Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Cost-Effective Server-side Re-deployment for Web-based Online Laboratories Using NGINX Reverse Proxy

Zhongcheng Lei * **Hong Zhou** * **Shengwang Ye** * **Wenshan Hu** *
Guo-Ping Liu *,**

* *Department of Artificial Intelligence and Automation, Wuhan University, Wuhan 430072, China (e-mail:wenshan.hu@whu.edu.cn).*
** *School of Engineering, University of South Wales, Pontypridd CF37 1DL, U.K.*

Abstract: Web-based online laboratories are easy-access and non-intrusive for online experimentation, which play a key role in online learning and distance education. For web-based online laboratories, the server-side deployment is vital to provide continuous and stable experimental services which are important features of online laboratories. This paper explores the re-deployment of two versions of a web-based online laboratory, the NCSLab system, with the leverage of the NGINX reverse proxy. With proper configuration, the deployed NGINX server enables a modular backend that can integrate different backend technologies such as Tomcat, PHP and file servers together. To support secure access, only one HTTPS certificate is required to be installed on the NGINX proxy server to support the two systems. The proposed methodology provides solutions for server re-deployment without the change of previous network topology. Moreover, the re-deployment is cost-effective for enabling the re-use of the previous domain name, standard HTTP and HTTPS ports as well as the HTTPS certificate.

Keywords: Server deployment, NGINX reverse proxy, Web-based system, Online laboratory.

1. INTRODUCTION

Online laboratories which provide remote and virtual experiments are effective tools for online learning and distance education (Restivo et al. (2018); Galan et al. (2019); Gustavsson et al. (2007); Rodriguez-Gil et al. (2017)). In the last two decades, online laboratories have been widely applied to science, technology, engineering, and mathematics (STEM) education (Jong et al. (2014); De Jong et al. (2013); August et al. (2016)), especially engineering education where theoretic knowledge is obscure and practical skills are vital (Santana et al. (2013); Gomes and Bogosyan (2009); Hu et al. (2017)).

To maximize the benefit of online laboratories, a great many of online laboratories have been setup throughout the world, which provides online experimental services with online accessibility and sharing features enabling unlimited participation. Users can conduct online experiments with shared test rigs anytime and anywhere via the Internet. Therefore, online laboratories can be a complement to both online learning and conventional hands-on laboratories.

⋆ This work was supported by the IFAC Foundation Young Author's Support (YAS) Program. This work was also supported in part by the National Natural Science Foundation of China under Grants 61773144, 61873195 and 61690212, in part by the Wuhan University Experiment Technology Project Funding under Grant WHU-2018-SYJS-05.

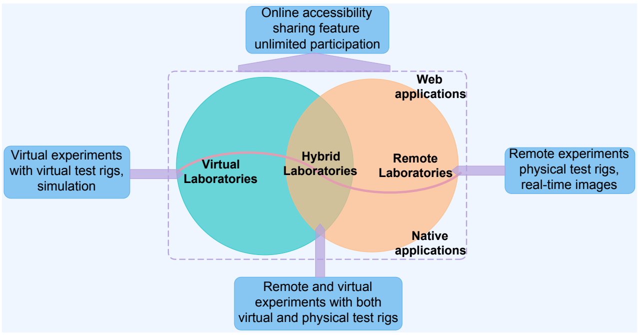

Fig. 1. Illustration of online laboratory classification

The illustration of online laboratory classification is shown in Fig. 1. In terms of experiments that online laboratories can provide, online laboratories can be classified into virtual laboratories, remote laboratories, and hybrid laboratories (Gomes and Bogosyan (2009); Lei et al. (2018b)); From the perspective of user-side applications, online laboratories can be classified into either web applications (web-based laboratories) or native applications (Hu et al. (2017)).

As web technologies advance, web-based laboratories gradually outperform native applications, for example, web-based online laboratories are more portable and less intrusive than native applications (García-Zubia et al. (2009)).

