Optimising resource costs of cloud computing for education

Fernando Koch, Marcos Dias de Assuncao, Carlos Cardonha, Marco Netto

Optimising Resource Costs of Cloud Computing for Education

Fernando Koch[a], Marcos D. Assunção[b], Carlos Cardonha[c], Marco A. S. Netto[c]

[a]*SAMSUNG Research Institute*
[b]*INRIA, LIP, ENS de Lyon*
[c]*IBM Research*

Abstract

There is a growing interest around the utilisation of cloud computing in education. As organisations involved in the area typically face severe budget restrictions, there is a need for cost optimisation mechanisms that explore unique features of digital learning environments. In this work, we introduce a method based on Maximum Likelihood Estimation that considers heterogeneity of IT infrastructure in order to devise resource allocation plans that maximise platform utilisation for educational environments. We performed experiments using modelled datasets from real digital teaching solutions and obtained cost reductions of up to 30%, compared with conservative resource allocation strategies.

Keywords:
cloud computing, education, digital teaching

1. Introduction

Digital teaching requires new methods to continuously evaluate student performance [1]. These methods revolve around collecting, classifying, and understanding events that happen during in-classroom activities [2]. They require the instrumentation of learning environments to generate multi-dimensional signals capable to define key contextual elements. This approach generates large amounts of data that need intense computing power and storage. As a solution, Sclater envisions that "the majority of educational services will be hosted in the cloud and institutions no longer host their own data centres with expensive hardware, power bills, staff salaries and computing resources which are rarely fully utilised" [3]. A challenge in this context is to balance resource demand, expected quality of services, and operational costs thus making the use of technology viable for the education environment.

In terms of cloud computing, this means to minimise the number of allocated resources subject to keeping quality of service at an acceptable level [4]. We claim that, given the unique features of digital education, one can devise mechanisms of resource allocation tailored for this domain. For instance, it is possible to estimate the number of resources required by a classroom during a specific class based on information such as features of the learning objects in digital education material, number of students, and historical resource demand. Traditional methods, however, estimate the *peak usage* and allocate resources considering a safety margin over the worst-case scenario; this over-allocation approach results in large and undesired resource waste. The work presented by Koch *et al.* compared different allocation strategies and evaluated their expenditures and impact upon Quality of Service (QoS) [5].

In order to optimally exploit the cost-effectiveness of cloud computing in education, we propose a probabilistic method that allows fine-grained adjustments of load forecast models and hence enables significant cost reductions in a pay-as-you-go business model. We consider the number of resources w_c for delivering a class c, a prime α_r of demand fluctuation based on limitations of the cloud infrastructure supporting activities in classroom r, and a prime β of safety margin which is adjusted according to the confidence level of the estimations. Special care must be taken with such methods, though, as they bring larger risks to QoS. The research questions addressed in this paper are:

- How to adjust prime α_r in order to optimise resource allocation?
- How to adjust prime β to achieve acceptable QoS levels?

To address these research questions, we constructed

scenarios based on real-world digital teaching initiatives. In particular, we evaluated these scenarios considering fluctuations of infrastructure availability, which is typical in developing countries. The proposed method is analysed via discrete-event simulations considering various numbers of classrooms and using resource savings and QoS violation as metrics.

The contributions of the paper are the following:

- A probabilistic resource allocation method for educational institutions to optimise their use of cloud resources;
- An evaluation of the method under several scenarios using data based on existing digital teaching initiatives.

The article is structured as follows. Section 2 introduces the motivation of this work by describing how resource demand behaves in a real world implementation of a digital teaching platform and presents the formal description of the problem. Section 3 describes the probabilistic algorithm used for resource allocation. Results of a computational evaluation involving the proposed method are presented in Section 4. Section 5 contains the description of related work in the literature, and Section 6 presents our conclusion.

2. Motivation and Problem Description

Digital teaching provides means for instrumenting learning environments and novel methods to collect, classify, and understand in-classroom learning activities. The workload of such systems varies over time depending on the context and elements composing the delivered education material. For example, there are demand peaks throughout the delivery of a class when the material comprises videos, pictures, tests, screen sharing, and so forth. Moreover, fluctuations of network availability highly influence the flow of incoming requests, which leads to an undesired decrease in resource demand. When using resources from a cloud, these nuances must be considered when optimising allocation of resources in order to minimise cost and avoid waste.

The scenario considered here is that of a service provider—or educational organisation—that needs to automatically allocate resources from a cloud to deliver education services required by a school or university. We considered Samsung School solutions in this article, a real world implementation of a digital teaching platform. The addressed problem can formally be defined as follows. Let $\mathcal{R}$ denote a set of classrooms and C denote a set of classes. We assume that all classes are presented in each classroom exactly once over T time-slots, so that we denote by $S_{r,t}$ the class c being taught in classroom r at the t-th time-slot, $1 \leq t \leq T$. Let us consider that there is a set of learning objects $L(c)$ associated to class c. Each object l is a media element of type $m(l)$, where type in this context may refer to text, image, and video. Let us denote the amount of resources consumed by learning objects of type m by $w(m)$. The sequence of events are as follows:

1. Educator signs into a classroom r at time-slot t and confirms that class $c = S_{r,t}$ will be delivered.
2. Students located in r sign in and the applications running on their devices load the links to content in $L(c)$.
3. Educator starts the class.
4. Educator requests students to go to specific objects or pages, act upon objects, respond to tests, watch videos, *etc*.
5. Students react to educator's command in heterogeneous ways, depending on the behaviour of the cloud infrastructure supporting material delivery in r and their level of engagement.
6. The cycle loops to Step 4 until the class ends (*i.e.*, until the end of time-slot t).
7. Applications upload log files reporting all activities during the class.
8. Students and educators are prepared to start activities scheduled for time-slot $t + 1$.
9. The cycle loops to Step 1, if more classes exist.

