A User-Based Model of Grid Computing Workloads

Marcus Carvalho and Francisco Brasileiro
Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Brazil
marcus@lsd.ufcg.edu.br, fubica@dsc.ufcg.edu.br

***Abstract*—A computational grid is a large scale federated infrastructure where users execute several types of applications with different submission rates. On the evaluation of solutions for grids, there are not much effort on using realistic workloads for experiments, and most of the time users' activities and applications are not well represented. In this work, we propose a user-based grid workload model which is based on clustering users according to their behaviour in the system and their applications. The results show that according to a new metric proposed, the model quality increases when using clustering and extracting models for the group of users with similar behaviour. Moreover, we compare our user-based modelling with a state-of-the-art system-based modelling approach. We show that by using our user-based model the system load can be easily changed by varying the number of users in the grid, creating different evaluation scenarios without affecting individual users' behaviour. On the other hand, varying the number of users in the system-based model does not affect the system load and change the way individual user's behave on the system, which can result in unrealistic users' activities.**

***Keywords*-grid computing; performance modelling; performance evaluation.**

I. Introduction

Computational grids are infrastructures where resources from different administrative domains are shared aggregating large scale processing capacities. Some characteristics of grid systems include the heterogeneity of its resources, the diversity of applications, as well as the way that users interact with the system. On the evaluation of solutions for grid systems (e.g. schedulers and resource management policies), it is desirable to analyse how the proposed solutions behave on several representative scenarios of the target system. Usually, the different scenarios are obtained by varying the system's workload (demand) and/or the system's resources (supply).

Generally, three approaches are used to create workload scenarios when evaluating grid environments. The first approach is to perform a parameter sweep on values for workload attributes related to job arrival rates and applications' characteristics [1], [2]. The problem with this approach is that, in general, the probability distributions used, as well as the values used to set their parameters are not well justified or validated with real data. For instance, a common mistake is to use a Poisson distribution to generate values for job arrival rates, what has already been shown to be incorrect for most parallel systems [3]. On the second approach, workload traces collected from existing grids or other parallel systems are simply reproduced [4]. The problem in this case is that the evaluation is limited to specific historic situations that happened when collecting the system data, not allowing an easy way to explore different scenarios that could happen in other periods. The third approach is to generate synthetic workloads according to models that represent the system workload. The problem with this approach is the dependency on representative workload traces to be used when extracting the model. However, the advantage of this approach is the flexibility to create different scenarios by varying the model input parameters, which allows the investigation of several scenarios while keeping general system properties [5].

We can broadly classify workload models as either descriptive or generative models. Descriptive models (which we will call *system-based* models) consider the workload source as a black-box, and only the characteristics of the workload that arrives in the system are modelled, no matter how it is generated. On the other hand, the generative models (which we will call *user-based* models) consider the workload source as a white-box and tries to model how the workload is actually generated. In this case, the total system workload is the aggregation of each individual workload generated by its users. An advantage of user-based models is that changing the system load can be easily done by varying the number of users in the system, without affecting the individual user's behaviour within the system [3].

In this work we propose a user-based workload model for grid computing systems, which describes the users' activities according to their characteristics, clustered by users' behaviour profiles. The workload model is focused on Bag-of-Tasks (BoT) applications, as most jobs from grid workloads that have been analysed present this structure [6]. The main contributions of this paper are three-fold:

- We propose a user-based workload model for grid computing systems, evaluate the quality of the model and compare it to a state-of-the-art system-based model.
- We describe a methodology to extract the proposed user-based model and to generate synthetic workloads. We also made available tools to apply this methodology.
- We propose a new metric to measure the quality of the model, according to *goodness-of-fit* tests for probability

distributions applied on clustered data.

The new metric proposed, called Group Goodness-of-Fit (GGoF) indicates the fraction of users that obtained successful distribution fittings according to a goodness-of-fit test (i.e. the test result was higher than a certain significance level).

An evaluation of the proposed model is done by applying the modelling methodology on 6 traces of existing grid computing systems. The results show that there is an increase on the model quality when clustering users according to their activity and extracting the model for each group of users, compared to the usual approach of having a single model for the entire workload (without clustering). The results also show that it is possible to have good results even with a small number of clusters, which makes the model more general while keeping it representative. Moreover, we applied the user-based model proposed to generate synthetic workloads and compare to synthetic workloads generated by a state-of-the-art system-based workload model for grid systems [7]. We show that by using our user-based model the system load can be easily changed by varying the number of users in the system, creating different grid scenarios without affecting individual users' behaviour. On the other hand, varying the number of users in the system-based model does not affect the system load and change the way individual user's behave on the system, which can result in unrealistic users' activities.

