Analysis, Modeling and Simulation of Workload Patterns in a Large-Scale Utility Cloud

Ismael Solis Moreno, Peter Garraghan, Paul Townend, and Jie Xu, *Member, IEEE*

Abstract—Understanding the characteristics and patterns of workloads within a Cloud computing environment is critical in order to improve resource management and operational conditions while Quality of Service (QoS) guarantees are maintained. Simulation models based on realistic parameters are also urgently needed for investigating the impact of these workload characteristics on new system designs and operation policies. Unfortunately there is a lack of analyses to support the development of workload models that capture the inherent diversity of users and tasks, largely due to the limited availability of Cloud tracelogs as well as the complexity in analyzing such systems. In this paper we present a comprehensive analysis of the workload characteristics derived from a production Cloud data center that features over 900 users submitting approximately 25 million tasks over a time period of a month. Our analysis focuses on exposing and quantifying the diversity of behavioral patterns for users and tasks, as well as identifying model parameters and their values for the simulation of the workload created by such components. Our derived model is implemented by extending the capabilities of the CloudSim framework and is further validated through empirical comparison and statistical hypothesis tests. We illustrate several examples of this work's practical applicability in the domain of resource management and energy-efficiency.

Index Terms—Cloud computing, workload characterization, cloud computing simulation, workload modeling

1 INTRODUCTION

Cloud computing environments are large-scale heterogeneous systems that are required to meet Quality of Service requirements demanded by consumers in order to fulfill diverse business objectives [1]. Such system characteristics result in a diversity of Cloud workload in terms of user behavior, task execution length and resource utilization patterns. In this context, Workload is defined as: "*The amount of work assigned to, or done by, a client, workgroup, server, or system in a given time period*" [12] and consists of two components: tasks and users. Tasks are defined as the basic unit of computation assigned or performed in the Cloud, and a user is defined as the actor responsible for creating and configuring the volume of tasks to be computed. In order to further enhance the effectiveness of managing Cloud computing environments there are two critical requirements. The first is that such environments require extensive and continuous analyses in order to understand and quantify the characteristics of system components. The second is the exploitation of the parameters derived from such analyses in order to develop simulation models which accurately reflect the operational conditions.

Analysis and simulation of Cloud tasks and users significantly benefits both providers and researchers, as it enables a more in-depth understanding of the entire system as well as offering a practical way to improve data center functionality. For providers, it enables a method to enhance resource management mechanisms to effectively leverage the diversity of users and tasks to increase the productivity and QoS of their systems. For example, exploiting task heterogeneity to reduce performance interference of physical servers or analyzing the correlation of failures to resource consumption. For researchers, simulation of Cloud workload enables evaluation of theoretical mechanisms supported by the characteristics of Cloud data centers.

Ideally such simulation parameters are derived from the empirical analysis of large-scale production Cloud data centers. Failure to do so results in misleading assumptions about the degree of workload diversity that exists within the Cloud and the creation of unrealistic simulation parameters. This consequently results in limitations to the usefulness and accuracy of simulation parameters. However, deriving such analyses is challenging in two specific areas. The first and most critical problem is that there are few available data sources pertaining to large-scale production utility Clouds, due to business and confidentiality concerns. This is a particular challenge in academia, which relies on the very few publicly available Cloud tracelogs. The second problem is analysis and simulation of realistic workloads; this is due to the massive size and complexity of data that a typical production Cloud can generate in terms of sheer volume of users and server events as well as recording resource utilization of tasks.

Recently, there has been initial work from the analysis of limited Cloud traces from Google [2], [3] and Yahoo! [4] in an effort to provide mechanisms to analyze and characterize workload patterns. However, such efforts are predominately constrained to traces of short observational periods [5] and coarse-grain statistics [6] which are not

- *The authors are with the School of Computing, University of Leeds, Leeds LS2 9JT, United Kingdom.*
E-mail: {scism, scpmg, p.m.townend, j.xu}@leeds.ac.uk.

Manuscript received 15 Sept. 2013; revised 3 Mar. 2014; accepted 3 Mar. 2014. Date of publication 1 Apr. 2014; date of current version 30 July 2014.
Recommended for acceptance by I. Bojanova, R.C.H. Hua, O. Rana, and M. Parashar.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCC.2014.2314661

sufficient to characterize the workload diversity of Cloud environments. In addition, there have been a number of approaches that analyze the diversity of workload by classifying tasks according to critical characteristics [7], [8], [9]. However, none of these provide a comprehensive study of the diversity of users and tasks, or provide a model containing sufficient details about the model parameters obtained from the analyses in order to be of practical use to researchers.

The objective of this paper is to present an in-depth empirical analysis of workload and its diversity in a large-scale production Cloud computing data center. Additionally, this work aims to provide a validated simulation model that includes parameters of tasks and users to be made available for other researchers to use. The analysis is conducted using the data from the second version of the Google Cloud tracelog [3], [10], which contains over 25 million tasks, submitted by 930 users over the observational period of a month. There are three core contributions within this work:

- An in-depth statistical analysis of the characteristics of workload diversity within a large-scale production Cloud. The analysis was performed over the entire tracelog time span as well as a number of observational periods to investigate patterns of diversity for both users and tasks within the system.
- An extensive analysis of distribution parameters derived from the workload analysis that can be applied to simulation tools by other researchers.
- A comprehensive validation of the simulation model based on empirical and statistical methods. A significant contribution of the simulation model provided is that it does not just replay the data within the tracelog. Instead, it creates patterns that randomly fluctuate based on realistic parameters. This is important in order to emulate dynamic environments and to avoid just statically reproducing the behavior from a specific period of time.

A secondary contribution of this paper is presenting practical applications of the model obtained to identify sources of inefficiencies and enhance resource-management and energy usage in virtualized Cloud environments.

This paper applies the methodology of analysis introduced in our previous approach [9], but is substantially different in a number of ways. First, this paper focuses specifically on a substantial analysis of Cloud diversity for tasks and users. Additionally, we analyze the entire tracelog time span and three additional observational periods, instead of just two days—which limited the original approach's applicability, as it could potentially omit crucial behavior within the overall Cloud environment. Furthermore, extensive analysis and parameter details are provided for user and task distributions.

The remainder of this paper is organized as follows: Section 2 presents the background; Section 3 discusses related work; Section 4 details the methodology used. Section 5 presents the cluster and distribution analysis of task and user diversity. Section 6 presents the validation of the model simulation. Section 7 describes the improvements to the model based on the validation results. Section 8 discusses practical applications of the results obtained within this paper. Sections 9 and 10 discuss the conclusions and further research directions of this work, respectively.

2 Background

2.1 Diversity Patterns in Cloud

According to the NIST [11], the Cloud computing model has the following five essential characteristics: on-demand self-service, resource pooling, broad network access, rapid elasticity and measured service. These characteristics create highly dynamic environments where customers from different contexts co-exist submitting workloads with diverse resource requirements at anytime. Workloads by themselves have properties or attributes that describe their behavior. These attributes are normally expressed by the type and amount of resources consumed and other attributes that could dictate where a specific workload can or cannot be executed. For example, security requirements, geographical location, or specific hardware constraints such as processor architecture, number of cores or Ethernet speed among others described in [13]. As discussed in [14], as more and more customers adopt Cloud platforms to fulfill their IT requirements, Cloud providers need to be prepared to manage highly heterogeneous workloads that are served on the top of shared infrastructure. Workloads can be broadly classified according to the fundamental resources that they consume in terms of CPU, memory and storage-bound workloads [15]. Moreover, depending on the interaction with the end-users, they can also be classified as latency-sensitive and batch workloads [16]. Common examples of workloads running in multi-tenant Cloud data centers according to [17] include Business Intelligence, scientific high-performance computing, gaming and simulation.

2.2 Importance of Workload Models in Cloud

Models abstract reality to aid researchers and providers in understanding system environments in order to develop or enhance such systems. Workload models enable a way to actually study Cloud environments and the effect of workload variability on the performance and productivity of the overall system. Specifically, they support researchers and providers in further understanding the actual status and conditions of the Cloud system and identify Key Performance Indicators (KPI) necessary to improve operational parameters. Such models can be used in a number of research domains including resource optimization, security, dependability and energy-efficiency. In order to produce realistic models, it is critical to derive their components and parameters from real-world production tracelogs. This leads to capturing the intrinsic diversity and dynamism of all co-existing components within the system as well as their interactions. Moreover, realistic workload models enable the simulation of Cloud environments whilst being able to control selected variables to study emergent system-wide behavior, as well as support the estimation of accurate forecasting under dynamic system conditions to improve QoS offered to users. This supports the enhancement of Cloud Management Systems (CMSs) as it allows providers to experiment with hypothetical scenarios and assess their decisions as a result of changes within the Cloud environment (i.e., Capacity planning for increased system size, alteration of the workload scheduling algorithm, performance tradeoffs, and service pricing models).

3 Related Work

The analysis of workload patterns for Cloud computing environments has been addressed previously [5], [6], [7], [8], [9], [18], [19], [20], [21], [22]. In this section, the most relevant approaches are described; their limitations and gaps are also discussed.

Wang et al. [22] present an approach to characterize the workloads of Cloud computing Hadoop ecosystems, based on an analysis of the first version of the Google tracelog [2]. The main objective of this work is to obtain coarse-grain statistical data about jobs and tasks to classify them by duration. This characteristic limits the work's application to the study of timing problems, and makes it unsuitable to analyze other Cloud computing issues related to resource usage patterns. Additionally, the analysis focuses on tasks and ignores the relationship with the users, a crucial component in Cloud workload as discussed previously.

