


RESEARCH

Open Access



# Multi-Dimensional Regression Host Utilization algorithm (MDRHU) for Host Overload Detection in Cloud Computing

Ali A. El-Moursy<sup>1\*</sup> , Amany Abdelsamea<sup>2</sup>, Rukshanda Kamran<sup>3</sup> and Mohamed Saad<sup>1</sup>

## Abstract

The use of cloud computing data centers is growing rapidly to meet the tremendous increase in demand for high-performance computing (HPC), storage and networking resources for business and scientific applications. Virtual machine (VM) consolidation involves the live migration of VMs to run on fewer physical servers, and thus allowing more servers to be switched off or run on low-power mode, as to improve the energy consumption efficiency, operating cost and CO<sub>2</sub> emission. A crucial step in VM consolidation is host overload detection, which attempts to predict whether or not a physical server will be oversubscribed with VMs. In contrast to the majority of previous work which use CPU utilization as the sole indicator for host overload, a recent study has proposed a multiple regression host overload detection algorithm, which takes multiple factors into consideration: CPU, memory and network BW utilization. This paper provides further improvement along two directions. First, we provide Multi-Dimensional Regression Host Utilization (MDRHU) algorithms that combine CPU, memory and network BW utilization via Euclidean Distance (MDRHU-ED) and absolute summation (MDRHU-AS), respectively. This leads to improved results in terms of energy consumption and service level agreement violation. Second, the study explicitly takes real-world HPC workloads into consideration. Our extensive simulation study further illustrates the superiority of our proposed algorithms over existing methods. In particular, as compared to the most recently proposed multiple regression algorithm that is based on Geometric Relation (GR), our proposed algorithms provide an improvement of at least 12% in energy consumption, and an improvement of at least 80% in a metric that combines energy consumption, service-level-violation, and number of VM migrations.

**Keywords:** Cloud Computing, Power management, Data center management, Virtual machine consolidation, Host overload detection, Multiple regression

## Introduction

Cloud computing [1] technology is acquiring a great deal of prominence across the computing and networking research communities. Nowadays, cloud data centers are the basic framework for the computing and data storage communities that offer expanded services to the end users. These data centers consume large amounts of electrical energy to process the cloud services bringing a large quantity of CO<sub>2</sub> emissions, high operational cost, and influencing the reliability of hardware equipments [2]. The

fundamental drain of power consumption in data centers are processing, network, cooling systems, and disk storage. Since conventional ways [3–5] to decrease the power consumption are no longer suitable for modern data centers, new adaptive software-oriented techniques are inevitable. To reduce energy consumption, it is important to address inefficiencies and waste in the manner electricity is conveyed to computing servers, and in the manner these resources are utilized to satisfy the running workloads. This may be possible by enhancing the data centers physical infrastructure, in addition to resource management and allocation algorithms.

One of the important methods to address the energy inefficiency in cloud data centers is to leverage the capabilities of the virtualization technology. Virtualization is

\*Correspondence: [aelmoursy@sharjah.ac.ae](mailto:aelmoursy@sharjah.ac.ae)

<sup>1</sup>Electrical and Computer Engineering Department, University of Sharjah, Sharjah, UAE

Full list of author information is available at the end of the article

a key player in cloud computing [6]. In particular, virtualization permits logical resource abstraction in isolation from their corresponding physical resources; the physical resources are transformed into virtual on-demand resources of cloud data centers. Virtualization is provided by the utilization of a hypervisor to logically allocate virtual machines (VMs) on physical server resources. The hypervisor permits the VM guest operating system to operate as if it were solely in control of the hardware, unaware that other guests are sharing it. Virtualization gives a chance to consolidate various VM instances running on under-utilized server machines into fewer nodes, allowing more servers to be switched-off, which leads to considerable energy savings. Virtual machine consolidation [7] involves scheduling/migrating several virtual machines into a few number of physical servers. Virtual machine consolidation has essentially an important tradeoff, namely reduction of energy consumption without sacrificing the Quality of Service (QoS) delivered by the system (energy-performance trade off). The improvement of the system QoS may be achieved indirectly by tuning parameters of the assigned host. The tuning is based on the utilization percentage achieved by the physical resources in a computing node. According to this percentage we can decide if a node is underloaded or overloaded. According to the workload information, a VM can be migrated from an overloaded host to an underloaded host. Server overload results in applications performance degradation and resource shortages, as well as an extensive increase in power consumption. Monitoring the overloaded hosts will enhance the host utilization and prevent thrashing of the VMs.

The capability to manage physical machine (PM) overload is a critical element of next-generation, competitive cloud services. If overload is not handled properly, cloud service providers will take the risk of violating their Service Level Agreements (SLAs) [8]. SLAs provide a level of assurance to customers that their requested resources are available when they request them [9]. An overloaded host directly affects the QoS because, if the resource capacity is fully utilized for a significant time window, it is expected that the applications will encounter performance degradation and resource shortage.

Most of the recent studies migrate VMs based on the CPU utilization of the physical server. This may be suitable for CPU-intensive applications, but will not be accurate for network, I/O and memory intensive applications [10]. Motivated by the fact that high-performance-computing (HPC) applications, and server applications usually utilizing cloud data centers, are sensitive to multiple factors (CPU, Memory and Network BW utilization), we propose a family of novel multi-dimensional regression host overload detection algorithms that explicitly take these orthogonal factors into consideration. In particular,

the most crucial aspect of multi-dimensional regression is how to combine the orthogonal factors (CPU, Memory and Network BW utilization) into a composite metric that accurately captures whether or not the host is overloaded. Making such a decision accurately is the ultimate goal of the regression algorithm. To this end, we introduce two alternative mathematical formulas for VM utilization, namely Euclidean Distance (ED) and Absolute Summation (AS), that are more general than the existing approach in [11].

More specifically, the contribution of this paper can be summarized as:

1. This paper introduces a Multi-Dimensional Regression Host Utilization algorithm (MDRHU) for host overload detection. In particular, we introduce two alternative mathematical composite metrics to measure the host utilization based on the VM utilization profiling of three independent factors: CPU, memory and network BW. The proposal is tuned and enhanced to meet the specs of the Cloud. The numerical results reveal the superiority of the proposed approach as compared to the existing state-of-the-art.
2. This study explicitly takes real-world HPC workloads into consideration. This ensures the trustworthiness of our numerical study in revealing the suitability of our proposed approach for HPC applications.

This paper is organized as follows: “[Related work](#)” section presents the related work, and summarizes the contribution of this paper. “[Multi-dimensional regression host utilization for host overload detection algorithms \(MDRHU\)](#)” section introduces the proposed regression-based, multi-dimensional host overload detection algorithm. “[Evaluation methodology](#)” section presents the evaluation methodology. “[Simulation results and analysis](#)” section presents the simulation results. Finally, “[Conclusion](#)” section concludes the paper.

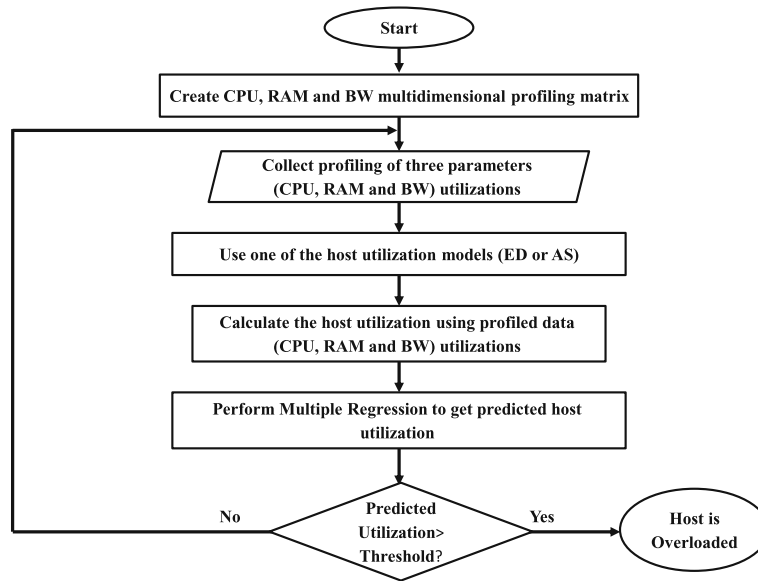
## Related work

Previous techniques to energy efficient host overload detection can be broadly divided into three categories: (1) static threshold based heuristics, (2) adaptive utilization based heuristics and (3) regression based heuristics. All categories enjoyed significant attention from the research community, so we focus here on the most relevant and significant work.

