ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

클라우드 데이터 센터에서 가상화된 자원의 SLA-Aware 조정을 통한 성능 및 에너지 효율의 최적화☆

Optimizing Performance and Energy Efficiency in Cloud Data Centers Through SLA-Aware Consolidation of Virtualized Resources

프랭크 엘리호데[1] 이 재 완[2*]
Frank I. Elijorde Jaewan Lee

요 약

클라우드 컴퓨팅은 사용자의 요구에 따라 IT서비스가 생성 및 조정되는 pay-per use 모델을 도입하였다. 그러나 서비스 제공자는 아직도 물리적인 인프라로 인해 발생하는 제약조건들에 대해 관심을 갖고 있다. 필요한 QoS나 SLA를 만족시키기 위해서는 가상화된 자료들이 에너지 소비량을 최소화시키면서 시스템 성능을 최대화시키기 위해 조정되어야 한다. 본 연구는 ANN을 사용하여 클라우드 환경에서 가상화된 자원들을 조정하기 위한 예측적 SLA 어웨어 방안을 제시한다. Qos를 유지하고, 성능과 에너지 효율간의 최적화를 위해서 서버 활용 임계치는 물리적 자원의 소비에 따라 동적으로 적용한다. 또한 많은 자원을 소비하는 VM들은 능력있고 평판이 좋은 호스트에 할당함으로써 부족한 프로비전닝을 방지한다. 제안한 기법의 성능을 평가하기 위해, 이질적인 클라우드 환경에서 최적화되지 않은 전통적인 접근방법 및 기존의 기법들과 비교하였다.

☞ 주제어 : 클라우드 컴퓨팅, 클라우드 데이터센터, 인공신경망, 자원 프로비저닝, 그린 컴퓨팅

ABSTRACT

The cloud computing paradigm introduced pay-per-use models in which IT services can be created and scaled on-demand. However, service providers are still concerned about the constraints imposed by their physical infrastructures. In order to keep the required QoS and achieve the goal of upholding the SLA, virtualized resources must be efficiently consolidated to maximize system throughput while keeping energy consumption at a minimum. Using ANN, we propose a predictive SLA-aware approach for consolidating virtualized resources in a cloud environment. To maintain the QoS and to establish an optimal trade-off between performance and energy efficiency, the server's utilization threshold dynamically adapts to the physical machine's resource consumption. Furthermore, resource-intensive VMs are prevented from getting underprovisioned by assigning them to hosts that are both capable and reputable. To verify the performance of our proposed approach, we compare it with non-optimized conventional approaches as well as with other previously proposed techniques in a heterogeneous cloud environment setup.

☞ keyword : Cloud Computing, Cloud Data Centers, Artificial Neural Network, Resource Provisioning, Green Computing

1. Introduction

There is an ongoing research interest to develop more flexible data centers which enable consolidated application platforms to seamlessly share resources from the same server. Such flexibility is an enabling factor in today's cloud computing infrastructures which aims to efficiently provision compute resources to multiple clients over the internet. Regulating the power consumption to reduce the cost of a computing infrastructure as well as to lessen its environmental effect has also become a very important consideration. According to the US Environment Protection Agency (EPA), the energy usage in cloud data centers is successively doubling every five years. Unfortunately, the average server utilization in many data centers is low, estimated to be only between 5% and 15% [1]. Underutilized resources would further add up to this waste because an idle server often consumes more

[1] Institute of ICT, West Visayas State University, Philippines
[2] Dept. of Information and Communication Engineering, Kunsan National University, Korea
* Corresponding author (jwlee@kunsan.ac.kr)
[Received 7 January 2014, Reviewed 13 January 2014, Accepted 7 April 2014]
☆ This research is partially supported by the Institute of Information and Telecommunication Technology of KNU

than 50% of its peak power [2], which means that a number of servers at low utilization consume significantly more energy than fewer servers at high utilization. The cause of the unreasonably high energy consumption is not just the quantity of computing resources and the power inefficiency of hardware, but most importantly the inefficient utilization of these resources.

Seeking ways to maintain the quality of service and uphold the Service Level Agreement (SLA) as well as to establish an optimal trade-off between performance and energy efficiency, we propose a resource provisioning approach for cloud systems which monitors and allocates compute resources in an adaptive manner using the ANN model. Effective consolidation of virtualized resources is achieved by enabling the server's utilization threshold to dynamically adapt to the physical machine's resource consumption as predicted by the system. Moreover, resource-intensive VMs that are prone to load spikes are protected from underprovisioning by assigning them to hosts that are both capable and reputable. Evaluation results from our simulated cloud environment setup confirm the superiority of our proposed approach over non-optimized conventional approaches and other previously proposed techniques.