Zhang et al. [5] present a study to evaluate whether the mean values for task waiting time, CPU, Memory, and disk consumption are suitable to accurately represent the performance characteristics of real traces. The data used in their study is not publicly available and consists of the historical traces of six Google compute clusters spanning five days of operation. The evaluation conducted suggests that mean values of runtime task resource consumption is a promising way to describe overall task resource usage. However, it does not describe how the boundaries for task classification were made and how members behave.

Mishra et al. [7] describe an approach to develop Cloud computing workload classifications based on task resource consumption patterns. The analyzed data consist of records from five Google clusters over four days. The proposed approach identifies workload characteristics, constructs the task classification, identifies the qualitative boundaries of each cluster and then reduces the number of clusters by merging adjacent clusters. This approach is useful to create the classification of tasks, but does not perform an analysis of the characteristics of the formed clusters in order to derive a detailed workload model. Finally, it is entirely focused on task modeling, neglecting user patterns.

Kuvulya et al. [6] present a statistical analysis of MapReduce traces. The analysis is based on ten months of MapReduce logs from the M45 supercomputing cluster [4]. Here, the authors present a set of coarse-grain statistical characteristics of the data related to resource utilization, job patterns, and source of failures. This work provides a detailed description of the distributions followed by the job completion times, but only provides very general information about the resource consumption and user behavioral patterns. Similar to [22], this characteristic limits the proposed approach mainly to the study of timing problems.

Aggarwal et al. [8] describe an approach to characterize Hadoop jobs. The analysis is performed on a data set spanning 24 hours from one of Yahoo!'s production clusters comprising of 11,686 jobs. This data set features metrics generated by the Hadoop framework. The main objective of this work is to group jobs with similar characteristics using clustering to analyze the resulting centroids. This work only focuses on the usage of the storage system, neglecting other critical resources such as CPU and Memory.

Our previous work [9] provides an approach for characterizing Cloud workload based on user and task patterns using the second version of the Google tracelog; it presents coarse-grain statistical properties of the tracelog, and classifies tasks and users using statistical mechanisms to select the number of clusters. A concise analysis of the clusters is performed as well as best fit distributions for each. Finally, the derived analysis parameters are simulated and compared against the empirical data for validation. This work has a number of limitations; the analysis performed is confined to only two days as opposed to the entire tracelog time span, resulting in the potential omission of crucial system environment behavior. Also, the cluster analysis and intra-cluster analysis do not contain sufficient detail to quantify the diversity of workload, instead presenting high-level observations. Furthermore, there is insufficient detail about the parameter distributions used; more detail is necessary in order for other researchers to simulate the workload obtained. Finally, the validation of the simulated model against that of the empirical data is based only on a visual match of the patterns from one single execution, and does not consider more rigorous statistical techniques.

From the analysis of the related work it is clear that there are few available production tracelogs to analyze workload patterns in Cloud environments. Previous analyses present gaps that need to be addressed in order to achieve more realistic workload patterns. It is imperative to analyze large data samples as performed by [5], [6], [9]. Small operational time frames as those used in [7], [8], [22] could lead to unrealistic models. Second, analyses need to explore more than coarse-grain statistics and cluster centroids. To capture the patterns of clustered individuals it is also necessary to conduct analysis of the parameters and study the trends of each cluster characteristic. Although previously approaches offer some insights about workload characteristics, they do not provide a structured model which can be used for conducting simulations. Finally, the workload is always driven by the users, therefore realistic workload models must include user behavioral patterns linked to tasks. The approaches previously described completely focus on tasks, neglecting the impact of user behavior on the overall environment workload. A summary of the main characteristics of the related work is presented in Table 1.

4 Methodology

The methodology, analysis and subsequent simulation within this paper was applied to the second version of the Google Cloud tracelog [3], [10] which contains over 12,000 servers, 25 million tasks and 930 users over the period of a month. The tracelog includes detailed data such as submission patterns, resource requests of users and resource consumption of tasks within the system.

The methodology is divided into two distinct steps: The first is defining the model that will be used for simulating the Cloud workload from the derived data set analysis. As stated

TABLE 1
Overview of Related Studies

Authors	Trace size	Analysis Methodology	Analyzed Components	Analyzed Parameters	Workload Model & Validation	Model Parameters
Smith [22]	7 Hours	Coarse-grain	Task	Task duration	No	No
Zhang [5]	30 Days (5 day sample)	Coarse-grain	Task	Task resource usage	No	Yes (Partially)
Mishra [7]	4 Days	Cluster centroids	Task	Task resource usage	No	No
Kavulya [6]	10 Months	Coarse-grain	Task	Task duration	Yes	Yes
Aggarwal [8]	24 Hours	Cluster centroids	Task	Task disk usage	No	No
Solis [9]	29 Days (2 day sample)	Coarse-grain, cluster centroids & intra-cluster analysis	User & Task	User resource estimation & task resource usage	Yes (Partially)	No
Proposed approach	**29 Days**	**Cluster centroids & distribtuion analysis**	**User & Task**	**User resource estimation & task resource usage**	**Yes**	**Yes**

previously, users are responsible for driving the volume and behavior of tasks in terms of requested resources and the volume of task submission. Therefore, three important characteristics that define this behavior within the tracelog are referred to as *parameters* that are fundamental to describe the user behavior: the submission rate α, and requested amount of CPU β and Memory ϕ. The submission rate is the quotient of dividing the number of submissions by the tracelog time span and is presented as task submissions per hour. Requested CPU and memory are represented as normalized resources requested by users taken directly from the task events log within the tracelog.

Tasks are defined by the type and amount of work dictated by users, resulting in different execution length and resource utilization patterns. Consequently, essential parameters that describe tasks are the length χ and the average resource utilization for CPU γ and Memory π. While the length is defined as the total amount of work to be computed, the average resource utilization is the mean of all the consumption measurements recorded in the tracelog for each task.

The Cloud workload can be defined as a set of users with profiles U submitting tasks classified in profiles T, where each user profile u_i is defined by the probability functions of α, β and ϕ, and each task profile t_i by χ, γ and π determined from the tracelog analysis. The expectation $E(u_i)$ of a user profile is given by its probability $P(u_i)$, and the expectation $E(t_i)$ of a task profile is given by its probability $P(t_i)$ conditioned to the probability of $P(u_j)$. The model components and their relationship are formalized in Equations (1) to (6).

$$U = \{u_1, u_2, u_3, \dots, u_i\} \tag{1}$$

$$T = \{t_1, t_2, t_3, \dots, t_i\} \tag{2}$$

$$u_i = \{f(\alpha), f(\beta), f(\phi)\} \tag{3}$$

$$t_i = \{f(\chi), f(\gamma), f(\pi)\} \tag{4}$$

$$E(u_i) = u_i P(u_i) \tag{5}$$

$$E(t_i) = t_i(P(t_i) \mid P(u_j)). \tag{6}$$

The second step of the methodology is to cluster tasks and users composed by the parameters defined for analyzing and creating realistic workload models derived from empirical data. k-means clustering is a popular data-clustering algorithm to divide n observations into k clusters, in which analyzed data sets are partitioned in relation to the selected parameters and grouped around cluster centroids [23].

One critical factor in such an algorithm is determining the optimal number of clusters. For the analysis, we use the statistical method proposed by Pham, et al. [24]. This method, shown in Equations (7) and (8), allows us to select the number of clusters based on quantitative metrics avoiding qualitative techniques that introduce subjectivity. This clustering method considers the degree of variability among all the elements within the derived clusters in relation to the number of analyzed parameters. A number of clusters k is suggested when this variability represented by *f(k)* is lower than or equal to 0.85 according to the observations presented by the authors. S_K is the sum of cluster distortions, N_d is the number of parameters within the population and α_k is the weight factor based on the previous set of clusters.

We run the k-means clustering algorithm for k ranging from 1 to 10. For each value of k we calculate *f(k)* using Equations (7) and (8). Based on the results we were able to formally determine the number of clusters for U and T (Equations (1) and (2)) respectively.

$$f(k) = \begin{cases} 1 & If\ k = I \\ \dfrac{S_k}{\alpha_k S_{k-1}} & If\ S_{k-1} \neq 0,\ \forall k > I \\ 1 & If\ S_{k-1} = 0,\ \forall k > I \end{cases} \tag{7}$$

$$\alpha_k = \begin{cases} 1 - \frac{3}{4N_d} & If\ k = 2,\ and\ N_d > I \\ \alpha_{k-1} + \dfrac{1 - \alpha_{k-1}}{6} & If\ k > 2\ and\ N_a > I. \end{cases} \tag{8}$$