For web-based online laboratories, enabling 24/7 accessible and stable web services is vital. With the advancement of technologies, the demand for system updates and upgrades are more and more frequent, and the transition from one version to another is common. Therefore, NGINX which is open-source, reliable with high-performance features can be helpful for different purposes such as reverse proxy (Oniga et al. (2018)) and load balancing (Abdullah et al. (2019)). This paper tries to offer feasible solutions to server-side re-deployment for web-based online laboratories using NGINX reverse proxy, which takes the two versions of Networked Control System Laboratory (NCSLab) as examples.

The organization of the remainder of this paper is as follows. Section 2 discusses the motivation and contribution. Section 3 introduces the previous network of the two NCSLab systems. In Section 4, the deployment of NGINX reverse proxy is explored, which gives a brief introduction to NGINX and details the configuration process. The re-deployment result is also provided in Section 4, followed by discussion in Section 5. Finally, the conclusion is presented in Section 6.

2. MOTIVATION AND CONTRIBUTION

The NCSLab (https://www.powersim.whu.edu.cn/ncslab) has been evolved through four major versions (Hu et al. (2017)), and has been mainly providing experimental services in control engineering covering physical and three-dimensional (3-D) virtual test rigs such as the dual tank system (Hu et al. (2013)), the fan speed control system (Zhou et al. (2017); Lei et al. (2018c)) and the ball and beam system (Hu et al. (2017)) for over 13 years, the latest version of which employs HTML5 to eliminate web plug-ins and achieve 3-D interactive experimentation (Hu et al. (2017); Lei et al. (2018b)). A Massive Open Online Course (MOOC) has also created and put online on a national MOOC platform called *China University MOOC* (China-University-MOOC (2018)) in China, which was aimed for the combination with NCSLab to promote control engineering education in 2018 (Lei et al. (2019)).

However, all the front-end technologies of the previous NCSLab system were integrated based on the Yahoo! User Interface (YUI) framework which is no longer actively maintained. With the development of technologies, in late 2017, the NCSLab team started to construct the new system. To keep up to date and also provide a good user experience similar to the old system, React that is one of the state-of-the-art front-end frameworks has been adopted, which also integrates other popular open source technologies such as *three.js* and *Echarts* using a front-end and back-end separation scheme (Guan et al. (2018)). The new React-based system has been open for experimentation via https://www.powersim.whu.edu.cn/react at the 2019-2020 Autumn Semester at Wuhan University.

The motivation is to address issues brought in the design, implementation, and deployment of the new system. Issues are caused mainly by the concurrency of the two systems. The two systems can definitely be deployed at two different and irrelevant domain names. However, it requires more time, effort and money and cannot form a unique portal that is consistent with the previous system. The issues can be concluded into the following three categories:

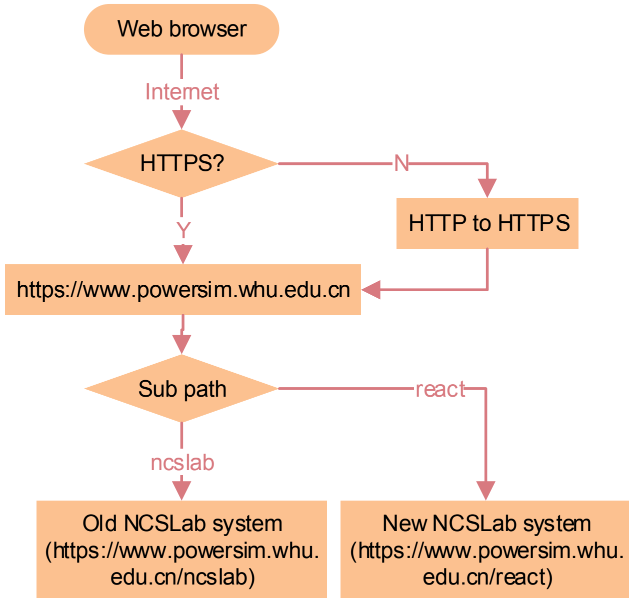

Fig. 2. Flowchart of intended access of NCSLab server

*1) **Hypertext Transfer Protocol Secure (HTTPS)***: HTTPS provides secure communication between users and the website, and also authentication of the accessed website. The maintenance of a website and HTTPS certificate requires considerable effort and financial support. For example, an HTTPS certificate costs over ¥4000 per year in China. Even for a free certificate, it could expire within a limited period of time and should be updated at least once in a year before its validity period is over. Therefore, maintaining two HTTPS certificates cost even much more effort and money.

