SSRG International Journal of Electrical and Electronics Engineering Volume 7 Issue 10, 1-11, October 2020
ISSN: 2348 – 8379 /doi:10.14445/23488379/IJEEE-V7I10P101

Shared Hardware Resources through Internet for Remote Experiments in Electronics and Electricity

Yassine Larbaoui[#1], Ahmed Naddami[#2], Ahmed Fahli[#3]

[#]*Electrical Engineering Department, University Hassan 1st, Settat, Morocco*

Abstract

This paper presents the importance of sharing hardware platforms between educational establishments for remote experimenting. In addition, it presents the principal pillars to adapt hardware resources and deploy shared hardware platforms for remote experiments through the Internet. Moreover, we present developed structures of a shared hardware platform of VISIR system, to share the exploit of our optimized resources of VISIR through the Internet; to be exploited by different establishments while using their own access platforms and their own web-user interfaces, which we propose in this paper as prepackaged auto-installers of optimized software to be used by potential exploiters.

Keywords — *Decentralized network, remote control, remote experiments, shared hardware platform.*

I. INTRODUCTION

Distance education has a significant length of history that spans over more than two centuries of expanding and developing [1]. This period represents significant changes in how learning advances from basic correspondence through postal service to the wide variety of tools available through the Internet, which support the trend of e-learning as a large important aspect of distance education.

E-learning [2] is the most used descriptor when referencing distance learning through the Internet. It often describes the efforts, the tools and the technologies that provide online access to educational materials for those who are geographically distant; and as a current result, computers, cell phones and the Internet [3] have become involved in the delivery of education as very powerful tools and frameworks. These technologies of tools and frameworks are, actually, principal basses of the delivery of varied online learning services of theoretical contents, simulation applications, virtual practices and remote experiments.

Remote laboratories [4], or remote labs as widely known, are considered as one of the most powerful online learning tools; by providing distance access to practical experiments on real physical materials. Many of these remote labs deliver considered levels of interaction and varied ranges of manipulation in different fields. These varying levels of interaction and manipulation are responding to the fundamental necessity of having practical experiments in the technical disciplines; in either educational areas or in the fields of scientific research.

The fundamental known place for practical exercising and experimenting is the hands-on laboratory, where real hardware and instruments are offered for in-place interaction with experiments. However, sometimes, many of these experiments require expensive instruments and materials, or the number of available equipment units is insufficient for all potential students, either due to the size of these materials or due to their failure and their maintenance requirements.

On the one hand, virtual laboratories, simulation laboratories and remote laboratories [4-5] can play principal roles in teaching specific areas of technical courses, such as analog electronic [6], [8], digital electronic [9-10] and electricity [7]. In addition, they can play important roles in other disciplines such as mechanic [11], radio communication [12], robotic [13-14], automation and instrumentation [15]. On the other hand, virtual laboratories and simulation laboratories can be used, principally, in certain experimental activities where simulation and virtualization may be sufficient or enough, even if they are not as effective as the laboratories where users can interactively exploit and manipulate real equipment [16]; according to the approach of learning-by-touching or the active learning approach [17].

Moreover, according to the declared statements by the Accreditation Board for Engineering and Technology (ABET): practical exercises and experiments in hands-on laboratories, as approximately in remote laboratories, should help students to achieve a set of competencies and abilities, which virtual laboratories can't satisfy if they lack the flexibility of interaction and the precision of results [18-19].

In a remote lab, the workbench, including the used instruments and equipment are not in front of the student because, instead of that, they are intended to be provided through the Internet [20]. Using this approach of online service, an experiment can be performed geographically anywhere as long as there will be an internet connection (at home, on the train, on the street, at cafe space, etc.).

This paper presents the work of developing and deploying a shared hardware platform for remote experimenting by installing, configuring and adapting the VISIR system (Virtual Instrument Systems In Reality) and separating its hardware part from its software parts in a certain manner to enable the hardware sharing through the Internet. This shared platform aims to be exploited by different webservers from other universities, institutions, e-learning platforms or other remote labs. This platform is adapted and deployed conforming to a decentralized architecture of the network, where the servers and resources of hardware and software are separated from each other. The entities and parts of this decentralized network are physically distributed while being connected only through the Internet.

VISIR system (Virtual Instrument Systems In Reality) was initiated and developed by Professor Ingvar Gustavsson with the help of other colleges and engineers; to offer to users, students principally, the possibility of the remote building of electronic and electrical circuits from the beginning by wiring discrete components.

VISIR system uses physical switching methods to control pre-established interconnections of wires between its placed-on components, and it conducts the requested measurements from any point of the inputs and outputs of these components while using different products of N.I. (National Instruments); such as by using the PXI and other N.I. Modules of measurement instruments: oscilloscope, dual-multimeter, function generator and D.C. (Direct Current) power supplier.

There are many related papers of research that present different contents of deploying [21], exploiting [22] or analyzing the VISIR system [23-24]. In addition, there are other papers that present the results of sharing specific resources of experiments between laboratories. However, this paper is distinguished because it presents the work of deploying, adapting and optimizing VISIR's resources of software to support a shared exploit of VISIR's hardware by different establishments through the Internet (Figure 1), where the software entities of VISIR are adapted to be separated from each other according to a decentralized architecture of the network.

This paper is structured as follows: Section 2 presents the main requirements to deploy shared hardware platforms for remote experiments. Section 3 presents the conducted process of deploying and sharing VISIR's hardware platform through the Internet, in addition to presenting various developed packages of software which integrate different parts of VISIR's software after adapting them; to be used by other webservers of different establishments in order to exploit our shared hardware platform of VISIR.

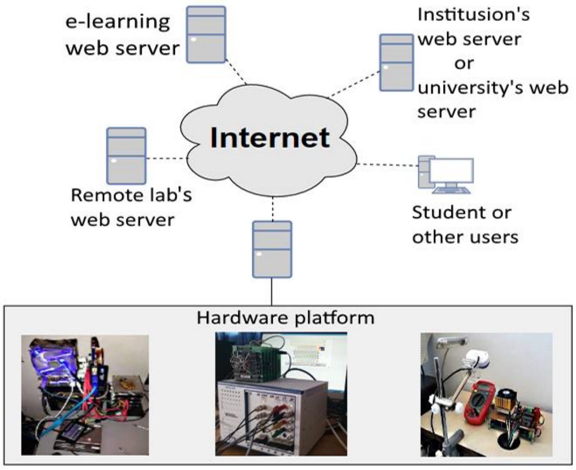

Fig. 1. Synthetized topology of a shared hardware platform for remote experiments.

