Improving Performance of Live Migration using Log Table Based Method without Violating Service Level Agreement

G Hemanth Kumar Yadav, K. Madhavi

Abstract:*The Cloud Service providers have to manage the resources that are available more efficiently to provide services according to Service Level Agreement. Virtual machines use the hardware effectively which reduces the need of physical hardware. Due to some circumstances like system crashes, increase in load on server, the virtual machines need to be migrated to other host machines. In live migration the virtual machines are moved to other host machines while they are running. In this paper, we propose a Log-Table based Migration algorithm for maximizing the availability of Cloud Services to the Cloud Users while migrating the Virtual Machines. During live migration, this algorithm maximizes the availability of high priority cloud services to the users under the constraints of load on network bandwidth and time. Experimental results shows that this algorithm performs well at heavy workload conditions by providing services to the cloud users according to Service Level Agreement.*

Index terms: Cloud Computing, Virtual Machines, Live Migration, Pre-copy, Service Level Agreement.

I. INTRODUCTION:

Cloud Computing provides the clients to access the compute resource, such as a Storage, software application or virtual machine, as a utility [1],[2]. Many commercial cloud providers (Microsoft, Amazon, IBM, and Google) have emerged and each typically provides its own cloud infrastructure, APIs and application description formats to access the cloud resources, as well as support for service level agreements (SLAs). Most of the organizations are using third-party cloud for computer infrastructure and maintenance rather than buying the infrastructure. The challenges for Cloud Service Providers have increased with the increase in number of cloud users. The Cloud Service Providers provides services to the users through data centers. The major challenge is to provide services to the users based on Service Level agreement, when more number of users tries to access the data centers at a time.

The Cloud Service provider needs to manage the Virtual Machines and the hardware resources more efficiently to ensure high quality service to be provided at lower cost. Virtual Machines are used by the Cloud Service Provider to separate the compute environment from the physical infrastructure, which makes multiple applications and Operating Systems (OS) run on the same machine simultaneously. Efficiently managing the virtual machines in a data center has became a big issue, because of increasing number of users choose cloud data centers to hold their applications. This problem can be solved by some extent using virtual machine migration. We can move VM (Virtual Machine) from one server to another server, from one data center to another data center. Usually VM Migration is done during hardware maintenance [3], [4], energy management [5], [6] load balancing [7], [8] and traffic management [9], [10].

VM Management not only has benefits, it also introduces some overhead problems. Two methods are used for Virtual Machine migration, non-live and live migration. In non-live migration, Virtual Machines are either shutdown or suspended to perform migration. During this migration all network connections are rebuilt when the Virtual Machines are restarted at the destination server. This type of migration is used mostly in the maintenance phases. But there will be situations where unplanned migration has to be done due to overloading of server and it is going to crash or power failures etc. In these situations live migration will be used to migrate the Virtual Machines from one Server to another Server. In Live Migration [11], the migration process will be done without any interruption of the Service to the users.

Revised Manuscript Received on December 22, 2018.

G Hemanth Kumar Yadav ,K. Madhavi' Research Scholar, Department of CSE, JNTUA, Ananthapuramu - 515002, Andhra Pradesh, India.hemu204@gmail.com[2]Department of CSE, JNTUA, Ananthapuramu - 515002, Andhra Pradesh,India. kasamadhavi@yahoo.com

In this migration the Virtual Machines are not shutdown or suspended, they will be providing services to the users. But the Cloud Services may be down for some time not more than the downtime specified in the Service Level Agreement. The Live migration of Virtual Machines is within the Data Center then the downtime is less compared to migration of Virtual Machines from one Data Center to another Data Center. The performance of the Service is severely affected when we need to migrate Virtual Machines from one Data Center to another Data center. During migration we can transfer either one Virtual Machine or multiple Virtual Machines. Virtual Machine migration can also be done in either pre-copy or post-copy methods [12], [13].

We need a migration strategy which moves the Virtual Machines form source to destination with less downtime and reduces the problems occur during the migration. The parameters that define the metrics to assess the migration policies are Service Degradation, Total Migration time, Network Traffic generated and total Downtime.

