I J C T A, 10(8), 2017, pp. 631-634 ISSN: 0974-5572

Dynamic Multi-Layer Approach for Load Balancing in Cloud Computing

N.R. Rejin Paul* and D. Paul Raj**

ABSTRACT

Cloud Computing is an emerged and inevitable field in IT Industry. Its main objective is to deliver cloud services efficiently to end users. One of the major issue in cloud computing is load balancing. Distributing load equally among all the VMs will help to achieve better response time and balancing. In this Paper we propose a Multi-Level Scheduler Algorithm which will effectively manage and distribute the load among various K-cluster group using K-means Clustering technique and grouping the VMs based on its capabilities. The proposed algorithm focus on improved data processing time, throughput, efficient resource utilization and response time

Keywords: Load Balancing, Cloud Computing, Virtual Machines, Super-Agent

1. INTRODUCTION

Cloud computing become one of the dominating Field in Information Technology, Various companies are now adopting cloud computing to make things simple and satisfy On demand services for cloud users, we know that cloud is a high speed computing provide flexibility and elasticity. Efficient Distribution of resources is possible by the idea of virtualization emerged.The main Challenges with cloud computing is to maintain Good response time, efficient resource utilization and minimizing energy consumption

Load Balancing is an important research area where Cloud Computing Technology need to be focus on in the modern era, Distributing equal amount of load among various nodes will give less response time and good resource utilization,Resource utilization need to be monitored within threshold utilization without violating Service Level Agreement (SLA) , Cuckoo Optimization Algorithm detect over and under utilization of computing Resource on virtual machines [1]

Load balancing algorithm categories as static load balancing and dynamic load balancing, static algorithm.

Will work based on static rules defined in the node based algorithm, it doesn't consider the dynamic behavior of the nodes and it focus only on the current state of the virtual machine. The dynamic load balancing algorithm will consider the previous state and current state of the node to take intelligent decision and distribute the traffic to various nodes in cloud computing environment. The performance study various dynamic scheduling algorithm like modified Throttled Algorithm, Particle Swarm Optimization algorithm, FCFS Algorithm are Discussed [2]

2. RELATED WORK

Moona Yakhchi[3] proposed an approach based on Cuckoo Optimization Algorithm detect over-utilized hosts and migrate the traffic to less utilized host with Minimum Migration Time and also allow idle host to sleep mode . In this paper a COA-MMT algorithm has been proposed to reduce the energy consumption in cloud computing infrastructures. They also considered SLA Violation for SLAV, SLATAH and PDM Metrics

* Department of CSE, Velammal Institute of Technology, Chennai-601 204, India, *Email: nrrejinpaul@gmail.com*

** Department of IT, R.M.D Engineering College, Chennai-601 206, India, *Email: kingrajpaul@gmail.com*

by measuring SLA violation. But here they have only considered limited SLA Metrics and response time(RT) was not taken into consideration for Improving the QOS

Mr. Mayur S. Pilavare[4] proposed a genetic algorithm that uses selection of random virtual machine as input for processing the algorithm also uses Logarithmic Least Square Matrix priority algorithm to improve the efficiency of the algorithm . This genetic algorithm out performs various existing loading balancing algorithm. they have considered all jobs requesting for the processor have same priority, this Invites starvation and required best selection Techniques for efficient utilization of computing Resources

Wei-Tao Wen[5] proposed Ant Colony Optimization based on metaheuristic algorithm which follows a distributed VM migration strategy , the idea behind this paper is local agent node monitors the resource utilization and launches the migration, previous and current system condition is considered to avoid unnecessary migration this will reduce the number of migration and maintain the required performance levels of cloud computing.

Kai Pan [6] proposed a improved particle swarm optimization algorithm to achieve optimized load balancing, further more established a resource task allocation model for complex networks in cloud computing

Reena Pan war[7] proposed a dynamic load balancing algorithm will evenly distribute load among various virtual machines , this paper focus on improving data processing time and response time .

