Muhammad Zakarya^{1,2} and Lee Gillam²

¹Abdul Wali Khan University, Mardan, Pakistan
²Department of Computer Science, University of Surrey, UK mohd.zakarya@awkum.edu.pk,l.gillam@surrey.ac.uk

Abstract. Simulations are often used to evaluate the performance of various scheduling and migration techniques in the context of large computing systems such as clouds and datacenters. To ensure that simulations match the real platform as close as possible, plausible assumptions and accurate statistical models are used in designing simulations; and that could also offer accurate results. However, it is not always possible that similar numerical results would also be achievable in a real cloud test-bed. The reason is that a simulator only abstracts a model and, hence, a system; but does not always reflect the real world scenarios. Therefore, the solution of any research problem using numerical simulation (experimentation) is not just to find a result, but also to ensure the quality and accuracy of the estimated results. CloudSim is largely used in the cloud research community to evaluate the performance of various resource allocation and migration policies. However, resources such as CPU, memory and application heterogeneities are not modelled yet. Moreover, its accuracy is rarely addressed. In this paper, we: (i) describe an extension to CloudSim that offers support for resource (CPU) and application heterogeneities; and (ii) demonstrate several techniques that could be used to measure the accuracy of results obtained in simulations, particularly, in the extended CloudSim. Based on our evaluation, we suggest that the accuracy and precision of the extended version of the CloudSim simulator may be as high as $\sim 98.63\%$ for certain energy and performance efficient resource allocation and consolidation with migration policies in heterogeneous datacenters.

Keywords: Simulations, performance modelling, accuracy, heterogeneous datacenters, resource management

1 Introduction

Quantifying the performance of various resource provisioning and consolidation policies, in a real cloud platform, for various workload models under transient conditions is challenging due to three reasons: (i) clouds show fluctuating demands, variable system sizes and heterogeneous hardware resources; (ii) cloud users have contending and heterogeneous quality of service (QoS) requirements; and (iii) cloud workloads have variable performance needs [1]. Furthermore, the use of real Infrastructure as a Service (IaaS) clouds to benchmark the workload performance under these variable conditions is constrained by real test-bed availability. Consequently, it is difficult to reproduce accurate results that could be trusted. In addition, it would be time-consuming and, hence, costly to re-configure benchmarking parameters across a large-scale IaaS cloud for multiple runs. Hence, it is not reasonable to conduct benchmarking experiments in a repeatable and scalable large-scale IaaS cloud.

A more feasible alternative is the use of simulations. Simulation tools make it possible to evaluate the research hypothesis (here benchmarking the workload) in a controlled platform which helps to reproduce the results easily. Simulations might also offer benefits by allowing to: (i) test services in a repeatable platform; and (ii) tune system bottlenecks before deploying on to real clouds. Furthermore, to develop and test adaptive application provisioning techniques, simulations must enable the evaluation of heterogeneous workloads and resources such as CPU, memory and networks. Note that CPU or platform heterogeneity is often seen as the major performance property of major cloud providers such as AWS EC2 and Microsoft Azure [2]. Moreover, significant efforts would be needed to choose plausible, reasonable, realistic assumptions and models to represent a real system/cloud. Note that due to the abstraction of the real system, it is not essential that the simulator must be as precise as a real world cloud platform in producing results. Therefore, it is very important to evaluate the precision and accuracy of the simulated results.

Resource management techniques in the context of cloud computing are largely evaluated using simulations. There are a number of cloud simulators suggested in the literature. These include but are not limited to DISSECT-CF [3], CloudSim [1], Green-Cloud [4] and DCSim [5]. Despite the popularity and number of citations, CloudSim neither offers support for modelling resource heterogeneity, neither validated and verified yet, although there are several models inside CloudSim which have been validated in the real world. For example, the linear power model and migration performance model are validated as accurate ($\sim 10\%$ performance degradation) [6]. However, the linear power model assumes that energy consumption exclusively depends on the level of CPU utilization only, and, therefore, ignores other components of the system such as network and memory. These components also consume significant amount of energy as demonstrated in [7]. Furthermore, CloudSim does not consider numerous important parameters in its VM migration model such as resource over-commitment and the dirtying rate of memory pages [8]. Similarly, there is no model to capture the overhead involved in the virtualisation, consolidation and migration technologies¹. Das et al. [9]have extended CloudSim with these models and have validated it through comparing with real world experiments with an approximate error of $\pm 2\%$. Most importantly, the current version of CloudSim cannot model resource and application heterogeneities, that certainly affect resource allocation and migration decisions.

In this paper, we: (i) model resource (CPU – platform) heterogeneity in CloudSim through offering extensions; and (ii) discuss the verification and validation of the extended CloudSim and its various statistical and mathematical models. In respect of (i), we consider, only, CPU or platform heterogeneity for various workloads (applications) which may cause variations in workload runtimes (performance), energy, and, therefore, costs. We are also aware that, besides platform (CPU architecture) heterogeneity which is often seen as a core performance property of public clouds [2], [10],other resources such as memory, networking have impacts on workload runtimes; however, they are not within the scope of this paper. We describe the modification being made to standard CloudSim in order to run the experiments presented in [11]. However, simulations may not always represent a real experimental set-up due to model abstraction, therefore, we demonstrate several statistical techniques to verify and validate the results produced in simulators, particularly, CloudSim [1]. Moreover, this paper extends our previously published results [12], [13] – as we measure their precision and accuracy. Several statistical techniques have been identified, and, implemented that could help the cloud research community to evaluate the accuracy of their simulated results with respect to a real cloud test-bed. We believe that, with the notable exception of [9], the accuracy and precision of the cloud simulator "CloudSim" is rarely addressed in the literature. Following are the major contributions of the work presented in this paper:

¹ http://www.cloudbus.org/cloudsim/

3

- several modifications and extensions such as VM migration energy consumption and performance, in terms of mathematical models, are presented to the wellknown cloud simulator "CloudSim";
- an approach to model resource (CPU platform or architecture) and workload (application) heterogeneities in CloudSim is presented;
- we, then, verify and validate these mathematical models using several statistical techniques; and
- we measure the accuracy of our previous published results [12], [13], which were produced in CloudSim, using real workload traces from Google production cluster [14].

The rest of the paper is structured as follows. In Sec. 2, we provide a background of our previous work as published in [12], [13], the obtained results and their accuracy. Furthermore, we model platform and application heterogeneities through offering extensions into the CloudSim simulator. We demonstrate the precision of CloudSim and, particularly, the obtained results in Sec. 3 and Sec. 4 using several statistical approaches. In Sec. 5, we validate our results using datasets from two real cloud platforms. Final comments and thoughts on the assessment are further elaborated in Sec. 6. An overview of the related work is presented in Sec. 7. Finally, Sec. 8 concludes the paper along with future research directions.

2 Background

This paper, basically, validates and verifies the results of our suggested scheduling and consolidation technique in [12], [13]. We proposed a VM allocation (FILLUP) and a consolidation with migration technique (CMCR) that migrate only long-running VMs to more energy efficient hosts – those that can recover their migration cost and, hence, continue to run for longer after migration to save energy. The readers are advised to read [11] for experimental set-up and results discussion. Following are the major findings demonstrated in [12]:

- 1. efficient VM allocation techniques might be at least 1.72% more energy efficient and economical when compared to consolidation with migration techniques.
- 2. it is approximately 3.52% more energy, hence more cost, efficient to migrate relatively long-running VMs. Long running VMs are those which run for two hours or longer.

The major reason to select a simulator instead of a real test-bed is that extensive and repeatable evaluation of a large system would require one to be readily available for such a purpose, as well as production of suitable code for this purpose. Sharkh et al. [15] state that CloudSim [1] is one of the most widely used simulation toolkits in cloud computing. Furthermore, CloudSim allows users to state and govern various resource allocation and provisioning policies, virtualisation, consolidation and energyaware resource management techniques. A detailed discussion of the CloudSim and other alternatives is presented in [11].

However, CloudSim does not provide a function to add or remove VMs while the simulation is running, which is necessary to the implementation of a real datacenter that runs interactive services. Normally, all the simulation entities are instantiated at the beginning and terminated at the end of the simulation. The main distribution of CloudSim expects the user to specify the submission times and service demands of all tasks (VMs) in advance, which is not possible for interactive services. Moreover, resources such as CPU, memory, disk, networks and workload (application) heterogeneities have not been modelled in CloudSim that could essentially affect resource allocation and migration decisions and, therefore, energy consumption, workload performance and users monetary costs. Therefore, to perform realistic simulations, along

with statistical models several modifications are needed in CloudSim, which are explained later in Sec. 2.1.

Furthermore, it is also important that a satisfactory amount of suitable data is available to build a simulation model and validate it accurately. The data should be from a reliable source. Usually, the lack or non-existence of suitable data could cause attempts, to verify and validate a model, failure. Therefore, we use real data from Google cluster [14] and PlanetLab² (CoMon project)³ to validate our proposed models. The PlanetLab dataset is briefly explained in Sec. 5.2 and further discussion can be found in [16]. A brief description of the Google's dataset is presented in Sec. 2.2; for detail discussion see [14].

2.1 Extended CloudSim

CloudSim, its layered architecture, and the features provided, are briefly explained in [1]. Fig. 1 shows the block diagram of CloudSim and its different components, offering a view of different capabilities for which CloudSim can be used such as scheduling, VM placement and consolidation techniques.



Fig. 1: CloudSim high level block diagram that shows a list of CloudSim features such as virtualisation, resource allocation and migration of VMs – users request VMs for their application (workload) and can see the desired output [1]

We found several limitations that need to be addressed before implementing our scheduling and migration techniques. For example, the available version of CloudSim does not support dynamic creation of VMs at runtime. To make it possible, we extend the *DatacenterBroker* class, with additional capabilities to read the Google data and monitor the submission time of each task to create VMs at runtime. If a task cannot be allocated to a VM, the *DatacenterBroker* puts the task in its waiting queue (W) to handle it later when enough resources are available. The *DatacenterBroker* also implements Alg. 1 to select a suitable VM type for each task and charge the user according to Google custom machine⁴ prices. We also extend the *PowerDatacenter*

 $^{^2}$ https://www.planet-lab.org/

 $^{^3}$ http://comon.cs.princeton.edu/

 $^{^{4}\ {\}rm https://cloud.google.com/custom-machine-types/}$



Fig. 2: Extended CloudSim class diagram [The classes shown in red, green or blue inside google and googlecluster packages are extensions to CloudSim default classes. In *util* package, we extend the **WorkloadModel** with **GoogleWorkload** to read the Google cluster data [14], the green classes in google package are the proposed allocation and migration policies while the green class in *models* package is the migration energy consumption model – the **VmLevelHostPowerModel** is implemented inside the *PowerModel* class – the blue *GooglePowerVm* class extends the *PowerVm* class to account for VMs past runtime and the blue *GooglePowerHost* class extends the *PowerHost* class to account for host reconfiguration costs]

σ

class to implement our proposed migration policies. An existing problem in this class i.e. several VMs were suddenly disappeared when they were in migration process was resolved. In essence, there was no check on the VM status during its migration and some VMs finish execution during the migration process.

