Teaching Cloud Computing: Motivations, Challenges and Tools

Please, cite this paper as:

Cosimo Anglano, Massimo Canonico, Marco Guazzone

"Teaching Cloud Computing: Motivations, Challenges and Tools,"

Proc. of the 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), New Orleans, LA, USA, May 18-22, 2020.

Accepted for publication.

Teaching Cloud Computing: Motivations, Challenges and Tools

Cosimo Anglano, Massimo Canonico and Marco Guazzone
{cosimo.anglano,massimo.canonico,marco.guazzone}@uniupo.it
Computer Science Institute, DiSIT, University of Piemonte Orientale, Italy

***Abstract*—Teaching Cloud Computing is becoming crucial since this recent computing paradigm is used in many fields and it is changing the way we use the applications and the technology. As a matter of the fact, most of the applications that we use everyday through the web are based on cloud services. Unfortunately, the difficulty to set up a real testbed for students and, at the same time, the lack of an easy, open and collaborative educational material freely available make teaching Cloud Computing a hard task. In this paper we discuss the state of the art concerning teaching Cloud Computing and we propose education materials and tools that make Cloud Computing easy to use even for students/educators without any computer science skills.**

***Index Terms*—Cloud Computing, Educational resources, Cloud tools**

I. Introduction

The NIST defines the *Cloud Computing* as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1]. In the last years, the adoption of this model has grown very fast and it is everywhere: any big IT company has implemented and proposed its cloud platform (i.e., Amazon with *Amazon Web Services* (AWS) [2], Google with *Google Cloud Platform* (GCP) [3], Microsoft with *Azure* [4] and *IBM Cloud* [5], just to name a few) and the most popular applications, which involve million of users, run over a cloud infrastructure (i.e., Facebook, Twitter, Salesforce.com, Netflix, just to name a few). Since Cloud Computing has become so important, it is necessary to teach this recent computational platform to all computer science students and not only. As a matter of the fact, the services provided by the Cloud Computing platform mentioned before are useful in many fields. For example, a biological scientist may need the *Genomic Analyses Service* provided by AWS [6] or the *Databiology's platform* on the IBM Cloud [7].

Due to the variety and the vastness of the potential audience of people interested in learning how Cloud Computing works, it is necessary to provide a course with different paths based on users skills (from computer science students to users without any knowledge about informatics) and focused on students/educators' fields of interest (i.e., biology, math, chemistry and so on). Of course, in order to create this multidisciplinary course, a team of educators specialized for these different paths is needed and a common platform where to share/add/delete/modify the contributions should be easily available form anywhere, through a Cloud Computing service. In 2011, within the *FutureGrid* project [8] (a distributed system which made possible for researchers to tackle complex research challenges by providing them free access to cloud platforms such as *Eucaliptus* [9] and *Nimbus* [10]), we shared our educational materials with the FutureGrid community and we created a specific web page to collect all the contributions from the various educators and researchers (the page is still available in [11] even if the FutureGrid project was dismissed in 2014). At that time, the cloud platforms were not ready for the big audience and most of them were designed for computer science expert users only. Now the most famous cloud platforms provide user-friendly web interface and many services that the user can access by using its browser, for example. This service model, called *Software-as-a-Service* (SaaS), has become a common delivery model for many business applications, including office software, messaging software, CAD software, online games, collaboration software, just to name a few. We believe that now it is the time to re-think about how we teach Cloud Computing in our universities by implementing two actions: (i) share our educational materials in a single place by exploiting all the collaborative tools available in order to easily improve this material constantly, and (ii) provide tools to easily access to different cloud platforms in order to avoid the vendor lock-in (in other words, the cloud users must be able to use concurrently different cloud platforms depending on their needs) and these tools must be easy to use for any kind of user, also the one without any computer science skill.

In this paper, we propose a new way of teaching Cloud Computing that takes into account the interdisciplinary nature both of the users involved and of the services proposed in order to obtain a common knowledge platform where any educators/students can find/add educational materials and tools to fully understand the strengths and weaknesses of the Cloud Computing. The rest of the paper is organized as follows: in Section II we present the state of the art concerning teaching Cloud Computing, while in Section III we discuss the challenge to face in order to provide a high quality educational materials. In Section IV we propose our "Cloud Computing class" by discussing in detail the contents of each lesson and, finally, we conclude the paper and we anticipate the next steps already ongoing (Section V).