ShikhaGarg [8] proposed a Synchronized Throttled Load Balancing to reduce overhead and under load the virtual machines faces. The idea here is to reduce response time and this help to improve the performance of cloud computing environment Comparative analysis also been made with various load balancing algorithm

3. PROPOSED WORK

This paper focus on efficient allocation of resources and task scheduling for load balancing algorithm, the load with different cloud servers may differ based on the traffic from various geographic location, using our multiple layer approach we can achieve improved data processing time, throughput, efficient resource utilization and response time. The following section introduce multiple layer approach and proposed algorithm.

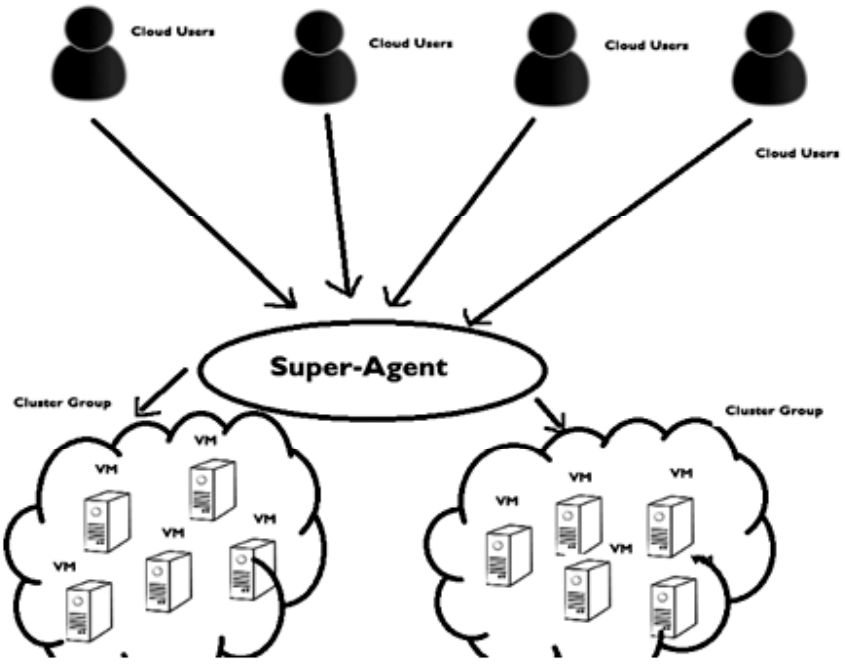

Figure 1: System Model

3.1. Multiple Layer Approach

Our Proposed method introduce two layers of operation of efficient allocation of resources and task scheduling, multiple Layer introduces super-agent, Host and Virtual Machines. Cloud Environment status information has been maintained by super-agent

First Layer includes Host and VM, super-agent node. First the super-agent nodewill divide multiple VMs with identical capabilities into cluster group using K-means clustering approach. Super-agent is responsible for allocating task to cluster Group. Cluster groupwill be dynamically created based on demand. Virtual machines will be removed from the cluster whenever the task is less, this will help to efficiently manage the resources by reducing the time required to analyze each virtual machines

Second Layer represent the mapping between the cluster group and the tasks carried out by super-agent.Mapping works on the basis of Best Fit Algorithm (BFA), bigger task is assigned to tremendous cluster group and lesser task assigned to less powerful cluster group. This mapping helps to achieve efficient utilization of resources

3.2. Multi-Level Scheduler Algorithm

Let us assume *L* represent a queue incoming task from ends users will be put in the queue using processed

Based on First In First out (FIFO) Manner. Let C_i represent group of clusters. H_i represent the host machines and *S* represent Super-agent Node, T_i represent Tasks from cloud users

Step up and initialize hosts, tasks,VMs the physical host will be available as VMs on demand and grouped into cluster based on similar capabilities

Step 1: Initialize VMs and create a K-cluster group using K-means Clustering technique based on the Memory Capabilities, CPU Power and Bandwidth

Step 2: If T‘‘0 ,go to Step 2a operation*else* Goto Step 8

Step 2a: Super-agent node receives tasks T_i from various cloud users

Step 3: Super-agent Node matches the task with available cluster $C_i$and ensure the Request can be Handled by the cluster group and then allocate pertinent cluster group for the particularTask, it Allocate H_i to cluster group if required based on the similar capabilities

Step 4: The availability of VMs will be identified by using Boolean parameter, once the VMs completes its task it set to AVAILABLE or NOT AVALIABLE

