EduCloud: PaaS versus IaaS Cloud Usage for an Advanced Computer Science Course

Luis M. Vaquero, Member, IEEE

Abstract—The cloud has become a widely used term in academia and the industry. Education has not remained unaware of this trend, and several educational solutions based on cloud technologies are already in place, especially for software as a service cloud. However, an evaluation of the educational potential of infrastructure and platform clouds has not been explored yet. An evaluation of which type of cloud would be the most beneficial for students to learn, depending on the technical knowledge required for its usage, is missing. Here, the first systematic evaluation of different types of cloud technologies in an advanced course on network overlays with 84 students and four professors is presented. This evaluation tries to answer the question whether cloud technologies (and which specific type of cloud) can be useful in educational scenarios for computer science students by focusing students in the actual tasks at hand. This study demonstrates that platform clouds are valued by both students and professors to achieve the course objectives and that clouds offer a significant improvement over the previous situation in labs where much effort was devoted to setting up the software necessary for course activities. These results most strongly apply to courses in which students interact with resources that are non-self-contained (e.g., network nodes, databases, mechanical equipment, or the cloud itself), but could also apply to other science disciplines that involve programming or performing virtual experiments.

Index Terms—Cloud, course, education, evaluation.

I. INTRODUCTION

HIGHER education is characterized by “the tension between the offered quality and the drive to provide affordable higher education to more and more people” [1]. Information technologies (IT) have an increasing role as supporting elements to improve the quality (access to more resources that can be globally widespread) and to reduce the cost of resource usage, thus easing the sustainability of education.

Different IT technologies have been proposed for educational support in a variety of courses, such as those in advanced networking and distributed systems. Service-oriented architectures (SOAs) [2], [3] and Grid [4], [5] are examples of this. However, they imply a high entry barrier for educators and students who have to learn the new tools. This is a well-known problem related to the usage of IT solutions as education-supporting elements [6]. Also, SOA and Grid solutions, as with many other computing technologies, require certain hardware and software infrastructure to be setup before they can be used for lab assignments. This demands specialized technical skills from both professors and students to set up the experimental environment. These tasks often distract students from the real goals of the course since they are focusing their effort on peripheral tasks more related to system administration than to the course topics.

Cloud computing [7] is a new paradigm for the provision of every network-available resource (X) as a service (XaaS). Although there are many definitions for the cloud, most of them include these four basic ingredients:

- On-demand resource provisioning: The required resources (machines, online libraries, version control systems, data management APIs, etc.) do not need to be online or deployed beforehand.
- Pay-per-use model: Users pay only for the actual usage of resources.
- Quality guarantees: Quality of service is guaranteed by the provider by the usage of service level agreements (SLAs).
- Scalability: The “size” of the service dynamically and automatically adapts to the actual demand.

Assuming that all clouds provide these four features, clouds are also usually categorized in three types, depending on the abstraction level of the provided services.

- Infrastructure-as-a-Service (IaaS) clouds supply virtual hardware resources such as machines, networks, or storage. A well-known example is Amazon EC2.1 Here, users must adapt, deploy, and control all the software stack their applications rely on.
- Platform-as-a-Service (PaaS) clouds provide a container environment for users to run their software components—for instance, an application container, an online database management system shared with several users, other mechanisms for data persistence to be used by online applications deployed over the platform, etc. Examples are Google App Engine2 (GAE) and Microsoft’s Azure.3
- Software-as-a-Service (SaaS) clouds give access to typical applications (e-mail clients, etc.) through the network. As such, they are intended for final users, not for system developers. The problem of dealing with education-supporting technologies is here reduced to selecting and aggregating in an appropriate order a set of prepackaged software services, and so this category is not included in this study.

These two first different levels of clouds are shown in Fig. 1. As can be observed, IaaS clouds offer an abstraction layer preventing users from dealing with low-level hardware configurations (but still dealing with the software and operating system, 1http://aws.amazon.com/ec2. Last viewed December 2010.

Manuscript received September 30, 2010; revised November 10, 2010; accepted November 30, 2010.

The author is with Hewlett-Packard Labs, Bristol BS34 8QZ, U.K., and also with Universidad Rey Juan Carlos, Móstoles 28933, Spain (e-mail: lmvaquero@ieee.org).