Another issue was the heap memory, particularly, when dealing with large-scale simulations that run for longer periods. As discussed, CloudSim deals with each entity as an object, and it is important to clear the corresponding memory states and references when some objects are destroyed or not referenced. We noted that the Java garbage collector was also unable to free space, as the objects were referenced even when they have been destroyed. To resolve this issue, we used two techniques: (i) modified the *DatacenterBroker* to explicitly destroy VM objects when they finish their execution; and (ii) force the Java garbage collector to periodically clear the memory.

A slight modification to ensure that an idle host consumes idle power was made to the *PowerHost* class. To implement realistic simulations (which resemble closely to a real test-bed), we extend the *Host* class for host reconfiguration costs. These costs matter when switching on/off the resources (hosts) using dynamic capacity planning (DCP). To create a history of VM past runtime, the Vm class is extended. The extended *PowerVmList* then ensures that VMs selected for migration are in descending order of their past runtime (R_{past}). All VM allocation policies were implemented by extending the *VmAllocationPolicy* class. However, there is no model implemented for the energy consumption of VM migration in CloudSim. We added a *MigrationPowerModel* class to CloudSim, which can be extended to any power model of the migration process. Initially, the migration power model demonstrated in [17] was added to perform our simulations. We also added a new class to CloudSim, i.e. *VmLevelHostPowerModel*, to estimate the energy consumption of VMs on a virtualised host. We extend this class with the VM power model as explained in [12] and which is also demonstrated on a real cloud test-bed in [18], [19].

CloudSim does not model the resource heterogeneity and performance variations due to CPU models and/or workload contention. However, the performance degradation due to migration is already taken into account [20]. To model resource heterogeneity and performance variations, we use several performance benchmarks from Amazon EC2, relate them to Google workload and feed them into CloudSim. We extend the *host* class with performance parameters (such as mean, standard deviation) to model variations in workload runtimes – as described in Sec. 2.3.

Fig. 2 shows the UML (Unified Modelling Language) class diagram of several extended classes of CloudSim. The classes in *google* package extend the CloudSim functionality for VM allocation and migration. Similarly, the classes defined in *googlecluster* package read Google data and create VMs according to their arrival rate (time). Readers who are unaware of the basics of CloudSim, should read [1] first to appreciate how these classes relate to each other and what they are supposed to do.

2.2 Google Cluster Dataset

The extended *WorkloadFileReader* class reads every task and requests the broker to create a VM using Alg. 1. We assume each Google cluster task is a VM, and extract its characteristics (required resources, duration etc.) from the trace [14]. The required resources (CPU, memory), submission (request) time and runtime of every task are known prior to VM creation. It is also possible that a user's task has a deadline, and it might be useful to know before launching a VM whether the VM would be able to finish its execution within time (deadline) or not. In that case, every task request also goes through Alg. 1, and a suitable, cheap (low price) instance type (VM) is selected from the available pool of instance (VM) classes.

The Google cluster dataset is briefly explained in [14]. We wrote a Python script in

7

Algorithm 1: Instance type selection algorithm [13]				
Input: Instance request (R) , available instance types (INSTANCES)				
Output: Return a suitable and cheap instance $(Instance_{suitable})$				
1 $Instance_{suitable} \leftarrow INSTANCES_{maxCores}$ [assume the larger instance as suitable];				
2 price \leftarrow Price.INSTANCES _{maxCores} [on-demand price per hour];				
3 minPrice \leftarrow Price.INSTANCES _{minCores} [on-demand price per hour];				
4 for VM in Instances do				
5 if VM can finish R within time (is suitable) then				
6 if $VM_{price} < price$ then				
7 Instance _{suitable} $\leftarrow VM;$				
8 price \leftarrow Price.Instance _{suitable} [on-demand price per hour];				
9 end if				
10 end if				
11 if $Price.Instance_{suitable} = minPrice$ then				
12 break for loop;				
13 end if				
14 end for				
15 return Instance _{suitable}				

order to read task durations and start times from the dataset (available at $GitHub)^5$. Using task runtime details from Google data, we wrote the dataset as a text file (.txt) which contains the fields as shown in Table 1. Note that the CPU needed for a VM is converted into the notion of MIPS (Millions of Instruction Per Second) i.e. instruction count to create consistency with CloudSim. For a particular application, MIPS is calculated as the instruction count divided by its runtime multiplied by 10^6 [10]. If we assume a CPU frequency of 1 GHz as equivalent of executing 10^3 millions instructions per second (MIPS) – because 1 $\text{GHz} = 10^3 \times 10^6 \text{ Hz} = 10^3 \text{ millions Hz}$ and 1 Hz corresponds to one clock cycle which is essential to complete a single instruction; then a particular task running on this CPU for a specific amount of time can be described as $[time(seconds) \times 10^3]$ number of MIs (million instructions). For example, an hour long task/workload on this particular CPU can be described as $3,600 \times 10^3$ million instructions where 3,600 refers to the number of seconds in one hour. In other words, a workload which consists of $3,600 \times 10^3$ million instructions (MIs) would take exactly 1 hour on a 1.0 GHz CPU, 0.5 hour on a 2.0 GHz CPU, and so on, to complete its execution.

Note that the above assumption is the implementational simplification of the CloudSim instruction execution methodology, which essentially means that every CPU instruction requires exactly one clock cycle to execute [1]. However, this is not realistic with respect to modern pipelined CPUs; because: (i) a particular instruction may take several clock cycles to execute [CPI – cycles per instruction is greater than one]; and (ii) multiple instructions might be executed in a single clock cycle [IPC – instructions per cycle is greater than one or CPI is lesser than one]. To simplify concerns, we assume that each instruction requires exactly one clock cycle to complete its execution. Moreover, the classical RISC (reduced instruction set computer) five stage pipeline also has CPI = 1 i.e. the best attainable value with a single-execution-unit processor⁶; which executes exactly one instruction in one clock cycle. In general, good performance metrics relate to application performance and not machine characteristics such as clock cycles, as demonstrated in [10]. We believe that due to these reasons it is reasonable to assume CPI = 1; however, in our future work, we would account for

 $^{^{5}}$ https://github.com/google/cluster-data

 $^{^{6}}$ https://en.wikipedia.org/wiki/Classic_RISC_pipeline

these plausible assumptions. The WorkloadFileReader class reads the file and sends an event for VM creation at runtime (start time). When a VM is created, the broker class sets its termination time (finish time) and sends a VmDestroy event when the time is reached. Note that reading large files containing millions of entries can create problems with the JVM memory (heap) as discussed in Sec. 2.1. We also use another dataset from the PlanetLab (data related to CoMon project) [21] to validate our proposed algorithms in [12]. The PlanetLab data consists of CPU utilization values for more than a thousand of VMs with an interval of five minutes as explained later in Sec. 5.2.

Field name	Type	Description
Start time CPU Memory Network	(int) (int) (int) (int)	start time of the VM in seconds total number of MIPS that the VM needs to execute memory needed for the VM in MB network resources needed for the VM in MB
Finish time	(int)	finish time of the VM in seconds

Table 1: Fields in the dataset [CPU requirements are in MIPS to make it consistent with CloudSim – for example if a VM, which has 1 GHz CPU, runs for one hour then it needs $60 \times 60 \times 10^3 = 3,600 \times 10^3$ million instructions (MIs) [1] – a workload of 10^3 million instructions would take 1 seconds to complete its execution on 1 GHz CPU while 0.5 seconds on a 2 GHz CPU]

2.3 Modelling Resource and Application Heterogeneities

In this section we describe how performance of VMs (running different kinds of application) varies due to different CPU models (architectures) that accommodate these VMs. O'Loughlin et al. [22] performed several experiments on Amazon EC2 cloud and suggested that similar VMs perform differently on similar hosts; and the performance variations for a CPU model/architecture can be modelled as multi-modal log-normal distribution – where multi-modality relates to the number of CPU architectures based on the total number of visible peaks. For example, Fig. 3 demonstrates the performance of AWS m1.small instance type on different CPU models (architectures). Such kind of heterogeneities would certainly affect the workload performance and, therefore, infrastructure energy efficiency. Unfortunately, CloudSim does not offer any model to represent this.

In order to model platform and application heterogeneities within the Google dataset, we can assume the priority of each task as the type of application. For example, all tasks of priority 0 belong to a particular application, say PRIORITY0; and there are 12 priorities in total. As the trace providers point out that each task priority affects billing [14], thus we believe that it will accurately reflect the workload type. However, heterogeneity of the CPU resources is not available in the dataset. Therefore, the runtimes of tasks that belong to a particular workload can be assumed as the performance variations and the number of peaks as the CPU models. As explained in [12], we use Monte Carlo simulations of the benchmarks, as demonstrated in [22], to generate a larger dataset in order to relate it to the Google dataset and extract the hosts and applications heterogeneities and performance parameters in terms of minimum (min), maximum (max), mean (μ) and standard deviation (σ). To do this, we put both datasets i.e. real benchmarks and Google data on the same scale, as



Fig. 3: Performance variations for POVRAY and BZIP2 benchmark workloads on m1.small instances [E5430 performs 'best' for BZIP2 but 'worst' for POVRAY] [22]

shown in Fig. 4 and Fig. 5. Through visual inspection, we identify similar histograms in the multi-modal distributions. Moreover, we choose the performance parameters, through mapping, in such a way that the differences between them are the least. From an experimental point of view, when a particular application is being migrated from one host to another host, the impact of host heterogeneity is added as an increase or a decrease in application's runtime. Further details on these parameters and the mapping mechanism to model resource and application heterogeneities can be found in [11], [23].



