Solis Moreno, I, Garraghan, P, Townend, PM and Xu, J (2013) *An Approach for Characterizing Workloads in Google Cloud to Derive Realistic Resource Utilization Models.* In: Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on. UNSPECIFIED. IEEE , 49 - 60 (12). ISBN 978-1-4673-5659-6

2013 IEEE Seventh International Symposium on Service-Oriented System Engineering

An Approach for Characterizing Workloads in Google Cloud to Derive Realistic Resource Utilization Models

Ismael Solis Moreno, Peter Garraghan, Paul Townend, Jie Xu
School of Computing
University of Leeds
Leeds, UK
{scism, scpmg, p.m.townend, j.xu} @ leeds.ac.uk

***Abstract*- Analyzing behavioral patterns of workloads is critical to understanding Cloud computing environments. However, until now only a limited number of real-world Cloud datacenter tracelogs have been available for analysis. This has led to a lack of methodologies to capture the diversity of patterns that exist in such datasets. This paper presents the first large-scale analysis of real-world Cloud data, using a recently released dataset that features traces from over 12,000 servers over the period of a month. Based on this analysis, we develop a novel approach for characterizing workloads that for the first time considers Cloud workload in the context of both user and task in order to derive a model to capture resource estimation and utilization patterns. The derived model assists in understanding the relationship between users and tasks within workload, and enables further work such as resource optimization, energy-efficiency improvements, and failure correlation. Additionally, it provides a mechanism to create patterns that randomly fluctuate based on realistic parameters. This is critical to emulating dynamic environments instead of statically replaying records in the tracelog. Our approach is evaluated by contrasting the logged data against simulation experiments, and our results show that the derived model parameters correctly describe the operational environment within a 5% of error margin, confirming the great variability of patterns that exist in Cloud computing.**

Keywords—Cloud computing workload patterns; MapReduce analysis; resource usage patterns; workload characterization.

I. INTRODUCTION

Gaining an understanding of Cloud system environments is increasing in importance as well as complexity due to a Cloud's ability to elastically scale-up and down provisioned resources on-demand. Additionally, such systems need to meet expected Quality of Service (QoS) requirements to fulfill the diverse business objectives demanded by consumers [1]. This diversity of objectives results in a complex workload imposed by users' behavior and task resource consumption patterns. As a consequence, it is a crucial requirement to characterize the workloads running within a Cloud environment. In this context, we define workload as the amount of work computed or processed within the Cloud datacenter that is mainly driven by two principal elements: tasks and users. A task is defined as the basic unit of computation assigned or performed in the Cloud e.g. *MapReduce* operations. A user is defined as the actor responsible for creating and configuring the volume of tasks to be computed.

As a result of business and confidentiality concerns, there has been a lack of available data from real Cloud operational environments to analyze. Recently, due to the publication of limited traces from Google [2] and Yahoo! [3], there has been an increasing effort to provide mechanisms to characterize workload dynamicity. However first efforts were strongly constrained by traces with very short observational periods [4]. Analyses and methodologies derived from just a few hours of production data are diminished by the uncertainty generated from the lack of realistic scenarios. Others that have had access to private large datasets introduce methodologies of analysis based on coarse-grain statistics [4, 5], which are appropriate to reveal general characteristics of the operational environment but not sufficient to describe and characterize the workload diversity that is generated in Cloud environments. Finally, more recent approaches [6, 7] have attempted to capture this diversity by classifying the different types of tasks discovered in the data.

Presently there is a lack of research that deeply analyses and models the relationship between users, tasks and their associated characteristics as an integrated concept of a workload in the Cloud environment. This is an extremely important factor to consider, as the volume and the behavior of tasks that exist within Cloud environments are driven by the demand of the users, and the resource consumption of these workloads is dependent on the users' estimation patterns and are subject to change over time. The establishment of methodologies to derive realistic models for highly diverse and dynamic environments such as Cloud computing datacenters is critical, as Cloud providers are required to understand the supply and demand of computational resources in order to outline inefficiencies and set up improvement goals. The methodologies and derived models need to identify general common characteristics but also specific behavioral patterns across different analyzed periods in order to capture such dynamicity.

The aim of this paper is to present a novel approach for characterizing Cloud datacenter workloads that creates a reusable generation model based on real operational data. To tackle this issue, we have analyzed the latest version of the Google Cloud tracelog [8], which spans a period of 1 month and contains information about over 25 million tasks and 925 users. The proposed approach differs from previous work in three main aspects. Firstly, it considers workload as a compound element not only integrated by task, but also by user behavioral patterns and the relationship between the two elements. This is critical to analyze how overall datacenter

978-0-7695-4944-6/12 $26.00 © 2012 IEEE
DOI 10.1109/SOSE.2013.24

utilization levels are affected by users' behavior and how datacenter efficiency can be improved as long as the Quality of Service offered to the users is maintained. Secondly, it provides a mechanism to create patterns that randomly fluctuate based on realistic parameters instead of statically replaying the records in tracelog. This is important to emulate dynamic environments and analyze their intrinsic variability of patterns. Finally, to feed the proposed model not only coarse-grained analysis was conducted but also a detailed study of user and task pattern distributions was performed. This allows extracting the general characteristics from the entire tracelog and modeling the behavioral patterns from specific scenarios at the same time. Initial simulation experimentation demonstrate that the derived model parameters reflect the real measurements in the tracelog within a 5% of error margin in most of the cases, and confirms the diversity of patterns that exist in Cloud computing environments for both tasks and users.

The primary contribution of this paper is to provide a reusable approach for characterizing the Cloud workload based on the patterns of both users and tasks. The objective is to derive models which can be used by providers and other researchers to capture the behavior of Cloud environments and support the analysis of Cloud computing problems based on realistic workload characteristics from large-scale commercial environments.

A secondary contribution of significance in this paper is the description of the statistical properties of the Google tracelog that will help further the research community's understanding of the utilization and performance of a large-scale commercial Cloud environment. Both of these contributions are essential to address further research challenges related to the improvement of Cloud datacenter operations.

The rest of this paper is organized as follows: Section II presents the background. Section III discusses related work. Section IV describes the analyzed dataset and the assumptions made. Section V presents the proposed workload model specification. Section VI presents the approach to derive the model parameters. Section VII presents the model evaluation and discusses the obtained results. Section VIII describes examples of the model's applicability. Finally, Section IX and X discuss the conclusions and further research directions respectively.

II. BACKGROUND

A. MapReduce and Cloud Computing

Defined by Google [9] "*MapReduce is a programming model and associated implementation for processing and generating large datasets*". It is commonly used for dividing work across a large distributed system since it enables automatic parallelization and distribution of large-scale computations. In this context, Cloud computing offers a unique opportunity for batch-processing and analyzing terabytes of data that would take considerable time to finish. Most Cloud providers such as Google, Amazon and Yahoo! have adopted MapReduce to build multi-tenant computing environments.

