CoLocateMe: Aggregation-based, EPC Aware VM Placement and Consolidation in Large-scale Heterogeneous Clouds

Hashim Ali, Muhammad Zakarya, Izaz Ur Rahman, Ayaz Ali Khan, Omer Rana, Lee Gillam, 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 same machines, they might suffer from contention, particularly, if they are competing for similar resources. High level of resource contention may degrade VMs performance, and, therefore, could potentially increase users' costs and infrastructure' energy consumption. Furthermore, segregation-based methods result in stranded resources and, therefore, less economics. The recent industrial interest of segregating workloads opens new directions for research. In this paper, we demonstrate how aggregation and segregation-based VM placement policies leads 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 utilization 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 jobs in separate clusters (hosts) that potentially needs more hosts and, therefore, may results in stranded resource. 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 exception of the Google's cluster [6]. Perhaps, inspired from benefits of segregation-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. The switch from aggregation to segregation-based approaches motivate us to perform this study.

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 minimized 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. Major contributions of the research conducted in this paper are:

- *H. Ali, M. Zakarya, I.U. Rahman, and A.A. Khan are with the Department of Computer Science, Abdul Wali Khan University, Pakistan. O. Rana is the University of Cardiff, UK. L. Gillam is with the University of Surrey, UK. R. Buyya is with the Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, University of Melbourne, Australia.*
E-mail(s): {hashimali, mohd.zakarya, izaz, ayazali}@awkum.edu.pk ranaof@cardiff.ac.uk, l.gillam@surrey.ac.uk, rbuyya@unimelb.edu.au

- 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;
- the proposed policies account for migration costs in terms of energy consumption, performance loss;
- 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 Sec. 2, we discuss the VM placement problem. In Sec. 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 Sec. 4. We offer an overview of the related work in Sec. 5. Finally, Sec. 6 concludes the paper and describes future research.

2 Problem Description

The runtime period or execution time (R) of a 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, performance and cost efficient resource management policies. Moreover, performance of workloads and possible loss due to resource contention must be taken into account.

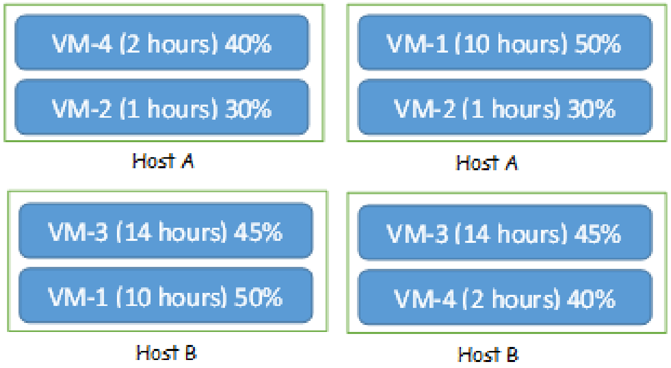


Fig. 1: Problem description [aggregation w.r.t runtimes]

Considering a datacenter which comprises two hosts 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, then both hosts will run 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. 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. Moreover, instances that run similar workloads and applications may also be aggregated to same resources or cluster of resources. However, if workloads compete for similar resources, then performance of the workloads may potentially be affected [3]. O'Loughlin et al. [9] demonstrated that this degradation can be more than 100% for certain workload types.

Furthermore, similar workloads may perform quite differently across same CPU models [4]. These variations may potentially affect workload runtimes, therefore, users' service costs; and energy consumption. Subsequently, energy consumption affects revenue of service providers (in terms of energy bills) and our environment (in terms of green house gases). Therefore, it is essential to account for these costs when deciding resource placement and consolidation, in particular, if placements are made with aggregation and segregation-based approaches. We believe, to the best of our knowledge, aggregation and segregation-based VM placement and consolidation approaches are not well explored in the context of energy and performance efficiencies of large-scale heterogeneous clusters and IaaS datacenters.

The above problem can be formulated as a multi-objective optimisation problem where objective are: (a) minimise total energy consumption of the datacenter ($E = \sum_{i=1}^{N} E_{host_i}$); improve or, at least, maintain workload performance (P); and minimise users' monetary costs (C). Note that, P and C are directly proportional where improving workload' P means reducing its R which subsequently means reducing C. Moreover, multi-objective optimisation problems can be solved in two different ways: (i) concurrently solve all objectives; and (ii) solve one objective first, and then make it a constraint on the next one. Moreover, various objectives (E, P, C) can be combined into a single metric ($ERC = \frac{EC}{P}$) 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.