The rest of the paper is organized as follows: Section II is devoted to the discussion of related work; the workload modelling methodology proposed in this work is described in Section III; an evaluation of the model is presented in Section IV; and in Section V we present our concluding remarks.

II. Related Work

The state-of-the-art workload models for parallel systems are based on the descriptive (system-based) model approach, on which a single model describes the workload that arrives in the entire system, either for grid computing [7], [8] or other parallel systems [5], [9]. As discussed before, this approach considers the workload generation source as a black-box.

However, in some studies, authors advocate that user-based models are more adequate for workload modelling of parallel systems, mainly for their flexibility to easily change the system load by varying the number of users, keeping the representativeness of the workload without affecting user's activity [10]–[12]. In this paper, we provide evidences that corroborate that user-based models are more appropriate than system-based models in some aspects, as some drawbacks presented by the latter are mitigated by the former.

Lublin and Feitelson [5] proposed a workload model for parallel systems extracted from traces of three real sites.

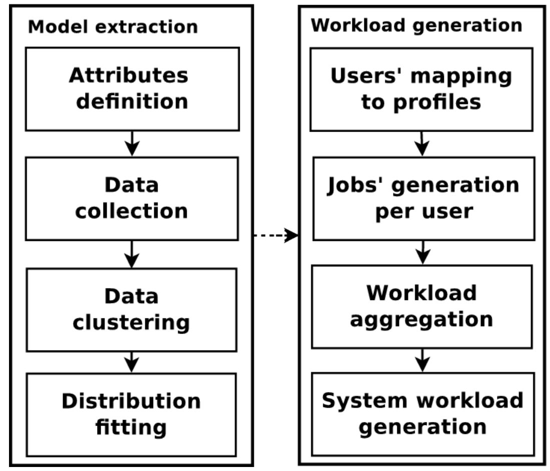

Figure 1. Methodology flow for workload modelling.

The model is obtained by fitting distributions to data of workload attributes. Iosup et al. [7] also used distribution fitting on the modelling of grid computing workloads, using seven traces of existing systems. They addressed in their model some specific characteristics of grid applications that are not found in other parallel systems (e.g. clusters and supercomputers), as the Bag-of-Tasks (BoT) nature of the jobs that comprises most of the grid's workloads. In our study, we propose a workload model for grid systems that also focus on BoT applications and uses distribution fitting on the modelling methodology. However, instead of adopting a system-based modelling approach as in previous works, we proposed a user-based modelling approach and highlight some advantages of this approach.

Javadi et al. [13] proposed a statistical modelling methodology applied to intervals of resource availability in large scale distributed systems. They used clustering techniques before applying distribution fitting in the data. In our study, we also apply clustering techniques in order to group users with similar behaviour and then apply distribution fitting for each group of users. Furthermore, we incorporated goodness-of-fit tests to guide the choice of the appropriate number of clusters by proposing a new metric that calculates the fraction of users that have successful goodness-of-fit results for the distribution fitting applied for their groups. We show that the quality of the model is increased when modelling groups of users clustered by their characteristics, according to the new metric proposed.

III. Workload Modelling Methodology

This section presents the methodology for the grid workload modelling proposed in this work. The first part covers the model extraction phase, while the second covers the synthetic workload generation that uses the extracted model as input. Figure 1 shows how the steps of the methodology are chained.

A. Model Extraction

The model extraction phase has four steps: define which workload attributes to use in the modelling, and their relations; define which workload traces to use on the extraction, and collect the proper data; cluster users according to the similarity of their activities in the system; and, apply distribution fitting to workload attributes' data for each group of users. These steps are described below.

1) Attributes definition: The grid workload is composed by many workload attributes, which are related to form the system workload. In this way, the first step is to identify which attributes to use in the workload modelling. As a user-based approach is used in this work, the users' activities in the system has to be represented by the model. Users interact with the grid by submitting jobs and receiving the results when the execution is completed. Most of the users' applications executed in grids are BoT [14], where a group of independent tasks are submitted in a short period of time, all of them being part of the same job. The workload attributes used in our modelling to represent the users' activities and their applications are:

- **Job Inter-Submission Time (JIST)**: Time interval duration between two consecutive job submissions by the same user.
- **Task Execution Time (TET)**: Time needed to execute a task in a reference machine.
- **Job Execution Time (JET)**: Time needed to execute a BoT job in a reference machine. It is defined as the sum of the execution time of all tasks that comprises the BoT job.

2) Data collection: We need to collect representative data from real systems in order to extract the workload model. The most common approach is to process logs from real systems, extract the required attributes, and store the data collected in an appropriate format. Many workload traces from grid and other parallel systems are available in repositories such as the Grid Workload Archive (GWA[1]) and the Parallel Workload Archive (PWA[2]). Grid workload traces available in the GWA repository were used in our work.