*2) **Web source access***: The two systems serve one purpose: to provide online experimentation to end users. The development of the two systems adopts a similar naming style. Thus, the same name, for example, the name of static web resources such as images and CSS, and also the real-time laboratory for real-time experimentation, exists for resources of different systems, which caused duplicate resources from the perspective of the web.

*3) **Various backend technologies:*** The backend of the previous YUI-based NCSLab system is Tomcat, while for the new React-based system, NGINX and PHP have been adopted. The two backend technologies are completely different, which requires different software configurations.

To spare the effort for application of more domain names and port configuration, especially for port 80 and 443 which are responsible for HTTP and HTTPS, respectively, as well as the HTTPS certificate, NGINX has been used for the reverse proxy to provide continues experimental services with server-side re-deployment. The intended deployment of the server is demonstrated in Fig. 2.

The adopted NGINX reverse proxy solution is cost-effective and high efficient and can address the above-mentioned issues requiring fewer resources than the deployment of two separate systems, thus, the domain name can host both the old and new experimentation systems simultaneously.

3. PREVIOUS NETWORK TOPOLOGY OF NCSLAB

Efforts have been devoted to the construction of the new system, and the old system now mainly provides experimental services to users outside the campus, especially to MOOC users, thus, the old system is running without major updates and upgrades. For students at Wuhan University, they gradually use the new system for experimentation.

Both the YUI-based system and React-based system are developed to provide web-based interactive online experiment services without any plug-ins. There are mainly two reasons why the two systems coexist, one of which is the old system is robust and has been online for years without major issues. Also, a related MOOC has been developed based on the old system, which is currently put online for the third round (Lei et al. (2019)). The second reason is that the new system has been put online recently, which is still under construction.

3.1 YUI-based NCSLab

Fig. 3 shows the web page of YUI-based NCSLab system. Before the re-deployment, the YUI-based NCSLab system had been deployed at the local area network (LAN) 192.168.1.10 with a Tomcat server, which can be accessed via https://www.powersim.whu.edu.cn/ncslab.

There are several publications regarding the design, implementation, and application of the old system of NCSLab, covering research and education topics. These scenarios including test rig integration such as remote controlled wireless power transfer system (Lei et al. (2018a)) and virtual DC motor control system (Lei et al. (2015)), system upgrading including 2-D to 3-D (Hu et al. (2013)) and Flash to HTML5 implementation (Hu et al. (2017)), system design methodologies such as modular design (Lei et al. (2018c)) and wiring practice (Lei et al. (2018b)), and laboratory work assessing method using online laboratories (Zhou et al. (2017)).

The YUI-based NCSLab has been applied to teaching and experiment scenarios at Wuhan University since 2011. A great many of on- and off-campus users have benefited from the online experimentation system.

3.2 React-based NCSLab

The React-based NCSLab system has recently been applied to teaching and experimentation. Fig. 4 illustrates the new React-based NCSLab web page. From Fig. 3 and Fig. 4, it can be seen that the two systems share the same web structure, where test rigs are cataloged into different sub-laboratories according to their functionalities rather than their virtual/remote attribute.

