International Journal of Grid and Distributed Computing
Vol. 9, No. 5 (2016), pp.25-30
http://dx.doi.org/10.14257/ijgdc.2016.9.5.03

A Desirable Strategy for Resource Allocation using Virtual Machine in Cloud

B. Abinaya. [1], J. Suganthi[2], R. G. Suresh Kumar[3] and T. Nalini[4]

[1,2]Master of Technology, Department of computer science and Engineering, Rajiv Gandhi College of Engineering and Technology, Pondicherry, India
[3]Research Scholar, Vels University, Chennai, India
[4]Professor, Department of computer science and Engineering, Bharath University, Chennai, India
[1]abinayabskr25@gmail.com , [2]suganthi.sofi@gmail.com, [3]aargeek@gmail.com
[4]brnalinichidimbaram@gmail.com

Abstract

Cloud computing is a facsimile of legalizing ubiquitous, expedient, on-demand network access to a shared pool of configurable computing resources that can be rapidly furnished and released with negligible management effort. It relies on sharing computing resources rather than having local servers or personal devices to handle applications. The resource allocation, still lack on sustaining tools that enable developers to compare different resource allocation strategies in cloud computing. In this paper we initiate the concept of "skewness" to measure the bumpy utilization of a server. By minimizing skewness, we can improve the overall utilization of servers in the face of multidimensional resource constraints. Here we use skewness metric to combine VMs with different resource characteristics suitably so that the capacities of servers are well utilized.

Keywords*: Skewness, virtual machine, resource allocation, cloud computing*

1. Introduction

Cloud computing is an expression used to narrate a variety of computing concepts that necessitate a large number of computers connected through a real-time communication network such as the Internet. In science, cloud computing is a synonym for distributed computing over a network, and means the ability to run a program or application on many connected computers at the same time. The phrase also more commonly refers to network-based services, which appear to be provided by real server hardware, and are in fact served up by virtual hardware, simulated by software running on one or more real machines. Such virtual servers do not physically exist and can therefore be moved around and scaled up (or down) on the fly without affecting the end user - arguably, rather like a cloud. The popularity of the term can be attributed to its use in marketing to sell hosted services in the sense of application service provisioning that run client server software on a remote location.

Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. The cloud also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. For example, a cloud computer facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach should

ISSN: 2005-4262 IJGDC

maximize the use of computing powers thus reducing environmental damage as well since less power, air conditioning, rackspace, *etc*. is required for a variety of functions.

1.1. Cloud Management

Legacy management infrastructures, which are based on the concept of dedicated system relationships and architecture constructs, are not well suited to cloud environments where instances are continually launched and decommissioned. Instead, the dynamic nature of cloud computing requires monitoring and management tools that are adaptable, extensible and customizable.

Cloud computing presents a number of management challenges. Companies using public clouds do not have ownership of the equipment hosting the cloud environment, and because the environment is not contained within their own networks, public cloud customers don't have full visibility or control. Users of public cloud services must also integrate with an architecture defined by the cloud provider, using its specific parameters for working with cloud components. Integration includes tying into the cloud APIs for configuring IP addresses, subnets, firewalls and data service functions for storage. Because control of these functions is based on the cloud provider's infrastructure and services, public cloud users must integrate with the cloud infrastructure management. Capacity management is a challenge for both public and private cloud environments because end users have the ability to deploy applications using self-service portals. Applications of all sizes may appear in the environment, consume an unpredictable amount of resources, and then disappear at any time.

Chargeback or, pricing resource use on a granular basis is a challenge for both public and private cloud environments. Chargeback is a challenge for public cloud service providers because they must price their services competitively while still creating profit Users of public cloud services may find chargeback challenging because it is difficult for IT groups to assess actual resource costs on a granular basis due to overlapping resources within an organization that may be paid for by an individual business unit, such as electrical power. For private cloud operators, chargeback is fairly straightforward, but the challenge lies in guessing how to allocate resources as closely as possible to actual resource usage to achieve the greatest operational efficiency. Exceeding budgets can be a risk.

1.2. Cloud Client

Users access cloud computing using networked client devices, such as desktop computers, laptops, tablets and smartphones. Some of these devices – cloud clients – rely on cloud computing for all or a majority of their applications so as to be essentially useless without it. Examples are thin clients and the browser-based Chromebook. Many cloud applications do not require specific software on the client and instead use a web browser to interact with the cloud application. With Ajax and HTML5 these Web user interfaces can achieve a similar, or even better, look and feel to native applications. Some cloud applications, however, support specific client software dedicated to these applications (e.g., virtual desktop clients and most email clients). Some legacy applications (line of business applications that until now have been prevalent in thin client computing) are delivered via a screen-sharing technology.

