A Two Phases Pre-Copy Strategy for the Live Migration to Optimize the Cloud Services

Shalini Ahirwar[1], Prof. Malti Nagle[2]
shalini3355@gmail.com[1]
Department of Computer Science & Engineering
Surabhi College of Engineering & Technology, Bhopal, India

Abstract:- **Cloud computing provides a cost-effective and dynamic scalable service to the user on the rent basic with the help of internet. These services can be used at anytime and anywhere in the world. Virtualization is the prime technology in the cloud which enables efficient resource management and isolation for various big data applications. As cloud provide services to the multiple user and cloud resources are share by the multiple user to increase the resource utilization. In order to increase the resource utilization provider creates the number of virtual machine into a single physical machine which may create a situation where some virtual machines are overloaded. In this case virtual machine is transferred to the other physical machine known as virtual machine migration. Several steps are involved in the migration process. These steps are source physical selection, virtual machine selection, target physical machine selection and selecting method for transferring the virtual machine data. Performance of the virtual machine is mainly depends on the method which is used to transfer the virtual machine data. Pre-copy, post-copy and stop and copy are the methods which are used for sending the data from one physic machine to another physical machine. This paper presents a pre copy approach which minimize the total migration time and down time by minimizing the number of transferred pages. Main concept which is use to reduce the number of transferred pages is to avoid the pages which is highly modified. CloudSim simulator is use to implement our approach.**

Keywords- Virtual machine migrations, pre-copy, post copy, stop and copy, dirty page, down time.

I. INTRODUCTION

Cloud computing is the fastest growing new technology in era of computer and IT industries [1]. It is so popular in due to its quality and cheapest services. It is a utility model and support for an on demand services. Since it support utility model so user pay only for the resources which they actually used [2]. Several definitions are available to define the cloud. According to the NIST standard "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" [3].

Cloud support various type of services and can be implemented in multiple ways. Figure 1 shows the various services and the deployment model of the cloud computing [2, 3].

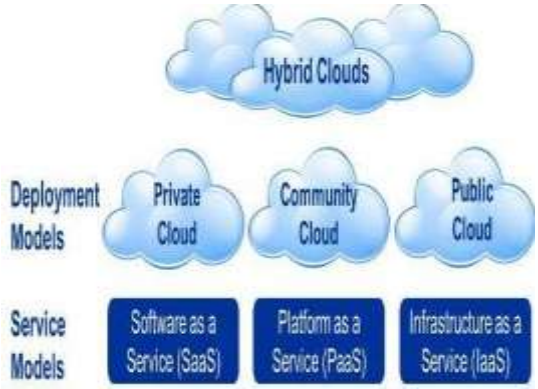


Figure 1: Cloud Services and Deployment Model

Virtualization [4, 5] is the core in the cloud and allows assigning the cloud resources to the multiple user by dividing the physical resources into multiple parts. Hypervisor is the software which makes cloud possible. It creates the virtual machine (VM) and allocated to the user. This VM is created according to the user demand and multiple VM can be given to the single physical machine (PM).

Virtual machine (VM) migration [6, 7, 8] is one of the important features of the virtualization because it enabled the provider to move the VM from one physical machine (PM) to another PM. . It facilitates in various situations like load balancing, server maintenance, server consolidation [7, 8, 9] etc. Several steps are involved in the migration process. These steps are source physical selection, virtual machine selection, target physical machine selection and selecting method for transferring the virtual machine data [9]. Number of methods is available for transferring the virtual machine data like pre-copy, post-copy and stop and copy.

Stop and copy: - It is an offline approach for transferring the VM data from one PM to another PM. According to this approach VM executing on the source PM is stopped and then moved all VM pages from the source PM to the target

PM. When all pages are transferred from source to destination PM then VM is resumed on the target PM.

Post-Copy: - Post copy [10] approaches provide the solution to migrate the VM without the suspension of the services running on the VM. It is a live migration approach for the migration. In these approaches stop the VM on the source PM and then send the minimum data of the VM that required to start the VM on the target PM. Now VM is start on the target PM and transferred the remaining pages to the target PM.

Pre-Copy [11]: **-** This approach also provides the solution to migrate the VM without the suspension of the services running on the VM. It is a live migration approach for the migration. In these approaches in the first iteration we transferred all memory pages to the detonation PM while VM is still running on the source PM. In the second iteration we transferred all the VM data which is modified in the first iteration. This process is continuing until it reached to the maximum number of iteration or minimum size. Figure 4 shows the architecture of the pre copy approach.

