DOI: http://dx.doi.org/10.26483/ijarcs.v8i8.4627

ISSN No. 0976-5697

Volume 8, No. 8, September-October 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# A HYBRID APPROACH FOR REDUCING DOWNTIME, MIGRATION TIME AND ENERGY CONSUMPTION OF LIVE VM MIGRATION.

Ritika Dhuria
Dept.of Computer Engg &Technology
Guru Nanak Dev University
Amritsar, Punjab

Kiranbir Kaur
Dept.of Computer Engg &Technology
Guru Nanak Dev University
Amritsar, Punjab

***Abstract:*** In this paper we discuss a hybrid approach of live VM migration that evaluates the hosts across the LAN. Here the best of both Pre and post copy methods are to be evaluated using the hybrid approach. Later the pre copy approach used, spawn the VM on destination host after that CPU state and memory is transferred and after transferring processor state the opposite work has to be done. In our approach we used checkpoint mechanism for recovery of page faults occurs over the network while transferring the data over the VM. We propose a method that uses hybrid approach of live VM migration with addition a new data recovery mechanism. This recovery mechanism utilizes checkpoint and also in this the load balancing on individual machine has been done. In results 20% to 30% improvement has been seen and also it is better implementation over pre and post copy mechanism.

***Keywords:*** Hybrid approach, pre copy, CPU state, checkpoint, post copy, load balance**.**

## I. INTRODUCTION

With the advent of virtualization, it gives an advantage to companies about computing and storage infrastructure over the cloud. In case of large scale deployment the efficient utilization of resources has been a major concern so we utilize the live migration as an impressive technology. The live migration technology is impressive for load balancing and optimizing VM deployment in a data center across physical nodes.[1]
The VM is useful following ways:

1. It can be migrated to new node if current physical node is failing
2. For resource utilization it can be migrated to other nodes from idle node
3. The load balancing can be done on various physical nodes.

Here in this paper we describe about the hybrid approach to live migration of virtual machines that will introduces recovery mechanism during the migration process and also balances load various VM machines.[2]
In recent years, the growth of IT infrastructure has triggered the demand for computational power and has led to the creation of huge data centers and has increased the energy demand. A solution to this problem is cloud computing. Cloud Computing is among the most trending technologies on the Internet which fulfills the need of computationally intensive demands of users. Cloud Computing offers access to shared pool of computing resources which includes storage space, computation power, network, applications and services on demand basis to the users over the internet. Cloud Computing introduces the concept of Everything as a Service, mostly referred as XaaS where X is Software, Infrastructure, Hardware, Platform, Data, Business etc. The user no longer need to worry about the initial investments on the resources since cloud computing provides an approach for leveraging computing resources with same ease as utilizing common utilities such as natural gas, water, electricity supply on pay per use bases through the concept of utility oriented computing thus ensuring Quality of Service at the same time. Due to the rising service demands of the users the cloud infrastructure is increasing day by day [3]. Cloud computing has exploited virtualization technology to provide on demand provisioning of resources in order to satisfy the cloud clients.[4], [5] A data centers under cloud infrastructure comprises of thousands of physical nodes and single physical node consists of multiple virtual machine instances each having its own operating system and work isolated from each other thus the complexity of cloud infrastructure is increasing due to which faults are inevitable.

The live VM migration process can be classified into two steps:

- Control can be switched to the destination
- Data transfer to the destination.

The two mechanisms that are most commonly used in live migration process are given below:

- Pre Copy mechanism
- Post Copy mechanism[3]

[6], [7]In pre copy the transfer of memory has to be done firstly and then transfers the execution. In post copy the execution transfer is done firstly and then the data transfer through memory are done.

## 2 STUDY OF LITERATURE

Energy efficiency is critical while resources are allocated to VMs. Work has been done towards this aspect. This section describes the techniques used to achieve energy efficiency by reducing load using computational offloading mechanism.
[8]propose energy efficient mobile cloud computing using wireless energy transfer. The technique combines mobile

cloud computing and microwave power transfer technique. Using this technique it is possible to perform computation in wearable devices. Set of policies are formulated for controlling CPU cycles in case of local computing and offloading for other mode of computing.