2. Related Work

Cloud infrastructures must accommodate changing demands for different types of processing with heterogeneous workloads and time constraints. In a similar context, dynamic management of virtualized application environments is becoming very important to exploit computing resources, especially with recent virtualization capabilities that allow

live sessions to be moved transparently between servers. Mauro Andreolini et.al proposed Dynamic Load Management of Virtual Machines. Their schemes offer novel management algorithms to decide about reallocations of virtual machines in a cloud context characterized by large numbers of hosts [1]. The novel algorithms identify just the real critical instances and take decisions without recurring to typical thresholds. Moreover, they consider load trend behavior of the resources instead of instantaneous or average measures. Work by Jeffrey S. Chase et.al provided the system is based on an economic approach to managing shared server resources, in which services "bid" for resources as a function of delivered performance [2]. The system continuously monitors load and plans resource allotments by estimating the value of their effects on service performance.

A greedy resource allocation algorithm adjusts resource prices to balance supply and demand, allocating resources to their most efficient use. Aameek Singh et.al proposed Integration and Load Balancing in Data Centers and their scheme offer novel load balancing algorithm called VectorDot for handling the hierarchical and multi-dimensional resource constraints in such systems [3]. The algorithm, inspired by the successful Toyoda method for multi-dimensional knapsacks, is the first of its kind. We evaluate our system on a range of synthetic and real data center testbeds comprising of VMware ESX servers, IBM SAN Volume Controller, Cisco and Brocade switches. Experiments under varied conditions demonstrate the end-to-end validity of our system and the ability of VectorDot to efficiently remove overloads on server, switch and storage nodes. Gong Chen et.al proposed Energy-Aware Server Provisioning and Load Dispatching for Connection Intensive Internet Services and it provide unique properties, performance, and power models of connection servers, based on a real data trace collected from the deployed Windows Live Messenger [4]. Their scheme design server provisioning and load dispatching algorithms can save a significant amount of energy without sacrificing user experiences. Pradeep Padala et.al proposed Automated Control of Multiple Virtualized Resources. Their scheme proposed Auto Control, means a resource control system that automatically adapts to dynamic changes in a shared virtualized infrastructure to achieve application SLOs [5]. Autocontrol is a combination of an online model estimator and a novel multi-input, multi-output (MIMO) resource controller.

3. Existing System

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. The capacity of PMs can also be heterogeneous because multiple generations of hardware coexist in a data center.

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 two main disadvantages are overload avoidance and green computing.

4. Proposed System

In this paper, we present the design and implementation of an automated resource management system that achieves a good balance between the two goals. Two goals are overload avoidance and green computing. 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. Green computing: 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.

We develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used. 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. 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.

5. Implementation

5.1. System Architecture

The system architecture consists of the virtual machine scheduler. The virtual machine scheduler predicts the hot spot and cold spot and manages the migration. This is done in order to increase the energy efficiency through green computing. N number of physical machines is used for the implementation. Our algorithm executes periodically to evaluate the resource allocation status based on the predicted future resource demands of VMs. We define a 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. We define the temperature of a hot spot p as the square sum of its resource utilization beyond the hot threshold: We define a server as a cold spot if the utilizations of all its resources are below a cold threshold.

