Journal of Universal Computer Science, vol. 19, no. 14 (2013), 2112-2126
submitted: 3/2/13, accepted: 29/7/13, appeared: 1/8/13 © J.UCS

Laboratories as a Service (LaaS): Using Cloud Technologies in the Field of Education

Rafael Pastor, Agustín Carlos Caminero, Daniel Sánchez, Roberto Hernández Salvador Ros, Antonio Robles-Gómez, Llanos Tobarra
(Control and Communication Systems Department
National Spanish Distance University of Spain, UNED
Madrid, Spain
{rpastor, accaminero, dsanchez, roberto, sros, arobles, llanos}@scc.uned.es)

Abstract: Society has evolved in such a way that individuals are required to embrace constant improvements in order to be able to perform their jobs properly. Distance education is a solution to this problem, as it allows students to obtain practical knowledge without the space and time constraints of classical face-to-face education, thus allowing them to fit their studies into possibly tight schedules. In order to obtain practical distance education on technical topics, the use of remote laboratories becomes more of a necessity rather than just being an option. To this end, the RELATED framework has been developed in order to permit the structural development of remote laboratories. It presents a structured methodology of remote/virtual lab development and also provides common facilities, such as user management, booking, or basic visualization. In the case that a high number of laboratories and students use RELATED, handling such information becomes a major issue for the proper functionality of RELATED. These issues can be efficiently tackled using cloud technologies. This paper proposes the use of cloud technologies to enhance RELATED and describes the cloud-based architecture that is under development at UNED, including details on its software components and the algorithms needed for resource provisioning.

Keywords: Cloud computing, On-line education, Remote laboratories, Web-based services
Categories: C.2.4, H.3.5, J.6, K.3, K.6, L.3.0, L.3.5, L.3.6

1 Introduction

Cloud computing allows on-demand network access to fully configurable computing resources. These resources can be provisioned and released with minimal management effort or service provider interaction [NIST, 2013]. Among the benefits of using cloud technologies, we can find [García-Peñalvo et al., 2012] [IBM, 2013] [IBM, 2009]: lower cost of ownership, the more efficient use of technical staff, cloud computing saves time, money, a shortened production cycle, organizations can store more data than can be stored on private computer systems, and cloud computing offers much more flexibility than past computing methods do.

The cloud allows a more efficient use of computing resources because it represents a shift from the previous "computer-guided" computing architectures to a more versatile "user-guided" system. In traditional systems, computers had static software features, and users of such resources had to "fit" into those features. For example, if a shared computer had a Linux operating system installed along with some programs and libraries, users who were willing to run their applications on it

had to make sure that their applications could run on such a system. Hence, the use of computing systems could be considered as "computer-guided," as users had to fit their applications in order to satisfy the computer features.

The cloud allows systems to dynamically provide the computing resources their users need, thus reducing expenses and energy consumption as well as improving their scalability [Vázquez et al., 2009] [Buyya et al., 2010]. Hence, if users want to run some applications in a cloud, it is the computer that has to "fit" into the user's needs. In the example above, virtual machines (VM) can be dynamically instantiated in order to meet the users' requirements. The cloud system can therefore be considered a "user-guided" system, as it is a computing resource that is adapted to users' needs. Furthermore, an appropriate cloud infrastructure manager (such as OpenNebula [Sotomayor et al., 2009] or Eucalyptus [Nurmi et al., 2009]) can provide on-demand instantiation and monitoring, along with the live migration of VMs. Consequently, fault tolerance and scalability are provided.

Another important point to keep in mind is computers' power consumption [Buyya et al., 2010]. According to [Kaplan et al., 2008], datacenters now drive more carbon emissions than do both Argentina and the Netherlands. Thus, cloud infrastructures should be managed in order to try to reduce computers' power consumption along with maintaining the efficient utilization of machines—with this being called green computing.

The evolution of education and the increase in the knowledge of the necessities our society requires have created significant changes in how the learning process takes place. Nowadays, there is a constant need to improve, to keep our knowledge up-to-date, or to obtain knowledge on new topics—with this being especially true in the case of technical studies, where technology is constantly evolving. Distance education is a solution to this problem because it allows students to obtain practical knowledge without the space and time constraints of classical face-to-face education; thus, they can fit their studies into possibly tight schedules.

