CoLocateMe: Aggregation-Based, Energy, Performance and Cost Aware VM Placement and Consolidation in Heterogeneous IaaS Clouds

Muhammad Zakarya, Lee Gillam, Khaled Salah, Omer Rana, Santosh Tirunagari, and Rajkumar Buyya

Abstract—In many production clouds, with the notable exception of Google, aggregation-based VM placement policies are used to provision datacenter resources energy and performance efficiently. However, if VMs with similar workloads are placed onto the same machines, they might suffer from contention, particularly, if they are competing for similar resources. High levels of resource contention may degrade VMs performance, and, therefore, could potentially increase users' costs and infrastructure's energy consumption. Furthermore, segregation-based methods result in stranded resources and, therefore, less economics. The recent industrial interest in segregating workloads opens new directions for research. In this article, we demonstrate how aggregation and segregation-based VM placement policies lead to variabilities in energy efficiency, workload performance, and users' costs. We, then, propose various approaches to aggregation-based placement and migration. We investigate through a number of experiments, using Microsoft Azure and Google's workload traces for more than twelve thousand hosts and a million VMs, the impact of placement decisions on energy, performance, and costs. Our extensive simulations and empirical evaluation demonstrate that, for certain workloads, aggregation-based allocation and consolidation is ∼9.61% more energy and ∼20.0% more performance efficient than segregation-based policies. Moreover, various aggregation metrics, such as runtimes and workload types, offer variations in energy consumption and performance, therefore, users' costs.

Index Terms—Clouds, datacenters, VM placement, resource consolidation, migrations, heterogeneity, energy efficiency, performance

1 Introduction

One of the major challenges in cloud datacenters is to manage computational resources energy and performance efficiently. Energy consumption affects our environment and account for large energy bills while performance affects cloud economics. Therefore, cloud service providers are focusing to design policies for energy, performance aware computing, encouraged by high operational costs of installed computer clusters [1]. The goal can be achieved in two different ways: (i) assigning only appropriate resources; and (ii) consolidating workload onto fewer machines using VM migration and switching off idle machines. On one side, the capability of VM migrations brings several benefits such as improved manageability, increased utilisation and energy savings. However, on the other side, it results in down time that decreases the performance of workloads. Migrations are expensive and in dynamic cloud environments, where thousands number of VM requests arrive in an hour, even they might not be suitable. Therefore, appropriate VM placement policies are essential to save energy and provide customers the expected level of workload performance [2].

VM placement policies can be categorized as: (a) segregation; and (b) aggregation based [1]. In segregation based policies, the providers run user-facing, batch, and production workloads in separate clusters (hosts). Therefore, if workloads demand is either low or even not available, then their resources still need to be switched on and this may also result in stranded resources. Large number of hosts in use can increase the providers' energy bill and have impact on our environment. Aggregation based policies run mixed workloads on same hosts which may degrade the workload performance, particularly, if they compete for similar resources (co-located VMs) [3]. Moreover, workload performance also varies across different CPU architectures – similar workloads may run quite differently over same CPU model [4]. Subsequently, lower workload performance could potentially increase infrastructure energy consumption and users monetary costs. The former approach is widely used in many production clouds, such as Alibaba cluster [5], with the notable

- *Muhammad Zakarya is with the Department of Computer Science, Abdul Wali Khan University, Mardan 23200, Pakistan. E-mail: mohd.zakarya@awkum.edu.pk.*
- *Lee Gillam is with the University of Surrey, GU2 7XH Guildford, U.K. E-mail: L.Gillam@surrey.ac.uk.*
- *Khaled Salah is with Khalifa University, Abu Dhabi 127788, UAE. E-mail: khaled.salah@ku.ac.ae.*
- *Omer Rana is with the University of Cardiff, CF10 3AT Cardiff, U.K. E-mail: ranaof@cardiff.ac.uk.*
- *Santosh Tirunagari is with Middlesex University, NW4 4BT London, U.K. E-mail: s.tirunagari@mdx.ac.uk.*
- *Rajkumar Buyya is with Cloud Computing and Distributed Systems Lab, School of Computing and Information Systems, University of Melbourne, Parkville, VIC 3010, Australia. E-mail: rbuyya@unimelb.edu.au.*

Manuscript received 5 March 2022; accepted 6 June 2022. Date of publication 10 June 2022; date of current version 10 April 2023.
This work was supported in part by AWK University, Pakistan, and in part by an Australian Research Council Discovery project at the University of Melbourne, Australia.
(Corresponding author: Muhammad Zakarya.)
Digital Object Identifier no. 10.1109/TSC.2022.3181375

exception of the Google's cluster [6]. Perhaps, inspired from benefits of aggregation-based approaches, Alibaba's cluster resources are also now offered to run workloads in mix. However, a detailed investigation of both methodologies is still needed in terms of energy efficiency and workload performance. In fact, the providers' switching efforts motivate us to investigate which technique is better than the other in term of energy, performance, and cost efficiency. Furthermore, which characteristics of the workloads should be used to aggregate them onto similar servers.

In this paper, we investigate how VMs and workloads would be placed onto physical hosts, in a heterogeneous cluster, so that the infrastructure energy consumption is minimised under the performance and users' cost constraints. We propose runtime-aware aggregation-based, energy, performance, cost (EPC) efficient VM placement and consolidation policies in order to execute several workloads in mix. Since, workloads are co-located, therefore, we call it CoLocateMe. Using real workload datasets from virtualised clouds, such as Google and Microsoft Azure clouds, we evaluate the performance of runtime-aware aggregation and segregation based placement policies, in an event driven cloud simulator i.e., CloudSim [7]. Our empirical evaluation suggests that the proposed, runtime-aware aggregation-based, VM placement and consolidation policies outperform segregation-based policies. Since, aggregation favours to colocate VMs having similar characteristics onto similar hosts, therefore, we call it "CoLocateMe". Major contributions of this paper are:

- an aggregation-based, energy, performance and cost (EPC) aware VM placement policy is proposed;
- a consolidation method is suggested that put similar workloads onto same resources;
- with respect to workload performance, we model resource heterogeneities in datacenters; and
- we evaluate the impact of aggregation and segregation-based VM placement and migration policies on infrastructure energy efficiency, workload performance and users' costs.

The rest of the paper is organized as follows. In Section 2, we discuss the VM placement problem. In Section 3, we propose an aggregation-based allocation and consolidation technique that places similar workloads on same resources. We validate the proposed scheme using real workload traces from Azure clusters in Section 4. We offer an overview of the related work in Section 5. Finally, Section 6 concludes the paper and describes future research.

2 Problem Description

The runtime period or execution time of a VM (R_{vm}) is dependent on data size to be processed and the quantity of resources i.e., CPU cores, memory, and bandwidth, assigned. The active period of a physical machine is proportional to the lengthiest runtime of the VMs running on the machine. If the duration of most VMs is much shorter than the runtime of the longest one, it indicates low machine runtime efficiency. To increase the runtime efficiency of machines, researchers have proposed techniques like aggregating VMs with similar runtime to a particular cluster or consolidate VMs by their capacities [8]. The former method can save more power than the later one through decreasing machine runtime. However, the impact of the runtime diversity of VMs and VM resource capacities on amount of machines should be considered when designing energy and performance efficient resource management policies. Furthermore, performance loss due to resource contention must be taken into account.

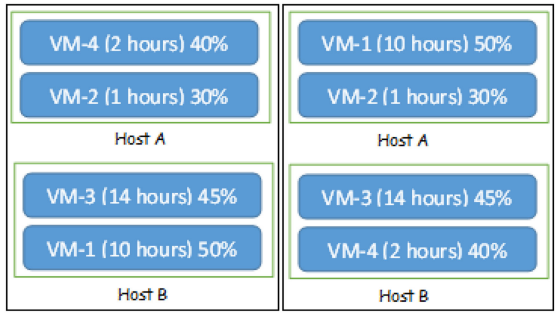

Fig. 1. Problem description [aggregation w.r.t runtimes].

Considering only two hosts, as part of a datacenter, as shown in Fig. 1; each one can accommodate two VMs. There are four VMs with different runtime requirements. If the placement is not runtime aware and consolidation is not assumed; then, host A will at least run for 10 hours and host B for 14 hours (right-hand side i.e., VM-1 and VM-2 are co-located on host A while VM-3 and VM-4 on host B), continuously. However, if runtimes are considered (left-hand side – VM-2 and VM-4 are co-located on host A while VM-1 and VM-3 on host B) then after 2 hours, host A can be switched off to save power; if there are no pending VM requests in the admission queue. Note that, the runtime of each VM depends on the application's workload which is highly unpredictable. If we can predict it, this does not essentially means that hosts in Fig. 1 (right) will run for 10 and 14 hours. When VM-2 and VM-4 finish their jobs, we can consolidate VM-1 and VM-3 to one host. Similar decisions can also be taken on VM capacities and instances of similar types can be placed together. However, various workloads if co-located or aggregated may not perform up to expected levels. However, if workloads compete for similar resources, then performance of the workloads may potentially be affected [3]. Furthermore, similar workloads may perform quite differently across same CPU models due to: (i) CPU models [4]; and (ii) resource contention [3]. These variations may potentially affect workload runtimes, therefore, users' service costs; and energy consumption. Subsequently, energy consumption affects revenue of service providers (energy bills) and our environment (green house gases). Therefore, it is essential to account for these costs when deciding resource placement and consolidation.

The above problem can be formulated as a multi-objective optimisation problem where objective are: (a) minimise total energy consumption ($E = \sum_{i=1}^{N} E_{host_i}$); improve or, at least, maintain workload performance (P); and minimise users' monetary costs (C i.e., VMs provisioning costs). Note that, P and C are inversely proportional where improving workload' P means reducing its R (the sum of all VM runtimes R_{vm} running the same workload) which subsequently means reducing C (the sum of all VMs provisioning costs).