It is clear that peak load can happen at distinct points, such as when the applications load the material (Step 2), when students act upon the content (Step 5), and when the application uploads the log files for processing (Step 7). Thus, the maximum resource demand per student throughout class c is roughly $\max_{l \in L(c)} w(m(l))$, and if the number of students located in classroom r is $n \in \mathbb{N}$, then the maximum resource demand of c is given by

$$w_c = n \left(\max_{l \in L(c)} w(m(l)) \right).$$

One may infer c from r and t given S. We will use $w_{r,t}$ and w_c interchangeably whenever $c = S_{r,t}$. We remark that maximum resource demand w_c is achieved if all students access the most resource-demanding learning content simultaneously.

Ideally, each class c is *expected* to have maximum demand w_c, independently from the classroom where it is being presented. However, as classrooms may be located in different regions and, consequently, subject to

different IT infrastructure, deviations on w_c may occur. For instance, in places where data transmission is inefficient, students may not be able to access the content smoothly. In these situations, *allocated resources may be underused*.

The goal of an optimum resource allocation method is to maximise system utilisation while delivering good QoS, where QoS is harmed whenever the number of allocated resources is insufficient for the load requirements.

We propose a method that considers usage variations caused by cloud infrastructure issues to reduce overallocation and set adequate safety margins that reduce risk of QoS degradation. This technique is specially useful in emerging countries such as Brazil, where fluctuations of data communication availability is a common reality; moreover, a successful implementation of this technique will rationalise the cost factor around cloud computing for education.

3. Probabilistic Workload-Aware Dynamic Resource Allocation

Let $w'_{r,t} = w'_c$ denote the actual resource consumption demand of class $c = S_{r,t}$ for classroom r, as explained in Section 2. This value can be smaller than w_c in cases where the underlying IT infrastructure r does not deliver optimal service. That is, quality of infrastructure influences directly upon resource utilisation. This scenario emerges in schools with poor Internet connection, as execution of specific content may be impacted by the slow communication—thus students give up from playing the content, consequently curbing the data load. Moreover, for each time-slot t, let

$$w_t = \sum_{r \in \mathcal{R}} w_{r,t}$$

denote the total expected number of resources and

$$w'_t = \sum_{r \in \mathcal{R}} w'_{r,t}$$

denote the actual resource demand at t

We assume that deviations on w_c for each class c presented in classroom r are given by a multiplicative factor α_r drawn from a gaussian distribution $\mathcal{N}(\mu_r, \sigma_r^2)$ whose values are truncated on the interval $[0, 1]$, that is, $w'_{r,c} = \alpha_r w_c$. Parameters μ_r and σ_r^2 depend only on the classroom r, *i.e.*, we assume that the variations are dependent exclusively on the classrooms, and not on the classes.

We extend the algorithm presented in our previous work by considering values α_r for the estimation of the required number of resources from the cloud [5]. In real-world settings, it is typically not possible to know *a priori* the values of μ_r and σ_r^2, so we employ *Maximum Likelihood Estimation* (MLE) to compute the values of these parameters for each r and for each time-slot t based on the actual values of α that have been observed in previous time-slots. Namely, our approach "guesses" a resource consumption value $w^*_{r,t}$ for each pair (r, t), so it assumes that the total allocation will be given by

$$w^*_t = \sum_{r \in \mathcal{R}} w^*_{r,t}.$$

The algorithm also employs a safety margin β for resource allocation, which changes over time according to the quality of forecast results but never goes below a given constant β' (*e.g.*, $\beta' = 1.3$ represents a minimum margin of 30%). Therefore, the actual allocation will be given by $\beta_t w^*_t$, where β_t denotes the current value of β at time-slot t.

The pseudo-code of the resource allocation algorithm is presented in Algorithm 1. As input data, it receives parameters $\mathcal{R}$, C, S, and T and as output it delivers a **QoS violation** value Q, which is given by

$$Q = \frac{1}{T} \sum_{t \in [1,T]} \frac{\max(w'_t - \beta_t w^*_t, 0)}{w'_t}.$$

Q can be interpreted as follows: in each time-slot t, if $\beta_t w^*_t < w_t$, the number of resources is insufficient, so a fraction of the students, given by $(w'_t - \beta_t w^*_t)/w'_t$, will suffer with bad QoS. The value of Q is the sum of these fractions for every time-slot t in $[1, T]$ divided by T, so it is equal to the average percentage of students that will receive bad QoS per time-slot.