II. RELATED WORK

To monitor the disruption and degradation of the services of cloud providers are detected using anomaly detection methods, which leads to Virtual Machine migration. Detection of Anomaly is based on System metrics of log-based [14]. Mhedheb and Streit [15] developed the method for Virtual Machine scheduling for energy efficiency. Thermal aware scheduler has been developed to inspect the changes in the load on servers and the temperature of the server machines to trigger the migration to avoid any overloading or overheating of the servers.

The difficulty of achieving the SLA for multi-tier web applications has been discussed in [16]. The proposed method tries to achieve profit and reduce the maintenance cost by managing dynamic resource allocation. Stochastic programming has been used for selecting the virtual machines to be loaded in the local data centers near by the host in [17]. The cloud service providers service has been analyzed by using Learning Automata based QoS [18] to check the whether the services are according to the agreement.

Paper [19] provides brief information on concept of cloud computing and various technologies that is being used for resource allocation. Reduction in the cost of deployment of services is primary concern of Resource allocation. Several resource allocation methods have been proposed based on the various important factors such as execution time, utility, VM, dependency on the hardware, application and SLA. In [20], Resource scheduling and costs have been designed by considering the expectations according to both customers and cloud service providers. The work [21] explains various resource allocation policies. Multi-dimensional SLA based resource allocation has been evolved from Service level agreement based resource allocation [22] which proposes algorithm that uses two SLA levels. Response time and allocated memory is calculated for the customers in Gold SLAs. The force between the customer and the cloud service provider is handled by this algorithm to maximize the profit for both.

Service Level Agreement based service provision for the cloud users with satisfaction level have been accounted as major criteria in [23]. This paper explores the willingness of the users to pay for the service and their satisfaction level to pay.

Lou et al. [24] developed tools that can be used to extract log data and process using machine learning techniques to discover the Invariants. The anomalies can be detected by matching the input with the already defined invariants. Any variation can define the anomalies. Xu et al. [25] developed a method to find the system errors and anomalies automatically by mining the console log.

III. LIVE MIGRATION PROCEDURE:

Cloud Services live migration requires to identify which of the Virtual Machines to be migrated when it is required and in which server they have to be migrated. VM Selection algorithm addresses this issue to select which Virtual Machine for migration and where to be migrated. When the Cloud Service Provider detects the problem in the host machine and requires migrating VMs to another Host, the exact time for Host failure is not known. So we need to move all the Virtual Machines to the destination Host server before failure of the Host Machine. During migration of the VMs from source to destination, it consumes a lot of network bandwidth and requires CPU cycles to complete the process. The migration process may consume a lot of time to complete due to dirty pages generated during user modifying the pages[26] as shown in the below figure 3.1.

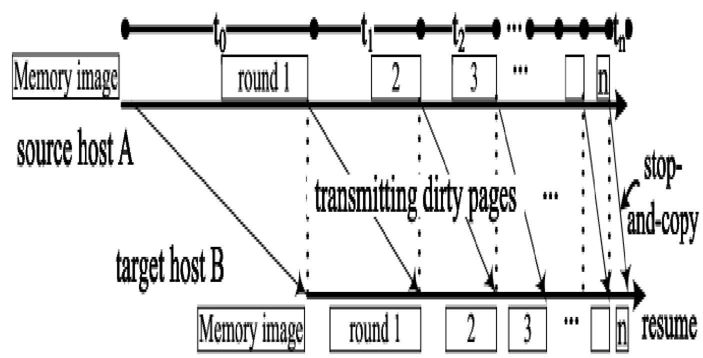

Fig. 3.1Live Migration of Memory pages

We consider Infrastructure-as-a-Service model where the Virtual Machine instances are managed by cloud service providers to offer services to the cloud users. The cloud service providers maintain one or more hypervisors to create and manage Virtual Machines whenever the cloud user requests.