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 [10], have been suggested to predict VMs runtimes. However, due to the unpredictable nature of the cloud workloads, many efforts are needed. In the first part of this section, we explain the aggregation-based placement and consolidation policies. In the second part, we explain the methodologies and approaches to implement these policies.

3.1 Placement and Consolidation

In this section, we explain how the placement and consolidation technique are being used in the optimisation module. The energy, performance aware placement policy, based on first fit technique (EPFF), is described in Alg. 1. The energy, performance aware migration approach (EPAM) is described in Alg. 2. The proposed aggregation-based placement is dependent on the runtime efficiencies of the hosts and VMs. First of all, the available hosts are being divided into groups based on their runtime efficiencies (as described in Sec. 3.1.1). Furthermore, all hosts in each group are being sorted in decreasing order of their runtime efficiencies. Then, based on the runtime of the VM, using a particular classification technique (such as K-means clustering algorithm), every VM is mapped onto each host in a particular group. As a result, the most runtime efficient host (i.e. at top of the list) is selected to run that particular VM. Note that the placement algorithm is a modified version of the first fit (FF) heuristic approach.

Algorithm 1: VM placement algorithm (EPFF)

```
Input: List of hosts ($H$), List of VM requests ($V$)
Output: Efficient VM placement
find runtime efficiency of each host $h \in H$ (using Eq. 2);
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 - runtime efficiency;
    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
```

The consolidation policy runs, periodically, every five minutes interval and looks for possibilities to aggregate 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 predefined 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 Alg. 3. Several metrics of the VMs are considered when deciding their migrations. For example, [11] 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, [12] uses volume-to-size (VSR) ratio of a VM to decide its migration. In this paper, we prefer to migrate long-running VMs. Finally, Alg. 1 is used to place then on appropriate hosts that consumes less energy and performance is assured.

Algorithm 2: Consolidation technique (EPAM)

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

Algorithm 3: VM selection policy

```
Input: List of migratable VMs ($V_m$)
Output: Select a suitable VM $VM_{fit}$ for migration
$VM_{fit} \leftarrow$ null;
for each $vm$ in $V_m$ do
    estimate runtime of $vm$;
end for
sort $V_m$ in decreasing order of runtimes;
$VM_{fit} \leftarrow V_m[0]$ (long-running VM is on top of the list);
return $VM_{fit}$
```

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 Alg. 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 Alg. 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 (Alg. 3), that prefers to migrate long-running VMs, ensures to control these repeatable migrations, but not essentially.

3.1.1 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 (R); thus, the least value offers an economical placement.

$$E_{host}^{vm} = P_{host}^{idle+dynamic} \times R_{vm}^{predict|past} \qquad (2)$$

Rich literature of the prediction offers various ways to estimate VM' runtimes, as described in Sec. 3.2. 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 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:

$$\alpha_{host}^{vm} = C_{host}.d_{vm} - \sum_{k \in N_{host}^{vm}} w_k(min\{r_k, r_{vm}\}) - s_{vm} \qquad (3)$$

where C_{host} denotes the host capacity, r_{vm}, s_{vm}, and w_i represent the VM release time, start time and CPU demand, respectively. Moreover, the VM runtime d_{vm} is computed as r_{vm} - s_{vm}. N_{host}^{vm} 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×time; therefore, it can be easily translated to energy consumption, performance and cost. 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 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 [10]. However, there would be other efficient ways of doing the same, for example using workload types, VM sizes, as described later in Sec. 4.3.6.

3.2.1 Past Runtimes

In practice, public clouds are opaque and service providers are not aware of the workloads they are hosting on their infrastructure. Therefore, from IaaS point of view, workloads and their runtimes cannot be predicted accurately. Albeit, we are aware of various efforts towards workload runtimes prediction [10], [13], [14], however, in practice they are not reasonable – as public cloud workloads differ significantly from private ones. 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 only reason which supports this idea is that Google's tasks that run for more than seven hours continue running for several days or even months [15], [16]. Albeit, this is further evidenced in Microsoft Azure cluster [10]; however, this may not be essentially true. The idea is based on our previous works [1], [17]; that use VMs or containers past runtimes (durations for which VMs/containers have already run) in workload allocation and consolidation decisions – *migrate only relatively long-running VMs and/or containers since they could recover their migration costs*. 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.

