WebLab-Deployer: exporting remote laboratories as SaaS through federation protocols

Pablo Orduña, Xabier Larrakoetxea, David Buján, Ignacio Angulo, Olga Dziabenko, Luis Rodriguez-Gil, Diego López-de-Ipiña, Javier García-Zubia
{pablo.orduna,xabier.larrakoetxea,david.bujan,ignacio.angulo,olga.dziabenko,luis.rodriguez,dipina,zubia}@deusto.es
DeustoTech – Deusto Institute of Technology, University of Deusto, Bilbao, Spain
Faculty of Engineering, University of Deusto, Bilbao, Spain

***Abstract*—During the last decade, remote laboratories have been extensively used as a primary learning tool in many universities around the world. However, today most of the remote laboratories are still only used by the same institution that provides or even develops them, or by direct partners in federated environments. There are two ways to support this type of federation: a) using a federated authentication system such as Shibboleth or b) installing a remote laboratory management system that supports federation natively. In both cases, the consumer institution must go through a process of deployment or complex configuration. This contribution explores providing access to laboratories using a Cloud Computing approach, considering the federated environments that do not have laboratories attached as a SaaS (Software as a Service) system. This approach not only makes adoption by other institutions easier, but also benefits from the existing features provided by Cloud Computing, such as elasticity to reuse the same resources for different institutions to balance the load.**

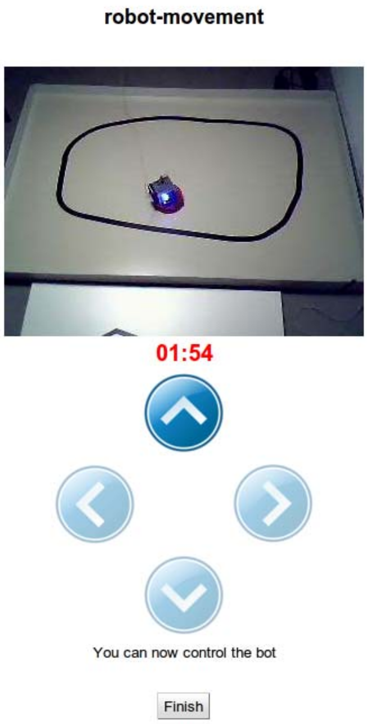
***Index Terms*—remote-laboratories, cloud-computing.**

I. Introduction

A remote laboratory is a software and hardware tool which enables students to use real equipment -located in an educational institution- through the Internet. This way, students can experiment as if they were using the laboratories with their own hands. For this reason, remote laboratories are an important learning tool for STEM (Science, Technology, Engineering, Mathematics) fields. In Figure 1, an example of remotely controlled robot is shown.

Developing a remote laboratory is a complex task that involves a multidisciplinary team. So as to ease common tasks, initiatives such as MIT iLabs[1], Labshare Sahara[2] or WebLab-Deusto[3] were created. These systems (refered as remote laboratory management systems –RLMS-) do not only provide software development toolkits for developing the remote laboratories, but also provide administration tools and common features (scheduling, user tracking, authentication, etc.) already implemented. This way, a remote laboratory developer can focus on the particular experiment itself [1]. However, even with these remote laboratory management systems, if an institution desires to build its own remote laboratory, certain design, development and maintenance efforts are required. While the particular costs have not been clearly addressed in the literature [2], it is clear that the creation of a remote (and traditional) laboratory has an economic impact.

