International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 4 No: 3 March 2019

Improving Electricity Usage based on Computational Modeling in Cloud Computing

[1]Azham Hussain, [2]Zarul Fitri Zaaba
[1]School of Computing, Universiti Utara Malaysia, Kedah, Malaysia
[2]School of Computer Science, University Sains Malaysia, Penang, Malaysia

Abstract: The achievement of energy efficiency is gradually receiving a lot of attention these days due to the budget and environmental issues. A prediction technique has been developed in our previous research to improve monitoring statistics. In this research, our new proposal can make the optimization to solve the energy issue of cloud computing by adopting the predictive monitoring information. Actually, the convex optimization technique is coupled with the proposed prediction method to produce a near-optimal set of physical machines hosting. After that, an appropriate migration instruction may eventually be created. The cloud orchestrator can relocate virtual machines to a designed sub-set of infrastructure on the basis of this instruction. The idle physical servers can then be switched off appropriately to save power and maintain system performance. For evaluation purposes, an experiment is conducted based on Google Traces 29-day period. By using this assessment, the proposed approach demonstrates the potential to significantly reduce power consumption without affecting service quality.

Keywords: Cloud Computing, virtual machines (VMs), Energy Efficiency, Convex Optimization.

1. INTRODUCTION

In current years, cloud computing has been recognized as a popular platform for managing most operations by a number of data centers. Cloud computing naturally enhances the use and scalability of the physical infrastructure underlying it. Cloud computing can deliver the ordered resource as a virtual package conveniently via internet connection as a substitute for independently allocating the computing facilities when requested. It should also be noted that cloud computing can be used to enhance infrastructure utilization by virtualizing the composition of the service to a higher level. Therefore, the physical facilities capacity can be unified to provide better service quality. Finally, cloud computing can reduce the cost of managing to save money as a consequence.

To achieve a reduction in power consumption in cloud computing, understanding the sources that consume the energy and how to reduce the corresponding consumption efficiently would be a must. Obviously, most internal components burn the power to do the assigned jobs when a computing system is online. Because of this, any devices running inefficiently that are in idle state will actually waste the power for very limited value. Critically, to save energy, this type of facilities should be minimized. The conventional approach is to decrease the number of physical machines working to an optimal amount capacity. Cloud computing has an opportunity to implement this approach by stacking virtual machines (VMs) by using virtualization. The VMs can be migrated to an optimally designated physical machine (PMs) for this purpose. The remaining idle PMs are then switched off to meet the requirement to mitigate the burning of power. In latest research, to improve monitoring statistics, we have developed an enhanced prediction technique based on Gaussian process regression. We would like to propose an optimization scheme in this research to reduce cloud computing power consumption.

2. RELATED WORKS

Cloud computing energy efficiency is mostly associated with the consolidation philosophy of VM. This means that the interest issue focuses on selecting the appropriate placement for VMs with respect to the use of PMs [3]. Basically, as a regular object - bin issue, we can model VMs and PMs. The consolidation of the VM can therefore be simplified to the problem of bin-packing, which is NP-hard [1]. The heuristics-oriented techniques could therefore be the promising solutions. Whereby, this methodology is popularly adopted by some well-known approaches, namely decreasing the best fit [1] and decreasing the first fit [5]. By

using these techniques, there is a tendency for the cloud orchestrator to assign VMs to minimize the number of PMs hosting. Because of this attractive feature, the bin - packing model mentioned above is widely used to generate the energy efficiency solution. However, when implementing in action, heuristics approaches have a critical drawback. This family requires the fixed number of objects and bins at the beginning of time to produce a good solution. In other words, it is necessary to recognize the amount of VMs and PMs in advance. This requirement is apparently unfeasible as it breaks the principles that make cloud computing, which are elasticity and multi-tenancy. Furthermore, the rapid changes in the utilization of infrastructure clearly degrade binpacking approaches in terms of accuracy. This issue ultimately causes bad effects on system performance because of that.

In other approaches use the prediction techniques as a pre-processing step to enhance input data to break through the mentioned obstacle. By predicting the workload of the infrastructure, the cloud orchestrator can make more reasonable decisions to reduce only unexpected fluctuation of use. A number of research would take this method into account in their proposals. Prediction algorithm candidates are different from hidden model Markov [8] to polynomial fitting [16]. Unfortunately, the designed philosophy of versatile resource provision in cloud computing is not given sufficient attention by these authors. Thus, these techniques may not provide the orchestrator with a good prospect of the underlying system. In addition, there is another research attempting to predict the workload using the Wiener filter [7]. Wiener filter, however, only performs properly with the stationary signal and noise spectrum to the best of our knowledge. It may not be a good idea to bring signal processing techniques to the cloud computing domain without rigorous analysis. For this reason, Wiener filter may be inapplicable in the domain of interest for the purpose of prediction.

