VOL. 10, NO. 10, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ADAPTIVE LOAD BALANCING ALGORITHM USING MODIFIED RESOURCE ALLOCATION STRATEGIES ON INFRASTRUCTURE AS A SERVICE CLOUD SYSTEMS

Lavanya M., Sahana V., Swathi Rekha K. and Vaithiyanathan V.
School of Computing, SASTRA University, Tamilnadu, India
E-Mail: m_lavanyass@ict.sastra.edu

ABSTRACT

Cloud computing is set of resources and services offered by the Internet. Cloud computing provides its consumers to access virtualized hardware, scalable, distributed and software infrastructure over the internet. Load balancing is the method which is used to distribute the task among multiple computers. Hence overload can be avoided and can achieve minimum data processing time, optimal resource utilization and minimum average response time. The existing Throttled Load Balancing (TLB) algorithm exhibits some drawbacks. To overcome the drawbacks in TLB this paper proposes an efficient scheduling algorithm, which can support the load balancing and can provide better improved strategies through efficient job scheduling, modified resource allocation techniques and reduce the power usage and context switching between the servers. Enhanced Throttled Load Balancing Algorithm provides improved results to Throttled Load Balancing Algorithm.

Keywords: cloud computing, virtual machine, global queue.

INTRODUCTION

Cloud computing is a model for on-demand network access to a shared pool of configurable computing resources that can be rapidly supply and release with minimal management effort or service provider interaction. Cloud computing provides its consumers to access hardware, software, distributed storage over the internet. Load balancing is a mechanism which is used to allot the work among different computers for the selection of virtual machine. Load balancing assures that there is approximately an equal amount of work among all the processors at any instance of time.

In the existing load balancing algorithms, there are some drawbacks like increase in waiting time, context switching and increase in response time. Because of the increase in waiting time the cost also increases. As a result turnaround time also increases. In Equally Spread Current Execution Algorithm, the main drawback is scanning the queue frequently. This results in additional computational overhead. In throttled load balancing algorithm the major drawback is that the index table is scanned again and again until or unless the particular virtual machine is available for allocating the resources. The overload is a major issue in data center. To overcome the drawbacks in load balancing algorithms, this paper proposes Enhanced Throttled Load Balancing Algorithm which supports the load balancing and can provide better results. This algorithm provides efficient throughput with less turnaround time and low waiting time.

RELATED WORKS

Load balancing is the method which is used to distribute the task among multiple computers for selecting the virtual nodes for execution. Hence overload is avoided and can achieve minimum average response time. Paper [1] proposes a new heuristic technique Priority-based on Deadline and Size (PDS) to maximize the throughput of the overall model by affording quick response to the tasks which prioritized based on deadline but it not discussed about the task scheduling with similar deadline . To minimize the operational costs [2] proposes to consider electricity price while load balancing. It also come up with distributed algorithms, which reduces the sum of both energy and delay cost. Social impact cost was defined as the measure for environment impact for the data center. By considering the availability of renewable energy and directing load to the data center, environmental impact has been reduced on the datacenter.

Conti *et al*. [3] proposes operational costs can be minimized by allocating more load to the data centers which consumes low electricity. Using Protocol level mechanisms, load balancing is achieved. [4] Cloud computing has a capacity to maintain internet power and network for using remotely available resources, thus provides the solution for the cost effectiveness. In order to service the heavy loads Liu *et al*. [5] expanded the model proposed in [2] to subtract locally generated clean energy from energy cost calculation. In addition to this, many proposals use locally generated clean energy.

Mathew *et al.* [6] Proposes an algorithm for reducing the energy consumption by controlling various machines in cloud. It maintains enough servers which are used to hold the current requests and spare capacity to handle the spikes in traffic in all datacenters. The performance of the load balancing algorithm is described in [7] in which request time is same but differs in cost calculation for Throttled Load Balancing algorithm. Technique of Order Preference by Similarity to Ideal Solution (TOPSIS) method is proposed in [8], which finds the suitable physical machines in the data center for the migrated virtual machines. With less VM migration load balancing is achieved in large scale cloud computing. Liu *et al*. [9] attempts to find the less cost for passing a particular volume of way over the internet.

Genetic algorithm based method with migration of virtual machines is proposed in [10] to reduce the carbon footprint of the cloud. While determining where to route load, we use carbon intensity data instead of weather data. In paper [12] Unsurpassed Composite Algorithm (UCA) algorithm is proposed to schedule the tasks based on the processing time to reduce billing of resources usage.

Even though there are many algorithms, still there exists some drawbacks. To overcome these drawbacks this project proposes an efficient scheduling algorithm, which can support the load balancing and reduce the power usage, context switching between the servers which reduces the amount of carbon level emitted from virtual machines.

