MostaLab: Performance Evaluation of Simultaneous Access in Analog Remote Laboratories

https://doi.org/10.3991/ijoe.v17i06.22485

Mohammed Moussa (✉), Abdelhalim Benachenhou,
Abderrahmane Boumehdi, Abderrahmane Adda-Benattia

Université Abdelhamid Ibn Badis Mostaganem, LEOG,
Mostaganem, Algeria
mohamed.moussa@univ-mosta.dz

Abstract—A service-oriented architecture for simultaneous access in the field of remote labs has been proposed and validated using stress load testing. The innovation of this work lies on the use of the parameters collected for the typical student and tested with the Artillery.io tool. Then, we have evaluated the performance of the laboratory by defining 5s the maximum waiting time that a request cannot be exceeded. This article also describes a use case showing how this architecture was designed and developed with 109 students.

Keywords—Remote laboratories, access sharing, stress load test, performance

1 Introduction

The practical work laboratories constitute an essential support for teaching and are par excellence the framework for learning an experimental approach. They allow students to see physical phenomena in action. However, practical labs require time and space for students as well as academic staff. In STEM (Science, Technology, Engineering and Mathematics) education, students can remotely control and collect data from a real physical laboratory [1]. These RLs (Remote Laboratories) are very expensive for universities with a large number of students and it is difficult to meet high demand. Salzmann et al, [2] reported that large-scale RLs are the next challenge in remote experimentation. This implies the obligation to share and pool these RLs between universities. However, the sharing and pooling of resources require the implementation of a service-oriented architecture [3] offering a very high level of abstraction and interoperability.

Two categories of managing access to RLs emerge. 1) Approach by reservation 2) Approach by queues [4].

For the first category, users will book a time slot first, and then they can control and interact with the experiment during that session. This approach is not efficient when the number of users is high.

In the second category, users have competitive access to RLs. For example, for certain experiments in analog electronics, the response time of requests does not exceed a few thousandths of second. In this type of experiment, it is possible to allow simultaneous access to the laboratory, while ensuring isolation between requests.

However, the response time of the physical lab increases with the number of concurrent users. To our knowledge, no study has evaluated the performance of the laboratory to answer the question: how many simultaneous users can use a RL without the response time being prohibitive. This article aims to discuss this problem and give some answers.

Section 2 of this article summarizes the main related works. Section 3 describes a service-oriented architecture for the design of RLs while guaranteeing exclusive access (access to a single user without conflicts). Section 4 shows through a concrete use case how to assess the maximum user capacity that can be achieved using a load testing approach.

2 Related Works

In recent years, multiple remote laboratory projects have emerged. Some of them are based on virtualization, others on the deployment of multiple instances of the same RL.

Neustock [5] has used a virtualization-based approach called Massively Scalable Online Labs (MSOL). It is based on the iLabs platform developed at Stanford in 1996 [6]. MSOL aims to create a laboratory based on transforming an existing experiment into a set of data by recording all of its possible states.

Markan et al. [7] have proposed an approach which offers great multi-user scalability thanks to a reduced-duration laboratory session in "batch mode". The advantage of this approach is that the access to RLs does not require any prior reservation.

A reservation system has been implemented in MIT's iLab project that includes additional operations and supports user reservation of platforms [8]. The EOLES (Electronics and Optics for Embedded System) project has enabled students from different countries to carry out remote experiments by sharing ten platforms, using the reservation of time slots [9].

The queue approach was adopted in earlier versions of LabShare Sahara [10-11] and weblab-deusto [12]. WebLab-Deusto does not use the reservation system. Each user requesting access to the laboratory is placed in a FIFO (First In First Out) queue.

Lowe [4], has demonstrated that the reservation / queue combination optimizes the performance of the RL management system, based on the indicator of the overall level of use (session duration) and the times of 'waiting queues.