Threshold-based techniques rely on setting a static CPU utilization threshold distinguishing the non-overload and overload states of the node. These techniques compare the current CPU utilization of the host against the predefined threshold. If the threshold is exceeded, a host overload is declared. See, e.g., [12] and [13]. Fixed

**Table 1** Comparison between the multi-factor related work

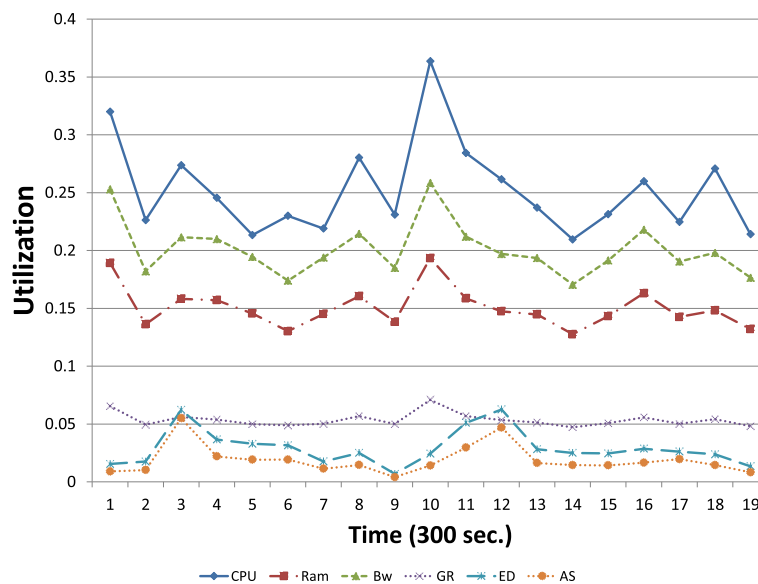
Related work	Parameters	Target	Algorithm name	Advantages	Disadvantages
Tian et al [16]	CPU, memory and network BW	Resource scheduling	Dynamic and Integrated Resource Scheduling (DAIRS)	<p>-It provides a dynamic and integrated resource scheduling which treats CPU, memory and network BW integrated for both physical machines and virtual machines</p> <p>-It provides integrated measurement of the total imbalance level of a Cloud data center as well as the average imbalance level of each server for performance evaluation</p>	It did not consider energy-efficiency for Cloud data centers
Tang et al [17]	CPU, Communication network in a data center	Virtual Machine Placement in server consolidation	Hybrid GA (HGA)	<p>-It considers the energy consumption in both physical machines and the communication network in a data center</p> <p>-Compared with existing heuristic algorithms, the HGA can generate much better solutions as the HGA is basically a global search algorithm while heuristic algorithms are local search algorithms, which may be trapped at a local optimum during their search process</p>	Its computation time is higher than other existing heuristic algorithms
Castro et al [18]	CPU and RAM	Virtual machine Placement	CPU and RAM Energy aWare (CREW)	<p>-It improves VM placement by employing a power model that considers the energy consumed by both CPU and RAM</p> <p>-CREW offers the best trade-off between energy saving and SLA violation</p>	It considers CPU and RAM only and it did not consider the communication network in a data center
Li et al. [19]	CPU and disk	Host overload/ underload double threshold detection and VM placement	Multi-resource double threshold method and Modified Particle Swarm Optimization (MPSO) VM placement	<p>-It designed a method of double threshold with multi resource utilization to trigger the migration of VMs. The Modified Particle Swarm Optimization method is introduced into the consolidation of VMs to avoid falling into local optima which is a common defect in traditional heuristic algorithms</p>	It did not consider the memory and communication network in a data center
Farahnakian et al. [20]	CPU and memory utilizations	A virtual machine consolidation	A Utilization Prediction-aware VM Consolidation (UP-VMC) approach	<p>-UP-VMC goes beyond the existing works which only consider CPU utilization by also considering memory. Combining both memory and CPU utilization, UP-VMC can better identify causes of SLA violations and consequently prevent them from happening.</p> <p>-In contrast to the existing VM consolidation methods which mostly rely on the current resource utilization of PMs, UP-VMC considers both current and future resource utilization</p>	It should take into account network resource utilization and traffic to optimize VM placement.
Abdelsamea et al. [11]	CPU and RAM and network BW	Host overload detection	Multiple Regression Host Overload Detection (MRHOD)	<p>-It significantly reduces energy consumption while ensuring a high level of adherence to Service Level Agreements (SLA)</p>	It uses a workload that is insignificant for BW



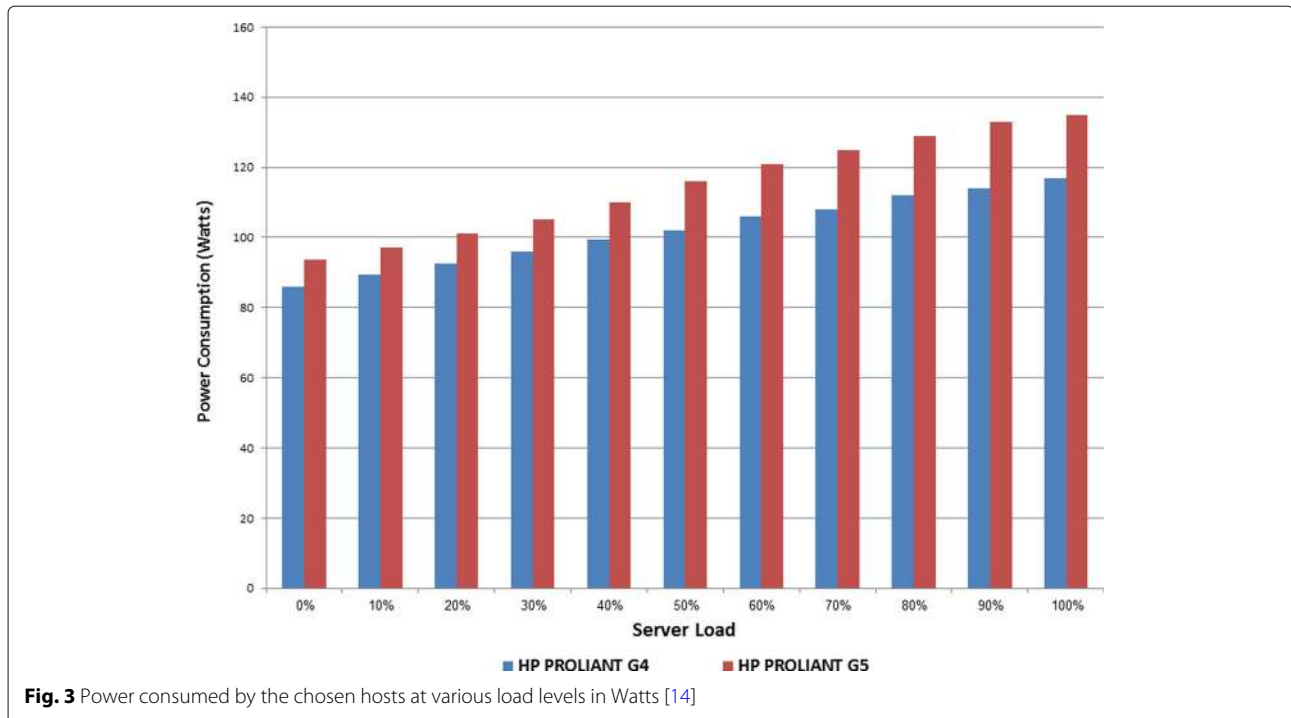
**Fig. 1** Multi-Dimensional Regression Host Utilization for Overload Detection (MDRHU)

utilization thresholds are not suitable for environments with dynamic and unpredictable workloads, where various types of applications may share a physical resource. The system should be capable of adjusting automatically its behavior based on the workload patterns. Adaptive utilization based algorithms provide auto-adjustment of the utilization thresholds depending on a statistical analysis of historical data calculated within the lifetime of the VMs. See, e.g., [7, 14, 15]. Adaptive utilization based

algorithms are efficient for dynamic environments, but provide poor prediction of host overloading. Hence, host overload detection may also benefit from the estimation of future CPU utilization. Regression-based techniques provide better prediction of host overloading because they depend on estimation of future CPU utilization. Although they are complex, their benefits may payoff. Example regression algorithms include the Local Regression (LR) algorithm [14]. The basic idea of local regression is to



**Fig. 2** CPU, RAM, BW, Geometric Relation (GR), Euclidean Distance (ED) and Absolute Summation(AS)



fit a simple model to the localized subsets of the data to produce a curve that approximates the original data.

The main drawback of the above-mentioned studies is that host overload detection is mainly based on the CPU utilization. Studies that combined multiple criteria, e.g., CPU, memory and/or BW utilization, do exist. Table 1 provides an overview of the main existing studies that considered multiple factors for detecting host overload. In a previous effort, Abdelsamea et al. [11] have developed a multiple regression algorithm that uses CPU utilization, memory utilization and BW utilization for host overload detection. They proposed preliminary results for this methodology using random and planetLab workload traces which are significant for CPU and RAM but are insignificant for BW utilization. This follow-up paper proposes a new family of multi-dimensional regression host overload detection algorithms using Euclidean Distance (ED) and Absolute Summation (AS) that are more general than that in [11]. Also, and in contrast to [11], we use real-life HPC workloads that are sensitive to CPU, RAM and network BW.

It is worth noting that host overload detection is considered as the first step in the VM consolidation process. Once the overloaded hosts are detected, VM

consolidation entails two other steps: (1) selecting the VMs to be migrated from the overloaded hosts to other hosts (known as VM migration), and (2) replacement of the VMs selected for migration on new hosts (known as VM placement). Some recent studies on energy-efficient VM placement and VM migration can be found, e.g., in [21–23]. In particular, the study in [21] addressed the VM placement problem with the objective of improving energy consumption and SLA violations. The authors proposed a novel VM placement algorithm based on the bin packing heuristic. The study in [23] also addressed the energy-efficient VM placement problem, but proposed a genetic algorithm meta-heuristic. Finally, the study in [22] presented new energy-efficient VM placement and VM migration policies. Our work, however, focuses on host overload detection.

