CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Conference Paper

APEnergy: Application Profile-Based Energy-Efficient Framework for SaaS Clouds

Basit Qureshi
Anis Koubaa

CISTER-TR-190619

APEnergy: Application Profile-Based Energy-Efficient Framework for SaaS Clouds

Basit Qureshi, Anis Koubaa

CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail:

https://www.cister-labs.pt

Abstract

In the past decade, there has been a steady increase in the focus on green initiatives for data centers. Various energy efficiency measures have been proposed and adopted, however the optimal tradeoff between performance and energy efficiency of data centers is yet to be achieved. Addressing this issue, we present APEnergy, an Application Profile-based energy efficient framework for small to medium scale data centers. The proposed framework leverages information on the completed application with certain workloads in the data center to build profiles for workflows. The framework utilizes a novel scheduler to obtain a near-optimal mapping for placement of workflow tasks in the data center based on three criteria including CPU utilization, power cost and task completion time. We compare the performance of the proposed scheduler to similar RTC and HEFT schedulers. Extensive simulation studies are carried out to verify the scalability and efficiency of APEnergy framework. Results show that the proposed Scheduler is 2% and 14% more energy efficient than RTC and HEFT respectively.

APEnergy: Application Profile-based Energy-efficient Framework for SaaS Clouds

Basit Qureshi*, Anis Koubaa*†
* Department of Computer Science, Prince Sultan University, Saudi Arabia
†CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal
Email: {qureshi, akoubaa }@psu.edu.sa

***Abstract*—In the past decade, there has been a steady increase in the focus on green initiatives for data centers. Various energy efficiency measures have been proposed and adopted, however the optimal tradeoff between performance and energy efficiency of data centers is yet to be achieved. Addressing this issue, we present APEnergy, an Application Profile-based energy efficient framework for small to medium scale data centers. The proposed framework leverages information on the completed application with certain workloads in the data center to build profiles for workflows. The framework utilizes a novel scheduler to obtain a near-optimal mapping for placement of workflow tasks in the data center based on three criteria including CPU utilization, power cost and task completion time. We compare the performance of the proposed scheduler to similar RTC and HEFT schedulers. Extensive simulation studies are carried out to verify the scalability and efficiency of APEnergy framework. Results show that the proposed Scheduler is 2% and 14% more energy efficient than RTC and HEFT respectively.**

***Index Terms*—Cloud computing, Energy efficiency, workload optimization, Scheduling**

I. Introduction

In the age of Big Data, the Software as a Service (SaaS) cloud computing model, provides a heterogeneous multi-tenant virtual environment in the data centers [1]. The ever-growing demand for SaaS services is escalating the surge of data centers predictably intensifying the demand for energy utilization, thus raising the operational costs of data centers operations. A recent report [2] shows that 1.1% to 1.5% of the global energy usage accounts to the data centers. In one year, 416.2 Tera watt-hours of electricity was utilized by data centers worldwide. To this day, energy efficiency in executing services on data centers remains a challenge.

Since 2013, various energy efficiency measures have been considered to address the challenge. Some of these measures include the design of energy-efficient data centers [4]–[7], energy efficient placement of servers in data centers [8] [12]–[15], efficient deployment workflows in virtualized data centers [16]–[19], and the effective use of resources and virtualization technology in data centers [24] [25]. Most of these enhancements are being put to practice in various hyper-scale data centers maintained by the likes of Facebook, Google, Amazon etc. Small to medium scale data centers are generally possessed by small enterprises, universities etc. Only 5% of data center related power consumption worldwide accounts to the hyper-scale data centers, the remaining 95% relates to the small and medium scale data centers [4]. The energy management in small-medium scale data centers with the steady workload is more noteworthy than the dynamic nature of workload in hyper-scale data centers.

In small-medium scale data centers, server consolidation works towards efficient utilization of resources [8], [9]; however, energy-aware deployment of tasks with varying workloads in virtualized environments is challenging [22] and is an ongoing research area. Researchers in [21], [24] have devised the concept of Application Profiles (AP) where information on each individual application executing in the small-medium scale data center is stored and regularly updated. APs consist of information on the application type, workload, start time, execution time, finish time, resources utilized (CPU, memory, IO) etc. Information within an AP was used to improve the efficiency of clusters in data centers.

