A Power and Thermal-Aware Virtual Machine Allocation Mechanism for Cloud Data Centers

Jing V. Wang[†], Chi-Tsun Cheng, and Chi K. Tse
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hunghom, Kowloon, Hong Kong
[†]Email: jing.j.wang@connect.polyu.hk

Abstract—**With the rapid growing number of Cloud applications, demands for large-scale data centers have raised to historical high. Cloud data centers allow dynamic and flexible resource provisioning to accommodate time varying computational demands. Recent studies have proposed several allocation policies based mainly on power consumption of servers. Host temperature, however, is rarely considered as a monitoring parameter. This work proposes a power and thermal-aware virtual machine (VM) allocation mechanism for Cloud data centers. The objective of the proposed mechanism is to reduce the overall energy consumption and VM migration numbers, while avoiding violations of Service Level Agreements (SLA) in Cloud data centers. The proposed mechanism was implemented and evaluated on CloudSim. Simulation results show that the proposed allocation mechanism brings significant benefits in terms of energy saving and other performance indices.**

Index Terms—**Cloud Computing, Virtualization, Data Centers, Thermal Aware, Energy Consumption**

I. Introduction

In recent years, the proliferation of scientific, business, and web applications has introduced huge demands for large-scale computing data centers. Power consumption has become a critical concern as data centers consume considerable amounts of power. Without a proper resource provisioning mechanism, poor utilization of resource and hot-spot problems may arise. Power-aware provisioning mechanisms can, not only bring substantial energy savings, but also reduce operational costs [1], [2].

Virtualization is one of the key techniques for reducing energy consumption [3]–[5]. By creating several virtual machines (VMs) on a physical host, virtualization helps improve the utilization of resources and reduces idling of computational equipment. An attractive mechanism for dynamic resource management is live VM migration [6], [7]. It is the process of migrating a VM from one physical machine to another, which aims to yield a better resource allocation or consolidate VMs onto fewer physical hosts.

In this work, a VM allocation mechanism, which considers both energy consumption and temperature of the hosts, is proposed. A new thermal-aware function is introduced for VM selection and identifying suitable hosts for VM migrations. The proposed mechanism is implemented and evaluated on CloudSim [8], a standard platform for simulating and modeling control mechanisms in Cloud data centers. Simulation results show the promising performance of the proposed mechanism in energy saving while keeping the migration number low without imposing significant violations on system constraints. The rest of the paper is arranged as follows. Related work is reviewed in Section II. Problem formulation is given in Section III. The proposed VM allocation mechanism is introduced and elaborated in Section IV. In Section V, performance of the proposed mechanism is evaluated using computer simulations. The results are further studied and discussed in Section VI. Finally, conclusions are given in Section VII.

II. Related Work

Several techniques have been proposed in literature for resource provisioning. One of the representatives is Dynamic Round-Robin in [9]. The authors in [9] have investigated the problem of power consumption in data center and came up with an extended version of the original Round-Robin method. Qavami et al. [10] have introduced a resource allocation method at the application level that allocates an appropriate number of VMs to an application according to resource requirements. In [11], Chen et al. proposed a VM allocation mechanism for Cloud data centers that consolidates complementary VMs with spatial/temporal-awareness. Viswanathan et al. [12] presented a brute-force algorithm for multiple dimensions application profiling. The work in [13] presented several approaches to tackle an energy-aware scheduling problem. Song et al. [14] presented a live migration approach based on application demands to optimize the number of active servers. However, none of them considered the temperature of the hosts, which can lead to serious hot-spot or cold-spot problems.

In contrast, scheduling algorithms in [15] and [16] solely consider temperature of the nodes in a Cloud data center. Mhedheb et al. [17] proposed a technique that imposes utilization and temperature thresholds to identify critical hosts for VM migrations. However, these algorithms do not consider penalties for violating Service Level Agreements (SLAs).

Most of the aforementioned methods are relying on host's power consumption for finding the target host in the migration process. Host temperature is rarely considered as a selection and/or allocation criterion.

