An Optimized Time Series based Two phase strategy Pre-Copy Algorithm for Live Virtual Machine Migration

Ruchi Tailwal
P.G. Scholar,
Graphic Era Hill University
Dehradun, India

Avita Katal
Assistant Professor,
Graphic Era Hill University
Dehradun, India

***Abstract*—Virtualization technology is core concept of cloud computing. Virtualization permits creation and migration of numerous virtual machines on physical machine. Advantage of virtualization is that live migration of VM is conceivable, which is handy in load balancing of VM. The key test of live migration is to accomplish least migration time and downtime so that the application running on virtual machine is suspended for insignificant time. In this paper an optimized time-series based pre-copy approach is proposed, which adjusts the benefits of enhanced time series based approach and gives second opportunity to the pages before transmitting to the target host. With the two-stage methodology if a page is unmodified for two successive rounds then page is migrated to destination, else it is sent in the last iteration. This leads to reduction in no of pages being transferred, resulting decreased migration time and total migration time.**

Keywords— Virtualization, Live migration, Virtual machine, Migration time, Downtime.

I. INTRODUCTION

Cloud computing is a huge distributed computing environment which contains large amount of virtualized computing resources available for individual or an organization. Cloud computing intended to construct an internet era without PC and fulfill the developing necessity of clients through incorporating the entire network computing service. [1].

Virtualization is a core concept of cloud computing, mostly determined by advantage of application isolation, resource sharing, fault tolerance, load balancing. In virtualized environment multiple applications can run concurrently. The hypervisor (Virtual machine monitor) creates multiple virtual machines on physical host. A virtual machine is a software computer that, like a physical computer, that runs its own particular operating system and applications like a physical computer.Each virtual machine has its own Operating system and virtualized CPU is shared among all virtual machines [4]. Virtualization permits construction and migration of a few virtual machines on physical computer. Live migration portrays the procedure of replication of VM from one physical host to the next physical host, without shutting down the VM. It gives unique advantage to server virtualization. Some of these incorporate [5]:

Power management: It aims to merge virtual machines through live migration on an ideal number of servers and specifically switch off underutilized servers to diminish data center power utilization.

IT maintenance: Administrator can evidently move virtual machines to free and power off hosts for maintenance reason.

Load balancing: It aims to fiddle VM placement to accomplish basic business objectives, for example, high throughput and high accessibility.

This paper is organized as follows. Section 2 describes the background work done in this area. Section 3 briefly outlines the existing work done. Section 4 introduces the proposed optimized two phase strategy algorithm followed by result and discussion in Section 5. Section 6 introduces future work.

II. BACKGROUND

Virtualization innovation empowers administrator to move VM of source computer to the destination computer, which empowers less power consumption, load adjusting and high resource sharing. Hypervisor (Virtual Machine Migration (VMM)) allows clean separation between hardware and software.

Live migration permits the server executive to migrate a running application (virtual machine) among various physical hosts without shutting down the VM, in order to achieve load balancing and less power consumption. Two sorts of live migration methodologies (pre-copy & post-copy) are adapted by hypervisor.

Post-copy Methodology:

In the post copy approach firstly the vm is suspended from the source host and minimum info of execution state of vm(memory, CPU, I/O files) is transferred to destination host. VM is now active on destination host. Simultaneously, source host transfessers enduring memory pages to target host. Page fault encountered at the target, if VM is accessing a page that is not available in memory.

Pre-copy Approach:

The pre-copy migration consists of following two stages:

1) Warm-up phase
2) Stop-copy phase

Warm-up phase:
The Virtual machine manager duplicates all the memory pages repeatedly from source to target while the VM is being executing at source.
If pages modified throughout memory duplicate procedure called messy (dirty) pages, messy pages will be re-replicated until the page dirtying rate is grater then rate of re-replicated pages.

Stop and Copy phase:
Following warm-up phase, the VM will be halted for hotspot and the remaining filthy pages will make duplicated at the target host, furthermore VM will now continued in target host.