For many request-response cloud workloads such as databases, with the notable exception of batch workloads, this statement might not be essentially true due to users' experiences. Furthermore, improving P through provisioning more resources will itself increase certain costs (e.g., C, E). For any request-response workloads such as web services, streaming analytics, getting a response faster does not reduce the energy cost, and would most likely increase it. Therefore, our aim is to improve P through running workloads on appropriate VMs and hosts (based on historical information). Moreover, various objectives (E, P, C) can be combined into a single metric $(ERC = \frac{E \times C}{P})$ where P is the inverse of runtime (R), and then solved as a single objective problem [1], given by

$$min(ERC). \tag{1}$$

As, energy is measured in Wh (watt hours), runtime in hours and users' monetary cost in $/hour. Thus, the above metric captures power to runtime ratio per unit cost [4]. The least values for ERC will translate to the best achievable performance & energy efficiency at the lowest cost. To deal with the above multi-objective optimisation problem, when one objective is preferred over another; then, the ERC can be elaborated as $ERC = \alpha.E \times \beta.R \times \gamma.C$ subject to $\alpha + \beta + \gamma \in \{0, \ldots, 1\}$ (where α, β, and γ are the domination values for energy, performance, and user's costs, respectively) [9], [10].

3 Proposed Solution

Aggregation, based on runtime of VMs might be useful if workload runtimes are predictable. Albeit, various machine learning based techniques, such as gradient boosted trees [11], have been suggested to predict VMs runtimes. However, due to the unpredictable nature of the cloud workloads, many efforts are needed. In Section 3.1, we explain the methodologies and approaches to implement resource management policies. In Section 3.2, we explain the aggregation-based placement and consolidation policies.

3.1 Implementation Methodologies

Hence, due to the efforts involved in accurately predicting the runtimes of VMs; we use two different methodologies to implement the above algorithms: (i) use previous runtimes of VMs to aggregate them; and (ii) predict VMs runtimes using the gradient boost tree method [11]. However, there would be other efficient ways of doing the same, for example using workload types, VM sizes, as described later in Section 4.3.9.

3.1.1 Past Runtimes

From IaaS point of view, workloads and their runtimes cannot be predicted accurately. Therefore, instead of using workload actual runtimes, an alternative approach is to use their past runtimes (the duration for which the workload has already run). In other words, we assume that workloads which have run (in past) for similar runtimes may probably run (in future) for similar durations – this is like a probabilistic approach. The idea is based on our previous works [1], [12]; that use VMs or containers past runtimes in workload placement decisions – *migrate only relatively long-running VMs and/or containers since they could recover their migration costs*. We do not use the model here, instead, we use the methodology of migrating relatively log running VMs. Furthermore, the initial placement (i.e., past runtime is zero) is achieved through the classic first fit (FF) heuristic algorithm. Using past runtimes for such decisions avoid complex prediction techniques (e.g., machine learning) that might not be reasonable in hyper-scale IaaS clouds.

3.1.2 Runtimes Prediction

If we assume that clouds are not opaque; then it is possible to predict their runtimes using historical data [13], [14]. Using past runtimes may not provide accurate estimates – for example, workloads which have run for longer durations have more probability and, therefore, higher tendency toward terminations. Therefore, it is essential to predict runtimes and use them in resource allocation and migration decisions. Note that, predicting a particular workload runtime may need identifying its type first i.e., CPU, memory, disk intensive. We assume that each VM requests certain resources (CPU, memory, storage), holds a priority and is initiated by a particular user. Moreover, the actual resource usage of each VM is also monitored. Since, submitting user, resource demand and actual usage have shown strong relationship to runtimes [11], [13]; therefore, we also used these features of more than ten millions tasks (categorised in three different groups w.r.t scheduling constants), using the Google's cluster dataset, to train our prediction model. We used simple (linear regression) to complex (boosted trees) machine learning algorithms to estimate workload runtimes. Moreover, various techniques offer various levels of accuracy and, therefore, variations in experimental outcomes. Moreover, accurate predictions decrease the likelihood of inappropriate migration decisions. Similarly, predicting the runtime of VMs is also influenced by the type of workload hosted in the VM. In [15], the authors describe various approaches to accurate historical data if workloads differ. Further details on workload predictions can be found in [11], [14], [16], that offer reasonable accuracy for public clouds where workloads fluctuate more than private ones, significantly.

3.1.3 Migration Durations Prediction

When consolidating short running workloads, it is possible that the migration efforts are being wasted if the VM terminates during migration or just after its migration process is finished [1]. However, if preference is given to migrate VMs that would highly likely run for a long time after migration, the actual migration time would be a very small fraction of the VMs total lifetime. Further, the probability of the VM powering off during the migration time would be equally minuscule. To decide effective migrations, it is also essential to estimate the migration durations for VMs running different services. In [16], the authors have trained a machine learning approach to predict migration durations and other metrics using real workload dataset. Their investigations suggest that migration durations and performance degradation are, largely, reliant on the migration approach (such as pre-copy, post-copy) and workload type. Moreover, there is a strong linear relationship among the amount of data being copied and migration durations [17]. In off-line migration,

the performance loss (downtime) is almost equal to migration duration. However, in live migration, downtime is different and, usually, smaller than the migration duration [1]. In addition, the memory size of the VM is not always the best indicator on the performance drop when migrating. In many cases it is the rate of change in the dirty pages [17], [18].

In total migration time, page dirty rate plays an important role. To predict migration duration, it is essential that a representative workload is available to train various predictive models. Further, neither Google dataset [19] nor Microsoft Azure dataset [11] contain migration statistics of VMs. Therefore, it is difficult to estimate migration durations for tasks relating to these both datasets. Fortunately, an interesting VM migration dataset is presented in [16]. Therefore, we choose comparable workloads from Google, Microsoft Azure, and the migration datasets provided in [16]. This gives us simplified assumptions for comparing workload benchmarks and, therefore, estimation of accurate migration durations. The model was then trained using various approaches such as linear regression and support vector regression (SVR). Various features, such as VM size, page dirty rate, resource utilisation of VMs, source and destination servers, are considered. We have spent considerable efforts on statistical mapping of various workloads so that plausible assumptions can be derived for simulation purposes. Further details on mapping the Google's workload traces [19] to real IaaS performance benchmarks [4] can be found in our previous works [1], [12].

3.2 Placement and Consolidation

In this section, we explain how the placement and consolidation techniques are being used in the optimisation module. The proposed policies ensure that the most runtime efficient hosts are selected to run particular VMs. The energy, performance aware placement policy, based on first fit technique (EPFF), is described in Algorithm 1. The energy, performance aware migration approach (EPAM) is described in Algorithm 2. The proposed aggregation-based placement is dependent on the runtime efficiencies of the hosts and VMs. First, the available hosts are being divided into groups based on their runtime efficiencies and temporal slack (as described in Section 3.2.1). Furthermore, all hosts in each group are being sorted in increasing order of their ERC values – the objective function (Eq. (1)). Then, based on the runtime of the VM (either predicted or past), using a particular classification technique (such as k-Means clustering algorithm as discussed in Section 3.2.2), every VM is categorised and mapped onto each host in a particular group. In fact, the latter step ensures that VMs of similar characteristics (runtimes, capacities, etc.) are placed onto similar hosts i.e., aggregation (CoLocateMe). In case of segregation, this step is ignored from Algorithm 1 to ensure that VMs are not classified. As a result, the most runtime efficient host (i.e., at top of the list) is selected to run that particular group of VMs. Note that, the placement algorithm is a substantially improved version of the first fit (FF) heuristic approach.

The consolidation policy EPAM runs, periodically, every five minutes interval and looks for possibilities to aggregate and consolidate the workload (VMs) onto fewer hosts. Note that, a shorter time interval will lead to additional overhead but a larger one may lead to poor system performance as reacting to the dynamicity of the workload will be too late. Furthermore, this could happen in two different ways: (i) through migrating VMs from underloaded and overloaded hosts, using some pre-defined threshold values in terms of their utilisation levels (e.g., an upper threshold value U_{upper} for overloaded hosts and a lower threshold value U_{lower} for underloaded hosts) [1]; and (ii) through migrating all VMs from hosts having the highest levels of runtime efficiencies. Once hosts are being identified, a list of migratable VMs is constructed using a particular VM selection policy, using Algorithm 3. Several metrics of the VMs are considered when deciding their migrations. For example, [20] chooses a VM that either: (a) has a small memory so that its migration can be completed quickly; or (b) has maximum utilisation level so that overloaded host is avoided up to maximum. However, [1] prefers to migrate relatively long-running VMs so that their migration efforts are ensured. Moreover, [21] uses volume-to-size (VSR) ratio of a VM to decide its migration. In this paper, we prefer to migrate long-running VMs. Finally, Algorithm 1 is used to place them onto appropriate hosts that consumes less energy and performance is assured along with aggregation or segregation.

Algorithm 1. VM Placement Algorithm (EPFF)

Input: List of hosts (H), List of VM requests (V)
Output: Efficient VM placement

find runtime efficiency of host $h \in H$ (using Eq. (2));
find the temporal slack of host $h \in H$ (using Eq. (3));
categorise H subject to their runtime efficiencies - H_c;
for *each* $vm \in V$ **do**
  estimate (past) or predict runtime of the vm;
  match vm to H_c and pick all suitable hosts (H_m);
  sort H_m in assending order - temporal slack (Eq. (3));
  compute and sort H_m w.r.t ERC values (Eq. (1));
  for *each* $h \in H_m$ **do**
    if *h has enough resources and can run the vm* **then**
      allocate vm to h;
      break the loop and pick the next vm;
    end if
  end for
  if *vm did not fit in any available h* **then**
    start new h and allocate vm;
  else
    "*vm* cannot be allocated";
    "push the *vm* request into W (waiting queue)";
  end if
end for
return output

From implementation perspective, all servers are classified into groups based on their energy consumption and performance (CPU architecture). Workload of particular type is, then, placed on separate server groups, as appropriate. In contrast, workload of any type can be placed on any suitable server in the segregation-based allocation approach. The worst case computational complexity of Algorithm 1 is $\mathcal{O}(nm) + T_p$ where n, m and T_p denote the number of VMs, hosts and runtime prediction time, respectively. Moreover, T_p is dependent on the workload type, historical data and the prediction algorithm. The best case occurs when each VM is allocated in the first iteration. This also applies to Algorithm

2 with additional time for finding migratable VMs and appropriate target hosts. Also, given that resource properties can change over time; and if a runtime approach is adopted, then, potentially there may be oscillatory or repeatable behaviour, e.g., move VM from host X to Y and then back to X. We can use techniques like CMCR i.e., Consolidation with Migration Cost Recovery [1] or put a constraint to avoid such repeatable migrations. We believe, the proposed VM selection policy (Algorithm 3), that prefers to migrate long-running VMs, ensures to control these repeatable migrations, but not essentially.

