Energy efficient VM migration revisited: SLA assurance and minimum service disruption with available hosts

Philip M. Glasser, Ovunc Kocabas, Burak Kantarci, Tolga Soyata, and Jeanna Matthews

***Abstract*—We propose a VM migration approach named Energy Saving Virtual Machine Migration with Minimum Disruption (ESVM^{3}D) that reduces SLA violations by running VMs in a data center with more available physical hosts as opposed to shutting down idle hosts to save energy. As compared to previously proposed power minimizing VM migration algorithms, simulation results show a 40% reduction in VM migrations and a 10% energy savings as a result of this approach. In summary, ESVM^{3}D achieves a 70% reduction in the number of host shutdowns, resulting in a negligible SLA degradation ($\leq$ 0.1%) as compared to the previously proposed approaches, translating to a similar SLA performance and without a degradation in energy consumption.**

I. Introduction

The Infrastructure as a Service (IaaS) model enables cloud customers to rent cloud computing resources such as virtual machines, physical servers, storage, load balancers and network [1]. The advantages of IaaS are reported as follows: Users can obtain the resources on demand based in a pay-as-you-go fashion such that they do not need to plan their infrastructure in advance. Furthermore, the virtual infrastructure provisioned for a cloud user can scale up based on the future needs of the business. Moreover, maintenance costs of cloud service providers are eliminated as the infrastructure providers take care of data center maintenance. Last but not least, scalability and reliability of cloud services can be guaranteed through IaaS systems [2] subject to service level agreements (SLAs). Virtualization systems such as Xen-based Citrix Cloud [3] or VMWare vCloud [4] are a key component in IaaS, enabling the partitioning of physical computing resources into many virtual machines.

Infrastructure as a Service (IaaS) clouds partition computing and storage hardware on physical servers into multiple virtual machines (VMs) so that multiple users can utilize the same physical medium without interfering with each other. Over time, it can be good to move VMs from one physical machine to another for reasons such as energy minimization, CPU utilization and/or maintenance. This is widely known as VM migration in the literature.

Philip M. Glasser and Burak Kantarci are with the Department of Electrical and Computer Engineering at Clarkson University, Potsdam, NY, USA, 13699. email: {glassepm,bkantarc}@clarkson.edu

Ovunc Kocabas and Tolga Soyata are with the Department of Electrical and Computer Engineering at University of Rochester, Rochester, NY, 14627, emails: {kocabas,soyata}@ece.rochester.edu

Jeanna Mattthews is with the Department of Computer Science at Clarkson University, Potsdam, NY, USA, 13699. email: jnm@clarkson.edu

Besides partitioning the hardware among multiple clients, virtualization also enables virtual machine migration (VMM) or dynamically moving a virtual machine between physical hosts. Benefits of virtual machine migration are server consolidation, load balancing, improved data and network locality, facilitated maintenance, reduced host costs, and reduced energy costs/carbon footprint [5]. In particular, the idea behind VM migration for reduced energy costs and carbon footprint is two-fold: i) Putting the idle hosts into sleep mode in order to save idle power consumption, and ii) Migrating VMs away from overloaded hosts to avoid hotspots in the data center [6]. It is worth noting that hotspots also increase the cooling costs and energy consumed to cool the hot aisles in the data center. Figure 1 illustrates a simple VM migration scenario where host-A is operating above a redline temperature. In order to prevent Host-A from becoming a hotspot, the virtual infrastructure manager (i.e., resource broker) migrates a VM. VM-2 is identified as the best partition to be migrated. Host-B is found to be the future host of VM-2 based on a pre-defined search criteria. Once VM-2 is migrated, both Host-A and Host-B are operating below the redline temperature.