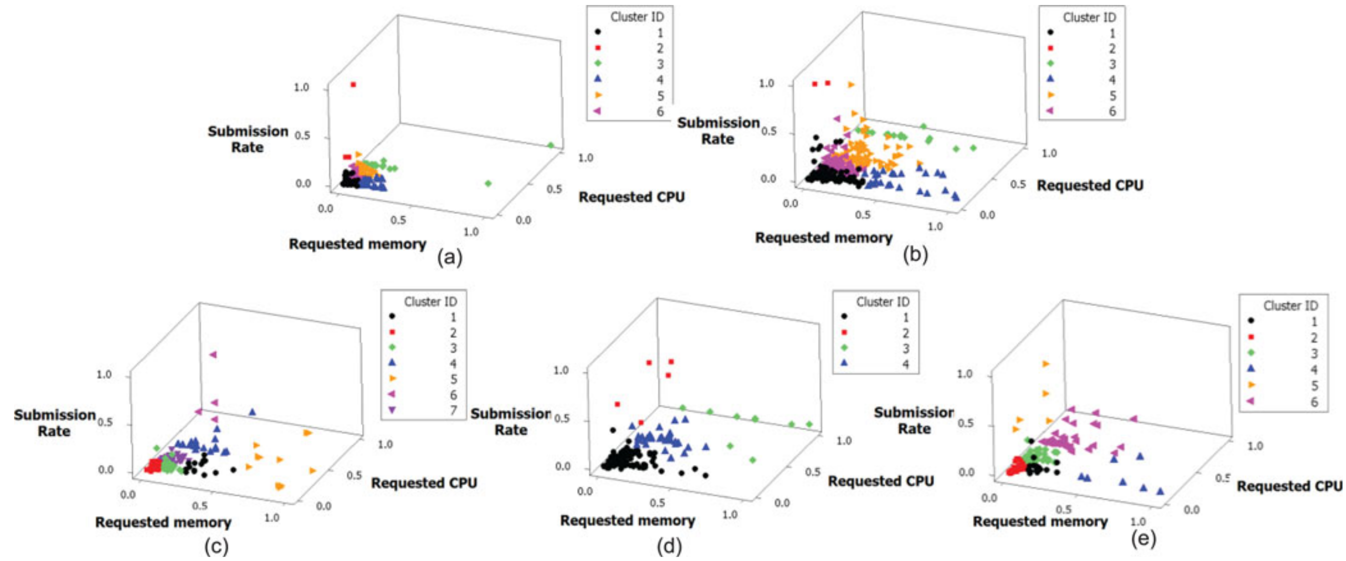


Fig. 1. Clusterization for users (a) entire month (b) entire month (omitting outliers) (c) Day 2, (d) Day 18, and (e) Day 26.

5 Analysis of Diversity

This section presents the analysis of user and task characteristics within the tracelog after performing the k-means clustering algorithm on the entire tracelog timespan as described in Section 4. Specifically, we are interested in quantifying and characterizing the diversity of user and task behavior that exists within the system environment. The analysis is divided into two sections; cluster analysis and distribution analysis.

The cluster analysis discusses the characteristics and behavior of the k-clusters and studies the statistical properties of each parameter within the clusters for users and tasks, including the Mean, Standard Deviation and Coefficient of Variation (*Cv*). The distribution analysis consists of analyzing the inner data distributions for each of the components within each cluster parameter for tasks and users. This required fitting the data to the closest theoretical distribution using a Goodness of Fit (GoF) test to obtain the parameters of their Probabilistic Distribution Functions (PDF). The data of each cluster is fitted to a parametric distribution by using the Anderson-Darling (AD) GoF statistical test. The theoretical distribution with the lowest AD-value is selected to represent the data distribution of each cluster. The objective is to use the PDFs of the parameters in the workload model described in Equations (3) and (4). A number of assumptions for the distribution analysis can be found in [9]. The main alteration to the methodology in order to improve the accuracy of the model is to consider the amount of CPU and memory requested by users instead of the proportions of overestimation and underestimation of resources. This is because the overestimation is an approximated value, whilst the amount of requested resources is a factual value which produces more accurate results.

Moreover, for both the cluster and distribution analysis we have also investigated the variance of task and user clusters and parameters over a number of observational periods. The reason for this is to inspect patterns that exist within the data and to explore the degree of variance over the system lifespan. As a result, this analysis comprises of four observational periods; the entire month trace, Day 2, Day 18 and Day 26. The latter three observational periods were selected for two reasons: First, they represent observational periods of low task length, high submission rate and an average of these two parameters respectively. Second, the periods are temporally far apart, and provide insight into system diversity at different system states.

5.1 Cluster Analysis

Fig. 1 illustrates the k-clusters partitioning that satisfies $f(k) < 0.85$ for users across observational periods. It can be observed from Fig. 1a that the majority of users across the entire month request similar portions of CPU and memory, and exhibit similar submission rates. Furthermore, there are three specific users that have a substantially high submission rate and request larger amounts of CPU and memory as shown in clusters 2 (*U2*) and 3 (*U3*), respectively. When omitting these three users from the cluster analysis in Fig. 1b it is clearer to observe that clusters characteristics are similar across additional observational periods as demonstrated in Figs. 1c, 1d, and 1e, with a substantial amount of users exhibiting a similar submission rates and resource request patterns.

Table 2 shows the statistical properties of each parameter for the defined clusters for the entire tracelog period. It is observable that users follow different resource utilization and submission patterns. For example, *U2* contains 0.71 percent of the total user population and has an incredibly high submission rate in comparison to other clusters. Another example is that *U3* has the highest average requested CPU and memory, but has the lowest submission rate, indicating this type of user infrequently submits more resource intensive tasks.

We observe that requested CPU and memory across most clusters exhibits low variance, with an average *Cv* of 0.42 and 0.79 respectively (*U3* requested memory appears to have higher variance due to the strong influence of three specific users discussed above). The parameter submission

TABLE 2
Statistical Properties of User Clusters for Entire System

		Requested CPU			Requested Memory			Submission Rate (Hourly)		
Cluster	Population %	*Mean*	*Stdev.*	*Cv*	*Mean*	*Stdev.*	*Cv*	*Mean*	*Stdev.*	*Cv*
U1	37.03	0.010	0.004	0.388	0.016	0.013	0.854	34.94	94.00	2.691
U2	0.71	0.016	0.011	0.689	0.019	0.013	0.658	2498.21	2034.6	0.814
U3	6.37	0.135	0.048	0.358	0.094	0.136	1.453	4.71	10.82	2.295
U4	6.37	0.025	0.018	0.718	0.092	0.031	0.342	13.49	19.47	1.444
U5	22.64	0.063	0.011	0.168	0.030	0.020	0.648	73.40	170.44	2.322
U6	26.89	0.032	0.006	0.197	0.014	0.010	0.752	43.63	105.18	2.411

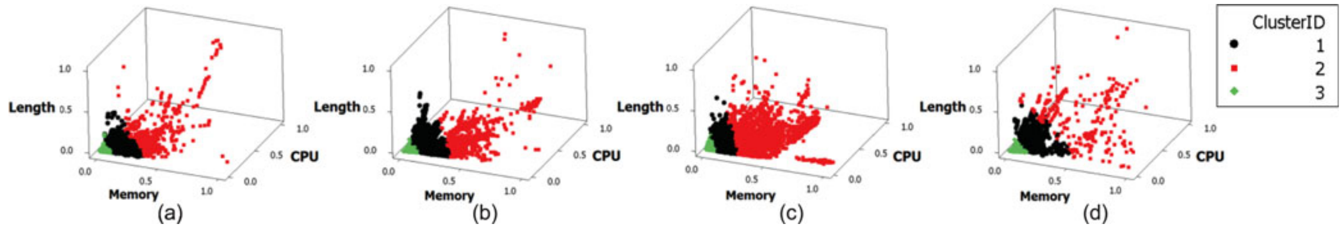


Fig. 2. Clusterization for tasks (a) entire month, (b) Day 2, (c) Day 18, and (d) Day 26.

rate exhibits highly variant behavior across all user clusters, with an average *Cv* of 1.97. *U2* is the only user cluster whose *Cv* submission rate is less than 1, which is most likely due to the cluster population size of 3.

There are three reasons for the above observations. First, as reported in previous works [9] the Cloud data center environment is naturally heterogeneous in workload due to user behavior. Second, requested resources by users are possibly a reflection of the application and system domain boundaries. For example, applications deployed or invoked within the Cloud environment have pre-defined resource requests to meet the demands of user QoS. Third, the submission rate is outside the boundaries of the system and is entirely driven by users; Such behavior is reflective of the definition of Cloud computing, which provides the illusion of infinite resource to users [25], allowing them to submit as many tasks as required without conscious thought about system limitations.

Figs. 2a, 2b, 2c, and 2d presents the k-clusters for tasks across all observational periods, and demonstrates that it was possible to define three clusters for all observational periods where $f(k) < 0.85$. It is observable that the cluster shapes are visually similar across all observational periods, with cluster 3 (*T3*) containing the lowest values for CPU, memory and length while *T2* exhibits more variant behavior. Moreover, *T2* composes less than 2 percent of the total task population and *T3* contains over 70 percent of the task population across all time periods as shown in Table 3. In addition, we observe that the proportions of tasks within the clusters stay relatively constant. In comparison to the heterogeneity of user clusters, task patterns appear to be more uniform across different observational periods.

Table 4 presents the statistical properties of the task parameters length, CPU and Memory utilization for all clusters across the four observational periods. It is possible to make a more balanced comparison of task clusters over different time periods in contrast to user clusters due to the observed stability. Similar to the characteristic of user submission rate, we observe that task length is highly heterogeneous across all clusters and observational periods with an average *Cv* of 2.36, indicating high variation between values. This is due to the same reasons as for the variability that exists for user submission rates; task length is a parameter that is outside the boundaries of the system environment and is entirely dependent on the demands of the user (i.e., Users will execute tasks of different execution length to meet their QoS demands). CPU and memory are less variable due to application domain constraints imposed by the system environment, reflected by an average *Cv* value of 0.93 and 0.83 for CPU and memory utilization respectively.

These results highlight two important findings. First, when quantifying the diversity of the Cloud environment, it appears that parameters that are outside the boundaries of the system environment introduce the highest level of heterogeneity. This is demonstrated by the parameters user submission rate and task execution length exhibiting highly variant behavior in comparison to CPU and memory requests and utilization for users and tasks, respectively. Second, the diversity of workload imposed by these two parameters introduces potential challenges to workload prediction; for this case, where the parameters are highly variable and dynamic, the expiration time of historical data seems to be considerably shorter. Therefore, there exists the need for adaptive and evolving mechanisms that allow providers to obtain more accurate predictions.