B. Available Cloud Computing Tracelogs

At the present time, there are a limited number of real-world Cloud computing tracelogs available. Tracelogs that are of sufficient observational period and system size to perform in-depth analyses are even more limited. This is mostly due to the business and confidentiality concerns of users and providers in commercial Clouds. These limitations restrict research regarding the Cloud model, as researchers face increased difficulty in justifying their work without realistic data and models derived from production environments. Recently, Google has contributed by releasing two versions of tracelogs from their Hadoop MapReduce clusters. The first version spans over a period of 7 hours with normalized processor and Memory usage metrics collected every 5 minutes. The trace describes the resource consumption for approximately 176,174 tasks grouped in 9,174 jobs. The trace has been public from December 2009 and is available in [2]. The second version of this trace spans 30 days and 12,583 servers in operation, providing information on 25 million tasks grouped in 650,000 jobs. The work presented in this paper is based on this tracelog, that has been public since November 2011 and is available in [8]. Additionally more information about the data structure, monitoring, and normalization process can be found in [10].

The other vendor that has provided data about their Cloud computing clusters is Yahoo! which was made available for selected universities from their M45 Hadoop cluster [3]. There is limited detail about the type and structure of the data provided except for [5]. Here, it is mentioned that M45 data spans a period of 10 months and comprises of 171,079 Hadoop jobs including large-scale graph mining, text and Web mining, and large-scale computer graphics.

C. Importance of Workload Models in Cloud Computing

The importance of workload models in Cloud computing cannot be understated. For researchers, they provide a way to not only simulate Cloud environments but also to manipulate the workload variables within those environments. Furthermore, it is of great benefit if these models are derived from production tracelogs, as this enables subsequent research involving simulation to be based on realistic scenarios. Additionally, these models can be used to support a wide variety of research domains, including resource optimization, energy-efficiency, and failure-analysis. For providers, workload models enable them to simulate their Cloud environments whilst being able to control the variables to study emergent system-wide behavior. Such models support the estimation of accurate forecasts under dynamic conditions to improve the QoS offered to customers.

III. RELATED WORK

The analysis of behavioral patterns and deriving models for Cloud Computing environments has been addressed previously [11-14]. In this section, the most relevant approaches are described. Furthermore, their flaws and gaps are also discussed.

Wang, et al. [15] present an approach to characterize the workloads of Cloud Computing Hadoop ecosystems, based on an analysis of the first version of the Google tracelogs [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 model focuses on tasks and ignores the relationship with the users.

Mishra, et al. [6] describe an approach to construct Cloud computing workload classifications based on task resource consumption patterns. It is applied to the first version of Google tracelogs [2]. In general terms, the proposed approach identifies the workload characteristics, constructs the task classification, identifies the qualitative boundaries of each cluster, and then reduces the number of clusters by merging adjacent clusters. The approach presented is useful to create the classification of tasks; however it does not perform intra-cluster analysis to derive a detailed workload model. Finally, it is entirely focused on task modeling, neglecting the user patterns which are as important as the tasks in the overall workload model.

Kuvalya, et al. [5] present a statistical analysis of MapReduce traces. The analysis is based on ten months of MapReduce logs from the M45 supercomputing cluster [3]. 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 the user behavioral patterns. Similar to [15], this characteristic limits the proposed approach mainly to the study of timing problems.

Aggarwal, et al. [7] describe an approach to characterize Hadoop jobs. The analysis is performed on a dataset spanning 24 hours from one of Yahoo!'s production clusters comprising of 11,686 jobs. This dataset features metrics generated by the Hadoop framework. The main objective is to group jobs with similar characteristics using clustering to analyze the resulting centroids. This work is only focused on the usage of the storage system, neglecting other critical resources such as CPU and Memory.

From the analysis of the related work it is clear that there are limited production tracelogs to analyze the workload patterns in Cloud environments available. Previous analyses present some gaps that need to be addressed in order to achieve more realistic workload patterns. Firstly, it is imperative to analyze large data samples as performed by [4, 5]. Small operational time frames as those used in [6, 7, 15] could lead to unrealistic models. Secondly, the analysis needs to explore more than coarse-grain statistics and cluster centroids. To capture the patterns of the clustered individuals it is also necessary to conduct intra-cluster analysis and study the trends of each cluster characteristic. Finally, the workload is always driven by the users, therefore realistic workload models must include user behavioral patterns linked to tasks.

IV. Dataset Overview

As mentioned previously, the data used in this work was collected from the second version of the Google MapReduce Cloud tracelog that spans a period of approximately one month [8, 10]. The log contains tens of millions of records for tasks, jobs, and server events. Furthermore, it provides the normalized CPU, Memory, and disk utilization per task in a timestamp every 5 minutes. The majority of our analysis is focused on two data structures: "*tasks events*" and "*task resource usage*". While the former provides information about the submission times and the link between users and tasks, the latter provides detailed information about the consumption of resources. The total size of the data is approximately 250GB.

A. Dataset Assumptions

In order to produce a fair and comprehensive analysis, it is necessary to rely on realistic assumptions to overcome the lack of context information and normalized data. These assumptions are listed and justified as follows:

- *A task is considered the basic element that consumes resources.* As the resource consumption is logged by tasks, the analysis is focused on them and jobs are considered just as grouping element.
- *The task duration is considered from the last submission event to successful completion.* This is because the total execution time is normally affected by other factors, such as resubmission events caused by task evictions and failures.
- *Task length is calculated based on the duration and the average CPU utilization and is measured in Millions of Instructions (MI).* Duration depends on the architectural characteristics of the server where the task is allocated [16]; describing the task in terms of length in MI allows us to perform an architecture-agnostic workload analysis.
- *To calculate the task length we consider the processing capacity of the Primergy RX200 S7 architecture, as it is described in SpecPower2008 benchmark results* [17]. Because the data that describes the servers' capacity is masked in the tracelog we are assuming characteristics from real operational systems based on the provided capacities.
- *Tasks that start before or finish after the tracelog time frame are not considered in the analysis.* It is impossible to derive the length parameter for "*incomplete-tasks*" where the start or finish time is unknown.
- *Every time a task fails, is evicted or killed we assume that it is restarted from the beginning.* A task failure is an interruption on a running task, requiring the system to re-execute the interrupted task [10, 18].
- *Disk usage is not considered due to uniform usage patterns.* As observed in the data, 98% of tasks present similar disk usage patterns [6, 10] which makes this dimension unsuitable for classification purposes.

B. *General Tracelog Statistical Analysis*

An overview of the statistics derived from the trace based on the prior assumptions is presented in Table I. One interesting observation about the data from this table is the average number of tasks per user, which is 3,981.06. This number is high and misleading due to the non-uniform distribution of submitted tasks per user as shown in Fig.1. A small portion of the users constitute a significant proportion of the submitted tasks, while the majority of users individually contribute less than 0.1 % of the total number.

TABLE I. DATASET OVERVIEW.

Trace span	29 Days	**Num of servers**	12,532
Num of tasks	17,752,951	**Avg tasks / day**	612,170.72
Num of users	430	**Avg users / day**	153.20
Avg task length	61,575,043.48	**Avg tasks / user**	3,981.06

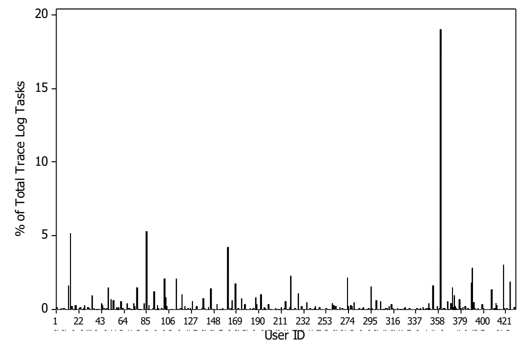


Figure 1. Distribution of Completed Tasks per User.

