

CORRELATION BASED VMS PLACEMENT RESOURCE PROVISIONION

Zar Lwin Phyo¹ and Thandar Thein²

¹University of Computer Studies, Mandalay ,Myanmar
zarlwinphyo@gmail.com

²University of Computer Studies, Yangon, Myanmar
thandartheinn@gmail.com

ABSTRACT

In Virtualized Data Center (VDC), a single Physical Machine (PM) is logically divided into one or more Virtual Machines (VMs) that share physical resources. Therefore, dynamically resource provisioning plays an important role in VDC. Moreover the resource provider would like to maximize resource utilization, which forms a large portion of their operational costs. To achieve this goal, several consolidations can be used to minimize the number of PMs required for hosting a set of VMs. However, consolidation is often undesirable for users, who are seeking maximum performance and reliability from their applications. These applications may indirectly increase costs due to under-provisioning of physical resources. Moreover frequent Service Level Agreement (SLA) violations result in lost business. To meet SLA requirements, over provisioning of resources is used. However, if the services do not use all the CPU they have been allocated; the provider will suffer from low resource utilization, since unused resources cannot be allocated to other services running in the provider. Therefore VM provisioning is the most vital step in dynamic resource provisioning. In this paper, correlation based VMs provisioning approach is proposed. Compared to individual-VM based provisioning, correlation based joint-VM provisioning could lead to much higher resource utilization. According to the experimental results, proposed approach can save nearly 50% CPU resource in terms of overall CPU utilization.

KEYWORDS

consolidation, resource provisioning, resource utilization, SLA, Virtualized Data Center

1. INTRODUCTION

Recent advances in hardware virtualization technologies enable VDC to consolidate the workload of the active virtual machines (VMs) into a small set of powerful servers in order to reduce the maintenance cost of the data center. However, such cost saving comes with the sacrifice of the VMs' Quality of Service (QoS). In this case, efficient server consolidation plays an important role in VDC.

Server consolidation has emerged as a promising technique to reduce the energy costs of a VDC and server consolidation can have a significant effect on overall data center performance. It is based on the observation that many enterprise servers do not maximally utilize the available server resources all of the time, and virtualization technologies facilitate consolidation of several physical servers onto a single high end system for higher resource utilization.

Co-locating applications (or virtual machines), perhaps in individual virtual machines, thus allows for a reduction in the total number of physical servers, minimizes *server sprawl* as well as the total data center space requirements. Consolidation not only can increase the total resource utilization at server but also can reduce the total power consumed by the applications because existing servers are not energy proportional, i.e., a significant amount of power is consumed even at low levels of utilization. Consolidation thus provides an opportunity to reduce the overall power consumed by operating the servers in a range with a more attractive performance and high resource utilization.

Generally, there are 3 types of server consolidation: static, semi-static and dynamic consolidation. In static consolidation, applications (VMs) are placed on PMs for a long time period (e.g. months, years), and not migrated continuously in reaction to load changes. In semi-static consolidation, these applications are placed based on a daily or weekly basis. However, dynamic consolidation spans a couple of hours and requires a runtime placement manager to migrate VMs automatically in response to workload variations [4].

When a VM controller needs to create and deploy a set of VMs on a set of PMs, server consolidation is performed. The goal of server consolidation is to determine a mapping of VMs to physical hosts such that the minimum numbers of hosts are used. Existing consolidation schemes consist of two steps: estimating the future size for each VM, and placing VMs on physical hosts. The first step, estimating VM size, is usually solved by first forecasting the future workload, then finding a capacity size that can sufficiently cover the forecasted workload. The second step, VM placement, usually requires solving a bin packing type of problem. Specifically, since each VM carries a size and each physical host has fixed capacity, the VM placement problem is equivalent to packing items (VMs) into the smallest number of bins (hosts) without violating the size limit on each bin. In practice, VM placement is tackled by either heuristics or solving an integer programming problem [11]. By exploiting VM multiplexing, it is possible to achieve even more compact consolidation. Therefore, correlation based VM multiplexing approach is proposed in which VMs are consolidated and provisioned together, based on their aggregate capacity needs. Correlation based VM provisioning (CBP) exploits statistical multiplexing among the dynamic VM demand characteristics. We also evaluate the effectiveness of the proposed approach against a First-Fit Decreasing (FFD) approach.

The rest of the paper is structured as follows: Section 2 presents the related work of this paper. Section 3 describes dynamic resource provisioning system for VDC. System evaluation is presented in section 4. It is followed by conclusion in section 5.

2. RELATED WORKS

Some related work of VM sizing and placement in VDC are presented in this section.

S. Govindan et al.[8] presented models for estimation of the CPU usage when dispersed VMs are moved to collocated placement. First, they empirically demonstrated and quantified the resource savings due to collocation of communicating VMs and then they build models based on different resource usage micro-benchmarks to predict the resource usages when moving from non-collocated placements to collocated placements and vice-versa. These models are general enough to be applicable to any application hosted in a virtualized environment.

