Optimizing the Virtual Machine Migrations in Cloud Computing Systems by using Future Prediction Algorithm

Vishnu S Sekhar
PG Scholar
Dept. of Computer Science
Mangalam College of Engineering
Ettumanoor,Kottayam,Kerala ,India

Neena Joseph
Assistant Professor
Dept. of Computer Science
Mangalam College of Engineering
Ettumanoor,Kottayam,Kerala ,India

***Abstract*— Virtualization is the root concept regarding the cloud based scenarios. With the widespread researches in the field of cloud computing, the need for virtual machine migration arises so as to deal with conditions of resource scarcity(hot spots) and excess spare capacity(cold spots).Our main focus is on the optimization of migration techniques so as to reduce the number of future migrations by making an optimal job or process allocation based on the size of the jobs coming to the scheduler and the processing capabilities of the underlying physical machine making use of a future prediction algorithm.**

Keywords:Virtual Machine(VM),Physical Machine(PM),VM mi-gration

I. INTRODUCTION

The concept of cloud computing systems [2] has undergone lots of advancements as well as researches in the past few years. Now also researches are going on regarding the various topics like dynamic resource management, cloud security etc..The cloud actually provides a pool of virtualized resources. The end users or clients using the services get only a virtualized view of the resources in the form of virtual machines. The physical resources are allocated to the virtual machine with the help of hypervisors (eg:XEN hypervisors)[1].That is the virtual machines are allocated dynamically to certain physical machines(PM).There can be multiple virtual machines(VM) working over a single PM. There are situations when the PMs may not be able to meet the requirements of virtual machines allocated to it. That is a condition in which the scarcity of resources occur(hotspots) or underutilization of resources(cold spots). In such situations, there arises the need for VM migration from one physical machine to another. The VM migration involves the transfer of the state of the virtual machine to another PM. That means the execution status till that time is retained and the rest is executed after reaching the destination PM. Consolidation process can be done to bring the VMs from a PM which is underutilized together to another PM with a medium utilization level and the source PM can be turned to an inactive state. Similarly when a PM is overloaded,VMs currently running on it can be migrated to a lightly loaded PM.The virtual machine migration is a popular concept widely used in cloud computing environments to avoid the situations described above. But the problem does not end there. By the time a VM subject to migration the state at the destination PM may change due to either the allocation of new VM or the retiring of the VMs that were already in execution at that PM.Thus there exists a chance for subsequent migration from the PM to the VM has been already migrated. This may further leads to recurring migrations which causes degradation in the performance of the system as a whole. Its an inevitable condition that might be occurring in the clouds environments. So there arises the need for reducing the number of further migration. In this paper we propose a future load prediction mechanism so as to make the optimal assignments of PMs that reduces the number of future VMmigrations[3].The decision regarding migration is done purely on the basis of searching for the optimal allocation that reduces the chances of further migrations or prevents the system from entering into an inconsistent state that may occur due to recurring migrations.

II. EXISTING SYSTEM

In the existing system when an overloaded condition or hotspots[1] occur, the decision to migrate is taken and the virtual machine is migrated to the physical machine with the least load in the present condition. The migration is done from a PM whose load is extremely high in comparison with all the other PMs present in the resource pool. But it doesn't make any prediction of the overall system load condition that may occur after migration and takes no methods to reduce the number of further migrations required. In the current cloud computing environments,VMresource scheduling only considers the current system condition and ignores the previous state of system which causes the system load imbalance. Number of VM migrations is more when most of the load balancing takes place. The cost of performing migration increases proportionate to the number of further migrations required. This is largely due to granularity of VM resources and the large amount of data transferred in the migration with suspension of VM service. Another factor of concern is the extremely high usage of the network bandwidth as the VM migration[5] involves the transfer of the state of a virtual machine and all related execution components through the network. It leads to the need for a scheduling strategy to enable effective load balancing. The method used in this scheduling strategy should compute its influence on the system in advance, when current VM resources are allocated to every physical node and will adopt for the deployment that will have the least load on the system. Our objective is to avoid overload and support green computing. To avoid overload, the capacity of a physical machines should be sufficient to satisfy the resource needs of all virtual machines running on it. Otherwise, the physical machines will be overloaded and can lead to degraded performance of its virtual machines.To support 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. The migration is done from a physical machine whose load is greater than a particular threshold. Thus we arrive at a decision to implement a new approach which will be stated in the following sections.