We also observe that the number of tasks per day varies significantly from the average, as is shown in Fig. 2(a); ranging from 313,927 to 950,449. In contrast, the variation in the number of users per day shown in Fig. 2(b) is more discrete. This suggests a loose correlation between the number of users and the number of tasks submitted per day at a coarse-grain analysis. Fig. 2(c) depicts the average length of tasks on a daily basis. An observation of interest is that when the task length is contrasted with the number of submissions per day, there is no clear correlation between them. The statistical properties of the average CPU and Memory utilized by a task per day are shown in Fig. 2(d). It is observed that resource utilization levels are very similar across all the analyzed days. This suggests a strong correlation to the number of users, but a very loose one to the number of tasks and their average length. This presentation of the statistical properties of the tracelog is important for two reasons. Firstly, it shows that the resource consumption and task completion behavior is not homogenous across the observation period. Secondly, at such a high level of analysis, there appear to be loose correlations between tasks, users, task length and task resource consumption. These observations are important, as they demonstrate that coarse-grained analysis is insufficient for modeling the behavior of tasks and users realistically.

V. MODEL DESCRIPTION

To provide a more precise workload description and reduce the gaps found during the coarse-grained analysis, our proposed workload model comprises of the concepts of users and tasks and their relationship in matters of amount work and utilization of resources.

Users are responsible for driving the volume and behavior of tasks based on the amount of resources requested for their execution. Therefore, three important characteristics that we will refer to as dimensions are fundamental to describe the users' shape: the submission rate (α) and the estimation ratios for CPU (β) and Memory (ϕ). While the submission rate is the quotient of dividing the number of submissions by the time span, the resource estimation ratio is

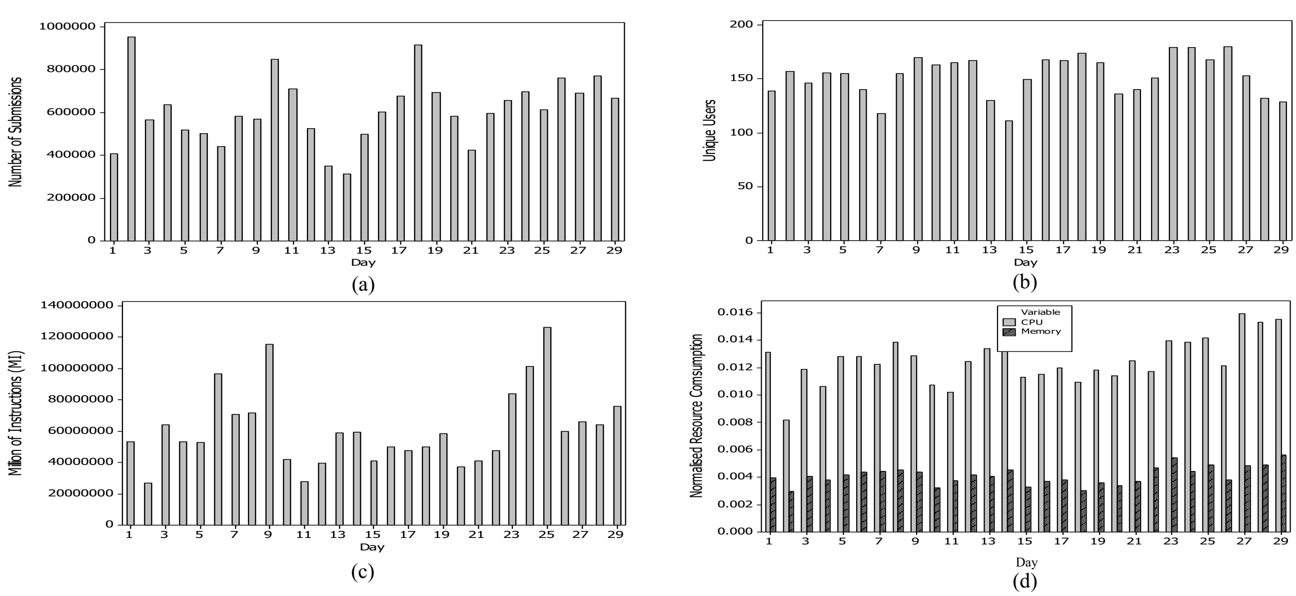


Figure 2. Statistical properties of (a) Completed-tasks, (b) Users, (c) Average task length, (d) Average resource consumption per task.

the result of dividing the difference in the amount of resource utilization by the original amount of requested resources.

Tasks are defined by the type and amount of work dictated by users, resulting in different duration and resource utilization patterns. Consequently, essential dimensions to describe tasks are: length (χ), 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. Therefore, the Cloud workload can be described 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 $P(t_j)$. The model components and their relationship are formalized in Equations 1 to 6.

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

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

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

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

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

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

VI. MODEL PARAMETERS

The objective of our approach is to characterize user and task behavior to derive the statistical parameters that define the workload model described in the previous section. This is performed in two steps: determine the set of profiles *U* and *T* defined in Equations 1 and 2, and derive the probabilistic functions for α , β , ϕ , χ , γ and π required in Equations 3 - 6.

A. Sampling Process

We select a sample size of 24 hours from the overall tracelog population to attain the classification for tasks and users. We decided to use sampling because of the fact that performing the analysis on a per day basis allows us to contrast the behavioral pattern results from different days. Additionally, from our coarse-grain statistical analysis that included the entire dataset, we identified that some days are representative of the overall average tracelog behavior due to the balance between submissions and task length. The selection of the sample population was calculated by comparing the variance between the average task length and number of submitted tasks per day against the entire tracelog. Using this technique, day 18 was selected as the sample population.

To determine the set of profiles for users and tasks we use *k*-means clustering [19] on a sample population from the tracelog to classify tasks and users based on to their respective dimensions. To derive the probabilistic functions of each profile we perform intra-cluster analysis to study the internal data distribution of each dimension. Moreover, before applying the described steps, we perform a sampling strategy to conduct the analysis on per day basis.

B. Cluster Analysis

Using the sample population, we used a clustering algorithm to classify tasks and users composed by the dimensions described in section 5. The *k*-means clustering is a popular data-clustering algorithm to divide *n* observations into *k* clusters, in which values are partitioned in relation of the selected dimensions and grouped around cluster centroids [19]. One critical factor in such an algorithm is determining the optimal number of clusters. For our analysis, we use the method proposed by Pham et al. [20]. 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 dimensions. 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 dimensions within the population and α_k is the weight factor based on the previous set of clusters.

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

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

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 determine the number of clusters for *U* and *T* (Equations 1 and 2) respectively. The plots of the determined clusters for users and tasks are shown in Fig. 3(a) and Fig. 3(b) respectively. An observation of interest in these two figures is that the values for some dimensions such as submission rate for cluster u_6 and length for cluster t_1 are widely spread and can deviate significantly from the centroid. Visually, it appears that t_2 contains the smallest number of elements compared to the other clusters. However, as Table II shows, t_2 in fact contains 72.56% of the task population. Additionally, Table II shows that 77.40% of users are clustered together in u_4.

