Research Journal of Applied Sciences, Engineering and Technology 7(22): 4702-4705, 2014
DOI:10.19026/rjaset.7.854
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: October 25, 2013 Accepted: November 11, 2013 Published: June 10, 2014

Research Article

A QOS Based Optimal and Cost Effective Load Balancing Strategy in Storage Cloud

[1]S.P. Jeno Lovesum, [2]K. Krishnamoorthy and [3]P. Blessed Prince
[1]Department of Computer Science and Engineering, Karunya University, India
[2]Department of Computer Science and Engineering, Sudharsan College of Engineering, India
[3]Department of Information Technology, Karunya University, India

Abstract: The aim of this paper is to create an optimal and cost effective load balancing strategy based on the QOS specified by the user which has been included as SLA in the request sent by the user to the Service provider. Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications. When a user wants to store his or her files in the cloud initially they need to select a cloud service provider to use the providers storage area. Storage cloud belongs to Iaas cloud. We consider the scenario that no VM should be overloaded and at the same time no VM should be idle. For this the data file sent by the user is not stored as a whole file instead it is split into smaller files and distributed among the available VM for storage. This paper mainly focuses on QOS parameters like reliability, stability and the overall throughput.

Keywords: Cloud, Iaas, load balancing, QOS, storage area

INTRODUCTION

Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. It is a pay-as-you-go service. All these factors increased the popularity of cloud computing The services are broadly divided into three categories Infrastructure-as a Service (IaaS), Platform as a Service (PaaS) and Software-as a Service (SaaS). Cloud computing is the combination of two characteristics, flexibility and scalability (Armbrust *et al*., 2009). Cloud computing provides flexibility and scalability by providing heterogeneous resources to the users (Rimal *et al*., 2009). Heterogeneity is achieved with the help of virtualization. Through virtualization Virtual Machines (VM) are capable of processing the user requests. VMs are guest operating systems deployed on a physical server. Physical host server will allocate memory, disk and other resources to different VMs deployed on them. IaaS is the common cloud service type. This is because infrastructure is costlier than other services and cloud can play an important role in reducing the infrastructure costs through the provisioning of the IaaS.

The end of this decade is marked by a paradigm shift of the industrial information technology towards a subscription based or pay-per-use service business model known as cloud computing. This paradigm provides users with a long list of advantages, such as provision computing capabilities; broad, heterogeneous network access; resource pooling and rapid elasticity with measured services. Huge amounts of data being retrieved from geographically distributed data sources and non-localized data-handling requirements create such a change in technological as well as business model. Cloud data storage is one of the salient service in cloud computing. The subscriber's data to be stored in data storage that is offered by cloud service provider's servers. The subscribers have to pay the cost to Service Providers (SP) for this storage service. In this paper An Optimal Proficient Load Balancing Strategy has been implemented to provide customers with data availability as well as optimal storage and therefore to minimize the storage cost of the customer by increasing the efficiency and throughput.

In this paper a cost effective load balancing strategy has been discussed and implemented by considering the QOS parameters specified by the user which is being created as SLA and sent along with the user request. The QOS parameters considered are reliability, stability and throughput. Using this parameters the cost effective path to schedule the workflow is calculated.

LITERATURE REVIEW

Token routing: The main objective of the algorithm Wickremasinghe *et al*. (2010) is to minimize the system cost by moving the tokens around the system. But in a scalable cloud system agents can not have the enough information of distributing the work load due to

Corresponding Author: S.P. Jeno Lovesum, Department of Computer Science and Engineering, Karunya University, India

communication bottleneck. So the workload distribution among the agents is not fixed. The drawback of the token routing algorithm can be removed with the help of heuristic approach of token based load balancing. This algorithm provides the fast and efficient routing decision. In this algorithm, Agent does not need to have an idea of the complete knowledge of their global state and neighbor's working load. To make their decision where to pass the token they actually build their own knowledge base. This knowledge base is actually derived from the previously received tokens. So in this approach no communication overhead is generated.

Round robin: In this algorithm (Beltran *et al*., 2011), the processes are divided between all processors. Each process is assigned to the processor in a round robin order. The process allocation order is maintained locally independent of the allocations from remote processors. Though the work load distributions between processors are equal but the job processing time for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle. This algorithm is mostly used in web servers where Http requests are of similar nature and distributed equally.

Randomized: Randomized algorithm is of type static in nature. In this algorithm (Beltran *et al*., 2011) a process can be handled by a particular node n with a probability p. The process allocation order is maintained for each processor independent of allocation from remote processor. This algorithm works well in case of processes are of equal loaded. However, problem arises when loads are of different computational complexities. Randomized algorithm does not maintain deterministic approach. It works well when Round Robin algorithm generates overhead for process queue.