**Multi-dimensional regression host utilization for host overload detection algorithms (MDRHU)**

In what follows, we present a novel family of multiple regression based algorithms for host overload detection. The presented algorithms explore different models for host utilization using multiple factors, namely CPU,

**Table 2** Planetlab Workload data [14]

Date	03-03-2011	06-03-2011	09-03-2011	22-03-2011	25-03-2011	03-04-2011	09-04-2011	11-04-2011	12-04-2011	20-04-2011
Number of VMs	1052	898	1061	1516	1078	1463	1358	1233	1054	1033

**Table 3** Relative Energy consumption for different algorithms vs. Number of VMs

Algorithm	898	1033	1233	1463	1516
HLRHOD	-0.077	-0.121	-0.054	-0.076	-0.096
MRHOD	-0.089	-0.144	-0.089	-0.081	-0.115
MDRHU-ED	-0.211	-0.243	-0.176	-0.245	-0.291
MDRHU-AS	-0.207	-0.244	-0.206	-0.242	-0.289

memory and BW. The flowchart for the proposed Multi-Dimensional Regression Host Utilization (MDRHU) algorithm is presented in Fig. 1. In order to predict future host utilization values, the regressor requires two major components. The first component is the profiled data for the independent factors (dimensions) of the running VMs that contribute to the host utilization evaluation. This data is readily available through the online profiling of the VMs and can be superimposed/aggregated across all VMs per host.

The second and the most critical component needed to perform the multiple regression process is the function to combine the independent factors together to derive a dependent metric representing the overall host utilization. This non-trivial component is required as the host utilization cannot be measured directly. A successful evaluation of the host utilization leads to a correct decision of whether the host is really overloaded or not, which is the ultimate goal of our proposed algorithm. Although the work in [11] incorporated the Geometric Relation (GR) function for the host utilization formulation, we believe it does not closely trace the actual host utilization behaviour. In this paper, two alternative models considering the space distance (the multi-dimensionality of the problem) for the host utilization are proposed as discussed, namely, Euclidean Distance (ED) and Absolute Summation (AS). The following paragraph discusses these two proposed models. In addition, GR is also discussed as a benchmark for comparison purposes.

1. **MRHOD** [11]: Geometric Relation (GR) is a multi-factorization relationship that combines multiple parameters in one metric that summarizes the overall system behavior. The most critical factors to be considered for VMs are CPU, memory and BW. However, the absolute values for those factors scores

**Table 4** Relative ESV metric vs. Number of VMs

Algorithm	898	1033	1233	1463	1516
HLRHOD	-0.161	-0.113	-0.078	-0.136	-0.072
MRHOD	-0.164	-0.112	-0.046	-0.126	-0.056
MDRHU-ED	-0.245	-0.233	-0.248	-0.269	-0.206
MDRHU-AS	-0.267	-0.216	-0.223	-0.27	-0.234

are not the desired parameters to be used. The utilization of the aforementioned factors relative to the maximum permissible utilization is more meaningful to make the factors dimensionless and representative for the host overload. The maximum utilization is either defined by the cloud service provider or the absolute available host utilization. Hence, the profiled data is to be normalized to represent a fractional utilization per factor. Geometric Relation (GR) for host utilization evaluation is used in [24] as shown in Eq. 1. However, the GR lacks the consideration of the orthogonality of the multi-dimensional space among the different factors.

$$HostUtilization = \frac{\omega_1}{1 - CPU} \times \frac{\omega_2}{1 - RAM} \times \frac{\omega_3}{1 - BW} \quad (1)$$

where  $\omega_i$  The weight of for factor  $i$ , where the factors are CPU, memory and BW

CPU The relative CPU utilization

RAM The relative memory utilization

BW The relative network utilization

An alternative method to the GR relation above is to use different space distance methodologies. Several methods can be used [25] such as Euclidean Distance (ED) or Absolute Summation (AS) to calculate the distances of two points in space.

2. **MDRHU-ED**: As mentioned above, the CPU, RAM and BW utilizations have different scales or measures. Therefore, the host utilization cannot be determined by simply summing up these values. To overcome this difficulty, the objective is divided by a normalization constant (normConst) as shown in Eq. 2.

$$HostUtilization = \frac{CPU + RAM + BW}{normConstED} \quad (2)$$

Our proposed algorithm MDRHU-ED uses the Euclidean distance between the current and previous host utilizations as the normalization constant. In particular, the normalization constant used by MDRHU-ED is shown in Eq. 3:

$$normConstED = \sqrt{d(CPU)^2 + d(RAM)^2 + d(BW)^2} \quad (3)$$

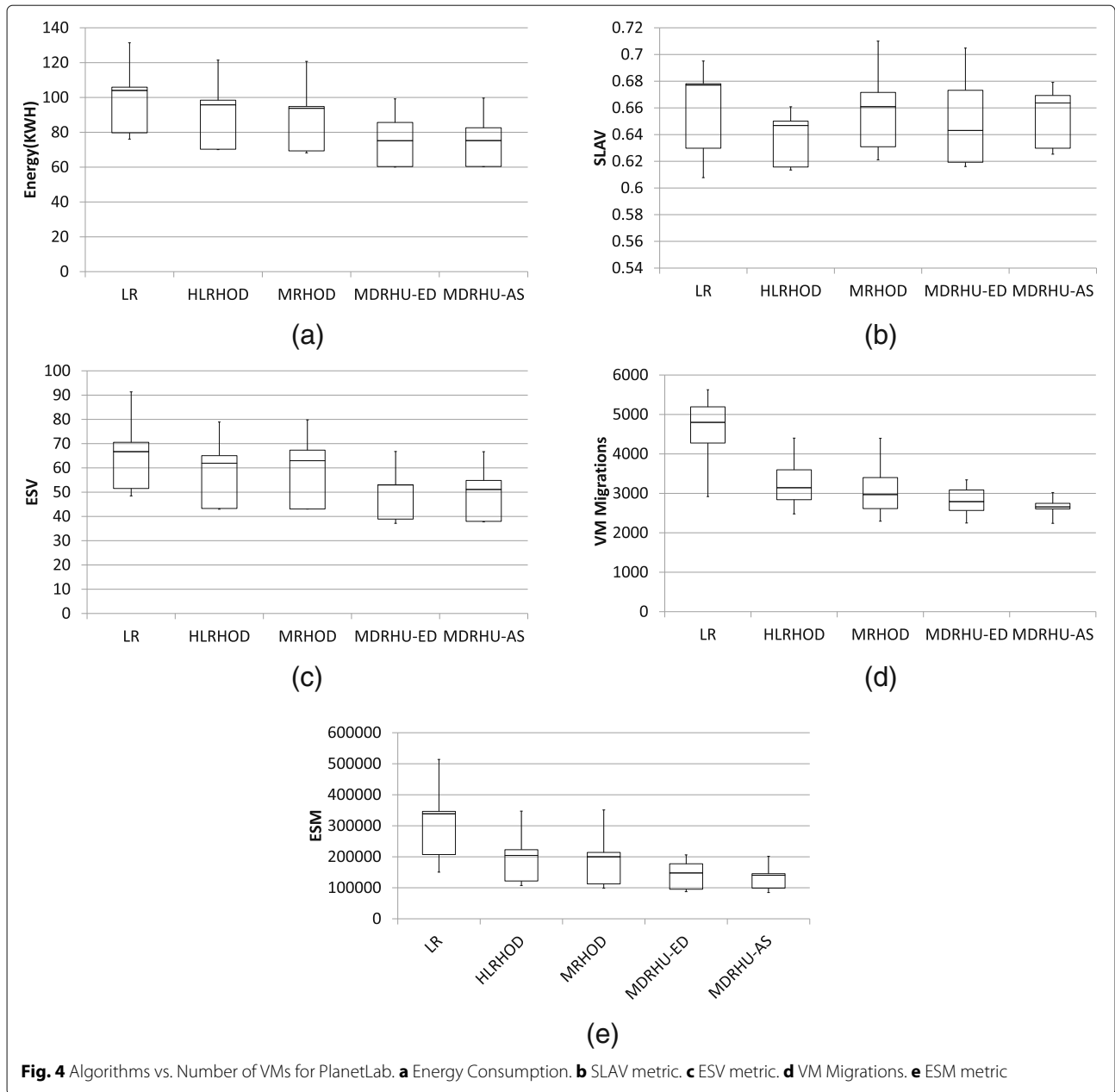
Note that  $d(CPU)$ ,  $d(RAM)$  and  $d(BW)$  denote the relative difference between the current and previous CPU utilizations, memory utilizations and BW utilizations, respectively.

