

Scheduling schemes among Internet Laboratories ecosystems

P. Orduña¹, J. García-Zubia²

¹ DeustoTech – Deusto Institute of Technology, University of Deusto, Bilbao, Spain

² Faculty of Engineering, University of Deusto, Bilbao, Spain

Abstract—The challenges of software architectures of Remote Labs have dramatically evolved during the latest years. The priorities of the provided features have also evolved. A key example is scheduling resources. If a rig management framework (such as MIT iLabs, LabShare Sahara or WebLab-Deusto) is provided with a single scheduling scheme, then experiments will have to be adapted to this scheduling scheme. For instance, if WebLab-Deusto only provides queues and has no support for calendar-based booking, all the experiments would have to be reserved through queues. This is an efficient approach when there is a single system, even if it is federated as in the case of MIT iLabs, where two universities have iLabs deployed and they share experiments relying on the iLabs scheduling scheme. However, efforts are being placed in supporting heterogeneous systems where different rig management frameworks can share their resources, responding to complex business models. In these ecosystems, where multiple rig management framework providers take part, the interfaces used to communicate them must support a flexible scheduling system. In this paper the different scheduling schemes used by Remote Labs in the literature are analyzed.

Index Terms— remote-laboratories, scheduling.

I. INTRODUCTION

The use of Remote Laboratories is particularly useful nowadays. Since their creation, Remote Labs enable students to remotely access experiments which are physically located in the university, so students can be at home or in traditional libraries while using experiments. The quality of the experiment can be improved by the integration of the Remote Laboratory in other digital learning tools, or enhancing the remote collaboration.

This increasing interest on Remote Laboratories has generated big efforts to share resources among different universities. The idea is that a provider university can share experiments that it is using with its students, so students of a consumer university can use them.

The MIT iLabs team¹ is pioneer in this field, having shared batch experiments among different universities since 2004 [1], and interactive experiments since 2008 [2]. The iLabs are the most popular remote laboratories platform, being used in different countries among the five continents.

In the same line, the LabShare project² in Australia is a publicly funded project focused on building a network of remote laboratories. As part of this project, the Sahara³ platform has been developed.

Efforts towards the integration of these efforts through interoperability bridges have been placed, as detailed in [3] for iLabs with LabShare Sahara. In fact, the Global Online Laboratory Consortium (GOLC⁴) was recently created to promote the development and sharing of remote laboratories.

One of the core research lines in the field of Remote Laboratories is the rig management frameworks or remote laboratory platforms, which are focused on the management of the common requirements of the remote laboratories.

Every Remote Laboratory has a set of requirements that can be reused among other laboratories. Authentication (asserting that the user is who claims to be) and authorization (asserting the permissions of the user, such as what experiments can run) are clear examples of these requirements, since it does not matter if the experiment is an electronics experiment or an experiment of physics to handle authentication and authorization.

A standalone Remote Laboratory can build all these features from scratch for that specific domain, without using a rig management framework. VISIR⁵ is an example of this: it is a domain specific -electronics- experiment, but it handles authentication, authorization, scheduling, user tracking, security, communications, etc. The problem of this approach is that it requires the experiment developers to keep and maintain all these layers of software.

As opposed to this approach, an experiment could be implemented using the tools provided by a rig management framework. With this approach, as the rig management framework publishes new versions, the experiment automatically supports new features developed by the particular framework. For instance, if developers of the rig management framework develop a new authentication scheme such as OpenID, the experiment will be accessible through OpenID. Without using a rig management framework, the experiment developers would need to implement this support for their particular experiment.

² <http://www.labshare.edu.au/project>

³ <http://labshare-sahara.sourceforge.net/>

⁴ <http://www.online-lab.org/>

⁵ <http://openlabs.bth.se/>

¹ <http://ilab.mit.edu/iLabServiceBroker/>

Scheduling could be considered a cross-laboratory requirement, since most Remote Laboratories will require or benefit from a scheduling system. Basic models of scheduling schemes in Remote Laboratories have already been studied in the literature [4]. However, the case of scheduling can be considered special, since depending on a number of factors detailed later in this paper, the scheduling scheme will be one or other. The functionality of the scheduling schemes is sometimes very domain dependent. For instance, there are experiments that can manage tens of concurrent users per second, where a calendar would not make sense, but even a queue API that requires several steps such as “start, check, stop” would add too much latency.

