iPlace: An Intelligent and Tunable Power- and Performance-Aware Virtual Machine Placement Technique for Cloud-based Real-time Applications

Faruk Caglar, Shashank Shekhar and Aniruddha Gokhale
Dept of EECS, Vanderbilt University, Nashville, TN 37235, USA
Email: {faruk.caglar, shashank.shekhar, a.gokhale}@vanderbilt.edu

***Abstract*—Power and performance tradeoffs are critical and challenging issues faced by cloud service providers (CSPs) while managing their data centers. On the one hand, CSPs strive to reduce power consumption of their data centers to not only decrease their energy costs but to also reduce adverse impact on the environment. On the other hand, CSPs must deliver performance expected by the applications hosted in their cloud in accordance with predefined Service Level Agreements (SLAs). Not doing so will lead to loss of customers and thereby major revenue losses for the CSPs. Addressing these dual set of challenges is hard for the CSPs because power management and performance assurance are conflicting objectives, particularly in the context of multi-tenant cloud systems where multiple virtual machines (VMs) may be hosted on a single physical server. The problem becomes even harder when real-time applications are hosted in these VMs. To address these challenges and make appropriate tradeoffs, we present iPlace, which is an intelligent and tunable power- and performance-aware VM placement middleware. The placement strategy is based on a two-level artificial neural network which predicts (1) CPU usage at the first level, and (2) power consumption and performance of a host machine at the second level that uses the predicted CPU usage. The efficacy of iPlace is evaluated in the context of a VM consolidation algorithm that is applied to running virtual machines and host machines in a private cloud.**

Keywords*—*virtual machine placement, cloud computing, deployment algorithm, power and performance tradeoffs.

I. Introduction

Cloud data centers are massive-scale farms of networked servers and other resource types, such as storage, that are used to host different kinds of services simultaneously from multiple different customers. Due to the use of commodity hardware for the resources, failures are common within data centers that can cause some disruptions in the hosted services. Another major factor that can cause disruptions in data centers stems from the massive power requirements of the data centers both to operate the hardware as well as for cooling. Power outages due to excess demand can result in substantial disruptions to the data center. For instance, several prominent cloud service providers (CSPs) reported days-long partial or complete outages of their cloud services platform.[1] As an example of the adverse impact this can cause, in 2012 an intrinsic outage in Amadeus airline reservation system's data center triggered long lines and delays at many airports worldwide. Power outages are also shown to have an adverse impact on environment because they produce diesel exhaust.

Reducing energy consumption in data centers is thus an important criteria to reduce the chances of outages. This issue is particularly important considering an increasing trend towards hosting applications with soft real-time requirements in the cloud [1], [2], which cannot sustain significant service disruptions. For these applications, performance concerns, such as response time and service availability, are vital requirements and hence disruptions in data centers is often not desirable.

One promising approach to maintaining availability and performance requirements of real-time applications after partial disruptions within the same data center is via live migration [3] of virtual machines (VMs).[2] VM migrations help in hardware maintenance, fault-tolerance and load-balancing. However, live migration may incur significant cost in terms of substantial network usage particularly when multiple simultaneous VM migrations are active at any given time thereby adding to the energy consumption. One key reason for the increased network usage is that existing approaches that use live VM migrations often tend to ignore the placement issues for the backup VMs, which in turn leads to unwanted usage of network and other resources thereby causing increased energy consumption.

For example, cloud-based high availability and performance solutions such as Remus [4], Paratus [5], and Kemari [6], require capturing the entire executions of the VM and transferring them to the backup machine as swiftly and seamlessly as possible. While these solutions are much desirable for maintaining application performance and availability, they tend to shift the responsibility of choosing the backup VMs to the cloud user. A solution that will relieve the cloud user of these responsibilities and automate the choice of backup VMs is desirable. In prior work [7], [8] we addressed this limitation in the Remus high availability solution by providing a backup VM placement mechanism that was based on simple bin packing heuristics. However, this work did not consider energy conservation as a criteria.

Another technique used by CSPs to improve resource utilization, reduce energy consumption, and thereby saving on energy bills is to employ "Resource Overbooking" [9], [10], [11]. Even in the case of resource overbooking, the placement

[1] A recent incident is reported at www.datacenterknowledge.com/archives/2012/07/10/major-outage-salesforce-com/.

[2] Live migrations may be feasible across data centers but will incur additional and unpredictable networking delays, which may not be suitable for real-time applications.

of VMs on aptly suited host machines where SLA is not violated is crucial. We have observed that even idle VMs that are overbooked on a host machine might affect the performance of applications running in other collocated VMs on that host because of performance interference between collocated VMs [12], [13], [14] when resources are overbooked. Thus, the need for effective VM placement is a key requirement.