5.2 Distribution Analysis

This section studies the data distributions for each cluster parameter for tasks and users. Figs. 3 and 4 present the

TABLE 3
Proportion of Task Clusters Population Percent

Cluster	Month	Day 2	Day 18	Day 26
T1	25.04	15.82	25.61	22.07
T2	1.38	1.8	1.84	1.99
T3	73.59	82.38	72.55	75.94

TABLE 4
Statistical Properties of Task Clusters

		Month			Day 2		
Param	**Clus**	**Mean**	**Stdev.**	***Cv***	**Mean**	**Stdev.**	***Cv***
CPU	*T1*	0.029	0.028	0.966	0.029	0.025	0.862
	T2	0.095	0.088	0.926	0.071	0.071	1
	T3	0.006	0.012	2	0.007	0.012	1.714
Mem	*T1*	0.011	0.01	0.909	0.013	0.01	0.769
	T2	0.049	0.031	0.633	0.047	0.021	0.447
	T3	0.002	0.003	1.5	0.003	0.003	1
Length	*T1*	16,605,683	32,753,760	1.972	9,787,032	1,551,9963	1.586
	T2	123,974,450	250,146,799	2.018	30,932,490	40,683,248	1.315
	T3	739,117	4,056,404	5.488	245,445	655,190	2.669
		Day 18			**Day 26**		
		Mean	**Stdev.**	***Cv***	**Mean**	**Stdev.**	***Cv***
CPU	*T1*	0.028	0.014	0.492	0.006	0.006	1
	T2	0.076	0.051	0.667	0.065	0.04	0.615
	T3	0.005	0.005	0.984	0.026	0.012	0.462
Mem	*T1*	0.009	0.006	0.632	0.001	0.001	1
	T2	0.040	0.017	0.428	0.031	0.018	0.581
	T3	0.001	0.001	1.075	0.009	0.004	0.444
Length	*T1*	41,329,800	103,613,335	2.507	13,669,736	16,538,165	1.21
	T2	117,493,568	388,077,476	3.303	82300581	54,360,253	0.661
	T3	7,658,844	25,068,810	3.273	613,803	1,450,884	2.364

Cumulative Distribution Function (CDF) as an example of the similarity between the theoretical distribution and the empirical data for parameters in *U1* and *T1* obtained as result of the fitting process with the AD test. Table 5 presents the probability of CPU and memory consumption equal to 0 for each task cluster, whilst Table 6 presents the best fit distributions with their corresponding AD value for task and user cluster parameters for the entire tracelog, as well as the parameters required for researchers to simulate the behavior of users and tasks.

Inspecting the different types of distributions and their respective parameter values, we see further statistical evidence of inherit workload diversity within the Cloud environment due to user behavior. For users, it is observable that the best-fit distribution for requested CPU varies between Logistic, three-Parameter Weibull and Loglogistic and Wakeby. Memory is equally heterogeneous, ranging from three-Parameter Lognormal, three-Parameter Loglogistic and Weibull. This gives us insight into the nature of how different users request different resources based on their requirements. For example, the Wakeby distribution used for *U3* and *U5* shows that a large portion of requested CPU is homogenous for those types of users, while *U4* requested CPU and memory is represented with three-Parameter Weibull, signifying that a large portion of users in the analyzed environment request smaller portions of CPU and memory with few users requesting large amounts.

Submission rate distributions predominantly best fit three-Parameter Weibull and 3-Parameter Lognormal. In conjunction with the parameter values, we observe that this data distribution is right-skewed as depicted in Fig. 3c. This indicates that the Cloud environment is composed of a majority of users that submit a small number of tasks and a few users that submit a large proportion of tasks (indeed, there exists one user that submits approximately 18 percent of the total tasks [9].)

For tasks, we observe that CPU and memory utilization across the three clusters follows a number of distributions including General Extreme Value, Weibull, three-Parameter Weibull and three-Parameter Lognormal. This indicates that a high proportion of tasks consume machine resources at lower rates as shown in Fig. 4a and 4b for CPU and memory, respectively. The length of a task shares similar behavior to that of the submission rate of a user, in that it is right-skewed which signifies that most of tasks have a short to medium duration as depicted in Fig. 4c.

Moreover, we contrast the best fit distributions for tasks across different observational periods as shown in Table 7. We present the distribution comparison for tasks, as every observational period shares the same number of task clusters that satisfies $f(k) < 0.85$. An observation of interest is that different days appear to best fit different distributions within that time frame. For example, Day 18 is composed of Loglogistic and Lognormal distributions for all parameters, while Day 26 is predominantly composed of three-Parameter Lognormal and Gamma distributions.

In addition, we observe that task length appears to have exhibit the most consistent distributions characteristics within a selected time observation, predominately following Lognormal and three-Parameter Lognormal. We also observe homogeneity of certain parameters across different observational periods, with the time periods following the same family of distributions for length (Lognormal and Three-Parameter Lognormal).

6 MODEL SIMULATION

In order to characterize and analyze the performance of similar large-scale Cloud data centers under a projected

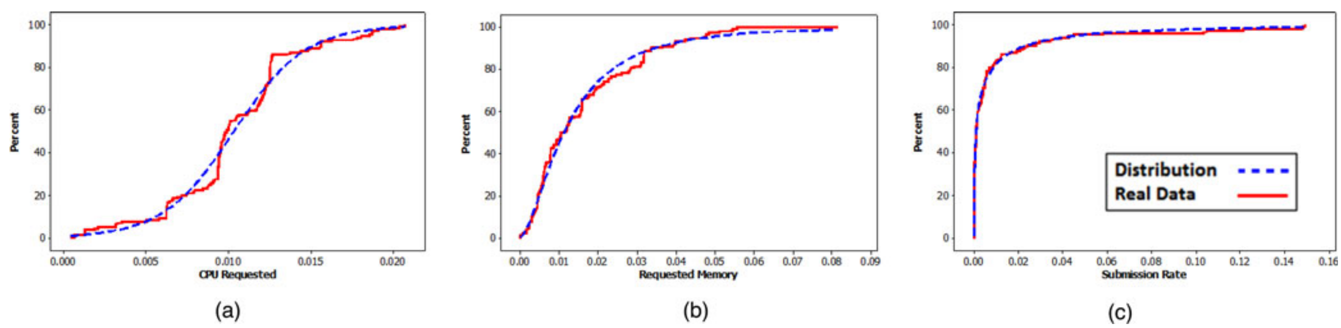


Fig. 3. CDF of user cluster *U1* (a) CPU requested, (b) memory requested, and (c) submission rate.

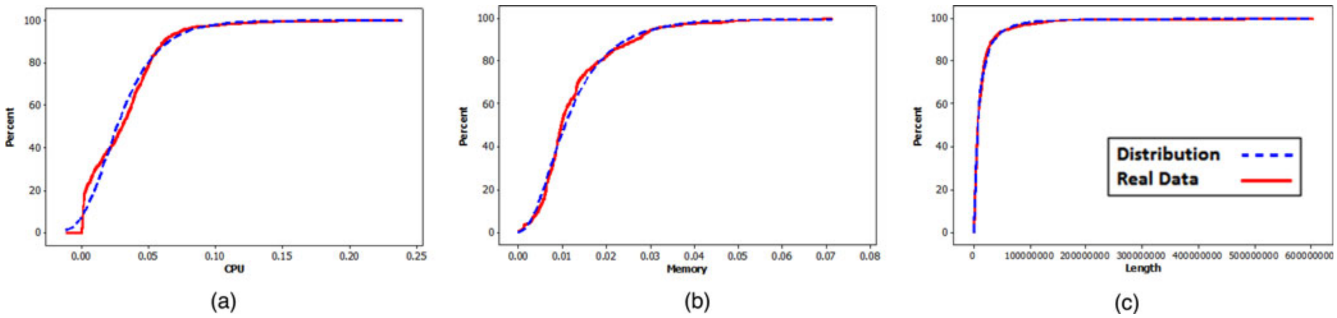


Fig. 4. CDF of task cluster *T1* (a) CPU, (b) memory, and (c) submission rate.

set of operating conditions, we implemented the task and user model parameters described previously as an extension to the CloudSim framework [26], [27], [28], [29]. CloudSim is a Java based framework that enables the simulation of complete Cloud Computing environments [27]. It provides abstraction of all the elements within the Cloud computing model and the interaction among them. However, as with any other simulation software, the quality and accuracy of the results entirely depends on how accurately the introduced parameters reflect the analyzed system in reality. The following subsections describe the implemented workload generator and the conducted simulation validation.

6.1 Workload and Environment Generator

The workload and environment generator is composed of six modules: The Profile Manager, Data center Generator, Customer Generator, Task Generator and Environment Coordinator. The user and task profiles describe respectively the user and task types identified during the clustering process and encapsulate the outlined behavioral patterns derived during the cluster and distribution analysis. The server profiles describe the capacities and characteristics of the data center hosts according to the data within the tracelog. These characteristics as well as the proportion of servers from each type are listed in Table 8.

The profiles manager loads each element description making them available to the generators. The User Generator creates the CloudSim user instances and connects them with a specific profile determined by their associated probabilities as described in Equation (5). The Task Generator creates the CloudSim task instances and connects them with a specific task profile determined by the conditional probability in Equation (6). Each one of the user and task characteristics defined such as submission rate, length and resource consumption described in the model are obtained by sampling the inverse CDFs of the distributions in Equations (3) and (4). Finally, the Environment Coordinator controls the interactions between the three generators and the CloudSim framework that executes the simulation with the created instances.