2. Related Work

2.1 SLA Management in Cloud-based Systems

SLAs have become increasingly important in today's cloud computing scenario because they define the terms and conditions for the provisioning and delivery of services to its consumers. Thus, upholding the SLA aims to prevent violations which results to costly penalties that the provider has to pay in cases of service degradation. In [3], they proposed and evaluated a dynamic server consolidation algorithm to reduce the amount of required capacity and SLA violation rate. The algorithm uses historical data to forecast future demand and relies on periodic executions to minimize the number of physical servers to support the virtual machines. In [4], SLA requirements are represented as the pre-determined response times for each type of transactions specific to the web-application. Based on the utility function, the migration controller decides whether an effective reconfiguration is possible in order to fulfill the SLA. The problem of dynamic consolidation of VMs running multi-tier web applications to optimize a global utility function, while meeting SLA requirements was investigated in [5]. The system adjusts the placement of VMs and the states of the hosts whenever the request rate deviates from the allowed interval. In the approach proposed in [6], they applied control loops to manage resource allocation under response time constraints at the cluster and server levels. If the server's available resource is insufficient to meet the applications' SLAs, a VM is migrated from the server. Our study is similar to [7] in the sense that we both employ VM placement and VM selection techniques for keeping the SLA violation at a low level. We go further by incorporating machine learning for the prediction of the VM's resource utilization therefore allowing the host to anticipate incoming resource consumption. Although there has been a considerable amount of work which considered the development of flexible and self-manageable cloud computing infrastructures, we believe that there is a lack of adequate monitoring schemes which can predict and prevent possible SLA violations.

2.2 VM Consolidation in Cloud Data Centers

VM consolidation itself is a big challenge especially that it involves various constraints such as performance, scalability, availability, network, and cost. The authors in [8] describe a system that combines various mechanisms for allocating virtual machines to physical hosts. The allocation process is done by considering three resources: the RAM, the number of CPU cores and the type of architecture. In [9] they developed a system which monitors and detects hotspots and initiates VM migration whenever a certain metric goes beyond the threshold for a given time and the next predicted value also exceeds the threshold. Another solution in [10] is a resource manager for homogeneous clusters which performs dynamic consolidation based on constraint programming and takes migration overhead into account. It assumes that the resource demand is known in advance. While in [11], they introduce a VM placement strategy based on the idea of equivalent capacity to consolidate an increased number of VMs for each server. In [12] they introduced some VM consolidation constraints for limiting the number of virtual machines in a physical server, assignment of some virtual machines to

different physical servers, mapping virtual machines to a specific set of physical servers, and limiting the total number of migrations for dynamic consolidation.

2.3 Energy Efficient Cloud Data Centers

One of the earlier works in this related field is discussed in [13] which provide energy distributed accounting on vertical structured OS with Virtual machines. They provide a framework for managing energy in multilayered operating system and accounts recursive energy consumption spent in virtualization layer of driver components. Over the last years, several researchers have addressed this problem by the well-known technique of Dynamic Voltage and Frequency Scaling (DVFS) [14] or by the introduction of heuristics to minimize power consumption [15]. The former can effectively reduce the dynamic power of the system, while the latter minimizes the total power of the data center. With regards to dynamic resource provisioning in virtualized servers, they aimed to manage VM configurations to minimize the number of physical hosts needed to support the computing environment[16]. However, they did not consider switching-off unused machines to save power. In [17], they described an approach for allocating virtual machines to physical servers with a focus on energy saving. Their approach is improved by utilizing an over-provisioning method that addresses the variability related to the resource requirements of the applications. Moreover, a study is conducted in [18]to understand how application type and the heterogeneity of servers impact the energy efficiency of data centers.

3. SLA-Aware and Energy Efficient Consolidation of Virtualized Resources

3.1 Resource Utilization Prediction and SLA Management

In a cloud environment, compute resources are provided using a pay-as-you-go flexible charging. As such, resource demand is considered more unpredictable as compared with traditional IT environments. A key challenge for cloud service providers is to automate the management and allocation of virtual resources while at the same time considering both the SLA requirements of the hosted services and resource management costs. To address this, we propose to integrate with the cloud system an SLA management mechanism capable of managing and predicting the VM resource utilization levels in the host machines. Using Artificial Neural Networks (ANN), the prediction can be dynamically tuned according to actual usage of the cloud infrastructure.

