Effective VM Scheduling Strategy for Heterogeneous Cloud Environment

B. A Patil[1], Amegha. K[2]

[1]Associate Professor, KLESCET, Belgaum, Karnataka, India

[2]M. Tech CSE, Visvesaraya Technological University, Belgaum, Karnataka, India

Abstract: ***Cloud computing is the set of distributed computing nodes. It is the use of computing resources that are delivered as a service over a network. Virtualization plays a crucial role in cloud computing. Typically VMs are offered in different types, each type have its own characteristics which includes number of CPU cores, amount of main memory, etc. and cost. Presently, static algorithms are being used for scheduling VM instances in cloud. Instead of these, an algorithm is proposed here which dynamically detects the load and then schedules the tasks. The main purpose of the proposed scheduling strategy is to find the minimally loaded computational node. Upon receiving task requests from the clients, server has to schedule these to a minimally loaded node among all available computing nodes.***

Keywords: Virtualization, UEC, Eucalyptus, Secure Shell Login

1. Introduction

Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. Users do not require knowledge of, expertise in, or control over the technology infrastructure in the cloud that supports them.

There are many open source software available for cloud deployment and management. The overall function of these systems is to manage the provisioning of virtual machines for a cloud providing infrastructure-as-a-service. Open-source Frameworks are flexible enough to work with many underlying systems. Commercial clouds only need their system to work with the hardware that they have.

Virtualization is closely related to cloud because an end user can use different services of a cloud using different virtual machine instances. Virtual machine is the software implementation of a computer which will execute different programs like a real machine. After virtualization, it has been possible to present compute resources in the form of Virtual Machine (VM) Images. Virtualization plays a special role in cloud computing. Typically VMs are offered in different types, each type have its own characteristics which includes number of CPU cores, amount of main memory, etc. and cost. To satisfy the needs of such a high performance computations evolves the use of federated clouds and multi-cloud deployments. Efficient scheduling on distributed systems such as heterogeneous cloud and grid computing is mostly affected by load balancing. The main purpose of scheduling here is to find the minimally loaded computational node. Upon receiving task requests from the clients, server has to schedule these to a minimally loaded node among all available computing nodes.

Here, the open-source framework, Eucalyptus is chosen for cloud deployment. Eucalyptus is included in Ubuntu Enterprise Cloud, UEC for short. UEC is a stack of applications included in Ubuntu Server Edition. It is an open-source infrastructure for cloud computing. Ubuntu Enterprise Cloud (UEC) provides virtualization capability, applications and flexibility to help deploy a cloud within an organization. Along with Eucalyptus, UEC includes a number of other open source software as libvirt, KVM or XEN virtualization technology also. It is very easy to install and configure cloud with UEC.

2. Existing System

Scheduling in Eucalyptus determines the method by which Virtual Machines are allocated to the nodes. This is done to balance the load on all the nodes effectively and to achieve a target quality of service. The need for a good scheduling algorithm arises from the requirement for it to perform multitasking and multiplexing. A scheduling policy can be chosen by changing the value of SCHEDPOLICY in eucalyptus.conf file. The two existing algorithms in Eucalyptus along with their limitations are listed below.

2.1 Greedy Algorithm

The Greedy algorithm is the default algorithm used for scheduling of Virtual Machines in Eucalyptus. The Greedy algorithm is very simple and straight forward. The greedy algorithm uses the first node that it finds with suitable resources for running the VM that is to be allocated. The first node that is identified is allocated the VM. This means that the greedy algorithm exhausts a node before it goes on to the next node. It is both simple to implement and also the allocation of VMs do not require any complex processing. The major drawback would be the low utilization of the available resources. Even if there are under loaded nodes, an overloading of a node might result.

2.2 Round Robin Algorithm

The Round Robin algorithm mainly focuses on distributing the load equally to all the nodes. Using this algorithm, the scheduler allocates one VM to a node in a cyclic manner.

The scheduler starts with a node and moves on to the next node, after a VM is assigned to that node. This is repeated until all the nodes have been allocated at least one VM and then the scheduler returns to the first node again. Hence, in this case, the scheduler does not wait for the exhaustion of the resources of a node before moving on to the next. The main advantage of this algorithm is that it utilizes all the resources in a balanced order. An equal number of VMs are allocated to all the nodes which ensure fairness. However, the drawback of using this algorithm is that it never checks for the load on the nodes. So if an instance is terminated in any of the previous node, it remains free while the next node in the cycle is overloaded with more instances.

3. Proposed System

The need for a new algorithm can easily be realized from the drawbacks of all the existing algorithms. The proposed algorithm considers the dynamic change in load of the nodes based on which the virtual machines are scheduled. This consideration of load factor leads to better utilization of the resources.

The proposed algorithm addresses the problem of scheduling virtual machine images to a set of distributed compute nodes in a cloud environment. The main purpose of scheduling here is to find the node which can give better performance. It checks for the system configuration and number of VM instances in the nodes and finds the minimally loaded node for assigning new VM instance. A computing node is said to be minimally loaded if its CPU utilization is minimum among all available computing nodes.

Server gets the performance statistics of the nodes, before assigning a task to it. Each of the users' tasks is executed by each VM instance. So, the total load on the node corresponds to the number of instances running on it. And hence, the new task is assigned to the node which has least number of VM instances, i.e., minimally loaded node. It checks for the system configuration and number of VM instances in the nodes. Computing nodes with higher configuration can run more number of VM instances than those with lower configuration. So the proposed algorithm reads the configuration of the nodes as well as the number of instances that are running currently on each of the nodes and accordingly create new instance for the incoming task.