Digital Object Identifier 10.1109/TE.2010.2100097
OS, in their virtual machines, VMs). IaaS clouds can be directly exposed to final users (such as Amazon’s EC2), or they can be used by providers of more abstract services to host their services. Fig. 1 illustrates how PaaS clouds can be built upon an underlying IaaS cloud. PaaS clouds, in turn, offer services to help developers throughout the whole life cycle of an application, from design to test and to production. In the examples in Fig. 1, it can be observed how data processing (e.g., data mining) and data storage services for developers are supported by either a database and/or a distributed file system. These latter elements can (since it is not mandatory to have an IaaS cloud to build a PaaS), in turn, be deployed inside one or more VMs on top of the capabilities enabled by the IaaS layer.

In educational environments, both IaaS and PaaS can be considered as supporting technologies for education. From the features described, a several-fold increase in efficiency (improved access to widespread heterogeneous resources with reduced costs [1]) can be expected, helping educators and students to reach the intended educational results faster. In addition, another fact is also attracting educators’ focus to cloud technologies: Several major IaaS vendors, such as Rightscale and Amazon, have recently presented their “education programs” to ease educational access to cloud infrastructures. Other players have also offered mail, collaboration, and more software in the cloud, such as Microsoft’s Live@Edu.

Given this is something of a Cambrian era for a widespread application of the cloud to education, the main goal of this study is to ascertain what type of cloud is more appropriate to reach the desired educational objectives. To accomplish that mission, students in an advanced course in computer science have been manually setting up networks and nodes and coding the required monitoring and routing algorithms for creating an overlay network that they used for assignments involving distributed software applications. This paper investigates whether the use of IaaS or PaaS (as infrastructure for the assignment) increases or decreases the performance of the students and how they perceive the use of IaaS and PaaS. That is, do students using IaaS or PaaS find it to be easier or harder than manually setting up the needed infrastructure to test the probing and routing algorithms (main focus of each assignment), and do they have a clear preference for one technology over the other? The perception of the professors was also taken into account.

The remainder of this paper is structured as follows. Section II presents related work in previous distributed systems and paradigms that tried to improve the efficiency in education. Having precedent technologies pros and cons clear, Section III introduces the educational context, the learning objectives of the course, and the cloud technologies to be used to achieve the objectives. Then, the obtained results are presented in Section IV. This quantitative evaluation is further explained by the inclusion of qualitative information in Section V. Finally, a wrap-up of the obtained results presented in Section VI.

II. RELATED WORK

Among available IT, several distributed technologies have been employed to support education by increasing the range of available resources at a reduced cost.

SOAs advocate for the usage of open and standard-based interfaces to facilitate the reutilization and integration of several educational tools [2], [3], [8], [9]. Examples of service-based educational environments include LearnServe [10]. This system allows pupils to search and select learning contents that are later presented sequentially.

Other service-oriented approaches introduced the usage of grid services for enabling widespread access to computational and hardware resources [4]–[6], [11], [12]. As discussed in [7], grids and clouds are different paradigms whose differences and similarities render the distinction fuzzy. For the sake of brevity, the differences can be summarized by recalling the four main elements offered by a cloud: on-demand provision, SLAs, scalability, and a pay-per-use billing model. The interested reader is referred to [7] for a more detailed comparison. Gimolas was an environmental learning grid-enabled tool consisting of a topological database joined to a visualization tool that let pupils study the effects of several phenomena on the ground [11]. Similarly, COVASE let users’ avatars interact in a virtual environment to learn fluid dynamics and finite state analysis [5]. These were shown in real time, and needed high computational power to reach optimum performance and usability. In terms of hardware resources, Isilab built a lab consisting of remotely controlled electronic equipment such as a signal generator, an oscil-

---


VAQUERO: EduCloud: PaaS VERSUS IaaS CLOUD USAGE FOR ADVANCED COMPUTER SCIENCE COURSE

TABLE I
EDUCATIONAL EVALUATION OF THE AVAILABLE CLOUDS AIMED AT EDUCATIONAL ENVIRONMENTS. NOTE: SaaS IS NOT COVERED IN THIS STUDY, BUT IT IS INCLUDED HERE FOR COMPLETENESS