Section 4 presents a conducted optimization on VISIR's equipment server by integrating a waiting line process within it (First In First Out waiting line), to receive and serve high numbers of online requests of experimenting from different distributed measurement servers through the Internet, in addition of presenting a conducted optimization on the online TCP (Transmission Control Protocol) service of VISIR's equipment server by improving its time response. Finally, section 5 for the conclusion.

II. REQUIREMENTS TO DEPLOY SHARED HARDWARE PLATFORMS FOR REMOTE EXPERIMENTS

Deploying shared hardware platforms for remote experiments will bring many advantages and opportunities in scientific fields and education (Figure 1); it will open the door to different universities and institutions around the world to have their own experiment's web services while exploiting shared resources through the Internet. In addition, deploying shared platforms for remote experiments will speed up the development process of the shared hardware and software in the form of collective contributions. Moreover, remote labs will have the opportunity to cover more technical fields of experimenting, without the need to spend more financial resources to buy the necessary hardware.

As an example of sharing the laboratory's resources, the presented work in [25], where the authors tried to elaborate and describe an open model of shared laboratories through the Internet.

There are many questions and technical aspects to consider before deploying any hardware to have a shared exploit of remote experimenting on it, such as:

- Does the hardware have interfaces to communicate with computers?
- Does the hardware have software frameworks, applications or web interfaces for its control?
- Are those software frameworks, applications or web interfaces resilient to be edited and modified?
- What are the physical limits and software barriers of the aimed hardware for online sharing?
- How flexible is the hardware for online use, and what is its supportable number of online access sessions of parallel experimenting?
- How many experiments queries on the hardware that can be handled and served simultaneously or nearly simultaneously?
- Moreover, how much is the hardware reliable, securable and adaptable to online sharing and to be exploited by a multiplicity of webservers?

In addition, there are many other technical questions, which depend on the perspective of use, the aimed fields of education, the aimed areas of scientific experimenting and the aimed hardware for online sharing.

Therefore, in this section, we analyze the radical aspects of the hardware and software parts as potential resources to be used in the design and deployment of shared hardware platforms for remote experiments. This analysis is to identify the essential technical requirements to deploy shared hardware platforms by discussing and answering the aspects of the previously mentioned questions.

A. *The Hardware Aspect of Shared Platforms*

There are many materials used in hands-on laboratories for experimenting, which diverse between electrical and electronic components, measurement instruments, power supply instruments, workbenches and multiplicity of products, which depend on the niched fields of experimenting. Some of these materials are designed to communicate with computers, or other auxiliary smart machines and devices, whereas others are structured only for hands-on use.

In the majority of hands-on experiments, there will always be some particular input parameters or input variables to control, and there will be certain output measurements to collect for their display or their visualization. Therefore, the first stand of view to consider prospective hardware for online experimenting is; what are the possible inputs and outputs of the prospective hardware to communicate with computers?

Even the materials with no support for computer use, they can be adapted by relying on exterior smart instruments or by relying on exterior devices and outside sensors; to control and manipulate the inputs and communicate the assembled results of experiments with computers. Otherwise, converting the results of experiments to other forms of measurable events to be converted and transferred to computers. Alternatively, relying on the use of cameras to visualize the hardware while showing the execution processes of experiments, or relying on cameras to visualize the displayed values of the used instruments for measurements while providing certain ways of inputs control.

There is also an elementary technical concept to consider; which is the use of physical switches to control the power supply, or to control the interconnections between the arranged components and integrated materials in binary states, or even switch or multiplex between them for more potential combinations of experiments such as what is designed for VISIR's switching system [26]. Using this mechanic of hardware switching also opens the way to put the used materials in standby modes.

A standby mode is where the experiment execution to apply the modification of the parameters and/or apply the measurements retrieving, will occur only for a slight of time under the condition of having a launched event. This event may be something like clicking on a button on an application or on a web user interface. After finishing an experiment execution process, releasing the hardware resources to be exploited by other triggered events of executions is an essential technical clew; to found the physical pillars to have multi-user services of remote experiments while exploiting the same hardware.

In this context, we are working on developing a general model of adapting and deploying shared materials through the Internet for remote experiments while distributing the resources of hardware and software through decentralized networks. This model is based on projecting the presented work in this paper on a larger architecture of network while using various resources of hardware and software.

B. The Software Aspect of Shared Platforms

After briefly treating and discussing both of the support and the breakthrough to communicate experiment hardware with computers or with other smart devices, we come to the point of analyzing the manner of using this hardware at the software level.

There are many instruments, smart devices and other products in the market provided with their own dedicated software, their own code sources or their own web-user interfaces and applications to handle their use on computers, such as N.I. (National Instruments) products and Arduino devices. However, there will be many requested characteristics and basics in those resources before buying and considering any of them for online use.

On the other hand, there are many provided smart microchips, microcontrollers and processors, which may be used to develop different materials or embedded systems; to be deployed to support certain educational experiments or scientific experiments. Those materials offer many advantages, such as having more flexible software frameworks for their programming processes, having more adaptability to communicate with other frameworks, or having the interfacing functionalities to be utilized by different programming languages.

Therefore, the first interrogation to drop at this point is, are the used resources at the software level to control and exploit the hardware parts editable for changing? To answer this interrogation mark, we need to conduct the next evaluations:

- They are determining whether the used software resources are proprietary or open resources.
- Determining how much the software resources may be considered as open resources.
- Determining the used programming languages to develop software resources,
- They are determining the resiliency of software resources to be modified.

Those evaluations are searching for the potential resiliency and flexibility of the software to be modified, to be altered or to be adapted in order to have an online exploit on the hardware.

Concerning the VISIR system, it is composed of a switching system and other N.I. Products (National Instruments), such as the PXI and N.I. Instrument modules. All materials of VISIR system rely on the use of LabVIEW platform as a software framework for their control and their support, which allows having a varied range of modification and manipulation to conduct on their manner of use and their exploit since LabVIEW resources are designed to handle different software resources and to be handled by other programming languages.