4. System Design

Whenever a request comes from a client it will be first accepted at the appropriate cloud server. At this local cloud site the incoming request will be served by the scheduler to better match the requirement at that cloud server. This is elaborated in the following steps:

- Client sends a request to perform some computation on cloud.
- First it goes to the respective server requested for computation.
- If the request is given to a cloud server then it is scheduled to the best computational node (actual compute nodes) with the help of proposed algorithm by the scheduler.
- Once the request reaches the desired computing node, then the client-server gets connected and computations are carried out.
- This process goes on continuously for all the incoming requests.

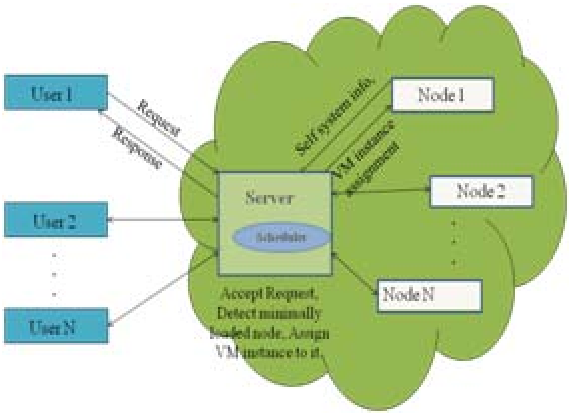

Figure 1: Detailed system architecture

The proposed algorithm is executed in the server. Server does password less login to the nodes through Secure Shell Login (SSH) so as to attain its system load as well as the details of VM instances running in it. These are the inputs to the algorithm. The proposed system works as follows.

- Gets the system load of all the computing nodes by logging into it using SSH and retrieving the system information.
- Compares the load of all the system and finds the one with minimum load.
- Assigns the new VM instance to the node which is minimally loaded.

5. Results and Discussions

On comparison with the other algorithms, the proposed scheduling strategy has been found to work more effectively. It has been implemented in a private cloud deployed using Eucalyptus. The proposed scheduling strategy does not exhaust a particular node like Greedy algorithm. Therefore the proposed strategy makes better utilization of the available resources.

The round robin algorithm allocates one VM to each node in a cyclic manner. It never checks for the load on the nodes. So if an instance is terminated in any of the previous node, it remains free while the next node in the cycle is overloaded with two instances. Hence, the proposed strategy gives slightly better performance than round robin algorithm.

6. Conclusion

In this paper, a new scheduling strategy for Virtual Machine instances in cloud has been proposed and implemented. A LAN scenario of cloud environment is created using Eucalyptus and tested the algorithm. Thus, the objective of distributing virtual machine to the physical computing nodes on cloud computing environment with efficient load balance is achieved successfully. It will also help to better utilization of cloud resources, solving the problem of underutilization and overutilization of computing resources.

In future, research can be done in users' task assignment to the VM instances efficiently. Factoring the tasks and distributing will be a good research work. The possible future enhancements are;

- Factoring large tasks for execution on multiple computing nodes by considering the task dependency.
- Integration of security issues while transferring the tasks.
- VM migration for load balancing.

References

[1] Ammar Khalid, HusnainMujtaba, “Data Processing Issues in Cloud Computing”, Second International Conference on Machine Vision, 2009.

[2] R. Moreno-Vozmediano, R.S. Montero, I. M. Llorente, “Multicloud Deployment of Computing Clusters for Loosely Coupled MTC Applications”, IEEE Transaction on Parallel and Distributed Systems, Vol 22, no 6, pp. 924-930, June, 2011.

[3] B. Rochwerger, D. Beritgand, E. Levy, A. Galis and others, “The Reservoir Model and Architecture for Open Federated Cloud Computing”, IBM Journal of Rsearch and Development, Vol 53, no 4, pp. 1-11, July, 2009.

[4] Jose Luis Lucas Simarro, Rafael Moreno-Vozmediano, Ruben S. Montero and I. M. Llorente, “Dynamic placement of virtual machines for cost optimization in multi-cloud environment”, International Conference on High Performance Computing and Simulation (HPCS), 2011

[5] Matthias Schmidt, NielsFallenbeck, Matthew Smith, Bernd Freisleben, “Efficient Distribution of Virtual Machines for Cloud Computing”, 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, 2010.

[6] Wen-Chung Shih, Shian-shyong Tseng, ”Performance Study of Parallel Programming on Cloud Computing Environment: Using MapReduce”, IEEE Computer Sociaty, 2010.

[7] Yang, C.T., K.W. Cheng, and K.C. Li, “An Efficient Parallel Loop Self- Scheduling Scheme for Cluster Environments”, Journal of Supercomputing, 2005. 34: p. 315-335.

[8] Yang, C.T. and S.C. Chang, “A Parallel Loop Self-Scheduling on Extremely Heterogeneous PC Clusters”, Journal of Information Science and Engineering, 2004. 20(2): p. 263-273.

[9] Wen-Chung Shih, Chao-Tung Yang, Ping-I Chen, Shian-Shyong Tseng, “A Hybrid Parallel Loop Scheduling Scheme on Heterogeneous PC Clusters”, Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005, pp. 56-58.

Author Profile

B. A Patil received his B.E. and M.E Degree in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum. He is currently working as Associate Professor in department of Computer Science and Engineering, KLESCET, Belgaum, Karnataka, India. His main research interests are in the area of distributed computing, network security and cryptography.

Amegha. K received her graduation in computer Science and engineering from Institution of Engineers India (IEI), Kolkata and now pursuing M Tech degree in Computer Science and Engineering from Visvesaraya Technological University, Belgaum, Karnataka, India. Her main research interest is in the area of cloud computing.