Advances in Computer Science and Information Technology (ACSIT)
Print ISSN : 2393-9907; Online ISSN : 2393-9915; Volume 2, Number 7; April – June, 2015 pp 87 – 91
Krishi Sanskriti Publications
http://www.krishisanskriti.org/ACSIT.html

A Survey on Dynamic Load Balancing algorithms in Cloud Computing

Surbhi Kapoor

M.tech, Department of Computer Science and Technology,
Jaypee Institute of Informaion and Technology, Noida
E-mail: surbhikapoor0509@gmail.com

Abstract: *Cloud Computing is amongst the latest emerging paradigms in IT sector where services are provided over the internet to the user on demand. Load balancing is one of the most challenging areas in cloud computing. The primary concern is to distribute load efficiently and effectively among virtual machines so as to reduce response time of tasks. Various static and dynamic load balancing algorithms have been developed to address this problem. The static ones are easier to implement and are relatively less complex but are not suitable for a cloud environment where the number of tasks being put up by the end users and the requirements of the tasks cannot be known before hand, so load on cloud servers changes dynamically. However dynamic load balancing algorithms although are complex but still better suited for cloud atmosphere. In this paper we survey various dynamic load balancing algorithms that have been proposed to solve the problem of load balancing in cloud. A discussion and comparison of these algorithms has been done to give a better insight into the pros and cons of the algorithms.*

1. INTRODUCTION

Cloud computing can be defined as a network of remote servers being hosted over the internet for storage, management and processing of data. Cloud is a conventional term for anything that incurs delivering hosted services over the internet. These are broadly divided into three categories: Infrastructure as a service, Platform as a service, and Software as a service.

A Cloud service has three different characteristics. The first is that it’s a pay as you use model which means that it is sold on demand, the second one is its elasticity which means that user can have as much of a service or as little of a service as they want and the third characteristic is that it is entirely handled by the provider. Load balancing is one of the most challenging areas in cloud computing. With the increase in number of cloud users, the load on servers of cloud is also increasing. It has been reported that from 2012 to 2017, data centres workload will grow 2.3-fold whereas cloud workload will amplify 3.7 fold. So balancing load conveyed to the servers of cloud is one of the key concerns in cloud computing.

Load balancing is basically dividing the amount of work among the servers so that more work can be done in same amount of time and therefore all users get served faster. It targets to optimize throughput, maximize resource usage, reduce response time to minimum possible value and avoid overloading of any single resource. Load balancing is the necessity of cloud computing.

Load balancing algorithms can be either static or dynamic. Static algorithms are suitable for homogeneous and stable environments whereas Dynamic algorithms are more flexible and take into account different types of attributes in the system both prior to and during run-time. These algorithms can adapt to changes and can provide better results in heterogeneous and dynamic environments like that of cloud.

Many researchers have proposed dynamic load balancing algorithms. In this paper we will be reviewing some of such algorithms and a comparison of these algorithms will be done.

The rest of the paper is organized as follows. In section 2, some of the existing dynamic load balancing algorithms will be reviewed. After that, we discuss and compare the relevant approaches in section 3.

2. DYNAMIC LOAD BALANCING ALGORITHMS

Dynamic load balancing algorithms are better suited for cloud computing environment as compared to static ones because the former ones keep in mind the prior information about nodes like available bandwidth, memory, processing power and so on, before assigning load as well as the run time changes in these parameters are also considered whereas the static ones keep in mind only the prior information about these parameters before assigning tasks to them. The run time changes are not considered by static load balancing algorithms. This may lead to a particular node getting overloaded while some other nodes may get underutilized. So in our paper we are focusing on dynamic load balancing algorithms for cloud environment.