The execution of live migration procedure relies upon two execution measurements (Downtime and Migration time). Migration time is when VM is executing on source host to the time when VM resumed at destination. Downtime is when VM will be halted at source and its execution states are transferred to destination host
Total migration time= Migration time + Downtime.

III. RELATED WORK

Different studies identified with the issue of live virtual machine migration. The fundamental reason for live migration is to relocate the running virtual machine from the hotspot host to target host, without closing the customer or host, so running application will be halted for immaterial time period. To resolve the issue of least downtime and migration time, many calculations and algorithms have been presented by analysts. However, a number of those algo's are not appropriate to lessen downtime and movement time in high messy page environment.
In this section, we will have a look over some existing migration algorithms which were introduced in order to provide minimum downtime and migration time in high messy page environment. Some of them are:

A) Standard Pre-Copy Based Approach:

This technique [2] identifies two bitmaps to_send (identify pages modified in last iteration) and to_skip (identify pages modified in current iteration) to recognize the high messy pages. If any page modified in previous iteration to_send is set as 1in iteration i, otherwise 0. If any page modified in current iteration to_skip is set as 1in iteration i, otherwise 0. For page (p) whenever to_send=0 and to_skip=1 or to_send=1 and to_skip=1, p will identified as high messy page and will be send in last round and it will transmit large number of dirty pages iteratively.
Disadvantage*:* Standard pre-copy methodology depends upon two bitmap to distinguish often altered pages, so likelihood of discovering high filthy page is low.

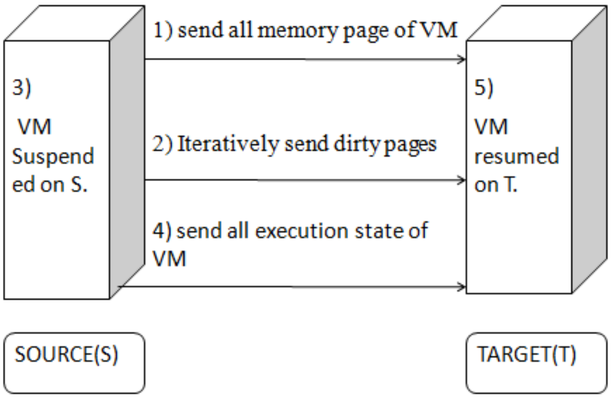
1) send all memory page of VM
2) Iteratively send dirty pages
4) send all execution state of VM
3) VM Suspend ed on S.
5) VM resumed on T.
SOURCE(S)
TARGET(T)

Fig.1: Pre-copy based migration.[4]

B) Time Series Based Approach:

This methodology [2] extends the standard pre-copy methodology. Time series approach records the high filthy pages by their chronological statics of to_send (to_send_h), which evade frequent re-transmission of memory pages to destination host in current round. Two key parameters K (threshold of high messy page) and N (upper limit of to_send_h[]). On the off chance that any page altered in any iteration i, then value of to_send_h will become 1 for specific page, otherwise 0. In every iteration i, time series array is checked and page is considered as high messy page when it satisfies following equation (1)[3]:

$$\sum_{i=1}^{N}(p€\ to\_send\_h[i]) >= k\ldots\ldots\ldots(1)$$

If threshold K is lesser then page modification rate, then page will identified as high filthy page and will be migrated at last round.
Disadvantage: Time series based approach depends upon two parameters K and N. If appropriate ration of K/N will not be chosen then time series approach will not give better performance in high dirty page environment.

C) Enhanced Time Series Approach With Two Phase Strategy:

Enhanced time series approach with two phase strategy [4] extends time series approach and eliminate the problem of taking threshold K.
Enhanced time series approach = Time series + Two phase approach.
In the first phase it checks the value of to_send and to_skip array. For page (p) if to_send=1 and to_skip=0, it checks array to_send_h[N] and following equation[2].

$$\sum_{i=1}^{N} Count\ One(to\_send\_h[i]) \geq \sum_{i=1}^{N} CountZero(to\_send\_h[i]$$

Equation (1)

When page fulfills above condition will be identified as high messy page and sent in last round, otherwise give second chance to page.

