Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 12, December 2013, pg.50 – 54

SURVEY ARTICLE

Survey on Efficient Dynamic Resource Allocation in Cloud

Kaleeswari.P[1], Guru Rani.G[1]
PG Student[1], Assistant Professor[2], Department of CSE
NPR college of Engineering and Technology, TamilNadu, India
Email: kaleeswari22@gmail.com; ranitcguru@yahoo.com

Abstract-Cloud computing is on demand as it offers dynamic flexible resource allocation for reliable and guaranteed services in pay as-you-use manner to public. In Cloud computing multiple cloud users can request number of cloud services simultaneously.So there must be a provision that all resources are made available to requesting user in efficient manner to satisfy their need. In this paper, virtualization technology is proposed for efficient dynamic resource allocation.

Keywords-Cloud Computing; Dynamic Resource Allocation; Virtual Machine; Virtualization Technology

I. INTRODUCTION

Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software and information are provided to users over the network. Cloud computing providers deliver application via the Internet, which are accessed from web browser, while the business software and data are stored on servers at a remote location. Cloud computing really is accessing resources and services needed to perform functions with dynamically changing needs. The cloud is a virtualization of resources that maintains and manages itself.

Cloud computing nowadays becomes quite popular among a community of cloud users by offering a variety of resources.

Cloud computing platforms, such as those provided by

Microsoft, Amazon, Google, IBM, and Hewlett-Packard, let developers deploy applications across computers hosted by a central organization. These applications can access a large network of computing resources that are deployed and managed by a cloud computing provider. Developers obtain the advantages of a managed computing platform, without having to commit resources to design, build and maintain the network. Yet, an important problem that must be addressed effectively in the cloud is how to manage QoS and maintain SLA for cloud users that share cloud resources.

In cloud platforms, resource allocation takes place at two levels. First, when an application is uploaded to the cloud, the load balancer assigns the requested instances to physical computers, attempting to balance the computational load of multiple applications across physical computers. Second, when an application receives multiple incoming requests, these requests should be each assigned to a specific application instance to balance the computational load across a set of instances of the same application. For example,

Amazon EC2 uses elastic load balancing (ELB) to control how incoming requests are handled. Application designers can direct requests to instances in specific availability zones, to specific instances, or to instances demonstrating the shortest response times. In the following sections a review of existing resource allocation techniques like Topology Aware Resource

Allocation, Linear Scheduling and Resource Allocation for parallel data processing is described briefly.

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 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.

Dynamic Resource Allocation Strategy (DRAS) 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 a user job. The order and time of allocation of resources are also an input for an optimal DRAS. An optimal DRAS should avoid the following criteria as follows:

a) **Resource contention** situation arises when two applications try to access the same resource at the same time.
b) S**carcity of resources** arises when there are limited resources.
c) **Resource fragmentation** situation arises when the resources are isolated. There will be enough resources but not able to allocate to the needed application.
d) **Over-provisioning** of resources arises when the application gets surplus resources than the demanded one.
e) **Under-provisioning** of resources occurs when the application is assigned with fewer numbers of resources than the demand.

The dynamic resource allocation based on distributed multiple criteria decisions in computing cloud. In it author contribution is tow-fold, first distributed architecture is adopted, and in which resource management is divided into independent tasks, each of which is performed by Autonomous Node Agents (NA) in a cycle of three activities: (1) VMPlacement, 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 its total resources use by hosted VM are monitored by NA, (3) In VMSelection, 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.

II. LITERATURE SURVEY

In[1] author proposed a virtualization technology for dynamic resource allocation in cloud computing environment.The virtualization technology is used 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 multidimensional resource utilization of a server. By minimizing skewness, we can combine different types of workloads nicely and improve the overall utilization of server resources.

Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources. This mapping is largely hidden from the cloud users. Users with the Amazon EC2 service for example, do not know where their VM instances run. It is up to the cloud provider to make sure the underlying physical machines (PMs) have 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. However, 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 heterogenous because multiple generations of hardware coexist in a data center.

