PREDICTION BASE VIRTUAL MACHINE MIGRATION IN CLOUD COMPUTING

[1]Prachi Soni, [2]NimeshVaidhya, [3]Rakesh Shah

[1]Student, [2]Asst. Professor, [3]PG Coordinator
[1]Computer Engineering
[1]GMFE, Himmatnagar, India

Abstract : Virtual Machine (VM) migration is an important technology to support Infrastructure as a Service (IaaS). Traditional pre-copy and post-copy strategies could function well in LAN but will need considerable time to migrate between remote hosts in WAN. In this paper, we propose a prediction-based strategy to optimize cloud VM migration process over WAN. In this strategy, information about size increments of snapshots is used to determine appropriate time points for migration in order to reduce the downtime during migration. Specifically, we utilize Markov Chain Model to predict the future increasing speed of snapshots. And also use prediction based algorithm for predict the necessary of future migration machine in prior basis.

***IndexTerms*:** Virtual Machine, Prediction, Migration, Cloud Computing

I. INTRODUCTION

Cloud computing helps enterprises take advantage of resources provided by large cloud service vendors. Typically, enterprises need to expand their IT capabilities during workload peaks; meanwhile migrating a VM to a cloud is a cost-efficient choice. As a result, attention is being attracted to live VM migration.The entire process of VM migration can be divided into three stages: the pre-copy, the down time and the synchronization stage [1]. During the pre-copy stage, a VM keeps running while the modified data is transferred [2]. After that, the VM shuts down and synchronizes the latest data [3]. In post migration, the VM resumes on the destination host before all the modified data is transferred [4]. So data on both sides should be synchronized. The durations of these three periods are important metrics and most of the migration strategies are designed for optimizing these metrics. There are three classic basic algorithms for VM migration, namely pure stop-copy, pre-copy and post-copy algorithm. Pure stop-copy algorithm is designed to shut down the VM and copy all its state to the destination host [5, 6, 7]. Although pure stop-copy algorithm can minimize the total migration time, it creates long down time. In order to reduce the down time, pre-copy algorithm is widely used. For example, Khaled Z. presents a pre-copy based algorithm on-line (OL) to provide minimal downtime [2]. Post-copy algorithm is another way to reduce the down time pre-copy algorithm is widely used. For example, Khaled Z. presents a pre-copy based algorithm on-line (OL) to provide minimal downtime[2]. Post-copy algorithm is another way to reduce the down time during VM Migrations work in the general Internet environment, and we can hardly transfer data through VPN. On contrary, the VM migration strategy we propose is suitable for the general Internet migration. Michael designs a post-copy based strategy using adaptive pre-paging across a Gigabit LAN [8]. Pre-copy algorithm and post-copy algorithm could reduce down time, but they both require a high bandwidth environment like LAN migration. While a VM running on a host, prediction base virtual machine migration work on a future forecast of a heavy traffic and then it migrate virtual machines from one data center to another data center.

II.VM MIGRATION TECHNIQUES:

A. Pre-Copy Memory Migration:
In pre-copy memory migration, the Hypervisor typically copies all the memory pages from source to destination while the VM is still running on the source. If some memory pages change (become 'dirty') during this process, they will continue re-copied until the rate of re-copied pages is not less than page dirtying rate.[1]

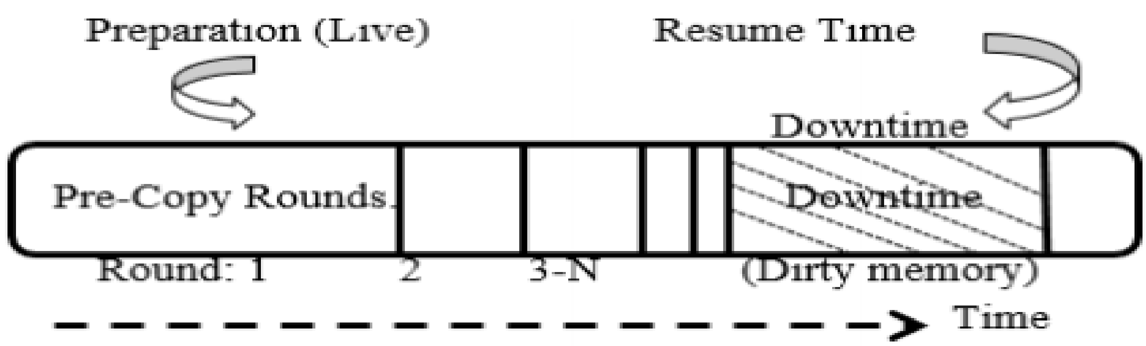

Fig 1. Pre-copy migration[1]