II. Related Work

Several educational materials has been proposed during the past years concerning Cloud Computing. In this study, we focus our attention on the free resources avoiding the commercial ones such as *Coursera* [12] where more than 10 courses are provided but related to only one platform, in particular the Google Cloud Platform.

The *Open Source Data Science Curriculum* [13] (OSDSC) proposes videos, courses and books for learning how to manage data science. The approach of this educational resource is to exploit free platforms such as *Github* and *Stack Overflow* in order to provide free reports, source code and hands-out. The problem with OSDSC is related to the freshness of the material proposed: the last commit in its repository was done more than 6 years ago, when Cloud Computing was really different with respect to what is now.

Another important educational resource is *Qwiklabs* [14] which claims in its webpage "Become a cloud expert with hands-on training." In its webpage, the proposed educational materials are only focused on two cloud platforms (i.e., Google Cloud Platform and Amazon Web Services) and only few hands-out are free (for some course you have to buy credits to see the whole materials).

Google Cloud Codelabs [15] is a website focused only on Google Cloud Platform which proposes labs that cover a wide range of topics such as Google Cloud Basics, Compute, Data, Mobile, Monitoring, Machine Learning and Networking. The labs proposed can be filtered by topic, duration and date of publication, but a filter related to the user competence levels (i.e. beginner, intermediate, advanced, etc.) is missing. This means that in the 100+ labs proposed, it is quite difficult for a user to figure out which are the best labs for his/her own competence.

Besides the above web contents aggregators, there are many educators' personal web sites which publish their own hand-outs or exercises. Although these educational materials are very interesting, they are often specific for a single type of users (for example, computer science graduated students with a high competence level), the resources proposed are not modifiable (e.g., they are in *pdf* format) and the source code of the proposed exercises is not available.

In this paper, we propose a new approach based on sharing the Cloud Computing teaching experiences of various educators. We provide a web content aggregator where all materials is open/shareable/modifiable by any educators. We also provide a open-source tool (developed by us) that enables an easy interaction with different cloud platforms.

III. Challenges

The three main challenges to face in order to properly teach Cloud Computing are related to (i) find out a real testbed where to practice with this recent computing paradigm, (ii) get confident with different cloud platforms in order to avoid the vendor lock-in issues and, finally, (iii) have a high-quality and wide-variety of educational materials ready to be exploited without much effort by the educators and easy to understand for students (despite their skills). In the rest of this section, we discuss in details these three challenges.

A. Real testbed

The first problem in teaching Cloud Computing is related to the testbed where the students can get practice on this computing paradigm. Conversely to other lab teaching activities where a typical student lab can be enough, a Cloud Computing testbed requires a (possibly big) set of computational resources geographically distributed worldwide. This scenario is almost impossible to replicate in a university: the money to run such a huge lab are difficult to find and, usually, a university has not computational resources available in different countries. At the time of writing, the only testbeds available ready to be used with the characteristics mentioned above are the one provided by the public cloud platforms such as AWS, GCP, IBM Cloud just to name a few. In particular, the educators/students may pay to rent computational power and/or service usage or they can use the free resources/services available in these platforms. These resources/services for free are very limited in terms of capacities. For example, AWS provides a "free-tier" resource that is a virtual machine with 1 vCPU with 2.5GHz, 1 GB of system memory, 8 GB of storage memory and low network performance (where low means "best effort") that can be executed by users for free. The other cloud platforms propose similar "low capacity" resources, so it is quite difficult for students/educators make some realistic exercise over these platforms. In the recent years, for educational purposes, AWS and GCP have proposed specific environment to provide better resources than free-tier. In particular, AWS started a program called *AWS educate* [16]: an academic gateway with educational resources for students and educators. In particular, in this program an educator can add his/her students in a virtual classroom and they will obtain 50$ of AWS credit. With this credit, the students can run more powerful virtual machines than the one proposed in the free-tier configuration or rent an Amazon AWS service for a limited amount of time. Google Cloud Platform provides a similar educational program with a limited amount of resources and for a limited period.