In addition, the modified specific schedulers in [6], [14] and [2] are a different type of approaches that should be included. These schedulers are the efforts to solve other energy efficiency aspects in the rates of network traffic, resource reconfiguration and communication. By proposing these schedulers, the authors claim to be able to optimize the network throughput as well as balance the use of resources, ultimately saving energy. These research, however, do not consider the importance of preserving system performance. The aforementioned schedulers are therefore unable to be implemented in the systems of service provision.

By investigating the research field, it can be concluded that while energy efficiency is a hot topic in computer engineering these days, not enough research has been successful in balancing energy savings with an acceptable performance, particularly in a predictive and optimized way. Therefore, we would like to propose a solution that incorporates our previous prediction method [4] and convex optimization technique to reduce cloud computing energy consumption. In the next sections the rest of the proposal is described.

3. PROPOSED ARCHITECTURE

3.1 System Description

Suppose the interest infrastructure is a homogeneous system. That means all the facilities for physical computing are the same. This assumption is only to make the derivatives of the equation more comfortable. Indeed, this configuration does not degrade the generality because only by adding some weighted arguments can the heterogeneous system is transformed into a homogeneous system. As previously stated, the research's goal is to reduce cloud computing power consumption. To do this, we follow the philosophy of stacking VMs. In other words, to compact the size of running PMs, VMs consolidation is chosen. This choice is based on the fact that an idle PM actually burns up to 60% of the peak power [9][11][10], which is used to keep the same PM in peak performance. It should be noted that booting a PM only burns 23.9% of the same power [13]. In addition, reducing the number of running PMs provides additional power reduction to maintain both the cooling system and the networking devices. Because of these reasons, stopping idle PMs can help save more power than leaving them to serve no specific purpose, even an additional cost is required to re-activate the offline computing facilities later. We design an architecture based on this reasoning, namely the energy efficiency management system (E2M) shown in Figure 1. The main objective of this architecture is to create an optimal consolidation strategy for VMs and periodically send it to the orchestrator. Finally, in order to reduce power consumption, idle PMs are temporarily deactivated. Following is the architecture functionality of each component:

International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 4 No: 3 March 2019

Ganglia: For both PMs and VMs, this component collects most operating statistics. The information collected is actually used in the next stage as the input for the prediction step. Note that for monitoring purposes, Ganglia is known to be trusted as a platform for years. This component is lightweight but powerful and versatile enough for any solution to be integrated.

Predictor: This component is the sink of data for statistics from Ganglia. The enhanced Gaussian process regression is activated to do the prediction step after receiving the above-mentioned data. The outcome of this step is the statistics of predictive monitoring. In other words, the predictor offers the futuristic view of the infrastructure's working status. For the optimization step, this type of anticipated system utilization is more valuable than the original data.

Energy optimizer: The predictive monitoring statistics obtained from the predictor can be used as the valuable input to create near - optimal consolidation instruction. If possible, the strategy must save as much power as possible without deteriorating service quality. In fact, it is this component's responsibility to decide the minimum but feasible set of PMs to host the growing VMs normally. Finally, the total VM consolidation package is delivered for implementation to the cloud orchestrator.

