© May 2016 | IJIRT | Volume 2 Issue 12 | ISSN: 2349-6002

Reducing VM Migration Time By Compression Algorithm And Setiing Threshold Of Dirty Page Rate

Nirali Raval[1], Rikin Thakkar[2]
[1]*Student, Computer Engineering, Silver Oak College of Engineering and Technology,*
[2]*Asst.Professor, Computer Engineering, Silver Oak College of Engineering and Technology,*
Ahmedabad, India

***Abstract*— This paper gives an overview of the different works in literature that consider virtual machine migration. The different works related to virtual migration are classified into different categories. Some of the works that consider less explored areas of virtual machine migration are discussed in detail. improved time-series based precopy approach technique is proposed. With the time-series prediction technique, frequently updated dirty pages (high dirty pages) are identified more precisely, and transmit them in the last round of iteration, in order to reduce unnecessary, repeated transmission of dirty pages. transferring huge number of unnecessary memory pages resulting into increase in the total migration time and downtime. Memory Compression, a faster migration technique, is the second step of a combined approach. Various compression methods named RLE, Huffman Coding, MEMCOM, WKdm and LZ are classified. Proposed approach of modifying optimized pre-copy will reduce unnecessary transfer of pages and its combination with Characteristic Based Compression (CBC) algorithm will handle two factors viz. (i) total migration time and (ii) total downtime and make migration process more effective.**

***Index Terms*— Cloud Computing,Virtualization,Virtual Machine Migration**

I.INTRODUCTION

Cloud computing aims to build an ―internet age without personal computer‖ and satisfy the growing without personal computer‖ and satisfy the growing requirement of customers through integrating the whole network computing service. Virtualization technology, being the most effective solution to the resource management of cloud computing, excels in its virtual machine real-time migration technology . Since the traditional virtual machine dynamic migration still exists, artificial-initiated migration cannot satisfy the requirements of the ―Cloud‖ and the migration can only happen between the isomorphic virtual machine monitors. These two defects are the bottleneck on the healthy development of the virtual machine dynamic migration technology.

Virtualization means more than one virtual machines (VM) on single physical machine. Virtualization is mostly useful to handle workload balancing between physical machines in datacenter when available resources are not enough for vms.

In process of migration, virtual machine move from one physical machine to another. In offline migration process is stopped till the virtual machine can continue on target machine and in live migration user can execute without interrupted. Live migration is a migration during which the VM seems to be responsive all the time from clients' perspective .Live migration is key selling point for state-of-the-art virtualization technologies.

Memory transfer is divided into three phases :
1) Push phase: Certain pages are pushed in advance to the new destination before the source virtual machine moves to destination. Some modified pages are resent to ensure consistency.
2) Stop and copy phase: It is a simple way in which source virtual machine is stopped at source and complete Virtual machine is copied or moved to destination and then resumed it at destination.
3) Pull phase: In pull phase virtual machine starts execution on destination machine and if it requires
a page that has not been copied, then this page is pulled across the network from the source virtual machine.
Aim of this paper is to review some of the existing live migration techniques.

Live Migration Approaches
There are two most frequently used live migration approach viz. (a) Pre-copy approach (b) Post-copy approach.Pre-copy serves the combination of push phase with stop and copy phase as shown in Fig.1. Post-copy serves combination of pull phase with stop and copy phase as shown in Fig.2.

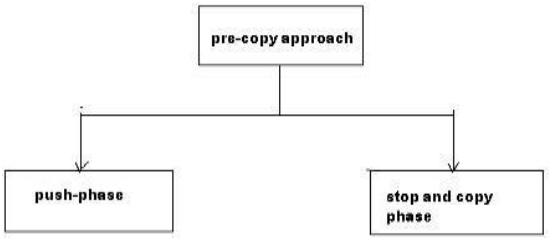

Figure 1. Pre-copy approach

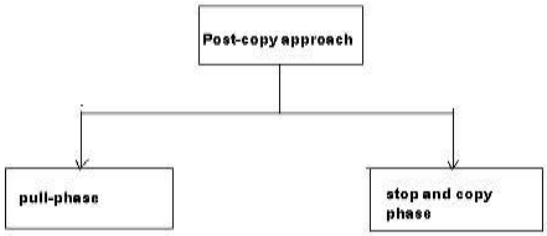

Figure 2. Post-copy approach

In pre-copy approach pages of memory are iteratively copied from the source machine to the destination host, all without ever stopping the execution of the virtual machine being migrated .In post-copy approach each memory page is transferred only once, which is the main benefit over pre-copy approach. Precopy approach provides high reliability against system fault. Because it contains copy of the memory pages in both sides, whereas in post-copy approach memory page can be found at one side only .

