



Leveraging Cloud Infrastructures for Teaching Advanced Computer Engineering Classes Experiences at the University of Pisa

Antonio Cisternino¹, Pietro Ducange²(✉), Nicola Tonello²,
and Carlo Vallati²

¹ Department of Computer Science,
University of Pisa, Pisa, Italy
a.cisternino@unipi.it

² Department of Information Engineering,
University of Pisa, Pisa, Italy
{p.ducange,n.tonello,c.vallati}@unipi.it

Abstract. In the framework of Problem-Based Learning (PBL), hands-on activities play a crucial role for allowing students to acquire skills and competences. PBL is particularly suitable for teaching advanced courses including topics such as artificial intelligence, cloud computing and big data. In order to carry out practical activities, students may need specific computing and storage infrastructures, that may be not physically available in the laboratories. Thus, virtualization tools and cloud computing are often adopted for building Virtual Lab for teaching purposes.

In this work, we present our recent experiences, at the University of Pisa, in delivering two advanced courses, namely *Cloud Computing* and *Large-Scale and Multi-Structured Databases*. We discuss how we built and exploited a Virtual Lab on top of the private Cloud infrastructure of the University of Pisa as a support for the teaching-learning workflow. Moreover, we present the quantitative feedback received by the students about their experience with the Virtual Lab, showing that the learning objectives that we fixed for our courses have been fully achieved.

Keywords: Innovative Degree Programs · Cloud Computing · Distributed Databases · Virtual Lab · Problem-Based Learning

1 Introduction

Nowadays, universities offer Innovative Degree Programs, especially in technological fields such as artificial intelligence, data science and cybersecurity [8,9].

This work was supported by the Italian Ministry of Education and Research (MIUR), in the framework of the CrossLab project (Departments of Excellence).

The curricula of such university programs include subjects requiring students to be involved in practical teaching activities and develop educational projects (usually in groups). Indeed, from a pedagogic point of view, the most effective way to get students to master IT tools and platforms is a Problem-Based Learning [2] approach based on practical activities and educational projects (preferably in a collaborative learning framework) [1]. The evolution of IT hardware with its requirements make it unsuitable for installation in classroom with instructor and students due to space, power and cooling requirements. This is the case of courses that handle subjects such as cloud and distributed computing, where an infrastructure with a large number of servers is required [10,11] to allow students to experiment with a plausible setup. These kind of problems have been reported into two recently published contributions [6,11]. The authors discussed their experiences on the transition from the use of a physical infrastructure to the exploitation of virtual resources for teaching computer networking and distributed computing courses.

Moreover, the infrastructure should be always available to students so that they can experiment through practical homework and educational projects. Finally, the ability to remotely access the infrastructure for practicing allows distance learning to support students unable to attend laboratory classes. This is what happened during the current COVID-19 pandemic emergency that required to transition from in-presence teaching to distance learning within few days [14].

In this work, we discuss our recent experience in teaching Cloud Computing and Distributed Databases courses, at the Master of Science Program in Artificial Intelligence and Data Engineering, offered for the first time in the 2019–2020 Academic Year, by the University of Pisa¹. In order to allow students to develop hands-on activities and the education projects that we assigned to them, we have implemented a Cloud-based Virtual Lab. Students were allowed to access to a set of virtual machines (VMs), to install the required tools and frameworks, and to actually develop the assigned activities, such as client-server applications interacting with distributed databases and distributed programming exercises.

The paper is organized as follows. Section 2 discuss some recent contribution on adopting virtualization techniques and cloud computing services in the context of education. Section 3 illustrates the architecture of our Cloud-based Virtual Lab. Afterward, Sect. 4 discusses some examples of teaching workflows carried out exploiting our Virtual Lab. Section 5 reports some discussions on feedback provided by students on the Virtual Lab adoption and usage. Section 6 draws some final remarks on our teaching experiences.

2 Related Work

In the last years, a special attention has been payed in exploiting virtualization techniques and cloud-computing services as a support for teaching both traditional and innovative courses [12]. These courses include subjects such as computer networking [6], operating systems [3], cybersecurity [8,15], parallel and

¹ <https://computer.ing.unipi.it/aide-lm>.

distributed computing [11], big data [4] and cloud computing [5,10]. Recently, in the work published in [7], the author argues on four dimensions on which virtualization techniques and cloud computing services should be integrated into the educational framework. Specifically, he suggests that these techniques and services should be a priority for the educational institutions in terms of: i) creating a computing infrastructure based on virtualization and cloud services, ii) including virtualization and cloud computing as a teaching subjects, iii) exploiting virtualization and cloud services as tools for teaching and iv) integrating the previous concepts as a whole point of strength of the institution itself.