PROPOSED TECHNIQUE

This paper proposes a new algorithm called Enhanced Throttled Load Balancing algorithm (ETLB) which includes the global queue along with combined approach of pull based technique and push based technique. Depending on the situation any one of the approaches will takes place. This global queue will maintain all the incoming jobs and it will push the job into the virtual machine which completes its current task faster than the other virtual machines based on task priority. In Throttled Load Balancing Algorithm (TLB), each and every virtual machine will maintain a separate queue. Because of the queuing there is an increase in waiting time. But in the proposed algorithm there exists only one global queue which provides efficient response time. The proposed algorithm reduces the waiting time, response time and provides the efficient throughput for both static and dynamic task scheduling than the existing algorithms.

ALGORITHM

The proposed algorithm consists of n task sets from T1 to Tn, n arrival times from A1 to An, n Burst times from B1 to Bn. This algorithm uses pull based technique and push based technique.

Input: Taskset (T1, T2…..Tn), Arrival Time (A1, A2…..An), Burst Time(B1,B2…..Bn).

a) Initialize n=number of task.
b) Get the input for arrival time and burst time for the set of tasks.
c) The jobs which are all in the global queue (Q) are assigned to the appropriate virtual machines (VM).
d) The remaining tasks and newly arrived jobs (if any) are maintained in the Q.
e) The waiting time, response time, finish time and turnaround time are evaluated for each and every job in virtual machines.
f) Assign the task from Q to VM which completes the current task first based on priority.
g) Repeat from the step 4 until all the jobs in Q are serviced.

EXPERIMENTAL ANALYSIS

Static method scheduling

To illustrate the proposed model, task set [12] is taken with four virtual machines. Scheduler module will take the tasks and scheduling is done based on the service time in the virtual machines. The job which came first into the global queue is assigned to the virtual machine which completes the job faster than the other virtual machines. The following table1 shows the burst time of the tasks in static method.

Table-1. Task Set referred from [12].

Task set (T)	**Burst time**
T1	4
T2	3
T3	2
T4	1
T5	5
T6	4
T7	6
T8	1
T9	2
T10	3

Let us consider the arrival time for the Static scheduling is 0 for all the tasks. Since all the tasks are coming at the same time, using Shortest Job First algorithm the job which has shortest burst time is given higher priority and the task is assigned to the corresponding virtual machine. If all the four virtual machines are busy the remaining tasks are maintained in the global queue and once the job is finished in the virtual machine the next task which has the shortest job will be assigned to the corresponding virtual machine which is free currently. After allocating the task to a virtual machine the waiting time, finish time and turnaround time for each and every tasks are calculated. The average waiting time and average turnaround time are calculated finally. The gunshot diagram for static scheduling is shown in Figure-1.

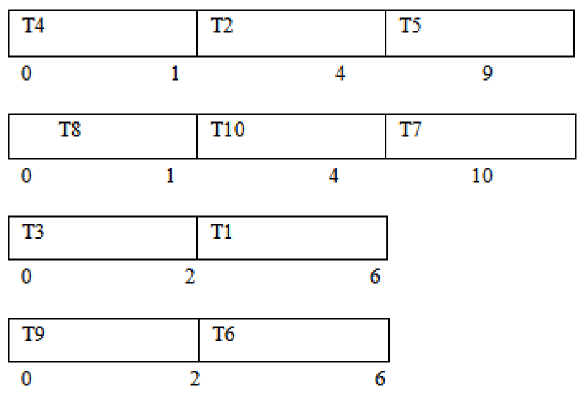

Figure-1. Proposed Gantt chart diagram for static method.

The average waiting time and average turnaround time in static method is shown in table 2.

Table-2. Comparison for Static method.

Algorithm	**Average waiting time**	**Average Turnaround time**
Throttled Load Balancing [11]	2.6	5.7
Enhanced Throttled Load Balancing [proposed]	1.4	4.5

The following Figure-2 confirms that Enhanced Throttled Load Balancing (ETLB) algorithm shows better results.

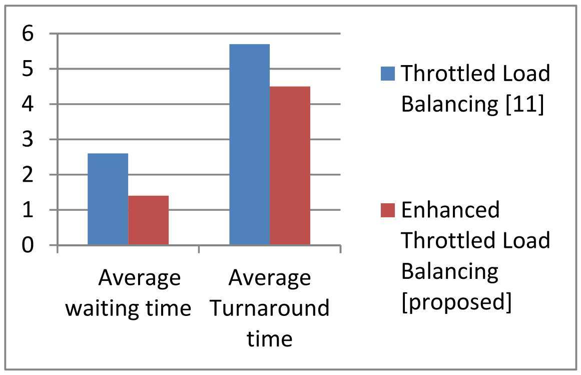

Figure-2. Comparison of ETLB and TLB.

This graph shows that ETLB algorithm is better than the TLB algorithm in static method.