Algorithm 2. Consolidation Technique (EPAM)

Input: List of hosts (H), List of VMs (V)
Output: Efficient VM placement
1 Using current states of H and V, find overloaded and underloaded hosts (H_{ou}) – predefined thresholds;
2 select all migratabale VMs (V_m) from H_{ou} using a VM selection policy (Algorithm 3);
3 **for** *each* $vm \in V_m$ **do**
4 find a list of all hosts H_n such that $H_n \not\subset H_{ou}$;
5 call VM placement algorithm (H_n, vm) [Algorithm 1];
6 **end for**
7 run this optimisation module periodically;
8 **return** *output*

Algorithm 3. VM Selection Policy

Input: List of migratable VMs (V_m)
Output: Select a suitable VM VM_{fit} for migration
1 $VM_{fit} \leftarrow$ **null**;
2 **for** *each* vm *in* V_m **do**
3 predict runtime or compute past runtime of the vm;
4 **end for**
5 sort V_m – decreasing order of runtimes, workload type;
6 $VM_{fit} \leftarrow V_m[0]$ (*the most suitable VM is on top of list*);
7 **return** VM_{fit}

3.2.1 Runtime Efficiency

The runtime efficiency of server $host$ denotes its total amount of energy consumed (E^{vm}_{host}) when it runs a particular VM up to some expected | past runtime R_{vm}, given by Eq. (2). Since, energy is the product of power consumed (P) for time t (here R); thus, the least value offers an economical placement.

$$E^{vm}_{host} = P^{idle+dynamic}_{host} \times R^{predict|past}_{vm}. \quad (2)$$

Rich literature of the prediction offers various ways to estimate VM' runtimes, as described in Section 3.1. In other research, runtime efficiency is the ratio between the, number of, short running VMs and the longer one (diversity of VM runtimes) while accounting for resource capacities. The slack of each host denotes the difference between its total CPU capacity C and amount of used CPU resources of running VMs i.e., C_{host} - $\sum_{i \in N_{vm}} vm_i$ [8]. The least the slack, the more appropriate will be the placement. While accounting for VMs durations, the temporal slack α of each host is given by

$$\alpha^{vm}_{host} = C_{host}.d_{vm} - \sum_{k \in N^{vm}_{host}} w_k(min\{r_k, r_{vm}\}) - s_{vm}, \quad (3)$$

where C_{host} denotes the host resource capacity or a notion of VM density [1] which can be computed using the host's number of cores and VM sizes (without any concerns whether the VM resources are less, more utilised). Further, r_{vm} and s_{vm} represent the VM release time (expected runtime) and start time, respectively. Similarly, r_k and w_k denote the VM' runtime and CPU demand on k^{th} host. In practice, VM release time and start time are determined by users not by administrators. However, we assume that workload type is known, and the VM runtime d_{vm} is computed as r_{vm} - s_{vm}. N^{vm}_{host} denotes the number of VMs on host. Assigning VMs to host with the least α offer opportunities for switching on/off hosts when it is most cost effective. Hence, α is measured in CPU $\times$ time; therefore, it can be easily translated to energy consumption, performance and cost. Note that, d_{vm} and r_{vm} aggregate demand w.r.t release time. In addition, we only consider CPU in this formulation, since it is predicated on the assumption that CPU is the largest driver of power consumption [1]. However, there are other components such as GPUs, memory, disks, and network devices whose power consumption is also worth considering. More details about the energy consumption and models of these devices can be found in [22]. After computing α for each host, all hosts H are classified into several clusters H_c, using α and k-Means clustering approach. Next, all hosts in each cluster are sorted in increasing order of their α values. Lastly, every VM is assigned, based on its runtime (past or predicted), to a suitable host cluster and, subsequently, holding the least α value.

3.2.2 k-Means Clustering

Using the k-Means clustering approach, we create a set of clusters in order to categorise all VMs. All VMs in a cluster retain comparable features and each VM is mapped to a single cluster. Furthermore, every VM is linked with different types of VM characteristics i.e., runtimes, capacities, workloads (aggregation criteria). To divide these VMs into different categories (z – selected in advance), we characterize each VM as a point in the multi-dimensional R_z space – where each point is a VM and coordinates of the point denote VM categories. The inputs of the algorithm, for certain VM characteristics, are the number of categories (clusters) that are specified by its center point. For each characteristic, the cluster centers are points in the R_z space. In fact, the algorithm allocates N data points to z clusters. In step 1, centers of the z clusters are selected randomly. In step 2, the algorithm allocates each data point to the cluster with the nearest center using a particular distance measure such as euclidean distance. In step 3, these cluster centers are recalculated based on the current assignment. The algorithm repeats step 2 and step 3 until no further changes occur [2].

3.3 Modelling Heterogeneity of Infrastructure

In this section, we explain how energy consumption of virtualised hosts and performance of various workloads (and co-located VMs that compete for similar resources) across several heterogeneous hosts can be modelled. These factors are essential to account for as, potentially, they might have impact on users' costs and service revenues.

3.3.1 Energy Consumption

Energy efficiency of a non-virtualised host could be accurately identified through profiling its various resources for energy measurement. However, the energy consumption of a virtualised host may, possibly, be related to the number of VMs they accommodate. This means that an energy expensive host (virtualised) may, possibly, run a VM more energy efficiently than an energy cheaper, but, non-virtualised host. This relationship could be understood more effectively through relating virtualised and non-virtualised hosts to a bus and a car, respectively. A bus consumes more fuels but still offers cheaper fare than a car. In a similar way, for a particular VM a less energy efficient machine might be more efficient if it can accommodate more VMs. Using the host (non-virtualised) linear power model which is more than 90% accurate, for certain workloads under reasonable assumptions [20], and the most widely used [1], the power/energy consumption of a single VM can be estimated using Eq. (4)

$$\mathcal{P}^h_{vm} = \frac{\mathcal{P}^h_{idle}}{N} + W^h_{vm} \times (\mathcal{P}^h_{busy} - \mathcal{P}^h_{idle}) \times U^h_{vm}, \tag{4}$$

where N is the total number of VMs on a particular host h, $\mathcal{P}^h_{idle}$ and $\mathcal{P}^h_{busy}$ are the power consumed when h is idle (0% utilised) and fully utilised, respectively. Further, W^h_{vm} are the host resources (cores) allocated to the VM and U^h_{vm} is the VM utilisation level. The total energy consumption of the datacenter is the sum of the energy consumed by all hosts; and each host energy consumption could be either: its benchmarked values; a linear relationship to its CPU usage – very similar to Eq. (4); or $\sum^m \mathcal{P}^h_{vm}$ for all m number of VMs running on the virtualised host [1]. Since, for the duration of migration there are exactly two VMs running on source and destination hosts which also cost energy. In order to account for migration energy cost, we use the model suggested in [17]. According to [17], energy is largely consumed by transferring the VM memory; and the amount of energy is directly proportional of the VM size (as given by Eq. (5)). We prefer to use this model because it is more than 90% accurate for certain workloads under reasonable assumptions [17].

$$E_{mig} = 0.512.(VM_{mem}) + 20.165, \tag{5}$$

where VM_{mem} denotes size of the VM and parameters of the linear model (α = 0.512, β = 20.165) are computed through empirical evaluation [17]. Besides memory, disk and network states will also consume energy. Moreover, once the duration of a particular VM migration is predicted, then it is also possible to compute the expected energy consumption through multiplying the source (server) and network energy profiles with durations. However, this might not produce accurate estimation compared to the model in Eq. (5) which already accounts for network and disk state costs.

3.3.2 Performance

Various studies suggest that performance of cloud applications or workloads perform quite differently due to: (i) CPU models [4]; and (ii) resource contention [3], [15], [23]. Regarding (i), similar VMs (workloads) run quite strangely even on same CPU models; which may be related to either design (fabrication process), cache levels and/or memory churns. Largely, the distribution of workload runtimes follows a log-normal pattern across different CPU models [4]. Moreover, a particular workload may run quickly on a specific CPU model, but, may run quite slow on another CPU model. Similarly, a CPU model may run a particular workload quickly, but, another one quite slow. For example, E5430 is faster for bzip2 benchmark than E5507, but, is slower for povray benchmark – as shown in Table 1. Regarding (ii), co-located VMs on a specific host may experience severe performance degradation, particularly, if they compete for same resources (resource interference). The degradation is dependent on the total number of co-located VMs and the workload type they are running on a particular host – as shown in Table 2. In order to model performance variations, we model: (i) CPU heterogeneity as log-normally distributed with respect to workload runtimes; and (ii) resource contention as regression line equation with respect to total number of co-located VMs on a particular host for certain workloads. Moreover, performance of workloads is also affected due to VM migrations; and we account for that, as described in Section 3.1.3. Note that performance, here, refers to sum of all VM runtimes that run workload W (most suitable to users which translates into costs), and is given by

TABLE 1
Execution Times (Seconds) of Various Applications Across Different CPU Models [4]

Workload type	CPU model	Execution times
bzip2	E5430	447 s
	E5507	641 s
povray	E5430	579 s
	E5507	544 s

$$R = \sum_{vm \in W} Runtime_{vm}, \tag{6}$$

where $Runtime_{vm}$ is the wall-clock time (measured in seconds) of each vm involved in running workload W. Further, users are billed according to runtime of each vm as described in Section 4.3.6. Similarly, cost of running a particular workload is the sum of all VM costs.