3.2.2 Runtimes Prediction

If we assume that clouds are not opaque which means that the provider has knowledge of the user's workloads; then it is possible to predict their runtimes using historical data [14], [18]. For example, Cortez et al. [10] 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. [14] predicted job runtimes using various characteristics of the workloads in order to automate resource allocation. 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 [10], [14]; 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 [19], the authors describe various approaches to accurate historical data if workloads differ. Further details on workload predictions can be found in [10], [13], [18], that offer reasonable accuracy for public clouds where workloads fluctuate more than private ones, significantly.

3.2.3 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]. To decide effective migrations, it is also essential to estimate the migration durations for VMs running different services. In [13], 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 [20]. 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].

Live migration durations are strongly dependent on various factors such as VM size and the workload its running. Moreover, page dirty rate plays an important role in total migration time. In order to predict migration duration, it is essential that a representative workload is available to train various predictive models. Further, neither Google dataset [15] nor Microsoft Azure dataset [10] 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 [13]. Therefore, we choose comparable workloads from Google, Microsoft Azure, and the migration datasets provided in [13]. 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 workload mapping can be found in our previous works [1], [17].

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 and most widely used [1], the power/energy consumption of a single VM can be estimated using Eq. 4:

$$P_{vm}^{h} = \frac{P_{idle}^{h}}{N} + W_{vm}^{h} \times (P_{busy}^{h} - P_{idle}^{h}) \times U_{vm}^{h} \tag{4}$$

where N is the total number of VMs on a particular host h, P_{idle}^{h} and P_{busy}^{h} are the energy consumed when h is idle (0% utilised) and fully utilised, respectively. Further, W_{vm}^{h} are the host resources (cores) allocated to the VM and U_{vm}^{h} is the VM utilisation level. 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 [20]. According to [20], 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.

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

where VM_{mem} denotes the size of VM. 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], [21]; and (ii) resource contention [3], [19], [22]. 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. 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 Sec. 3.2.3. 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:

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

where $Runtime_{vm}$ is the wall-clock time of each vm involved i running workload W. Further, users are billed according to runtime of each vm as described in Sec. 4.3.7. 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

TABLE 1: Execution times (seconds) of various applications across different CPU models [4]

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

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

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 [15]. However, this is not reasonable for virtualised workloads [10]. Moreover, our investigation of the Google cluster and Microsoft Azure 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 follow certain features (resource demand and usage, arrival time, submitting users) of the original traces.

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

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) [9], 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.

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 [23]. However, for large-scale systems consisting thousands of servers and variety of workloads, we prefer quickness rather than optimality.

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. 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 [10], [13]. 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 (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 [11].

Our simulated datacenter 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 [11]. 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)	P_{IDLE} (Wh)	P_{MAX} (Wh)	Amount of hosts
E5430	2,830	8	22.4	16	4	166	265	12,583
E5507	2,533	8	20	8	8	67	218	
E5645	2,400	12	28.8	16	4	63.1	200	
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 2GHz CPU. From implementational simplification point of view, performance loss or gain was modelled as subtraction or addition of MIPS to the workload size, respectively.

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

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 workload runs in mix 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 prediction techniques to predict runtimes [13].

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. 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], [17]. However, if performance is taken into account, significant 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

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 affective and EPC aware placement policy]

Policy		No. of	Energy	Performance	ERC	No. of	Energy	Performance	ERC	Absolute error	
allocation	migration	migrs	(KWh)	(hours)	$\times 10^6$	migrs	(KWh)	(hours)	$\times 10^6$	AE_{alloc}	AE_{mig}
SEGREGATION-BASED PLACEMENT											
		Past runtimes				Runtimes prediction					
FF	NO	0	511.93	302.78±0.02	287.2	0	511.93	302.78±0.12	287.2	-	-
	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
EFF	NO	0	503.39	291.43±0.08	261.7	0	503.39	291.43±0.14	261.7	-	-
	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
EPFF	NO	0	465.96	241.67±0.62	166.6	0	465.96	241.67±0.92	166.6	-	-
	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					
FF	NO	0	510.76	301.23±0.03	283.6	0	510.76	301.23±0.13	283.6	-	-
	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
EFF	NO	0	495.4	280.81±0.47	239.1	0	495.4	280.81±0.42	239.1	-	-
	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
EPFF	NO	0	463.23	238.03±0.11	160.6	0	463.23	238.03±0.18	160.6	-	-
	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

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 8. In short, segregation-based policies offers high levels of datacenter utilisation, with the least performance loss, for particular workloads.
Similarly, if VMs are aggregated on VM sizes, then resources are wasted (stranded resources) [16]. 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. 8 (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 best results. However, if runtimes and migration durations are being predicted, then inaccurate predictions may lead to worse results even than segregation-based methods. This suggests to further investigate other metrics for aggregation-based resource management in IaaS heterogeneous clouds.