An alternative with respect to the public cloud platforms provided by the big of the IT is represented by an open-source project called *OpenStack* [17] which provides a free software to build up a cloud platform. This means that if a group of universities or companies can share their computational resources over the Internet, they can install and configure a cloud platform and use it for free without limitation in resource capacities (the limit depends only on the capacities of the resources shared) or time. In case these entities cannot share their computational resources, a viable solution is to use *Chameleon* [18]: a project which provides a real cloud testbed running the OpenStack software. In particular, Chameleon provides access to nearly 15,000 CPU cores, 5PB of total disk space, hosted across two sites, the University of Chicago and Texas Advanced Computing Center, connected by 100 Gbps network. Last but not least, Chameleon is completely free: a

```
$ aws ec2 run-instances \
    --image-id ami-1a2b3c4d \
    --count 1 \
    --instance-type c3.large \
    --key-name MyKeyPair \
    --security-groups MySecurityGroup
```

(a)

```
$ openstack server create --flavor 1 --image 397e713c-b95b-4186-ad46-6126863ea0a9 \
  --security-group default --key-name KeyPair01 --user-data cloudinit.file \
  myCirrosServer
```

(b)

```
$ gcloud compute instances create "my-new-instance" \
      --zone="us-west1-b" \
      --image-family="tf-latest-cu92" \
      --image-project=deeplearning-platform-release \
      --maintenance-policy=TERMINATE \
      --accelerator="type=nvidia-tesla-v100,count=8" \
      --machine-type="n1-standard-8" \
      --boot-disk-size=120GB \
      --metadata="install-nvidia-driver=True"
```

(c)

Fig. 1. How to run a new virtual machine by using the command-line client software provided by (a) AWS, (b) OpenStack and (c) GCP respectively.

very important characteristic for educators/students (and even for researchers).

B. Different cloud platforms

The second problem is related to the lack of standardization for interconnecting cloud platforms and this often results in users being locked into a specific provider and platform. Even if the basic features in each cloud provider are almost the same (e.g., run a Virtual Machine) the terminology used and how these features can be implemented are very different. This means that when a user starts to get confident with a specific platform, he/she is discouraged to try the same actions into another platform since he/she has to learn new terms and new procedures to replicate the same actions he/she did in the previous platform. From our point of view, it is really important that an educator/student get immediately confidence with more than one cloud platform. This because any cloud platform has its strengths and weaknesses and it is important to easily decide which is the best platform to adopt depending on the features required. The difference between cloud platforms is even more complicate when an expert user uses the command-line client software instead of the web interface provided by the various vendors. For example, in Figure 1 we show how a user can start a new virtual machine in AWS, OpenStack and GCP. By observing the three commands in the figure, it is worth to notice how different is the syntax and the number of parameters to be specified in order to execute the exact same action (i.e., starting a virtual machine) in the three platforms. Once a user is getting confident with one specific command-line client software, it is hard for him/hers learning another set of commands to execute the same actions. In order to face this problem, the scientific community has proposed some tools such as *Cloudmesh* [19] and *EasyCloud* [20] (where EasyCloud is an evolution of our previous toolkits [21], [22]): both projects provide an easy text user interface to manage virtual machines and bare metal provisioned operating systems in a multicloud environment. Just to give an overview concerning the interface of these tools,

```
******************* EasyCloud *******************

Select a platform
-------------------------
1) Amazon Web Services
2) Google Cloud Platform
3) OpenStack
4) Close application

Please make a choice:
> 
```

Fig. 2. The EasyCloud main menu.

```
******************* EasyCloud *******************

What would you like to do?
-------------------------

1)  Create new instance
2)  Show running instances
3)  Terminare instance
4)  Reboot instance
5)  Manage floating IPs
6)  Manage volumes
7)  Extra functions
8)  Start monitor
9)  Manage rules
10) Edit configuration file
11) Back to main manu

Please make a choice:
> 
```

Fig. 3. The EasyCloud main features.

in Figure 2 we show the main menu provided by EasyCloud where a user can select which Cloud Computing platform to use and then, in Figure 3 we show the main actions available for each Cloud platform. Thanks to these tools, the user does not have to remember the exact syntax of each Cloud platform client software and can easily not only run instances of virtual machines but also perform some advanced actions such as implementing a load balancer or a fault tolerant mechanism. The details concerning all the features provided by these tools is out of the scope of this paper. The interested reader can install/configure/run the tools by downloading them from their website (e.g., for EasyCloud the source code is in [23]).