6.2 Simulation Configuration

We have executed a model simulation of a data center composed of 12,000 servers with 160 customers submitting tasks during 24 hours a total of five iterations. The user and task profiles are configured using the statistical parameters derived for the entire month analysis as described in Tables 5 and 6. The profiles of the simulated servers are outlined from the tracelog as presented in Table 8 where the values of CPU and memory are normalized. The normalization is a scaling relative to the largest capacity of the resource on any server in the trace which is 1.0.

6.3 Simulation Validation

Model validation is defined as the "*substantiation that a computerized model with its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model*" [3]. In the case of the historical data of trace-driven models where the analyst does not have access to the real system or to a different dataset sample from the same system, a common validation technique consists of using a portion of the available data to construct the model and the remaining data to determine whether the model behaves as the real system does. This is typically addressed by sampling the analyzed tracelog where both the input and the actual system response must be collected from the same period of time [31]. According to Sargent [30], there are two basic approaches in comparing the simulation model to the behavior of the real system. The first consists of using graphs to empirically evaluate the outputs and the second involves the application of statistical hypothesis tests to make an objective decision.

To validate our model simulation we use both techniques; the proportions of categorical data such as task, user and server types as well as tasks priorities are contrasted empirically by plotting comparative charts and evaluating the absolute error between the average output from the simulations and the data in the real system. Additionally we analyze the variability of results and their corresponding confidence interval (CI). On the other hand, continuous data such as the user and task resource request and consumption patterns are compared statistically using the Wilcox Mann-Whitney test

TABLE 5
Probability of 0 for Task Resource Utilization

Cluster	CPU p(0)	Memory p(0)
T1	0.121	0.156
T2	0.069	0.073
T3	0.322	0.460

TABLE 6
Best Fit Distribution Parameters of User and Task Clusters for Entire System

	Distribution (AD Value)	Parameters	Distribution (AD Value)	Parameters	Distribution (AD Value)	Parameters
Cluster	Requested CPU		Requested Memory		Submission Rate (Hourly)	
U1	Logistic (1.911)	$\mu = 0.0103$, $\sigma = 0.00216$	3P Lognormal (0.875)	$\mu = -4.355$, $\sigma = 0.802$, $T = -1.607E\text{-}3$	3P Weibull (0.278)	$k = 0.372$, $\lambda = 0.0024$, $T = 3.86E\text{-}7$
U2	Normal (0.431)	$\mu = 0.016$, $\sigma = 0.0109$	Normal (0.191)	$\mu = 0.01916$, $\sigma = 0.01261$	Lognormal (0.471)	$\mu = -0.5679$, $\sigma = 0.7496$
U3	Wakeby (5.620)	$\alpha = 41.734$, $\beta = 334.62$, $\delta = 0.973$, $\xi = 0.00$, $\gamma = 0.0003$	3P Loglogistic (0.876)	$\alpha = 2.156$, $\beta = 0.06334$, $T = -6.1E\text{-}3$	3P Weibull (1.591)	$k = 0.2546$, $\lambda = 0.00007798$, $T = 3.951E\text{-}7$
U4	3P Weibull (0.742)	$k = 1.190$, $\lambda = 0.02372$, T = 2.903E-3	3P Weibull (0.342)	$k = 1.095$, $\lambda = 0.0392$, T = 0.0541	3P Lognormal (0.577)	$\mu = -6.757$, $\sigma = 1.779$, $T = -1.328E\text{-}4$
U5	Wakeby (4.212)	$\alpha = 0.22515$, $\beta = 11.859$, $\gamma = 0.00383$, $\delta = 0.38933$, $\xi = 0.0395$	Weibull (0.629)	$k = 1.570$, $\lambda = 0.03392$	3P Weibull (0.367)	$k = 0.338$, $\lambda = 0.00426$, $T = 3.6E\text{-}7$
U6	3P Loglogistic (2.171)	$\alpha = 5.4452$, $\beta = 0.01896$, $\gamma = 0.01256$	3P Weibull (0.563)	$k = 1.186$, $\lambda = 0.0132$, $T = 1.207E\text{-}3$	3P Weibull (0.523)	$k = 0.034$, $\lambda = 0.0026$, $T = 3.86E\text{-}7$
	CPU		Memory		Length	
T1	Gen Extr Value (5.323)	$\xi = -0.016$, $\sigma = 0.02098$, $\mu = 0.01954$	3P Lognormal (6.946)	$\mu = -4.342$, $\sigma = 0.569$, $T = -2.399E\text{-}4$	Lognormal (12.048)	$\mu = 15.83$, $\sigma = 1.240$
T2	Weibull (16.934)	$k = 0.9594$, $\lambda = 0.09795$	3P Weibull (3.203)	$k = 2.528$, $\lambda = 0.0703$, $T = -9.294E\text{-}3$	3P Loglogistic (10.692)	$\mu = 17.70$, $\sigma = 0.640$
T3	3P Lognormal (15.934)	$\mu = -6.120$, $\sigma = 1.897$, $T = 6.41E\text{-}6$	3P Lognormal (2.756)	$\mu = -5.907$, $\sigma = 0.877$, $T = -2.204E\text{-}4$	3P Lognormal (8.045)	$\mu = 11.87$, $\sigma = 1.855$, $T = -255.9$

(WMW) [32], [33]. WMW is one of the most powerful non-parametric tests for comparing two populations. According to Mauger [34], "*it is based on the test of the null hypothesis that the distributions of two populations, although unspecified, are equal, against the alternative hypothesis that the distributions have the same shape but are shifted, so the outcomes of one population tends to be larger than the othe*r". It is commonly applied instead of the two-sample *t*-test when the analyzed data does not follow a normal distribution as is the case of the outlined user and tasks patterns. Additionally, in order to verify the consistency of the WMW test, we have applied the Fisher's Method [35]; a meta-analysis technique to combine *p*-values from different and independent tests which have the same null hypothesis. The objective is to verify whether the rejections are statistically significant given the variances reported, or are consistent with the results of the other simulations.

6.4 Validation Results

The results from our simulation experiments demonstrate the accuracy of the derived model to represent the operational characteristics of the workload within the Cloud computing data center for the analyzed scenario. Fig. 5 illustrates the proportion of components (users, tasks, task priorities and servers) created during the simulations which are contrasted against the observations from the real system. Comparing the average simulation outputs with the real values, it is possible to observe that simulated proportions of fundamental elements consistently match the proportions of the elements in the actual system. From the detailed results presented in Table 9, it can be observed that while the proportions of tasks do not significantly fluctuate, the proportions of users and servers across different simulation executions present a higher variability. This is mainly produced by a very small population of specific clusters. For example cluster *U2* represents only 0.70 percent of

TABLE 7
Best Fit Distribution Comparison for Task Clusters

Parameter	Cluster	Month	Day 2	Day 18	Day 26
CPU	*T1*	General Extreme Value	Normal	Loglogistic	3-Parameter Lognormal
	T2	Weibull	Weibull	Lognormal	Lognormal
	T3	3-Parameter Lognormal	Lognormal	Lognormal	Gamma
Memory	*T1*	3-Parameter Lognormal	Lognormal	Lognormal	Gamma
	T2	3-Parameter Weibull	Normal	Loglogistic	3-Parameter Gamma
	T3	3-Parameter Lognormal	Lognormal	Loglogistic	3-Parameter Lognormal
Length	*T1*	Lognormal	Lognormal	Loglogistic	3-Parameter Lognormal
	T2	3-Parameter Loglogistic	Lognormal	Lognormal	3-Parameter Lognormal
	T3	3-Parameter Lognormal	Lognormal	Lognormal	3-Parameter Lognormal

TABLE 8
Server Characteristics of Tracelog

Server Type	CPU Capacity	Memory Capacity	Proportion in Datacenter %
S1	0.25	0.2498	1.00
S2	1	1	6.32
S3	1	0.5	0.02
S4	0.5	0.2493	30.70
S5	0.5	0.749	7.96
S6	0.5	0.4995	53.50
S7	0.5	0.9678	0.04
S8	0.5	0.1241	0.41
S9	0.5	0.03085	0.04
S10	0.5	0.06158	0.01