II.MODIFIED PRECOPY AND COMPRSSION ALGORITHM

A Modified PreCopy Algorithm

The original precopy approach of Xen is not suitable for the situation of high dirty pages. This is because it would repeatedly transmit a large number of identical dirty pages in iteration when VM's memory is in read and write intensive access. Thus it is not efficient for live migration of VM to converge at a specific time. To address this drawback, a time-series based precopy algorithms is proposed. In this algorithm, high dirty pages are identified by their historical statistics of the TO_SEND b it map, to avoid them being transferred again to the target host in this iteration. Until at the end of the last iteration, will be these high dirty pages transferred to target host. The benefits brought in are obvious: the number of iterations of migration decreases, the down time and migration time also decrease. In other words, the VM can be moved to the target host much faster and more transparently to meet SLAs objectives .

Two key parameters K and N, and an array of historical bitmap called TO_SEND_H with size N are used in the proposed algorithm. The TO_SEND_ H is used to save the past statistics of the TO_SEND bit map. The N stands for the max size of the time -series array TO_SEND_H, and the K stands for the threshold of high dirty page, which means only dirty pages whose appearing times in the TO_SEND_H array passing K can be identified as high dirty pages. During the process of iteration, the data of the TO_SEND bitmap will be saved to array TO_SEND_H according to pointer ptr which indicates the current position of the array, and then pointer ptr will move to next position.

To better understand the proposed approach, consider memory page as p, dirty pages as to_send, dirty pages that should be skipped in iteration as to_skip, historical dirty pages set as to_send_h. The following equations are used to determine when not to send dirty page p:

$$p \in to\_send \,\&\, p \in to\_skip \quad \ldots (1)$$
$$p \notin to\_send \,\&\, p \in to\_skip \quad \ldots (2)$$
$$p \notin to\_send \,\&\, p \notin to\_skip \quad \ldots (3)$$

If dirty page p meets one of the above equations , in an iteration we would not send p in this iteration. For equation (1), it means memory page p belongs to the type of dirty page, but it is marked that it should be skipped to send. For equations (2) and (3), memo ry page *p* is not the type of dirty page that should be sent. So it is obvious that p should not be sent to target host in this iteration.

$$p \in to\_send \,\&\, p \notin to\_skip \quad \ldots (4)$$

If dirty page p meets the above equation (4) in iteration, then further consideration must be taken. In the traditional precopy algorithm of Xen, memory page p would be sent to target host, because it has been modified. Actually it is not advisable. It does not consider the fact that some memo ry pages are the type of frequently updated dirty pages. If these frequently updated dirty pages are transferred repeatedly, in each iteration, it would consume a lot of unnecessary migration time. Thus, here we use the historical dirty pages set to_send_h to assist in making decision of whether to send p or not. Below is the equation of identifying high dirty page p:

$$\sum_{i=1}^{N} (p \in to\_send\_h[i]) \geq k \quad \ldots (5)$$

For equation (5), it compares the count of dirty page p that appears in the historical dirty pages set to_send_h with size N,

with the preset threshold K. If equation (5) is true, consider dirty page p to be high dirty page, which means that p should not be sent in this iteration until the end of all iterations. Otherwise consider dirty page p to be the type of low dirty page, which means that it should be sent to the target host, as there is high probability that it would not be modified in the future.

B Memory Compression Algorithm

Live migration can take advantage to complete migration of virtual machine in minimum time. Faster migration is achieved by compressing memory pages. In this mechanism, pages compressed at the source and decompressed at the destination using various compression techniques.