For the new React-based system, before the re-deployment, it had been deployed at LAN 192.168.1.100, which can be accessed via https://control.whu.edu.cn. NGINX has already been deployed in the new React-based NCSLab to work with PHP (FastCGI). NGINX is used to work as a load balancer for PHP-FastCGI and also serve static content, while the combination of NGINX with PHP can

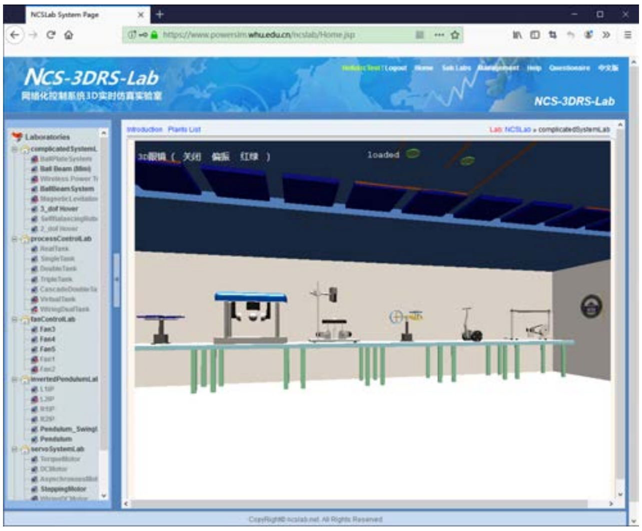

Fig. 3. Old version of NCSLab web page based on YUI framework

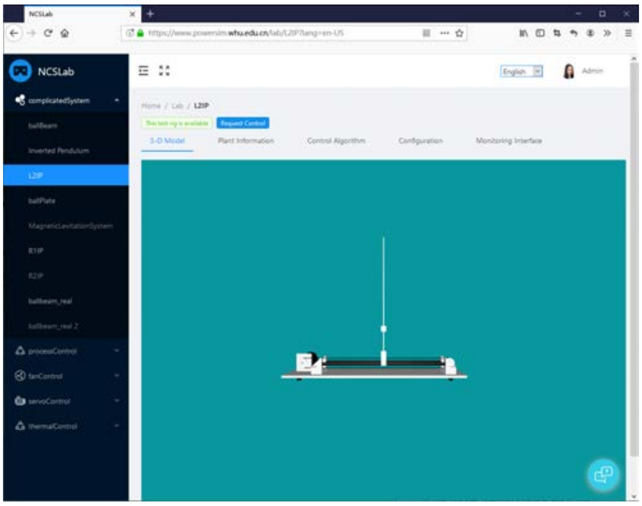

Fig. 4. New version of NCSLab web page based on React framework

be used to serve dynamic content via FastCGI. The configuration of the *upstream* block for load balancing for FastCGI is as Listing 1.

Listing 1. Load balancing configuration for React-based NCSLab using NGINX

```
upstream myfastcgi {
  server 127.0.0.1:9000 weight=1;
  server 127.0.0.1:9001 weight=1;
}
```

Currently, there are already more 20 test rigs integrated into the React-based NCSLab platform focusing on proportional-integral-derivative (PID) tuning, algorithm implementation and verification, visualized monitoring, and control. Part of the methodologies adopted in YUI-based system has been reused in the design and implementation of the new React-based system.

The two systems also have other features in common, for example, user-customizable monitoring interface and user-defined control algorithms. As for 3-D interactive features, the new React-based system needs to be enhanced as currently only 3-D animations which can demonstrate the dynamic process of virtual experimentation is available, while 3-D interactions which allow user interactions in the 3-D scene are still working in progress compared with the YUI-based system.

The YUI-based NCSLab and React-based system occupy two domain names, two HTTPS certificates, and require double effort, and the possibility of security attacks is doubled. What's more, even though the two systems are relevant, it cannot be recognized from the URL (Uniform Resource Locator), which may cause more issues, for example, future system replacement.

4. NGINX REVERSE PROXY SERVER DEPLOYMENT

The deployment of the NGINX reverse proxy server is also the process of server-side re-deployment of the two NC-SLab systems, which is designed to receive requests from users and distribute it to a certain server. After receiving server responses, the NGINX reverse proxy forwards them to the users.

