Efficient Virtualization Technique for Cloud to Achieve Optimal Resource Allocation

Suguna Marappan¹, Sharmila Dhandapani ²

¹ Assistant Professor (SRG), Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore -641049, TamilNadu, India. Email: Suguna.marappan@gmail.com

² Professor and Head, Department of Electronics and Instrumentation Engineering, Bannari Amman Institute of Technology, Sathyamangalam-638 401, TamilNadu, India. Email: hodei@bitsathy.ac.in

Abstract

Cloud computing technique has become an unavoidable technology in current IT world and now all the IT enterprises have switched from client-server technology to cloud technology. The underlying architecture of cloud is virtualization technology which multiplexes the server resources and utilizes the server in maximum level. Though the virtualization technology is used by many cloud providers it is not efficient in worst case scenario. The worst case scenario is when the cloud server is overloaded; resource need for an incoming task is insufficient in the cloud server. So in this paper we propose some algorithm and methodologies to make the cloud efficient in a worst case scenario.

Keywords: Cloud computing, Big and Small Cloud, Virtualization, Load rebalancing

1. Introduction

In recent years there is a drastic change in IT world which has gave birth a technology called Cloud Computing[9]. Although the technology is new but the techniques used in it is not new because the techniques are borrowed from existing technologies like Distributed Computing, Utility Computing and Grid Computing. As the cloud computing technique is very clear and adopted by many Enterprises and common people it has some drawbacks concerning its operation. Here we address some of the main problems associated with cloud that is resource allocation, load balancing and fault tolerance. Cloud environment is a web-based environment so it has to behave in dynamic manner. Incoming request for resources may vary in cloud, either it
is high or low according to the state it should change its behavior and this technique is known as rapid elasticity. Even though the technique is there it has some inefficiencies, so to make it more efficient some algorithms and techniques are introduced.

**Fig. 1(a): System Architecture, 1(b): Internal Snapshot of a Cloud Server**

The Fig.1 (a) shows the System Architecture for Dynamic Resource Allocator in which various clients try to access the cloud resources. So this Dynamic resource allocator is responsible for load balancing, resource mapping and dynamic resource allocation. Fig.1 (b) shows the internal snapshot of a cloud server, where it shows how the physical machine is deployed to run various virtual machines (VMs).

2. **Problem 1: Resource Allocation**

The Resource allocation is broadly classified as Static and Dynamic resource allocation[3][4], So, dynamic resource allocation is the best resource allocation method when compared to static resource allocation and it outperforms well in resource allocation and when there is need for resource allocation. In the cloud, virtualization technique is used and it is the heart of any cloud architecture, without virtualization cloud computing is impossible. So here whenever a client raises request for resources, with the virtualization[2] technique, a Virtual Machine (VM) is allocated for each client.

So the resource allocation is happening in this way some inefficiency may occur in Cloud server (CS) it is like when virtual machines (VMs) are allocated for clients CS has a capability to handle (i.e.) maximum 10VMs and minimum 3VMs when it is allocated to handle 15 VMs then it is overloaded and at the same time it is allocated to handle 1 VM then it is underutilized.

As shown in the Fig.2 the CS1 is overloaded with jobs and likewise CS2 has few jobs but the CS3 and CS4 is severely underutilized so it consumes lots of energy even it has no jobs running so energy is wasted due to running state of the cloud server.
3. **Problem2: Resource Insufficient**

The resource insufficient [6] problem occurs when there is a sudden requirement for a running virtual machine to occupy more resources and the available resource is insufficient. This degrades the performance of currently running Virtual machine and also to relinquish the execution.

As shown in the below Fig.3: the previously required resource is lower than the currently required resource and the currently required resource is three fold higher than the previously required if such a condition arises the virtual machine requests the cloud server to allocate the resource at the time CS checks the resource available if the sufficient resource is not available then the cloud notifies the virtual machine and withdraws its execution so this degrades not only the performance of virtual machine and also the application running on the top of the virtual machine.

Fig.3: Resource insufficient problem
4. Solution1: Dynamic Resource Allocation

In dynamic resource allocation[3][4] the resource in a cloud is allocated when there is a need for resources, unless, until it won’t allocate any resources in a cloud server. Similarly, when a cloud server is overloaded instantaneously the VMs are migrated[1][4] to other cloud server in order to avoid the overload and the same way when a cloud server is underutilized the VMs are migrated to other cloud server to avoid underutilization. This is also known as Cloud server consolidation[7] in order to utilize cloud server in maximum level and also to support green computing. The proposed model not only performs dynamic resource allocation and also responds instantaneously to the overload and avoid overload by VM migration process.

Fig. 4: Dynamic resource allocation in Cloud Server

As shown in the Fig.4 the VMs in a cloud server exceeding maximum threshold or fall behind minimum threshold then the VMs are migrated from one cloud server to other cloud server and unutilized cloud servers are switched off and the running cloud servers have consolidated VMs that will utilize the cloud server properly.

4.1 Load prediction and Overload avoidance algorithm

Load prediction and overload avoidance algorithm[3][4][5] is responsible for predicting the load in a cloud server and to avoid when it is overloaded. It is done by the algorithm which is intended to check the available memory in the cloud server, which is capable to run the VMs or not. Load prediction[10] and overload avoidance algorithm not only avoids the load in a cloud server and it also responds to the overload instantaneously.

