Resource Optimization and Allocation over Virtualized Cloud System

Kanika Takkar[1] Neha Khatri [2]
[1,2]Department of Computer Science
[1,2]CBS Group Of Institutions, Maharishi Dayanand University, Rohtak

Abstract— Cloud Computing Is A Model For Enabling Ubiquitous, Convenient, On-Demand Network Access To A Shared Pool Of Configurable Computing Resources (E.G., Networks, Servers, Storage, Applications, And Services) That Can Be Rapidly Provisioned And Released With Minimal Management Effort Or Service Provider Interaction. This Paper Reviews Resource Optimization and Allocation Over Virtualized Cloud System. 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[1].In A Virtualized Cloud Computing Environment, Customers May Never Know Exactly Where Their Data Is Stored. In Fact, Data May Be Stored Across Multiple Data Centers In An Effort To Improve Reliability, Increase Performance, And Provide Redundancies. This Geographic Dispersion May Make It More Difficult To Ascertain Legal Jurisdiction If Disputes Arise. 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. All Implementation Work Is Carried Out In Java Netbeans And Matlab (Matrix Laboratory) Could Be A Problem-Oriented Language And Interactive Surroundings For Numerical Computation, Image, And Programming. Matlab Is De Facto Normal For Analyzing Information, Developing Algorithms, And Making Models And Applications.

Key words: Virtualized Cloud System, Virtual machine monitors, Virtual Machines

I. INTRODUCTION

In a virtualized cloud computing environment, customers may never know exactly where their data is stored. In fact, data may be stored across multiple data centers in an effort to improve reliability, increase performance, and provide redundancies. This geographic dispersion may make it more difficult to ascertain legal jurisdiction if disputes arise.As discussed earlier, a host can simultaneously instantiate multiple VMs and allocate cores based on predefined processor sharing policies (space-shared, time-shared). Every VM component has access to a component that stores the characteristics related to a VM, such as memory, processor, storage, and the VM's internal scheduling policy, which is extended from the abstract component called VM Scheduling A system which can automatically scale its infrastructure resources is designed in. The system composed of a virtual network of virtual machines capable of live migration across multi- domain physical infrastructure. By using dynamic availability of infrastructure resources and dynamic application demand, a virtual computation environment is able to automatically relocate itself across the infrastructure and scale its resources. But the above work considers only the non-preempt able scheduling policy.

Several researchers have developed efficient resource allocations for real time tasks on multiprocessor system. But the studies, scheduled tasks on fixed number of processors.

A. Advantages of Using Virtualization

Despite many difficulties that accompany the VM scheduling problem, using them as the basic building blocks of a distributed system is very beneficial. In this section, we discuss the two main benefits, namely, easy reservation and VM elasticity.

Other Advantages include:

1) Ease of Resource Reservation:

Since distributed systems are used in a widespread range of applications, the need for delivering special resources at particular times has been essential. As a result, a set ofresource reservation mechanisms have been proposed. *VM*

2) Elasticity:

Many scheduling challenges arise from the nature of processing units available in today's computer systems.

B. Understanding a Virtual Machine

A more sophisticated neuron is the McCulloch and Pitts model Neuron [19]. The main difference from the previous model is that the inputs are 'weighted'; the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These inputs (weighted) are added together and if they exceed a preset threshold value, the neuron fires. Else the neuron does not fire.

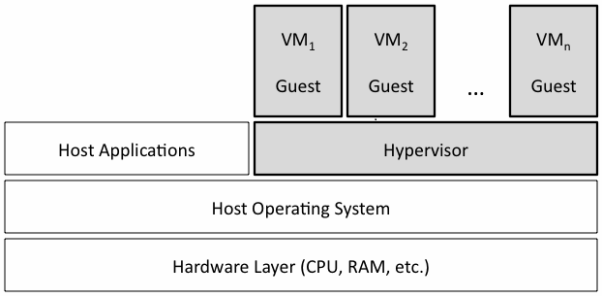

Fig. 1.4: Virtual Machine Architecture

A system which can automatically scale its infrastructure resources is designed in. The system composed of a virtual network of virtual machines capable of live migration across multi- domain physical infrastructure. By using dynamic availability of infrastructure resources and dynamic application demand, a virtual computation environment is able to automatically relocate itself across the infrastructure and scale its resources. But the above work considers only the non-preempt able scheduling policy.

II. RESEARCH PROPOSED WORK

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 reduction of Physical Machines used.

A. Overload avoidance

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.

B. Reduction of PM

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.

Following are the Objectives of my work:

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

(2) We will 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.