4.1 NGINX Description

NGINX includes the free and open source NGINX software, as well as the commercial product from NGINX, Inc., NGINX Plus (DeJonghe (2019)). This paper is based on the open source NGINX, which is available at https://nginx.org, where the description of NGINX is given as "*NGINX is an HTTP and reverse proxy server, a mail proxy server, and a generic TCP/UDP proxy server, originally written by Igor Sysoev*". NGINX can be a powerful and high-efficient reverse proxy server (DeJonghe (2019); Soni (2016)).

4.2 NGINX Server Configuration

NGINX is cross-platform and can be deployed at any kind of OS such as Linux (either Debian, Ubuntu, RedHat or CentOS), Windows OS or macOS. Without loss of versatility, in this paper, NGINX has been deployed on the Windows system as other web servers are currently deployed on Windows OS.

Fig. 5 illustrates the flowchart of the process of NGINX reverse proxy configuration used in this paper. From Fig. 5, it can be concluded that the NGINX reverse proxy configuration process includes server hardware preparations such as OS installation and IP configuration (assigned to the LAN 192.168.1.xx), and corresponding router (the one that hosts https://www.powersim.whu.edu.cn and has Internet access) configuration that rebinds HTTP and HTTPS ports to the NGINX server, as well as the installation of NGINX software.

Following the NGINX configuration example provided by DeJonghe (2019) and *https://nginx.org*, the configuration process of NGINX configuration file *ngin.conf* is as follows, which are inside of the HTTP module.

1) Basic configuration: Basic configuration includes ports for NGINX to listen on, 80 and 443 in this case, *server_name* to configure the specific server, *index* directive to provide NGINX with a default file or list of files, the HTTPS certificate (one or more sets of certificate and key pair locations should be disclosed) and SSL (Secure Sockets Layer) status.

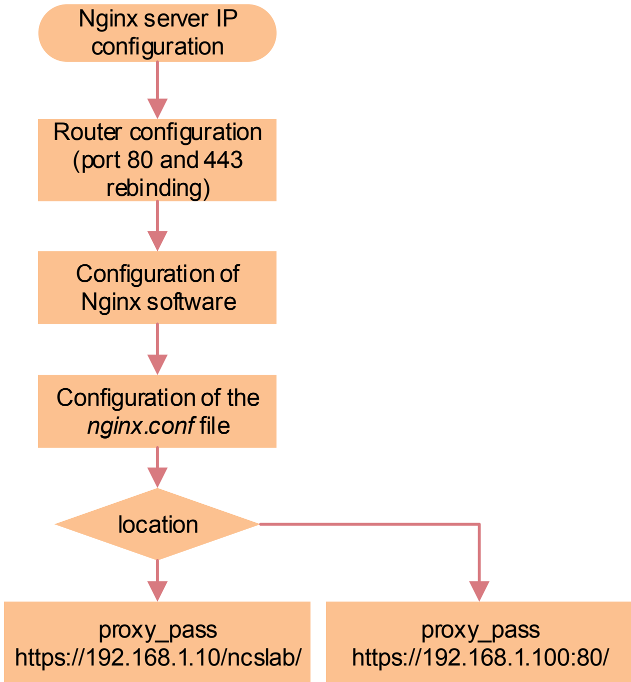

Fig. 5. Flowchart of NGINX reverse proxy configuration

2) Proxy location block: NGINX will best match the URI (Uniform Resource Identifier) (DeJonghe (2019)) requested to a *location* block. The *ncslab* location block configuration for YUI-based system which is based on a Tomcat server is shown in Listing 2, in which line 2 directs the *ncslab* request to the server 192.168.1.10 that hosts the YUI-based system. The *proxy_http_version* directive in line 3 instructs the proxy module to use HTTP version 1.1. Line 4 to 7 of Listing 2 are crucial for the Tomcat server. If it is not well configured, part of the system resources cannot be accessed.