1) Delta Compression based RLE

RLE is a Run Length Encoding method to compress strings of 0s' or 1s'. Delta compression finds out portions in a dirty page which cloud be changed known as page delta. This delta is replicated to the delta of backup instead of sending entire page and the whole page is XOR'ed with previous version of it. During this process, differences are Run Length Encoded which is later on required to restore the page. To get original page, RLE result is decoded, and the outcome is XOR'ed with the content of the page. This way, system throughput can be increased and it decreases total migration time duirng iterations.

2) MEMCOM based Adaptive Algorithm

MEMCOM [13] uses memory compression to allow efficient, stable virtual machine migration using zero-aware compression algorithm for maintaining cost and performance of live migration. Data being transferred in each round is first compressed and on the target, they are decompressed. To avoid the additional overhead in terms of compression time, multi-threading technique is used to parallelize compression tasks. Base on the minimum threshold of dirty pages at which pre-copy processes stopped, high compression ratio is gained. Word similarity of each one memory page with a 16-word dictionary (which means 16 words recently seen) in a page is maintained. The number of matched words in page is named word-similarity of a page. It works based on high word similarity (using WKdm algorithm which is combination of dictionary and statistical techniques), low word similarity (using LZO algorithm which is based on dictionary technique) and pages consists of a so many of zero bytes and random non-zero bytes. LRU and direct mapped cache are used for a replacement policy of strong similarities. MEMCOM allows high dirty page rate to compare to the pre-copy of Xen and it was also tested on different application scenarios that produced good results. When many clients access servers simultaneously and a number of virtual machines access a large amount of data at the same time like at save or restore function, ComIO [22] can quickly save/restore virtual machines using page compression. Fast enhanced characteristic based com-pression algorithm (ECBC) is proposed with multi-threaded technique, which is hybrid of several compression methods and the compression tasks are parallelized for much short ended compression time. In this method, the data of needed page are directly extracted from the compressed block so it indirectly augments the effective storage space.

3) Huffman Coding for Log Data Compression

Huffman coding is used in log compression technique for making the most effective and fast migration of the virtual machine. The parallel multithreading technique is used in association with Huffman coding. A checkpoint is the process of storing the state of a system. Replay is used for re-execution of the past system. It is helpful for debugging and proactive failure. The trace daemon is continuously monitoring the discrete event and generates the logs. It transfers checkpoint to the target after that logs are passed to the destination host. Before transmitting the log files to the destination, the files are compressed and transmitted to the destination. The discussion given in the paper [23] is used to quantify the tradeoffs between the usability and costs of several checkpoint compression techniques and provide insights which can direct selection decisions. Three compression techniques which includes gzip, delta compression, and similarity compression are evaluated on different workloads. Similarity compression is used to identify content redundancy for memory pages, or the VM checkpoints, and send only one copy of the duplicate elements therefore it is able to reduce replication traffic. In global clock algorithm [24], if pages are not being recently accessed, then they will not be accessed in near future. This kind of concept is proposed in Difference Engine which supports multiple compression algorithms named LZO and WKdm.

4) Device driver based LZ algorithm

In Dynamic migration , memory compression is per-formed by a loadable device driver program. Migrated Pages are put on compression memory, sealed and then migrated into the target machine. Three modules named compression memory management module, compression algorithm module and

block device driver are combined to get compressed pages. Lempel-Ziv compression algorithm is used in this paper. In [19], similar memory compression modules are introduced which is given in dynamic migration approach.

C Watermarking Algorithm

The modified QIM (quantization index modulation) technique is used for embed the medical information of patient in medical image.A modified QIM based watermarking method of medical image is proposed in this work where each watermark bit is spreaded over N-mutually orthogonal signal points. First cover signal is projected on *N*-mutually orthogonal signal points in *N*-dimensional signal space. The projection may be accomplished in down sampling the cover by a factor of *N*. The cover (***X***) can mathematically be represented as $\boldsymbol{X = \{X_1, X_2 ,\ldots, X_N\}}$where$\{X_i\}$ is the signal coefficients corresponding to complete orthogonal basis function set[7]

III.PROPOSED ALGORITHM AND PROPOSED ALGORITHM WITH COMPRESSION TECHNIQUE

A)Run Length Encoding:

RLE is a Run Length Encoding technique to pack series of 0s' or 1s'. Delta pressure discovers bits in a grimy page which cloud be changed known as page delta. This delta is duplicated to the delta of reinforcement as opposed to sending whole page and the entire page is XOR'ed with past adaptation of it. Amid this procedure, contrasts are Run Length Encoded which is later on required to reestablish the page. To get unique page, RLE result is decoded, and the result is XOR'ed with the substance of the page. Along these lines, framework throughput can be expanded and it diminishes absolute relocation time duirng cycles.