<table>
<thead>
<tr>
<th>Name/Reference</th>
<th>Cloud Level</th>
<th>Description</th>
<th>Educational Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>StarHPC [14]</td>
<td>IaaS</td>
<td>e-learning infrastructure</td>
<td>None</td>
</tr>
<tr>
<td>[15]</td>
<td>PaaS</td>
<td>deploys a framework for managing resources, sessions in a virtual cloud computing lab</td>
<td>None</td>
</tr>
<tr>
<td>Seattle [17]</td>
<td>PaaS</td>
<td>API for controlling cloud-enabled resources and online libraries for easing concurrent programming</td>
<td>None</td>
</tr>
<tr>
<td>[18]</td>
<td>PaaS</td>
<td>runtime environment with visibility and connecting tasks to a SaaS service</td>
<td>None</td>
</tr>
<tr>
<td>[19]</td>
<td>SaaS</td>
<td>course sharing platform</td>
<td>None</td>
</tr>
<tr>
<td>[20]</td>
<td>SaaS</td>
<td>online services supporting a course on mechanical engineering</td>
<td>Learning effectiveness of blended/flex course redesign compared with traditional online course delivery</td>
</tr>
</tbody>
</table>

There are already some proposals to apply cloud technologies in educational environments; see Table I. Some relevant efforts to use IaaS cloud technologies in education include StarHPC [14]. StarHPC prepackaged a VM image used by students, the scripts used by an administrator, and a virtual image of the Amazon Elastic Computing Cloud (EC2) machine used to build the cluster shared by the class. At the PaaS level, more open and testable experimental platforms are needed in a lab-level with PCs. Tian et al. [15] presented a higher abstraction-level framework for managing PaaS in a virtual cloud computing lab. Yin et al. built a GAE-based course sharing platform [16], but did not perform an evaluation of the usefulness of the cloud for learning purposes. Also, Seattle offered a community-based effort to set up a platform environment [online APIs offering capabilities for managing files, network resources, threads (called timers and locking APIs), and miscellaneous (other online libraries for doing exits, etc.)] to help to deliver a cloud in the educational environment that promoted resource usage [17]. Other authors proposed a layered architecture for an e-Learning cloud [18], focused on delivering SaaS. Similarly, Hu and Zhang presented a case study of a mechanical engineering course redesign where learning effectiveness is compared to its traditional and online-only course delivery [19]. As mentioned in Section I, SaaS aims at offering online services to end-users, so in this regard it is similar to any general online learning tool. As another type of cloud, SaaS clouds still benefit from the SLAs, dynamic provision, automated scaling, and pay-per-use models that characterize the cloud capabilities mentioned in the Introduction, which is an important difference to traditional online learning platforms.

All this interest in clouds is justified by the perceived advantages they bring. However, this increasing number of tools and architectures for IaaS/PaaS supported education has not been accompanied by a real assessment on the effectiveness of these approaches (Table I). More specifically, column 3 of Table I, reveals that an individual and comparative evaluation is still missing for IaaS and PaaS in order to ascertain their real potential in some scenarios.

Of all the cloud-related works, only Hu et al. [19] performed a comparison on the efficiency of online models versus traditional models. This evaluation was not focused on the impact of supporting technologies or the perceived ease of use and acceleration of the learning process. Furthermore, the appropriate level of abstraction (i.e., IaaS or PaaS) that should be delivered to students to enable them to focus on the course topics has not been analyzed by previous works. That is, previous reports do not present an exhaustive evaluation of the advantages of using different type of clouds and the educational outcome obtained (average qualification, perceived ease of usage, time to obtain the desired results, etc.).

This paper presents data on how the utilization of clouds can be useful for achieving not only resource-usage/cost optimization, but also for enhancing students’ results. Furthermore, the most appropriate abstraction level at the cloud stack (IaaS or PaaS) is analyzed.

