The following publication J. V. Wang, N. Ganganath, C. T. Cheng and C. K. Tse, "A heuristics-based VM allocation mechanism for cloud data centers," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 2017, pp. 1-4 is available at http://dx.doi.org/10.1109/ISCAS.2017.8050470

A Heuristics-based VM Allocation Mechanism for Cloud Data Centers

Jing V. Wang*, Nuwan Ganganath, Chi-Tsun Cheng, and Chi K. Tse
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
*Email: jing.j.wang@connect.polyu.hk

***Abstract*—Live migration, which allows virtual machines (VMs) to move across distinct physical hosts, is widely adopted for realizing energy saving and load balancing in modern Cloud data centers. However, putting VMs with high correlations on their CPU utilization patterns onto the same host is very likely to trigger overloading incidents and commit Service Level Agreements (SLA) violations, even when the host has not yet reached its critical limits. To avoid such SLA violations and to maintain a low energy consumption, this work proposes a heuristics-based VM allocation mechanism. Under the proposed mechanism, VMs showing high CPU utilization correlations to other co-located peers and hosts with extreme utilization values are both assigned with high heuristic values. VMs with high heuristic values are therefore less preferred to be co-located onto a host that has a high heuristic value. Performance of the proposed mechanism was evaluated using CloudSim with real-world workload data. Simulation results show that when comparing with other existing mechanisms, the proposed idea can further reduce both energy consumption and SLA violations.**

Keywords*—*VM allocation, multiple correlation, CPU utilization, heuristics, cloud computing

I. Introduction

Cloud data centers are now expanding in unprecedented scales and complexities to catch up with the soaring demands from Cloud service subscribers. With massive numbers of servers and networking equipment, Cloud data centers consume a tremendous amount of energy in their daily operation, including energy used in cooling their infrastructures. Therefore, reducing energy consumption while providing the required services to their subscribers has become a major concern to Cloud service providers all over the world. Toward energy saving, computing resources need to be provisioned dynamically and efficiently. This requires the technique of virtualization, which yields better resource utilization by accommodating multiple virtual machines (VMs) onto each physical host. However, co-located VMs with high correlations in their CPU utilization patterns are associated with higher risks of overloading their host, as these VMs are more likely to reach their peak utilization levels at the same time and exhaust all the available resource on the host. An overloaded host may result in violations of Service Level Agreements (SLA) which may lead to additional economic losses. The necessary migration procedures after an overloading event will also introduce extra energy consumption to the system. Therefore, both utilization levels of the hosts and CPU utilization correlations among co-located VMs need to be taken into account in an allocation process.

Extensive studies have been conducted on using CPU utilization correlations among co-located VMs as a migration criterion. Verma *et al.* [1] conducted some pioneering work regarding the resource utilization correlations (RUC) among co-located applications and sheded light on the possibilities of applying it as a criterion in VM placement processes. In [2], Zhang *et al.* proposed a VM migration algorithm that minimizes the number of VM migrations in an over-committed data center. In their proposed idea, VMs with high inter-RUC should be scattered onto different hosts. Canali and Lancellotti [3] exploited the correlations on various resource utilization indices among VMs to cluster VMs according to their behavioral patterns. In [4], Hwang and Pedram modeled resource demands as random variables and then considered the correlations among these random variables to solve the VM consolidation problem. In their work, the RUC between two co-located VMs is adopted for performance evaluation purposes instead of making decisions in the optimization process. However, most aforementioned work did not consider the status of the source and destination hosts in their migration processes. Pillai *et al.* [5] proposed a resource allocation mechanism based on the principles of coalition formation and the uncertainty principle of game theory. In an earlier work of the authors in this paper [6], host's temperature was considered as a migration criterion, which provides a better option in reflecting the utilization levels of hosts. In [7], the VM provisioning process was formulated as a stable matching problem that allows VMs and hosts to have different objectives in the allocation process.