When modified time series approach is completed, selected pages will passed through second chance strategy. If it is unmodified for 2 successive rounds will sent to target host. Following figure 2 explains this approach

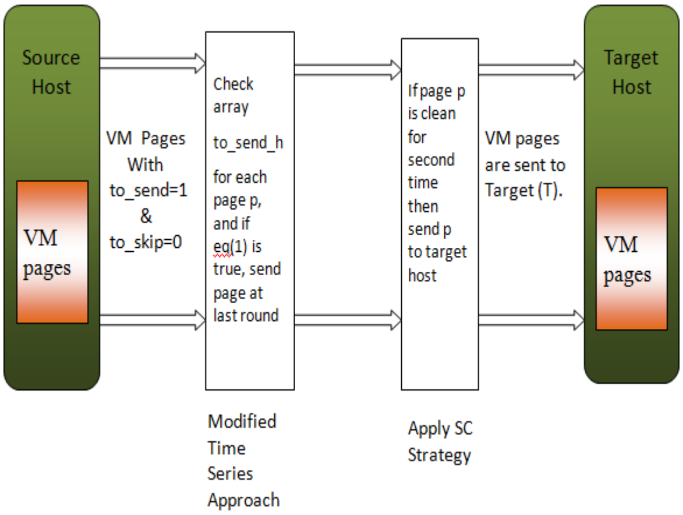
Source Host
VM pages
VM Pages With to_send=1 & to_skip=0
Check array to_send_h for each page p, and if eq(1) is true, send page at last round
If page p is clean for second time then send p to target host
VM pages are sent to Target (T).
Target Host
VM pages
Modified Time Series Approach
Apply SC Strategy

Fig.2: Modified time series approach

Disadvantage: This approach only gives second chance for which value of to_send=1 and to_skip=0, but not providing any chance to which to_send=0 and to_skip=0. Due to this pages having value to_send=0 and to_skip=0 will be sent in last round and migration time will increase.

IV. PROPOSED WORK

Optimized Two Phase strategy Approach:

Optimized two phase strategy eliminates the problem of enhanced time series approach. Bitmap to_send (record pages updated at earlier round), to_skip (records the pages updated in current round), to_send_h (contains record of history of page in last N iteration) and N (size of array to_send_h) are used. For page (p) if updates are made in any cycle i then the estimation of to_send_h is set as 1 for that specific page in cycle i, otherwise 0. The issue with modified time series approach is it only gives second chance to the pages containing values to_send=1 and to_skip=0, due to which the pages those were not modified in previous iteration and in the current iteration will wait and sent in the last round, which increases the migration time. Our approach also gives second chance to the pages having values to_send=0 and to_skip=0.

$$\sum_{i=1}^{N} Count\ One(to\_send\_h[i]) \geq \sum_{i=1}^{N} CountZero(to\_send\_h[i]$$

Equation (1)

Cont One counts number of ones in to_send_h and count zero counts number of zero's in to_send_h[].

Pages having values to_send=1 and to_skip=0 or to_send=0 and to_skip=0, we will check the array to_send_h[N]. If eq.(1) will be satisfied by the page, page will be high messy page and will migrate in last iteration, otherwise give second chance to the page.

Second Chance to pages:

When the enhanced time series approach is completed, and equation is not satisfied by the pages, for which to_send= 1 and to_skip=0 or to_send=0 and to_skip=0, then give second chance to page. If pages will be clean for two consecutive rounds (For both round value of pages will to_send=0 and to_skip=0) after that page will be sent to the target host.

