Live migration of Virtual Machines in Cloud Environment using Prediction of CPU Usage

Mr. Vikas Malik Computer Department, G. E. S'S R. H. Sapat College Of Engineering, Management Studies And Research, Nashik, India Prof. C. R. Barde Computer Department, G. E. S'S R. H. Sapat College Of Engineering, Management Studies And Research,Nashik,India

ABSTRACT

The virtualization gives the power of partitioning the physical host in to multiple virtual machines. We can manage the number of active host and their power consumption by migrating the virtual machines according to their resource requirement and current status on that particular host. Service level agreement is the main thing and essential one for giving the reliable quality of service between customers and the data centers. Live migration of the virtual machines managing the over load and under loaded host which gives the ability of dynamic resource allocation on another host. Dynamic virtual machine consolidation and switching off the idle host allow data centers to minimize the resource and power consumption. The proposed technique will provide the ability of dynamic virtual machine consolidation using adaptive utilization threshold based o CPU usage prediction which can easily manage the high level of SLA and reduces the number of VM migrations in between the host. The validation of the proposed technique on multiple workload traces of the Planet lab servers.

Keywords:

Virtualization, resource, utilization, prediction, live migration, consolidation.

1. INTRODUCTION

The computational world become very big and complex, cloud computing is a popular computing model to provide large data centers for users. Cloud offering the different services [1] which user can consider as per the business requirement and cost limit. IaaS, PaaS and SaaS are the three main services of the cloud computing and every data center offering these services to multiple client. In the SaaS type only installed software hand overs to the client for using it's functionalities user need not to know the real infrastructure of the software environment and it's requirement. User just uses the services of the software and saves his work on line despite of knowing the location of actual server. Paas(Platform as a service) offers the platform on which services and applications can be developed for the business.

All the basic infrastructure things included in the IaaS, which delivers the infrastructure for installing and maintaining the own environment in the specified IaaS. This layer is most useful from the data center perspective which can use the virtualization [2] technique for enhancing the service accessibility. Physical

machines can be divided in to number of virtual machines with the help of virtualization technique for addressing the efficiency of the utilization by sharing the resources of physical server in to multiple virtual machines. The hot and cold live migration of virtual machines is the best feature provided by the hyper visors and their virtualization technique. The live migration is most useful in the load balancing, high performance throughput, fault tolerance and the physical server maintenance [3][4][5]. Live migration is beneficial in the dynamic virtual machine consolidation according to their resource requirement and minimizes the number of active host for energy saving purpose. Migration of virtual machines in live environment can saves the power consumption as it keeping the low number of active host in the data center. Quality of service; managed by the service level agreement (SLA) which describes the response time and throughput of the system. The live migration should maintain the power management, SLA level and guarantee of the performance after the migration of the virtual machine.



Fig. 1. Cloud services

This approach describes the phenomenon of the under and overload detection whenever the host is under loaded all the virtual machines from that host migrated on the another host and that particular host is switched off for the time being. Decreasing the number of active host automatically reduces the power consumption in data center. Moreover, the requested amount CPU remand exceeds the available capacity on that host then this host considered as a over loaded host and some VMs migrate to the another host for maintaining the SLA level.

Live migration and the SLA management addressing the following challenges in dynamic VM consolidation:

1. Determination of the overloaded host.

2. Determination of the under-loaded host.



Fig. 2. System view

- 3. Selection of the virtual machine for migration.
- 4. Placement of the selected virtual machine.

The first two issues considerably influence the decisions of other problems. So the expected CPU requirement of the host for deciding the overloaded and under-loaded host is major and interested thing. Predicting the CPU utilization on the basis of dynamic threshold will help in dynamic VM consolidation and generates the efficient load balanced structure on the every host. In the following chapters proposed system is described in detail. Section I gives introduction. Section III describes the related study i.e. study of existing systems. Section III describes the motivation and problem definition of the system. In section IV the system architecture is presented using SLA violation metric and dynamic threshold calculation methodology. Section V gives the algorithms for this system. Finally, conclusions and acknowledgement are stated and also references are given at the end of the report.

2. RELATED STUDY

There is a huge amount of literature survey on the virtualization and virtual machine consolidation which gives the idea of how virtualization works and how hypervisor plays an important role in the live migration of the virtual machine. The architecture utilizes the power system based model on global and local policies to reallocate the virtual machine in the cloud environment. The data center power management issues and work has been carried out by the Nathuji and Schwan [8]. They have distributed the policies in the global and local level, at the local level it manages the guest OS power management and gives the information to the global manager, the global manager catches the information from current resource allocation and applies its procedure to choose the virtual machine placement.