We assume that the hypervisors are compatible for migration Virtual Machines from one hypervisor to another. The cloud user can get services from one or more virtual machines running in the host machine i.e. the user service may have more than one applications, where one application running in one Virtual Machine and other application in another Virtual Machine.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In live migration process to provide services according to Service Level Agreements the Virtual Machines to be migrated to the nearest data centers as this migration process also increases the network traffic between the source and destination nodes. The parameters to calculate the cost of the migration includes CPU cycles, priority, Virtual Machine image size, Dirty Page rate, bandwidth. For a Cloud Service provider, let the set of cloud services can be defined as S= {S_1, S_2, S_3 ….S_n} and a set of Virtual Machines as V= {V_1, V_2, V_3…V_n}. Each Cloud Service can be comprised of one or more virtual machines. Each Cloud Service is given a priority, a non negative number based on the availability of Service parameter which is defined in Service Level Agreements.

The cost of Cloud Service can be defined as sum of the costs of all virtual machines comprised in that service. The cost of each virtual machine is the total memory consumed by it i.e., image size of each virtual machine. The cost of Cloud Service can be defined as follow

$$C(S_i) = \sum_{V_j \in S_i} C(V_j)$$

The cost of live migration of virtual machines is dependent on following parameters: Image size M of virtual machines of a Cloud Service in MB, total dirty page rate R of the Virtual Machines of a Cloud Service in MB/s, bandwidth of the channel used for migration L in MB/s, Downtime Specified in the availability parameter in ms, and amount of data that can be transferred in the channel in each iteration as *B* in MB.

Total Migration time is the summation of time taken to transfer all the memory pages from source to destination and is defined as follow

$$MT = \frac{V_{mem}}{B}$$

Where, V_{mem} total size of memory pages and *B* is the bandwidth of the channel.

Downtime is the time when the service is temporarily unavailable to the cloud user due to the migration process. Downtime *DT* depends on the dirty page rate *R*, Number of Pages *N*, T_n time taken in the last iteration of migration and Bandwidth *B*, Lui et al. [27] defines as follow

$$DT = \frac{R*N*T_n}{B}$$

The time taken T_i for virtual machine migration at each round is calculated as

$$T_i = \frac{V_m*R^i}{L^{i+1}}$$

Where V_m is the amount of virtual memory, R is the dirty page rate, L is the transmission rate of memory during Virtual Memory migration and the Network Traffic generated NT_{mig} during the Virtual Machine migration is defined as:

$$NT_{mig} = \sum_{i=0}^{n} V_m \left(\frac{R}{L}\right)^i$$

The Latency of total live migration can be defined as

$$T_{latency} = \sum_{i=0}^{n} T_i$$

IV. Log Table based Migration:

Our implementation of the algorithm keeps track to information retrieved from the log files regarding memory locations of Virtual Machines. Whenever any modifications are made to the pages while migration process that information is recorded in the log table. The log table can be used to select the pages in the first round of iterations where we try to migrate only the pages which are not frequently updated, so less number of pages to be retransmitted again to the destination server.

The algorithm is given as:

Log Table based Migration Algorithm

$input : V_{mem}, B, R, P[i]$
$for\ i = 1\ to\ n$
$V_i \leftarrow P[i]$
$T_i \leftarrow V_i / B$
$V_{mem} \leftarrow V_{mem} - V_i$
$if\ \ V_{mem} == 0\ then$ /* to end migration */
$V_j \leftarrow R * T_{i-1}$
$T_j \leftarrow V_j / B$
$DT \leftarrow T_j + T_r$
$end\ if$
$end\ for$
$MT = \sum_{i=1}^{n-1} T_i + T_j$
$MV = \sum_{i=1}^{n-1} V_i + V_j$

The process starts by calculating the total amount of memory to be transferred of all services. Then select the memory pages to be transferred in the iteration which are not modified or least modified by the users. The pages can be selected by using the log maintained regarding the updated memory pages. Calculate Down time and Time taken to transfer the memory pages in the iteration.

The process continues till all the pages are transferred to the destination server. In this algorithm the number of duplicate pages to be transferred in the live migration due to dirty page rate can be reduced which increases the efficiency of live migration.