Meng et al. [11] proposed a joint-VM provisioning approach in which multiple VMs are consolidated and provisioned based on an estimation of their aggregate resource need. Verma et al. [3] presented a power-aware application placement methodology and system in the context of a runtime placement controller. Verma et al. [4] presented two new consolidation methods that

take both load correlation and dynamics into VM placement considerations. Bobroff et al. [7] outlined a systematic approach to identify the servers that are good candidates for dynamic placement, and present a mechanism for dynamic migration of VMs based on a load forecast. Govindan et al. [9] proposed to consolidate VMs based on their communication patterns. S. Govindan et al. used statistical multiplexing of applications to identify applications that fit into given power budgets [8].

In this paper, we apply VM multiplexing to consolidate VMs more densely on hosts. This work provides a general performance model and enabling techniques that ensure the application performance is not degraded by correlation based VM multiplexing.

3. DYNAMIC RESOURCE PROVISIONING SYSTEM FOR VDC

To place VMs on PMs, we have to know their resource requirements. Moreover, in VDC, the workload on servers changes over time which may change the resource requirements of VMs. To provide desired performance of applications with efficient resource utilizations in data-center, VMs sizing is the most vital step. The objective of VM sizing is to ensure that VM capacity is commensurate with the workload. While over provisioning wastes costly resources, under-provisioning degrades application performance and may lose business. There are number of factors that we have to consider during VM sizing, most of which are given in Service Level Agreement (SLA) for the user’s applications running in VM.

There are many important parameters that should be defined in the SLA such as CPU capacity, memory size, availability, response time, etc. In this paper, we emphasize on computing capacity (CPU) of a resource provider to specify fine-grain QoS guarantees. Figure (1) shows the procedure of VMs placement in VDC in the form of flowchart.

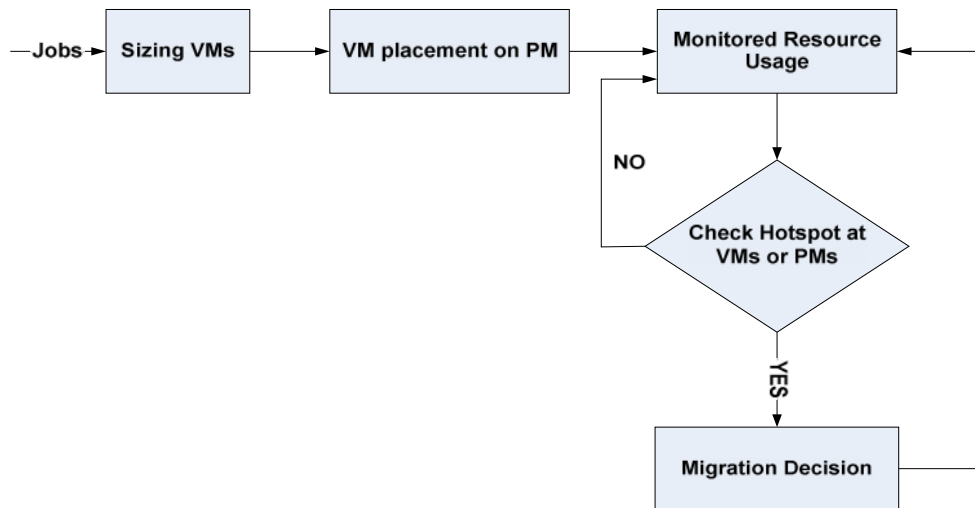


Figure 1. Procedure of VM Placement in VDC

Correlation Based Provisioning (CBP) approach is used to place VMs on PMs. For a given set of VMs, correlation matrix (C) is built based on the historical resource demand behaviour of each VM. To consider the correlation between a pair of VMs, Pearson correlation coefficient is used. In this case, we consider VM pairs with strong negative correlations as good candidates for VM multiplexing because a negative correlation indicates that these VMs demand behaviour changes

in opposite directions. Therefore, we combine VMs with not similar temporal peaks and troughs pattern. Therefore unused resources of a low utilized VM can be directed to the other co-located VMs at their peak utilization. Although there are additional virtualization overheads associated with scheduling depend on the number of VMs, CBP is suitable for VM sizing and placement.

3.1. CPU Based Service Level Agreement

A Service level agreement is a document that includes a description about agreed service, service level parameters, guarantees, and actions and remedies for all cases of violations [2]. The SLA is very important as a contract between consumer and provider. The main idea of SLAs is to give a clear definition of the formal agreements about service terms like performance, availability and billing.

In VDC, there are many different architectures: processor models, each of them with different clock frequencies, etc. For this reason, a good resource level metric has to be platform-independent so it can be used in all these architectures. In this section, we describe our approach for unifying computing capacity (CPU capacity) metrics among machines with heterogeneous architectures.