3 MostaLab Architecture Design

Figure 1 describes the MostaLab architecture used in this study. It consists of a set of services, which can be deployed in different location. The architecture design is based on a service-oriented approach using the Lab as a service (LaaS) model introduced in [13-14-15-16]. These services are loosely coupled and offer a high level of abstraction and virtualization. The architecture also provides a simple API that hides the details of the access layer implementation to the different components of the lab. The user interface (UI) allows students to access the RL transparently from the Moodle platform [17]. The requested experiment is dynamically loaded and a relay pre-configuration step is performed. The UI allows students to select a particular state of the circuit or modify the values of a possible configuration of the experiment.

The functional architecture presented in figure 1 is based on three components: (1) The laboratory manager (2) The laboratory server (3) The user interface.

In the following section, we describe in detail the three components:

3.1 The laboratory manager

The Lab Manager is a central component of the architecture, and the entry point for requests to the various RLs. It is called by remote clients and performs the following functions:

- Management and allocation of resources for RLs: depending on the state of the resources, the service manager identifies the resources available to allocate a given experiment.
- Exploitation and monitoring of the RLs: The laboratory manager controls and monitors the RLs, by recording the status of each RL in a log file.
- Audit and logging of the activities of the various users: The Lab manager records the actions performed by remote users, sends requests to the lab server in order to view or modify the state of the laboratory.

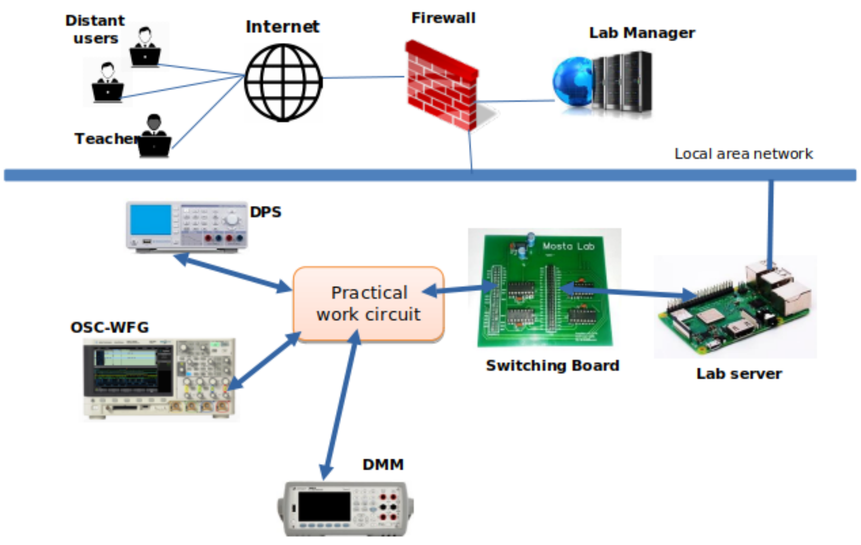


Fig. 1. The architecture of MostaLab. DMM: Digital Multimeter) - OSC-WFG Oscilloscope and WaveForm Generator- DPS: Digital Power Supply

3.2 The lab server

Lab server is a component that manages access to RL hardware. It is also responsible for sending the various requests to the instruments involved in the experiment, and will return to the Lab Manager the answers given by the various laboratory devices. In addition, the lab server implements an API for the management of the relay board and the communication with the instruments via the LXI standard. The lab server is often deployed alongside the lab manager, and can also be deployed separately from it.

3.3 The User Interface (UI)

It is a graphical interface using the HTML5 functionalities. The end user does not need to install any application other than the Internet browser. It allows students to send requests to the service manager to display or modify the status of the experiment.

4 Methodology

Figure 2 shows the two-step process that we choose to answer the research problem. In the first phase of the work, the RL has modelled, developed and then deployed for two weeks with 109 students. The results of this phase lead us to properly characterize a typical student. The second phase, called load stress test, we build a Bot with the Artillery tool. This bot is built with the parameters of the typical student. then this bot is run several to determine the maximum capacity that a single instance of RL can reach.