In our case, the National Distance Education University of Spain (Universidad Nacional de Educación a Distancia, UNED) is the largest university in Spain, with more than 200,000 students. We provide totally distant education, so the use of remote laboratories to obtain practical knowledge on technical topics becomes a necessity more than just an option. For this, the RELATED framework [Pastor et al., 2006] has been developed in order to permit the structural development of remote laboratories. It presents a structured methodology for the development of remote/virtual labs and also includes common features, such as user management, booking, or basic visualization. In the case that a high number of laboratories and students use RELATED, handling such an amount of information and such a workload becomes a major issue for the proper functionality of RELATED. Besides, the use of cloud technologies allows us to adapt the RELATED infrastructures in order to fit them to the current or forecasted workload—thanks to which we can reduce expenses in terms of power consumption.

This paper proposes the use of cloud technologies to enhance RELATED and to tackle these issues. This paper's structure is as follows. Section II presents the fundamentals of virtual/remote laboratories and their related distance-learning scenarios. Section III provides brief information on the RELATED framework. Section IV presents the extensions that are harnessing the cloud technologies under

development at UNED. Section V details the way we perform resource provisioning, and finally, Section VI presents conclusions and future work.

2 Remote and virtual laboratories in distance education

The acquisition of practical knowledge is one of the most challenging issues in engineering education—by far, more challenging than the acquisition of theoretical knowledge is. in giving students a taste of real situations, for example, the behavior of measure instruments, or how control action works, is an essential aspect nowadays. All of these issues can be solved using on-site laboratory experiments, although this can be tricky with a large number of students or in a distance-learning environment—which is the case of the National Distance University of Spain (UNED). In a distance-education model, such as UNED's, students have no classrooms but rather have e-learning environments instead. Typically, these e-learning environments, provided via learning management systems (LMS), offer communication, assignments, and content tools (among others) for all of the participants of each subject (e.g. students, faculty).

However, for experimental subjects, no specific tools are available inside the e-learning space. In order to provide full access to online laboratories, Web-based applications must be provided in order to achieve "universal" access to students using Web browsers. Additionally, integrating such applications within the courses' e-learning spaces is easy. However, first, it is necessary to look inside the different categories associated with online laboratories. Three main categories exist:

- Software laboratories. These are software packages that can be downloaded from different sources (ftp, mail, or Web) and require installation on the client computer in order to run the laboratory. Obviously, the client computer must fit some requirements, so students should have a suitable computer in order to use the laboratory.
- Web-based virtual laboratories. Again, they are software packages that represent the behavior of the laboratory (a physical system, for example) using a proper simulation software. The difference between this and the above category is in the installation issues and computer requirements. In this case, only a Web browser is needed—no installation of software on the students' computers is required, with the exception of some browser plug-ins, such as Flash, Java, etc. Thus, access to and usage of these laboratories is given to every student who has a recent browser installed on his/her computer.
- Web-based remote laboratories. In this case, laboratories are physical equipment in remote locations (universities, companies, etc.) and are managed using Web interfaces through Web browsers. No installation issues are involved (maybe some plug-ins are needed, as for the previous case). Again, a Web browser is the only tool that students need to use for this kind of laboratory. In fact, the virtual and remote laboratories have the same requirements (Web browser and plug-ins), but the main difference is the entity (a simulation in the case of the virtual laboratory or real equipment/instruments in the case of the remote laboratory) and the behavior they are representing. Thus, the development of virtual and

remote laboratories is very similar (despite the software requirements for the physical equipment/instrument technology).

The use of software laboratories has several disadvantages associated mainly with the problems that stem from the non-fulfillment of a student's computer requirements and the installation of software packages. In a distance-learning model, it is very important to avoid these issues due to the non-presence of lecturers in a classroom. A virtual/remote laboratory is a more controlled environment because the errors/problems can be solved in a remote way by updating the related laboratory software running inside the Web browser. This is the main reason for the adoption of virtual/remote laboratories instead of software laboratories.

The application of virtual laboratories in learning scenarios for distance education allows students to interact with simulation models for several types of systems, such as physical systems (motors, level tanks, etc.) or event-based systems (traffic control, database systems, etc.). In the case of remote laboratories, the idea is to allow students to be able to perform actual experiments [Gomes and Bogosyan, 2009] [Guimaraes et al., 2011] [Sancristobal et al., 2011] in real time and on real equipment over the Internet (in a way similar to that of on-site experiments). This idea is, in fact, a widely discussed issue [Nourdine et al., 2010].