3.3.3 Workloads

Various workloads have different impacts on infrastructure energy consumption, workload performance, and migration durations. Therefore, it is necessary to characterize workload types, even, if real datasets are used or replayed in simulations [1]. An easy way to characterize workloads is to use their resource utilisation levels. For example, CPU intensive workloads would have large impact on CPU utilisation; but not, essentially, on memory or disk usage. Similarly, memory or disk intensive workloads will have little impact on CPU usage. However, in real, scenarios are completely different, probably, due to CPU heterogeneities. Another approach is to use tasks' priorities, that affects billings, as a proxy to represent workload type [19]. However, this is not reasonable for virtualised workloads [11]. Moreover, our investigation of the Google cluster and Microsoft Azure

TABLE 2
Execution Times (Seconds) of Various Applications on Co-Located VMs [3]

Workload type	CPU model	Number of co-located VMs: 2	4	6	8	10	12
		Execution times					
Grep	E5620	13	14	16	21	31	36
	E7420	20	22	25	29	38	44
Sort	E5620	16	22	38	59	69	78
	E7420	21	28	43	65	76	85

datasets suggests that these workloads (containerised, virtualised) perform quite differently [1]. Therefore, we use monte-carlo simulations to create synthesized workload from real benchmarks workloads that were produced in a real IaaS cloud [4]; and obtain certain features (resource demand and usage, arrival time, submitting users) of the original traces using the laws of statistical distributions and mapping [1]. The synthesized traces follow the features in the original traces.

Table 3 describes the performance (runtimes) of various benchmark workloads (Povray, Namd, Stream) when executed over different CPU platforms [4]. Povray is short-running, Namd is long-running, and Stream is of mixed nature; when run at maximum speed. Note that, stream values originally represent the bandwidth (i.e., data transfer) [4], however, we assume these as durations – since the less data copied, the least time it will take [1]. However, if utilisation levels are normally distributed, then execution times vary. Moreover, distributions of runtimes for a particular workload necessarily follow multi-modal lognormal patterns; where multi-modality relates to CPU architectural heterogeneity. Using laws of lognormal distributions [1], we generated three different synthesized workloads from the reported values i.e., mean (μ), standard deviation (σ), minimum, and maximum, as shown in Table 3. We believe, the generated workloads closely match real workloads; and can be assumed as mix of workloads. Further details on workload modelling and mapping them to real applications' performance benchmarks are described in [12], [24].

4 Performance Evaluation

We assume energy, performance and cost efficient VM placement and consolidation as types of bin-packing problem that can be solved using various heuristics such as first fit, best fit. Energy can be decreased via increasing the resource utilisation levels; that subsequently minimises the number of used servers. Similarly, performance can be ensured either via: (a) relocating workloads to best performing hosts; and/or (b) minimising co-location. In both cases, the proposed scheduler ensures to put similar workloads onto same hosts such that energy and performance efficiencies are achieved. Albeit, techniques like linear programming can be used to come up with an optimal or approximate solution [25]. However, for large-scale systems consisting thousands of servers and variety of workloads, we prefer quickness rather than optimality – as happens in real clouds such as Intel [11].

TABLE 3
Different Benchmarks Runtime Parameters [4]

Benchmark workload	CPU model	Real benchmarks runtimes: (μ)	(σ)	Min	Max	CoV
Povray	E5430	439	11	421	467	0.025
	E5-2650	468	12	451	500	0.026
	E5645	507	10	490	535	0.02
Namd	E5-2651	1994	41.9	1952	2036	0.021
	E5-2650	2007	28.5	1978	2036	0.014
	E5645	2043	96.4	1946	2140	0.047
	E5430	2160	20.7	2135	2189	0.01
	E5507	2187	18.1	2162	2217	0.008
Stream	E5430	1446	66	1328	1572	0.045
	E5507	2348	104	2078	2448	0.044
	E5645	3395	287	2995	4008	0.085
	E5-2650	5294	191	4935	5860	0.036

4.1 Experimental Set-Up

In order to evaluate the proposed policies, the CloudSim [7] simulator was extensively modified to simulate a real heterogeneous datacenter as close as possible. For example, classes were added to account for: CPU architectural heterogeneity, performance of co-located VMs, migration costs in terms of energy consumption and performance loss, VM level power consumption, and predicting workload runtimes, migration durations. Details of the extended version of the CloudSim i.e., PerficientCloudSim are elaborated in [24]. Moreover, performance degradation due to migrations, migration durations, and workload runtimes are predicted using various machine learning techniques such as linear regression, SVR and gradient boost trees [11], [16]. Note that, we used the well-known implementations of these algorithms using the *sklearn* package. The energy consumption of various servers is computed according to SPECpower[1] benchmarks. Furthermore, if servers are idle with no workload running (0% utilised), we still assume them as switched on and, therefore, consume their idle power ($\mathcal{P}_{idle}$). The energy consumption of a single VM and virtualised host is computed using the linear power model which is suggested more than 90% accurate [20].

Our simulated datacenter is a direct modelling of the Google cluster [19] that comprises 12,583 heterogeneous servers that belong to five types, as shown in Table 4. Speeds of various servers were mapped to millions of instructions per second (MIPS) in order to be consistent with the CloudSim. For aggregation-based VM placement, all available servers are grouped into five different clusters – based on these five types of CPU models. For example, all servers of CPU model "E5430" denote a separate cluster. Virtual machines of six various sizes and speeds were assumed running three different kinds of workloads (as shown in Table 3). The utilisation levels of all workloads were modelled as normally distributed with respect to prior studies [20]. Frequencies of VMs, as shown in Table 5, were mentioned in vCPUs (cores), converted to ECUs (EC2 Compute Unit) and mapped to MIPS rating, accordingly. The ECU is described as: "equivalent

1. https://www.spec.org/power_ssj2008/

TABLE 4
Servers Types and Characteristics for Simulated Datacenter [ECU = CPU Speed (GHz) × Number of Cores]

CPU model	Speed (MHz)	No of Cores	No of ECUs	Memory (GB)	Storage (TB)	$\mathcal{P}_{idle}$ (Wh)	$\mathcal{P}_{busy}$ (Wh)	Amount of hosts
E5430	2,830	8	22.4	16	4	166	265	
E5507	2,533	8	20	8	8	67	218	
E5645	2,400	12	28.8	16	4	63.1	200	12,583
E5-2650	2,000	16	32	24	8	52.9	215	
E5-2651	1,800	12	21.6	32	12	57.5	178	

CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor" and its rating is per vCPU/core; therefore, the VM total rating is the multiple of cores (number) and ECU rating. The rating is, then, translated to MIPS for consistency with CloudSim as it does not support the notion of ECU. Note that, the large difference in storage capacities of VMs, which ensures heterogeneity, but this will have a clear impact on the migration costs. Performance parameters for servers and VMs (workloads) were taken from real experimental values, as demonstrated in [1], [4]. Various heuristics, that aggregate or segregate workloads using different features such as runtimes, were considered for initial VM placement. At five minutes interval, the optimisation module searches for consolidation opportunities – if utilisation level of a server exceeds 80% or drops below 20% which are two pre-defined threshold values. Our empirical evaluation was accomplished using two different approaches for VM live migration i.e., pre-copy and post-copy. Moreover, workload sizes (runtimes) were transformed to equivalent MIPS over a rating of 2 GHz CPU. From implementational simplification point of view, performance loss or gain was modelled as subtraction or addition of MIPS to the workload size, respectively.

In order to demonstrate the impact of EPC-aware VM placement and optimisation on infrastructure energy consumption, workload performance and service costs, we consider different approaches to VM placement (first fit - FF, energy aware first fit - EFF, energy and performance aware first fit - EPFF) and consolidation with migration (no migration - NO, migrate all - ALL, energy aware migration - EAM, energy and performance aware migration - EPAM) [1]. Note that, VMs selected for migrations are also placed on target servers using these heuristics. In addition, we account for migration energy and performance costs. For example, in ALL approach, all migratable VMs are given chances to migrate; however, in EAM and EPAM those migratable VMs are migrated which can recover their migration costs [1]. Moreover, these policies are implemented using two different methodologies to placement i.e., segregation-based and aggregation-based. The former one ensures that various workloads run in mix (i.e., colocated across a single cluster) while the later one puts similar workloads (based on runtimes, workload types) on same servers (same CPU architectures, similar runtime efficiencies). Similarly, the proposed methodologies have been implemented in two different ways: (i) using past runtimes [1]; and (ii) using certain prediction techniques to predict workloads' runtimes [16].

TABLE 5
Amazon Various Instances and Their Characteristics – Mem Means Memory & vCPU Denotes a Hyperthreaded Core

Instance type	No of vCPUs	No of ECUs	Speed (GHz) MIPS	Mem (GB)	Storage (GB)
t2.nano	1	1	1.0	0.5	1
t1.micro	1	1	1.0	0.613	1
t2.micro	1	1	1.0	1	1
m1.small	1	1	1.0	1.7	160
m1.medium	1	2	2.0	3.75	410
m3.medium	1	3	3.0	3.75	4

4.2 Evaluation Metrics

Data for various metrics, such as energy consumption (KWh), performance or runtime (seconds), total number of migrations, ERC, resource usage statistics, was collected during simulations. Moreover, prediction accuracy is computed in terms of absolute error both for runtimes (AE_{alloc}) and migration durations (AE_{mig}). The AE denotes the divergence of the estimated value from the actual value in absolute units i.e., seconds and converted to hours.