III. COURSE BACKGROUND

A. Course Description

The course on overlay networks described here. “Infraestructura de Redes” is taught in the fifth year of study in the Computer Engineering degree at Universidad Rey Juan Carlos (URJC), Móstoles, Spain. This degree requires the completion of approximately 35–40 credits (European Credit Transfer Systems, ECTS) per semester during a five-year period. The course introduces overlay networks and peer-to-peer systems and discusses their general properties and applications, drawing its material from all these content areas. The range of topics dealt with is very varied, ranging from the optimal placement of a node in the overlay, how to monitor the servers and networks connecting them, and how to discover, access, and distribute information in large-scale graphs.
This course is closely related to material taught in several computer science and engineering curricula: System Architecture and Organization and Algorithms and Complexity (ACM CS2008 curriculum); in Algorithms, Computer Systems Engineering or Computer Networks (ACM’s CE2004 curriculum); Algorithms and Data Structures (ACM’s IS2002 curriculum); Networking, Platform Technologies, (ACM’s IT2008 curriculum); and Net-centric Computing and Algorithms and complexity in K12’s education curriculum. These are very important content areas in their respective curricula and, as shown below, can be benefited from the usage of cloud technologies. Clouds are relevant to several subjects in these curriculum such as networking, system administration, economics, algorithms, platforms, etc. More specifically, the course covers the following topics:

- introduction to complexity theory;
- structured and unstructured networks, power-law distributions;
- currently deployed peer-to-peer systems and how they work;
- examples of today’s overlay networks and differences with their antecedents;
- distributed Hash tables as a basis for structured peer-to-peer systems;
- overlay networks design considerations and general architecture from internetworking to resource management: monitoring and routing algorithms;
- content delivery networks, their hierarchy, and their performance evaluation;
- performance issues, monitoring performance, networking tricks, legal aspects, and privacy issues;
- content distribution algorithms;
- peer/node discovery algorithms (central versus distributed systems).

This syllabus is accompanied by the realization of several lab assignments and a final practical session whose purpose is for students to connect the concepts they met in the previous labs. This last practical is explained in detail in Section IV.

### B. Evaluation Setup

Eighty-four students carried out two lab experiments using three different setups. The first (IaaS) uses an IaaS cloud for the provision of VMs, the second (PaaS) uses a PaaS cloud to host student software, and the third (control) is similar to previous years’ settings where students were in charge of the whole setup.

The first group consisted of a series of 54 computer science undergraduate students at Telefónica Labs, Madrid, Spain, between May and July 2010, as well as 10 junior programmers (former undergraduate students) with less than two months of working experience. Since most of the undergraduate students had little practical experience, the presence of both graduate students and junior programmers with higher expertise helped to analyze whether the familiarity with programming languages, server configuration, and the like made any difference. This also helped to improve the statistical power of the tests. All students underwent an individual evaluation of their expertise.6

Later, a second group of 20 fifth-year undergraduate computer science students with college-level programming skills at Universidad Rey Juan Carlos joined the study, repeating the same experimental conditions between July and September 2010. The experiments were directed by four adjunct professors working at Telefónica Labs and Universidad Rey Juan Carlos.

The aim of the lab experiment was to build an overlay network where nodes communicate by using conventional HTTP protocols, so that there is a single central register by which all nodes are known (this frees students from having to spend their time designing and implementing node discovery algorithms). Every node has a list of the other nodes and the distance at which they lie. This distance is obtained by aggregating different network measurements (every node has a probe attached to it to get this data; students design the way the nodes and the network are probed). Data are sent through the fully connected network via the less expensive path according to probe measurements (routing algorithms are an essential design decision). To build such an application, three different scenarios are envisioned.

- Designing a simple servlet by relying on traditional programming frameworks such as Eclipse and Apache Tomcat.7 Students had to install the operating system, set up networking, set up certificates with a local certification authority to access lab machines via SSH, and after this, configure Tomcat and start developing a simple servlet serving as central registry for overlay nodes (which are servlets themselves). This is the setting denoted as “control” since it allows traditional scenarios to be compared to cloud-based ones. See Fig. 2(a).
- Deploying of a worldwide spread network of five Amazon EC2 Linux nodes so that students just have to deploy the required VMs (a simple click) when needed, install Tomcat, and start developing, as shown in Fig. 2(b). This is the setting denoted as “IaaS.”
- Finally, employing [Fig. 2(c)] GAE to build the application (students just have to install an Eclipse plugin to interact transparently with GAE and start developing). This is the setting denoted as “PaaS.”

