Towards Energy-Efficient and Real-Time Cloud Computing

Taha Tekreeti, Ting Cao, Xiaopu Peng, Tathagata Bhattacharya
Jianzhou Mao, Xiao Qin, and Wei-Shinn Ku
Department of Computer Science and Software Engineering Auburn University - Auburn, AL
Email: {tka0002, tzc0028, xzp0007, tzb0063, jzm0122, xqin, weishinn}@auburn.edu

2021 IEEE International Conference on Networking, Architecture and Storage (NAS) | 978-1-7281-7744-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/NAS51552.2021.9605453

***Abstract*—In modern cloud computing environments, there is a tremendous growth of data to be stored and managed in data centers. Large-scale data centers demand high utilization of computing and storage resources, which lead to expensive operational cost for energy usage. Evidence shows that consolidating virtual machines (VMs) can conserve energy consumption in clouds through VM migrations. VM-consolidation techniques, however, inevitably induce a burden on performance. To address this issue, we propose a holistic solution - *EGRET* - to boost energy efficiency of cloud computing platforms by seamlessly integrating the DVFS scheme with the VM-consolidation technique. EGRET dynamically determines the most energy-efficient strategy by issuing a command to either scale CPU frequencies on a VM or marking the VM as underutilized. We conduct extensive experiments to evaluate the performance of EGRET. The experimental results show that EGRET substantially improves the energy efficiency of cloud computing platforms.**

I. Introduction

We develop a DVFS-based consolidation strategy - *EGRET* - for virtual machines running real-time applications, aiming to uplift energy efficiency of cloud computing infrastructures. This research is motivated by the following three technological trends in cloud computing.

Motivation 1: Representative real-time services made available in cloud computing systems include video streaming [1], real-time location tracking [2], [3], and real-time social networks [4]. Most of these creative and practical real-time services are implemented and maintained in real-world computing clouds like google clouds [5] and Microsoft clouds [6].

Motivation 2: Enterprises shifting to the cloud sets high expectations and requirements, from the services provided by cloud-providers, to maintain a competitive edge in the market [7]. These requirements mean providing reliable, distributed and load-balanced services which translates to more data center, more power consumption and thus more heat dissipation and green-house gas emissions.

Motivation 3: Running idle and under-utilized servers is one of the most prominent reasons in energy wasted by cloud data centers. These servers can consume up to 50% of peak power [8]. Therefore, Virtual Machine (VM) consolidation techniques have been presented to minimize the power loss from under-utilized machines.

II. Related Work

A. Virtual Machine Consolidation

An myriad of studies centered around virtual machine (VM) consolidation techniques by moving workloads away from under-utilized VMs and shutting down corresponding physical machines in order to conserve power [9]. Unlike the aforementioned VM-consolidation technique, our proposed EGRET optimizes the energy efficiency of clouds by seamlessly integrating a VM-consolidation scheme with the practical DVFS technique.

B. Dynamic Voltage and Frequency Scaling

Energy-efficient scheduling plays a major role in reducing energy consumption of large-scale computing systems. For example, to implement high energy-efficient clusters, Zong *et al.* proposed two duplication-based scheduling techniques - EAD and PEBD algorithms [10]. Different from the existing DFVS-based schedulers, our EGRET combines the DVFS and VM-consolidation techniques to form a hybrid energy-saving approach.

III. System Framework

EGRET framework consists of two modules - a scheduling module (see Section III-A) and a resource management module (see Section III-B).

A. The EGRET Scheduling Module

The EGRET scheduling module is comprised of two key pieces, namely, a coordinator and a scheduler. The coordinator serves as a bridge between the scheduler in the EGRET module and the two components in the resource manager (see Section III-B). On one hand, the coordinator gathers critical DVFS and VM status information from the resource manager; on the other hand, the coordinator relays scheduling information to the resource manager. The scheduling decisions along with the DVFS and VM-consolidation decisions are made by the EGRET module, which relies on the coordinator to pass on the decisions to the resource manager.