Variables Q and β are initialised with 0 and 0.2, respectively (Lines 1 and 2). Moreover, the algorithm maintains a "list of lists" α' that, in each time-step t, will be given by $\alpha'_r = \{\alpha'_{r,1}, \alpha'_{r,2}, \ldots, \alpha'_{r,t-1}\}$, with

$$\alpha'_{r,t'} = \frac{w'_{S_{r,t'}}}{w_{S_{r,t'}}}$$

for $1 \leq t' < T$. For each r, α'_r is initialised as an empty list (Lines 3-4).

Algorithm 1 main loop iterates over each time-step t (Lines 6–17). Initially, it obtains an estimation of resource demand using Algorithm 2, described below (Line 7). Afterwards, it computes the actual resource demands (Lines 8 and 10) and updates α' by appending $\alpha'_{r,t}$ to each list α'_r (Line 10). Once the difference Δ

Algorithm 1: Proposed resource allocation.

Input: $\mathcal{R}, C, S, T$
Output: QoS estimation

1 $\beta \leftarrow 0.2$;
2 $Q \leftarrow 0$;
3 $\beta' =$ // administrator' specified minimum safety margin (*e.g* 1.3);
4 **for** *r in* $\mathcal{R}$ **do**
5 $\quad \alpha'_r = []$;
6 **for** *t in* $[1, T]$ **do**
7 $\quad w^*_t =$ **Algorithm 2**$(\mathcal{R}, C, S, t, \alpha')$;
8 $\quad w'_t \leftarrow 0$;
9 $\quad$ **for** *r in* $\mathcal{R}$ **do**
10 $\quad\quad w'_t \leftarrow w'_t + w'_{r,t}$;
11 $\quad\quad \alpha'_r.append(\alpha'_{r,t})$;
12 $\quad \Delta = \beta w^*_t - w'_t$;
13 $\quad$ **if** $\Delta > 0$ **then**
14 $\quad\quad \beta \leftarrow \max(\beta(1.0 - s'), \beta')$;
15 $\quad$ **else**
16 $\quad\quad \beta \leftarrow (1.0 + s)\beta$;
17 $\quad\quad Q \leftarrow Q - \Delta / w'_t T$;
18 **return** Q;

Algorithm 2: Estimation of resources demand.

Input: $\mathcal{R}$, C, S, time-slot t, α'
Output: Number w^*_t of resources to be allocated

1 $w^*_t \leftarrow 0$;
2 **for** *r in* $\mathcal{R}$ **do**
3 $\quad c = S_{r,t}$;
4 $\quad \mu_r, \sigma^2_r \leftarrow MLE(\alpha'_r)$;
5 $\quad \alpha_r = draw(\mathcal{N}(\mu_r, \sigma^2_r))$;
6 $\quad w^*_t \leftarrow w^*_t + w_c \alpha_r$;
7 **return** w^*_t;

between the estimated and the actual resources demand has been computed (Line 12), the algorithm adjusts the value of β according to the progression of this error. Namely, if the error in t is greater than 0, β becomes the maximum between $(1.0 - s')\beta$, $s' \geq 0$, and β' (Line 14). Both s and s' are constants that contain the value 0.005. Conversely, if it is smaller than 0, β is multiplied by $(1.0+s)$ and Q is incremented by $-\Delta/(w'_t T)$ (Lines 16 and 17). Finally, after the end of the loop, Q is returned.

Algorithm 2 is used to estimate the number of required resources. As input data, it receives the set $\mathcal{R}$ of classrooms, the set C of classes, the schedule S, the time-slot t for which the allocation is being performed, and α'. Initially, it sets w^*_t to zero (Line 1). Then, for each classroom r (Lines 2-6), the algorithm retrieves the class c that is going to be presented at r in time-step t (Line 3). Afterwards, it estimates the values of parameters μ_r and σ^r_R characterising the normal distribution that describes α_r using MLE on α'_r (Line 4). Once μ_r and σ^2_r have been computed, the algorithm draws a value α_r from $\mathcal{N}(\mu_r, \sigma^2_r)$ (Line 5) and w^*_t is incremented by the product of the estimated resource consumption w_c and α_r (Line 6). For a class r for instance, if $\alpha_r = 0.7$ and the maximum number of resources for delivering the class $w_c = 100$, then w^*_t is incremented by 70 resources. After the end of the main loop, the method returns the number w^*_t of resources required in time-slot t (Line 7).

We assume that the behaviour of the IT infrastructure is stable along the delivery of material in each classroom r. That is, μ_r and σ^2_r do not change over time. We remark that the method can be easily adapted to situations where this assumption does not hold by simply restricting the application of MLE to the last values of vector α'_r, eliminating hence the influence of values w'_c/w_c that do not reflect the current distribution.

4. Performance Evaluation

4.1. Experimental Setup and Metrics

In order to perform experiments in a controlled and repeatable manner we generated a set of synthetic workloads that resemble the log files created by the digital teaching problem described in Section 2. We consider a number of classrooms, with each having a maximum number of students who attend a number of classes per day. As detailed later, a generated workload contains the hourly demand for number of resources for each classroom over a number of days.