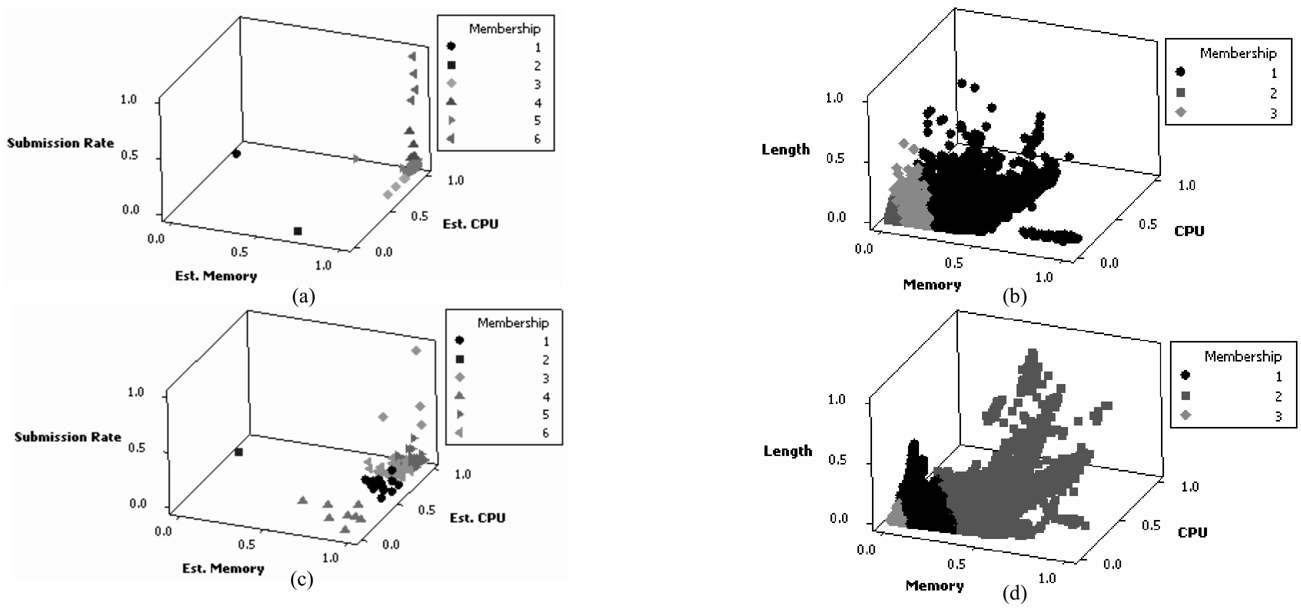


Figure 3. Clusterization results for (a) Users day 18, (b) Tasks day 18, (c) Users day 2 and (d) Tasks day 2.

To evaluate whether the derived user and task types are consistent across other periods of time in the same tracelog, the clustering process was repeated on another sample population. Day 2 was chosen as the statistical properties described in Fig. 2. Principally because it has the highest number of submissions and the number of users is lower than that of Day 18. This is helpful to determine whether the variation of submissions and users can affect the number of clusters derived from the "*average*" day.

We found that it was possible to create the same number of k clusters as Day 18 that satisfies $f(k) < 0.85$ for both users and tasks; as shown in Fig. 3(c) and Fig. 3(d) respectively. An observation of interest is that the general shape of the clusters for Day 2 are comparable to that of the cluster shapes for day 18 even though the workload for the two sampled days is different. When comparing the centroid values for the two analyzed days as shown in Tables III and IV, there is marginal difference of centroid values for tasks, which suggest a consistent resource utilization patterns during the two analyzed days. User centroid dimensions however experience a slightly increased discrepancy, especially for submission rate. The variability is clearly introduced by a larger number of submissions in day 2 being performed by a lower number of users in comparison to day 18. This increases the average submission rate per user close to 15% in day 2.

From our cluster analysis, we are able to make three clear observations. First, tasks and users exhibit similar behavioral patterns across the two different observational periods. Second, by comparing similar clusters it is possible to observe that although the patterns are close, they present differences evidently introduced by changes in the environment. Finally, the cluster analysis depicts in general terms the individual user and tasks patterns but does not provide the fine-grained parameters required to characterize realistic utilization models. This makes necessary to perform intra-cluster analysis to capture the fine details of individuals' (tasks and users) behavior.

TABLE II. PROPORTION OF ELEMENTS WITHIN CLUSTER FOR DAY 18.

Cluster	Pop %	Cluster	Pop %	Cluster	Pop %
u_1	0.68	u_4	77.40	t_1	1.84
u_2	0.68	u_5	15.75	t_2	72.56
u_3	2.74	u_6	2.74	t_3	25.60

TABLE III. CENTROID COMPARISON FOR USERS.

Day 2 Cluster	**2**	**3**	**1**	**6**	**5**	**4**
Sub Rate	0.0002	0.592	0.0152	0.0138	0.0124	0.0047
Est. CPU	0.7444	0.9089	0.6024	0.7978	0.948	0.1854
Est. Mem	0.0000	0.9624	0.9142	0.9057	0.9558	0.8557
Day 18 Cluster	**1**	**6**	**3**	**5**	**4**	**2**
Sub Rate	0.0003	0.7901	0.0027	0.0113	0.1801	0.0000
Est. CPU	0.8126	0.9838	0.7428	0.9718	0.9638	0.0000
Est. Mem	0.0000	0.9974	0.9947	0.9888	0.9922	0.7178
Euclidean Dist	**0.0682**	**0.2147**	**0.1623**	**0.1928**	**0.1723**	**0.2311**

TABLE IV. CENTROID COMPARISON FOR TASKS.

Day 2 Cluster	**3**	**1**	**2**
Length	0.0006	0.0244	0.075
CPU	0.0147	0.1041	0.2841
Memory	0.0115	0.0994	0.3849
Day 18 Cluster	**2**	**3**	**1**
Length	0.0007	0.0038	0.0107
CPU	0.0149	0.0810	0.2206
Memory	0.0089	0.0585	0.2556
Euclidean Dist	**0.0026**	**0.0512**	**0.1577**

C. Intra-Cluster Analysis

The intra-cluster analysis consists of studying the data distributions for each one of the cluster dimensions. The process requires fitting the data from the logs to specific distributions using a Goodness of Fit (GoF) test to obtain the parameters of their Probabilistic Distribution Functions (PDF). The objective is to use these PDFs as components of the workload model as described in Equations 3 and 4. The following considerations have been made during the intra-cluster analysis:

- *To analyze dimensions represented by non-averaged values such as submission rate and task length, the data is taken directly from the cluster.* However, if the dimension is represented by averaged values such as CPU and Memory estimation the data needs to be taken from the detailed measurements to capture the existing variability.
- *Due to the large population of data for each dimension, we perform the GoF test over samples with a confidence interval of 95% and margin of error as +/- 5%.* However, in cases where the number of records is small (<1000 records), the entire population is used to derive the statistical parameters.
- *Regarding to the behavioral patterns there are two special cases: resource estimation and consumption.* While the latter depict the way on how tasks consume resources, the former outline how users request resources. This involves two possible scenarios: overestimation (OE) and underestimation (UE) which are treated as separated data distributions.
- *For CPU and Memory consumption within the task-clusters, there are a substantial amount of records where utilization is equal to 0%.* This makes impossible to fit the resource consumption to a continuous distributions. Therefore these especial cases are treated as "*zero-inflated*" distributions [21] where the analyzed data is divided in two sets: continuous for values greater than zero and discrete for zero values.