In [1], the goal is to find such an algorithm that considers priorities of users to assign their tasks. Chen Proposed an algorithm called PA-LBIMM (user Priority Aware Load Balanced Improved Min Min) scheduling algorithm which takes the characteristic of min-min scheduling algorithm as foundation to minimize the completion time of all resources and improve the load balance factor. It divides the user submitted tasks into two categories, VIP user tasks and ordinary tasks such that VIP tasks are executed first and are assigned to only VIP resources. After all the VIP tasks are assigned, then ordinary tasks are scheduled to both VIP and ordinary resources. After all the tasks get assigned, load balancing is then done by selecting the task with minimum completion time from the most heavily loaded resource. Now the completion time of this task is calculated on all the other resources and a minimum value is obtained. If this minimum value of completion time is less than the makespan of the tasks, then the particular task is reassigned to that resource. This continues until no other task on the most heavily loaded resource with the minimum completion time needs rescheduling. Since cloud is a pay according to use model, so issuing tasks on the grounds of user priorities is a good idea for scheduling the load.

In [2], the author introduced a Periodic_Ant Colony Optimization based scheduling algorithm (PACO) that uses the basic ant colony optimization algorithm in cloud computing and improves its pheromone update policy by including a periodic strategy. Pheromone intensity is a factor that gives the load assigned to a particular resource. In basic ACO, when a particular resource is assigned to a task, its pheromone intensity will get increased. This will increase the chances of that resource being selected in future for other tasks and hence load on that resource will be increased gradually. This particular drawback is overcome by PACO algorithm in which if a particular resource is selected, then its pheromone intensity gets reduced which will gradually lower the chances of selection of that resource by other tasks. If pheromone intensity of a resource gets reduced to a minimum threshold value, then that particular resource will turn off. This way if all the resources are turned off, then this leads to the end of a scheduling period and the number of tasks being assigned in that period are recorded. Remaining tasks will be assigned in new scheduling periods. In the end, if the scheduling scheme is best by far, then the pheromone intensity of that resource gets increased which increases the chances of selection of that resource by ants in future. Thus, ants get to select the better scheme. In this way, a periodic scheme is introduced. The author has compared his strategy with min-min algorithm on the basis of resources load and makespan of tasks has shown that PACO performs better than min-min.

A LBACO (Load Balancing Ant Colony Optimization) algorithm is proposed in [3] that will balance load in a cloud environment using Ant colony Optimization algorithm as the foundation. It reduces the makespan of a given tasks set and balances the entire system load. The algorithm initializes the pheromone of all VMs on the basis of number of processors, their capacity and the bandwidth factor and places the ants on VMs randomly. It then choses the next VM for an ant on the basis of probability which is computed as the ratio of pheromone of that machine to the ratio of pheromone of all machines. The one with the maximum probability among neighbouring VMs is chosen as the next VM by the ant. When an ant completes the tour, the pheromone is updated locally and if the solution obtained is current optimal solution, then global updation of pheromone is done. The algorithm continues as long as iterative condition satisfies. For each iteration, all the ants complete their tour. This algorithm is simulated in CloudSim and the results have been compared with FCFS and ACO on the basis of load imbalance and makespan of the tasks and it has been shown that the proposed approach performs better than these algorithms.

An SLA-aware two level decentralized load balance architecture (tldlb) has been proposed by Li in [4] which focuses on reducing the SLA violation rate. SLA is the service level agreement between the service provider and the end user. The two levels in the architecture are the global load balancer and the local load balancer which is SLA aware. The local balancer keeps track of the load of the VMs in its virtual zone and share this information with the corresponding global load balancer. The local load balancer choses VMs for the current task using nn-dwr (neural network based dynamic weighted round robin) algorithm which is also being proposed by the author. If the current working VMs can’t bear the load, then local load balancer will generate VMs from the spare VM pool. If there is no VM available even in the spare VM pool to serve the incoming requests, then corresponding global load balancer will be informed by the local load balancer. The global load balancers are connected to each other via P2P connections. It will then forward the requests to another lightly loaded virtual zone of some other global load balancer. The comparison of proposed nn-dwr has been done with other algorithms and it is 1.49 times faster than Artificial neural network based algorithm, 1.86 times faster than weighted round robin and 1.21 times faster than capacity based load balancing algorithm