3. **MDRHU-AS**: Similarly to MDRHU-ED, our alternative proposed algorithm MDRHU-AS estimates the host utilization using:

$$HostUtilization = \frac{CPU + RAM + BW}{normConstAS}$$

The normalization constant (normConst), however, is calculated as the summation of the absolute values





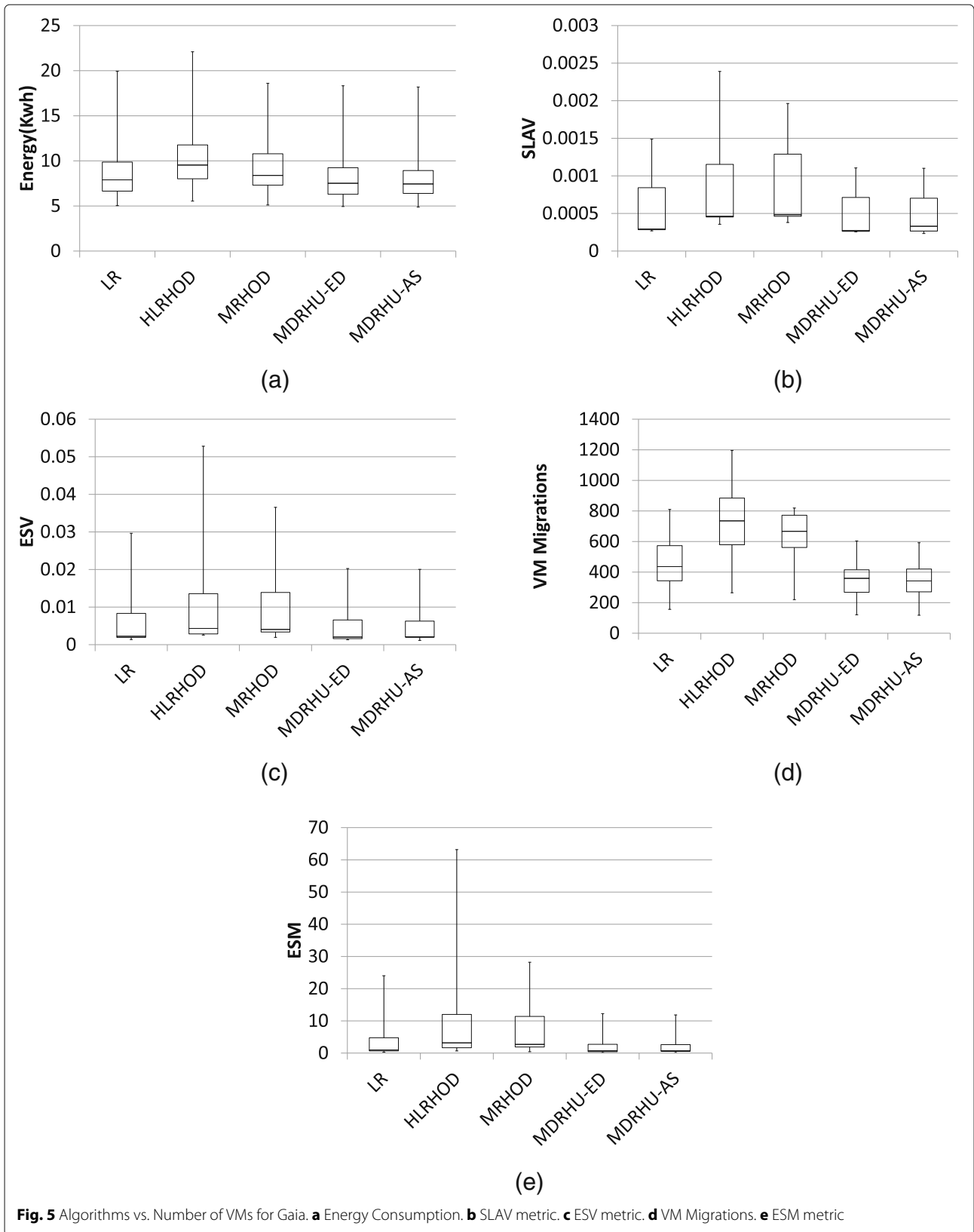
of the utilization distances. In particular, the normalization constant used by MDRHU-AS is shown in Eq. 4:

$$normConstAS = |d(CPU)| + |d(RAM)| + |d(BW)| \quad (4)$$

Again,  $d(CPU)$ ,  $d(RAM)$  and  $d(BW)$  denote the relative difference between the current and previous CPU utilizations, memory utilizations and BW utilizations, respectively. This strategy results in a bigger divisor as compared to the Euclidean distance strategy, and is considered to be preferable for scaling problems that are considered badly

incommensurable. On the other hand, it is characterized by a similar robustness to the Euclidean distance method.

It is worth noting that our proposed alternative approaches (MDRHU-ED and MDRHU-AS) are used to estimate the host utilization using profiled data related to CPU, memory and BW utilizations, respectively. As shown in Fig. 1, the next step is to build a general model that maps the independent variables (here CPU, memory and BW) to their corresponding dependent variable (here host utilization). This is accomplished using a multiple





regression algorithm. See, e.g., [26]. Multiple regression is an extension of simple linear regression. Its objective is to predict the value of a dependent variable (here host utilization) based on the values of multiple independent variables (here CPU, memory and BW). The outcome of the multiple regression algorithm is a prediction of the future host utilization. After the *predicted Host Utilization* is obtained through the regressor, its value is evaluated. The host is considered to be overloaded if the predicted host utilization exceeds some threshold as recommended by [14]. Consequently, a VM is selected to be migrated from the overloaded host.

To examine the correlation between the multi-dimensional approach and the utilization of the single factor parameters, Fig. 2 depicts the VM utilization for each individual factor and the host utilization calculated through GR, as compared to the two proposed multi-dimensional models (MDRHU-ED and MDRHU-AS) at different time slots. We can observe that the GR host utilization is somewhat averaging the individual utilization behaviour. On the other hand, ED and AS trace the overall utilization behaviour better with more pronounced variation in the curves.

## Evaluation methodology

### Experimental setup

Since it is hard to conduct repeatable large scale experiments on a real infrastructure [14], simulations are chosen as an approach to highlight the superiority of our proposed algorithms. The Cloudsim toolkit [27] has been utilized as a simulation framework because of the following reasons. Cloudsim supports VM provisioning at two levels: the host level and the VM level. At both levels, Cloudsim performs space-shared and time-shared provisioning techniques. Space-shared techniques distribute certain CPU cores among the VMs. These techniques behave similarly to the First Come First Serve (FCFS) scheduling algorithm. Time-shared techniques variably distribute the capacity of one core among the VMs. These techniques act similarly to the Round-Robin (RR) scheduling technique. Also Cloudsim permits the modeling of virtualized frameworks, sustaining on demand resource provisioning and resource management. Therefore, we choose the Cloudsim 3.0.3 toolkit as our simulation platform. We have extended Cloudsim with energy-aware simulations, which were originally not available in the core framework [27]. For the multi-dimensional regression process, Ordinary Least Square (OLS) Multiple Regression function [26, 28] is used for the regression coefficient calculation, then host utilization prediction.

To assess the performance of the newly proposed algorithms (MDRHU-ED and MDRHU-AS), we use the following host overload detection algorithms from the literature as benchmarks:

- The Local Regression (LR) algorithm from [14].
- The Hybrid Local Regression Host Overload Detection (HLRHOD) algorithm from [11].
- The Multiple Regression Host Overload Detection (MRHOD) algorithm from [11].

It is worth noting that LR is already implemented in the CloudSim simulator. However, we add an implementation of HLRHOD and MRHOD, as well as our proposed algorithms (MDRHU-ED and MDRHU-AS), to CloudSim.

A critical parameter to be adjusted for the host overload detection process is the threshold used to identify whether the host is overloaded or not. This parameter is called *safety parameter* in Cloudsim. The safety parameter defines how aggressively the system consolidates VMs on physical servers. If the safety parameter is too tight, opportunities for energy savings become too low. On the other hand, if the safety parameter is too relaxed, the levels of service level agreement violations become too high. Therefore, we perform an experimental (via simulation) selection for the safety parameter value, as to achieve an acceptable tradeoff between energy saving and SLA violation, as indicated by the overall performance metric. All results reported in the following sections are using the experimentally adjusted safety parameter.

Note also that, once a host overload has been detected, the next step is to select particular VMs to be migrated from the overloaded host to other hosts. To this end, we use the Minimum Migration Time (MMT) algorithm [14] for VM selection, and the modified Best Fit Decreasing (BFD) algorithm proposed in [14] for VM migration. It is worth noting that MMT and BFD are already implemented in Cloudsim.

### Power model

The Power consumed by computing resources in cloud data centers is mostly consumed by the CPU, disk storage, memory, power supplies and cooling devices [29]. Establishing exact analytical models for power consumption is a complicated task due to the complexity of the power models of modern multicore CPUs. Consequently, we utilize real data on power consumption offered by the outcomes of the SPECpower benchmark [14] as an alternative to the usage of an analytical power consumption model. The host overload is frequently examined every scheduling interval chosen to be 300 sec. The host types are: HP ProLiant ML110 G4 (Intel Xeon 3040/2 cores/1860 MHz/4 GB), and HP ProLiant ML110G5 (Intel Xeon 3075/2 cores/2660 MHz/4 GB), and their power consumption features are shown in Fig. 3.

### Performance metrics

The metrics we use to assess the performance of our proposed algorithms are summarized as follows:

- **Total energy consumption (E)** is defined as the sum of energy consumed by the physical resources of a data center, and is computed using the model presented in [14, 30].
- **Number of VM migrations** is an important metric reflecting the time needed to migrate VMs from the overloaded host to other underloaded hosts.
- **SLA Violation (SLAV)** captures the performance degradation due to host overloading and performance degradation due to VM migration. SLA Violation (SLAV) [14] occurs when a VM cannot obtain its promised Quality of Service (QoS) [31, 32].
- **Energy and SLA Violations (ESV)** is a metric that combines energy consumption and SLA Violations. This metric is originally proposed in [14], and is shown in Eq. 5. Note that the energy consumed by physical hosts and SLAV are adversely related, as energy can be frequently minimized on the expense of increasing SLA violations. The lower the ESV metric lower, the better the performance.

$$ESV = E \times SLAV \quad (5)$$

- **Energy-SLA-Migration (ESM)** [33] captures the simultaneous minimization of energy, SLA violation, and number of VMs migrations, and is given by Eq. 6.

$$ESM = E \times SLAV \times \text{Number of VM migrations} \quad (6)$$

### Workloads

We have used two real workloads for our experiments. The first workload data is provided as a part of the CoMon project, a monitoring infrastructure for *PlanetLab*. We have randomly chosen 10 days of the workload traces collected during March and April 2011 [14]. Each day is characterized by a specific number of VMs as shown in Table 2.