However, a unique feature of remote laboratories when compared to traditional laboratories is that the distance of the student is not an issue, so remote laboratories can be shared with other schools or universities.



robot-movement

01:54

You can now control the bot

Figure 1 Robot experiment in WebLab-Deusto

The interest on this unique characteristic of remote laboratories -federating them to increase the types of practices and reduce costs- is growing. The Labshare project survey [3], made on all 34 Australian universities offering undergraduate engineering programs, reflects that interviewed executives were more interested in getting involved for the pedagogic merits of the remote laboratories, and that they were more inclined on initially being laboratory consumers than providers. Indeed, the European Union Commission is going to spend 60 million euro in research actions, projects and network of excellences in Technology enhanced Learning (TEL), under the objective ICT-2011.8.1 of the call FP7-ICT-2011-8. One of the target outcomes is precisely supporting a European-wide federation and use of remote laboratories

[1] http://ilab.mit.edu/wiki/
[2] http://www.labshare.edu.au
[3] http://www.weblab.deusto.es/

and virtual experimentations for learning and teaching purposes.

This concept of sharing laboratories through federation can be implemented in two different ways: deploying a RLMS that supports federation in the consumer institution or supporting cross-institutional authentication mechanisms such as Shibboleth[4]. From the management point of view, the former is richer since it can establish more clear policies for sharing the laboratory, and the consumer institution acts as an administrator in the system. However, both approaches require some infrastructure on the consumer institution. The former requires a server where the RLMS is deployed and its maintenance. The latter requires that such schemes to be available in this institutions (e.g. it might be common in universities but not in secondary schools). A third approach, not federated, would be signing up remote students in the provider institutions and be managed by the provider institution, with all the additional management involved for the provider system and the lack of control for the consumer institution.

The key concept of this contribution is that a remote laboratory management system is essentially a pure software component which, especially if it supports federation, it can be deployed anywhere else. For those institutions that are remote laboratory providers (they share or sell their own resources to other institutions), this may not have sense, since they are already assuming higher costs. But for other institutions, which are exclusively consumers (they only use remote laboratories provided by other institutions, such as secondary schools or even universities starting in the field of remote laboratories), this situation makes sense. Indeed, third parties could provide resources in their own servers to share and reuse costs.

Using cloud computing nomenclature, this means that remote laboratory management systems can be considered SaaS (Software as a Service) applications, which could be deployed in cloud environments. This contribution presents WebLab-Deployer, which is a subproject of the spanish mCloud project[5], focused on building a web-based wizard that creates new instances of the WebLab-Deusto RLMS as requested. The main feature of this system is that teachers of secondary schools and universities can register and automatically create their own, customized WebLab-Deusto hosted in servers of the University of Deusto, becoming administrators of that instance: adding users, permissions, managing policies, etc.

The rest of the paper is organized as follows: section II presents remote laboratories federations, section III discusses how cloud computing technologies can help the presented scenario, and section III describes the WebLab-Deployer solution. Finally, I presents the conclusions.

II. Federating Remote Laboratories

As previously stated, a unique characteristic of remote laboratories when compared to traditional laboratories is that the distance of the student to the real equipment is not an issue, so remote laboratories can be shared with other institutions. One institution can share a laboratory to other institution. We refer to institutions rather than universities since they do not need to be universities: research centers may be interested in sharing local resources as part of an agreement, and secondary schools would reasonably be consumers.

This sharing can be managed in a direct, simple way: the provider entity (the one where the equipment is located) creates accounts of users of the consumer entity (the one interested in using the provider university's equipment for their students). Students of the consumer entity directly access in the provider entity and the provider entity does all the work: it authenticates the user, authorizes him to use the laboratory and provides the laboratory.

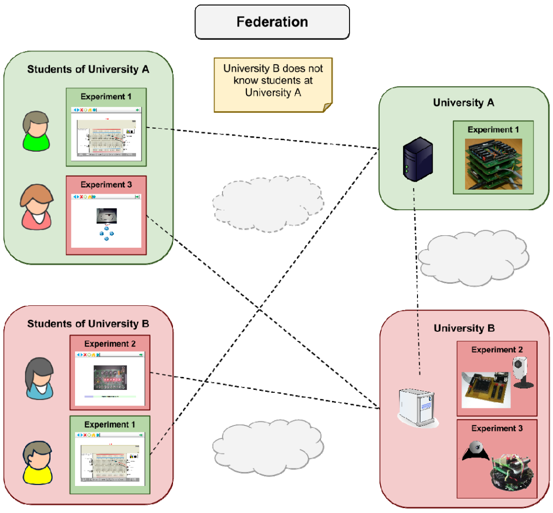


Figure 2 Example of remote laboratory federation: students of universities A and B accessing remote laboratories of each other