Listing 2. NGINX proxy configuration for *ncslab* location with a Tomcat server

```
1 location ^~ /ncslab/ {
2   proxy_pass https://192.168.1.10/ncslab/;
3   proxy_http_version 1.1;
4   proxy_set_header Host $http_host;
5   proxy_set_header X-Real-IP $remote_addr;
6   proxy_set_header X-Forwarded-For
      $proxy_add_x_forwarded_for;
7   proxy_set_header X-Forwarded-Proto $scheme;
8 }
```

Listing 3 shows the details on NGINX proxy configuration for *react* location with the NGINX and PHP (FastCGI) server. It is the most common proxy location configuration type which only contains the *proxy_pass* directive. NGINX will direct the *react* requests to the server 192.168.1.100, where the React-based new system is hosted.

Listing 3. NGINX proxy configuration for *react* location with NGINX and PHP(fastCGI) server

```
location ^~ /react/ {
  proxy_pass http://192.168.1.100:80/;
  proxy_http_version 1.1;
}
```

3) Connection with WebSocket protocol: Listing 2 and 3 are about the HTTP protocol. In some cases, WebSocket connection is needed, which requires the upgrade of the HTTP protocol. Listing 4 shows the configuration for *rtlab* block which is responsible for real-time experimentation. *proxy_set_header* directive is employed to switch a protocol to WebSocket via "Upgrade" and "Connection".

Listing 4. NGINX proxy configuration for *rtlab* location

```
location ^~ /rtlab/ {
  proxy_pass http://192.168.1.100:80/rtlab/;
  proxy_http_version 1.1;
  proxy_set_header Upgrade $http_upgrade;
  proxy_set_header Connection "Upgrade";
}
```

4) HTTPS configuration: Listing 5 sets up two *server* blocks: one for HTTP on port 80 and the other for HTTPS on port 443. Line 2 and 7 instruct NGINX to listen on port 80 and 443, respectively, the latter of which is encrypted with SSL and the *ssl* parameter is enabled. To ensure secure communication, a *rewrite* or *return* statement should be used to redirect unencrypted HTTP requests to HTTPS, which is shown in line 4 of Listing 5.

Listing 5. HTTP and HTTPS configuration

```
1  server {
2    listen 80;
3    server_name www.powersim.whu.edu.cn;
4    rewrite ^(.*) https://$server_name$1
         permanent;
5  }
6  server {
7    listen 443 ssl;
8    ssl on;
9    # other
10 }
```

Apart from the preceding configuration, other *location* blocks are also configured to serve static contents such as images, CSS and 3-D models. A remained crucial issue is to address the web resource conflicts caused by similar naming styles of the two systems. A feasible solution that has been used in the re-deployment is to rename the resource in the new system to avoid conflicts.

4.3 Re-deployment Result

Fig. 6 shows the access illustration of re-depolyment of the two NCSLab servers, in which users can access the web-based online experimentation system with end-user devices such as tables, PC and smartphones. The access requests first arrive at the intended NGINX proxy server which would then proxy the requests to corresponding servers according to the specific *location* directive.