The availability of editable resources at the software level to control and use the hardware parts of an experiment opens the way to create a control entity, which to be in charge of applying the modification of the parameters and conducting the measurements. This control entity may be designed as an equipment controller limited to the tasks of answering local requests and responding to local commands of a user or responding to the requests of another software entity at the same machine. Otherwise, this controlling entity may be designed to behave as an equipment server to respond to the online requests of experimenting through the Internet.

In the first possible choice of developing an equipment controller, there will be an absolute necessity to have another software object, or entity (measurement server), to handle the online requests of experiments; by receiving and managing those requests and then forwarding them to the equipment controller entity. After executing the requested modifications of an experiment by the equipment server, the retrieved measurements will be sent to the online users through the first-mentioned entity (the measurement server), which will be responsible for receiving and managing the requests of online experiments.

This measurement server may be convenient, in charge of the authentication tasks of potential online users, to verify their credentials and/or their authorizations; to verify whether they are allowed to experiment and to be served at that time or not. It will be convenient for this measurement server to integrate a waiting line, such as by integrating a FIFO (First In First Out) waiting line, to manage all received requests of experimenting through the Internet.

Using an equipment server dedicated only to control the hardware field and to respond to the forwarded requests from a measurement server; will deliver a more responsive service of online experimenting on the hardware field to fulfil as many as possible of incoming requests. In this scenario, the measurement server will be the responsible entity for managing and serving the online requests of experiments.

Separating the functionalities of the measurement server from the equipment server where both of them function as independent entities while providing the equipment server with its own waiting line (FIFO) to handle as much as possible of forwarded requests from different distributed measurement servers through the Internet; will give more flexibility and more extended range to share the exploit of the hardware field.

As a result of separating the functionalities of the measurement server from the equipment server, many web servers of different educational establishments will have the opportunity to deploy their own independent measurement servers; to be in charge of receiving and authenticating their student's requests of experimenting on the shared field of hardware.

Separating the deployed entities of measurement servers from the equipment server will help to distribute the resources of hardware and software through a decentralized architecture of the network, which will enable us to augment the number of deployed entities of resources and extend the range of the deployed network.

In this context, the VISIR system relies on a measurement server to receive and manage the online requests of experimenting, and it relies on an equipment server to control the hardware field and to execute the measurements retrieving of requested experiments. It was complicated for us to separate the measurement server of VISIR from the equipment server (Section 3) because many of their integrated functionalities are dependent on each other.

C. The Aspect of Online Web Client Module

This part treats the concerns of developing online web-user interfaces or developing web applications; to be used through the Internet to request specific measurement servers for remote executions of experimenting. Those interfaces must offer to their end-users, students principally, the hand on manipulating the values of input fields for parameters changing, and then display or visualize the retrieved results after launching specific events, such as clicking on a one-state button to execute an experiment.

The development process of those web-user interfaces, or web applications, may generously consider the distribution ability of the final products to be integrated within different webservers.

One of the principal aims while developing those web-user interfaces and web applications should be to communicate them with certain measurement servers to exploit the deployed hardware while using software techniques of time multiplexing or time division to elevate the supportable number of potential requests of experiments.

The techniques of software switching, or software multiplexing, may play major parts to control the executed codes of experiments; to switch or multiplex between them in case of using the same components of the hardware. These techniques of switching and multiplexing are also recommended to optimize the allocated space of RAM (Random Access Memory) and optimize the exploit of the processing capacities; to handle more online incoming requests of different experiments while relying on the same resources.

An online web client module may serve a high-approximated environment of experimenting to the usual environments of hands-on laboratories; by using images and shapes of instruments, and by using similar methods of manipulation and handling such as by dragging visual objects and by wiring between the shapes of components for circuits building. Otherwise, these web client modules may offer only static user interfaces with standard fields of inputs and outputs while offering simple visualization frames, serving only an experiment content without trying to approximate its usual hands-on environment of experimenting.

Developing the web client module independently from a specific web access platform, with its own configuration to communicate with a specific measurement server, will be convenient to enable the integration of this web client module within different web services of numerous establishments. Otherwise, it will be complicated to separate the files and functionalities of this web client module from its default parent access platform.

The decisions of choosing among different client-side technologies (simple HTML, HTML5, XML, AJAX Adobe Flash, Java Applets, Microsoft Silverlight, etc.) to develop the web client module is fundamental and critical [27-28], to allow its use on different browsers and allow its exploit on various operating systems. Choosing among these technologies is also critical to enable having certain degrees of network security and interactivity, in addition to enabling the support of different multimedia resources (video, sound, 3D graphics, etc.) on the web client modules.

Depending on the used client-side technology, the server-side of the software will be oriented to interact with the client-side using specific types of protocols and software objects (Sockets, JSON, SOAP, REST, RMI, etc.).

Web applications and software resources may have different exploits of the processing capacities, either at the experiments hosting server or at the client-side; where they may have different influences in term of the number of exploited processors and the exploited space of RAM (Random Access Memory), which may augment the latency of experiments and slow down the measurements conduction.

D. The Web Access Platform Aspect

Generally, web access platforms consider the processes of identification and authentication of the potential users to verify their credentials and/or their authorizations before allowing any of them to access the supported online services. Moreover, they take into concern the security aspect of transferred data, such as login accounts and passwords, which is an essential part of providing an integrated online service; to stand against the possible attacks of spoofing and to sniff the transmitted data between the client-side and the databases of access platforms.

It is essential to take into account the use of different countermeasures and security technologies while developing any web access platform; to avoid the attacks of code injections [29] (such as SQL, LDAP, XPath, NoSQL, XML parser and SMTP header injections), and to avoid web intrusions and MITM attacks (Man In The Middle attacks) [30].

Concerning the web access platform for remote experiments, in addition to the previously mentioned points, it should offer a reservation service for scheduling or booking to manage the time allocation between the clients, where any experiment must be scheduled or booked before taking place. In addition, it is important to elaborate an account for administrative or instructive access, or elaborate a guest service, to allow having direct access to the web client module without reservation, to give away to test and validate the operability of served experiments before their official exploit by online students.

Web access platforms, as resources, also open the door to serve theoretical educational contents for the supported experiments on the deployed hardware, such as by using videos, documents, PDFs or by using other forms of digital files to cover the pedagogical tasks of experimenting.

In this context, we deploy the Open lab platform of the VISIR system and the Moodle platform [31] to support various services of e-learning and online experimenting while providing the digital files of theoretical contents for the pedagogical tasks of experimenting.