The author in [5] Proposed an Agent Based Dynamic Load Balancing (ABDLB) algorithm in cloud computing. The proposed approach focuses on two factors one is load balancing of all the servers and second is reducing the CPU time units being consumed. In this algorithm, the mobile agent which is a software program is responsible for balancing the load being put on cloud servers. The mobile agent takes two walks. In the first one, it finds the total number of jobs assigned to each server. Using this, average number of jobs is calculated. This average is then used to assign a status to each server. If the total number of jobs being assigned to a server is more than the average number of jobs, then status of sever is overloaded and if the number of jobs assigned to a particular server is less than the average number of jobs, then server is considered to be underloaded. In the second walk, the agent transfers the jobs from the overloaded servers to the underloaded servers. This particular algorithm depends on a single mobile agent responsible for entire load balancing. The light weight mobile agent moves from server to server to collect the load information. This will not affect the network load too much. This scheme has been compared with a centralized server based load balancing policy on the basis of throughput, CPU time unit consumed and average waiting time and ABDLB performs better for all the factors.

Jiang[6] proposed a Predictive Dynamic Load Balancing algorithm that considers into account service type demanded by the user. Most of the traditional load balancing algorithms monitor load on the servers periodically and assigns requests to the servers on the basis of periodic load monitoring. However load between monitoring intervals can change. This proposed algorithm overcomes this advantage by predicting resource utilization between monitoring intervals. Also the type of service has been considered by the algorithm so that resource overhead of a server is computed separately for CPU services, memory services, disk I/O services and network bandwidth services. Consequently, the following two parameters being introduced in the paper surely improve the response time and throughput of the system.

A Dynamic Compare and Balance Algorithm (DCABA) has been proposed by Sahu[7]. Other load balancing strategies focus only on balancing load based on CPU usage, RAM usage and bandwidth usage in physical servers but this proposed algorithm also decreases the number of current active servers so as to support green computing concept and also to reduce the cost of cloud providers. In this algorithm, load on the host machines in cloud is evaluated dynamically in terms of total capacity of the server. The total capacity limits of the host machine will be the host limit. On multiplying the host limit by weight coefficients, upper and lower threshold values are computed. The weight coefficients are computed by the cloud provider on the basis of dynamic behaviour of services and applications. If the load is greater than a predetermined upper threshold value, then the extra load is transferred to other suitable host machines. On the other side, if load on host machine is lesser than lower threshold value, then server consolidation is applied which means switching off this particular server and transferring its entire load to other suitable host machines. This is done to save energy being consumed up in the cloud system. Due to this server consolidation approach being followed, there is a trade-off existing between throughput of the system and the energy being consumed. It has some working challenges like selection of appropriate threshold values and selection of migration policies.

Ghafari[8] proposed a Bee-MMT algorithm (Artificial Bee Colony algorithm- Minimal Migration Time) that makes use of bee colony algorithm to detect over utilized hosts and then using MMT selection for VMs, it decides which VMs can be shifted from the over utilized hosts. It then also finds the underutilized hosts and shifts all VMs assigned to these hosts to the other suitable hosts and then moves these hosts to switch off mode to save energy. Like the DCABA algorithm proposed in [7], this approach also cannot entirely solve the trade-off between response time of the tasks and energy consumption. This proposed algorithm can get greater power consumption than other algorithms thus supporting green computing concept. Also it is noteworthy that Bee-MMT has very less number of VM migrations. SLA violation has been used to evaluate the performance of the algorithm and author has introduced two metrics to calculate SLA violation. These are SLATAH (SLA violation time per active host) and PDM(performance degradation due to migration). The proposed algorithm has been simulated using CloudSim toolkit and results are compared with other algorithms on the basis of SLA violation, SLATAH ,PDM and VM migration. It has been shown that Bee-MMT can reduce the power consumptions and but its performance is weaker than other algorithms in case of SLA violation. However the ration of increase in SLA violation is much lesser than ratio of decrease in power consumption.

