ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

# A Novel Study on Pre-copy Method for Classification of Live Migration of Virtual Machines in Cloud Computing

Minal Patel
Assistant Professor, A.D. Patel Institute of Technology, Karamsad, Anand, Gujarat, India

*Abstract* - **Cloud computing is the framework of recent technology and it can set the vision of computing utilities into a reality. The most promising part of cloud computing is that it provides dynamic provisioning and delivers facility for computation, networking, storage etc using datacenters. The working of cloud resources provided through the Internet and it gives services on a pay-per-use basis from different cloud vendors. IaaS services used to develop background processing, PaaS provides programming platforms for developing applications and SaaS provides benefit to cloud users from the elastic scalability without making processes such as software development, installation, configuration, maintenance etc. The migration of a virtual machine has advantages such as increasing energy efficiency, resource sharing and load balancing, fault resilience etc. The pre-copy algorithm of Xen hypervisor is very well known and it is also used in real applications to run migration activities. The migration is categorized into three major types to improve pre-copy: i) compressing memory pages ii) predicting dirty pages iii) reducing dirty pages (performance modeling). Each category has its own importance for the live migration. At first, the compression model is proposed to achieve efficient virtual machine migration. The time series based techniques are developed using historical analysis of past data in second category. The time series is generated with consecutive iterations of memory pages. Here, two different regression-based models of time series are proposed. In the third category, the performance modeling of live migration is proposed by vMeasure approach. These models are evaluated for measuring the computation of virtual machine (VM) downtime, the total number of pages transferred and total migration time.**

**Keywords— Cloud, Virtualization, Infrastructure service, Datacenter**

## I. INTRODUCTION

Cloud computing [1-4] is the framework of recent technology which can set the vision of computing utilities into a reality. The most promising part of cloud computing is that it provides dynamic provisioning and it is not limited to apply with only cloud services but it is also provides facility for computation, networking storage etc. The working of cloud resources provided through the Internet and it gives services on a pay-per-use basis from different cloud vendors. The framework of cloud computing is shown in Figure 1. The user can make the VM request by two ways: i) computation of data using compute cloud and ii) storing of data using storage cloud.
The three major datacenter operations are network, storage and server virtualization [1]. The network virtualization is used to split up network bandwidth into multiple channels. The virtual server is made with using multiple servers to provide server virtualization. The storage virtualization is used for pooling of resources from multiple network storage devices.

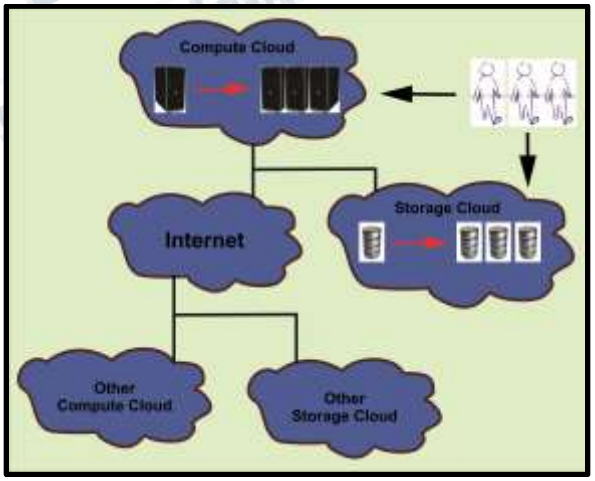


***Figure 1. Fundamentals of Cloud Computing***

The cloud computing is the collection of five different characteristics, three basic service models and four deployment models. Five characteristics of cloud computing [1,4] are i) on-demand self-service ii) broad network access iii) resource pooling iv) rapid elasticity and v) measured service.
i) On-demand self-service: When the services (like computing, storage etc.) provided to consumers by service

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

provider are provision automatically without demanding any human interaction, then this concept is known as on-demand self service. For example, the user can take benet of cloud hosting services rovided by cloud host provider securely. The user has access to use the services and he can also authorize to change services through the service provider's online facility. He can have full control to add or remove software's, services and storage facility whenever needed.

ii) Broad network access: The network access is provided with the help of heterogeneous thin or thick client platforms like mobile phones, laptops, tablets etc.