[9]suggests energy constraint mechanism to ensure job execution efficiently. Code migration is suggested to optimize energy efficiency. Pre-copy with remote execution takes place. With remote execution, job executes from the remote server. In case of deterioration, job is migrated through code and hence progress of job is saved and it is executed again from the place it is stopped on previous machine. Results show considerable improvement in terms of downtime and migration time.

[10]researched a task computing and cost of file offloading to minimize energy consumption. Radio resource allocation is primarily considered in this literature. Energy efficient computational offloading (EECO) on 5G network is proposed in this paper. Uplink and Downlink transmission rate is considered through the following equations.

Uplink Transmission Rate

$$r_{i,k} = W \log_2\left(1 + \frac{p_i^M g_i^M}{I^S + \sigma^2}\right)$$

Equation 1: Uplink Transmission Rate

Where ‘P’ is the power of mobile device, ‘I’ denotes the interference, ‘g’ indicate the channel gain, ‘σ’ is the noise.

Downlink Transmission Rate

$$r_{i,k} = W \log_2\left(1 + \frac{p_i^M g_i^M}{I^M + \sigma^2}\right)$$

Equation 2: Downlink Transmission Rate

Channel for accessing used is M. Cost under the delay constraint is reduced considerably.

[11] Proposes a decentralized approach for mobile computational offloading. Decentralized approach follows multiple virtual machines on which load is distributed. The computation is considerably reduced on individual machine. The energy efficiency is achieved since priority while allocation is considered. Results indicate improved performance.

[12] Proposes duty cycling mechanism to achieve energy efficiency in scheduling of resources in wireless sensor network. Duty cycling is divided into power management and topology control mechanisms. Node redundancy is considered in topology control and power management is considered in case of sensor allocation. Sensors have limited power and energy associated with them. This work effectively manages both energy and power and hence a result obtained is better in terms of energy efficiency. Minimum load a node can tackle is given through the following equation.

$$\gamma\big(L(G)\big) \leq K_n(G) \leq K_E(G) \leq \min\left(\deg(G)\right)$$

Equation 3: Load equation for nodes

‘G’ indicates the graph of the form G= {V, E}, ‘V’ is the set of vertex and ‘E’ is the set of edges. ‘n’ indicates total number of nodes.

[13]consider both dynamic power as well as leakage power for energy efficiency during scheduling. Precedence constraint is employed in this case. Jobs hence are executed in terms of precedence rather than sequential. The execution time is calculated in terms of following equations.

$$C_m = (1 - \beta) * CB_m + \beta * CW_m$$

Equation 4: Execution time calculation.

Jobs are executed on 1, 2 and 4 cores for checking the power consumption. Results show better scheduling as compared to other scheduling mechanisms. [14]

Next section describes the detail proposed methodology which enhance Green computing by lowering the $CO_2$ level and reduces energy consumption also.

## 3 PROPOSED METHODOLOGY

In proposed methodology we combine pre copy and post copy approach along with hybridization introducing checkpoint mechanism.

### PRE COPY

[15], [16]In pre copy approach the transfer from memory to the destination are to be done first and then limit the iteration reaches.

**Algorithm**

1. In destination node the memory and VCPUs are restrained first
2. A scan on page writes is initiated and all contents from source RAM are transferred to destination when relocation is issued.
3. In next step until iteration limit is reached the pages have been transferred.
4. When all transfer has to be done then the source is stopped and current state of CPU registers. After that state of virtual device and last memory pages are transferred to destination.
5. At destination the VM is resumed. .

The number of remaining pages to be copied for a given point in time t is then determined by

$$f(t) = e(t) + p(t) + h(t)$$

### POST COPY

[17], [18]In post copy the transfer of device state and VCPU is transferred first on destination and then the execution on destination starts. The steps which are performed as given below

1. The VM at source are stopped
2. The states of devices are copied and VCPU registers on the destination VM.
3. Execution at destination are resumed
4. If not yet fetched page is accessed by VM then Page Fault occurs and page is transferred to the destination.