(3) We will measure the uneven utilization of a server on cloud system in the process. By minimizing uneven utilization, we can improve the overall utilization of servers in the face of multidimensional resource constraints.

(4) We will validate our approach by conducting a performance evaluation study using the simulation toolkit.

Tools which are used in my work are:

C. Hardware Requirements:

- Processor: Pentium IV or Higher
- RAM: I GB or Higher
- Hard Disk: 10 GB or Higher

D. Software Requirements:

- Platform: Windows 7 Professional
- Java 7, Netbeans 6.9 and above
- Matlab 2012b for Interpolation of results, and for graphs and charts.

III. PRACTICAL IMPLEMENTATION AND RESULT

Firstly the described resource allocation approach adaptively allocates the system resources of servers to their services in runtime in order to satisfy the requirements of multiple cloudlets.

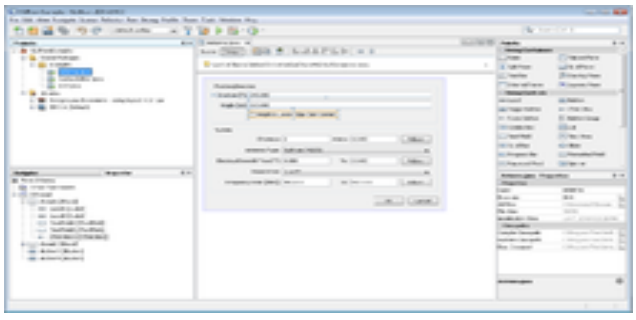

Fig. 5.1 NetBeans GUI Builder

A. Result

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Finish Time
9	Success	3	16	120	120.2
21	Success	3	16	120	120.2
33	Success	3	16	120	120.2
1	Success	2	2	160	160.2
13	Success	2	2	160	160.2
25	Success	2	2	160	160.2

Then start MATLAB and import the data which is collect as input and target and run

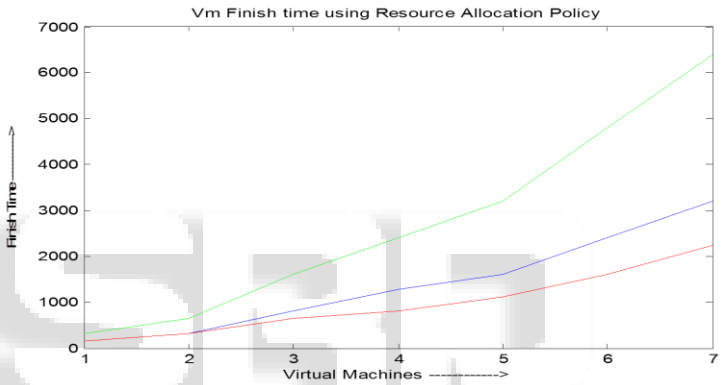

Fig. 5.2: Finish time of against other algorithms

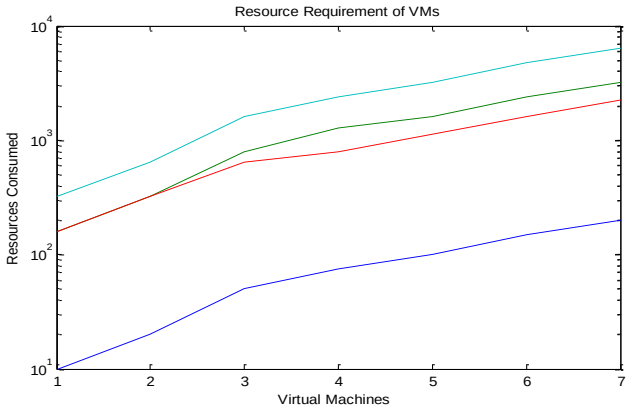

Fig.5.3: Resource Requirement of VMs for proposed algorithm (blue) against other algorithms

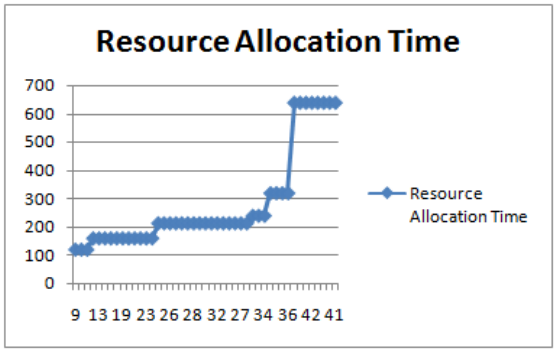

Fig. 5.4: Time for Resource Allocation for a given VM

IV. CONCLUSION

Cloud computing offers utility oriented IT services to users globally. It is Based on a pay as you go model, it enable hosting of pervasive applications from customer, scientific, and business domains. The data centers hosting Cloud applications consume large amounts of electrical energy, contributing to high operational costs and carbon footprints to the environment. The basic principle of cloud computing is that user data is not stored locally but is stored in the data center of internet.