The second workload contains 3 months worth of HPC data from the Gaia cluster at the University of Luxembourg [34]. The workload data includes CPU and memory usage [35]. The log is available directly in standard workload format (SWF). We use UniLu-Gaia-2014-1.swf. It is based on accounting data collected by the scheduler. We build three utilization models that read the CPU and RAM, and calculate the BW, respectively, from the workload and pass it to the cloudlets.

## Simulation results and analysis

### Sensitivity analysis

In this subsection, we assess how the performance of our proposed algorithms are affected by varying the number of VMs, and by varying the scheduling interval. Experiments for the sensitivity analysis uses *PlanetLab* workload traces by default. However, the sensitivity for the number of VMs is also studied for the Gaia workload.

### Number of VMs

Varying the number of VMs with fixed 800 hosts and 300 sec. scheduling interval is tested and evaluated. The effect on the E and ESV metrics are shown in Tables 3 and 4, respectively. MRHOD is used as the baseline algorithm using Eq. 7. The higher the energy consumption, the more inferior the algorithm. Accordingly, a positive value for this equation indicates worse performance, while a negative value indicates better performance.

$$\text{Relative Improvement} = \frac{\text{Algorithm} - \text{Reference}}{\text{Reference}} \quad (7)$$

As shown in Table 3, MDRHU-ED is the best since it causes a reduction in energy consumption by about 21% as compared to the LR algorithm. As the number of VMs increases, MDRHU-AS is the best when the number of VMs is 1033 and 1233. HLRHOD is the simplest multiple regression algorithm since it depends on local regression. Its best improvement is when the number of VMs is 1033 where the energy consumption is enhanced by about 12% compared to LR. MRHOD based on multiple regression outperforms HLRHOD in term of energy consumption. MRHOD gives an improvement more than HLRHOD of about 2% for 898 VMs and about 14% for 1033 VMs. Also it is clear from Table 3 that space distance based multiple regression algorithms (MDRHU-ED and MDRHU-AS) give better enhancement in term of energy consumption than Geometric Relation (GR) based multiple regression algorithm (MRHOD). MDRHU-ED is the best when number of VMs are 898, 1463 and 1516 while MDRHU-AS is the best when number of VMs are 1033 and 1233.

In Table 4, when the number of VMs is small (898), MDRHU-AS gives the best results as it causes a reduction of the ESV metric by about 26%. For the largest number of VMs (1516), MDRHU-AS is the best since it decreases the ESV by about 23%. HLRHOD gives better results than MRHOD across all number of VMs except when the number of VMs is 898. MDRHU-ED is the best when number of VMs are 898, 1463 and 1516, respectively, while MDRHU-AS is the best when number of VMs are 1033 and 1233, respectively. The results in Table 4 prove that multi-dimensional regression algorithms do not sacrifice energy consumption with SLA violations.

**Table 5** Relative Energy consumption vs. Scheduling interval (sec.)

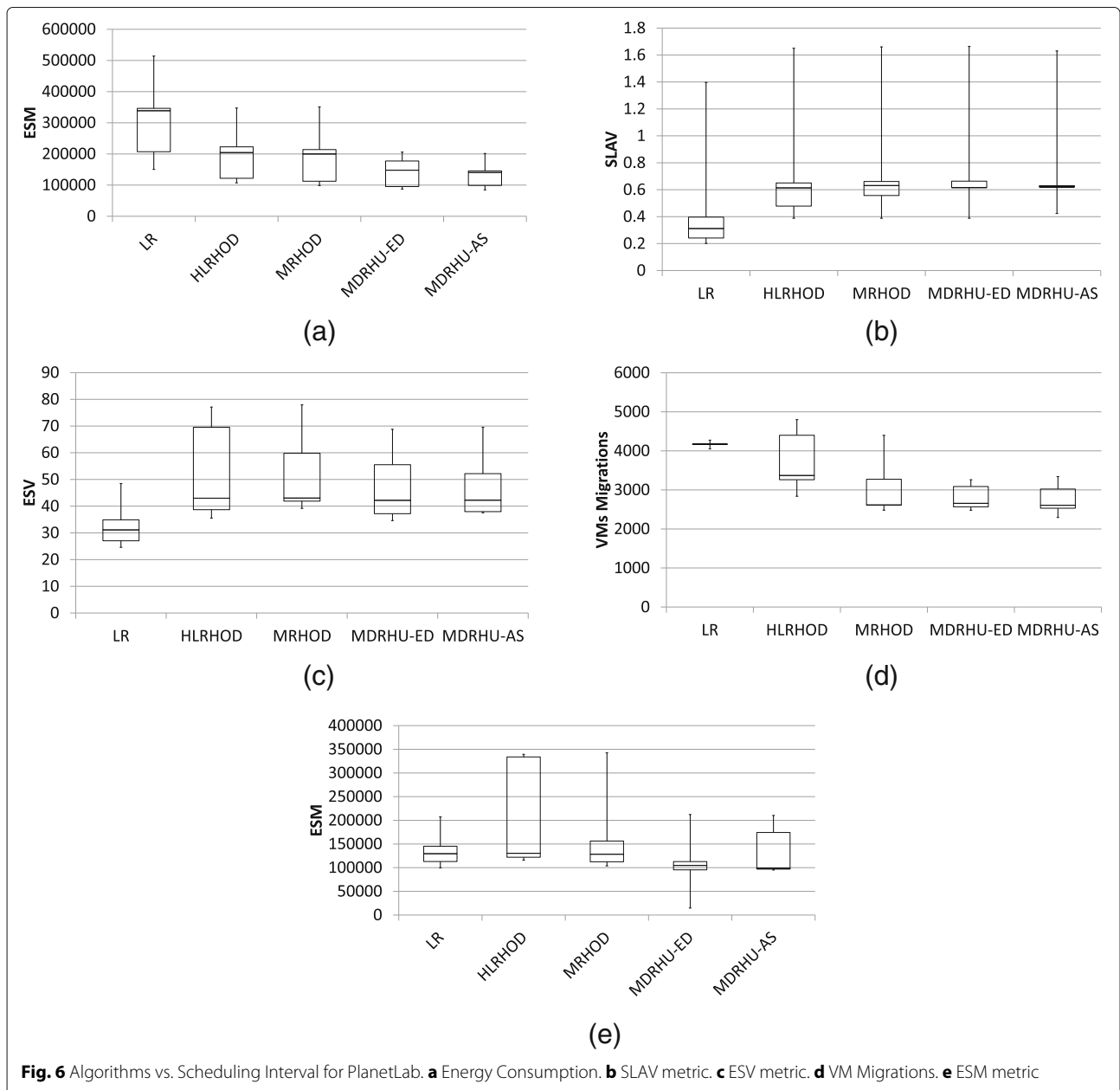
Algorithm	300	500	700	900	1100
HLRHOD	-0.122	-0.08	-0.085	-0.084	-0.028
MRHOD	-0.144	-0.145	-0.097	-0.099	-0.033
MDRHU-ED	-0.243	-0.222	-0.162	-0.205	-0.163
MDRHU-AS	-0.244	-0.227	-0.156	-0.208	-0.148

Figure 4 shows the variation of the results for different number of VMs per algorithm as a box plot to indicate the maximum, minimum, first quartile and third quartile using *PlanetLab* workloads. Each part of the Figure shows one of the evaluation metrics as (a) E, (b) SLAV, (c) ESV, (d) VM migrations and (e) ESM across different algorithms. For the energy consumption, both newly proposed MDRHU-ED and MDRHU-AS algorithms are very close to each other, and better than all other algorithms. MDRHU-AS is very stable and shows the least variation in its results for the VM migrations, followed by MDRHU-ED which comes in the second place. For the SLAV, HLRHOD is the best. However, for the combined

metrics (ESV and ESM) MDRHU-ED and MDRHU-AS are the best. Similar behaviour is noticed for the Gaia workload with even better results for MDRHU-ED and MDRHU-AS, as depicted in Fig. 5. However, it is interesting to see HLRHOD and MRHOD are not performing as good as LR for Gaia workloads. This supports the importance of using the multi-dimensional regression approach for the HPC workloads.

**Scheduling Interval**

Relative energy consumptions with respect to LR while varying the scheduling interval with fixed 800 hosts and 1033 VMs are shown in Table 5. Increasing the scheduling