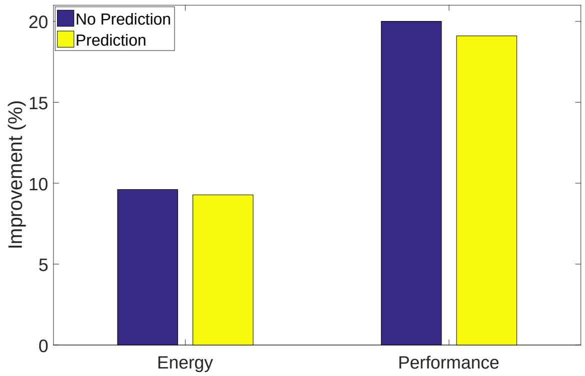


Fig. 2: %age improvements in energy and performance using aggregation-based VM placement instead of segregation, using EPFF allocation and EPAM migration [B. TREE prediction]

4.3.2 Energy vs. 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 vs. Performance Aware Migrations

Previous research findings, as demonstrated in [1], [17], suggest that migrations are costly and sometimes it might be even more economical not to migrate. Moreover, if a particular

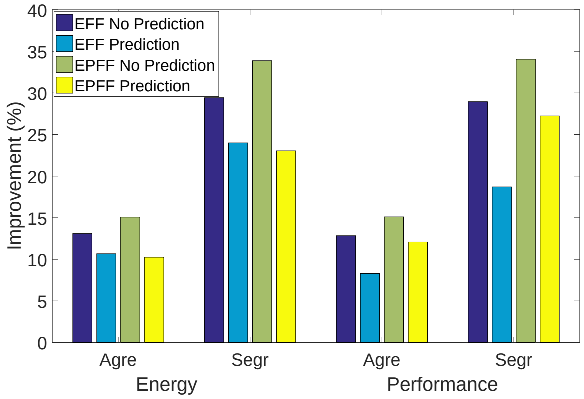


Fig. 3: Percentage improvements in energy and performance using EFF and EPFF placement techniques rather than FF [B. TREE prediction]

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.

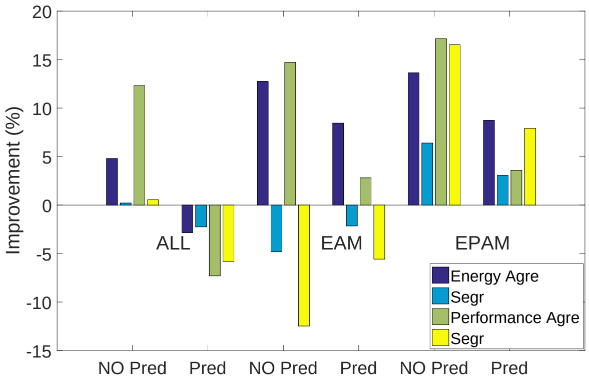


Fig. 4: %age improvements in energy and performance using ALL, EAM and EPAM migration with B. TREE prediction rather than no migration [the bars below 0% indicate worse approaches – EPAM outperforms ALL and EAM policies]

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 [17], 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) offers relatively accurate results over linear regression and, therefore, large savings and performance gains. This suggests the importance of workload prediction in cost-efficient management of datacenter' resources.

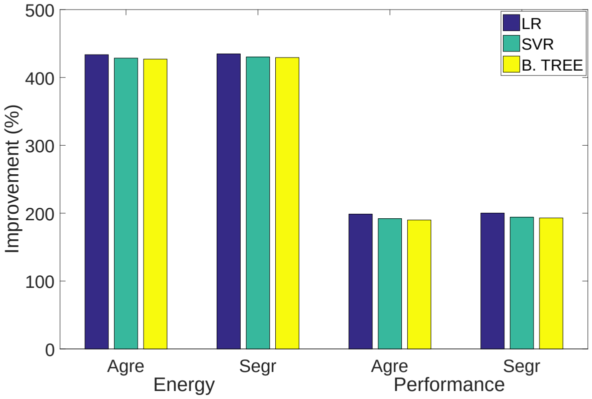


Fig. 5: Impact of various runtimes prediction techniques on energy consumption and workload performance – the lowest values are the best [LR - linear regression, SVR - support vector regression, B. TREE - boosted tree]

4.3.5 *Running Containerized Workloads over VMs*

Since, containers are replacing VMs, therefore, it is essential to account for containerised workloads [17]. 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 [24]. In addition to earlier experimental set-up, as explained in Sec. 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.