III. PROPOSED SYSTEM

The proposed system makes use of a future prediction algorithm which considers the allocation to various physical machines and makes a prediction that the migration to which PM leads to minimum number of future migrations. Whenever a VM utilizing the resources from any of the PMs in the resource pool retires then the average of the load over each of the PMs in the pool is calculated and the PM with a load factor less than that value is recommended to be the destination node for a VM under migration. In addition to this we apply the future prediction algorithm which will be explained in more detail in the coming section. In brief, what the future prediction algorithm does is that it calculates the approximate processing speed of each of the physical machines present in the resource pool. We know that for each job or request provided from the client to the cloud system for processing, a new VM with desired specification is generated. Based on the specification of the job and processing power of the processing nodes(physical machines), we will come to know the time required to process each job of given specifications . When a job needs to be migrated, the algorithm firstly checks the consequences that may occur after migrating it to each of the physical machines other than the one in which it was already residing. Based on that a matrix is formed for each allocations assumed and the number of migrations expected to be occurring in each of them is evaluated. A matrix is maintained for the allocation to each physical machines present in the resource pool. The allocation which gives the least value for the number of future migrations will be selected as the target machine to which the job or VM is migrated .

IV. SYSTEM ARCHITECTURE

The system architecture basically consists of two basic blocks

1. Cloud controller
2. Migration controller

A. Cloud Controller

Controls all the activities in the cloud. The cloud controller consists of the following blocks

1) Job allocator: It allocates the job to the best node in the resource pool. The allocation is done on the basis of Node's (PM) load conditions.

2)Future Predictor: This module makes the prediction of load levels in each of the processing nodes(PMs) in mere future and charts out the various optimal allocations.

3) Hot & Cold Pool Manager:- a thread continuously check the load of each node and fix whether a node is hot or cold ie,overloaded or underloaded.

I. *Migration Manager*

It constitutes three parts

1) Overload detector: It detects the overload conditions in the system.
2) Victim VM identifier: Finds the VM which is to be migrated.
3) Remote PM Finder: Decides the PM to which migration is to be done.