TABLE 9
Simulation Results for Proportions of Cloud Data Center Components

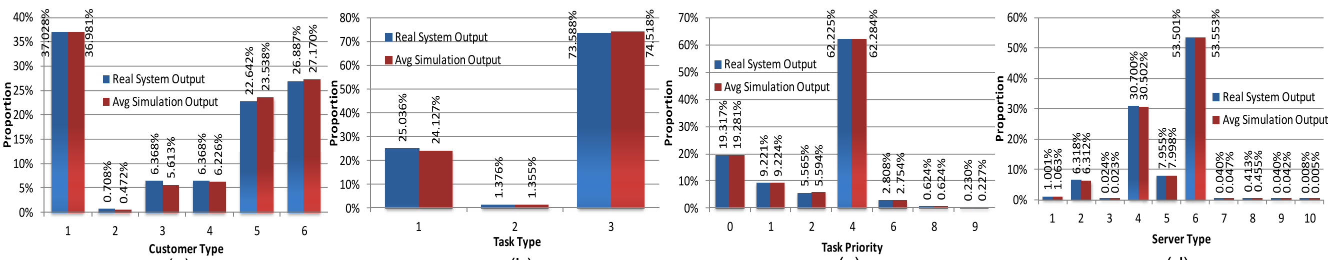
Component	Mean Sim.	Std Dev.	*Cv*	95% CI	A. Error
U1	36.981	2.943	7.958	(34.40,39.56)	0.047
U2	0.472	0.167	35.355	(0.32,0.61)	0.236
U3	5.613	0.840	14.974	(4.87,6.34)	0.755
U4	6.226	1.186	19.053	(5.18,7.26)	0.142
U5	23.538	1.086	4.614	(22.58,24.49)	0.896
U6	27.170	3.574	13.156	(24.04,30.30)	0.238
T1	24.127	1.159	4.803	(23.11,25.14)	0.909
T2	1.355	0.067	4.945	(1.29,1.41)	0.021
T3	74.518	1.120	1.503	(73.53,75.50)	0.930
S1	1.063	0.078	7.364	(0.99,1.13)	0.062
S2	6.312	0.310	4.912	(6.04,6.58)	0.006
S3	0.023	0.007	29.881	(0.01,0.03)	0.001
S4	30.502	0.199	0.651	(30.32,30.6)	0.198
S5	7.998	0.150	1.872	(7.86,8.12)	0.043
S6	53.553	0.282	0.527	(53.30,53.8)	0.053
S7	0.047	0.021	44.821	(0.02,0.06)	0.007
S8	0.455	0.021	4.597	(0.43,0.47)	0.042
S9	0.042	0.017	40.000	(0.02,0.05)	0.002
S10	0.005	0.007	149.071	(0.00,0.01)	0.003

the customers' population but introduce a variability of 35.35 percent. The average *Cv* for tasks is estimated at 0.78 percent against 15.85 and 28.37 percent for customers and servers, respectively. Although the creation of tasks depends on the type of users created, the variability observed in the generation of users is not sufficiently statistically significant to affect the correct proportions of tasks generated during the different simulations. This can be confirmed by analyzing the absolute error between the means of real and simulated populations. For the generation of users, the average absolute error is calculated at 0.39 percent while for tasks and servers is calculated as 0.62 and 0.04 percent, respectively. A breakdown of these statistics with their corresponding 95 percent CI for the obtained mean is presented in Table 9, where it is also observable that in all the cases the difference between the simulated and real system proportions is lower than 1 percent.

In regards to the user and tasks patterns derived from the distribution analysis, we have plotted the empirical CDF of the real data for each cluster parameter and compared them against the empirical CDF of their corresponding simulation outputs. Due to space constraints, we are exemplifying this comparison in Figs. 6 and 7 with the parameters of *U1* and *T3* respectively which represent the largest populations for each element in the tracelog. From these figures, it is noticeable that simulated component patterns are consistent with those observed in the real data. The most significant differences are found in task CPU consumption patterns. This is confirmed in Tables 10 and 11, where the significance values (p-values) obtained by applying WMW test for each simulation output against the real system measurements are listed. Significance values between 0.30 and 0.99 suggest that the simulated parameters for user submission rate, task length and task memory utilization strongly follow the distributions of the real system. In the case of parameters such as CPU requested, memory requested and CPU utilization, 90 percent of the results have a moderate to strong significance value ranging from 0.05 to 0.99. However, there are instances (highlighted in grey) in which there is no statistical evidence to support the WMW null hypothesis.

The results of Fisher's p-value calculation for the clusters with at least one rejection are also presented in Tables 10 and 11. Fisher's p-values $>= 0.05$ support the hypothesis that all separate WMW null hypotheses are true. On the other hand, p-values < 0.05 suggest that the WMW null hypothesis holds in some simulations but not in others. From the total 120 evaluated cases there are six solid rejections which represent an error of 5 percent where the most affected parameter is CPU utilization for *T2* and *T3*.

Regarding to the tasks execution times, it is observed that like the actual system, the simulated tasks follow a lognormal distribution. That is, most of the tasks have a short to medium duration; while a small proportion of tasks have a considerable elapsed time as illustrated Fig. 8. Comparing the average location obtained during the simulations against the location for the data in the tracelog we obtain a percentage of error of 1.27 percent for *T1*, 8.07 percent for *T2* and 5.91 percent for *T3*. This is consistent with the results of WMW test in Table 11, since the Length and CPU utilization is more accurate for *T1* the execution time



(a) (b) (c) (d)

Fig. 5. Comparison of proportions of real and simulated data for (a) users, (b) tasks, (c) task priority, and (d) servers.

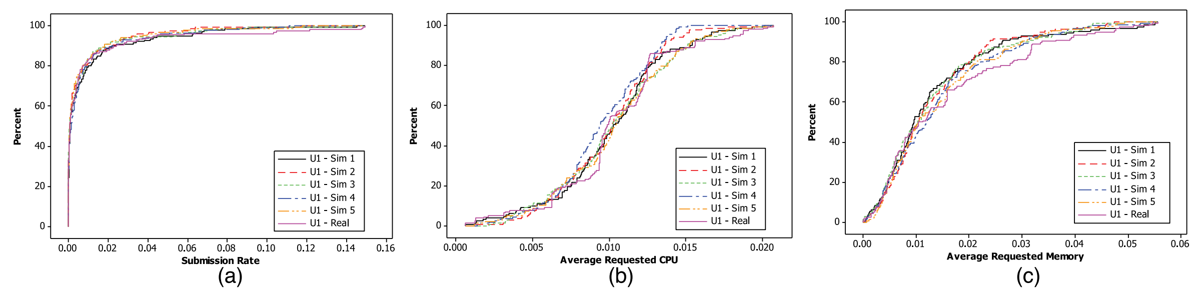


Fig. 6. CDF of user patterns between real and simulated data for *U1* (a) requested CPU, (b) requested memory, and (c) submission rate.

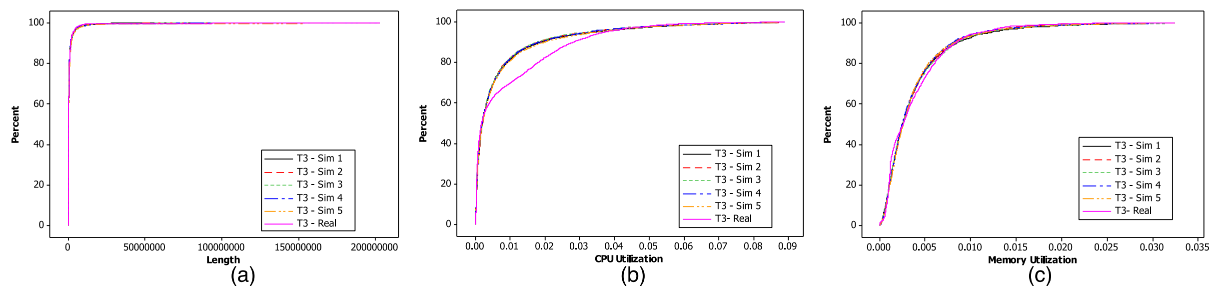


Fig. 7. CDF of task patterns between real and simulated data for *T3* (a) CPU utilization, (b) memory utilization, and (c) length.

TABLE 10
Wilcox Mann-Whitney and Fisher's P-Value Test for User Clusters

Simulation	Submission Rate					Average CPU Requested					Average Memory Requested				
	U1	*U3*	*U4*	*U5*	*U6*	*U1*	*U3*	*U4*	*U5*	*U6*	*U1*	*U3*	*U4*	*U5*	*U6*
Sim 1	0.49	0.82	0.80	0.84	0.84	0.75	0.05	0.55	0.51	0.42	0.30	0.22	0.39	0.37	0.42
Sim 2	0.73	0.53	0.55	0.57	0.68	0.64	0.00	0.78	0.94	0.41	0.54	0.43	0.91	0.44	0.27
Sim 3	0.33	0.36	0.82	0.25	0.56	0.76	0.96	0.61	0.55	0.91	0.26	0.94	0.55	0.50	0.20
Sim 4	0.36	0.85	0.77	0.92	0.97	0.04	0.42	0.71	0.74	0.24	0.84	0.68	0.76	0.99	0.11
Sim 5	0.58	0.98	0.21	0.77	0.82	0.79	0.65	0.76	0.03	0.07	0.83	0.74	0.65	0.04	0.017
Fisher´s p-Value						0.57	0.009		0.43					0.34	0.03

TABLE 11
Wilcox Mann-Whitney and Fisher's P-Value Test for Task Clusters

Simulation	Length			CPU Utilization			Memory Utilization		
	T1	*T2*	*T3*	*T1*	*T2*	*T3*	*T1*	*T2*	*T3*
Sim 1	0.77	0.52	0.43	0.05	0.05	0.05	0.24	0.35	0.68
Sim 2	0.25	0.68	0.19	0.05	0.03	0.07	0.10	0.57	0.55
Sim 3	0.86	0.50	0.72	0.06	0.05	0.00	0.18	0.25	0.83
Sim 4	0.83	0.38	0.99	0.05	0.03	0.02	0.45	0.62	0.88
Sim 5	0.90	0.11	0.91	0.06	0.05	0.15	0.93	0.60	0.40
Fisher's p-Value					0.0005	0.0002			

is closer to that observed in the real system. Conversely, differences in CPU utilization for *T2* and *T3* increase the error in execution time for these two clusters.

