Dynamic Resource Allocation Using Skewness Algorithm in Cloud Computing

D.Golden Jemi
PG Student, Department of CSE
Loyola Institute of Technology and Science

C.S Soumiya
Assistant Professor, Department of CSE
Loyola Institute of Technology and Science

Abstract
Cloud Computing is the latest technology used by many organizations in this competitive world. As many organizations are using cloud computing technology, the major issue is resource allocation as pay-per-use on demand basis. In Cloud environments, efficient resource provisioning and management is a challenging issue because of the dynamic nature of the Cloud and the need to satisfy heterogeneous resource requirements. In such dynamic environments where end-users can arrive and leave the Cloud at any time, a Cloud service provider should be able to make accurate decisions for scaling up or down its data-centers while taking into account several utility criteria, the delay of virtual resources setup, the migration of existing processes, the resource utilization. In order to satisfy parties, an efficient and dynamic resource allocation strategy is mandatory. Dynamic Resource Allocation dealing with virtualization machines on physical machines. The results confirmed that the virtual machine which loading becomes too high, it will automatically migrated to another low loading physical machine without service interrupting. This can be done with the help of Skewness Algorithm. The flexible resource provisioning and migrations of machine state have improved efficiency of resource usage and dynamic resource provisioning capabilities. Allocating virtual machine to an appropriate physical machine is important to enhance the performance of cloud computing environment.

Keywords
Virtualization, Cloud Computing, Load prediction, Hot spot mitigation, Green Computing, Resource Management.

1. Introduction

One of the most significant benefits of cloud computing is reducing the operating cost of data center through virtualization to support cost reduction, the resource of physical machines (PM) in data center should be efficiently utilized. However, if the provider only considers maximizing the utilization of data centers (i.e., maximizing the utilization level of physical machines), eventually, it has a bad influence upon the performance of virtual machines (VM) in data center due to high workload of each associated physical machine.

To prevent such a performance degradation, an appropriate VM allocation scheme is needed for the overall performance of cloud computing. In addition, the provider needs to set an appropriate threshold of utilization level that does not affect to the performance degradation of virtual machines in the data center. For better performance of VMs in terms of response time, Consider the location of each VM to provide worldwide services, the provider should have several data centers according to geographically locations. Since cloud computing services are delivered over the public internet which does not guaranteed reliability in general, there may be undesirable performance degradations such as slow response time.

Although the provider can designate the allocation for new VMs to a low utilized PM that guarantees no performance degradation due to utilization level, a performance degrade is still possible to occur. If the location of a PM that is providing the user's VM is far from the location of a user, the geographical distance between the PM and the user affect to the response time of the VM.

Therefore, a cloud provider needs to consider not only the utilization level of PMs, but also the location of a PM to allocate the user request as a VM. To address these issues, this paper proposes a dynamic resource allocation model. The new model considers 1) the location of PMs, and 2) the dynamic utilization level of PMs.

2. Related Works

There are numerous advantages of cloud computing, the most basic ones being lower costs, re-provisioning of resources and remote accessibility. Cloud computing lowers cost by avoiding the capital expenditure by the company in renting the physical infrastructure from a third party provider. Due to the flexible nature of cloud computing, we can quickly access more resources from cloud providers when we need to expand our business.

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 following section discusses the significance of resource allocation.

A. Significance of Dynamic Resource Allocation

In cloud computing, Dynamic Resource Allocation is the process of assigning available resources to the needed cloud applications over the internet. Resource allocation

Manuscript received January 5, 2015
Manuscript revised January 20, 2015

starves services if the allocation is not managed precisely. Resource provisioning solves that problem by allowing the service providers to manage the resources for each individual module.

Resource Allocation Strategy is all about integrating cloud provider activities for utilizing and allocating scarce resources within the limit of cloud environment so as to meet the needs of the cloud application. It requires the type and amount of resources needed by each application in order to complete user requirements. The order and time of allocation of resources are also an input for resource allocation. From the perspective of a cloud provider, predicting the dynamic nature of users, user demands, and application demands are impractical. Cloud resources consist of physical and virtual resources.

The physical resources are shared across multiple compute requests through virtualization and provisioning. In Cloud environments, efficient resource provisioning and management present today a challenging issue because of the dynamic nature of the Cloud on one hand, and the need to satisfy heterogeneous resource requirements on the other hand.