In this paper we have discussed mainly about 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 reduction of Physical Machines used.

V. FUTURE SCOPE

Our main objective was to develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used. We were able to successfully designe a Resource algorithm that can capture the 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.

The allocation scheme was based upon of queue based resource allocation policy to manage resources dynamically over the cloud systems. The allocation approach works by adaptively allocating the system resources of servers to their services in runtime in order to satisfy the requirements of multiple cloudlets. The demonstration results show that our dynamic resource allocation approach can substantially increase the throughput of a cloud system. Future research may include the extension of our adaptive resource allocation approach to QoS features, such as timeliness, accuracy and security..

REFERENCES

[1] N.Krishnaveni, G.Sivakumar,"Survey on Dynamic Resource Allocation Strategy in Cloud Computing environment", Dept. of CSE Erode Sengunthar Engineering College Thudupathi,India, International Journal of Computer Applications Technology and Research, Vol. 2, Issue 6, pp. 731 - 737,2013.

[2] Shabnam Khan,"A survey on scheduling based resource allocation in cloud computing", Computer Science and Engineering Dept., Sobhasaria Engineering College, Sikar,Rajasthan, India. International Journal For Technological Research In Engineering, Vol. 1, Issue. 1, Sep – 2013.

[3] VaghelaAnkita,"A survey on various resource allocation policies in cloud computing environment", Department of Computer Engineering, Alpha College of Engineering and Technology, Gujarat, India, Vol. 2, Issue 5, pp. 760 – 763.

[4] Anshul Rai, Ranjita Bhagwan, Saikat Guha,"GeneralizedResource Allocation for the Cloud", Microsoft Research India.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of Cloud computing environments for scaling of application services," in Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'10), ser. Lecture Notes in Computer Science, vol. 6081. Busan: Springer, May 2010, pp. 13–31.

[6] "Amazon Elastic Compute Cloud (Amazon EC2)," http://aws.amazon. com/ec2.

[7] J. Varia, "Best practices in architecting Cloud applications in the AWS Cloud," in Cloud Computing: Principles and Paradigms, R. Buyya, J. Broberg, and A. Goscinski, Eds. Wiley Press, 2011, ch. 18, pp. 459–490.

[8] Q. Zhang, E. Gurses, R. Boutaba, and J. Xiao, "Dynamic resource alloca- ¨ tion for spot markets in Clouds," in Proceedings of the 2nd Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE '11). Boston: USENIX, Mar. 2011.

[9] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Resource pool management: Reactive versus proactive or let's be friends," Computer Networks, vol. 53, no. 17, pp. 2905 – 2922, 2009, virtualized Data Centers.

[10] ´I. Goiri, F. Julia, J. Fit ` o, M. Mac ´ ´ias, and J. Guitart, "Resource-level QoS metric for CPU-based guarantees in Cloud providers," in Economics of Grids, Clouds, Systems, and Services, ser. Lecture Notes in Computer Science, J. Altmann and O. Rana, Eds. Springer Berlin / Heidelberg, 2010, vol. 6296, pp. 34–47

[11] M. Gallet, N. Yigitbasi, B. Javadi, D. Kondo, A. Iosup, D. Epema, A model for space-correlated failures in large-scale distributed systems, in: Proceedings of the 16th International European Conference on Parallel and Distributed Computing, Euro-Par 2010, Springer-Verlag, Berlin, Ischia, Italy, 2010, pp. 88–100

[12] Javadi, Bahman, Jemal Abawajy, and Rajkumar Buyya. "Failure-aware resource provisioning for hybrid Cloud infrastructure." Journal of parallel and distributed computing 72, no. 10 (2012): 1318-1331.

[13] D.G. Feitelson, Workload Modeling for Computer Systems Performance Evaluation, e-Book. http://www.cs.huji.ac.il/~feit/wlmod/, 2009

[14] M.D. deAssunção, R. Buyya, S. Venugopal, InterGrid: a case for Internetworking islands of Grids, Concurrency andComputation: Practice andExperience20 (8) (2008) 997–1024

[15] D. Ford, F. Labelle, F.I. Popovici,M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, S. Quinlan, Availability ingloballydistributed storage systems, in: Proceedings of the9thUSENIXConferenceonOperatingSystemsDesignand Implementation, USENIX Association, Berkeley, CA, Vancouver, BC, Canada, 2010, pp. 1–7.