There are multiple problems with this solution. First, the provider entity must create and manage the user accounts of all the interested consumer universities. In a complex scenario, where a wide variety of consumers are involved -such as foreign universities and even secondary schools that simply do not speak the provider entity's language-, this approach does not scale. Second, the management of this approach is cumbersome: consumer universities would need to notify providers every change, and local databases or protocols such as LDAP would not be available. Third, the consumer entity cannot carry a proper accounting of the uses performed: it must trust the provider entity.

If both institutions come to an agreement where users of the consumer entity can access up to 10,000 times, there will be no way for the consumer entity to audit this if the provider entity at some point says “you have already reached the limit".

In order to handle these and other problems, a two-side model is required (see Figure 2), where both universities have the same software framework that manages this sharing. The consumer entity can authenticate and authorize local students, and once authorized, the local framework will contact the provider entity and request a slot. This way, the provider entity does not need to manage students and courses of the consumer entity, and the consumer entity can track all the requests performed to the provider entity, being able to track students and audit

[4] http://www.shibboleth.net
[5] http://innovation.logica.com.es/web/mcloud

the overall use. In this sense, MIT iLabs have been effectively sharing remote laboratories around the world for years [4]. Different universities can use the MIT iLabs framework to develop, maintain and share their remote laboratories with other universities. In the federation model defined by the iLabs Shared Architecture (ISA), two types of remote laboratories can be shared: batch laboratories (using queues) and interactive laboratories (using a calendar-based booking system). As shown on Figure 3, WebLab-Deusto has also adopted a federation model described in [5], which provides new features such as federated load balance and transitivity which are outside the scope of this contribution.

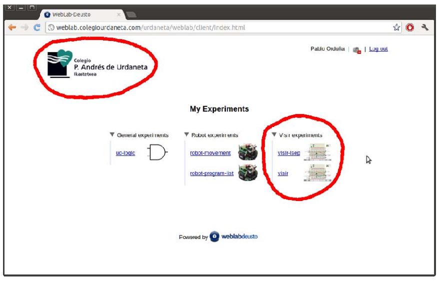


Figure 3 Secondary school Colegio Urdaneta accessing laboratories deployed in the University of Deusto through the WebLab-Deusto federation protocol

III. Cloud Computing for Remote Laboratories

This section discusses the adoption of cloud computing technologies for remote laboratories.

A. Cloud computing features

Cloud computing is a well established paradigm on which services are provided through a network and hosted by the service provider [6]. This paradigm has the potential to decrease computational costs of multiple applications by running them on a shared environment. To use an example, a provider may have 10 dedicated servers for 20 applications. If in a particular moment, 15 of them idle and there is a load peak in the others of them, they can use most of the resources, and during the rest of the time be idle using few resources.

There are different types of levels in cloud architectures, being the most relevant to this contribution SaaS (Software as a Service), and IaaS (Infrastructure as a Service).

The former refers to software applications provided in a cloud basis. An example would be wordpress.com[6], which creates instances of the popular open source blogging software Wordpress[7] for each user. These users can customize it and administrate them, changing templates, adding supported plug-ins, managing collaborators, etc. All the wordpress.com users share the same equipment resources, administrated by wordpress.com. The deployment of different instances of the same web application in the same virtual server is a recommended practice to achieve this, and this is feature is called multitenancy.