III. Problem Formulation

In this work, temperature of hosts is considered in the VM selection and allocation process. The system under consideration is an ordinary Cloud data center with N heterogeneous physical hosts and M heterogeneous VMs. Users can submit requests for provisioning of these VMs with resource requests characterized by the parameters of the hosts. With VM migrations and consolidations, multiple VMs may be allocated onto a single physical host. The following models are adopted in the current project.

A. Power Model

Power consumption of computing nodes is mainly determined by their CPU, memory, storages, and network interfaces utilizations. In comparison, CPU is the main contributor to the host power consumption. It has been shown that there is a linear relationship between the host power consumption and its CPU utilization [18]. Nevertheless, [18] also shows that idle servers can still consume approximately 70% of their peak power. Therefore, switching off idle servers is highly recommended. In this work, real data on power consumption provided by SpecPower08 [22] will be utilized in the simulations.

B. Temperature Model

In this paper, a lumped RC thermal model [19]–[21] is adopted to describe the thermal behavior of the processors. The temperature for a processor operated at power P for a time t can be calculated as

$$T(t) = P \times R_{\text{th}} + T_{\text{amb}} - (P \times R_{\text{th}} + T_{\text{amb}} - T_{\text{init}}) e^{-t/R_{\text{th}}C_{\text{th}}}, \quad (1)$$

where R_{th} and C_{th} represent the equivalent thermal resistance and thermal capacitance, respectively. T_{amb} is the ambient temperature, and T_{init} is the initial temperature.

C. Host Overloading Detection Algorithms

In the simulation, we adopt different algorithms in [13] to detect whether a host is overloaded.

1) Static Threshold (THR): Fixed values of utilization thresholds are chosen for deciding the migrations of VMs.

2) Interquartile Range (IQR): set an upper utilization threshold depending on the difference between the first and third quartiles in CPU utilization;

3) Median Absolute Deviation (MAD): set the threshold according to the absolute distance from the median of host CPU utilization;

4) Local Regression Robust (LRR): fit a trend polynomial to the last k observations of CPU utilization to estimate the next observation and check whether it satisfies some conditions of host overloading detection.

D. VM Selection Policies

In this work, four VM selection policies in [13] are selected for comparison purposes.

1) Maximum Correlation (MC): migrate a VM with the highest correlation of CPU utilization with other VMs on the same host;

2) Minimum Migration Time (MMT): migrate a VM with the shortest time to complete a migration;

3) Minimum Utilization (MU): migrate a VM with the minimum utilization;

4) Random Selection (RS): select a VM to be migrated randomly.

IV. Proposed VM Allocation Mechanism

The proposed VM allocation mechanism can be divided into three steps: (1) identifying critical hosts: overloaded hosts and underutilized hosts; (2) selecting VMs for migrations: migrate VMs away from an overloaded host; (3) reallocating those migrated VMs onto underutilized host(s).

In this work, different algorithms are chosen for host overloading detection. If the CPU usage of a physical node exceeds the utilization threshold, one or multiple VMs will have to be migrated away in order to prevent a potential SLA violation. By the end of the VM migration process, hosts with zero VM will be hibernated for energy saving.

Once a host has been identified as being overloaded, VM(s) will be migrated away from this host. In this step, we propose a new selection method for choosing VMs for migration. The new selection method migrates a VM j such that the overloaded host can achieve the minimum temperature distance (TD) after the migration. Here, TD is estimated as the absolute distance between a desirable temperature and the temperature of the host after migration. Both energy consumption and SLA violations can remain at relatively low values if servers can operate at the optimal temperature given in [17]. The new selection method chooses a VM j that minimizes TD, which is defined as

$$\text{TD} = |T_{\text{opt}} - T_i|, \quad (2)$$

where T_i represents the temperature of a host i and T_{opt} is the optimal temperature according to [17], which tries to obtain a reasonable trade-off between enenry consumption and SLA violations.