4.3 Results and Discussion

The results, averaged over ten runs, are shown in Table 6. Our evaluation suggests that workloads run more energy and performance efficiently and, therefore economically, if aggregated onto separate clusters or co-located w.r.t certain metrics and scheduling policies. Moreover, a significant decrease in total number of migrations can be observed; as workloads were initially placed on appropriate servers. Effective allocation techniques are more economical than consolidation approaches; and we suspect, perhaps, this might be a reason that public service providers do not migrate workloads for energy or performance aware computation in their clusters. Furthermore, if migration costs (in terms of energy consumption and performance loss) are considered, then the migrate all approach can be much expensive than the no migration approach. Similarly, if we migrate things only to energy efficient servers, it degrades workload performance and, therefore, may consume more energy due to the existing trade-off between energy consumption and performance (runtimes) [1]. These findings are, largely, consistent with previous outcomes [1], [12]. However, if performance is taken into account, significant

TABLE 6
Average Results for Various Combinations of VM Allocation and Migration Policies – The Lowest Values are 'Best' [± Denotes Standard Deviation, the Least Value for ERC Denotes the Most Effective and EPC Aware Placement Policy]

Policy allocation	migration	No. of migrs	Energy (KWh)	Performance (hours)	ERC $\times 10^6$	No. of migrs	Energy (KWh)	Performance (hours)	ERC $\times 10^6$	Absolute error AE_{alloc}	Absolute error AE_{mig}
SEGREGATION-BASED PLACEMENT											
		Past runtimes				Runtimes prediction					
	NO	0	511.93	302.78±0.02	287.2	0	511.93	302.78±0.12	287.2	-	-
FF	ALL	5231	547.23	349.71±0.21	409.6	6390	552.7	356.98±0.29	431.1	0.35	0.08
	EAM	3211	493.31	278.02±0.09	233.4	4009	525.66	321.03±0.26	331.5	0.42	0.07
	EPAM	1021	443.4	211.67±0.1	121.6	1921	461.52	235.76±0.21	157	0.29	0.09
	NO	0	503.39	291.43±0.08	261.7	0	503.39	291.43±0.14	261.7	-	-
EFF	ALL	4123	520.04	313.56±0.41	312.9	4522	525.56	320.9±0.51	331.2	0.32	0.06
	EAM	2198	511.25	301.87±0.32	285.1	2390	525.97	321.45±0.42	332.6	0.39	0.06
	EPAM	1082	444.89	213.66±0.21	124.3	1693	457.28	230.12±0.39	148.2	0.44	0.08
	NO	0	465.96	241.67±0.62	166.6	0	465.96	241.67±0.92	166.6	-	-
EPFF	ALL	3382	443.57	211.9±01.2	121.9	3319	479.24	259.32±1.3	197.2	0.49	0.1
	EAM	1502	439.22	206.11±0.44	114.2	1699	460.86	234.89±0.56	155.6	0.5	0.11
	EPAM	921	434.77	200.2±0.31	106.6	1256	459.45	233.01±0.34	152.7	0.49	0.07
AGGREGATION-BASED PLACEMENT											
		Past runtimes				Runtimes prediction					
	NO	0	510.76	301.23±0.03	283.6	0	510.76	301.23±0.13	283.6	-	-
FF	ALL	2898	494.63	279.78±0.73	237	3033	501.41	288.79±0.76	255.9	0.29	0.12
	EAM	1677	459.52	233.1±0.56	152.8	1799	485.75	267.98±0.51	213.5	0.38	0.11
	EPAM	922	434.45	199.78±0.33	106.1	1209	458	231.08±0.42	149.7	0.4	0.09
	NO	0	495.4	280.81±0.47	239.1	0	495.4	280.81±0.42	239.1	-	-
EFF	ALL	1999	488.51	271.65±0.35	220.6	2777	508.35	298.02±0.33	276.3	0.27	0.13
	EAM	911	491.36	275.43±0.21	228.1	1455	503.75	291.9±0.45	262.7	0.5	0.12
	EPAM	706	445.16	214.01±0.08	124.8	951	468.38	244.88±0.9	171.9	0.48	0.06
	NO	0	463.23	238.03±0.11	160.6	0	463.23	238.03±0.18	160.6	-	-
EPFF	ALL	2001	462.25	236.73±0.54	158.5	2231	473.65	251.89±0.67	183.9	0.2	0.11
	EAM	1109	485.56	267.72±0.39	213	1589	473.21	251.3±0.87	182.9	0.46	0.13
	EPAM	799	433.61	198.66±0.64	104.7	988	449.05	219.19±0.55	132	0.33	0.08

energy, performance gains and, therefore, users costs can be saved.

4.3.1 Aggregation versus Segregation

Table 6 shows that aggregation-based placement and/or consolidation (based on workload runtimes) is approximately 9.61% energy and 20.0% performance efficient than segregation-based methodology. The least value for ERC shows the most EPC efficient placement. Fig. 2 describes the percentage improvements, in energy consumption and performance, of using runtime-based aggregation rather than segregation. However, this may not be essentially true for all workloads – as there are certain applications that could perform the best if segregated using other metrics such as workload type, VM sizes etc. For example, if various workloads are placed aggregated (W_1 is placed on servers with CPU model E5430, while W_2 is placed on servers with CPU model E5-2650, and so on), they result in lower utilisation level of resources, as shown in Table 12. In short, segregation-based policies offers high levels of datacenter utilisation, with the least performance loss, for particular workloads.

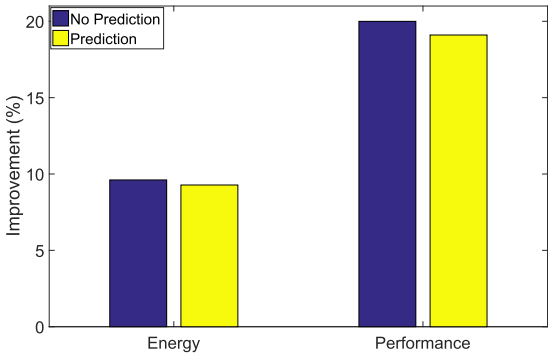

Fig. 2. Percentage improvements in energy and performance using aggregation-based VM placement instead of segregation, using EPFF allocation and EPAM migration [Boosted Gradient Trees prediction].

Similarly, if VMs are aggregated on VM sizes, then resources are wasted (stranded resources) [26]. If VM sizes are same, then both approaches are comparable. However, for various sizes of VMs segregation packs them closely, which: (a) increases resource utilisation (energy efficient); and (ii) higher chances of resource contention (less performance and cost-efficient). Our evaluation suggests that aggregation of VMs, based on workload type, is not ensuring EPC aware placement at all – as shown in Table 12 (observe ERC values for various workloads and methodologies). This is justifiable as similar workloads often compete for same resources which results in worse performance issues. Furthermore, we observed that using past runtimes for aggregation-based placement and migration of workloads always produces

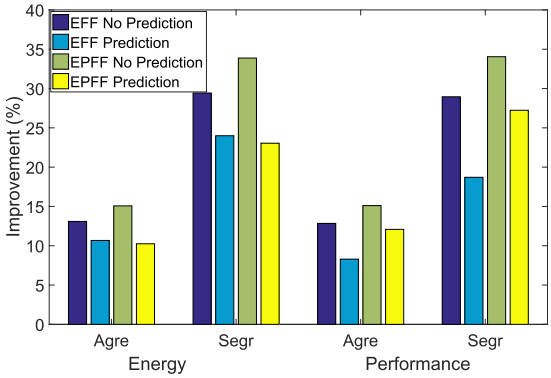

Fig. 3. Percentage improvements in energy and performance using EFF and EPFF placement techniques rather than FF [Boosted Gradient Trees prediction].

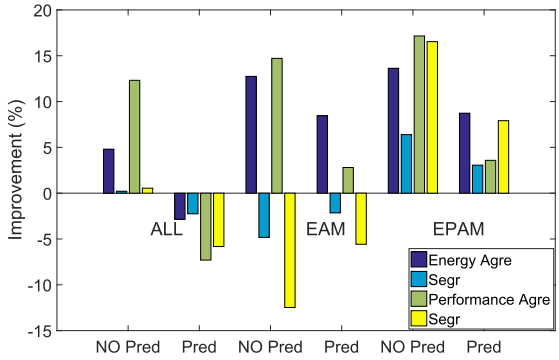

Fig. 4. Percentage improvements in energy and performance using ALL, EAM and EPAM migration with Boosted Gradient Trees prediction rather than no migration [the bars below 0% indicate worse approaches – EPAM outperforms ALL and EAM policies].

best results. However, if runtimes and migration durations are being predicted, then inaccurate predictions may lead to worse results even than segregation-based methods. There are no EPC benefits derived from runtime prediction; and using machine learning methods may decrease the metrics of interest. This suggests to further investigate other metrics for aggregation-based resource management in IaaS heterogeneous clouds.

4.3.2 Energy versus Performance Aware Allocation

If we allocate workloads on energy efficient servers (or through energy aware placement policy - EFF), then neither energy nor performance efficiency is assured – since energy efficient servers are not essentially performance efficient. Theoretically, energy efficiency is guaranteed; however, lower performance means longer runtimes and these longer durations translate to more energy consumption (i.e., energy performance trade-off) [1]. Moreover, if workloads are placed initially to energy, performance efficient servers (or through energy, performance aware scheduling - EPFF), then both energy and performance are assured. Fig. 3 shows the percentage improvement, in energy consumption and performance, of using EFF and EPFF allocation polices instead of a simple FF approach.

4.3.3 Energy versus Performance Aware Migrations

Previous research findings, as demonstrated in [1], [12], suggest that migrations are costly and sometimes it might be even more economical not to migrate. Moreover, if a particular workload is being migrated several times, repeatedly, it may suffer from severe performance degradation and, therefore, may consume more energy. Therefore, if migrations are controlled through some methodology e.g., (i) migrate relatively long-running workloads [1]; (ii) migrate to energy efficient servers - EAM; then energy might be saved. Further, if migrations are performed to energy, performance efficient servers (or through energy, performance aware policies - EPAM), them both energy and performance are guaranteed. Fig. 4 shows the percentage degradation or improvement, in energy consumption and workload performance, of using ALL, EAM and EPAM migration polices instead of no migration approach (using boosted tress i.e., B. TREE prediction method). Furthermore, due to the existing trade-off between energy consumption and performance (runtime), migration to energy efficient servers only is not economical.

4.3.4 Impact of Predictions on Energy and Performance

As described earlier, workload runtimes and migration durations play an important role in placement and consolidation decisions, particularly, if their objectives are energy efficiency and/or performance gains. To decide energy efficient migrations, such as CMCR [1] and CPER i.e., Consolidation with migration Energy, Performance Cost Recovery [12], runtimes and migration durations are being compared. Therefore, their predictions and accuracy will have an impact on total number of migrations, which may subsequently affect energy consumption and performance. Fig. 5 shows that good prediction technique (such as boosted trees) may offer relatively accurate results over linear regression, SVR (comparable); and, therefore, almost negligible savings and performance gains. Albeit, the improvement is actually trivial (for certain kinds of workloads), and may even be worse if the overhead of the prediction strategy is taken into account. However, for other workloads' types, considerably higher improvements were observed. This is, possibly, due to the sizes of different datasets gathered for various applications. These findings are inline with [27] that demonstrates the efficiency of using a simple averaging method over using complex learning approaches. Figs. 2, 3, and 4 indicate that the performance of using prediction techniques to predict runtimes is worse than

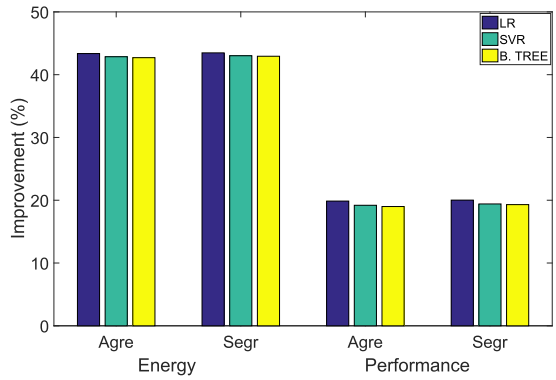

Fig. 5. Impact of various runtimes prediction techniques on energy consumption and workload performance – the lowest values are the best [LR - linear regression, SVR - support vector regression, B. TREE - Boosted Gradient Trees].