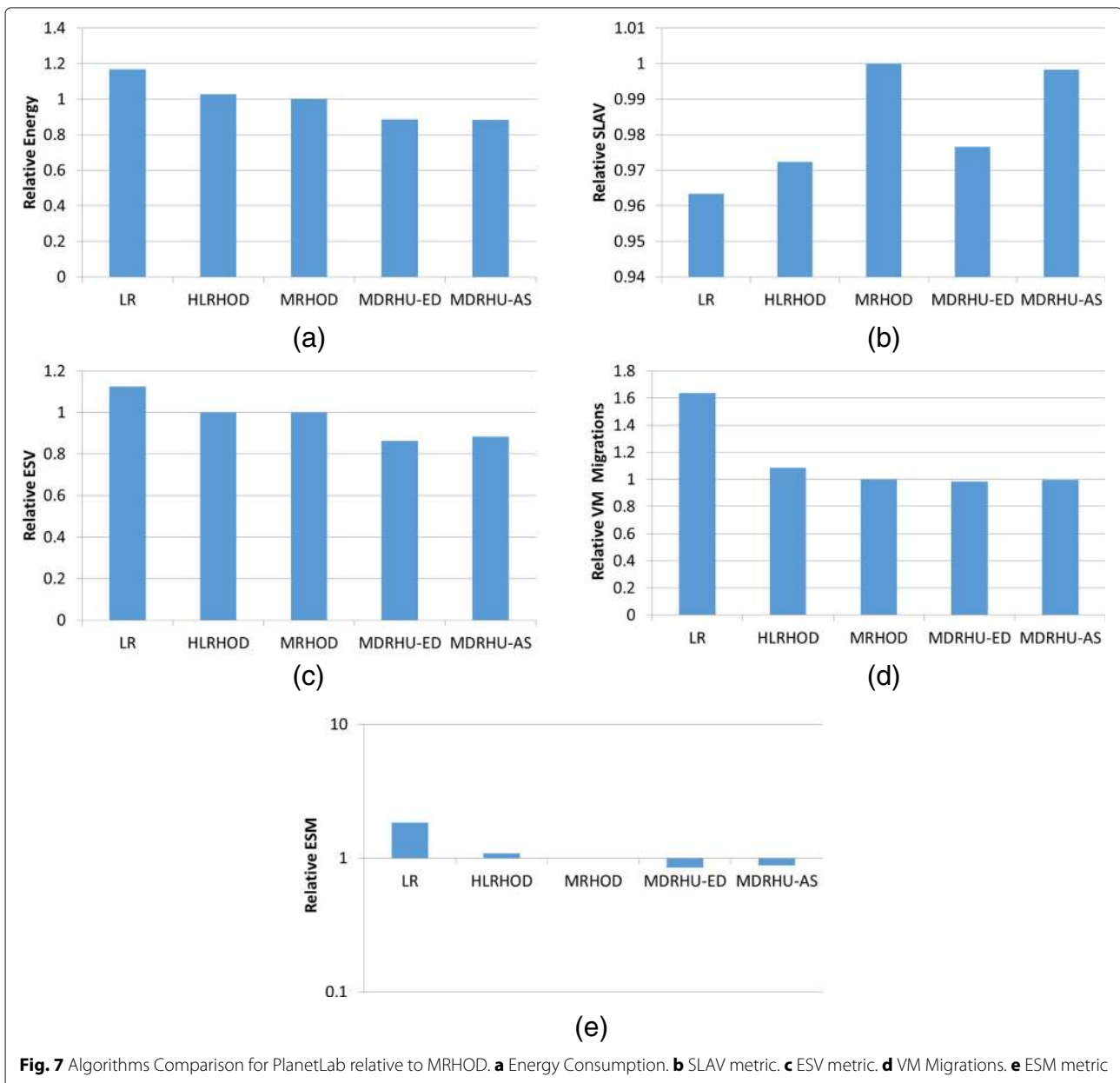


**Fig. 6** Algorithms vs. Scheduling Interval for PlanetLab. **a** Energy Consumption. **b** SLAV metric. **c** ESV metric. **d** VM Migrations. **e** ESM metric

interval indicates a lower frequency of running the VM consolidation and vice versa. HLRHOD provides better results than single factor algorithm since it causes a reduction in energy consumption by about 12% as compared to LR when the scheduling interval is 300 sec. Our new proposed multi-dimensional regression based algorithms outperform single factor algorithm as well as the HLRHOD multiple regression algorithm across all values of scheduling interval. MDRHU-AS gives the best energy consumption. In particular, it reduces the energy consumption by about 24% when the scheduling interval is 300 sec. It is worth noting that the scheduling interval specifies how often the overload detection is performed.

So, a small scheduling interval allows a better prediction of host overloading. This leads to better reduction in energy consumption as compared to when the scheduling interval is large. However, this comes at the cost of an increased computational complexity.

Similar to Fig. 4, Fig. 6 shows the variation of the results for different scheduling intervals. For the energy consumption, MDRHU-ED and MDRHU-AS outperform all other algorithms. MDRHU-AS is very robust, and shows no variation in the SLAVs. However, LR has the lowest SLAV and ESV values among all algorithms. However, it has the worst performance in terms of the VM migrations. MDRHU-ED is the best in terms of the ESM metric.



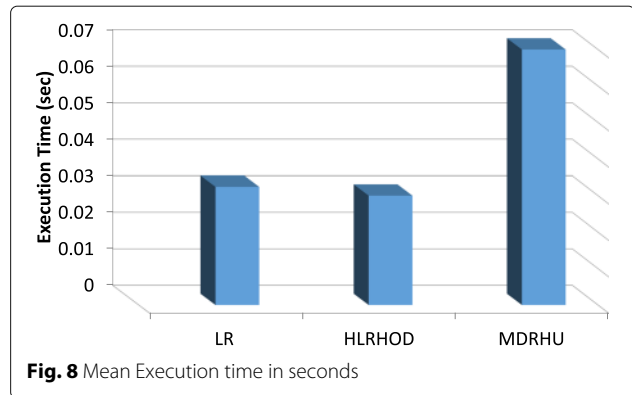
### Algorithms comparative analysis using PlanetLab Workload

The comparison between our proposed multi-dimensional family of algorithms (MDRHU-AS and MDRHU-ED) and the existing multiple factor algorithms (HLRHOD and MRHOD) is shown in Fig. 7. Note that the results are normalized with respect to MRHOD. Note also that the  $y$ -axis of Fig. 7e is in log-scale. We use a *PlanetLab* workload trace with the number of hosts set to 800 and the number of VMs set to 1033. For the *PlanetLab* workload we consider two factors only (CPU and RAM) in MDRHU and HLRHOD. The reason is that *PlanetLab* workloads are insensitive to the BW utilization because the VMs are not HPC in nature, and thus not communicating among each other. The next subsection, however, is devoted for results using a Gaia HPC workload trace that is sensitive to *all* three (CPU, RAM and BW) factors.

Across different metrics for power/performance and QoS, our proposed MDRHU-ED and MDRHU-AS algorithms outperform all other algorithms. In particular, the energy consumption is reduced by about 28% as compared to LR. This can be justified by the robustness of the absolute summation, which resists data outliers. Moreover, our newly proposed MDRHU-ED and MDRHU-AS algorithms outperform HLRHOD and MRHOD algorithms by about 12%. The SLAV is lower in LR as compared to other multiple factor algorithms by a negligible percentage (2%) because the energy and SLAV are negatively correlated. However, MDRHU-AS and MDRHU-ED improve the combined ESV metric by about 25% as compared to LR. MDRHU-AS and MDRHU-ED have lower (better) ESV than the recently proposed MRHOD by 12% and 14%, respectively.

Multi-dimensional regression algorithms outperform multi-factor algorithms due to their superiority in predicting host overloading, which leads to a decreased number of VM migrations as well. The latter leads to a decrease in the SLAV metric. The effect of number of VM migrations on energy consumption and ESV is observed when using the ESM metric. MDRHU-AS and MDRHU-ED have lower (better) ESM than the recently proposed MRHOD by 12% and 15%, respectively.

Although multiple regression enhances the energy consumption, its higher complexity adds overhead as compared to the simpler techniques. For the sake of fair comparison, Fig. 8 depicts the execution times of the different algorithms. HLRHOD is the best in terms of execution time. In spite of the higher execution time of MDRHU, the overhead is insignificant for typical job dispatch rates in cloud data centers. The other components of VM consolidation have even a higher overhead.



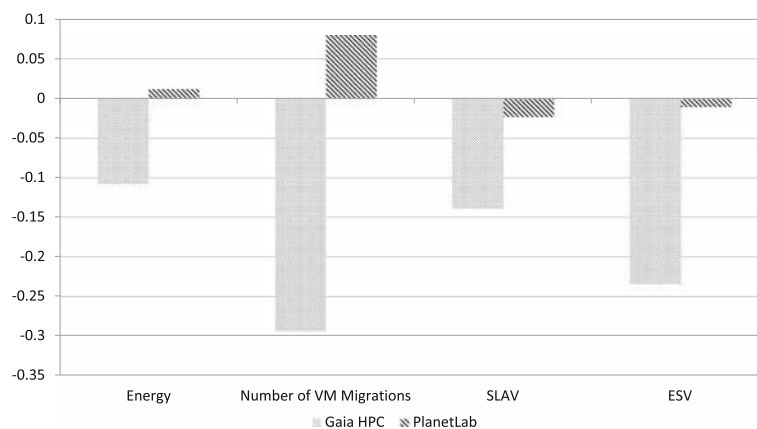
**Fig. 8** Mean Execution time in seconds

### Algorithms comparative analysis using Gaia Workload

In this part of our simulation study we use a Gaia HPC workload trace with the number of hosts set to 800 and the number of VMs set to 1001. Using the Gaia workload is motivated by its HPC nature, and its significance to the BW utilization parameter. To this end, we re-address the *PlanetLab* workload with 800 hosts and 1033 VMs as to illustrate, in contrast to Gaia, its insignificance to the BW utilization. In particular, we test the recently proposed algorithm MRHOD with only two factors (CPU and RAM) taken into consideration in the regression model, then with all three (CPU, RAM and BW) factors taken into consideration in the regression model. The same (two vs. three factors) comparison is done on the *PlanetLab* workload. Figure 9 depicts the parameters relative improvement results for MRHOD considering all three factors as compared to considering only two factors when using the Gaia HPC workload and the *PlanetLab* workload, respectively. In particular, considering three factors in MRHOD improves the energy consumption by more than 10% for the Gaia workloads, while considering three factors improves the energy consumption by only 0.3% for the *PlanetLab* workloads. MRHOD improves the ESV metric by more than 23% for the Gaia workloads, while considering three factors in the proposed MRHOD only makes the ESV metric worse for *PlanetLab* workloads. This proves that the *PlanetLab* workload is insignificant to the BW utilization, i.e., non-HPC in nature. In contrast, the Gaia workload is indeed HPC in nature due to its dependance on the BW utilization of the VMs. This partly illustrates the importance of this work in testing the proposed algorithms for real HPC workloads, in contrast to [11], which used only the non-HPC *PlanetLab*.