7 Improvement of CPU Consumption Patterns

Inaccurate CPU utilization patterns for *T2* and *T3* are result of multimodal data distributions. This makes fitting such data sets with a single theoretical distribution unsuitable and creates significant gaps between the simulated and real data as observed in Fig. 7b. To improve the accuracy of our model, we applied *"multi-peak histogram analysis for region splitting"* [38] and fitted the derived dataset sub-regions to new parametrical distributions. Essentially, the data is ranked and presented in a histogram, which is split based on the lowest points of the different valleys created by the multimodal distribution. To identify the peaks and valleys of a given multimodal data set, we smooth the histogram by applying the LOWESS [36] (Locally-Weighted Scatterplot Smoother) technique using the Minitab statistical package [37]. Then, the derived sub-regions are fitted to new parametrical distributions following the same process described in Section 5.2. Consequently, the CPU utilization patterns of the affected clusters comprise a combination of different distributions which are sampled by the model simulator based on the proportional size of the derived sub-regions. The distribution parameters and sizes of the obtained sub-regions are presented in Table 12.

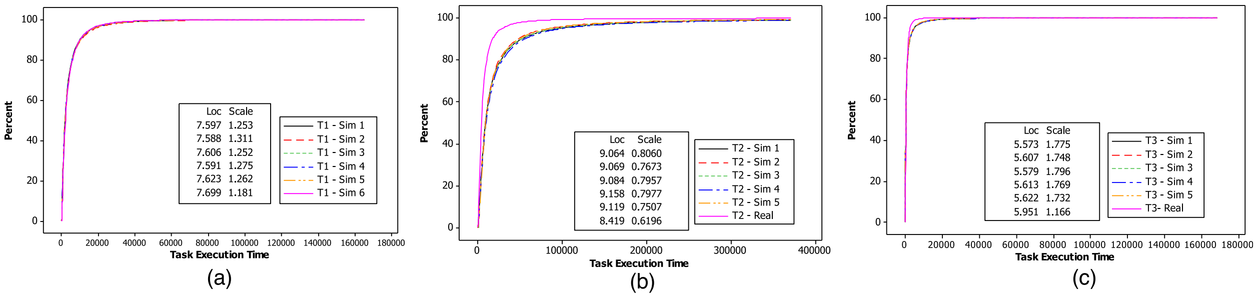


Fig. 8. CDF of task patterns between real and simulated data of task execution time (seconds) for (a) *T1* (s), (b) *T2*, and (c) *T3*.

The results of this process are illustrated in Fig. 9 where it can be observed that the split distributions improve the fitting between the simulated and real datasets. The p-values of the WMW test for both clusters are sufficiently statistically strong to support the equality of patterns. This reduces the error for execution time from 8.07 to 0.42 percent and from 5.91 to 0.13 percent for *T2* and *T3*, respectively.

8 Application of Work

The workload model presented in this paper enables researchers to simulate request and consumption patterns considering parameters and patterns statistically close to those observed from a production environment. This is critical in order to improve resources utilization, reduce energy waste and in general terms support the design of accurate forecast mechanisms under dynamic conditions to improve the QoS offered to customers. Specifically, we use the proposed model to support the design and evaluation of two energy-aware mechanisms for Cloud computing environments.

The first is a resource overallocation mechanism that considers customers' resource request patterns and the actual resource utilization imposed by their submitted tasks. Taking into account these parameters from the proposed model it is possible to estimate the resource overestimation patterns. The main idea is to exploit the resource overestimation patterns of each user type in order to smartly overallocate resources to the physical servers. This reduces the waste produced by frequent overestimations and increases data center availability. Consequently, it creates the opportunity to host additional Virtual Machines in the same computing infrastructure, improving its energy-efficiency [39].

The second mechanism considers the relationship between Virtual Machine interference due to competition for resources and energy-efficiency. The core idea is to co-allocate different types of workloads based on the level of interference that they create, to reduce resultant overhead and thus improve the energy-efficiency of the data center. By considering the resource consumption patterns of each task type we estimate the level of interference and energy-efficiency decrement when they are co-located in a physical server. We classify incoming tasks based on their resource usage patterns, pre-select the hosting servers based on resources constraints, and make the final allocation decision based on the current servers' performance interference level [40]. In both cases the proposed workload model and the parameters derived from the presented analysis are used to emulate the user and tasks patterns required by the energy-aware algorithms. The model integrates the relationship between user demand and the actual resource usage—essential in both scenarios where the aim is to achieve a balance between resource request and utilization in order to reduce resource waste.

Another important benefit of our approach is that as values of customer and task parameters are represented as proportions of resources requested or consumed, they are agnostic of underlying hardware characteristics. Therefore, the proposed model can be used to evaluate the performance of different data center configurations under the same workload.

Furthermore, the comprehensive analysis at cluster and intra-cluster level, the workload model that integrates user and tasks patterns, and the applicability of the model independently of the hardware characteristics represent unique advances in comparison with the related work previously discussed in Section 3. Additionally, the proposed model supports the assessment of resource management mechanisms such as those recently presented in [41], [42] and [43] with parameters from a large-scale production Cloud environment.

TABLE 12
Sub-Regions Distribution Fitting to Improve CPU Utilization for T2 and T3

Cluster	Distributions	Parameters	Region Proportion
T2	Gen. Extreme Value	μ= 0.00593, σ=0.00583, ξ= -0.01822c	22.90%
	3-Parameter Lognormal	μ =-2.9072, σ =0.20621, T =-0.00888	32.44%
	Gen. Extreme Value	μ = 0.11193, σ =0.0242, ξ= -0.20605	16.10%
	3-Parameter Weibull	k=1.3318, λ=0.05718, T=0.16661	28.56%
T3	3-Parameter Lognormal	μ =-7.7268, σ =0.64993, T =-4.9626E-5	45.34%
	3-Parameter Weibull	k =0.89629, λ =0.00364, T=0.00136	28.21%
	3-Parameter Weibull	k =1.1097, λ =0.0152, T=0.01314	26.45%

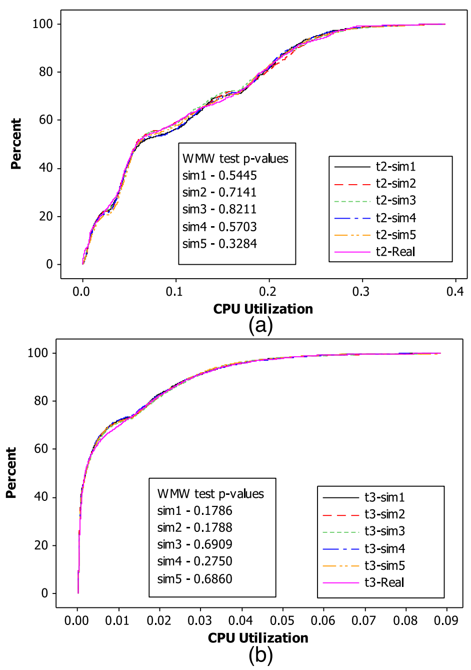


Fig. 9. CPU utilization pattern improvement for (a) *T2* and (b) *T3*.

9 Conclusions

This paper presents an analysis that quantifies the diversity of Cloud workloads and derives a workload model from a large-scale production Cloud data center. The presented analysis and model captures the characteristics and behavioral patterns of user and task variability across the entire system as well as different observational periods. The derived model is implemented using the CloudSim framework and extensively validated through empirical comparison and statistical tests. From the observations presented within this work and the results obtained from the simulations, a number of conclusions can be made. These are as follows:

- *Workload in Cloud data centers is driven not only by tasks characteristics but also by user behavioral patterns.* Related approaches on workload analysis are focused on parameters such as the duration and the resources consumed by tasks. However, as observed from the presented analysis, in some scenarios specific types of users impose a strong influence on the overall Cloud workload. Therefore, comprehensive workload models must consider both tasks and users in order to reflect realistic conditions.
- *User patterns tend to be significantly more diverse than task patterns across different observational periods.* Depending on the type of service offered, providers can control the type of tasks and the environment in which they are running (i.e., SaaS and PaaS). This can create more "*stable*" tasks patterns over the time. On the other hand, user patterns tend to change according to needs derived from their own business objectives which are completely out of the boundaries of Cloud providers. This creates new challenges on workload prediction mechanisms that need to evolve and adapt according to such dynamic characteristics.
- *Describing Cloud analyses is an important first step, but providing the parameters and characteristics derived from these analyses is critical.* This supports the development and validation of simulation models as presented in this work. Such simulations can support the evaluation of new operational policies, new system designs, and support the decision-making process as result of changes in the Cloud environment.
- *Workload models can be exploited to improve diverse and critical operational parameters.* This paper has presented two examples of how the derived model can be used to improve performance and energy-efficiency by exploiting the diversity of users and tasks. In addition, the workload model can be used to improve parameters such as security, dependability, and economics.

10 Future Work

Future research directions includes extending the model to include tasks constraints based on server characteristics; this will allows us to analyze the impact of hardware heterogeneity on workload behavior. Other extensions include analyzing the workload from the jobs perspective specifically modeling the behavior and relationship of users and submitted jobs, accurately emulating and analyzing workload energy consumption and reliability enabling further research into energy-efficiency, resource optimization and failure-analysis in the Cloud environment. Finally, it is important to enable a collaboration link with the CloudSim group in order to integrate the proposed workload generator as an add-in of the current framework implementation allowing it to be made publicly available.

Acknowledgments

The work was supported by CONACyT (No. 213247), the National Basic Research Program of China (973) (No. 2011CB302602), and the UK EPSRC WRG platform project (No. EP/F057644/1).