Additionally, when building a federation of Remote Laboratories, properly selecting the scheduling schemes is crucial. Both the provider university (the university which hosts the physical experiment) and the consumer university (the university whose student is using the experiment) should support the same schemes if there is any plan to support load balancing among different institutions. If both universities use the same rig management framework, this is not an issue. However, when building a heterogeneous ecosystem of Remote Laboratories, with different infrastructure providers, this would not be the case, and the scheduling schemes must be carefully considered.

In this paper, different scheduling schemes used in Remote Laboratories are presented.

II. CLASSIFYING SCHEDULING SCHEMES

In a first classification, the experiments can be classified into the following two categories:

- Experiments where users are interacting with the experiment while it is running.
- Experiments that do not require this interaction, letting the user submit a set of instructions that will run in the experiment and will return results that will be processed.

Most interactive experiments could be classified in the first category. The key factor is that the user will acquire a resource during a whole session. Users will send commands, files, will press buttons or write texts in a remote desktop, and will see the results in real time, without anybody else using the experiment at the same time.



Figure 1 VISIR experiment running in Facebook through WebLab-Deusto

However, some experiments that might be classified as interactive should actually be classified as the second category. A clear example is VISIR [5], developed by the

BTH in Sweden. In VISIR, students see a breadboard that can be manipulated, and the configurations are submitted to the server. In the server, the system checks that the experiment is not harmful and with a set of relays it builds the circuit in a matter of hundredths of seconds. The measurement is then performed and the result is returned, while the experiment can be used by other request. In Figure 1 the oscilloscope is shown. Since the time required by the components for providing reliable values tend to be a fraction of a second, the system can handle a number of students concurrently -in a normal setup, up to 16 concurrent users- without losing the perception of having acquired the experiment. For instance, if the student uses the oscilloscope, one request will be performed after another, so the graphic in the screen will continuously show real measurements, even if other students are at the same time using the experiment.

Batch experiments may be classified in the second category. In these experiments, the user sends an experiment setup, and the user does not need to interact with the experiment while it is running. The experiment can take a long time running the experiment, or it can take just fractions of seconds (as in VISIR), and the user will retrieve the results during the same session or in the future. The MIT Microelectronics WebLab [6] is one of the first remote experiments, and it falls in this category. The scheduling scheme behind these experiments have been a queue of users -before and after reimplementing it under the iLabs Shared Architecture-, which is the efficient approach for this category. Furthermore, no additional consideration is required in this category as long as the resources are independent (i.e. users can only request a single resource so no resource is spared while a user is in the queue), which is usually the case.

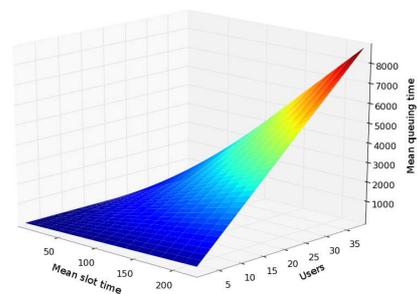


Figure 2 Factors that have an impact on the length of a queue of users

While the use of a queue is common in experiments of the second category, experiments of the first one offer a wider range of scheduling schemes. Therefore, a second classification within the first category can be established, referring to the scheduling scheme used by the experiment. Experiments in the first category may use queuing, booking, or a mixture of them.

The decision depends mainly on three factors, as stated in Figure 2:

- Number of expected concurrent users
- Number of available rigs of the experiment
- Duration of each user

If there is a single rig which uses a queue, users take an average of 1 minute per session, and 15 concurrent users

are expected, then the latest user will have to wait the queue for 15 minutes, which is acceptable. If 120 concurrent students are expected, or if the average time is 8 minutes, then the latest user will have to wait for 2 hours, which might cause a worse user experience, and a booking system will become more suitable. If 7 rigs of the experiment are added and the system can balance the load of users among the different rigs automatically, the latest user will again have to wait for only 15 minutes. On the other hand, if the number of concurrent users is low and the average of time per session is also low, a booking system becomes particularly inconvenient: users would have to state when they want to use it even if the experiment is free at that moment. Therefore, the same experiment can find more suitable one scheduling scheme or other depending on the usage of the experiment rather than on the nature of the experiment.