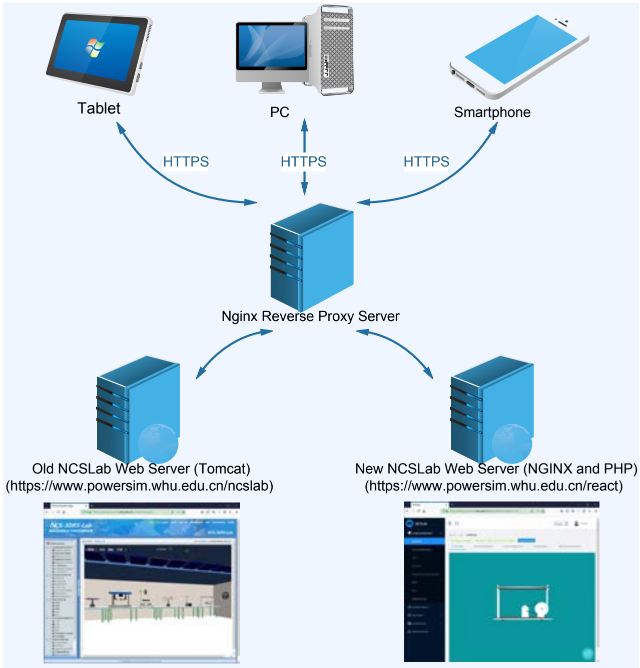

Fig. 6. Access illustration of re-deployment of NCSLab servers

The NGINX reverse proxy solution does not have to change anything about the YUI-based NCSLab system, and only a few static files have to be renamed for the React-based NCSLab system. From the perspective of users of the YUI-based system, there is no difference for their online experimentation. Though the React-based system uses a PHP with NGINX server different from the Tomcat server of YUI-based system, they can be integrated together seamlessly using NGINX reverse proxy.

5. DISCUSSION

NGINX has been widely used in web-based systems (Oniga et al. (2018); DeJonghe (2019)). There are 300 million websites throughout the world choose to run on NGINX (DeJonghe (2019)). This paper takes the online laboratory system NCSLab as an example to demonstrate the NGINX-based server-side re-deployment, which shows that NGINX is reliable and high efficient. For those systems which need to be updated and upgraded, NGINX reverse proxy can be employed to provide continuous services without more domain names or HTTPS certificates, thus time and efforts can be spared.

The NGINX reverse proxy has been utilized in the case that the two systems are coexisting. In the future, when the React-based new system has reached a fully-developed or mature stage and been applied in large scale, the old YUI-based NCSLab system may not be necessary, thus, the NGINX reverse proxy server discussed in this paper are no longer useful. However, the methodologies and solutions provided in this paper are cost-effective and highly universal, especially for those demands with various different kinds of servers/systems to serve multi-purposes.

NGINX can also be used as the HTTP load balancer to improve system performance, scalability, and reliability with a distributed server structure. In the future, NGINX can be applied to other scenarios in NCSLab, for example, HTTP load balancing for server clusters will be useful to handle mass visiting request.

6. CONCLUSION

In this paper, the server-side re-deployment of NCSLab systems using NGINX reverse proxy is introduced, which spares time and effort for system deployment and maintenance, and is also cost-effective. The previous network topology of the NCSLab systems which are currently coexisting for web-based online experimentation is presented. The deployment of NGINX server is discussed in details, the results of which show that the deployed NGINX server enables a modular backend that can integrate different backend technologies such as Tomcat and PHP (with NGINX) together regardless of the previous network structure. Moreover, with proper configuration, all of the web resources of the two systems such as images and CSS and real-time experimentation can be accessed by the reverse proxy provided by NGINX. Owing to the NGINX proxy server, efforts have been spared and costs have been saved for fewer HTTPS certificates and domain names, *etc.* The methodologies and solutions provided in this paper can be employed in other web-based systems and scenarios.

REFERENCES

Abdullah, M., Iqbal, W., and Erradi, A. (2019). Unsupervised learning approach for web application autodecomposition into microservices. *Journal of Systems and Software*, 151, 243–257.

August, S.E., Hammers, M.L., Murphy, D.B., Neyer, A., Gueye, P., and Thames, R.Q. (2016). Virtual engineering sciences learning lab: Giving STEM education a second life. *IEEE Transactions on Learning Technologies*, 9(1), 18–30.

China-University-MOOC (2018). Remote and virtual simulation experimentation for classic control theory. `https://www.icourse163.org/course/WHU-1003368016/`. Online; Accessed October 20, 2019.

De Jong, T., Linn, M.C., and Zacharia, Z.C. (2013). Physical and virtual laboratories in science and engineering education. *Science*, 340(6130), 305–308.

DeJonghe, D. (2019). *NGINX Cookbook*. O'Reilly, Sebastopol, CA, USA.

Galan, D., Chaos, D., De La Torre, L., Aranda-Escolastico, E., and Heradio, R. (2019). Customized online laboratory experiments: A general tool and its application to the furuta inverted pendulum. *IEEE Control Systems Magazine*, 39(5), 75–87.

