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. 3, Issue. 12, December 2014, pg.30 – 36

RESEARCH ARTICLE

Optimizing Resource Allocation through Virtual Machines in Cloud Computing

Valli Kumar M[1], V.Santosh Kumar[2], Sirisha K L[3]

Student, Department of CSE, Sreyas Institute of Engineering and Technology, Hyderabad, India [1]

Associate Professor, Department of CSE, Sreyas Institute of Engineering and Technology, Hyderabad, India [2]

Asst.Professor, Dept. of CSE, Keshav Memorial Institute of Technology, Hyderabad, India [3]

[1] mvallikumar@gmail.com, [2] veenu.santosh@gmail.com, [3] klssirisha@gmail.com

---

Abstract-- Cloud computing has emerged to be a modern computing model based on Internet. This technology can enable organizations and individuals to share state-of-the-art computing resources in pay as you use fashion. Virtualization is the technology that enabled cloud computing to be affordable. The cloud resources are to be utilized optimally. In order to achieve this dynamic resource allocation is the best solution. However, the dynamic resource allocation is a challenging problem to be addressed. In this paper we proposed an algorithm that takes care of dynamic resource allocation using two possible cases. In the first case capacity of a VM is increased at runtime while in the second case VM migration takes place to different physical machine. We built a prototype application to demonstrate the simulations for proof of concept. We considered memory as a resource. The experimental results are encouraging.

Index Terms – Cloud computing, virtualization, load balancing, optimal resource allocation

I. INTRODUCTION

Virtualization is the technology that enabled cloud computing to be affordable and viable. Resource virtualization at various levels made cloud computing possible. As cloud resources are costly and need huge investment, from the service provider point of view it is essential to have dynamic resource allocation so as to optimize the utilization of resources. If there is no such mechanism, it ends up with consumption of more resources that ultimately leads to unsuccessful implementation of cloud. Many algorithms or techniques came into existence. They include MUSE [14] where replicas are maintained for optimal performance; load balancing strategies in [15]; integrated load dispatching approach in [10]; delay scheduling [18], HARMONY where multiple resource layers are used [20]; and skewness algorithm for dynamic resource allocation [28]. These works focused on various strategies. However, in this paper we use hybrid approach to solve the problem of dynamic resource allocation.

In this paper we proposed a novel architecture to solve the problem of dynamic resource allocation. Our contributions in this paper are as described here.

- We proposed architecture for cloud infrastructure to achieve optimal resource allocation.
- We proposed an algorithm by name over-commitment mitigation algorithm for dynamic resource allocation and optimization of resource allocation and utilization. This algorithm has different strategies to handle the problem of dynamic resource allocation.
- We built a web based prototype application that demonstrates the proof of concept. The application simulates the cloud environment and dynamic resource allocation as per the proposed algorithm.

The remainder of the paper is structured as follows. Section II provides review of literature that exists about the prior works on dynamic resource allocation problem. Section III presented the proposed architecture. Section IV provides the algorithm proposed. Section V presents experimental results while section VI concludes the paper.

II. RELATED WORKS

With respect to data centers and resource allocation to various applications at application level was given importance to research in [15] and [14]. Load dispatching and server provisioning were introduced in [10]. However, these works did not use virtual machines. Nevertheless, these experiments were done in multi-tier architectures. The work proposed in [28] targeted a cloud environment that is Amazon EC2-style environment where experiments are made with dynamic resource allocation. There was concept of VMs and their allocations. In [16] MapReduce is explored which is a framework for distributed programming. Scheduling algorithm was used in [17] to minimize cost while executing jobs. The delay scheduling concept was introduced in [18] for effectiveness with local data processing. Dynamic priorities to jobs was explored in [19] for optimal resource allocation.
Live VM migration concept was explored in [20], [12] and [8]. In this approach a VM is migrated from one physical resource to another physical resource. In [21] load balancing mechanisms are explored. The load balancing concept has been around and it is used in all cloud environments where server machines are given requests based on their load. In this paper we combined many strategies and built a hybrid algorithm that can cope with various runtime requirements and resource dynamics for optimal resource allocation and utilization.

III. HYBRID APPROACH FOR LOAD PREDICTION AND DYNAMIC RESOURCE ALLOCATION

In cloud, virtualization plays an important role in resource utilization. In fact the virtualization technology made the cloud computing paradigm affordable as it can improve the utilization of physical resources. With less physical resources it is possible to serve more number of requests from clients. In this process virtualization mechanism in cloud data center is crucial. Therefore we considered the optimal resource allocation problem as non-trivial and studied the existing algorithms that are used for dynamic resource allocation. In this section we propose an architecture that can have our proposed algorithm running to allocate resources dynamically in order to optimize the usage of cloud resources. The architecture focuses on VM assignments, overload estimation and other features that will leverage the functionality of cloud data center. Figure 1 shows an overview of our architecture.