II. RELATED WORK

Li Jiaxin et al. [12], proposed a VM scheduling approach for assigning the resources to the multi-tenant in the cloud environment. In this approach first they design the formula for allocating the VM to the multi-tenant. This formula considered the requirement of multi-tenant. After this they proposed approach an algorithm for placing the VM which uses the Multiple Knapsack Problem (LP-MKP). this approach successfully reduced the resource fragmentation in cloud.

Ma et al. [13], proposed an improved version of the pre copy approach. This approach try to removed the limitation of the traditional pre copy approach where the page is send to the target PM where the page is modified in the last iteration. This approach allow to maintain the record of previous history of the page and then take the decision on the basis of previous history. Three types of bit map is used by this approach i.e.to-send, to-skip & to-fix. Page which is frequently modified is stored in send- to- last bitmap. Page is send to target PM if and only if to-send=1 and to-skip=0.

to-send	0	1	1	0
to-skip	0	1	0	1
Send or not	no	no	yes	no

M.R. Desai et.al [14], present an improved version of migration approach proposed by the F. Ma []. Main limitation of the above approach is that if the page is change or modified sequentially then this approach may increase the number of page transferred. To remove this limitation they used the ratio of K/N where K is threshold value of the dirty pages and N is the number of previous history. In this approach if the page is modified more than K times then page is considered as a high modified page and can't transferred to the target VM. Page is send to the target PM when the page is modified less than K times. This approach may reduce the total number of page transferred in the migration process. Main limitation of this approach is to set the accurate value of K.

S. Joshi et al. [15], proposed cuckoo search approach for the virtual machine consolidation. Main objective of this approach is to reduce the energy consumption and resource wastage. Main concept of this approach is the life cycle of Cuckoo bird which keeps their eggs to the other bird nest. In order to reduce the resource wastages vector projection method is used which the PM to the PM where resource utilization of the VM is opposite to the resource used by the PM. This approach reduced the resource wastage.

Hines et al. [16] introduced a post copy approach for transferring the VM data from one PM to another. In the post copy VM's CPU state has been send into the first iteration and then VM is resume on the target PM. Remaining pages or data is send when it is needed. This approach reduces the total migration time but increases the total migration time which will increase the total migration time and reduce the performance of the PM.

H. Jin et al. [17]. presents the solutions for transferring the VM data by using the logs file. In this approach instead of transferring the VM data they send the log files which generated during the execution. Main idea behind this approach is that they think that logs files are smaller in size as compare to the normal VM data pages. So it will reduce the total migration time. This approach may minimized the total down time but it is very difficult to implement practically because to recover the data at the destination they replay the data. So if the source and destination PM has the different configuration then this approach can't be implemented because if two PM has different CPU then have different clock rate, speed etc.

Liu et al. [18] proposed a pre copy based approach which is based on the checkpointing/recovery and trace/replay. To reduce the total migration time and down time they create a Checkpointing/recovery and trace/replay technology and send these checkppoint instead of pages. This approach may minimized the total down time but it is very difficult to implement practically because to recover the data at the destination they replay the data. So if the source and destination PM has the different configuration then this

approach can't be implemented because if two PM has different CPU then have different clock rate, speed etc.

III. Proposed Work

After reading the theory of various method used for transferring the VM data, it is found that pre copy based approach gives better result in term of total migration time and down time as compare to the stop and copy and post copy approach. So in this paper we proposed two phase pre copy approach for transferring the VM data. Most of the existing methods [14] used to transfer the VM data used two type of threshold one for the number of iteration and another on the maximum size in the stop and copy phase. These two threshold are used to apply the stop and copy phase. Since each VM have stop and copy phase during the migration hence it must be stop for a time which is required for the stop and copy phase. During this phase VM services is not available to the user.

To assist the quality of various data transfer method in cloud total migration time and down time are used. As we know that two types of threshold are used by most of the pre copy approaches, the accurate value of these thresholds is difficult to calculate. In this paper we proposed a two phase pre copy approach that optimized the performance of the cloud services by minimizing the down and total migration time.