3) Data clustering: Grid systems are comprised by many users which commonly have different characteristics [6] regarding the way they interact with the system. In this work we have focused in the following characteristics: job submission rates, execution time of tasks, and number of tasks per BoT job. In order to address the differences on users' activities, clustering techniques can be used to identify users with similar behaviour and group them according to their profiles.

We analysed two clustering techniques to use: k-means and hierarchical clustering, as in the work by Javadi et al. [13], and found the hierarchical clustering as more appropriate for our modelling. The hierarchical clustering receives as input the distance matrix between each element, which in this case are the workload attributes' values for each user. Starting from the case where each user is a group itself, it iteratively combines users based on the distance matrix to form the groups, providing many grouping levels until there is only one group comprising all users. In the context of our work, the main benefits of this technique in comparison to the k-means are: the hierarchical clustering gives deterministic results if executed many times; it is easier to model in terms of distance between users; the iterative process gives an overview of many possible options for the number of clusters to be chosen.

The distance matrix used as input for the hierarchical clustering must represent the similarity of users in terms of workload attributes. We process each of the three workload attributes (*job inter-submission time, task execution time* and *job execution time*) separately when applying the clustering. We define the distance between two users in terms of a workload attribute based on the distance between the empirical cumulative distribution functions (ECDF) of attribute values for each user. We used the two-sample Kolmogorov-Smirnov (KS) test to measure the maximum distance between two distributions by their ECDFs, provided by the D statistic resulted from the KS test [3]. The D statistic value for each pair of users for each workload attribute was used to fill the distance matrix used in the hierarchical clustering process.

The groups of users defined in this step for different numbers of clusters and for each workload attribute are provided for the next step where distribution fitting is applied to extract probability distributions that represent the characteristics of workloads generated by the users of each group.

4) Distribution fitting: The output of the model extraction phase is one probability distribution for each group for each workload attribute. The goal of the distribution fitting step is to find the best distribution that describes each attribute for each group of users.

In order to apply the distribution fitting, we have to choose a set of probability distributions as candidates for the fitting. As done in related work [7], [15], we also selected often used distributions which have low complexity functions. The selected candidate distributions are: *exponential, normal, log-normal, gamma* and *weibull*. To estimate the parameters' values for each distribution, we used the *Maximum Likelihood Estimation* (MLE) method, which was applied in most workload modeling studies [5], [7], [15] and is less sensitive to outliers than the alternative moment matching method [3].

After fitting distributions to the data, we have to calculate how close each distribution is to the original data, which can be measured by Goodness-of-Fit (GoF) tests. We used the Kolmogorov-Smirnov (KS) test, which checks the null hypothesis that the original data comes from a certain theoretical distribution. The null hypothesis is rejected (resp.

[1] http://gwa.ewi.tudelft.nl/

[2] http://www.cs.huji.ac.il/labs/parallel/workload/

not rejected) if the p-value calculated by the test is lower (reps. higher) than a certain significance level [16].

The KS test has an issue of being sensitive for large samples. As the data samples used to extract the workload models are all very large, we applied an approach used in other modelling studies to mitigate this issue [13], [15], [17]. We select $1,000$ random samples of size 30 for each fitted data, obtain the p-values for the KS test applied to each sample and then calculate the average p-value.

The distribution fitting is done separately for each workload attribute. For a workload attribute, given a group of users clustered in the previous step, the distribution fitting process uses all values of the workload attribute being modelled for all users of the group to perform the fitting. For instance, when fitting distributions for the Task Execution Time attribute, considering that we want to have 2 groups A and B of users, one fitting is applied to all values of task execution times for all users of group A, and another fitting is applied the same way for the users of group B, resulting in two sets of candidate distributions with fitted parameters, one for each group.

B. Synthetic Workload Generation

The model extracted in the previous phase serves as input for this second phase, where we describe how synthetic workloads are generated. The process consists in choosing the desired number of users in the system and mapping them to groups of users' profiles; generating jobs for each user from the probability distribution of its group for each workload attribute; aggregating the generated jobs of all users; and, finally generating the total system workload.

1) Mapping users to profiles: Grid systems are composed of many resource centers or sites, and many users are associated to each site. In the modelling extraction phase, each user is mapped to a cluster for each workload attribute, which represents the profile for the user's activity. The fraction of users that each cluster covers in the original trace is also calculated in the extraction phase. This fraction is used in the mapping of users to profiles when generating synthetic workloads. For each user created in the synthetic workload, the probability of this user being associated to a profile is equal to the fraction of users that are associated to the corresponding cluster in the original trace. For instance, considering a trace of a system comprised by 100 users and 3 clusters, where *Profile 1* covers 60 users, *Profile 2* covers 25 users and *Profile 3* covers 15 users, then the probability of a user in the synthetic workload being mapped to each profile is 60%, 25% and 15%, respectively.