For example, Apple iCloud is a well known service in which user is allowed to store documents on the cloud and access them from any device connected. This can facilitate to take a picture while traveling with a mobile and edit the same picture on laptop. The user can check the updated picture on tablet. The user does not need to set up cables and connect devices with each other.

iii) Resource pooling: In resource pooling, multiple consumers are served using multi-tenant service and computing resources are pooled between physical and virtual resources, dynamically assigned and reassigned according to consumer demand.

For example, different resources provided are storage, network, processing and these are shared with location independence way in that the customer is not able to know the exact location of the provided resources but the location can be provided by understanding the higher level of scenario.

iv) Rapid elasticity: The capabilities of cloud resources can be elastically provisioned and released to scale outward and inward with demand is known as elasticity. The capabilities available is considered unlimited and with any quantity at any time.

v) Measure service: The resource usage is controlled and optimized with a metering capability at some level of abstraction for the services like computing,storage, bandwidth etc. The resource usage can be monitored and reported so that transparency for the provider and consumer can be maintained.

For example, the metering service of cloud is charged per usage metrics just like electricity or municipality water services. The more the services are utilized the customer has to pay higher bill.

## II. AIM AND SCOPE OF WORK

VM migration is mainly performed in cloud with two approaches: i) pre-copy and ii) post-copy. The post copy has its limitation compare to pre-copy and it also degrades performance with memory intensive pages. The aim of pre-copy based live migration is to reduce repetitive transmission of memory pages. This method copies all pages in first round and iteratively copying modified pages [2,5].

The key issue to manage workloads is to know how efficient migration can take place between servers. The migration process has main three parameters to perform optimal migration: i) VM size ii) dirty rate and iii) data rate (bandwidth).

## III. PRE-COPY METHOD CLASSIFICATION FOR LIVE MIGRATION OF VIRTUAL MACHINES

In this section, improved pre-copy approaches are discussed in three major fields (shown in Figure 2):

i) compression methods based improved pre-copy approaches ii) improved pre-copy algorithms designed for regression models and iii) performance models based improved pre-copy algorithms. The primary techniques are used to migrate virtual machines using reduction of dirty pages.

### *A. Techniques for Reducing Dirty Pages:*

•Stunning rogue processes & freeing page cache pages [6]
•CPU scheduling techniques [7-10]

### *A.1 Stunning Rogue Processes & Freeing Page Cache Pages*

The basic methods of process migration like stunning rogue processes and freeing page cache pages have been developed in [6]. Stunning rogue process Stunning rogue method is developed to limit dirty pages transmission, which skips pages that have been dirtied in the previous iteration. Pre-copy migration can perform optimal when dirty rate becomes lower than transmission rate. At the time of migration starts, a tread is being generated to mitigate the risk of dirty rate comparing to transmission rate using kernel analysis.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

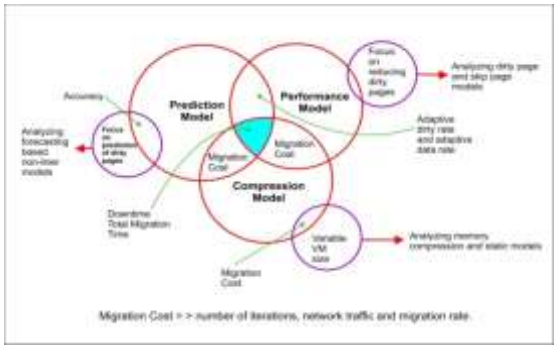


***Figure 2. Classification of Live Migration using Pre-copy***

***A.2 Free Page Cache Method***

In Linux, non volatile storage is used to store data and it is known as a cache. When this data is read again, it can available quickly from cache. A number of free pages are available in OS at any time so when migration demands pages, it will provide from these pages to Xen. This way, it is able to reduce migration time at the first iteration and processes further iterations.

***A.3 CPU Scheduling Techniques***

The objective of CPU scheduling algorithms [7-10] is to reduce the rate of dirty memory generation. In this procedure, allocated VCPU can control the writing into memory. At the time of faster writing, VCPU gets slow down and it can control overall data transferred in the last round. CPU is being scheduled when VM writes too fast memory during pre-copy. It manages dirty rate for allocation of CPU with proper time distribution to work effectively.