There have been several proposals to reduce power consumption and/or improve energy efficiency in data centers via VM migration [7]–[11]. Related work also aims at addressing migration latency, performance degradation due to VM migration and service downtime [12]–[14]. In this paper, we revisit the VM migration as a tool for improving enercy cost and carbon footprint. We adopt a previously proposed heuristic, Power-Aware Best Fit Decreasing (PABFD) algorithm, that balances energy savings as well as overall SLA time [6] by aiming at minimum disruption and minimum SLA violation while reducing the energy consumption in the data center. Our proposed approach is called Energy Saving Virtual Machine Migration with Minimum Disruption (ESVM^{3}D). When searching for a physical host to which we can migrate a VM, we use a multi-objective search that leads to minimum marginal SLA if the VM is placed on a candidate host, and maximum marginal CPU utilization on the corresponding host. Additionally, we define an aggressive mode for the proposed scheme, which further defines a marginal energy consumption criterion besides these two metrics. ESVM^{3}D is evaluated through simulations, and we show that disruption is minimized as the number of VM migrations can be reduced by 40% while energy consumption can be further reduced by 10% when compared to a previously proposed power minimizing VM migration approach. Furthermore, SLA degradation due

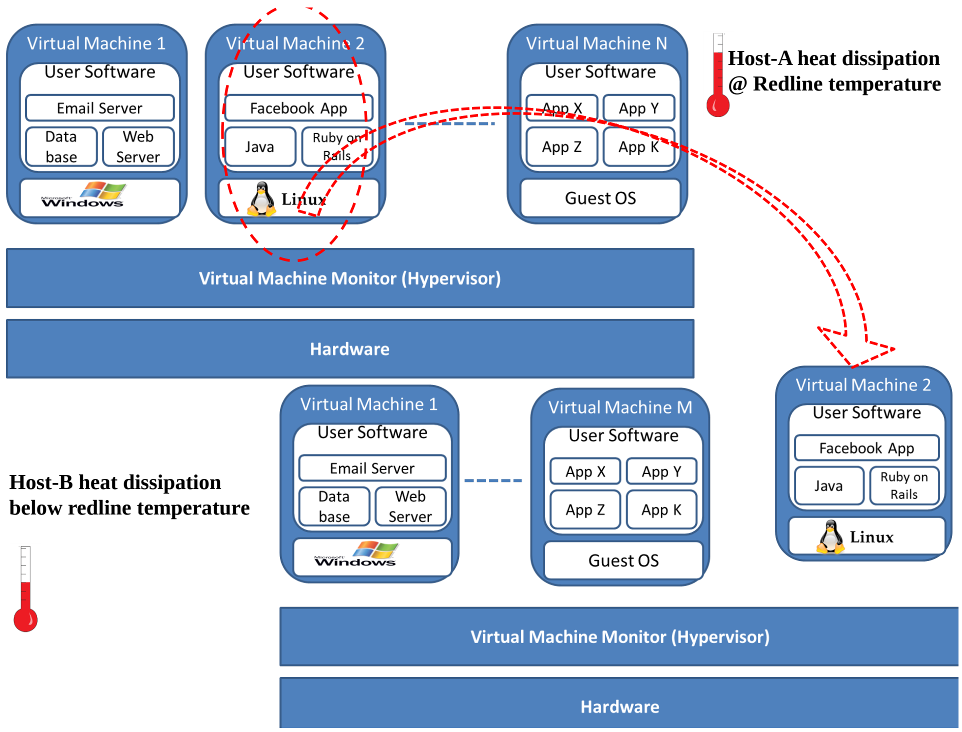

Fig. 1: Virtual machine migration to avoid hot spots in a data center.

to migration is 0.10% under the proposed approach while the previously proposes scheme can lead to SLA violation at the order of 14%. Moreover, the number of hosts that are shutdown can be reduced by 70% so more physical hosts are maintained available for future requests without any degradation in energy consumption or SLA performance.

This paper is organized as follows. Section II presents related work. We describe our proposal in detail in Section III. Numerical results of performance evaluation, as well as related discussions are provided in Section IV. We finally conclude the paper and give future directions in Section V.

II. Background

In [5], service discontinuity has been reported as a notable cost of VM migration. The authors report that VM migration leads to around 3 seconds of downtime and over 44 seconds of downgraded performance. Moreover, the authors also report that SLA violations lead to up to 20% penalty during VM migration. Indeed, VM migration costs include security vulnerabilities as well however VM security is out of scope of this paper.