In summary, energy conservation in data centers is increasingly becoming the focus of CSPs who are seeking ways to save on energy bills, reduce the chances of outages, and reduce adverse impact on the environment. Reducing energy consumption would imply shutting down large portions of the data center and employing resource overbooking. However, a naive approach to conserving energy may lead to applications not meeting their performance requirements, which is not acceptable to real time applications hosted in the cloud. Techniques that support both performance and availability in the cloud must continue to work. As we have seen above, a common theme that pervades these requirements is the need for effective VM placement in the data center. A common practice for VM placement decisions at the hypervisor level is bin packing heuristics such as first-fit, best-fit, and next-fit. However, these bin packing techniques do not consider power concerns of the CSPs nor performance requirements of applications.

To address these objectives, this paper presents *iPlace,* which is an intelligent and tunable power- and performance-aware virtual machine placement technique that is realized as cloud infrastructure middleware. The key contributions of iPlace include:

- An intelligent tunable power- and performance-aware virtual machine placement strategy in virtualized environments that satisfies soft real-time application QoS. The novelty of our VM placement approach stems from its use of a two-stage neural network which predicts (1) CPU usage at the first level and (2) uses the predicted CPU usage at the first level to predict the power consumption and performance of a host machine at the second level. Section III delves into the details of this contribution.
- It analyzes how energy consumption of data centers can be reduced while performance of soft real-time applications are ensured by employing iPlace. Section IV presents results of our empirical studies.

The remainder of this paper is organized as follows: Section II introduces, compares, and discusses several recent prior efforts synergistic with iPlace; Section III provides an architectural view of the iPlace middleware and the design of the two-stage artificial neural network; Section IV-A presents the test and evaluation results of iPlace; and finally, Section V presents concluding remarks alluding to future work.

II. Related Work

This section explores prior work that employ schemes like live migration and server consolidation techniques that aim to address one or more of the performance, availability and energy consumption issues in cloud data centers.

Akiyama et. al propose MiyakoDori [15] which employs "memory reusing" technique to reduce the amount of memory transferred thereby reducing unnecessary energy consumption during the live migration. When a virtual machine monitor (VMM) initiates a live migration command, MiyakoDori retains the memory image of the VM on the source node. Identical memory pages are not transferred; only the manipulated memory content is transferred when that VM is migrated back to the original node. MiyakoDori saves substantial amount of memory between migrations thereby reducing energy consumption. This related work considers identical memory pages to reduce the energy consumption during live migrations whereas our work focuses on both reducing energy consumption and guaranteeing application performance. It is feasible that our work can leverage MiyakoDori in the live migration process.

Deshpande et al. address the problem of migrating several collocated VMs simultaneously [16]. In a data center, it is highly likely that collocated VMs might have the same operating system, similar software, and libraries installed on it. Therefore, the basic idea in the paper is transferring identical contents across the collocated VMs only once. Our work is once again complementary to this approach since we focus on finding an aptly suited host machine for a VM. Thus, it is possible for our approach to leverage this related work for additional benefits.

The work closest to ours is by Hirofuchi et al. [17], [18] who propose an energy-efficient VM consolidation technique for optimizing VM locations to achieve energy savings while guaranteeing performance. In this work, post-copy live migration is utilized as opposed to pre-copy live migration since post-copy migration reacts to sudden load changes more quickly than pre-copy. Data center servers are categorized as shared and dedicated servers. Shared servers host the idle VMs while dedicated servers host CPU-intensive VMs. Shared servers take advantage of extra memory to host many idle VMs. The technique utilized in their paper is to migrate as many idle machines into shared servers as possible from dedicated servers and finally switch-off the dedicated servers in which no more VMs are left. Our work also comes under the purview of consolidation algorithms. In contrast to this work, our work does not differentiate between shared and dedicated servers, which reduces the complexity of our technique.

Berral et al. [19] propose a framework that provides intelligent dynamic consolidation of VMs in which deadline-sensitive applications are executing. A machine learning-based technique is employed to reduce the energy consumption while meeting SLA requirements for high performance computing (HPC) environments where applications have deadline constraints. This work differs from our work in that their work targets a HPC environment, which are more controlled and where exclusive access to resources is granted is targeted, whereas we primarily target public cloud environments.

Piao [20] proposes a VM placement and migration approach to optimize the heavy data transfer over the network. Due to the nature of network- and data-intensive workloads, applications hosted on various VMs often communicate with each other frequently over the network which in turn adversely affects the application performance and network overhead. Moreover, it might lead to network congestion and unex-

pected network latency. Therefore, migrating these kinds of applications within a close proximity of their counterparts reduces the traffic on the network and ultimately optimizes the performance. The work in that paper differs from our work in that it attempts migrating highly coupled VMs to closeby locations. In contrast, we target compute-intensive applications and discover aptly suited host machine for their VMs with respect to power and performance. As future work, we will consider accounting for network usage as suggested in this prior work.

