A Stable Matching-based Virtual Machine Allocation Mechanism for Cloud Data Centers

Jing V. Wang*, Kai-Yin Fok, 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](mailto:jing.j.wang@connect.polyu.hk)

***Abstract*—Virtualization is the enabling technology that makes resource provisioning in Cloud computing feasible. With virtualization, virtual machines (VMs) can be migrated across physical hosts to achieve better utilization of resource with a minimum impact on service quality. The VM allocation problem can be formulated as a stable matching problem. In this paper, we propose a VM allocation mechanism based on stable matching. A deferred acceptance procedure is adopted to handle conflicts among preferences of VMs and physical hosts. Unlike ordinary stable matching problems, both involving party groups in our matching process are having a mutual objective, that is to reduce the overall energy consumption of a Cloud data center while maintaining a high level of Quality of Service. The proposed mechanism is evaluated using CloudSim with real-world workload data. Simulation results show that Cloud data centers with the proposed mechanism can reduce energy consumption and avoid violations of Service-Level Agreement.**

***Keywords*-stable matching; VM allocation; cloud computing;**

I. Introduction

Cloud computing allows provision of infrastructure, platform, and software as services to users with a pay-as-you-go model [1]. Energy consumption of Cloud data centers increases rapidly with the demands on Cloud applications. Besides energy bills, costs associated with cooling and hardware failure due to overheating become critical concerns of Cloud service providers (CSPs) nowadays. Nevertheless, CSPs are required to maintain Quality of Service (QoS) to their subscribers. All these have put CSPs into dilemma situations. With virtualization technology, better resource utilization and thermal distribution can be achieved by allocating virtual machines (VMs) onto physical hosts strategically. In contrast, a poor resource provisioning will lead to undesirable resource utilization and incur performance degradation.

The VM-host allocation problem in Cloud computing can be viewed as a stable matching problem. An early attempt of formulating the VM allocation problem as a stable matching problem was given by Xu and Li [2]. In their work, the two matching party groups are considered as having opposite objectives. Xu and Li tried to maintain fairness between the two groups by achieving an egalitarian stable matching between them. A similar idea was given by Dhillon *et al.* in [3]. In their work, they formulated the VM co-scheduling problem as a cascade of a stable roommates problem and a stable matching problem. A recent work done by Kim *et al.* in [4] suggested to formulate the VM migration process as a Hospital Residents problem. In their work, they considered VMs with equal requirement on resource.

In this paper, a VM allocation mechanism based on stable matching is proposed. In our stable matching framework, hosts and VMs are matched according to their individual preferences. A matching is regarded as stable when no individual would prefer another individual to its current partner. In ordinary stable matching problems, both participating party groups are usually having different and opposite preferences. The matching result can be biased toward either party depending on which party is taking the initiative and being the proposing group. In contrast, VMs and hosts in the proposed mechanism are sharing a mutual objective, that is to consolidate VMs such that active hosts can operate close to a desirable utilization threshold. The proposed mechanism is implemented and evaluated on CloudSim [5] with real-world workload data. Simulation results show that the proposed allocation mechanism can bring significant benefits in terms of energy saving and QoS to Cloud data centers.

The rest of this paper is arranged as follows. Section II introduces and elaborates the proposed VM allocation mechanism. In Section III, performance of the proposed mechanism is evaluated using extensive simulations. The results are further studied and discussed in Section IV. Finally, Section V gives some concluding remarks.

II. Proposed VM Allocation Mechanism

In this work, the scenario under study is based on an Infrastructure as a Service (IaaS) model. Consider a Cloud data center consists of N heterogeneous physical hosts with different resource capacities. At any given time, multiple independent users submit their requests for provisioning M VMs. These VMs, characterized by their requirements, are then allocated to some or all of the physical hosts. In general, a Cloud resource provisioning process can be

Table I
NOMENCLATURE

Symbol	Definition
u_i	Current CPU utilization of VM V_i
U_j	Current CPU utilization of physical host P_j
m_i	MIPS of VM V_i
M_j	MIPS provided by physical host P_j
U_{ij}	Estimated CPU utilization of physical host P_j after the allocation of a migrated VM V_i
$V_{\text{list\_}j}$	A list of migrated VM(s) which regard physical host P_j as their most preferred host.

divided into three major procedures: (1) identifying critical hosts (i.e. overloaded or underutilized hosts), (2) selecting VM(s) on critical hosts to be migrated, and (3) reallocating those VM(s) onto underutilized host(s). Note that the proposed mechanism is designed for the last procedure. To demonstrate the improvement introduced by the proposed method, state-of-the-art mechanisms are adopted in the first two procedures.