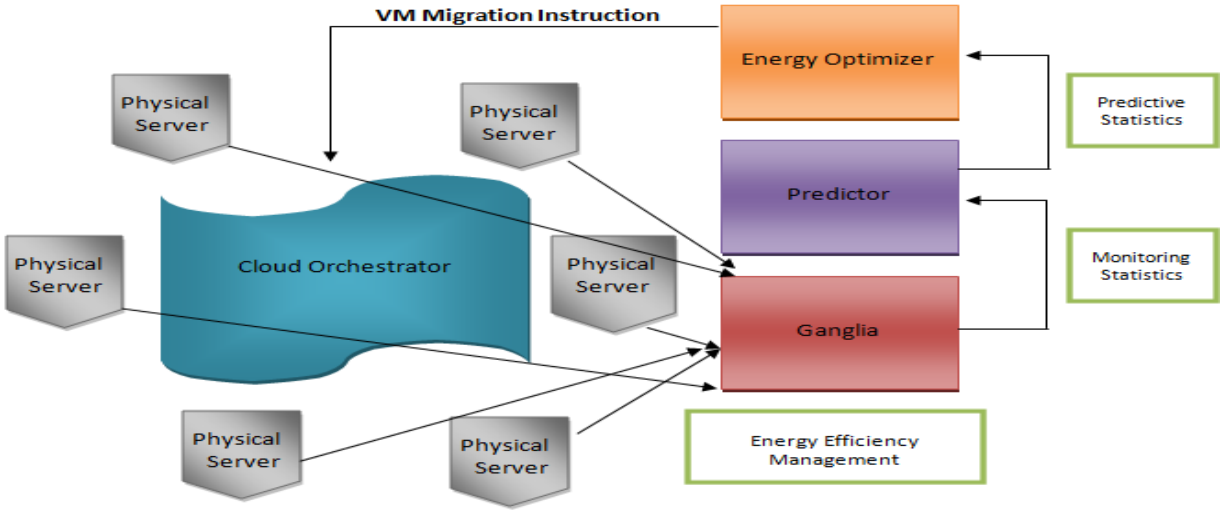

Figure 1. Architecture of energy efficiency management (E2M) system

2. Prediction Model

As mentioned above, in the energy optimizer component, the migration instruction would be created. It is compulsory to improve the monitoring data in advance before the optimization procedure can be issued. The reason for this enhancement's need is double. First, it is known to be true that the statistics of monitoring are always the information that has been delayed. It means that the data we received at the time t actually reflects the status of the system at the time $t - \tau$, in which the monitoring window triggers the process of data collection. At the time the reaction is executed, any decision - making based on this obsolete data may not be reasonable. There is obviously a requirement for data prediction in view of this fact. The second reason is that the use of a proactive reaction rather than a reactive model is sometimes better. In this case, the orchestrator would have a higher chance of reducing the violation to service quality in advance. The predictor's goal is regularly to provide the optimizer with the futuristic use of resources. To do that, to make the regression, the Bayesian learning and the Gaussian process regression are chosen. In our previous research [4], the guidance on how to build this prediction model is provided in detail.

4. ENERGY OPTIMIZATION

Coming to this step, we assume that sufficient information is received from the predictor by the energy optimizer, it is time to conduct the optimization for power consumption. As mentioned earlier, the output of this stage requires a minimum - but - feasible number of PMs. Note that the output is then used to build the VM migration instruction. In the energy optimizer, there are primarily two sub -

components, namely power management and cluster optimizer. The power management follows the resource pool and incorporates the energy decision made from the optimizer of the cluster. The final decision can be called the VM migration instruction. This instruction will be sent for action to the cloud orchestrator.

4.1 Performance modeling

As CPU is one of the most responsive parameters among usage information, this factor should be chosen to model the presentation. Identify global use as $U_m^{fi} \in \mathbb{R}^+$ and individual use as $I_m^{fi} \in \mathbb{R}^+$ with respect to resource fi (for example, fc is CPU). The number of active PMs at the monitoring window m denoted by a_m is the target to be calculated. It is important to mention that consolidating the VMs into a number a_m of PMs could result in its peak performance being infrastructured. This procedure must therefore be monitored. Otherwise, the entire system may suffer very high latency[15] and violate the service quality described in the Service Level Agreement (SLA) document. The use of CPU resources should therefore be formulated as follows:

$$I_m = \max_{f_c}\{I_m^{f_c}\} = \max_{f_c}\{\frac{U_m^{f_c}}{a_m C^{f_c}}\}. \quad (1)$$

Observing (1), I_m is known to be a function of am decreasing. In other words, decreasing the number of PMs could give the entire system high latency. Denote the average task processing latency as *l*m in CPUs. This parameter can be calculated by the exhausted CPU's expected waiting time E(f_c):

$$l_m(I_m) = \mathbf{E}(f_c) = \frac{\lambda_m{}^1/\mu^2}{2(1-\lambda_m{}^1/\mu)}, \quad (2)$$

Where λ*m* is the task's arrival rate, μ is the homogeneous C PU's service rate.