In this paper, we present the design and implementation of an automated resource management system that achieves a good balance between the two goals. We make the following contributions:

a) We develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used.
b) We introduce the concept of “skewness” to measure the uneven utilization of a server. By minimizing skewness, we can improve the overall utilization of servers in the face of multidimensional resource constraints.
c) We design a load prediction algorithm that can capture the future resource usages of applications accurately without looking inside the VMs. The algorithm can capture the rising trend of resource usage patterns and help reduce the placement churn significantly.

In [2] author proposed xen, an x86 virtual machine monitor which allows multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion, but without sacrificing either performance or functionality. This is achieved by providing an idealized virtual machine abstraction to which operating systems such as Linux, BSD and Windows XP, can be ported with minimal effort. Our design is targeted at hosting up to 100 virtual machine instances simultaneously on a modern server. The virtualization approach taken by Xen is extremely efficient. We allow operating systems such as Linux and Windows XP to be hosted simultaneously for a negligible performance overhead.

Full virtualization was never part of the x 86 architectural designs. Certain supervisor instructions must be handled by the VMM for correct virtualization, but executing these with insufficient privilege fails silently rather than causing a convenient trap. Efficiently virtualizing the x86 MMU is also difficult. These problems can be solved, but only at the cost of increased complexity and reduced performance.In Oder to overcome these problems to introduce the paravirtualization. Paravirtualization is necessary to obtain high performance and strong resource isolation on uncooperative machine architectures such as x86.Xen currently uses an algorithm called the Borrowed Virtual Time algorithm to schedule domains.It also improves the performance because paravirtualization allows many OS instances to run concurrently on a single physical machine.

In[3] author proposed migration of virtual machine which is used to migrating operating system instances across distinct physical hosts is a useful tool for administrators of data centers and clusters.It allows a clean separation between hardware and software, and facilitates fault management, load balancing and low-level system maintenance. Migrating an entire OS and all of its applications as one unit allows us to avoid many of the difficulties faced by process-level migration approaches. The narrow interface between a virtualized OS and the virtual machine monitor (VMM) makes it easy avoid the problem of residual dependencies in which the original host machine must remain available and network-accessible in order to service certain system calls or even memory accesses on behalf of migrated processes.

Live OS migration is an extremelely powerful tool for cluster administrators, allowing separation of hardware and software considerations and consolidating clustered hardware into a single coherent management domain. If a physical machine needs to be removed from service an administrator may migrate OS instances including the applications that they are running to alternative machines, freeing the original machine for maintenance. We achieve this live migration by using a pre-copy approach in which pages of memory are iteratively copied from the source machine to the destination host, all without ever stopping the execution of the virtual machine being migrated. Page level protection hardware is used to ensure a consistent snapshot is transferred, and a rate-adaptive algorithm is used to control the impact of migration traffic on running services. Finally, the virtual machine copies any remaining pages to the destination and resumes execution there. Live OS migration is used to balancing downtime and total migration time and also avoid problem of residual dependencies.

In[4] authors propose an VMware ESX Server which is a thin software layer designed to multiplex hardware resources efficiently among virtual machines running unmodified commodity operating systems. This paper introduces several novel ESX Server mechanisms and policies for managing memory. A ballooning technique reclaims the pages considered least valuable by the operating system running in a virtual machine. An idle memory tax achieves efficient memory utilization while maintaining performance isolation guarantees. Content-based page sharing and hot I/O page remapping exploit transparent page remapping to eliminate redundancy and reduce copying overheads. These techniques are combined to efficiently support virtual machine workloads that overcommit memory.

In many computing environments, individual servers are underutilized, allowing them to be consolidated as virtual machines on a single physical server with little or no performance penalty. Similarly, many small servers can be consolidated onto fewer larger machines to simplify management and reduce costs. ESX Server manages system hardware directly, providing significantly higher I/O performance and complete control over resource management. High-level resource management policies compute a target memory allocation for each VM based on specified parameters and system load. These allocations are achieved by invoking lower-level mechanisms to reclaim memory from virtual machines.