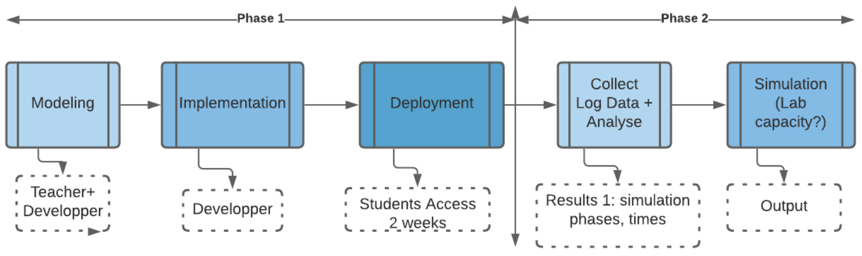
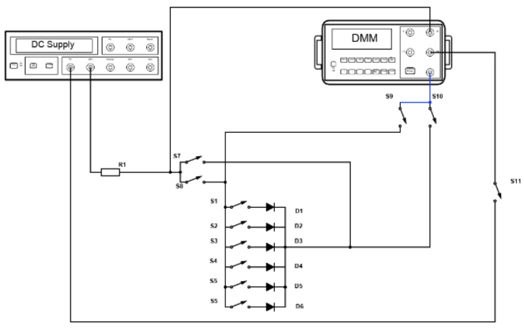
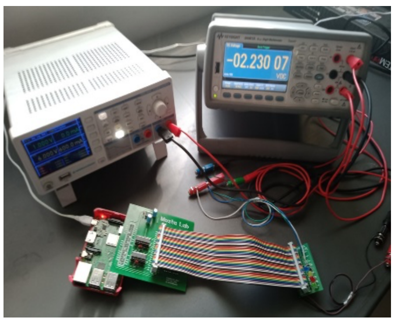


Fig. 2. The research methodology

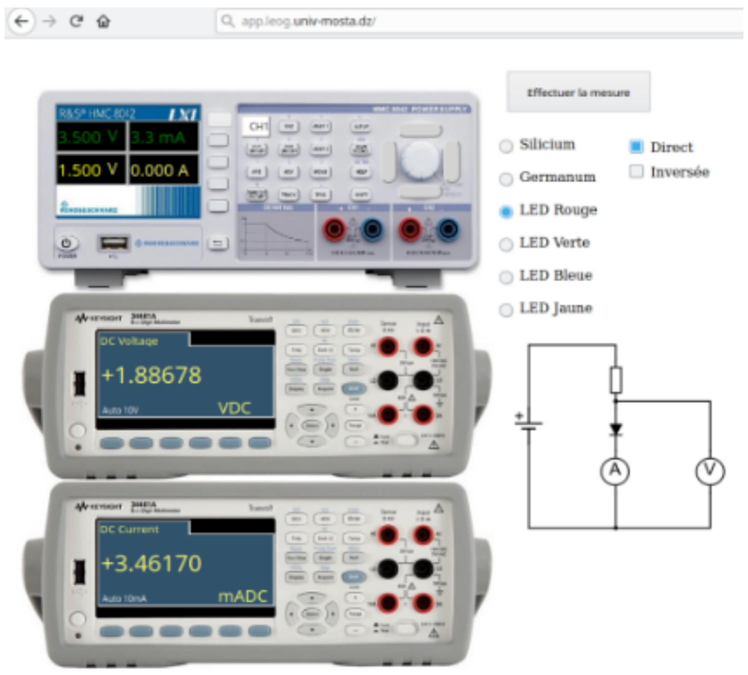
4.1 Experimental setup



a) Global circuit



b) Hardware implementation



○ Silicium
○ Germanum
● LED Rouge
○ LED Verte
○ LED Bleue
○ LED Jaune

[✓ Direct]
☐ Inversée

c) Graphical user interface

Fig. 3. Experimental setup

FIG. 3a illustrates the circuit diagram comprising the various possible combinations. It allows to select one diode among 6 (silicon diode, germanium diode, red LED, green LED, blue LED, yellow LED), as well as the direction (direct or reverse). The DMM can be used as a voltmeter or as an ammeter.