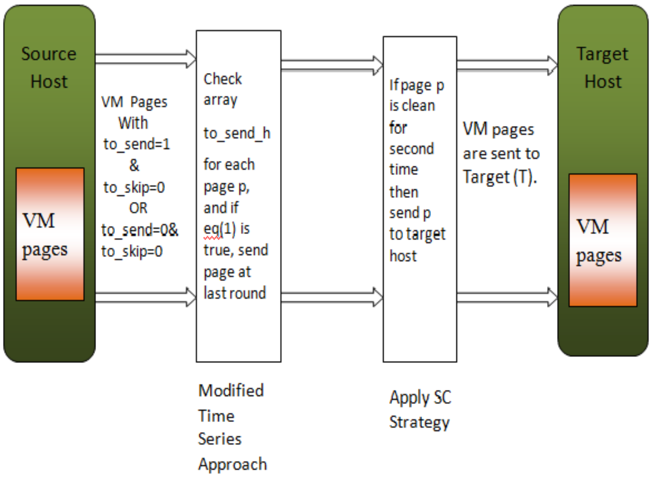
Source Host
VM pages
VM Pages With to_send=1 & to_skip=0 OR to_send=0& to_skip=0
Check array to_send_h for each page p, and if eq(1) is true, send page at last round
If page p is clean for second time then send p to target host
VM pages are sent to Target (T).
Target Host
VM pages
Modified Time Series Approach
Apply SC Strategy

Fig.3: Modified two phase strategy approach

In modified two phase strategy we have considered the advantages of time series and two phase strategy approach. We give second chance to the page for which to_send=0&to_skip=0, if page will be clean for next two rounds then page will transferred in current iteration, resulting reduced migration time.

Our approach is described by the following algorithm.

```
Algorithm:

Input:
N: size of array to_send_h
to_send_h: time series array
to_send: dirty bitmap of previous iteration
to_skip: dirty bitmap of current iteration

  Send all VM's memory pages in first time.
to_send ← dirty pages;
to_skip ←      NULL;
i←0;
    While (True) {
        Get dirty bitmap of Virtual Machine (VM);

            For each page p {
                IF   (to_send= =0 & to_skip= =1)   OR
                     ( to_send= =1 & to_skip= =1)  then
                          Continue;

Else IF  (to_send==1&to_skip==0)   OR

                     (to_send= =0 & to_skip= =0 ) THEN
           IF equation (1) is true THEN
                          Continue;
ELSE
Give a second chance (SC) to page p;
IF page  is kept clean for 2 consecutive round
THEN

Send page to target host.

ELSE
    Continue;  }

to_send_h[i] ← to_send;
i ← (i+1)%N;
to_send ← to_skip;
Update to_skip;

IF (last iteration & to_send= =1) THEN
send page p to destination;
break;
}
```

Fig.4: Optimized Enhanced Time Series Based Algorithm

The pseudo code and flowchart of optimized time series based two phase strategy approach for live virtual machine migration is represented in Fig.4 and in Fig.5.

Flowchart:

Terms Used in Flowchart:

N: range of to_send_h[];
to_send_h: time sequence array contains history of pages (p);
to_send: messy page of earlier round;
to_skip: messy pages of present round;
to_send==0&to_skip==0: non dirty page;
to_send==0&to_skip==1: page p wasn’t messy in earlier round but is dirty in present iteration;
to_send==1&to_skip==1: page p was dirty in previous iteration and is also messy in present iteration;
to_send==1&to_skip==0: page p was dirty in previous iteration but is dirty in current iteration.

