AN EFFICIENT LOAD BALANCING ALGORITHM FOR CLOUD ENVIRONMENT

V.Bharath[1], D. Vijayakumar[2], R. Sabarimuthukumar[3]

[1,2,3] Department of CSE – PG, National Engineering College Kovilpatti, Tamilnadu, (India)

ABSTRACT

Cloud computing is internet-based computing in which large groups of remote servers are networked to allow the centralized data storage, and online access to computer services or resources. Major issue of cloud computing is in its load balancing. In this proposed work, an efficient dynamic load balancing algorithm for cloud environment is introduced. So many clients requesting for same resource at a time for example, exam results election results. It is one of the drawbacks of cloud computing technologies. The load balancing is one of the solutions for this situation. Proposed work consists of three main phases. First phase involves prioritize the user request based on Service Level Agreement. Second phase involves allocation of resource based on the priority. Third phase involves load balancing of the resource. From this three phases user get the continuous services and fast access of resource without in long time queue.

Keywords: Cloud Computing; Load Balancing; Service Level Agreement (SLA); Resource Allocation;

I. INTRODUCTION

Cloud Computing provides us a means by which we can access the applications as utilities, over the internet. It allows us to create, configure, and customize the business applications online. The cloud makes it possible to access information from anywhere at any time. While a traditional computer setup requires you to be in the same location as data storage Device, the cloud takes away that step. The cloud removes the need for you to be in the same Physical location as the hardware that stores your data. Cloud provider can both own and House the hardware and software necessary to run home or business applications. This is especially helpful for businesses that cannot afford the same amount of hardware and Storage space as a bigger company. Small companies can store their information in the cloud, removing the cost of purchasing and storing memory devices. Additionally, because you only need to buy the amount of storage space use, a business can purchase more space or reduce their subscription as their business grows or as they find they need less storage space.

The main contributions of this proposed work are summarized below.

- Load balancing is the pre-required for increasing, the cloud performance and complete utilizing the resource. It reduces the response time, execution time and performance of the speed. Load balancing is networking solution responsible for incoming traffic among server hosting the same application content.
- Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work.

- Our objective is to develop an effective load balancing algorithm using Divisible load scheduling theorem to maximize or minimize different performance parameters (for example throughput, latency) for the clouds of different sizes (virtual topology depending on the application requirement).

II. RELATED WORK

Cloud computing involves sharing resources to multiple users. The multiple users have to be provisioned with resources without any congestion or traffic in the network. Those issues can be overcome with efficient load balancing techniques. There are various algorithms for balancing the load among the nodes proposed in international journals and they are discussed below briefly.

Nader Mohamed et al (2014) proposed file download make up a large percentage of the internet traffic to satisfy various clients using distributed environments for their cloud, internet applications. Cloud data server replicate data server, storage infrastructure and servers at various sites to meet overall high demand for their client and increase availability. To reduce use of redundancy and to enhance downloads speed. This paper introduce fast and efficient concurrent technique for downloading large files from replicated files on distributed servers to enhance file download times through concurrent download of flies from opposite direction in the files. Implement the DDFTP and experimentally demonstrated considerable performance gains for file downloads compared to other concurrent/parallel file/data download models.

Bin Dong, Xiuqiao Li et al many solution have proposed the load imbalance issue of parallel files system. The existing solution will prohibitively in efficient in large scale parallel file systems. To address this parallel paper presents SALB. SALB employs an optimization model for file migration. The load balancing algorithm for parallel file systems needs to deal with the following three new challenges. The first challenge for the load balancing algorithm is how to provide the scalability and the availability required by the steadily growing parallel I/O system. The second challenge for the load balancing algorithm is how to take the network transmission into account. The third challenge for the load balancing algorithm is how to effectively realize its load migration.

