Gurpreet Singh, International Journal of Computer Science and Mobile Applications,
Vol.2 Issue. 9, September- 2014, pg. 30-38 ISSN: 2321-8363

AN ADAPTIVE DISTRIBUTED LOAD BALANCING TECHNIQUE FOR CLOUD COMPUTING

Gurpreet Singh

M.Phil Research Scholar, Computer Science Dept. Punjabi University, Patiala
gurpreet.msa@gmail.com

Abstract: ***Cloud Computing is demand based service model where services, information, software and resources hosted on datacenters which are geographically distributed. Load balancing is very important due to the increase in cloud services traffic. Load Balancing is techniques are used that the whole load is distributed among total available datacenters in a distributive cloud system. So there should not exist such condition where some processing nodes having larger proportion of load while some others processing nodes remains idle. Although many methods has been proposed for load balancing, some are static in nature and others are based on central or completely distributed load balancing techniques. We proposed energy efficient adaptive distributed technique which combines the properties of central and distributed load balancing. It properly utilizes the available resources and automatically responds with variations in demand for cloud services. We compared proposed technique with existing sequential technique for load distribution, energy per task and VM tasks distribution by using green cloud simulator.***

Keywords: cloud computing, virtual machine, physical machine, load distribution, dynamic load balancing

1. Introduction

Cloud computing is demand based service in which resources, user's data, software and other infrastructure are hosted on cloud nodes, which are distributed at different geographically locations. Load balancing techniques are used to avoid such condition where some processing nodes having larger proportion of load while some others processing nodes remains idle. Load balancing techniques helps to effectively utilize the available cloud resources and improves the overall response time of system [1].

Load balancing techniques are very important due to need of balancing the load of cloud system where processing nodes are distributed. Load balancing technique distributes the total load among all available processing nodes. It checks current load status of every processing node or server on arrival of every new request for cloud resource and reassign total load if required [2]. In case of the failure of any processing node or any physical machine, the cloud load balancing technique will automatically forward traffic to other processing node of datacenter. It provides reliability to Cloud Computing services.

2. Types of Load balancing

On the Basis of nature of Load Balancing techniques it can be divided into following two categories:

2.1 Static Load Balancing:

Static load balancing technique does not depend on the current state of the system. It requires node's total processing capabilities detail and other properties of node to implement this technique. Static load balancing technique cannot match with the demand if there are any variations in demand at run time [3]. Static load balancing technique is suitable homogenous environment and where demands for cloud services are constants.

2.2 Dynamic Load balancing:

Dynamic load balancing technique depends on current load status of system. It can handle the variations in user's demands for cloud services. There is no need of prior knowledge of system such as nodes property to implement this technique. Cloud computing provide elasticity so anytime user can demand for additional resources such as computational resources or storage resources [4]. Dynamic load balancing checks current status of system rearranges the load if required. Dynamic load balancing techniques can further divide into two categories distributed and non-distributed. In dynamically distributed technique load balancing is achieved by all nodes of cloud system and task of load balancing is distributed among all available nodes. In non-distributed load balancing technique, central node is responsible for load balancing process.

3. Challenges in Load balancing

There are several challenges in Load balancing for Cloud Computing some are following:

3.1 Cloud Nodes distribution

Cloud Computing services hosts on datacenters which are distributed at different geographically locations. Some load balancing techniques are designed which are efficient only for closely located nodes. Because these techniques do not consider the distance between node and communication delay for replicate the load information. It is a great challenge to design a load balancing technique which can balance load for the system where nodes are distributed at different locations. Load balancing technique must consider the factors available bandwidth of network links among the nodes, distance between processing nodes and client requesting for cloud services so response delay can be calculated based on these factors. [5].

3.2 Replication

Same data can be stored at many locations to achieve higher availability of cloud services. In case of failure of any specific server user can access their information from any other available server. User will automatically direct to available server. But require replication mechanism which is responsible to keep information consistent. A full replication replicates the same data to every node of system. It doesn't proper utilize the available storage capacity. It also imposes higher cost because it has to replicate same data every time. There should be partial replication instead of full replication. Partial replication technique can utilize available storage by making decision which data set should be replicated to which node based on properties of node [6]. This technique better utilizes the total available storage, but it may lead to more complexity because every time it has to decide which dataset is appropriate for which node based on node's properties.