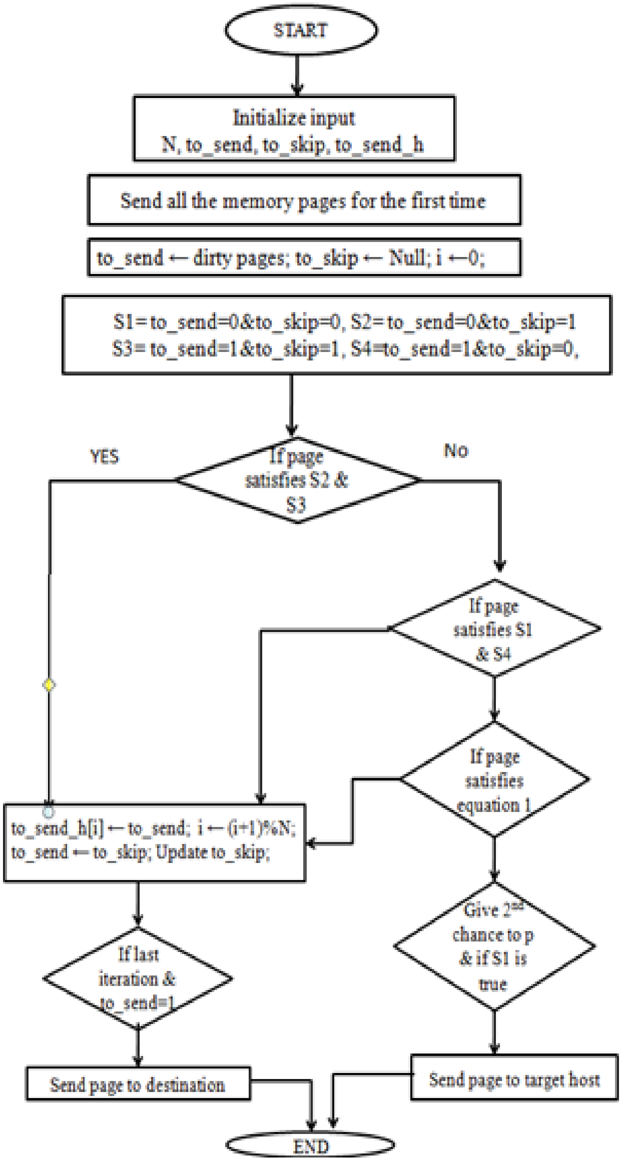
START
Initialize input
N, to_send, to_skip, to_send_h
Send all the memory pages for the first time
to_send ← dirty pages; to_skip ← Null; i ←0;
S1= to_send=0&to_skip=0, S2= to_send=0&to_skip=1
S3= to_send=1&to_skip=1, S4=to_send=1&to_skip=0,
YES
If page satisfies S2 & S3
No
If page satisfies S1 & S4
If page satisfies equation 1
to_send_h[i] ← to_send; i ← (i+1)%N;
to_send ← to_skip; Update to_skip;
If last iteration & to_send=1
Give 2nd chance to p & if S1 is true
Send page to destination
Send page to target host
END

Fig.5: Flowchart Of Optimized Enhanced Time Series Based Algorithm

Performance Matrices to Analyze Algorithm:

1) Down time
2) Migration time
3) Total Migration time

Downtime is when VM will be halted at source and its execution states and modified pages are migrated to destination host. VM now resumed at destination host.

Migration time is when VM is executing on source host yet pages are relocating to destination host to the time when VM resumed at destination.

Total migration time is the time taken to perform downtime and iterations.

Our proposed algorithm also gives the second chance to the pages having values to_send==0&to_skip==0. If page satisfies equation 1 and remains unmodified for 2 successive rounds, will transmitted to destination host in current iteration instead of waiting for the last round. So the migration time will be decreased. Our approach is appropriate for minimum and maximum messy situation, effect on various migration parameters can be easily analyzed.

Pages transferred in iterations: It relies on upon the transmission times of messy pages amid the live movement of VM. Only the page for which value of to_send=0&to_skip=0 OR to_send=1&to_skip=0 and doesn't satisfies eq(1), will be transmitted to destination host, which reduces the quantity of pages moved in different cycles.

Migration time: Page satisfies all three conditions and will kept clean for two consecutive rounds, will sent to target, so less number of pages is transferred and time taken by pre-copy iterations will also be reduced.

Downtime: Less number of pages (non-dirty pages) is transferred and rest of pages waits for last round to be sent.

V. RESULTS AND DISCUSSION

Analyze of Migration time and Downtime:
To evaluate and compare Optimized time series based two phase strategy with existing enhanced time series based two phase strategy, the simulation is done using Java with netbeans 7.2 Technology.