In such dynamic environments where end-users can arrive and leave the Cloud at any time, a Cloud service provider (CSP) should be able to make accurate decisions for scaling up or down its data centers while taking into account several utility criteria, the delay of virtual resources setup, the migration of existing processes, the resource utilization, etc. In order to satisfy parties (the CSP and the end-users), an efficient and dynamic resource allocation strategy is mandatory.

Dynamic Resource Allocation dealing with virtualization machines on physical machines. The results confirmed that the virtual machine which loading becomes too high, it will automatically migrated to another low loading physical machine without service interrupting.

The remainder of this paper is organized as follows: Section 2 presents a system overview of the proposed model. Section 3 describes the proposed system of dynamic resource allocation model. Section 4 describes the experimental environment and shows the performance evaluation of the proposed model in terms of the allocation outcomes (response time of user's VM). Finally, Section 5 concludes this paper with a list of future works.

3. System Overview

The system is designed to provide flexible and on-demand service, the proposed system allows to user to request an arbitrary amount of resources at any time and from anywhere. For managing flexible user request, the proposed system is designed as a hybrid architecture that is combined with centralized and distributed resource management architectures.

Physical Machine is a real resource (i.e. it is combined different types of resources such CPU, memory, or network bandwidth) that can allocate many VMs. In a PM, there is a specialized layer for virtualization called a hypervisor. The hypervisor generally has the responsibility to allocate VMs and share its resources like traditional operating systems.

Each PM runs the hypervisor which supports one or more applications such as Web server, remote desktop, DNS, Mail, Map/ Reduce, etc. Assume all PMs Share backend storage. The load predictor predicts the future resource demands of VMs and the future load of PMs based on past statistics. Compute the load of a PM by aggregating the resource usage of its VMs. The hot spot solver detects if the resource utilization of any PM is above the hot threshold (i.e., a hot spot).

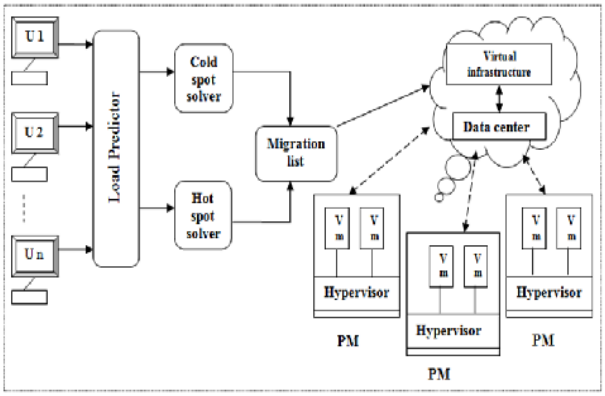

Figure No 1: System Architecture

If some PM is present VMs running on them will be migrated away to reduce their load. The cold spot solver checks if the average utilization of actively used PMs (APMs) is below the green computing threshold. If so, some of those PMs could potentially be turned off to save energy. It identifies the set of PMs whose utilization is below the cold threshold (i.e., cold spots) and then attempts to migrate away all their VMs. It then compiles a migration list of VMs and passes it to the controller for execution.

4. Proposed Work

A. Resource Management:

Resource management poses particular challenges in large-scale systems, such as server clusters that simultaneously process requests from a large number of clients. Dynamic resource management in large scale cloud environment includes the physical infrastructure and associated control functionality that enables the provisioning and management of cloud services.

Cloud resources consist of physical and virtual resources. The physical resources are shared across multiple compute requests through virtualization and provisioning. The request for virtualized resources is described through a set of parameters detailing the processing, memory and disk needs.

Provisioning satisfies the request by mapping virtualized resources to physical ones. The hardware and software resources are allocated to the cloud applications on-demand basis. The Load Predictor is the entity responsible for optimizing resource allocation. When it receives a resource request, the Load Predictor iterates through the possible subsets of available resources

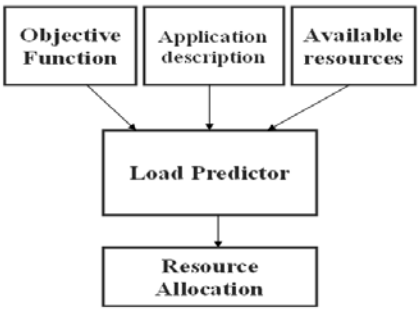

Figure No 2: Resource Allocation

Load Predictor: The Load Predictor maps resource allocation candidates to the user with respect to a given objective function.

Objective Function: The objective function defines the metric that should optimize. For example, given the increasing cost and scarcity of power in the data center, an objective function might measure the increase in power usage due to a particular allocation.