C. Educational resources

The third problem is related to the quality, clarity and accessibility of the educational resources freely available in the web. As mentioned in the previous section neither the web portal specific to collect cloud educational resources (such as Google Codelabs or Qwiklab) nor the page of the courses provided by educators are able to satisfy the accessibility, clarity and quality requirements. From our point of view, there is not the perfect educational material who fits all the necessity of a variety of users interested in Cloud Computing. Only a collaborative platform, where each educator can share his/her experience and provide his/her material can be actually useful. Thanks to this collaborative platform, it will also be easy for educators to update their materials: the constant evolution of the software tools available makes the whole cloud ecosystem different from one year to the next (i.e., the commercial cloud platforms such as AWS and GCP, are constantly adding new products and adapting/removing old ones). Few years ago, around 2011, we start an experiment: we share online all our educational materials and we ask to cloud communities to comment and use our material in their class. The outcome was promising: many educators and cloud users appreciated our work and they proposed updates and new contents. Our educational materials is available for everyone, we are constantly hearing from any suggestions which come from the user cloud community (we have proposed our course to the *Chameleon users group* [24] and *Google Cloud Faculty Community group* [25]) and we have also created a mailing list where students/educators can ask/propose new educational resources. The material mentioned above is available in [26] and in the next section we describe in details the syllabus of our "collaborative" Cloud Computing course.

IV. Syllabus

In this section, we introduce the syllabus of the Cloud Computing course proposed to the various user cloud communities. The course is organized in two main paths: the path for beginners and the path for advanced users. The *beginner path* (presented in Section IV-A) has been designed specifically for new users without any computer science skills, while the *advanced path* (presented in Section IV-B) has been designed specifically for users with some basic program skills. All the lessons proposed are freely available for download and any educator can propose new educational materials. For each lesson, we report in parentheses the estimate amount of hours necessary to teach the proposed arguments. As mentioned before, to support students and educators, a mailing list has also been created to comment and to help both educators and students to better understand what is Cloud computing and how it works. In the rest of this section we describe in detail the two paths.