The mathematical formula for calculating

Source Contention=

$$\sum_{i=1}^{n} \text{Rate of outgoing traffic in Mbps for VM , if migrated with pre copy} + \text{outgoing background traffic}$$

Destination Contention=

$$\sum_{i=1}^{n} \text{Rate of incoming traffic in Mbps for VM , if migrated with post copy} + \text{Incoming background traffic}$$

## CHECKPOINTING

[19], [20]The checkpoint is a mechanism that is used to back up the data before the updates are done on VM in live migration. The administrator can return the virtual machine to its state prior to the update. The action that will used to return the state to checkpoint is recovery action. Each virtual hard disk that is attached to each virtual machine uses checkpoint for each to save the state. The recover action is utilized after the creation of checkpoint to restore the virtual machine.

The logs are maintained in real time environment till all the memory spaces fill out. In the checkpoint mechanism all the previous logs are removed from the system and stored in storage disk permanently.

## HYBRID APPROACH

In this we combine pre copy and post copy approach along with checkpoint. The hybridization of pre and post copy is followed by using the condition. If pre copy generates better result than it is followed otherwise post copy is followed.

ALGORITHM

```
1. Firstly the result of pre and post copy are analyzed and then VM chooses which mechanism is followed
2. If result produced by pre copy approach is effective than it is used for transfer otherwise
3. Post copy approach has to be followed.
4. After that checkpoint mechanism is applied for recovery.
```

## 4. PERFORMANCE ANALYSIS AND RESULTS

Performance monitoring is done to prove the worth of study. The parameters used for analysis are load, downtime and migration time. Load is estimated by observing the total load disbursed over the data centers.

$$load = \sum \frac{Burst_{time}}{n}$$

N is total VMs present within the datacenter and used for allocation. The results obtained are as under

Table 1: Showing Load Distribution among 10 VMs with and without checkpoint approach.

| *VM* | *Load(existing work)* | *Load (proposed work)* |
|---|---|---|
| 1 | 900 | 850 |
| 2 | 850 | 801 |
| 3 | 870 | 852 |
| 4 | 920 | 911 |
| 5 | 780 | 763 |
| 6 | 830 | 822 |
| 7 | 810 | 799 |
| 8 | 930 | 923 |
| 9 | 840 | 833 |
| 10 | 800 | 788 |

The plots corresponding to load distribution is given as follows

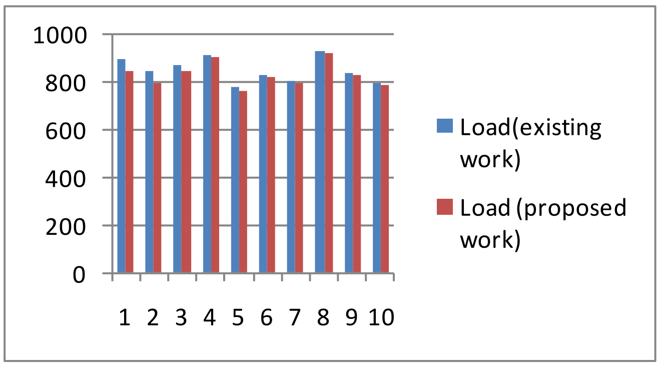

Figure 1: Showing Load on individual Machine

The downtime and migration time are also better in hybrid approach with checkpoint. Downtime indicates idle time of VMs. Average idle time is calculated using following equation

$$Downtime = \sum \frac{Idle_i}{n}$$

'n' is total number of VMs in the system. Idle$_i$ is the idle time of VMs.

Table 2: Downtime of various VMs

| *VM* | *Downtime* |
|---|---|
| *1* | *15.0* |
| *2* | *3.0* |
| *3* | *10.0* |
| *4* | *15.0* |
| *5* | *20.0* |
| *6* | *13.0* |
| *7* | *10.0* |
| *8* | *5.0* |
| *9* | *4.0* |
| *10* | *5.0* |