ESX Server maintains a pmap data structure for each VM to translate physical page numbers (PPNs) to machine page numbers (MPNs). VM instructions that manipulate guest OS page tables or TLB contents are intercepted, preventing updates to actual MMU state. Separate shadow page tables, which contain virtual-to-machine page mappings, are maintained for use by the processor and are kept consistent with the physical-to-machine mappings in the pmap. This approach permits ordinary memory references to execute without additional overhead, since the hardware TLB will cache direct virtual-to-machine address translations read from the shadow page table.ESX Server supports overcommitment of memory to facilitate a higher degree of server consolidation than would be possible with simple static partitioning. Overcommitment means that the total size configured for all running virtual machines exceeds the total amount of actual machine memory. The system manages the allocation of memory to VMs automatically based on configuration parameters and system load.

ESX Server uses a ballooning technique to achieve such predictable performance by coaxing the guest OS into cooperating with it when possible. In this ballooning technique, a small balloon module is loaded into the guest OS as a pseudo-device driver or kernel service. It has no external interface within the guest and communicates with ESX Server via a private channel. When the server wants to reclaim memory, it instructs the driver to inflate by allocating pinned physical pages within the VM using appropriate native interfaces. Similarly, the server may deflate the balloon by instructing it to deallocate previously-allocated pages.

Server consolidation presents numerous opportunities for sharing memory between virtual machines. For example, several VMs may be running instances of the same guest OS, have the same applications or components loaded or contain common data. ESX Server exploits these sharing opportunities, so that server workloads running in VMs on a single machine often consume less memory than they would run on separate physical machines. As a result, higher levels of overcommitment can be supported efficiently.

In[5] authors propose a Dynamic server provisioning technique for energy consumption in hosting internet services.This dynamic server provisioning techniques are effective in turning off unnecessary servers to save energy. Internet services such as search, web-mail, online chatting, and online gaming, have become part of people's everyday life. Such services are expected to scale well, to guarantee performance and to be highly available. To achieve these goals, these services are typically deployed in clusters of massive number of servers hosted in dedicated data centers. Each data center houses a large number of heterogeneous components for computing, storage, and networking, together with an infrastructure to distribute power and provide cooling.

Data center energy savings can come from a number of places: on the hardware and facility side, e.g., by designing energy-efficient servers and data center infrastructures and on the software side, e.g., through resource management. In this paper, we take a software-based approach, consisting of two interdependent techniques such as dynamic provisioning and load dispatching. In dynamic provisioning that dynamically turns on a minimum number of servers required to satisfy application specific quality of service and in load dispatching that distributes current load among the running machines.

Our approach is motivated by two observations from real data sets collected from operating Internet services. First, the total load of a typical Internet service fluctuates over a day. For example, the fluctuation for the number of users logged on to Windows Live Messenger can be about 40% of the peak load within a day. Second, an active server, even when it is kept idle, consumes a non-trivial amount of power. The first observation provides us the opportunity to dynamically change the number of active servers, while the second observation implies that shutting down machines during off -peak period provides maximum power savings.

In this paper, we develop power saving techniques for connection services and evaluate the techniques using data traces from Windows Live Messenger (formerly MSN Messenger), a popular instant messaging service with millions of users. We consider server provisioning and load dispatching in a single framework, and evaluate various load skewing techniques to trade off between energy saving and quality of service. Although the problem is motivated by Messenger services, the results should apply to other connection-oriented services. The contributions of the paper are

a) We characterize performance, power, and user experience models for Windows Live Messenger connection servers based on real data collected over a period of 45 days.

b) We design a common provisioning framework that trades off power saving and user experiences. It takes into account the server transient behavior and accommodates various load dispatching algorithms.

c) We design load skewing algorithms that allow significant amount of energy saving (up to 30%) without sacrificing user experiences, i.e., maintaining very small number of SIDs.

In performance model, to characterize the effect of load dispatching on service loads, it is important to understand the relationship between application level parameters such as user login and physical parameters such as CPU utilization and power consumption. In other words, we need to identify the variables that significantly affect CPU and power. This would enable us to control CPU usage and power consumption of the servers by controlling these variables.

In this paper, Load balancing algorithms are used for try to make the numbers of connections on the servers the same or as close as possible. By setting a uniform login rates for all the servers, regardless of the fluctuations of departure rates on individual servers , it leaves the number of connections on the servers diverge without proper feedback control. The principle of load skewing is exactly the opposite of load balancing. Here new login requests are routed to busy servers as long as the servers can handle them. The goal is to maintain a small number of tail servers that have small number of connections. When user login requests ramp up, these servers will be used as reserve to handle login increases and surge, and give time for new servers to be turned on.The load skewing algorithms generate small number of SIDs when turning off servers, they also maintain unnecessary tail servers when the load ramps up. So a hybrid (switching) algorithm that employs load balancing when load increases and load-skewing when load decreases seems to be able to get the best energy-SID tradeoff.

