TECHNOLOGY HORIZONS JOURNAL

Vol. 2 (3), 10 March 2018, pp. 83-88

Machine Learning Based Approach for Virtual Machine Migration

M.K.Hassan[1*], Amin Babiker [2], Suliman Zobli [3], Magdi B. M. Amien [3]Mohammed E.A. Kanona [1]

1Future University, Khartoum, Sudan

Collage of Engineering, Al-Neelain University Khartoum, Sudan

Corresponding author E-mail: memo1023@hotmail.com

Received: 09 December 17

Accepted: 05 January 18

ABSTRACT

Nowadays the utilization of Cloud computing in industry, government and academia are growing substantially, this is due to their ability to deliver resilient, robust, and scalable computational power. In cloud computing data centers, high-speed networks interconnect both virtual and physical computers. The dynamic provisioning of these systems is based on end-user computing resources requirements. However, high efficient resource utilization is yet far to reach, therefore the operational costs of these data centers are considerably high. Currently, systems allocate a maximum number of resources in a manner that attempts to ensure that all the Service Level Agreements (SLA) are satisfied. Virtualizations represents the core technology of cloud computing. By creating multiple Virtual Machine (VMs) instances, it allows Cloud providers to manage its data center resources more efficiently and accordingly improving the utilization of resources. In addition to that, by using live migration the VMs can be dynamically consolidated on the minimal number of physical nodes according to their current resource requirements and maintaining SLAs. Therefore, non-optimized and inefficient VMs consolidation may cause performance degradation when an application encounters variable workloads. Thus, to ensure acceptable Quality of Service (QoS) and Service Level Agreements (SLA) a novel machine learning based technique for dynamic consolidation of VMs based on an adaptive prediction of utilization thresholds is proposed to satisfy the Service Level Agreements (SLA). Workload traces from Planet Lab servers have been utilized to validate the efficiency of the proposed technique with different of workload patterns.

Keywords— Traffic; machine learning; SLA; Virtualization;

I. INTRODUCTION

Virtualization introduced opportunities for improved efficiency in cloud data center by increasing isolation between application resource utilization, allocation and management [1]. Live migration is one of the most interesting features in virtualization. That makes virtualization more attractive. By using live migration, running virtual machines (VMs) can be moved seamlessly between physical hosts. This allows service providers hosting high availability applications to better commits their level of service governed by a Service Level Agreement (SLA) and in most cases represented in terms of application availability. Availabilities such as 99.99% which is common in the telecommunication industry obviously permit small fraction of downtime [2].

Consequently, activities such as hardware maintenance are very difficult under such a policy.Moreover, SLA violation may occur due to servers being over utilized e.g CPU utilization is 100 %. Thus, high availability fault-tolerant systems are crucial in order to maintain such demanding policy which is costly in term of CAPEX and OPEX. Live migration can mitigate this problem by moving VMs with little interruption as much as possible subject to the agreed SLA.

However, these interruptions can cause performance degradation and it varies between applications due to different usage patterns, ranging from a few milliseconds [3] to 3 seconds in the case of High-Performance Computing benchmarks [4]. This is can be still unacceptable for some application. Consequently, predicting live migration at the earliest time as much as possible will significantly contribute to reducing any performance degradation resulted from the duration of any interruption. Our objective is to provide a machine learning a predictive model to dynamically consolidate resources based on load and consequently maintaining the SLA. The is heuristic based predictive model where future loads are to be predicted Then decision of migration will be made

by machine learning algorithm which will be used as a classifier based on CPU utilization, inter-VM bandwidth utilization and memory utilization.

II. RELATED WORK

Actually, there has been a tremendous research work that is being taken place in this area. Nathuji [5] have suggested that the resources in the virtualized data center to be classified into global and local policies where live migration is handled by global policies applied to redistribute the VMs. Song et al. [6]have adopted resource allocation according to application priorities in virtualized clusters. In [7] a heuristic for the bin packing problem-based approach has been used to deal with the dynamic consolidation.