Local Regression Robust (LRR) algorithm introduced in [6] is adopted to identify critical hosts in the first procedure of the provisioning process. LRR method is an adaptive predicted threshold detection algorithm. It fits a trend polynomial to the last k CPU utilization observations of the current host to predict the next observation, and thus determines whether a host is going to be overloaded. In the results presented in [6], LRR method outperforms other existing host overloading detection methods. Therefore, we adopted LRR method for detecting overloaded hosts.

In the second procedure, we adopt Minimum Migration Time (MMT) policy in [6] for selecting VMs on critical hosts to be migrated. The MMT policy migrates a VM that requires the shortest time to complete a migration. In [6], it is shown that the MMT policy can produce better results than other existing selection policies for VM selection.

The proposed mechanism is utilized in the last procedure of the process to identify suitable host(s) to accommodate the migrated VM(s). Details on the operation of the proposed mechanism are elaborated as follows.

Step I: Identify the most preferred host of each migrated VM.

(a) Suppose there are $\alpha \leq M$ VMs to be migrated and $\beta \leq N$ physical hosts. Denote the VMs to be migrated as a set $V = \{V_1, V_2, \cdots, V_\alpha\}$ and denote the available hosts as another set $P = \{P_1, P_2, \cdots, P_\beta\}$.

(b) For each VM in V, estimate the utilization of each host in P if such VM is assigned to them as

$$U_{ij} = \frac{U_j M_j + u_i m_i}{M_j}. \quad (1)$$

(c) Compute the difference between a target utilization threshold U_{th} and the estimated utilization U_{ij}. If $U_{\text{th}} \geq U_{ij}$, the difference is calculated as

$$\Delta U_{ij} = U_{\text{th}} - U_{ij}. \quad (2)$$

Otherwise, the host is not considered as a suitable host for migration.

(d) Physical host $P_{k(i)}$ is the most preferred host of a VM V_i, where

$$k(i) = \arg\min_{1 \leq j \leq \beta} \Delta U_{ij}. \quad (3)$$

Multiple VMs can select the same physical host as their most preferred host. Here, we denote $V_{\text{list\_}j}$ as a list of migrated VM(s) which regard physical host P_j as their most preferred host.

Step II: Matching VMs with the hosts.

(a) For each physical host in P, match P_j with a VM V_i in its $V_{\text{list\_}j}$ list, which can yield a minimum ΔU_{ij} .

(b) After each matching, discard $V_{\text{list\_}j}$ and remove V_i from V.

Step III: Repeat Steps I and II if $V \neq \emptyset$, otherwise terminate and return the final migration map.

The pseudocode of the proposed algorithm is presented

Algorithm 1: UT-based Stable Matching

```
Require: hostList,VMsToMigrateList
Ensure: allocation of VMs
 1: for vm in VMsToMigrateList do
 2:     minUTDiff1 ← MAX
 3:     preferredHostList ← NULL
 4:     for host in hostList do
 5:         if host has enough resources for vm then
 6:             UTDiff1 ← estimateUTDiff1(host,vm)
 7:             if UTDiff1<minUTDiff1 then
 8:                 preferredHost ← host
 9:                 minUTDiff1 ← UTDiff1
10:             end if
11:         end if
12:     end for
13:     if preferredHostList ≠ NULL then
14:         preferredHostList.add(vm,preferredHost)
15:     end if
16: end for
17: for preferredHost in preferredHostList do
18:     minUTDiff2 ← MAX
19:     finalSelectedVM ← NULL
20:     for VM in VMsSelectpreferredHost do
21:         UTDiff2 ← estimateUTDiff2(host,vm)
22:         if UTDiff2<minUTDiff2 then
23:             minUTDiff2 ← UTDiff2
24:             finalSelectedVM ← VM
25:         end if
26:     end for
27: end for
28: if finalSelectedVM ≠ NULL then
29:     allocation.add(finalSelectedVM,preferredHost)
30: end if
31: return allocation
```

in Algorithm 1. After each matching, the matched VM(s) would be removed from the VMsToMigrate List. The proposed algorithm is repeated until all the VMs in the VMsToMigrate List are matched. A final migration map is then returned.