Commonly, to measure the computing capacity of a machine, *megahertz* has been used. However, computing power is depending on the processor architecture and it does not directly measure. For instance, using this measure an Intel Pentium III with 500 MHz would be 20 times slower than an Intel Xeon 4-core with 2.6 GHz ($4.2600 / 500 = 20.8$). However, a simple performance comparison (e.g. *BogoMips* [10]) demonstrates that it can be up to 40 times slower ($45322.2 / 499.71 = 42.6$).

In order to consider the heterogeneity of the different processor architectures, Amazon uses EC2 compute units (ECU) in its services [1]. For CPU-related SLA metrics among heterogeneous machines we use ECUs in this work. The maximum computing capacity of ECU associated machine is measured by using $CPU_A \times ECU_s$.

3.1.1. Derivation of CPU-based SLA Metric

Traditionally, over provisioning of resources is used to meet SLA requirements. However, since servers operate most of the time at very low utilization level, it leads to waste of resources in non-peak times. This over provisioning of resources results in extra maintenance costs including server cooling and administration [6]. Today, most of the data centres run different types of applications on separate VMs without any awareness of their different SLA requirements such as deadline, which may result in resource under-utilization and management complexity.

In this paper, we derive a resource-level metric for specifying QOS guarantees regarding the computing capacity (CPU). To achieve better resource utilization in the provider, it allows the provider to implement dynamic resource provisioning and allocate to the different services only the amount of CPU they need. In fact, most of them only support SLAs with very simple metrics based on resource availability. It particular, metrics related to the provider's CPU is especially susceptible to naïve usage. In the SLA establishes, the provider must maintain the agreed metric during the whole execution of the service whether the applications use all allocated capacity during some phases of its execution [5].

The user is only required to specify the computing performance he requires, which will be accepted by the provider if he is able to provide that performance. The allocated CPU (CPU_A) can

vary over time depending on the provider's status, and it is periodically obtained by means of the monitoring subsystem.

$$\frac{CPU_A \times ECUs}{CPU_R} \geq 1 \quad (1)$$

In Equation (1), we introduce the metric, in which CPU_A to the application is greater than or equal to the required capacity (CPU_R) at time t . Otherwise, SLA violation (SLA_V) will occur. After that we find number of intervals (V_I) which are violated SLA agreement value. However SLA could be violated in two situations in this case. First, real violation, when the provider assigns to the service an amount of CPU that is not enough to fulfil the SLA. The provider is responsible for it, and must pay the corresponding penalty. Second, although the provider assigns to the service an amount of CPU that should be enough to fulfil the SLA, the service does not use all the assigned CPU. This can be defined *fake* violation, since the provider is not causing it, and for this reason, he should not pay any penalty.

We consider the real SLA violation (SLA_{RV}). Therefore we calculate real SLA violated intervals (SLA_{RVI}).

$$SLA_{RVI} = < V_I \quad (2)$$

In equation (2), factor acts as a SLA agreement value to calculate when the service is considered to be violation of SLA agreement or not. A penalty q will incur if number of such violation increases beyond a threshold .

$$SLA_V = \frac{SLA_{RVI} * 100}{no\ of\ epoches} \quad (3)$$

The SLA is evaluated every time interval and the real SLA violation is calculated by using equation (3). Hence, SLA_V is defined as SLA violation and V_I stands for the number of SLA violated interval. If the SLA is violated in every time interval, maximum number of SLA violations occurs.

3.2. Correlation based VM provisioning (CBP)

There are many applications which are running in VDC. The workload on applications changes over time and they do not fully utilize the available physical resources all the time. There is high probability that the workloads of these VMs reach at peak level at the same time.

Obviously, an important factor contributing to the realized capacity savings with VM multiplexing is the way VMs are combined: a method that combines VMs with similar temporal peak and trough patterns cannot achieve savings compared to a method that identifies and favours VMs with complementary temporal behaviour.

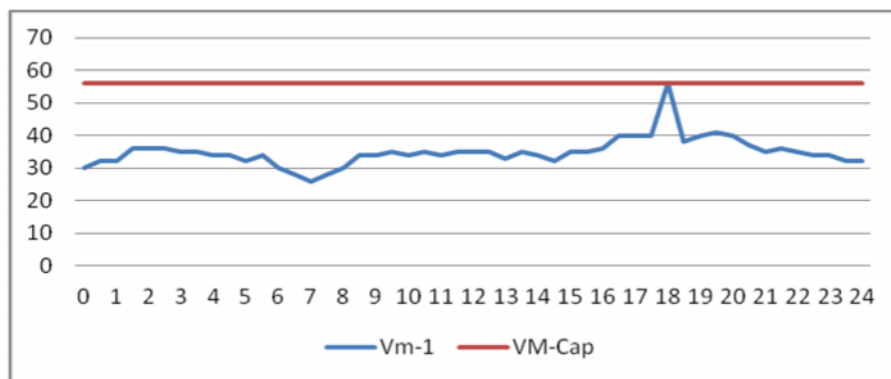
Therefore, correlations among VMs need to be considered during VM sizing and placement to avoid SLA violations and reduce migration cost. The workloads of some applications are periodic or stable for a specific period of time. So we can predict the CPU requirements of VMs containing such applications based on historical CPU utilization statistics.