3.3 Failure

Load balancing technique should have been designed in a way there should not exist single point of failure. In central load balancing system, central node is responsible for collecting the load information of system and taking decision of reassigning load. Central load balancing technique provides efficient solution for load distribution but problem with this technique is, there is single controller. If the case of failure, it will halt the complete system. So any load balancing technique should be designed by taking into account the single point of failure [7].

4. Proposed Technique

We proposed a technique, which divides the load balancing process into two levels, Central level and Distributed level. Main load controller work at central level and PM load controller work at distributed level. All PM controller at datacenters, communicate with main controller and provide current status of load. Main controller is responsible for assigning any specific task to physical machine and taking decision of migration for virtual machine by considering current status of datacenter. When main load controller receives request for cloud service, it decides that which cloud PM should receive the request.

Our proposed technique is self-adaptive in nature. When it find any variations in demand, PM Load balancer sends signal to the main load controller with current status of physical machine. Main load controller is responsible for allowing virtual machine to continue its execution at physical machine or migrating task from one physical machine to another physical machine. Load balancing process is divided into two levels, central level and distributed level as described below:

4.1 Central Level:

The load balancing solution is done by the main Load controller and PM Load controllers. The main load controller first receives the task and then find the datacenter with an idle status, datacenter whose any resource is not assigned to any virtual machine created by the user. If there is only one idle datacenter then it will be chosen for creating virtual machine. But if there are more than one idle datacenters available, dynamic round robin will be used for choosing idle datacenter. If there are no idle datacenter at any specific time then datacenter having minimum load will be chosen for creating virtual machine for a specific task.

After successful creation of virtual machine, it will activate the PM load controller and task would be assigned to PM load controller at distributed level.

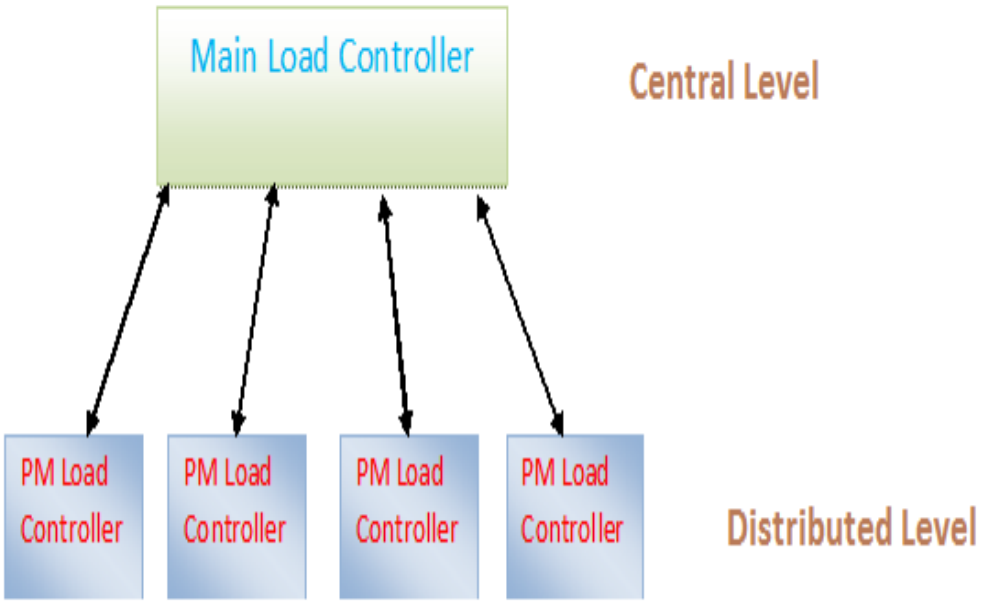

Figure1: Main load controller and PM controller

4.2 Distributed Level:

At distributed level, status of PM will be updated. Variation in cloud service demand will be checked, if there would be no variations in demand at run time then virtual machine would continue its execution at local physical machine. But if found any variations in user's demand, either increase or decrease at run time, then PM load controller will send message to main controller with current load status of PM. Main load controller will compare current load of physical machine with other physical machines of cloud system. If current physical machine will be having load less than other physical machine of cloud, virtual machine will continue its working at local physical machine otherwise task will be migrated to any other PM.

