Volume 2, Issue 11, November-2015, pp. 761-764 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

A Study on Dynamic Resource Allocation using Virtual Machines for IAAS

[1]Jella.Kishore, [2]B.Kishore

[1]M.Tech (CSE), Department of Computer Science & Engineering, Malla Reddy College Of Engineering,

[2]Assistant Professor, Department of Computer Science & Engineering, Malla Reddy College of Engineering, Maisammaguda (v) I Dhulapally (M), Secunderabad.

Abstract: - Cloud computing allow business customers to scale up and down their resource usage based on needs. Many of the touted gains in the cloud model come from resource multiplexing through virtualization technology. In this paper, we present a system that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. We introduce the concept of "skewness" to measure the unevenness' in the multi-dimensional resource utilization of a server. By minimizing skewness, we can combine different types of workloads nicely and improve the overall utilization of server resources. We develop a set of heuristics that prevent overload in the system effectively while saving energy used. Trace driven simulation and experiment results demonstrate that our algorithm achieves good performance.

.**Keywords** – Cloud computing, Green computing, Resource, Skewness, Virtual machine.

1. INTRODUCTION

Cloud computing is the delivery of computing and storage capacity as a service to a community of end recipients. The name comes from the use of a cloud shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts services with a user's data, software and computation over a network. The remote accessibility enables us to access the cloud services from anywhere at any time. To gain the maximum degree of the above mentioned benefits, the services offered in terms of resources should be allocated optimally to the applications running in the cloud. The elasticity and the lack of upfront capital investment offered by cloud computing is appealing to any businesses. In this paper, we discuss how the cloud service provider can best multiplex the available virtual resources onto the physical hardware. This is important because much of the touted gains in the cloud model come from such multiplexing. Virtual Machine Monitors (VMMs) like Xen provide a mechanism for mapping Virtual Machines (VMs) to Physical Resources [3]. This mapping is hidden from the cloud users. Users with the Amazon EC2 service [4], for example, do not know where their VM instances run. It is up to the Cloud Service Provider to make sure the underlying Physical Machines (PMs) has sufficient resources to meet their needs VM live migration technology makes it possible to change the mapping between VMs and PMs While applications are running [5], but, a policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized. This is challenging when the resource needs of VMs are heterogeneous due to the diverse set of applications they run and vary with time as the workloads grow and shrink. The capacity of PMs can also be heterogeneous because multiple generations of hardware co-exist in a data center. To achieve the overload avoidance that is the capacity of a PM should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its VMs. And also the number of PMs used should be minimized as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off to save energy. In this paper, we presented the design and implementation of dynamic resource allocation in the Virtualized Cloud Environment which maintains the balance between the following two goals.

Goals to Achieve:

Overload Avoidance. The capacity of a PM must satisfy the resource needs from all VMs running on it. Or else, the PM is overloaded and leads to provide less performance of its VMs.

Green computing. The number of PMs used should be optimized as long as they could satisfy the needs of all VMs. And Idle PMs can be turned off to save energy.

There is an in depth tradeoff between the two goals in the face of changing resource needs from all VMs. To avoid the overload, should keep the utilization of PMs low to reduce the possibility of overload in case the resource needs of VMs increase later. For green computing, should keep the utilization of PMs reasonably high to make efficiency in energy [7]. A VM Monitor manages and multiplexes access to the physical resources, maintaining isolation between VMs at all times. As the physical resources are virtualized, several VMs, each of which is self-contained with its own operating system, can execute on a physical machine (PM). The hypervisor, which arbitrates access to physical resources, can manipulate the extent of access to a resource (memory allocated or CPU allocated to a VM, etc.).

II. RELATED WORK

In [2] author proposed architecture, using feedback control theory, for adaptive management of virtualized resources, which is based on VM. In this VM-based architecture all hardware resources are pooled into common shared space in cloud computing infrastructure so that hosted application can access the required resources as per there need to meet Service Level Objective (SLOs) of application. The adaptive manager use in this architecture is multi-input multi-output (MIMO) resource manager, which includes 3 controllers:CPU controller, memory controller and I/O controller, its goal is regulate multiple virtualized resources utilization to achieve SLOs of application by using control inputs per-VM CPU, memory and I/O allocation. The seminal work of Walsh et al. [3], proposed a general two-layer architecture that uses utility functions, adopted in the context of dynamic and autonomous resource allocation, which consists of local agents and global arbiter. The responsibility of local agents is to calculate utilities, for given current or forecasted workload and range of resources, for each AE and results are transfer to global arbiter. Where, global arbiter computes near-optimal configuration of resources based on the results provided by the local agents. In [4], authors propose an adaptive resource allocation algorithm for the cloud system with preempt able tasks in which algorithms adjust the resource allocation adaptively based on the updated of the actual task executions. Adaptive list scheduling (ALS) and adaptive min-min scheduling (AMMS) algorithms are use for task scheduling which includes static task scheduling, for static resource allocation, is generated offline. The online adaptive procedure is use for re-evaluating the remaining static resource allocation repeatedly with predefined frequency. The dynamic resource allocation based on distributed multiple criteria decisions in computing cloud explain in [6]. In it author contribution is tow-fold, first distributed architecture is adopted, in which resource management is divided into independent tasks, each of which is performed by Autonomous Node Agents (NA) in ac cycle of three activities:

(1) VM Placement, in it suitable physical machine (PM) is found which is capable of running given VM and then assigned VM to that PM, (2) Monitoring, in it total resources use by hosted VM are monitored by NA, (3) In VM selection, if local accommodation is not possible, a VM need to migrate at another PM and process loops back to into placement and second, using PROMETHEE method, NA carry out configuration in parallel through multiple criteria decision analysis. This approach is potentially more feasible in large data centers than centralized approaches.