We have fitted the resulting data subsets from the previously described considerations to their closest theoretical distributions applying Anderson-Darling and Kolmogorov–Smirnov GoF tests. We have used Minitab [22] and R [23] statistical packages to efficiently perform such analyses due to the large amount of records. For each parameter we evaluated several distributions including normal, lognormal, exponential, weibull, gamma, logistic, loglogistic, and extreme value among others. To determine the best candidate in case that more than one distribution fit the data, we have selected the one with the highest above 0.05 P-value that determinates the statistical significance according to the process described in [24]. The entire set of distributions obtained from this procedure is presented in Table V and Table VI for users and tasks respectively.

Regarding to the user behavioral patterns, it is observed from Table V that the General Extreme Value (GEV) distribution best fits CPU and Memory overestimations in both analyzed scenarios. This indicates that users tend to highly overestimate both resources when tasks are submitted. On the other hand, resource underestimations generally follow distributions such as Lognormal, Weibull, and Gamma. This indicates that when users underestimate they do it in small proportions especially for Memory. According to [25], Memory requests are rarely underestimated by a large factor because tasks are killed when Memory requests greatly exceed enforced limits. Finally, submission rates follow distributions such as Lognormal, Gamma, and Weibull. This remarks the observations from coarse-grain analysis in Fig. 1 that indicates that most of the users have low submission rates in comparison to very few users having high submission rates. The diversity of distributions for user clusters across the analyzed days especially for underestimation of resources reveals the details of the differences between clusters measured in Table III.

TABLE V. SET OF DATA DISTRIBUTIONS DERIVED FROM USER CLUSTERS.

		Day 2	**Day 18**
Cluster	***Dimension***	***Best Fit Distribution***	***Best Fit Distribution***
u_1	***Submission Rate*** ***CPU UE / OE*** ***Memory UE / OE***	-Lognormal -Gamma / General Extreme Value - Lognormal / General Extreme Value	-Uniform - Lognormal / General Extreme Value -Gamma / NA
u_2	***Submission Rate*** ***CPU UE / OE*** ***Memory UE / OE***	- Uniform -NA / Normal -Normal / NA	-Uniform -NA -NA
u_3	***Submission Rate*** ***CPU UE / OE*** ***Memory UE / OE***	- Uniform -Weibull / General Extreme Value -NA / General Extreme Value	-Uniform -Weibull / General Extreme Value -Gamma / General Extreme Value
u_4	***Submission Rate*** ***CPU UE / OE*** ***Memory UE / OE***	- Lognormal -Weibull / NA -NA / General Extreme Value	-Uniform - Lognormal / General Extreme Value - Lognormal / General Extreme Value
u_5	***Submission Rate*** ***CPU UE / OE*** ***Memory UE / OE***	- Lognormal -LogLogistic / General Extreme Value -NA / General Extreme Value	-Gamma - Lognormal / General Extreme Value - Lognormal / General Extreme Value
u_6	***Submission Rate*** ***CPU UE / OE*** ***Memory UE / OE***	-Weibull -Gamma / Normal -Loglogistic / General Extreme Value	-Uniform -Weibull / General Extreme Value - Lognormal / General Extreme Value

TABLE VI. SET OF DATA DISTRIBUTIONS DERIVED FROM TASK CLUSTERS.

		Day 2		Day 18	
Cluster	***Dimension***	***Best Fit Distribution***	***P(0)***	***Best Fit Distribution***	***P(0)***
t_1	***Length*** ***CPU*** ***Memory***	-Lognormal -Normal -Lognormal	-NA -16% -22%	-Loglogistic -Lognormal -Loglogistic	-NA -20% -37%
t_2	***Length*** ***CPU*** ***Memory***	-Lognormal -Weibull -Normal	-NA -18% -22%	-Lognormal -Lognormal -Loglogistic	-NA -13% -30%
t_3	***Length*** ***CPU*** ***Memory***	-Lognormal -Lognormal -Lognormal	-NA -41% -60%	-Lognormal -Lognormal -Loglogistic	-NA -13% -30%

In the case of tasks consumption patterns, it is observed from Table VI that the length generally follows a lognormal distribution indicating that even within the clusters most of the tasks have a short length. The same occurs with CPU and Memory consumption where lognormal, loglogistic, and Weibull distributions indicate a high proportion of tasks consume resources at lower rates. The homogeneity of distributions across the two days for task clusters confirms that the consumption patterns are very similar in both scenarios as was measured in Table IV. As observed, the context of the two analyzed days is different and affects the workload patterns. Day 2 presents a higher number of submissions with a lower number of users and considerable reduced amount of work computed in comparison to day 18 as illustrated in Fig. 1. Nevertheless, the approach proposed in this paper allows us to abstract the same general types of users and tasks for both scenarios and at the same time outline the specific contextual behavioral patterns for each one.

VII. MODEL EVALUATION

To assess the quality of the models derived with the proposed approach we have performed simulation experiments and contrasted the results against the production data from Google traces. To perform these simulations we have develop a workload generator that extends the capabilities of the CloudSim simulator [26-29]. The workload generator takes as input the model parameters for users and tasks and produces as an output a set of instructions to be executed by the simulator to mimic the operational environment behavior.

A. Workload Generator

The workload generator is integrated by 5 modules: User Profiles, Task Profiles, User Generator, Task Generator, and Workload Coordinator. The interaction of these components to produce the Cloud workload is described in Fig. 4.

The User and Task Profiles describe respectively each one of the user and task types identified during the clustering process and encapsulate the behavior outlined through the intra-cluster analysis. The User Generator creates the CloudSim user instances and links them with a specific profile determined by their associated probabilities as described in Equation 5. The Task Generator creates the CloudSim task instances and links them with a specific task profile determined by the conditional probability in Equation 6. Each one of the task parameters, including the resources requested by the users is obtained by sampling the inverse CDFs of the distributions defined in Equation 3 and Equation 4. Finally, the coordinator controls the interactions between the workload generator and CloudSim framework.

B. Simulation Environment

We have simulated a datacenter composed by 12,583 servers based on the capacities described in the Google tracelog. The simulation time is set up to emulate 29 days and contains 153 users per day. The user and task profiles are configured using the statistical parameters derived for day 2 and described in Section VI. Day 2 is preferred over day 18 due to the irregularity of the data distributions found during the analysis. The results of the simulation are compared against random selected data from the same day in the tracelog. It is important to mention that although we are simulating the execution of tasks, we are not comparing the task duration against the real measurements registered in the tracelog. This is because the duration of tasks can be affected by other factors such as scheduling priorities, performance interference, or failure occurrence which are out of the scope of this paper.