In this paper, we focus on small to medium scale data centers due to the low variability and high certainty in application workloads resulting in a near constant number of Virtual machines (VMs). We present APEnergy, a power aware energy efficient framework based on APs. In contrast to the scheduler algorithm in [24], we include energy utilization parameters in addition to resource usage (CPU, memory) etc within the APs. A system model for the workflows assignment based on task execution times and energy consumption parameters is developed. A novel power-aware task scheduling algorithm is presented with a focus on reducing the number of active physical servers and VM migrations in data centers whereas maintaining the overall workload performance. The performance efficiency of the proposed Scheduler is validated through extensive simulation studies. We compare the proposed APEnergy Scheduler with two scheduling algorithms namely Stochastic Heterogeneous Earliest Finish Time (HEFT) [10] and Robust Time Cost (RTC) [11]. Results show that the APEnergy scheduler is 2% and 14% more energy efficient than RTC and HEFT respectively.

The rest of the paper is organized as follows. Related works are presented in section 2. Section 3 details the APEnergy framework for Scheduling and placement of applications in the data center. Section 4 presents detailed experimental evaluations followed by conclusions in section 5.

II. Related works

Researchers in [4]–[7] focus on design of energy-efficient data centers. Dayarathna et.al in [5] provide a survey on data

978-1-5386-9355-1/18/$31.00 ©2018 IEEE
DOI 10.1109/FIT.2018.00053

center energy consumption models. Wan et.al in [6] present a framework that optimizes data center energy consumption by controlling the cooling units. Authors in [3], [4] present a novel approach to reduce energy consumption in a data center by optimizing the network traffic. This is achieved by strategically locating servers based on network traffic. Authors in [8] detail energy efficient placement strategies of Linux based software routers in software defined networks. They present an energy-aware multi-level control system and discuss design and implementation issues.

Varasteh in [12] present a survey on server consolidation techniques. Authors in [13] discuss a scheduling technique based on trading inspired approach for dynamic server consolidation in data centers. The proposed approach in [14] considers energy conservation mechanisms to migrate overloaded physical machines (PM) based on power utilization threshold strategy. Shaw et.al in [15] discuss the energy performance tradeoff for VM consolidation in cloud data centers by restricting the repeated migration of the same VMs. They present a heuristic based algorithm that utilizes a threshold to restrict migration in the data center.

Efficient deployment of workflows in virtualized data centers is the focus of [**?**], [16]–[19]. Hossain et. al in [16] present a belief rule based expert system to predict Power Usage Effectiveness (PUE) for uncertain workloads in data centers. They evaluate the system using real world data from a data centers. Poola et. al. in [17] presented a robust scheduling algorithm with resource allocation policies that schedules workflow tasks on heterogeneous Cloud resources while trying to minimize the total elapsed time and the cost. Xiangming Dai in [18] proposes algorithms to reduce energy consumption in data centers by considering the placement of VMs onto the servers intelligently. Hilman et. al. in [19] focus on cost of scheduled workflows in IaaS clouds. They propose a budget-distribution algorithm that assigns a part of the workflow budget to the individual tasks. This task-level budget guides the scheduling process to avoid incurring unexpected costs. Qureshi et.al. in [25] consider the role of power profiles in determining the power consumption in scheduling various applications in data centers by conducting experiments on Hadoop based clusters and measuring power consumption for different workloads. Chen et.al. [22] present an energy-efficient workload aware task Scheduler using online profiling that collects workload information of tasks for CPU-bound parallel applications. The aforementioned related works concur, the energy efficiency and the performance of the data center are correlated.

In this work, we present a framework for power-aware, energy-efficient placement of workloads in a data center by leveraging the concept of APs presented in [23], [24]. The proposed APEnergy framework considers various runtime parameters to build and update APs. A novel APEnergy Scheduler is developed that considers application-profile parameters for energy-efficient placement of VM in the data center hardware. The proposed framework achieves energy efficiency by scheduling near optimal mapping and placement of application workflows in the data center.