Figure 3b shows the hardware implementation comprising a lab server, a switching device and the component board.

Figure 3c illustrates the graphical interface. The end user selects a diode and the forward or reverse direction, the voltage V of the power supply and sends a request to read the voltage across the diode and the current. In the physical device, the DMM is configured as a voltmeter, measures the voltage across the diode then as an ammeter, and measures the current flowing through the diode. The results are displayed in separate virtual instruments giving the feeling of having two instruments.

4.2 The sharing model and data logging

The sequence diagram in Figure 4 illustrates the interaction between the different services of the application. The objective is twofold: 1) to orchestrate the various web services; 2) It allows RLs managers to have application load time data for system performance monitoring and to have reliable data for future performance analysis

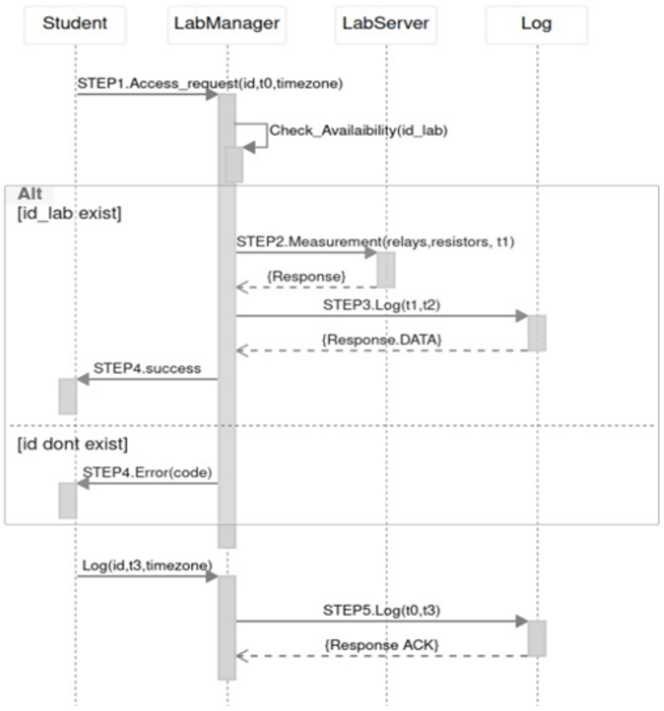


Fig. 4. The sharing sequence diagram

The five steps of the interaction between the various components are:

1. The student sends an access request to the Lab server; the request contains: id_lab, t0 (start time of the transaction), session_id and the data for the RL (relay, input voltage, component values).
2. The lab manager notes the arrival time t1 of the request, checks the user's authorization and sends the request to the lab server.
3. The lab server configures the circuit with the parameters of the request and returns the results of the measurements to the lab manager. The Lab manager notes the response time t2, calculates the time required to perform the measurement (t2-t1) and records the request parameters, the different times in a log file.
4. The lab manager returns the requested measurements to the client or an error code if the request failed.
5. Finally, the client notes the response time t3, displays the measurements in the appropriate places. It calculates the overall time of the transaction (t3-t0) and returns it to the lab manager which records it in the log file.

An important note is that in this type of flow, the client's clock and the one of the laboratory manager are not necessarily synchronized. Times t0 and t3 have the same reference, as well as t2 and t4.

4.3 Implementation

The algorithm below explains this principle based on the reconfiguration of the same instrument in several modes. The digital Multi-meter switches from Voltmeter mode to Ammeter mode to obtain measurements of the same circuit.

The exclusive access to the instruments is provided by the Lab Server. The socket mechanism is chosen because it is very suitable to prevent two requests from accessing the same instrument at the same time. The operations of configuring the relay board and the digital potentiometer can be performed by the Lab Server in parallel. To ensure the isolation of requests between users, Mutual exclusion is implemented. As the pseudo code above shows.