García-Zubia, J., Orduna, P., López-de Ipiña, D., and Alves, G.R. (2009). Addressing software impact in the design of remote laboratories. *IEEE Transactions on Industrial Electronics*, 56(12), 4757–4767.

Gomes, L. and Bogosyan, S. (2009). Current trends in remote laboratories. *IEEE Transactions on Industrial Electronics*, 56(12), 4744–4756.

Guan, S., Hu, W., and Zhou, H. (2018). Front-end and back-end separation-react based framework for networked remote control laboratory. In *2018 37th Chinese Control Conference (CCC)*, 6314–6319. IEEE.

Gustavsson, I., Zackrisson, J., Håkansson, L., Claesson, I., and Lagö, T.L. (2007). The visir project–an open source software initiative for distributed online laboratories. In *REV 2007*.

Hu, W., Lei, Z., Zhou, H., Liu, G.P., Deng, Q., Zhou, D., and Liu, Z.W. (2017). Plug-in free web-based 3-D interactive laboratory for control engineering education. *IEEE Transactions on Industrial Electronics*, 64(5), 3808–3818.

Hu, W., Liu, G.P., and Zhou, H. (2013). Web-based 3-D control laboratory for remote real-time experimentation. *IEEE Transactions on Industrial Electronics*, 60(10), 4673–4682.

Jong, T.d., Sotiriou, S., and Gillet, D. (2014). Innovations in STEM education: the Go-Lab federation of online labs. *Smart Learning Environments*, 1(1), 1–16.

Lei, Z., Hu, W., Zhou, H., and Zhang, W. (2018a). Integrating a wireless power transfer system into online laboratory: Example with NCSLab. In *Online Engineering & Internet of Things*, 278–289. Springer.

Lei, Z., Hu, W., Zhou, H., Zhong, L., and Gao, X. (2015). A DC motor position control system in a 3D real-time virtual laboratory environment based on NCSLab 3D. *International Journal of Online Engineering (iJOE)*, 11(3), 49–55.

Lei, Z., Zhou, H., and Hu, W. (2019). Combining MOOL with MOOC to promote control engineering education: Experience with NCSLab. *IFAC-PapersOnLine*, 52(9), 236–241.

Lei, Z., Zhou, H., Hu, W., Deng, Q., Zhou, D., and Liu, Z.W. (2018b). HTML5-based 3-D online control laboratory with virtual interactive wiring practice. *IEEE Transactions on Industrial Informatics*, 14(6), 2473–2483.

Lei, Z., Zhou, H., Hu, W., Deng, Q., Zhou, D., Liu, Z.W., and Lai, J. (2018c). Modular web-based interactive hybrid laboratory framework for research and education. *IEEE Access*, 6, 20152–20163.

Oniga, B., Munteanu, A., and Dadarlat, V. (2018). Open-source multi-protocol gateway for internet of things. In *2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 1–6. IEEE.

Restivo, M.T., de Fátima Chouzal, M., Abreu, P., and Zvacek, S. (2018). The role of an experimental laboratory in engineering education. In *International Conference on Interactive Collaborative Learning*, 644–652. Springer.

Rodriguez-Gil, L., García-Zubia, J., Orduña, P., and López-de Ipiña, D. (2017). Towards new multiplatform hybrid online laboratory models. *IEEE Transactions on Learning Technologies*, 10(3), 318–330.

Santana, I., Ferre, M., Izaguirre, E., Aracil, R., and Hernandez, L. (2013). Remote laboratories for education and research purposes in automatic control systems. *IEEE Transactions on Industrial Informatics*, 9(1), 547–556.

Soni, R. (2016). *Nginx*. Apress, Berkeley, CA.

Zhou, H., Lei, Z., Hu, W., Deng, Q., Zhou, D., and Liu, Z.W. (2017). A multi-criteria method for improving the assessment of students' laboratory work using online laboratory. *International Journal of Engineering Education*, 33(5), 1654–1663.