TABLE 7
Container Types and Their Characteristics [12]

Container type	Speed (MHz)	Cores	ECU's	Memory (MB)
A	1,000	1	1	128
B	1,225	1	1.23	256
C	1,500	1	1.5	512

TABLE 8
Costs Savings [Energy and Users Monetary Costs are Described in US Dollars]

Policy	Energy costs ($)	Users monetary costs ($)	Total costs savings (%)
Segregation	2202.78	1149.87	-
Aggregation	1732.65	931.56	18.99

that using past runtimes. For migration scenarios (Fig. 4), the performance of prediction-based approaches in many cases is even worse than no migration approaches. However, the prediction overhead of boosted trees is higher due to its computational complexity. Therefore, if the accuracy difference among the three techniques is small, simple linear regression may be a better choice. Therefore, it is recommended, as a future work, to take the prediction overhead of the three techniques into account to provide more convincing experimental results. This suggests the importance of workload prediction in cost-efficient management of datacenter' resources.

4.3.5 Running Containerised Workloads Over VMs

Since, containers are replacing VMs, therefore, it is essential to account for containerised workloads [12]. We are aware that lot of more discussions and experiments is needed to show and discuss the differences between VM-based and container-based cloud infrastructures. Therefore, readers should look for our previous works to understand our observations and findings for containerised workloads [12], [28]. In this section, we describe how aggregation and segregation based placement and consolidation policies would affect energy consumption, performance and costs of workloads that run within: containers directly; or containers that subsequently run within VMs [29]. In addition to earlier experimental set-up, as explained in Section 4.1, we illustrated three container types with characteristics shown in Table 7. We assume that each VM can run several containers. Further, the same allocation policy, which is used to place VMs on servers, was also used to place containers on VMs.

We observed comparable outcomes when containers run on virtualised IaaS resources (inside VMs), as shown in Fig. 6. Albeit, servers were largely more utilised, but, with no benefits. This demonstrates that increased levels of datacenter utilisation may not be always beneficial from energy savings point of view. Moreover, significant performance loss was seen, surprisingly, when containerised workloads that run directly on servers were aggregated based on the workload type. We suspect this might be a possible reason for service providers' that prefer to segregate their workloads. Unexpectedly, when containers were aggregated onto VMs based on their runtimes; then, besides reduced total number of migrations potential energy savings and comparable performance was achieved. This experiment suggests that, for diverse workload types, segregation-based approaches outperform aggregation-based techniques.

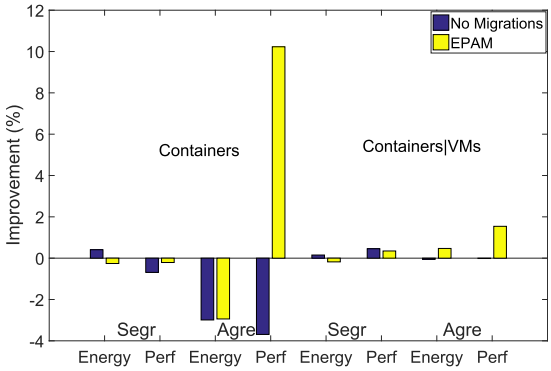

Fig. 6. Percentage improvements in energy and performance when running workloads in: (i) containers; and (ii) virtualised containers, instead of only VMs.

4.3.6 Costs Savings

The total electricity bill, user monetary costs and costs savings (in US dollars - $) are described in Table 8. For this analysis, we assume a PUE[2] of 1.10 and energy price of $0.88 per KWh[3] that mimic a Google datacenter located in the Oklahoma State, USA. Moreover, we assume that users' bills are computed at the rate of $0.0017 per second.[4] The cost of running a particular user's workload is $C_{user} = \sum_{vm}^{user} 0.0017.Runtime_{vm}$; where the runtime of each VM is in seconds. For certain workloads, service providers could save up to ~21.34% energy costs (bills) using aggregation-based placement techniques instead of segregation. Moreover, users' monetary costs could be reduced up to ~8.39 to 18.99%.

Although, the least users' monetary costs would certainly affect the providers' economics (less profit), however, they can attract more customers which can recoup back these losses (large business). Moreover, the above savings will translate to a million dollars per year for hyper-scale IaaS clouds, such as Amazon AWS and Google, that consist of clusters with more that millions servers to offer resources at large scale. Table 9 shows the percentage of savings possible in energy consumption, performance improvement and users' costs, when using various techniques in relation to CoLocateMe. It is clear that "CoLocateMe" (aggregation-based policies) offers significant performance improvements and energy savings.

4.3.7 Significance of Results

To demonstrate the, significant, statistical differences between the means of the obtained results using proposed methods and others, we performed the *t-test* analysis. Table 10 shows

2. https://www.google.co.uk/about/datacenters/efficiency/
3. https://www.eia.gov/electricity/monthly/
4. https://aws.amazon.com/ec2/pricing/

TABLE 9
Percentage of Savings Possible, Using Various Techniques, in Terms of Energy Consumption, Performance and Cost [+ Means Performance Gains and - Indicates Performance Loss]

Work	[8]	[12]	[18]	[1]	[28]	CoLocate Me
Energy	∼30	43.31	-	3.66	30.47	9.61
Performance	-	+1.09	+16.0	+1.87	-2.14	+20.0
Cost	-	14.78	-	13.56	-	18.99

TABLE 10
Statistical Significance of Results [FF and ALL are "Base" Allocation and Migration Policies for Comparison]

Policy	Allocation			Migration		
	FF	EFF	EPFF	ALL	EAM	EPAM
Segregation	-	0.311	0.048	-	0.045	0.044
Aggregation	-	0.135	0.042	-	0.041	0.043

the p values for various allocation and migration policies. It can be seen that energy aware allocation (EFF), only, may be worse than non-energy aware placement (FF). Similarly, aggregation-based policies offers lower p values (*t-critical* = 2.774) than segregation-based policies. The failure of the *t-test* for FF and EFF policies is, perhaps, due to the overlaps that exist in the collected dataset; however, EFF outperforms FF based on the mean values.

Further, PerficientCloudSim is suggested to produce approximately 98.63% accurate results as compared to a real IaaS cloud [1], [12]. This means that approximately $\pm$1.37% error is expected in our simulated outcomes. Thus, the proposed aggregation-based policy is approximately 9.61[$\pm$0.13]% more energy, and 20.0[$\pm$0.27]% more performance efficient than segregation-based policies. Table 9 describes that these savings, in energy and performance gains, are significant as compared to other segregation-based policies.

4.3.8 Comparative Study

Table 11 offers a comparative study of the proposed CoLocateMe approach to other competing methods including Heifer [3], iAware [18], Granite [30], and IGGA [31]. The results were obtained both for aggregation and segregation methodologies. In case of aggregation, compared with other approaches, CoLocateMe could save approximately 3.02% – 7.22% energy and improve the workload's performance i.e., 0.34% – 6.6%. The CoLocateMe approach is $\sim$ 1.11% – 2.54% more energy efficient and 0.58% – 4.48% more performance efficient than other methods when segregation is taken into account. As, CoLocateMe prefers aggregation, therefore, the savings are potentially small in the segregation environment. Furthermore, the proposed approach has lower ERC values than all the closest rivals. The results demonstrate that aggregation is ∼11.08% – 24.67% more energy efficient but 0.98% – 8.86% less performance efficient than the segregation. Moreover, aggregation by workloads decreases performance but could be more energy efficient than through aggregating with runtimes. Using various statistical methods, we believe that these outcomes are reasonable against a real IaaS cloud i.e., ∼98.99% accurate [24].

4.3.9 Generalisation of Outcomes

In order to find consistency in our results and scalability of our proposals, we evaluated the proposed techniques using a variety of heterogeneous dynamic workloads, heterogeneous servers, various metrics for aggregation (such as runtime, workload type), and datacenter sizes. The experiments were carried out using experimental set-up and mathematical models, as described earlier in Section 4.1. In additions, three workload types W_1, W_2 and W_3 which belong to tasks of three different priorities (0, 2, 9) from Google's cluster dataset, are also investigated. Furthermore, besides workload runtimes, their type (based on the priority or resource usage – CPU, memory, disk intensive) are considered for aggregation and segregation. We observed that certain workloads, if aggregated using other features such as workload type, may perform 'best' using segregation-based placement. However, our findings are largely consistent regarding datacenter and workloads sizes which means that our approach can be scaled for cost-efficient resource management in hyper-scale datacenters such as Google, AWS, and Azure.

Table 12 describes the results which were obtained using previous experimental parameters and set-up, as initially was described in Section 4.1. Largely, we observed that segregation-based VM placement offers fewer opportunities for migrations. Less number of migration opportunities may ensure workload performance, however, it reduces resource utilisation levels. Moreover, workloads aggregated or segregated using various metrics (such as VM sizes, workload type, submitting users) offer variations in energy consumption and performance, therefore, costs. Our evaluation suggests that aggregating VMs based on their workload types in not ensuring EPC aware placement at all. For example, similar VM types may not be tightly packed on servers, in aggregation, and resources are wasted. However, segregation can ensure tight packing of VMs, but, increases resource contention due to co-location. Furthermore, aggregating workloads of similar duration allowing for more servers to be powered down to save energy. Segregation may imply either: (a) having all servers switched on, and minimising the number of VMs per host; or (b) putting the shortest runtime VMs onto hosts with the longest runtime (the greatest runtime efficiency). In respect of (b), workload performance should be better than (a); because there are more servers and resource contention is lessened (the period of the short runtime), rather than a period of time closer to the longest runtime.