In pre copy approach all pages are transferred in the first iteration whereas in the next iteration only those pages are transferred which is modified in the last iteration. This process is continuing till the pre define iteration. Hence in the pre copy approach one is transferred several times which increase the total number of page transferred. In order to minimize the total migration time and down time we record the history of previous pages. If the page is modified or change during the transmission in the current iteration then entry for the corresponding pages is 0 otherwise it is 1. In our approach how many pages are recorded as a history pages is decided by the user. Our proposed approach is consisting of two phases. These phases are:

Phase - 1

To store the history of each pages bitmap array is used. Size of the bitmap array is equal to the number of pages in the VM. Each entry of the bitmap array is conation either 0 or 1. If page is change then the corresponding entry for that page is 1 otherwise 0. Since how many number of pages is recorded as a history is decided by the user so width (column) of the bitmap array is depends on this user entry. In this phase we only compare the current page history with the previous page history. If the page is modified in the last iteration and not modified in the current iteration then it can be send and send to the second phase otherwise page is not transferred in to the current iteration. For this purpose we are using two bitmap arrays to store the history of the pages into the current and previous history named to_ current and to_last respectively. Page is only send when to_last=1 and to_current=0 otherwise page is not transferred to the destination physical machine.

Phase - 2

Page is send to the second phase when to_last=1 and to_current=0. In the phase two we check the previous record of the pages. Most of the existing approach [13, 14], use the threshold to decide whether the page is belongs to the higher dirty category or not. If the page is change more than the threshold value then page is belongs to the higher dirty category otherwise it comes into the normal category. Page belongs to the higher dirty category is not send into the current iteration. The performance of the pre copy approach is depends on this threshold value which is very difficult to find.

In our approach we eliminating the use of this threshold value. Instead of this threshold value we decides whether the page belongs to the higher dirty category on the basic of number of modification in the history. To declare the page as a higher dirty category we count the number of once and zeros for the corresponding page in the history bitmap. If number of once is larger than number of zeros the page is declared as a number of once and zeros and not transferred in the current iteration otherwise it is transferred to the destination physical machine.

If m represent the size of the history in each VM and p is the boolean variable, Then number of times when the page is modified is given by A.

$$A = \sum_{i=1}^{m} \mathrm{p} \qquad (1)$$

Hence, number of times when the page is not modified is given by B.

$$B = 1 - P \qquad (2)$$

Page is send to the destination PM only when B > A, otherwise wait and not send page.

Proposed Two Phase Pre Copy Approach

Pages in VM =n,
Maximum number of iteration = m
Number of sequence generated = 20
Number of recorded pages for each VM=p,
to_current=0, to_last=0, to_count[i].

History Of each page for the last four iterations	0	0	0	0	0	0	0	0	1	1	1
	0	0	0	0	1	1	1	1	0	0	0
	0	0	1	1	0	0	1	1	0	0	1
	0	1	1	0	1	1	1	0	0	1	0
	1	1	1	0	0	0	1	0	0	1	0
	0	1	1	0	1	1	1	0	0	1	0
	1	1	0	0	0	0	1	0	0	1	0
	1	1	1	0	1	1	1	1	0	1	1
	0	1	0	1	0	0	0	0	0	1	0
Send or Not	Y	N	Y	N	N	Y	N	Y	N	N	Y

[1] **i=1**
[2] **while (i < 20) do**
[3] **j=1**
[4] **while(j < m) do**
[5] For each page of the VM check whether it is change or not
[6] **for each** pages of VM **do**
[7] **if** (to_last=0 &&to_ current =1|| to_last=1 &&to_ current =1|| to_last=0 &&to_ current =0) **then**
[8] Page is not transferred and wait for the next iteration
[9] **else**
[10] **if** (to_last=1 &&to_ current =0) **then**
[11] Calculate the number of times when the page is modified by using the previous record (history of the page)
[12] Calculate number of times when the page is modified according to the previous equation 1and 2.
[13] B = P-A //Number of times when the page is not modified
[14] **if** A < B **then**
[15] Send pages to the destination virtual machine into the current iteration
[16] **else**
[17] Don't send the page.
[18] **end if**
[19] **end if**
[20] j=j+1
[21] **end while**
[22] Stop the VM at the source PM and transferred all remaining VM pages. Now start the VM on the target PM
[23] **end while**

For example if the 9 previous pages is record as a history for each page then page if to_last = 1 and to _send= 0 then page is transferred to the second phase and if the number of zeros is greater 5 then only page is transferred to the final physical machine .