Pseudo_Algorithm()

```
Create_PS_Socket() // Power supply
  Execute_ parallel(Configure_circuit, Configure_Pot)
  Create_DMM_Socket()
    Read_Volt()
  Close_DMM_Socket()
  Configure_Switch_step2
  Create_DMM_Socket()
    Read_Current()
  Close_DMM_Socket()
Close_PS_Socket() //LOG time response
```

Create_PS_Socket() is a function which guarantees that at most one request at any time can access the critical section (lines 1 to 10). It is only at the end of this critical section that other queries can access the hardware setup. This avoids conflicts when two or more requests access the instruments. The critical part is responsible for several tasks (configuration of the relays in voltmeter mode, measurement of the voltage at the diode terminal, and finally the reconfiguration of the relays in ammeter mode and current measurement).

For the load test, the Artillery.io is chosen [18] for its ease of scripting in YAML (Yet Another Markup Language). The two essential elements of Artillery are "scenarios" and "phases". The scenarios are used to write the various HTTP requests to be made (Flow).

4.4 Deployment

A use case (Characterization of a diode) as shown in figure 6 was deployed and tested for 2 weeks during the 2018/2019 academic year at the Faculty of Science and Technology at the University of Mostaganem. The students were invited to take several measurements, draw the i (v) curves, write and submit the report on the Moodle platform. The lab was available 24 hours a day for a class of 109 students.

5 Result and Discussion

From the data logged in the first phase, we draw the following conclusions:

- 91% of students complete the Lab in two attempts;
- The maximum connection time to the lab for a student is two hours;
- The student performs an average of 200 requests;
- The average time between two requests is 26.6 seconds of the same student.

The times t2-t3 and t0-t4 were almost equal. This brings us back to paying less attention to network latency in the rest of the simulation.

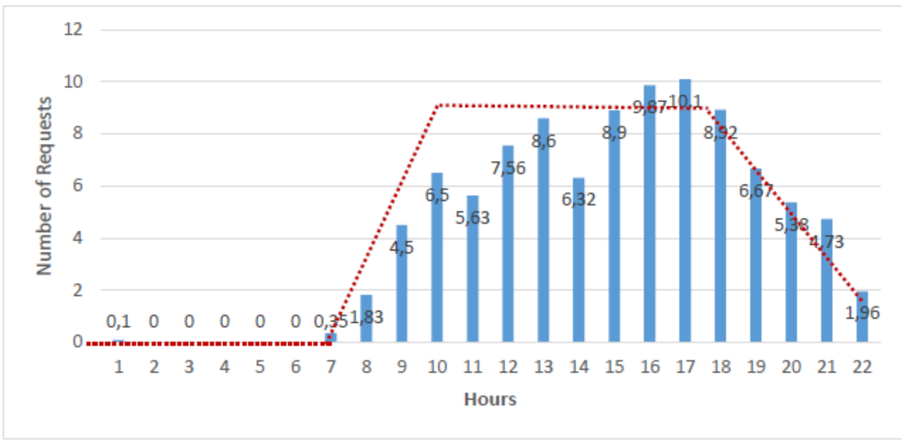
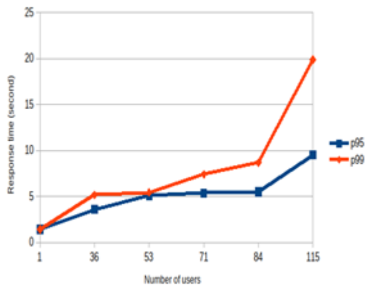


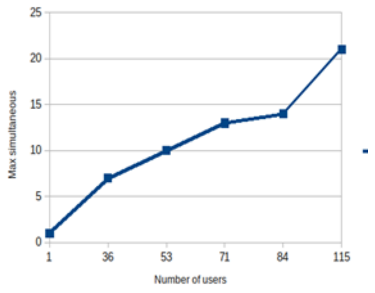
Fig. 5. Average distributions of requests by hour of day

Figure 5 illustrates the average number of requests per hour, it is calculated during the entire two-week lab period. This allows having a better approximation in three phases:

1. A first phase between 8:00 am and 10:00 am or the requests rise gradually (this phase represents 10% of the traffic)
2. A second phase between 10:00 a.m. and 6:00 p.m. where the arrival of important students (this phase represents 60% of the traffic)
3. A third and final phase between 6:00 p.m. and 12:00 p.m. where the arrival of students follows a downward curve (this phase represents 30% of the traffic)



(Left) Lab response time versus number of users' message rate.