Fig. 4: Priority 0 workload and execution time in (seconds) for host types A, B & C [log-normal distribution of runtimes – right, Google data vs. real performance benchmarks – middle, identify platforms – left] – PRIORITY0 workload is similar to POVRAY benchmarks on E5430, E5-2650 and E5645; where, E5430 performs better than E5645 [11]

We extend the *PowerHost* class of the CloudSim simulator to account for platform heterogeneities. The extended class offer a way to either: (i) increase or decrease the workload runtimes (using the z-score normalisation technique); and (ii) reduce the number of MIPS (execution) in the range of variations. CPUs of the same model with different speeds would be expected to need to execute the same number of instructions in order to complete a given workload. Conversion from CPU speed to MIPS, as described in Sec. 2.2, retains the relative speed differential for the CPU model. However, variations between CPU models do have to be accounted for, and this is



Fig. 5: Priority 2 workload and execution time in (seconds) for host types A, B & C [log-normal distribution of runtimes – right, Google data vs. real performance benchmarks – middle, identify platforms – left] – PRIORITY2 workload is similar to NAMD benchmarks on E5-2651, E5-2650, E5645, E5430 and E5507; where, E5430 performs worse than E5645 [11]

readily achievable by associating to workload runtimes and determining the MIPS rate that leads to this. Moreover, the *GoogleWorkloadFileReader* class uses priority of each task to put it with the right workload type. With the above modification to CloudSim, we evaluated the proposed resource allocation and migration algorithms for more than a million tasks (VMs) that belongs to various applications and 12,583 heterogeneous hosts – that concluded the two major findings, as, initially, described in Sec. 2.

The results were obtained on several plausible and realistic assumptions in a simulated environment using the well-known cloud simulator "CloudSim". However, it is not always possible that same or similar results and savings are also achievable in a real cloud platform or test-bed. The reason is that a simulator only abstracts a model but does not always reflect the real world scenarios. Therefore, the solution of any research problem through numerical simulation (experimentation) is not only to find a result, but also to guarantee the quality, precision and accuracy of the results [24]. There are several questions that come to mind regarding the results produced in a numerical simulator. For example:

- 1. can we assume that the obtained results are accurate regarding the system behaviour?
- 2. how can we enumerate the similarity and, therefore, dissimilarity between simulation and reality?

To answer the above two questions, it is essential to establish verification and validation⁷ criteria that "allow objective quantification of the difference between the results and the reality" [24]. Thacker et al. [25] describe that numerical simulations never completely match "real" experimental results due to the model abstraction. Therefore, the best option available is to make sure that the numerical solutions would be a good approximation of the reality. One approach that is widely used for this purpose is verification and validation, which collects evidence of a model's accuracy for a particular scenario. Nevertheless, this approach cannot prove that a particular model is accurate for all possible scenarios, but, rather, it can provide evidence that a model is sufficiently accurate.

 $[\]label{eq:constraint} 7 \ {\rm https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/other-se-lifecycle-building-building-blocks/other-se-lifecycle-buildin$

3 The Verification and Validation Approach

Usually, two different approaches are most commonly used to judge how good/bad a model is with respect to the real system: (i) verification; and (ii) validation. Verification ascertains that the model implements correct assumptions. Validation makes sure that the assumptions are reasonable with respect to the real system. The authors, of the manuscript in footnote $[^8]$, have described different methods including anti-bugging, simplified models, and tracing, that can be used to verify simulation models. Furthermore, Jauregui et al. [24] have discussed different kinds of validation that can be performed using: (i) other numerical solutions; (ii) analytical solutions; (iii) experimental results; (iv) intermediate results; and (v) convergence. Once a kind of validation is chosen, the next problem is to compare the obtained results with the pattern (established models) quantitatively. In numerous research fields including computer science, modest visual/photographic inspection is, largely, used as a method of validation. However, this method forces the necessity to examine reliable statistical techniques (or computational methods) to equate the differences (statistical significance) among the numerical and real results; and measure their accuracy and quality. Several other validation methods like: (a) correlation; and (b) reliability factor are also discussed in [24].

All verifications and validations are undertaken through a comparison of a reference/pattern model (i.e. the ground truth) with the simulated model under observation. The most common technique is the use of comparative analysis, performed in three different ways: (i) comparing simple cases of a simulation model to known results of analytical models (simplified); (ii) comparing the simulation model to other simulation models that have been already verified and validated; and/or (iii) experimental results (theoretical analysis). This paper provides an analysis of CloudSim (as the approaches in [12] are evaluated through this simulator) to make sure that it produces correct results. In respect to (i), the paper starts with associating reallife measurements (small-scale) to results obtained from simulations using plausible assumptions [Sec. 4.1]. During the comparisons, the precision and accuracy of the numerical results and, therefore, simulation is analysed. In respect to (ii), the paper focuses on a comparative performance and accuracy study of CloudSim with an existing simulator DISSECT-CF [3] [Sec. 4.2]. In respect to (iii), we repeatedly conduct the simulations using the same experimental set-up but with differences in: (a) VM arrival and termination rate, (b) submission time, and (c) workload datasets [Sec. 5.1]. Next we use different statistical approaches to approximate our results to a real test-bed. Note that both Sec. 4.1 and Sec. 4.2 fall under the category of model verification, while Sec. 5.1 depicts the approach towards model validation.

4 Model Verification

Verification is intended to assure that the model does what it is designed to do. There are several other cloud simulators that have been already verified (as discussed in Sec. 6) [5], [3]; however, CloudSim is not. Therefore, we use two kinds of approaches to make sure that the simulated system behaviour inside CloudSim is like a real system and is expected to produce accurate and precise results: (a) simplified model – comparing the simulated model to analytical models [Sec. 4.1]; and (b) simulated models – comparing the simulated model with other simulated but verified and validated models [Sec. 4.2].

 $^{^{8}}$ http://www.inf.ed.ac.uk/teaching/courses/ms/notes/note14.pdf

4.1 Simulation Models vs Analytical Models

The easiest (perhaps less efficient) way to verify our results is to compare them with analytical results which are obtained through a real cloud platform or test-bed. However, there is no guarantee that a similar approach is always available, so this might not be easy to do. Furthermore, this technique can be used, only, in simple scenarios, since finding analytical solutions for large-scale real problems could be either impossible or, at least, difficult, and most probably, this could be the reason for using numerical simulations instead of a real test-bed [1].

Fortunately, a part of our migration approach, the VM level host efficiency model is already taken into account by several researchers. For example, Alzamil et al. [18] proposed a similar model and produced their results at Leeds cloud test-bed at University of Leeds, UK⁹. Recently, the authors extended their own model in [18] to address heterogeneous workloads [19]. Since we use the VM level host efficiency model as a baseline for our VM placement and migration policies [12]. Therefore, if we compare our baseline to the results demonstrated in [18] and [19], we can come up with an approach to approximate our results and their accuracy to a real platform.

Real Test-bed A homogeneous cluster consisting of just four hosts is used in the real experiments in [18], [19]. Each host consists of a four core X3430 Intel Xeon CPU, running at the default clock speed of 2.40GHz, and 8GB of RAM. These hosts are connected via Gigabit Ethernet. Each host is attached to a watt meter in order to measure the energy consumed. These hosts are also used in the Leeds test-bed, which comprises a cluster of Dell commodity servers as described in [18]. If we know the minimum (P_{idle}) and maximum (P_{max}) power consumption of the real hosts in the Leeds test-best, we can use a linear power model (to measure energy usage) which is already validated [refer Sec. 4.2]. Unfortunately, except for CPU, memory and disk, we do not know other details of the hosts. Therefore, for simulating the energy usage behaviour, we use the benchmark values from SPECpower¹⁰. We further assume that the VM workload is stochastic (random change) and is always utilizing the requested resources as described in [1].

The first experiment in [18] demonstrates that over-provisioning the number of vCPU's (from one to four) on a single VM does not have an impact on the host's overall energy consumption. There are two reasons: (i) if the VM has one CPU, then dividing that CPU into one or four vCPUs would still consume the same maximum energy [18]; and (ii) if the VM has assigned more than one CPU, but the VM is not utilizing the overprovisioned resources. In respect to (ii), if the VM is utilizing the over-provisioned resources then the energy usage and the VM runtime would vary. Despite this dynamic behaviour (real), the current version of CloudSim implements a static behaviour in over-provisioning situations; that would affect infrastructure energy efficiency. To move away from this static behaviour in CloudSim, simulation developers should offer a way (or statistical model) to dynamically change the host's processing capabilities (i.e. speed or the MIPS ratio) depending on the level of over-provisioning they observe in real platforms. However, this requires significant efforts due to lack or nonavailability of resource over-provisioning datasets. Similar to resource heterogeneity model [as described in Sec. 2.3], we can estimate various parameters through observing real platforms and, then, embed those in the *PowerHost* class of the CloudSim simulator.

In the second experiment, the authors demonstrated that increasing the number of VMs in a single host has an impact on the host's overall energy consumption that can be represented by a linear relationship. The host level energy consumption and the

 $^{^{9}\ {\}rm https://institutes.engineering.leeds.ac.uk/computing/research/distsys/facilities.shtml}$

 $^{^{10}\ \}mathrm{https://www.spec.org/power_ssj2008/results/res2010q2/power_ssj2008-20100420-00252.html}$



Fig. 6: Modelling host level energy consumption (4 VMs) – the experiments are performed in four stages (60 second intervals), scaling-up the VM from one vCPU to four vCPUs [18]

linear growth in energy usage w.r.t the number of VMs is shown in Fig. 6 [18], [19] and is given by Eq. 1:

$$y = 10.127x + 72.587 \ (idle \ power) \qquad [R^2 = 0.9996] \tag{1}$$

where x is the number of VMs running on host and y is the host power consumption. Using virtualised host energy consumption model in [12], the VM level energy usage for all four VMs is similar to host's overall energy consumption as shown in Fig. 6 and Fig. 7 [18]. We use a similar virtualised host energy model to the baseline for our proposed VM allocation and migration policies. Therefore, if the simulated results of this model have similarities to those results demonstrated in [18], then the model may be considered verified.