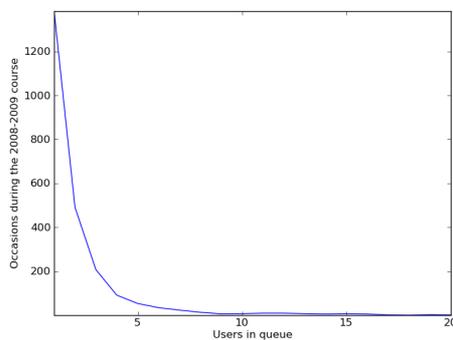


Figure 3 Users in queue in WebLab-Deusto during the course 2008-2009

It is important to remark that the number of concurrent users may differ considerably from the number of total users. The CPLD and FPGA experiments of WebLab-Deusto in the University of Deusto, had together 97 students during the course 2008-2009. Both experiments are managed with a priority queue. The students used one or the other experiment a total of 4541 times during that course. The maximum assigned time was always inferior to 4 minutes (200 seconds). However, in 2169 times (47% of the times), the student directly started using the experiment, without entering in the queue, so 47% of the times the number of concurrent students was actually 1. The distribution of the remaining is represented in Figure 3, showing the highest position in the queue achieved by each session. As the figure shows, 1383 times the user was in the position 0 of the queue (this is, there was one student using the experiment and nobody else in the queue), 491 times the position was 1208 times position 2, 92 times position 3, and the number of times descends quickly, achieving 2 times where the user entered in the position 20 of the queue. Therefore, 89.0% of the times the user had to wait less than 7 minutes (400 seconds), and 95.6% of the times the user had to wait less than 800 seconds (13 minutes and 20 seconds) to acquire the desired experiment. These numbers consider the maximum assigned time (200 seconds), but surely most of the students finished their work before that time so the queuing time was even shorter.

III. SCHEDULING SCHEMES USED IN REMOTE LABORATORIES

In this section, different algorithms used in different universities during the 2009-2010 course are presented.

A. WebLab-Deusto

WebLab-Deusto [7] is an Open Source⁶ Remote Laboratory⁷, developed by the University of Deusto, Spain. WebLab-Deusto has deployed experiments that require short sessions. This is so since students have to program electronic devices in their computer and use the Remote Laboratory to test their design. In all the supported experiments, users send these programs which are programmed in the device and then the user starts interacting with the experiment for a small amount of time to check the behavior. The user acquires the experiment for a certain time while the user interacts with the experiment.

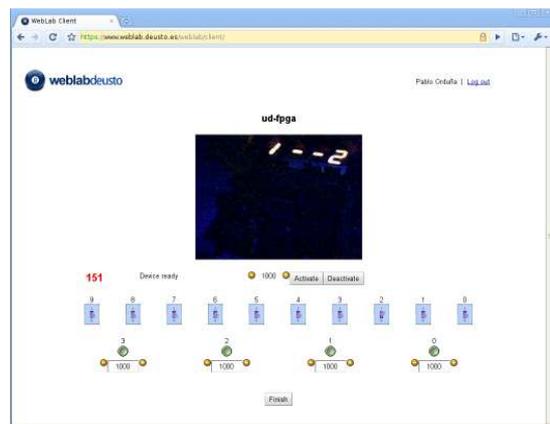


Figure 4: FPGA experiment running in WebLab-Deusto

The duration of the experiments has always been short (from 100 to 300 seconds), so the number of concurrent users tends to be low, as previously detailed. Therefore, the use of a priority queue, where the staff will go before students, is used.

B. MIT iLabs

MIT iLabs⁸ is an Open Source platform on top of which developers can build new remote experiments, and it is certainly the most spread remote laboratories system in the world.