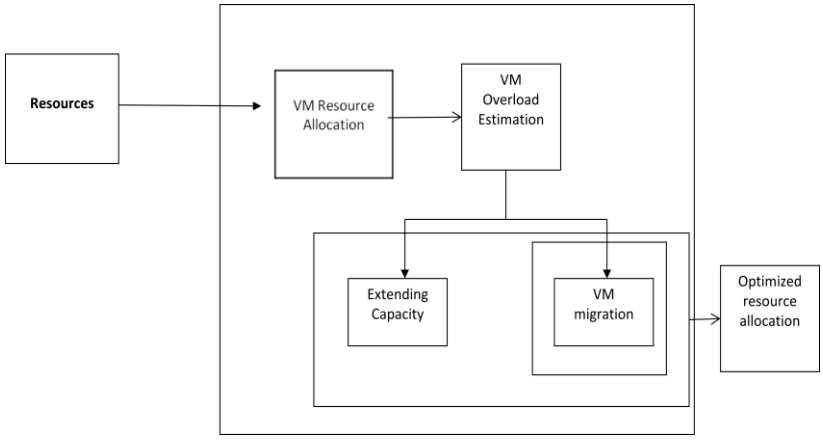

Figure 1 – Architectural overview of the proposed system

Here is the description about the proposed architecture. There are number of physical resources on top of which number of virtual machines is assigned. The physical resources have certain capacity limit. However, the physical resources are available so as to allocation of new physical machine is possible for business continuity and disaster mitigation. Nevertheless, it is imperative that the data center should have strategies to allocate resources optimally. Towards it we could identify two strategies which are very useful. The first strategy is known as extending the capacity of physical machine while the second strategy is migrating VM from one physical machine to another physical machine. When any VM experiences over commitment our algorithm (described in the next section) starts working. It combines two possible cases for optimizing resource allocation. The first case is, when a VM over commitment is witnessed, the VM's capacity gets increased by using either free memory of PM or idle memory of other VM. The second case is that a VM migration technique is employed so as to move over-committed VM to other PM. Thus the proposed algorithm achieves optimal resource allocation dynamically.

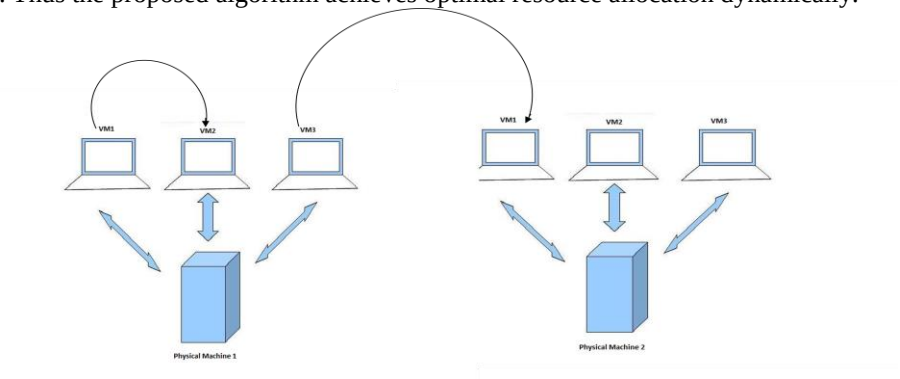

Figure 2 – Live VM migration

As shown in Figure 2, the VM3 of physical machine 1 is migrated to physical machine 2 as per the runtime dynamics according to the proposed algorithm which is described in ensuing section.

IV. PROPOSED ALGORITHM

This algorithm is meant for achieving optimal resource utilization in cloud computing environment. This is in tune with the architecture proposed in the previous section. Optimized resource allocation is the goal of this algorithm. It takes the available resources as input and allocates resources optimally. The algorithm is strategically executed in cloud data center so as to ensure optimal resource utilization that can increase the scalability of the cloud.

Algorithm: Over-Commitment Mitigation Algorithm
Inputs : Resources Availability
Outputs : Optimized Resource Allocation

For each PM in Data Center
 IF any VM is overloaded THEN
CASE 1: CAPACITY EXTENSION
  IF PM has available free memory THEN
    Allocate the free memory to VM in question
  END IF
  IF other VMs have idle memory THEN
    Allocate the idle memory of other VM to the VM who has been overloaded
  END IF

STEP 2: LIVE VM MIGRATION
  IF capacity extension fails THEN
    Identify other PM with enough resources
    Migrate the VM who has been overloaded to new PM
  END IF
END

Listing 1 – Over commitment mitigation algorithm

As seen in Listing 1, the algorithm combines two possible cases for optimizing resource allocation. The first case is, when a VM over commitment is witnessed, the VM's capacity gets increased by using either free memory of PM or idle memory of other VM. The second case is that a VM migration technique is employed so as to move over-committed VM to other PM. Thus the proposed algorithm achieves optimal resource allocation dynamically. This will increase resource utilization and avoids unnecessary deployment of physical resources. This phenomenon is also linked to the possible reduction of investment risks in cloud computing from the stand point of cloud service provider.