### IV. RESULTS OBTAINED: EVALUATION

#### A. Evaluation Process Design

As with previous studies [6], [20], [21], the evaluation process is divided into three different stages.

1) **Categorization of the concepts under study.**

6Indeed, in this evaluation, three junior programmers got low rates and were classified as elementary students.

TABLE II
INITIAL SCALE ITEMS FOR PERCEIVED USEFULNESS AND EASE OF USE. EVERY ITEM IN THE TABLE WAS RATED FROM 1 TO 4, MEANING DISAGREE, PARTIALLY DISAGREE, PARTIALLY AGREE, AND TOTALLY AGREE, RESPECTIVELY

<table>
<thead>
<tr>
<th>Item #</th>
<th>Usefulness</th>
<th>Ease of Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Better quality of the work</td>
<td>Reduced error proneness</td>
</tr>
<tr>
<td>2</td>
<td>Better control over the work</td>
<td>Little dependence on online documentation (intuitive)</td>
</tr>
<tr>
<td>3</td>
<td>Saves time</td>
<td>Flexible</td>
</tr>
<tr>
<td>4</td>
<td>Critical to achieve the objectives</td>
<td>Ease of remembering</td>
</tr>
<tr>
<td>5</td>
<td>Helps to focus</td>
<td>Ease of learning</td>
</tr>
<tr>
<td>6</td>
<td>Useful</td>
<td>Easy to use</td>
</tr>
</tbody>
</table>

Fig. 4. Results of the data analysis process performed to the systems by students and professors. The results include crossover comparisons for different groups and considering survey answers as in Table II and quantitative data of time to completion, time devoted to each phase, and percent of success.

2) Data collection, in this case including diverse data sources such as pretest surveys in which students were asked their degree of experience and classified into advanced (meaning more experienced) and elementary (less experienced) students. The surveys were designed to cover all the categories identified in the first stage of the evaluation. Every item in these categories was rated from 1 to 4, meaning disagree, partially disagree, partially agree, and totally agree, respectively (Table II). Also, data regarding the order of introduction to IaaS/PaaS, the time devoted to each task, and the level of knowledge acquired were collected. The second data source was discussion groups. An individual meeting was held with students and professors to discuss key elements in the surveys and to deepen in the evaluation these performed.

3) Data analysis. The data gathered were analyzed by combining a quantitative and qualitative approach, combining experimental data, surveys, and individual interviews [21]. Section IV-A1 introduces some of the relevant elements for the surveys and data the analysis. Some results of the quantitative analysis are better understood in conjunction with explanations and reasons given by professors and students. In other words, this qualitative analysis supports and explains the statistical findings expressed before. This combination is shown in Section V.

1) Employed Categories: A first category is related to the previous experience of the students in related tasks. As mentioned, an initial survey is employed for categorizing students into two classes: advanced (more experienced) and elementary (less experienced) (see Fig. 3). The survey analyzed questions such as their previous experience in system management, and it was performed by formulating four simple questions to the students trying to get a hint of their knowledge on Java Servlet programming and hardware configuration (including security, networking, etc.).

Having this initial segmentation, crossover experiments were selected. Six groups were built, each one corresponding to a sequence of two of the three possible configuration settings (control, IaaS, and PaaS; see Section III). Students were randomly assigned to one of those six groups. The goal is to be able to perform fair comparisons between all possible settings to describe the “goodness” of every employed system (as shown in Fig. 3). In this individual comparison, the terms employed for students to rate different approaches are based on the work by Davis [22]. More specifically, this value is analyzed from a student point of view (perceived ease of use and the perceived usefulness, as shown in Table II) and from the lecturer’s perspective (time to complete the task, average rating and percentage of students completing the task in time). An estimation of the time devoted to each stage by students was also included in the survey.

2) Data Analysis: After completing the data collection stage (see Fig. 4), a per-group (per-sequence) evaluation of the results was obtained after students were placed in their first set (control, IaaS or PaaS) in the sequence. Then, every student serves as his/her own control for the comparison by performing an analysis of variance test (ANOVA) between the two settings with which he/she has worked. The effects of the order in the sequence of settings are also studied. Professors are given a final survey on pedagogic questions.