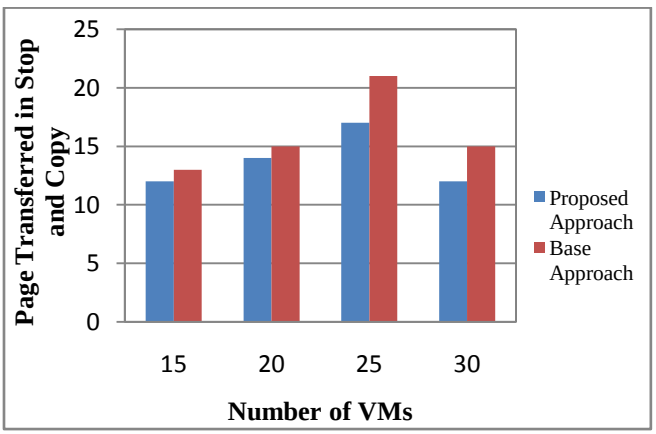
Table 1: Working of Two Phase Pre Copy Approach

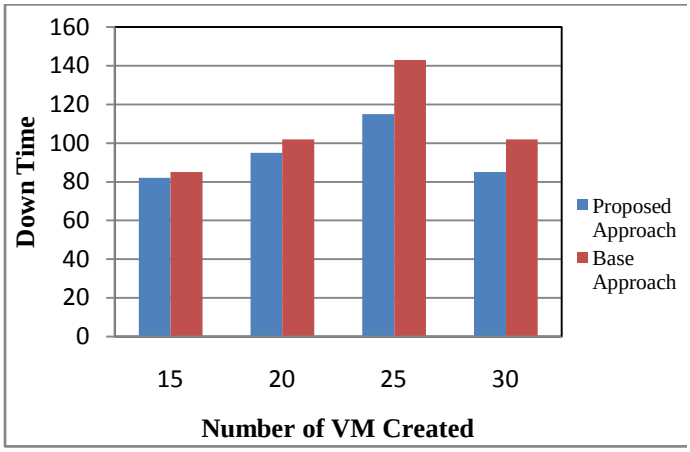
IV. Results Analysis

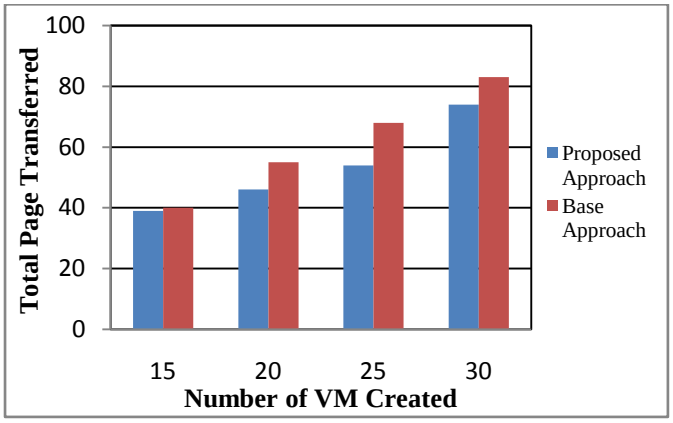
CloudSim [19] is the tool which is used for measuring the performance of the proposed approach. For this purpose it is compare with the already exist approach named " Efficient Virtual Machine Migration in Cloud Computing" [14] and check the total migration time and down time. Both approaches are implemented in cloudSim simulator. Following environment is created during the experiment:

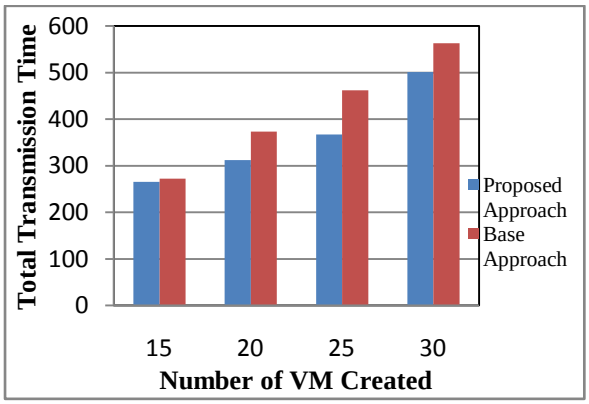
For the proposed two phase pre copy approach
Number of VM created during the experiment = 2
Pages in each VM= 15, 20, 25
Previous Recorded data for each page =8
Iterat_threshold =7