III. SHARING VISIR'S HARDWARE PLATFORM THROUGH THE INTERNET

Many universities, educational institutions, schools and remote labs around the world lack the financial resources to buy all the sufficient hardware to support various fields of experimenting. In addition, other similar establishments do not have all the necessary resources to support large numbers of experimenters. Moreover, other establishments suffer from failed or destroyed materials of experiments; caused during unsupervised manipulations or caused by an excess of incautious hands-on use. Furthermore, the majority of remote labs around the world do not have the same materials, and they do not offer the same fields of experimenting.

Deploying online shared hardware platforms for remote experiments as physical resources will offer the possibility to different educational establishments to install their own access platforms and their own web-user interfaces; to support varied fields of experiments through the Internet.

The deployment of shared hardware platforms will allow numerous benefits to having their own remote services of experimenting and their own web interfaces of experiments for their aimed courses while exploiting shared resources located in distance laboratories or other named places owned by different establishments.

There are many technical questions to answer about the deployment of shared hardware platforms for remote experiments; in order to respond to the technical requirements of the online exploit, such as:

- Pointing how much the hardware resources are flexible and resilient for online use.
- Defining what the usage limits of the hardware in the function of the supportable number of online access sessions in parallel are.
- Determining how many of experiment queries that may be handled and served simultaneously.
- Defining how much the material resources are reliable, securable and adaptable to the online sharing and to be exploited by a multiplicity of remote labs, e-learning web services or other webservers.

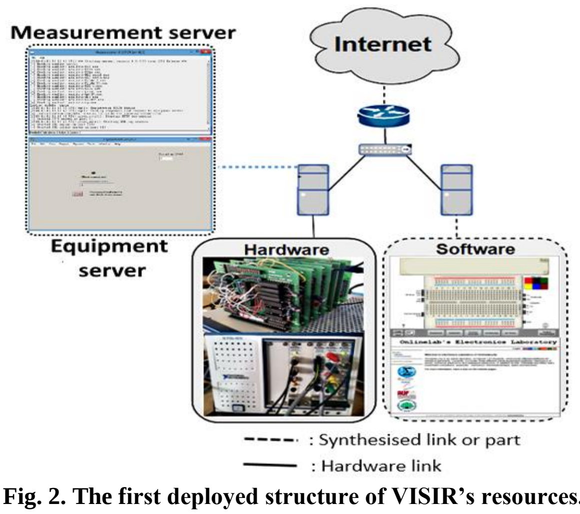

Fig. 2. The first deployed structure of VISIR's resources.

Concerning our deployed hardware platform of the VISIR system in our remote-lab, we managed successfully to install and adapt VISIR's materials on an independent hosting machine, separately from the hosting machine of the web-user interface and independently from VISIR's Open lab platform.

In the installed hardware field of the VISIR system in our remote lab, we integrate the used measurement server and equipment server to control the hardware resources according to improvised hosting structures. The first elaborated structure was by deploying and configuring the measurement server and the equipment server on an independent hosting server; to support their direct exploit through our online web services. The second elaborated structure was by deploying and configuring another measurement server on its own independent hosting machine separately from our deployed equipment server.

To separate VISIR's measurement server from the equipment server, we had to deploy an independent hosting machine for the measurement server, where we configured this hosting machine to listen to the incoming requests of online experimenting on the VISIR system and then forwarding those requests to the measurement server. In addition, we had to project the same procedure of deployment of the measurement server on VISIR's equipment server.

We managed to install VISIR's Open lab platform and VISIR's web-user interface on a separate web server, which is accessible through our online web services. This web-user interface is able to exploit our shared hardware platform of VISIR by being configured to communicate through the Internet to the adapted measurement server (Figure 2). These adapted resources of the VISIR system are dedicated to being shared with the partners of our university and to be exploited for educational purposes.

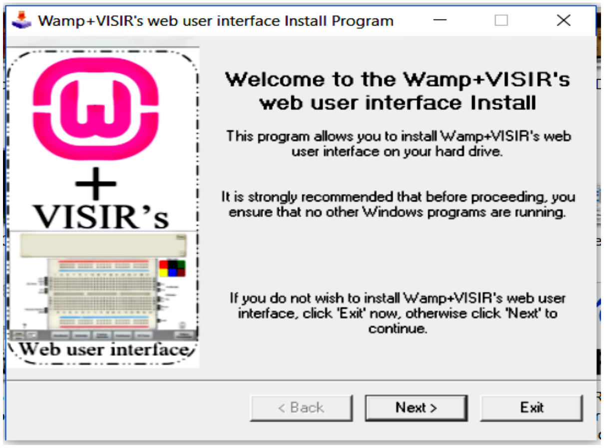

Fig. 3. Proposed auto-installer for a preconfigured web-user interface of the VISIR system, which is integrated within a WAMP server.

VISIR's measurement server is configured, by default, to support sixteen experimenting sessions using its waiting line. This number of supportable sessions is an editable variable to increment or decrement according to the requested types of experiments and their need for allocated time to be executed.

The variable of a supportable number of online sessions of experimenting on VISIR's hardware should be oriented to deliver a respected value of service quality to as many as possible of parallel served clients, which is also defined by the online availability and reliability of the hardware platform. Therefore, we modified, initially, this editable variable to twenty supportable sessions of simultaneous experimenting queries, taking advantage of the technical fact that an experiment query will not happen until an online experimenter clicks on the execution button of experimenting on VISIR's web user interface. In addition, we are taking benefit of releasing the hardware resources of used components for circuits building by the equipment server, after a short slight of time from a received request of experimenting, from microseconds to few milliseconds depending on the type and amount of measurements.

We disabled the continued mode of measurement conduction, which is used by the oscilloscope module of the VISIR system, due to its behaviour of consuming considerable time of hardware exploit where other online queries of experiment executions may occur and take place.

Our developed software of distributed auto-installers for VISIR's Open lab platform and VISIR's web-user interface are developed in two versions.

The first version of proposed auto-installers is providing an ordinary installation of VISIR's Open lab platform and VISIR's web-user interface, with all needed packages of software and with the principal configurations.

The second proposed auto-installer is an advanced copy of the first auto-installer. It is configured to give the opportunity to other remote labs of our partners to install their own web platforms of VISIR's Open lab platform for identification, authentication and reservation. In addition, this auto-installer gives our partners the opportunity to have their own installed web interfaces of the VISIR system while exploiting our deployed hardware for remote experiments through those web interfaces.