Central queuing: This algorithm (Hsu and Liu, 2010) works on the principal of dynamic distribution. Each new activity arriving at the queue manager is inserted into the queue. When request for an activity is received by the queue manager it removes the first activity from the queue and sends it to the requester. If no ready activity is present in the queue the request is buffered, until a new activity is available. But in case new activity comes to the queue while there are unanswered requests in the queue the first such request is removed from the queue and new activity is assigned to it. When a processor load falls under the threshold then the local load manager sends a request for the new activity to the central load manager, central manager then answers the request if ready activity is found otherwise queues the request until new activity arrives.

Connection mechanism: Load balancing algorithm (Jaspreet, 2012) can also be based on least connection mechanism which is a part of dynamic scheduling algorithm. It needs to count the number of connections for each server dynamically to estimate the load. The load balancer records the connection number of each server. The number of connection increases when a new connection is dispatched to it and decreases the number when connection finishes or timeout happens.

METHODOLOGY

System architecture: In this paper an Optimal Proficient Load Balancing Strategy has been discussed to provide customers with data availability as well as optimal storage and therefore to minimize the storage cost of the customer. The architecture is shown in

Fig. 1: Architecture for optimal load balancing

Fig. 1. In this architecture the user sends their request to the various cloud providers available. In the request they specify the amount of resource they need for storage also they specify their QOS expectation in the form of SLA to the providers. The providers calculate the cost for the requested resource and sent it back to the user. The user chooses the minimal cost provider and who can meet up their QOS specification.

Proposed load balancing strategy: After choosing the cost effective service provider we go for storing the data in the form of files. For placing it, the load in the nodes has to be analyzed. Here the load is considered as number of executing files in that node, its memory usage details etc. With these things the minimal loaded node is identified. To calculate the load on VM there is a master VM, the master VM contains a VM table that is being updated with the current information about the VM. The VM table contains information like no of VM, load on each VM, less loaded VM, overloaded VM, idle VM etc. once a VM is allocated a task it is immediately updated in the table by the master VM. Then the Data queue is considered for allocating the task to a VM here task refers to the file to be saved or stored. In the data queue the files are queued according to their arrival time and so based on their arrival time the task is being assigned to a VM. With the growing number of Cloud offerings, even though it opens the chance to leverage the virtually infinite computing resources of the Cloud, it has also becomes challenging for Cloud customers to find the best Cloud services which can satisfy their QoS requirements in terms of parameters such as performance and security. To choose appropriately between different Cloud services, customers need to have a way to identify and measure key performance criteria that are important to their applications.

The Cloud model consists of a resource provider which provides services to its users and is capable of executing scientific workflows. The resource provider provides different services with different Quality of Service for each task of every workflow. For each workflow task, ti, can be processed by mi services, Si = {si, 1, si, 2,..., si, mi}, with different QoS attributes. Real Clouds, like Xignite and StrikeIron, use the pricing model for each resource. The cost of a service usually depends on its execution time, shorter execution times are more expensive. The services of each task are sorted in an increasing order according to their execution times, i.e., si, j is faster (and more expensive) than si, j+1. ET (ti, s) and EC (ti, s) are defined as the execution time and the execution cost of processing task ti on service s, respectively. Throughput and efficiency are important measures to evaluate the performance of infrastructure services provided by Clouds. Throughput is the number of tasks completed by the Cloud service per unit of time. It is slightly different from the Service Response Time metric, which measures how fast the service is provided. Throughput depends on several factors that can affect execution of a task. Let an user application have 'n' tasks and they are submitted to run on 'm' machines from the Cloud provider.

Algorithm:

Step 1: User sends request for data storage to SP along with the SLA
Step 2: SP receives the request and the SLA
Step 3: SP Sends received acknowledgement
Step 4: Calculates the cost for the requested storage area also considers the SLA
Step 5: Sends the calculated cost back to the User
Step 6: User submits the data to be saved
Step 7: Files of Data submitted are queued up in the data queue based on their arrival time A_T
Step 8: Master VM Splits the file into smaller files e.g., A file of 10 MB can be divided 2 MB of five files
Step 9: Goes back to the data queue and picks up the next file to be saved
Step 10: Step 6 to 9 are repeated till the task is completed

RESULTS AND DISCUSSION

This section describes about the experimental evaluation of the algorithm based on cloud sim using the virtual machine creation and finding the cost effective path. The Cloud model consists of a service provider and offers services to execute the workflow.

The experimental steps include the virtual machine creation, task creation, task and path scheduling based on the qos. For plotting the graph take two parameters like the execution time in milliseconds and also the path of the workflow. Here taking a workflow having some tasks and consider their dependencies. The qos attributes for each path has been calculated. The Cloud system efficiency indicates the effective utilization of leased services. Therefore, a higher value for efficiency indicates that the overhead will be smaller.