Simulated Model We created (simulated) a model of the above real test-bed. The host level energy consumption and the linear relationship between energy usage and number of VMs, are shown in Fig. 8. The exact idle power consumption of hosts (real test-bed) is unknown to us. According to the SPECpower benchmarks that we use in our experiments, the idle power of each host is 62 Watts per hour (Wh). If we consider the y intercept in the linear equation as the host idle power (62 Wh), then the slope ($m_1 = 10.267$) of simulated results is very near to the slope ($m_2 = 10.127$) of the linear equation in real experiments [Eq. 2]. These slopes represent a similar rate of change in energy consumption w.r.t the number of VMs on a single host. Unfortunately, we are not aware of the workload type inside the real test-bed. Therefore, we assume a stochastic workload (random behaviour in utilizing the VMs) in our simplified (simulated) model. The stochastic utilization model is already available in CloudSim class file "UtilizationModelStochastic()". We ran the experiment ten times and then compared the average values to the real test-bed results.

$$y = 10.267x + 62.0 \ (idle \ power) \qquad [R^2 = 0.9149] \tag{2}$$



Fig. 7: VM level energy consumption (4 VMs) – the experiments are performed in four stages (60 second intervals), scaling-up the VM from one vCPU to four vCPUs; the dips in the graph (left side) represent the transition of terminating the current stage and starting the next stage [18]

In Eq. 2, x is the number of VMs running on host and y is the host overall power consumption. As Fig. 8 shows, the simulated results (host level energy consumption w.r.t number of VMs) match the real-life measurements most closely (the relative error (RE) in terms of the difference in both slopes of the linear equations (m_1, m_2) is less than 1.37%). This is computed as:

$$RE = \pm \left[100 - \left(\frac{m_1}{m_2} \times 100\right) \right] \tag{3}$$

Therefore, compared to the real test-bed experiments, our simulations of the simplified model and results can be assumed ~ 98.63% (100 - 1.37) accurate. With this number we suggest that our approaches can save approximately $3.66\pm0.05\%$ more energy than no migration approach [11]. Note that ± 0.05 is the RE which is approximately 1.37% of 3.66 – computed as $\frac{1.37}{100} \times 3.66$. However, with this it is understood that a model which works for simple scenarios is not assured to work for complex scenarios; however, a model which does not work for simple scenarios will absolutely not work for complex ones.

4.2 Simulation Models vs Validated Simulation Models

We did not find any study that have validated the simulator under consideration i.e. CloudSim [1] and its models as a whole. However, Tighe et al. [5] developed and validated a cloud simulator "DCSim" that can be used for comparison with CloudSim (in terms of their models). Similarly, Kecskemeti et al. [3] presented "DISSECT-CF (DIScrete event baSed Energy Consumption simulaTor for Clouds and Federations) and analysed by firstly validating it against the behaviour of a small-scale infrastructure cloud at the University of Innsbruck, Austria. According to their evaluations, the system's simulated behaviour almost matches real-life experiments with negligible error (with respect to application runtime, larger-scale network transfers and energy consumption)". These simulators are briefly described in [11]. In this section



Fig. 8: Host level energy consumption for 1 host running 1 to 4 VMs [the workload inside VMs implements a stochastic utilization model]

we verify several models that we used in CloudSim by comparing them to similar models in the validated simulator "DISSECT-CF". These models include: (i) host's energy consumption (linear relationship with CPU utilization and benchmarks); (ii) migration performance degradation; (iii) migration energy usage; (iv) migration time and (v) resource (CPU or platform) heterogeneity.

Linear Power Model The energy model (linear relationship with CPU usage %, as given by Eq. 4) has been validated in [3], with relative error of 0.21% with a sample standard deviation of 0.4%. The resulting power readings of a real host and simulated model using DISSECT-CF simulator are presented in [3].

$$P = P_{idle} + (P_{max} - P_{idle}) \times u \tag{4}$$

where P_{idle} and P_{max} denote the power consumption of host when it is idle (0% utilised) and 100% utilised, respectively; and u is the utilisation level of the host. Further details of the experimental set-up and parameters are discussed in [3]. Note that for accurate energy readings, one must be aware of the CPU's energy characteristics as they might not follow a linear trend and thus the offered linear consumption model should be exchanged with a custom model of which we have full control (SPECpower benchmarks). The creation of custom models is described in [8]. A few example models are also available in Vincenzo's fork of the simulator¹¹. Teng et. al [26] also found a linear relationship between a DVFS enabled CPU and its power consumption as given in Eq. 5:

$$P_t = C_1 \cdot f^3 \times u_t + P_{idle} \tag{5}$$

where f is the CPU frequency and C_1 is the scale coefficient. The energy consumption e_t is defined as an integral of the power consumption function over a period (t) as given in Eq. 6:

$$e_t = \int_{t_s}^{t_f} P_t \, dt \tag{6}$$

 $^{^{11} \}rm \ https://github.com/vincenzo-uibk/dissect-cf/$

where t_s is the time when the server starts and t_f is the time when the server finishes its assigned workload (job). The total energy consumption E of a datacenter with mactive servers is measured as in Eq. 7.

$$E = \sum_{i=1}^{m} e_t \tag{7}$$

Similar to DCSim and DISSECT-CF, CloudSim also provides support for modelling the SPECpower¹² real benchmarks (power consumption) for a number of hosts. We also use the SPECpower⁶ benchmarks for hosts energy consumption in our experiments, which does not need to be validated.

Migration Performance Model The migration model that we used in our experiments i.e. *each migration costs 10% loss in performance (degradation)*, has been validated for web application (workload) on a cluster of 6 hosts running 5 VMs as demonstrated in [27]. Beloglazov et al. [20] also used a similar model to demonstrate performance degradation due to migrations in datacenters.

Migration Power Model The refined energy model that we used for estimation of migration cost has been validated on a real Xen virtualised platform in [28], where it is demonstrated to be 90% accurate. The results are shown in [28], which describes that the refined model error is as low as 10%.

Huang et al. [29] demonstrated that during a migration, the energy consumption of the target host is almost stable (low increase). However, the energy consumption of the source host goes down with increase in CPU utilization of the migrated VM. Another energy consumption model for live migration is presented in [30]. This model increases the accuracy of the model presented in [29] by $\sim 3.9\%$ (11.8% vs. 15.7% error) for the source host and by $\sim 7.9\%$ (5% vs 12.9% error) for the target host. The authors [30] have suggested an accuracy improvement of $\sim 24\%$ over the model in [29]. With these contradictory results, it is reasonable to assume the VM size (data copied) as a major factor to estimate the energy consumption of a VM migration, as suggested in [12].

Migration Time The length of a live migration depends on the total amount of memory used by the VM and available network bandwidth [16]. The model in Eq. 8 which is used to estimate the time needed to complete a migration is mathematically valid: (i) if the size (memory and disk) of the migrated VM is known; and (ii) if the VM is idle. Beloglazov et al. [16] describe that "this model is justified (provided evidence for) and mathematically valid since the images and data of VMs are stored on a shared storage accessible over the network, which is required to enable live migration; therefore, copying the VM's storage is not required".

$$Migration_{time} = \frac{VM_{memory}}{Network_{bandwidth}}$$
(8)

Liu et al. [28] also demonstrated a slightly different model of Eq. 8 and have validated it in a real cloud platform. The model is suggested to be over 90% accurate. However, in our simulations we assume that a given VM workload is homogeneous and does not change, therefore, we believe that the selected approach (Eq. 8) is mathematically valid and accurate.

¹² https://www.spec.org/power_ssj2008/

Resource Heterogeneity Model To evaluate the trade-off between energy and performance (hence cost), we considered the heterogeneity of resources (hosts), with respect to CPU/platform, in our simulations using CloudSim. To date, CloudSim (and even DCsim and DISSECT-CF simulators) does not have any model to represent performance and heterogeneity of CPU resources. Therefore, we made several assumptions through relating the Google data with real performance benchmarks as demonstrated in [22], [31] [as described in Sec. 2.3]. Using the relationship between Google data and real performance benchmarks, we were able to identify various performance parameters (such as minimum - min, maximum - max, mean - μ and standard deviation - σ that denote variations in runtime) of hosts for different kinds of workload that we take into account.

Albeit, there would be other efficient ways of mapping; but, we use simple statistical method. First we put both datasets (i.e. real benchmarks and Google data) on the same scale, then identify the number of peaks using visual inspection. Moreover, through comparing the means, minimums and maximums of each single distribution, in both datasets, this could offer a way of identifying various platforms for various applications. We believe that this mapping (Google data – real benchmarks [14], [22]) is undertaken accurately and it also describes the hosts' heterogeneity well. However, as described earlier, we simply assume the CPU or platform heterogeneity as it is often considered as a core performance property of major cloud service providers such as EC2 and Azure [2]; but we are aware that other aspects such as memory, disk and networks have also impact on workload performance. Further discussion on the mapping is given in [12]. Note that a generalized mathematical model is still needed that can be used inside simulators to represent real heterogeneities for cloud resources and workloads.

Based on the above discussion of individual models (that we use in CloudSim) and their correctness, it is reasonable to assume that each sub-model produces valid results and the integration of models (CloudSim) is verified to be accurate. Due to dependencies among the models, it is also possible that the simulation can still produce invalid results even if all its sub-models are validated as described in [¹³]. However, in Cloudsim the above models are independent and it is reasonable to assume that this will not happen.

5 Model Validation

In model validation, there are three aspects of a model that should be taken into account: (i) assumptions; (ii) inputs and distributions; and (iii) outputs and conclusions. Furthermore, there are three approaches for validation that can be applied to these different aspects of a particular model as appropriate. These approaches are: (a) real system measurements; (b) expert intuition; and (c) theoretical results/analysis. Out of these three, "comparison with a real system is the most efficient and reliable method of validating a simulation model" [24]. However, this is often infeasible because: (i) the real system does not exist; or (ii) the measurements might be either impossible, or too difficult and expensive to carry out. Therefore, we adopt the validation approach using theoretical results and their analysis - (c). This is the most popular technique due to the fact that numerical measurements show consistency of the simulated model with the reality. Nevertheless, it is mandatory to introduce a measuring instrument whenever we perform a measurement and consequently this directly or indirectly affects the system being measured/observed. Therefore, it is very important to have the highest similarity between simulation configurations and measurements [24]. There are several methods such as correlation, that can be used to find similarities and

 $^{13\} https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/other-se-lifecycle-building-blocks-articles/verification-and-validation-of-simulation-models$

dissimilarities among different datasets (measurements) [24]. The correlation among two or more variables is perfect if the output value of the correlation factor is closer to either 1 or -1 and gets worse if the output value approaches to 0. The sign (\pm) designates the nature of the correlation: (+1) shows a perfect positive linear relationship. A (+1) represents that the relationship between two or more variables has exactly the same behaviour: when one of them decreases, the other decreases as well. Similarly, a (-1) indicates that there is a perfect negative relationship (linear), one will decrease as the other increases and vice versa. To compare the obtained results, a better and more efficient statistical approach than correlation is Feature Selective Validation (FSV) which is most widely used in the literature because of its versatility and simplicity [24].

In Sec. 5.1, we discuss the accuracy of the results presented in [12] using datasets from Google. In Sec. 5.2, we use datasets from another real cloud (PlanetLab) to evaluate the validity and accuracy of our approaches.