Algorithm 1: VM Selection Policy

```
Input: OverUtilizedHosts, vmList
Output: VmsToMigrateList
foreach host in OverUtilizedHosts do
    minTD ← MAX
    migratableVM ← NULL
    foreach vm in vmList do
        if TD<minTD then
            migratableVM ← vm
            minTD ← TD
        end
    end
    VmsToMigrateList.add(migratableVM)
end
return VmsToMigrateList
```

The idea of the proposed selection method is to select VMs to be migrated, such that after the migration the host can operate at a temperature closer to the optimal temperature in [17]. We denote this VM selection method as Minimum Temperature Distance (MTD). The pseudocode for the proposed method is presented in Algorithm 1.

The last step of the VM reallocation process is to find a suitable host for accommodating the migrated VMs. This problem can be viewed as a bin packing problem. The CPU resource available at each physical host is regarded as the bin size. Items are representing VMs that have to be allocated while the prices are corresponding to the Temperature Distance Ratio (TDR) value of the hosts. Details on the proposed TDR function is elaborated as follows.

Modern processors have the capabilities to measure on-chip temperature. A high on-chip temperature indicates a high energy consumption and also a high possibility of overheating that may result in hardware failure. It is therefore desirable to migrate VMs to hosts with both low power consumption and temperature. We define TDR as

$$\mathrm{TDR} = \frac{1}{|T_{\mathrm{opt}} - T_i|}, \tag{3}$$

where T_i represents the current temperature of host i and T_{opt} is the optimal temperature according to [17].

Algorithm 2: TDR Best Fit Decreasing

```
Input: hostList,VmsToMigrateList
Output: allocation of VMs
VmsToMigrateList.sortDecreasingUtilization()
foreach vm in VmsToMigrateList do
    minTDRDiff ← MAX
    allocatedHost ← NULL
    foreach host in hostList do
        if host has enough resources for vm then
            TDRDiff ← estimateTDRDiff(host,vm)
            if TDRDiff<minTDRDiff then
                allocatedHost ← host
                minTDRDiff ← TDRDiff
            end
        end
    end
    if allocatedHost ≠ NULL then
        allocation.add(vm,allocatedHost)
    end
end
return allocation
```

The TDR function utilizes the power model and the thermal model introduced in previous sections for making an allocation decision. Note that under the same level of utilization, a host with a higher MIPS (Million Instructions Per Second) value will normally consume more power than one with a lower MIPS value. Under the same condition, however, the two hosts may accommodate different numbers of VMs. Therefore, it is difficult to have a proper temperature control by simply considering power or utilization of the machines. The rationale of the proposed TDR function is to migrate VMs to hosts with temperature closer to the optimal temperature.

TABLE I: The simulation setup

Physical Host Type	HP−G4	HP−G5
Host MIPS	1860	2660
Host RAM[MB]	4096	4096
Host Storage[TB]	1	1
Virtual Machine Type	**MIPS**	**RAM[MB]**
Type1	500	613
Type2	1000	1740
Type3	1500	1740
Type4	2000	870
Thermal Constants	**Value**	**Unit**
Initial CPU Temperature(T_{init})	318	Kelvin
Ambient Temperature(T_{amb})	308	Kelvin
Thermal Resistance(R_{th})	0.34	Kelvin/Watt
Thermal Capacity(C_{th})	340	Joule/Kelvin

The proposed TDR function is then used in a modified Best Fit Decreasing algorithm (BFD) [13] called TDR-BFD. In TDR-BFD, the selected VMs are sorted in a decreasing order of their current CPU utilizations. The sorted VMs are then allocated to hosts that can yield the least increase in their TDR values sequentially. The algorithm reallocates the selected VM to a host that provides the least difference between the current TDR value and that value after reallocation. This allows VMs choosing the most energy and performance efficient hosts. The pseudocode for the proposed allocation algorithm is presented in Algorithm 2.

V. Simulations

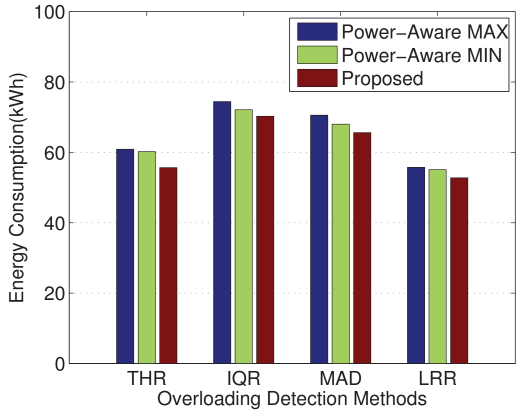
The proposed mechanism is implemented and evaluated using CloudSim [8], which is commonly used for modeling Cloud computing environments. It supports modeling of on-demand virtualization resource and application management, which is highly suitable for evaluating the proposed heuristics.