III. The AP-Energy Framework

In small to medium scale data centers, typically, a finite set of applications with definite workloads are executed. The APEnergy framework leverages the availability of an applications data, including expected run time, power requirements, the frequency of multi tenancy of VMs per PMs, resource allocation to VMs etc. to build APs. Before the initiation and deployment of an application on the cluster, the information available in the AP is used by the APEnergy framework to optimally schedule the task placement by considering the expected run times and power consumption parameters. An AP consists of data including the energy consumption for an application, its resource requirements including CPU, memory, Network I/O, the power consumption of underlying PMs, expected completion times and frequency of applications initiated per unit time. The profiles are periodically updated to maintain the ever-changing energy efficiency requirements in the clusters. We define VM profile and AP as data structures within the framework to support power aware workflow scheduling in data centers. A VM Profile consists of a PM id; a VM id; CPU resource requested by the VM, memory requested, and power attributes. In this work, we assume that power is directly correlated to the CPU utilization. Power attributes of a PM are used for optimization purposes. Finally, *task_id* in a workflow, contains the task assigned to a VM. An AP consists of the following parameters:

- Workflow ID (W_k). At the initiation, the application manager would initiate a workflow that contains various applications/tasks to be executed on the cluster. A workflow identifier is a unique identifier.
- Task ID (T_{ki}). A workflow consists of multiple applications with certain number of tasks to be executed. The task id is a unique identifier.
- Arrival Time (a_{ki}). The arrival time is the instant in seconds when the task arrived. This time is utilized in a tasks assignment to a VM.
- Completion time (e_{ki}). Based on the cost of executing a task, an expected completion time is computed.
- Requested resources. This is a combination of two parameters, requested CPU and memory workload in bytes. The requested resources are used in the assignment of VMs to tasks.
- Power profile. This contains information about the power consumption at various time intervals. These values are used in conjunction with the power attributes defined in VM profile for placement of tasks in the data center.

Fig. 1 shows the application and VM profiles data structures. As a new workflow arrives (algorithm 1) with multiple tasks, individual tasks are registered by the Scheduler in the framework. If an AP exists for this task, the status information is pulled; based on workflow parameters, resources requested including CPU, memory etc. are determined. Information related to task initialization such as task arrival time, workload, expected completion time and power consumption is analyzed. Once the cost analysis is completed, the task is assigned to a

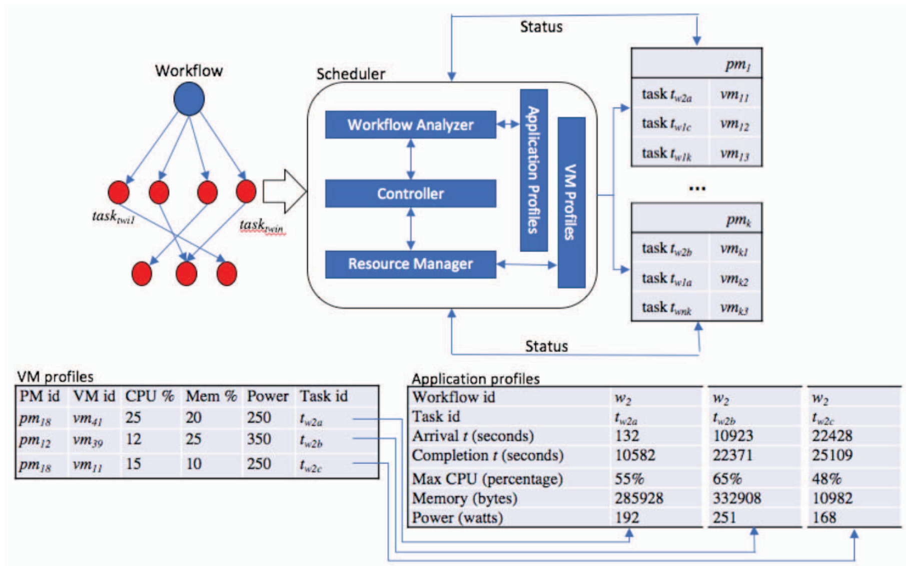

Fig. 1. APEnergy Scheduling model with VM and APs

VM, and the VM profile is updated. As the task executes, runtime status parameters such as completion rate, power consumption, starting time, completion time etc. are recorded. In case, when an AP for a task does not exist, the scheduler estimates the runtime based on the workflow size using general distribution. Over a period of time, with larger data availability, the cost analysis normalizes the overall efficiency of the framework.