The pattern recognition ability of ANN makes it an excellent tool for classification and forecasting in various applications. In this work, multi-layer neural network perceptrons have been applied to predict the volume of resources to be consumed by the VMs. Periodically, the utilization history of the VMs from the previous hour is derived which is then used to predict the amount of resources that need to be consolidated by the host machine in the succeeding periods. As shown in **Figure 1,** the ANN model used in this study is the standard three-layer feedforward network. Since the one-step-ahead forecasting is considered, only one output node is employed.

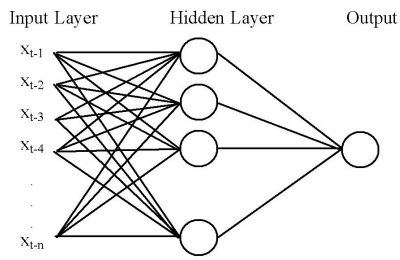

Figure 1.The ANN model.

Through the input layer, the ANN model receives 12 inputs *x(n)* composed of the VM's resource utilization level for 1 hour divided at an interval of 5 minutes. Each neuron comprises two units. The first unit sums up the products of weights coefficients and input signals; while the second one implements a nonlinear neuron activation function. The desired response vector is obtained at the output layer of the computation nodes. As shown **in Figure 2**, each connection is modified by a weight, and each node has an extra input assumed to have a constant value of 1. The weight that modifies this extra input is called the bias.

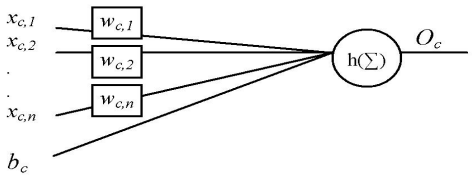

Figure. 2. A neural network node with inputs, weights, and bias.

When the network is run, each layer performs the calculation on the input and transfers the result *Oc* to the succeeding layer.

$$O_c = h(\sum_{i=1}^{N} x_{c,i} w_{c,i} + b_c) \quad (1)$$

where $h(x)$

$$= \begin{cases} \frac{1}{1+e^{-x}}, & hidden\ layer\ node \\ x, & output\ layer\ node \end{cases} \quad (2)$$

$$w_{c,i} = (learning\ parameter\ n) * (local\ gradient) * (input\ signal) \quad (3)$$

In equation 1, O_c is the output of the current node, n is the number of nodes in the previous layer, $X_{c,i}$ is the input of the current node from the previous layer, b_c is the bias and $W_{c,i}$ is the modified weight based on the error which is derived using Equation 3. Meanwhile, $h(x)$ is a sigmoid activation function for the hidden layers Activation function.

We train the ANN model in a supervised manner using a back-propagation algorithm. Many variants of back-propagation training algorithm were developed; in our case we adopted the Resilient Back-propagation technique [19]. The reason behind this is that it can combine fast convergence, stability, and generally produce good results. It differs to the traditional back-propagation in the sense that the effort of adaptation is not distorted by gradient behavior; instead it only depends on the sign of the derivative instead of its value.

After the neural network had been trained within tolerable error, the remaining workload trace data was fed and the Mean Squared Error (MSE) is calculated from the predicted values. The calculation of the MSE is shown below, where d and $\overline{d}$ are the actual and predicted utilization levels respectively, and N is the number of samples.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(d - \overline{d})^2 \quad (4)$$

Figure 3 shows an example of how the ANN model was applied to predict resource utilization levels using historical data. As can be seen, the utilization patterns indicate a close match between the actual and the predicted resource utilization levels.

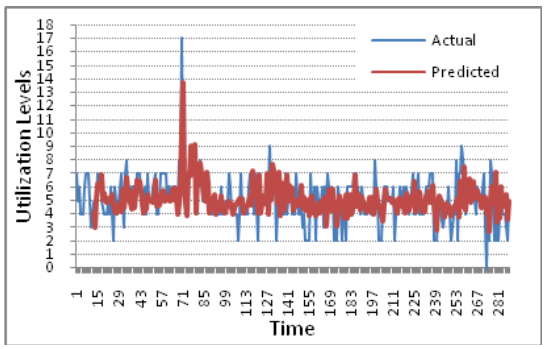

Figure 3. Comparison of the predicted and actual VM utilization levels.