A. Experiment Setup

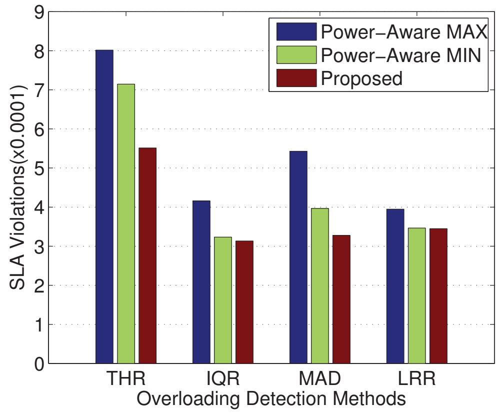
The simulated data center comprises several heterogeneous physical hosts, half of which are HP ProLiant G4 servers, and the other half is composed of HP ProLiant G5 servers. The properties of the servers are given in Table 1. Each node is defined to have one CPU core with performance equivalent to 1860 or 2660 MIPS. The corresponding power model is adopted from SpecPower08 [22].

The characteristics of the four modeled VM types are also given in Table 1. Each VM is a single-core machine with various MIPS and RAM values to simulate real world scenarios. For all VMs, they all run with 100Mbit/s of bandwidth and 2.5 Gigabytes of VM size individually.

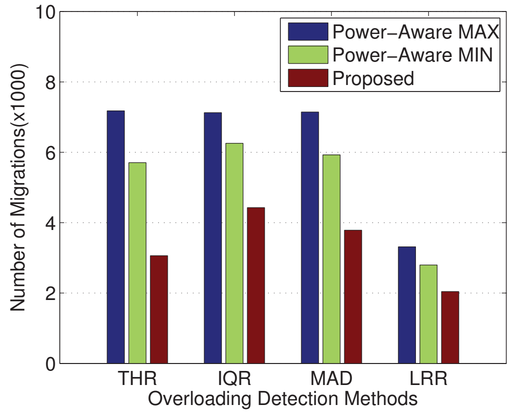
The thermal constants for the lumped RC thermal model are listed in Table 1. Such thermal model is typically used for modeling a single core CPU [23]. The value of optimal temperature threshold is set as 343 Kelvin [17], where a reasonable trade-off between power consumption and SLA



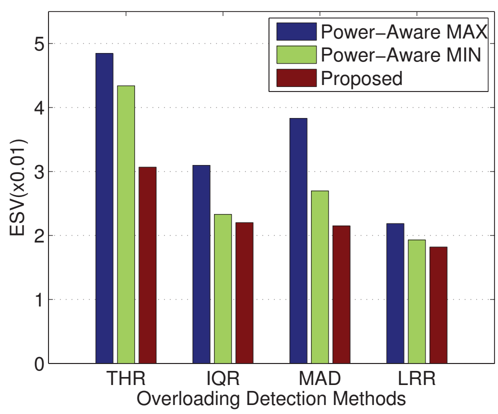
(a) Energy Consumption



(b) SLA Violations



(c) Number of Migrations



(d) Energy and SLA Violations

Fig. 1: Comparisons of the proposed method with other existing power-based methods (Results presented are the average results of 50 simulations using random workload)

violations can be achieved. In the experiments, each simulation lasts for one simulated day with an interval of five minutes. Simulation results are shown in Figs. 1 and 2.

B. Performance Metrics

1) SLA Violation Metrics: Meeting Quality of Service (QoS) is essential for Cloud providers. QoS is usually negotiated in terms of SLA. In this paper, two metrics in [13] are adopted to measure the level of SLA violation: (1) SLA violation Time per Active Host (SLATAH); and (2) Performance Degradation due to Migrations (PDM). Therefore, the SLA Violation (SLAV) is calculated as

$$\text{SLAV} = \text{SLATAH} \times \text{PDM}, \tag{4}$$

where SLATAH is the percentage of time which active hosts have experienced 100% CPU utilization and PDM is the overall performance degradation due to VM migrations. Interested readers may refer to [13] for the explanations of SLATAH and PDM.