As regards teaching cloud computing, authors of [10], discuss their experience at the University of Zilina, in Slovakia, where they exploit a private cloud-based lab, based on OpenStack, for allowing students to carry out hands-on activities. The activities include: the installation of the virtualization platform, building VMs, creating a virtual networks with several machines and writing orchestration scripts. In a recent work published in [11], authors argue on their approach for teaching parallel and distributed computing at the Clemson University, in South Carolina, US. Specifically, authors share their experience in leveraging an US publicly available computing resource, namely CloudLab², as a platform to develop and host teaching materials, to allow students to carry out their practical activities and to support instructors to monitor the teaching-learning workflow. Students were first asked to take familiarity with the basics of Linux Operating Systems and of computer networking, in the virtualized framework offered by CloudLab. Then, by adopting a Problem-Based learning approach, group of students had the opportunity of building different topology of clusters on which they experimented the implementation of parallel and distributed applications. For the implementation of their application, students adopted the tools of the Hadoop environment, including Hadoop Distributed File System.

In 2019, a contribution on adopting virtualization techniques for teaching, learning, and assessing students' performance, in computer networking subjects, was presented in [6]. The author discusses how his "Networking and Telecommunications Management" course, at the University of the Pacific, in Stockton, California, has been redesigned in order to improve the low quality of the teaching-learning workflow. The author states that he achieved unsatisfactory results of teaching mainly due to the adoption of physical machines and network equipment for students hands-on activities. The new version of the course includes traditional lectures and discussion sessions that focus on theoretical and fundamental topics. Moreover, virtualization techniques are mainly used for providing students with hands-on experience in computer network management. For the practical activities, students are required to install VirtualBox³, on the computers in a lab or on their own laptops, and then to create VMs, to install different operating systems, to organize computer networks and so on. Thanks to virtualization, many advantages are offered to the students, such as portability of their work, low cost, and freedom of experimentation.

² www.cloudlab.us.

³ <https://www.virtualbox.org/>.

As regards teaching cybersecurity using cloud-based services, a recent contribution has been published in [8]. Authors report the results of a feasibility study that they carried out by evaluating two Cloud Service Providers (CSP), namely Google Cloud and Microsoft Azure. During summer of 2018 a group of students were involved in laboratory sessions and some practical activities were assigned to them. These activities included a number of tasks such as to connect to the cloud provider to initiate sessions, to create VMs and virtual networks and to simulate security attacks. After the lab sessions, students were interviewed face to face to evaluate the quality of their experience, in terms of usability of the services offered by the two CSPs.

Recently, a group of researchers of the eCampus University, an Italian distance learning institution, designed a virtual-lab, also based on cloud computing, for carrying out the practical activities of a sport and exercise science university program [13].

Finally, interesting suggestions can be found in [16], in which authors carry out a deep analysis of different tools for exploiting virtualization and cloud computing in higher education.

3 Cloud-Based Virtual Lab Architecture

We implemented our Virtual Lab using the private cloud infrastructure of the University of Pisa. The infrastructure runs on over 70 servers with around 3100 physical cores managed by Microsoft System Center Virtual Machine Manager⁴ (SCVMM); servers run either VMware ESXi or Microsoft Hyper-V⁵ to run the virtual workloads of the University. A private cloud has been allocated to support the Virtual Lab using Hyper-V servers. Using a scriptable PowerShell it is possible to manipulate VM creation, configuration, and deletion using SCVMM.

A set of tools have been implemented to automate the creation of the VMs through SCVMM. One script, in particular, is responsible for creating all the VMs for one specific class, in order to create and assign one VM for each enrolled student. The script behaves as follows: starting from the list of the students enrolled in the class, it automatically creates and configures one VM for each student; after the creation of VMs, the script automatically notify the student via emails including all the details to connect to the assigned VM. In order to keep track of the association between a VM and a student, the script automatically populate an Excel file with the IP address of each VM with the corresponding student. In this way the instructor can keep track of the assignment and connect to each VM when need to support and troubleshooting.

Each VM has a virtual network adapter connected to an overlay virtual network isolated from the internet. A virtual gateway, implemented using *pfSense*, allows the VMs to connect to Internet. A VM running OpenVPN⁶ allows students to securely connect to the assigned VMs through VPN.