Zehua Guo et al Software-Defined Networking (SDN) is a new network technology that decouples the control plane logic from the data plane and uses a programmable software controller to manage network operation and the state of network components. In an SDN network, a logically centralized controller uses a global network view to conduct management and operation of the network. The centralized control of the SDN network presents a tremendous opportunity for network operators to refractor the control plane and to improve the performance of applications. For the application of load balancing, the logically centralized controller conducts Real-time Least loaded Server selection (RLS) for multiple domains, where new flows pass by for the first time. In this paper, we propose a new type of controller state synchronization scheme, Load Variance-based Synchronization (LVS), to improve the load-balancing performance in the multi-controller multi-domain SDN network. Compared with PS-based schemes, LVS-based schemes conduct effective state synchronizations among controllers only when the load of a specific server or domain exceeds a certain threshold, which significantly reduces the synchronization overhead of controllers.

Yang Xu et al (2011) in this paper map reduce are providing complex job decomposition and sub task management. Map reduce model with an agent-aided layer and abstract working load request for data blocks as tokens. The token based work is performed .The token routing algorithm is used. When the size of cloud scales up, cloud computing is required to handle massive data accessing requests such as distributed data mining.

Jaspreet Kaur et al (2012) this paper presents an approach for scheduling algorithm that can maintain the load balancing and provides better improved strategies through efficient job scheduling and modified resource allocation techniques. The load can be CPU load, memory capacity, delay or network load. For best resource utilization distribution of load among distribution system. The nodes are heavily loaded then the loads are equally spread.

Nitin S.More et al (2012) Load Balancing is a method to distribute workload across one or more servers, network interface, hard drivers and other computing resources. In large powerful computing hardware and network infrastructure, hardware failure, power and network interruption and resource limitation in time of high demand. Cloud computing has a key way of manage resources, Now Cloud computing allows companies to outsource some resource and application to third parties and it means less hassle and less hardware in the company. Just like any outsourced system, though, cloud computing requires monitoring. Get the resource like RAM, hard disk space, etc... And set some threshold value to each and every resource through which can divert the load to another node present in the cloud. Jobs are making as a thread. The thread are submitted to load balance and verifies the threshold value of the node as well as threshold value upcoming load if it is satisfied the request will be forwarded to next step.

Pankraj Sharma et al (2012) load balancing are core and challenges issues in cloud computing. How to use cloud computing resources efficiently and gain the maximum profits with efficient load balancing algorithm is one of the cloud provider’s ultimate goals. In this paper firstly a analysis of different virtual machine(VM) load balancing algorithm has been proposed and implemented in virtual machine environment of cloud computing in order to achieve better response time and cost. Virtual Machine enables the abstraction of an OS and Application running on it from the hardware. It is controlled by Data centre. Data centre manages the data centre management activities such as VM creation and destruction and does the routing of user requests received from user bases via the internet. It maintain a record of the state of each virtual machine, if a request arrive concerning the allocation of virtual machine, throttled load balancer send the ID of deal virtual machine to the data centre controller and data centre controller allocates the ideal virtual machine.

Dzmitry Kliazovich et al (2013) energy consumption accounts for a large percentage of the operational expenses in data centers that are used as backend computing infrastructure for cloud computing. Existing solutions for energy efficiency and job scheduling are focusing on job distribution between servers based on the computational demands, while the communication demands are ignored. This work emphases the role of communication fabric and presents a scheduling solution, named e-STAB, which takes into account traffic requirements of cloud applications providing energy efficient job allocation and traffic load balancing in data center networks. Effective distribution of network traffic improves quality of service of running cloud applications by reducing the communication-related delays and congestion-related packet losses. The validation results, obtained from the Green Cloud simulator, underline benefits and efficiency of the proposed scheduling methodology.

III. RELATED WORK

In this section the system model and the three phases of the entire work is presented in detail.

3.1 System Model

In this proposed work, Load balancing technique is used to control the network congestion. Load balancing is the process of distributing the load among various resources in any system. Load balancing is a core networking solution responsible for distributing incoming traffic among servers hosting the same application content. The user request got from users and prioritizes the user request based on the Service Level Agreement (SLA). Then based on the priority the resources are allocated to the priority user. Then Virtual Machine (VM) are balanced if too many request allocated.