B. The Resource-Management Module

The scheduling algorithm searches available servers to create VMs for allocating a job under the requirements of the job. In the algorithm, the SLA also has to be taken into account. The scheduling algorithm selects the solution that consumes the least energy. If all available servers cannot satisfy the requirements of the job, the scheduling algorithm has to notice the VM manager to

978-1-7281-7744-1/21/$31.00 ©2021 IEEE
Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:12:50 UTC from IEEE Xplore. Restrictions apply.

power on proper servers. The VM manager then creates required VMs from relative servers and deploys the job on these VMs assigned by the scheduling algorithm. The VM manager also has to set up the working frequency of the server and shut down the servers or VMs that have finished their works or issue a VM consolidation/migration based on workload variations determined by the EGRET scheduling algorithm.

IV. Frequency-Aware VM Consolidations

In this section we propose the dynamic real-time algorithm - *EGRET* - to leverage the DVFS model to conserve energy by consolidating virtual machines.

A. Dynamic Voltage and Frequency Scaling in EGRET

EGRET makes use of the DVFS technique to reduce PM power consumption by scaling down both voltage and CPU frequency. Energy savings tend to have a toll on a system's performance, which in turn may violate SLA requirements. To guarantee the prescribed SLAs, the DVFS module in our EGRET ought to manipulate frequency/voltage in a way that prolonged execution times still meet the SLA requirements. EGRET overcomes this problem by calculating the optimal frequency to be used in correlation with the current utilization of a host machine. Such a correlationship is calculated by taking into account the CPU performance measured in terms of *MIPS* (a.k.a., million instructions per second), which is a popular metric used to quantify the CPU performance of a physical machine.

Algorithm 1: The DVFS frequency changer algorithm.

Result: A set F_{conf} of optimal frequencies for all the hosts h in host set H.

initialization;
foreach $h \in H$ **do**
  $u_h \leftarrow get_host_utilization(h)$;
  foreach $f_{req} \in F_h$ **do**
    $u_h^a \leftarrow \frac{f_{req}}{f_{Max}}$;
    if $u_h^a \geq u_h$ **then**
      $f_h \leftarrow f_{req}$;
    end
  end
  $F_{conf} \leftarrow put_optimal_frequency(f_h)$;
end
return F_{conf};

Prior to presenting our frequency selection algorithm, we introduce the important notation used throughout this manuscript. Let $speed_{conf}$ and $speed_{max}$ be the configured and maximum CPU speed measured in MIPS. We denote f_{max} as the maximum frequency that enable the CPU to run at its full capacity of $speed_{max}$. By the same token, we refer to f_{conf} as a required frequency that leads to the CPU speed of $Speed_{conf}$.

Given a host h and a specified task in the inner loop (see also **foreach** $f_{req} \in F_h$ **do**), Algorithm 1 discovers the most efficient f_{req} customized for accomplishing the task by iterating through all the candidate frequencies in set F_h. The most appropriate frequency f_h for host h, of course, must fulfill the task's utilization requirement. Such a utilization requirement check is furnished by the boolean expression $u_h^a \geq u_h$. After determining the optimal frequency f_h of host h, the optimal frequency is augmented into the set F_{conf} of optimal frequencies for all the hosts (see $F_{conf} \leftarrow put_optimal_frequency(f_h)$ in the algorithm). Finally, the algorithm returns the optimal frequency set F_{conf}.

B. Dynamic VM Consolidation in EGRET

We advocate for VM consolidation as an alternative to dynamic frequency CPU scaling in the case where DFVS fails to maintain good balance between energy efficiency and SLA. The VM consolidation procedure consists of the following four stages:

- To detect over-utilized physical machines or PMs.
- To detect under-utilized physical machines or PMs.
- To identify new hosts to which VMs are placed.
- To migrate VMs from over- or under- utilized PMs to new PMs.

C. DVFS-aware VM migrations

To make a wise tradeoff between performance and energy efficiency, we propose a frequency-aware virtual-machine placement algorithm. The overarching goal of the algorithm is make optimal placement decisions by electing the most appropriate target physical machines, to which virtual machines should be migrated. At the heart of our design is the best fit decreasing algorithm or BFD [11]. We choose to leverage the *BFD* algorithm to govern the virtual machine migrations in our system.