Migration time = Time at which VM is activated on target host – time at which VM migration initiated on source.

Table 1 demonstrates migration time calculated by our approach and enhanced time series based two phase strategy approach. Our proposed algorithm produces same Migration time or less rather than other. Since only the pages satisfying all three conditions will transmitted to destination host, so less number of pages will be transferred in various rounds, hence time taken by pre-copy iteration will also reduced, which results decreased migration time.

Patterns	Enhanced time series based two phase strategy approach(seconds)	Optimized time series based two phase strategy approach(seconds)
1	114	107
2	243	239
3	285	279
4	157	150
5	349	341
6	280	267
7	216	210
8	345	338
9	113	105
10	196	192

Table 1: Migration time calculated by time series based two phase approach and our approach

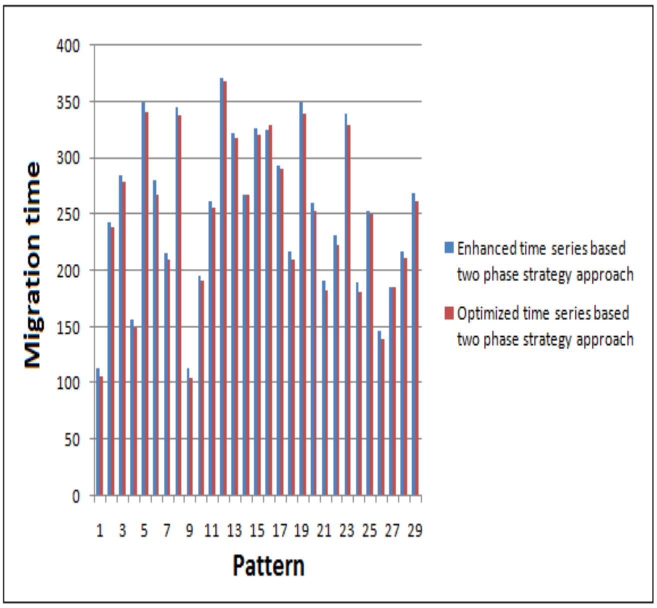

Fig.6: Analysis of Migration time for enhanced time series based two phase strategy approach and our approach

Table 2 demonstrates downtime calculated by our approach and enhanced time series based two phase strategy approach. Downtime for enhanced time series approach and approach is illustrated by fig.7.

Patterns	Enhanced time series based two phase strategy approach	Optimized time series based two phase strategy approach
1	97	88
2	55	52
3	113	110
4	54	45
5	79	70
6	90	81
7	58	51
8	57	48
9	59	49
10	88	85

Table 2: Downtime calculated by enhanced time series based two phase approach and our approach