Along with the concept of virtual/remote laboratories (considered a service for students), other typical services that Web applications need also appear. These services include issues related to the management of users, booking/access control, sessions, and role assignments. All of these services are available using a software/hardware infrastructure with several common elements in Web applications: Web services, databases, hardware servers, and network connections. This infrastructure must allow multiple concurrent users in order to provide solid performance from the user's point of view (students and lecturers). Thus, the infrastructure must be built as a set of hardware/software resources with high availability. Therefore, cloud computing becomes a real alternative in order to provide high availability to this infrastructure The infrastructure used in a typical distance-learning scenario in UNED works as follows:

1) A lecturer makes available his/her virtual/remote laboratory in the infrastructure (using his/her own computing resources).
2) Students register to get access to these laboratories using a Web portal (which is part of the infrastructure).
3) A student performs a real experiment using the remote laboratory, thus creating an experimental session (data are saved in the corresponding service inside the infrastructure).
4) Lecturers analyze experimental data in saved sessions to provide an evaluation of the experiments that students perform. Again, infrastructure services provide these experimental data.
5) Lecturers use the e-learning environment of UNED to inform students about their evaluations.

This learning scenario is implemented in several subjects included in two official degrees of UNED:

- Information and communication technology degree (computer science faculty)
- Industrial electronic and control engineering (electrical and electronic engineering faculty)

3 The RELATED framework

RELATED framework [Pastor et al., 2006], [Pastor et al., 2012a], [Pastor et al., 2012b] proposes a structured methodology of remote/virtual lab development and also provides common features as user management, booking, basic visualization (trend graphs and direct interaction using interactive variables), data logging, and experimental session control. A Remote LABoratory (RLab) system is defined using a formal specification (which is called LEDML, laboratory experimentation description language, based on XML).

The RELATED architecture is based on the module paradigm that leads to a structured development strategy. In this way, laboratories are developed in a more rational way, thus reducing development times and optimizing human resources. With RELATED, starting the process of remote laboratory development from scratch is not necessary.

Section 2.1 describes the development process of a remote laboratory using the RELATED framework. Section 2.2 describes the communication architecture of a RELATED RLab with the experimentation client.

3.1 Development using RELATED

An RLab system consists of several components, as can be seen in Figure 1. The main component is an experiment that is defined using XML. Experiments are composed of modules and views. Modules and views can be reutilized in other experiments of the same or different RLab systems. This modular approach leads to code-reusing and facilitates the expansion of existing laboratories.

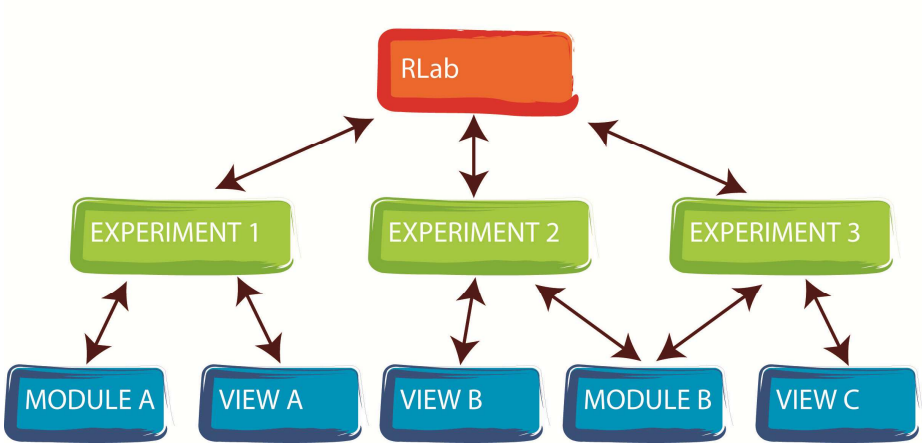

Figure 1: Elements of an RLab

The lab designer develops modules in order to provide local access to laboratory equipment. The RELATED facilities start these modules, which are run-able entities, in order to get/set data from/to the laboratory equipment. This data will be sent over the Internet to the experimentation client, too (which is executed in the student's computer using a Web browser and a Java virtual machine (JVM).

A "view" provides a graphical user interface (GUI) to the final user. These views use data from modules to update the experiment visualization. Updating the modules' values from the view entity is possible. That is, the user can interact directly with the experiment using one or more views. Some samples of view interfaces can be seen in Figure 2.

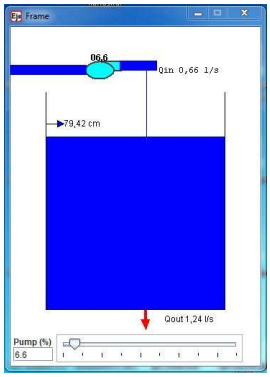

Figure 2: Examples of views

Java is used to develop the RLab entities, such as the modules and views. In the case of views, several utilities simplify the programming process. Easy Java Simulations (EJS) [Esquembre, 2004] is a free authoring tool that helps non-programmers to create interactive simulations and GUI interfaces in Java. Lecturers or teachers who wish to develop remote laboratories using the RELATED framework can leverage EJS to develop views with a simple reusing procedure of code produced by EJS.