2) Generation of jobs per user: After associating each user of the synthetic workload to a profile, the values for each attribute are generated by a random number generation process using the probability distribution associated to the profile. Then, the values generated by each workload attribute are combined to create new jobs for each user.

In order to generate a new job for a user, first a random value for the Job Inter-Submission Time (JIST) is generated. It means that the submission time of a new job is the submission time of the previous job increased by the new JIST value generated. We consider that the first job of a user is submitted at a pre-defined workload start time increased by the first JIST value generated. After that, a random value for the Job Execution Time (JET) is generated. Recall that this corresponds to the sum of the execution times of all tasks that belong to the BoT job. After that, random values for Task Execution Time (TET) are generated until the sum of all execution times reaches the job execution time previously generated. The last task execution time generated is truncated in case the sum of all tasks exceed the job execution time. The number of tasks that composes the BoT job is the amount of task execution times that needed to be created to achieve the total job execution time.

3) Workload aggregation: The jobs generated by each user are aggregated to compose the entire grid workload. The process of generating new jobs is interrupted when a pre-defined condition is met. We defined this condition as the maximum duration of the synthetic workload, so when a new job submission time exceeds the desired duration, the process of generating jobs for that user is finished.

4) System workload generation: In the end, all generated jobs to be submitted to the grid are recorded, ordered by their submission time. The information included in the record includes the submission time of each job; the identification of the user to submit the job; and the list of tasks associated to the job with the time that takes to execute the task in a reference machine.

IV. Evaluation

In this section we present the evaluation of the workload model proposed. First we describe the six real grid computing traces used in the evaluation. After that, the evaluation methodology is presented, including the definition of a new metric to measure the quality of the modelling. Finally, we present the evaluation results as well as a comparison of the user-based model proposed with a state-of-the-art system-based model.

A. Workload Traces

Six workload traces from real grid systems available in the Grid Workload Archive were used in the model evaluation. Table I lists the workload traces used, providing information about the systems used for trace collection, as the grid location and type (academic, production or both).

B. Evaluation Methodology

The original grid traces do not contain information about the relation of tasks to BoT jobs, needed for the modelling. We applied the approach proposed by Iosup et al. [14] to group tasks in BoTs, in which tasks submitted by the same

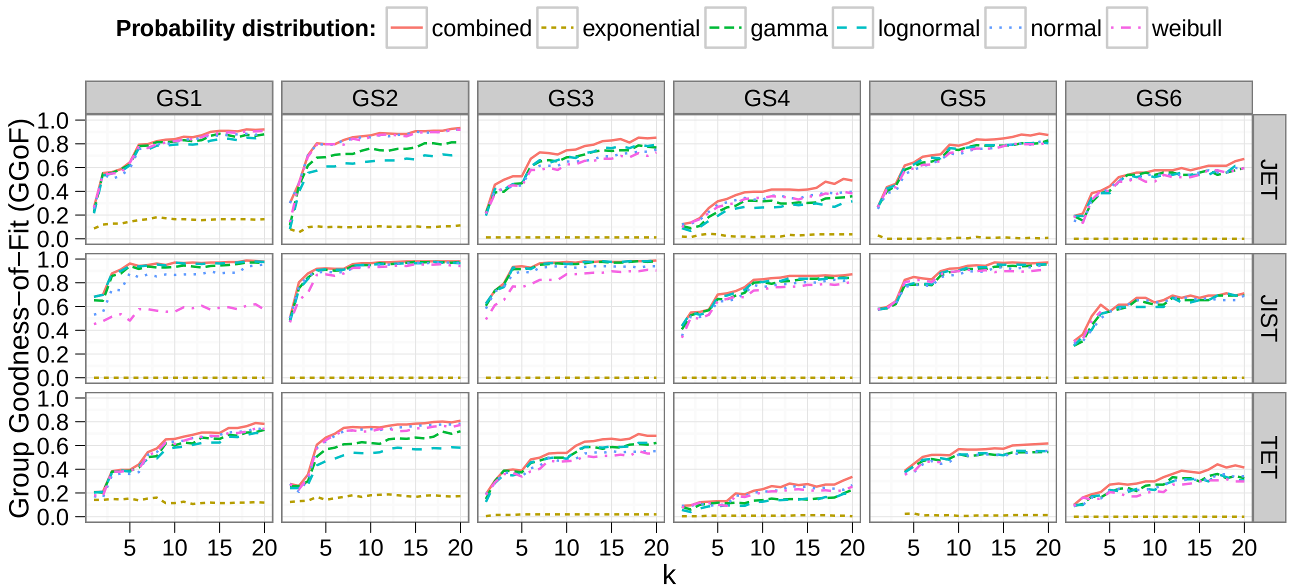

Figure 2. Group Goodness-of-Fit ($GGoF$) results for different fitting scenarios.