We associate the conduction of each class in a classroom to a time-step in T, so an iteration of Algorithm 1's main loop (Lines 5-16) is executed before all classes taking place in the associated time-step. In the beginning of each iteration, Algorithm 2 estimates the number of resources that will be required by the set of classes being conducted in the associated time-step. Estimations made in Algorithm 2, safety margin β, and QoS violation are updated in each iteration according to actual demands registered in previous steps.

To emulate the difficulty that students located in a given classroom may experience while accessing the educational content due to infrastructure issues, we sort parameters $\mu_r \in [0.5, 1.0]$ and $\sigma^2_r \in [0.01, 0.2]$ uniformly for each classroom r. Then, in each time-slot t,

we draw $\alpha_{r,t}$ from the gaussian distribution $\mathcal{N}(\mu_r, \sigma_r^2)$ truncated at the interval $[0, 1]$ and multiply it by the maximum number of students in the class [6]. If $\alpha_r = 0.8$, only 80% of the students in the class are able to properly access the educational content.

We generate the set of classes C considering the average content used by Samsung School solutions in Brazil. We vary the number of classes rather than subject areas and teacher behaviour in order to enable a more fair comparison of the allocation models. Each subject consists of a number of pages uniformly distributed between 18 and 25; each page is accessed at a specific time during class and contains from 6 to 12 learning objects, which can be text, image, or video. Typically, approximately 20% of the learning objects are text, 30% are images, and 50% are videos. In our experiments, we purposefully extended the amount of video considering new pedagogical models being applied in digital teaching solutions and the fact that this type of learning object constitutes the *de facto* standard of add-on support material. For the sake of this analysis, let us consider that the objects' sizes are pre-established: texts are between 2KB and 8KB; images are between 180KB and 1.5MB, and videos are between 6MB and 15MB. These numbers are aligned with actual parameters observable in Samsung School solutions. Peak load is computed based on (i) the number of students able to access the content during a class, and (ii) the number of HTTP requests (*i.e.*, HTTP chunks) required to transfer the content into the students' devices. The peak parameter is applied to determine the maximum resource demand of the class.

We evaluate two performance metrics to determine the provided QoS levels and the resource savings achieved by the proposed allocation method:

Resource saving: the ratio between the amount of resources allocated by the algorithm, given by $\beta \sum_{t\in[1,T]} w_t^*$, and the maximum estimated amount of resources, given by $\sum_{t\in[1,T]} w_t$.

QoS violations: the value Q defined in Section 3.

Finally, for the sake of comparability, we remark that "conservative" approaches allocate resources according to $\sum_{t\in[1,T]} w_t$ (possibly with an additional safety margin) and, therefore, typically have $Q = 0$, since they avoid situations with insufficient resource allocation.

4.2. Results and Analysis

We evaluate the performance of the proposed allocation method by considering scenarios with various numbers of classrooms. In the first set of experiments $|\mathcal{R}|$ varies from 1 to 301 (with step size 10) and different values for the minimum safety margin β' are considered. We argued in our previous work that a safety margin of 30% (*i.e.*, $\beta' = 1.3$) is commonplace in the industry [5], and therefore we included it for comparison. Note that the margin is necessary not only because of possible errors with predictions, but also because Cloud resources may not offer a steady performance [7].

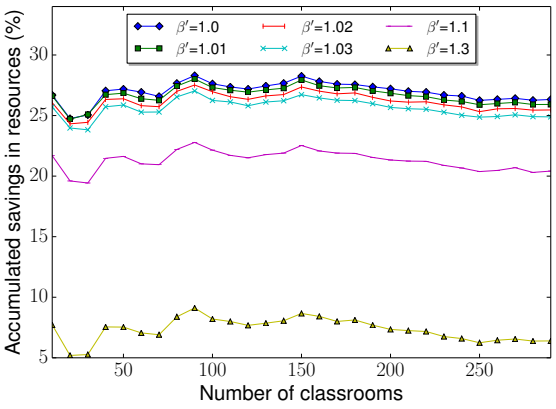

Figure 1: Resource savings under various numbers of classrooms ($\sigma_r^2 \in [0.01, 0.2]$); where β' is the minimum resource safety margin.

Figure 1 shows resource savings under various numbers of classrooms and values of β', where the savings are computed as the difference between the resources required to handle peak load and the actual provision. Initially, we observe that, not surprisingly, low values of β' lead to large savings, while $\beta' = 1.3$ is less effective. In most cases, savings are above 25%, except under certain scenarios where the number of classrooms is small.