3.2 Adaptive Utilization Thresholds

After a VM is instantiated, a resource monitoring mechanism needs to keep track of the physical machine's utilization level in relation to the actual resource usage of the VMs being hosted. To attain this, a number of considerations have to be met. First, we need to know whether a host is overloaded which would require migration to a less loaded host. Similarly, an underloaded host also needs to migrate its VMs to another host so it can be put to a low-power mode.

In an environment where heterogeneous services share the same physical resources, workloads are highly dynamic; this makes fixed utilization thresholds unsuitable. For this reason, we devise an adaptive method which uses time series data of the host's most recent utilization history and the predicted utilization level. Both the actual and predicted values should be taken into account; thus, the upper utilization threshold is derived as:

$$T_U = u - \left[\frac{(\frac{1}{N}\sum_{i=1}^{N} x_i)}{u} * s \right] \quad (5)$$

$$\text{where } u = \begin{cases} u_p, & u_p \geq \frac{1}{N}\sum_{i=1}^{N} x_i \\ (\frac{1}{N}\sum_{i=1}^{N} x_i), & otherwise \end{cases}$$

In the equation, {x1, x2,..,xN} are the observed values of the host's utilization history, while u_p is the predicted utilization level for the given N samples. The parameter s is a value which influences the tradeoff between quality of service and energy efficiency. As shown in **Figure 4**, setting the value higher would result to a lower SLA violation rate although at the expense of a higher energy consumption. The implication of this observation is that a host configured to strictly observe low SLA violation rate would tend to maintain a low utilization level in exchange for better performance. This behavior would require the data center to keep a higher number of active servers to host the virtual machines, leading to higher energy consumption.

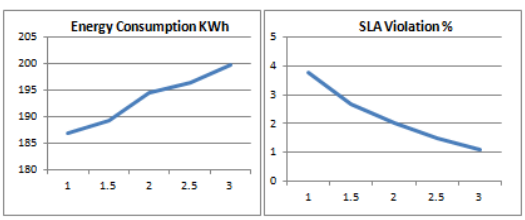

Figure 4. The effect of tweaking parameter s on the energy consumption and SLA violation.

Once the upper threshold has been determined, the next step is to derive the lower threshold. To obtain the lower threshold, we use the following equation:

$$T_L = T_U - (T_U * p) \quad (6)$$

The parameter p is used to determine the gap between the upper and lower threshold. An interesting observation is that the distance between the thresholds also has a significant impact on the performance of the cloud system's provisioning mechanism. Part of the goals of this study is to find the most optimal value and combination for parameters s and p in order to set an excellent balance between performance and energy efficiency.

3.3 Virtual Machine Migration Strategy

With the goal of minimizing the overhead during migration, we propose an approach which considers the VM's current and predicted resource utilization pattern. The predicted utilization level u_p is used as parameter for monitoring the consolidation of virtualized resources. We look for the VM i with the highest difference between the current utilization c_u and the predicted utilization:

$$max(c_u - u_p)i | c_u > u_p \quad (7)$$

These calculations are then used in the VM selection process. As shown in **Algorithm 1**, the process starts by enumerating the hosts that need to perform migration according to decreasing utilization levels. Each host will then have their respective VM lists traversed, in which the variables *utilDifference* and *maxVM* are updated in each iteration should the function *GetUtilDifference (vm)* generate a new maximum value. The update process resumes until the algorithm has inspected all VMs, which in return appends the VM with the highest utilization difference to the migration list. The same process applies on the remaining hosts until the VM selection routine is terminated and the final VM migration list is returned.

```
Algorithm: Maximum Utilization Difference
Input: HMigList, //host migration list
Output: VMList //VM migration list

1.  Sort(HMigList, utilization)//sort hosts,
    decreasing utilization
2.  For each host in HMigList {
3.  utilDifference = Max
4.  For each vm in host{
5.      difference = GetUtilDifference(vm)
6.  ifdifference>utilDifference
7.  {
8.  utilDifference = difference
9.  maxVM=vm
10.     }
11. VMList.Add(maxVM)
12.   }
13. }
14. ReturnVMList
```

Algorithm 1. The VM Migration Strategy.