V. EXPERIMENTAL RESULTS

We built a prototype web application, a custom simulator, which demonstrates the proof of concept. The empirical results are encouraging. The environment used for simulation study include JDK 1.7, Tomcat 7.0, MY SQL, Chrome running in Windows 7 machine with 4 GB RAM, core 2 dual processor. The prototype demonstrates our architecture and the underlying algorithm. Local machine resources are utilized in order to simulate the cloud. The study reveals that it is possible to achieve successful optimization of resource allocation and utilizations by using the proposed algorithm named "Over-Commitment Mitigation Algorithm". Here are some snapshots to understand how the simulation results reveal the dynamic resource allocation as per the proposed solution.

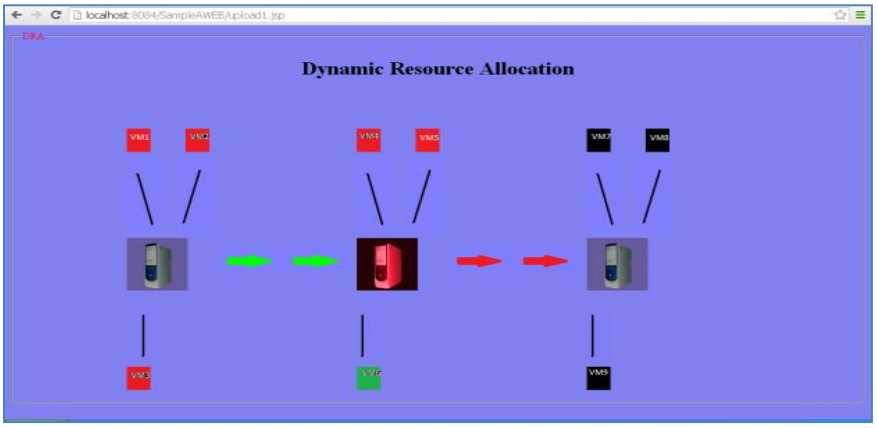

Figure 3 – Dynamic resource allocation

As seen in Figure 4, it is evident that resource allocation could not be possible with physical machine as indicated in red color VMs. The resource allocation is done for a VM on physical machine 2.

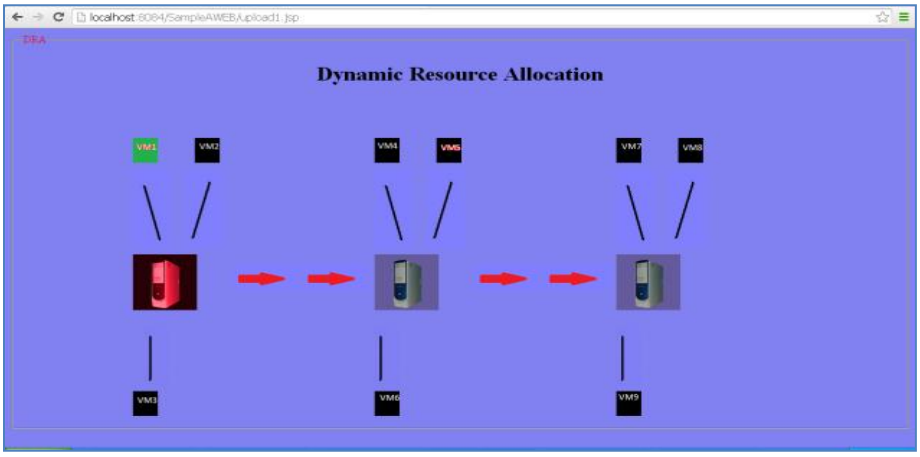

Figure 4 – Dynamic resource allocation

As seen in Figure 4, it is evident that resource allocation could be possible with physical machine 1 as indicated in green color VM. The resource allocation is done for a VM on physical machine 1.