⁴ <https://docs.microsoft.com/en-us/system-center/vmm/overview>.

⁵ <https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/>.

⁶ <https://openvpn.net/>.

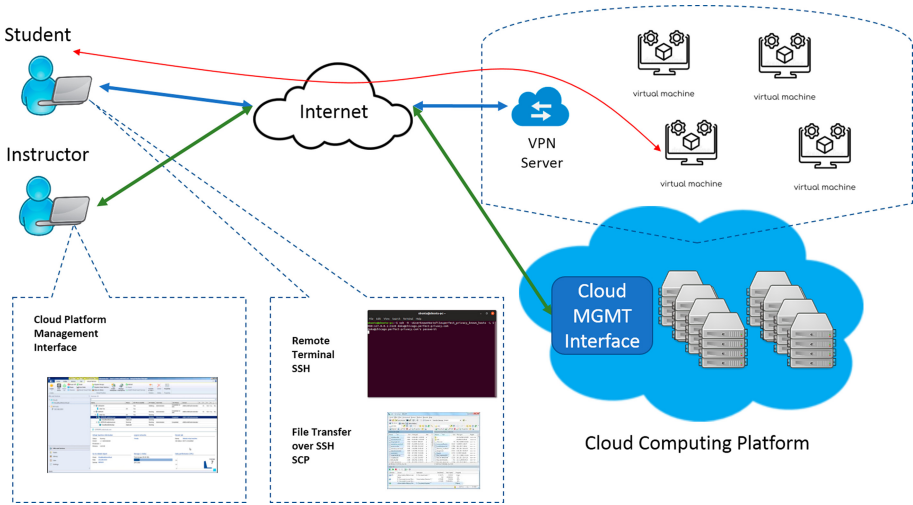


Fig. 1. Cloud-based Virtual Lab Architecture

The overall architecture of the cloud-based Virtual Lab is presented in Fig. 1. Instructors can access the management interface exposed by the cloud platform to manage the VMs for the students, also in an automated fashion by exploiting the tools we implemented. Instructors can also exploit additional functionalities offered by the cloud platform. An example is the console functionality that allows to connect to each VM. This functionality is useful to connect to a specific VM to check the progress of the students in carrying out an assignment or to troubleshoot a specific issue. On the other hand, students interact with the VMs allocated through remote connection, e.g. a remote terminal like SSH or a file transfer. Through them they can perform any kind of assignment/exercise on the VM, from compiling and executing a program to install and configure a specific software. Exploiting the fact that VMs are all connected to the same virtual overlay network, students can also carry out group assignment in which the VMs assigned to each student can connect with other VMs assigned to other students. This is specifically useful to design assignments involving distributed software, i.e. systems made of different component running on different machines interacting through an internet/intranet connection.

4 Teaching Activities on the Virtual Lab

Once all the VMs have been allocated in the Virtual Lab, we divided students in groups of 4. Each group was required to pool together their VMs to build up a small but fully functional cluster of computers using the full management rights granted on the cluster VMs. This group organization allowed us (1) to illustrate the state-of-the art tools and frameworks in distributed environments, and (2)

Table 1. Tools adopted in the teaching activities.

Name	Description
GitHub ^a	Web platform offering distributed version control and source code management using Git
Jupyter ^b	Web-based computational environment supporting interactive data science and scientific computing
Docker ^c	Platform-as-a-service software using use OS-level virtualization to deliver software in packages (containers)
Kubernetes ^d	Container-orchestration system for automating computer application deployment, scaling, and management
MongoDB ^e	Cross-platform document-oriented database management systems supporting dataset replication and sharding
Neo4J ^f	Cross-platform graph-oriented database management systems supporting transactions and advanced queries in graphs
HDFS ^g	Distributed, scalable, and portable file system for the Hadoop framework
YARN ^h	Resource management handling distributed applications for the Hadoop framework
Hadoop ⁱ	Software framework for distributed processing of big data using the MapReduce programming model
Spark ^j	Distributed general-purpose cluster-computing framework using the MapReduce programming model

^a<https://github.com/>.

^b<https://jupyter.org/>.

^c<https://www.docker.com/>.

^d<https://kubernetes.io/>.

^e<https://www.mongodb.com/>.

^f<https://neo4j.com/>.

^g<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>.

^h<https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>.

ⁱ<https://hadoop.apache.org/>.

^j<https://spark.apache.org/>.

to design practical activities involving the installation, setup, configuration and use of these tools and frameworks on the virtual cluster.