For the already exist efficient virtual machine migration approach
Number of VM created during the experiment = 2
Pages in each VM= 15, 20, 25
Previous Recorded data for each page =8
Iterat_threshold =7
Dirty_Threshold=5









V. CONCLUSION AND FUTURE WORK

Virtualization is the prime technology in the cloud which enables efficient resource management and isolation for various big data applications. As cloud provide services to the multiple user and cloud resources are share by the multiple user to increase the resource utilization. In order to increase the resource utilization provider creates the number of virtual machine into a single physical machine which may create a situation where some virtual machines are overloaded. In this case virtual machine is transferred to the other physical machine known as virtual machine migration. Several steps are involved in the migration process. These steps are source physical selection, virtual machine selection, target physical machine selection and selecting method for transferring the virtual machine data. Performance of the virtual machine is mainly depends on the method which is used to transfer the virtual machine data. Pre-copy, post-copy and stop and copy are the methods which are used for sending the data from one physic machine to another physical machine. This paper present an pre copy approach which minimize the total migration time and down time by minimizing the number of transferred pages.

References

[1] G. Pallis, “Cloud Computing: The New Frontier of Internet Computing”, IEEE Journal of Internet Computing, Vol. 14, No. 5, September/October 2010, pages 70-73.

[2] R. K. Gupta et al., “A Complete Theoretical Review on Virtual Machine Migration in Cloud Environment”, International Journal of Cloud Computing and Services Science (IJ-CLOSER), Vol.3, No.3, June 2014, pp. 172-178.

[3] B. P. Rima, E. Choi, and I. Lumb, “A Taxonomy and Survey of Cloud Computing Systems”, Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, August 2009, pages 44-51.

[4] A. Kivity et al. "kvm: the linux virtual machine monitor,” in Proceeding of the 2007 Ottawa Linux Symposium, 2007, pp. 225–230.

[5] P. Barham, et al. “Xen and the art of virtualization,” in Proceedings of the 19th ACM symposium on Operating Systems Principles (SOSP’03), 2003, pp. 164–177.

[6] D.Perez Botero, “A brief tutorial on live migration of virtual machine from a security perspective”, 2011

[7] C. Clark et al. “Live migration of virtual machine”, proceeding of the 2nd conference on symposium on network system design and implementation, 2007.

[8] T. Wood et al., “Black-Box and Gray-Box strategies for virtual machine migration”, proc. 4th conference on symposium on network system design and implementation, 2007.

[9] R. K. Gupta et al.,” Survey on Virtual Machine Placement Techniques in Cloud Computing Environment”, International Journal on Cloud Computing: Services and Architecture (IJCCSA) ,Vol. 4, No. 4, August 2014, pp. 1-7.

[10] M. R. Hines and K. Gopalan, “Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning,” in Proceedings of the ACM/Usenix International Conference on Virtual Execution Environments (VEE’09), 2009, pp.51-60.

[11] W. Cui, M. Song, “Live Memory Migration with Matrix Bitmap Algorithm”, IEEE, 978-1-4244-6359-6/10,pp 277-281,2010.

[12] Li Jiaxin et al., " Efficient Multi-Tenant Virtual Machine Allocation in Cloud Data Centers", international journal of Tsinghua Science and Technology, vol. 20, no. 1, Feb 2015, pp 81- 89.

[13] F. Ma, F. Liu, Z. Liu,” Live virtual machine migration based on improved pre-copy approach ,” In Proc. Software engineering and Services science, pp. 230-233, 2010.

[14] M.R. Desai et al., "Efficient Virtual Machine Migration in Cloud Computing", proceeding of Fifth International Conference on Communication Systems and Network Technologies (CSNT), , pp. 1015-1019, 2015

[15] S. Joshi et al., " Cuckoo search Approach for Virtual Machine Consolidation in Cloud Data Centre ", proceeding of the IEEE International Conference on Computing, Communication and Automation (ICCCA), 2015, pp. 683-686.

[16] M. R. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning," in Proceedings of the ACM/Usenix International Conference on Virtual Execution Environments (VEE'09), 2009, pp.51-60.

[17] H. Jin, H. Liu, X. Liao, L. Hu, P. Li," Live migration of virtual machine based on full system trace and replay,", 2012.

[18] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live migration of virtual machine based on full system trace and replay," in Proceedings of the 18th International Symposium on High Performance Distributed Computing (HPDC'09), 2009, pp.101–110.

[19] R. Calheiros et al. " CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services", 2011.