Khosravi et al. [21] have taken into account carbon footprint rate and power usage effectiveness (PUE) for designing VM placement strategy in data centers. The VM placement problem is considered as a bin packing problem with $(datacenter \times cluster \times host)$ placement options. The authors propose Energy and Carbon-Efficient (ECE) VM placement algorithm based on best-fit heuristic to find a solution to the problem, and evaluate the algorithm using simulation. A difference with our work is that we have evaluated our solution inside a cloud data center and applied machine learning to account for a large set of factors affecting power and performance which are difficult to model in the system. In this related work, the authors considered power consumption as a function of CPU frequency, whereas we have taken several factors including memory, network, overbooking rate etc. into account for predicting performance and power.

Dong et al. [22] propose a VM placement scheme to reduce both the number of physical machines and network elements in a data center to reduce overall energy consumption. The optimization of physical servers is considered as a bin packing algorithm, while network optimization is formulated as a quadratic assignment problem. The proposed method is a combination of hierarchical clustering and best-fit to solve the optimization problem for VM placement and is evaluated based on simulations. In contrast, our work provides an intelligent placement algorithm which considers VM-based host overbooking, power consumption and performance, which is evaluated in a real data center.

III. Virtual Machine Placement using iPlace

Figure 1 depicts the strategy of iPlace, which is our intelligent power- and performance-aware virtual machine placement algorithm. The goal of iPlace is to find an aptly suited host machine by carefully considering the energy efficiency of the data center and performance requirements of soft-real time applications running on host machines. iPlace takes power changes and performance effects to the applications running on VMs for its placement decision. A tunable parameter named *performance preference level* is provided to iPlace in advance to set the performance requirement.

To find the aptly suited host machine, a two-level artificial neural network (ANN) is employed by our VM placement middleware, which are at the core of our system design and serve as the predictor mechanism. To train the ANNs, iPlace employs the Levenberg-Marquardt back-propagation algorithm [23]. At the first level, the mean CPU usage of a host machine after a VM were to be migrated to it is predicted by running the CPU usage predictor ANN. Subsequently, this predicted CPU usage value is utilized by the second level ANN. At the second level, power consumption and mean performance of the host machine is predicted by the power and performance predictor ANN. At runtime, the middleware will consult the prediction engine and if the predicted values are acceptable, the middleware will take the decision of placing the VM on a given host.

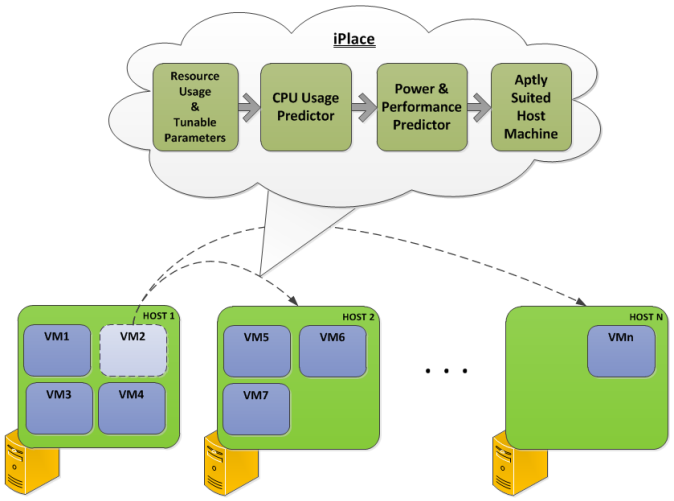


Figure 1: Illustration of iPlace's Virtual Machine Placement Strategy

To understand how these ANNs are used to make runtime decisions, consider the case when one of the consolidation algorithms, high availability solutions, or scheduling mechanisms would like to migrate a VM from one host machine to another one. iPlace finds the aptly suited host machine by predicting the power consumption and performance values for each host machines in the cluster as though the VM was migrated on to it. As illustrated in Figure 1, iPlace employs both CPU usage predictor and power and performance predictor sequentially by feeding their required input values.

In our current design, iPlace targets only compute-intensive applications, therefore $1/(CPUtime)$ metric was utilized in this work as the performance indicator of an application. The higher the performance value, the better the performance. Additionally, we assume that CSPs overbook their underlying cloud infrastructure to save energy costs. Details of the ANNs are described below.