The main idea of the algorithm is, if the memory is written too fast, it is able to schedule the CPU time for different VMs to mange percentage ratio, so that the dirty rate could be adjusted. This way it can give small value for downtime and total migration time in improved pre-copy algorithm.

***B. Improved Pre-copy Methods using Compression:***

Delta Compression [11], MEMCOM [12], Huffman Coding [13,14] and LZ method [15] Compression techniques are used in computer networks and data storage systems for their effective outcome and the same techniques are used to increase the efficiency of data transfers and reduce space issues on the destination device. In compression mechanism, pages are compressed at the source to generate faster transfer and decompressed at the destination using various compression techniques. Compression of memory pages can be performed by various methods named delta compression, MEMCOM, Huffman coding, andLZ algorithm.

Existing non-compression methods have their ability to migrate VM with downtime and total migration time, but these methods have limitation to compress pages and it can lead to more iterations during migration for these methods. Compression is very useful technique to work with memory intensive pages and the higher performance gain is achieved at the time of various heavy workloads.
The delta compression is the most effective method for evaluation of performing heavy workloads. The delta compression is also an efficient method for combined approach and it does not give much memory overheads which are main issues with the methods such as MEMCOM, comIO, hu man algorithm etc.

***C. Basic Improved Pre-copy Techniques***

- LRU Algorithm [16,17]
- Matrix Bitmap Algorithm [18]
- HCA Hierarchical copy algorithm [19]
- Workload adaptive live migration with CLB (Central Load Balancing) [20]
- Combined techniques [15, 21-23]

***C.1 LRU Algorithm***

LRU working set prediction [16] operates with working set contains most recently used pages while least recently used pages are in the inactive list. A page is placed into working set list when a new page is faulted into main memory. If the working set list is full then most recently page is kept in working set by transferring least recently used page. If inactive list is full then it shows the memory of VM is full and the least recently used page of inactive list is rejected.

***C.2 Matrix Bitmap Algorithm***

The main challenge of this algorithm [18] is that at the time of monitoring the dirty page before iteration migration, it is needed to lock the whole structure so it can generate overhead in the migration process.
The algorithm depends on the past values and predict the next values. In this algorithm, dirty bitmap is collected many times to decide whether to send the page or not. This process makes little overhead and combining compression module with matrix bitmap would improve the performance.

***C.3 HCA Hierarchical copy algorithm***

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

HCA algorithm [19] is developed to modify kernel structure to improve pre-copy algorithm. HCA has following data structures:
Dirtypage amount: to record the total amount of dirty pages, its initial value is 0.
Total dirtycount: to record the modification time of dirty page, its initial value is 0.
Dirty count(new data structure): to get number of modified times of page.
It works based on three layers to deal with dirty pages: Layer 0 (no dirty page), Layer 1 (pages without high dirty rate) and Layer 2 (pages with high dirty rate not transfer in next round).

***C.4 Workload adaptive live migration with CLB (Central Load Balancing)***

The algorithm [20] works for generic and memory intensive workloads. The main part of the workload adaptive algorithm is to provide load balancing activity using centralized approach. The limitation of this algorithm is that the live migration is not supported with network aware mapping decisions to reduce network traffic.

***C.5 Combined Techniques for Live Migration [15, 21-23]***

There are number of pre-copy algorithms have been implemented with this approach. An optimized algorithm for dynamic migration is presented in [15]. A working set based method is proposed to have two modules: i) probability prediction and ii) memory compression. The device driver based LZ algorithm has been for the dynamic migration. The dynamic migration mechanism based on a combined approach including layer based copy algorithm and memory compression algorithm has been discussed in [21].

D. Time-series Methods
•Time Series based Pre-copy Approach [24]
•Context Based Prediction algorithm (CBP) [25]
•Kalman filter method [26]

Both methods CBP and Kalman filter are able to shorten the parameters like total migration time, downtime and total pages transferred significantly compared to default time-series based pre-copy algorithm. The advantage of these methods named time series based pre-copy approach [24] and Kalman filter [26] are that they are able to work on past observations and prediction of future data efficiently. The core idea of these methods is to send frequently updated pages in the last round by analyzing WWS in each iteration.

The performance is better in the case of high dirty page rate or low network bandwidth in [25]. This method works with two dimensional array of bitmap which can indicate the historical statistics of the bit of a page. This method is faster than [24] because it works with fixed sizes of pages.