Application Description: The application description consists of three parts: i) the framework type that identifies the framework model to use, ii) workload specific parameters that describe the particular application's resource usage and iii) a request for resources including the number of VMs, storage, etc.

Available Resources: The final input required by the Load Predictor is a resource snapshot of the IaaS data centre. This includes information derived from both the virtualization layer and the IaaS monitoring service.

By using Resource Management, the Cloud providers provide resources for more number of users with less response time and can share their resources over the internet with high performance.

B. Virtualization:

In virtualization based Cloud Computing model the user not need to know the specific location of each physical device, operating system, memory, number of processor cores, middleware technology and so on. The users simply send their requests to the cloud. Through the unified user interface, send the request to the cloud system, then Cloud platform received request and allocated resources for users. The heart of virtualization is the "virtual machine" (VM), a tightly isolated software container with an operating system and application inside. Because each VM is completely separate and independent, many of them can run simultaneously on a single computer. A thin layer of software called a hypervisor decouples the VMs from the host, and dynamically allocates computing resources to each VM as needed. This architecture redefines computing equation, to deliver:

Many applications on each server. As each VM encapsulates an entire machine, many applications and operating systems can be run on one host at the same time.

Maximum server utilization, minimum server count. Every physical machine is used to its full capacity, allowing you to significantly reduce costs by deploying fewer servers overall.

Faster, easier application and resources provisioning. As self-contained software files, VMs can be manipulated with copy-and-paste ease. This brings unprecedented simplicity, speed, and flexibility to IT provisioning and management. VMs can even be transferred from one physical server to another while running, via a process known as live migration.

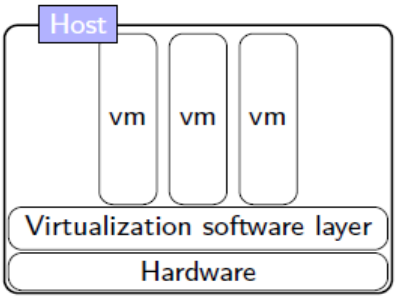

Figure No 3: Virtualization

***C.** Migration:*

Migration of VMs has been investigated as a mean to adjust data-centers utilization. The VMs are periodically reallocated using migration according to their current resource demand. Virtual machine migration takes a running virtual machine and moves it from one physical machine to another..

When a VM is running a live service it is important that this transfer occurs in a manner that balances the requirements of minimizing both downtime and total migration time. The only perceived change should be a brief slowdown during the migration and a possible improvement in performance after the migration because

the VM was moved to a machine with more available resources.

In terms of VM migration, there have been a few proposals to enable a VM to migrate from one physical machine to another for different purposes such as improving the power efficiency and satisfying performance requirements. Some migration approaches also considered the network performance and communication costs when determining migration policies.

However, the existing research heavily relies on the statistical methods to migrate the files that most frequently communicate with each other. Thus, the optimization procedure has to be conducted after a relatively long period to obtain statistics. This drawback makes the statistic method based VM allocation or migration approaches hard to fit a runtime circumstance.

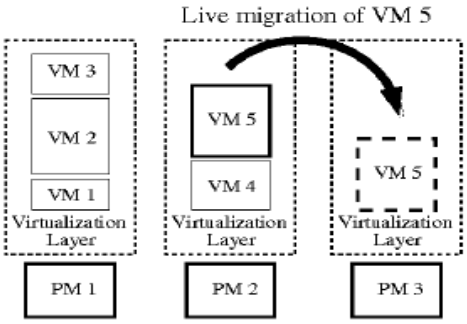

Figure No 4: Migration

Three physical machines with a virtualization layer are used to execute five virtual machines. In the initial configuration, PM 3 can be in low power state or powered down because it is not hosting any VMs. In response to demand change PM 3 is activated and VM 5 (denoted by solid lines) is migrated from PM 2 to PM 3. The migration occurs without service interruption.

D. Green Computing:

Green computing is the term used to denote efficient use of resources in computing. Core objectives of Green Computing Strategy is to Minimizing energy consumption, Purchasing green energy, Reducing the paper and other consumables used, Minimizing equipment disposal requirements and Reducing travel requirements for employees/customers. It also used for reduce costs

Computing Power Consumption has Reached a Critical Point in Data centers have run out of usable power and cooling due to high densities. Computer virtualization is the process of running two or more logical computer systems on one set of physical hardware.