A. System Model

As the workflow arrives, the scheduler controller communicates with the workflow analyzer and resource manager to determine number of tasks, types of tasks, resources required by each task by querying information in application and VM profiles. The scheduler using a power-aware scheduling algorithm, determines the optimal cost of efficiently executing the workflow with minimal power requirements by producing a mapping. Once the Price / Cost of executing the workflow is determined, the tasks are assigned to VMs. We assume that the workflows $W = \{w_0, w_1, \ldots, w_k\}$ are continually submitted. Each workflow w_k consists of a list of tasks $T_{ki} = \{T_0, T_1, \ldots, T_i\}$; $T_{ki} \in w_k$; with arrival time a_{ki}; expected completion time e_{ki}; and a power-profile P_{ki}. The expected arrival time a_{ki} is the system time at initiation of a task in the workflow k_i, the power-profile is determined from a vector containing the power consumption levels $P_{T_{ki}} = \{p_0, p_1, \ldots, p_j\}$ of a task *i* in the workflow *k*; where $p_0, p_1, \ldots, p_j$ are the power consumption levels for a task at various times during its execution. Tasks defined in a workflow are assigned to a set of VMs *V* such that only one task is executed per VM. A pool of VMs $V = \{v_0, v_1, \ldots, v_x\}$ is available, VMs can be gained and released by any task T_{ki} in a workflow. Each VM has a $Price(v_x)$ associated with it which indicates the cost to execute a task on this VM for a time frame $\Delta t = t_{eki} - t_{aki}$ where t_{aki} is the arrival time for task T_{ki} in Workflow w_k and t_{eki} is the expected completion time for this task. The variable $m_{T_{ki,x}}$ reflects mapping between tasks and VMs. The value of $m_{T_{ki,x}}$ is 1 if task T_{ki} is mapped to VM and 0 otherwise.

$$m_{Tki,x} = \begin{cases} 1 & when\ T_{ki}\ is\ assigned\ to\ v_x \\ 0 & otherwise \end{cases} \tag{1}$$

The real starting times $S_{T_{ki,x}}$ and completion times $f_{T_{ki,x}}$ on v_x become available as the task progresses. We give the completion time as:

$$f_{T_{ki,x}} \geq S_{T_{ki,x}} + t_{m_{ki,x}} \tag{2}$$

where $t_{m_{ki,x}}$ gives the time for placement of task in a VM. Equation 2 gives the actual completion time of all tasks in a workflow.

$$max_{z \in T}\{f_{T_z}\} \leq f_k \tag{3}$$

where $max_{z \in T}$ is the actual completion time f_k of the workflow w_k.

If we assume $D = \{d_0, d_1, \ldots, d_y\}$ PMs are available in the cluster, then $n_y^{m_{T_{ki,x}}}$ gives the mapping of tasks T_{ki} in a VM v_x to PM d_y. Here, the optimization objective is to *i)* minimize the cost of task execution in the workflow and *ii)* maximize the utilization of PM by optimizing the multi-tenancy of VM placement on servers. This can be determined as:

$$minimize \sum_{t=1}^{x} Price(v_t).\Delta t_{ki} \tag{4}$$

where v_t is the total number of VMs deployed, w_k, $Price(v_t)$ represents the price associated with this VM and Δt_{ki} is the time to completion. The optimal placement of VMs per PM is determined by the minimization of power consumption cost of a VM. We give the average for executing a VM with mapping $m_{Tki,x}$ on server d_y as:

$$Price_y^{m_{Tki,x}} = 1/j \sum_{t=0}^{j} p_{T_{ki,x}}.\Delta t_{ki} \tag{5}$$

where $p_{T_{ki,x}}$ is the power consumption level at the time interval Δt_{ki} and $0 < Price_y^{m_{Tki,x}} < 1$. To obtain an optimal $n_y^{m_{T_{ki,x}}}$ we need to maximize the placement of *r* number of VMs on a PM y such that:

$$n_y^{m_{T_{ki,x}}} = max_{y \in D} \sum_{r=1}^{x} Price_y^{m_{Tki,x}} \tag{6}$$

The maximum CPU utilization θ_y threshold on a PM d_y where $\theta_y \in [0, 1]$.