(Right) Max simultaneous versus the number of users. Average distributions of requests by hour of day.

Fig. 6.

The lab server is configured to receive requests sent by Artillery.io a load testing toolkit. To configure the parameters of each phase, we set the number N of virtual users per day, then we calculate for each phase the different parameters (ArrivalRate, duration and rampTo) according to the real observations already seen in the previous section.

The previous step is repeated several times, increasing the number of virtual users at each step to test the maximum capacity of the lab server. Then, at the end of each load test, performance metrics are taken such as P95 and P99 (a p99 request latency value of 5000ms means 99 out of 100 requests took 5000ms or less). Finally, based on these metrics, it remains to be decided whether or not to reduce the number of virtual users.

Figures 6 and 7 clearly show that for 53 users per day, the values P90 or P95 are close to the max response time (5s) already set. Knowing that one RL is left for a week at the disposal of students: we can reach approximately 370 students per RL and per week.

Table 1. Specific Artillery.io bot load test parameters and aggregate data results

	Categories	**Attributes**	**Value**
Input data for Artillery stress bot	Phases 1	arrivalRate	0.005555555
		rampTo	0.0009523
		Duration	7200 s (2 hours)
		Mode	poisson
	Phases 2	arrivalRate	0.00095230
		rampTo	-
		Duration	25200 s (7 hours)
		Mode	poisson
	Phases 3	arrivalRate	0.00095230
		rampTo	0.00055555
		Duration	21600 s (6 hours)
		Mode	poisson
Aggregate Artillery output data	Latency	Min	1435.7 ms
		Max	8779.7 ms
		Median	1822 ms
		P95	5079.8
		P99	5439.9
	Requests	Number requests	10600

6 Conclusion

In this paper, a RL sharing model is proposed and evaluated for teaching analog electronics. This simultaneous access model makes it possible to share an RL for several students at the same time. The implementation of this solution on a use case (characterization of the diodes) shows how to reach 370 users if we set at 5 seconds the maximum response time that the request cannot exceed. Our future work will focus on improving the performance study to include more instances of RL.

7 Acknowledgement

This work is supported by the e-LIVES and ReLaTraP-AUF projects.

8 References

[1] De Jong, T., Sotiriou, S., & Gillet, D. (2014). Innovations in STEM education: The Go-Lab federation of online labs. Smart Learning Environments, 1(1), 1-16. https://doi.org/10.1186/s40561-014-0003-6

[2] Salzmann, C., Gillet, D., & Piguet, Y. (2016, February). MOOLs for MOOCs: A first edX scalable implementation. In 2016 13th international conference on remote engineering and virtual instrumentation (rev) (pp. 246-251). IEEE. https://doi.org/10.1109/rev.2016.7444473

[3] Moussa, M., Benachenhou, A., Belghit, S., Benattia, A. A., & Boumehdi, A. (2020, February). An Implementation of Microservices Based Architecture for Remote Laboratories. In International Conference on Remote Engineering and Virtual Instrumentation (pp. 154-161). Springer, Cham. https://doi.org/10.1007/978-3-030-52575-0_12

[4] Lowe, D. (2013). Integrating reservations and queuing in remote laboratory scheduling. *IEEE Transactions on Learning Technologies*, *6*(1), 73-84. https://doi.org/10.1109/tlt.2013.5