When the size of VMs has been known, the next step is to place VMs on the minimum number of PMs. In this case, we calculate the Correlation matrix among VMs. CPU utilization traces of VMs and parameter in the SLA model are taken as input to the CBP. These traces have been

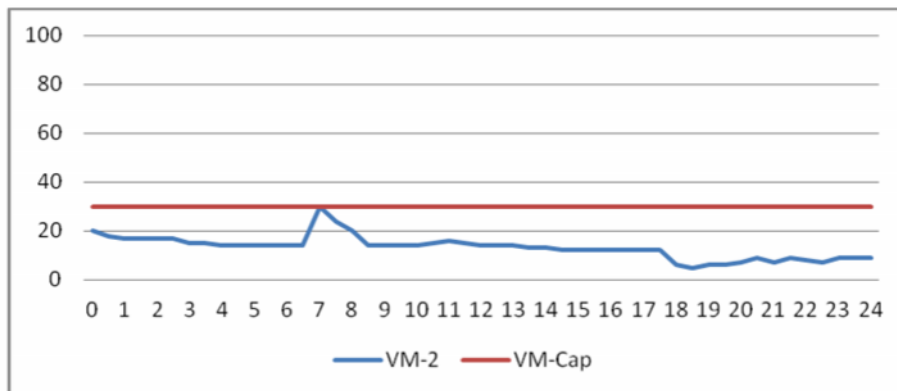
taken as slices of time series t_0, t_1, \dots, t_n . Based on these traces, it allocates CPU resource to VMs. CBP uses Correlation variable which imposes constraint on correlation between VMs. The correlation between a pair of applications (VMs) with time-series $\{x_1, x_2, \dots, x_n\}$ and $\{y_1, y_2, \dots, y_n\}$ is represented by the Pearson correlation coefficient :

$$C_{xy} = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{N \sum x_i^2 - (\sum x_i)^2} \cdot \sqrt{N \sum y_i^2 - (\sum y_i)^2}} \quad (4)$$

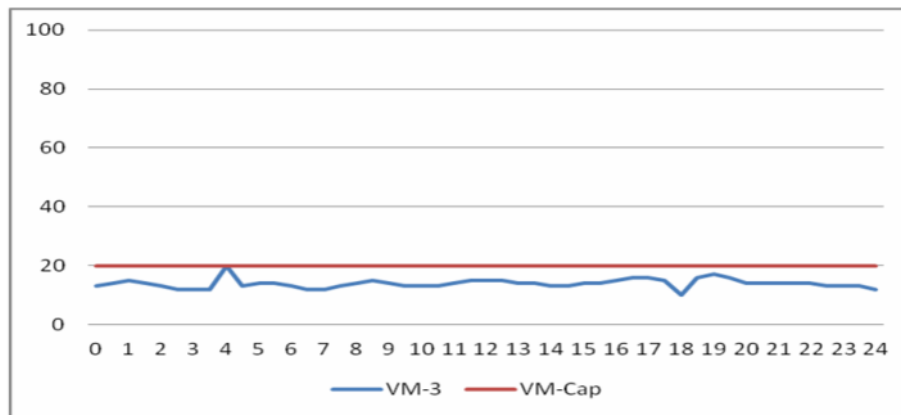
Firstly, correlation matrix C is built based on the CPU utilization behaviour of each VM. In this case, positive correlations of VMs pairs indicate that these VMs demand behaviour are similar and negative correlation of VMs pairs indicate that these VMs demand behaviour changes in opposite directions. Therefore, VM pairs with strong negative correlations are considered as good candidates for VM multiplexing.



(a)



(b)



(c)