This difference between instances with small and large values of $|\mathcal{R}|$ can be explained by the possibilities that large $|\mathcal{R}|$ provides to *error cancellation*. In each time-slot t, our algorithm assumes that the amount of resources that will be used by classroom r is $w_{r,t}^* = w_{r,t}\alpha'_r$, where α'_r is a random variable drawn from a truncated gaussian distribution. It is clear that $w_{r,t}^*$ will almost always be wrong, but since errors in the estimation of the distribution decrease over time, deviations tend to decrease. Moreover, $w_{r,t}^* \geq w'_{r,t}$ in some cases and $w_{r,t}^* \leq w'_{r,t}$ in others, so underestimations are frequently compensated by overallocation. For higher values of $|\mathcal{R}|$, such cancellations happen frequently, whereas extreme cases with just 1 classroom depend strongly on the accuracy of its distribution's estimation. This strategy fits very well in digital teaching environments as it is pos-

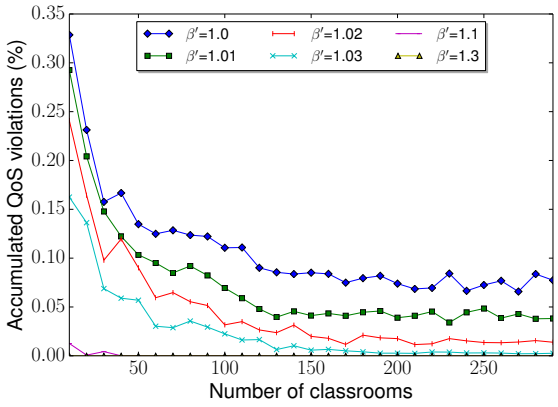

Figure 2: QoS violations under various numbers of classrooms ($\sigma_r^2 \in [0.01, 0.2]$); where β' is the minimum resource safety margin.

sible to have a fairly accurate estimate on resource usage since classes happen in pre-defined time slots and their content are known in advance. Apart from that, classes happen in parallel, thus amortising error among the classrooms. It is also important to notice that we included experiments with small numbers of classrooms for the sake of completeness, even though the average number of classrooms in public schools in Brazil is often greater than the point where the proposed method presents substantial savings.

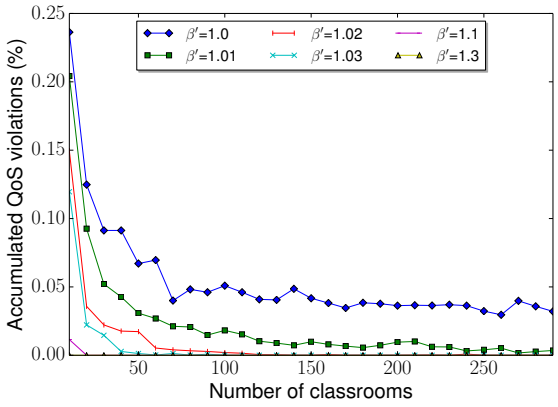

Figure 3: QoS violations under various numbers of classrooms ($\sigma_r^2 \in [0.01, 0.1]$); where β' is the minimum resource safety margin.

Figure 2 presents QoS violations for scenarios with varying number of classrooms; for the sake of better visualisation the values have been multiplied by 100. Two observations should be made regarding these results. First, the accumulated QoS violation is considerably low (always under 2%), and this holds for all the values of β'. Second, direct comparison between curves show that adding 0.01 to β' reduces considerably the percentage of QoS violations. This fact is a direct consequence of the relatively large standard deviation values; clearly, in cases where σ is large, mistakes in predictions are more likely to occur and to be larger. Therefore, for scenarios where β' is too low, the adjustments it yields may not be sufficient to overcome an overall estimation $w^*_{r,t}$ that was too low.

The observations above motivated an investigation of scenarios where σ_r^2 assume significantly smaller values. Figures 3 and 4 show the results of experiments in scenarios where $\sigma_r^2 \in [0.01, 0.1]$ for each r in $\mathcal{R}$. Figure 3 corroborates the discussions above by showing that QoS violations are significantly smaller in scenarios with low variance (as presented in Figure 2). Conversely, improvements on QoS typically lead to more costs, and this is exactly what can be observed in a direct comparison between Figure 4 (lower variances) and Figure 1 (higher variances). In resume, we observe a clear trade-off between QoS and resource savings, and this aspect becomes more evident in scenarios where $|\mathcal{R}|$ is small (as discussed below).

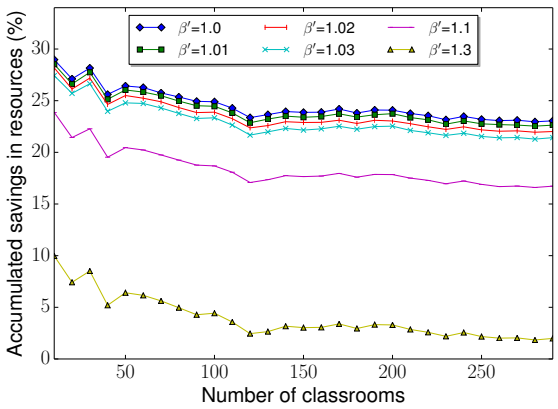

Figure 4: Resource savings under various numbers of classrooms ($\sigma_r^2 \in [0.01, 0.1]$); where β' is the minimum resource safety margin.