Table 1: Performance parameters used in Live Migration of Virtual Machines

V_{mem}	*Total memory size of all virtual*
V_i	*Total number of pages transfer*
T_i	*Total Time to transfere pages in*
$P[i]$	*priority array to select virtual m*
B	*Transmission rate of memory*
R	*Dirty Page Rate*
n	*number of iterations*
V_j	*Total number of pages transfer*
T_j	*Total Time to transfere pages in*
T_r	*Time taken to resume Virtual ma*
DT	*Total Down Time*
MT	*Total Migration Time*
MV	*Total Volume of memory migra*

V. IMPLEMENTATION:

This section describes the environmental setup for simulation to perform experiments on live migration of virtual machines across various loads. It shows the performance evaluation of total number of Virtual Machine Memory pages transferred, total downtime and total migration time. The effectiveness of Log Table based Migration algorithm is compared with pre-copy method under various loads. The Simulation setup for Virtual Environment has Xen hypervisor installed in two laptops with 8 GB RAM and Ubuntu 14.04 OS. VMs are installed with Ubuntu 14.04 as guest OS. The Virtual Machine memory is migrated and can be accessed with NFS protocol as an image file.

The performance of the algorithm is tested under idle condition where no applications are running and workload conditions where other applications are running during the migration. The Xen hypervisor generates a log file with the number of pages transferred, time taken for transferring pages, total iterations. The log file can be used to get the information to analyze the results. The performance of Log Table Based Migration is compared with pre-copy method and also with prediction based reuse-distance algorithm. The parameters considered for comparison are Total migration time, Number of Pages transferred and Total Downtime.

Total number of pages transferred: Table 2 and Figure 5.1 shows the total number of pages migrated in pre-copy, prediction based reuse-distance method and Log Table Based Migration method under idle and workload conditions. It shows that Log Table Based Migration method performs almost equally at idle conditions but it outperforms other two methods in workload condition. At idle condition the number of pages updated is less as the user not modifying the pages which leads to less number of dirty pages and more pages are sent in all the methods. In this condition all the methods perform similarly. At heavy workload condition many pages are modified and the pages are to be retransmitted. In this case the number of pages retransmitted in Log Table Based Migration algorithm is reduced compared to other two methods because of selecting the pages which are not frequently modified. As a result, Log Table Based Migration algorithm performs well and reduces the total number of pages to be transferred to the destination server.

Table 2: Total number of pages transferred

Condition	Pre-copy	PBRS	LTBM
Idle	284,163	277,106	279,487
Workload	1,022,344	576,193	301,590

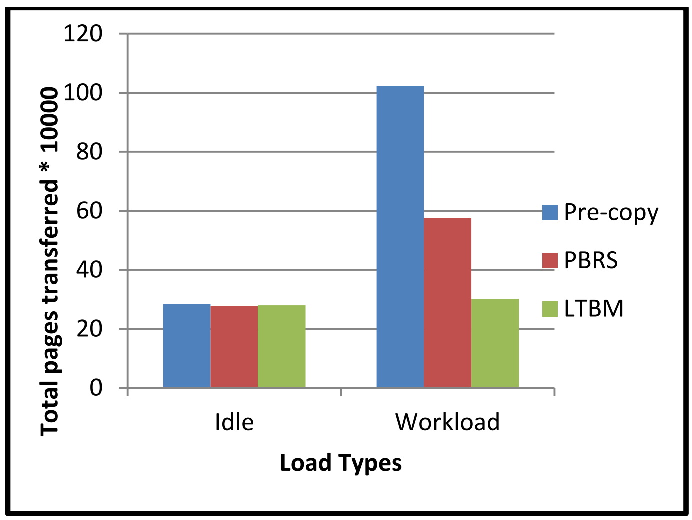

Fig. 5.1Total pages transferred at different loads

Total Live Migration Time: The total time taken for all iterations to complete the total pages of the Virtual machines from source server to destination server. The Log Table Based Migration algorithm performs well by reducing the total number of Virtual Machine Memory pages to be retransmitted by selecting pages which are not frequently updated. The last iterations will transfer the pages which are frequently updated. Table 3 and Figure 5.2 show that the time taken for migrating pages is reduced, total migration time is also reduced compared to pre-copy and prediction based reuse-distance method.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 3: Total Time taken for Live Migration