Sandpiper is a VM migration scheme which aims to avoid machine overloading [15]. The control plane of Sandpiper consists of a profiling engine, hotspot detector and a migration manager. Status of the physical hosts are collected by the profiling engine and are reported to the hotspot detector. The migration manager takes care of the VMs be migrated, and determines the destination host.

In [16], the authors propose a VM migratino scheme that aims to achieve dynamic server consolidation constrained to VM capacity requirements such as CPU and memory. The authors partition the VM migration problem into two stages. The first stage is called the VM packing problem whereas the second stage is the VM replacement problem. The authors in [17] consider the physical conditions in the data center and aim to address the cooling costs which is introduced by heat imbalance between the aisles of the active physical hosts. Thus, besides the CPU utilization, power consumption and memory requirements, it aims at reducing unstable host selection during VM migration.

The authors in [12] propose a VM migration planning scheme to reduce the service downtime and the migration time. The proposed approach basically determines the execution order of massive VM migration in a data center so that when the final VM-physical host mapping is reached, data center will be impacted minimally in terms of time and performance.

As VM migration utilizes data center network links, data center network performance is also affected by VM migration. As reported by the authors in [13], aggressive utilization of available resources may introduce network performance degradation in the data center. To this end, the authors propose static and dynamic VM migration approaches are proposed which aims at maximum energy savings and minimized maximum network link utilization.

In [14], the authors propose naming the VMs with the services running on them and routes towards destination hosts are selected by using means of named data networking in lieu of IP addressing. By doing so, it is aimed at the services are not interrupted during migration. The authors further propose a load balancing algorithm to optimize the performance of named data networking-based VM migration.

The authors in [6] propose several heuristics that aim to optimize energy consumption and SLA violation times in a data center. Among those schemes, power-aware best fit decreasing succeeds in significantly reducing the energy consumption and also reduces SLA violations when compared to its SLA-unaware counterparts. Moreover, the algorithm is practical and easy to implement in real data centers. Therefore, it is this scheme we chose to compare with our algorithm in this paper.

III. Energy-efficient VM migration with minimum disruption (ESVM^{3}D)

Our proposal adopts the online adaptive heuristics in [6] and integrates a lightweight heuristic for VM placement under the condition denoting the overloaded and underutilized physical hosts. The steps of the proposed approach are briefly presented in Algorithm 1. Thus, the proposed solution takes the list of active hosts as the input, and for each active host, it checks if the host is overloaded. In case a physical host is overloaded, the algorithm decides which VMs to migrate to other physical hosts. Once the list of VMs to be migrated is obtained, a resource broker runs a local search based heuristic to determine the new physical host-VM mapping. Detection of overloaded hosts can be done via various methods. In this paper, we adopt the local regression-based overloaded host detection in [6]. Thus, the timeline for the observation on CPU utilization of host-i is partitioned into q slots, and based on the CPU utilization in millions instructions per second (MIPS) on host-i (ρ_i^j), a polynomial curve is aimed to be fitted. For each observation-j on host-i at t_i^j, a trend line, $\rho_i^j = b_i \cdot t_i^j + a_i$ is aimed to be fit where a_i and b_i coefficients are obtained through minimization of the function in Eq. 1. In the equation, observation t_i^k denotes the middle of the entire observation window for host-ii, i.e., $k = \lceil q/2 \rceil$ whereas t_i^l denotes the k^{th} observation from the right boundary of the observation window. Once the coefficients are obtained, the next value of CPU utilization (i.e., ρ_i^{k+1}) is estimated through ρ_i^k, and the host is marked as overloaded if the conditions in (2) hold. In the equation, t_{min} is the upper bound for VM migration duration between two physical hosts. For further details of the regression, the reader is referred to [6].

$$min \sum_j^k \left(1 - (\frac{t_i^k - t_i^j}{t_i^k - t_i^l})^3\right)^3 (\rho_i^j - a_i - b_i t_i^j)^2 \qquad (1)$$

$$s \cdot \rho_i^{k+1} \geq 1 \; and \; t_i^{k+1} - t_i^k \leq t_{mig}; \;\; s \in \mathbb{R}^+ \qquad (2)$$

Upon migrating the VMs of the overloaded hosts, the original VM migration algorithm in [6] proceeds with detecting the underloaded hosts. The underloaded hosts are potentially to be shut down once their VMs can be successfully migrated to other active physical hosts. For each underloaded host, the algorithm retrieves the list of VMs to be migrated, and repeats the steps for VM migration from overloaded hosts. We aim at improving the performance of this approach by reducing the number of underutilized hosts. As mentioned before, the novelty of the proposed solution is the local search based lightweight heuristic introduced to the VM migration heuristic in [6] to determine new VM-host mapping in case overloaded hosts are detected with the ultimate goal of energy efficiency and minimum disruption.