The second version of proposed auto-installers opens the door to its deplorers to exploit our installed platform of VISIR's hardware in our remote laboratory, including the switching system and the PXI product with its instrument modules, while exploiting our installed measurement server and equipment server through the Internet. This version serves our goal of sharing installed hardware platforms with other establishments since VISIR's switching system is able to serve numerous online sessions of experimenting nearly simultaneously.

Sharing the exploit of the VISIR system through our remote lab's web services is relying on using the equipment server for the hardware exploit during short slights of time, which vary from microseconds to few milliseconds. In addition, this shared exploit is relying on the use of the FIFO waiting line of the measurement server to have multi-users web-based service of experimenting; to deliver a respected level of exploit on VISIR's hardware.

We developed a third software package as a proposed version of auto-installers, which integrates only VISIR's web-user interface within a Wamp server (Figure 3). This integrated web-user interface is already preconfigured to communicate with our deployed measurement server and exploit our deployed shared hardware of the VISIR system.

By default, installing and using VISIR's web-user interface directly within any hosting platform of web serving (Wamp, Easyphp or Xample) gives the online users a hand on a web interface with additional data of I.P. address of hosting server and additional data of saved circuit structures of experiments; which, basically, online users should not have access to them. Therefore, we created a source code file by using JavaScript programming language and HTML (HyperText Markup Language) to control the form, and the visualized content of the integrated web-user interface within the proposed auto-installer in Figure 3.

The principal goal of developing the third version of proposed auto-installers is to enable direct integration of VISIR's web-user interface within any web server while using any access platform other than VISIR's Open lab; such as by using the Moodle platform [31].

In a second configured structure of VISIR's software entities, we successfully managed to adapt the equipment server as an independent entity from the measurement server; to receive the online requests of experimenting from different distributed measurement servers. This separation of the measurement server from the equipment server is important in order to deploy distributed systems of measurement servers and web-user interfaces of the VISIR system through a decentralized architecture of the network.

In addition, we developed a fourth auto-installer as an upgraded version of the third proposed auto-installer, which integrates the adapted web-user interface of VISIR along with a preconfigured version of the measurement server; to exploit the shared hardware platform by other webservers of our partners while interacting directly with our installed equipment server.

The principal goal of developing the fourth package of auto-installers is to enable the exploitation of the shared hardware platform by distributed measurement servers through a decentralized network. Therefore, other establishments will be able to have their own installed web-user interfaces and measurement servers to support their own services of remote experimenting while exploiting the deployed hardware and the deployed equipment server in our remote lab.

The used equipment server to control the shared platform of VISIR's hardware is configured, adapted and optimized to function in two structures: the first architecture structure in Figure 2 and a second architecture structure where the measurement server is on its own hosting machine independently from the equipment server of VISIR system. Therefore, the deployed shared hardware in our remote lab is enabled to be exploited by other web servers while using their own software entities; either by using VISIR's Open lab platform and the web-user interface, which are integrated within the second proposed auto-installer, or by using only the adapted web-user interface integrated within the third auto-installer, or by using the measurement server and the web-user interface integrated within the fourth auto-installer.

IV. OPTIMIZING VISIRS'S EQUIPMENT SERVER

VISIR's equipment server relies on the measurement server to receive and manage the incoming online sessions of experiments while using a First In First Out waiting line for their reception and ordering. Configuring and adapting the equipment server to run properly on its own independent hosting machine, separately from the measurement server, helps us to found the pillars to serve many distributed measurement servers to exploit one shared hardware platform of the VISIR system. However, these configuration and adaptation drop many technical questions, such as:

- How may many simultaneous requests of online experiments the equipment server handle from different distributed measurement servers?
- What is the limit where the yield of the equipment server will start decreasing while serving different measurement servers through the Internet?
- Moreover, does the equipment server use some utility or software technique as a waiting line to handle its incoming requests of experimenting?

To analyze the aspects of the previously mentioned questions, we deployed a measurement server on its own hosting machine separately from the equipment server of VISIR, and we used the Wireshark software to capture the transmitted data between the measurement server and our deployed equipment server after adapting them to function independently (Section 3). We conducted those processes of testing and analyzing to synthesize the scenario of transferred packets between the measurement server and the equipment server, which is illustrated in Figure 4.

We used the tools of Wireshark and TCP replay to generate different numbers of TCP packets, similar in form and contents to the transmitted packets from the measurement server to the equipment server, to define the online limits of VISIR's equipment server while measuring its responding behaviour and its service capacities. As a result, as shown in Figure 5, the equipment server's capacity of receiving, treating and responding to its incoming requests of experimenting decreased proportionally to the number of incoming packets, where the loss of some online packets while the equipment server was treating and serving other packets of experimenting requests.

At the conducted testing processes, we sent different numbers of [SYN] packets (synchronization

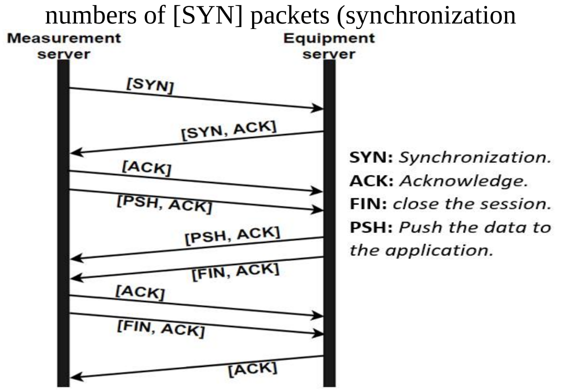

Fig. 4. TCP packets exchanging scenario between VISIR's measurement server and VISIR's equipment server.

packets) to the equipment server by using TCP replay; to generate them in different perspectives while using Wireshark to capture their accorded responses, which are shown in Figure 5. The used packets were generated nearly simultaneously by decreasing the shifting time between them to few milliseconds.

After many conducted tests on the equipment server with different time intervals of shifting between the generated packets by TCP Replay, we adopted the shifting interval of ten milliseconds to precede the analyzing tests; because it was the minimum interval of time-shifting between packets where we had an acceptable speed of packets responding from VISIR's equipment server. Moreover, we relied on using Wireshark to calculate the time response of the equipment server while conducting those tests.