Figure 2 shows that QoS violation is very close to zero for instances with 10 and 20 classes if the algorithm is set with $\beta' = 1.3$ and $\beta' = 1.1$, respectively. Higher values of β' clearly improve QoS, but as lower values lead to significant resource savings (up to 20% in some scenarios), we decided to investigate the trade-off between QoS and resource savings for relatively small values of β' and $|\mathcal{R}|$, *i.e.*, $1 \leq \mathcal{R} \leq 70$ and $\beta' \in \{1.04, 1.06, 1.08, 1.1, 1.2\}$. The results are reported

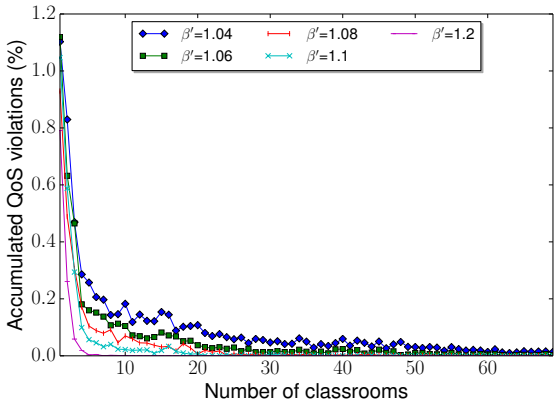

Figure 5: QoS violations between 1 to 70 classrooms ($\sigma_R^2 \in [0.01, 0.2]$); where β' is the resource minimum safety margin.

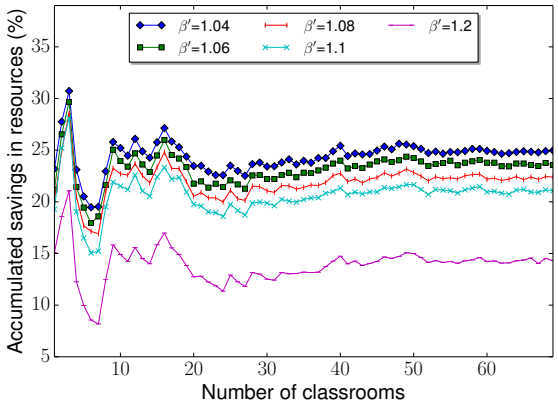

Figure 6: Resource savings between 1 to 70 classrooms ($\sigma_R^2 \in [0.01, 0.2]$); where β' is the resource minimum safety margin.

in Figures 5 and 6, which show QoS levels and resource savings, respectively. Both figures suggest that there is a certain "critical mass" related to the quality of our algorithm; in particular, the curves reach a certain stability when $|\mathcal{R}|$ becomes larger than 30 due to error cancellation.

To further explain the reasons of these results, we note that scenarios with few classrooms reduce the possibilities of error cancellation, making hence the performance of our method more unpredictable and more susceptible to the mean and, more important, to the variance of the distributions. For instance, a sequence of time-steps where most deviations are either positive or negative lead to "abnormalities" such as the one observed with $|\mathcal{R}| \approx 6$, where QoS violations and resource savings abruptly increased and decreased, respectively. Therefore, scenarios with small values of $|\mathcal{R}|$ make our method more susceptible to the values of σ^2. Nevertheless, our results show that these effects can be mitigated if we set $\beta' = 1.1$ because, in this case, QoS violations are negligible and resource savings are significant (above 20%). In summary, higher values of β' make the MLE method less sensitive to the parameters of the distributions and reduce QoS violations at the expense of additional expenses with resources.

Finally, we observe that the MLE method in Algorithm 2 has a convergence rate that is satisfactory for digital teaching scenarios. Namely, after few time-steps (approximately 10), it is possible to observe that the effects of noise start to be adequately filtered out and parameters (μ_r, σ_r^2) become indeed subject to fine-grain adjustments. Typically, after 20 time-steps, significant changes on (μ_r, σ_r^2) cease to happen, suggesting hence that the algorithm is also suitable for scenarios where the behaviour of the IT infrastructure may change over time.

5. Related Work

Cloud Computing is being used by educational institutions as a platform for offering modern and up-to-date IT resources to students [8, 9]. This is particularly important in developing countries and for meeting the limited budgets that institutions often have as a result of the current economic turmoil [10, 11, 12, 13]. The 2010 UNESCO Report points out that Cloud computing offers opportunities for cost reduction due to the economies of scale, thus resulting in a shift away from locally-hosted services [3]. The same report highlights the benefits of cloud computing for institutions and students. Apart from the claimed benefits of cost reduction, elasticity [14], and concentration on core business, the report mentions enhanced resource availability, better end-user satisfaction, and augmented learning process and collaboration. Cloud is also an interesting mechanism for schools to use software licenses over the Internet [15].

Another study has focused on the opportunities of cloud computing to increase collaboration among multiple institutions [16]. In addition, as discussed by Sultan, there are several examples of educational institutions that have adopted cloud computing not only to rationalise the management of IT resources, but also to make the education process more efficient [13].

Cost reductions and quality of service are key factors for educational institutions and are impacted by how cloud providers manage their resources. Having

appropriate tools for doing so is an important differentiator. The following projects, for instance, have investigated aspects related to Service Level Agreements (SLAs) and load prediction methods for optimising resource management. Emeakaroha *et al.* investigated monitoring time intervals for detecting SLA violations and for informing the resource allocation system of such violations [17]. The solution is reactive and does not use service workload for proactively predicting resource consumption. Li et al. introduced an approach to optimal virtual-machine placement for predictable and time-constrained load peaks [18]. The solution, although focuses on a proactive resource allocation using prediction techniques, does not leverage specific information about the workload domain. Similar approaches were investigated by Ali-Eldin *et al.* [19]. McGougha *et al.* compared on-premise and cloud resources [20]—such a study is relevant for better understanding the cost-benefit of moving workloads to the Cloud.