Since there is improvement when using three factors as opposed to using two factors, a comparison between different multiple factors algorithms (MRHOD and HLRHOD) and multi-dimensional algorithms (MDRHU-AS & MDRHU-ED) when taking all three factors (CPU, RAM, BW) into consideration is presented in Fig. 10. The results are normalized with respect to MRHOD. In





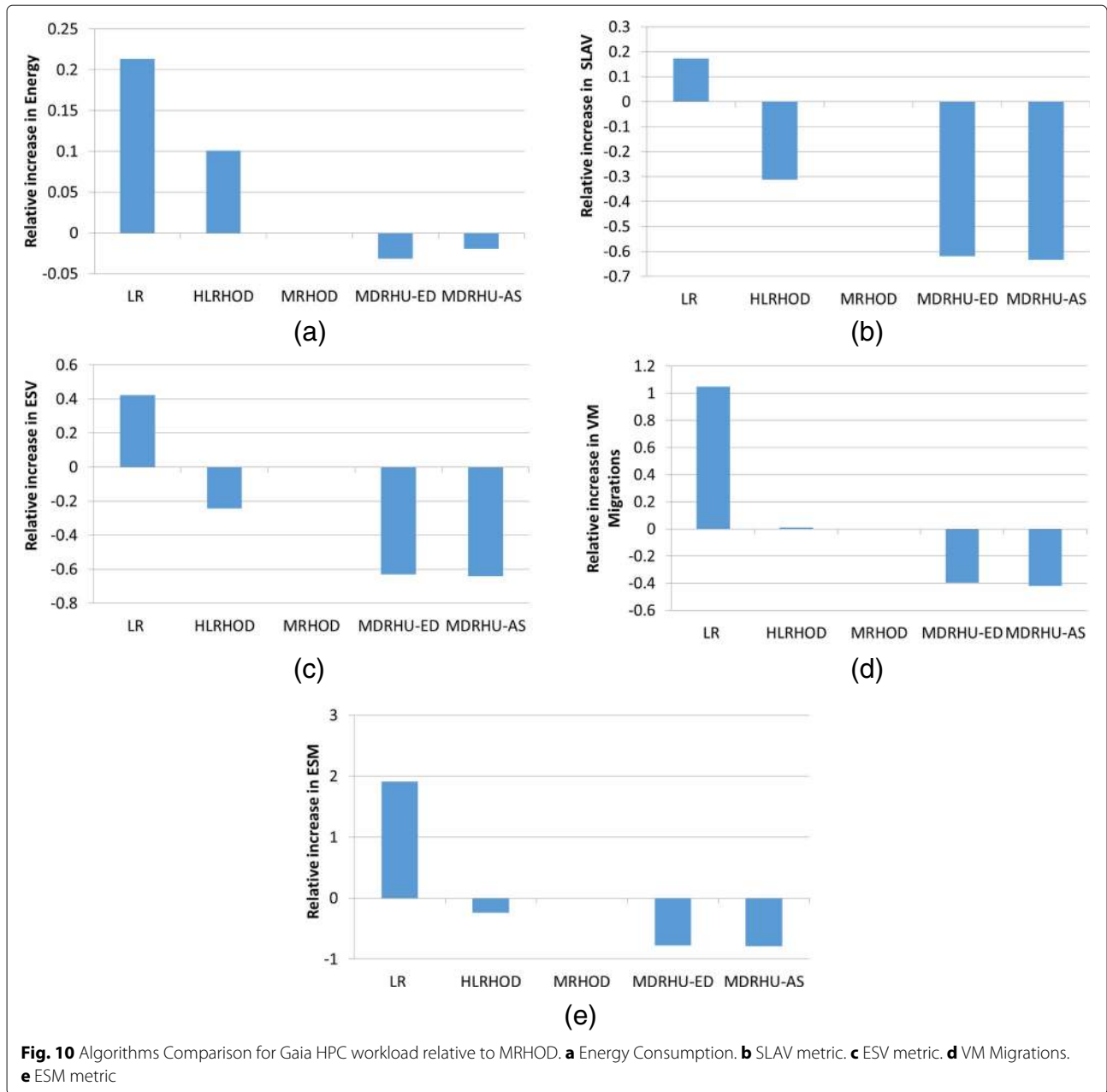
**Fig. 9** Percentage improvement for using three factors relative to using two factors in MRHOD for the Gaia HPC workload and the PlanetLab workload

particular, Fig. 10 shows that the consideration of the BW utilization in HPC workloads causes an improvement in all metrics for multiple factor algorithms. Using MRHOD leads to about 23% better (less) energy consumption and HLRHOD leads to about 14% better (less) energy consumption as compared to LR. However, the new multi-dimensional regression used in MDRHU-AS and MDRHU-ED achieve a slight energy saving of 2% and 4%, respectively. The SLAV metric is reduced in MRHOD by about 18% as compared to that of LR. MRHOD violates the service level agreement more than 60% as compared to the newly proposed MDRHU-AS and MDRHU-ED. Hence, the ESV metric is greatly improved for MDRHU-AS and MDRHU-ED as compared to using MRHOD or LR algorithm. The number of VM migrations is also reduced significantly by the newly proposed algorithms. MDRHU-AS and MDRHU-ED reduce the VM migrations by about 40% as compared to MRHOD. Hence, the ESM metric shows a significant improvement (reduction) for the new proposals. MDRHU-AS and MDRHU-ED reduce the ESM metric by about 78% and 80%, respectively, as compared to MRHOD. The improvement achieved using the new formulas in multi-dimensional regression using the Gaia HPC workload is six times better than the improvement achieved using the *PlanetLab* workload. However, we can hardly claim a winner when comparing MDRHU-AS and MDRHU-ED. Both of these multi-dimensional distance measurement concepts are significantly superior than using the GR formula; yet they are very close to each other to discriminate a winner.

## Conclusion

The use of cloud computing data centers has gained a lot of interest as a viable solution to satisfy the tremendously increased demand for high-performance computing (HPC), storage and networking resources for

business and scientific applications. Such large-scale data centers lead to excessive amounts of energy consumption, operating costs and  $CO_2$  emissions. Virtual machine (VM) consolidation, which involves the live migration of VMs to run on fewer physical servers, comes as an important solution because it allows more servers to be switched off or run on low-power mode, which helps reduce the energy consumption, operating cost and  $CO_2$  emission. A crucial step in VM consolidation is host overload detection, which attempts to predict whether or not a physical server will be oversubscribed with VMs. Unlike most of the previous work, which use the CPU utilization as the sole indicator for host overload, this paper took multiple factors into consideration: CPU, memory and network BW utilization. This is motivated by the fact that HPC applications are not only constrained by the CPU, but also by the memory and BW requirements. Therefore, this paper presented a family of novel multi-dimensional regression host overload detection algorithms, which combine CPU, memory and network BW utilization via Euclidean Distance (ED) and Absolute Summation (AS), respectively. The contribution of this paper is two-fold. First, the presented algorithms are based on multi-dimensional regression, leading to improved results in terms of energy consumption and service level agreement violation. Second, the proposed algorithms were tested using real-world HPC workloads. Our extensive simulation study illustrated the superiority of our proposed algorithms over existing methods. In particular, as compared to the most recently proposed multiple regression based on Geometric Relation (GR), our proposed algorithms provide an improvement of at least 12% in energy consumption, and an improvement of at least 80% in a metric that combines energy consumption, service level agreement violation and number of VM migrations.



**Abbreviations**

AS: Absolute Summation; E: Total energy consumption; ED: Euclidean Distance; ESV: Energy and SLA Violations; GR: Geometric Relation; HLRHOD: Hybrid Local Regression Host Overload Detection algorithm; LR: Local Regression; MDRHU: Multi-Dimensional Regression Host Utilization; MRHOD: Multiple Regression Host Overload Detection algorithm; VM: Virtual Machine

**Acknowledgements**

Not Applicable.

**Funding**

This research was supported in part by the the cloud computing center of excellence grant number 5220, Science and Technology Development Fund (STDF), Egypt, and in part by the Distributed and Networked Systems Research Group Operating Grant number 150410, University of Sharjah, UAE. Through those grants the Cloud system is established to perform the experiments and simulations.

**Availability of data and materials**

The data, through which the experiment is conducted, is available at: - The Gaia HPC workload. [http://www.cs.huji.ac.il/labs/parallel/workload/l\\_unilu\\_gaia/index.html](http://www.cs.huji.ac.il/labs/parallel/workload/l_unilu_gaia/index.html) (2014) - Parallel Workloads Archive. <http://www.cs.huji.ac.il/labs/parallel/workload> (2017) - <https://www.planet-lab.org/db/pub/slices.php> However, the full description of the experiment setup is provided in the manuscript in sections [Experimental setup](#) and [Workloads](#).

**Authors' contributions**

The authors equally contributed to this research and the paper initiated by the first author. All authors read and approved the final manuscript.