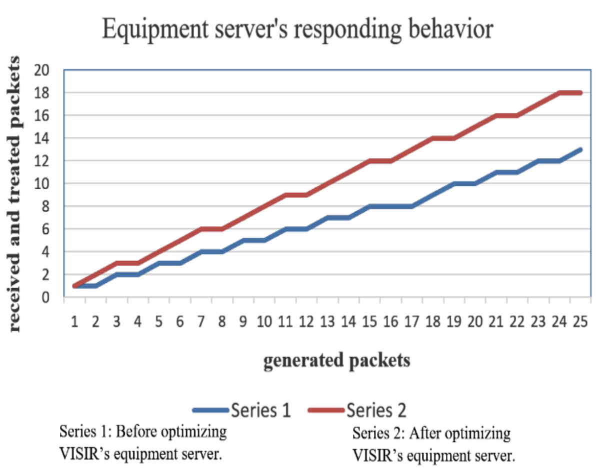

Fig. 5. Received and treated packets in the function of generated packets that were transmitted to VISIR's equipment server.

The equipment server's program, which is shown in Figure 6, relies on an array object to register and save the incoming TCP packets in order to be treated and responded to by the equipment server. This array object is the founding base of a queuing process, which is developed to function as a reception line service to manage the received packets by the equipment server. This queuing process is within a while loop object of LabVIEW platform, Figure 6 element b, separately from the execution level of the listening process of the TCP port service (Figure 6 element a). This functioning comportment is responsible for desynchronizing the queuing process from the listening process when the equipment server receives high numbers of nearly simultaneous TCP packets.

Receiving high numbers of TCP packets by the equipment server causes the desynchronization event between the queuing process and the listening process, which is responsible for the loss of some TCP packets of requested experiments while the equipment server treats other packets.

Technically, when the measurement server receives no response from the equipment server, it resends the same packet three more times, shifted in time, by using a configurable waiting interval; before concluding that the equipment server is out of service. Using this common technique at networking protocols offers more opportunity to an experimenting request to be treated and served properly; in case of having the event of nearly simultaneous requests of experimenting from different distributed measurement servers.

However, the short come of handling each packet of an experimenting request at a time by the equipment server without having a reliable waiting line, and the desynchronization between the TCP port listening process and the queuing process of incoming packets into the reception line, cause the loss of some online incoming packets without being responded. Moreover, being executed under the LabVIEW platform within its processes and threads while lacking the high priority of execution and the elevated frequency of processing cause the decreasing of the responding behaviour and the processing capacities of the equipment server of the VISIR system.

To solve the previously described limitation of packets received at the level of the equipment server of the VISIR system, we programmed a waiting line process (First In First Out waiting line) within the equipment server entity (Figure 7); to handle and manage more incoming packets of experimenting requests at the same execution level of the TCP listening process. In addition, we modified the principal program of VISIR's equipment server to elevate its execution priority and augment its processing frequency; in order to improve its online capacities by handling and

TABLE I

TCP packet responses of VISIR's equipment server before and after optimization.

Transmitted packets	[SYN[1],ACK[2]]	[PSH[3],ACK]	[FIN[4],ACK]	[ACK]	Total
Equipment server's response before optimization	0.000826 (s)	0.001883 (s)	0.000809 (s)	0.000806 (s)	0.004324 (s)
Equipment server's response after optimization	0.000403 (s)	0.002253 (s)	0.00002 (s)	0.000294 (s)	0.002970 (s)
Time difference optimization in seconds	+0.000423 (s)	-0.000370 (s)	+0.000789 (s)	+0.000512 (s)	+0.001354 (s)
Time difference optimization (%)	+51.210653 (%)	-19.649495 (%)	+99.527812 (%)	+63.523573 (%)	+31.313135 (%)

[1]*Synchronization: used packet to synchronize between the measurement server and the equipment server.*
[2]*Acknowledge: used packet to validate the reception of another type of packet.*
[3]*Push: used packet to push the data to the application.*
[4]*Finish: used packet to finish the communication scenario by closing the session.*

serving higher numbers of incoming packets.

In a second scenario of testing and analyzing, which was conducted on the optimized equipment server; we used TCP replay to generate high numbers of TCP packets, where we used the same time shifting interval of ten milliseconds while repeating the same processes of the first conducted scenario of testing and analyzing. During the second analyzing scenario, we used Wireshark to measure the conducted optimizations, which are interpreted in Figure 5 and Table 1.

The conducted optimizations on VISIR's equipment server are the result of conducting the next modifications:

- The integration of a FIFO waiting line process within the equipment server of VISIR,
- The elevation of the execution priority of the equipment server to the level of 65535, which is the highest level of execution priority.
- The elevation of the execution frequency of the equipment server to a higher stable level near 1GHz.
- The modification of the program structure of the equipment server by using a while loop conditioned by frequency and priority of execution; to have more suitable online services when receiving many packets of experimenting requests in a nearly simultaneous way.

From the results of the second conducted scenario of testing and analyzing, we deduce that the responding behaviour of the modified equipment server is improved and optimized by decreasing the number of lost incoming packets without being treated and served, as shown in Figure 5. In addition, the equipment server's time of response (calculated by using Wireshark) is decreased, as shown in Table 1, by having an average of more than 31,3% of time optimization.

Having a scenario where the equipment server receives high numbers of nearly simultaneous experimenting requests is less expected than the scenario of receiving inferior numbers of experimenting requests; however, its probability will considerably increase where there will be many distributed measurement servers to be served through a distributed architecture of the network. Therefore, modifying and improving the program structure of the equipment server and improving its functionalities help us to share the exploit of our deployed hardware platform of the VISIR system through the Internet.