B. APEnergy Scheduler

Two criterion are considered for ranking the tasks, i) earliest finishing time of a task, ii) Lowest *Price* of tasks available in the scheduling queue. The proposed scheduling algorithm continually creates and updates new and existing mappings of tasks, VMs and PMs based on the profiles. Algorithm 1 shows the process for a workflow arrival in the framework. It analyzes every task considering its expected completion time and Price which is available in the AP. The *TaskList* queue is frequently updated considering the tasks processed in a workflow. Once a *TaskList* queue is completed, the scheduler is called given in algorithm2. All the remaining tasks that were rejected, are appended to the *Taskpool* queue for the next cycle of workflow processing. An example of a rejected task is a task that may not have been completed after Δt. The scheduler initiates the VMs, tasks and PMs mapping m_k for a workflow w_k. The tasks time and prices is periodically updated in the *TaskList*. An appropriate VM is selected with minimal expected runtime and lowest Price and is added to the VM map m_k. Once a mapping is produced, the tasks are removed from the *TaskList*. The scheduler also computes the cost of executing the VM in PMs, considering the frequency of multi tenancy on a PM and the load factor, the scheduler determines the minimal price of executing VMs on a PM using equations (5) and (6) where as considering the limit for θ_y. PMs satisfying the criterion $isvalid(n_y^{m_{T_{ki,x}}})$, receive the VMs and the task is initiated. As an alternative, the next PM satisfying the criterion is sought from the pool of available PMs. Consequently, the load factor and the price of the selected PM is updated in the profiles defined in the framework.

IV. Performance Evaluation

In this section, we detail the experimental evaluation of APEnergy based on simulation studies. We compare the proposed scheduling algorithm with two scheduling algorithms; Stochastic Heterogeneous Earliest Finish Time (HEFT) [11] and Robust Time Cost (RTC) [12]. CloudSim framework [9] is extended to simulate a cloud computing cluster environment. We assume that the number of VMs for each type is

Algorithm 1: On arrival of a new workflow

Taskpool ← *null*
Tasklist ← *null*
for each task $t_i \in w_k$ **do**
  if $t_i \in APL$ **then**
    Calculate the $f_{T_{ki,x}}$ from AP as in equation (2)
  else Generate expected completion time $t_{e_{ki}}$
  end if
  Calculate the power consumption of t_i
  Determine the Price for t_i as in equation (4)
  TaskList ← *TaskList* ∪ $\{t_{ki}, Price\}$
end for
Call Scheduler() to schedule tasks in *TaskList*
Add all non-scheduled tasks in *Taskpool*

infinite, the VM instances can be acquired at any time. Two workflow templates are created for evaluation based on real-world scientific application workflows obtained from Pegasus Workflow repository [20]. We consider small and medium data-sets with about 30 and 60 tasks per workflow respectively. For experimental evaluation, we assume that the VM boot time is 2 seconds. Two different test setups 1 and 2 are used. Test setup 1 is used to verify the feasibility and Task completion efficiency for the three scheduling algorithms. Test setup 2 is used to compare the energy efficiency of the APEnergy against HEFT and RTC. Various simulation parameters can be seen in table 1.

TABLE I
Test setup parameters

Test Setup 1. (100 PMs)					
Scenarios	1	2	3	4	
VMs	200	400	500	800	
Workflows	200	400	800	2000	
Test Setup 2. (150 PMs)					
Scenarios	5	6	7	8	9
VMs	200	300	400	800	1000
Workflows	400	900	1600	2000	3000

A. CPU utilization and task placement efficiency

In order to determine the efficient utilization of resources, we consider the task placement efficiency of the proposed scheduler. This can be determined by analyzing the ratio of CPU resources requested by a VM versus the actual utilization of the CPU at the PM. The CPU utilization efficiency $\eta_{CPU(j)}$ is given as:

$$\eta_{CPU(j)} = \frac{\sum_{i=0}^{x} v_{i,CPU}}{\mu_{d_j}}; v \in V, d_j \in D, \eta_{CPU(j)} < 1 \tag{7}$$

where μ_{d_j} is the CPU utilization at the server, and $\sum_{i=0}^{x} v_{i,CPU}$ is the sum of the requested CPU resources of all VMs placed on PM d_j. The value of $\eta_{CPU(j)}$ is computed at task placement time before the actual execution of the workload in the cluster. We compare the task placement efficiency in terms of CPU utilization efficiency of the proposed scheduler against the HEFT and RTC. Table 2 shows the average CPU utilization for Scenarios 1 to 4 for Test Setup