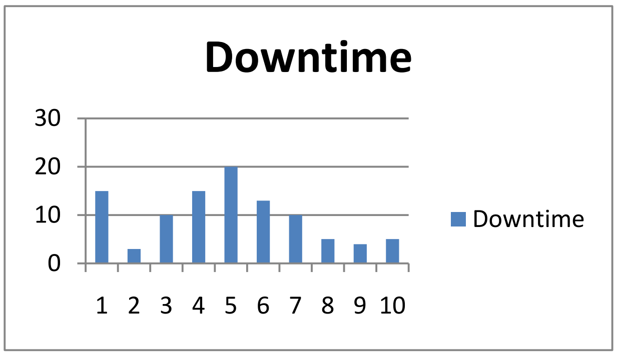

Figure 2: Downtime of VMs

Migration time is also observed calculated using the formula.

$$Migration - time = \sum \frac{Load_{Th_i}}{n}$$

Load_th$_i$ is threshold load which VM can handle. As load threshold expires, load is shifted to next machine. Time taken to shift the load is known as migration time. Migration time is also observed to be optimal as given through table 3

Table 3: Migration time associated with checkpoint approach

| *VM* | *Migration time* |
|---|---|
| *1* | *0.30000000000004* |
| *2* | *0.3* |
| *3* | *0.2* |
| *4* | *0.20000000000003* |
| *5* | *0.4* |
| *6* | *0.30000000000005* |
| *7* | *0.1* |
| *8* | *0.2* |
| *9* | *0.1* |
| *10* | *0.5* |

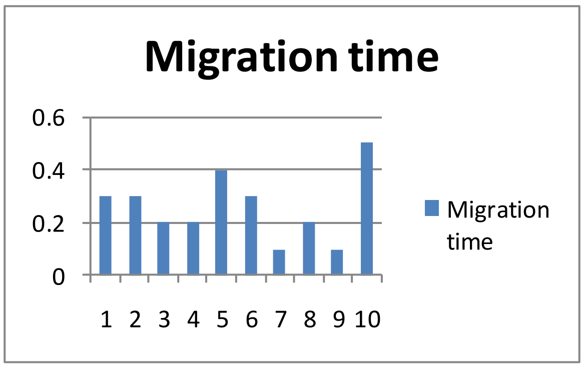

Figure 3: Migration time observed through Checkpoint approach

The comparison of existing approach without checkpoint with hybrid approach is given in terms of downtime and migration time as

TABLE 4: PERFORMANCE COMPARISON OF Hybrid approach without and with checkpoints.

| ***NUMBER OF VM*** | ***DOWNTIME*** | | ***MIGRATION TIME*** | |
|---|---|---|---|---|
| | ***EXISTING*** | ***PROPOSED*** | ***EXISTING*** | ***PROPOSED*** |
| ***10*** | ***19*** | ***9*** | ***1.23*** | ***0.3*** |
| ***20*** | ***25*** | ***11*** | ***5.23*** | ***0.6*** |
| ***30*** | ***29*** | ***13*** | ***6.32*** | ***0.9*** |
| ***40*** | ***32*** | ***17*** | ***8.98*** | ***1.2*** |
| ***50*** | ***35*** | ***19*** | ***9.32*** | ***1.5*** |

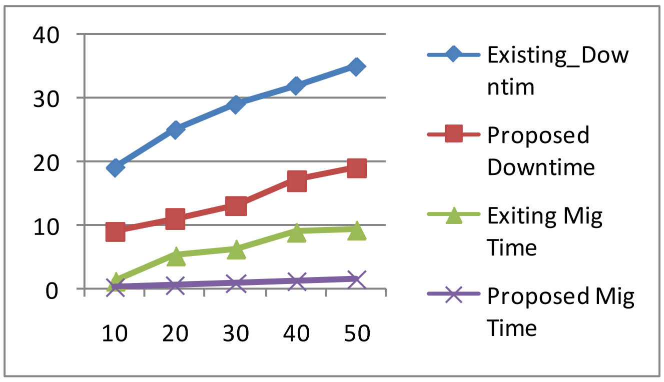

Figure 4: comparison in terms of number of VMS, Migration and Downtime

Comparison indicates Hybrid approach with checkpoint is better and result is improved by 20 to 30%.