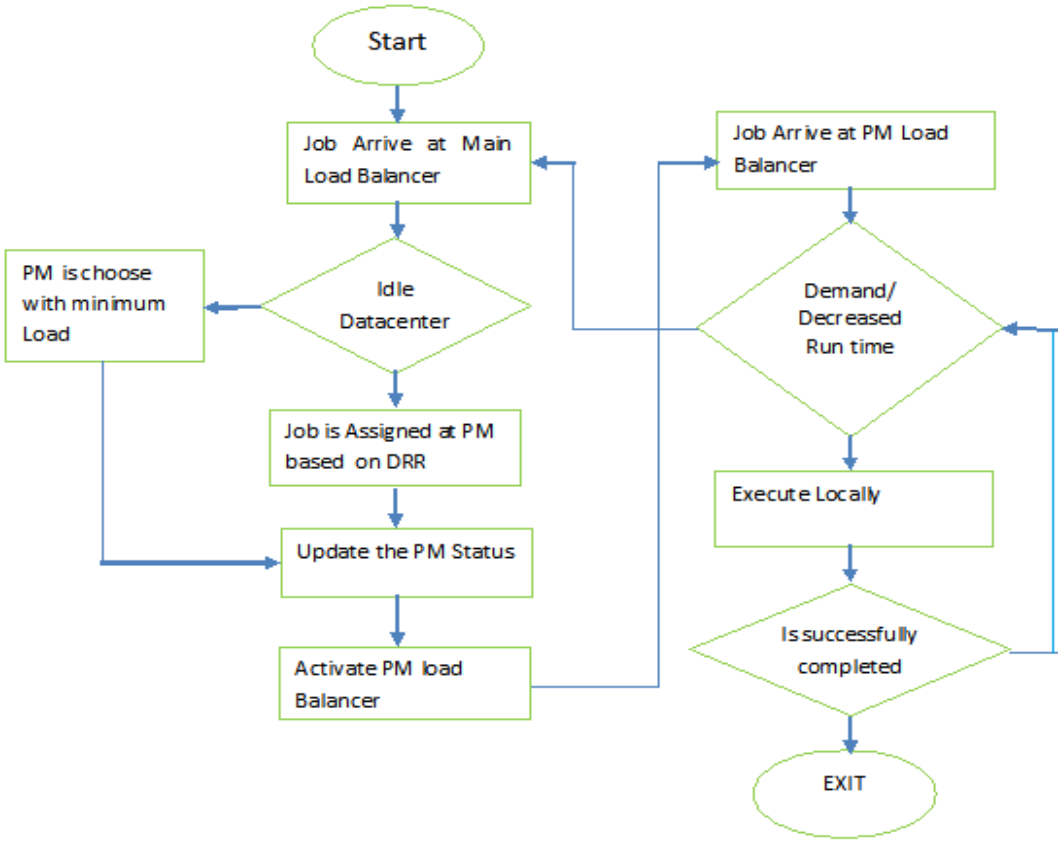

Figure 2: Flowchart for Proposed Technique

4.3 Proposed Techniques steps:

Step 1: Main load controller receives the request for cloud service and finds the available resources.
Step 2: Main load controller checks the status of PM to find the idle PM.
Step 3: If there does not exist any idle PM then it finds the data center having a minimum load status and assign task to that PM and go to step 5.
Step 4: If there exists idle PM then main load controller chooses PM by using dynamic Round-Robin method and assigns task to PM.
Step 5: Load Status of PM is updated and activates the PM load controller.
Step 6: If there are any variations in user's requests for services, either increased or decreased at runtime ,then go to Step 2 else go to step 7.
Step 7: Task continues its execution till the completion of task on local PM.
Step 8: If task is successfully completed then go to step 9 else go to step 6.
Step 9: Exit.