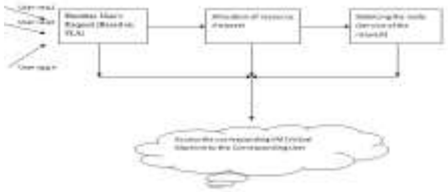


Fig: System Architecture

3.2 Three Phases of Load Balancing

Our proposed work involves three phases such as Prioritize user request phase, Allocation of resource / request phase and Balancing the Node phase. They are discussed briefly below.

Phase 1 : Prioritize user request

In this phase, User request is processed and classified based on the priority assigned to the user request. The priority will be assigned based on SLA (Service Level Agreement).SLA has agreement of cloud user to access the cloud. Agreement contains cost, level, usage etc. Each user has to select the priority such as immediate, normal and lower.

If the user selects the priority as “immediate” then those user requests will be processed immediately. If the user selects the “Normal” priority, then those user requests will be processed normally (processed when available). If the user selects the priority as “lower”, then those user requests will be processed with less importance compared to others.

Phase 2 : Allocation of resource / request

After load balancing the user request (based on the priority assigned to the user), the requested resources has to be allocated to them. This phase involves creation of virtual machine with all the requirements of the user was performed. Each combination of the user requirements is created as single instance and provision to the user. If the required configuration is currently unavailable or the number of instances exceeds (a limit assigned) then there will be the trigger for creating new instance. If the maximum number of instance exceeds for the particular service, then the user has to wait for a while until the infrastructure was created for them.

Phase 3: Balancing the Node

According to the prioritized user request the node are balanced for corresponding user. Based on the priority, the node are allocate to the corresponding user. The user requests were classified based on priority. Each class of user request are processed separately or assigned to various nodes. In this scenario, the user requests were separated and thus loads were balanced.

Weighted Load Balancing Algorithm

1. Create VM’s of different Datacenter according to computing power of host/physical server in terms of its core processor, processing speed, memory, storage etc.

2. Allocate weighted count according to the computing power of the VM’s in Datacenter. If one VM is capable of having twice as much load as the other, the powerful server gets a weight of ‘2’ or if it can take four times load then server gets a weight of ‘4’ and so on.

3: WeightedActiveVmLoadBalancer maintains an index table of VMs, associated weighted count and the number of requests currently allocated to the VM. At start all VM's have 0 allocations.

4: When a request to allocate a new VM from the Datacenter Controller arrives, it parses the table and identifies the least loaded VM.

5: After identifying the least loaded VM's in different datacenters, it allocate requests to the most
Powerful VM according to the weight assigned. If there are more than one, the first identified is selected.

6: WeightedActiveVmLoadBalancer returns the VM id to the Datacenter Controller. The Datacenter Controller sends the request to the VM identified by that id.

7: DataCenterController notifies the WeightedActiveVmLoadBalancer of the new allocation.

8: WeightedActiveVmLoadBalancer updates the allocation table increasing the allocations count for that VM.

9: When the VM finishes processing the request, and the Datacenter Controller receives the response cloudlet, it notifies the WeightedActiveVmLoadBalancer of the VM de-allocation.

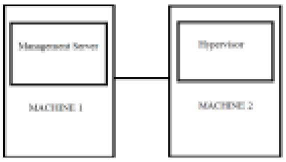
10: The WeightedActiveVmLoadBalancer updates the allocation table by decreasing the allocation count for the VM by one.

IV. EXPERIMENT AND EVALUATION

In this section we present the cloud setup and experimental results of our proposed work in detail.

4.1 Experimental Setup

The cloud environment for implementing the proposed system was created using the cloud developing tool called "Cloud-Stack" which provides Infrastructure as a Service (IaaS) for the cloud providers. The basic deployment of the cloud involves two separate machines as shown in Fig 3.