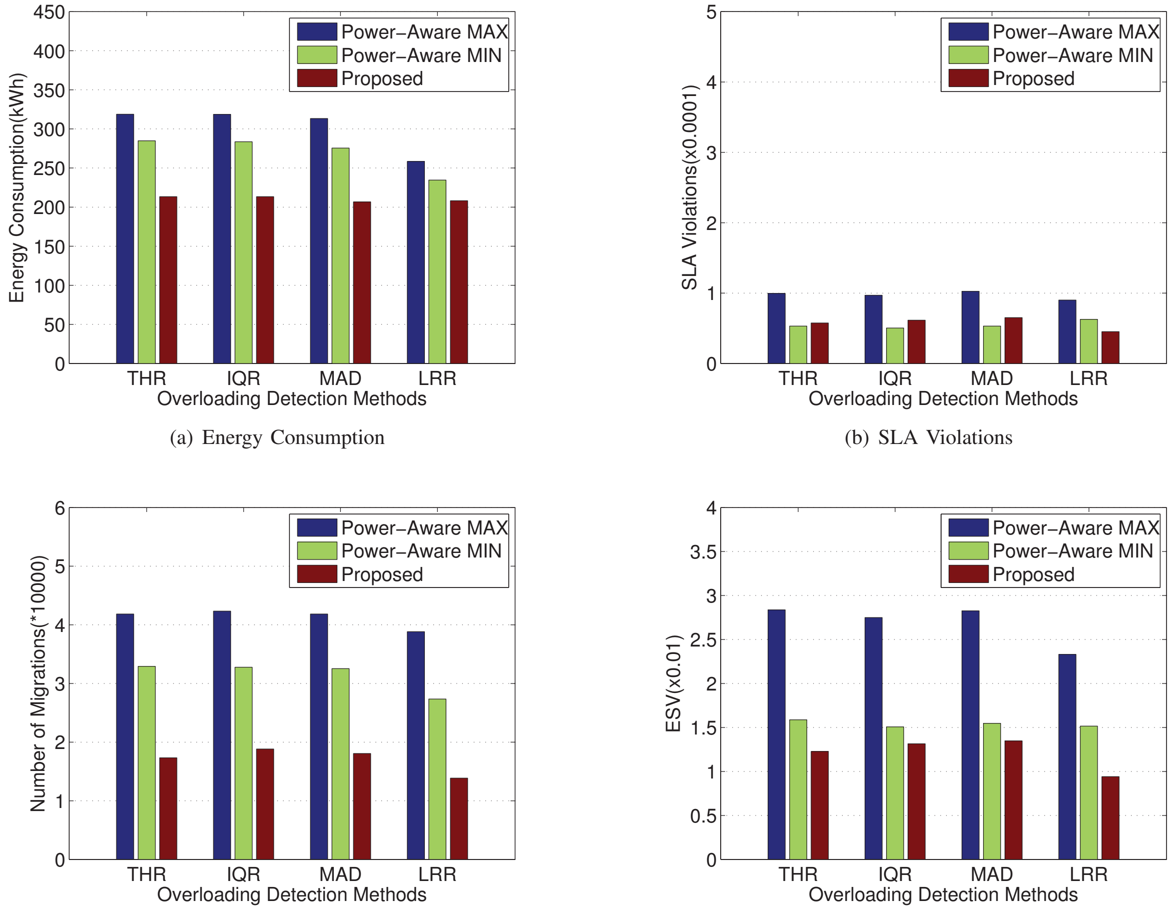
2) Energy and SLA Violations Metrics: The performance metrics of energy and SLA violations are often conflicting. Energy can usually be decreased with a cost of an increased level of SLA violations. Our objective is to achieve a trade-off between power consumption and SLA violations. Therefore, we adopt the metric Energy and SLA Violations (ESV) in [13] that combines energy consumption metric and SLA violations metric together to evaluate the overall performance of Cloud data centers.

$$\text{ESV} = \text{E} \times \text{SLAV}, \tag{5}$$

where E is the total energy consumption of a data center and SLAV is expressed in (4).