A. The beginner path

1) **Introduction to Cloud Computing ($\sim 2h$):** in this section, we introduce the definitions and basic concepts related to the Cloud Computing. Then we describe both the service models (IaaS, PaaS, SaaS) and the deployment modes (Public, Private, Community, Hybrid). We also discuss pros and cons of Cloud Computing in various realistic scenarios.
2) **Cloud Platforms ($\sim 2h$):** in this lesson, we provide an overview of the main Cloud Computing platforms discussing their characteristics. In the final part of the lesson we compare cloud platforms with respect to these characteristics by highlighting the strengths and weaknesses of each cloud platform with respect to the others.
3) **OpenStack and Chameleon project($\sim 6h$):** this sequence of lessons are dedicated to the OpenStack project, an open source Cloud Computing platform. The practical exercises are performed on the Chameleon infrastructure. The topic of each lesson follows:
    a) *Introduction to OpenStack ($\sim 2h$)*: after a brief introduction concerning the characteristics of OpenStack, we describe in detail how its web console works in order to show to the students the steps to start/stop/clone/access into a Virtual Machine.
    b) *LAMP+CMS ($\sim 2h$)*: in this lesson, the students create a Virtual Machine (VM) running a web server with a *Content Management System* (CMS) (i.e., Joomla or Wordpress) over a *Linux, Apache, MySQL, PHP/Perl/Python* (LAMP) software stack (i.e., a software stack consisting of a Linux operating system, an Apache HTTP server, a MySQL database management system, and a set of scripts written in the PHP/Perl/Python programming languages). This VM is used in the next lessons to perform the exercises proposed.
    c) *Manage volumes ($\sim 2h$)*: in this lesson, the students learn how to manage storage volumes by performing some simple exercises where they simulate a service failure and recovery. In particular, the students have to periodically save the data of a service into a volume (that is, they backup data related to the service) so that when the service dies (that is, the virtual machine where the service is running becomes unreachable), the new VM (started automatically to replace the one that just died) can recover the data from the backup stored in the above volume and, therefore, continue the service that was interrupted. In other words, the students have to implement what is called *businesses continuity*.
4) **Amazon Web Service (AWS) ($\sim 4h$):** Cloud Computing was popularized with Amazon releasing its *Elastic Compute Cloud* (EC2) product in 2006. Thus, in this sequence of lessons, we introduce the "father" of Cloud Computing platforms with the following topics:
    a) *Introduction to AWS ($\sim 2h$)*: in this lesson, we present AWS and its vast ecosystem.
    b) *The AWS's Console ($\sim 2h$)*: in this lesson, the students get confidence with this cloud platform by interacting with its web console in order to perform the basic actions such as starting/stopping/cloning a VM.
5) **Google Cloud Platform (GCP) ($\sim 4h$):** The cloud platform proposed by Google in discussed in detail in this sequence of lessons as described below:
    a) *Introduction to GCP ($\sim 2h$)*: we introduce the GCP features and we compare them with AWS.
    b) *The GCP's Console ($\sim 4h$)*: the students start to use GCP by exploiting its web console and performing the basic actions.
6) **Cloud Federation ($\sim 6h$):** in these lessons the students will learn how to use the *Cloudmesh* and *EasyCloud*

tools in order to interact with various cloud platforms. As mentioned before, this is really important in order to avoid the lock-in vendor "trap" and to fully exploit the characteristics of each cloud platform. The topics of each lesson follow:

a) *Cloud standardization ($\sim 2h$)*: in this lesson, we describe the various attempts to create standard API to interact with any cloud platform and what is still missing in order to achieve this important goal.
b) *The Cloudmesh project ($\sim 2h$)*: the students use the *Cloudmesh* tool to perform some basic actions such as starting/stopping/cloning a VM.
c) *The EasyCloud project ($\sim 2h$)*: the students use the *EasyCloud* tool to perform some basic actions such as starting/stopping/cloning a VM.

At the end of the last two lessons, the students are invited to discuss about these two tools and how it is important standardization in Cloud Computing.

7) **Beyond Cloud Computing** ($\sim 4h$)**:** in these lessons we discuss what will be the Cloud Computing in the next future.
   a) *Containers ($\sim 2h$)*: we introduce the concept of Container and we discuss the differences between Virtual Machines and Containers. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another [27].
   b) *Fog/Edge computing ($\sim 1h$)*: Fog/Edge Computing [28]–[30] is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively, located at the edge of network. This new paradigm has been specifically proposed to satisfy the real-time requirements of the services that will delivered in the near future to end users (e.g., autonomous vehicles and smart grids). In this lesson, we present this new important paradigm, by introducing its basic concepts and discussing how it can work in synergy with the cloud.
   c) *Femtocloud systems($\sim 1h$)* : a Femtocloud system is an emerging computing system consisting of a ephemeral set of heterogeneous mobile devices whose owners allow to run tasks offloaded by other users [31]–[33]. In this lesson, we present this recent system and we discuss how it can be integrated in both Fog/Edge and Cloud Computing platforms.