B. Evaluation Results

As mentioned, for the initial survey’s results, students were evenly classified into advanced (42 students who had generally used the required technologies many times) and elementary (42 who had used most of the technologies a few times or never) students in order to find differences in the results due to their background level (see Fig. 5).

Usefulness and ease of use were analyzed by calculating the average rating of every student of all the items in Table II. First, the answers from all the students going first to control versus first to IaaS and versus first to a PaaS were analyzed. Results are shown in Fig. 6. Control-first students (group size \( n = 28 \)) rated this approach’s usefulness and ease of use as 1.77 ± 0.03 and 2.09 ± 0.01, respectively. Students first submitted to an IaaS environment (\( n = 28 \)) rated usefulness as 2.16 ± 0.01 and ease of use as 2.43 ± 0.03. Finally, students under a PaaS environment first (\( n = 28 \)) gave 1.75 ± 0.03 to usefulness and 2.17 ± 0.02 to ease of use. Students rated IaaS better than control or PaaS, although the differences in the initial evaluation of the different scenarios were minor with regard to their usefulness and ease of use. These results remained unaltered in spite of the initial experience of the students.

However, an “apple to apple” comparison called for the consideration of every individual student and his/her perceived variation in usefulness and ease of use when comparing two different situations. Results of such comparisons are shown in Fig. 7. When elementary students were first put in a control environment, they rated the usability of the approach at 1.55 ± 0.05 (\( n = 7 \)). When this same set of students was asked to rate again after producing the same code in an IaaS scenario, this was significantly increased to 2.86 ± 0.04 (\( p < 0.05; n = 7 \)) [see left-hand side of Fig. 7(a)]. Doing the opposite [see Fig. 7(b)] experiment revealed similar features, and students rated the option of IaaS significantly higher than the control in both usefulness and ease of use (see Fig. 7).

Fig. 7(c) and (d) reveals the differences in the perceived usefulness and ease of use by students using control conditions and a PaaS cloud for developing the desired overlay. As can be observed, students consistently rated PaaS higher than the control no matter the order of the sequence. It should be pointed out that students tend to rate the control option lower after having used the PaaS tools for developing their code (as also happened whenever students first used IaaS and were then transferred to a more conventional control scenario). Finally, Fig. 7(e) and (f) directly compares students first exposed to IaaS and then to PaaS, and vice versa. Students found PaaS to be more useful and easier to use than IaaS no matter their initial level of expertise in the tasks at hand. Again, IaaS was rated lower after having used a PaaS environment.

A general observation from Fig. 7 is that, in all sequences, there were no relevant differences between the perceptions of advanced (experienced) and elementary (less experienced) students. Also notable is that 85% of the students using PaaS rated it as very easy to use, learn, and remember (4 out of 4 rating). Similarly, 80% of the students rated this to be most useful (4 out of 4 rating) for saving time and helping to focus on the course topics (algorithms and overlay layout). This unanimity was not achieved for IaaS over control, in spite of the significantly improved student perception.

With these data at hand, the discussion meetings were started. All students were prompted to quantify the “cloud-induced time reduction” depending on the group they were in, with the following results:

1) Control versus IaaS: 15.0 ± 1.2 h vs 11.3 ± 0.8 h (\( n = 28 \));
2) Control versus PaaS: 13.8 ± 1.1 h vs 7.7 ± 0.6 h (\( n = 28 \));
3) IaaS versus PaaS: 10.6 ± 1.1 h vs 6.8 ± 1.4 h (\( n = 28 \)).

A second question addressed in these briefings was the estimated time devoted to the tasks required for completing the process; see Table III. The values obtained do not exactly add up...
to those reported in the list above, but they remain within reasonably similar limits. During the briefings in discussion groups, most students agreed that cloud computing techniques made their lives easier and helped them to focus on the “overlay” problem “rather than dealing with stuff for other subjects.” After the briefings, a test was prepared by the professors to check the knowledge acquired by students regarding overlay probing and algorithms (typical questions included the theoretical design of a content distribution system taking into account node’s location and the most appropriate techniques for probing and routing content across the overlay). The number of passing students (considering students at Universidad Rey Juan Carlos only) increased by 30% as compared to the previous year’s data. These data are not directly comparable since the students in this course are exposed to two different setups (in the previous year, students underwent “control” conditions only).