4. Simulation and Evaluation Results

4.1 Simulation Setup

To evaluate our proposed approach, it needs to be deployed in an environment which emulates the cloud computing paradigm. Regarding the simulation platform, we used CloudSim toolkit [20] which is a simulation framework made in Java. After we modified and extended parts of the simulator, we implemented our proposed approach and performed extensive simulation using 10 days worth of resource usage data by more than a thousand PlanetLab[21] VMs provisioned for multiple users. The simulated data center is set by using realistic models of VM instances and host machines. For the 400 VM instances, we used 4 types of VM instances with characteristics similar to the Amazon EC2 instance types [22] shown in **Table 1.**

Table 1. VM instances specification.

Instance Type	CPU (1 compute unit = 1.0 Ghz)	RAM (GB)
M1 Small Instance	1 core with 1 EC2 Compute Unit	1.7
M1 Medium Instance	1 core with 2 EC2 Compute Units	3.75
M1 Large Instance	2 cores with 2 EC2 Compute Units each	7.5
High-CPU Medium Instance	2 cores with 2.5 EC2 Compute Units each	1.7

For the heterogeneous data center setup, we considered 100 physical machines which were equally distributed among the two types of servers with specifications and power (in Watts) consumptions derived from [23] and [24] as shown in **Table 2**. The first variant is HP ProLiant DL380 G7 (6 cores, Intel Xeon X5675 3.07 GHz processor, 12GB RAM) with 2 CPUs enabled. The other is IBM System X3550 M3 (6 cores, Intel Xeon X5670 2.9 GHz processor, 12GB RAM) with 2 CPUs enabled. Both servers were configured with 1000 Mb network bandwidth.

Table 2. Server power consumption at varying loads.

VM Instance	Target Load(%)									
	100	90	80	70	60	50	40	30	20	10
DL380 G7	222	199	180	163	147	136	126	116	106	93.6
X3550 M3	247	229	211	191	173	156	143	131	120	107

4.2 Evaluation Results

Extensive experiments on different values for the parameters discussed in Section 3.2 were done to determine the optimum configuration for our proposed approach. Based on the results shown in **Figure 5** and **Figure 6** we conclude that the best configuration for the parameters p and s are 0.7 and 3.0 respectively. Setting parameter p to a value higher than 0.7 would increase the gap between the upper and lower utilization thresholds resulting to a significantly reduced lower threshold which affects the algorithm's judgment towards underutilized hosts, resulting to fewer VM migrations, fewer host shutdowns, and disproportionately high energy consumption.

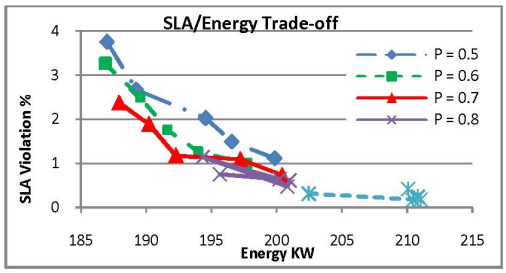

Figure 5. Relationship between SLA violation and energy consumption based on parameter *p*.

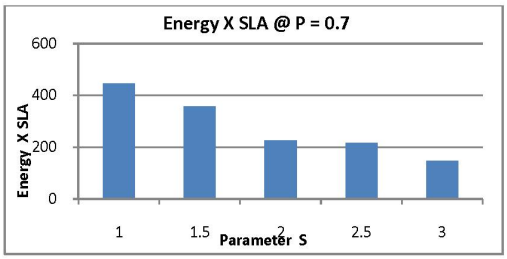

Figure 6. Result of combining SLA violation and energy consumption based on parameter *s*.

After we derive the best configuration for our proposed scheme, we evaluate its performance and compare it with other methods presented in [25]. The methods chosen for comparison are: a) The Non Power-Aware (NPA) policy, which does not employ energy efficient techniques and assumes 100% CPU host utilization thereby consuming maximum power at any given instance. b) Dynamic Voltage and Frequency Scaling (DVFS), which uses dynamic voltage scaling to reduce the energy consumption of hosts. c) Threshold-Based (THR) approach, which requires setting the upper limit for host utilization and keeping the total CPU utilization below such threshold. d) Random Selection (RS), which randomly selects a number of VMs and migrating it to less loaded hosts. e) Median Absolute Deviation (MAD), which uses residuals from the CPU utilization data's median. The derived value is then used to set the upper utilization threshold for detecting overloaded hosts. f) Inter Quartile Range (IQR), using the given CPU utilization history it measures the dispersion of data which is used to decide on host overloading. g) Local Regression (LR), which builds a trend line that estimates the next observation for the CPU utilization which will decide if a host is overloaded. h) Local Regression Robust (LRR), an improved version of LR made resilient against outlier data.