Kusic et al. [9] have defined the power management problem in the virtualized environment as sequential optimization achieved it on the limited look ahead control (LLC). The main objective and motive is to take the full advantage of the resource, revenue and reducing the power and SLA. Kalman filter has proposed the work on future demand forecasting and evaluate the necessary reallocation but require the modifications in the application which is not possible in the IaaS. The continuous optimization problem of heterogeneous system has been expressed by the Verma et al. [10] as a power aware placement of the applications; this optimization was useful in reducing the power and improvement in the performance. In this performance of the application can be degraded due to the workload inconsistency. Recently, Verma et al. [10] has proposed this approach in to static, semi static, and dynamic consolidation. The multi-tier web application consolidation problem using live migration have examined by the Jung et al. [13] [14].

In static CPU threshold the decision is made on the threshold set in the policy if the CPU utilization value goes on the 85% of the total capacity then host is declared as a over loaded host but it do not adapt the workload changes so these are not suitable in the dynamic workload. Therefore, decision making on the dynamic workload based on the analysis of historical data is important.

An approach of dynamic VM consolidation established on the stability approximation- this is an possibility that the reallocation of the virtual machine will stay active for the provided time frame in the future; has been suggested by the Kumar et al [11]. The proposed adaptive migration threshold for dynamic VM consolidation at run time utilization of the resources and migration of the virtual machines switches off the idle nodes and reducing the power consumption. Live migration process varies according to the type of hypervisor and its supporting environment. We have conduct the review on architecture of Leading Hypervisors and Their Live Migration Techniques [12]; which explains the process of virtual machine migration and the architecture of hyper visor.

3. MOTIVATION

Considering a data center scenario which having M hosts with heterogeneous configuration, users are submitting the requests to N VMs which are allocated on these hosts. All virtual machines experiencing the dynamic workloads so the CPU usage and utilization varies arbitrarily. The power consumption is depend on the CPU utilization [3][6][7]of the server and this CPU performance is defined in the format of Millions instruction per second (MIPS). The large scale data center need to maintain the SLA level and Power consumption by the servers and this can be handled by the live migration of the virtual machines but currently all this process going on the static threshold which cannot handle the multiple workload and violating the SLA level. The live migration process includes the Over load detection, under load detection, VM selection and VM placement strategy [3]. We propose the CPU usage prediction and handle the live migration on the dynamic threshold. So it will forecast the over load host and migrates the some VM to balance the load in same way some underutilized host can be switched off by migrating the all VMs for reducing the power consumption. Cloud providers need to concentrate on the energy efficiency and SLA levels; that's why the dynamic VM consolidation is more important in the data center. As the static threshold migration policy causes unnecessary overhead on the physical machines and resulting in to the SLA violation. To ensure the user's application performance during the virtual migration process and maintaining the SLA level we propose a resource scheduling strategy based upon the adaptive migration threshold of the virtual machine.

3.1 Problem Definition

A system design which handles the auto live migration of virtual machines using dynamic migration threshold on the basis of CPU utilization which manages the high level of SLA, reduces the number of migrations and decreasing the power consumption in data center. This addresses the following objectives: Input: CPU utilization data of the Planet lab servers. Output: Percentage of SLA violation, Number of VM Migrations and Power consumption.

4. SYSTEM ARCHITECTURE

This system targets the IaaS environment of the data center which contains the N heterogeneous host and every node is characterized by the RAM, bandwidth, disk storage and the CPU performance (MIPS) and this environment not having the knowledge of the workload and the time for virtual machines are provisioned in the data center. Multiple users are submitting the request for the provisioning of the VMs. The virtual machines are owned by the independent users so we cant get the information of the type of applications installed by the other user.



Fig. 3. System Architecture

This environment having the shared storage connected to all the physical nodes without shared storage live migration of virtual machines can not be possible. The main advantage of using the shared storage while forming the cloud cluster all the virtual disk of the virtual machines are placed on the shared storage and all Nodes are having connectivity to this storage whenever the virtual machine needs to be migrated the disk will be remain on the shared storage and only memory pages will be transferred from one node to another this will improves the performance and never gives the load on network and manages the application performance. For virtualization management we need to install the hypervisor on each node which is having the VMM(virtual machine monitor) and responsible for monitoring the local resources of the host server. The monitoring system fetches the data from this module. The system having multiple VMs created on the physical node and local manger taking the information from VMM. The local mangers are responsible for monitoring the CPU utilization and taking the decision for VM migration. The global manager works like a VM scheduler which are connected to all VMM and local manager of each node for maintaining the over utilizations and issues the commands for the VM placement.