The latter refers not to software applications but to infrastructure (disk, processing and memory) provided as a service to customers. An example would be Amazon Web Services (AWS[8]), which enables customers to create their own virtual machines (administrated by the customers) on the Amazon infrastructures. This way, customers can contract 10 virtual machines that they will configure and setup their applications on it to use it. In the same way, AWS provides storage through the Amazon Simple Storage Service (Amazon S3[9]), where applications can store their information relying on the scalable amazon infrastructure. IaaS support different accounting systems for supporting infrastructure costs. For this contribution, the Amazon EC2 (Elastic Compute) is particularly interesting, since it enables customers to dynamically increase or decrease the required resources as users demand it. This way, customers select when they need to pay for more infrastructure. It is essentially equivalent as if one institution would rent 10 times more equipment for only certain overloaded moments.

SaaS and IaaS are not incompatible, and indeed SaaS providers may internally rely on own or contracted IaaS for the computational resources.

B. Remote laboratories as SaaS

As previously discussed, remote laboratory management systems (RLMSs) are pure software applications that manage remote laboratories. If a RLMS supports federation, it makes sense to deploy a RLMS where secondary schools or universities (or even just particular teachers of universities) have access as administrators and they enable certain students to access certain federated laboratories provided by other institution. If the RLMS supports multitenancy (being able to have more than one copy of the RLMS with different customizations in the same machine) it is suitable to be deployed as a SaaS application.

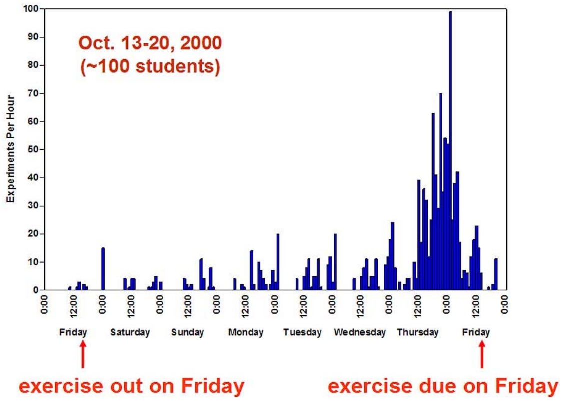


Figure 4 Load of users accessing a MIT iLab laboratory in 2000 [4]. It represents the load peaks that remote laboratories need to handle.

Additionally, remote laboratory management systems are applications that may easily benefit by elastic features of certain IaaS. Remote laboratories are not used in an equally distributed basis during the whole year. As shown

[6] http://www.wordpress.com/
[7] http://www.wordpress.org/

[8] http://aws.amazon.com/
[9] http://aws.amazon.com/s3/

in Figure 4, students tend to access the laboratories the days before exercises due, and there are not exercises during the whole year. Remote laboratory administrators know in advance when more resources would be desirable and when the available resources are enough. So using an existing IaaS to assign more resources during certain periods would be useful, particularly for stressed systems.

Depending on the architecture of the RLMS, once the stress on the RLMS is properly managed, the remote laboratories will not be affected. If one thousand students are accessing a RLMS deployed in a proper environment to support one thousand students, given that the scheduling is usually managed at RLMS level, the final machines attached to the real equipment will not have a relevant stress.

IV. WebLab-Deployer

WebLab-Deployer is a software system which generates WebLab-Deusto instances automatically. WebLab-Deusto is a remote laboratory management system which supports both federation and multitenancy.

Using WebLab-Deployer, a teacher can create its own WebLab-Deusto instance hosted remotely and administrate it: adding users, adding permissions on these users, etc. From the very beginning, as seen on the screenshot of Figure 5, the generated WebLab-Deusto is customized with the logo, links and names of the consumer institution. Given that WebLab-Deusto supports OpenID and OAuth 2.0 [1], the teacher can rely on these protocols so students are logged with the same credentials they use in their consumer system, as long as the consumer system supports these protocols. Otherwise, the authentication will be local at the RLMS system.

WebLab-Deployer has been developed using the Flask microframework [10], and its source code is available in the WebLab-Deusto github repository[11].

I. Conclusions

This contribution shows how remote laboratory management systems (RLMSs) can be considered a common SaaS application and indeed they may benefit from the elastic features of cloud computing, as long as a proper load model is established.