Algorithm 2: BFD-enabled VM Placement

Data: VM, PM
Result: VM placement into PM

$initialization(VM, PM)$;
$l_{vm} \leftarrow get_decreased_utilization_order(VM)$;
foreach $vm \in VM$ **do**
  $p_{min} \leftarrow p_{max}$;
  $map_h \leftarrow NULL$;
  if $u_h \geq u_(h, vm)$ **then**
    $p \leftarrow get_power_usage_(h, vm)$;
    if $p \leq p_{min}$ **then**
      $map_h \leftarrow h$;
      $p_{min} \leftarrow p$;
    end
  end
  if $map_h \neq NULL$ **then**
    $placement.add(map_h, vm)$;
  end
end
return $placement$;

Algorithm 2 presents a BFD-enabled scheme to handle virtual machine placement. Taking a set of virtual machines (VM) and a set of physical machines (PM) as an input, the algorithm

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:12:50 UTC from IEEE Xplore. Restrictions apply.

originates the VM placement decisions. Our algorithm strives to elect a physical machine that embraces the least adverse impacts from VM migrations in terms of increased frequency. To achieve this goal, our strategy first sorts the list of VMs with their estimated CPU utilization in a decreasing order (see Line 2 in Algorithm 2). Then, the algorithm seeks a host that can accommodate the migrated VM with the minimal increase in overall power consumption (see Lines 3-12). Incorporating BFD, our algorithm nominates the tightest physical machine to host newly migrated virtual machines (see Lines 8-11). After each placement attempt, Algorithm 2 updates the current utilization of each physical host for future placements (see Lines 13-15). Then, we incorporate our DVFS scaling algorithm on the host in order to reduce the power consumption more.

V. Experimental Framework

In this section, we present the configuration settings of the implemented simulator that processes a real-world workload data. We rely on simulation studies, because the simulator is capable of resembling the behaviors of large-scale data centers. We expand the CloudSim 3.0 simulator [12] by incorporating our proposed EGRET module.

A. Workload Conditions

EGRET-powered CloudSim is driven by real datasets harvested from the CoMon project [12]. To resemble real-world scenarios, we advocate for the *PlanetLab* workload traces that were recorded to track CPU utilization of more than a thousand VMs placed on physical hosts located in various locations around the world. More specifically, the traces were collected from randomly selected datasets "20110303, 20110325 and 20110420" to conduct our simulations.

B. VM Assignment Policies

Before diving into the performance analysis, we shed some light on the VM assignment policies implemented in CloudSim.

1) Overloaded and Under-loaded Hosts Detection: Thresholds are configured to detect overloaded and under-loaded hosts in a cloud computing platform. Beloglazov and Buyya elaborated a holistic way of setting up the upper- and lower-bound thresholds in CloudSim [12]. We consider the following three strategies for overloaded and under-loaded host detection.

- *Interquartile Range (IQR):* This strategy is a common statistical measure of variability range in dispersion. IQR is calculated using $IQR = Q_3 – Q_1$, where Q_3 is the third quartile of the 75th percentile and Q_1 is the first quartile or the first percentile. IQR is computed for each host, thereby providing a measure of over or under utilization of that host.
- *Median Absolute Deviation (MAD):* It is another statistical measure of dispersion calculated by finding the median of the absolute deviation of a data range. In contrast to standard deviation, MAD is less prune to the influence of outliers. Thus, MAD offers more accurate results than its counterparts.
- *Static Threshold:* This strategy simply maintain static thresholds; we set a static value to define when hosts are treated as overloaded ones.

2) Virtual Machine Selection: Nominating a VM as a candidate to migrate from under-loaded or overloaded hosts can be achieved by the virtue of effective algorithms implemented in CloudSim [12]. CloudSim offers the following versatile algorithms to elect to-be-migrated VMs to facilitate the purpose of power management.

- *Random Selection:* A VM is randomly selected from over- or under-utilized hosts.
- *Minimum Migration Time (MMT):* A chosen VM takes the minimum time to be migrated from the host.
- *Maximum Correlation:* Elected VMs exhibit more peak loads in terms of CPU usage.