In [6], 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. In each reevaluation process, the schedulers are re-calculating the finish time of their respective submitted tasks, not the tasks that are assign to that cloud.

The problem of resource allocation is considered in [7], to optimize the total profit gained from the multi dimensional SLA contracts for multi-tire application. In it the upper bound of total profit is provided with the help of force-directed resource assignment (FRA) heuristic algorithm, in which initial solution is based on provided solution for profit upper bound problem. Next, distribution rates are fixed and local optimization step is use for improving resource sharing. Finally, a resource consolidation technique is applied to consolidate resources to determine the active (ON) servers and further optimize the resource assignment.

The previous work on web application scalability implemented for static load balancing solution with server clusters but the dynamic scaling of web applications in virtualized cloud computing has not been much discussed. Because such kinds of work load require minimum response time and high level of availability and reliability from web applications. A solution for dynamic scaling of web application provided in [8] by describing an architecture to scale web application in dynamic manner, based on

threshold in a virtualized cloud computing environment. As per the demand these virtual machines are started and provisioned by a provisioning subsystem. But the action of provisioning and de-provisioning of web server virtual machine instances control by a dynamic scaling algorithm based on relevant threshold of web application. Efficient resource manager for dynamic virtualized resources allocation in cloud environment was presented in [9]. This proposed mechanism consists of three components: User Interface (UI), Subscriber server (SS) and Resource Manager (RM). User can access cloud in authorized manner with corresponding user name and password, through user interface. Resource manager accepts VMs request from user interface and subscription message to Subscriber server. The responsibility of Subscriber server is to maintain and manage the user profiles.

III. CONCLUSIONS

A literature review shows that dynamic resource allocation is growing need of cloud providers for more number of users and with the less response time. Hence the on-demand resource allocation based SLA as per defined task priority helps to satisfy the efficient provisioning of cloud resources to multiple cloud users.

REFERENCES

[1] Zhen Xiao, Senior Member, IEEE, Weijia Song, and Qi Chen proposed a "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment", IEEE Transactions on Parallel and Distributed System, vol.24, No.6, June 2013.
[2]P.Barham,B.Dragovic,K.Fraser,S.Hand,T.Harris,A.Ho,R.Neugebauer,I.Pratt,and A.Warfield Proposed a "Xen and the Art of Virtualization," Proc. ACM Symp.Operating Systems Principles(SOSP'03),Oct.2003.
[3] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul,C. Limpach, I.Pratt, and A. Warfield Proposed a "Live Migration of Virtual Machines,"Proc. Symp. Networked Systems Design and Implementation (NSDI '05), May 2005.
[4] C.A. Waldspurger Proposed a "Memory Resource Management in Vmware ESX Server," Proc. Symp. Operating Systems Design and Implementation (OSDI '02), Aug. 2002.
[5] G. Chen, H. Wenbo, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao Proposed a "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services," Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI '08), Apr. 2008.
[6] Jiayin Li, Meikang Qiu, Jian-Wei Niu, Yu Chen, Zhong Ming, "Adaptive Resource Allocation for Pre-empt able Jobs in Cloud Systems," in 10th International Conference on Intelligent System Design and Application, Jan. 2011, pp. 31-36.
[7] Goudarzi H., Pedram M., "Multi-dimensional SLA based Resource Allocation for Multi-tier Cloud Computing Systems," in IEEE International Conference on Cloud Computing, Sep. 2011, pp. 324-331.
[8] Chieu T.C., Mohindra A., Karve A.A., Segal A.,"Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment," in IEEE International Conference on e-Business Engineering, Dec. 2009, pp. 281-286.
[9] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer, Ivan Breskovic, "SLA-Aware Application Deployment and Resource Allocation in Clouds", 35th IEEE Annual Computer Software and Application Conference Workshops, 2011, pp. 298-303.

