of each resource and status of VMs.

STEP 2: Cluster the n VMs into K clusters using K-means clustering

STEP 3: Cloud controller receives a new request

STEP 4: Cloud controller queries appropriate node controller or load balancer for next allocation.

STEP 5: Load balancer scans the range specifier list of clusters to see that which cluster can handle the incoming request.

STEP 6: Load balancer assigns request to the appropriate VM of the chosen cluster from the list of cluster members which will match the specific demands of the task and whose status is AVAILABLE. In case more than one VM may satisfy this, then the first one which is found will get the task.

STEP 7: Remaining resource quantities of that VM in the VM list of that cluster is updated.

STEP 8: $STATUS_{VM}$= BUSY from AVAILABLE

STEP 9: After processing requests, $STATUS_{VM}$=AVAILABLE

STEP 10: Go to STEP 3 unless no more requests arrive.

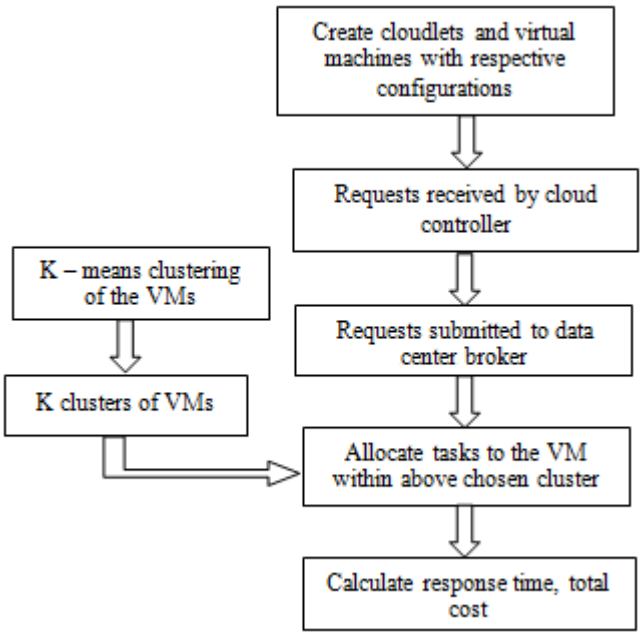

Figure 1. Flow diagram of proposed algorithm

V. CONCLUSION

Cloud computing being the most advancing and useful technology it is required to tackle the possible issues regarding its proper working and implementation. Load optimization is one of the major problems existing in cloud system. Various algorithms have been proposed in cloud literature which try to handle this issue in order to get higher throughput, efficient resource utilization, low response time, etc. This paper discusses and surveys various existing load optimization algorithms in cloud computing and also proposes a K-mean clustering based algorithm based on our research work.

REFERENCES

[1] A. Aditya, U. Chatterjee and S. Gupta, A Comparative Study of Different Static and Dynamic Load Balancing Algorithm in Cloud Computing with Special Emphasis on Time Factor, International Journal of Multidisciplinary and Current Research, pp. 141-149, 2015.

[2] B. Mondal, K. Dasgupta, P. Dutta, Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach, Elsevier Publications, pp. 783-789, 2012.

[3] B. Adrian, L. Heryawan, Analysis of K-means Algorithm for VM Allocation in Cloud Computing, IEEE transactions, pp. 48-53, 2015.

[4] Dhinesh Babu L.D., P. Venkata Krishna, Honey bee behavior inspired load balancing of tasks in cloud computing environments, Elsevier Publications, pp. 2292-2303, 2013.

[5] F. Ramezani, Jie Lu, F. K. Hussain, Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization, Springer Publications, pp. 739-756, 2014

[6] J. Zhao, L. Hu, Y. Ding, G. Xu, and M. Hu, A heuristic placement selection of live virtual machine migration for energy-saving in cloud computing environment, PloS One, vol. 9, no. 9, p. e108275, Sep. 2014.

[7] R. Gao, Juebo Wu, Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization, Future Internet Publications, pp. 445-483, 2015

[8] Zhen Xiao, W. Song, and Qi Chen, Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment, IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 6, pp. 1107-1117, 2013

[9] Bhasker Prasad Rimal, Eummi Choi, Lan Lump (2009), A Taxonomy and Survey of Cloud Computing System, 5th International Joint Conference on INC, IMS and IDC, IEEE Explore 2527Aug 2009, pp. 44-51

[10] Kansal.N. J and Chana.I, Cloud Load Balancing Techniques: A Step towards Green Computing, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012.

[11] Geetha Megharaj, Dr. Mohan K.G., A Survey on Load Balancing Techniques in Cloud Computing, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. I (Mar-Apr. 2016), PP 55-61

[12] M. Randles, D. Lamb, and a. Taleb-Bendiab, A Comparative study into distributed load balancing algorithms for cloud computing, IEEE 24th nt. Conf. Adv. Inf. Netw. Appl. Work., pp. 551–556, 2010.

[13] Kousik Dasguptaa, Brototi Mandalb, Paramartha Duttac, Jyotsna Kumar Mondald, Santanu Dame, A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing, International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013 .

[14] Ronak R Patel,Swachil J Patel,Dhaval S Patel,Tushar T Desai, Improved Ga Using Population Reduction For Load Balancing In Cloud Computing, 2016 Intl Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India.

[15] Chandrakanta Korat1, Piyush Gohel2, A Novel Honey Bee Inspired Algorithm for Dynamic Load Balancing In Cloud Environment, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 8, August 2015.

[16] Bhawna Mallick, Reena Panwar, Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm, 2015 International Conference on Green Computing and Internet of Things (ICGCloT).

[17] Jia Zhao, Kun Yang, Senior Member, IEEE, Xiaohui Wei, Member, IEEE, Yan Ding, Liang Hu, and Gaochao Xu, A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment, IEEE Transactions on Parallel and Distributed Computing, Vol. 27, No. 2, February 2016.

[18] S. K. Dhurandher, M. S. Obaidat, I. Woungang, P. Agarwal, A. Gupta, A. and P. Gupta, A Cluster-Based Load Balancing Algorithm In Cloud Computing, Proc. IEEE International Conference on Communications (ICC), 2014, 2921-2925.

[19] Shridhar G.Domanal and G. Ram Mohana Reddy, Load Balancing in Cloud Environment using a Novel Hybrid Scheduling Algorithm referred as DCBT, 2015 IEEE International Conference on Cloud Computing in Emerging Markets.

[20] Rahul Malhotra, Prince Jain, Study and Comparison of CloudSim Simulators in the Cloud Computing, The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Vol. 1, No. 4, September-October 2013.