Table I
WORKLOAD TRACES USED IN THE MODEL EVALUATION [6].

ID	System (location, type)	Jobs	Sites	Users
GS1	DAS-2 (Netherlands, acad.)	602k	5	332
GS2	Grid'5000 (France, acad.)	915k	15	473
GS3	NorduGrid (Europe, both)	781k	75+	387
GS4	AuverGrid (France, prod.)	404k	5	405
GS5	SHARCNET (Canada, prod.)	1.2M	10	412
GS6	EGEE/LCG (Europe, prod.)	188k	220+	216

user with inter-submission times lower than 2 minutes are considered to be from the same BoT job.

As seen in Section III-A, each group of users for each workload attribute has fitted probability distributions associated to it. However, we need a metric to measure how good the model is for the chosen number of clusters and for each candidate distribution. Thus, we proposed a new metric to evaluate the quality of a model, which can be applied on any workload modelling approach that combines clustering and distribution fitting techniques. The metric is defined below:

- **Group Goodness-of-Fit** ($GGoF$): the fraction of users that obtained successful distribution fitting for a workload attribute according to a Goodness-of-Fit (GoF) test, considering there is a probability distribution that describes the attribute for each group of users. A GoF test result is considered successful if the p-value calculated is higher than a certain significance level.

We have chosen the Kolmogorov-Smirnov (KS) test as the GoF test with a significance level of 5% to calculate the GoF metric. For each workload attribute and each user, the KS test measures the distance between the ECDF of all attribute values for the user and the cumulative distribution function (CDF) of the probability distribution associated to the group the user belongs. The $GGoF$ metric is calculated for all configurations of number of clusters and candidate distributions.

A base-2 logarithmic transformation was applied on the workload attributes' data to reduce the impact of extreme values in the fitting. The transformation does not affect the conclusions about the fitting [5], [7], but a base-2 exponential function has to be applied on the data when generating synthetic workloads from the distributions.

For reproducibility, the tools developed for model extraction and synthetic workload generation implementing the modelling methodology proposed are available at: http://www.lsd.ufcg.edu.br/~marcus/grid-workload-model.

C. Model Evaluation Results

This section presents the results for the evaluation of the proposed model. The modelling was applied to 6 workload traces (GS1–GS6) and for the 3 workload attributes (JIST, TET and JET). Figure 2 shows the results for the $GGoF$ metric for different values of the number of clusters (k) and candidate distributions. Each graphic presents a scenario of workload trace and attribute. The different lines in the graphics represent the results for each candidate distribution listed

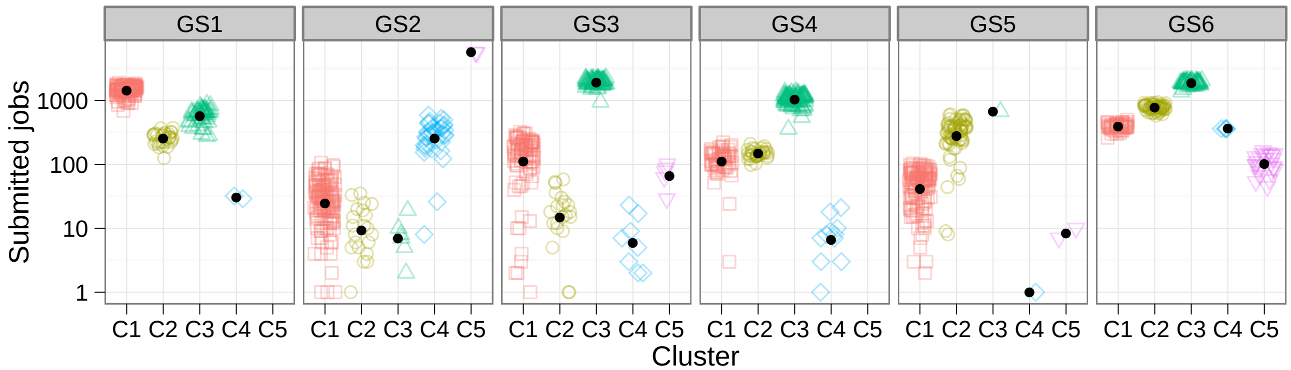

Figure 3. Submitted jobs per user divided in clusters for a 1 month synthetic workload and 6 different traces.

in Section III-A. Another line called *combined* is added to the distributions' list, which represents the combination of the best candidate distributions for each group according to the KS test result.

From the graphics we see that the $GGoF$ tends to increase when the number of clusters k is increased. It is an expected behaviour, as the higher the number of clusters, the higher the number of distributions used in the model and the lower the number of users per group. In this case, as each distribution is fitted for fewer users, the probability of having successful goodness-of-fit for all users is higher. In most of the cases, the $GGoF$ curves have a high increase rate for the first increments of k, but at some point the curves tend to reach a plateau where increasing k does not increase the $GGoF$ significantly.