In this paper, by considering hosts' utilization levels as well as RUC among co-located VMs, we proposed a VM allocation mechanism by assigning heuristics to those entities. VMs with high heuristic values (i.e. high correlations in their resource utilization patterns) are intended to *repel* each other, such that they are less willing to be allocated onto the same host. Similarly, hosts with high heuristic values (i.e., high utilization levels) will also intend to repel VMs from migrating onto them and may even expel their VMs. The proposed mechanism aims to reduce the overall energy consumption of a Cloud data center and keep its SLA violations at a relatively low value. Performances of the proposed mechanism were verified using CloudSim, an open-platform that supports modeling Cloud applications and services. Simulation results show that the proposed heuristics-based mechanism can achieve better performances comparing with other existing mechanisms under test.

The rest of the paper is organized as follows. Preliminaries are given in Section II. In Section III, formulations of the heuristics used in the proposed mechanism are introduced. Section IV explains and elaborates the proposed heuristics-based VM allocation mechanism. Simulation setup and results are presented and discussed in Section V. Finally, Section VI draws some conclusions.

II. Preliminaries

Here, we adopt the multiple correlation coefficient [8] to estimate the RUC between co-located VMs. It is commonly used in multiple regression analysis to measure the quality of a prediction of dependent variables from independent variables. The value of a multiple correlation coefficient varies between 0 and 1. A value of 0 indicates that there is no linear relationship between those variables. If there exists a perfect prediction, the multiple correlation coefficient equals 1.

Suppose there are n co-located VMs on a host and they are represented by vector $\mathbf{V} = [V_1, V_2, ..., V_n]$. The RUC strength of the i^{th} VM toward the other $n-1$ VMs is measured based on their last q CPU utilization observations. We denote the vector $\mathbf{y}_i$ to represent the last q observations of the i^{th} VM. Similarly, we denote $\mathbf{X}$ as an augmented matrix comprises the q observations of the remaining $n-1$ VMs on the host. Expressions of $\mathbf{y}_i$ and $\mathbf{X}$ are given as

$$\mathbf{y}_i = \begin{bmatrix} y_{1,i} \\ \vdots \\ y_{q,i} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,m} & \cdots & x_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & x_{p,1} & \cdots & x_{p,m} & \cdots & x_{p,n-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & x_{q,1} & \cdots & x_{q,m} & \cdots & x_{q,n-1} \end{bmatrix}.$$

Here, $x_{p,m}$ is the p^{th} CPU utilization observation of V_m. The multiple correlation coefficient $R^2_{V_i,\mathbf{V}\setminus V_i}$ for each V_i can then be calculated as

$$R^2_{V_i,\mathbf{V}\setminus V_i} = \frac{\sum_{k=1}^{q}(y_{k,i} - m_{\mathbf{y}_i})^2(\hat{y}_{k,i} - m_{\hat{\mathbf{y}}_i})^2}{\sum_{k=1}^{q}(y_{k,i} - m_{\mathbf{y}_i})^2\sum_{k=1}^{q}(\hat{y}_{k,i} - m_{\hat{\mathbf{y}}_i})^2},$$

where $m_{\mathbf{y}_i}$ and $m_{\hat{\mathbf{y}}_i}$ are the means of $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$, respectively. Here, $\hat{\mathbf{y}}_i$ is a vector of predicted utilization values of the i^{th} VM, which can be obtained as

$$\hat{\mathbf{y}}_i = \mathbf{X}(\mathbf{X}^{\text{T}}\mathbf{X})^{-1}\mathbf{X}^{\text{T}}\mathbf{y}_i.$$

In this work, the multiple correlation coefficient between the i^{th} VM and other co-located VMs represents the corresponding RUC between both parties.

III. Heuristics Formulations

As mentioned earlier, VMs with high RUC to their co-located VMs are more likely to trigger overloading events. Unfortunately, such a problem cannot be completely resolved by imposing utilization thresholds to control the utilization level of hosts. Besides, hosts with relatively low utilizations, even idle hosts, could still consume up to 70% of their peak power [9]. This inspires us to formulate the RUC among co-located VMs and the host utilization level as two heuristics. They are then consolidated into a single heuristic function to evaluate the state of each VM for making allocation decisions.