B. The advanced path

1) **OpenStack and Chameleon project($\sim 6h$) :**
   Once the students are confident with the Chameleon testbed they are ready for more advanced tasks. The topic of each lesson follows:
   a) *Load balance ($\sim 2h$)*: the students learn how to automatically clone/stop a VM depending on load intensities. In other words, the students have to implement what is called *elasticity*.
   b) *Advanced usage of the command-line client software ($\sim 2h$)*: a set of exercises to exploit the OpenStack command-line interface are proposed.
   c) *Customization scripts ($\sim 2h$)*: with this option the user can specify a script to be executed immediately after the Virtual Machine boot. This option has two advantages: (i) the user can save time by avoiding to perform this installation/configuration steps manually, and (ii) the user can configure a single Virtual Machine base image and use it for different services, by simply providing the right customization scripts, instead of having different Virtual Machine images for the different services.
2) **Amazon Web Service (AWS)**($\sim 4h$) **:**
   a) *The AWS's command-line client software*: by using the AWS's command-line client software, the students can create bash scripts to automate some actions. Various exercises are proposed to the students in order to learn the potentiality of this command-line client software.
3) **Google Cloud Platform (GCP)**($\sim 4h$) **:** For the main AWS competitor, we provide two advanced lessons:
   a) *The GCP's command client ($\sim 2h$)*: the students will use the basic commands of the GCP's command-line client software and compare them with the command-line interface offered by AWS.
   b) *A MapReduce example ($\sim 2h$)*: the students implement an easy *MapReduce* [34] example. MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.
4) **Cloud Federation** ($\sim 6h$)**:** Behind tools like EasyCloud and Cloudmesh there are several libraries which enable access to different cloud systems. In this sequence of lessons, we describe in detail how they work and we propose some exercises based on them. The topic of each lesson follows:
   a) *The Boto library [35] ($\sim 2h$)*: this is the Python library for AWS and it supports more than 50 AWS services. We propose to the students to write scripts to interact with the services related to elastic Cloud Computing and autoscaling.
   b) *The Apache libcloud [36] ($\sim 2h$)*: this is a Python library that provides a unified API for interacting with many popular cloud service providers (e.g., AWS, GCP, and OpenStack, just to name a few). We propose a working script and ask to the students to improve it by adding new features such as monitor some parameter of a virtual machine.

c) *The EasyCloud tool ($\sim 2h$)*: advanced actions are possible with our tool. In particular, it is possible to set some automatic procedure in order to implement customized load balancing policies.

5) **Cloud as a platform ($\sim 16h$):** beyond data storage and computing, there are high-level services that are particularly well suited to research applications. In particular, after an introduction concerning the big data concepts, we discuss the most recent solutions for data analysis, machine learning and streaming data analysis. We also propose some practical exercises by using the *Apache Hadoop* [37]: a software library that allows for the distributed processing of large data sets across clusters of computers using simple programming models. The topic of each lesson follows:
   a) *Introduction to Big Data ($\sim 2h$)*: we introduce the concepts of Big Data and we discuss the challenges for storing, processing and visualizing huge amount of data.
   b) *Data analytics ($\sim 6h$)*: we discuss the basic principles for designing a Big Data processing platform. We also present the Apache Hadoop project (and its ecosystem) and we discuss in detail its architecture, including the HDFS and YARN components. Then, we present the MapReduce paradigm and we propose to students several lab activities based on the MapReduce implementation provided by Apache Hadoop. Finally, we discuss the issues of the MapReduce and we introduce the directed acyclic graph computing model as a way to overcome those issues. To this end, we present the *Apache Spark* [38] platform and we propose to students some lab activities based on it.
   c) *Stream analytics ($\sim 6h$)*: we discuss the basic principles and challenges for designing real-time event processing platforms. Then, we present the *Apache Spark Streaming* [39], the *Apache Storm* [40], the *Apache Flink* [41] and the *Amazon Kinesis* [42] platforms, by discussing their architecture and programming model. Finally, we propose to students some lab activities based on the above platforms.
   d) *Machine learning ($\sim 2h$)*: we introduce the basic concepts of machine learning and we discuss the challenging for running machine learning algorithms in large-scale distributed systems and with huge amount of data. Then, we present the *Apache Spark MLlib* [43] library and we propose some lab activities to students based on this library.

V. Conclusions

In this paper, we discuss the state of the art, the challenges and the resources related to teaching Cloud Computing. This recent computing paradigm is everywhere and is being used not only by computer scientist but also by a wide variety of users with different backgrounds and skills. For these reasons, it is important to teach Cloud Computing not only in the computer science curriculum, and in order to achieve this goal the educational materials must be understandable for any kind of users. For this purpose, we provide educational materials free to use and tools to help user to explore different cloud platforms.