5.1 Validation using Experimental Results (Google dataset)

In this section, we evaluate the proposed migration techniques on 29 different traces from Google's cluster dataset (each trace corresponds to a single day workload) [14] and discuss the similarities among the obtained results to get the validity of the proposed techniques. Unfortunately, we do not have any measurements (ground truth) from a real test-bed that we can use them to compare with our results. Therefore, the only option to validate our findings from all 29 datasets is to statistically prove that for each day's workload, there is a significant difference (statistical) between the results generated by the proposed allocation and migration policies and other scheduling techniques.

Significance Differences As described in [11], [13], for all 29 datasets, the proposed VM allocation policy (FILLUP) combined with CMCR (consolidation technique) achieved higher energy efficiency compared to state-of-the-art allocation and migration policies. For all 29 datasets, the VM allocation policies (RR, R, BRS, MPD, FF, FILLUP) combined with different migration approaches (no migration [NO], migrate all [ALL], CMCR) that achieved higher energy efficiency are marked (**bold face**) in Table 2. For all datasets, the results demonstrate that the FILLUP approach combined with CMCR technique achieves higher energy efficiency than other approaches. Similarly, CMCR always achieves higher energy efficiency than NO and ALL which indicates that our migration approach could be more energy and hence cost efficient to datacenter service providers. Note that the CMCR approach minimizes the energy consumption by migrating VMs running for longer [as explained in [12], [13]]. In order to determine that there is a statistical difference (significant) between the results produced from our approaches and others, we use two statistical tests: (i) the standard error; and (ii) analysis of variance.

Standard Error If the sampling distribution is normally distributed, the sample average, its standard error (SE), and the quantiles (0.975) of the normal distribution can be used to calculate Confidence Intervals (CI) for the average. The statistics for our experiments are shown in Table 3. The SE is estimated by the standard deviation (σ) of the sample divided by the square root of the sample size (n) [Eq. 9]. Note that we use standard errors to study the significance of the differences between the means.

$$SE = \frac{\sigma}{\sqrt{n}} \tag{9}$$

Zakarya M., Gillam L.

DAY	(RR			R			BRS			MPD			FF			FILLUF	,
	NO	ALL	CMCR	NO	ALL	CMCR												
1	29.06	21.05	21.05	27.99	20.87	20.87	15.72	15.34	15.19	20.69	19.32	19.32	15.94	15.8	15.7	13.6	13.31	13.3
2	30.19	22.41	22.41	29.13	22.21	22.21	16.68	15.7	15.68	22.18	19.98	19.82	16.85	16.57	16.37	14.36	13.93	13.93
3	46.78	31.4	31.4	44.76	31.2	31.2	25.82	24.94	24.82	36.56	30.49	30.49	26.05	25.59	25.27	22.29	25.59	21.52
4	38.26	25.03	25.03	34.25	24.81	24.81	19.99	18.98	18.9	28.55	23.23	23.23	19.82	19.64	19.33	16.94	16.54	16.51
5	27.35	18.3	18.3	26.36	18.0	18.0	12.85	12.3	12.07	17.9	15.53	15.53	12.89	12.82	12.54	10.71	10.39	10.37
6	34.5	21.16	21.16	32.7	21.28	21.28	17.0	15.39	15.35	21.36	20.29	19.54	16.91	16.05	15.77	14.16	13.61	13.58
7	20.28	13.33	13.33	18.2	13.08	13.08	82.45	82.17	8.12	11.78	11.53	11.51	8.28	8.44	8.34	7.01	6.97	6.97
8	43.28	23.96	23.96	37.83	24.33	24.33	18.39	17.63	17.43	26.99	23.62	22.59	18.1	17.91	17.61	15.31	15.12	15.11
9	120.4	102.7	102.7	119.2	102.5	102.5	92.15	90.89	90.28	109.3	101.4	101.3	92.1	92.99	92.39	82.7	82.05	81.85
10	62.2	52.8	52.8	61.52	52.72	52.72	43.86	42.88	42.79	53.26	48.26	48.26	43.87	44.11	43.67	38.74	38.3	38.2
11	38.8	25.58	24.69	38.62	26.1	26.1	17.88	16.47	16.47	25.14	23.98	23.14	17.8	17.7	17.51	14.79	14.41	14.4
12	16.97	9.44	9.44	16.57	9.74	9.74	7.94	6.99	6.81	7.94	11.09	10.8	7.91	7.18	7.1	6.51	6.02	6.02
13	21.95	13.37	13.37	20.8	13.31	13.31	9.77	9.06	9.06	12.61	12.13	12.04	9.96	9.48	9.43	8.29	7.96	7.96
14	12.47	8.0	8.0	11.47	7.94	7.94	4.62	4.66	4.55	7.15	7.09	7.09	4.64	4.75	4.6	3.89	3.85	3.85
15	21.12	14.41	14.41	20.01	14.39	14.39	9.67	9.59	9.51	14.28	13.71	13.71	9.63	9.92	9.55	8.19	8.04	8.03
16	26.37	19.17	19.17	25.66	19.09	19.09	14.43	13.93	13.77	19.22	17.47	17.47	10.98	14.92	14.33	12.87	12.66	12.56
17	18.85	12.99	12.99	17.48	12.91	12.91	7.99	7.94	7.82	10.85	10.4	10.4	8.01	8.09	7.94	6.76	6.68	6.68
18	90.29	58.04	58.04	85.47	57.55	57.55	50.98	46.63	46.63	87.31	63.16	62.73	50.52	51.16	47.48	45.42	44.3	44.04
19	70.43	37.49	37.49	69.2	38.2	38.2	35.3	30.92	30.77	46.49	37.0	37.0	34.59	31.22	31.0	28.48	26.72	26.68
20	46.17	25.21	25.21	45.35	25.29	25.29	22.51	19.98	19.14	30.24	26.95	26.95	21.87	20.27	19.58	17.66	16.38	16.38
21	44.27	21.93	21.93	43.57	22.2	22.2	21.08	17.43	17.15	23.18	22.91	22.91	20.68	17.62	17.49	16.43	15.19	15.18
22	49.17	29.2	29.2	47.12	28.62	28.62	24.05	21.5	21.5	37.49	30.2	29.1	23.76	22.39	22.19	20.11	19.4	19.38
23	63.34	40.57	40.57	61.09	40.81	40.81	37.06	33.52	33.25	48.68	40.28	39.67	36.69	34.41	34.21	31.67	30	29.94
24	64.75	35.78	35.78	47.81	35.81	35.81	30.46	28.96	28.71	41.44	35.34	35.34	30.46	29.86	29.35	26.29	25.6	25.56
25	34.84	24.1	24.1	32.99	23.78	23.78	18.33	18.04	17.97	27.81	22.07	22.07	18.35	18.54	18.36	15.81	15.79	15.69
26	48.39	34.04	34.04	44.42	34.91	34.91	25.29	25.67	25.55	40.61	34.37	34.28	25.39	25.9	25.52	22.29	22.43	22.34
27	43.18	27.31	27.31	42.05	27.23	27.23	23.2	21.2	21.16	31.38	26.96	20.66	22.92	21.78	21.49	19.38	18.76	18.74
28	40.99	30.66	30.66	40.31	30.42	30.42	23.98	23.41	22.89	34.55	28.01	28.01	23.91	24.32	23.64	20.71	20.54	20.47
29	25.75	21.66	21.66	24.73	21.52	21.52	16.45	16.7	16.44	21.19	19.71	19.71	16.51	16.9	16.53	14.31	14.28	14.24

Table 2: Results in terms of total energy consumption (10^3 kWh) , for all 29 datasets using different allocation and migration policies [minimum values shown in boldface are better – CMCR migrates relatively long-running VMs which run for 2 hours or longer [as discussed in [12], [13]]]

[FILLUP+CMCR is more energy efficient than all other approaches to allocation and migration – for all 29 datasets FILLUP performs better than efficient approaches such as BRS, FF and MPD]

The problem with this technique arises when the $[mean \pm SE]$ of several groups overlaps. Such situations either shows the inaccuracy of the results, failure of the proposed model, or may be due to small differences. These differences might still be significant and can be identified by other statistical methods like t test, hypothesis test etc.

Analysis of Variance Another approach is hypothesis testing statistics which allows us to use statistical data analysis to make statistical inferences about whether or not the data we gathered support a particular hypothesis¹⁴. In other words, is there any significant difference between the results? To ascertain if the results in Table 2 are statistically significant is determined by calculating the probability of error (p value)by the t ratio. The difference between two datasets is statistically significant if it can not be explained by chance alone. The difference between two datasets is judged to be statistically significant when $p \leq 0.05$. At p = 0.05, the differences between the two datasets have only a 5% probability of occurring by chance [32]. We choose repeated measure ANOVA (Analysis of Variance) to show that there are significant differences between the results (using different allocation policies combined with different migration approaches) because: (i) the data is quantitative; (ii) each column represents a group or sample, so we have three samples for each independent experiment; and (iii) the purpose of the test is to compare the data. As our dependent variable (energy consumption) is dependent on two variables (independent) i.e. allocation policy and migration technique, so we use two-way ANOVA test. The ANOVA test only tells us that there are statistically significant differences among the groups, however, it does not really compare the groups with each other. Therefore, we use the t-test (post-hoc) repeatedly to show the significant differences among the groups [32].

The null hypothesis that we assume is that the mean (average value of the dependent

19

 $^{^{14}\ \}mathrm{http://www.statisticallysignificant consulting.com}$

Scheduling	Migration	Average Energy	Standard Deviation	Standard Error
approach approach		(MWh)	(σ) (Energy)	(SE) * 1.96
	NO	42.4276	22.9767	± 8.3627
RR	ALL	28.3134	18.074	± 6.5738
	CMCR	28.2828	18.0794	± 6.5802
	NO	40.2297	22.3156	± 8.122
R	ALL	28.3041	18.0516	± 6.5701
	CMCR	28.3041	18.0516	± 6.5701
	NO	25.7207	19.7506	± 7.1885
BRS	ALL	24.4421	19.4593	± 7.0825
	CMCR	21.7166	16.251	± 5.9148
	NO	31.5907	21.9187	± 7.9776
MPD	ALL	27.4648	18.3633	± 6.6836
	CMCR	27.2645	18.3397	± 6.675
	NO	22.9445	16.8353	± 6.1274
FF	ALL	22.6321	16.8539	± 6.1342
	CMCR	22.2169	16.5872	± 6.0371
FillUp	NO	19.851	15.1087	± 5.499
	ALL	19.4766	14.9682	± 5.4479
	CMCR	19.2924	14.8819	± 5.4165

20 Modelling Heterogeneities in Cloud Simulations and Quantifying their Accuracy

Table 3: Standard error with 95% confidence interval for different scheduling and migration algorithms [29 traces from Google's cluster – energy consumption, minimum values are better which are shown in boldface] – this suggests that the FILLUP combined with CMCR is more energy efficient than the other policies

[the allocation policies have stronger impact on energy consumption than the migration policies - FILLUP can save $\sim 53\%$ more energy than RR when no migrations are considered – CMCR can save $\sim 3\%$ energy when combined with FILLUP]

variable - energy consumption) is the same for all groups - combination of scheduling and migration policies. The alternative, or research hypothesis, is that the average is not the same for all groups. The ANOVA test shows that the differences among different allocation policies i.e. RR, R, BRS, MPD, FF and FILLUP are statistically significant having p = 7.37888E-07. Similarly, the p value for different migration approaches i.e. NO, ALL and CMCR is 0.00488 which confirms that the results are significantly different. Therefore, ANOVA rejects the null hypothesis and there are at least 2 groups different from each other. In order to determine which groups are different from which (among NO, ALL and CMCR), the t-test (post-hoc) details are shown in Table 4. Note that for efficient allocation, both migration techniques do not show any significant differences compared to NO approach. However, the p values shows a clear efficiency of CMCR over the ALL approach. Fig. 9 shows the box plot for NO, ALL and CMCR when combined with all allocation approaches.