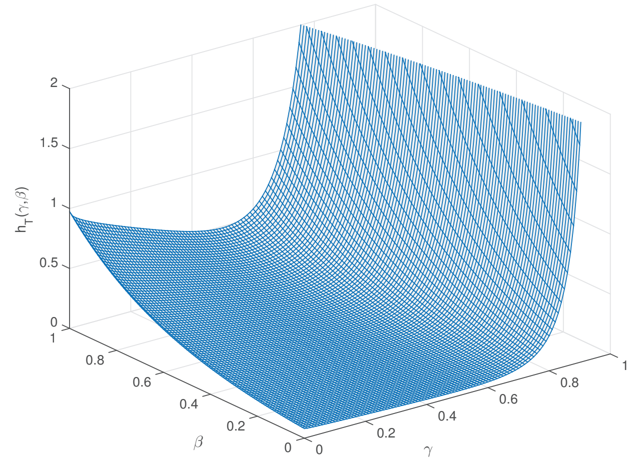

Fig. 1: An illustration of the heuristic function h_{T}.

We first formulate the heuristic corresponds to the utilization level of a host. Hosts with extreme utilization levels are usually operating outside their maximum efficiency range. In contrast, it is more desirable for them to keep their utilization at relatively moderate levels. Because of that, the first heuristic h_1, which corresponds to a host's utilization, is defined as

$$h_1(\gamma) = \frac{(a-c)(1-\sqrt{b})^2}{b}\left(\frac{1}{1-\sqrt{\gamma}} - \frac{1}{1-\sqrt{b}}\right)^2 + c,$$

where $\gamma \in [0,1]$ denotes the CPU utilization of the host. The parameters should be selected as $a > 0$, $1 \geq b \geq 0$, and $a > c \geq 0$. In this work, they are selected as $a = 0.4$, $b = 0.5$, and $c = 0.2$, correspondingly to ensure hosts with extreme utilization will have relatively higher h_1.

We then formulate the second heuristic corresponds to the RUC among co-located VMs. An exponential function is chosen here such that VMs with similar CPU utilization patterns will have higher heuristic values, which will encourage them to migrate onto different hosts. The second heuristic h_2 is formulated as

$$h_2(\beta) = m\exp(n\beta).$$

Here, $\beta \in [0,1]$ is the RUC of the VM under evaluation to other co-located VMs. The parameters should be selected as $m > 0$ and $n > 0$. In this work, they are selected as $m = 0.2$ and $n = 2.5$ to give higher heuristic values to VMs with higher RUC values.

The two heuristics are then consolidated into a single heuristic function h_{T} as

$$h_{\text{T}}(\gamma,\beta) = h_1(\gamma) \times h_2(\beta).$$

An illustration on the characteristics of h_{T} versus host's utilization level γ and VM's RUC value β is shown in Fig. 1.

IV. Proposed VM Allocation Mechanism

In this work, an ordinary Cloud data center with heterogeneous physical hosts and VMs is modelled based on an Infrastructure as a Service (IaaS) environment. Multiple VMs can be co-located onto a single physical host to reduce energy consumption. Furthermore, VMs can be migrated between different hosts for efficient resource management. This dynamic resource provisioning is triggered periodically according to the specified interval of the Cloud service provider. The proposed heuristics-based VM allocation mechanism is composed of the following three major steps:

- Step 1 **Identifying critical hosts:** The mechanism checks for VMs with high values of h_{T}. Here, if the h_{T} value of a VM on a host exceeds a system threshold T_1, the VM is considered as critical. In the proposed mechanism, if there exists any critical VMs on a host, such host will be regarded as a critical host. Some VMs on a critical host will have to be migrated away to prevent a potential SLA violation.
- Step 2 **Selecting VM(s) for migration:** Once a host has been identified as critical, the next step is to select one or multiple of its VMs to migrate away from it. In the proposed mechanism, a critical VM with the shortest migration time on a critical host will be given a higher priority to be migrated first. As long migration time can cause negative impacts on application performance, such design can lower the chance of having SLA violations. After each migration, h_1 and h_2 will be updated. Therefore, Step 2 is executed iteratively until no more critical VM can be found on the host.
- Step 3 **Reallocation of migrated VMs:** The last step of the VM allocation process is to find new hosts to accommodate the migrated out VMs. This problem can be viewed as a bin packing problem with variable bin sizes and prices. Here, bin sizes are representing the available CPU resource of the physical hosts, while prices are corresponding to the h_{T} values of the selected VMs if they are reallocated onto some hosts. As the bin packing problem is an NP-hard problem, we adopt a modified Best Fit Decreasing (BFD) algorithm known as heuristics-based BFD to solve it. In heuristics-based BFD, the selected VMs obtained from Step 2 are sorted in a decreasing order based on their current CPU utilizations. Each sorted VM will be allocated to a host that can yield the lowest value of h_{T} which is lower than the system threshold T_2.