4.1 SLA Violation

The QOS is the thing which should be managed in the cloud computing environment by the cloud service provider and this requirement is formulated in the format of SLA which is characterized in the terms of maximum response time and minimal throughput. As these characteristics can be vary application to application. We have defined this generic terms in our experiment by fraction of the requested MIPS by all VMs Urj(t) and allocated MIPS in actual Uaj(t) relatively to the total requested MIPS during the life time of the VMs. Number of VMs is M.

$$SLA = \frac{\sum_{j=1}^{M} \int_{t} U_{rj}(t) - U_{aj}(t)dt}{\sum_{j=1}^{M} \int_{t} U_{rj}(t)dt}$$
(1)

This represents the percentage of CPU has not been allocated when demanded by the applications.

4.2 Power Consumption

The CPU utilization always changes as workload varies time to time, the workload variability (t), we can calculate the total energy consumption by the server with following model.

$$E = \int_{t} P(\mu(t))dt \tag{2}$$

4.3 Dynamic Threshold

Previous work on static threshold as mentioned earlier is not suitable for the dynamic and random workload which is unpredictable. We are looking for the system which is suitable for all types of workload and should adjust his behaviour according to the workload. Therefore we propose the novel technique which adjusts the utilization threshold based on the analysis of past (historic) data of the virtual machine.

The random variable (uj) describes the CPU utilization created by the each VM with a particular distribution which occurs over some period of time. The CPU utilization of the host represented by a random variable (Ui) which is a sum of utilization of m VMs of the particular host. We are assuming that the distributions created by the virtual machine is differ VM to VM also the distribution of the host is approximately normal and modelled by the t-distributions. Predicting the future resource requirement of the host is quite impossible as every virtual machine application resource requirement n different and due to the virtualization layer of hyper visor it quite complicated to predict it but we can try to predict it from the past CPU usage in recent time in terms of sample mean(Ui) and standard deviation (SUi)

$$\bar{U}_i = \sum_{j=1}^m \bar{u}_j, SU_i = \sqrt{\sum_{j=1}^m S_{uj}^2}$$
(3)

We will collect the data of each VM separately and then we will do the summation that is VM is migrated with its data and that will be same after the migration. With the above information and inverse probability function(tinvn(P)) finding the CPU utilization interval is possible which will reach with lower probability and we can set the upper threshold (Tui), lower threshold (Pul)and upper limit (Puu) with the following formula where n is number of data points and n-1 is the degree of freedom for host I preserving the spare CPU capacity.

$$T_{ui} = 1 - \left((t_{inv_n - 1}(P_{uu}) \cdot s_{U_i} + \bar{U_i}) - (t_{inv_n - 1}(P_{ul}) \cdot s_{U_i} + \bar{U_i}) \right)$$
(4)

Whenevr the host have low CPU utilization a limit (Ul) for capping the lower utilization threshold is introduced. For efficient we can set it to 30%. The lower threshold (Tl)can be calculated as:

$$\bar{U} = \frac{1}{N} \sum_{i=1}^{N} \bar{U}_i, s_U = \frac{1}{N} \sqrt{\sum_{i=1}^{N} (\bar{U}_i - \bar{U})^2}$$
(5)

5. ALGORITHM

5.1 The reallocation algorithm using the dynamic thresholds

Following dynamic relocation algorithm using the dynamic threshold uses the minimum migration policy for VM selection. The complexity of an algorithm is the summation of number of non over utilized host plus the product of over loaded host and their respective allocated VMs.

Algorithm 1. Dynamic Thresholds (DT)

Input: hostList, vmList **Output:** migrationList vmList.sortDecreasingUtilization() foreach h in hostList do $hUtil \leftarrow h.util()$ bestFitUtil \leftarrow MAX while hUtil > h.upThresh() do foreach vm in vmList do if vm.util() > hUtil - h.upThresh() then $t \leftarrow vm.util() hUtil + h.upThresh()$ if t < bestFitUtil then bestFitUtil ← t $bestFitVm \leftarrow vm$ else if bestFitUtil = MAX then $bestFitVm \leftarrow vm$ break $hUtil \leftarrow hUtil - bestFitVm.util()$ migrationList.add(bestFitVm) vmList.remove(vm) if hUtil < lowThresh() then migrationList.add(h.getVmList()) vmList.remove(h.getVmList()) return migrationList

5.2 VM Placement

If the host is overloaded this algorithm selects the VM for migration using VM selection policy. Once the migration list is built the VM placement algorithm invokes the new host information for VM placement. Also it works for under loaded host and migrates the VM form these host.