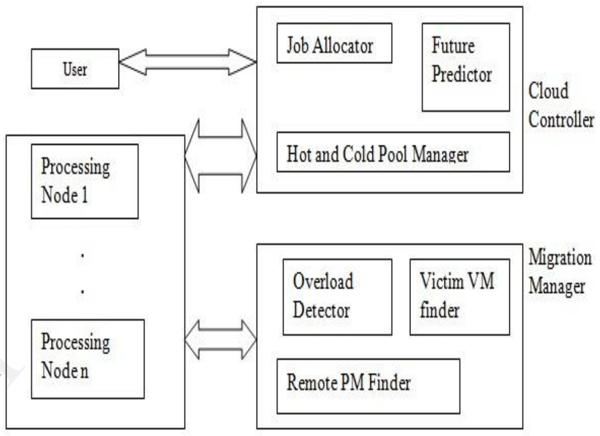


Fig 1. System Architecture

V. IMPLEMENTATION

At the implementation point of view we consider a set of machines and there communication. There can be one or more clients requesting to perform one or more jobs with given number of inputs. Whenever a new job arises a virtual machine with the specification required by the job is created. Next step is done by the scheduler. The virtual machine needs to be allocated to the best processing node which will be physical machine present within the resource pool of the cloud system. The allocation is done to the node having the least load when compared to rest of the nodes in the resource pool . The scheduler monitors the overall load conditions after the completion of each job. Immediately after the completion of each job, the average load in the system is calculated and it is multiplied by factor say α which is fixed based on the number of consequent migrations supported so that the system performance remains optimal. This value gives the threshold. When the load at a particular node is high relative to the threshold then the decision for migration is taken. When the value of α is so high then the number of migrations will probably be low and if the value is less then the number of migrations will be high. The concept is explained in more detail in the following sections. The important modules in the project ie, cloud controller and migration controller described in the earlier sections are residing at the scheduler. The problems associated with virtual machine migrations are solved by the scheduler. The diagrammatic representation of the communication among the modules is shown in fig 2.

The basic problems to be solved in resource management in cloud environment by virtual machine migration includes

•When to migrate?

• Where to migrate?

• Which VM to migrate?

• From which PM to migrate?

The decision to migrate is taken when the overloaded condition occurs at a physical machine. That means the condition when the load or the number of jobs executing at a physical machine exceeds a threshold value which is given by

Jthreshold=javg*α

Javg=Jtotal/Total no. of processing nodes, where Jtotal is the total number of jobs to be executed. The decision to which physical machine the job is to be migrated is based on the future prediction algorithm which is described in the following section.

VI. FUTURE PREDICTION ALGORITHM

This algorithm aims at making a better decision regarding where to migrate in such a way so as to reduce the possibility of further migrations. Firstly the approximate processing speed in number of jobs executed per unit time of each processing node is calculated. Based on that the time needed to process each job can be calculated.

Let’s take an example. Consider a system with 8 jobs say j1,j2...j8 executing at present. Let there be three processing nodes (PMs) say p1,p2 and p3.Let jobs j3 and j5 be executing at p1;j1,j2 and j4 executing at p2 and j6,j7,j8 executing at p3.Let j9 be a job that has come to the scheduler or cloud controller to take migration decision. At such a situation the number of future migrations expected will be calculated with respect to each possible allocation. For example consider the allocation of j9 to p1.When j9 is added to p1 the execution time of each of the jobs i.e., j3 and j5 will be increased by a particular value. That value ’Exec extra’ can be calculated a follows.

Avglen=Size of j9/no. of jobs at the node to which migration is assumed.

Exec extra=avglen/speed of exec node

After adding that value to that jobs in p1,jobs in all the processing nodes say p1,p2 and p3 are arranged in increasing order of their execution time and placed in that order along the column of a matrix and processing node along the rows. Each entry in the matrix constitutes the number of remaining jobs when the particular job given in the corresponding column has finished execution. The matrix for allocation of j9 to p1 is shown below.

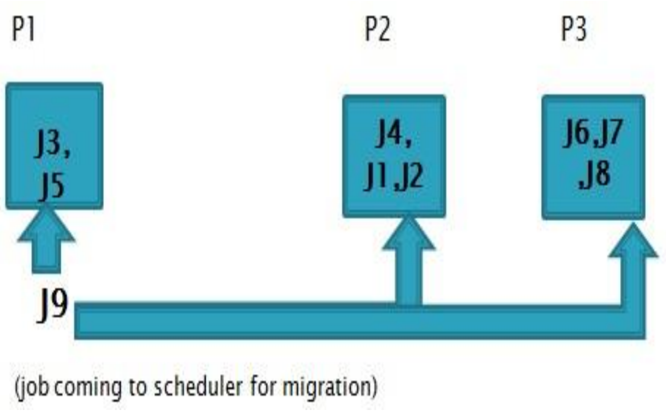
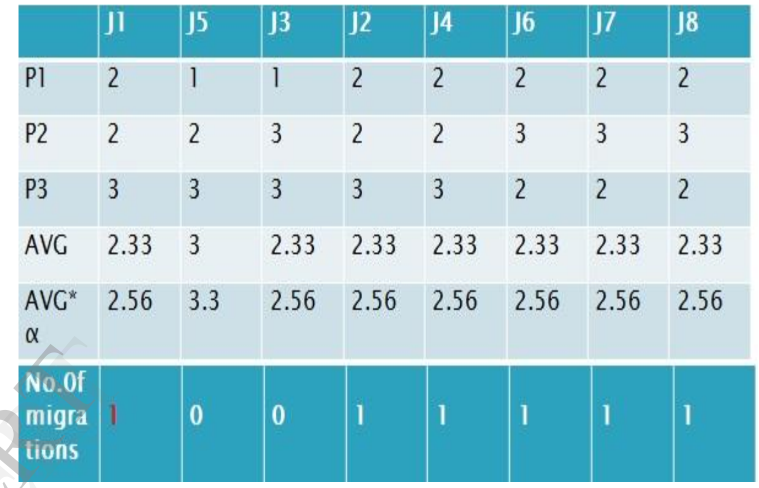