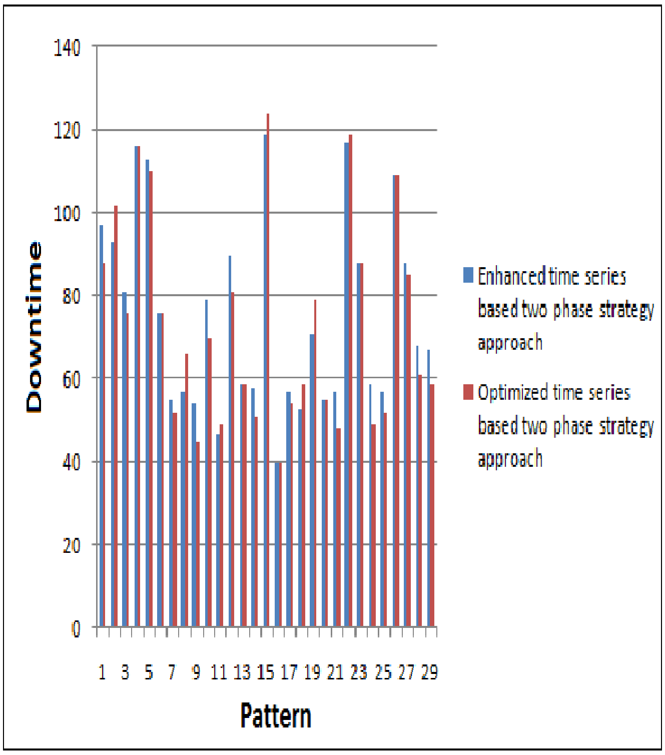

Fig.7: Analysis of Downtime for enhanced time series approach and our approach

VI. CONCLUSION AND FUTURE WORK

Enhanced time series based two phase strategy approach combines the time series based pre-copy method and two phase strategy to get features of both algorithm and to remove the need to take appropriate combination ratio of K and N. In our approach some modifications are done in enhanced time series based two phase approach to extend the performance of live virtual machine migration using pre-copy approach. Our proposed approach provides second chance to the pages for which value of to_send=0&to_skip=0 OR to_send=1&to_skip=0, if page remains unmodified in 2 successive rounds then page will be transmitted in current iteration instead of waiting last round, otherwise will be sent in last iteration. Hence fewer pages transferred and resulting decreased migration time and downtime.

The further enhancement of our proposed work can be done in order to achieve decreased downtime as compared to enhanced time series approach.

REFERENCES

[1] LIU Peng-cheng. "Cloud Computing Dynamic Migration of Virtual Machine Research[D]". Shanghai: Fudan University, 2009.

[2] Anju Mohan, Shine S . "Survey on Live VM Migration Techniques", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 1, January 2013

[3] J.Arputharaj Johnson Department of Computer Science and Engineering Anna University, "OPTIMIZATION OF MIGRATION DOWNTIME OF VIRTUAL MACHINES IN CLOUD",IEEE - 31661,4th ICCCNT 2013 .

[4] Neha Agrawal1 & R.K. Pateriya2. "Enhanced time series based pre-copy method for live migration of virtual machine",2013

[5] Pradip D. Patel elect. "Live Virtual Machine Migration Techniques in Cloud Computing", International Journal of Computer Applications (0975 – 8887), January 2014

[6] LIU Peng-cheng. "Cloud Computing Dynamic Migration of Virtual Machine Research[D]". Shanghai: Fudan University, 2009.

[7] Bolin Hu , Zhou Lei, Yu Lei, Dong Xu, Jaindum Li, " A Time Series Based Approach for Live Migration of Virtual Machines," IEEE 17th International Conference on Parallel and Distributed System, 2011,pp.947-952.

[8] Cho-Chin Lin, Yu-Chi Huang and Zong-Dejian, "A Two Phase Iterative Pre-copy Strategy for Live Migration of Virtual Machines," in Proc ICCM, Taiwan ,2012 ,PP-29-34.

[9] W.Liu and Tao Fan elect, the Live Migration of Virtual Machine Based on Recovering System and CPU Scheduling," J. Network and Computer Application, vol.34, Issue 4, 2011,pp.1088-1096.

[10] Yanqing Ma, Hongbo Wang, Jiankang Dong, Yangyang li,, Shiduan Cheng,"ME2: "Efficient live migration of virtual machine with memory exploration and encoding," IEEE international conference on cluster computing,2012,pp.610-613.

[11] F. Ma, F. Liu, and Z. Liu, ― "Live Virtual Machine Migration Based on Improved Pre-copy Approach",‖ Proc. Software Engineering and Service Sciences, 2010, pp. 230-233.

[12] Z. B. Liu, W. Y. Qu, T. Yan, and H. T. Li, ― "Hierarchical Copy Algorithm for Xen Live Migration",‖ Proc. The 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2010, pp. 361-364.

[13] LIU Peng-cheng. "Cloud Computing Dynamic Migration of Virtual Machine" Research[D]. Shanghai: Fudan University, 2009.

[14] Wang Peng. "Into the Cloud Computing" [M]. Beijing: People's Posts and Telecommunications Press, 2009-06