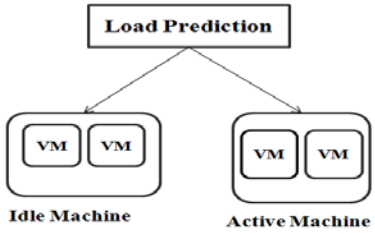

Figure No 5: Green Computing

When the resource utilization of active servers is too low, some of them can be turned off to save energy. This is handled by Green Computing. The challenge is to reduce the number of active servers during low load without sacrificing performance either now or in the future.

5. Objective of Skewness Algorithm

In proposed Dynamic Resource Allocation, the resources are allocated using Skewness Algorithm. The concept of skewness Algorithm is to quantify the unevenness in the utilization of multiple resources on a server. Let n be the number of resources consider is and r_i be the utilization of the i^{th} resource. Define the resource skewness of a server p as

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

where r is the average utilization of all resources for server p. In practice, not all types of resources are performance critical and hence only need to consider bottleneck resources in the above calculation. By minimizing the skewness, different types of workloads can combine nicely and improve the overall utilization of server resources.

A. Hot and Cold Spots

Algorithm is executed periodically to evaluate the resource allocation status based on the predicted future resource demands of VMs. The server as a hot spot if the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VMs running on it should be migrated away. The temperature of a hot spot p is defined as the square sum of its resource utilization beyond the hot threshold.

$$temperature(p) = \sum_{r \in R}(r - r_t)^2,$$

Where R is the set of overloaded resources in server p and rt is the hot threshold for resource r. The temperature of a hot spot reflects its degree of overload. If a server is not a hot spot, its temperature is zero. A server is defined as a cold spot if the utilizations of all its resources are below a cold threshold.

This indicates that the server is mostly idle and a potential candidate to turn off to save energy. Different types of resources can have different thresholds. For example, define the hot thresholds for CPU and memory resources to be 90 and 80 percent, respectively. Thus a server is a hot spot if either its CPU usage is above 90 percent or its memory usage is above 80 percent.

B. Hot Spot Mitigation

Sort the list of hot spots in the system in descending temperature (i.e., handle the hottest one first). Our goal is to eliminate all hot spots if possible. Otherwise, keep their temperature as low as possible. For each server p, first decide which of its VMs should be migrated away. Sort its list of VMs based on the resulting temperature of the server if that VM is migrated away.

The aim is to migrate away the VM that can reduce the server's temperature the most. In case of ties, select the VM whose removal can reduce the skewness of the server the most. For each VM in the list, the algorithm finds a destination server to accommodate it. The server must not become a hot spot after accepting this VM. Among all such servers, algorithm select one whose skewness can be reduced the most by accepting this VM. Note that this reduction can be negative which means algorithm selects the server whose skewness increases the least.

If a destination server is found, the algorithm records the migration of the VMs to that server and updates the predicted load of related servers. Otherwise, move onto the next VM in the list and try to find a destination server for it. As long as find a destination server for any of its VMs, Consider this run of the algorithm a success and then move onto the next hot spot. Note that each run of the algorithm migrates away at most one VM from the overloaded server. This does not necessarily eliminate the hot spot, but at least reduces its temperature. If it remains a hot spot in the next decision run, the algorithm will repeat this process. It is possible to design the algorithm so that it can migrate away multiple VMs during each run. But this can add more load on the related servers during a period when they are already overloaded. The algorithm decide to use this more conservative approach and leave the system some time to react before initiating additional migrations.

C. 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. The challenge is to reduce the number of active servers during low load without sacrificing performance either now or in the future. Green computing algorithm is invoked when the average utilizations of all resources on active servers are below the green computing threshold. Sort the list of cold spots in the system based on the ascending order of their memory size. Since algorithm needs to migrate away all its VMs before shut down an under-utilized server, define the memory size of a cold spot as the aggregate memory size of all VMs running on it.

For a cold spot p, check if all of its VMs are migrated somewhere else. For each VM on p, try to find a destination server to accommodate it. The resource utilizations of the server after accepting the VM must be below the warm threshold. The energy can be saved by consolidating under-utilized servers, overdoing it may create hot spots in the future. The warm threshold is designed to prevent that.

If multiple servers satisfy the above criterion, then prefer one that is not a current cold spot. This is because increasing load on a cold spot reduces the likelihood that it can be eliminated. However, it will accept a cold spot as the destination server if necessary.

All things being equal, then select a destination server whose skewness can be reduced the most by accepting this VM. If the destination server is finding for all VMs on a cold spot, then record the sequence of migrations and update the predicted load of related servers. Otherwise, do not migrate any of its VMs.