Algorithm 2: Tasks scheduler

```
Initialize VM map m_k
for each {t_i, Price} ∈ TaskList do
    Target_vm ← null, minCost ←∝ □
    for each v_j ∈ V□ do
        Compute Price(v_t) for t_i
        if Price(v_t)□ < Price and f_{t_ij} < minCost then
            Target_vm ← v_j, minCost ← f_{t_ij}
        end if
    end for
    if Target_vm ≠ null then
        Assign t_i to v_j
    else
        Add a v_j satisfying equation (4) and (5)
        Allocate task t_i to v_j
    end if
    m_k ← m_k + v_j
    V□ ← {V□ − v_j}
end for
for each d_y in Cluster do
    for each v_x ∈ m_k do
        if isvalid (n_y^{m_{T_{k,x}}}) and load(d_y) < maxload then
            Assign v_x to d_y
            m_k ← m_k − v_x
            update n_y^{m_{T_{k,x}}} in equation (6)
            load(d_y) ← load(d_y) + 1
        end if
    end for
end for
```

1. We observe a high variance in the average CPU utilization efficiency of the proposed scheduler when compared to HEFT and RTC in Scenario 1. However, as the ratio of the number of applications and VMs increases per PM, the efficiency also increases. In Scenario 4, we observe an average CPU efficiency of 0.719 compared to 0.778 and 0.701 for HEFT and RTC, respectively. The proposed Scheduler in this work achieves results that are close to the utilization efficiency to the RTC and HEFT Schedulers with the increase in the problem size. Due to a low number of VMs per PM, the APEnergy scheduler increases the ratio of the number of VMs per PMs. This reflects in increased number of PMs with near zero load. On the other hand, the HEFT and RTC are comparatively not very efficient with only 1 and 3 VMs per PM. Consequently, the average CPU utilization efficiency for scenario 1 is lower for the APEnergy scheduler in comparison with HEFT and RTC. As the workload increases, in scenario 3 and 4, the APEnergy scheduler maintains 8 and 3 PMs with zero VM placements, however, HEFT and RTC manage 0. This observation provides evidence that, with the smaller workload, the APEnergy scheduler tends to be more efficient in terms of workload placement in the cluster.

B. Energy Efficiency

We analyze the energy efficiency by providing variations in three parameters,*i)* increasing the number of physical and

TABLE II
COMPARISON OF SCHEDULERS

Average CPU utilization efficiency η_{CPU}				
Scenario	1	2	3	4
Proposed Scheduler	0.397	0.495	0.593	0.719
HEFT	0.528	0.583	0.628	0.778
RTC	0.418	0.492	0.588	0.701
Number of Idle PMs $\|\varphi\|$				
Scenario	1	2	3	4
Proposed Scheduler	19	15	8	3
HEFT	1	1	0	0
RTC	3	2	0	0
VM Placement time (seconds) in the cluster				
Scenario	1	2	3	4
Proposed Scheduler	5	12	17	24
HEFT	3	18	31	37
RTC	4	22	42	83

VMs, *ii)* increasing the applications and consequently increasing the total workload, and *iii)* tweaking the *maxload* and threshold θ_y. The objective of this experimentation is to analyze the effect of larger workloads, increased number of resources, and the threshold of the energy efficiency in the proposed scheduler. We assume that the cluster is composed of PMs similar in characteristics in terms of processor architecture, frequency, physical memory size, etc. We assume the power usage at idle time for a server to be 150 W. In Test Setup 2, the number of PMs is increased to 150, with the CPU availability threshold increased to $0.35 < \theta_y < 0.85$ and the *maxload* set to 20. We observe the average CPU utilization efficiency and number of idle servers at the initiation of workflows for all three schedulers, as seen in Table 3. We notice the average CPU utilization efficiency for HEFT is better compared to the RTC and the proposed scheduler. This clearly shows that, on average, HEFT performs better in terms of average CPU utilization; however, we also note that the number of idle PMs for the proposed scheduler is better compared to HEFT. The APEnergy scheduler on average consumes less power compared to the RTC and the HEFT for Scenarios 5, 6, and 7 as can be seen in table 3; however, its power consumption is comparable for Scenarios 8 and 9 with larger workloads. We interpret from this experimentation that the APEnergy scheduler increases the multi-tenancy per physical server compared to the RTC, consequently increasing the number of idle machines. In Scenarios 5 and 6, where the workload is comparatively lower, the APEnergy scheduler with 802.1 and 825.6 kWh is 2% and 14% more efficient than the RTC and HEFT, respectively. We concur that the effect of increasing the number of idle machines by efficient placement of VMs in the cluster improves the power efficiency of the cluster.