[5] Neustock L.T., Herring G.K., Hesselink L., Remote Experimentation with Massively Scalable Online Laboratories. In: Auer M., Zutin D. ed. Online Engineering & Internet of Things. Lecture Notes in Networks and Systems, vol 22. Springer, Cham, 2018, doi, Print ISBN 978-3-319-64351-9. https://doi.org/10.1007/978-3-319-64352-6_24

[6] Harward, V. J., Del Alamo, et al. The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories. Proceedings of the IEEE, 2008, 96(6), 931-950. https://doi.org/10.1109/jproc.2008.921607

[7] Markan C. M., Gupta P., Kumar G., et al. Scalable Multiuser Remote Laboratories provide on-demand hands-on laboratory experience. In: IEEE Conference on Technology and Society in Asia (T&SA). IEEE, 2012. p. 1-7. https://doi.org/10.1109/tsasia.2012.6397981

[8] http://icampus.mit.edu/projects/ilabs (accessed on 20/12/2020)

[9] Andrieu, G., Farah, S., Fredon, T., Benachenhou, A., Ankrim, M., Bouchlaghem, K., ... & Cristea, M. (2016, February). Overview of the first year of the L3-EOLES training. In 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV) (pp. 396-399). IEEE. https://doi.org/10.1109/rev.2016.7444511

[10] Lowe, D. (2014, February). Mools: Massive open online laboratories: An analysis of scale and feasibility. In 2014 11th International Conference on Remote Engineering and Virtual Instrumentation (REV) (pp. 1-6). IEEE. https://doi.org/10.1109/rev.2014.6784219

[11] Lowe, D., Machet, T., Kostulski, T., Zubía, J. G., & Alves, G. R. (2012). Uts remote labs, labshare, and the sahara architecture. Using Remote Labs in Education: Two Little Ducks in Remote Experimentation, 8, 403.

[12] Orduña, P., Irurzun, J., Rodriguez-Gil, L., Garcia-Zubia, J., Gazzola, F., & López-de-Ipiña, D. (2011). Adding New Features to New and Existing Remote Experiments Through Their Integration in WebLab-Deusto. International Journal of Online Engineering, 7. https://doi.org/10.3991/ijoe.v7is2.1774

[13] Tawfik, M., Salzmann, C., Gillet, D., Lowe, D., Saliah-Hassane, H., Sancristobal, E., & Castro, M. (2014). Laboratory as a Service (LaaS): A Novel Paradigm for Developing and

Implementing Modular Remote Laboratories. International Journal of Online Engineering, 10(4). https://doi.org/10.3991/ijoe.v10i4.3654

[14] IEEE Standard for Networked Smart Learning Objects for Online Laboratories AS IEEE 1876-2019, 2019.

[15] Zutin, D.G., Auer, M., Ordu, P., Kreiter, C., et al.: Online lab infrastructure as a service: Anew paradigm to simplify the development and deployment of online labs. In: 2016 13thInternational Conference on Remote Engineering and Virtual Instrumentation (REV). pp.208–214. IEEE (2016). https://doi.org/10.1109/rev.2016.7444467

[16] Caminero, A. C., Robles-Gomez, A., Ros, S., Tobarra, L., Hernandez, R., Pastor, R., & Castro, M. (2014). On the Creation of Customizable Laboratory Experiments: Deconstruction of Remote Laboratories to Create Laboratories as a Service (LaaS). *International Journal of Online Engineering*, *10*(6). https://doi.org/10.3991/ijoe.v10i6.3989

[17] http://www.moodle.org (accessed on 20/12/2020)

[18] https://artillery.io/ (accessed on 05/01/2021)

9 Authors

Mohammed Moussa, Abdelhalim Benachenhou, Abderrahmane Boumehdi, and **Abderrahmane Adda-Benattia** are affiliated with Université Abdelhamid Ibn Badis Mostaganem, LEOG, Mostaganem, Algeria.

Article submitted 2021-03-07. Resubmitted 2021-04-01. Final acceptance 2021-04-02. Final version published as submitted by the authors.