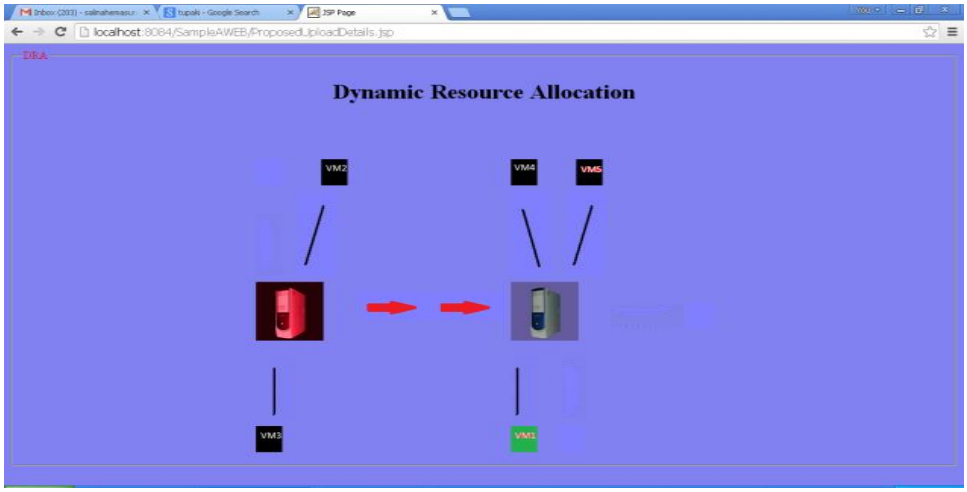

Figure 5 – Dynamic resource allocation

As seen in Figure 5, it is evident that VM migration took place. Initially there were 3 VMs with first physical machine. As one of the VMs experienced over-commitment and there was no possibility of extending resources for VM, it is migrated to other physical machine where resources are available. The migrated VM is shown in green color.

VI. CONCLUSION

In this paper, the dynamic resource allocation is studied in the context of resource allocation in cloud computing. As cloud became a reality and virtualization concept is used to make it successful, there is resource allocation concern from cloud service provider point of view. From the review of literature it is understood that there are many approaches to dynamic resource allocation including skewness. In this paper we focused on a hybrid approach that considers two cases. The first case is to see the possible increase in resources to JVM while the second case is live VM migration from one physical machine to another physical machine. We proposed an algorithm to this effect. We also built a custom simulator which demonstrates the proof of concept. The experimental results visualize the two cases as per the experiences with regard to resource utilization dynamics at runtime. In future we would like to implement this algorithm in the real cloud environment.

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] 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.
[12] 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.
[13] "TPC-W: Transaction processing performance council, http://www.tpc.org/tpcw/."
[14] 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.
[15] 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.
[16] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, and I. Stoica, "Improving MapReduce performance in heterogeneous environments," in *Proc. of the Symposium on Operating Systems Design and Implementation (OSDI'08)*, 2008.
[17] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: Fair scheduling for distributed computing clusters," in *Proc. of the ACM Symposium on Operating System Principles (SOSP'09)*, Oct. 2009.
[18] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling," in *Proc. of the European conference on Computer systems (EuroSys'10)*, 2010.
[19] T. Sandholm and K. Lai, "Mapreduce optimization using regulated dynamic prioritization," in *Proc. of the international joint conference on Measurement and modeling of computer systems (SIGMETRICS'09)*, 2009.
[20] 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.
[21] Y. Toyoda, "A simplified algorithm for obtaining approximate solutions to zero-one programming problems," *Management Science*, vol. 21, pp. 1417–1427, august 1975.
[22] R. Nathuji and K. Schwan, "Virtualpower: coordinated power management in virtualized enterprise systems," in *Proc. of the ACM SIGOPS symposium on Operating systems principles (SOSP'07)*, 2007.
[23] D. Meisner, B. T. Gold, and T. F. Wenisch, "Powernap: eliminating server idle power," in *Proc. of the international conference on Architectural support for programming languages and operating systems (ASPLOS'09)*, 2009.
[24] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, "Somniloquy: augmenting network interfaces to reduce pc energy usage," in *Proc. of the USENIX symposium on Networked systems design and implementation (NSDI'09)*, 2009.
[25] 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.
[26] 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.
[27] 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," in *Proc. of the ACM European conference on Computer systems (EuroSys'12)*, 2012.

AUTHORS

Valli Kumar Masam received the MCA degree in computer science and technology from Jawaharlal Nehru Technological University, Hyderabad, India in 2011. He is currently working towards his M.Tech degree in Sreyas Institute of Engineering and Technology, Hyderabad, India. His research interests include data mining and cloud computing.

Vennu Santosh Kumar received the Masters degree in Computer Science and Engineering in the year 2010. He is Microsoft Certified System Engineer & CISCO Certified Network Administrator, he worked as a System Engineer in WIPRO Technologies(INDIA). In 2011 he joined as an Associate Professor at Sreyas Institute of Engineering and Technology in Computer Science Department. He has been involved in several tutorials, workshops, technical paper presentations .His research interests are focused on Computer Networks, Network Security & Mobile Computing.

Sirisha K L S received the Masters degree in Computer Science and Engineering in the year 2010. In 2010 she joined as an Assistant Professor at Keshav Memorial Institute of Technology in Computer Science and Engineering Department. Her research interests are focused on Network Security, Information Retrieval & Machine learning.