The Kalman prediction works based on muti-stage transmission strategy in which threshold gets changed dynamically. The simulation model using Kalman is designed in [26] and the above methods [24, 25] have implemented real model of live migration using time-series model. This prediction method is not much popular due to its complex working model.

The novel study of time-series is based on prediction based models using time series analysis given in [24-26]. The existing techniques are simple but it is not able to give an optimal accuracy of dirty pages prediction. In future work, the statistical prediction model and learning based prediction model can be applied for further improvement.

***E. Performance Modeling***

The existing methods on performance modeling are shown below:
•Performance and energy models for live migration [27]
•Analytic performance model [28]
•iAware system [29]
•Pacer system [30]

In [27], two models based on performance and energy parameters are evaluated. These models are used to evaluate network traffic and downtime. The performance modeling part is discussed in first model and the energy modeling is shown in second part. The exponential method is used to calculate total migration time, service time and network traffic using performance modeling. The energy modeling is discussed with the concepts of green computing and the main advantage of this method is to migrate virtual machine with minimum energy cost.
The uniform dirtying rate is maintained using analytic performance model [28]. This analytical model is based on two types of hot pages which are used to minimize the migration of pages effectively. The focus of this method is to work at the service downtime phase. An iAware system [29] is a live virtual machine migration technique that is designed with lightweight interference-aware VM.

It is designed with two points i) VM performance interference and ii) key elements. The iAware system has been developed to manage effective load and reduce power while migrating virtual machine. In [30], the hybrid cloud is discussed to manage VM migration. The advantage of Pacer is that it can predict the migration time in all pre-copy rounds, and also, synchronizing the migrations of multiple application elements to improve the performance.The novel study of dirty page rate models are developed on performance modeling given in [27-30]. Existing dirty models [27,28] have considered adaptive dirty rate scenario with their variable WWS size which can allow transferring few dirty pages of memory intensive pages in migration system. In extended dirty page model, the WWS size is set with a maximum size for write (memory) intensive pages. The extended dirty model can outperform better than existing dirty models.

## IV. CONCLUSION

Following are conclusions of the comparative analysis:
1. Combined technique is able to give optimal performance compare to basic pre-copy based techniques. Large VM requires more pages to manage so migration can be stuck at any point of time. Compression of VM is the necessary action to avoid such issues.
2. Size of VM has rational impact in VM migration. The non-adaptive data rate can improve the performance compared to adaptive. Network traffic can be increased but it minimizes downtime. Total migration time is also reduced.
3. Proposed time-series pre-copy models are able to identify dirty pages accurately in-advance and improve the performance by transferring frequently modified dirty pages in last round.
4. Adaptive dirty rate outperforms better than non-adaptive dirty rate during migration of VM. The network traffic is minimized in adaptive dirty rate compared to non-adaptive. The downtime may increase and total migration time is also affected due to unique dirty pages increased.

## V. REFERENCES

[1] Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). Mastering cloud computing: foundations and applications programming. Newnes.
[2] Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: towards a cloud denition. ACM SIGCOMM Computer Communication Review, 39(1), 50-55.

[3] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616.

[4] Mell, P., Grance T. NIST working definition on cloud computing. National Institute of Standard and Technology (NIST); [Online Document] Available at http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[5] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., & Warfield, A. (2003). Xen and the art of virtualization. ACM SIGOPS Operating Systems Review, 37(5), 164-177.

[6] Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., & Warfield, A. (2005, May). Live migration of virtual machines. In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2 (pp. 273-286). USENIX Association.

[7] Jin, H., Gao, W., Wu, S., Shi, X., Wu, X., & Zhou, F. (2011). Optimizing the live migration of virtual machine by CPU scheduling. Journal of Network and Computer Applications, 34(4), 1088-1096.

[8] Liu, Z., Qu, W., Liu, W., & Li, K. (2010, December). Xen live migration with slowdown scheduling algorithm. In Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2010 International Conference on (pp. 215-221). IEEE.

[9] Liu, W., & Fan, T. (2011, August). Live migration of virtual machine based on recovering system and CPU scheduling. In Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International (Vol. 1, pp. 303-307). IEEE.