B)Characteristics Based Compression:

CBC include two Parts: (1) Determining the sort of pages being compacted. The expectation of the progression is to choose pages with solid regularities, which mean high comparability or to a great degree high recurrence of zero bytes. With the assistance of a lexicon, we tally the comparable words (complete or incomplete coordinating with the lexicon) and zero-bytes. The lexicon dependably keeps as of late seen words. So as to abatement the coordinating overhead, our calculation deals with this lexicon as a direct mapped reserve and LRU (Least Recently Used) will be utilized as the swap calculation for the word reference. (2) Making decision of fitting solid pressure calculation. A banner demonstrating the sort of a page is additionally included into the compacted yield for future decompression.

Step1: Host Request for new virtual machine
Step 2: Check condition VM having load less than 50%
Step 3: If available than then allocate VM to host having load less than 50% else allocate VM to host having lowest load
Step 4: After every 10 second check host load
Step 5:Any host having load more than 90%
Step 6:If no then again check host load else find host having load less than 50%
Step 7:If host having load less than 50% available then find all dirty pages else find lowest load host and then find dirty pages
Step 8: After find dirty pages . transfer the pages which not dirty to new VM
Step 9: then apply compression algorithm on dirty pages
Step 10:then transfer dirty pages to new VM

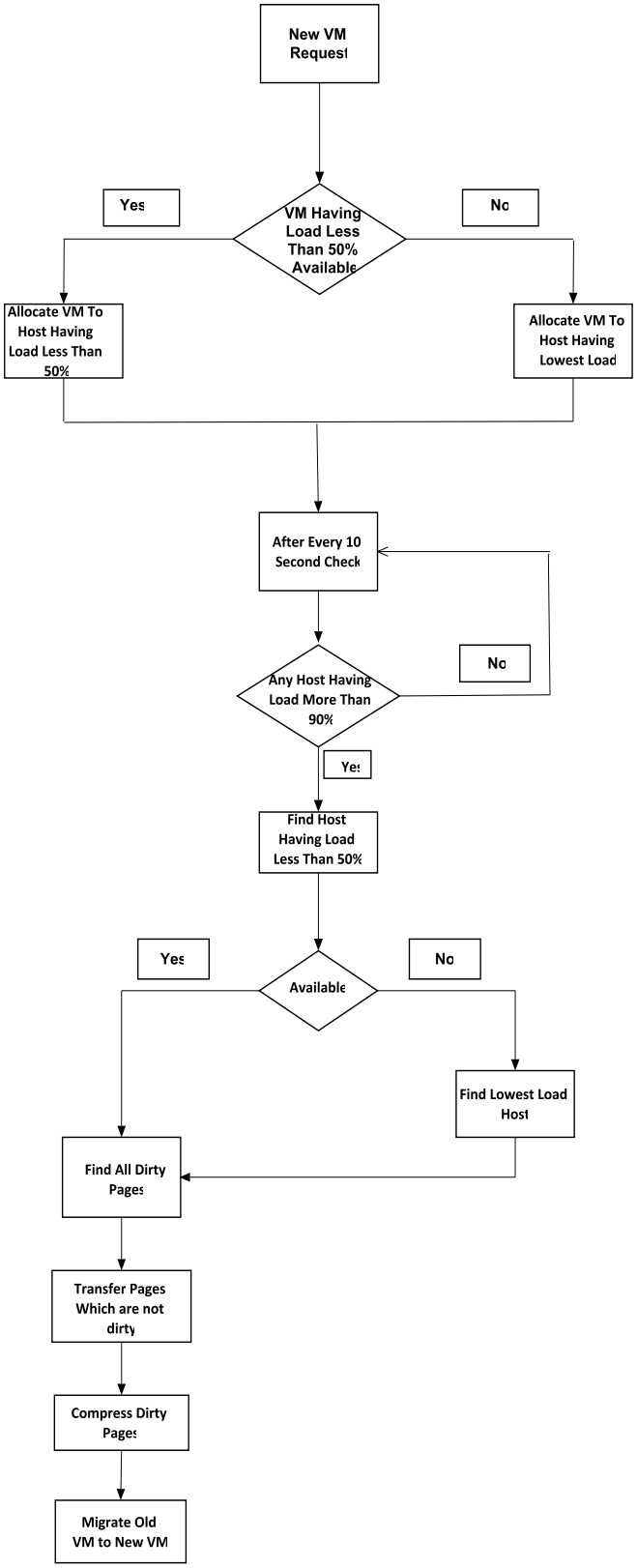

Figure 3.1 PROPOSED ALGORITHM

IV. EXPERIMENTAL RESULTS

Memory Size	Pre Copy Algorithm		Proposed Algorithm		Proposed Algorithm With Compression			
	Down Time	Migration Time	Down Time	Migration Time	Down Time	Migration Time	Down Time	Migration Time
128 MB	9.234	205.224	7.217	191.307	6.942	197.635	6.902	192.859
256 MB	13.0	397.556	11.051	362.645	10.584	371.942	10.529	364.310
512 MB	18.322	782.22	16.714	726.729	16.284	739.284	16.159	728.641
1028 MB	24.654	1469.12	21.571	1406.942	20.983	1454.495	20.776	1436.423

Table4.1 Result Table