TABLE 7: Container types and their characteristics [17]

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

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 containerized 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.

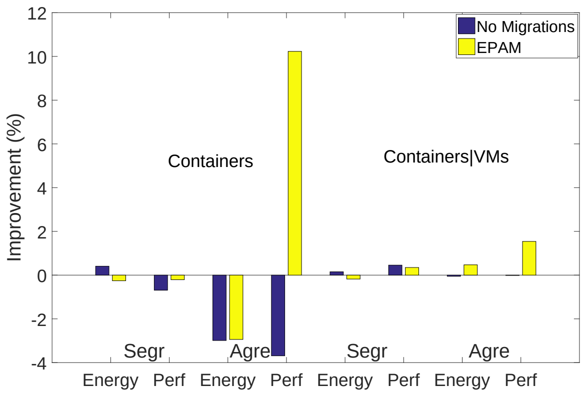


Fig. 6: Percentage improvements in energy and performance when running workloads in: (i) containers; and (ii) virtualised containers, instead of only VMs

4.3.6 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 Sec. 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.

Table 8 describes the results which were obtained using previous experimental parameters and set-up, as initially was described in Sec. 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. 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$12.2%) for CPU activity, as shown in Table 8. However, if VM sizes are assumed as running different types of workloads e.g. CPU, memory, disk intensive; then, contention will be low. Therefore, it is useful to aggregate VM types that have different resource requirements – as this will reduce energy use ($\sim$7.51%) and performance overheads ($\sim$13.63%), as shown in Table 8. This type of profiling is particularly relevant in a real-time context.

4.3.7 Costs Savings

The total electricity bill, user monetary costs and costs savings (in US dollars - $) are described in Table 9. 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 $\sim$21.34% energy costs (bills) using aggregation-based placement techniques instead of segregation. Moreover, users' monetary costs could be reduced up to $\sim$8.39 to 18.99%.

TABLE 9: 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

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.

4.3.8 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. This is carried out through computing the probability error (p value) by the *t-critical* ratio. If $p \leq 0.05$, then the difference between two datasets is statistically significant. At p = 0.05 (with 95% confidence interval), the differences between means have a 5% probability of occurring by chance [17]. Table 10 shows 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

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

TABLE 8: 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	runtime	3k - 50k	672	71.22	19.34	0.16	528	71.26	19.56	0.17
W_2		6k - 70k	563	167.13	88.2	7.96	500	167.78	90.01	8.32
W_3		9k - 0.1m	501	295.73	171.9	53.48	487	311.69	201.56	77.5
W_1, W_2	runtime	6k - 0.12m	0	174.91	109.89	12.93	0	175.92	112.7	13.67
W_1, W_2		6k - 0.12m	1098	171.81	101.23	10.78	1001	175.51	111.56	13.37
W_2, W_3		9k - 0.17m	1891	352.75	277.89	166.71	1792	203.25	279.01	96.83
W_1, W_3		6k - 0.15m	1056	203.49	189.56	44.75	934	207.75	201.44	51.59
W_1, W_2, W_3		12k - 0.22m	3221	578.79	391.67	543.39	2875	584.83	399.7	571.81
W_1, W_2, W_3	workload type	12k - 0.22m	0	702.63	556.3	1330.75	0	652.16	489.21	955.2
W_1, W_2		6k - 0.12m	1389	174.99	110.11	12.98	1238	171.36	99.98	10.48
W_2, W_3		9k - 0.17m	1690	365.42	301.43	203.2	1782	358.24	288.09	181.98
W_1, W_3		6k - 0.15	980	208.27	202.88	52.46	995	207.39	200.45	51
W_1, W_2, W_3		12k - 0.22m	2150	617.33	442.9	741.11	2201	576.7	388.89	533.77
W_1, W_2	VM size	6k - 0.12m	1288	174.04	107.45	12.3	1499	178.3	119.34	15.54
W_2, W_3		9k - 0.17m	1185	359.86	291.09	186.61	1282	376.79	322.57	239.94
W_1, W_2, W_3		12k - 0.22m	1976	584.89	399.77	572.07	2019	632.35	462.87	829.14

EFF policies is, perhaps, due to the overlaps that exist in the collected dataset; however, EFF outperforms FF based on the mean values.