TABLE III
COMPARISON OF THE ENERGY CONSUMPTION

Scenario	Proposed Scheduler			RTC			HEFT		
	η_{CPU}	P_{avg}	kWh	η_{CPU}	P_{avg}	kWh	η_{CPU}	P_{avg}	kWh
5	0.492	248.4	802.1	0.404	230.8	815.4	0.531	256.2	917.2
6	0.513	252.6	825.6	0.436	237.2	843.5	0.581	266.2	955.5
7	0.608	271.6	928.1	0.556	261.2	940.3	0.619	273.8	982.7
8	0.682	286.4	1001.6	0.689	287.8	1036.1	0.709	291.8	1050.5
9	0.742	298.4	1067.1	0.734	296.8	1068.5	0.783	306.6	1103.8

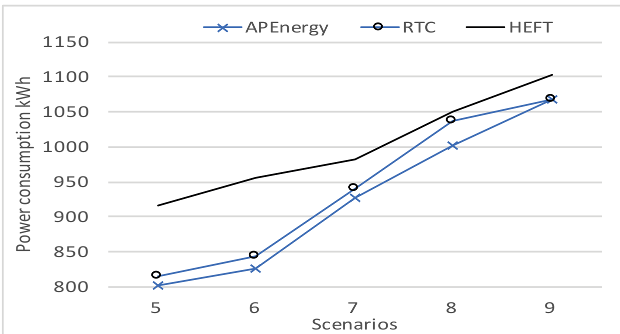

Fig. 2. Comparison of power consumption among the schedulers

V. CONCLUSIONS

In this paper, we present APEnergy, a profile based energy efficient framework with a novel Scheduler that makes a good trade-off considering various parameters consisting of cost of VM placement, power usage, CPU utilization and PMs load factor. The performance of the APEnergy Scheduler is compared to RTC and HEFT Schedulers extensively through simulation studies. Results show that the APEnergy scheduler dominates the benchmarked Schedulers in energy efficiency and exploits the data center resources by increasing the multi tenancy of VMs per PM, therefore, increasing the number of idle machines in the cluster. For smaller workloads, the APEnergy Scheduler is comparatively 2% and 14% more energy efficient than RTC and HEFT, however, for larger workloads, the energy efficiency is only slightly better.

ACKNOWLEDGMENT

This work is partially supported by Robotics and Internet of Things Lab at Prince Sultan University.