Methods e to *f* use the Minimum Migration Time Policy (MMT) policy to migrate a VM which requires the least time to complete a migration compared to other VMs hosted by the physical server. As for the evaluation metrics, we compare our work with the aforementioned methods in terms of energy consumption, SLA violation rate, number of VM migrations, VM performance degradation, number of host shutdowns, and the *Energy X SLA* combination.

We show in **Figure 7** the result of evaluating the energy consumption of the given resource provisioning techniques. Expectedly, the NPA approach has the highest energy consumption at around 570 then followed by DVFS at around 540. All the rest including our proposed Adaptive Threshold Prediction (ATP) consumed the least energy at around 200. The results presented in **Figure 8** show that the ATP approach has the lowest overall SLA violation rate at around 0.70%, followed by MAD approach at about 1.8%. This implies that in the entire operation of the data center, the ATP approach performed best and was able to deliver the agreed SLA level at 99.30%. This is attributed to its ability to predict the incoming VM utilization levels which enables it to provision the right amount of resources. On the other hand, the THR scheme has a high SLA violation rate at about 4.75% which is quite far from its counterparts. Looking at the LR scheme, it generated a lower SLA violation rate than LRR for the reason that it is more reactive to sudden load variations while LRR is more focused on smoothing out the utilization data prior to load analysis. For the succeeding evaluation results, we omit NPA and DVFS since the metrics presented does not apply to them. This is because both approaches have no capabilities to dynamically optimize resource allocation, as well as to monitor SLA violations and energy consumption.

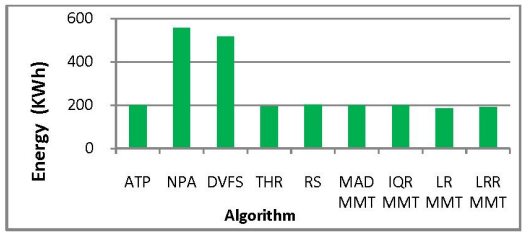

Figure 7. Total energy consumption.

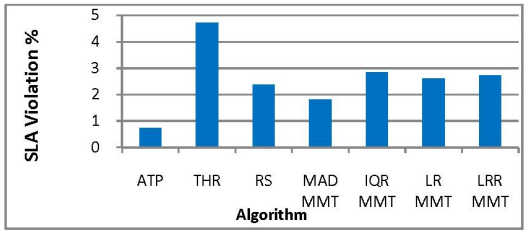

Figure 8. Overall SLA violation rate.

Shown in **Figure 9** is the number of VMs migrated during the entire operation of the simulated data center. On the said metric, ATP has the least number of migrated VMs, followed by LR, while THR has the most migrations at more than 20000. ATP had the least number of VM migrations due to its adaptive threshold mechanism. Through this, resource utilization among physical hosts is optimized therefore eliminating unnecessary or premature VM migrations. Meanwhile, we can see that THR has the most aggressive behavior with regards to migration which is due to its fixed utilization threshold. In **Figure 10** we show each method's respective performance degradation whenever a VM migration is executed. Looking at the figure, it is consistent with the trend shown in **Figure 8** which confirms that VM migrations cause SLA violations. Although

ATP doesn't have the lowest VM degradation rate per migration, its superiority over other methods particularly LR and LRR is justified by its minimal migration resulting to a very low SLA violation rate.

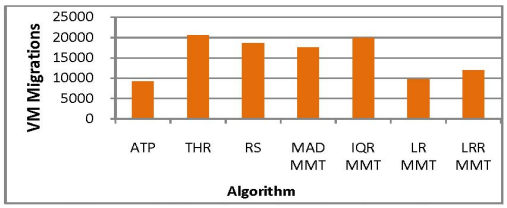

Figure 9. Number of VMs migrated.

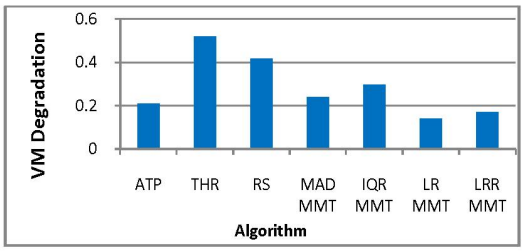

Figure 10. VM performance degradation.