A. CPU Usage Predictor

The structure of the CPU usage predictor ANN is depicted in Figure 2. The purpose of the CPU usage predictor is to estimate the amount of CPU usage of the host machine after a VM were to migrate onto it. Due to the CPU contention in over utilized virtualized environments, the mean CPU usage of a host machine might not increase by the same amount of CPU usage currently been illustrated by the VM being migrated. Thus, a simple subtraction on one machine and addition on another machine does not work. Therefore, iPlace employs a CPU usage predictor ANN for this prediction so that the power consumption and performance of the host machine could be determined effectively by knowing the CPU usage of the host machine.

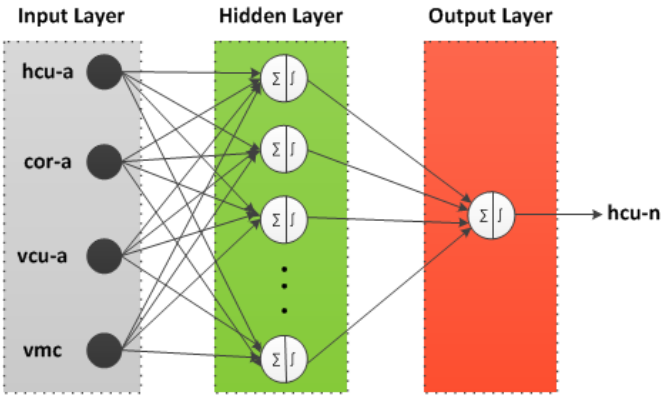


Figure 2: Structure of the CPU Usage Predictor ANN

The topology of the CPU usage predictor ANN is shown below.

Input Layer : $hcu-a, cor-a, vcu-a, vmc$
Hidden Layer : 9 neurons
Activation Function (in hidden layer) : Tangent Sigmoid
Output Layer : $hcu-n$
Transfer Function (in output layer) : Pure Linear

where
$hcu-a$ = Actual mean CPU usage of the host machine before VM is migrated on it
$cor-a$ = Actual CPU overbooking ratio of the host machine before VM is migrated on it
$vcu-a$ = Actual CPU usage of the VM being migrated onto the host machine
vmc = Actual VM count on the host machine before VM is migrated on it
$hcu-n$ = CPU usage of the host machine after VM is migrated onto it

The CPU overbooking ratio and mean CPU usage of the host machine provided to this ANN are computed by Equations (2) and (4).

$$Total\ vCPU\ Requested = \sum_{i=0}^{m} vCPU_i \quad (1)$$

$$CPU\ Overbooking\ Ratio = \frac{Total\ vCPU\ Requested}{Total\ pCPU\ Cores} \quad (2)$$

$$Total\ CPU\ Usage = \sum_{i=0}^{m} vmCPUUsage_i \quad (3)$$

$$Host\ Mean\ CPU\ Usage = \frac{Total\ CPU\ Usage}{m} \quad (4)$$

where
$Total\ vCPU\ Requested$: Total number of virtual CPU cores requested on a host machine
m : Total number of the guest VMs on a host machine
$vCPU$: Number of virtual CPU cores of a VM
$Total\ pCPU\ Cores$: Total number of physical CPU cores of a host machine

The number of neurons in the hidden layer is determined based on experimentation by trying different numbers and examining the system results. The reliability and accuracy of ANNs employed by iPlace is examined by carefully looking into the mean squared error (MSE) and regression (R) values. The MSE value provides average squared difference between input and output whereas the R value describes how the input of the system is correlated with its output. The best performance of the CPU usage predictor ANN was produced with 9 neurons in the hidden layer, and MSE of 0.00044 and R of 0.99. As shown in Figure 3, these MSE and R values clearly indicate that CPU usage predictor precisely estimates the host machine's CPU usage.

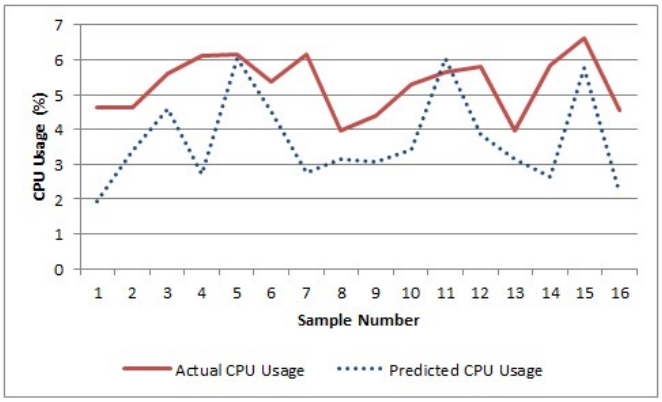
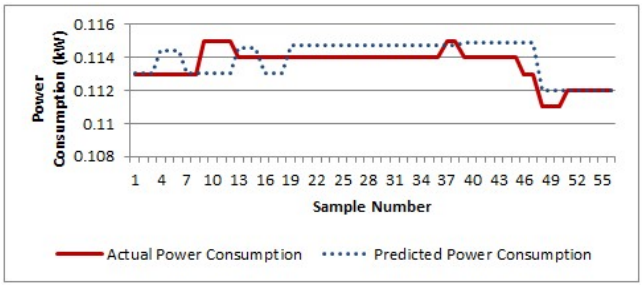