By compared *l*m to the threshold *l* (pictured in the SLA document), it is possible to estimate the quality of services to be violated or not. If the violation happens, it is necessary to calculate the penalty cost C_m^p as follows:

$$C_m^p = w_m s_p(l_m(I^m) - l)^+, \quad (3)$$

in which, respectively, w_m and s_p represent the weight factor reflecting the magnitude of the violation and the fine payable for the penalty. Also supposed to extinguish the trend of the average increase in latency is the weight factor w_m. In other words, this parameter flexibly allows as a preventive method for reducing overhead system a controlled variety of underperformance PMs. This weight essentially plays the role of preserving the execution of SLA.

4.2 Energy Modeling

As we all know, energy consumption in running clusters can be broken down to two periods: the computation period of the assigned tasks and the maintenance period of the idle state. The following equation may be used to model this fact:

$$e_m = P_{idle} + P_{running}. \quad (4)$$

Assume that s_m represents the electricity fine at the monitoring window m, the electricity expense denoted by C_m^e, is represented as follows:

$$C_m^e(a_m) = s_m a_m e_m = s_m a_m(P_{idle} + P_{running}). \quad (5)$$

The energy used to process the tasks is untouchable in (5). As a result, in the optimization procedure, the represented parameter P *running* should not be considered. Therefore, (5) is reduced to:

$$C_m^e(a_m) = s_m a_m P_{idle}. \quad (6)$$

4.3 Cluster optimizer

This section represents the core of the energy optimizer. As a brief summary, our goal is to reduce power consumption while maintaining the quality of services. By minimizing the number a_m of active PMs, this goal can be achieved. Mathematically, it is necessary to find the variable a_m^* in an optimal way.

Table 1. Summary of Google Traces' Characteristics

Time span	# of PMs	#VM requests	# of users
29 days	12583	>25M	925

This function can be solved by applying the conditions of Karush−Kuhn−Tucker (KKT) to find a near - optimal value.

5. PERFORMANCE EVALUATION

5.1 Experiment Design

The testbed is a 16 homogeneous server cluster. For the detailed configuration, a 2.4Ghz and 12 GB RAM Intel

Xeon E7 - 2870 is designed to host up to 8 VMs in each serving. The infrastructure can host up to 128 VMs to conduct the experiment with these equipment. We use Google traces as a workload simulation for the dataset. These traces, announced by Google, actually include monitoring data from over 12,500 machines over 29-day duration. However, only a set of 6732 machines are selected to meet the homogeneous system assumption. We also randomly extract 2.26 GB of compressed data from 39 GB for the experiment in this set. The selected dataset is made up of many parts. Each part is a 24-hour trace period. We scale the maximum measuring length to 60 seconds for the convenience of presentation. As the monitoring window, this length is also adopted. In addition, a summary of the characteristics of Google Traces is described in Table 1.

5.2 Implementation

The experiment is conducted in four comparison schemes as follows:

- The standard schemes: all PMs are always activated. No power savings is at all acquired.
- The greedy first fit decreasing (FFD) scheme [12]: the VMs are sorted into queue in terms of internal CPU utilization by decreasing order. This queue is submitted to the first host that corresponds to the requirement of the resource. Basically, relocating VMs is using the bin - packing approach.
- The proposed approach (E2M) scheme: to create near-optimal energy consumption and preserve the quality of services, the proposed method is implemented.
- The optimal energy-aware scheme: to achieve minimum energy consumption, an optimal solution is pre-calculated. The quality of services is not taken into consideration in this scheme. To put it another way, the quality of services is sacrificed to save the energy significantly.

6. RESULT

The traces of Google are in fact a set of synthesized data. Therefore, an external energy equivalent calculation [13] is used to calculate the result in order to measure the energy consumption. The calculation description and associated parameters are shown in the original paper and summarized in Table 2. As shown in Figures 2 and 3, the default scheme consumes an enormous amount of power due to the constant activation of the PMs.

Table 2. Energy Estimation Parameters

Parameter	Value	Unit
E_{sleep}	107	Watt
E_{idle}	300.81	Watt
E_{peak}	600	Watt
$E_{active \rightarrow sleep}$	1.530556	Watt-hour
$E_{sleep \rightarrow active}$	1.183333	Watt-hour
$E_{active \rightarrow off}$	1.544444	Watt-hour
$E_{off \rightarrow active}$	11.95	Watt-hour