We observed that aggregating VMs that have similar types of workloads could lead to high resource contention, interference (and possibly performance degradation that can be $\sim 12.2\%$) for CPU activity, as shown in Table 12. However, if VM sizes are assumed as running different types of workloads e.g., CPU, memory, disk intensive; then, the contention will be low. Therefore, it is useful to aggregate VM types that have different resource requirements – as this will reduce energy use ($\sim 7.51\%$) and performance overheads ($\sim 13.63\%$), as shown in Table 12. For aggregating and segregating with respect to the workload type, we

TABLE 11
Comparison of CoLocateMe With Other Methods [Energy is Measured in KWh and Performance in Seconds]

Policy	Aggregation						Segregation		
	Runtime			Workload					
	Energy	Performance	ERC	Energy	Performance	ERC	Energy	Performance	ERC
Granite	15178.56	654.48	9.93	17032.34	699.98	11.92	19834.51	681.56	13.52
IGGA	15694.43	653.74	10.26	17521.4	696.09	12.2	20122.1	673.9	13.56
Heifer	16168.9	612.17	9.9	17011.22	667.11	11.35	19981.66	655.67	13.1
iAware	16285.09	613.73	9.99	17634.11	661.26	11.66	19831.9	654.82	12.99
CoLocateMe	15108.87	611.28	9.24	16498.3	659.04	10.87	19611.6	651.03	12.77

TABLE 12
Results Generalisation Using Various Approaches to Aggregation and Different Kinds of Workloads (Using EPFF Allocation and EPAM Migration Policies); Datacenter Size Denotes the Total Number of Servers and VMs; and VM Sizes Refer to Different Workloads i.e., CPU, Memory, Disk Intensive – the Lowest Values for ERC Represent EPC Aware Placement

Workload type	Agg. | seg. metric	Datacenter size	No. of migrs	Energy (KWh)	Performance (hours)	ERC 10^6	No. of migrs	Energy (KWh)	Performance (hours)	ERC 10^6
				Aggregation				Segregation		
W_1		3 k - 50 k	672	71.22	19.34	0.16	528	71.26	19.56	0.17
W_2	runtime	6 k - 70 k	563	167.13	88.2	7.96	500	167.78	90.01	8.32
W_3		9 k - 0.1 m	501	295.73	171.9	53.48	487	311.69	201.56	77.5
W_1, W_2		6 k - 0.12 m	0	174.91	109.89	12.93	0	175.92	112.7	13.67
W_1, W_2		6 k - 0.12 m	1098	171.81	101.23	10.78	1001	175.51	111.56	13.37
W_2, W_3	runtime	9 k - 0.17 m	1891	352.75	277.89	166.71	1792	203.25	279.01	96.83
W_1, W_3		6 k - 0.15 m	1056	203.49	189.56	44.75	934	207.75	201.44	51.59
W_1, W_2, W_3		12 k - 0.22 m	3221	578.79	391.67	543.39	2875	584.83	399.7	571.81
W_1, W_2, W_3		12 k - 0.22 m	0	702.63	556.3	1330.75	0	652.16	489.21	955.2
W_1, W_2		6 k - 0.12 m	1389	174.99	110.11	12.98	1238	171.36	99.98	10.48
W_2, W_3	workload	9 k - 0.17 m	1690	365.42	301.43	203.2	1782	358.24	288.09	181.98
W_1, W_3	type	6 k - 0.15	980	208.27	202.88	52.46	995	207.39	200.45	51
W_1, W_2, W_3		12 k - 0.22 m	2150	617.33	442.9	741.11	2201	576.7	388.89	533.77
W_1, W_2		6 k - 0.12 m	1288	174.04	107.45	12.3	1499	178.3	119.34	15.54
W_2, W_3	VM size	9 k - 0.17 m	1185	359.86	291.09	186.61	1282	376.79	322.57	239.94
W_1, W_2, W_3		12 k - 0.22 m	1976	584.89	399.77	572.07	2019	632.35	462.87	829.14

observed opposite findings as compared to VM sizes – segregation is better than aggregation. This type of profiling is particularly relevant in a real-time context; which needs further investigation.

5 Related Work

In cloud computing, resource allocation algorithms use certain features of the infrastructure, and workloads in order to run them in respect of achievable objective (energy, performance, cost). For example, [1] used the host (virtualised) or VM efficiency factor (E_f) to minimise IaaS energy consumption and improve (at least maintain) the workload performance levels. Other works [20], also use the host energy efficiency metric (i.e., hosts with the least energy consumption) for efficient placement; however, this metric may not accurately measure the efficiency of a heterogeneous virtualised host [1]. However, due to the existing trade-off among energy and runtime (performance), energy cannot be saved with these methods. In such circumstances, performance must be considered during allocation and migration decisions. In the cloud literature, various research, as demonstrated in [1], [15], [18], [30], [32], have considered performance of workloads along with energy efficiency during resource placement and migration decisions. In [30], the authors proposed a scheduling technique, called Granite, to reduce datacenter energy consumption that accounts for CPU temperature. In Granite, a VM is assigned to a server that results with the least increase in energy consumption after allocation. Further, the migration policy selects VMs from servers based on their temperature levels. Dabbagh *et al.* [8] used workload runtimes to place relatively similar-running VMs onto same hosts. In practice, predicting workload runtimes can be a daunting task. Moreover, all workloads of a particular user could be, possibly, placed on same hosts or VMs.

In practice, public clouds are opaque and service providers are not aware of the workloads they are hosting on their infrastructure. Albeit, there are various efforts towards workload runtimes prediction [11], [13], [16], but, in practice they are not reasonable – as public cloud workloads differ significantly from private ones. Therefore, an alternative approach is to use their past runtimes. The only reason which supports this idea is that Google's tasks that run for

TABLE 13
Summary of the Related Work, Closest to CoLocateMe, With Respect to Various Evaluation Criteria

Parameters		Related Work [8]	[5]	[33]	[34]	[35]	[18]	[6]	[31]	[29]	[28]	[36]	[1]	[20]	CoLocate Me
Platform	VMs	✓	✓	✓	✓		✓		✓		✓	✓	✓	✓	✓
	Containers							✓			✓				✓
	Containers | VMs					✓				✓	✓				✓
Metrics	Energy	✓	✓			✓			✓		✓	✓	✓	✓	✓
	Performance		✓				✓	✓	✓		✓		✓	✓	✓
	Migration cost								✓				✓		✓
	User costs										✓		✓		✓
	Co-location						✓								✓
Placement method	Aggregation	✓		✓											✓
	Segregation		✓					✓							✓
Scheduler	Single				✓					✓	✓				✓
	Distributed				✓			✓				✓			
	Hierarchical				✓								✓	✓	
Aggregation criteria	Runtimes	✓													✓
	Workload type														✓
	VM size														✓
Management policy	Allocation	✓	✓		✓			✓		✓	✓	✓	✓	✓	✓
	Migration				✓	✓	✓		✓		✓		✓	✓	✓
	Sharing resources			✓											

more than seven hours continue running for several days or even months [19], [26]. This is further evidenced in Microsoft Azure cluster [11]; however, this may not be essentially true. If we assume that clouds are not opaque; then it is possible to predict their runtimes using historical data [13], [14]. For example, Cortez *et al.* [11] used gradient boosted tree method to predict VMs runtimes in Microsoft Azure cloud. They also found a close relationship among VMs runtimes, submitting users, and job names (logical). Tumanov *et al.* [13] predicted job runtimes using various characteristics of the workloads in order to automate resource allocation.

A resource level server disaggregation technique, as described in [33], integrates various resources (such as CPU, memory, storage) from multiple servers into a single pool. With server disaggregation it is also possible to run a single VM on multiple servers which provides higher chances for maximising resource utilisation. Moreover, it offers an easy way to enable vertical resource scaling (adding more resources) of VMs. From resource allocation perspective, server disaggregation simplifies the VM scheduling problem to only one dimension. However, aggregation and segregation based VMs placement and consolidation techniques are not explored. Lebre *et al.* [34] have also discussed various VM placement and consolidation techniques in terms of three different schedulers: centralised, hierarchical and distributed. Tchana *et al.* [35] suggested software or application migration to achieve energy efficiency in datacenters. Wu *et al.* [31] also studied VMs consolidation while accounting for energy consumption and migration costs i.e., performance loss in terms of downtime. The authors suggested a consolidation scheme i.e., improved grouping genetic algorithm (IGGA). The swapping criteria ensures that energy costs are reduced through avoiding unnecessary migrations. Jiang *et al.* [36] proposed an adaptive resource allocation algorithm that dynamically allocates resources to VMs energy efficiently.

In [37], a solution for improved IaaS energy consumption while keeping SLA violations minimum, is proposed. The proposed scheme uses energy aware methods that are founded over adaptive three threshold framework (ATF), policies for VM selection like maximum ratio of CPU utilisation to memory utilisation (MRCU), minimum product of CPU and memory utilisation (MPCU), and maximum energy efficient VM placement (VPME). The results show that optimal energy efficiency and lower SLA violations could be achieved. In [38], the authors offer a survey of proactive methods for VM placement while classifying them as per their application in forecasting procedures. The predicted strategies are efficient and produce minimal VM overheads. An in-depth analysis of predictive VM placement algorithms is illustrated. A dynamic consolidation policy (ETAS) holistically manages IaaS resources through creating a trade-off between computing resources and cooling systems [39]. An improvement in energy utilisation is seen as compared to thermal aware methods.

The peak efficiency-aware scheduling (PEAS) algorithm initially defines metrics like peak power efficiency, and optimal energy usage for heterogeneous servers [40]. Later, it allocates and consolidates VMs over servers so that maximum power efficiency is achieved. It helps to optimise energy usage, overall system performance, and QoS metrics. Similarly, a VM placement algorithm (PLVMP) ensures to achieve better VM performance and balanced server workload between user and service provider [41]. The PLVMP method improves VM placement and creates an effective load balance among various servers. In terms of energy consumption, an in-depth analysis of proactive VM consolidation techniques and algorithm is offered in [42]. In [43], authors focused over maximising the IaaS profits while solving the problem as multi-objective optimisation issue. An evolutionary algorithm is

suggested that optimises revenue of the IaaS providers while SLAs are guaranteed.