The rationale of the proposed VM allocation mechanism is to arrange VMs with low h_2 values to be operated under hosts with appropriate utilization levels (i.e. low h_1 values). The proposed idea reallocates critical VMs to hosts that can yield minimum h_{T} values. This allows critical VMs to choose more capable hosts and avoid co-locating with VMs with similar utilization patterns. If no active host can accommodate the migrated out critical VMs, an inactive host will be turned on. On the other hand, under-utilized hosts will be turned off for energy saving. The mechanism is repeated until all the critical VMs have been re-allocated.

V. Performance Evaluation

A. Simulation Setup

We carried out extensive simulations on CloudSim-4.0 [10] to evaluate the effectiveness of our proposed mechanism. The data set we used in the simulations is obtained from real-world workload traces of PlanetLab [11]. We choose 10 days from the dataset as in [7] and average out the simulation results for comparisons. Two kinds of servers were emulated in the simulated data center: HP ProLiant G4 (2 cores $\times$ 1860 MIPS), and HP ProLiant G5 (2 cores $\times$ 2660 MIPS). In the simulations, 800 heterougenous hosts were deployed equally with these two configurations. The corresponding power models of the selected hosts were adopted from SpecPower08 [12]. We have simulated four different types of VMs with various characteristics. Each VM is a single-core machine configured with 100Mbit/s of bandwidth and 2.5 Gigabytes of VM size. In the simulations, VM provisioning processes were triggered every five simulated minutes.

In this paper, we adopt two independent metrics given in [13] to evaluate the level of SLA violation: (1) SLA violation Time per Active Host (SLATAH) is an estimate of the percentage of time which the CPU utilization of physical hosts have reached 100%, and (2) Performance Degradation due to Migrations (PDM) is an estimate of the overall performance degradation during VM migration processes. These two metrics are with equal importance. By combining both metrics, a parameter called SLA Violation (SLAV) is defined as

$$\mathrm{SLAV} = \mathrm{SLATAH} \times \mathrm{PDM}.$$

The objective of a VM allocation mechanism is to achieve a reasonable trade-off between energy consumption and SLA violations. Toward these two conflicting metrics, in this work, we adopt a combined metric, the Energy and SLA Violations (ESV) [13] to evaluate the overall performance of a Cloud data center. Here, ESV is calculated as

$$\mathrm{ESV} = E \times \mathrm{SLAV},$$

where E is the total energy consumption of a data center.

In the simulations, four other existing mechanisms were selected for comparison purposes, namely the power-based LRR method [13] and the three allocation mechanisms adopting different correlation-based criteria in [14]. In the power-based LRR method, the increase in host's power consumption after receiving a VM is regarded as the migration criterion. While for the other three mechanisms, their criteria presented in [14] were adopted correspondingly.

In the simulations, the system thresholds T_1 and T_2 of the proposed mechanism were chosen as 0.9 and 0.3, respectively. Six different metrics were chosen for evaluating the performance of the mechanisms under test, namely energy consumption, VM migration number, number of hot-spots, number of cold-spots, SLAV, and ESV. Hosts with CPU utilization above 90% or below 25% are considered as hot-spots or cold-spots, respectively [15]. Table I shows average daily results over 10 simulated days.