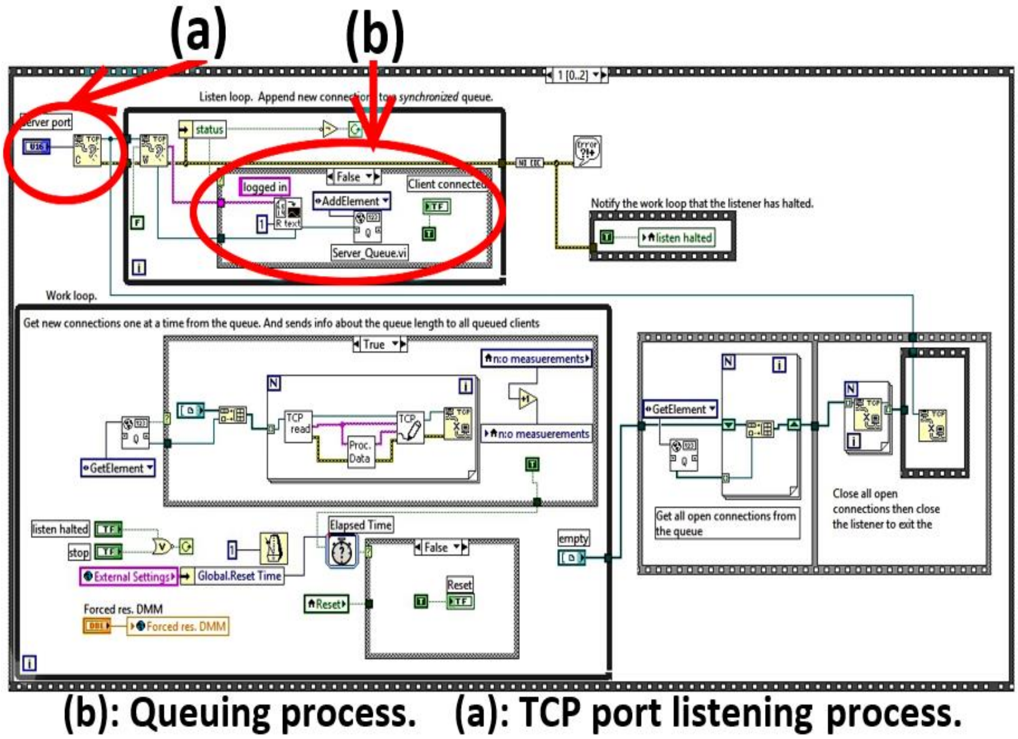

Fig 6: VISIR's equipment server program before optimization.

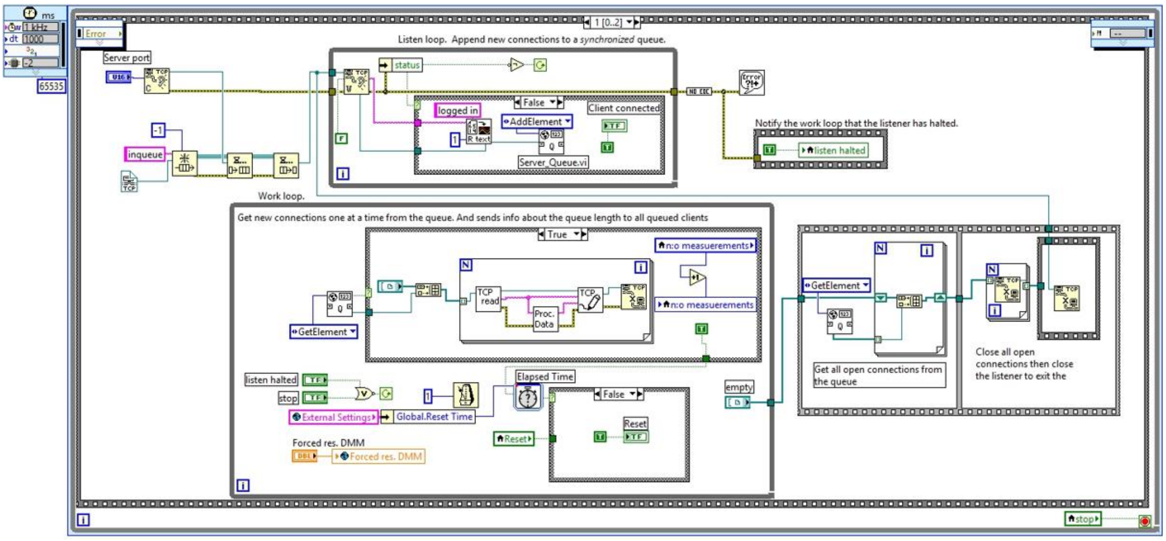

Fig 7: VISIR's equipment server program after optimization.

Our deployed platform of VISIR's hardware is adapted to be exploited by many partners of universities and institutions after deploying their own VISIR's web-user interfaces and their own measurement servers; by using our proposed packages of auto-installers, which are presented in section 3.

V. CONCLUSION

Shared hardware platforms, as physical resources, will give the opportunity to many universities and institutions around the world to deploy their own web services of online experimenting while exploiting dislocated shared hardware owned by other remote labs or different establishments.

The shared resources, which are presented in this paper, will enable their benefits to deploy their remote experimenting services by using distributed software of web-user interfaces and web access platforms, to have more simplicity of service deployment. In addition, these shared resources will facilitate and speed up the development process of the used materials and shared software in the form of collective contributions. Moreover, many remote labs will have the opportunity to cover more technical fields of experimenting without the necessity of having the financial resources to buy the needed hardware.

REFERENCES

[1] M. W. Rodrigues, L. E. Zárate, S. Isotani, "*Educational Data Mining: A review of the evaluation process in the e-learning*," Telemat. Inform. , Vol. 35, pp. 1701-1717, Sep. 2018.

[2] M. Aparicio, F. Bacao, T. Oliveira, "*Grit in the path to e-learning success*," J. Comput. Hum. Behav., Vol. 66, pp. 388-399, Jan. 2017.

[3] O. Said, Y. Albagor, "*Internet of Things-Based Free Learning System: Performance Evaluation and Communication Perspective*," IETE Journal of Research, pp. 31-44, 2016.

[4] R. Heradio, L. de la Torre, D. Galan, F. J. Cabrerizo, E. H. Viedma, S. Dormido, "*Virtual and remote labs in education: A bibliometric analysis*," Comput. Educ., Vol. 98, pp. 14-38, Jul. 2016.

[5] S. W. Director, A. W. Westerberg, M. R. Lightner, "*Simulation Procedures for Large-Scale Electronic Systems*," IETE Journal of Research, pp. 271-285, 2015.

[6] S. Rapuano, F. Zoino, "*A learning management system including laboratory experiments on measurement instrumentation*," IEEE Trans. Instrum. Meas., Vol. 55, pp. 1757–1766, Oct. 2006.

[7] J.A. Asumadu, R. Tanner, J. Fitzmaurice, M. Kelly, H. Ogunleye, J. Belter, S.C. Koh, "*A web-based electrical and electronics remote wiring and measurement laboratory (RwmLAB) instrument*," IEEE Trans. Instrum. Meas., Vol. 54, pp. 38–44, Feb. 2005.