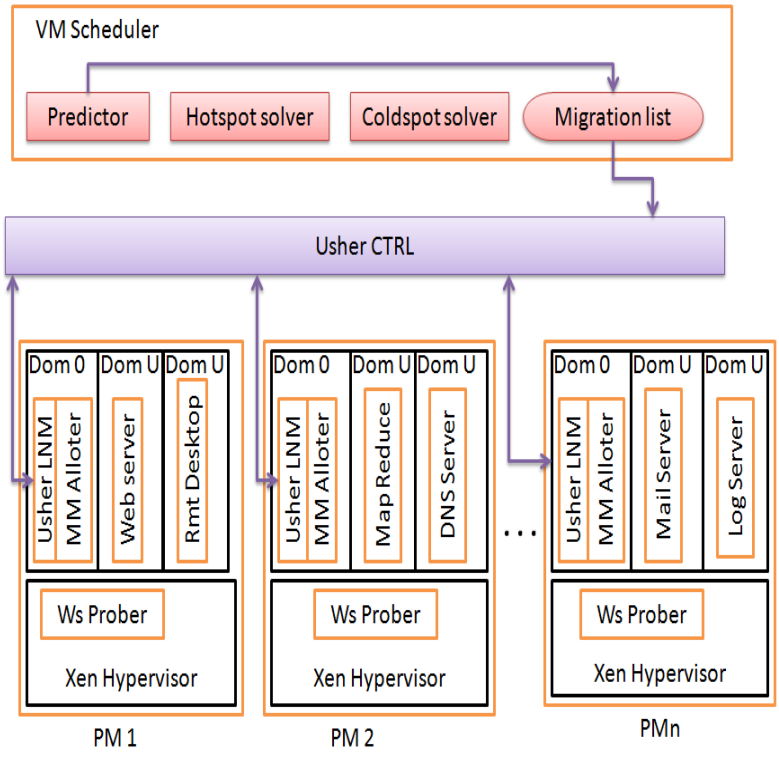

Figure 1. System Architecture of Resource Allocation

This indicates that the server is mostly idle and a potential candidate to turn off to save energy. However, we do so only when the average resource utilization of all actively used

servers (i.e., APMs) in the system is below a green computing threshold. A server is actively used if it has at least one VM running. Otherwise, it is inactive. Finally, we define the warm threshold to be a level of resource utilization that is sufficiently high to justify having the server running but not so high as to risk becoming a hot spot in the face of temporary fluctuation of application resource demands. We sort the list of hot spots in the system in descending temperature (i.e., we 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, we first decide which of its VMs should be migrated away. We sort its list of VMs based on the resulting temperature of the server if that VM is migrated away.

We aim to migrate away the VM that can reduce the server's temperature the most. In case of ties, we select the VM whose removal can reduce the skewness of the server the most. For each VM in the list, we see if we can find a destination server to accommodate it. The server must not become a hot spot after accepting this VM. Among all such servers, we select one hose skewness can be reduced the most by accepting this VM. Note that this reduction can be negative which means we select the server whose skewness increases the least. 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 here is to reduce the number of active servers during low load without sacrificing performance either now or in the future. We need to avoid oscillation in the system. Our green computing algorithm is invoked when the average utilizations of all resources on active servers are below the green computing threshold. We sort the list of cold spots in the system based on the ascending order of their memory size. Since we need to migrate away all its VMs before we can shut down an under-utilized server, we define the memory size of a cold spot as the aggregate memory size of all VMs running on it. Recall that our model assumes all VMs connect to share back-end storage. Hence, the cost of a VM live migration is determined mostly by its memory footprint.

6. Conclusion

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

References

[1] M. Andreolini, S. Casolari, M. Colajanni and M. Messori, "Dynamic load management of virtual machines", Institute for the computer science, social informatics and Telecommunication Engineering, **(2010)**, pp. 201-204.
[2] J. S. Chase, D. C. Anderson, P. N. Thakar and A. M. Vahdat, "Managing Energy and Server Resources in Hosting Centers", Proceedings of the 18th ACM Symposium on Operating System Principles (SOSP).
[3] A. Singh, M. Korupolu and D. Mohapatra, "Server-storage virtualization: Integration and load balancing in data centers", International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1-12.
[4] 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", Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI'08), **(2008)**.
[5] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt and A. Warfield, "Live migration of virtual machines", Proceedings of the Symposium on Networked Systems Design and Implementation (NSDI'05), **(2005)**.
[6] 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", Proceedings of the ACM European conference on Computer systems (EuroSys'09), **(2009)**.
[7] C. A. Waldspurger, "Memory resource management in VMware ESX server", in Proceedings of the symposium on Operating systems design and implementation (OSDI'02), **(2002)**.

[8] T. Wood, P. Shenoy, A. Venkataramani and M. Yousif, "Black-box and gray-box strategies for virtual machine migration", Proceedings Of the Symposium on Networked Systems Design and Implementation (NSDI'07), **(2007)**.
[9] L. He and J. Walrand, "Pricing Differentiated Internet Services", IEEE Transactions on Cloud computing, **(2011)**.
[10] 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", Proceedings of the ACM Symposium on Operating Systems Principles (SOSP'03), **(2003)**.
[11] N. Bobroff, A. Kochut and K. Beaty, "Dynamic placement of virtual machines for managing sla violations", Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM'07), 2007. L. Siegele, "Let it rise: A special report on corporate IT," in The Economist, Oct.
[12] M. McNett, D. Gupta, A. Vahdat and G. M. Voelker, "Usher: An extensible framework for managing clusters of virtual machines", Proceedings of the Large Installation System Administration Conference (LISA'07), **(2007)**.
[13] M. Nelson, B.-H. Lim and G. Hutchins, "Fast transparent migration for virtual machines", Proceedings of the USENIX Annual Technical Conference, **(2005)**.