We organized the practical activities into two main categories, namely *tools setup* and *distributed programming* activities. The learning objectives of the tools setup activities are to allow students to setup and to correctly configure complex software stacks in distributed environments. In addition, we enabled students to experience the tools on distributed runtime support environments. The tools adopted in our courses are reported in Table 1.

Manual setup and configuration allow students to face the real system administration problems and challenges that these tools present, having to understand the different abstractions in complex configurations each one introduces. Once the learning objectives are achieved, it is possible to leverage one of the many management software to automatically configure these systems being aware about the performed operations. This approach is two-fold: firstly, the students will manually configure the different tools to understand their deployment and configuration potentialities, then the automatic configuration will help them to reproduce with almost no effort the settings required to perform the following distributed programming activities.

The main learning goal of the distributed programming activities is to let students learn how to design and develop programs suitable to interact with or run onto state-of-the-art platforms for cloud and distributed computing and storage.

The practical activities requires a careful preparation setup. Besides the preparation of lecture notes and slides describing the architecture, implementation, usage and application programming interfaces (APIs) of the selected tools, the instructor must devote time to setup, test and debug the steps to perform the activities on a specific Virtual Lab. After the preparation steps, the administration of these practical activities requires a mix of face-to-face teaching, using slides and electronic notes, with supervised and unsupervised hands-on sessions.

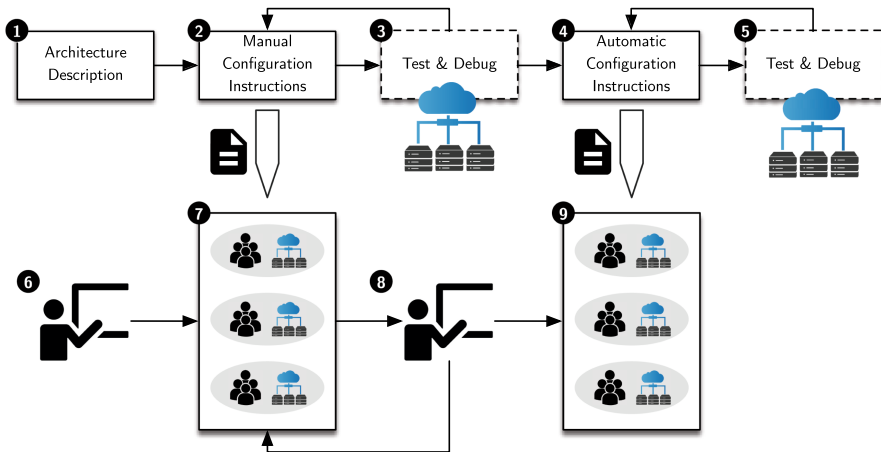


Fig. 2. The workflow of the tools setup activities.

The workflow of the tools setup activities is illustrated in Fig. 2, and the steps are described below:

1. The instructor prepares the slides and/or lecture notes describing the architecture and implementation of the selected tool.

2. The instructor prepares the manual tool configuration instructions to be carried out by the student groups. These instructions must provide a step-wise approach to the setup and configuration of the tool, illustrating the different options and specific details of each tool's underlying infrastructure; the instructions are uploaded on `GitHub`.
3. The instructor tests and debugs the prepared instructions on the dedicated Virtual Lab, refining and correcting the configuration instructions prepared in the previous step.
4. The instructor prepares the automatic tool configuration instructions to configure the selected tool with minimal user interaction. Once the students have carried out the manual configuration setup, it is important to provide automatic configuration procedures to avoid spending additional time to configure from scratch the tool in case of faults, reboots or migrations of the Virtual Lab underlying resources. The instructions are uploaded on `GitHub`.
5. The instructor tests and debugs the prepared instruction on the dedicated Virtual Lab, refining and correcting the configuration instruction prepared in the previous step.
6. The instructor presents the architecture and the implementation design of a selected tools. Students download the instructions of the practical activities and the instructors discusses with the electronic hands-on notes the main activities to be carried out during the unsupervised practical sessions.
7. In an unsupervised way, the student groups focus on reproducing the instructions and commands specified in the electronic notes, at their own pace.
8. After a given amount of time, the instructor supervises the groups and provides feedback and help on the main problems the students could have encountered during the unsupervised practical sessions.
9. At the end, the students are allowed to download the automatic configuration instructions from `GitHub` to avoid the manual setup in case of persistent errors or availability issues of the Cloud platform.