Condition	Pre-copy	PBRS	LTBM
Idle	103,435	90,125	82,476
Workload	305,277	178,784	88,223

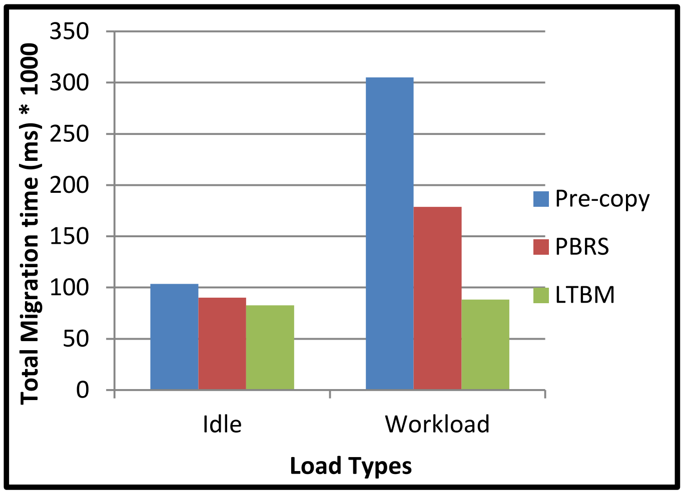

Fig: 5.2 Total Migration Time at different load types

Total Downtime: The time taken after migrating virtual machines from source to destination, the source virtual machine is stopped and will resume at the destination server after the last iteration is the downtime for the service to the user. This parameter is important as it directly affects the service provider for his Service Level Agreement of Availability. The Downtime should be less to avoid breaching the Service Level Agreements. Table 4 and Fig. 5.3 show that the total downtime for the three methods. The Downtime in The Log Table Based Migration method is similar to other two methods in idle condition. The Log Table Based Migration method gives less downtime than the pre-copy and prediction based reuse-distance method for the higher workloads because the number of retransmission of pages is reduced.

Table 4: Total Downtime in Live Migration

Condition	Pre-copy	PBRS	LTBM
Idle	233	231	231
Workload	62,820	60,986	60,213

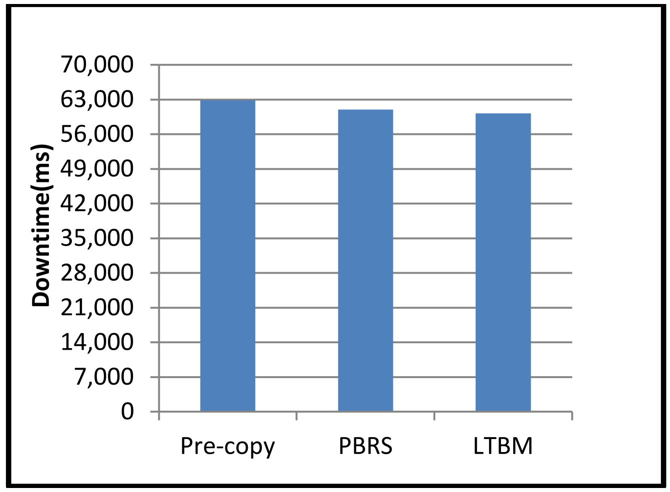

Fig: 5.3 Total Down time at Heavy Workload

VI. CONCLUSION:

In this paper, we presented a Log Table based algorithm for maximizing the Cloud Services availability for the Cloud Users. The Live Migration of Virtual machines normally leads to unavailable of services to the users for some time which causes service disruption. This algorithm preserves the availability of Cloud Services during the live migration of virtual machines and provides services according to Service Level Agreement. Live migration process consumes large amount of network bandwidth which affects the overall efficiency of Cloud Services. To improve the efficiency and reduce the overheads created to network bandwidth, we proposed Log Table based algorithm. The performance of the algorithm has been assessed based on the parameters like Total number of page transfers, Total Migration time and Total Downtime. Pre-copy method is mostly used method in live migration and its drawback is transferring more number of pages in the live migration process. The Log Table based algorithm will reduce the transfer of same pages multiple times by selecting the pages which are not frequently updated.