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

Further, CloudSim is suggested to produce approximately 98.63% accurate results as compared to a real IaaS cloud [1], [17]. This means that approximately ±1.37% error is expected in our simulated outcomes. Thus, the proposed aggregation-based policy is approximately 9.61[±0.13]% more energy, and 20.0[±0.27]% more performance efficient than segregation-based policies. Table 12 describes that these savings, in energy and performance gains, are significant as compared to other segregation-based policies.

5 Related Work

In cloud computing, rich literature has addressed the VM placement and consolidation problems. They can be classified based on their goals and objectives such as energy efficiency, users' costs minimisation and performance gains. Both, VM placement and consolidation are treated as bin-packing issues; and are, largely, solved using various heuristics instead of optimal solutions due to problem sizes. These heuristics use certain features of the infrastructure, and workloads in order to run them in respect of achievable objective. 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 [11], 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], [19], [25], [26], [27], have considered performance of workloads along with energy efficiency during resource placement and migration decisions.

Largely, datacenters are not well utilised, and various placement policies try to pack workloads onto fewer hosts for increasing resource utilisation levels. A very simple approach is to classify the workloads, aggregate similar workloads, and schedule them onto similar hosts. However, [6] shows that instead of using aggregation-based policies, segregation-based placement could result in high levels of resource utilisation. This unpredictability can, largely, be related to the heterogeneity of hosts and workloads. Note that, workloads classification are achieved through investigating various characteristics of the workloads such as priority, submitting user, and resource demands. Besides the difficulties involved in predicting workloads, various other metrics could be used to aggregate workloads. For example, [8] used workload runtimes to place relatively similar-running VM 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. Albeit, this may offer additional security and privacy to containerised users, but, VMs and hosts are relatively safe.

Before 2015, Alibaba datacenters used to run various workloads on separate clusters [5]. However, appropriate scheduling, which may result in high utilisation levels, is possible through coordinated schedulers – that had become a common practice in today's datacenters [6]. Aggregation-based placement can also be used to assign appropriate resources to workloads, in hybrid clouds that run different kinds of sand-boxing technologies such as containerisation, virtualisation, nested containers, and bare-metal. Workloads could be placed where they could perform to their expected levels of performance. For example, workloads that perform best in containers might be scheduled on containerised infrastructure instead of bare-metal and virtualised resources (VMs). More-

TABLE 11: Summary of the related work, closest to CoLocateMe, with respect to various evaluation criteria

Parameters		Related Work													CoLocate Me
		[8]	[5]	[29]	[30]	[31]	[25]	[6]	[32]	[24]	[28]	[33]	[1]	[11]	
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			✓											

over, appropriate workload allocation and relocation decisions are possible through a centralised scheduler that runs on top of multiple schedulers (distributed or hierarchical) for various sand-boxing technologies that cooperatively manage hybrid clouds [28].

A resource level server disaggregation technique, as described in [29], 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. Note that, in horizontal scaling VM sizes or their number are scaled/increased over resources of a single or multiple server(s). 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. [30] have also discussed various VM placement and consolidation techniques in terms of three different schedulers: centralised, hierarchical and distributed. Tchana et al. [31] suggested software or application migration to achieve energy efficiency in datacenters. Wu et al. [32] also studied VMs consolidation while accounting for energy consumption and migration costs i.e. performance loss in terms of downtime. Jiang et al. [33] proposed an adaptive resource allocation algorithm that dynamically allocates resources to VMs energy efficiently.

Majority of the above techniques, including our own methods [1], [28], consider segregation-based placement and consolidation; while aggregation remains relatively un-addressed in terms of energy consumption and performance. We believe, to the best of our knowledge an investigation of aggregation and segregation-based resource management techniques is not available in the existing cloud state-of-the-art. Furthermore, with notable exception of [2], [8], VM runtimes, sizes and workloads they run, are not evaluated for similar placement and consolidation decisions. Table 12 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. The summary of the comparison between our proposed technique "CoLocateMe" and other closely related works is given in Table 11. We believe, the information in Table 11 would also help our readers to quickly identify gaps for further research.