In Table 5, we compare the FILLUP approach with all other allocation policies combined with NO, ALL and CMCR. As shown in Fig. 10, the t-test indicates that there are significant differences between FILLUP and RR, R, MPD when considered in combination with NO, ALL, CMCR. However, the behaviour of FILLUP is almost similar to BRS and FF and the test fails to show that there are significant differences between these approaches. However, the high probability (p) value is not evidence that the null hypothesis is true. It is even impossible to distinguish a null effect from a very small effect as demonstrated in the article at footnote [¹⁵].

To summarize this discussion, although we were unable to statistically prove that the proposed FILLUP and CMCR approach produces significantly different results

 $^{^{15}\ {\}rm http://onlinestatbook.com/2/logic_of_hypothesis_testing/nonsignificant.html}$

Allocation policy	MIGRATION POLICY	t Stat	$p(T \le t)$
			one-tail
	NO vs. ALL	2.555	0.00669
RR	NO vs. Cmcr	2.56	0.0066
	ALL vs. CMCR	0.006	0.49748
	NO vs. ALL	2.199	0.016
R	NO vs. Cmcr	2.199	0.01603
	ALL vs. CMCR	0	0.5
	NO vs. ALL	0.244	0.40405
BRS	NO vs. CMCR	0.828	0.20548
	ALL vs. CMCR	0.569	0.28586
	NO vs. ALL	0.764	0.22418
MPD	NO vs. Cmcr	0.801	0.21326
	ALL vs. CMCR	0.041	0.48378
	NO vs. ALL	0.069	0.47246
FF	NO vs. Cmcr	0.163	0.43559
	ALL vs. CMCR	0.093	0.46316
	NO vs. ALL	0.093	0.46305
Fillup	NO vs. Cmcr	0.139	0.44482
	ALL vs. CMCR	0.046	0.48167

Zakarya M., Gillam L. 21

Table 4: The t-test multiple comparisons for different allocation and migration policies [CMCR can be seen as more energy efficient that NO and ALL approaches w.r.t the p values shown in boldface] – the t-critical value is 1.673, [efficient allocation do not have an impact on consolidation policies]



Fig. 9: The t-test details for NO, ALL and CMCR migration policies ['*' represents the degree of difference and 'o' means there is no significance difference – however, the minimum, maximum and mean values show that CMCR is more energy efficient than NO and ALL]

compared to all other allocation and migration approaches. However, based on the minimum, maximum and mean results as shown in Fig. 9 and Fig. 10, we can see that FILLUP and CMCR are more energy efficient than all other allocation and migration policies. Lastly, there are different methods (like Fisher's method, weighted Z-method)

Allocation policy	MIGRATION POLICY	t Stat	$P(T \le t)$
			one-tail
	NO	4.344	2.967E-05
RR	ALL	1.993	0.0256
	CMCR	2.0316	0.02348
	NO	4.001	9.339E-05
R	ALL	1.992	0.02563
	CMCR	2.038	0.02313
	NO	1.249	0.10843
BRS	ALL	1.07	0.14455
	CMCR	0.582	0.28141
	NO	2.333	0.01162
MPD	ALL	1.784	0.0399
	CMCR	1.786	0.03975
	NO	0.724	0.23615
FF	ALL	0.741	0.23097
	Cmcr	0.694	0.24514

22 Modelling Heterogeneities in Cloud Simulations and Quantifying their Accuracy

Table 5: The t-test multiple comparisons of FILLUP approach with different allocation and migration policies [the FILLUP approach is much similar to BRS and FF, hence there are no significant differences] – approaches with p values in boldface are less efficient than FILLUP – the t-critical value is 1.673, [*BRS and FF are comparable to* FILLUP]



Fig. 10: Comparison of FILLUP to other approaches using t-test ['*' represents the degree of difference and 'o' means there is no significance difference – however, the minimum, maximum and mean values show that FILLUP is more energy efficient than RR, R, BRS, MPD and FF]

that can be used to combine the probabilities (p values) if the differences are not significant in multiple iterations. Based on non-significant findings, these methods can reach to an accurate decision about whether the differences are significant or not. This is illustrated with an example in the article at footnote [¹⁶].

 $^{16\ {\}rm http://onlinestatbook.com/2/logic_of_hypothesis_testing/nonsignificant.html}$

5.2 Validation using Experimental Results (PlanetLab dataset)

To validate our approaches we also use the data provided by a monitoring infrastructure for PlanetLab¹⁷, as part of the CoMon project [21]. The data consists of CPU utilization values taken at 5 minute intervals. The data is collected from more than a thousand VMs running inside hosts that are located at more than 500 locations around the world. We use the traces which span over 10 days and are by default available inside CloudSim [1]. These traces have been randomly selected from the workload traces collected during March and April 2011. Table 6 shows the dataset various characteristics for each day. By using this data, our intention is to compare the results of our proposed algorithms to other approaches that have been already published in [33].

Date	Number of	Util. Mean	Util. St. dev.	Quartile 1	Median	Quartile 3
	VMs	μ	σ	(%)	(%)	(%)
03/03/2011	1,052	12.31	17.09	2	6	15
06/03/2011	898	11.44	16.83	2	5	13
09/03/2011	1,061	10.70	15.57	2	4	13
22/03/2011	1,516	9.26	12.78	2	5	12
25/03/2011	1,078	10.56	14.14	2	6	14
03/04/2011	1,463	12.39	16.55	2	6	17
09/04/2011	1,358	11.12	15.09	2	6	15
11/04/2011	1,233	11.56	15.07	2	6	16
12/04/2011	1,054	11.54	15.15	2	6	16
20/04/2011	1,033	10.43	15.21	2	4	12

Table 6: Workload data characteristics (CPU utilization %) from CoMon project, PlanetLab [33]

To be consistent with the results in [33], we simulated a datacenter of 800 heterogeneous hosts comprised of two different types as shown in Table 7. For the sake of simulation, the frequency of the host's CPUs are mapped onto MIPS ratings with respect to a single core. Moreover, to simplify implementation, we assume that each host and, therefore, VM can execute exactly one instruction per clock cycle i.e. CPI =1, as initially described in Sec. 2.2. In addition, each host is configured to have 1 GB/s network bandwidth. The energy consumption of these hosts relates to the benchmark values from SPECpower. The characteristics of various VM types, as shown in Table 8, relate to well-known Amazon EC2 instance types with a single exemption that we assume all VMs as single core machines. This is described by the fact that the PlanetLab's workload data comes from single core VMs, as well [33]. Therefore, for each VM type its corresponding memory (RAM) is divided by the number of cores as described in Table 8.

Initially, all VMs are placed onto hosts according to: (i) the resource needs as defined by VM types; and (ii) VM allocation policy. However, using the workload data, if VMs utilize their provisioned resources less, this create opportunities for consolidation. Note that every VM is assigned a workload trace, randomly, from one of the VMs (PlanetLab dataset) that lasts for various execution times – as execution times of VMs matches tasks execution times in Google dataset. As demonstrated in [33], the Power Aware BFD heuristic (PABFD also known as MPD) is used as the default

 $^{^{17}}$ https://www.planet-lab.org/

Speed MIPS No of Memory Host Host $P_{idle} | P_{max} | \text{Amount}$ (GHz) Rating Cores Name Type (GB)(Wh) (Wh) HP ProLiant Intel Xeon ML110 G4 400 3040 1.861,860 $\mathbf{2}$ 4 86 117HP ProLiant Intel Xeon $\mathbf{2}$ ML110 G5 3075 2.662,660 4 93.7 135400

24 Modelling Heterogeneities in Cloud Simulations and Quantifying their Accuracy

Table 7: Host characteristics and number used in the experiments demonstrated in [33] – MIPS: Million of Instructions Per Second

Instance type	Speed (MHz)	Memory (GB)
High CPU medium	2,500	0.85
Extra large	2,000	3.75
Small	1,000	1.7
Micro	500	0.613

Table 8: Instance types and their characteristics [33], [34]

allocation policy in CloudSim. During consolidation, the optimization module determines the overloaded hosts using two techniques: (i) the static threshold policy (THR) which uses a static upper utilization threshold (80%) for each host; and (ii) the local regression dynamic threshold policy (LR) which uses dynamic (adaptive) thresholds instead of a static one. The underutilized hosts are those which have a minimum utilization level compared to all hosts. If there are several migratable VMs, then the migration policy uses MMT (minimum migration time) approach to prioritize those VMs that will take minimum time in migration. Beloglazov et al. [33] suggested that the MMT policy performs better than all other methods such as random, MCC, MU as presented in [33], and when combined with LR, it produces minimal results in terms of energy consumption.

We made a slight modification to the above experiments in order to account for the migration cost (both in terms of energy consumption and performance degradation). As, there are two instances running during the migration duration (one on source host and a second one on the target host); the energy cost relates to the consumed energy by the least energy efficient host [11]. Moreover, in our evaluation, we account for 10% performance degradation due to migration that is merely modelled based on the memory, disk size and available network bandwidth [1], [23]; however, complete network heterogeneity is not within the scope of this paper. Similarly, co-locating¹⁸ VMs when they compete for similar resources can also degrade VMs performance [31]. However, the performance degradation due to co-locating VMs is not considered in this paper.