III. Simulations

We implemented and evaluated our proposed mechanism on CloudSim [5], a simulation platform that supports modeling of applications and services of cloud infrastructure. In the simulated data center, there are 800 heterogeneous physical hosts including same amount of HP ProLiant G4 servers and G5 servers. Each host has two CPU cores. The core of G4 and G5 servers are assigned with 1860 and 2660 MIPS respectively. The corresponding power models of the physical hosts are obtained from SpecPower08 [7]. The simulated cloud data center comprises four different types of single-core VMs. Each VM is modeled to have 100Mbit/s of bandwidth and 2.5 Gigabytes of VM size. Each simulation is conducted using the data set of one simulated day, and the sampling interval is five minutes. It is essential for CSPs to maintain QoS to their subscribers, while Service-Level Agreement (SLA) violations is usually adopted as an indicator. The level of SLA violation is measured using the two metrics in [6] : (1) SLA violation Time per Active Host (SLATAH); and (2) Performance Degradation due to Migrations (PDM). SLATAH and PDM are independent to each other and are with equal importance. SLA Violation (SLAV), which integrated both metrics, is defined as

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

where SLATAH is the percentage of time which the CPU utilization of active hosts have reached 100% and PDM is the overall performance degradation caused by VM migrations.

Energy consumption and SLA violations are conflicting metrics. A reduction in energy comsumption often implies an increase of SLAV. The proposed allocation mechanism aims to achieve a reasonable trade-off between power consumption and SLAV. Therefore, the metric, Energy and SLAV (ESV) in [6], is adopted to evaluate the overall performance of Cloud data centers under test. ESV is expressed as

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

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

IV. Performance analysis and discussions

Extensive simulations were carried out using real-world workload data to evaluate the performance of the proposed mechanism on CloudSim 3.0.3 [8]. In the simulations, 10 days from the workload traces of PlanetLab [9] are randomly chosen. The utilization threshold of the proposed mechanism has been manually tuned to 0.8. In the simulations, unaltered version of power-based LRR method comes with CloudSim 3.0.3 is used for comparison. Power-based LRR method in [6] is chosen as a benchmark due to its outstanding performance over other existing methods. The key difference between our proposed mechanism and the power-based LRR method is at the VM reallocation procedure. Power-based LRR method regards the VM reallocation problem as a bin packing problem and adopted host's power as a migration criterion. In contrast, the proposed mechanism regards that as a stable matching problem with a mutual objective for both matching parties, which is the main contribution and novelty of this work.

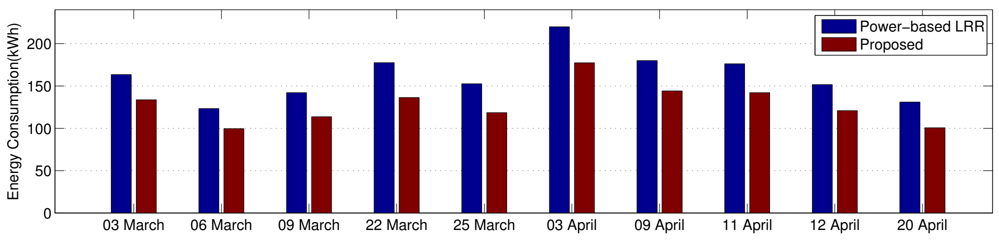
We compared the proposed mechanism with the power-based LRR method over the four aforementioned metrics: energy consumption, SLA violations, migration number, and ESV. Results are summarized in Figure 1. The total energy consumption of Cloud data centers with different VM allocation mechanisms are reported in Figure 1(a). It shows that the proposed method consumed less power than the power-based LRR method. Figure 1(b) shows SLA violations of Cloud data centers with different VM allocation mechanisms. Our proposed mechanism committed significantly fewer SLA violations than its counterpart, which indicates a lower impact on service quality. Each VM migration may result in committing SLA violations, hence it is important to reduce the migration number whenever possible. As observed from Figure 1(c), the proposed mechanism invoked less migrations compared to the benchmarking method. In addition, the ESV metric demonstrates a balanced trade-off between energy consumption and QoS. Systems with lower ESV values are more capable of reducing energy consumption and avoiding SLA violations. From Figure 1(d), it is shown that our proposed mechanism can outperform the power-based LRR method. Under the proposed mechanism, the total utilization of hosts' CPU can be kept close to the target utilization level where hosts can achieve their maximum efficiency.