5. Results and Discussion

We use three-tier architecture of Green Cloud simulator. We simulate our proposed model with 144 data centers virtually distributed on different geographically locations. Five users from virtually different location accessing cloud services. Total 30588 tasks are generated which in total takes 65 seconds to complete the simulation.

Table1: Summary for proposed technique

Proposed Technique	
Data Center Architecture	Three-tier
Server	144
Total Task	30588
Avg. Task/server	212 approx.
User	5
Switches (core)	1
Switches (agg.)	2
Switches (access)	3

Figure 3 presents simulation results of datacenter Virtual machine tasks for proposed technique, Virtual machines are created for all 144 servers. No server is idle at any time and every virtual machine is assigned with equal no. of tasks. Every virtual Machine is assigned with approximately 212 tasks. All virtual machines are assigned with an equal load.

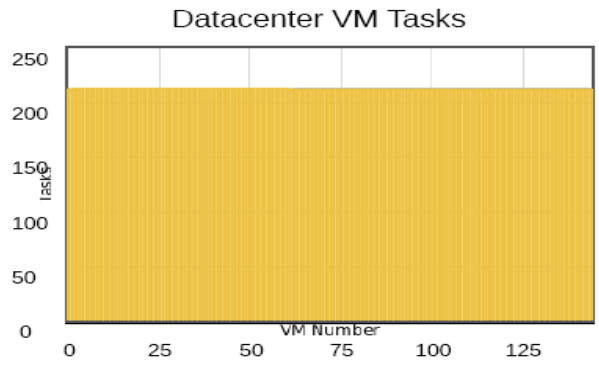

Figure 3: Datacenter VM Tasks

Figure 4 presents a Workload distribution among datacenters. Total simulation time is 65 sec and graph shows that load is constant during period of 65secs. It show that our proposed technique properly balance the load till the competition of task. All datacenters are loaded with equal no. of loads; an approximate load on Every Datacenter is 60%.

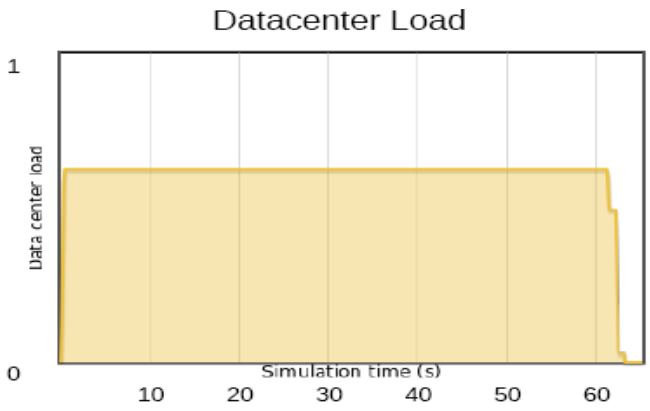

Figure 4: Datacenter Load

We compare our approach with sequential approach. For comparison reasons, we also simulate load balancing technique with 144 servers for both techniques. Therefore, there are 144 data centers virtually distributed at different geographically locations. Total of 2456 tasks are generated by existing sequential technique, simulation time is 65 seconds.

Table 2: summary for Sequential Technique

Sequential Technique	
Data Center Architecture	Three-tier
Server	144
Total Task	2456
Avg. Task/server	17 approx.
User	1
Switches (core)	1
Switches (agg.)	2
Switches (access)	3

Figure 5 presents distribution of total workload among all available processing servers. One third of servers containing the whole workload of datacenter. Result of this one third of servers is overload and remaining two third of server idle or has nothing to execute at this time. Approximately 40 servers out of 144 servers assigned with tasks and remaining approximately 100 servers has no task to execute or idle at this time. It shows that sequential technique not proper utilizing the available resources because most datacenters remain idle during 65 sec of simulation time.