Figure2. (a), (b) and (c) CPU Resource Requirement of VM-1, VM-2 and VM-3

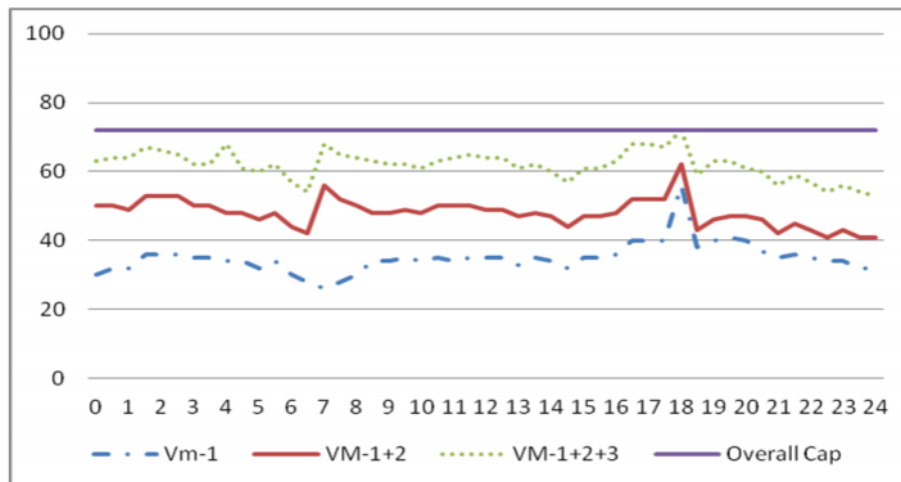


Figure3. Aggregate CPU Resource Requirement of VMs

Figure (2) and (3) depict an example with three sets of VM which are selected from the production data centre. Figure (2) illustrates the monitored CPU demand of each VM over a 24 hour period. This figure also depicts with the capacity bound required by each VM that is based on conservatively satisfying peak demand of each VM. However, the peaks of each VM occur at different time. Based on this capacity model, the total capacity required for this three VM is 108%. Figure (3) shows the aggregated CPU requirements of three VMs when they are jointly provisioned. Here we observe the total capacity required to satisfy the demand of all the three VMs is only 72%, a dramatic improved compared to the separate provisioning.

4. SYSTEM EVALUATION

In this section, we demonstrate and evaluate CBP with three datasets and demonstrate its substantial benefits with dramatic improvements in resource use efficiency.

4.1. Experimental Setup

To demonstrate the effectiveness of proposed CBP approach, we used trace collected from a commercial data centres. It includes 10- thousand VMs and spans 3 years (2007, 2008, 2009).All the evaluations are based on this dataset. These data are tasted on the test bed which is shown in Table 1.

Table 1. Specification of Test Bed

Test Bed	
PM	2.3 GHz Intel processor
	128 GB of RAM
VM	1.3 GHz CPU
	16 GB of RAM

The data centre had a large number of datasets and, firstly, we extract the server information from this dataset. VMsSuit 1 contains 255 VMs and in VMsSuit 2, which contains 1000 VMs. The purpose of this experiment is to try various VM provisioning schemes by consolidating these VMs into the hosts. The detailed information about the applications running in the data centre and the virtual server to physical server mapping are withheld for privacy and business reasons. Then we analyze the nature of datasets. Figure (4) shows the demand trace of dataset 1 and Figure (5) and Figure (6) show that of dataset 2 and dataset 3.

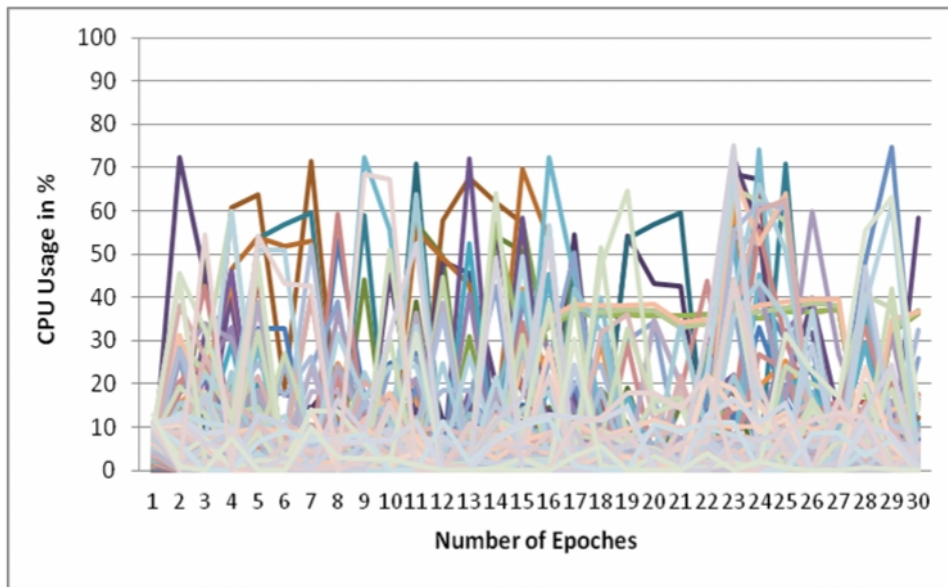


Figure 4. Workload Demand Patterns of Dataset 1

According to Figure (4), we note that not all resource usage is high peak usage in Dataset 1. Firstly we analyze the resource usage pattern of Dataset1. In this Dataset, 72.8 % of all VMs have low resource usage pattern and only 7.2% have peak usage pattern.