Result table clearly show the comparison between different algorithm. Proposed algorithm reduce both parameter down time and migration time as compare to pre-copy algorithm. Using compression technique CBC and RLE with proposed algorithm reduce better downtime as compare to proposed algorithm but increase total migration time that's why we use multithreading technique with CBC and reduce both down time and total migration time.

V. CONCLUSION

Virtual machine migration is latest research topic and many researchers are working on different Virtual machine migration algorithms to reduce down time as well as migration time. By applying base paper algorithm, we can reduce the down time. By applying our algorithm it can much reduce down time but, migration time is increased. So I am proposing such formula which can reduce down time as well as migration time.

REFERENCES

[1] Migration, Christina Terese Joseph, Prof. K Chandrasekaran, Robin Cyriac. A Perspective Study Of Virtual Machine 978-1-4799-3080-7/14/$31.00_c 2014 IEEE

[2] Shoaib Hassan,AsimAbbas Kamboh,Dr. Farooque Azam.Analysis Of Cloud Computing Performance,Scalability,Availability,&Security 978-1-4799-4441-5/14/$31.00 ©2014 IEEE

[3] J.Arputharaj Johnson. Optimization Of Migration Downtime Of Virtual Machines In Cloud ICCCNT 4 JULY 2013, India IEEE

[4] Petter Sv¨ard and Johan Tordsson, Benoit Hudzia, Erik Elmroth High Performance Live Migration Through Dynamic Page Transfer Reordering And Compression 2011 Third IEEE International Conference on Cloud Computing Technology and Science.

[5] Adel Amani, Kamran Zamanifar. Improving The Time of Live Migration Virtual Machine by Optimized Algorithm Scheduler Credit.

[6]Minal Patel, Sanjay Chaudhary Survey on a Combined Approach using Prediction and Compression to Improve Pre-copy for Efficient Live Memory Migration on Xen 978-1-4799-7683-6/14/$31.00©2014 IEEE

[7]Hai Jin, Li Deng,Song Wu, Xuanhua Shi,Xiaodong Pan Live Virtual Machine Migration with Adaptive Memory Compression 978-1-4244-5012-1/09/$25.00 ©2009 IEEE

Computer and Telecommunication Systems 1526-7539/14 $31.00 © 2014 IEEE

[8] Megha R. Desai, Hiren B. Patel Efficient Virtual Machine Migration in Cloud ComputingIEEE 2015