Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Allocation of Resources in Cloud Computing Environment

A. Nagesh, Shraddha Agarwal, Padma Naveen Kumar
Department of Computer Science and Engineering,
JNTU, Hyderabad, India

Abstract— Cloud computing allows the scaling of resources as per the needs of the customers. In this paper, we present a system, in which the cloud administrator allocates the domain registered by the cloud user as per the SLA plan etc. If the domain requested by the user is free to be allocated, then the cloud administrator allocates the domain as per the usage requirements to the cloud user. The utilization of server and the resources are analyzed by the cloud administrator.

Keywords— VMM, XEN, green computing, virtualization, SLA

I. INTRODUCTION

Cloud computing, the long held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. They need not be concerned about over-provisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or under- provisioning for one that becomes wildly popular, thus missing potential customers and revenue [1]. In some ways the cloud is already hanging in the sky, especially for the consumers. According to a recent study, 69% of the Americans connected to the web use some kind of "cloud service", including web-based e-mail or online data storage. The best example is Google, the biggest online search company by far, which now offers a plethora of web-based applications such as word-processing or online spreadsheets [2].

Modern computers are sufficiently powerful to use virtualization to present the illusion of many smaller *virtual machines* (VMs), each running a separate operating system instance. This has led to a resurgence of interest in VM technology. Xen, a high performance resource-managed virtual machine monitor (VMM) enables applications such as server consolidation, co-located hosting facilities, distributed web services, secure computing platform and application mobility [3]. Successful partitioning of a machine to support the concurrent execution of multiple operating systems poses several challenges. Firstly, virtual machines must be isolated from one another: it is not acceptable for the execution of one to adversely affect the performance of another. This is particularly true when virtual machines are owned by mutually untrusting users. Secondly, it is necessary to support a variety of different operating systems to accommodate the heterogeneity of popular applications. Thirdly, the performance overhead introduced by virtualization should be small. The elasticity and the lack of upfront capital investment offered by cloud computing is appealing to many businesses. There is a lot of discussion on the benefits and costs of the cloud model and on how to move legacy applications onto the cloud platform. Here we study a different problem: how can a cloud service provider best multiplex its virtual resources onto the physical hardware? This is important because much of the touted gains in the cloud model come from such multiplexing. Studies have found that servers in many existing data centers are often severely under-utilized due to over-provisioning for the peak demand. The cloud model is expected to make such practice unnecessary by offering automatic scale up and down in response to load variation [5]. Besides reducing the hardware cost, it also saves on electricity which contributes to a significant portion of the operational expenses in large data centers.

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 [5]. 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 heterogeneous because multiple generations of hardware co-exist in a data center [6]. 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 project, we presented the design and implementation of dynamic resource allocation in the Virtualized Cloud Environment which maintains the balance between the overload avoidance and green computing [7]. 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.) [8].

In this paper, we present a system, which allocates the resources requested by the cloud user on the server, according to the memory requirement and other criteria and the user can access the resource once it has been approved by the cloud administrator. Here, the resource is nothing but the domain requested by the cloud user.

II. LITERATURE SURVEY

In [11], Zhen Xiao 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 heterogeneous 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 allocate the resources requested by the cloud user as per the Service Level Agreement and the memory usage of the resource on the cloud.

We allocate the resource requested by the user on the server of user choice and check if it gets overloaded. If the server is not overloaded, then the resource is allocated on that server which the user can make use of.

In [12], Barham 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 order 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 [13], Fraser 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 extremely 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 [14], C.A. Waldspurger 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 project 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 over commit 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. 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 project, 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.

III. PROPOSED SYSTEM

In this paper, we present a system that allocates the resources requested by the cloud user such as domain of a website on a server after the approval of a cloud administrator.

The cloud user initially gets registered for accessing the cloud resources. If he is an already existing user, then he can get authenticated using his username and password. If he not an existing user, then he has to get registered by filling the registration form. He can login to the cloud website using this username and password. The cloud user has various functionalities like he can register for a domain, view his domain status, view his details, creation of a site, viewing the site status (which the cloud administrator has to approve), view all the hosted sites by the user. The cloud user registers for a domain of his choice, by providing the details such as the username of cloud user, the domain for which he is registering, the SLA plan (the number of months for which he is registering the domain), the memory requirements of the domain on the server, and other personal details. After successfully registering for the domain, the user is directed for the payment options. The payment can be done using credit card and all the card details have to be specified by the cloud user in order to successfully register for the domain. The cloud user can view the details of the domain registering by him in the ***viewdomaindetails*****.** The cloud user has to add a site to the domain registered by him, by uploading a html file, which is the home page of the site on the domain registered by the user.