The four professors were asked to rate the following sentences from 1–4 depending on their degree of agreement (mean ± standard error in the mean; ratings were between 1 and 4 as shown):

1) IaaS helps knowledge acquisition: 1.5 ± 0.14 (n = 4);
2) IaaS eases evaluation: 1.25 ± 0.13 (n = 4);
3) IaaS facilitates students work: 3.25 ± 0.13 (n = 4);
4) IaaS helps students’ focus: 3.0 ± 0 (n = 4);
5) IaaS helps attain objective: 3.25 ± 0.24 (n = 4);
6) PaaS facilitates knowledge acquisition: 1.5 ± 0.14 (n = 4);
7) PaaS helps evaluation: 1.25 ± 0.13 (n = 4);
8) PaaS helps students work: 3.5 ± 0.14 (n = 4);
9) PaaS facilitates students’ focus: 3.75 ± 0.13 (n = 4);
10) PaaS helps target accomplishment: 4.0 ± 0 (n = 4).

As can be observed, both cloud approaches increase the perception by professors that cloud computing helps attain the educational objective and helps students to focus on the problem at hand, although neither help in evaluating students.

Rather, briefings with professors revealed it took them more time to prepare classes for cloud-related lab assignments, ≃ 30% increase in the preparation time. Professors had to learn how to manage and set up some VMs in Amazon (simple Web interface for VM deployment, which also implies some configuration steps). These steps took only a little time for them to learn (the average estimated time to control this environment was ≃ 0.5 ± 0.05 h, n = 4) with little perceived complexity (“the Web interface certainly helped to reduce errors and to set up the environment” or “having VMs deployed in three minutes is a definite help that allows one to forget about low level stuff”). The increase in preparation time is principally derived from the newness of the technologies since appropriate images had to be generated and uploaded to Amazon’s simple storage service (S3) for later use. Amazon EC2 tools had to be included in the images to allow for persistence of the running images (otherwise, all the changes are lost upon reboot of the machine). Learning these APIs and creating and updating the images was deemed as the most time-consuming task for lab assignment preparation. For GAE, two of the professors involved had less background on software programming. Even so, they found no problems in integrating GAE’s Eclipse plugin and using it for implementing the probing and routing algorithm.

V. DISCUSSION

The changes the cloud is already bringing to how students and academics (and administrators) actually work can be observed in a variety of recent software [14]–[17], [19]. These present
tools, frameworks, and software packages are offered as a service in educational scenarios. However, a detailed evaluation of the educational benefits of cloud computing for education was still missing. The evaluation of infrastructure and platform clouds, either alone or as compared to one another, yields a hint of what the most appropriate cloud type could be in an advanced course on overlay networks, in the assignments for which students need to deploy distributed infrastructure and decide/implement the most appropriate probing and routing algorithms.

An initial comparison of students subject to the first scenario in their sequence (either control, IaaS, or PaaS) revealed minor differences between these technologies. This is explained by the fact that students tend to complain about the platform they are using, so they were put in a more debatable scenario and forced to compare two different approaches. Thus, a crossover study was employed with all 84 students, randomly gathered in seven-member groups (sequences). An important point with these sequences of conditions has to do with the selected order since this may affect the outcome. For instance, a user first randomized to a PaaS platform may have a better opinion than one first randomized to a control group. To analyze this bias, a comparison of reverse-order sequences (e.g., results when students first used IaaS and then PaaS versus results when using first PaaS and then IaaS) was performed, showing minimal differences in order. Of course, a “learning” effect is inherently there in the evaluation. This effect has to do with the knowledge that students carry over to the next sequence (e.g., knowledge on probing and routing—see Fig. 2)—will be useful at all stages, while knowledge on hardware configuration will not prove useful for an IaaS or PaaS scenario). This was taken into account when analyzing PaaS to IaaS, PaaS to control, and IaaS to control groups. To minimize this effect, a whole month “washout” period was allowed in which students did not work on related topics, and the algorithms (for appropriately probing, for routing information, for content distribution, etc.) used in each experiment were changed. This minimization is shown by the fact that no differences in the final evaluation were found and the time for developing probing and routing (see Table III) is not significantly different for all three scenarios considered.