2. Basic Deployment

3. System Arch3it

Fig: Basic Deployment of cloudstack

The Machine 1 acts as the management server that manages cloud resources. By interacting with the management server through its UI or API, we can configure and manage our cloud infrastructure. It controls allocation of virtual machines to hosts and assign storage and IP addresses to virtual machine instances. The Machine 2 acts as the Hypervisor that creates and runs virtual machines and that was managed by machine 1.Using the management server configuration setup, we configured our cloud in the form of availability zones. At most 1500 files with its size varying from 1 KB to 114,725 KB, was used for our experiment. Various formats like video contents, image files, and text files are used. The maximum network bandwidth of our system is configured to 100 Mbps. The entire implementation of the system was done using java.

4.2 Experimental Results

The Load Balancing Technique involves providing a login page for each user to make a request for the resources such as email, storage and eBooks. Once log in, each user can make a request to the resources. The user can select the service (email, eBooks or storage). Apart from service, the user can specify the priority like normal, immediate and low priority. Once users select all the requirements, it will be stored in Request catalog. Each user will assigned with unique ID and date time of request and it was shown in Figure 3.

Fig. 3. User Request Catalog

User request catalog for three users and they have selected various priorities such as immediate, Normal, and lower.

Fig. 4 . User Request Catalog with various priorities.

For each service, status is maintained separately in service catalog. The Service catalog has information about Maximum number of instance, Number of available instances, and Maximum number of request. Here the maximum number of instance for assigned as 2, 3, and 1 for email, storage and eBooks respectively. And the Maximum number of a request allowed for each service as assigned as 5, 2 and 3 respectively for each service .Since the number of request and number of available instance are within the limit, then the status will be "Current Instance Proceeds". If there is no request for a service then the status will be "No Instance Assigned"

Fig 5. Service Catalog and VM status

Here for the service "email", the number of request equals 5 and also Number of available instance also equals to Maximum number of allowed instance. So the status turns to "request limit exceeds, trigger to next instance".

Fig 6. Service Catalog "Need to extend infrastructure

V. CONCLUSION

An efficient load balancing algorithm based on the user priority is proposed. The user priority is assigned based on the SLA. Initially the user requests are assigned a unique ID and priority and they are maintained in separate Request Catalog. Then the service details and the available instances are maintained in Service Catalog. Based on the available instance details in service catalog, the instances are assigned to the user. In Future, multiple user requests will be balanced on their load in the network. Thus, this method aims to enhance the entire system performance and reduces the network congestion.

REFERENCES

[1] Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao, Li Ruan (2012), ‘A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers’, j.Parallel Distrib. Comput 72, pp. 1254-1268.

[2] Dzmitry Kliazovich, Sisay T. Arzo, Fabrizio Granelli, Pascal Bouvry and Ullah Khan, (2013) ‘e-STAB: Energy-Efficient Scheduling for Cloud Computing Application with Traffic Load Balancing’, IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, pp. 7-13.

[3] Jianying Luo, Lei Rao, and Xue Liu (2014), ‘Temporal Load Balancing with Service Delay Guarantees for Data Center Energy Cost Optimization’, IEEE Transactions on Parallel and Distributed System, Vol. 25 No. 3, pp. 775-783.

[4] Jaspreet Kaur (2012), ‘Comparison of load balancing algorithms in a cloud’ International Journal of Engineering Research and Application, Vol.2 Issue 3, pp. 1169-1173.

[5] Nader Mohamed, Jameela Al-Jaroodi, Abdulla Eid (2013), ‘ A dual-direction technique for fast file download with dynamic load balancing in the cloud’, Journal of Network and Computer Application 36 , pp. 1116-1130.

[6] Nitin S.More, Swapnaja R. Hiray, Smita Shukla Patel (2012), ‘Load Balancing and Resource Monitoring in Cloud’, International Journal of Advance in Computing and Information Researches ISSN: 2277-4068, Volume1- No.2, pp. 19-22.

[7] Qiaomin Xie, Yi Lu, Gabriel Kliot, Alan Geller, James R. Larus, Albert Greenberg (2011),’ Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web service’, Performance Evaluation 68, pp. 1056-1071.

[8] Pankraj Sharma, Meenakshi Sharma, Sandeep Sharma (2012), ’Efficient Load Balancing Algorithm in VM Cloud Environment’, International Journal of Computer Science And Technology, Vol.3, Issue 1, pp. 439-441.