The cloud administrator has to accept the domain registration request of the cloud user in order to access the domain successfully by the user. The cloud administrator logs into his account on the cloud website to check the pending requests by all the cloud users and allocates the domain requested by them, if the domain does not violate the memory requirements of each server. The cloud administrator adds the site on the server which has minimum overload and then hosts the site. The cloud administrator can view the memory left on all the servers and also the resource allocation graph which varies as per the resources allocated on the servers. The site which has been hosted can now be successfully used by the cloud user. The site hosting status will be visible to the cloud user in his user account on the cloud website, in the ***viewhostedsites*** tab.

IV. RESULTS

Fig 1: Cloud user entering new domain

Fig 1 shows the domain name options that can be entered by the cloud user for domain allocation by cloud administrator.

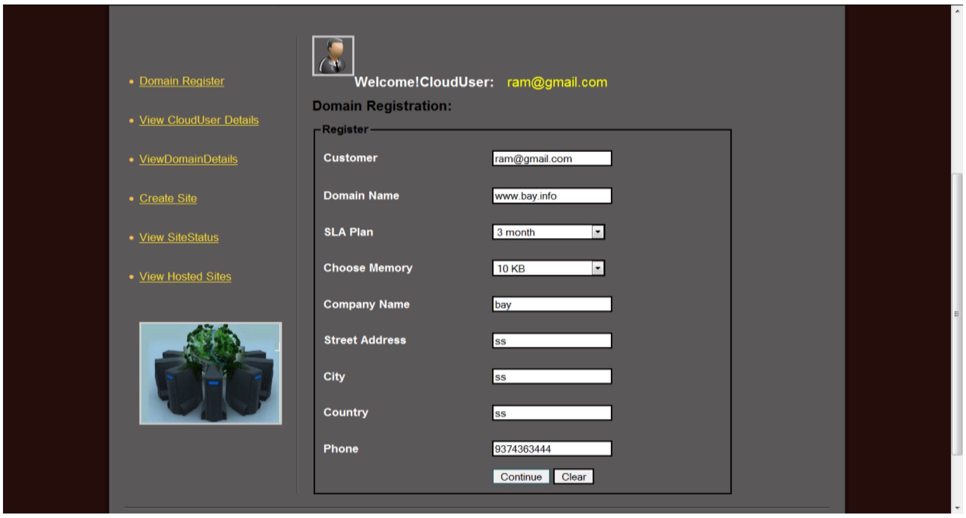

Fig 2: Memory required for domain

Fig 2 shows the details that have to be entered by the cloud user for domain registration.

Fig 3: Site creation

Fig 3 shows how a site can be created by the cloud user using the domain with which he has been registered. An HTML file is uploaded, so that it forms the home page of the site created.

Fig 4: Adding site to server

Fig 4 shows how a cloud administrator adds a site to a server based on its memory requirements, thereby avoiding overload on the server.

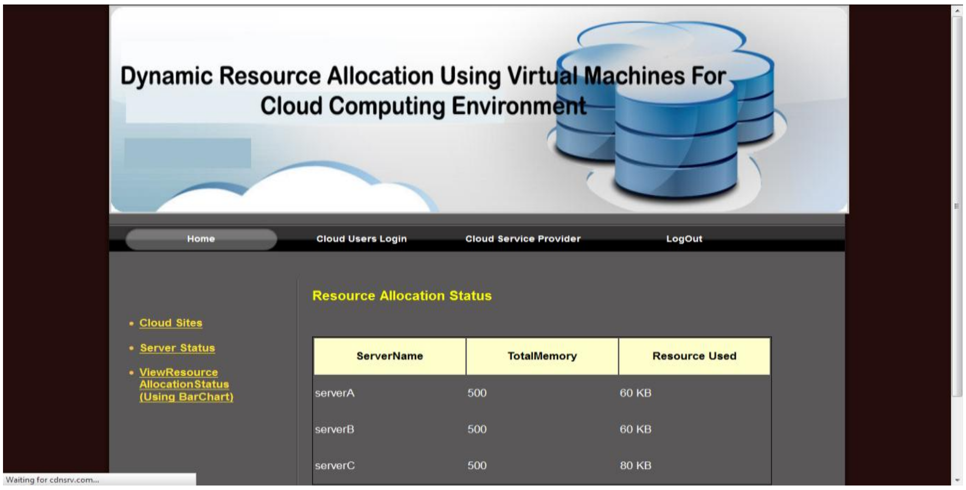

Fig 5: Resource allocation status

Fig 6 shows the total memory of each server and the allocation status of the memory to the domains on all the servers. The cloud administrator can check the status of memory on each server and allocate domain on the server which has been utilized minimally, thus avoiding the server overload.