III.PROPOSED SYSTEM

This proposed system consists of number of servers, predictor, hotspot and cold spot solvers and migration list. Set of servers used for running different applications. Predictor is used to execute periodically to evaluate the resource allocation status based on the predicted future demands of virtual machines.

A. System Overview

The architecture of the system is presented in Figure 1. Each physical machine (PM) runs the Xen hypervisor (VMM) which supports a privileged domain 0 and one or more domain U [7]. Each VM in domain U encapsulates one or more applications such as Web server, remote desktop, DNS, Mail, Map/Reduce, etc. We assume all PMs share a backend storage. The multiplexing of VMs to PMs is managed using the Usher framework [8]. The main logic of our system is implemented as a set of plug-ins to Usher. Each node runs an Usher local node manager (LNM) on domain 0 which collects the usage statistics of resources for each

VM on that node. The statistics collected at each PM are forwarded to the User.

The VM Scheduler is invoked periodically and receives from the LNM the resource demand history of VMs, the capacity and the load history of PMs, and the current layout of VMs on PMs. The scheduler has several components. The predictor predicts the future resource demands of VMs and the future load of PMs based on past statistics. We compute the load of a PM by aggregating the resource usage of its VMs. The LNM at each node first attempts to satisfy the new demands locally by adjusting the resource allocation of VMs sharing the same VMM. The MM Allotter on domain 0 of each node is responsible for adjusting the local memory allocation. The hot spot solver in our VM Scheduler detects if the resource utilization of any PM is above the hot threshold (i.e., a hot spot). The cold spot solver checks if the average utilization of actively used PMs (APMs) is below the green computing threshold.

B. Skewness Algorithm

We introduce the concept of "skewness" to measure t he unevenness in the multi-dimensional resource utilization of a server. By minimizing skewness, we can combine different types of workloads nicely and improve the overall utilization of server resources. Let n be the number of resources we consider and ri be the utilization of the i-th resource. We define the resource skewness of a server p as

$$skewness(p) = \sqrt{\sum_{i=1}^{n} (\frac{r_i}{\overline{r}} - 1)^2}$$

Where r is the average utilization of all resources for server p. In practice, not all types of resources are performance critical and hence we only need to consider bottleneck resources in the above calculation. By minimizing the skewness, we can combine different types of workloads nicely and improve the overall utilization of server resources. The flow chart represents the flow of an algorithm in Fig 2. Our algorithm executes periodically to evaluate the resource allocation status based on the predicted future resource demands of VMs.

C. Hotspot Mitigation

We handle the hottest one first I.e. sort the list of hot spots in the system Otherwise, keep their temperature as low as possible. Our aim is to migrate the VM that can reduce the server's temperature. In case of ties, the VM whose removal can reduce the skewness of the server the most is selected. We first decide for each server p which of its VMs should be migrated away. Based on the resulting temperature we sort list the VMs of the server if that VM is migrated away. We see if we can find a destination server to accommodate it for each list of in the VM. After accepting this VM the server should not become hot spot. We select one skewness which can be reduced the most by accepting this VM among all servers. We record the migration of the VM to that server and update the predicted load of related servers when the destination server is found. Else we move on to the next VM in the list and try to find a destination server for it.

D .Green Computing

When the resource utilization of active servers is too low, some of them can be turned off to save energy. This is handled
in our green computing algorithm. Our green computing algorithm is invoked when the average utilizations of all resources on active servers are below the green computing threshold. We check if we can migrate all its VMs somewhere else for a cold spot p. For each VM on p, we try to find a destination server to accommodate it. The utilizations of resources of the server after accepting the VM must be below the warm threshold. Section 7 in the supplementary file explains why the memory is a good measure in depth. We try to eliminate the cold spot with the lowest cost first. We select a server whose skewness can be reduced the most. If we can find destination servers for all VMs on a cold spot, we record the sequence of migrations and update the predicted load of related servers. Otherwise, we do not migrate any of its VM.

IV. CONCLUSION

We have presented the design, implementation, and evaluation of a resource management system for cloud computing services. Our system multiplexes virtual to physical resources adaptively based on the changing demand. We use the skewness metric to combine VMs with different resource characteristics appropriately so that the capacities of servers are well utilized. Our algorithm achieves both overload avoidance and green computing for systems with multi resource constraints.

REFERENCES

[1] M. Armbrust et al., "Above the Clouds: A Berkeley V iew of Cloud Computing," technical report, Univ. o California, B erkeley, Feb. 2009.
[2] L. Siegele, "Let It Rise: A Special Report on Corpo rate IT," The Economist, vol. 389, pp. 3-16, Oct. 2008.
[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," Proc. ACM Symp. Operating Systems Principles (SOSP '03), Oct. 2003E. H. Miller, "A note on refle ctor arrays (Periodical style—Accepted for publication)," IEEE Trans. Antennas Propagat., to be published.
[4] "Amazon elastic compute cloud (Amazon EC2)," http://aws. amazon.com/ec2/, 2012.
[5] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machine s," Proc. Symp. Networked Systems Design and Implementation (NSDI '05), May 2005..
[6] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast Trans parent Migration for Virtual Machines," Proc. USENIX Ann. Technical Conf., 2005.M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.
[7] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Place ment of Virtual Machines for Managing SLA Violations," Proc. IFIP/I EEE Int'l Symp. Integrated Network Management (IM '07), 2007 .
[8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-Box and Gray-Box Strategies for Virtual Machine Migration," Proc. Symp. Networked Systems Design and Implementation (NSDI '07), Apr. 2007.