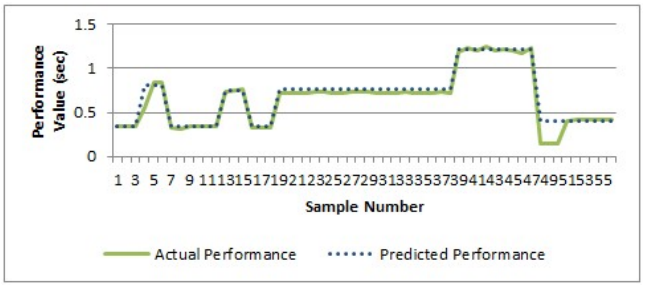
Figure 3: Comparison of Actual and Predicted CPU Usage of Host Machine

B. Power and Performance Predictor

The structure of the power and performance predictor ANN is depicted in Figure 4. The output of the CPU usage predictor ANN is provided as input to the power and performance predictor ANN. The purpose of power and performance predictor is to predict power consumption and performance of the host machine if the VM were to migrate to it.



(a) Power Consumption



(b) Performance

Figure 5: Comparison of Actual and Predicted Power Consumption and Performance Value Results of Host Machine

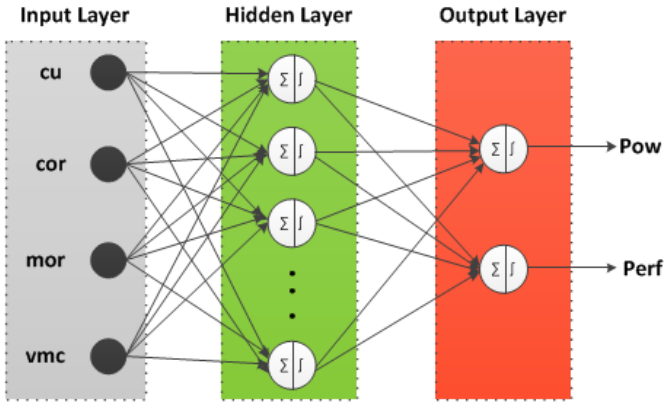


Figure 4: Structure of the Power and Performance Predictor Artificial Neural Network

The topology of power and performance predictor ANN is detailed below.

Input Layer : cu, cor, mor, vmc
Hidden Layer : 12 neurons
Activation Function (in hidden layer)
: Tangent Sigmoid
Output Layer : $Pow, Perf$
Transfer Function (in output layer)
: Pure Linear

where

cu = Mean CPU usage of the host machine
cor = CPU overbooking ratio of the host machine
mor = Memory overbooking ratio of the host machine
vmc = VM count on the host machine
Pow = Power consumption of the host machine
$Perf$ = Performance value of the host machine (*i.e.* mean performance of all the guest VMs running on the host machine)

The best performance of the power and performance predictor ANN was produced with 12 neurons in the hidden layer, MSE of 0.008, and R of 0.97. These MSE and R values clearly show that the power and performance predictor ANN precisely estimates the host machine's power consumption and performance. Figure 5 depicts the comparison of actual power consumption and performance values of host machine along with the predicted values of power and performance predictor.

By carefully observing the data generated by our simulation software, we determined the mean performance value (μ) as 1.75 and the standard deviation (σ) as 1.17. These values are the assumed indicators for performance requirement of the soft real-time application and utilized to check whether the performance requirement of the soft real-time application validated by the Equation (5). This performance indicator is assured on the host machine where it will be migrated with best effort. α in Equation (5) is basically the performance preference level parameter passed by the system user. The tighter performance requirement might cause iPlace not to be able to find any host machine.

$$Pr = \mu + \alpha * \sigma \tag{5}$$

where

Pr : Performance requirement of the VM
μ : Mean performance value computed by looking the values in the cluster
α : Performance preference level
σ : Standard deviation value values in the cluster computed by looking the values in the cluster

This performance parameter along with the resource usage information is provided to iPlace. iPlace employs the CPU usage predictor ANN first and feeds the predicted CPU usage

of the host machine to the power and performance predictor ANN then.