Fig 6: Graph for resource allocation

Fig 6 shows the allocation status of memory on all the servers in the form of a graph, which acts as a visual cue to the cloud administrator in the process of allocating servers to the domains.

Fig 7: Site status in cloud user account

Fig 7 shows the status of the domain, if approved or rejected by the cloud administrator.

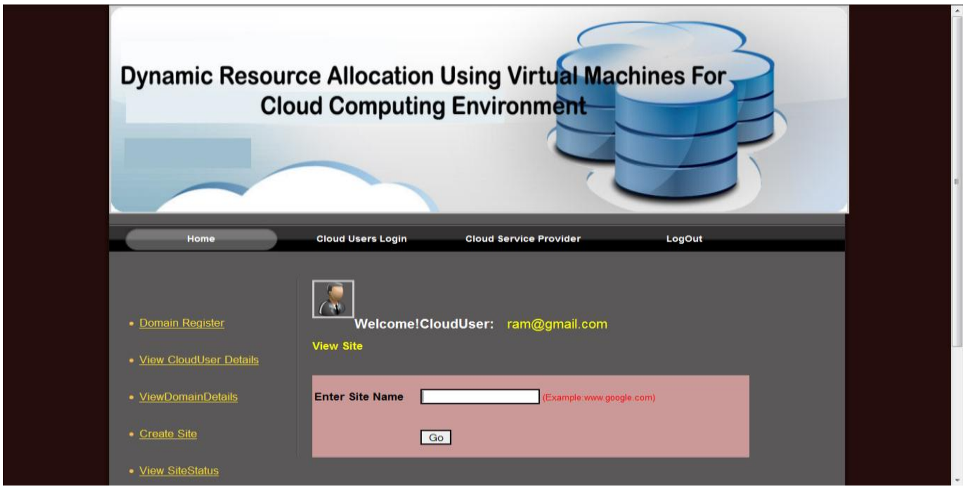

Fig 8: View site

Fig 8 shows the viewing of site by entering the site name that has been provided by cloud user at the time of creation of site.

Fig 9: View site

Fig 9 show sthe home page of the site created by the cloud user.

V. CONCLUSION

In this paper, 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. Our implementation achieves both overload avoidance and effective resource allocation for systems with multi-resource constraints. The systems presented by us, will thus effectively utilize the server resources and avoid the domain usage by multiple clients. The cloud users can choose his service level agreement plan and the memory required by him for the registered domain and pay for the domain as per the usage. Thus, the system is cost as well as resource effective.

REFERENCES

[1] M. Armbrust *et al.*, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep., Feb 2009.

[2] L. Siegele, "Let it rise: A special report on corporate IT," in *The Economist*, 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," in *Proc. of the ACM Symposium on Operating Systems Principles (SOSP'03)*, Oct. 2003.

[4] "Amazon elastic compute cloud (Amazon EC2),http://aws.amazon.com/ec2/."

[5] 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.

[6] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast transparent migration for virtual machines," in *Proc. of the USENIX Annual Technical Conference*, 2005.

[7] M. McNett, D. Gupta, A. Vahdat, and G. M. Voelker, “Usher: An extensible framework for managing clusters of virtual machines,” in *Proc. of the Large Installation System Administration Conference (LISA’07)*, Nov. 2007.

[8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, “Black-box and gray-box strategies for virtual machine migration,” in *Proc. Of the Symposium on Networked Systems Design and Implementation (NSDI’07)*, Apr. 2007.

[9] C. A. Waldspurger, “Memory resource management in VMware ESX server,” in *Proc. of the symposium on Operating systems design and implementation (OSDI’02)*, Aug. 2002.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] C.A. Waldspurger Proposed a “Memory Resource Management in Vmware ESX Server,” Proc. Symp. Operating Systems Design and Implementation (OSDI ’02), Aug. 2002.

[15] 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.