TABLE I: SIMULATION RESULTS

VM allocation mechanisms	Energy consumption	Migration number	Hot-spots	Cold-spots	SLAV (x0.00001)	ESV (x0.001)
Power-based [13]	161.71 kWh	28438	1201	5841	5.02	8.03
Correlation of migrated VM(s) [14]	120.99 kWh	11549	1411	618	1.6	1.89
Average correlation level (ACL) [14]	121.07 kWh	11899	1426	620	1.69	1.98
Variation of correlation level (VCL) [14]	121.00 kWh	11477	1391	647	1.55	1.82
Proposed heuristics-based	120.11 kWh	11098	712	448	1.07	1.25

B. Results and Discussions

According to the results, the energy consumption of data centers with the proposed mechanism is much lower than that in [13]. Nevertheless, such value of the proposed mechanism is slightly better compare to that in [14] which can be explained by the lower number of cold-spots. Moreover, the proposed mechanism helps data centers to considerably reduce SLAV compare to its counterparts. The low SLAV values of systems with the proposed mechanism is a result of lower numbers of hot-spots among active hosts and fewer VM migrations. As a result, systems with the proposed mechanism can yield more desirable ESV values.

In the proposed mechanism, system threshold T_1 should be chosen higher than T_2. Systems with high values of T_1 can allow VMs with moderate RUC to be co-located and yield a better utilization. However, having T_1 being too high will trigger more overloading incidents. On the contrary, a low value of T_2 can ensure the destination host of a reallocated VM to have sufficient resource. A high value of T_2 can lead to a high number of migrations as the selected host is more likely to be regarded as critical in future evaluations. However, an extremely low value of T_2 is also not desirable as it will trigger over-provisioning and introduce more under-utilized hosts to the systems.

VI. Conclusions

Modern cloud data centers under the IaaS model present a major challenge in resource management. In this paper, we propose a virtual machine (VM) allocation mechanism based on heuristics to improve the VM consolidation processes in Cloud data centers. The designs of the proposed heuristics incorporate both the utilization levels of hosts and resource utilization correlations among co-located VMs. Under the proposed mechanism, a lower heuristic value indicates a more desirable operating environment for both VMs and hosts. Using real-world data traces from PlanetLab, simulation results show that when comparing with other existing mechanisms under test, the proposed VM allocation mechanism can lower the risk of overloading, reduce Service Level Agreements violations, and achieve reductions in overall energy consumption.

Acknowledgment

This work is supported by the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University (Projects G-YBKH and RTMR).

References

[1] A. Verma, G. Dasgupta, T. K. Nayak, P. De, and R. Kothari, "Server workload analysis for power minimization using consolidation," in *2009 conference on USENIX Annual technical conference*. USENIX Association, 2009, pp. 28–28.

[2] X. Zhang, Z.-Y. Shae, S. Zheng, and H. Jamjoom, "Virtual machine migration in an over-committed cloud," in *2012 IEEE Network Operations and Management Symposium*, 2012, pp. 196–203.

[3] C. Canali and R. Lancellotti, "Automated clustering of vms for scalable cloud monitoring and management," in *2012 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2012, pp. 1–5.

[4] I. Hwang and M. Pedram, "Hierarchical virtual machine consolidation in a cloud computing system." in *2013 IEEE International Conference on Cloud Computing (CLOUD)*, 2013, pp. 196–203.

[5] P. S. Pillai and S. Rao, "Resource allocation in cloud computing using the uncertainty principle of game theory," *IEEE Systems Journal*, vol. 10, no. 2, pp. 637–648, 2016.

[6] J. V. Wang, C.-T. Cheng, and C. K. Tse, "A power and thermal-aware virtual machine allocation mechanism for cloud data centers," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 2850–2855.

[7] J. V. Wang, K.-Y. Fok, C.-T. Cheng, and C. K. Tse, "A stable matching-based virtual machine allocation mechanism for cloud data centers," in *2016 IEEE World Congress on Services (SERVICES)*, 2016, pp. 103–106.

[8] H. Abdi, "Multiple correlation coefficient," *The University of Texas at Dallas*, 2007.

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

[10] 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," *Software: Practice and experience*, vol. 41, no. 1, pp. 23–50, 2011.

[11] K. Park and V. S. Pai, "Comon: A mostly-scalable monitoring system for planetlab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006.

[12] "Specpower08," (Accessed: 2016-10-29). [Online]. Available: http://www.spec.org

[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," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[14] J. V. Wang, C.-T. Cheng, and C. K. Tse, "Effects of correlation-based vm allocation criteria to cloud data centers," in *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2016, pp. 398–401.

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