After receiving the CPU usage, iPlace finds all the host machines satisfying the performance requirement in Equation (5) by comparing these predicted performance values with the value returned from Equation (5) by starting from the *performance preference level*. If none of the host machines satisfies the requested *performance preference level*, iPlace gradually lower the *performance preference level* by one and checks each host machine again to find another host machine that will satisfy this new performance requirement. Then, iPlace computes the power change on the each host machine satisfying the performance requirement to see how much power will increase. Finally, the placement decision is made onto the host machine which satisfies the performance requirement and has the minimum power change.

IV. Validating the iPlace Approach

The iPlace framework consists of two stage neural network. For accurate predictions, the training data set for both the stages should be as close as possible to real-world data. To achieve the same, we have used a private data center consisting of five hosts on which we have emulated a workload that is similar to that of a production data released by Google Inc. from one of their cluster's trace log [24]. Our private data center comprises a homogeneous set of machines and is managed by the OpenNebula cloud management solution version 3.2.1. Table I provides the configuration of each host used as a cluster node. Each host is connected to a *Watts Up? Pro* power meter, which can report consumed power with frequency of once a second and an accuracy of a tenth of a watt.

Table I: Hardware and Software specification of Cluster Nodes

Processor	2.1 GHz Opteron
Number of CPU cores	12
Memory	32 GB
Hard disk	8 TB
Operating System	Ubuntu 10.04 64-bit
Hypervisor	Xen 4.1.2
Guest virtualization mode	Para

The Google cluster trace contains a dataset for about 12,000 distinct machines collected over a 29 day period in the month of May, 2011. We chose one of the host with ID 257408495 from the cluster and reproduced the workload on one of the host in our data center for one day. The VM configuration and resource usage in the dataset was normalized which we scaled according to host configuration and pruned the data which did not fit the characteristics of our host.

The workload on the host was generated using our simulation software coded in Python which used OpenNebula to create and delete virtual machines. We executed *Lookbusy*, a synthetic load generator, processes to mimic the CPU and memory workloads. Our resource monitoring application was coded in C++, which runs to collect the resource usage information of the host using *libvirt* library at certain specified interval.

Matlab software is used to train and run the ANNs as well as deciding the placement decision. Additionally, a SQL server database management system is utilized to import the resource information data of each host machine and prepare the training set for ANNs.

A. Experimental Results

In this section we show the experimental results of iPlace that we have tested and evaluated by following two test cases we have defined. The initial configuration of our cluster is depicted in Figure 6. In Figure 6, the tuple under each VM name represents the resource capacity of the VM in the format of <*cpu, memory*>. Additionally, initial resource usage and overbooking ratios of each host machine in the cluster is illustrated in Table II.

Table II: Initial Resource Usage of Host Machines in the Cluster

	CPU Usage	CPU Overbooking Ratio	Memory Overbooking Ratio
HOST 1	16%	4	0.75
HOST 2	18%	0.17	0.06
HOST 3	30%	2.6	0.81
HOST 4	23%	3.3	0.69
HOST 5	1.5%	1.3	0.5

Use Case 1: In this use case, we assumed that there was an abnormal activity causing performance degradation of VM2, which is a high priority VM on Host 1. Therefore, a decision was made to migrate it to another host machine in the cluster. We analyzed and compared the results of placement decision of iPlace with a first-fit heuristic of bin packing algorithm in the context of power and performance. We have tested iPlace with three different performance preference levels (i.e. α values of -1, 0, 1).

Recall that performance preference level is the tunable performance parameter passed by the system user. It is also used in the performance requirement equation 5 with α as the parameter. Based on that performance preference level, the standard deviation is adjusted and performance requirement is either tightened or softened. As the performance preference level goes below 0, -1, -2 ..., the performance requirement of the application is softened and vice versa.

As expected, the placement decision of first-fit heuristic is to place the VM on the first host machine in which it fits. Therefore, a first-fit heuristic placed VM2 on to Host 2. iPlace decides the target host machine by observing power changes on host machines and performance effects on the applications running on the VM. Thus, the placement decision for each performance preference level presented in Table III for this use case might be dissimilar for any other use cases.

As shown in Table III, iPlace decided to migrate VM2 to Host 5 at both performance preference levels of α = -1 and α= 0. iPlace assured the performance requirement of VM2 on none of the host machines in the cluster for the tighter performance preference level of α = 1.