Another tool that can help non-programmers to create view modules is the GLG Toolkit [GENLOGIC, 2013]. View samples showed in Figure 2 are created using EJS.

The next step once every module and view is developed is to prepare the XML file, which provides the definition of the laboratory (the RLab system). Specific tags are used for experiments, modules and views. Figure 3 shows a sample of an XML experiment definition.

Several sub-tags exist for each <*module*> or <*view*> tag. The most important one is the <*implementation*> tag. With this <*implementation*> tag, the developer of the remote laboratory can specify the Java Archive (JAR) file where RELATED can find the classes that implement the module or view. Some tags also specify the module variables—among others we can find the tag <*var name="name"*>—which has several options for specifying the type of the variable as well as the initial, maximum, and minimum values, and so on. For <*view*> tags, a special sub-tag that is used is the

<use name> tag. This tag is helpful for specifying every variable of a module that will be used in a view. More information about developing remote laboratories using RELATED can be found in [Pastor et al., 2012b].

```
- <experiment name="Speed control" logging="no" sampleTime="100">
    Speed control using RT PID
    <duration type="time" time="300"/>
  - <run type="remote" module="motorRemotoMan" source="//62.204.199.218:1099/D71F73A7AE4E112B5DF4172E60E059E5">
      <interactives show="false,false,false,false,false,true" names="current,tacho,position,voltage,speed,setLight"/>
      <paint names="voltage" colors="black"/>
    </run>
  - <run module="RT PID module">
      <interactives show="true,true,true,true,true,true,true" names="yref,kp,ti,td,Beta,Tr,N"/>
      <paint names="y,yref" colors="yellow,red"/>
      <in name="y" module="motorRemotoMan" source="//62.204.199.218:1099/D71F73A7AE4E112B5DF4172E60E059E5" var="speed"/>
      <out name="u" module="motorRemotoMan" source="//62.204.199.218:1099/D71F73A7AE4E112B5DF4172E60E059E5" var="voltage"/>
    </run>
    <run module="VIDEO SERVER module"/>
    <open view="ImageViewer"/>
    <open view="MotorQETEJSViewMan"/>
  </experiment>
```

Figure 3: XML experiment sample

When the XML RLab laboratory definition is ready, the last step involves publishing the laboratory. In order to do this, RELATED provides a publish application that can be seen in Figure 4. This application parses the XML file and then uploads to the RELATED Server every file needed to run the lab. When the process is complete, a new laboratory can be accessed from the RELATED Web page [RELATED, 2013].

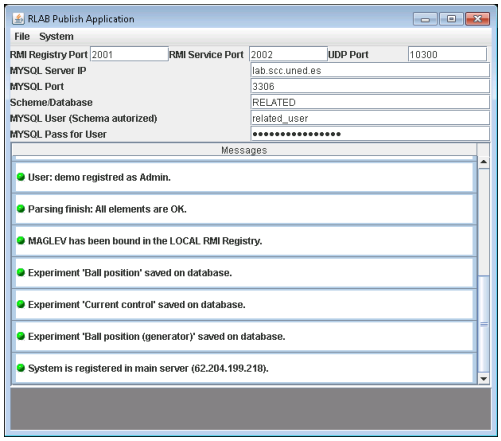

Figure 4: RLAB publish application

One of RELATED's interesting features is that it has booking and user control features. In this way, only authorized users can see/manage a laboratory depending on their assigned roles.

The publish application must be run on the computer that will be the laboratory server because it is also the piece of software that manages communications between lab equipment and the RELATED Web server. That is, the application keeps running on the lab machine in order to provide access to the lab equipment.

RELATED also provides several interesting features for the final user. One of these features is the experiment control panel (the experimentation client). The experiment control panel is the place where most of the activity associated with using the remote lab takes place. First of all, the user must log in on the RELATED Web page. Then, all of the experiments allowed for this user will be shown; when the user tries to access one of these experiments, a login window appears. Then, the user credentials must be entered, and if the credentials are valid, the experiment control panel pop ups. The experiment control panel can be seen in Figure 5.

When the student logs into the experiment, he/she reserves a time slot; this slot time is assigned to avoid multiple concurrent users and can be set using the booking system provided for RELATED, so a start and a finish date is assigned to the running experiment and the user. The experiment control panel shows a clock to indicate to the student the time remaining to complete the experiment.