Step 5: Super-agent need to maintain the capabilities of VM with the status of VMs

Step 6: Check for AVAILABLE VMs and migrate to sleep Mode

Step 7: Update status Information of VMs to super-agent

Step 8: Remove VMs from Cluster group

To create a cluster group based on the resource capabilities of VMs using K-Means clustering technique, To assign VMs to Cluster Group Euclidean distance formula has be used The value of K is the number of clusters has Been chosen to be the highest prime factor of n where n is the number of VMs [9]

To find VMs distance with other cluster centers[9] can be represented as

$$EUD(VM_i)(C_j) = sqrt[(CPU_i - CPU_j)^2 + (Mem_i - Mem_j)^2 + (B W_i - B W_j)^2 \quad (1)$$

If there is no task T_i for VMs in the cluster group to process, VMs can be removed from cluster group and change VMs to sleep mode. As soon as super-agent node receives tasks from the cloud users its checks

for the available VMs in the cluster Group $C_i$and satisfy the capability constraints based on the Best Fit Algorithm, Super-agent will allocate task to VMs in that particular Cluster Group

Bigger Task T_i can be accomplished by grouping similar Capabilities VMs and assigning the task , thus we can Achieve less data processing time, throughput, efficient resource utilization and response time. Super-agent will also maintain the status information of all Available VMs to ensure the efficient utilization of resources

4. CONCLUSION AND FUTURE WORK

This paper represents efficient Multi Level scheduling algorithm for load balancing in cloud computing environment, this proposed algorithm focus on improved data processing time, throughput, efficient resource utilization and response time. Cluster groups will be formed based on the user Task demand help to utilize the resources efficiently and grouping the VMs based on the capabilities will help to achieve better throughput and response time. Plan to implement this using cloudsim and In future the proposed work can be implemented in real cloud Environment so as to evaluate its performance in various other metrics

REFERENCES

[1] MoonaYakhchi_,SeyedMohssenGhafari_,ShahparYakhchi_,Mahdi Fazeliy,AhmadPatooghi "Proposing a Load Balancing Method Based on Cuckoo Optimization Algorithm for Energy Management in Cloud Computing Infrastructures" Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on 27-29 May 2015.

[2] PriyadarashiniAdyashaPattanaik,Sharmistha Roy,Prasant Kumar Pattnaik "Performance Study of Some Dynamic Load Balancing Algorithms in Cloud Computing Environment" 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)

[3] MoonaYakhchi_,SeyedMohssenGhafari_,ShahparYakhchi_,Mahdi Fazeliy,AhmadPatooghi "Proposing a Load Balancing Method Based on Cuckoo Optimization Algorithm for Energy Management in Cloud Computing Infrastructures" Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on 27-29 May 2015.

[4] Mayur S. Pilavare, Amish Desai "A Novel Approach Towards Improving Performance of Load Balancing Using Genetic Algorithm in Cloud Computing" *ICIIECS'15*2nd International Conference on Innovations in Information Embedded and Communication Systems.

[5] Wei-Tao Wen, Chang-Dong Wang, De-Shen Wu and Ying-Yan Xie "An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment" 2015 Ninth International Conference on Frontier of Computer Science and Technology 26-28 Aug. 2015.

[6] Kai Pan and Jiaqi Chen "Load Balancing in Cloud Computing Environment Based on An Improved Particle Swarm Optimization",Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on 23-25 Sept. 2015.

[7] Reena Pan war Prof. Dr. BhawnaMallick, "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm "Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on 8-10 Oct. 2015.

[8] ShikhaGarg, Rakesh Kumar Dwivediy and Himanshu Chauhan "Efficient Utilization of Virtual Machines in Cloud Computing using Synchronized Throttled Load Balancing" 1st International Conference on Next Generation Computing Technologies (NGCT-2015)Dehradun, India, 4-5 September 2015.

[9] HuiXiong; Junjie Wu; Jian Chen, "K-Means Clustering Versus Validation Measures: A Data-Distribution Perspective," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on , vo1.39, no.2, pp.318-331, April 2009 doi: 10.1109rrSMCB.2008.2004559.