In [8] a threshold-based reactive approach to dynamic workload consolidation has been proposed. However, it was suitable for certain types of applications. VMware Distributed Power Management operates on fixed threshold values which apparently is not suitable for dynamic and unpredictable workloads [9]. In our developed model, we propose an approach to set the threshold values dynamically, depending on VMs historical predicted data of the resource usage by each VM and machine learning as a decision making an approach.

In [10] and [11] static and dynamic resource assignment policies in Virtualized data centers are discussed. [12-13] classified server consolidation as centralized and decentralized. In [11, 14] It was discussed that VM consolidation approaches trigger VM migration based on a predefined threshold, where on other side other approaches [12],[15] trigger migration after workload behaviour analysis for a certain amount of time. Based on learned-intelligent QoS-based threshold and predictive heuristic methods [14].

However, a few sets of approaches [13],[14] investigated on workload-independent QoS-based threshold approaches for strict SLA violation avoidance and efficient migration management [14].Authors in [16] proposed a Virtual Machine Placement Problem with Traffic-Aware Balancing (VMPPTB) and proved it to be NP-hard then a Longest Processing Time Based Placement algorithm (LPTBP algorithm) is designed to solve it. Moreover, Locality-Aware Virtual Machine Placement Problem with Traffic-Aware Balancing (LVMPPTB) is proposed.

Authors in [17] proposed a (VM) placement algorithm named ATEA (adaptive three-threshold energy-aware algorithm), In order to reduce the energy consumption and SLA violation. It utilizes historical data from resource usage by VMs, then it migrates VMs on heavily loaded or little-loaded hosts to lightly loaded hosts, while the VMs on lightly loaded and moderately loaded hosts remain unchanged. All previous work did not consider InterVm bandwidth and memory utilization effects on VM consolidation problem and on SLA definition especially in an application that does not tolerate any downtime or performance degradation such as in telecommunication applications [18].

III. Cost of live migration

Live migrations can have negative impact on the performance of applications in a VM during a migration. It has been estimated to be 10% of the CPU utilization depending on the workload of the application [17]. The length of a live migration depends on the total amount of memory used by the VM and available network bandwidth. The migration time and performance degradation experienced by a VM j as expressed in (1)[14].

$$Tm_{j=} \; Mj/B_J \qquad (1)$$

$$U_{dj} \; = 0.1 \int_{t0}^{t0+Tm_j} U_j\,(t)\; d(t) \qquad (2)$$

U_{dj} Is the performance degradation by Vm_j during migration, t_0 is the time when the migration starts,Tm_j is the time taken to complete the migration, U_j is the CPU utilization byVm_j, Mj is the amount of memory used by$\mathrm{Vm_j}$, and $\mathrm{B_J}$ is the available network bandwidth [14].

$$S_j = x_{dj} + U_{dj} \qquad (3)$$

In (3) x_{dj} is the performance degradation when the allocated CPU utilization forVm_j, is 100% and S_j is the Total performance degradation byVm_j,Thus, from equation (3) in order to minimize the total performance degradation ,either U_{dj} should be minimized or x_{dj}. In this work we concentrate o,,,,,,,n minimizing x_{dj} as U_{dj} is not only influenced by the CPU utilization U_j but with the amount of memory used and the available network Bandwidth. Moreover, it is more likely that Vm_j will experience performance degradation while the host CPU utilization is 100% more than during the live migration.

IV. Host Overload Detection

In order to avoid SLA violation performance degradation. The host should perform a regular check on the system utilization where an overload detection algorithm should be executed to de-consolidate VMs. The following part describes recently known host overload detection methods. One of the earliest methods is static CPU Utilization Threshold which relies on setting static CPU utilization threshold to differentiate between overload and the non-overload states of the host. It

is simple but inefficient for dynamic workloads, particularly when different types of applications share a physical node. In such case that the system should be able to automatically adjust its behaviour based on the workload patterns adopted by the applications[14].