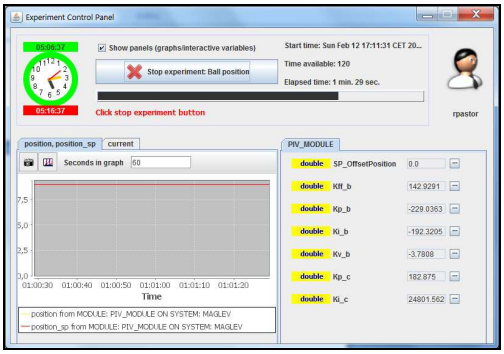

Figure 5: RLAB experiment control panel

Some other parts of the experiment control panel window are the trend graphs and the interactive variables panel. Several trend graphs can be seen in each graph. Several graphs can be set up for each experiment. This is a common way to show information for control systems. In the right part of the window is the interactive variables place, which features several variables that can be the user can configure, thus interacting with the laboratory equipment.

3.2 RLAB communication architecture

Among the possible improvements for RELATED, one of the most prominent is a communication structure that defines how RELATED provides remote laboratory functionality from a more technical point of view.

Remote laboratories developed under RELATED can be accessed using two different technologies. The first is a distributed object system, which uses Java Remote Method Interface (RMI) to control the experiment state, including a self-developed protocol on top of the User Datagram Protocol (UDP) in order to get/set data from/to the experiment. The use of UDP allows us to obtain better performance in data transmission. The second is a REpresentational State Transfer (REST) system that can be viewed as a Web-service technology.

However, at a more organizational level, what are the main components of a RELATED RLab system? In a RELATED RLab system are four important components that can be seen in Figure 6. First is the final *user computer*, that is, a personal computer (PC). Second is the *RELATED Web server* that handles access to the experiments and the communication between the *laboratory server* and the *user computer*. The *laboratory server* is the piece of equipment used to handle the communications between the laboratory equipment and the *RELATED Web server*.

Then, some kind of technology must be used in order for the laboratory equipment to communicate with the *RELATED laboratory server*. One way of doing this is to use an HIL USB board that comes with analogical and digital input/outputs and encoder-specific inputs. The developer must code the module entities needed to communicate between RELATED and this HIL board.

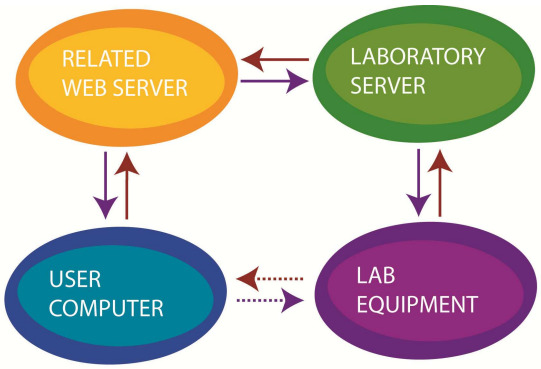

Figure 6: RLAB communication architecture

In the end, this paper will show the new communication architecture in RELATED that uses cloud technology to enhance the functionality of the RELATED Web server as well as that of the laboratory server.

4 Remote laboratories as a service

UNED is working on harnessing cloud technology to manage its technological infrastructure so that fault-tolerance, scalability, and low power consumption are achieved. Similar to [Caminero et al., 2011a], a cloud-based architecture can be implemented in order to improve on RELATED's scalability. This architecture will rely on cloud and virtualization principles to provide the efficient and scalable use of RELATED.

This architecture is presented in Figure 7 and has the following components:

- RLab component servers: one for each laboratory; provides access to the lab to which it is connected (as described in [Pastor et al., 2006])
- Database: keeps information on the labs and their available time slots
- RLab control Web server: works as a reference server; grants access to the labs based on permissions (as described in [Pastor et al., 2006])

- Load balancer: balances the incoming connections from users between the servers available at each moment; an example of a load balancer is Nginx [Nginx, 2013].
- Monitor: monitors the servers; checks several parameters, such as their central processing units (CPUs) or memory usage. An example of a monitor is Ganglia [Massie et al., 2004].
- Virtual infrastructure manager (VIM): performs the deployment of VMs that run the Web server. It adapts the infrastructure (by means of deploying VMs in a public cloud provider, such as Amazon Elastic Compute Cloud, EC2 [Amazon, 2013]) in order to meet the current workload. An example of VIM is OpenNebula [Sotomayor et al., 2009].