The worst results are for the scenarios where only one group covers all users ($k = 1$). The fact that $GGoF$ is increased when increasing the number of clusters corroborates the importance of using clustering techniques on the workload modelling. Having only one group to represent all users makes the model more generic, but the modelling quality is much lower compared to higher values of k. On the other hand, even though having more clusters gives higher $GGoF$ values, it turns the model more complex and more specific for the base system users, as higher level patterns of users' activities are not defined. In this way, it is more appropriate to choose a number of clusters that is not so large, but also reaches high $GGoF$ values. In practice the number of clusters should be chosen around the value for which increments on k do not imply on significant increases on the quality of the model.

The exponential distribution has $GGoF$ values close to zero for all scenarios. This is a relevant result, as this distribution is largely used when modelling inter-arrival times, when it is assumed that the arrival rate comes from a Poisson distribution, and the inter-arrival times comes from an exponential distribution. However, in our evaluation the exponential distribution presented very low $GGoF$ results for all workload attributes, including job inter-arrival time.

There are no significant differences between the results of distributions other than the exponential, except in a few scenarios. However, using a combination of distributions as in the *combined* approach presents always the best results, as shown in the graphics. The *combined* approach was used in the application of the model to generate synthetic workloads. It is out of the scope of this paper to propose a technique to find the best number of clusters k for each attribute and trace. For the sake of simplicity, we chose $k = 5$ as the number of clusters to be used in the model application for all workload attributes. As we can see from Figure 2, there are not significant increases in $GGoF$ for number of clusters higher than 5.

D. Model Application

In this section we apply the proposed user-based workload model to generate synthetic workloads with different grid configurations. We analyse the characteristics of the user-based generated workloads and compare to a state-of-the-art system-based grid workload model.

We applied the model for the 6 traces described in Section IV-A, using 5 clusters ($k = 5$) for each workload attribute modelled. The synthetic workload generation process receives as input the number N of users to compose the grid workload.

Figure 3 presents the number of jobs submitted per user for a 1 month period of synthetic workloads, where each workload is configured to have $N = 200$ users. The users are divided in $k = 5$ clusters for the job inter-submission time attribute. The workloads are separated in different graphics by the traces used in the model extraction. The number of jobs presented in the y-axis are in log scale. The black filled points are the average number of submitted jobs for the users of each cluster, while the not-filled color symbols are the values for each user individually.

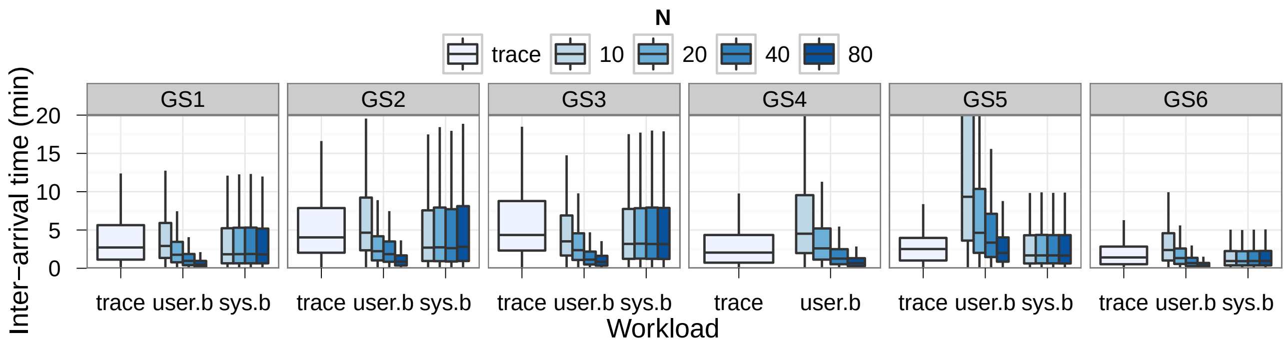

Figure 4. Box-plots for system's job inter-arrival times for real traces, user-based and system-based synthetic workloads.