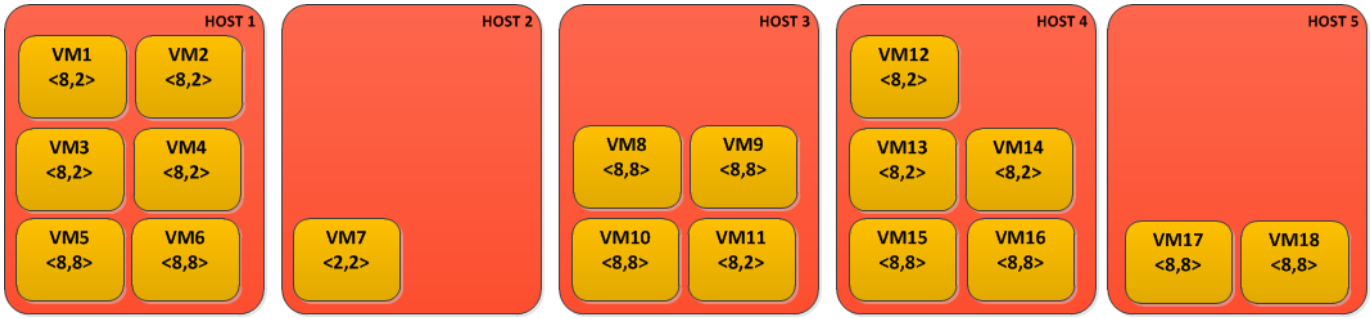


Figure 6: Initial Configuration of the Cluster Utilized in Test Cases

Table III: Test Results of Use Case 1

Performance Preference Level (α)	Placement Decision
-1	HOST 5
0	HOST 5
1	NONE

To detail the case where performance preference level of α =0, iPlace predicted that only Host 3 and Host 5 satisfied the performance requirement in Equation (5). However, iPlace decided to migrate VM2 onto the Host 5 due to the prediction of lower amount of power increase by 0.0274kW on Host 5 versus 0.0281kW on Host 3.

Compared to the first-fit heuristic, iPlace could not assure the performance requirement of VM2 on Host 2 even though VM2 fits on it. Therefore, it discarded Host 2 for its placement decision for VM2.

Use Case 2: In this use case, we assumed that a decision was made to migrate all VMs residing on one of the less utilized host machines onto the rest of the host machines decided by iPlace. Host 2 was selected as the target due to having only one VM. Therefore, VM7 will be migrated and Host 2 will be shut down.

At initial run with performance preference level of α = 0, iPlace could not find a host machine for that criteria. It determined Host 3, Host 4, and Host 5 after iterating till the performance preference level of α = -2. However, iPlace decided to migrate VM7 onto the Host 3 due to the power change concerns.

After migrating VM7 onto the Host 3, Host 2 becomes idle and started to consume 0.091kW power with no VMs running on it. Therefore, we assumed it was turned off and computed the overall power consumption of the cluster. The total power consumption of the cluster dropped from 0.708kW to 0.614kW which saved about 13% of power consumed by the cluster.

V. Conclusion

In this paper we proposed iPlace, an intelligent tunable power- and performance-aware virtual machine placement strategy. The work was motivated by the need to conserve energy in data centers yet manage the performance and availability requirements of soft real-time applications that are increasingly being hosted in cloud data centers. To that end, we have developed a two-level artificial neural network (ANN) with stage one responsible for CPU usage prediction, and stage two responsible for power and performance prediction. The two stage ANN was designed, trained and employed to forecast the host machine's CPU usage, power consumption, and performance. For training purposes and evaluation, we generated workloads in our private cloud that emulated data from a Google's production server. We have tested and evaluated iPlace in our private cloud and compared results with first-fit bin-packing heuristic. The results shows that iPlace could help to save certain degrees of power consumption by satisfying variety of performance requirements. Compared to the first-fit heuristic, iPlace places VMs on host machines where application performance is assured and energy efficiency is maximized.

Since the private cloud environment where we tested and evaluated iPlace is a homogeneous environment, our test results were validated only in a homogeneous environment. However, iPlace could easily be employed in a heterogeneous environment by providing additional host machine capacity information parameters to the ANNs, such as number of cores and memory size. In this work, we targeted only the compute-intensive applications due to the performance metric we utilized. By integrating more generic application performance metrics, such as response time or throughput, iPlace could support a variety of application types in the cloud environment. These dimensions will form the basis of our future work.

The source code for iPlace is available for download at www.dre.vanderbilt.edu/~caglarf/download/iPlace.

Acknowledgments

This work was supported in part by the National Science Foundation NSF CAREER CNS 0845789 and DDDAS FA9550-13-1-0227. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] T. M. Takai, "Cloud computing strategy," DTIC Document, Tech. Rep., 2012.

[2] A. Corradi, L. Foschini, J. Povedano-Molina, and J. M. Lopez-Soler, "Dds-enabled cloud management support for fast task offloading," in *Computers and Communications (ISCC), 2012 IEEE Symposium on*. IEEE, 2012, pp. 000 067–000 074.