Since the setup, configuration and administration of software tools often require commands to be issues at the command line, it is important to provide students with clear, concise and easy “cut-and-paste” notes to speed up the most difficult configuration commands and to avoid errors.

The workflow of the distributed programming activities is summarized in Fig. 3, and the steps are described below:

1. The instructor prepares the slides and/or lecture notes describing the constructs and APIs of a selected programming environment.
2. The instructor prepares the description of the programming exercises to be carried out by the student groups during the unsupervised practical session, uploaded on `GitHub`.
3. The instructor implements the solutions of the proposed exercises, and uploads them on `GitHub`.
4. The instructor tests and debugs the solutions on the dedicated Virtual Lab, refining and correcting them; the solutions are uploaded on `GitHub`.

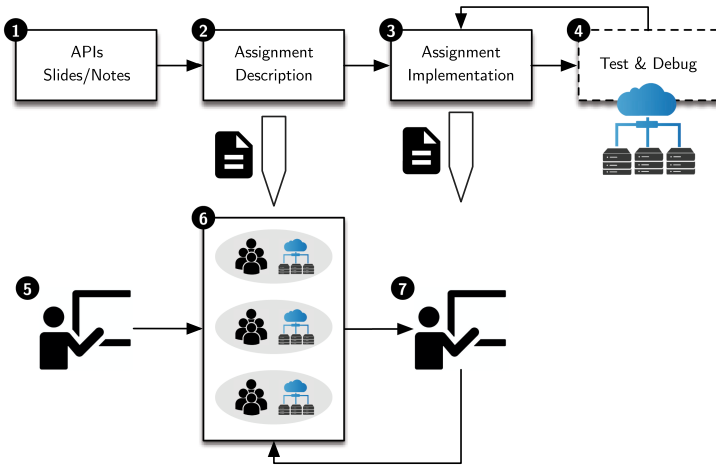


Fig. 3. The workflow of the distributed programming activities.

5. The instructor presents the constructs and APIs of the selected programming environment. Then, he/she provides and discusses the exercises to be carried out during the unsupervised practical sessions.
6. In an unsupervised way, the student groups focus on solving the programming assignments.
7. After a given amount of time, the instructor supervises the groups and provides feedback and help the on main problems the students could have encountered during the unsupervised practical sessions. The groups are now allowed to download and test a proposed solution to the programming exercises hosted on **GitHub**.

It is worth noticing that, in both workflows discussed above, students are not enabled to view the automatic configurations and the proposed solutions before finishing the assignments.

Since groups and students in the same groups have different learning paces, it is important that the instructors provide and discuss the exercises by providing a priori the solutions of the programming assignments. In doing so, groups and/or students within groups can perform during the practical session different activities, depending on their own learning pace, such as discussing and analysing the source code of the provided solution, compiling, testing and adapting the provided solution, and/or developing their own solution and comparing the results with the given one.

Once the student groups successfully master different tools and programming environment through the provided exercises, they are requested to implement a final project. As regards the Cloud Computing course, the project focuses on developing a specific distributed algorithm on the top of MapReduce, and providing a demo of its implementation on the group’s Virtual Lab; in the last edition of the course, students have been requested to implement a version of

the kNN algorithm, namely a simple supervised machine learning algorithm for classification task, in MapReduce, on both the Hadoop and Spark platforms. As regards the Large-Scale and Multi-Structured Databases, students have been requested to design and implement a Java application which interacts with both Document-Based and Graph-Based databases, taking appropriately into consideration data replica, sharding and consistency. Students were required to deploy and run the application on a virtual cluster built on the Virtual Lab.

5 Discussions

The Virtual Lab that we briefly described in this work was adopted for carrying out the teaching activities of two courses, namely *Large-Scale and Multi-Structured Databases* and *Cloud Computing*, delivered by the University of Pisa. The first course was delivered in presence mode, from September to December 2019. As regards the second course, due to the emergency of the COVID-19 pandemic, most of the lectures have been held in on-line mode. Around 100 students were enrolled to both courses and 25 working groups were created. The practical classes were attended by 40–70 students. As for the resources occupied by the VMs, we have allocated about 2TB of RAM, 200 Computing Cores and 9TB of hard disks. The ability to allocate resources on the University on private cloud allowed us to allocate enough resources that in a public cloud setup would have been impractical and expensive, and it has been a key to the Virtual Lab delivery.