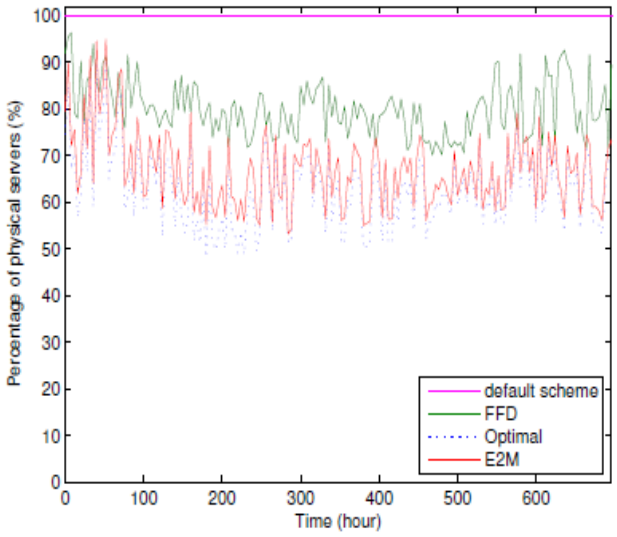

Figure 2. Percentage of active physical servers in Google traces experiment

FFD scheme, even power usage is less than the standard scheme, a remarkable amount of power is wasted because when the workload fluctuates, many idle PMs are kept alive. The reason for this issue is that the FFD is unable to perform the bin-packing algorithm properly in most cases without the ability to predict. An another reason is the underlying computing facilities ' obsolete status information. On the other hand, the proposed approach, namely E2M, can save significantly better energy by equipping the resource utilization prediction and optimizing the active PM pool. This achievement also has another additional aspect, the gap between E2M and the optimal scheme. Apparently, regardless of system performance, the optimal scheme has better energy savings. Because in this scheme the quality of service is totally not considered, this

optimal solution brings too much overhead to the infrastructure and tends to violate the SLA frequently.

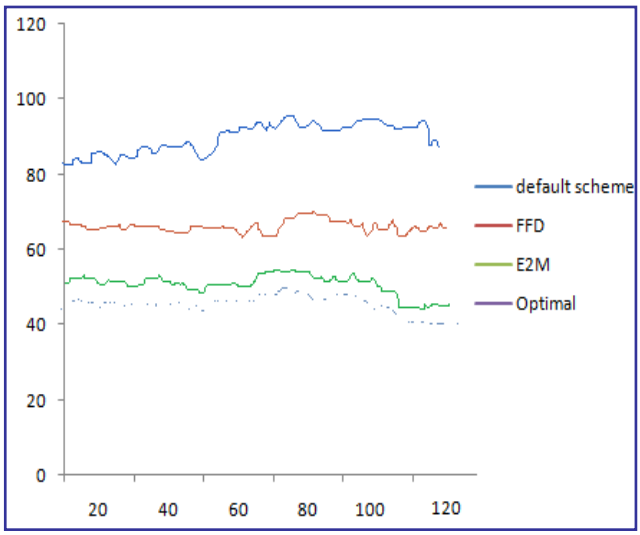

Figure 3. Power consumption evaluation of the proposed method in Google traces experiment (lower is better)

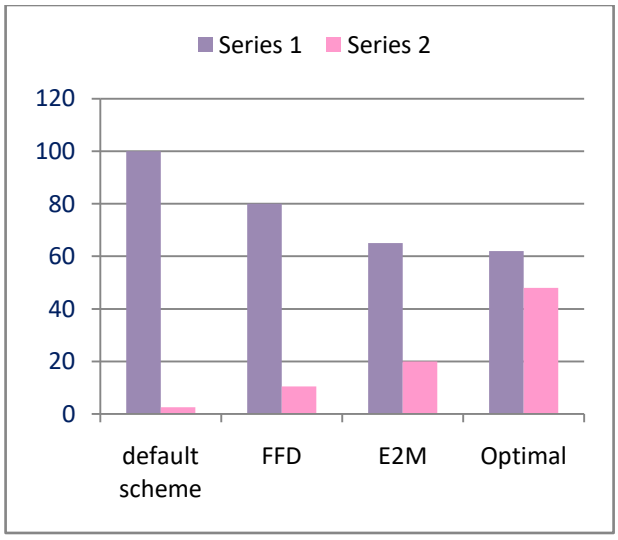

Figure 4. Power consumption vs average latency in Google traces experiment.

Our proposal can achieve a reduction in power consumption of up to 34.89 percent compared to the default scheme for more details on quantitative measurement of energy savings. You can find the detailed evaluation in Figure 4. This achievement can be considered as a major improvement. The optimal scheme can only reach up to 37.08 percent as a side note. It means that the method proposed can be viewed as an almost optimal solution. Our method also suffers in Figure 4 about 54.72 percent less than the optimal solution in terms of average system scheduling latency. The quality of services can therefore be maintained at an acceptable level.