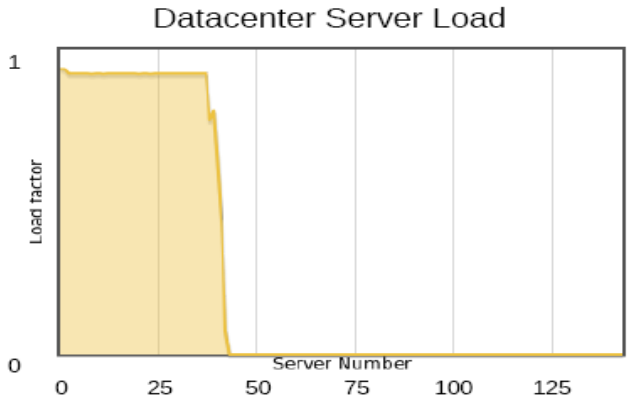

Figure 5: Datacenter Load

Figure 6 presents the simulation result of sequential approach for VM tasks. There are a total of 144 servers and 2456 tasks are generated Approximately 17 tasks should be assigned to every virtual machine if efficient distribution approach would be used. But this result shows it mapped load onto approximately one third of the virtual machine, maintaining load at a peak rate and no virtual machine is created for remaining two third of the servers. Approximately 40 VM are overloaded, assigned by approximately 70 tasks.

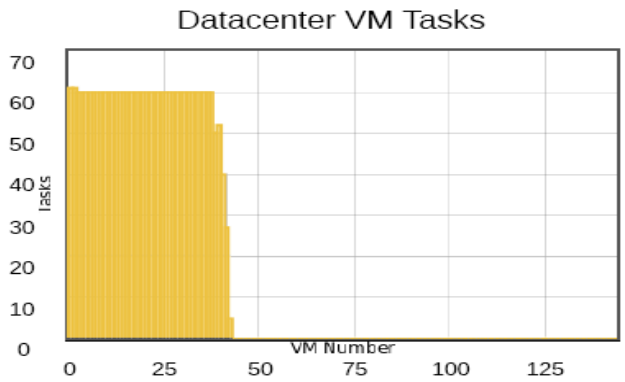

Figure 6: Data Center VM Tasks

Figure 7 presents the detail of total energy consumption for proposed technique. Proposed technique, energy consumption graph shows total energy consumed by proposed technique is 838.8 w/h. Processing server share for 676 w/h, switches used at different layers consume 162 w/h. Proposed technique energy consumption graph shows that the processing servers share around 81% of total data center energy consumption, while the communicational links and switches account for the rest 19%. Furthermore, the consumption of switches breaks with 6% allocated for core switches, 12% for aggregation switches,1% for the access switches.

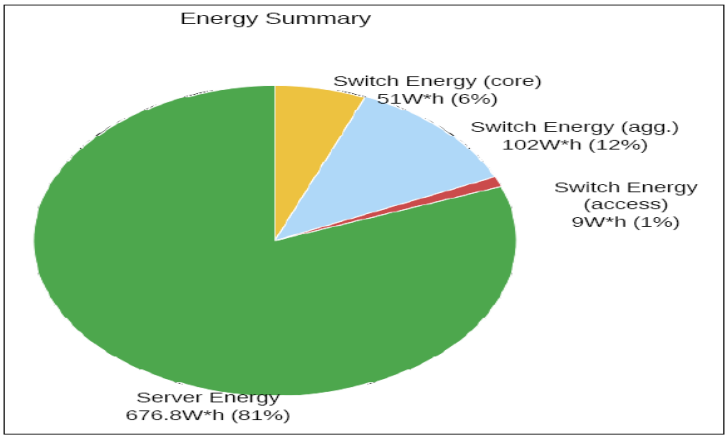

Figure 7: Energy consumption by proposed tech.

Figure 8 presents the detail of total energy consumption for sequential technique. Existing technique, energy consumption graph shows total energy consumed by existing technique is 752.2 w/h. Processing server consumes 590 w/h, switches used at different layers consume 162 w/h.

Existing technique energy consumption graph shows that the processing servers share around 78% of total data center energy consumption. Switches used at different layers consuming rest 22% of total energy. Furthermore, energy consumption of switches breaks with 7% allocated for core switches, 14% for aggregation switches and 1% for the access switches. Comparison of energy consumption per task shows that our proposed adaptive distributed technique is energy efficient per task.