V. Local Regression

The main idea of behind our approach as depicted in Fig (1) is to rely firstly on workload prediction based on statistical analysis of historical data collected during the VMs lifetime. Local regression (LR) has proved its efficiency as predictor method as discussed in. It is simple models used build up a curve from localized subsets of data that approximates the original input the original data. LR algorithm, using the described method was derived from Loess, for each new observation a new trend line [14].

$$\hat{g}(x) = \hat{a} + \hat{b}x \quad (4)$$

Is found this trend line is used to predict the next observation$\hat{g}(x_{k+1})$. The new observation can be host resource utilization such as CPU and memory the algorithm detects a host overload, requiring some VMs to be offloaded from the host.

$$\hat{g}(x_{k+1}) \geq 1, \qquad x_{k+1} - x_k \leq t_m \quad (5)$$

Where t_m is the maximum time required for a VM migration.

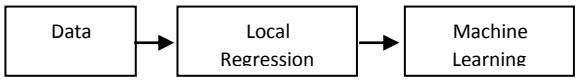


Figure (1) Proposed Method

VI. Classification Trees

Classification trees are effective when a class is already known beforehand in the observation of the learning sample Classes in learning sample can follow a certain hypothesis or be provided by the user. Mathematically, Let t_p be a parent node andt_l, t_r respectively left and tight child nodes of parent the node t_p.Assume the learning sample with variable matrix X with M number of variables x_j and N observations. Let class vector Y consist of N observations with total amount of the K classes.

A classification tree is built according to splitting to rule the hat performs the splitting of learning sample into smaller parts. We already know that each time data have to be divided into two parts with maximum homogeneous the city of left and right child nodes will be equivalent to the maximization of change of impurity function $\Delta i(t)$: [17]

$$\Delta i(t) = i(t_p) - E[i(t_c)] \quad (6)$$

Where t_c - left and right child nodes of the parent nodet_p. Assuming that the P_l, P_r - probabilities of right and left nodes, we Get: [18]

$$\Delta i(t) = i(t_p) - P_l i(t_l) - P_r i(t_r) \quad (7)$$

Therefore, at each node classification trees solves the following

Maximization problem: [18]

$$arg\ max\ xj \leq x_j^R,$$

$$j = 1, \ldots, M\ [\,i(t_p) - P_l i(t_l) - P_r i(t_r)] \quad (8)$$

From Equation (8) all possible values of all variables in matrix X for the best split question will be searched through $xj < x_j^R$ which will maximize the change of impurity measure$\Delta i(t)$ [18].

VII. Support Vector Machines

Support vector machine (SVMs) support nonlinear classification and can find the hyperplane of maximal margin Equation (9) shows an example of hard-margin SVM with noise-free training data to be correctly classified by a linear function. The data points D (or training set) can be represented mathematically as follows. [19]

$$D = \{(x_1, y_1), (x_2, y_2), \ldots . (x_m, y_m)\} \quad (9)$$

Where x_i is an n-dimensional real vectorr, y_i is either 1 or -1

Indicating the class to which the point x_1 belongs to. The SVM classification function $F(x)$ takes the form [19]

$$F(x) = w.x - b \quad (10)$$

b is the bias and w is the weight vector , which will be calculated by SVM during the training process. First, to correctly classify the training set .*F(x)* (or w and b) must return negative numbers for negative data points and positive numbers otherwise, that is, for every point x_i in D as expressed mathematically in (11)

$$w.x - b > 0\ if\ y_i = 1, and\ w.x - b < 0\ if\ y_i = 1 \quad (11)$$

The k-nearest neighbours (KNN) is one of the simplest methods for pattern classification. However, it is proven classification technology in many domains and is used to produce significant results when combined with prior knowledge. In [19].

The KNN each unlabeled example is classified by the majority label among its k -nearest neighbours in the training set. Therefore its classification performance depends on the distance metric used to identify nearest neighbours. In the case of the prior knowledge is missed, Euclidean distances between examples represented as vector inputs are used to measure the similarities in most KNN classifiers, do not capitalize on any Statistical regularities in the data that might be estimated from a large training set of labelled examples.