The number of host shutdowns for each approach are compared in **Figure 11**. As shown, both non-power-aware schemes have the lowest results at around 50, followed by ATP at about 350, while THR and IQR are the highest at around 2600. Considering only the power-aware techniques, even if the ATP approach has the least number of host shutdowns its energy consumption is comparable to those of its counterparts. The reason behind this is its efficient resource provisioning mechanism combined with a predictive technique for monitoring resource utilization. With such characteristics, SLA violation is kept at a very low rate while avoiding excessive power cycling which could reduce its reliability. This shows us the importance of optimizing resource allocation in order to balance system performance and energy consumption.

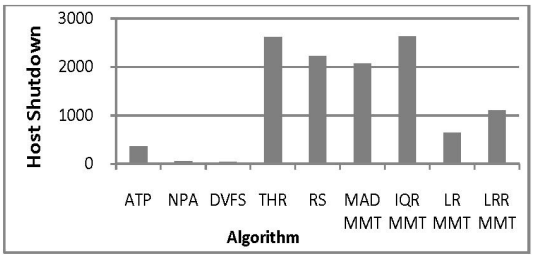

Figure 11. Number of host shutdown

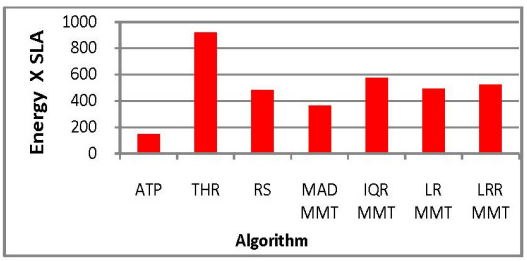

Figure 12. Energy and SLA combination

At this point, we compare the performance of the various approaches using the metric which combines energy efficiency and service quality. As shown in Figure 12, ATP has the best result at around 140, followed by RS at about 490, and the worst is that of THR at more than 900. Despite the similarity in the total energy consumption of ATP and other power-aware techniques in Figure 7, the low Energy X SLA value of ATP is mainly attribtued to its remarkably low SLA violation rate which is less than 1%. This gives us a clear idea that the manner in which virtualized resources in a cloud data center are provisioned and utilized is extemely important towards efficient VM consolidation.

5. Conclusion

The efficient management and allocation of virtualized resources in a cloud system is not a trivial task. This is for the reason that clients usually acquire resources on-demand which results to a highly dynamic workload on the part of the cloud provider. In order to maintain the required quality of service, it is especially important for the cloud provider to avoid SLA violations.

The aforementioned concerns where tackled in this paper using techniques that efficiently consolidate virtualized resources to maximize system throughput as well as to keep energy consumption at a minimum. Using ANN, the server's utilization threshold can dynamically adapt to the physical machine's resource consumption based on predicted utilization levels. Resource-intensive VMs that are bound to shoot up their resource consumption level are protected from underprovisioning by having them hosted by servers that are both capable and reputable. We verified the performance of our work by comparing it with non-optimized conventional methods and also with previously proposed techniques in a heterogeneous cloud environment setup. The superiority of our approach is emphasized by its *Energy* X SLA value which is very well compensated by its remarkably low SLA violation rate. From these results, we conclude that upholding the SLA by means of minimizing service disruption is indeed a huge contributor in the trade-off between the performance and energy consumption of a cloud data center.

References

[1] U. S. Environmental Protection Agency, Report to congress on server and data center energy efficiency public law 109-431. *Technical report, EPA ENERGY STAR Program*, 2007.

[2] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms", *In SIGMETRICS*, pp. 157–168, 2009.

[3] N. Bobroff, A. Kochut, and K.A. Beaty, "Dynamic placement of virtual machines for managing sla violations", *In Proc. of the 10th IFIP/IEEE International Symposium on Integrated Network Management*, 2007.

[4] G. Jung, K.R. Joshi, M.A. Hiltunen, S.D. Schlichting, and C. Pu, "A cost-sensitive adaptation engine for server consolidation of multi-tier applications", *Proc. of the 10th ACM/IFIP/USENIX International Conference on Middleware*, pp.1–20, 2009.

[5] G. Jung, M. A. Hiltunen, K. R. Joshi, R. D. Schlichting, and C. Pu, "Mistral: Dynamically managing power, performance, and adaptation cost in Cloud infrastructures", *In Proc. of the 30th Intl. Conf. on Distributed Computing Systems*, pp. 62–73, 2010.

[6] X. Wang and Y. Wang, "Coordinating power control and performance management for virtualized server clusters," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, pp.245–259, 2011.