We chose the MMT combined with THR and LR techniques as the benchmark policies for comparison because they perform better than all other techniques in [33]. We simulated the combination of FILLUP and CMCR (PR – past runtime) techniques and observed that our approaches outperform even the best combination of allocation and migration policies presented in [33]. The substitute for MMT policy is our own VM selection policy i.e. migrate VMs having maximum past runtime. The mean results (energy consumption and migrations) for all ten workload traces, different allocation and migration policies are shown in Table 9.

Fig. 11 and Fig. 12 show the results obtained for all ten workload traces. Using the THR and LR policies, our consolidation approach (FILLUP+CMCR) is \sim 51% more energy efficient than THR-MPD-MMT and \sim 53.4% more energy efficient than LR-

 $^{^{18}}$ several VMs that reside on same host sharing memory and other resources

	Policy	Avg. energy	Avg. number	
Host overload	Allocation	MIGRATION	(kWh)	of migrations
	MPD	MMT	246.069	26,779
THR	FillUp	MMT	192.483	37,100
	FillUp	CMCR	127.849	1,391
	MPD	MMT	233.638	28,389
LR	FillUp	MMT	170.816	31,202
	FillUp	CMCR	124.759	$1,\!340$

Table 9: Average energy consumption and number of migrations for ten different traces from PlanetLab [THR: uses an upper threshold (0.8) for host overload, MPD, default VM allocation policy in CloudSim, MMT: minimum migration time FU: FILLUP, PR: CMCR – 'best' approaches are shown in boldface]

MPD-MMT. If we assume that the results presented in [33] are correct, then certainly our results can be assumed valid and accurate. Note that in these experiments all VMs are running for 24 hours and the resource usage varies according to the PlanetLab workload. We expect that our approach would be more energy efficient if VMs are submitted and terminated dynamically at runtime which creates more migration opportunities.



Fig. 11: Energy consumption using different allocation and migration techniques for ten workload datasets from PlanetLab as described in Table 6 [MMT: minimum migration time FU: FILLUP, PR: CMCR]

Note that simulations does not represent the reality and the above results are not guaranteed to be accurate, even if simulators are verified and validated. Therefore, there is no guarantee that the same results, as presented in [12], [13], would also be achievable in a real cloud test-bed. Although, we have tried our best to verify and validate the findings, however, errors and inaccuracy might still be expected due to the abstraction of mathematical and statistical models until a real heterogeneous cloud test-bed is used for evaluation. Table 10 summarizes the findings and associated variables, extensively discussed in this paper. This could also fit the discussion in Sec. 6, neatly.



Fig. 12: Number of migrations using different allocation and migration techniques for ten workload datasets from PlanetLab as described in Table 6 [MMT: minimum migration time FU: FILLUP, PR: CMCR]

Criteria	Method	Findings
Verification	Simulation vs.	(a) CloudSim is approximately 98.63% accurate and pre-
	Analytical [Sec.	cise [regression analysis].
	4.1]	
		The expected error is approximately $\pm 1.37\%$.
	Simulation vs.	(a) The linear power model is 99.79% accurate.
	Validated	(b) VM migration results in 10% loss in performance.
	[Sec. 4.2]	(c) The migration power model in 90% accurate.
		(d) The migration time model is mathematically valid.
		(e) The resource heterogeneity model is based on real
		benchmarks and real workload datasets.
Validation	Experimental Re-	(a) Statistical techniques such as standard error (SE) and
	sults [Sec. 5.1 and	analysis of variance (ANOVA) were used to show that the
	Sec. 5.2]	obtained results have significant differences i.e. among
		various allocation and migration policies using the real
		workload dataset from Google.
		(b) A comparison of the proposed policies vs. the closest
		rivals, using another real workload dataset from Plan-
		etlab cloud, also demonstrates the energy savings and
		performance gains.
	Allocation and	The energy-performance efficient migration approach
	Consolidation	"FILLUP+CMCR", suggested in $[12]$ and $[13]$, can be up
		to 3.66% (±0.05 error) more energy efficient, and 1.87%
		$(\pm 0.025 \text{ error})$ more performance efficient, than a no mi-
		gration strategy.

Table 10: Summary of findings and associated variables or statistical methods

6 Discussion

A simulated model is more abstract than the real system it represents¹⁹. From a performance point of view, this abstraction allows us to focus on the system elements which are really important, that creates two major problems: (i) the model assumption we make could, possibly, eliminate details of the real system; and (ii) the abstraction process introduces inaccuracy due to (i). That said, some degree of inaccuracy may be necessary or even desirable, to make the model solution tractable. Without doubt, some assumptions are made in order to construct the model. However, assumptions should be realistic, plausible and reasonable with respect to the real systems. In other words a simulated system should exactly match a real system. Moreover, when using assumptions and simulations, we must put efforts into validation of the mathematical models. Similarly, if a numerical simulator has been used to produce results from a model, then the validity of the simulator also matters. If a particular simulator is verified and validated, then we are assured that the obtained results are accurate or at least approximated for a real test-bed as well.

CloudSim [1] is among the most popular and highly cited IaaS cloud simulators. The three major problems with CloudSim are: (i) there is no complete documentation to show the full working of its different classes; (ii) there is no statistical model to model and account for resource heterogeneities; and (iii) no study is available at the moment about its accuracy (to the best of our knowledge). Another cloud simulator "DISSECT-CF" is shown to be accurate and performs better than CloudSim [3]. The authors [3] demonstrated that the simulation offers sufficient accuracy even for complex CPUs and network set-ups. In case of networking, though, one has to keep in mind that the simulator only does rudimentary modelling of network traffic and it is not capable to model traffic with small sized transfers. However, this is by design as Kecskemeti et al. describe in [3]; the main aim for the networking of the simulator is to allow proper modelling of VM image transfers. The relative error of the DISSECT-CF simulator is shown to be less than 0.3% for CPU intensive tasks, and less than 5% for memory intensive tasks while network transfers of large files have a relative error of less than 0.5%. Most importantly, the accuracy of the modelled energy behaviour is shown to be in line with the accuracy of the real CPU model in use [SPECpower benchmarks]. Moreover, DISSECT-CF simulator also models the energy consumption of a VM migration as described in [30], [35]. Note that similar energy models and benchmarks, except the VM migration, are also used in CloudSim and several other cloud simulators such as DCSim [5]. Unfortunately, DISSECT-CF and DCSim do not offer any model for resource heterogeneities.

For all simulators, one can only achieve some level of accuracy by ensuring that the simulator's models are aligned with the actual hardware he/she is trying to model. For example, to model a real life host, one has to adjust the processing capability of resource consumptions and energy usage of the host (object) [3]. Moreover, degradation in performance, with respect to execution times, must be kept in view. For example, as discussed in [36], performance (execution times) of cloud workloads are strongly dependant on the CPU models (architectures). At the moment, similar performance models are not available neither in CloudSim, DISSECT-CF and DCSim. As a result, several researchers [11] have used performance benchmarks values to model performance of their workloads in simulations. This means that for accurate results, the required level of adjustment must be experimentally/empirically identified for actual hardware and software combination. Fortunately, this adjustment is not necessary for non-tricky hardware set-ups (such as non-HT enabled, SMP systems) running CPU intensive applications. Those researchers who are interested to know how to model

 $^{^{19}\ {\}rm http://www.inf.ed.ac.uk/teaching/courses/ms/notes/note14.pdf}$

software behaviour with processing limits (capabilities), may refer to [37] for more discussion and details.

7 Related Work

To perform larger-scale experiments and feasibility studies, cloud simulators could be used. A wide variety of simulators (both open source and commercial) is available with different features and characteristics. They are designed with one or more objectives; however, none of them can be used to simulate a whole cloud. Sharkh et al. [15], has presented a systematic review of several cloud simulators which are often used by the cloud research community.

Among these, CloudSim [1] is one of the most popularly/widely used and cited. By using CloudSim, industrial developers and cloud researchers can focus on various design issues of a particular system they want to investigate, deprived of being involved in any low level details of cloud infrastructures and services. Largely, CloudSim is used to evaluate resource scheduling, allocation and consolidation techniques. However, it provides little or no support for modelling datacenter networks. Moreover, Greencloud [4] is a classical packet-level simulator for energy-aware datacenters that, largely, focuses on datacenter networks and communications. Moreover, it offers an exhaustive fine-grained modelling facility for the energy consumption of various datacenter's IT equipment such as hosts, communication links and network switches. iCanCloud [38] is an event-driven simulation environment which is used to model and simulate infrastructure clouds. The key focus of iCanCloud is to foresee the existing trade-offs between monetary costs and performance of heterogeneous workloads run on particular hardware [11]. Moreover, iCanCloud also offers valuable information regarding resource provisioning costs.

DCSim [5] differs from GreenCloud in that it is focused on virtualised datacenters which provide IaaS platform to multiple tenants, similar to CloudSim. However, it differs from CloudSim in that it focuses, largely, on transactional and continuous workloads. As such, DCSim has the ability to model replicated VMs that share incoming workloads and, in addition to, dependencies among VMs that are part of a multi-tiered application. In addition, DCSim has a more layered and realistic cloud architecture (host inside a rack, racks inside a cluster and clusters inside a datacenter) in comparison to CloudSim. Moreover, DCSim provides support for inter-racks and inter-clusters VMs migration.

DISSECT-CF [3] is a compact, highly customizable cloud simulator (open source) that focuses on the internal organization and behaviour of IaaS clouds. This simulator provides more in depth energy estimation techniques both at host and VM level and is largely validated with real world experiments. The high level components (e.g., IaaS level VM and host schedulers) of the simulator were not validated with real life measurements but by comparing its results with two other simulators: CloudSim and GroudSim [39]. The experiments are demonstrated in [3], and the relative error of DISSECT-CF compared to other simulators was revealed to be as low as 0.29%. It is important to note that the offered high level schedulers are not necessarily following the implementation details of any scheduler of real life IaaS systems and hypervisors, although they are having similar behaviour, and are offered only to show simple example implementations. Moreover, Kecskemeti et al. [3] describe about the accuracy of DISSECT-CF simulator that "similar to the CPU models, for highly accurate results, we must suggest a custom VM scheduler that matches the one used by our modelled real life IaaS".

GreenCloud is an extension of the well-known network simulator "NS2" which is largely used to evaluate energy-aware cloud datacenters. The key characteristic of GreenCloud is its capability of exhaustive modelling of the communication networks in a datacenter. MDCSim [40] is a discrete event-driven simulator (commercial) that models particular hardware characteristics of various datacenter components such as hosts, communication links and network switches. Nunez et al. [38] proposed iCan-Cloud, which is a hypervisor-based simulator specifically with a focus on simulating instance types offered by Amazon EC2 cloud. The tools and simulators that could simulate a complete cloud stack comprise CloudSim [1] and DISSECT-CF [3]. Nevertheless, CloudSim offers either limited or no support for more realistic and complex application types such as the work-flows (communicating tasks), and has no capability of cross-layer interaction. Modelling the cross-layer interaction is very important, particularly, for intra-layer resource management techniques i.e. self-configuration or adaptation [41]. Moreover, DISSECT-CF permits access to internal cloud information (such as VMs and workloads) and, therefore, precisely models energy consumption of IaaS clouds at two levels: (i) hosts (coarse-grained); and (ii) VMs (fine-grained) – as described in [11].