In the heuristic, the resource broker and participating physical hosts aim at maximizing their utility. Utility is defined as the inverse of CPU utilization for a physical host while the resource broker defines the utility as the inverse of the weighted sum of energy consumption and SLA violation in the data center. These two expressions are formulated in Eq. 3 and Eq. 4, respectively.

$$U_i^h = 1/(\rho_i) \qquad (3)$$

$$U^{rb} = \frac{1}{\alpha(\xi) + (1-\alpha)(SLATAH \times PDM)} \qquad (4)$$

In these equations, ρ_i denotes the CPU utilization of physical host-i in terms of millions instructions per second (MIPS), and ξ denotes the overall energy consumption in the data center. Furthermore, SLA violation is defined as the product of SLA violation time per active host ($SLATAH$) and performance degradation due to migration (PDM). As seen in the equation, $\alpha \in [0, 1]$ is a weight factor which prioritizes SLA violation over energy consumption or vice versa. As in most applications, SLA violation and energy efficiency are equally important; hence we design our proposed procedure accordingly. Moreover, delays due to shutting down and starting up the physical hosts lead to inefficiencies in energy savings [18] and reduced lifetimes of the disks in case of frequent shutdown and startup [19]. On the other hand, shutting down the idle or underutilized hosts introduce savings in power consumption due to CPU utilization, as well as the power needed to cool the servers and the clusters [20], [21]. Therefore, the resource broker has three objectives as follows: i) Maximum utility per host, i.e., overall energy savings (see Eq. 3), ii) Maximum utility for the resource broker, i.e., minimum SLA violation due to migration (see Eq. 4), and iii) Minimum host shutdowns.

As ξ is a function of total CPU utilization in the data center, we set $\alpha = 0$ to avoid energy savings' suppressing the SLA violation objective. Thus, we call this operation mode *the non-aggressive mode* where the resource broker maximizes its utility through the objectives $ii{-}iii$ whereas the physical hosts maximize their utility through the objective i. It is worthwhile noting that we categorize the objectives $ii - iii$ as *minimum disruption goals*.

As shown in the pseudo-code, in order to meet the objective-$i - iii$, for each VM to be migrated, the algorithm seeks the physical host that would lead to maximum marginal CPU utilization ($\Delta\rho_{h'}$) where h' is a potential destination host for the corresponding VM. Thus, the algorithm searches for the physical host that would experience minimum marginal utility. As opposed to the conventional approaches, our proposed approach aims at placing the VM on the physical host that is expected to lead to higher CPU utilization and consequently higher increase in its energy consumption as formulated in Eq. 5. In the equation, $\rho_{h'}^t(V_{h'})$ denotes the CPU utilization at host-h' when the VMs in the list $V_{h'}$ are allocated on it. Having reduced the energy consumption of the original host will enable meeting the objective-i whereas selection of the host with minimum marginal utility change will help reduce the number of underutilized hosts and in turn resulting with mitigating the shutdown and restart delays (i.e., disruption) due to migration. Furthermore, in order to meet the goals in $ii - iii$, the algorithm searches for the host that would lead to maximum marginal utility for the resource broker, i.e., minimum SLA violation due to migration. In the pseudo-code, by $\Delta SLAV_{h'}^{\nu}$ we denote the change in SLA violation at the potential host h' before and after allocating resources for the VM-ν as formulated in Eq. 6 where $SLAV_{h'}^{V_{h'}}$ denotes the SLA violation at the physical host-h' with the VMs allocated on itself ($V_{h'}$).