VI. Performance Analysis and Discussions

In this section, we carry out a series of experiments using a random workload and a real-world workload to examine the efficiency of the implemented mechanism. In order to highlight the improvement due to the proposed idea, we compare the proposed VM allocation mechanism with all combinations of the power-based allocation mechanism and the four VM



(a) Energy Consumption

(b) SLA Violations

(c) Number of Migrations

(d) Energy and SLA Violations

Fig. 2: Comparisons of the proposed method with other existing power-based methods (Results presented are the average results of 50 simulations using real-world workload)

selection algorithms (MC, MMT, MU and RS) mentioned earlier under different host overloading detection algorithms. To ease reading, we present only the maximum and minimum values of those methods.

A. Random Workload

We first provide a performance comparison under random workload. The simulated data center comprises 50 heterogeneous physical hosts and 50 VMs. The CPU utilization of a VM is generated as a uniformly distributed random variable. Fig. 1 shows the average performance measured in 50 simulations with different methods under different host overloading detection algorithms.

In Fig. 1(a) the total energy consumption for the proposed method and power-based methods is reported. It can be observed that the proposed method is able to bring extra energy savings comparing with other power-based methods. In addition, we examine the SLA violations of the system in Fig. 1(b). This result highlights the ability of the thermal-aware policy to achieve higher level of QoS. Fig. 1(c) compares the number of migrations under different methods. It is observed that the proposed mechanism outperforms the power-based methods by 33%-52%. To evaluate how the performance can be optimized, we focus on the Energy and SLA violations value which is reported in Fig. 1(d). As expected the results indicate that the proposed method has a better overall performance than other power-based methods.

B. Real-world Workload

After that, we provide a performance comparison under a real-world workload. We have randomly chosen one day from the workload traces of PlanetLab [24]. A system with 800 hosts and 1052 VMs was simulated and evaluated. The results produced by different methods are shown in Fig. 2.

Fig. 2(a) shows that the proposed method can reduce the overall energy consumption by 30% when comparing with the other methods. The results of SLA violations are shown in Fig. 2(b). According to the figure, all methods under test can achieve SLAV values < 0.0001. Fig. 2(c) shows the number of migrations achieved by different strategies. The migration number of the proposed method is 50% smaller than its counterparts all the time. A similar result is obtained for the ESV parameter. In terms of ESV, the proposed mechanism outperforms the other power-based algorithms by 38%-51.2%.

When comparing the number of active hosts under different provisioning mechanisms, it is observed that the amount of active hosts in systems with the proposed TDR-based mechanism is much smaller than those with power-based provisioning mechanisms. Within the active hosts, the number of hosts with low MIPS utilized by the proposed mechanism is slightly higher than other mechanisms under test. This explains the promising energy saving performance of the proposed TDR-based mechanism as it tends to migrate VMs to hosts with lower power consumption. However, the value of SLA violation is slightly increased as the proposed mechanism allocates VMs to hosts that are more likely to experience 100% CPU utilization rather than hosts with higher MIPS values. Nevertheless, the proposed mechanism can successfully locate proper hosts for the migrated VMs, which can achieve reasonable trade offs among energy consumption, temperature, and SLA violations.

Each VM migration may result in SLA violations, hence it is essential to minimize the migration number whenever possible. We can observe that the proposed mechanism uses less migrations when comparing with its counterparts. Our proposed mechanism is able to keep hosts working near the optimal temperature through VM reallocations. The total utilization of host's CPU will be kept between the utilization threshold and the utilization deduced from the optimal temperature. As a result, the proposed mechanism enables better consolidations of VMs. Compared with other power-based methods, the proposed method shows better performance in reducing migration number, SLA violations, and energy consumption.