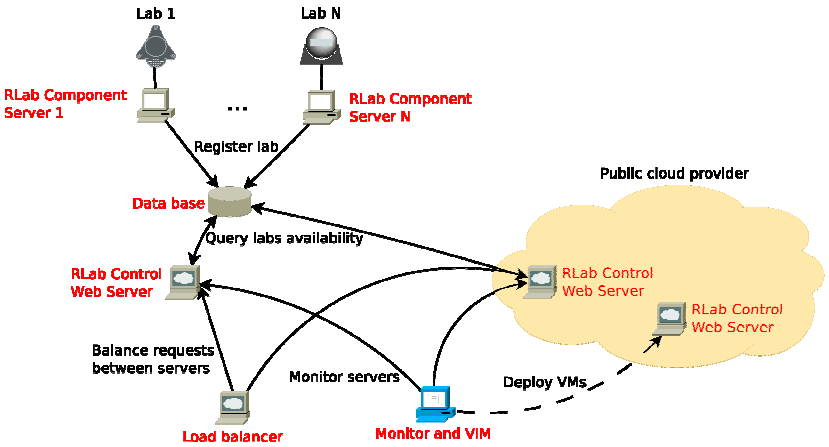

Figure 7: Proposed architecture

In order to provide scalability, a sharding architecture [Seovic et al. (2010)] can be implemented for the database, in which the database can be split into a number of databases. Each database would hold a subset of the data (the shards), where shards can be replicated to provide fault-tolerance and scalability. In addition, concerning the load balancer, load monitor, and the VIM, other machines could be set to back them up in the case of failures. Even more, data in our local premises can be de-duplicated [Dong et al., 2011] so that no data are lost in the case of local failures.

On the other hand, in order to provide efficient quality of service (QoS), a load forecasting technique could be implemented, similar to [Caminero et al., 2011a]. In this way, resources could be allocated based on the expected workload we plan to receive so that the system is adapted to it. Thanks to this, the system could be made of as few machines as possible (thus saving power), but at the same time, it could be providing efficient service to its users—thus providing green computing.

The way in which we perform the provision of resources is detailed in the next section.

5 Resource provisioning

As stated before, in order to work efficiently, the provision of resources for the RLab control Web server must be performed in advance so that users (e.g. students and faculty) do not experience congestion, which jeopardizes their work experiences. This task must be performed by the VIM, using information that comes from the resource monitor. In order to do this, a resource provisioning algorithm must be developed that adapts the RLab control Web server infrastructure to the expected workload, and this is depicted in Table 1. Although cloud providers, such as Amazon EC2, provide the automatic scaling of computing resources (e.g., more resources are devoted to an existing server), this is a coarse grain scalability based on predefined metrics. In our case, we need a resource provisioning algorithm because we want to maintain tight control over our system's infrastructures (both our internal and external infrastructures), and use metrics specifically defined to fit our needs. The development of such metrics is also among future work.

Algorithm: Resource provisioning
1: Let ProvisioningInterval = the provisioning interval 2: loop 3: {Get the predictions for the expected work load} 4: Get-WorkLoadPredictions() 5: {If there are underloaded machines, then switch off one of the machines} 6: while (NumberUnderloadedMachines()>0) do 7: ScheduleSwitchOffOneMachine() 8: {Estimate workload once machines are eventually switched off} 9: EstimateWorkload() 10: end while 11: {If there are overloaded machines, then switch on one of the off machines, or instantiate VMs in Amazon in the case that allmachines are already running} 12: while (NumberOverloadedMachines()>0) do 13: if (NumberOffMachines()>0) then 14: ScheduleSwitchOnOneMachine() 15: else 16: ScheduleUseAmazon() 17: end if 18: {Estimate workload once machines are eventually switched on} 19: EstimateWorkload() 20: end while 21: {Switch machines on/off, instantiate VMs in Amazon} 22: CommitDecisions() 23: Wait(ProvisioningInterval) 24: end loop

Table 1: Algorithm for resource provisioning

This algorithm requires the calculation of predictions on our machines' workloads, based on machine parameters, such as the number of concurrent users,

CPU, memory or network utilization, or a combination of them, such as [Caminero et al., 2011b]. The way in which the algorithm performs the provisioning of resources is detailed next. First, the predictions on the machines' workloads are obtained (line 4). Then, we check if any machines are underused (line 6), and we try to switch them off (line 7). Although the switching-off of machines is decided at this point, this decision will not be committed until the end of the algorithm (line 22) in order to avoid switching on/off the same machine multiple times during the same interval. When switching machines off, the processes that are running on them (e.g., queries from users who want to access a laboratory) finish before the machine is actually switched off. These queries are stateless queries that flush the access information to the database when they are done. Another running machine will process upcoming queries.

After checking underused machines, we must estimate the workload of each running machine when this decision is eventually committed (line 9). This is necessary in order to have a clear picture of the status of the system when these decisions are committed, so that we do not overload any machine. These steps (turning machines off and estimation of the workload) are repeated until no under-loaded machines are present (line 6).