$$\Delta\rho_{h'}^{\nu} = \rho_{h'}^{t+1}(V_{h'} \cup \{\nu\}) - \rho_{h'}^{t}(V_{h'}) \tag{5}$$

$$\Delta SLAV_{h'}^{\nu} = SLAV_{h'}^{V_{h'} \cup \{\nu\}} - SLAV_{h'}^{V_{h'}} \tag{6}$$

As mentioned above, if the utility of the resource broker were defined as a function of the overall energy consumption and the SLA violation due to migration, objective-i is expected to aggressively suppress objectives-$ii - iii$. In that case, the following search function would have been added to the related block of the algorithm:

$$argmin_{h' \in H, h' \neq h}\{\Delta\xi\};$$

Thus, in the *aggressive mode*, for a VM in migration, the algorithm searches for the physical host $i)$ whose CPU utilization will increase the most among all candidate physical hosts, $ii)$ that will lead to minimum increase in SLA violation due to migration, and $iii)$ that will lead to the minimum increase in the overall power consumption of the data center. Since overall power consumption (ξ) is a function of the list of CPU utilization of the physical hosts ($\{\rho_h\}$), power savings are included twice in the heuristic, i.e., explicitly through Δxi and implicitly through checking $\Delta\rho_{h'}^{\nu}$.

Algorithm 1 Pseudocode of the proposed VM migration heuristic

```
Begin
for each h ∈ H do
  if (h is overloaded) then
    VML ← getVMsToMigrate(h)
    // New VM mappping
    for each ν ∈ VML do
      h' ← {argmin_{h'∈H,h'≠h}{ΔSLAV^ν_{h'}};
      argmax_{h'∈H,h'≠h}{Δρ^ν_{h'}};
      h' is not overloaded by ν
      Allocate h' for ν
    end for
  end if
  VML.clear()
end for
for each h ∈ H do
  if (h is underloaded) then
    VML ← getVMsToMigrate(h)
    // New VM mappping
    for each ν ∈ VML do
      h' ← {argmin_{h'∈H,h'≠h}{ΔSLAV^ν_{h'}};
      argmax_{h'∈H,h'≠h}{Δρ^ν_{h'}};
      h' is not overloaded by ν
      Allocate h' for ν
    end for
  end if
  if (h is idle) then
    Shutdown h
  end if
end for
return Map(H,V)
End
```

IV. Numerical Results

We evaluated our proposed scheme and compared its performance to the previously proposed Power-Aware Best Fit Decreasing (PABFD) algorithm in [6], and adopted the same simulation settings in our study. To this end, we used the CloudSim simulator which was implemented in Java [22]. The datacenter is generated with 800 physical nodes consisting of HP ProLiant ML110 G4 (1860 MIPS/core) and HP ProLiant ML110 G5 (2660 MIPS/core) servers. The former consumes 86 W and 117 W in idle and fully utilized status, respectively, whereas the latter consumes 93.7 W when it is idle and 135 W when it is 100% utilized. Power consumption of these servers at different load levels between 0–100% can be found in [6]. We ran simulations for one virtual day (i.e., 86,400 seconds), and the workload profile for the day was selected randomly from the monitoring data in [23]. Thus, during the selected days, the number of VMs varies within 898–1463 while the average CPU utilization varies within 9.26%–12.39% with standard deviations 12.78%–17.09%, respectively.

Figure 2 illustrates the energy consumption under PABFD and the two modes (i.e., aggressive and non-aggressive) of our proposed VM migration method. As seen in the figure, under the aggressive mode, the proposed scheme reduces the energy consumption by 5.7% whereas the non-aggressive mode reduces the power consumption of PABFD by 8.5%. As mentioned above, seeking the physical host which will lead to the highest marginal CPU utilization ensures that underutilized hosts get assigned VMs such that the average workload is lower improving user utility.