## 5. CONCLUSION

In this paper we present a best way to recover and save the state of VM machine during the migration process by using checkpoint mechanism. In pre copy approach read intensive workload is well proven but not in case of write intensive. In this page faults are large and it won't work in worst case. The post copy approach only transfer the CPU register an virtual devices state so it has least downtime but intensive workload will degrade its performance.

## 6. REFERENCES

[1] L. Hu, J. Zhao, G. Xu, Y. Ding, and J. Chu (2013) 'HMDC□: Live Virtual Machine Migration Based on Hybrid Memory Copy and Delta Compression', vol. 646, no. 2, pp. 639–646.

[2] S. Sahni(2012), 'A Hybrid Approach To Live Migration Of Virtual Machines' .

[3] D. Gupta, S. Jain, and S. Goel (2016), 'Performance Study of Live Virtual Machine Migration using KVM Hypervisor', vol. 6, no. 6, pp. 195–207.

[4] C. Wang, Q. Wang, K. Ren, and W. J. Lou (2009), 'Ensuring Data Storage Security in Cloud Computing', *Iwqos 2009 Ieee 17th Int. Work. Qual. Serv.*, pp. 37–45\n302.

[5] V. Yadav, P. Malik, A. Kumar, and G. Sahoo (2015), 'Energy Efficient Data Center in Cloud Computing', *2015 IEEE Int. Conf. Cloud Comput. Emerg. Mark.*, pp. 59–67.

[6] F. Ma, F. Liu, and Z. Liu (2010), 'Live Virtual Machine Migration based on Improved Pre-copy Approach', pp. 230–233.

[7] Y. Zhong, J. Xu, Q. Li, H. Zhang, and F. Liu (2014), 'Memory State Transfer Optimization for Pre-copy based Live VM Migration', pp. 290–293.

[8] C. You, K. Huang, and H. Chae (2016), 'Energy Efficient Mobile Cloud Computing Powered by Wireless Energy Transfer', vol. 8716, no. c, pp. 1–14.

[9] D. Marculescu, N. H. Zamora, P. Stanley-marbell, and R. Marculescu (2003),'Fault-Tolerant Techniques for Ambient Intelligent Distributed Systems-', pp. 348–355.

[10] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, and X. Peng (2016), 'Energy-efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks', vol. 3536, no. c, pp. 1–10.

[11] X. Chen (2015), 'Decentralized Computation Offloading Game for Mobile Cloud Computing', *IEEE Trans.*

*Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983.

[12] L. Aslanyan, H. Aslanyan, and H. Khosravi (2013), 'Optimal node scheduling for integrated connected-coverage in wireless sensor networks', *CSIT 2013 - 9th Int. Conf. Comput. Sci. Inf. Technol. Revis. Sel. Pap.*.

[13] I. Transactions, O. N. C. Design, and O. F. Integrated (2011),'Reliability-Driven Energy-Efficient Task Scheduling for Multiprocessor Real-Time Systems', vol. 30, no. 10, pp. 1569–1573.

[14] D. Miraculine (2016), 'Efficient Data Transmission during Virtual Machine Failures Using Hybrid Copy Live Migration', pp. 119–126.

[15] D. Kapil, E. S. Pilli, and R. C. Joshi (2013), 'Live virtual machine migration techniques: Survey and research challenges',in *3rd IEEE International Advance Computing Conference (IACC)*, 2013, pp. 963–969.

[16] G. J. Jeincy (2016), 'Space Prediction for Cloud Computing', 2016.

[17] W. Zhang, K. T. Lam, and C. L. Wang(2014),'Adaptive Live VM Migration over a WAN: Modeling and Implementation', in *2014 IEEE 7th International Conference on Cloud Computing*, 2014, pp. 368–375.

[18] P. Kaur and A. Rani (2015), 'Virtual Machine Migration in Cloud Computing',vol. 8, no. 5, pp. 337–342.

[19] J. Devale (1999), 'Checkpoint / Recovery Overview Checkpointing - Recovery', *Memory*.

[20] J. Hursey, J. M. Squyres, T. I. Mattox, and A. Lumsdaine, 'The Design and Implementation of Checkpoint / Restart Process Fault Tolerance for Open MPI ∗.