References

[1] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," *Proc. 10th Int. Conf. Algorithms Archit. Parallel Process.*, 2010, pp. 13–31.
[2] Google. Google Cluster Data V1 (2010). [Online] Available: http://code.google.com/p/googleclusterdata/wiki/TraceVersion1
[3] Google. Google Cluster Data V2 (2011). [Online] Available: http://code.google.com/p/googleclusterdata/wiki/ClusterData2011_1
[4] Yahoo. Yahoo! M45 Supercomputing Project. (2007). [Online]. Available: http://research.yahoo.com/node/1884
[5] Q. Zhang, J. Hellerstein, and R. Boutaba, "Characterizing task usage shapes in Google compute clusters," in *Proc. 5th Int. Workshop Large Scale Distrib. Syst. Middleware*, 2011, pp. 2–8.
[6] S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan, "An analysis of traces from a production MapReduce cluster," in *Proc. IEEE/ACT Int. Conf. Cluster, Cloud Grid Comput.*, 2010, pp. 94–103.
[7] A. K. Mishra, J. Hellerstein, W. Cirne, and C. R. Das, "Towards characterizing cloud backend workloads: Insights from Google compute clusters," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 37, pp. 34–41, 2010.
[8] S. Aggarwal, S. Phadke, and M. Bhandarkar, "Characterization of Hadoop jobs using unsupervised learning," in *Proc. 2nd Int. Conf. Cloud Comput. Technol. Sci.*, 2010, pp. 748–753.

[9] I. Solis Moreno, P. Garraghan, P. Townend, and J. Xu, "An approach for characterizing workloads in Google cloud to derive realistic resource utilization models," in *Proc. IEEE Int. Symp. Serv. Oriented Syst. Eng.*, 2013, pp. 49–60.

[10] C. Reiss, J. Wilkes, and J. Hellerstein, "Google Cluster-Usage Traces: Format + Schema," Google Inc., Mountain View, CA, USA, White Paper, 2011.

[11] P. Mell and T. Grance, "The NIST definition of cloud computing," *NIST Spec. Publication*, vol. 800, p. 145, 2011.

[12] M. A. El-Refaey and M. A. Rizkaa, "Virtual systems workload characterization: An overview," in *Proc. IEEE Int. Workshops Enabling Technol. Infrastructures Collaborative Enterprises*, 2009, pp. 72–77.

[13] B. Sharma, V. Chudnovsky, J. Hellerstein, R. Rifaat, and C. R. Das, "Modeling and synthesizing task placement constraints in Google compute clusters," in *Proc. ACM Symp. Cloud Comput.*, 2011, pp. 1–14.

[14] J. Zhan, L. Wang, W. Shi, S. Gong, and X. Zang, "PhoenixCloud: Provisioning resources for heterogeneous workloads in cloud computing," arXiv preprint arXiv:1006, vol. 1401, 2010.

[15] V. Vasudevan, D. Andersen, M. Kaminsky, L. Tan, J. Franklin, and I. Moraru, "Energy-efficient cluster computing with FAWN: Workloads and implications," in *Proc. Int. Conf. Energy-Efficient Comput. Netw.*, 2010, pp. 195–204.

[16] T. N. B. Doung, X. Li, R. S. M. Goh, X. Tang, and W. Cai, "QoS-aware revenue-cost optimization for latency-sensitive services in IaaS clouds," in *Proc. IEEE/ACM Int. Symp. Distrib. Simul. Real Time Appl.*, 2012, pp. 11–18.

[17] IBM, "Get more out of cloud with a structured workload analysis," White Paper IAW03006-USEN-00, 2011.

[18] A. Bahga and V. K. Madisetti, "Synthetic workload generation for cloud computing applications," *J. Softw. Eng. Appl.*, vol. 4, pp. 396–410, 2011.

[19] A. Beitch, B. Liu, T. Yung, R. Griffith, A. Fox, and D. A. Patterson, "Rain: A workload generation toolkit for cloud computing applications," Elect. Eng. Comput. Sci. Univ. California, Berkeley, CA, USA, White Paper UCB/EECS-2010-14, 2010.

[20] Y. Chen, A. S. Ganapathi, R. Griffith, and R. H. Katz, "Analysis and lessons from a publicly available Google cluster trace," USA, EECS Dept., Univ. California, Berkeley, CA, UCB/EECS-2010-95., Jun. 2010.

[21] J. W. Smith and I. Sommerville, "Workload classification & software energy measurement for efficient scheduling on private cloud platforms," presented at the *ACM SOCC*, Cascais, Portugal, 2011.

[22] G. Wang, A. R. Butt, H. Monti, and K. Gupta, "Towards synthesizing realistic workload traces for studying the Hadoop ecosystem," in *Proc. IEEE Int. Symp. Modeling, Anal. Simul. Comput. Telecommun. Syst.*, 2011, pp. 400–408.

[23] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, pp. 645–678, 2005.

[24] D. T. Pham, S. S. Dimov, and C. D. Nguyen, "Selection of K in K-means clustering," *Proc. Inst. Mech. Eng., Part C: J. Mech. Eng. Sci.*, vol. 219, pp. 103–119, 2005.

[25] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley view of cloud computing," Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, Feb. 2009.

[26] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *Proc. Intl Conf. High Perform. Comput. Simul.*, 2009, pp. 1–11.

[27] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. Practice Experience*, vol. 41, pp. 23–50, 2010.

[28] S. K. Garg and R. Buyya, "NetworkCloudSim: Modelling parallel applications in cloud simulations," in *Proc. IEEE Intl. Conf. Utility Cloud Comput.*, 2011, pp. 105–113.

[29] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A CloudSim-based visual Modeller for analysing cloud computing environments and applications," in *Proc. IEEE Intl. Conf. Adv. Inf. Netw. Appl.*, 2010, pp. 446–452.

[30] R. G. Sargent, "Verification and validation of simulation models," in *Proc.Conf, Winter Simul.*, 2010, pp. 166–183.

[31] O. Balci and R. G. Sargent, "Some examples of simulation model validation using hypothesis testing," *Proc. Conf. Winter Simul.*, vol. 2, pp. 621–629, 1982.

[32] D. Brown and P. Rothery, "Models in biology: Mathematics, statistics and computing," *Proc. 14th Conf. Winter Simul*, 1993.

[33] A. Gold, "Understanding the Mann-Whitney Test," *J. Property Tax Assessment Admin.*, vol. 4, pp. 55–57, 2007.

[34] D. T. Mauger and G. L. Kauffman Jr, "82 - statistical analysis—specific statistical tests: Indications for use," *Surgical Research* W. S. Wiley and W. W. Douglas, eds., San Diego, CA, USA, Academic, 2001, pp. 1201–1215.

[35] D. A. S. Fraser, A. K. M. Saleh, and K. Ji, "Combining p-values: A definitive process," *J. Statist. Res.*, vol. 44, pp. 15–29, 2010.

[36] D. Borcard and P. Legendre, "Exploratory data analysis,"in *Numerical Ecology*, New York, NY, USA, Springer, pp. 9–30, 2011.

[37] Minitab, Version: Release 16 (2010). MINITAB statistical software [Online]. Available: http://www.minitab.com.

[38] S. Pal and P. Bhattacharyya, "Multipeak histogram analysis in region splitting: A regularization problem," in *Proc. IEEE Comput. Digit. Tech.*, 1991, vol. 138, pp. 285–288.

[39] I. Solis Moreno and J. Xu, "Neural network-based overallocation for improved energy-efficiency in real-time cloud environments," in *Proc. IEEE Int. Symp. Object/Compon./Serv.-Oriented Real-Time Distrib. Comput.*, 2012, pp. 119–126.

[40] I. Solis Moreno, R. Yang, J. Xu, and T. Wo, "Improved energy-efficiency in cloud datacenters with interference-aware virtual machine placement," in *Proc. IEEE Int. Symp. Auton. Decentralized Syst.*, 2013, pp. 1–8.

[41] X. Lu, H. Wang, J. Wang, J. Xu, and D. Li, "Internet-based virtual computing environment: Beyond the data center as a computer," *Future Generation Comput. Syst.*, vol. 29, pp. 309–322, 2013.

[42] M. Kesavan, I. Ahmad, O. Krieger, R. Soundararajan, A. Gavrilovska and K. Schwan, "Practical compute capacity management for virtualized datacenters," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 88–100, Jan.-Jun. 2013.

[43] J. Doyle, R. Shorten, and D. O'Mahony, "Stratus: Load balancing the cloud for carbon emissions control," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 116–128, Jan.-Jun. 2013.

Ismael Solis Moreno received the PhD degree from the University of Leeds, and the MSc degree from the CENIDET, Mexico. He has worked as a researcher for the Mexican Electrical Research Institute. His current work on energy-efficient Cloud computing is funded by the CONACyT. He has received best paper awards at IEEE SOSE-2013 and IEEE ISADS-2013.

Peter Garraghan received the BSc degree from Staffordshire University, United Kingdom, and is currently working toward the PhD degree in the Distributed Systems and Service Group at the University of Leeds. He has worked as an IT specialist at HP, Germany. His current research on Cloud computing and energy-aware dependability is funded by the UK EPSRC WRG platform project. He has received an award for best conference paper at the IEEE SOSE-2013.

Paul Townend is a research fellow in the School of Computing, University of Leeds. He has been a lead researcher on major projects dealing with HPC, decision support, large-scale simulations, Cloud computing, and dependable and secure systems. He has extensive experience in collaborating with academia, local government, and industry, and has authored and coauthored more than 40 international publications.

Jie Xu is a chair of computing and head of the I-CSS at the University of Leeds. He is the director of the UK EPSRC WRG e-Science Centre. He is also a guest professor of Beihang University, China. He has published more than 300 academic papers in areas related to dependable distributed systems and has industrial experience in designing and implementing large-scale networked computer systems. He has led or coled many research projects to the value of more than $30M. He is a member of the IEEE.