Algorithm 2. Modified Best Fit Decreasing (MBFD) Input: hostList, vmList Output: allocation of VMs vmList.sortDecreasingUtilization() foreach vm in vmList do minPower \leftarrow MAX $allocatedHost \leftarrow NULL$ foreach host in hostList do if host has enough resource for vm then power \leftarrow estimatePower(host, vm) if power < minPower then allocatedHost \leftarrow host $minPower \leftarrow power$ if allocatedHost 6 = NULL then allocate vm to allocatedHost return allocation

Table 1. Random Workload

SR. No	Algorithm	No. of	%	Energy(
		Migration	s of SLA	Kwh)		
			violations			
1	DT	2872	10.89	35.37		
2	MAD	5265	10.91	45.61		
3	THR	4839	12.81	41.81		
4	IQR	5502	10.44	47.85		

SR. No	Algorithm	No. of	%	Energy(
		Migration	s of SLA	Kwh)
			violations	
1	DT	26292	9.6	163.15
2	MAD	26292	10.18	184.88
3	THR	26634	10.14	191.73
4	IQR	27476	10.8	188.86

6. EXPERIMENTAL SET UP AND RESULTS

As the proposed system is a cloud computing environment and this should be evaluated in the large scale data center architecture environment. It is really difficult to conduct the experiment on real cloud set up as it requires huge amount of resources and cost. Whenever we are referring the live migration of virtual machines it requires the Nodes with huge resources and shared storage attach to all nodes. If we will choose the proprietary hyper visors then we need to purchase the license of their required software's and few open source hyper visors varies in the supporting guest OS. Whenever we need to reproduce the experiment with same condition to compare the multiple algorithms it is quite difficult to have the same state of the environment. We have chosen the CloudSim framework for testing the algorithms in which we can set up the data center environment. This modern framework is widely used in the cloud computing environment for testing the multiple algorithms due to the support for large scalability. In this framework we can scale the resources as well as we can apply the dynamic workloads. CloudSim can be easily integrated with the Eclipse which requires JRE and JDK 1.7 above version.

6.1 Workload data and result

6.1.1 Random workload. The users submit requests for provisioning of 50 heterogeneous VMs that fill the full capacity of the data centre. The random workload provided to every VM, which is modelled to generate the utilization of CPU according to a uniformly distributed random variable. Every application has a specific length that determines the number of instructions with MI. Table 1 illustrates the SLA violation levels caused by different methods in the random workload. The obtained results can be explained by the fact that dynamic threshold can predict the load in a host before it will overloaded and a SLA violation happens.

6.1.2 *Real workload traces.* It is important to have the real workload traces for testing the outcome of the algorithms. For this system we are using the data provided of the CoMon monitoring project. This is the real monitoring project which monitors the PlanetLab servers. We have taken the CPU utilization data of more than thousand servers which has placed globally. The data centre consisting of 800 nodes which includes the HP ProLiant ML110 G4(Intel Xeon 3040, 2 cores x 1860 MHz, 4 GB) servers and HP ProLiant ML110 G5(Intel Xeon 3075, 2 cores x 2660 MHz, 4 GB) servers.

We are considering some other algorithm for comparing the performance with our algorithm.

THR: This includes the static threshold in which over load detection has been set to 80%.

MAD and IQR: These are adapting the utilization threshold using the Median absolute deviation and Inter quartile range. While comparing the dynamic adaptive threshold with other algorithms it gives the better results. The results taken on 800 hosts on which 1052 VMs were provisioned in real workload traces and 50 host, 50 VMs were provisioned in the random workload. The minimum migration time policy was applied for the VM migration in CloudSim tool-kit.

7. CONCLUSION

With the help of above algorithms we can maximize the performance of the cloud servers and we can minimize the number of migrations of VMs from one PM to other PM. The proposed system supports the heuristics of static and dynamic threshold based migration. This system basically focuses on the dynamic VM consolidation and support for green computing by reducing the total number of physical host requirement. To maximize their Return On Investment (ROI), Cloud providers need to apply multiple strategies, like dynamic VM consolidation with switching idle servers to power-saving modes. This system has proposed and investigated a suite of novel techniques for implementing distributed dynamic VM consolidation in IaaS Clouds under workload-independent QoS constraints. The proposed approach improves the utilization of data center resources and reduces energy consumption, while satisfying the defined QoS requirements.