[7] A.Beloglazov, J. H. Abawajy, R.Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing", *Future Generation Comp. Syst. (FGCS)*,pp. 755-768, 2012.

[8] R. Nielsen, C. Iversen, and P. Bonnet, "Private Cloud Configuration with MetaConfig", *Proc. for IEEE 4th International Conference on Cloud Computing*, 2011.

[9] T. Wood, P. J. Shenoy, A. Venkataramani, and M. S. Yousif, "Black-box and gray-box strategies for virtual machine migration", *In Proc. of NSDI*, 2007.

[10] F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall, "Entropy: a consolidation manager for clusters", *In Proc. of VEE*, 2009.

[11] M. Chen, H. Zhang, Y.-Y. Su, X. Wang, G. Jiang, and K. Yoshihira, "Effective VM sizing in virtualized data centers", *In Proc. of the IFIP/IEEE International Symposium on Integrated Network Management*, 2011.

[12] M. Bichler, T. Setzer, and B. Speitkamp, "Capacity planning for virtualized servers", *In Proc. of the 16th Annual Workshop on Information Technologies and Systems*, 2006.

[13] J. Stoess and L. Bellosa, "Energy Management for Hypervisor-based Virtual Machines", *In Proc. of IEEE Symposium on USENIX Annual Technical Conference*, pp. 28-37, 2007.

[14] J. Heo, D. Henriksson, L. Xue, and T. Abdelzaher, "Integrating adaptive components: An emerging challenge in performance-adaptive systems and a server farm case-study," *In Proc. of the 28th IEEE International Real-Time Systems Symposium*, pp. 227–238, 2007.

[15] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE Trans. Parallel Distrib. Syst.*, pp. 1458–1472, 2008.

[16] G. Khanna, K. A. Beaty, G. Kar, and A. Kochut, "Application performance management in virtualized server environments," *In Proc. of Network Operations and Management Symposium (NOMS)*, pp.373-381, 2006.

[17] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "EnaCloud: An Energy Saving Application Live Placement Approach for Cloud Computing Environments", *IEEE International Conference on Cloud Computing*, pp. 17-24, 2009.

[18] G. Metri, S.Srinivasaraghavan, S.Weisong, M.Brockmeyer, "Experimental Analysis of Application Specific Energy Efficiency of Data Centers with Heterogeneous Servers," *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on* , pp.786,793, 2012.

[19] M. Reidmiller, H. Braun, "A Direct Adaptive Method for Faster Back-propagation Learning: The RPRO Algorithm.", *In Proc. of the IEEE International Conference on Neural Networks*. 1993, p. 135-147.

[20] "CloudSim": a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms", Software: Practice and Experience, pp. 23–50, 2011.

[21] Park KS, Pai VS. "CoMon: a mostly-scalable monitoring system for PlanetLab.", ACM SIGOPS Operating Systems Review 2006.

[22] "Amazon EC2 Instance Types", http://aws.amazon.com/ec2/instance-types

[23] "Standard Performance Evaluation Corporation", http://www.spec.org/power_ssj2008/results/res2011q1/power_ssj2008-20110209-00353.html

[24] "Standard Performance Evaluation Corporation", http://www.spec.org/power_ssj2008/results/res2010q2/power_ssj2008-20100315-00239.html

[25] A. Beloglazov and R. Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers", Concurrency and Computation: Practice and Experience (CCPE), John Wiley & Sons, Ltd, pp. 1397-1420, 2012.

◐ 저 자 소 개 ◑

프랭크 엘리호데 (Frank I. Elijorde)

2003년 Western Visayas College of Science and Technology, Philippines
BS in Information Technology
2007년 Western Visayas College of Science and Technology, Philippines
MS in Computer Science
2011～현재 Kunsan National University, South Korea, Graduate Student in Ph. D. Course
관심분야 : Distributed systems, cloud computing, data mining, ubiquitous sensor networks, RFID
E-mail : frank@kunsan.ac.kr

이 재 완 (Jaewan Lee)

1984년 중앙대학교이학사-전자계산학
1987년 중앙대학교이학석사-전자계산학
1992년 중앙대학교공학박사-컴퓨터공학
1996년 3월~ 1998년 1월 한국학술진흥재단 전문위원
1992 ~ 현재 군산대학교교수
관심분야 :분산시스템, 운영체제, 유비쿼터스 시스템, 클라우드컴퓨팅 등
E-mail: jwlee@kunsan.ac.kr