**Authors' information**

**Ali A. El-Moursy** (Senior Member, IEEE) received the Ph.D. in the area of High-performance Computer Architecture from University of Rochester, Rochester, NY, USA, in 2005. Dr. El-Moursy has worked for Software Solution Group, Intel Corp., CA, USA till early 2007. In 2007, he has joint Electronics



Research Institute, Giza, Egypt. His research interest is in high-performance computer architecture, multi-core multi-threaded micro-architecture, power-aware micro-architecture, simulation and modeling of architecture performance and power, workload profiling and characterization, cell programming, high performance computing, parallel computing and Cloud computing. Dr. El-Moursy has also participated with IBM Cairo Technology Development Center, Egypt, as a visitor research scientist for the period from Feb.2007 till Jan.2010. In Sep.2010 Dr. El-Moursy has joint ECE Dep.at University of Sharjah, Sharjah, UAE as an Assistant Prof.. In Jan.2017, Dr. El-Moursy has been promoted to the Associate Prof. Rank.

**Amany Abd El Samea** received the BSc degree from Shoubra Faculty of Engineering in 1999, the MSc degree from Faculty of Engineering Cairo University in 2006 and the PhD degree in Computer Engineering from Faculty of Engineering Cairo University in 2017. In 2000, she has joint Electronics Research Institute, Giza, Egypt. She attended the Chain-Reds school on Cloud Computing held at INFN Catania April 2015. Her research interest focus on the fields of parallel computing, distributed computing, grid computing and cloud computing.

**Rukshanda Kamran** has completed her Masters in Information Technology in 2005 from Preston University, Pakistan and Bachelors in Computer Science in 1999 from Shah Abdul Latif University Karachi, Pakistan with Honors. Rukshanda Kamran has working experience as Faculty of Information Technology in Bedfordshire University, UK (Dubai Campus) Sep 2011 till January 2014, Preston University Dubai for the period from early 2003 till May 2013 and as Research Assistant at University of Wollongong Dubai from July 2014 till Feb 2015 in the area of A Hybrid Cloud Enabled Supply Chain Network. Rukshanda Kamran has joined SZABSIT University Dubai In Sep 2013 and currently working as Assistant Professor SZABSIT University and In Feb 2016 Rukshanda Kamran as joint ECE Dep.at University of Sharjah, Sharjah, UAE and working as Research Assistant. Rukshanda Kamran is an enrolled Student for PhD. in the area of High-Performance Cloud Computing at University of Malaysia, Sarawak (UNIMAS) 2014. Her research interest are High performance computing and Cloud Technologies.

**Mohamed Saad** (Senior Member, IEEE) received Ph.D. degree in electrical and computer engineering from McMaster University, Hamilton, Canada, in 2004. He is currently an Associate Professor at the Department of Electrical and Computer Engineering, University of Sharjah, UAE. His research interests include networking, communications and optimization, with current activity focused on the optimal design of wireless and wired communication networks, and optimal network resource management. He has also held research positions with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada, and the Advanced Optimization Laboratory at the Department of Computing and Software, McMaster University, Hamilton, Canada. Dr. Saad is an editor for the International Journal of Distributed Sensor Networks. He was the recipient of the best paper award in the IEEE Symposium on Computers and Communications, Riccione, Italy, June 2010. He was the recipient of the University of Sharjah "Annual Incentive Award for Distinguished Faculty Members", for excellence in research, April 2010 (university-wide). He received also two best teaching awards by the IEEE Women in Engineering Society, University of Sharjah (in 2007 and 2009). He was also the recipient of a 2005-2006 Natural Sciences and Engineering Research Council of Canada (NSERC) post-doctoral fellowship. He is a senior member of the IEEE.

#### Competing interests

The authors declare that they have no competing interests.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

#### Author details

<sup>1</sup>Electrical and Computer Engineering Department, University of Sharjah, Sharjah, UAE. <sup>2</sup>Computers and Systems Department, Electronics Research Institute, Giza, Egypt. <sup>3</sup>Senior IT Faculty, SZABIST University, Sharjah, UAE.

Received: 1 January 2019 Accepted: 17 April 2019

Published online: 13 June 2019

#### References

1. Chen Q, Deng QN (2009) Cloud computing and its key techniques. *Jr. Comput Appl* 4:25–62

2. Kaushar H, Ricchhariya P, Motwani A (2014) Comparison of sla based energy efficient dynamic virtual machine consolidation algorithms. *Int J Comput Appl* 102:0975–8887
3. Prakash K, Edwin B (2013) Survey about power management techniques for high performance data centers in cloud environment. *Int J Eng Res Technol (IJERT)* 2:3403–3407
4. Wang L, Von Laszewski G, Dayal J, Wang F (2010) Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS. In: *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE Computer Society, pp 368–377
5. Von Laszewski G, Wang L, Younge AJ, He X (2009) Power-aware scheduling of virtual machines in dvfs-enabled clusters. In: *2009 IEEE International Conference on Cluster Computing and Workshops*. IEEE, pp 1–10
6. Thakral D, Singh M (2014) Virtualization in cloud computing. *Int J Comput Sci Mob Comput* 3:1262–1273
7. Beloglazov A, Buyya R (2010) Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. *MGC, Bangalore, India*. Copyright 2010 ACM 978-1-4503-0453-5/10/11. T. R. V
8. Kaushar H, Ricchhariya P, Motwani A (2014) Comparison of sla based energy efficient dynamic virtual machine consolidation algorithms. *Int J Comput Appl* 102:0975–8887
9. Zhu F, Li H, Lu J (2012) A service level agreement framework of cloud computing based on the cloud bank model. *Comput Sci Autom Eng (CSAE)*, IEEE 1:255–259
10. Vigiotti F, LDMPA, Batista MD (2014) A green network-aware vms placement mechanism. In: *Proceedings of the IEEE Globecom*. IEEE, Austin
11. Abdelsamea A, El-Moursy AA, Hemayed EE, Eldeeb H (2017) Virtual machine consolidation enhancement using hybrid regression algorithms. *Egypt Inform J* 18(3):161–170
12. Sharma O, Saini H (2016) Vm consolidation for cloud data center using median based threshold approach. *Twelfth Int Multi-Conference Inf Process-2016 (IMCIP-2016)*, *Procedia Comput Sci* 89:27–33
13. Zhou Z, Hu Z, Song T, Yu J (2015) *J Cent South Univ* 22:94–98
14. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurr Comput: Pract Experience (CCPE)* 24:1397–1420
15. Monil MAH, Rahman RM (2016) VM consolidation approach based on heuristics, fuzzy logic, and migration control. *J Cloud Comput* 5:8
16. Tian W, Zhao Y, Zhong Y, Xu M, Jing C (2011) A dynamic and integrated load-balancing scheduling algorithm for cloud datacenters. In: *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*. IEEE, pp 311–315
17. Tang M, Pan S (2015) A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. *Neural Process Lett* 41:211–221
18. Castroa PP, Barreto V, Corrêaa SL, Granville LZ, Cardoso KV (2016) A joint cpu-ram energy efficient and sla compliant approach for cloud datacenters. *Comput Netw* 94:1–13
19. Li H, Zhu G, Cui C, Tang H, Dou Y, He C (2016) *Computing* 98:303–317
20. Farahnakian F, Pahikkala T, Liljeberg P, Plosila J, Hieu NT, Tenhunen H (2016) Energy-aware vm consolidation in cloud data centers using utilization prediction model. *IEEE Trans Cloud Comput*. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=7593250>
21. Moges FF, Abebe SL (2019) *J Cloud Comput* 8(1):2. <https://doi.org/10.1186/s13677-019-0126-y>
22. Wang H, Tianfield H (2018) Energy-aware dynamic virtual machine consolidation for cloud datacenters. *IEEE Access* 6:15259–15273. <https://doi.org/10.1109/ACCESS.2018.2813541>
23. Yousefipour A, Rahmani AM, Jahanshahi M, Energy and cost-aware virtual machine consolidation in cloud computing. *Softw: Pract Experience* 48(10):1758–1774. <https://doi.org/10.1002/spe.2585>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2585>
24. Anandharajan, Bhargavan D, Bhagyaveni AM (2013) Vm consolidation techniques in cloud data center. *J Theor Appl Inf Technol* 53:267–273
25. Tamiz M, Jones D, Romero C (1998) *Eur J Oper Res* 111:569–581
26. Wooldridge JM (2015) *Introductory econometrics: A modern approach*. Nelson Education

27. Rodrigo C, Rajiv R (2011) Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Wiley Press, New York, USA
28. Bowerman B, O'Connell R, Murphree E (2013) Business Statistics in Practice. 7th edn. McGraw-Hill/Irwin
29. Beloglazov A, Buyya R, YLee, Zomaya A (2011) J Adv Comput 82:47–111
30. Zhu F, Li H, Lu J (2012) A service level agreement framework of cloud computing based on the cloud bank model. Comput Sci Autom Eng (CSAE), IEEE 1:255–259
31. Saravanan S, Venkatachalam V, Malligai ST (2015) Optimization of SLA violation in cloud computing using artificial bee colony. Int J Adv Eng 1(3):410–414
32. Han G, Que W, Jia G, Shu L (2016) An efficient virtual machine consolidation scheme for multimedia cloud computing. Sensors 16(2):246
33. Arianyan E, Taheri H, Sharifian S (2016) J Inf Sci Eng 32:1575–1593
34. (2014) The Gaia HPC workload. [http://www.cs.huji.ac.il/labs/parallel/workload/l\\_unilu\\_gaia/index.html](http://www.cs.huji.ac.il/labs/parallel/workload/l_unilu_gaia/index.html). Accessed 8 Dec 2005
35. Feitelson DG, Tsafirir D, Krakov D (2014) Experience with using the parallel workloads archive. J Parallel Distrib Comput 74(10):2967–2982

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---