In the future, we plan to improve the educational material by adding new resources dedicated to other cloud platforms such as IBM Cloud, Microsoft Azure just to name a few and we also plan to increase the number of practical exercises, possibly by introducing additional tools suitable for experimenting with virtualized infrastructures (e.g., [44]). Moreover, we are recording some lessons and we will put them available for everyone in our website like a *Massive Open Online Courses* (MOOCs). We are also studying how to improve the collaboration between educators and students by testing specific platforms for educational purposes such as *Moodle* [45], *MIT Star platform* [46] and *Piazza* [47]. Finally, we plan to take surveys among students at the end of the course, to collect their feedbacks and improve the course organization consequently.

Acknowledgment

This research has a financial support of the Università del Piemonte Orientale and of the INdAM – GNCS Project 2020.

References

[1] P. M. Mell and T. Grance, "Sp 800-145. the nist definition of cloud computing," Gaithersburg, MD, USA, Tech. Rep., 2011.
[2] Amazon, "Amazon Web Services," available: https://aws.amazon.com. Accessed: Jan 25, 2020.
[3] Google, "Cloud," available: https://cloud.google.com/. Accessed: Jan 25, 2020.
[4] Microsoft, "Azure," available: https://azure.microsoft.com/. Accessed: Jan 25, 2020.
[5] IBM, "IBM Cloud," available: https://cloud.ibm.com/login. Accessed: Feb 9, 2020.
[6] Amazon, "Genomic in the Cloud," available: https://aws.amazon.com/health/genomics/. Accessed: Feb 9, 2020.
[7] I. Cloud, "The databiology's platform," available: https://databiology.com/. Accessed: Feb 9, 2020.
[8] XSEDE, "FutureGrid," available: http://archive.futuregrid.org/. Accessed: Feb 9, 2020.
[9] UCSB, "Eucalyptus," available: https://www.eucalyptus.cloud/. Accessed: Feb 9, 2020.
[10] University of Chicago, "Nimbus," available: http://www.nimbusproject.org. Accessed: Feb 9, 2020.
[11] FutureGrid community, "Community Educational Materials," available: http://archive.futuregrid.org/community_edu_materials.html. Accessed: Feb 9, 2020.
[12] Coursera, "Cloud computing courses," available: https://www.coursera.org/browse/information-technology/cloud-computing. Accessed: Feb 9, 2020.
[13] Clare Corthell, "The Open Source Data Science Masters," available: http://datasciencemasters.org/. Accessed: Feb 9, 2020.
[14] qwiklabs, "Hands-On Cloud Training," available: https://www.qwiklabs.com. Accessed: Feb 9, 2020.
[15] Google, "Codelabs," available: https://codelabs.developers.google.com/. Accessed: Feb 9, 2020.
[16] Amazon, "Aws educate," available: https://aws.amazon.com/education/awseducate/. Accessed: Feb 9, 2020.
[17] OpenStack, "OpenStack Compute (nova)," available: https://docs.openstack.org/nova/latest/. Accessed: Jan 25, 2020.
[18] K. Keahey, P. Riteau, D. Stanzione, T. Cockerill, J. Mambretti, P. Rad, and P. Ruth, "Chameleon: a scalable production testbed for computer science research," in *Contemporary High Performance Computing*. CRC Press, 2019, pp. 123–148.

[19] G. von Laszewski, B. Abdul-Wahid, F. Wang, H. Lee, G. C. Fox, and W. Chang, "Cloudmesh in support of the nist big data architecture framework," Technical report, Indiana University, Bloomingtion IN 47408, USA, Tech. Rep., 2017.

[20] C. Anglano, M. Canonico, and M. Guazzone, "Easycloud: a rule based toolkit for multi-platform cloud/edge service management," in *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2020, accepted for publication.

[21] M. Canonico and D. Monfrecola, "Cloudtui-fts: A user-friendly and powerful tool to manage cloud computing platforms," in *Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS'15. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, p. 220–223. [Online]. Available: https://doi.org/10.4108/eai.14-12-2015.2262718

[22] M. Canonico, A. Lombardo, and I. Lovotti, "Cloudtui: A multi cloud platform text user interface," in *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools*, ser. ValueTools '13. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, p. 294–297. [Online]. Available: https://doi.org/10.4108/icst.valuetools.2013.254413

[23] The UPO's Distributed Computing Systems group, "EasyCloud repository," https://gitlab.di.unipmn.it/DCS/easycloud/, 2020, [Online; accessed 25-Jan-2020].