The list of cold spots is also updated because some of them may no longer be cold due to the proposed VM migrations in the above process. The above consolidation adds extra load onto the related servers. This is not as serious a problem as in the hot spot mitigation case because green computing is initiated only when the load in the system is low.

Restrict the number of cold spots that can be eliminated in each run of the algorithm to be no more than a certain percentage of active servers in the system. This is called the consolidation limit. Note that the elimination of cold spots in the system is done only when the average load of all active servers (APMs) is below the green computing threshold.

6. Analysis of Skewness Algorithm

The skewness algorithm consists of three parts: load prediction, hot spot mitigation, and green computing. Let n and m be the number of PMs and VMs respectively. The number of resources (CPU, memory, I/O, etc.) that need to be considered is usually a small constant. Thus the computation of the skewness and the temperature metrics for a single server takes a constant amount of time. During load prediction, The time complexity is O(n+m).

A. Complexity of Hot Spot Mitigation

For hot spot mitigation, let nh be the number of hot spots in the system during a decision. Sorting them based on their temperature takes O(nh*log(nh)). Hence, the sorting takes a constant amount of time. For each VM, need to scan the rest of the PMs to find a suitable destination for it, which takes O(n). The overall complexity of this phase is thus O(nh * n).

B. Complexity of Green Computing

For the green computing phase, let nc be the number of cold spots in the system during a decision run. Sorting them based on their memory sizes takes O(nc*log(nc)). For each VM in a cold spot, it takes O(n) time to find a destination server for it. The overall complexity of this phase is O(nc *n).

7. Conclusion and Future Work

Dynamic migrations of virtual machines are becoming an interesting opportunity to allow cloud infrastructures to accommodate changing demands for different types of processing with heterogeneous workloads and time constraints. The proposed system multiplexes virtual to physical resources adaptively based on the changing demand.

The skewness algorithm metric is used to combine VMs with different resource characteristics appropriately so that the capacities of servers are well utilized. The algorithm achieves both overload avoidance and green computing for systems with multi resource constraints. This improves the performance of the Cloud Data Centers.

The use of computing resources as a delivered service is an important development in the world. At present Cloud Computing is a promising paradigm for delivering IT services as computing utilities. People around the world are making use of different cloud services provided by many companies.

Security is the important factor that everyone thinks before choosing a cloud provider. The security issues can be solved by providing security for the user resources while allocating resources dynamically. This will improve the performance and security of Data centers.

REFERENCES

[1] Zhen Xiao, Senior Member, IEEE, Weijia Song, and Qi Chen-"Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment"- IEEE Transaction on parallel and Distributed System, vol. 24, no. 6, June 2013.

[2] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in Proc. of the Symposium on Networked Systems Design and Implementation (NSDI'05), May 2005.

[3] G. Chen, H. Wenbo, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy aware server provisioning and load dispatching for connection-intensive internet services," in Proc. of the USENIX Symposium on Networked Systems Design and Implementation (NSDI'08), Apr. 2008.

[4] P. Padala, K.-Y. Hou, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant, "Automated control of multiple virtualized resources," in Proc. of the ACM European conference on Computer systems (EuroSys'09), 2009.

[5] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing sla violations," in Proc. of the IFIP/IEEE International Symposium on Integrated Network Management (IM'07), 2007.

[6] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," in Proc. Of the ACM Symposium on Operating System Principles (SOSP'01), Oct. 2001.

[7] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise data centers," in Proc. Of the International World Wide Web Conference (WWW'07), May 2007.

[8] A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: integration and load balancing in data centers," in Proc. of the ACM/IEEE conference on Supercomputing, 2008.

[9] T. Das, P. Padala, V. N. Padmanabhan, R. Ramjee, and K. G. Shin, "Litegreen: saving energy in networked desktops using virtualization," in Proc. of the USENIX Annual Technical Conference, 2010.

[10] Y. Agarwal, S. Savage, and R. Gupta, "Sleepserver: a software-only approach for reducing the energy consumption of pcs within enterprise environments," in Proc. of the USENIX Annual Technical Conference, 2010.

[11] 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. 2003.

[12] 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.

[13] N. Bila, E.d. Lara, K. Joshi, H.A. Lagar-Cavilla, M. Hiltunen, and M. Satyanarayanan, "Jettison: Efficient Idle Desktop Consolidation with Partial VM Migration," Proc. ACM European Conf. Computer Systems (EuroSys '12), 2012.