[3] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 273–286.

[4] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, and A. Warfield, "Remus: High availability via asynchronous virtual machine replication," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. USENIX Association, 2008, pp. 161–174.

[5] Y. Du and H. Yu, "Paratus: Instantaneous failover via virtual machine replication," in *Grid and Cooperative Computing, 2009. GCC'09. Eighth International Conference on*. IEEE, 2009, pp. 307–312.

[6] Y. Tamura, K. Sato, S. Kihara, and S. Moriai, "Kemari: Virtual machine synchronization for fault tolerance," *In USENIX 2008 Poster Session*, 2008.

[7] K. An, F. Caglar, S. Shekhar, and A. Gokhale, "Automated Placement of Virtual Machine Replicas to Support Reliable Distributed Real-time and Embedded Systems in the Cloud," in *International Workshop on Real-time and Distributed Computing in Emerging Applications (REACTION), 33rd IEEE Real-time Systems Symposium (RTSS '12)*. San Juan, Puerto Rico, USA: IEEE, Dec. 2012.

[8] K. An, S. Shekhar, F. Caglar, A. Gokhale, and S. Sastry, "A Cloud Middleware for Assuring Performance and High Availability of Soft Real-time Applications," *Submitted to Special Issue of REACTION 2012 Workshop, Elsevier Journal of Systems Architecture (JSA)*, 2014.

[9] F. Caglar and A. Gokhale, "iOverbook: Managing Cloud-based Soft Real-time Applications in a Resource-Overbooked Data Center," in *Submitted to the 20th IEEE Real-time and Embedded Technology and Applications Symposium*. Berlin, Germany: IEEE, Apr. 2014.

[10] S. A. Baset, L. Wang, and C. Tang, "Towards an understanding of oversubscription in cloud," in *Proceedings of the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*. USENIX Association, 2012, pp. 7–7.

[11] (2013, Sep.) Best practices for oversubscription of cpu, memory and storage in vsphere virtual environments. [Online]. Available: https://software.dell.com

[12] F. Caglar, S. Shekhar, and A. Gokhale, "A Performance Interference-aware Virtual Machine Placement Strategy for Supporting Soft Real-time Applications in the Cloud," Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA, Tech. Rep. ISIS-13-105, 2013.

[13] R. Nathuji, A. Kansal, and A. Ghaffarkhah, "Q-clouds: managing performance interference effects for qos-aware clouds," in *Proceedings of the 5th European conference on Computer systems*. ACM, 2010, pp. 237–250.

[14] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell, "Modeling virtual machine performance: challenges and approaches," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 3, pp. 55–60, 2010.

[15] S. Akiyama, T. Hirofuchi, R. Takano, and S. Honiden, "Miyakodori: A memory reusing mechanism for dynamic vm consolidation," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 2012, pp. 606–613.

[16] U. Deshpande, X. Wang, and K. Gopalan, "Live gang migration of virtual machines," in *Proceedings of the 20th international symposium on High performance distributed computing*. ACM, 2011, pp. 135–146.

[17] T. Hirofuchi, H. Nakada, S. Itoh, and S. Sekiguchi, "Making vm consolidation more energy-efficient by postcopy live migration," in *CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization*, 2011, pp. 195–204.

[18] ——, "Reactive consolidation of virtual machines enabled by postcopy live migration," in *Proceedings of the 5th international workshop on Virtualization technologies in distributed computing*. ACM, 2011, pp. 11–18.

[19] J. L. Berral, Í. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, "Towards energy-aware scheduling in data centers using machine learning," in *Proceedings of the 1st International Conference on energy-Efficient Computing and Networking*. ACM, 2010, pp. 215–224.

[20] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*. IEEE, 2010, pp. 87–92.

[21] A. Khosravi, S. K. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Euro-Par 2013 Parallel Processing*. Springer, 2013, pp. 317–328.

[22] J. Dong, X. Jin, H. Wang, Y. Li, P. Zhang, and S. Cheng, "Energy-saving virtual machine placement in cloud data centers," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*. IEEE, 2013, pp. 618–624.

[23] J. J. More, "The Levenberg-Marquardt Algorithm: Implementation and Theory," in *Numerical Analysis*, ser. Lecture Notes in Mathematics, G. Watson, Ed. Springer Berlin Heidelberg, 1978, vol. 630, pp. 105–116. [Online]. Available: http://dx.doi.org/10.1007/BFb0067700

[24] C. Reiss, J. Wilkes, and J. Hellerstein, "Google cluster-usage traces: format+ schema," *Google Inc., White Paper*, 2011.