Dynamic method scheduling

In Dynamic method scheduling, the arrival time is not fixed. The task may arrive at any time and the tasks are allocated to the virtual machines in the First Come First Serve manner from the global queue. If the tasks arrive at same time then the ETLB works as SJF manner. If more than one virtual machines is available for task allocation then tasks are allocated in FCFS manner.

The Task set for dynamic scheduling is shown in Table-3.

Table-3. Task set referred from [13].

Task set (T)	**Arrival time**	**Burst time**
T1	0	3
T2	0	4
T3	0	5
T4	2	2
T5	2	2
T6	3	4
T7	3	20
T8	4	14
T9	5	9

Average waiting time and Average turnaround time in dynamic method scheduling for TLB and ETLB algorithms is shown in Table-4.

Table-4. Comparison for Dynamic method.

Algorithm	Average waiting time	Average Turnaround time
Throttled Load Balancing [11]	0.44	7.4
Enhanced Throttled Load Balancing [proposed]	0.44	7.4

Hence ETLB algorithm shows better results in static method and dynamic method as compared to TLB algorithm. ETLB algorithm works 27% more efficient than the TLB algorithm.

CONCLUSIONS

This paper proposes ETLB algorithm which includes global queue along with combined approach of pull based technique and push based technique. From the experimental analysis and the comparative survey this algorithm confirms Enhanced Throttled Load Balancing algorithm provides the high throughput with low response time compared to Throttled Load Balancing algorithm. Instead of assigning a task to a particular virtual machine, in ETLB the tasks are assigned to the virtual machine based on priority manner. As queuing time decreases, waiting time is reduced. Hence ETLB provides the better results as it maintains a global queue. The average waiting time and average turn around time of ETLB algorithm in static method are 1.4 and 4.5 respectively. This work can be extended to design the usage of virtual machine based on hot spot and cold spot techniques to reduce carbon emission.

REFERENCES

[1] M. Lavanya, Dr.V. Vaithyanathan. 2013. Deadline aware Group based algorithm in CLOUD computing environment. In International Journal of Applied Engineering Research. 8(20): 2613-2616.

[2] Z. Liu, M. Lin, A. Wierman, S.H. Low, L.L. Andrew. 2011. Greening Geographical Load Balancing. In Proc. ACM SIG- METRICS Joint Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS). pp. 233-244.

[3] M. Conti, E. Gregori and F. Panzieri. 2000. Load Distribution among Replicated Web Servers: A Qos-Based Approach. In SIGMETRICS Performance Evaluation Rev. 27(4): 12-19.

[4] Nidhi JainKansal. 2012. Cloud Load Balancing Techniques: A Step towards Green Computing. In IJCSI International Journal of Computer Science Issues. 9(1, 1): 238-246.

[5] Z. Liu, M. Lin, A. Wierman, S.H. Low, L.L. Andrew. 2011. Geographical Load Balancing with Renewables. In Proc. Green- METRICS, June 2011, pp. 1-5.

[6] V. Mathew, R.K. Sitaraman, P. Shenoy. 2012. Energy-Aware Load Balancing in Content Delivery Networks. In: Proc. IEEE INFOCOM. pp. 954-962.

[7] Hemant S. Mahalle, Parag R. Kaveri and Vinay Chavan. 2013. Load Balancing On Cloud Data Centres. In International Journal of Advanced Research in Computer Science and Software Engineering. 3(1): 1-4.

[8] Ei Ma, Feng Liu and Zhen Liu. 2012. Distributed load balancing allocation of virtual machine in cloud data center. In Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on. pp. 22-24.

[9] L. Rao, X. Liu, L. Xie and W. Liu. 2010. Minimizing Electricity Cost: Optimization of istributed Internet Data Centers in a Multi- Electricity-Market Environment. In Proc. IEEE INFOCOM. pp. 1-9.

[10] F.F. Moghaddam, M. Cheriet, and K.K. Nguyen. 2011. Low Carbon Virtual Private Clouds. In Proc. IEEE Int'l Conf. Cloud Computing. pp. 259-266.

[11] Bagwaiya and V. Raghuwanshi S.K. 2014. Hybrid approach using throttled and ESCE load balancing algorithms in cloud computing. In Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on. vol., no, pp. 1, 6, 6-8.

[12] M. Lavanya, Dr.V. Vaithyanathan. 2013. Unsurpassed Composite Algorithm to Improve System Efficiency of CLOUD Computing. In International Journal of Applied Engineering Research. 8(20): 2617-2620.

[13] M. Lavanya, Dr.V. Vaithyanathan, S. Saravanan and D. Muthu. 2013. Improved Partitioned Queue Scheduling in Multiprocessor soft Real Time Systems. In International Journal of Applied Engineering Research. 8(20): 2621-2624.