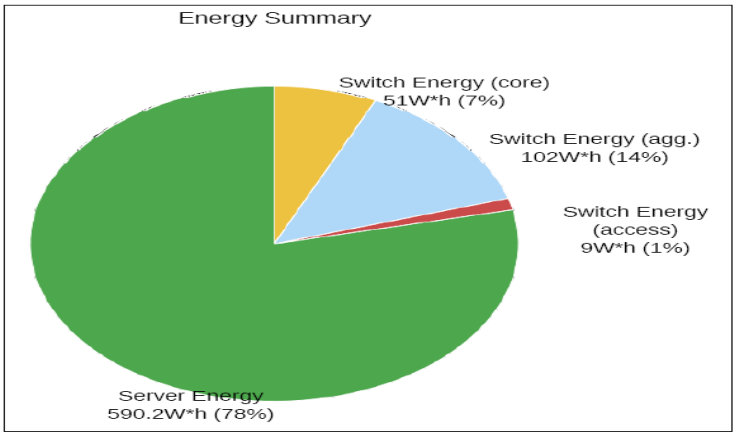

Figure 8: Energy consumption by sequential tech.

Table 3:Comparison of energy per task

Technique	No. of Tasks	Total Energy consumed	Energy/Task
Proposed Technique	30588	838.8 w/h	0.027 w/h
Sequential Technique	2456	752.2 w/h	0.306 w/h

Table 3 presents the comparison of energy consumption per task. Total energy consumed by our proposed approach 838 w/h for executing 50388 tasks. Therefore, energy consumption per task is 0.027 w/h. Total energy consumed by existing technique is 652.2 w/h for executing 2456 tasks. Therefore, energy consumption per task is 0.306 w/h.

6. Conclusion and Future Work

Load Balancing is emerging research area due to increase in cloud service traffic where millions of data are hosted on different datacenters, which are geographically distributed. Load balancing is very important as there is need of balancing the load on this heavy traffic. We have proposed a load balancing technique which properly utilizes the available resources and it transfer upcoming request for cloud services to appropriate physical machine based only on the ability of the node to process new requests. For equal distribution of load we select the datacenter with minimum load. Our proposed technique is energy efficient per task and adaptive in nature, which automatically responds to requests for changes in demand. We have not relied upon static method and dynamic central load balancing. Static method has not the ability to handle flexibility in demand and dynamic central load balancing method has greater chance of failure. Our proposed Technique combines the property of distributed and central load balancing.

We have compared our proposed technique with existing sequential load balancing technique. Our proposed technique provided better results for public and private cloud environment. Therefore as a future work, we plan to test our approach with hybrid cloud environment.

References

[1] M. A. Vouk.,Cloud Computing Issues, Research and Implementations. Journal of Computing and Information Technology, pp. 235–246,2008.

[2] yashpalsinh jadeda,kirit modi, cloud computing-concept architecture and challenges , International Conference on Computing, Electronics and Electrical Technologies ICCEET,2012.

[3] Xu, Gaochao ,Pang, Junjie ,Fu, Xiaodong, A load balancing model based on cloud partitioning for the public cloud, Tsinghua Science and Technology, Volume:18 , Issue: 1 , pages:34 – 39. February 2013.

[4] Jinhua Hu, Jianhua Gu, Guofei Sun, A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment, in 3rd International Symposium on Parallel Architectures, Algorithms and Programming, IEEE 2010.

[5] Radojevic, B. and M. Zagar, Analysis of issues with load balancing algorithms in hosted (cloud) environments, in proc.34th International Convention on MIPRO, IEEE, 2011.

[6] Network Load Balancing Technical Overview, http://technet.microsoft.com/en- us/library/bb742455.aspx

[7] Shufen Zhang, Shuai Zhang, Xuebin Chen, Shangzhuo Wu, Analysis and Research of Cloud Computing System Instance, Second International Conference on Future Networks, 2010.