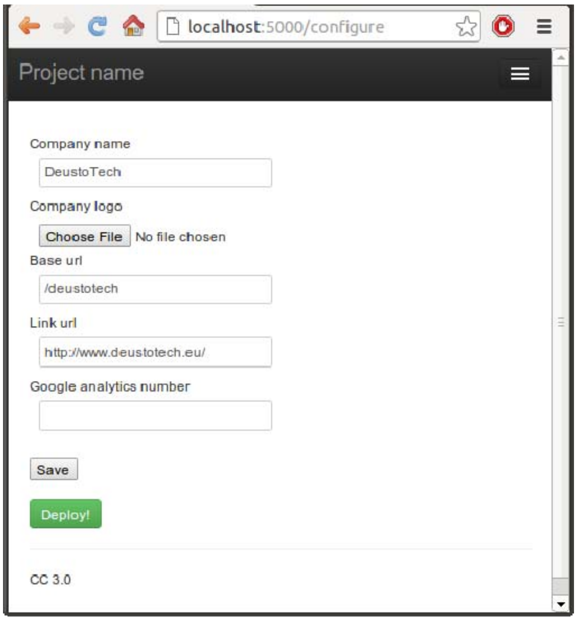
An application called WebLab-Deployer is presented as a wizard for the creation of new instances of the WebLab-Deusto RLMS. This enables that consumer-only institutions (such as secondary schools or universities not interested in developing their own remote laboratories) to minimize risks and costs. It will also enable remote laboratory newcomers to take this as a first step towards deploying their own remote laboratories.

As future work, the deployment of WebLab-Deployer on top of IaaS infrastructures (such as OpenStack[12]) is expected, as well as measurements of the impact of this tool on final users.

[10] http://flask.pocoo.org/

[11] http://github.com/porduna/weblabdeusto/

[12] http://www.openstack.org/



localhost:5000/configure

Project name

Company name

DeustoTech

Company logo

Choose File No file chosen

Base url

/deustotech

Link url

http://www.deustotech.eu/

Google analytics number

Figure 5 WebLab-Deployer user interface. By clicking on "Deploy", the system starts a new WebLab-Deusto instance with the selected logo, link, etc. which can be administrated by the user.

Acknowledgment

This work has been supported by research grants IPT-2011-1558-430000 (mCloud) funded by the Spanish Ministerio de Ciencia e Innovación.

References

[1] Pablo Orduña, Jaime Irurzun, Luis Rodriguez-Gil, Javier Garcia-Zubia, Fabricio Gazzola, Diego López-de-Ipiña. Adding New Features to New and Existing Remote Experiments through their Integration in WebLab-Deusto. International Journal of Online Engineering (iJOE) (ISSN: 1861-2121). 2011.

[2] D. Lowe, M. de la VilJefromoy, K. Jona, and L.R. Yeoh. Remote Laboratories: Uncovering the True Costs. In Remote Engineering Virtual Instrumentation 2012, REV 2012, July 2012.

[3] T. Kostulski, S. Murray. "The National Engineering Laboratory Survey – A review of the delivery of practical laboratory education in Australian undergraduate engineering programs". December 2010. Part of the outcomes of the LabShare Project: http://www.labshare.edu.au/media/img/labshare_report_panel_website.pdf

[4] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour et al.,"The iLab shared architecture: A web services infrastructure to build communities of internet accessible laboratories," Proceedings of the IEEE, vol. 96, no. 6, pp. 931–950, 2008.

[5] P. Orduña, L. Rodriguez-Gil, D. López-de-Ipiña, J. Garcia-Zubia. Sharing the remote laboratories among different institutions: a practical case. In Remote Engineering Virtual Instrumentation 2012, REV 2012, July 2012.

[6] Buyya, R., Yeo, C., Venugopal, S., Broberg, J. and Brandic, I. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems (25:6), 2009, pp. 599-616.