Wenhong [9] proposed the DAIRS (Dynamic and Integrated Resource Scheduling) algorithm. Unlike other traditional algorithms, this algorithm not only considers CPU load but also takes into account memory and network bandwidth load for calculating the load on a particular machine. Four types of queues are maintained by the algorithm. These are waiting queue containing requests that are not allotted immediately but are waiting, request queue containing the requests that are new, optimizing queue is for those tasks that need to be reallocated and delete queue keeps the tasks whose end time is pending. New tasks are accepted from the queues on the basis of priority of the queues where priority list is such that waiting queue is having the highest priority, then is the request queue, then optimal queue and then delete queue with the lowest priority. The allocation algorithm which is being used for assigning tasks from the queues to the servers firstly sorts the servers in increasing order of utilization and then it divides the utilization of physical servers into multiple intervals. The comparison of DAIRS algorithm has been done with three other algorithms that are ZHCJ, ZHJZ and rand algorithm and it hs been shown that DAIRS performs better than others on terms of average imbalance level of a cloud datacentre, average imbalance value of each server and running time of the algorithms.

3. DISCUSSION AND COMPARISON

In this section we will be discussing and comparing various algorithms being mentioned in section. Table I gives a comparison among the discussed load balancing algorithms on the basis of their pros and cons. The PA-LBIMM considers the priorities of the users whether they are VIP or ordinary. This is an advantage because cloud is a pay-per-use model, so giving priority to the users as they pay will be beneficial for the reputation of cloud vendor. However the drawback is that dependencies of the tasks, their deadlines, the geographic location of tasks and resources, factors like these have not been considered in this algorithm. Whereas one of the other algorithm discussed i.e. nn-dwr when applied on the two level decentralized load balancing architecture (tldlb) keeps in mind that SLA should not be violated. Thus response time, throughput or we can say deadlines of tasks are considered in this algorithm. Another advantage of this algorithm is that the architecture is decentralized. This avoids the problem of central point of failure. But the drawback with this approach is that transfer of information between the two levels of architecture may consume a lot of network bandwidth. DCABA also has the same disadvantage that it adds congestion to the network and consumes a lot of network bandwidth because each host periodically broadcasts its load to all the other hosts. Another drawback is that the energy consumption factor added will affect the quality of service provided to the user. Its advantage is that besides balancing load, it also focuses on decreasing the number of active servers so as to reduce power being consumed and support green computing. The same advantage is also found in Bee-MMT where if underutilized hosts are found, then if possible all the VMs of this host are migrated to other hosts and it is turned to sleep mode. This way it also supports green computing. But unlike DCABA, Bee-MMT also focuses on reducing SLA violation. Another advantage is that it reduces the number of VMs being migrated. It's drawback is that still a trade-off exists between power consumption and SLA violation. One of the other dynamic load balancing algorithms discussed is ABDLB (Agent Based Dynamic Load Balancing) algorithm has a light weight agent that can move from server to server carrying the load information without much affecting the bandwidth or the load of the network. However the scheme is centralized. If the mobile agent fails, the entire setup of load balancing fails. Also the time being consumed to balance load with this approach is a drawback. The two ant colony optimization based approaches have been discussed that are LBACO and PACO. LBACO algorithm reduces the makespan of a given task set but it does not consider the dependencies of the tasks. Also the pheromone update strategy of LBACO is such that it can increase the load on a particular node. This drawback is not in PACO approach. The pheromone update strategy of PACO fairly distributes the load among the nodes. But the problem of task dependencies is not addressed in this algorithm too. PACO is the first periodic scheduling strategy. While most of the other algorithms discussed consider only CPU utilization for measurement of load, the DAIRS algorithm also consider network bandwidth and available memory. However additional overhead has been added due to maintenance of four different kinds of queues. Another challenge is to set threshold values to judge over or under utilization of CPU, network bandwidth and memory.