[8] L. Benetazzo, M. Bertocco, F. Ferraris, A. Ferrero, C. Offelli, M. Parvis, V. Piuri, "*A web-based distributed virtual educational laboratory*," IEEE Trans. Instrum. Meas., vol. 49, pp. 349–356, Apr. 2000.

[9] 14M. Corrado, L. De Vitoa, H. Ramosb, J. Saliga, "*Hardware and software platform for ADCWAN remote laboratory*," J. Meas, Vol. 45, pp. 795–807 May. 2012.

[10] L.S. Indrusiak, M. Glesner, R. Reis, "*On the evolution of remote laboratories for prototyping digital electronic systems*," IEEE Trans. Ind. Electron., Vol. 54, pp. 3069–3077, Nov. 2007.

[11] J. Saenz, J. Chacon, L. la Torre, S. Dormido, "*An open software - open hardware lab of the air levitation system*," IFAC-PapersOnLine, vol. 50, pp. 9168-9173, Jul. 2017.

[12] A. Melkonyan, A. Gampe, M. Pontual, G. Huang, D. Akopian, "*Facilitating remote laboratory deployments using a relay gateway server architecture*," IEEE Trans. Ind. Electron., Vol. 61, pp. 477–485, Jan. 2014.

[13] F. Barreto, V. Benitti, "*Exploring the educational potential of robotics in schools: A systematic review*." Comput. Educ., Vol. 58, pp. 978–988, Apr. 2012.

[14] J. P. C. de Lima, L. M. Carlos, J. P. S. Simão, J. Pereira, P. M. Mafra, J. B. Silva, "*Design and implementation of a remote lab for teaching programming and robotics*," IFAC-PapersOnLine, vol. 49, pp. 86-91, 2016.

[15] W. Hu, G. P. Liu, H. Zhou, "*Web-based 3-D control laboratory for remote real-time experimentation*," IEEE Trans. Ind. Electron., Vol. 60, pp. 4673–4682, Oct. 2013.

[16] L. Gomes, S. Bogosyan, "*Current trends in remote laboratories*," IEEE Trans. Ind. Electron., Vol. 56, no. 12, pp. 4744–4756, Dec. 2009.

[17] R. Kapadia, "*Teaching and learning styles in engineering education*," Presented at the Frontiers in Education (FIE), 38th Annu. Conf., Saratoga Springs, NY, USA, Oct. 22–25, 2008.

[18] L.D. Feisel et al., "*Learning objectives for engineering education laboratories*," Presented at the Frontiers in Education (FIE) 32nd Annu. Conf., Boston, MA, USA, Nov. 6-9, 2002.

[19] M. Ghone, M. Schubert, J.R. Wagner, "*Development of a mechatronics laboratory-eliminating barriers to manufacturing instrumentation and control*," IEEE Trans. Ind. Electron., Vol. 50, pp. 394–397, Apr. 2003.

[20] M. Tawfik, E. Sancristobal, S. Martin, G. Diaz, J. Peire, M. Castro, "*Expanding the boundaries of the classroom: implementation of remote laboratories for industrial electronics disciplines*," IEEE Ind. Electron. Mag., Vol. 7, pp. 41–49, Mar. 2013.

[21] M. Tawfik, S. Monteso, F. G. Loro, P. Losada, J. A. Barba, E. Ruiz, E. Sancristobal, G. Diaz, J. Peire, M. Castro, "*Online Experiments With DC/DC Converters Using the VISIR Remote Laboratory*," IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 10, pp. 310 – 318, 2015.

[22] R. Najimaldeen, G. Ribeiro, A. Pedro, G. I. Gustavsson, "*Using uml models to describe the visir system*," International Journal of Online Engineering, Vol. 12, 2016.

[23] U. H. Jayo, J. G. Zubia, "*Remote measurement and instrumentation laboratory for training in real analog electronic experiments*," International Journal of Online Engineering, Vol. 12, pp. 123 – 134, 2016.

[24] J. G. Zubia, J. Cuadros, S. R. Yesa, U. Hernandez, P. Orduña, M. Guenaga, I. Gustavsson, "*Empirical analysis of the use of the visir remote lab in teaching analog electronics*," IEEE Transactions on Education, vol. 60, pp. 149–156, 2017.

[25] A. Bagnasco, M. Chirico, A. M. Scapolla, "*A new and open model to share laboratories on the Internet*," IEEE Trans. Instrum. Meas., vol. 54, pp. 1111–1117, Jun. 2005.

[26] P. Orduña, P. H. Bailey, K. DeLong, D. López-de-Ipiña, J. G. Zubia, "*Towards federated interoperable bridges for sharing remote educational laboratories*," Comput. Hum. Behav., vol. 30, pp. 389-395, Jan. 2014.

[27] J. G. Zubia, P. Orduña, D. López-de-Ipiña, U. Hernández and I. Trueba, "*Section III - Remote labs development issues, Remote Laboratories from the Software Engineering point of view*," in Advanceces on remote laboratories and e-learning experiences, Espain, Bilbao, Ed. Universidad de Deusto, 2007, pp. 131-150.

[28] J. G. Zubia, J. Cuadros, S. Romero, U. H. Jayo, P. Orduña, M. Guenaga, L. G. Sabate, I. Gustavsson, "*Empirical Analysis of the Use of the VISIR Remote Lab in Teaching Analog Electronics*," IEEE Trans. Educ., vol. 60, pp. 149 – 156, 2017.

[29] G. C.Oproiu, "*A T. K.Georgea, K. P. Jacobb, R. K.James, "Token-based Detection and Neural Network-based Reconstruction framework against code injection vulnerabilities*," J. Inform. Secure. Appl., Vol. 41, pp. 75-91, Aug. 2018.

[30] V. R.Mouli, K.P.Jevitha, "*Web Services Attacks and Security- A Systematic Literature Review*," Procedia. Comput. Sci. Vol. 93, pp. 870-877, 2016.

[31] G. C.Oproiu, "*a study about Using E-learning Platform (Moodle) in University Teaching Process*," Procedia. Soc. Behav. Sci., Vol. 180, pp. 426-432 May. 2015.

[32] Gokulahari.U , Padmavathy.C , Aishwarya.PL, Dhanuu Shri. R.V, "Decentralized Application" SSRG International Journal of Computer Science and Engineering 7.7 (2020): 45-50.