C. Results Analysis

The results from the simulation experiment show the accuracy of the derived model to represent the operational characteristics within the Cloud computing datacenter for the selected scenario. The proportions of users, tasks and the task per users classified by cluster membership

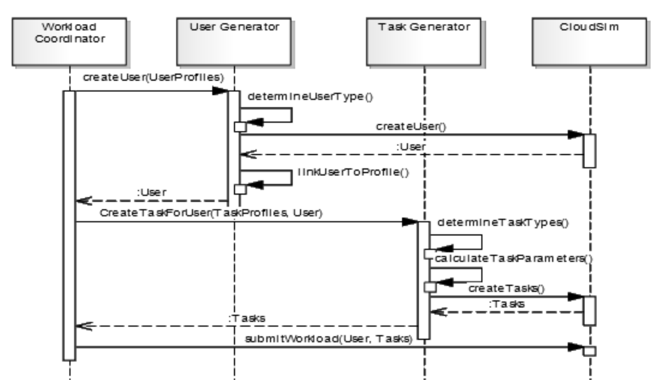


Figure 4. Workload Generator Components Interaction.

are contrasted as presented in Fig. 5. It is important to highlight the similarity that in all cases the data obtained from the simulation (S) presents in comparison to the real data logged in the traces (R). The discrepancies in most of the cases lie between the error margins of +/- 5%. The less precise cases are illustrated in Fig. 5(b) for t_2 and t_3 where the differences have been measured as 6% and 8% respectively. This behavior can be explained by the stochastic nature of the model and the coefficient of variation (CV) of users per day calculated as 12.2 from the data analysis. Nevertheless, this seems to have a negligible impact in the distribution of tasks per user where the differences are no greater than +/- 0.6% as shown in Fig. 5(c). The behavioral patterns of simulated users and task were also evaluated. Fig. 6 illustrates the patterns of the 3 dimensions for u_5 that composes just under 60% of all the individual users in the Cloud environment for the selected day. From the presented plots, it is observed that the distributions for the user dimensions during the simulation match closely the patterns observed from the logged data. In this case, the percentage of error was calculated as 1.68%, 2.0%, and 1.03% for submission rate, CPU and Memory estimation ratio respectively.

Table VII focuses on the evaluation accuracy for all the user clusters. It is based on the calculation of the percentage of error between the locations of observed and simulated data distributions. It is observed that the percentage of error is considerably low for CPU and Memory estimation, calculated on average as 2.14% and 1.41% respectively. However, in the case of submission rate with an average of 4.38%, u_3 introduces a moderately high error value. This is due to a lower number of cluster elements (4 users) therefore making it impossible to determine the data distribution. In this case we use a uniform distribution to select with equal probabilities any of the 4 possible values resulting in a percentage of error approximately 13.42%. If instead of using a uniform distribution we fit the data to a normal distribution, the percentage of error is minimized to 6.05% resulting in an average error of 3.15% for the submission ratio. This suggests the use of normal distribution to characterize clusters with few elements. Fig. 7 shows the comparison of patterns for t_3 which describes just under 80% of all the tasks in the tracelog. The plots calculated describe a very accurate simulation of the task patterns. The percentage of error was determined as 0.19%, 0.55%, and 2.5% for task length, CPU and Memory consumption respectively. The error measurements for the complete list of task clusters are presented in similar to users, Table VIII compares the locations of the logged and simulated data distributions for task length, CPU and Memory usage.

It is noticeable that the average percentage of error is low for length and Memory usage, calculated as 0.24% and 3.33% respectively. However, for CPU usage the average

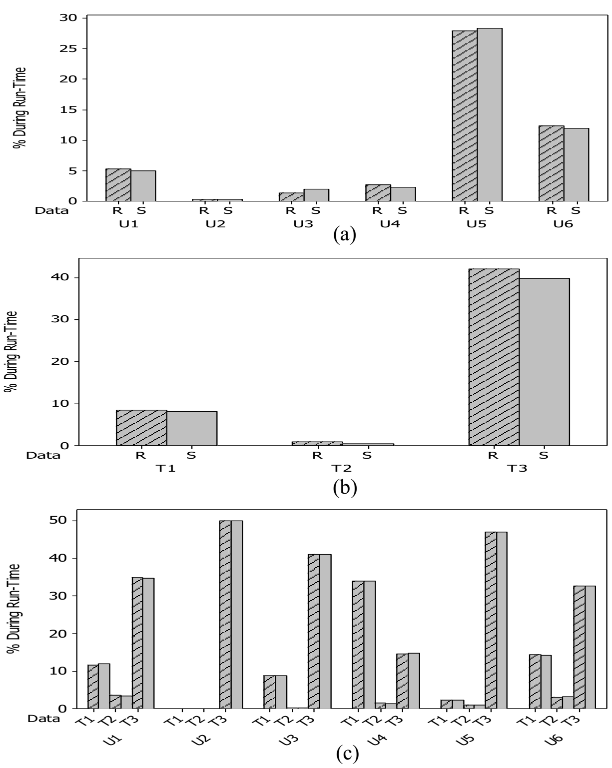


Figure 5. Comparisons of the proportions between logged data and the outcome from the simulation. (a) Proportions of user per cluster membership, (b) Proportions of task per cluster membership, and (c) illustrates the comparison of task per user.

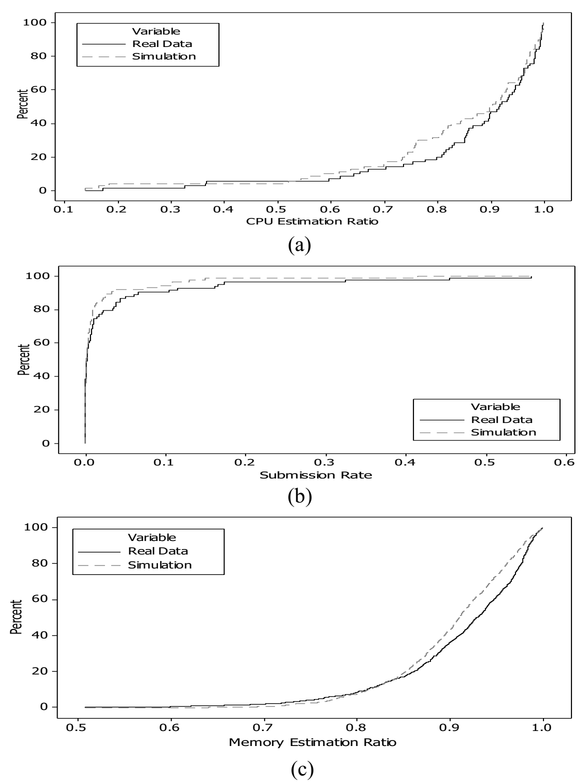
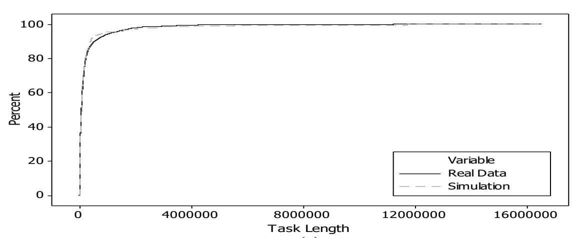


Figure 6. Comparison of user patterns between real and simulated data for u_5 (a) estimation ratio for CPU, (b) submission rates, and (c) estimation ratio for Memory requests.