TABLE 12: %age of savings possible, using various techniques, in terms of energy consumption, performance and cost [+ means performance gains and - indicates performance loss]

Work	[8]	[17]	[25]	[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

6 Conclusions 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. Moreover, using performance efficient migrations, certain workloads can finish their executions quickly and, thus, reducing users monetary costs. 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; and can run more energy and performance efficiently without being segregated, aggregated and/or migrated. Similarly, investigation of workload runtimes, their accurate prediction and other suitable metrics such as workload type, sizes, is needed for segregation-based VM placement which is currently used in many production clouds with the only exception of Alibaba's cloud and Google's cluster. Albeit, similar VM placement techniques have been demonstrated in the literature [8], however, it is assumed that runtimes of workloads are known in advance. Furthermore, there is a need for the investigation of other metrics-based aggregated and segregated VM placement and consolidation techniques and their potential impact on infrastructure energy consumption and workload performance. A study of robust deep learning based prediction techniques might be useful to estimate the migration and runtimes of workloads; and the heterogeneity of resources which can ensure workload independent Epc-aware resource/VM allocation and consolidation in IaaS clouds. In future research, we will investigate how aggregation and segregation based resource management would affect oversubscribed resources.

Acknowledgments

This work was supported, in part, by the AWK University, Pakistan, and, in part, by an Australian Research Council (ARC) Discovery project at the University of Melbourne, Australia. We 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, University of Surrey, 2017.

[2] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "An energy-efficient vm prediction and migration framework for overcommitted clouds," *IEEE Transactions on Cloud Computing*, 2016.

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

[4] J. O'Loughlin and L. Gillam, "Performance evaluation for cost-efficient public infrastructure cloud use," in *International Conference on Grid Economics and Business Models*. Springer, 2014, pp. 133–145.

[5] Y. Cheng, A. Anwar, and X. Duan, "Analyzing alibaba's co-located datacenter workloads," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 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 *Proceedings of the Tenth European Conference on Computer Systems*. ACM, 2015, p. 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," *Software: Practice and 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 *Globecom Workshops (GC Wkshps), 2014*. IEEE, 2014, pp. 122–126.

[9] J. O'Loughlin, "A workload-specific performance brokerage for infrastructure clouds." Ph.D. dissertation, University of Surrey, 2018.

[10] 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 *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 2017, pp. 153–167.

[11] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

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

[13] C. Jo, Y. Cho, and B. Egger, "A machine learning approach to live migration modeling," in *Proceedings of the 2017 Symposium on Cloud Computing*. ACM, 2017, pp. 351–364.

[14] A. Tumanov, A. Jiang, J. W. Park, M. A. Kozuch, and G. R. Ganger, "Jamaisvu: Robust scheduling with auto-estimated job runtimes," Technical Report CMU-PDL-16-104. Carnegie Mellon University, Tech. Rep., 2016.

[15] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format+ schema," *Google Inc., Mountain View, CA, USA, Technical Report*, 2011.

[16] 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 *Proceedings of the Third ACM Symposium on Cloud Computing*. ACM, 2012, p. 7.

[17] 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 Transactions on Cloud Computing*, 2019.

[18] W. Smith, I. Foster, and V. Taylor, "Predicting application run times with historical information," *Journal of Parallel and Distributed Computing*, vol. 64, no. 9, pp. 1007–1016, 2004.

[19] N. Rameshan, Y. Liu, L. Navarro, and V. Vlassov, "Augmenting elasticity controllers for improved accuracy," in *2016 IEEE International Conference on Autonomic Computing (ICAC)*. IEEE, 2016, pp. 117–126.

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

[21] J. O'Loughlin and L. Gillam, "A performance brokerage for heterogeneous clouds," *Future Generation Computer Systems*, vol. 87, pp. 831–845, 2018.

[22] P. Leitner and J. Cito, "Patterns in the chaos—a study of performance variation and predictability in public iaas clouds," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 3, p. 15, 2016.

[23] 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 Generation Computer Systems*, vol. 27, no. 8, pp. 1027–1034, 2011.

[24] I. Mavridis and H. Karatza, "Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing," *Future Generation Computer Systems*, vol. 94, pp. 674–696, 2019.

[25] 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 Transactions on Computers*, vol. 63, no. 12, pp. 3012–3025, 2014.

[26] 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," *Proceedings of the IEEE*, vol. 102, no. 1, pp. 11–31, 2014.

[27] X. Li, P. Garraghan, X. Jiang, Z. Wu, and J. Xu, "Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1317–1331, 2017.

[28] A. A. Khan, M. Zakarya, and R. Khan, "H^2 – a hybrid heterogeneity aware resource orchestrator for cloud platforms," *IEEE Systems Journal*, vol. 13, no. 4, pp. 3873–3876, 2019.