Initially, iLabs were designed with batch experiments in mind: developed experiments would be put in a queue, and as soon as a resource is available, it will take the request and process it. A clear example is the Microelectronics WebLab, where the user logs in and when he wants to perform a request, the system reports that the submission have been submitted to a queue, and as soon as it is processed the results are presented.

Later iLabs also aimed at interactive experiments [2]. Since the time slots were particularly long (from 20

⁶ <http://code.google.com/p/weblabdeusto/>

⁷ <http://www.weblab.deusto.es/>

⁸ <http://ilab.mit.edu/wiki/>

minutes to several hours), the interactive experiments use a calendar-based booking system.

Therefore, the system supports queuing for the batch experiments and booking for the interactive experiments.

C. LabShare Sahara

Labshare⁹ is a project backend by the Australian Government's Diversity and Structural Adjustment Fund, brought by five Australian universities. The target was to build a network of Remote Laboratories at Australian level that can be shared among different universities.

The first Open Source version was released in May 2010, and supported priority queues, while in the second release in February 2011, calendar-based booking is supported.

A key feature that LabShare Sahara introduced was the ability of reusing a resource for different types of experiments. For example, an available experiment is a structural beam that can be loaded. This loaded beam can be in a horizontal or a vertical position. Depending on the exercise, the inclination does or does not matter. Therefore, if there are 2 vertical beams and 2 horizontal beams working, and all of them are being used by different students, when a new student wants to use a vertical beam, his request will be added to a queue for the vertical beams. If another student comes, requesting any beam, his request will be handled as soon as any beam is available. This feature was added to WebLab-Deusto in the release 4.0M1.

D. VISIR

As previously stated, VISIR -Virtual Instrument Systems in Reality- is an Open Source remote experiment developed by the BTH (Blekinge Institute of Technology) in Sweden, which supports concurrent users. The usual limit of concurrent users is 16, but this number may decrease if the components added to VISIR require more time in providing reliable measurements. The typical session lasts for 10 minutes approximately. During the course 2009-2010, the VISIR deployed in the BTH had 1128 sessions. The University of Deusto, which shared it with UNED, had around 80 users, which in total used the system 480 times. The FH Campus Wien had 17 users.

E. BTH Security Lab

The Blekinge Institute of Technology, authors of VISIR, provides other interesting Remote Laboratory, called BTH Security Lab [8]. In this laboratory, the user selects a set of machines that his experiment will require, performing booking requests such as "3 Microsoft Windows machines, 2 FreeBSD machines and 1 Linux system" for a certain moment. The system has a pool of 32 machines, and each machine has a modified GRUB boot loader installed. When the system is forced to be rebooted, the boot loader requests to a global controller what image to deploy. If the system has reserved a FreeBSD, the controller will reply that the loader must deploy a FreeBSD image. The boot loader downloads that image and boots it. The user can then connect to the experiment through Remote Desktop, VNC or SSH. When time is finished, the controller can restart the system by resetting the power distribution unit used by the machine. Then the

hard disk will be overwritten to make sure that there is no original data left in the next session.

During the 2009-2010 course, 1403 reservations were performed. Each reservation usually does not take longer than 3 hours. However, the system requires around 10 minutes (depending on the hard drives and on the bandwidth) for downloading the image, and around 40 minutes to fully erase the 80 GB hard disks. The time slots per machine are in an hourly basis, but one additional hour is always reserved for initialization and another for cleaning up.

This experiment is especially interesting from the scheduling point of view, since the users actually require more than one resource at the same time. If a user requests 4 machines, he will start the experiment only if there are 4 machines at the same time available. If there are 3 machines available at 2pm and 4 machines available at 3pm, then this user will have to wait until 3pm. If later another student only requires 3 machines, this user will be able to access the system at 2pm.

In the case of BTH Security Lab this is not a problem, since the users see a timetable and select among the available dates. However, if this characteristic of acquiring more than one resource at the same time appeared in an experiment based on queues, well known complex situations would arise.

F. ISILab

ISILab¹⁰, which stands for Internet Shared Instrumentation Laboratory, is an Electronics Remote Laboratory developed in the University of Genoa [9]. It is compounded by a several independent experiments which have been developed on top of ISILab.