7. CONCLUSION

A near-optimal energy-efficient solution is proposed in this research based on infrastructure utilization prediction and optimization of power consumption. By using the above techniques, our proposal can create an appropriate strategy for VM migration. The cloud orchestrator can issue more reasonable consolidation of VMs based on this migration scheme and condense the pool of active PMs almost optimally. As a result, it is possible to achieve a significant reduction in energy consumption while maintaining the SLA. We plan to integrate the heuristics algorithm in the future in order to build a knowledge base that may help to reduce the overhead when predicting. This integration could boost the part of the prediction to create the VM migration instruction even faster.

REFERENCES

[1] Yasuhiro Ajiro and Atsuhiro Tanaka. 2007. Improving packing algorithms for server consolidation. In Int. CMG Conference. 399–406.

[2] Enzo Baccarelli, Nicola Cordeschi, Alessandro Mei, Massimo Panella, Mohammad Shojafar, and Julinda Stefa. 2016. Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study. IEEE Network 30, 2 (2016), 54–61.

[3] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems 28, 5 (2012), 755–768.

[4] Dinh-Mao Bui, Huu-Quoc Nguyen, YongIk Yoon, SungIk Jun, Muhammad Bilal Amin, and Sungyoung Lee. 2015. Gaussian process for predicting CPU utilization and its application to energy efficiency. Applied Intelligence 43, 4 (2015), 874–891.

[5] Edward G Coffman Jr, Michael R Garey, and David S Johnson. 1996. Approximation algorithms for bin packing: a survey. In Approximation algorithms for NP-hard problems. PWS Publishing Co., 46–93.

[6] Nicola Cordeschi, Mohammad Shojafar, and Enzo Baccarelli. 2013. Energy-saving self-configuring networked data centers. Computer Networks 57, 17 (2013), 3479–3491.

[7] Mehiar Dabbagh, Bechir Hamdaoui, Mohsen Guizani, and Ammar Rayes. 2015. Energy-efficient resource allocation and provisioning framework for cloud data centers. IEEE Transactions on Network and Service Management 12, 3 (2015), 377–391.

[8] Christopher Dabrowski and Fern Hunt. 2009. Using markov chain analysis to study dynamic behaviour in large-scale grid systems. In Proceedings of the Seventh Australasian Symposium on Grid Computing and e-Research-Volume 99. Australian Computer Society, Inc., 29–40.

[9] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power provisioning for a warehouse-sized computer. In ACM SIGARCH Computer Architecture News, Vol. 35. ACM, 13–23.

[10] Ajay Gulati, Anne Holler, Minwen Ji, Ganesha Shanmuganathan, Carl Waldspurger, and Xiaoyun Zhu. 2012. Vmware distributed resource management: Design, implementation, and lessons learned. VMware Technical Journal 1, 1(2012), 45–64.

[11] David Meisner, Brian T Gold, and Thomas F Wenisch. 2009. PowerNap: eliminating server idle power. In ACM Sigplan Notices, Vol. 44. ACM, 205–216.

[12] Thiago Kenji Okada, Albert De La Fuente Vigliotti, Daniel Macêdo Batista, and Alfredo Goldman vel Lejbman. 2015. Consolidation of VMs to improve energy efficiency in cloud computing environments. (2015), 150–158.

[13] Imad Sarji, Cesar Ghali, Ali Chehab, and Ayman Kayssi. 2011. CloudESE: Energy efficiency model for cloud computing environments. In Energy Aware Computing (ICEAC), 2011 International Conference on. IEEE, 1–6.

[14] M. Shojafar, N. Cordeschi, and E. Baccarelli. 2016. Energy-efficient Adaptive Resource Management for Real-time Vehicular Cloud Services. IEEE Transactions on Cloud Computing PP, 99 (2016), 1–1. https://doi.org/10.1109/TCC.2016.2551747

[15] Qi Zhang, Mohamed Faten Zhani, Shuo Zhang, Quanyan Zhu, Raouf Boutaba, and Joseph L Hellerstein. 2012. Dynamic energy-aware capacity provisioning for cloud computing environments. In Proceedings of the 9th international conference on Autonomic computing. ACM, 145–154.

[16] Yuanyuan Zhang, Wei Sun, and Yasushi Inoguchi. 2006. CPU load predictions on the computational grid*. In Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on, Vol. 1. IEEE, 321–326.