Once this is done, the opposite case is checked—that is, the algorithm makes sure that no machines are overloaded. In the case that overloaded machines are indeed present, one of the machines that are off must be turned on (line 14). In the case that all of the machines are already running but some machines are still overloaded, VMs must be instantiated in a public cloud provider, such as Amazon EC2 [Amazon, 2013] (line 16). At first, instances of the standard small-instance type are going to be created, but work on developing efficient strategies to choose different instance types that can yield better performance is among the future work. When it is decided to complete the instantiation, we must estimate the workloads of the machines again (as we did after turning machines off) [line 19].

Recall that at this point, the system has not yet been modified (e.g., no machine has been turned on/off, and no machine has been instantiated in a public cloud provider); thus, we need an estimation of the workload of the adapted system (e.g., where local or remote machines have been turned on/off). These steps (turning machines on, instantiation in a public cloud provider, and estimation of the workload) are repeated until no overloaded machines are present (line 12). Finally, all of the decisions (e.g. switching on/off machines, instantiation of VMs in Amazon EC2) are committed in line 22. With regard to the provisioning interval, this is a customizable parameter. A reasonable value could be one hour, as that is the hiring period of machines in Amazon EC2.

6 Conclusions and future work

Thanks to cloud computing, a number of benefits can be obtained with regard to the management of computing infrastructures, such as lower power consumption and improved system utilization. This paper presents the efforts executed at UNED, the largest university of Spain, to extend a remote laboratory technology with cloud principles. This remote laboratory technology (called RELATED) has been in use in our university for several years with satisfactory results. The current paper explains

the developments being made in our university in order to extend RELATED to cloud technologies so that it can handle large workloads and minimize its power consumption. The paper includes our system's software components and the algorithm needed to perform the resource provisioning.

As for future work, we are working on a full implementation of the architecture presented in this paper. This includes the development of an appropriate metric to decide about the utilization (that is, underutilization, overutilization) of machines. This metric may be based on a combination of system parameters, such as CPU or memory consumption, or network utilization, such as [Caminero et al., 2011b].

Acknowledgements

The authors would like to acknowledge European Union Leonardo Project 142788-2008-BG-LEONARDO-LMP and Spanish Ministry of Science and Innovation for the Project TIN2008-06083-C03/TSI "s-Labs – Integración de Servicios Abiertos para Laboratorios Remotos y Virtuales Distribuidos". We also thank Erasmus Program RIPLECS—Remote-labs access in Internet-based Performance-centered Learning Environment for Curriculum Support (517836-LLP-1-2011-1-ES-ERASMUS-ESMO) and PAC—Performance-centered Adaptive Curriculum for Employment Needs (517742-LLP-1-2011-1-BG-ERASMUS-ECUE). We also thank the community of Madrid for the support of E-Madrid Network of Excellence S2009 TIC-1650.

References

[Amazon, 2013] Amazon Elastic Compute Cloud: Web page at http://aws.amazon.com/ec2/, Date of last access: 11/13/2013.

[Buyya et al., 2010] Buyya, R., Beloglazov, A., Abawajy, J. "Energy-efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges"; in Proc. of the Intl. Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), Las Vegas, USA (2010).

[Caminero et al., 2011a] Caminero, A. C., Ros, S., Hernández, R., Robles-Gómez, A., Pastor, R.: "Cloud-based E-learning Infrastructures with Load Forecasting Mechanism Based on Exponential Smoothing: A Use Case"; Proc. of the ASEE/IEEE Frontiers In Education Conference (FIE), Rapid City, USA (2011).

[Caminero et al., 2011b] Caminero, A., Rana, O., Caminero, B., Carrión, C.: "Network-aware Heuristics for Inter-domain Meta-scheduling in Grids"; JCSS (Journal of Computer and System Sciences), 77, 2 (2011), 262–281.

[Dong et al., 2011] Dong, W., Douglis, F., Li, K., Patterson, R. H., Reddy, S., Shilane, P. "Tradeoffs in Scalable Data Routing for Deduplication Clusters"; Proc. of the 9th USENIX Conference on File and Storage Technologies (FAST), , San Jose, USA (2011).

[Esquembre, 2004] Esquembre, F.: "Easy Java Simulations: A Software Tool to Create Scientific Simulations in Java"; CPC (Computer Physics Communications), 156, 2 (2004), 199–204.

[García-Peñalvo et al., 2012] García-Peñalvo, F. J., Alier, M., Lytras, M.: "Some Reflections about Service Oriented Architectures, Cloud Computing Applications, Services and Interoperability"; J.UCS (Journal for Universal Computer Science), 18, 11 (2012), 1405–1409.

[GENLOGIC, 2013] GENLOGIC: Web page at http://www.genlogic.com, Date of last access: 11/13/2013.