B. Post-copy memory migration:
Post-copy VM migration is initiated by suspending the VM at the source, by suspending the VM a minimal subset of the execution state of the VM (CPU registers and non-pageable memory) is transferred to the target. The VM is then resumed

at the target, even though most of the memory state of the VM still resides at the source. At the target, when the VM tries to access pages that have not yet been transferred, it generates pagefaults. These two faults are trapped at the target and redirected towards the source over the network. Such faults are referred to as network faults. The source host responds to the network-fault by sending the faulted page. Since each page fault of the running VM is redirected towards the source, this technique can degrade performance of applications running inside the VM. However, pure demand-paging accompanied with techniques such as pre-paging can reduce this impact by a great extent.[1]

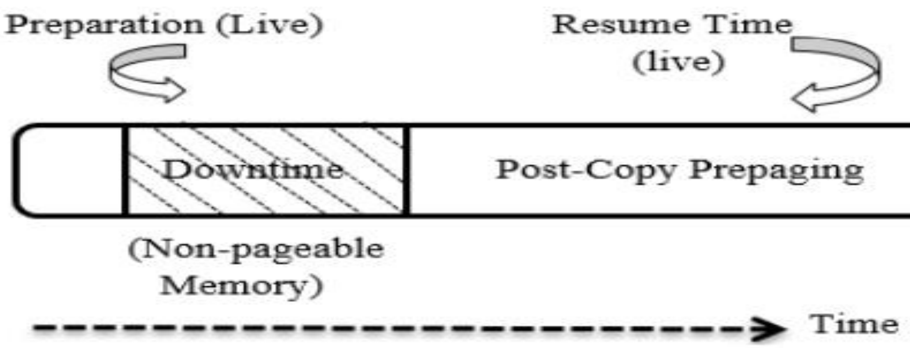

Fig 2. Post-copy migration [1]

III.VM MIGRATION GOALS:

A. Server Consolidation:

Server consolidation algorithms are required to decrease server sprawl in data centers. Such algorithms are in actual the VM packing heuristics attempt to pack whatever number VMs as would be prudent on a PM so that asset utilization is enhanced and unused or under-used machines could be turned off. This brings about reduced power consumption and in this way decreasing general operational expenses for data center administrators.[3]

B. Load balancing:

This decreases the imbalance of resource use levels over all the PMs in the group. This keeps a few machines from getting over-burden in the vicinity of lightly loaded machines with sufficient extra limit. Movement could be utilized to adjust the framework. The general system load could be adjusted by moving VMs from over-burden PMs to under-loaded PMs.[2]

C. Hotspot & Cold spot Mitigation:

Typically, a higher resource utilization esteem near greatest is situated as the upper limit and a low resource use worth is situated as the lower edge. PMs having resource utilization values past the upper limit are said to have shaped hotspots, and whose use values underneath the lower edge are said to have structured cool spots.[3]