REFERENCES

[1] A. Shehabi et al., "United states data center energy usage report", 2016, [online] Available: https://publications.lbl.gov/islandora/object/ir%3A1005775/datastream/PDF/view. Last accessed July 2018.
[2] I. M. Llorente, "The Limits to Cloud Price Reduction", IEEE Cloud Computing, vol. 4, no. 3, pp. 8-13, 2017.
[3] F. AlMudarra, B. Qureshi, "Issues in adopting Agile Development Principles for Mobile Cloud Computing Applications", 6th Intl Conf. on Ambient Systems, Networks and Tech., UK June 2-5, 2015.
[4] Z. Li and Y. Yang, "A Novel Network Structure with Power Efficiency and High Availability for Data Centers",IEEE Trans on Para. and Dist. Sys. vol. 29, no. 2, pp. 254-268, Feb. 2018.
[5] M. Dayarathna, Y. Wen and R. Fan, "Data Center Energy Consumption Modeling: A Survey", IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 732-794, 2016.
[6] J. Wan, X. Gui, R. Zhang and L. Fu, "Joint Cooling and Server Control in Data Centers: A Cross-Layer Framework for Holistic Energy Minimization", IEEE Systems Journal, vol. 12, No. 3, 2018.
[7] B. Qureshi et.al., "Countering the collusion attack with a multidimensional decentralized trust and reputation model in disconnected MANETs", Multimedia tools and apps., vol 66. no.2, pp.303-323, 2013.
[8] M. P. Karpowicz et.al., "Energy-Aware Multilevel Control System for a Network of Linux Software Routers: Design and Implementation," inIEEE Systems Journal, vol. 12, no. 1, pp. 571-582, March 2018.
[9] Rodrigo N. Calheiros, et.al.,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, January 2011.
[10] Xiaoyong Tang, et.al., A stochastic scheduling algorithm for precedence constrained tasks on Grid, Future Generation Computer Systems, vol 27, no. 8, pp. 1083-1091, 2011.
[11] D. Poola, et.al., "Robust Scheduling of Scientific Workflows with Deadline and Budget Constraints in Clouds", 28th IEEE Intl Conf on Adv. Info. Networking and Applications (AINA), pp. 858-865, 2014.
[12] A. Varasteh and M. Goudarzi, "Server Consolidation Techniques in Virtualized Data Centers: A Survey," inIEEE Systems Journal, vol. 11, no. 2, pp. 772-783, June 2017.
[13] G. Wu and M. Tang, "A Trading-Inspired Approach to the Dynamic Server Consolidation Problem in Data Centers", IEEE Intl conf. on Computer and Information Technology (CIT), Nadi, pp. 776-782, 2016.
[14] X. Wu, Y. Zeng and G. Lin, "An Energy Efficient VM Migration Algorithm in Data Centers,"in 16th Intl Symp. on Dist. Comp. and App. to Business, Engineering and Science (DCABES), Anyang, 2017.
[15] S. B. Shaw, J. P. Kumar and A. K. Singh, "Energy-performance trade-off through restricted VM consolidation in cloud data center",Int'l Conf. on Intelligent Computing and Control (I2C2), Coimbatore, pp. 1-6, 2017.
[16] M. S. Hossain et.al., "A Belief Rule Based Expert System for Datacenter PUE Prediction under Uncertainty," inIEEE Trans. on Sustainable Computing, vol. 2, no. 2, pp. 140-153, April 2017.
[17] D. Poola, S. K. Garg, R. Buyya, Y. Yang and K. Ramamohanarao, "Robust Scheduling of Scientific Workflows with Deadline and Budget Constraints in Clouds",28th IEEE Intl. Conf. on Advanced Information Networking and Applications (AINA), Canada, pp. 858-865, 2014.
[18] Xiangming Dai et.al., Energy-Efficient VMs Scheduling in Multi-Tenant Data Centers IEEE Trans. on Cloud Computing, pp. 210-221, 2016.
[19] M. Hilman, et.al., "Task-Based Budget Distribution Strategies for Scientific Workflows with Coarse-Grained Billing Periods in IaaS Clouds", 13th IEEE Intl. Conf. on e-Science, pp. 128-135, 2017.
[20] Gideon Juve, et.al., Characterizing and profiling scientific workflows, Future Generation Computer Systems, vol. 29, no. 3, pp. 682-692, 2013.
[21] F. Alharbi, Y. C. Tain, M. Tang and T. K. Sarker, "Profile-Based Static VM Placement for Energy-Efficient Data center,"18th IEEE Conf. on High Performance Computing and Comm. (HPCC2016), 2016.
[22] Q. Chen, L. Zheng, M. Guo, Z. Huang, "Eewa: Energy-efficient workload-aware task scheduling in multi-core architectures", IEEE Intl Parall. Dist. Proc. Symp., IPDPSW, pp. 642651, 2014.
[23] K. Ye, et.al., "Profiling-based workload consolidation and migration in virtualized data centers", IEEE Trans. Parallel Distributed Systems, vol. 26, no. 3, pp. 878890, 2015.
[24] Meera Vasudevan, et.al., "Profile-based application assignment for greener and more energy-efficient data centers", Future Generation Computer Systems, Volume 67, Pages 94-108, February 2017.
[25] B Qureshi, S AlWehaibi, A Koubaa, "On Power Consumption Profiles for Data Intensive Workloads in Virtualized Hadoop Clusters",in IEEE INFOCOM 2017, Atlanta, USA. pp.653-659, May 2017.