Fig 2. Job allocation to each node



	J1	J5	J3	J2	J4	J6	J7	J8
P1	2	1	1	2	2	2	2	2
P2	2	2	3	2	2	3	3	3
P3	3	3	3	3	3	2	2	2
AVG	2.33	3	2.33	2.33	2.33	2.33	2.33	2.33
AVG* α	2.56	3.3	2.56	2.56	2.56	2.56	2.56	2.56
No.Of migra tions	1	0	0	1	1	1	1	1

The total number of expected migrations is calculated. In the case of migration to p1,,here the total number of expected further migrations is 6.In the similar way the total number of migrations expected is calculated for each nodes. The allocation is done to the processing node with minimum number for total number of migrations.

VII. PERFORMANCE EVALUATION

A. *Effect of Future Prediction Algorithm On the Number of migrations*

In the existing scenarios the migrated job is allocated to a physical machine which is less resource constrained in present situation but never considers the future conditions that may arise in the system as a whole. But the future prediction algorithm as described in the previous chapter tries to minimize the number of migrations occurring in the future after allocation of the job to a particular node.

The efficiency of the future prediction algorithm can be analyzed by determining the number of migrations in the same compared to the earlier technique corresponding to given number of virtual machines. Without prediction, the algorithm tries to consolidate a PM as soon as its load drops below the threshold. With prediction, the algorithm correctly foresees that the load of the PM will increase above the threshold shortly and hence takes no action. This leaves the PM in the “cold spot” state for a while. However, it also reduces placement churns by avoiding unnecessary migrations due to temporary load fluctuation. Consequently, the number of migrations in the system with load prediction is smaller than that without prediction. We can adjust the conservativeness of load prediction by tuning its parameters, but the current configuration largely serves our purpose (i.e., error on the side of caution). The only downside of having more cold spots in the system is that it may increase the number of available physical machines. Here we compare the average number of migrations per VM in each decision with and without load prediction. It shows that each VM experiences fewer migrations with load prediction.

The graphical representation corresponding to each algorithms algm1 which is already existing and algm2 which is the future prediction algorithm is given below. In the x axis is shown the number of virtual machines(in other words the number of jobs) and the number of migrations along the Y-axis. The graph depicts the reduction in the number of future migrations by introducing the future load prediction algorithm. In the earlier migration techniques used the future load conditions in the system as a whole was not considered. As the load conditions are dynamic or varying with time, there are chances of further migrations from the node at which the job has arrived after migration. The future prediction algorithm makes allocation which is optimal so that the number of future migrations can be minimized. The curve in red color indicates the number of migrations after the implementation of future load prediction algorithm and blue curve depicts the number of migrations in a situation without using future load prediction algorithm for a given number of virtual machines in the system.

The graph (fig 4) shows an appreciable reduction in the number of future migrations which is achieved by incorporating a future predictor block which provides the optimal allocation of a VM under migration by making use of the future load prediction algorithm, By implementing future predictor block the overall system performance can be improved and prevents the system from going into an inconsistent state due to repeated or in other words recurring migrations.