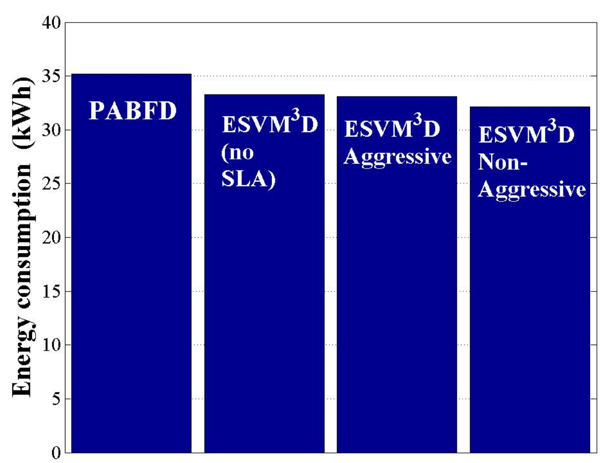

Fig. 2: Energy consumption under the benchmark scheme and the proposed scheme without SLA-awareness, agreessive mode and non-aggressive mode.

Figure 3 depicts the SLA violation, as well as the SLA violation due to VM migration under the three methods.

Under the aggressive mode of the proposed scheme, the SLA violation is increased by approximately 10% whereas the non-aggressive mode achieves the same level of SLA with PABFD by making a compromise between the energy efficiency and SLA-awareness. Furthermore, since fewer hosts are left underutilized under the proposed scheme, the resource broker requests fewer host shutdowns and fewer VM migrations as shown in the next figure. Hence, the proposed scheme under its non-aggressive mode does not lead to any further performance degradation in comparison to its predecessor, PABFD.

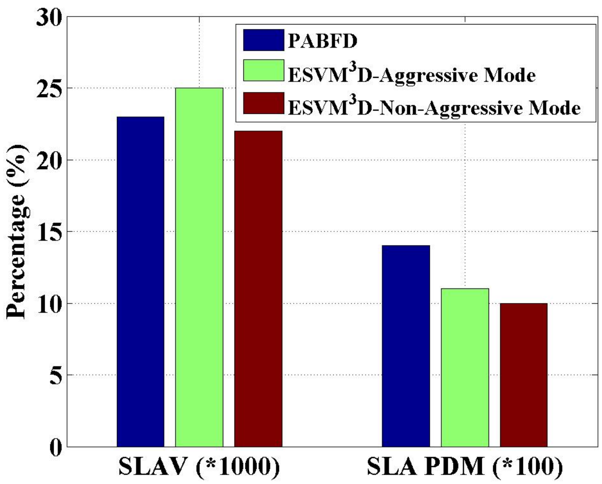

Fig. 3: SLA violation and SLA performance degradation due to migration.

Figure 4 presents the number of host shutdowns and the VM migrations under the three schemes. As seen in the figure, regardless of its mode, the proposed approach significantly reduces the number of host shutdowns, which is 6% and 7% under the aggressive and non-aggressive modes, respectively. Furthermore, the number of VM migrations is reduced by 35% and 39% under the aggressive and non-aggressive modes, respectively. This suggests that the energy usage and SLA violation due to migration is correlated to these two metrics. Though PABFD has the lowest overall SLA violations (as shown in Fig. 5), it needs improvement in terms of the utility of the hosts as it maximizes workload on each host which can cause migrations to happen more often as shown in Fig. 4.

V. Conclusions and Future Work

Hardware virtualization enables dynamically moving a virtual machine (VM) between physical hosts for the sake of improved utility and reduced operational expenditures due to energy costs. In this paper, we have proposed a VM migration scheme which adopts a previously proposed power-aware best fit decreasing method [6] and improves it in terms of energy consumption and SLA violation due to migration. The proposed scheme is called Energy Saving VM Migration with Minimum Disruption (ESVM^{3}D). Our proposed scheme adopts the same local regression technique to detect overloaded hosts which require VM migration. When VM migration is in progress, it uses marginal CPU utilization and marginal SLA violation on a potential destination hosts as a key metric while it aims at minimizing the marginal SLA violation and maximizing the marginal CPU utilization. The idea behind maximizing the marginal CPU utilization is that the proposed scheme aims at leaving fewer underutilized hosts so that SLA violation due to migration is reduced by fewer host shutdowns and fewer VM migrations. We have shown that the savings in consolidating VMs and then shutting down underutilized hosts under the power minimizing benchmark scheme does not outweigh the savings gained from the proposes solution which also aims to optimize workload for the hosts. The proposed scheme operates in two modes, namely aggressive and non-aggressive modes where the former suppresses the energy metric over the SLA violation. It has been shown that regardless of its operation mode, the proposed VM migration technique makes a compromise between energy use in the data center, SLA violations, and migration overheads.