The principal observations to be made from this analysis follow:

- Overall, students and professors agreed that cloud techniques helped to maintain students’ focus. Professors indicated their perception that cloud technologies saved a lot of time for developing the course contents. Of course, using PaaS clouds for courses on (for example) operating system administration may soon prove to be a poorly conceived idea since these hide the underlying heterogeneity of the operating system employed and the software tools used. Clouds should not be applied when the cloud could isolate students from the technologies/concepts they are supposed to learn.

- In general, students valued PaaS for its ease of learning, remembering, and use and for the time it saved. More variability was found in the final quality of the work (no differences were found that were dependent on the previous background of the students). Regarding this overall quality, it should be pointed out that in spite of saving a lot of time, cloud technologies did not prompt students to devote the time saved to overlay probing and routing algorithms. These data obtained by student ratings are in full agreement with the lack of improvement in the grades obtained in final tests (e.g., using PaaS did not lead to better marks than did using IaaS). They are also in agreement with the experience of the professors, who did not consider any cloud abstraction level to improve the results obtained or the acquisition of new knowledge by students. Of course, technology “per se” is not enough to help to motivate students further so that they devote more hours to the routing and probing problem and thus increase their final grade. This should be considered as a by-product of the study, not as a main target.

- A significant result of including cloud in this course was the marked increase in the percentage of passing students as compared to the previous year. Both students and professors, interviewed after the tests, agree that cloud technologies helped them to be more focused on the task at hand, resulting in a higher percentage of passing students. Quoting a student, “I felt I could focus on designing the algorithms and testing them in a real environment, rather than either waiting a lifetime on platform tests or performing mere simulations,” while another one mentioned that “I liked GAE (PaaS) the most, since it took out most of the unnecessary underlying complexity, while letting one focus on the algorithmic design.” On the other hand, professors realize that “I can devote more time to explaining mathematical tricks and algorithms, instead of wasting time in setting up the environment,” or “Although I was reluctant at the beginning, the experiment was worth it and students seemed satisfied.” Of course, comparing these data to those of the previous year only cannot be considered as statistically significant, so readers should take this cautiously.

- Another positive issue that arose in the discussion with students after the tests had to do with the ubiquitous access and worldwide infrastructure that can be set up for them, rather than their relying on local resources. This fact has also been pointed out by others as “the key mobile role for education” [1]. There, the unbundling by institutions (separate courses from academic programs and even course contents) and custom use of resources for professors and students to access the content were highly recommended [1].

- An inconvenience highlighted by professors was that the cloud itself did not only fail to help in boosting evaluation, but also it increased the time devoted to lab assignment preparation. The fact that they were doing this for the first time and did not have an in-depth knowledge of the technological environment was pointed out by most of the professors. These factors will diminish as technologies get established in the course curriculum and should not be a problem for future offerings of the course. Students, in turn, complained about how it is hard for them to get used to this cloud philosophy when they are introduced to it in the first lab session. A “penalty effect” was also noticed by which control rates are lower after having used IaaS or PaaS, and IaaS is poorly rated after having used PaaS. This observation was confirmed by some students claiming that “we felt like dealing with unneeded stuff when turned back down to an IaaS/PaaS option.”

Examining curricula above the cloud could potentially benefit subjects such as networking, system administration, algorithms,
and platforms, depending on the required abstraction level and where the learning elements should be located. For instance, if a lecturer wants lab assignments to focus on high-level tasks, such as software development, algorithms, or the like, a PaaS cloud hides the underlying layers’ complexity. On the contrary, if the aim is to train a system administrator, a more integrative approach including all layers (from hardware to software) may be required (this is where the cloud does not come in handy in computer science and engineering).

While it seems ideal for selecting and focusing on the tasks where students should focus their attention, removing unneeded complexity, IaaS or PaaS clouds are more difficult to extrapolate to other subjects beyond computer science and engineering. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located. For instance, if the learning elements should be located.