Let $\{(\vec{x_i}, yi)\}_{i=1}^{n}$ denote a training set of n labeled examples with inputs $\vec{x_i} \in R$ and discrete (in our case binary) class labels y_i ,We use the binary matrix $y_i, j \in \{0,1\}$ to indicate whether or not the labels y_i and y_j match .Our goal is to learn a linear transformation which we will use to compute squared distances as:

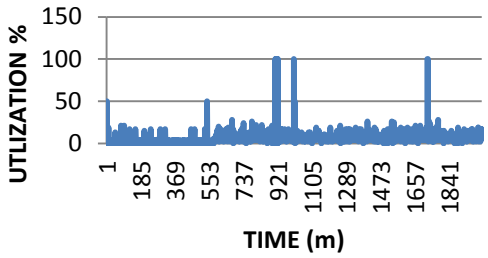
$$D(\vec{x_i}, \vec{x_j}) - ||L(\vec{x_i} - \vec{x_j})||^2 \quad (12)$$

VIII. Results and Analysis

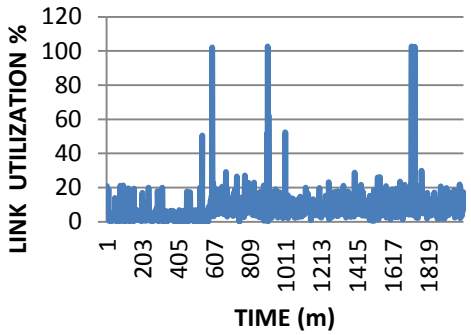
Real data from real live servers is crucial to perform the experiments. In this work, workload collected from CoMon project has been used. [17].The extracted data was part of more than a thousand VMs CPU utilization distributed across the world. Samples were selected randomly from 3 servers for a period of one week in the period from March to April 2011 with 5 minutes were used as measurement interval. It is observed that the average CPU utilization is far below 50%.Therefore CPUs Utilization was adjusted artificially where high CPU utilization is injected for the purpose of this experiment.

Moreover, the collected data did not contain memory and Inter VM bandwidth utilization. Therefore it was generated artificially. In this study, the evaluation metrics true positive rate (TPR), and false negative rate (FNR) have been used to assess the performance of the classification techniques. A true positive (TPR) is defined as VM migration being correctly classified to be migrated due to high utilization, and the classification output is considered as a false negative rate (FNR) if VM migration is classified to not to migrate from the current server. Both metrics are represented in percentage. For the training and testing Since the provided data set is not very large, Cross-validation has been used to train, test and validate the classification techniques, data is divided into 5 folds and each fold is held out in turn for training and testing.

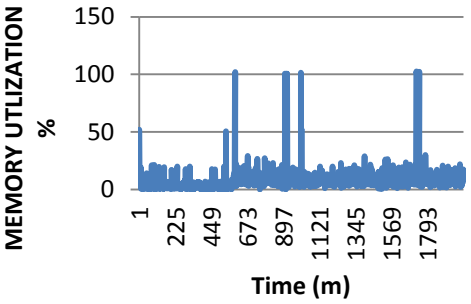
First of all, data are predicted using local regression as described in above provided that prediction window is less than or equal to the migration time as in the bond $x_{k+1} - x_k \leq t_m$and this is crucial to maintain the SLA then different classification techniques are used. Data were collected from three servers named Venus, Chimay and Alladein. Figure (2) shows sample resource utilization for Venus Server for one week.



(a)



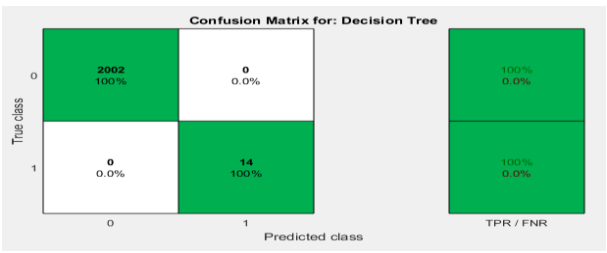
(b)



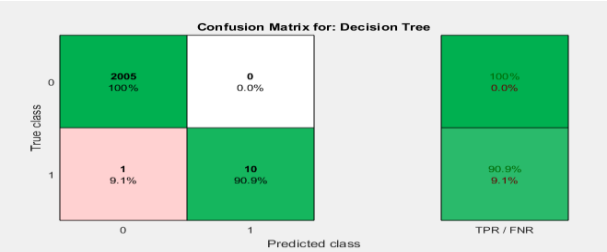
(c)