As in other domains, educational institutions can also utilise hybrid clouds to balance their workloads between on-premise and remote infrastructure [21, 22, 23]. The selection of a cloud can range from a single provider to multiple providers. For the latter, recommendation systems could be leveraged to select providers [24, 25].

Bodenstein *et al.* have focused on resource allocation decisions, ignoring application information to predict when resource allocation should be adapted [26]. Gong *et al.* introduced a system called PRedictive Elastic ReSource Scaling (PRESS), which aims at avoiding resource waste and service level objective violations in the context of cloud computing [27]. Their goal is to avoid the use of application profiling, model calibration, and understanding of user applications. Our work takes another direction where cloud customers provide information about their workloads to avoid SLA violations and reduce resource waste. Gmach *et al.* also investigated capacity planning using historical data, but without considering the nature of the workload [28]. Other projects [29, 30] have also explored the use of resource consumption prediction to better allocate resources. However they have not considered IT cost reductions and QoS in their studies. Adaptive resource allocation and demand prediction have also been explored in Grid and cluster computing environments [31, 32, 33, 34].

Our work exploits a gap in the state-of-the-art for a method to assess the impact of using specific domain information of a workload to assist resource allocation considering both IT costs and QoS for educational institutions.

6. Conclusion

We presented in this work a probabilistic resource allocation method that can be specially tailored for cloud computing environments providing services to education institutions. The proposed method explores the fact that IT infrastructure instrumenting physical classrooms may be heterogeneous and may prevent students from accessing the education material, which leads to under used resources. The method improves system utilisation (and, simultaneously, reduce allocation costs) at the expense of a minor impact on QoS.

For our evaluation, we generated datasets reproducing scenarios that are similar to those identified in real-world digital teaching initiatives. Using resource savings and QoS violation as metrics, we investigated several configurations, typically characterised by the number of classrooms and the minimum safety margin being employed. Results show that error cancellation plays a major role and allows for considerable cost reductions and that QoS violation is small even in situation where large deviations in the behaviour of the IT infrastructures are to be expected. This error cancellation is possible in digital teaching environments due to two main reasons: (i) it is possible to have a fairly accurate estimate on resource usage since classes happen in predefined time slots and their content are known in advance; and (ii) classes happen in parallel, thus making error amortisation possible among the classrooms.

The results of our experiments allow us to conclude that the probabilistic resource allocation method is very satisfactory, since it is able to deliver allocation plans which are considerably more economic and largely compensate the marginal impacts they have on QoS when compared with a typical worst-case-oriented approach. Finally, we believe that the performance of this algorithm has the potential to motivate several education institutions to employ cloud solutions to deliver electronic material to their students.

Acknowledgement

This material is based upon work supported by the FINEP under Contract 03.11.0371.00, MCT/FINEP/FNDCT 2010, related to the project "Platform for the Development of Accessible Vocational Training". Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of FINEP or any other related institution. Part of the work was performed when Fernando Koch and Marcos D. Assunção were at IBM Research.

References

[1] C. Dede, J. Richards, Digital Teaching Platforms: Customizing Classroom Learning for Each Student, Teachers College Press, 2012.

[2] F. Koch, C. Rao, Towards massively personal education through performance evaluation analytics, International Journal of Information and Education Technology 4 (4) (2014) 297–301.

[3] N. Sclater, Cloud computing in education, Iite policy brief, UNESCO Institute for Information Technologies in Education (September 2010).

[4] A. Quarati, D. D'Agostino, A. Galizia, M. Mangini, A. Clematis, Delivering cloud services with qos requirements: an opportunity for ict smes, in: Proceedings of the 9th International Conference on Economics of Grids, Clouds, Systems, and Services (GECON'2012), Springer, 2012, pp. 197–211.

[5] F. Koch, M. D. Assunçao, M. A. Netto, A cost analysis of cloud computing for education, in: Proceedings of the International Conference on Economics of Grids, Clouds, Systems, and Services (GECON'12), Springer, 2012, pp. 182–196.

[6] V. Mazet, Simulation d'une distribution gaussienne tronquee sur un intervalle fini, Technical report, Universite de Strasbourg/CNRS (2012).

[7] J. O'Loughlin, L. Gillam, Performance evaluation for cost-efficient public infrastructure cloud use, in: Proceedings of the 11th International Conference on Economics of Grids, Clouds, Systems, and Services (GECON'2014), Springer, 2014.

[8] R. Katz, The tower and the cloud: Higher education in the age of cloud computing, Educause, 2010.

[9] H. Katzan Jr, et al., The education value of cloud computing, Contemporary Issues in Education Research (CIER) 3 (7) (2010) 37–42.

[10] N. Kshetri, Cloud computing in developing economies, Computer 43 (10) (2010) 47 –55.

[11] S. Greengard, Cloud computing and developing nations, Communications of the ACM 53 (5) (2010) 18–20.