[Gomes and Bogosyan, 2009] Gomes L., Bogosyan, S.: "Current Trends in Remote Laboratories"; IE (Industrial Electronics, IEEE Transactions), 56 (2009), 4744–4756.

[Guimaraes et al., 2011] Guimaraes, E. G., Cardozo, E., Moraes, D. H., Coelho, P. R. "Design and Implementation Issues for Modern Remote Laboratories"; LT (Learning Technologies, IEEE Transactions), 4 (2011), 149–161.

[IBM, 2009] IBM Corporation: "The Benefits of Cloud Computing"; Tech. Rep., (2009), also available online at ftp://public.dhe.ibm.com/common/ssi/ecm/en/diw03004usen/DIW03004USEN.PDF, Date of last access: 11/13/2013.

[IBM, 2013] IBM Corporation: "Cloud Computing Saves Time, Money and Shortens Production Cycle"; Web page at http://www-01.ibm.com/software/success/cssdb.nsf/CS/ ARBN-7QK2YV? OpenDocument&Site=corp&cty=en_us, Date of last access: 11/13/2013.

[Kaplan et al., 2008] Kaplan, J. M., Forest, W., Kindler, N.: "Revolutionizing Data Center Energy Efficiency"; MacKinsey Tech. Rep., (2008).

[Massie et al., 2004] Massie, M. L., Chun, B. N., Culler, D. E.: "The Ganglia Distributed Monitoring System: Design, Implementation, and Experience"; PC (Parallel Computing), 30, 5–6 (2004), 817–840.

[Nginx, 2013] Nginx: Web page at http://wiki.nginx.org, Date of last access: 11/13/2013.

[NIST, 2013] National Institute of Standards and Technology (NIST): "Cloud Computing"; Web page at http://csrc.nist.gov/groups/SNS/cloud-computing/, Date of last access: 11/13/2013.

[Nourdine et al., 2010] Nourdine, A., Pastor, R., Vivas, G.: "Limitations of Remote Laboratories in Control Engineering Education"; IJOE (International Journal of Online Engineering), 6 (2010), 31–33.

[Nurmi et al., 2009] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D. "The Eucalyptus Open-Source Cloud-Computing System"; in Proc. 9th Int. Symp. on Cluster Computing and the Grid (CCGrid), Shangay, China (2009).

[Pastor et al., 2006] Pastor, R., Hernández, R., Ros, S., Castro, M.: "Methodological Specification of Implementation and Development of Experimental Environments"; LLTJ (Latin-American Learning Technologies Journal [RITA]), 3 (2006), 27–35.

[Pastor et al., 2012a] Pastor, R., Sánchez, D., Aliane, N., Hernández, R., Robles-Gómez, A., Caminero, A., Ros, S., Díaz, G., Castro, M.: "Practical Experiences on Building Structured Remote and Virtual Laboratories from the Student's Point of View"; Proc. of the ASEE/IEEE Frontiers In Education Conference (FIE), Seattle, USA (2012).

[Pastor et al., 2012b] Pastor, R. Sanchez, D., Aliane, N., Hernandez, R., Mariscal, G., Robles-Gomez, A., Caminero, A., Ros, S., Tawfik, M., Cristobal, E.S., Diaz, G., Castro, M.: "Structured Remote Laboratory Development"; TAEE (Technologies Applied to Electronics Teaching [TAEE]), vol., no. (2012),

[RELATED, 2013] RELATED Web Site, Web page at http://lab.scc.uned.es/related, Date of last access: 11/13/2013.

[Sancristobal et al., 2011] Sancristobal, E., Castro, M., Martin, S., Tawkif, M., Pesquera, A., Gil, R., Diaz, G, Peire, J.: "Remote Labs as Learning Services in the Educational Arena"; Global Engineering Education Conference (EDUCON), Amman , Jordan (2011).

[Seovic et al. (2010)] Seovic, A., Falco, M., Peralta, P.: "Oracle Coherence 3.5"; Packt Publishing, (2010).

[Sotomayor et al., 2009] Sotomayor, B., Montero, R. S., Llorente, I. M., Foster, I.: "Virtual Infrastructure Management in Private and Hybrid Clouds"; IEEE Internet Computing, 13, 5 (2009), 14–22.

[Vázquez et al., 2009] Vázquez, C., Huedo, E., Montero, R. S., Llorente, I. M. "Dynamic Provision of Computing Resources from Grid Infrastructures and Cloud Providers"; Proc. of the 1st Intl. Workshop on Grids, Clouds and Virtualization (WGCV), held jointly with the Grid and Pervasive Computing Conference (GPC), Geneva, Switzerland (2009).