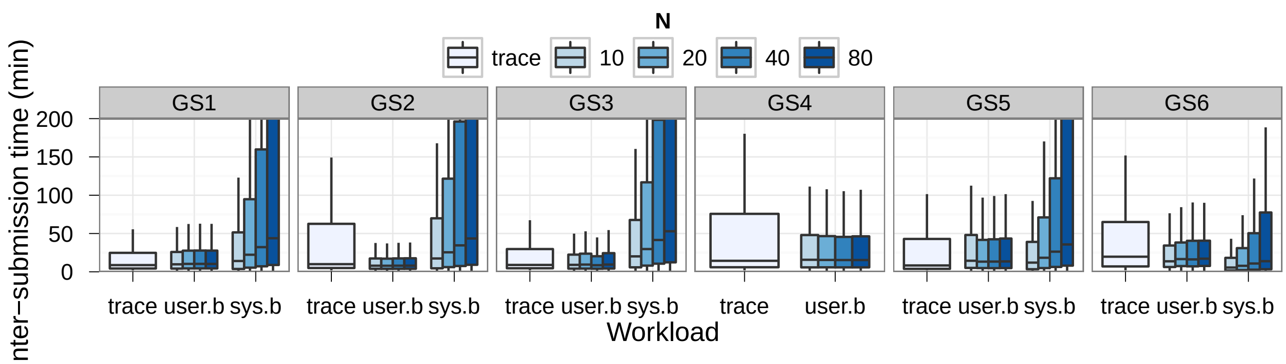

Figure 5. Box-plots for users' job inter-submission times for real traces, user-based and system-based synthetic workloads.

Note that there are different patterns on users' activities related to job submission rate. On the one hand, some users submitted more than 1,000 jobs in one month workload, as the users in the clusters $\langle GS1, C1\rangle$, $\langle GS2, C5\rangle$, $\langle GS3, C3\rangle$, and $\langle GS6, C3\rangle$, where the tuple $\langle GSi, Cj\rangle$ represents the cluster Cj of the workload model extracted from trace GSi. On the other hand, some groups of users submitted less than 10 jobs for the same time period, as the users in $\langle GS2, C2\rangle$, $\langle GS2, C3\rangle$, $\langle GS3, C4\rangle$, $\langle GS4, C4\rangle$, $\langle GS5, C4\rangle$, and $\langle GS5, C5\rangle$. We can also see that there is a difference on the number of users per cluster. For instance, if we look at the results for the model of trace $GS1$, we notice that the clusters $C1$ and $C3$ have a large number of points in a high density around the average point, which means that a large number of users are part of this cluster. On the other hand, cluster $C4$ has only a few users and cluster $C5$ has no user at all. It means that in the original trace the fraction of users clustered in these groups is very small, which results in a low probability of users in the synthetic workload being associated to these clusters. The difference in users' activities is an important feature to be represented in the evaluation of grid systems. A user-based workload model can be used in fine-grain studies to check how each user behaviour pattern affects the system, by changing the number of users in each group and analysing each group separately.

We also compared our *user-based* model with a state-of-the-art workload model for grid computing, proposed by Iosup et al. [7], which we called *system-based* model. In the system-based approach, the workload that arrives in the entire grid is modelled considering the source of jobs as a black box. The jobs are then assigned to users according to a Zipf distribution. We implemented the model described by Iosup et al. to generate the system-based synthetic workloads, using the distributions provided in their work [7].

Figures 4 and 5 show the box-plots of users' job inter-submission times and system's job inter-arrival time, respectively, for synthetic workloads with different numbers of users in the system (N), generated by the two different modelling approaches: the user-based model proposed in this paper (*user.b*) and the system-based model proposed by Iosup et al. (*sys.b*). We also show as a reference the box-plot for the values obtained in the original *trace*. Each graphic shows the synthetic workloads from different traces used in the extraction. The results of the system-based approach for the $GS4$ trace are not presented, as the original paper do not provide all distributions needed for the workload generation.

From Figure 4 we see that the system's job inter-arrival times for the *user-based* approach tends to decrease when the number of users in the system are increased, i.e. jobs arrive in the system in higher rates when there are more users submitting jobs. On the other hand, the system's job inter-arrival times for the *system-based* approach do not change when the number of users in the system increases, i.e. the system load is not affected when increasing the number of users submitting jobs. We believe that the system load should be increased when the number of users in the system is increased, as seen in the *user-based* approach.

We see in Figure 5 that the box-plots of users' job inter-submission times for the *user-based* approach are not affected when we change the number of users in the system, i.e. the users do not change their behaviour when the number of users changes. On the other hand, for the *system-based* approach the job inter-submission times tend to increase when the number of users are increased, i.e. the users submit jobs in lower rates when more users are active in the system. We believe that changing individual users' behaviour when varying the number of users in the system is an undesirable effect, as the users' activities resulted by this change may be unrealistic.

In this way, the *user-based* approach seems more appropriate than the state-of-the-art *system-based* approach, as we can easily change the system load by varying the number of users in the system, without affecting users' activities, thus preserving realistic users' behaviour.