All these experiments have the same scheduling scheme, based on queues. As in VISIR or in the user enters in a session but does not acquire the experiment. During this session, the user will submit configurations that will be queued and run in the physical experiments. These submissions usually take 500-700 milliseconds, but might achieve the 2 seconds in a single experiment.

The scheduling scheme used is therefore a queue. During the 2009-2010 course, the system was used between 800 and 1000 times by 30-40 students, using all the experiments listed above. The queue has achieved queues of less than 5 users

G. Ilmenau University of Technology

The Ilmenau University of Technology, Germany, runs a hybrid interactive laboratory [10]. The user will be able to use a real experiment. However, if the experiment is busy, students can use a virtual version that supports multiple students concurrently. In fact, students can practice with the virtual version and then check their algorithm with real world conditions in the physical experiment.

During the 2009-2010 course, it was used around 200 times. The experiment lasts 10 minutes, and it requires 30 seconds for synchronizing with the model and other 30 seconds for disposing the resources. At the moment, no

⁹ <http://www.labshare.edu.au/project>

¹⁰ <http://isilab.dibe.unige.it/>

scheduling system is used: the users can see that the experiment is busy or not and they can request using it.

H. RLab

RLab, a Remote Laboratory developed in the University of Maribor, Slovenia. The experiment consists in controlling and monitoring a real robotic arm [11].

During the 2009-2010 course, 2 robotic arms were used by 50 students, which used it in total 200 times. The average session lasts for an hour, so a booking system is used. No disposing time is required, and only 30 seconds are required to download the Java applet the first time that it is used.

I. RemotElectLab

The RemotElectLab is an electronics remote laboratory developed in the Polytechnic Institute of Porto - Higher School of Engineering, Portugal. The experiment uses a Elvis breadboard and national instruments software both in client and server side [12].

The experiment takes around 1 minute in complete all the measurements, and it has no initialization or disposing time. Since it was used by tenths of users, and a single Elvis setup (although they have planned to use more in the future), the system uses a priority queue.

J. Fachhochschule Düsseldorf University of Applied Sciences

The Dusseldorf University of Applied Sciences (Fachhochschule Düsseldorf) in Germany provide a set of industrial remote experiments [13][14].

During the 2009-2010 course, 588 users accessed these systems. Each user's session lasted for around 5 to 30 minutes, depending on the experiment, although some experiments actually have no time limit. The system currently does not support any scheduling system at the moment, but since there is a single rig of each experiment, users just request the use in case that the experiment is not busy.

K. Purdue University

The Purdue University in Indiana, United States, offers different experiments to its students under the CASPiE (Center for Authentic Science Practice in Education) effort¹¹.

In both the HPLC (High Performance Liquid Chromatography) and the Raman experiments, the user will connect to a remote application that will permit the user to monitor the experiment and gather generated data.

This data gathering may take 2.5 to 9 minutes, with an additional time to dispose the resources (1-2 minutes). However, the scheduling system is handled through a booking system managed by the lab administrator. The Lab administrator will book a slot of 3-4 hours to each class, and during this time the users will be able to use it. Therefore, the scheduling system works at group level, supporting only some students at the same time. This model is also unique compared with the rest of experiments studied, since it establishes exclusivity: the students that can use the experiment during those 3-4 hours can not use it during any other hour.

Last year, the HPLC experiment was used 784 times by 200 students, while the Raman experiment was used 17 times by 25 students.

IV. SUMMARY OF SCHEDULING SCHEMES USED

As previously detailed, the range of scheduling systems available in currently used online laboratories is wide. It is possible to identify batch and interactive experiments:

- Batch experiments (iLabs batch experiments, VISIR, ISILab)
- Interactive experiments (WebLab-Deusto, Labshare Sahara, RemotElectLab, iLabs interactive experiments, RLab, BTH Security Lab, Purdue University, Ilmenau University of Technology, Fachhochschule Düsseldorf University of Applied Sciences)

Among interactive experiments it is also possible to classify the following schemes:

- Queues and priority queues (WebLab-Deusto, Labshare Sahara, RemotElectLab)
- Booking (iLabs interactive experiments, Labshare Sahara, RLab)
- Shared resources booking (BTH Security Lab)
- Imposed group booking (Purdue University)
- No system (Ilmenau University of Technology, Fachhochschule Düsseldorf University of Applied Sciences)

This denotes the high diversity of scheduling systems actually used by Online Laboratories.