Acknowledgment

I want to thank all people who help me in different way. Especially I am thankful to my guide Prof. C. R. Barde for his continuous support and guidance in my work. Also, I would like thank our H.O.D. of Computer Engineering Prof. N. V. Alone and P.G. coordinator Prof. A. S. Vaidya for motivating me.

8. REFERENCES

- BP. Rimal, E. Choi, I. Lumb, A Taxonomy and, Survey of Cloud Computing Systems, Proceedings of the Fifth International Joint Conference on INC, IMS and IDC, pp.4451, 2009.
- [2] 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 nineteenth ACM symposium on Operating Systems Principles (SOSP03), pp.164-177, 2003.
- [3] A. Beloglazov and R. Buyya, Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers, Concurrency and Computation: Practice and Experience (CCPE), Vol.24, pp.1397-1420, 2012.
- [4] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, VMs live migration of Virtual Machine Based on Full System Trace and Replay, Proceedings of the 18th International Symposium on High Performance Distributed Computing (HPDC'09), pp.101-110, 2009.

- [5] A. B. Nagarajan, F. Mueller, Ch. Engelmann and S. L. Scott, Proactive Fault Tolerance for HPC with Xen Virtualization, Proceedings of the 21st ACM International Conference on Supercomputing (ICS07), pp.23-32, 2007.
- [6] X. Fan, WD. Weber, LA. Barroso, Power provisioning for a warehouse-sized computer, Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA 2007), pp.13 23, 2007.
- [7] D. Kusic, JO. Kephart, JE. Hanson, N. Kandasamy, G. Jiang, Powerand performance management of virtualized computing environments via lookahead control, In Proceedings of the International Conference on Autonomic Computing (ICAC), pp.3-12, 2008.
- [8] R. Nathuji and K. Schwan. Virtualpower: Coordinated power management in virtualized enterprise systems. ACM SIGOPS Operating Systems Review, 41(6):265278, 2007.
- [9] D. Kusic et al. Power and performance management of virtualized computing environments via lookahead control. Cluster Computing, 12(1):115, 2009.
- [10] A. Verma et al. pMapper: power and migration cost aware application placement in virtualized systems. In Proc. of the 9th ACM/IFIP/USENIX Intl. Conf. on Middleware, pages 243264, 2008.
- [11] Kumar S, Talwar V, Kumar V, Ranganathan P, Schwan K. vManage: loosely coupled platform and virtualization management in data centers. Proceedings of the 6th international conference on Autonomic computing (ICAC 2009),Barcelona,Spain, 2009;127136.
- [12] V. Malik, C. R. Barde, "Survey on Architecture of Leading Hypervisors and Their Live Migration Techniques", International Journal of Computer Science and Mobile Computing,IJCSMC, Vol. 3, Issue. 11, November 2014.
- [13] Jung G, Joshi KR, Hiltunen MA, Schlichting RD, Pu C. A cost-sensitive adaptation engine for server consolidation of multitier applications. Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware (Middleware 2009), Urbana Champaign, IL, USA, 2009;120.
- [14] M. Wang, X. Meng, L. Zhang, Consolidating Virtual Machines with Dynamic Bandwidth Demand in Data Centers, Proceedings of IEEE INFOCOM 2011 MINI-CONFERENCE, pp. 71-75, 2011.
- [15] VMware Inc. VMware distributed power management concepts and use, 2010.
- [16] Zhen Xiao, Senior Member, IEEE, Weijia Song, and QiChen, Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment, IEEE Transactions on Parallel and Distributed Systems, Vol.24, No. 6, June 2013.
- [17] Fahimeh Farahnakian, Pasi Liljeberg, and Juha Plosila LiRCUP: Linear Regression based CPU Usage Prediction Algorithm for Live Migration of Virtual Machines in Data Centers 39th Euromicro Conference Series on Software Engineering and Advanced Applications 2013.
- [18] Anton Beloglazov and Rajkumar Buyya, Energy Effcient Resource Management in Virtualized Cloud Data Centers, 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010.
- [19] K.S. Park, V.S. Pai, CoMon: a mostly-scalable monitoring system for PlanetLab, ACM SIGOPS Operating Systems Review, pp.65-47,2006.