The Experimental results show that Log Table based algorithm improves the performance of live migration by reducing the total number of memory pages transferred to the destination server, reduces total migration time required for live migration and reduces total downtime comparing to pre-copy method and prediction based reuse-distance. To enhance the existing method we can add compression in the process which reduces the total memory to be transmitted to the destination, but the compression increases overheads. This can be studied more in detail as future work.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

REFERENCES:

1. G. Boss, P. Malladi, D. Quan, L. Legregni, and H. Hall, "Cloud computing," IBM white paper, vol. 321, pp. 224–231, 2007.
2. A. Weiss, "Computing in the clouds," Computing, vol. 16, 2007.
3. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation (NSDI'05), Vol. 2, pp. 273–286. USENIX Association, Berkeley, CA, USA, 2005.
4. Huang, T., Zhu, Y., Wu, Y., Bressan, S., Dobbie, G.: Anomaly detection and identification scheme forVMlive migration in cloud infrastructure. Futur. Gener. Comput. Syst. **56**, 736–745, 2016.
5. Li, X., He, Q., Chen, J., Ye, K., Yin, T.: Informed live migration strategies of virtual machines for cluster load balancing. IEEE Trans. Softw. Eng. **PP**(99), 111–122, 2011.
6. Lu, P., Barbalace, A., Palmieri, R., Ravindran, B.: Adaptive live migration to improve load balancing in virtual machine environment. In: Euro-Par 2013: Parallel Processing Workshops 2013, Aachen, Germany, pp. 116–125, August 26–27, 2013.
7. Wood, T., Shenoy, P.J., Venkataramani, A., Yousif, M. S.: Blackbox and gray-box strategies for virtual machine migration. NSDI **7**, 17–17, 2007.
8. Zhao, Y., Huang, W.: Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud. In: International Conference on Networked Computing and Advanced Information Management, NCM, pp. 170–175, 2009.
9. Nagafuchi, Y., Teramoto, Y., Hu, B., Kishi, T., Koyama, T.,Kitazume,H.: Routing optimization for live VM migration between datacenters. In: 2015 10thAsia-Pacific Symposium on Information and Telecommunication Technologies (APSITT), pp. 1–3. IEEE , 2015.
10. Shetty, S., Yuchi, X., Song, M.: Towards a network-aware VM migration: Evaluating the cost of vm migration in cloud data centers. In: Moving Target Defense for Distributed Systems. Springer,Berlin, 2016.
11.] A. Shribman and B. Hudzia, "Pre-copy and post-copy vm live migration for memory intensive applications," in European Conference on Parallel Processing. Springer, pp. 539–547, 2012.
12. F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, "iaware: Making live migration of virtual machines interference-aware in the cloud," IEEE Transactions on Computers, vol. 63, no. 12, pp. 3012–3025, 2014.
13. U. Deshpande, D. Chan, T.-Y. Guh, J. Edouard, K. Gopalan, and N. Bila, "Agile live migration of virtual machines," in Parallel and Distributed Processing Symposium, 2016 IEEE International. IEEE, pp. 1061–1070, 2016.

Authors Profile:

Dr. K. Madhavi, working as Assistant professor in the Department of computer science and Engineering, JNTUA college of Engineering, Anantapur, Andhra Pradesh, India. She received **B.Tech** in Electronics and Communication Engineering from JNT University, Anantapur, India, **and M. Tech** in Computer Science from JNT University, Anantapur, India, and **Ph.D** from JNT University, Anantapur, India in August, 2012. Her research interests include computer networks, Cloud Computing, Mobile communications and wireless systems. She has published several papers in national and international refereed journals and conferences.

G. Hemanth Kumar Yadav is Pursuing Ph. D at JNTU Ananthapuramu and working as Assistant Professor in department of Computer Science and Engineering at Srinivasa Ramanujan Institute of Technology, Ananthapuramu, Andhra Pradesh, India. He received **B.Tech.** degree in Information Technology from the JNT University, Hyderabad, India in 2006 and **M.Tech** in Computer Science and Engineering from JNT University, Anantapur in 2010. His research interest includes Cloud Computing, Wireless Communication, Mobile Ad hoc networks, and Sensor Networks.