This work is currently being extended by considering the physical constraints and conditions in cloud data centers such as consideration of cooling costs by using thermal maps generated through fluid mechanics models. We are investigating the impact of reduced disruption by ESVM^3 on cooling energy consumption, and aiming at addressing possible trade-offs.

References

[1] Serrano Nicolas, Gorka Gallardo, and Josune Hernantes, "Infrastructure as a service and cloud technologies," *IEEE Software*, vol. 32, no. 2, pp. 30–36, 2015.

[2] Rajkumar Buyya, James Broberg, and Andrzej M Goscinski, *Cloud Computing: Principles and Paradigms*, Wiley, ISBN: 978-0-470-88799-8.

[3] Jiaxin Wang, Lianhe Yang, Miao Yu, and Shuai Wang, "Application of server virtualization technology based on citrix xenserver in the information center of the public security bureau and fire service department," in *Proceedings of the 11th Int. Symp. on Computer Science and Society*. IEEE, 2011, pp. 200–202.

[4] Pradeep Padala, "Resource management in vmware powered cloud: Concepts and techniques," in *Proceedings of the 27th Intl. Symp. on Parallel and Distributed Processing (IPDPS)*. IEEE, 2013, p. 581.

[5] Raouf Boutaba, Qi Zhang, and Mohamed Faten Zhani, "Virtual machine migration in cloud computing environments: Benefits, challenges, and approaches," in *Communication Infrastructures for Cloud Computing*, Hussein Mouftah and Burak Kantarci, Eds. IGI Global.

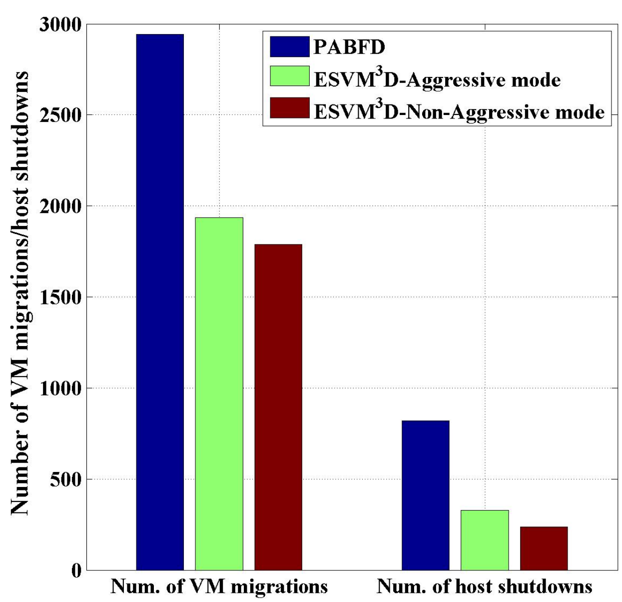

Fig. 4: Total VM migrations and number of physical host shutdowns.

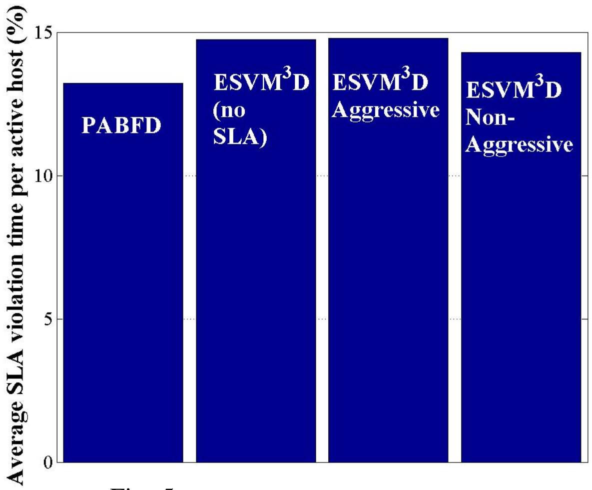

Fig. 5: Average SLA time per active host.