[12] M. Mircea, A. Andreescu, Using cloud computing in higher education: A strategy to improve agility in the current financial crisis, Communications of the IBIMA 53 (5).

[13] N. Sultan, Cloud computing for education: A new dawn?, International Journal of Information Management 30 (2) (2010) 109–116.

[14] P. D. Kaur, I. Chana, A resource elasticity framework for qos-aware execution of cloud applications, Future Generation Computer Systems 37 (2014) 14–25.

[15] C. Cacciari, D. Mallmann, C. Zsigri, F. D'Andria, B. Hagemeier, A. Rumpl, W. Ziegler, J. Martrat, SLA-based management of software licenses as web service resources in distributed environments, in: Proceedings of 7th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON'10), Springer, 2010, pp. 78–92.

[16] B. Wheeler, S. Waggener, Above-campus services: shaping the promise of cloud computing for higher education, Educause Review 44 (6) (2009) 52–67.

[17] V. C. Emeakaroha, M. A. S. Netto, R. N. Calheiros, I. Brandic, R. Buyya, C. A. F. D. Rose, Towards autonomic detection of sla violations in cloud infrastructures, Future Generation Computer Systems 28 (7) (2012) 1017–1029.

[18] W. Li, J. Tordsson, E. Elmroth, Virtual machine placement for predictable and time-constrained peak loads, in: Proceedings of 8th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON'11), Vol. 7150 of Lecture Notes in Computer Science, Springer, 2012.

[19] A. Ali-Eldin, J. Tordsson, E. Elmroth, An adaptive hybrid elasticity controller for cloud infrastructures, in: Proceedings of the IEEE Network Operations and Management Symposium (NOMS'12), 2012.

[20] A. S. McGough, M. Forshaw, C. Gerrard, S. Wheater, B. Allen, P. Robinson, Comparison of a cost-effective virtual cloud cluster with an existing campus cluster, Future Generation Computer Systems 41 (2014) 65–78.

[21] M. M. Kashef, J. Altmann, A cost model for hybrid clouds, in: Proceedings of 8th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON'11), Springer, 2012, pp. 46–60.

[22] C. De Alfonso, M. Caballer, F. Alvarruiz, G. Moltó, An economic and energy-aware analysis of the viability of outsourcing cluster computing to a cloud, Future Generation Computer Systems 29 (3) (2013) 704–712.

[23] G. Mateescu, W. Gentzsch, C. J. Ribbens, Hybrid computing-where hpc meets grid and cloud computing, Future Generation Computer Systems 27 (5) (2011) 440–453.

[24] M. Zhang, R. Ranjan, S. Nepal, M. Menzel, A. Haller, A declarative recommender system for cloud infrastructure services selection, in: Proceedings of the 9th International Conference on Economics of Grids, Clouds, Systems, and Services (GECON'2012), Springer, 2012, pp. 102–113.

[25] S. K. Garg, S. Versteeg, R. Buyya, A framework for ranking of cloud computing services, Future Generation Computer Systems 29 (4) (2013) 1012–1023.

[26] C. Bodenstein, M. Hedwig, D. Neumann, Strategic decision support for smart-leasing infrastructure-as-a-service, in: Proceedings of the International Conference on Information Systems (ICIS'11), 2011.

[27] Z. Gong, X. Gu, J. Wilkes, Press: Predictive elastic resource scaling for cloud systems, in: Proceedings of the 6th International Conference on Network and Service Management (CNSM'10), 2010.

[28] D. Gmach, J. Rolia, L. Cherkasova, A. Kemper, Capacity management and demand prediction for next generation data centers, in: Proceedings of the IEEE International Conference on Web Services (ICWS'07), 2007.

[29] A. Ganapathi, Y. Chen, A. Fox, R. H. Katz, D. A. Patterson, Statistics-driven workload modeling for the cloud, in: Proceedings of the 26th International Conference on Data Engineering (ICDE'10), 2010.

[30] A. Chandra, W. Gong, P. J. Shenoy, Dynamic resource allocation for shared data centers using online measurements, in: Proceedings of the 11th International Workshop on Quality of Service (IWQoS'03), 2003.

[31] F. Berman, R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman, S. M. Figueira, J. Hayes, G. Obertelli, J. M. Schopf, G. Shao, S. Smallen, N. T. Spring, A. Su, D. Zagorodnov, Adaptive computing on the grid using apples, IEEE Transactions on Parallel Distributed Systems 14 (4) (2003) 369–382.

[32] F. Berman, R. Wolski, S. Figueira, J. Schopf, G. Shao, Application-level scheduling on distributed heterogeneous networks, in: Proceedings of the 1996 ACM/IEEE Conference on Supercomputing, IEEE, 1996.

[33] M. A. S. Netto, C. Vecchiola, M. Kirley, C. A. Varela, R. Buyya, Use of run time predictions for automatic co-allocation of multi-cluster resources for iterative parallel applications, Journal of Parallel and Distributed Computing 71 (10) (2011) 1388–1399.

[34] L. T. Yang, X. Ma, F. Mueller, Cross-platform performance prediction of parallel applications using partial execution, in: Proceedings of the ACM/IEEE Conference on High Performance Networking and Computing (SC'05), 2005.