IV. PROPOSED MECHANISM:

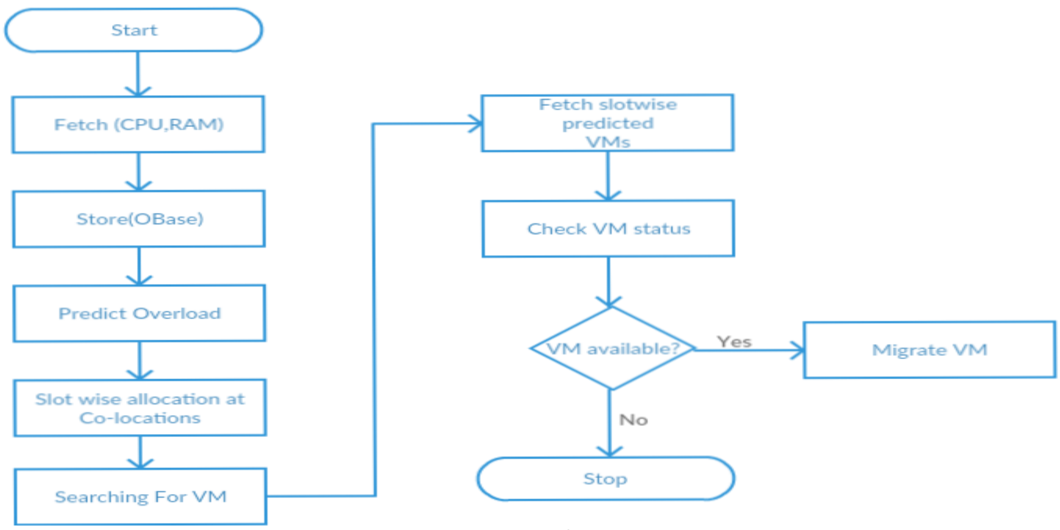

Fig 3. Flowchart of proposed work

- **Algorithm Steps:**

PM: Physical Machine

VM: Virtual Machine

Step 1:Repeat: Regular Interval

Step 2:For PMi to PMn

For VMi to VMn

Fetch(CPU,RAM)usage

Store (OBase)

End for

End for

Step 3: Fetch Slotwise predicted VMs

Step 4: Check for availability

If available

Then

Migrate Vms

Else

End

Step 5: End Repeat

Every time it find usage factor from OBase, Baased on the usage factor process pool of VM need Migration. Migration VM for underloaded VM.

V. SOFTWARE REQUIREMENTS:

- Amazon EC2
- Eclipse
- Amazon S3
- PyCharm
- Anaconda (3.0):

VI. IMPLEMENTATION RESULT:

- **SVC**

In SVC data points are mapped from data space to a high dimensional feature space using a kernel function. In the kernel's feature space the algorithm searches for the smallest sphere that encloses the image of the data using the Support Vector Domain Description algorithm. This sphere, when mapped back to data space, forms a set of contours which enclose the data points. Those contours are then interpreted as cluster boundaries, and points enclosed by each contour are associated by SVC to the same cluster.

This model gives 97% accuracy.

```
=====================================================================================
Accuracy: 0.9762140733399405
Precision: 0.951063829787234
Recall: 0.9977678571428571
Specificity: 0.9153846153846154
10-fold cross validation average accuracy: 0.968
SVC-RBF Kernel : >>> Final Accuracy in percentage 96.82709330535702 %

Process finished with exit code 0

6: TODO    Python Console    Terminal
```

Fig.4 SVC

- **Linear Regression:**

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable [18].

Logit Regression Results

Dep. Variable:	y	No. Observations:	4035
Model:	Logit	Df Residuals:	4028
Method:	MLE	Df Model:	6
Date:	Wed, 03 Apr 2019	Pseudo R-squ.:	0.09228
Time:	11:12:00	Log-Likelihood:	-2521.7
converged:	True	LL-Null:	-2778.1
		LLR p-value:	1.528e-107

	coef	std err	z	P>|z|	[0.025	0.975]
vCPU	-0.0502	0.010	-4.999	0.000	-0.070	-0.030
cpuLoad	0.0125	0.001	9.780	0.000	0.010	0.015
C_time_min	7.096e-05	0.000	0.152	0.879	-0.001	0.001
RAM	-0.0343	0.004	-8.753	0.000	-0.042	-0.027
ramLoad	0.0142	0.001	10.450	0.000	0.012	0.017
R_time	-0.0027	0.000	-5.799	0.000	-0.004	-0.002
applicationType	-0.3295	0.045	-7.271	0.000	-0.418	-0.241

Fig 5: Linear Regression

In our model we implemented linear regression model to predict the data and check the accuracy. Here is the result of the data set by applying model. Here the model is accurate by knowing its p value which is very low.

VII. CONCLUSION

For optimizing VM migration over WAN, we propose a prediction-based strategy which can forecast the increasing curve of snapshots about VMs. Our main contribution is to predict the increments of VM snapshot and select the proper segment to shut down the VM which minimizes the VM down time. Compared with two migration strategies, the evaluation shows that our PB strategy works well and stably during migration, which minimizes the down time among all the strategies. In the future, there are two parts of work we can focus on.

In Future we will try to make the dynamic migration over WAN, with low downtime or without interrupting the system. We will use prediction algorithm to predict overload before some time. We will migrate the virtual machine from one region to another.