An overview of state-of-the-art algorithms for energy efficient datacenters and large-scale multimedia services is offered in [44]. It also outlines key challenges while designing and managing green datacenters. Further, the authors highlight several possibilities for green streaming services. In [45], a continuous replica placement method is proposed that is based on greedy heuristics. The placement method creates a new replica while taking into account the optimisation criteria and various constraints such as energy, costs, etc. Majority of the above techniques consider segregation-based placement and consolidation; while aggregation remains relatively unexplored. Furthermore, with the notable exception of [2], [8], VM runtimes, sizes and workloads they run, are not evaluated for similar placement and consolidation decisions. The summary of the comparison between our proposed technique "CoLocateMe" and other closely related works is given in Table 13. We believe, Table 13 would help our readers to quickly identify gaps for further research.

6 Conclusion and Future Work

In this paper, through empirical evaluation we demonstrated how various approaches to VM placement and consolidation, and methodologies such as aggregation and segregation, would affect the energy, performance and cost efficiencies of large-scale IaaS providers. Our findings show that, for certain workload types, significant energy could be saved while their performance is ensured; through aggregating them on same servers. Moreover, aggregating workloads of similar duration allows for more servers to be switched off to save energy. However, if workloads are aggregated based on their types or other metrics, then they suffer from severe performance degradation. Our evaluation also suggests that if containers (instead of VMs) are aggregated based on their workloads types (instead of runtimes), then segregation-based placement methods might potentially outperform aggregation-based techniques.

Further research is needed to determine what kinds of workload are not suitable for aggregation, segregation, and/or migration. Similarly, investigation of workload runtimes, their accurate prediction and other suitable metrics such as workload type, sizes, is needed for segregation-based VM placement. Furthermore, there is a need for the investigation of other metrics-based aggregated and segregated techniques and their potential impact on energy consumption, performance, and costs. In future research, we will investigate how aggregation and segregation based resource management would affect oversubscribed resources. Moreover, the linear power model is hard to take at face value; as it only covers CPU resource. Modern processors have P-states and C-states (as opposed to merely busy/idle) and fairly involved automated switching between them, which we should consider in future research. Moreover, as a future plan we will work in order to implement the proposed strategies in a real-world public cloud.

Acknowledgments

The authors are thankful to Changyeon Jo, from SNU, Korea, for providing us VMs migration dataset.

References

[1] M. Zakarya and L. Gillam, "Energy and performance aware resource management in heterogeneous cloud datacenters," Ph.D. dissertation, Dept Comput. Sci., Univ. Surrey, Guildford, U.K., 2017.

[2] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "An energy-efficient VM prediction and migration framework for overcommitted clouds," *IEEE Trans. Cloud Comput.*, vol. 6, no. 4, pp. 955–966, Oct.–Dec. 2018.

[3] F. Xu, F. Liu, and H. Jin, "Heterogeneity and interference-aware virtual machine provisioning for predictable performance in the cloud," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2470–2483, Aug. 2016.

[4] J. O'Loughlin, "A workload-specific performance brokerage for infrastructure clouds." Ph.D. dissertation, Dept Comput. Sci., Univ. Surrey, Guildford, U.K., 2018.

[5] Y. Cheng, A. Anwar, and X. Duan, "Analyzing Alibaba's co-located datacenter workloads," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 292–297.

[6] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with borg," in *Proc. 10th Eur. Conf. Comput. Syst.*, 2015, Art. no. 18.

[7] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw.: Pract. Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[8] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Release-time aware VM placement," in *Proc. Globecom Workshops*, 2014, pp. 122–126.

[9] V. Gupta, "Stochastic models and analysis for resource management in server farms," Ph.D. dissertation, Sch. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2011.

[10] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. Kozuch, "Energy-efficient dynamic capacity provisioning in server farms," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-10–108, 2010.

[11] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms," in *Proc. 26th Symp. Oper. Syst. Princ.*, 2017, pp. 153–167.

[12] A. A. Khan, M. Zakarya, R. Buyya, R. Khan, M. Khan, and O. Rana, "An energy and performance aware consolidation technique for containerized datacenters," *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1305–1322, Oct.–Dec. 2021.

[13] A. Tumanov, A. Jiang, J. W. Park, M. A. Kozuch, and G. R. Ganger, "JamaisVu: Robust scheduling with auto-estimated job runtimes," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-PDL-16-104, 2016.

[14] W. Smith, I. Foster, and V. Taylor, "Predicting application run times with historical information," *J. Parallel Distrib. Comput.*, vol. 64, no. 9, pp. 1007–1016, 2004.

[15] N. Rameshan, Y. Liu, L. Navarro, and V. Vlassov, "Augmenting elasticity controllers for improved accuracy," in *Proc. IEEE Int. Conf. Autonomic Comput.*, 2016, pp. 117–126.

[16] C. Jo, Y. Cho, and B. Egger, "A machine learning approach to live migration modeling," in *Proc. Symp. Cloud Comput.*, 2017, pp. 351–364.

[17] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster Comput.*, vol. 16, pp. 249–264, 2011.

[18] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, "iAware: Making live migration of virtual machines interference-aware in the cloud," *IEEE Trans. Comput.*, vol. 63, no. 12, pp. 3012–3025, Dec. 2014.

[19] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: Format+schema," Google Inc., Mountain View, CA, USA, White Paper, 2011, pp. 1–14.

[20] 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 Computation: Pract. Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[21] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Sandpiper: Black-box and gray-box resource management for virtual machines," *Comput. Netw.*, vol. 53, no. 17, pp. 2923–2938, 2009.

[22] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Commun. Surv. Tut.*, vol. 18, no. 1, pp. 732–794, Jan.–Mar. 2016.

[23] P. Leitner and J. Cito, "Patterns in the chaos—A study of performance variation and predictability in public IaaS clouds," *ACM Trans. Internet Technol.*, vol. 16, no. 3, 2016, Art. no. 15.

[24] M. Zakarya, L. Gillam, A. A. Khan, and I. U. Rahman, "PerficientCloudSim: A tool to simulate large-scale computation in heterogeneous clouds," *J. Supercomputing*, vol. 77, no. 4, pp. 3959–4013, 2021.
[25] T. C. Ferreto, M. A. S. Netto, R. N. Calheiros, and C. A. F. De Rose, "Server consolidation with migration control for virtualized data centers," *Future Gener. Comput. Syst.*, vol. 27, no. 8, pp. 1027–1034, 2011.
[26] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proc. 3rd ACM Symp. Cloud Comput.*, 2012, Art. no. 7.
[27] D. Tsafrir, Y. Etsion, and D. G. Feitelson, "Backfilling using system-generated predictions rather than user runtime estimates," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 6, pp. 789–803, Jun. 2007.
[28] A. A. Khan, M. Zakarya, and R. Khan, "H^2 – A hybrid heterogeneity aware resource orchestrator for cloud platforms," *IEEE Syst. J.*, vol. 13, no. 4, pp. 3873–3876, Dec. 2019.
[29] I. Mavridis and H. Karatza, "Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing," *Future Gener. Comput. Syst.*, vol. 94, pp. 674–696, 2019.
[30] X. Li, P. Garraghan, X. Jiang, Z. Wu, and J. Xu, "Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1317–1331, Jun. 2018.
[31] Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, "Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters," *IEEE Trans. Serv. Comput.*, vol. 12, no. 4, pp. 550–563, Jul./Aug. 2019.
[32] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions," *Proc. IEEE*, vol. 102, no. 1, pp. 11–31, Jan. 2014.
[33] P. Svärd, "Dynamic cloud resource management: Scheduling, migration and server disaggregation," Ph.D. dissertation, Fac. Sci. Technol., Dept. Comput. Sci., Umeå universitet, Umeå, Sweden, 2014.
[34] A. Lebre, J. Pastor, A. Simonet, and M. Südholt, "Putting the next 500 VM placement algorithms to the acid test: The infrastructure provider viewpoint," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 1, pp. 204–217, Jan. 2019.
[35] A. Tchana, N. De Palma, I. Safieddine, and D. Hagimont, "Software consolidation as an efficient energy and cost saving solution," *Future Gener. Comput. Syst.*, vol. 58, pp. 1–12, 2016.
[36] H.-P. Jiang and W.-M. Chen, "Self-adaptive resource allocation for energy-aware virtual machine placement in dynamic computing cloud," *J. Netw. Comput. Appl.*, vol. 120, pp. 119–129, 2018.
[37] Z. Zhou *et al.*, "Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms," *Future Gener. Comput. Syst.*, vol. 86, pp. 836–850, 2018.
[38] M. Masdari and M. Zangakani, "Colorbluegreen cloud computing using proactive virtual machine placement: Challenges and issues," *J. Grid Comput.*, vol. 18, no. 4, pp. 727–759, 2020.
[39] S. Ilager, K. Ramamohanarao, and R. Buyya, "ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation," *Concurrency Computation: Pract. Experience*, vol. 31, no. 17, 2019, Art. no. e5221.
[40] W. Lin, W. Wu, and L. He, "An online virtual machine consolidation strategy for dual improvement in performance and energy conservation of server clusters in cloud data centers," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 766–777, Mar./Apr. 2022.
[41] H. Zhao, Q. Wang, J. Wang, B. Wan, and S. Li, "VM performance maximization and PM load balancing virtual machine placement in cloud," in *Proc. 20th IEEE/ACM Int. Symp. Cluster, Cloud Internet Comput.*, 2020, pp. 857–864.
[42] S. Ismaeel, R. Karim, and A. Miri, "Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres," *J. Cloud Comput.*, vol. 7, no. 1, pp. 1–28, 2018.
[43] S. Khalid, A. Ishfaq, and K. E. Mohammad, "An evolutionary approach to optimize data center profit in smart grid environment," in *Proc. 2nd Int. Conf. Data Intell. Secur.*, 2019, pp. 89–96.
[44] H. Yuan, C.-C. J. Kuo, and I. Ahmad, "Energy efficiency in data centers and cloud-based multimedia services: An overview and future directions," in *Proc. Int. Conf. Green Comput.*, 2010, pp. 375–382.
[45] T. Loukopoulos, P. Lampsas, and I. Ahmad, "Continuous replica placement schemes in distributed systems," in *Proc. 19th Annu. Int. Conf. Supercomputing*, 2005, pp. 284–292.