3) Virtual Machine Placement: The goal of our algorithm design is to locate the most appropriate physical host to run VMs in order to deliver the optimal energy efficiency. We advocate for applying our *EGRET* - an extension of the BFD algorithm - to allocate and migrate VMs in accordance with the frequency usage of physical nodes.

VI. Experimental Results

We implement the EGRET system in using CloudSim driven by three randomly picked PlaneLab workloads, which are collected from three separate dates. We systematically evaluate nine different combinations of the overloading detection and VM selection algorithms described in V. Next, we undertake our simulations using these combinations against the CloudSim's VM allocation policy as a baseline. The overarching goal is to demonstrate that EGRET has an edge over CloudSim thanks to the DVFS performance optimizer combined with the frequency-aware VM placement strategy.

Table I unveils the overall performance of EGRET, FIFD, and BFD in terms of the following five metrics, namely, energy consumption, the number of migrations and SLA performance degradation.

The results plotted in Table I show that our proposed *EGRET* slashes the number of VM migrations of FIFD by to an average of 2000, which represents an average reduction rate of 15.7%. EGRET's edge over both FIFD and BDF with respect to VM migrations is attributed by the deployment of the DVFS-optimizer that curtails the overall frequency in migrating VMs while accepting new jobs. When it comes to the SLA performance, EGRET's SLA violation metric is on par with those of the FIFD and BFD strategies. The SLA performance degradation is contributed by the overloaded hosts due to VM migrations. Importantly, EGRET has an edge in terms of the power consumption over the two alternatives by an average 25kWh, which signifies that EGRET is adroit at adopting frequency scaling to suppress heavy VM migration overhead.

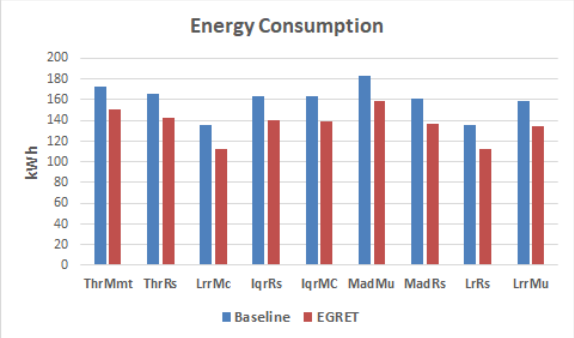
VII. Conclusions and Future Work

Recognizing that the overall resources consumption of modern data centers have dramatically increased, we have addressed in this paper a novel way of cutting back the energy consumption to build environmental-friendly clouds without adversely affecting service-level agreements (SLAs).

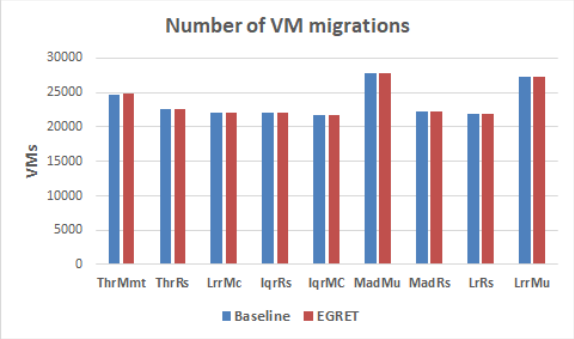
Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:12:50 UTC from IEEE Xplore. Restrictions apply.

TABLE I. Comparison of VM scheduling policies.