Figure (2) Venus server resource utilization. (a) CPU (b) Inter VM (c) Memory

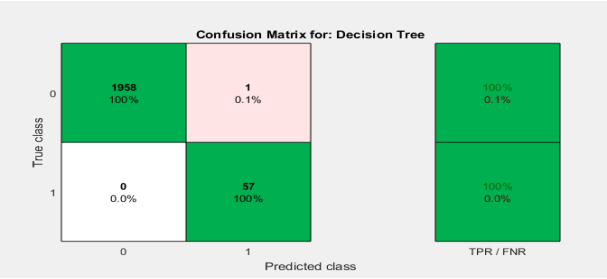
Figure (3) shows confusion matrix when regression trees are used for the three server readings. It is clearly seen that in Venus data. regression trees were able to correctly classify and predict the VM migration as indicated in 100 %TPR and 100 % FNR since all zero classes are correctly classified (predicted) as zero classes and likewise, all 1 classes are correctly classified (predicted) as 1 classes. In Chimay data, TPR and FNR were 100 % and 90.9% respectively and for Alaadein data TPR and FNR were almost 100 %.



(a)



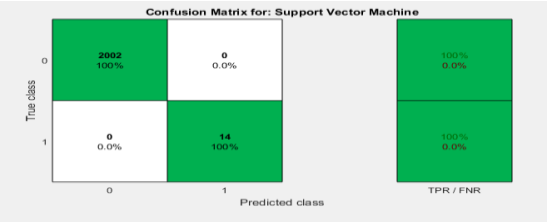
(b)



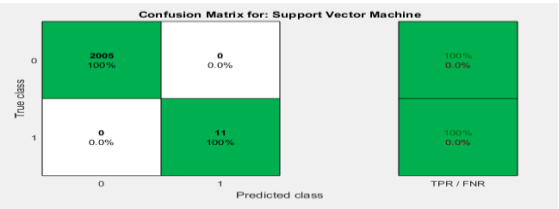
(c)

Figure (3) Confusion Matrix for regression trees (a) Venus (b) Chimay (c) Alaadein

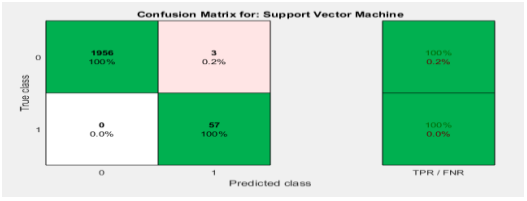
In Figure (4), classification results of using SVM is presented, It shows the confusion matrix when Support vector machine is used for the three server readings. It is clearly seen that in Venus data. SVM was able to correctly classify and predict the VM migration as indicated in 100 %TPR and 100 % FNR while in Chimay data, TPR and FNR were 100 % and 100% respectively and for Alaadein data TPR and FNR were almost 100 % with only less than 0.5% was misclassified. It is obvious that SVM has better improvement in general classification performance



(a)



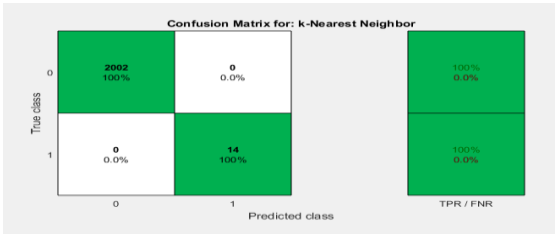
(b)



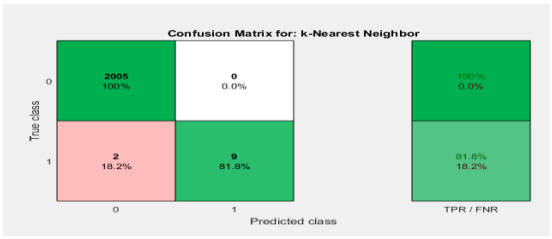
(c)

Figure (4) Confusion Matrix for SVM. (a) Venus (b) Chimay (c) Alaadein

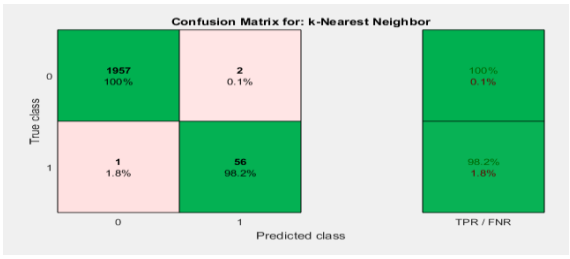
Figure (5) shows the results of using kNN as a classifier, from the provided confusion matrix.