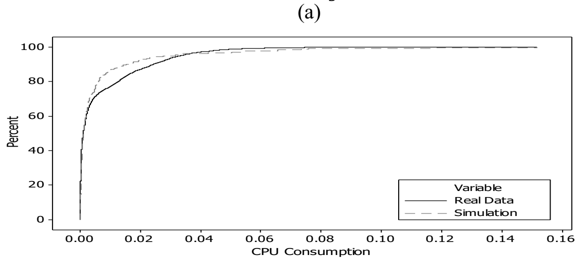
error percentage is approximately 10.97% caused by a very highly-inaccurate simulated CPU usage pattern within t_1. The root cause of this irregularity is a result of a multimodal distribution of the data in this cluster as is shown in Fig. 8(a). Initially, following the proposed approach we attempted to fit the data to the closest distribution. However, this produces very imprecise results as was discussed previously and presented in Fig. 8(b). To improve the accuracy of our model we applied multi-peak histogram analysis for region splitting [30] and fitted the sub-regions to new distribution parameters. As a result we minimized the error percentage from 24.66% to 0.95%. The comparison between the improved simulated distribution and the logged data is presented in Fig. 8(c).

VIII. Model Applicability

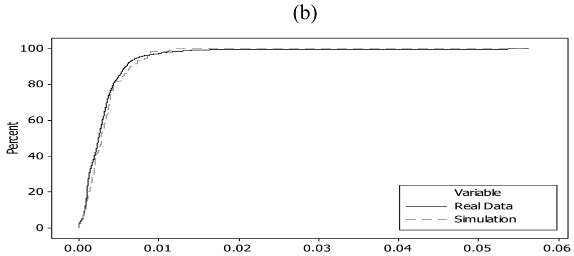
As previously mentioned, the workload model presented in this paper enables researches and providers to simulate realistic request and consumption patterns. This is critical in order to improve resources utilization, reduce energy waste and in general terms support the design of accurate forecast mechanisms 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.



(a)



(b)



(c)

Figure 7. Comparison of task patterns between real and simulated data for t_3 (a) task length, (b) CPU consumption, and (c) Memory consumption.

The first is a resource overallocation mechanism that considers customers' resource request patterns and the actual resource utilization imposed by their submitted tasks. The main idea is to exploit the resource utilization patterns of each customer for smartly underallocating resources to the requested Virtual Machines. This reduces the waste produced by frequent overestimations and increases the datacenter availability. Consequently, it creates the opportunity to host additional Virtual Machines in the same computing infrastructure improving its energy-efficiency [31].

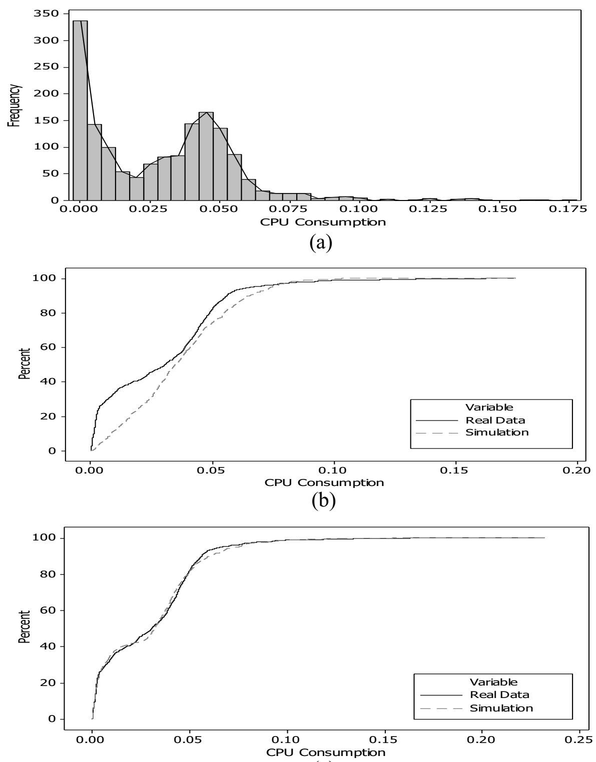
The second mechanism considers the relationship between Virtual Machine interference due to competition for resources and energy-efficiency. Therefore, a model to reduce the energy waste by exploiting the workload heterogeneity that exists in Cloud environments is proposed. The core idea is to co-allocate different types of workloads based on the level of interference that they create to reduce the resultant overhead and consequently to improve the energy-efficiency of the datacenter. The approach classifies the incoming workloads based on their resource usage patterns, pre-selects the hosting servers based on resources constraints, and makes the final allocation decision based on

TABLE VII. Evaluation of the Accuracy of User Patterns.

	Cluster	Real	Simulation	%Error
Submission Rate	u_1	5.248	5.107	2.69
	u_2	0.006	0.006	0.00
	u_3	1.543	1.336	13.42
	u_4	6.171	6.318	2.38
	u_5	6.648	6.760	1.68
	u_6	5.600	5.943	6.13
CPU Estimation	u_1	0.648	0.622	3.99
	u_2	0.423	0.412	2.48
	u_3	0.848	0.863	1.74
	u_4	0.848	0.863	1.74
	u_5	0.092	0.090	2.00
	u_6	0.585	0.580	0.85
Memory Estimation	u_1	0.906	0.901	0.54
	u_2	1.148	1.146	0.17
	u_3	0.968	0.963	0.54
	u_4	0.488	0.461	5.59
	u_5	0.941	0.931	1.03
	u_6	0.889	0.894	0.60

TABLE VIII. Evaluation of the Accuracy of Task Patterns.

	Cluster	Real	Simulation	%Error
Length	t_1	11.07	11.1	0.27
	t_2	16.57	16.53	0.24
	t_3	15.46	15.43	0.19
CPU Utilization	t_1	0.029	0.036	24.66
	t_2	0.071	0.065	7.70
	t_3	6.56	6.596	0.55
Memory Utilization	t_1	4.294	4.294	0.00
	t_2	0.046	0.050	7.50
	t_3	6.196	6.041	2.50



(c)

Figure 8. Multi-peak histogram analysis to improve the accuracy of CPU consumption patterns $\mathbf{t_1}$. (a) Multimodal distribution of CPU utilization, (b) imprecise results for CPU consumption, (c) adjusted distribution to reduce the percentage of error.

the current servers' performance interference level. 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. One big advantage is that the model does not just replay the data in 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 reproduce the behavior from a specific period of time. Another important benefit is that the model integrates the relationship between user demand and the actual resource usage which is essential in both scenarios where the aim is to achieve the balance between requests and utilization in order to reduce the waste of resources.

IX. CONCLUSIONS

An approach to derive realistic workload models that encompass user and task behavior has been presented in this paper. Furthermore, a 30 day tracelog from the Google Cloud has been analyzed to derive the statistical parameters to describe the proposed model. Our evaluation demonstrates that by following the approach described in this paper, it is possible to obtain the statistical parameters to emulate production environments in most cases within a margin of error of +/-5%. In this work, an exhaustive analysis of the data has been performed at three different levels: coarse-grain, cluster, and intra-cluster from which a number of observations and conclusions can be made. These are listed as follows:

- *Modeling user behavior is a critical factor when characterizing Cloud workloads*. Our analysis shows that user behavior affects workload characteristics and consequently the Cloud environment. Coarse-grain analysis suggests that there is a significant variation in the resource utilization and inferred submission rates of tasks within an observed time period. This indicates that resource utilization and number of tasks is dependent on user patterns, an element that has been overlooked in related works.
- *Workloads are highly variable across different observation periods* [31]. Our analysis is the first to measure and model this heterogeneity to simulate realistic production environments. Our analysis has revealed that task and user dimensions differ significantly on a daily basis. Furthermore, performing cluster and intra-cluster analysis demonstrates the variance in behavioral patterns between different types of users and tasks. Distributions modeled for each dimension exhibit a variance in shape as well as scale, making evident the diversity between cluster characteristics.
- *The Cloud environment does not exhibit obvious cyclic behavior.* In contrast to other models such as Grid Computing where seasonal patterns are detectable, the analyzed data does not provide strong correlation between the amount of work and specific periods of time. This confirms the dynamicity that exists in Cloud environments, where users are not tied to predefined schedules imposed by the system. Additionally, this indicates the diversity of users and their strong influence on the workload.
- *Users grossly overestimate the resources required to meet business objectives.* The intra-cluster analysis reveals that in over 90% of cases, users tend to overestimate the amount of resources that they require, wasting in some cases near to 98% of the requested resource. This type of exhibited user behavior has been the focus of studies in [31, 32] but this is the first time that the phenomena has been modeled and quantified.
- *Performing analysis on large-scale tracelogs is fundamental to deriving realistic models.* Although the analyzed tracelog is restricted to one month of operation, it is large enough to outline realistic workload patterns on per day basis. It has been previously observed that the datacenter usage level is stable across the different days [25]. This provides a better understanding of the environment in comparison to the hourly analyses previously conducted. However, Cloud environments are very dynamic and in order to develop realistic models, providers require to continuously analyze their datacenter tracelogs. This remarks the importance of methodologies of analysis

such as the one presented in this paper which allows providers to outline general workload characteristics as long as specific behavioral patterns.

X. Future Work

Currently, only two independent days from the tracelog have been modeled; nevertheless, it is important to generate a model representative of the entire tracelog. As future work, our methodology will be applied to a dataset representative of the entire month, in order to compare the derived parameters with the preliminary results obtained in this paper. Future directions will also include 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 accurately emulating and analyzing workload energy consumption and reliability, enabling further research into energy-efficiency, resource optimization and failure-analysis of the Cloud environment.

Acknowledgment

The work in this paper has been supported in part by the Mexican Council of Science and Technology CONACyT (No. 213247), the National Basic Research Program of China (973) (No. 2011CB302602), the UK EPSRC WRG platform project (No. EP/F057644/1), and the Major Program of the National Natural Science Foundation of China (No. 90818028).

References

[1] R. Buyya, *et al.*, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.,* vol. 25, pp. 599-616, 2009.

[2] Google. *Google Cluster Data V1*. Available: http://code.google.com/p/googleclusterdata/wiki/TraceVersion1

[3] Yahoo. *Yahoo! M45 Supercomputing Project*. Available: http://research.yahoo.com/node/1884

[4] Qi Zhang, *et al.*, "Characterizing Task Usage Shapes in Google Compute Clusters," in *In Proc. of The 5th International Workshop on Large Scale Distributed Systems and Middleware*, 2011.

[5] S. Kavulya, *et al.*, "An Analysis of Traces from a Production MapReduce Cluster," in *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, 2010, pp. 94-103.

[6] A. K. Mishra, *et al.*, "Towards characterizing cloud backend workloads: insights from Google compute clusters," *SIGMETRICS Perform. Eval. Rev.,* vol. 37, pp. 34-41, 2010.

[7] S. Aggarwal, *et al.*, "Characterization of Hadoop Jobs Using Unsupervised Learning," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 2010, pp. 748-753.

[8] Google. *Google Cluster Data V2*. Available: http://code.google.com/p/googleclusterdata/wiki/ClusterData2011_1

[9] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. ACM,* vol. 51, pp. 107-113, 2008.

[10] C. Reiss, *et al.*, "Google Cluster-Usage Traces: Format + Schema," Google Inc., White Paper, 2011.

[11] A. Bahga and V. K. Madisetti, "Synthetic Workload Generation for Cloud Computing Applications," *Journal of Software Engineering and Applications,* vol. 4, pp. 396-410, July 2011.

[12] A. Beitch, *et al.*, "Rain: A Workload Generation Toolkit for Cloud Computing Applications," Electrical Engineering and Computer Sciences University of California at Berkeley, White paper UCB/EECS-2010-14, 2010.

[13] Y. Chen, *et al.*, "Analysis and Lessons from a Publicly Available Google Cluster Trace," EECS Department, University of California, Berkeley UCB/EECS-2010-95, June 14 2010.

[14] J. W. Smith and I. Sommerville, "Workload Classification & Software Energy Measurement for Efficient Scheduling on Private Cloud Platforms," presented at the ACM SOCC, 2011.

[15] G. Wang, *et al.*, "Towards Synthesizing Realistic Workload Traces for Studying the Hadoop Ecosystem," in *Proceedings of the 19th Annual Meeting of the IEEE International Symposium on Modeling, Analysis and Smulation of Computer and Telecommunication Systems (MASCOTS)*, Singapore, 2011.

[16] P. Garraghan, *et al.*, "Real-Time Fault-Tolerance in Federated Cloud Environments," in *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on*, 2012, pp. 118-123.

[17] Standard Performance Evaluation Corporation. (2012, July 7). *SPECpower_ssj2008 Results*. Available: http://www.spec.org/power_ssj2008/results/

[18] J. A. Quiane-Ruiz, *et al.*, "RAFTing MapReduce: Fast recovery on the RAFT," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, 2011, pp. 589-600.

[19] X. Rui and D. Wunsch, II, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on,* vol. 16, pp. 645-678, 2005.

[20] D. T. Pham, *et al.*, "Selection of K in K-means clustering," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* vol. 219, pp. 103-119, January 1, 2005.

[21] W. Tu, "Zero-Inflated Data," in *Encyclopedia of Environmetrics*, ed: John Wiley & Sons, Ltd, 2006.

[22] Minitab, "Distribution Analysis," in *Minitab Users' Guide*, ed, 2011.

[23] Institute of Statistics and Methemathics. (2012, July 22). *The R Project for Statistical Computing*. Available: http://www.r-project.org/

[24] Minitab, "Quality Control," in *Minitab Users' Guide*, ed, 2003.

[25] C. Reiss, *et al.*, "Towards understanding heterogeneous clouds at scale:Google trace analysis," Intel Science & Technology Center for Cloud Computing, White Paper, ISTC-CC-TR-12-101, 2012.

[26] R. Buyya, *et al.*, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *Proc. of the International Conference on High Performance Computing & Simulation.* , 2009, pp. 1-11.

[27] R. N. Calheiros, *et al.*, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software: Practice and Experience,* 2010.

[28] S. K. Garg and R. Buyya, "NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, 2011, pp. 105-113.

[29] B. Wickremasinghe, *et al.*, "CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, 2010, pp. 446-452.

[30] S. Pal and P. Bhattacharyya, "Multipeak histogram analysis in region splitting: a regularisation problem," *Computers and Digital Techniques, IEE Proceedings E,* vol. 138, pp. 285-288, 1991.

[31] I. Solis Moreno and X. Jie, "Neural Network-Based Overallocation for Improved Energy-Efficiency in Real-Time Cloud Environments," in *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2012 IEEE 15th International Symposium on*, 2012, pp. 119-126.

[32] A. Quiroz, *et al.*, "Towards autonomic workload provisioning for enterprise Grids and clouds," in *Grid Computing, 2009 10th IEEE/ACM International Conference on*, 2009, pp. 50-57.