[29] P. Svärd, "Dynamic cloud resource management: Scheduling, migration and server disaggregation," Ph.D. dissertation, Umeå universitet, 2014.

[30] 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 Transactions on Parallel and Distributed Systems*, vol. 30, no. 1, pp. 204–217, 2019.

[31] A. Tchana, N. De Palma, I. Safieddine, and D. Hagimont, "Software consolidation as an efficient energy and cost saving solution," *Future Generation Computer Systems*, vol. 58, pp. 1–12, 2016.

[32] Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, "Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters," *IEEE Trans. Services Computing*, vol. 12, no. 4, pp. 550–563, 2019.

[33] H.-P. Jiang and W.-M. Chen, "Self-adaptive resource allocation for energy-aware virtual machine placement in dynamic computing cloud," *Journal of Network and Computer Applications*, vol. 120, pp. 119–129, 2018.

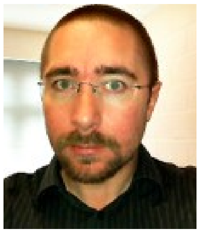
Hashim Ali is an Assistant Professor at the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. He is currently a PhD student in the same department. He completed his M.Phil (MS) in Computer Science from COMSATS Institute of Information Technology (CIIT), Abbotabad, Pakistan. His research interests include cloud computing, software testing, Agile processes, energy and performance efficient distributed systems and enterprise systems. He is proficient in computer systems both theoretically and practically. He has teaching experience of various courses in software engineering.

Muhammad Zakarya received the PhD degree in Computer Science from the University of Surrey, Guildford, U.K. He is currently a Lecturer with the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. His research interests include cloud computing, mobile edge clouds, Internet of Things (IoT), performance, energy efficiency, algorithms, and resource management. He has deep understanding of the theoretical computer science and data analysis. Furthermore, he also owns deep understanding of various statistical techniques which are, largely, used in applied research. His research has appeared in several international conferences, journals and transactions of repute.

Izaz Rahman received the PhD degree in computer science from the Department of Electronic and Computer Engineering, Brunel University, London, UK. He is currently an Assistant Professor with the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. His research interest includes power systems, optimization algorithms such as particle swarms, fuzzy logic, distributed systems, machine learning and artificial intelligence. He is teaching AI and fuzzy systems to graduate and undergraduate level students.

Ayaz Ali Khan is currently a PhD student in the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. He completed his M.Phil (MS) in computer science from COMSATS Institute of Information Technology (CIIT), Islamabad, Pakistan. His area of research includes energy-aware and performance-efficient scheduling, resource allocation, placement and management, at datacenter level. Moreover, he has enough knowledge of distributed systems, optimisation, game theory and computer programming. His work has appeared in several conferences, journals and transactions of repute.

Omer Rana is a Professor of Performance Engineering in School of Computer Science & Informatics at Cardiff University and Deputy Director of the Welsh e-Science Centre. He holds a PhD from Imperial College. His research interests extend to three main areas within computer science: problem solving environments, high performance agent systems and novel algorithms for data analysis and management. Moreover, he leads the Complex Systems research group in the School of Computer Science and Informatics and is director of the “Internet of Things” (IoT) Lab, at Cardiff University, UK. He is also an Associate Editor for Transactions on Parallel and Distributed Systems (TPDS), IEEE.



Lee Gillam is a Senior Lecturer in the Department of Computer Science at the University of Surrey, UK and Chartered IT Professional Fellow of the British Computer Society (FBCS CITP) and a member of the EPSRC Peer Review College. He is the Founding Editor-in-Chief of Springer’s Open Access Journal of Cloud Computing Advances, Systems and Applications (JoCCASA), which published its first articles in April 2012, and which followed on from co-editorship of the first Springer book on Cloud Computing in 2010. He has been a co-author of two reports for EPSRC/JISC on Cloud Computing (Research Use Cases, and Costs), and has been teaching a bespoke module on Cloud Computing to Masters students since 2010.

Rajkumar Buyya is a Fellow of the IEEE, Professor of Computer Science and Software Engineering, Future Fellow of the Australian Research Council (ARC), and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, at the University of Melbourne, Australia. His research interests include cloud, grid, distributed, and parallel computing. Prof. Rajkumar Buyya has a PhD in Computer Science from Monash University, Australia. He is a well-known and most cited scientist in the field of cloud computing systems. He is the former lead editor of Transactions on Cloud Computing (TCC), IEEE.