(a)



(b)



(c)

Figure (5) Confusion Matrix for KNN. (a) Venus (b) Chimay (c) Alaadein

VX. Conclusion

It is clearly seen that in Venus data, KNN classified and predicted the VM migration as indicated in 100 %TPR and 100 % FNR. In Chimay data, TPR and FNR were 100 % and 81.2% respectively and for Alaadein data TPR and FNR were around 100 % and 98.2 % respectively.Obviously, classification performance is less in KNN compared to regression trees and support vector machine

References

1. Barham, P., et al. *Xen and the art of virtualization*. in *ACM SIGOPS operating systems review*. 2003. ACM.
2. Akoush, S., et al. *Predicting the performance of virtual machine migration*. in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*. 2010. IEEE.
3. Clark, C., et al. *Live migration of virtual machines*. in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*. 2005. USENIX Association.
4. Nagarajan, A.B., et al. *Proactive fault tolerance for HPC with Xen virtualization*. in *Proceedings of the 21st annual international conference on Supercomputing*. 2007. ACM.
5. Nathuji, R. and K. Schwan. *Virtual power: coordinated power management in virtualized enterprise systems*. in *ACM SIGOPS Operating Systems Review*. 2007. ACM.
6. Song, Y., et al. *Multi-tiered on-demand resource scheduling for VM-based data center*. in *Cluster*

Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on. 2009. IEEE.

7. Verma, A., P. Ahuja, and A. Neogi. *pMapper: power and migration cost aware application placement in virtualized systems*. in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*. 2008. Springer-Verlag New York, Inc.
8. Gmach, D., et al., *Resource pool management: Reactive versus proactive or let's be friends.* Computer Networks, 2009. 53(17): p. 2905-2922.
9. Ferreto, T.C., et al., *Server consolidation with migration control for virtualized data centers.* Future Generation Computer Systems, 2011. 27(8): p. 1027-1034.
10. Wood, T., et al., *Memory buddies: exploiting page sharing for smart colocation in virtualized data centers.* ACM SIGOPS Operating Systems Review, 2009. 43(3): p. 27-36.
11. Hirofuchi, T., et al. *Reactive consolidation of virtual machines enabled by post-copy live migration*. in *Proceedings of the 5th international workshop on Virtualization technologies in distributed computing*. 2011. ACM.
12. Kakadia, D., N. Kopri, and V. Varma. *Network-aware virtual machine consolidation for large data centers*. in *Proceedings of the Third International Workshop on Network-Aware Data Management*. 2013. ACM.
13. Beloglazov, A., *Energy-efficient management of virtual machines in data centers for cloud computing*, 2013.
14. Chen, T., X. Gao, and G. Chen, *Optimized Virtual Machine Placement with Traffic-Aware Balancing in Data Center Networks.* Scientific Programming, 2016. 2016.
15. Sindelar, M., R.K. Sitaraman, and P. Shenoy. *Sharing-aware algorithms for virtual machine colocation*. in *Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures*. 2011. ACM.
16. Zhou, Z., Z. Hu, and K. Li, *Virtual machine placement algorithm for both energy-awareness and SLA violation reduction in cloud data centers.* Scientific Programming, 2016. 2016: p. 15.
17. Alla, M.K.H., et al., *REVIEW IN CLOUD-BASED NEXT GENERATION TELECOMMUNICATION NETWORK.* JURNAL TEKNOLOGI, 2016. 78(6): p. 51-57.
18. Timofeev, R., *Classification and regression trees (CART) theory and applications.* Humboldt University, Berlin, 2004.
19. Weinberger, K.Q. and L.K. Saul, *Distance metric learning for large margin nearest neighbor classification.* Journal of Machine Learning Research, 2009. 10(Feb): p. 207-244.