4. CONCLUSION AND FUTURE WORK

Load balancing is a prime area of concern in cloud computing. There is a need to balance this load so that the user's tasks are completed on time. Various researchers have come up with their ideas for load balancing in cloud. Static algorithms are used in an environment where there is prior knowledge of VM's capacity before execution starts whereas dynamic algorithms keep track of capacity of VMs during runtime. So such algorithms although are more complex but still better fit

Table 1

Algorithms	Pros	Cons
PA-LBIMM	• Considers the priorities of the user	• Deadlines of the tasks, their dependencies, geographic location of tasks and resources has not been considered.
LBACO	• Minimizes the makespan of a given task set.	• Pheromone update strategy being used can accumulate the load on a particular VM • Dependencies of the tasks has not been considered
PACO	• First algorithm to introduce periodic strategy • Good Pheromone update policy has been used which avoids accumulation of load on a particular VM	Does not consider task dependencies

nn-dwr	• Reduces SLA violation rate • Considers deadlines of the tasks • Decentralized architecture, so no central point of failure	• Consumes a lot of network bandwidth
DCABA	Supports green computing	• Consumes a lot of network bandwidth • QOS provided to the user is not good
Bee-MMT	• Reduces SLA violation • Less number of VMs being migrated • Reduces energy consumption	• QOS provided to the user is still not good
ABDLB	• adds very less congestion to the network	• Centralized approach • Takes up a lot of time
DAIRS	• Considers not only CPU utilization but also memory and network bandwidth for computation of load	• Setting threshold values is an issue • Causes a lot of overhead

to a cloud environment which is dynamic. We have reviewed various dynamic load balancing algorithms in this paper. We have compared these algorithms using their pros and cons. User priority, reduced response time, energy conservation, service differentiation are some of the factors which have been considered in the reviewed algorithms. In some algorithms that focus on energy conservation, there is still a trade-off between energy consumption and the response time of the tasks. In future, such dynamic load balancing algorithms can be developed for cloud computing which completes the tasks within their deadlines and also reduces energy being consumed tasks.

5. REFERENCES

[1] Huankai Chen; Wang, F.; Helian, N.; Akanmu, G., "User-priority guided Min-Minscheduling algorithm for load balancing in cloud computing," *Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on* , vol., no., pp.1,8, 21-23 Feb. 2013

[2] Weifeng Sun; Ning Zhang; Haotian Wang; Wenjuan Yin; Tie Qiu, "PACO: A Period ACO Based Scheduling Algorithm in Cloud Computing," *Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on* , vol., no., pp.482,486, 16-19 Dec. 2013

[3] Kun Li; Gaochao Xu; Guangyu Zhao; Yushuang Dong; Wang, D., "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization," *Chinagrid Conference (ChinaGrid), 2011 Sixth Annual* , vol., no., pp.3,9, 22-23 Aug. 2011

[4] Chung-Cheng Li; Kuochen Wang, "An SLA-aware load balancing scheme for cloud datacenters," *Information Networking (ICOIN), 2014 International Conference on* , vol., no., pp.58,63, 10-12 Feb. 2014

[5] Grover, J.; Katiyar, S., "Agent based dynamic load balancing in Cloud Computing," *Human Computer Interactions (ICHCI), 2013 International Conference on* , vol., no., pp.1,6, 23-24 Aug. 2013

[6] Jiawei Jiang; Haojiang Deng; Xue Liu, "A predictive dynamic load balancing algorithm with service differentiation," *Communication Technology (ICCT), 2013 15th IEEE International Conference on* , vol., no., pp.372,377, 17-19 Nov. 2013

[7] Sahu, Y.; Pateriya, R.K.; Gupta, R.K., "Cloud Server Optimization with Load Balancing and Green Computing Techniques Using Dynamic Compare and Balance Algorithm,"*Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on* , vol., no., pp.527,531, 27-29 Sept. 2013

[8] Ghafari, S.M.; Fazeli, M.; Patooghy, A.; Rikhtechi, L., "Bee-MMT: A load balancing method for power consumption management in cloud computing," *Contemporary Computing (IC3), 2013 Sixth International Conference on* , vol., no., pp.76,80, 8-10 Aug. 2013

Wenhong Tian; Yong Zhao; Yuanliang Zhong; Minxian Xu; Chen Jing, "A dynamic and integrated load-balancing scheduling algorithm for Cloud datacenters," *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on* , vol., no., pp.311,315, 15-17 Sept. 2011