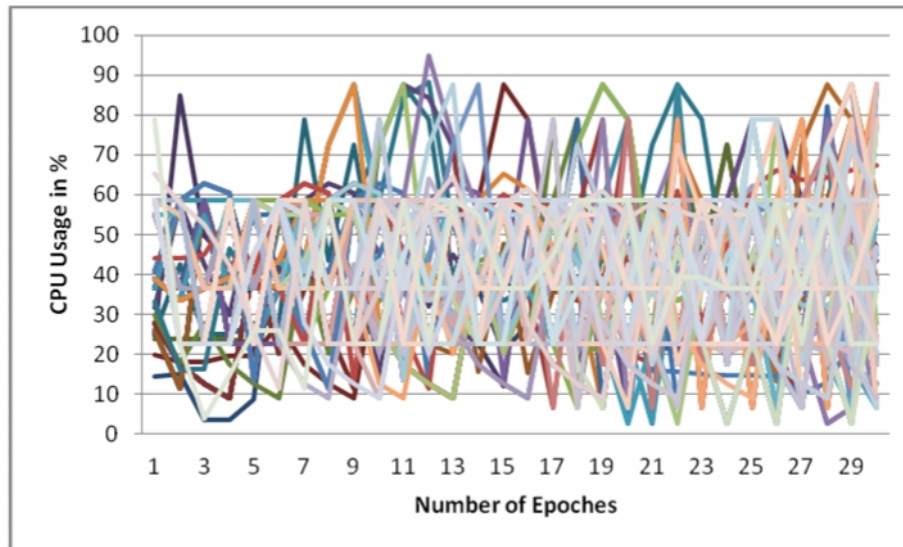


Figure 5. Workload Demand Patterns of Dataset 2

However, in Dataset 2, 72% of all VMs have CPU resource usage pattern with different high peaks and valleys and 28 % of all VMs have peak usage pattern. In Dataset 3, 92% of all VMs that exhibit demand patterns with different high peaks and valleys according to Figure (5) and Figure (6).

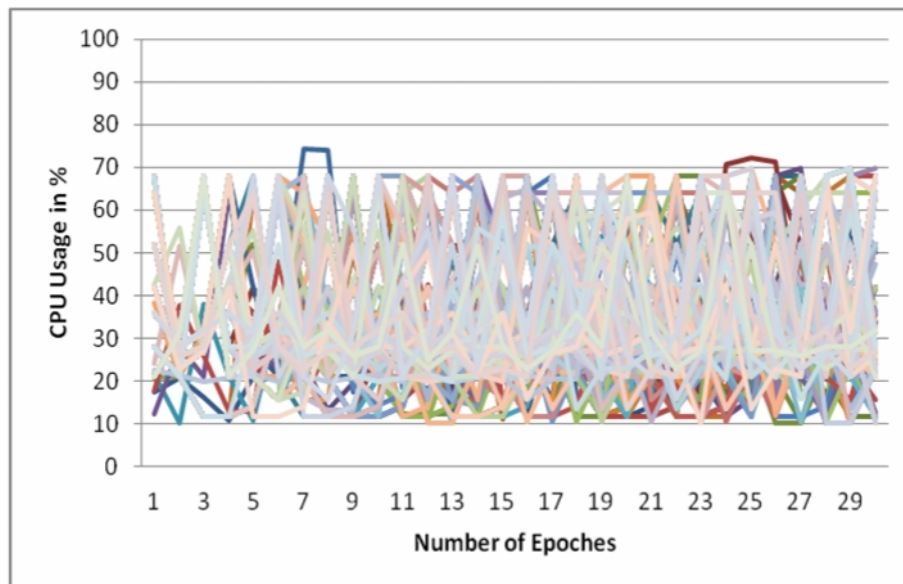


Figure 6. Workload Demand Patterns of Dataset 3

4.2. Experimental Results

By exploiting CBP approach, it is possible to achieve even more compact consolidation. We evaluate the quality of the proposed approach against a FFD approach. Figure (7) and Figure (8) demonstrates the comparison of capacity saving between CBP approach and FFD approach of VMsSuit 1 and VMsSuit 2 respectively. For this evaluation, we assumed a provider that is able to predict the future VM demand.