V. Conclusions and Future Work

In this paper, we presented a user-based grid workload model which describes the users' activities according to their characteristics, clustered by users' behaviour profiles. We also proposed a new metric to measure the quality of models that combine clustering techniques with distribution fitting. The results show that there is an increase on the model quality when clustering users according to their activity, and extracting the model for each group of users, compared to the usual approach of having a single model for the entire workload (without clustering). We compared our user-based model with a state-of-the-art system-based model and showed that by using our user-based model the system load can be easily changed by varying the number of users in the system, creating different grid scenarios without affecting individual users' behaviour. We have also shown that varying the number of users in the system-based model does not affect the system load and changes the way individual users behave on the system, which compromises the model representativeness.

For future work, we would like to apply the modelling methodology to other grid workloads as well as other systems, further explore the trade-offs for choosing different number of clusters and analyse the impact of each user profile on system performance. Moreover, we would like to use synthetic workloads generated by the model proposed in this work to evaluate grid computing solutions.

Acknowledgment

The authors would like to thank the support from CAPES and CNPq, Brazil. They also thank the institutions that made the grid traces available through the Grid Workload Archive (GWA), as well as the GWA people for the initiative.

References

[1] C. Anglano, J. Brevik, M. Canonico, D. Nurmi, and R. Wolski, "Fault-aware scheduling for bag-of-tasks applications on desktop grids," in *IEEE/ACM International Conference on Grid Computing*, ser. GRID '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 56–63.

[2] W. Cirne, F. Brasileiro, D. Paranhos, L. F. W. Ges, and W. Voorsluys, "On the efficacy, efficiency and emergent behavior of task replication in large distributed systems," *Parallel Computing*, vol. 33, no. 3, pp. 213 – 234, 2007.

[3] D. Feitelson, *Workload Modeling for Computer Systems Performance Evaluation*. Book draft, 2011, vol. 0.34. [Online]. Available: http://www.cs.huji.ac.il/~feit/wlmod/

[4] M. Carvalho, R. Miceli, P. D. M. Jr., F. Brasileiro, and R. Lopes, "Predicting the quality of service of a peer-to-peer desktop grid," in *Workshop on Desktop Grids and Volunteer Computing Systems (PCGrid)*. IEEE Computer Society, 2010.

[5] U. Lublin and D. G. Feitelson, "The workload on parallel supercomputers: modeling the characteristics of rigid jobs," *Journal of Parallel and Distributed Computing*, vol. 63, pp. 1105–1122, November 2003.

[6] A. Iosup and D. Epema, "Grid computing workloads," *IEEE Internet Computing*, 2011.

[7] A. Iosup, O. Sonmez, S. Anoep, and D. Epema, "The performance of bags-of-tasks in large-scale distributed systems," in *International Symposium on High Performance Distributed Computing (HPDC)*, 2008, pp. 97–108.

[8] H. Li and R. Buyya, "Model-based simulation and performance evaluation of grid scheduling strategies," *Future Generation Computer Systems*, vol. 25, no. 4, pp. 460 – 465, 2009.

[9] N. M. Tran and L. Wolters, "Towards a profound analysis of bags-of-tasks in parallel systems and their performance impact," in *Proceedings of the 20th international symposium on High performance distributed computing*, ser. HPDC '11. New York, NY, USA: ACM, 2011, pp. 111–122.

[10] J. Zilber, O. Amit, and D. Talby, "What is worth learning from parallel workloads? a user and session based analysis," in *Proceedings of the 19th annual international conference on Supercomputing*, ser. ICS '05. New York, NY, USA: ACM, 2005, pp. 377–386.

[11] E. Shmueli and D. G. Feitelson, "On simulation and design of parallel-systems schedulers: Are we doing the right thing?" *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, pp. 983–996, 2009.

[12] E. Shmueli and D. Feitelson, "Using site-level modeling to evaluate the performance of parallel system schedulers," in *14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, sept. 2006, pp. 167 – 178.

[13] B. Javadi, D. Kondo, J.-M. Vincent, and D. P. Anderson, "Discovering statistical models of availability in large distributed systems: An empirical study of seti@home," *IEEE Transactions on Parallel and Distributed Systems*, vol. 99, 2011.

[14] A. Iosup, M. Jan, O. Sonmez, and D. Epema, "The characteristics and performance of groups of jobs in grids," in *Euro-Par*. Springer-Verlag, 2007, pp. 382–393.

[15] D. Kondo, B. Javadi, A. Iosup, and D. Epema, "The failure trace archive: Enabling comparative analysis of failures in diverse distributed systems," *IEEE International Symposium on Cluster Computing and the Grid (CCGrid)*, pp. 398–407, 2010.

[16] R. B. D'agostino and M. S. Stephen, Eds., *Goodness-of-Fit Techniques*. New York and Basel: Marcel Dekker, 1986.

[17] D. Nurmi, J. Brevik, and R. Wolski, "Modeling machine availability in enterprise and wide-area distributed computing environments," in *Euro-Par*, 2005, pp. 432–441.