Overall, students appreciated the acquisition of skills using the Virtual Lab and the instructors collected a number of positive feedbacks during classes. However, some problems on the cloud infrastructure occurred especially during the March–May 2020 period and students experienced some difficulties for connecting to their VMs. These problems were probably due to the network congestion related with the COVID-19 emergency. With this first edition of the course we validated that it is practical to let students experience setup and configuration of complex and distributed software platforms, thus we intend to continue developing and improving this format in the future editions of the courses.

In order to collect a detailed feedback and highlight issues and drawbacks with our approach, a feedback form was provided to the students on July 2020, at the end of both the courses. The main goals of the questionnaire are summarized in the following:

1. To assess the overall experience from the point of view of the students;
2. To measure the overall usability of the platform and of the tools adopted;
3. To measure the extent of the connectivity issues experienced during the period March–May 2020.

In order to assess the overall experience, a set of three questions were included in the questionnaire for students. The first two questions focused on evaluating the overall experience of using the platform in two different phases of the labs, namely class exercises and final project, respectively. The third question focused

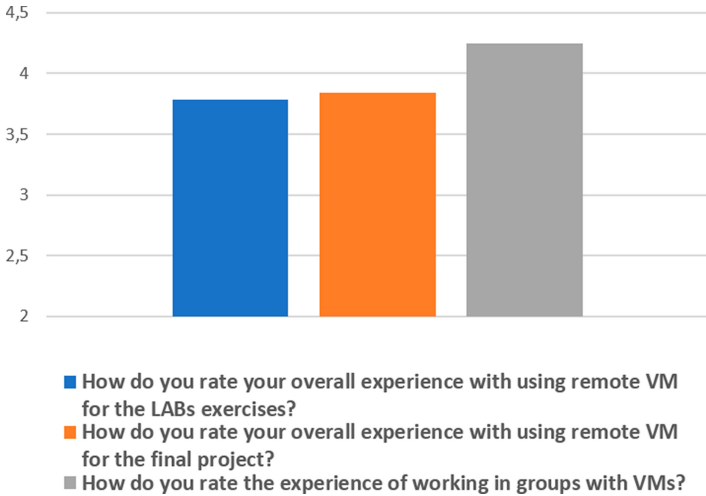


Fig. 4. Overall experience assessment.

on assessing the experience of working in groups using the cloud infrastructure. For each question the student was asked to rate the experience between 1 and 5, with 5 corresponding to ‘very good’ and 1 corresponding to ‘very bad’.

The detailed questions and the average results, over the 42 forms that we got back from students (83.3% Male, 16.7% Female, average age equal to 24.5), obtained for each question are reported in Fig. 4. As can be seen, the overall experience with both the lab exercises and the final projects is very positive. The experience with working in groups using the cloud infrastructure got the highest average score, demonstrating that the platform is well suited for distance group collaboration.

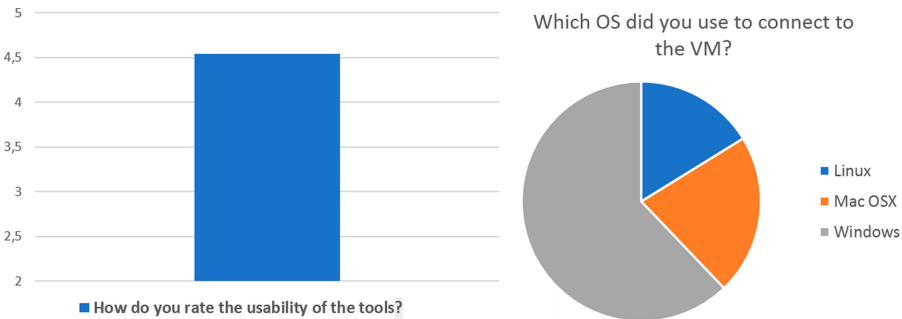


Fig. 5. Tools usability.

In order to assess the usability of the tools, students were asked to report the Operating System (OS) adopted to connect to the cloud platform and rate

the overall usability of the tools, from 1 (very bad) to 5 (very good). Figure 5 reports the collected results. As can be seen, the average usability perceived by the students is very good, slightly above 4.5. In addition to this, our results showed that different OSs were used by the students, although a large portion of students used Windows. The good usability of the tools were confirmed in all the OSs used by the users, confirming that the selection of the tools for the different OSs were adequate.