[24] Chameleon, "Users," available: https://lists.chameleoncloud.org/mailman/listinfo/users. Accessed: Jan 25, 2020.

[25] Google, "Cloud Faculty Community," available: https://groups.google.com/forum/#forum/googlecloudfaculty. Accessed: Jan 25, 2020.

[26] M. Canonico, "Teaching cloud computing," available: https://sites.google.com/uniupo.it/massimocanonico/teaching-cloud-computing. Accessed: Feb 9, 2020.

[27] D. Bernstein, "Containers and cloud: From lxc to docker to kubernetes," *IEEE Cloud Computing*, vol. 1, no. 3, pp. 81–84, 2014.

[28] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[29] C. Anglano, M. Canonico, P. Castagno, M. Guazzone, and M. Sereno, "A game-theoretic approach to coalition formation in fog provider federations," in *Proc. of the* 3^{rd} *International Conference on Fog and Mobile Edge Computing (FMEC)*. Barcelona, Spain: IEEE, April 2018, pp. 123–130.

[30] ——, "Profit-aware coalition formation in fog computing providers: A game-theoretic approach," *Concurrency and Computation: Practice and Experience*, 2019, in Press.

[31] K. Habak, M. Ammar, K. A. Harras, and E. W. Zegura, "Femtoclouds: Leveraging mobile devices to provide cloud service at the edge," in *Proceedings of the 2015 IEEE 8th International Conference on Cloud Computing*, ser. CLOUD '15. IEEE, 2015, pp. 9–16.

[32] C. Anglano, M. Canonico, and M. Guazzone, "Online user-driven task scheduling for FemtoClouds," in *Proc. of the* 4^{th} *International Conference on Fog and Mobile Edge Computing (FMEC)*. Rome, Italy: IEEE, June 2019, pp. 5–12.

[33] ——, "WQR-UD: An online scheduling algorithm for FemtoClouds," in *Proc. of the* 12^{th} *EAI International Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS 2019, 2019, pp. 179–182.

[34] Wikipedia, "Maprduce," available: https://en.wikipedia.org/wiki/MapReduce. Accessed: Feb 9, 2020.

[35] C. hackers, "Boto," available: https://github.com/boto/boto3. Accessed: Feb 9, 2020.

[36] Apache, "Libcloud," available: https://libcloud.apache.org/. Accessed: Feb 9, 2020.

[37] P. Zikopoulos, C. Eaton *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.

[38] Apache, "Spark," available: https://spark.apache.org/. Accessed: Feb 9, 2020.

[39] ——, "Spark streaming," available: https://spark.apache.org/streaming/. Accessed: Feb 9, 2020.

[40] ——, "Storm," available: https://storm.apache.org/. Accessed: Feb 9, 2020.

[41] ——, "Flink," available: https://flink.apache.org/. Accessed: Feb 9, 2020.

[42] A. W. Services, "Kinesis," available: https://aws.amazon.com/kinesis/. Accessed: Feb 9, 2020.

[43] Apache, "Mllib," available: https://spark.apache.org/mllib/. Accessed: Feb 9, 2020.

[44] C. Anglano, M. Canonico, and M. Guazzone, "Prometheus: A flexible toolkit for the experimentation with virtualized infrastructures," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 11, p. e4400, 2018.

[45] J. Cole and H. Foster, *Using Moodle: Teaching with the popular open source course management system.* " O'Reilly Media, Inc.", 2007.

[46] MIT, "Star platform," available: http://star.mit.edu/. Accessed: March 12, 2020.

[47] I. Tatarinov, Z. Ives, J. Madhavan, A. Halevy, D. Suciu, N. Dalvi, X. Dong, Y. Kadiyska, G. Miklau, and P. Mork, "The piazza peer data management project," *ACM Sigmod Record*, vol. 32, no. 3, pp. 47–52, 2003.