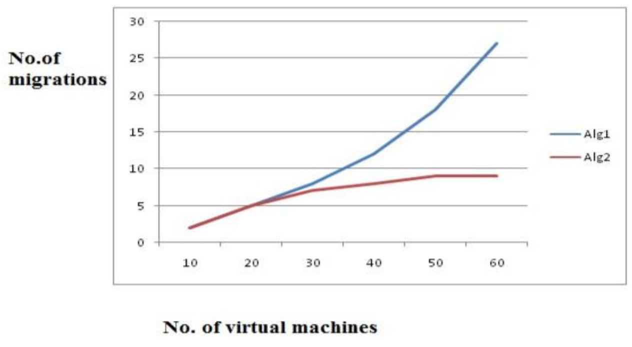


Fig. 4 Performance of future prediction algorithm

VIII. CONCLUSION AND FUTURE SCOPE

The live virtual machine migration in dynamic resource management of virtualized cloud systems making use of future load prediction algorithm has been discussed. Migration enables several resource management goals like consolidation, load balancing, and hot spot mitigation. Researchers have leveraged live virtual machine migration to come up with efficient resource management mechanisms. The components when to migrate, which VM to migrate, and where to migrate and approaches followed by different heuristics to apply migration techniques for goals of consolidation and hot spot mitigation. With the increase in the popularity of cloud computing systems, virtual machine migrations across data centers and diverse resource pools will be greatly beneficial to data center administrators. Live virtual machine migration is an indispensable tool for dynamic resource management in modern day data centers. There is much ground for improvement when it comes to securing the Live VM Migration technique.

ACKNOWLEDGEMENT

With prayers to God for his grace and blessings, for without his unforeseen guidance, this work would have remained only in my dreams. I would like to acknowledge with thanks the contributions given by the management of our college. I express my sincere gratitude to our Principal, Head of the Department of Computer Science for permitting me to do the same and for encouraging me. I would like to extend my regards to my Guide for her valuable ideas, inspiration, cooperation and technical guidance at every stage of the project which helped to keep the work on right track. Last but not the least, I would like to thank my parents, friends, teaching, and non-teaching staff of the college.

REFERENCES

[1] Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo Indian Institute of Technology Bombay "Dynamic Resource Management Using Virtual Machine Migrations",Cloud computing:And Communication Challenges IEEE Communications Magazine • September 2012 pp.34-39.

[2] M. Armbrust et al., "A View of Cloud Computing,"Commun. ACM, vol. 53, no.4, 2010, pp. 50–58.

[3] C. Clark et al., "Live Migration of Virtual Machines,"Proc. 2nd Conf. Symp.Networked Systems Design Implementation, vol. 2, USENIX Association, 2005,pp.273–86.

[4] A. Singh et al., "Server-Storage Virtualization: Integration and Load Balancing in Data centers,"Proc. 2008 ACM/IEEE Conf. Supercomputing, IEEE Press, 2008,pp.1–12

[5] R. Bradford et al., "Live Wide-Area Migration of Virtual Machines Including Local Persistent State," Int'l. Conf.Virtual Execution Environments (VEE), 2007

[6] T. Wood et al., "CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines,"Int'l. Conf. Virtual Execution Environments,2011.

[7] M. Mishra and A. Sahoo, "On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitiga- tion Using a Novel Vector Based Approac*h*," Proc. 4th Int'l. Conf. Cloud Computing, 2011, pp. 275–82.

[8] T. Wood et al., "Memory Buddies: Exploiting Page Sharing for Smart Colocation in Virtualized Data Centers," Proc. ACM SIGPLAN/SIGOPS Int'l. Conf. Virtual Execution Environments, VEE, 2009, pp. 31–40.

[9] M. McNett, D. Gupta, A. Vahdat, and G.M. Voelker, "Usher: An Extensible Framework for Managing Clusters of Virtual Ma chines," Proc. Large Installation System Administration Conf. (LISA '07), Nov. 2007.