VIII. ACKNOWLEDGEMENT

I forward my sincere thanks to Prof. Nimesh Vaidhya and Prof. Rakesh Shah for their valuable help during the report design of Research Skills. There suggestions were always there whenever I needed it. As supervisor they spent their valuable time for the in depth discussion on the topics. Also I forward my hearty thanks to other Faculty Members Department of Computer Engineering for their support.

REFERENCES

[1] Zhang, Fei, Guangming Liu, Xiaoming Fu, and Ramin Yahyapour. "A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues." *IEEE Communications Surveys & Tutorials* 20, no. 2 (2018): 1206-1243.

[2] Yang, Yaodong, Bo Mao, Hong Jiang, Yuekun Yang, Hao Luo, and Suzhen Wu. "SnapMig: Accelerating VM Live Storage Migration by Leveraging the Existing VM Snapshots in the Cloud." *IEEE Transactions on Parallel and Distributed Systems* 29, no. 6 (2018): 1416-1427.

[3] Tziritas, Nikos, Maria Koziri, Areti Bachtsevani, Thanasis Loukopoulos, George Stamoulis, Samee U. Khan, and Cheng-Zhong Xu. "Data Replication and Virtual Machine Migrations to Mitigate Network Overhead in Edge Computing Systems." *IEEE Transactions on Sustainable Computing* 2, no. 4 (2017): 320-332.

[4] Osanaiye, Opeyemi, Shuo Chen, Zheng Yan, Rongxing Lu, Kim-Kwang Raymond Choo, and Mqhele Dlodlo. "From cloud to fog computing: A review and a conceptual live VM migration framework." *IEEE Access* 5 (2017): 8284-8300.

[5] Farahnakian, Fahimeh, Tapio Pahikkala, Pasi Liljeberg, Juha Plosila, and Hannu Tenhunen. "Utilization prediction aware VM consolidation approach for green cloud computing." In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pp. 381-388. IEEE, 2015.

[6] Tao, Fei, Chen Li, T. Warren Liao, and Yuanjun Laili. "BGM-BLA: a new algorithm for dynamic migration of virtual machines in cloud computing." *IEEE Transactions on Services Computing* 9, no. 6 (2016): 910-925.

[7] Farahnakian, Fahimeh, Adnan Ashraf, Tapio Pahikkala, Pasi Liljeberg, Juha Plosila, Ivan Porres, and Hannu Tenhunen. "Using ant colony system to consolidate VMs for green cloud computing." *IEEE Transactions on Services Computing* 8, no. 2 (2015): 187-198.

[8] Dabbagh, Mehiar, Bechir Hamdaoui, Mohsen Guizani, and Ammar Rayes. "An energy-efficient VM prediction and migration framework for overcommitted clouds." *IEEE Transactions on Cloud Computing* (2016).

[9] Velde, Venkateshwarlu, and B. Rama. "A Virtual Machine Based Load Balancing Algorithm for Cloud Computing."“

[10] Chen, Changyuan, and Jian Cao. "Prediction-based optimization of live virtual machine migration." In *IFIP International Conference on Network and Parallel Computing*, pp. 347-356. Springer, Berlin, Heidelberg, 2014.

[11] A-young-son, Eui-Nam huh, Sang-Ho Na. “Migration scheme based machine learning for QoS in Cloud Computing: Survey and research challenges.” *IEEE Transactions on Cloud Computing* (2015).

[12] Khaled Z. Ibrahim, Steven A. Hofmeyr, Costin Iancu, Eric Roman. “Optimized pre-copy live migration for memory intensive applications.” *IFIP International Conference on Network and Parallel Computing*. Springer.

[13]Google

[14] wikipedia.org [online] available:
https://en.wikipedia.org/wiki/Eclipse_(software)

[15] Amazon Web Service [online] available:
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1

[16] wikipedia.org [online] available:
https://en.wikipedia.org/wiki/PyCharm

[17] wikipedia.org [online] available:
https://en.wikipedia.org/wiki/Anaconda_3:_Offspring

[18] wikipedia.org [online] available:
https://en.wikipedia.org/wiki/Linear_regression

[19] wikipedia.org [online] available:
https://en.wikipedia.org/wiki/Cloud_computing