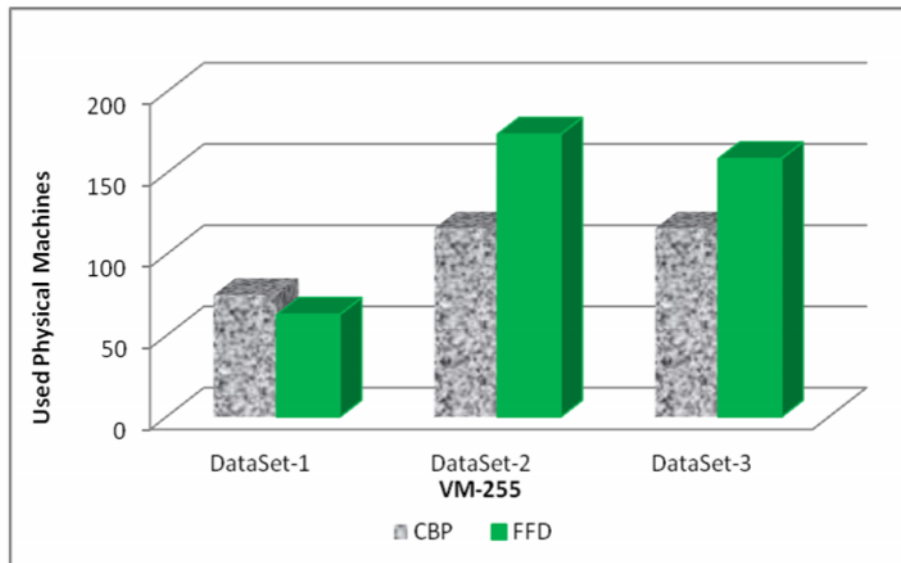


Figure 7. Total Required Capacity of CBP and FFD

According to the experimental result of Figure (7), CBP approach is suitable for Dataset 2 and Dataset 3 not for Dataset 1. In Dataset 1, most of the resource usage patterns are not peak and valley. There are VMs that exhibit demand patterns with different high peaks and valleys in Dataset 2 and Dataset 3. In our experiment, the more physical machines are used by using CBP than FFD in Dataset 1. However, in Dataset 2 and Dataset 3, the usage of physical machine of CBP approach is lower than FFD. Therefore, we note that CBP approach is more suitable for VMs that exhibit demand patterns with different peaks and valleys. However it might not be applicable for off peaks usage pattern.

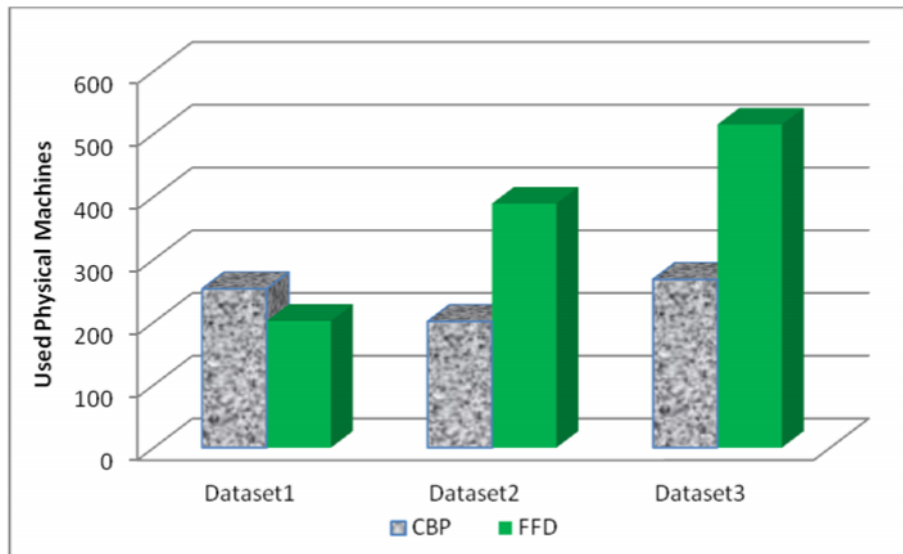


Figure 8. Total Required Capacity of CBP and FFD

According to the figure (8) of the experimental result, CBP approach can save nearly 50% CPU resource in terms of overall CPU utilization. Although CBP incurs an overhead in the computation of correlation for all VM pairs, the more VMs with high peaks and troughs resource usage pattern get, the more capacity saving achieved in CBP approach.

4.3. SLA Evaluation

The violation of the SLA specified by the capacity requirement is also considered. With forecasting, the actual CPU usage in future time horizon could be potentially greater than the given allocated CPU (CPU_A) and thus leads to SLA breach. Figure (9) shows the percentage of VMs with the actual CPU usage exceeding the given CPU_A . Overall more than 87% of the VMs still meet their SLA based on CBP approach. In these experiments, we have considered that $\alpha = 0.2$ and 0.3 these correspond to the 80% and 70% of CPU agreement in Equation 2.