[6] Anton Beloglazov and Rajkumar 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.

[7] Carlo Mastroianni, Michela Meo, and Giuseppe Papuzzo, "Probabilistic consolidation of virtual machines in self-organizing cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 215–228, 2013.

[8] Fahimeh Faraknakian, Adnan Ashraf, Tapio Pahikkala, Pasi Liljeberg, Juha Plosila, Ivan Porres, and Hannu Tenhunen, "Using ant colony system to consolidate vms for green cloud computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187–198, 2015.

[9] Pablo Graubner, Matthias Schmidt, and Bernd Freisleben, "Energy-efficient virtual machine consolidation," *IEEE IT Professional*, vol. 15, no. 2, pp. 28–34, 2013.

[10] Canturk Isci, Juixing Liu, Bulent Abali, Jeffrey O Kephart, and Jack Kouloheris, "Improving server utilization using fast virtual machine migration," *IBM Journal of Research and Development*, vol. 55, no. 6, pp. 4:1–4:12, 2011.

[11] Yang Song, Haoliang Wang, and Tolga Soyata, "Hardware and software aspects of vm-based mobile-cloud offloading," in *Enabling Real-Time Mobile Cloud Computing through Emerging Technologies*, Tolga Soyata, Ed., chapter 8, pp. 247–271. IGI Global, 2015.

[12] Faizul Md Bari, Mohamed Faten Zhani, Qi Zhang, Reaz Ahmed, and Raouf Boutaba, "Cqncr: Optimal vm migration planning in cloud data centers," in *Proceedings of the IFIP Networking Conference*. IFIP, 2014, pp. 1–9.

[13] Dong Jiankang, Wang Hongbo, and Cheng Shiduan, "Energy-performance tradeoffs in iaas cloud with virtual machine scheduling," *China Communications*, vol. 12, no. 2, pp. 155–166, 2015.

[14] Ruitao Xie, Yonggang Wen, Xiaohoua Jia, and Haiyong Xie, "Supporting seamless virtual machine migration via named data networking in cloud data center," *IEEE Transactions on Services Computing*, pp. 1–14, 2015 (accepted).

[15] Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif, "Sandpiper: Black-box and gray-box resource management for virtual machines," *Comput. Netw.*, vol. 53, no. 17, pp. 2923–2938, 2009.

[16] Fabien Hermenier, Xavier Lorca, Jean-Marc Menaud, Gilles Muller, and Julia Lawall, "Entropy: A consolidation manager for clusters," in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*. 2009, pp. 41–50, ACM.

[17] Jing Xu and José Fortes, "A multi-objective approach to virtual machine management in datacenters," in *Proceedings of the 8th ACM International Conference on Autonomic Computing*. 2011, pp. 225–234, ACM.

[18] Chang-Hao Tsai, Kang G. Shin, and John Reumann, "Online web cluster capacity estimation and its application to energy conservation," *IEEE Transactions on PArallel and Distributed Systems*, vol. 18, no. 7, pp. 932–945, 2007.

[19] K R Krish, Guanying Wang, Puranjoy Bhattacharjee, Ali R Butt, and Chris Gniady, "On reducing energy management delays in disks," *Journal of Parallel and Distributed Computing*, vol. 73, no. 6, pp. 823–835, 2013.

[20] Vimal Mathew, Ramesh K. Sitaraman, and Prashant Shenoy, "Energy-efficient content delivery networks using cluster shutdown," *Sustainable Computing: Informatics and Systems*, vol. 6, pp. 58–68, 2015.

[21] Kashif Bilal, Saif Ur Rehman Malik, Osman Khalid, Abdul Hameed, Enrique Alvarez, Vidura Wijaysekara, Rizwana Irfan, Sarjan Shrestha, Debjyoti Dwivedy, Mazhar Ali, Usman Shahid Khan, Assad Abbas, Naumanl Jalil, and Samee U. Khan, "A taxonomy and survey on green data center networks," *Future Generation Computer Systems*, vol. 36, no. 1, pp. 189–208, 2014.

[22] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. DeRose, and Rajkumar Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience (SPE)*, vol. 41, no. 1, pp. 23–50, 2011.

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