[10] Stage, A., & Setzer, T. (2009, May). Network-aware migration control and scheduling of differentiated virtual machine workloads. In Proceedings of the 2009 ICSE workshop on software engineering challenges of cloud computing (pp. 9-14). IEEE Computer Society.
[11] Svard, P., Hudzia, B., Tordsson, J., & Elmroth, E. (2011). Evaluation of delta compression techniques for efficient live migration of large virtual machines. ACM Sigplan Notices, 46(7), 111-120.

[12]Jin, H., Deng, L., Wu, S., Shi, X., & Pan, X. (2009, August). Live virtual machine migration with adaptive memory compression. In Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on (pp. 1-10). IEEE.

[13] Pigeon, S., & Bengio, Y. (1997). A memory-efficient Huffman adaptive coding algorithm for very large sets of symbols. Universit de Montral, Rapport technique, 1081.

[14] Knuth, D. E. (1985). Dynamic human coding. Journal of algorithms, 6(2), 163-180.

[15] Sun, G. F., Gu, J. H., Hu, J. H., & Zhao, T. H. (2011). al, Improvement of Live Memory Migration Mechanism for Virtual Machine Based on Pre-copy. Computer Engineering, 37(13).

[16] Zaw, E. P., & Thein, N. L. (2012). Improved live VM migration using LRU and Splay tree algorithm. International Journal of Computer Scienceand Telecommunications, 3(3), 1-7.

[17] Alamdari, J. F., & Zamanifar, K. (2012, December). A reuse distance based precopy approach to improve live migration of virtual machines. In Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on (pp. 551-556). IEEE.

[18] Cui, W., & Song, M. (2010, August). Live memory migration with matrix bitmap algorithm. In Web Society (SWS), 2010 IEEE 2nd Symposiumon (pp. 277-281). IEEE.

[19] Liu, Z., Qu, W., Yan, T., Li, H., & Li, K. (2010, October). Hierarchical copy algorithm for Xen live migration. In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2010 International Conference on (pp. 361-364). IEEE.

[20] Lu, P., Barbalace, A., Palmieri, R., & Ravindran, B. (2013, August). Adaptive Live Migration to Improve Load Balancing in Virtual Machine Environment. In Euro-Par Workshops (pp. 116-125).

[21] Jing, Y. (2012, January). Key technologies and optimization for dynamic migration of virtual machines in cloud computing. In Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on (pp. 643-647). IEEE.

[22] Zhang, Z., Xiao, L., Zhu, M., & Ruan, L. (2014). Mvmotion: a metadata based virtual machine migration in cloud. Cluster Computing, 17(2), 441-452.

[23] Cui, W., & Song, M. (2010, August). Live memory migration with matrix bitmap algorithm. In Web Society (SWS), 2010 IEEE 2nd Symposiumon (pp. 277-281). IEEE.

[24] Hu, B., Lei, Z., Lei, Y., Xu, D., & Li, J. (2011, December). A time-series based precopy approach for live migration of virtual machines. In Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th In-ternational Conference on (pp. 947-952). IEEE.

[25] Yong, C., Yusong, L., Yi, G., Runzhi, L., & Zongmin, W. (2013, November). Optimizing Live Migration of Virtual Machines with Context Based Prediction Algorithm. In 1st International Workshop on Cloud Computing and Information Security. Atlantis Press.

[26] Sun, Z. H,, Hu, X. L. (2014). Live Migration for Virtual Machine Based on Kalman Prediction of Dirty Pages. Applied Mechanics and Materials,Vols. 668-669, pp. 1363-1367.

[27] Liu, H., Jin, H., Xu, C. Z., & Liao, X. (2013). Performance and energy modeling for live migration of virtual machines. Cluster computing, 16(2), 249-264.

[28] Aldhalaan, A., & Menasc, D. A. (2013). Analytic performance modeling and optimization of live VM migration. In Computer Performance Engineering (pp. 28-42). Springer Berlin Heidelberg.

[29] Xu, F., Liu, F., Liu, L., Jin, H., Li, B., & Li, B. (2014). iaware: Making live migration of virtual machines interference-aware in the cloud. Computers, IEEE Transactions on, 63(12), 3012-3025.

[30] Zheng, J., Ng, T. S., Sripanidkulchai, K., & Liu, Z. (2013). Pacer: A progress management system for live virtual machine migration in cloud computing. Network and Service Management, IEEE Transactions on, 10(4), 369-382.