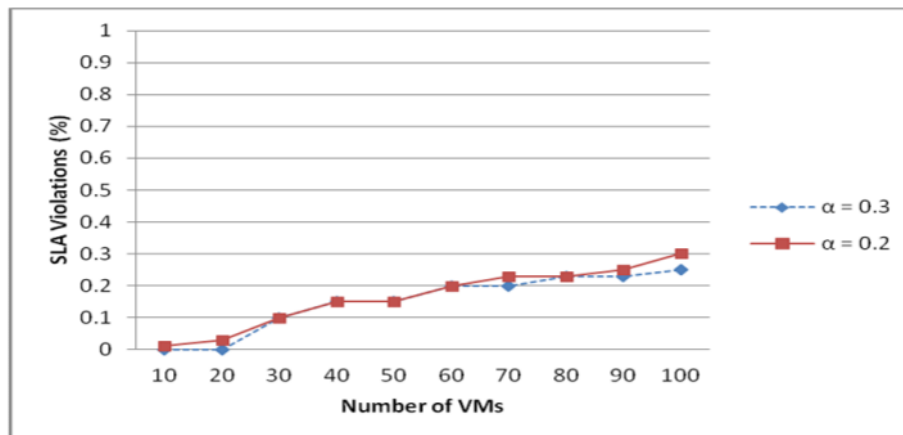


Figure 9. SLA Violations (%) vs Number of VMs with Different SLA Agreement Value

5. CONCLUSIONS

In this paper, correlation based VMs provisioning model is proposed for reducing the maintenance cost of VDC. Experimental evaluation revealed that our model leads to significant reduction in the number of active physical hosts compared to that of the FFD placement algorithms. In case of negatively and uncorrelated VMs our model resulted in 30 - 40% reduction in the number of active servers required compared to FFD provisioning. Whereas in case of a dataset consisting mostly of positively and non-peak usage pattern, VMs the reduction goes down about 5% - 10% under FFD approach. Although there is initial placement overhead, our proposed model is suitable for not only peaks and valleys resource usage pattern but also high resource usage pattern. Especially, the potential savings occurs if application were sized based on values as opposed to peak values. Because of the correlation between same applications has significant positive correlation, therefore, when non-correlated applications are provisioning together, high capacity saving occur. Therefore correlation needs to be considered during resource provision to avoid resource underutilization.

In this paper we only discuss VM multiplexing on CPU demand. However server consolidation is a multi-dimensional bin-packing problem. Therefore, in the future, other resources like disk I/O, network bandwidth are considered in VM multiplexing. Moreover, the workloads of the servers change with time which in turn change the resource demands of VMs. Overloaded conditions are formed in VDC when the resource requirement of one or more VMs on a physical host exceeds the host's capacity. Therefore, in the future, in order to detect this condition and provision resources to overloaded VMs, migration decision model for the virtualized data centre will be proposed.

REFERENCES

- [1] Amazon EC2, 2010. Website: <http://aws.amazon.com/ec2/>.
- [2] A.Andrieux, et al. Web services agreement specification (WSAgreement), Grid Resource Allocation Agreement Protocol (GRAAP) WG, March 14, 2007 <http://forge.gridforum.org/sf/projects/graap-wg>
- [3] A.Verma, 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, New York, NY, USA, 2008.
- [4] A.Verma,V.Dasgupta,T.Nayak.P.De and R.Kothari, "Server workload analysis for power minimization using consolidation", In proceedings of the 2009 USENIX Annual Technical Conference, 2009.
- [5] I.Goiri,F.Julia,J.O.Fito,M.Macias and J.Guitart, "Supporting CPU-based guarantees in cloud SLAs via resource level QOS metrics". © 2011 Elsevier B.V. All rights reserved.doi:10.1016/j.future.2011.11.004
- [6] J.K.Kim, H.J. Siegel, A.A.Maciejewski, R.Eigenmann, "Dynamic resource management in energy constrained heterogeneous computing systems using voltage scaling" IEEE Trans. Parallel Distrib. Syst. 19(11), 1445–1457 (2008)
- [7] N.Bobroff, A.Kochut, and K.Beaty, "Dynamic placement of virtual machines for managing SLA violations", IFIP/IEEE International Symposium on Integrated Network Management, 2007.
- [8] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini, "Statistical profiling-based techniques for effective power provisioning in data centers", In EuroSys '09: Proceedings of the 4th ACM European conference on Computer systems, pages 317–330, 2009.
- [9] S. Govindan, A. R. Nath, A. Das, B. Urgaonkar, and A. Sivasubramaniam, "Xen and co.: communication-aware CPU scheduling for consolidated xen-based hosting platforms", In ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE), pages 126–136, 2007.
- [10] W. van Dorst, "BogoMips mini-Howto", Tech. Report, Clifton Scientific Text Services V38, March 2006.
- [11] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, Efficient resource provisioning in compute clouds via VM multiplexing, In ICAC '10: Proceeding of the 7th international conference on Autonomic computing, pages 11-20, New York, NY, USA, 2010. ACM.

AUTHORS

Zar Lwin Phyo is now Ph.D. student in the University of Computer Studies, Mandalay (UCSM). She obtained her Bachelor degree in Computer Science and Master degree in Computer Science from the UCSM in 2003 and 2005, respectively. She is an assistant lecturer. Her current research interest is in the area of Virtualization. E-mail: zarlwinphyo@gmail.com



Thandar Thein received her M.Sc. (Computer Science) and Ph.D. (Information Technology) degrees in 1996 and 2004, respectively from University of Computer Studies, Yangon (UCSY), Myanmar. She did her post doctorate research fellowship in Computer Engineering Department of Korea Aerospace University, the scholarship awarded by Korea Research Foundation Grant funded by the Korean Government. She is a faculty member of UCSY since 1996. Currently she is a professor and guiding the Ph.D students. Her current research interests are in the areas of Software Aging, Virtualization, Green Computing, Wireless Sensor Networks and Natural Language Processing. E-mail: thandartheinn@gmail.com