Unfortunately, all of the above simulators do not offer a statistical model to capture the platform heterogeneities with respect to performance variations as demonstrated in [36]. Moreover, with the notable exception of [3], migration energy and performance costs models are also very rarely addressed.

8 Conclusions

To support and quicken the research associated to clouds, applications and services, it is very important that accurate simulators and software tools are designed and developed to aid the cloud research community and industrial developers. Simulationbased approaches to examine and evaluate cloud and its application behaviour also offer various important benefits, as they let cloud researchers to: (a) determine the performance of various resource provisioning and service delivery strategies in a governable environment; (b) conduct the experimental evaluation repeatedly and, largely, free of cost; and (c) tune the performance bottlenecks of various policies before they are deployed in real public or private clouds.

This paper provides: (a) an extension to the well-known cloud simulator "CloudSim" to account for resource heterogeneities; and (b) an analysis and evaluation of our previously published results [12], inside the extended simulator, to demonstrate their accuracy and preciseness [1]. We compared small-scale real measurements (obtained on University of Leeds test-bed) to the results obtained from simulations in CloudSim (simplified model) [18]. During the comparisons, accuracy/precision of the simulator (extended version) is analysed and is suggested to be approximately 98.63%. With this number we suggest that our proposed approaches [as described in [12], [13]] could save approximately $3.66\pm0.05\%$ more energy than no migration approach. We then focused on comparative performance and accuracy study of CloudSim with an existing validated simulators DISSECT-CF. This verifies various statistical/mathematical models used in CloudSim to confirm the precision of the simulations performed. We made this choice because:

- 1. the authors in [18], [19], have already proved their effectiveness in testing a part of our VM migration model, and
- 2. this allows us to check the accuracy of our VM migration model against their mathematical models.

Furthermore, we statistically validated our proposed allocation and migration policies with 29 different datasets (Google's cluster) and demonstrated that both our techniques i.e. allocation (FILLUP) and consolidation with migration cost recovery (CMCR) achieve higher energy efficiency compared to other policies. Additionally, we chose to validate our results using another dataset from the CoMon project inside the

PlanetLab cloud platform.

Note that, in this paper, the verification and validation is limited to only energy consumption and performance. The results are limited to a minimalistic linear power model of a single host that uses results of another research; that tests one of the baseline algorithms – VM level host efficiency model [12]. Moreover, the migration performance model is very rough and only considers 10% cost; that is merely modelled based on the memory, disk size and network bandwidth. Furthermore, "CloudSim" [1] is a generic tool for modelling and simulation of cloud computing infrastructures and services and its validation requires more in-depth analysis of all other aspects such as memory, networking, task scheduling, etc. Moreover, the accuracy claimed for CloudSim is obtained for a particular case and it is not shown that it is verified for other servers (CPUs). In future research, our aim would be to consider these aspects for modelling in cloud simulations; and, then, for versification, validation and accuracy.

Acknowledgement

This work is supported in part by the University of Surrey, UK, Abdul Wali Khan University, Mardan (AWKUM) and in part by the Abdul Wali Khan University Mardan (AWKUM), Pakistan. The research was conducted under the supervision of Dr. Lee Gillam at University of Surrey, UK.

References

- 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.
- 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.
- G. Kecskemeti, "Dissect-cf: a simulator to foster energy-aware scheduling in infrastructure clouds," Simulation Modelling Practice and Theory, vol. 58, pp. 188–218, 2015.
- D. Kliazovich, P. Bouvry, and S. U. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers," *The Journal of Supercomputing*, vol. 62, no. 3, pp. 1263–1283, 2012.
- M. Tighe, G. Keller, M. Bauer, and H. Lutfiyya, "Dcsim: A data centre simulation tool for evaluating dynamic virtualized resource management," in 2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualization management (svm). IEEE, 2012, pp. 385–392.
- A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya *et al.*, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in computers*, vol. 82, no. 2, pp. 47–111, 2011.
- W. L. Bircher and L. K. John, "Complete system power estimation using processor performance events," *IEEE Transactions on Computers*, vol. 61, no. 4, pp. 563–577, 2012.
- V. De Maio, G. Kecskemeti, and R. Prodan, "An improved model for live migration in data centre simulators," in *Proceedings of the 9th International Conference on Utility* and Cloud Computing. ACM, 2016, pp. 108–117.
- 9. K. Das, "Cloud computing simulation," Ph.D. dissertation, Indian Institute of Technology, Bombay Mumbai, 2011.
- J. O'Loughlin and L. Gillam, "Good performance metrics for cloud service brokers," in The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization. Citeseer, 2014, pp. 64–69.
- 11. M. Zakarya and L. Gillam, "Energy and performance aware resource management in heterogeneous cloud datacenters." Ph.D. dissertation, University of Surrey, 2017.

- —, "An energy aware cost recovery approach for virtual machine migration," in International Conference on the Economics of Grids, Clouds, Systems, and Services. Springer, 2016, pp. 175–190.
- 13. M. Zakarya, "An extended energy-aware cost recovery approach for virtual machine migration," *IEEE Systems Journal*, 2018.
- C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format+ schema," Google Inc., Mountain View, CA, USA, Technical Report, 2011.
- M. A. Sharkh, A. Kanso, A. Shami, and P. Öhlén, "Building a cloud on earth: A study of cloud computing data center simulators," *Computer Networks*, vol. 108, pp. 78–96, 2016.
- 16. A. Beloglazov, "Energy-efficient management of virtual machines in data centers for cloud computing," Ph.D. dissertation, 2013.
- Q. Huang, F. Gao, R. Wang, and Z. Qi, "Power consumption of virtual machine live migration in clouds," in *Proceedings - 2011 3rd International Conference on Communi*cations and Mobile Computing, CMC 2011, 2011, pp. 122–125.
- I. Alzamil, K. Djemame, D. Armstrong, and R. Kavanagh, "Energy-aware profiling for cloud computing environments," *Electronic Notes in Theoretical Computer Science*, vol. 318, pp. 91–108, 2015.
- I. Alzamil and K. Djemame, "Energy prediction for cloud workload patterns," in *Interna*tional Conference on the Economics of Grids, Clouds, Systems, and Services. Springer, 2016, pp. 160–174.
- A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.
- K. Park and V. S. Pai, "Comon: a mostly-scalable monitoring system for planetlab," ACM SIGOPS Operating Systems Review, vol. 40, no. 1, pp. 65–74, 2006.
- 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.
- 23. M. Zakarya and L. Gillam, "Managing energy, performance and cost in large scale heterogeneous datacenters using migrations," *Future Generation Computer Systems*, 2018.
- R. Jauregui and F. Silva, "Numerical validation methods," in *Numerical Analysis-Theory* and Application. InTech, 2011.
- B. H. Thacker, S. W. Doebling, F. M. Hemez, M. C. Anderson, J. E. Pepin, and E. A. Rodriguez, "Concepts of model verification and validation," Los Alamos National Lab., Los Alamos, NM (US), Tech. Rep., 2004.
- F. Teng, L. Yu, T. Li, D. Deng, and F. Magoulès, "Energy efficiency of VM consolidation in IaaS clouds," *The Journal of Supercomputing*, vol. 73, no. 2, pp. 782–809, 2017. [Online]. Available: "http://dx.doi.org/10.1007/s11227-016-1797-5
- W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *IEEE International Conference on Cloud Computing.* Springer, 2009, pp. 254–265.
- H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster computing*, vol. 16, no. 2, pp. 249–264, 2013.
- Q. Huang, F. Gao, R. Wang, and Z. Qi, "Power consumption of virtual machine live migration in clouds," in *Communications and Mobile Computing (CMC)*, 2011 Third International Conference on. IEEE, 2011, pp. 122–125.
- V. De Maio, G. Kecskemeti, and R. Prodan, "A workload-aware energy model for virtual machine migration," in 2015 IEEE International Conference on Cluster Computing. IEEE, 2015, pp. 274–283.
- J. O'Loughlin and L. Gillam, "Sibling virtual machine co-location confirmation and avoidance tactics for public infrastructure clouds," *The Journal of Supercomputing*, vol. 72, no. 3, pp. 961–984, 2016.
- 32. J. DeCoster, "Testing group differences using t-tests, anova, and nonparametric measures," *Accessed November*, vol. 30, no. 2010, pp. 202 006–0, 2006.
- 33. 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.

- 32 Modelling Heterogeneities in Cloud Simulations and Quantifying their Accuracy
- 34. —, "Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers," *Proceedings of the 8th International Workshop* on Middleware for Grids, Clouds and e-Science, no. December 2010, p. 6, 2011.
- V. De Maio, R. Prodan, S. Benedict, and G. Kecskemeti, "Modelling energy consumption of network transfers and virtual machine migration," *Future Generation Computer Systems*, vol. 56, pp. 388–406, 2016.
- J. OLoughlin and L. Gillam, "Towards performance prediction for public infrastructure clouds: An ec2 case study," in *Cloud Computing Technology and Science (CloudCom)*, 2013 IEEE 5th International Conference on, vol. 1. IEEE, 2013, pp. 475–480.
- G. Kecskemeti, W. Hajji, and F. P. Tso, "Modelling low power compute clusters for cloud simulation," in *Parallel, Distributed and Network-based Processing (PDP), 2017* 25th Euromicro International Conference on. IEEE, 2017, pp. 39–45.
- A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, "icancloud: A flexible and scalable cloud infrastructure simulator," *Journal of Grid Computing*, vol. 10, no. 1, pp. 185–209, 2012.
- S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer, "Groudsim: an eventbased simulation framework for computational grids and clouds," in *European Confer*ence on Parallel Processing. Springer, 2010, pp. 305–313.
- S.-H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das, "Mdcsim: A multi-tier data center simulation, platform," in *Cluster Computing and Workshops*, 2009. CLUS-TER'09. IEEE International Conference on. IEEE, 2009, pp. 1–9.
- M. Zakarya, "Energy, performance and cost efficient datacenters: A survey," *Renewable and Sustainable Energy Reviews*, vol. 94, pp. 363–385, 2018.