The qos parameters are used for calculating the execution time for each path in the workflow. In Fig. 2 Reliability reflects how a service operates without failure during a given time and condition. Therefore, it is defined based on the mean time to failure promised by the Cloud provider and previous failures experienced by the users. It is measured by Reliability = probability of violation * pmttf, where pmttf is the promised mean time to failure.

In Fig. 3 Stability is defined as the variability in the performance of a service. For storage, it is the variance in the average read and write time. For computational resources, it is the deviation from the performance specified in SLA.

In Fig. 4 Throughput is an important measure to evaluate the performance of infrastructure services provided by Clouds. Throughput is the number of tasks completed by the Cloud service per unit of time. It is

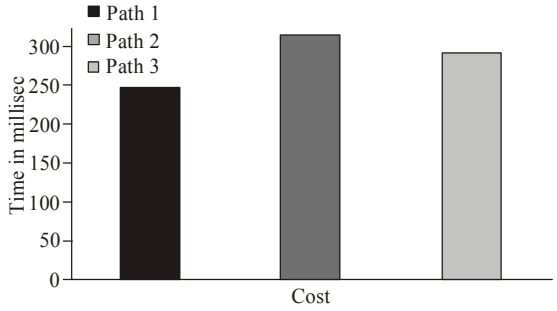

Fig. 2: Normalized cost for load balancing the workflow (reliability)

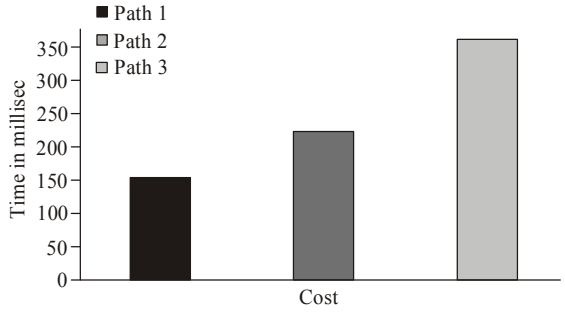

Fig. 3: Normalized cost for load balancing the workflow (stability)

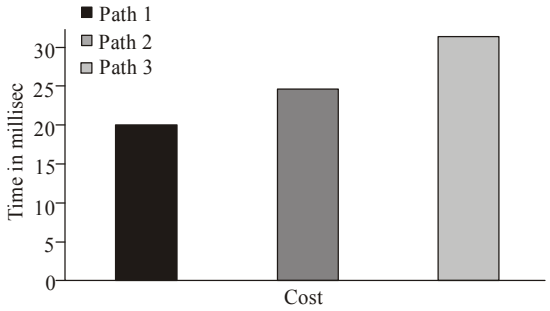

Fig. 4: Normalized cost for load balancing the workflow (throughput)

slightly different from the Service Response Time metric, which measures how fast the service is provided. Throughput depends on several factors that can affect execution of a task.

CONCLUSION AND RECOMMENDATIONS

This paper proposes an optimal and cost effective load balancing technique that can be used for cloud storage. Load balancing is done by splitting the file that the user wants to save into smaller files and distribute it among the available VMs so that the load on the VM is equally distributed and that no VM is overloaded or no VM is idle. At present we have considered only text files and as future work other file formats can be considered. It also considers the QOS parameters to schedule the task to achieve the load balanced. In future this work can be extended so that fault tolerance that occurs during saving a file can be handled and effective algorithms for the communication cost will be also be proposed.

REFERENCES

Armbrust, M., A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, 2009. Above the clouds: A berkeley view of cloud computing. Technical Report No. UCB/EECS-2009-28, University of California, Berkley, USA.

Beltran, M., A. Guzman and J.L. Bosque, 2011. Dealing with heteroginity in clusters. Proceeding of the 5th International Symposium on Parallel and Distributed Computing (ISPDC).

Hsu, C.H. and J.W. Liu, 2010. Dynamic load balancing algorithms in homogeneous distributed system. Proceedings of the 6th International Conference on Distributed Computing Systems, pp: 216-223.

Jaspreet, K., 2012. Comparison of load balancing algorithms in a Cloud. Int. J. Eng. Res. Appl., 2(3): 1169-1173.

Rimal, B.P., C. Eunmi and I. Lumb, 2009. A taxonomy and survey of cloud computing systems. Proceeding of the 5th International Joint Conference on INC, IMS and IDC, pp: 44-51.

Wickremasinghe, B., R.N. Calheiros and R. Buyya, 2010. CloudAnalyst: A cloudSim-based visual modeller for analysing cloud computing environments and applications. Proceeding of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA). Perth, WA, pp: 446-452.