VII. Conclusions

In this paper, a thermal-aware virtual machine (VM) allocation mechanism for Cloud data centers is proposed. The proposed mechanism selects and allocates VMs to hosts based on host's temperature. As a result, the proposed mechanism helps reduce energy consumption and migration number significantly while avoiding SLA violations in Cloud data centers. It can cooperate well with different host overloading detection algorithms under different types of workloads. The performance of the proposed mechanism has been verified using extensive simulation experiments based on CloudSim. This work provides an insight on the importance of considering both host's temperature and energy consumption when performing resource provisioning in Cloud data centers.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[2] L. Wang and S. Khan, "Review of performance metrics for green data centers: a taxonomy study," *The Journal of Supercomputing*, vol. 63, no. 3, pp. 639–656, 2013.

[3] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *Communications Magazine, IEEE*, vol. 50, no. 9, pp. 34–40, September 2012.

[4] D. Jiankang, W. Hongbo, L. Yangyang, and C. Shiduan, "Virtual machine scheduling for improving energy efciency in iaas cloud," *Communications, China*, vol. 11, no. 3, pp. 1–12, March 2014.

[5] J. Zheng, T. Ng, K. Sripanidkulchai, and Z. Liu, "Pacer: A progress management system for live virtual machine migration in cloud computing," *Network and Service Management, IEEE Transactions on*, vol. 10, no. 4, pp. 369–382, December 2013.

[6] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, "iaware: Making live migration of virtual machines interference-aware in the cloud," *Computers, IEEE Transactions on*, vol. 63, no. 12, pp. 3012–3025, Dec 2014.

[7] 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*, ser. NSDI'05. Berkeley, CA, USA: USENIX Association, 2005, pp. 273–286.

[8] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[9] C.-C. Lin, P. Liu, and J.-J. Wu, "Energy-aware virtual machine dynamic provision and scheduling for cloud computing," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, July 2011, pp. 736–737.

[10] H. Qavami, S. Jamali, M. Akbari, and B. Javadi, "Dynamic resource provisioning in cloud computing: A heuristic markovian approach," in *Cloud Computing*. Springer International Publishing, 2014, vol. 133, pp. 102–111.

[11] L. Chen and H. Shen, "Consolidating complementary vms with spatial/temporal-awareness in cloud datacenters," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 1033–1041.

[12] H. Viswanathan, E. Lee, I. Rodero, D. Pompili, M. Parashar, and M. Gamell, "Energy-aware application-centric vm allocation for hpc workloads," in *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, May 2011, pp. 890–897.

[13] 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," *Concurr. Comput. : Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.

[14] W. Song, Z. Xiao, Q. Chen, and H. Luo, "Adaptive resource provisioning for the cloud using online bin packing," *Computers, IEEE Transactions on*, vol. 63, no. 11, pp. 2647–2660, Nov 2014.

[15] A. Kaur and S. Kinger, "Temperature aware resource scheduling in green clouds," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, Aug 2013, pp. 1919–1923.

[16] S. Kinger and K. Goyal, "Energy-efficient cpu utilization based virtual machine scheduling in green clouds," in *Communication and Computing (ARTCom 2013), Fifth International Conference on Advances in Recent Technologies in*, Sept 2013, pp. 28–34.

[17] Y. Mhedheb, F. Jrad, J. Tao, J. Zhao, J. Koodziej, and A. Streit, "Load and thermal-aware vm scheduling on the cloud," in *Algorithms and Architectures for Parallel Processing*. Springer International Publishing, 2013, vol. 8285, pp. 101–114.

[18] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 13–23, Jun. 2007.

[19] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 94–125, Mar. 2004.

[20] F. Beneventi, A. Bartolini, A. Tilli, and L. Benini, "An effective gray-box identification procedure for multicore thermal modeling," *Computers, IEEE Transactions on*, vol. 63, no. 5, pp. 1097–1110, May 2014.

[21] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. Stan, "Hotspot: a compact thermal modeling methodology for early-stage vlsi design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 14, no. 5, pp. 501–513, May 2006.

[22] "Specpower08," http://www.spec.org.

[23] "Hotspot," http://lava.cs.virginia.edu/HotSpot/.

[24] K. Park and V. S. Pai, "Comon: A mostly-scalable monitoring system for planetlab," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 65–74, Jan. 2006.