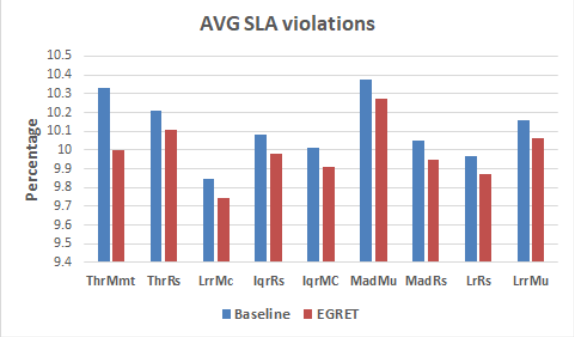
Policy	PlanetLab Workload	Energy Consumption kWh	Number of VM Migrations	Avg. SLA Violation
FIFD	20110303	188.86	28482	9.98%
BFD	20110303	188.86	26476	9.82%
EGRET	20110303	179.75	24008	9.86%



(a) Energy Consumption



(b) Number of VM Migrations



(c) Average SLA Violations

Fig. 1. Continued overall performance comparison using: Static Threshold Minimum Migration Time ThrMmt; Static Threshold Random Choice ThrRs; Local Regression Median absolute deviation LrrMC; Interquatile Range Random choice IqrRs; Interquartile Range Maximum correlation IqrMC; Median absolute deviation Minimum utilization MadMu; Median absoltue deviation Random choice MadRs; Local Regression Random choice LrRs; Local Regression Minimum utilization LrMu and our proposed schema *EGRET*.

We developed a resource-management module called *EGRET*, aiming at optimizing the energy efficiency of cloud-based data centers by seamlessly integrating the DVFS technique with the VM-placement and VM-consolidation strategies.

The experimental results unveil that our scheduler offers significant energy savings up to 41.62% while maintaining Qos. Our EGRET technique is adroit at cutting back unnecessary VM migrations to suppress the overhead incurred in power management.

There are two open issues to be addressed in our future research directions. First, we are slate to extend our proposed *EGRET* scheduler catering for a diversity of application domains such as high-performance computing, graphical computing, and big data processing. Second, we intend to explore new power models that embrace memory and disk resources.

ACKNOWLEDGMENT

This work is supported in part by the U.S. National Science Foundation under Grants IIS-1618669, OAC-1642133, and CCF-0845257 (CAREER).

REFERENCES

[1] M. Usman, X. He, K.-M. Lam, M. Xu, S. M. M. Bokhari, and J. Chen, "Frame interpolation for cloud-based mobile video streaming." *IEEE Trans. Multimedia*, vol. 18, no. 5, pp. 831–839, 2016.

[2] C.-T. Huynh, H.-Q. Nguyen, X.-Q. Pham, T.-D. Nguyen, and E.-N. Huh, "Cloud-based real-time location tracking and messaging system: a child-care case study," in *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*. ACM, 2015, p. 23.

[3] W. Ku and R. Zimmermann, "Nearest neighbor queries with peer-to-peer data sharing in mobile environments," *Pervasive Mob. Comput.*, vol. 4, no. 5, pp. 775–788, 2008.

[4] Y. Li, Z. Bao, G. Li, and K.-L. Tan, "Real time personalized search on social networks," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 639–650.

[5] S. Challita, F. Zalila, C. Gourdin, and P. Merle, "A precise model for google cloud platform," in *Cloud Engineering (IC2E), 2018 IEEE International Conference on*. IEEE, 2018, pp. 177–183.

[6] M. Filer, J. Gaudette, M. Ghobadi, R. Mahajan, T. Issenhuth, B. Klinkers, and J. Cox, "Elastic optical networking in the microsoft cloud," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, no. 7, pp. A45–A54, 2016.

[7] E. Masanet, "The energy efficiency potential of cloud-based software: A use case study," 2013.

[8] F. Farahnakian, P. Liljeberg, and J. Plosila, "Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning," in *Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on*. IEEE, 2014, pp. 500–507.

[9] Y. Ho, P. Liu, and J.-J. Wu, "Server consolidation algorithms with bounded migration cost and performance guarantees in cloud computing," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. IEEE, 2011, pp. 154–161.

[10] Z. Zong, A. Manzanares, X. Ruan, and X. Qin, "Ead and pebd: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters," *IEEE Transactions on Computers*, vol. 60, no. 3, pp. 360–374, 2011.

[11] R. E. Korf, "A new algorithm for optimal bin packing," in *Aaai/Iaai*, 2002, pp. 731–736.

[12] 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.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:12:50 UTC from IEEE Xplore. Restrictions apply.