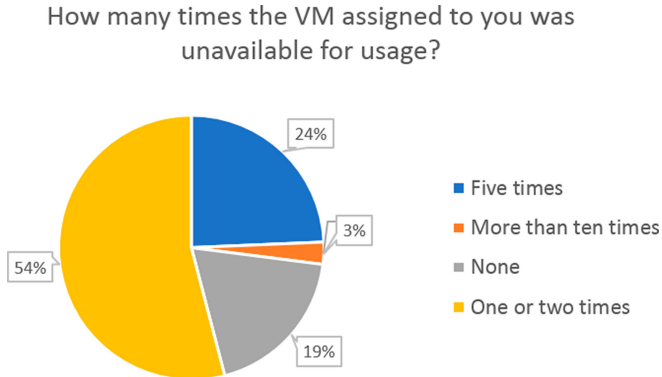


Fig. 6. Platform unavailability.

In order to measure the full extent of the problems on the cloud infrastructure occurred during the March–May 2020 period, we asked the students to quantify for how many times the VMs assigned to them were unavailable. The results are reported in Fig. 6. As can be seen, the majority (almost 3/4) of the students experience VM unavailability only one or two times or no episodes at all. Only a minority experienced more than five unavailability episodes. Although such episodes are restricted to a small minority, the fact that some students experienced more than ten unavailability episodes highlight the need to improve the reliability of the infrastructure.

In order to assess the reliability of the platform when the service is available, i.e. when VMs are reachable, we also asked the students to report the average duration (in hours) of their working sessions. The distribution of the average duration of a working session is reported in Fig. 7. As can be seen, students reported an average duration of working sessions between 1 h and 6 h. This shows that when VMs are available they offer a reliable service that enables students to work for hours on their assignments and final projects.

In order to evaluate the validity of Teaching–Learning workflow of the two courses that exploited the Virtual Lab, handled by the instructors following a Problem-Based Learning approach, in the following we discuss the results achieved by the students at the exam. As regards Large-Scale and Multi-Structured Databases course, after the first exam period (January–February

2020), 15 groups concluded their projects and a total of 55 students passed the exam with average mark higher than 28 points (30 points is the highest mark). Then, after the second exam period (June–July 2020), 14 more groups concluded their projects and the number of students that passed the exam reached a number equal to 82, with average mark higher than 28.5 points. Finally, as regards Cloud Computing course, after the first exam period (June–July 2020), 16 groups concluded their projects and a total of 52 students passed the exam with average mark higher than 28.5 points.

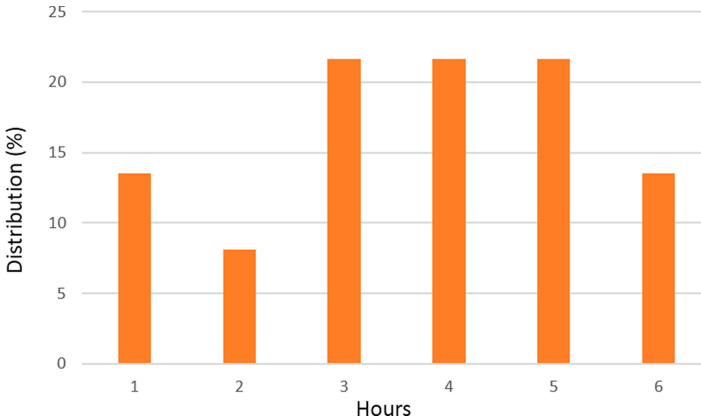


Fig. 7. Session duration.

6 Conclusions

In this work, we have discussed and shared our experience in delivering two advanced courses, in the framework of an innovative Master Degree in Artificial Intelligence and Data Engineering, offered by the University of Pisa. Specifically, their subjects are Cloud Computing and Distributed Databases. The two courses have been approached by the instructors following a Problem-Based learning approach. Indeed, students have been required to carry out hands-on activities for acquiring skills for designing, implementing and experimenting applications in a distributed fashion. Moreover, students have been organized in small groups for developing a final project as part of the exam.

We have mainly discussed our experience in building and exploiting, as a support of the teaching-learning workflow, a Virtual Lab, that has been built on the private cloud infrastructure of the University of Pisa. The Virtual Lab has been used by the instructors for delivering the practical class and by the students for carrying out assigned activities and for developing the final projects. Indeed, thanks to the Virtual Lab, both instructors and students were able to build and use clusters of virtual machines on which distributed application and storage systems were installed and run.