V. CONCLUSIONS AND FUTURE WORK

In this paper, authors have classified scheduling schemes used in Remote Laboratory, analyzing the state of the art through a survey to many Remote Laboratory maintainers.

The authors are working on an extensible scheduling system that supports the scheduling schemes detailed in this paper, focusing on its impact in federated remote laboratories.

REFERENCES

- [1] J. Harward, et al. "iLabs: A scalable architecture for sharing online laboratories". International Conference of Engineering Education. 2004. Gainesville, Florida. October 16-21, 2004.
- [2] J. Harward, 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, June 2008.
- [3] H. Yeung et al., "Interoperability of Remote Laboratories Systems," International Journal of Online Engineering (iJOE). Vol 6. Special issue REV2010.
- [4] A. Maiti, R. Nicole, "Time Scheduling Schemes for Online Laboratory Management Systems", Remote Engineering and Virtual Instrumentation. Stockholm, Sweden. July 2010.
- [5] I. Gustavsson, et al. "The VISIR project--an open source software initiative for distributed online laboratories." Proceedings of the REV 2007 Conference, Porto, Portugal.
- [6] Del Alamo, JA and Brooks, L. and McLean, C. and Hardison, J. and Mishuris, G. and Chang, V. and Hui, L.. (2002). World Congress on Networked Learning in a Global Environment, Berlin (Germany). 2002.

¹¹ <http://www.caspie.org/>

- [7] P. Orduña et al. "Designing Experiment Agnostic Remote Laboratories". Remote Engineering and Virtual Instrumentation. Bridgeport, CT, USA. June 2009. ISSN:1939-1382. DOI: <http://doi.ieeecomputersociety.org/10.1109/TLT.2009.51> Volume 99. 2009
- [8] Zackrisson, J., Svahnberg, C.. (2008). OpenLabs Security laboratory-The online security experiment platform. iJOE International Journal on Online Engineering. ISSN: 1861-2121. Volume 4. Number: special issue REV 2008. Pages 63-68. 2008.
- [9] Ponta, D., Scapolla, A.M., Buschiazzo, P. (2009). Survey of Remote Laboratories Using Service Oriented Architectures. International Journal of Online Engineering (iJOE). ISSN: 1861-2121. Volume 5. Issue 1. 2009.
- [10] Henke, K. and Wuttke, HD. (2006). A Mixed-Reality Environment for Digital Control Systems. International Journal of Online Engineering (iJOE). ISSN: 1861-2121. Volume 2. Issue 2. 2006.
- [11] Safaric, R., Truntic, M., Hercog, D., Pacnik, G.. (2005). Control and robotics remote laboratory for engineering education. iJOE International Journal on Online Engineering. ISSN: 1861-2121. 2005.
- [12] Sousa, N., Alves, GR., Gericota, MG.. (2009). An Integrated Reusable Remote Laboratory to Complement Electronics Teaching. IEEE Transactions on Learning Technologies.
- [13] Langmann, R.. (2005). Web-based remote control by liveconnect. International Journal of Online Engineering (iJOE). ISSN: 1861-2121. Volume 1. Issue 1. 2005.
- [14] Langmann, R.. (2004). E-learning & doing-a remote lab network for the education on the engineering. Proceedings of the TELEEC, Santiago de Cuba. 2004.

AUTHORS

P. Orduña is with DeustoTech – Deusto Institute of Technology, University of Deusto, Avda. Universidades 24, 48007 Bilbao. Spain. (e-mail: pablo.orduna@deusto.es).

J. García-Zubia, is with the Faculty of Engineering, University of Deusto, Avda. Universidades 24, 48007, Bilbao. Spain. (e-mail: zubia@deusto.es).