Step1: Initialize the Variables such as Input CS→CloudServers with VM→VirtualMachines associated with each Cloud Server AM→AvailableMemory, UM→UtilizedMemory L→LoadforVirtualMachine, N→NumberOfUserProcess U→UserProcess.
Step 2: Assign priority to each CloudServers.
Step 3: for i=0 to N
    Identify the priority in Cloud and associated VM.
Step 4: if (UM < AM)
Step 5: Perform V(M)=U.
Step 6: else
Step 7: Identify the FreeResource on priority cloud to perform the allocation.
Step 8: else
Step 9: print InsufficientResource.
Step 10: for i=0 to N
Step 11: if Load(VM) > Threshold(VM) or Load(VM) < Threshold(VM)
Step 12: Print “Migration Required”.
Step 13: End.

4.2 Overhead Computation Algorithm

While migrating virtual machines (VMs) between cloud servers to avoid overload some overheads like time and cost should be concerned because the time delay and cost overhead are major aspects that should be taken into account. These overheads also calculated in order to make virtual machine (VM) migration very efficient so that the virtualization is done properly for the running virtual machines (VMs). So the virtualization process is done efficiently without hanging or crashing the server.

Algorithm 2 Overhead Computation Algorithm

Step 1: Initialize the variables such as M→CloudServers, V→VirtualMachines.
Step 2: set VMcost=0.10.
Step 3: for each(V : M)
Step 4: collect t = dt in pmt.
Step 5:
    compute c = \( \frac{VM_{\text{Migration}} \rightarrow CS_{i}}{CS_{n}} \)
Step 6: End.

5. Solution 2: Load Rebalancing

Resource insufficient problem is avoided by load rebalancing[8] mechanism in which the VMs have sudden requirement so that the loads in other cloud servers are estimated and the best performing cloud server is also examined by Performance Estimation algorithm to fit the VMs requirement and also achieve 99.9% uptime of the cloud server. Load rebalancing is the process
of rearranging the VMs by estimating the previous requirement and current requirement of resources. In that consolidation of the VMs which need only low requirement of resources are migrated to cloud server’s named as Small clouds and consolidating the VMs which need high requirement of resources are migrated to cloud server’s named as Big clouds.

As shown in the Fig.5 Small clouds and Big clouds will be predefined by estimating the different cloud servers performance that is based on their previous statistics result like downtime percentage, fault tolerance are considered and in that best performing clouds are added to the category Big clouds and average performing clouds are added to the category Small clouds. Load rebalancing algorithm is used to rebalance the VMs which require low requirement of resources to Small cloud and the VMs which require high requirement of resources to Big cloud. By this mechanism we can achieve a better utilization of cloud servers and also the insufficient resource problem can be solved.

5.1 Performance Estimation Algorithm

Before load rebalancing the performance of the cloud servers should be estimated it is done by Performance estimation algorithm

Step1: Initialize the variables such as $S \rightarrow$ Small Cloud, $B \rightarrow$ Big Cloud, $CS \rightarrow$ Cloud Server.
Step2: Collect Statistics $T_s$ of CS$n$.
Step3: if $T_s$ == Good
Step4: then $CS_i \rightarrow B$.
Step5: else if $T_s$ == Average
Step6: then $CS_i \rightarrow S$.
Step7: else reject from the list.
Step8: End.

5.2 Load Rebalancing Algorithm

Once the Performance estimation is done then load rebalancing can be carried out in order to achieve efficient virtualization and resource utilization.
Step1: Initialize the variables such as $S\rightarrow$ Small Cloud, $B\rightarrow$ Big Cloud, $R_{req}\rightarrow$ Resource requirement, $VM_{mig}\rightarrow$ Virtual Machine Migration, $T_l \rightarrow$ Threshold for Small Cloud, $T_h \rightarrow$ Threshold for Big Cloud, $T_l = 4000$bytes, $T_h = 10000$bytes.

Step2: if $(R_{req} <= 10000$bytes$)$

Step3: then $VM_{mig} \rightarrow B$.

Step4: else if $(R_{req} <= 4000$bytes$)$

Step5: then $VM_{mig} \rightarrow S$.

Step6: End.

6. Output

Fig. 6 (a): Before efficient resource allocation (b): After dynamic resource allocation and load rebalancing (c): Equal utilization of resources by the VMs.

The output of the proposed system is shown in the above figure where Fig. 6(a) shows the overloaded cloud server which is having peak loads that is before efficient resource allocation, Fig.6(b) shows, the peak loads are drastically reduced by the proposed algorithms, Fig.6(c) shows the equal resource utilization by all virtual machines (VMs) in a cloud server.

7. Conclusion and Future Enhancement

An effective Resource allocation system is proposed for cloud computing environment to make its underlying architecture-virtualization to behave in an efficient manner and also by consolidating virtual machines (VMs) to achieve green computing.

Load prediction and Overload avoidance algorithm is proposed to predict the load and avoid the overload and an overhead computation algorithm is proposed to compute overheads when virtual machines (VMs) are migrated between cloud servers and also load rebalancing mechanism is proposed to rebalance load between low resource utilization and high resource utilization VMs.

Future Enhancement will focus on fault tolerance where cloud environment are prone to fault occurrence in cloud servers.
8. References


9. Bibliography

1. M. Suguna is an Assistant Professor (SRG) at the Department of Computer Science and Engineering in Kumaraguru College of Technology, Coimbatore. Her research interests include Distributed Computing and Software Project Management. She is a member of IEEE, IAENG, IACSIT and CSTA. Currently, she is pursuing a Ph.D. in Information and Communication Engineering at Anna University, India.

2. D. Sharmila is an Professor at the Department of EIE, Bannari Amman Institute of Technology, Sathyamangalam, India. She received her Ph.D. in Wireless Security from Anna University, India. Her areas of interest include wireless security, low power VLSI, Distributed computing and ad hoc networks. She has published more than 20 research articles in international journals and international conferences. She is a member of IEEE, IAENG.