V. Conclusion

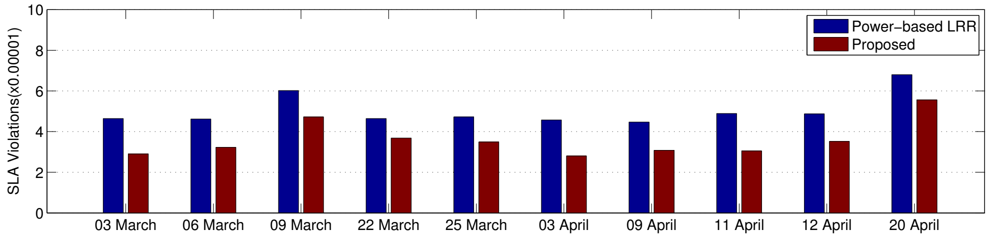
In this paper, a stable matching-based virtual machine (VM) allocation mechanism for Cloud data centers is proposed. The matching parties, VMs and hosts, are sharing a mutual objective, that is to consolidate VMs such that fewer active hosts are needed and active hosts can operate close to a desirable utilization threshold. The performance of the proposed mechanism has been verified using extensive simulations on CloudSim with real-world workload traces. Simulation results show that Cloud data centers with the proposed mechanism can yield lower energy consumption and commit fewer Service-Level Agreement violations.

Acknowledgment

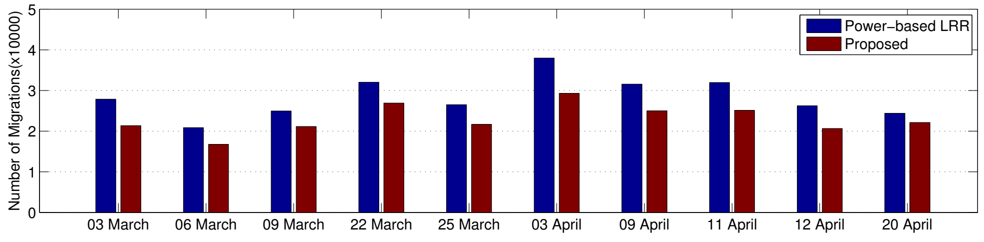
This work is supported by the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University (Projects RTMR, RU9D, and G-YBKH).



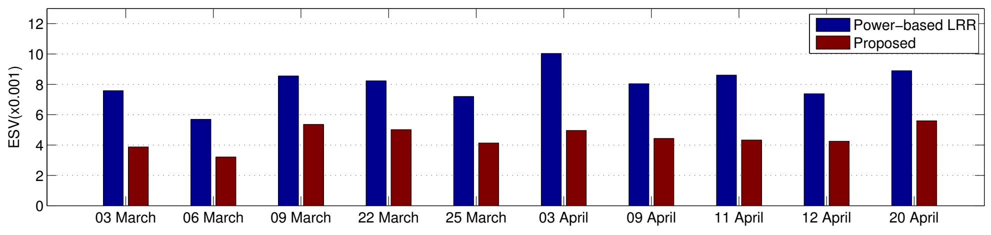
(a) Energy Consumption



(b) SLA Violations



(c) Number of Migrations



(d) Energy and SLA Violations

Figure 1. Comparisons of the VM allocation mechanisms under test

References

[1] F. Xu, F. Liu, H. Jin, and A. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions," *Proceedings of the IEEE*, vol. 102, no. 1, pp. 11–31, Jan 2014.

[2] H. Xu and B. Li, "Egalitarian stable matching for vm migration in cloud computing," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, April 2011, pp. 631–636.

[3] J. Dhillon, S. Purini, and S. Kashyap, "Virtual machine coscheduling: A game theoretic approach," in *Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on*, Dec 2013, pp. 227–234.

[4] G. Kim and W. Lee, "Stable matching with ties for cloud-assisted smart tv services," in *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, Jan 2014, pp. 558–559.

[5] 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.

[6] 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.

[7] "Specpower08," (Accessed: 2015-10-29). [Online]. Available: http://www.spec.org

[8] "Cloudsim," (Accessed: 2016-02-14). [Online]. Available: http://github.com/Cloudslab/cloudsim/releases

[9] 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.