We have shown the results of a campaign that collected student opinions about their experience with the Virtual Lab. To this aim, we submitted a simple questionnaire to each student, adopting an online feedback form. Results have shown that, even though connectivity issues were experimented, students appreciated both the Virtual Lab and the teaching approach adopted by the instructors. Moreover, by analyzing the records of winter and summer exam periods, more than 80% of the students passed the Large-Scale and Multi-Structured Databases exam. As regards the Cloud Computing course, more than 50% of students passed the exam, but we have to consider that student still have the winter period for taking the examination. Finally, the average mark achieved by the students is higher than 28.5 over 30.

By analyzing the students feedback and the results achieved by the students at the exam, from the instructors point of view, we can remark that the learning objectives that we fixed for our courses have been fully achieved. A great effort from the technical point of view has been made for the development and the maintenance of the Virtual Lab. However, there are still some problems on which we are working with the technicians of the University of Pisa, in order to have a more stable and efficient virtualization system. Moreover, the Virtual Lab described in this work will be tested in the next academic year also by instructors of other innovative courses. A more in-deep analysis of the validity of our methodological approach, supported by the novel technological infrastructure, will be carried out, in order to overcome the limitations of the current analysis. These limitations are strictly related with the reduced number of students and courses involved in the usage of the proposed virtual-lab. Moreover, a control group was missing in our analysis because both courses involved in the experimentation were at their first edition.

References

1. Admiraal, W., et al.: Students as future workers: cross-border multidisciplinary learning labs in higher education. *Int. J. Technol. Educ. Sci.* **3**(2), 85–94 (2019)
2. Chis, A.E., Moldovan, A.N., Murphy, L., Pathak, P., Muntean, C.H.: Investigating flipped classroom and problem-based learning in a programming module for computing conversion course. *Educ. Technol. Soc.* **21**, 232–247 (2018)
3. Dobrilovic, D., Stojanov, Z.: Using virtualization software in operating systems course. In: 2006 International Conference on Information Technology: Research and Education, pp. 222–226. IEEE (2006)
4. Eckroth, J.: Teaching big data with a virtual cluster. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education, pp. 175–180 (2016)
5. Gillam, L., Li, B., O’Loughlin, J.: Teaching clouds: lessons taught and lessons learnt. In: Cloud Computing for Teaching and Learning: Strategies for Design and Implementation, pp. 82–94. IGI Global (2012)
6. Huang, A.: Teaching, learning, and assessment with virtualization technology. *J. Educ. Technol. Syst.* **47**(4), 523–538 (2019)
7. Klement, M.: Models of integration of virtualization in education: virtualization technology and possibilities of its use in education. *Comput. Educ.* **105**, 31–43 (2017)

8. Kolhar, M., Alameen, A.: Before teaching cybersecurity in the university: usability analysis of the cloud platform with learner's perspective. *TEM J.* **8**(3), 733 (2019)
9. Li, Y., Wang, X., Xin, D.: An inquiry into AI university curriculum and market demand: facts, fits, and future trends. In: *Proceedings of the 2019 on Computers and People Research Conference*, pp. 139–142 (2019)
10. Moravčík, M., Segeč, P., Uramová, J., Kontšek, M.: Teaching cloud computing in cloud computing. In: *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 1–6. IEEE (2017)
11. Ngo, L.B., Denton, J.: Using CloudLab as a scalable platform for teaching cluster computing. *J. Comput. Sci.* **10**(1), 100–106 (2019)
12. O'Donnell, E., O'Donnell, L.: A review of virtualisation and cloud computing for higher education. In: *Encyclopedia of Information Science and Technology*, 3rd edn., pp. 7487–7496. IGI Global (2015)
13. Picerno, P., Pecori, R., Raviolo, P., Ducange, P.: Smartphones and exergame controllers as BYOD solutions for the e-tivities of an online sport and exercise sciences university program. In: Burgos, D., et al. (eds.) *HELMeTO 2019. CCIS*, vol. 1091, pp. 217–227. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31284-8_17
14. Rapanta, C., Botturi, L., Goodyear, P., Guàrdia, L., Koole, M.: Online university teaching during and after the Covid-19 crisis: refocusing teacher presence and learning activity. *Postdigit. Sci. Educ.* **2**, 923–945 (2020). <https://doi.org/10.1007/s42438-020-00155-y>
15. Salah, K., Hammoud, M., Zeadally, S.: Teaching cybersecurity using the cloud. *IEEE Trans. Learn. Technol.* **8**(4), 383–392 (2015)
16. Segeč, P., Moravčík, M., Kontšek, M., Papán, J., Uramová, J., Yeremenko, O.: Network virtualization tools-analysis and application in higher education. In: *2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 699–708. IEEE (2019)