Development of Remote Laboratories Using Cloud Architecture with Web Instrumentation

Ananda Maiti[1] and Chinmay K Maiti[2]
Indian Institute of Technology
Kharagpur, India
[1]anandamaiti@live.com, [2]ckm@ece.iitkgp.ernet.in

***Abstract*—Remotely Accessed Laboratory (RAL) is an environment which enables students to interact with instruments to perform hardware-based experiments from distant locations. In this paper, we discuss an approach to design remote experiments by representing instruments in the internet as objects or resources. The technology used to implement the architecture on the web is based on SOAP and RESTful web services and cloud computing. An integrated remote microelectronic device characterization and parameter extraction experiment is presented as a case study using Agilent 4156C parameter analyzer along with LabVIEW software. This technology is most useful for developers of remote laboratory and management system for adding new experiments. From a student's perspective the output and the interface remain the same, but the developers have more flexibility in designing experiments.**

***Index Terms*— web service, remote laboratory, microelectronics, semiconductor device characterization, LabVIEW**

I. INTRODUCTION

In engineering education, laboratory experimentation plays an important role. However, laboratory experiments are commonly not available in distance engineering education. The cutting edge internet technology and virtual instrumentation together may provide a solution for those who may not have access to instruments [1-3]. Also to perform an experiment, proper knowledge of handling instruments and performing experiments distant students need to have familiarity with the operational procedure.

Cloud based architecture may be used to provide instruments over the web to external developers. Web Instrumentation (WI) is the process of representing an instrument [4] or an experiment composing of single or multiple instruments as an object on the web, such that they are accessible at a remote location. WI also allows one to call any service of the instrument from a remote location. The advantages are as follows:

- Remote developers can develop a laboratory without or minimum support from actual providers of services.
- Security is enhanced by using standard protocols to implement the objects.
- The code to create web services and the objects are written once and can be reused still maintaining standard in communication.

II. REPRESENTATION OF INSTRUMENTS ON THE WEB

A. *Laboratory architecture at the Instruement site*

The laboratory consists of programs written using LabVIEW and are used to run the instruments. The LabVIEW VI file includes a set of commands that uses the drivers to interact with the instrument. The VI first initializes the instrument and then passes the user-provided input parameters to the instrument to run the experiments.

A local server is assigned to operate the instruments. The server is on the same network as in the NETLab servers. A Power control unit (PCU) is used to control the supply of power to the instrument when necessary. The computer sends commands to the PCU to switch on the instrument or turn it off.

B. *Role of Web Services*

The role of web services is to provide the instruments functions as a set of web services. An instrument has a set of functions that are used in combinations for running experiments. Hence to develop the laboratory, the user interface must contact and collect data from the server. These commands are in form of web services. A web service is a piece of code that resides on a remote computer and can be invoked from local platforms. The web service completes some tasks and returns the results [5-6]. In NETLab, web services are used for followings:

a. Once the data is collected, it is sent back to the users through RESTful web services.
b. For switching on or off the instrument, RESTful web service is used.
c. For presenting the instrument as a set of functions on the web, SOAP web services are used.

REST services are easy and quick way to create a web service that can be easily parsed using other programs. SOAP services on the other hand use objects that can be created from remote interfaces and then invoke its functions. RESTful web services or internal web services are used to exchange data from the instrument because the structure of data varies from experiment to experiment. Also the amount of data may vary substantially and usually in huge quantities from each run of experiment. Hence an internal web service can best describe the output of an experiment. The response for the users is in XML format. Furthermore, the XML output can very easily be parsed in any user client program using Java, C, and Flex etc.

SOAP services or external web services provide the package of classes that represents the instrument. These functionalities are available from outside the laboratory network domain and hence must be secured and abstract to the external developers. The SOAP provides higher level of reliability and security for application. It basically encapsulates or forward the internal web service output to the users in a secure and abstract manner.

C. *Instrument as a SOAP Object*

Simple Object Access Protocol or SOAP based web services operates as if the system is a collection of objects and their methods (see Figure 1). In the NETLab the objects are the instruments and commands that can be executed are methods associated with it. There can be some basic method of the instrument object such as:

- Returning the name or status of the instrument.
- Switching the instrument ON or OFF
- Running the instrument
- Returning data

In order to access the web services, the users must register with the NETLab system and a unique user id is given to him/her. This user id must be passed with each method. The methods will check whether the user request can be permitted or not and then proceeds. This is used to keep a record of all the user interactions as well as authentication of instrument access for external developers.

D. *Consuming the Web Services*

An instrument when put on the web becomes available to an user for use. If it is used by independent user, there is a chance that multiple requests may arrive at the same time. Thus a scheduling mechanism i.e., how to manage a group of users accessing the same instrument in a time slot is required. Since the instruments operate in a Non-pre-emptive manner, it can perform only one request at a time in the NETLab system and must make sure that no two requests are executed simultaneously. For this purpose the Hybrid Time scheduling scheme is employed in the remote developers end [7] .

E. *Security Issues*

To prevent misuse or accidental collapse of the system, security issues must be addressed properly. This includes the user management and instrument safety procedures. The users must create an identification number with the system. Once an experiment is started, a session is allotted to that particular user. The user's requests are then validated by this information. For instrument safety, the values passed for execution are validated by checking with the maximum allowed safety limit of each parameter. The system can keep track of user input behavior and bar them from the system, if necessary.

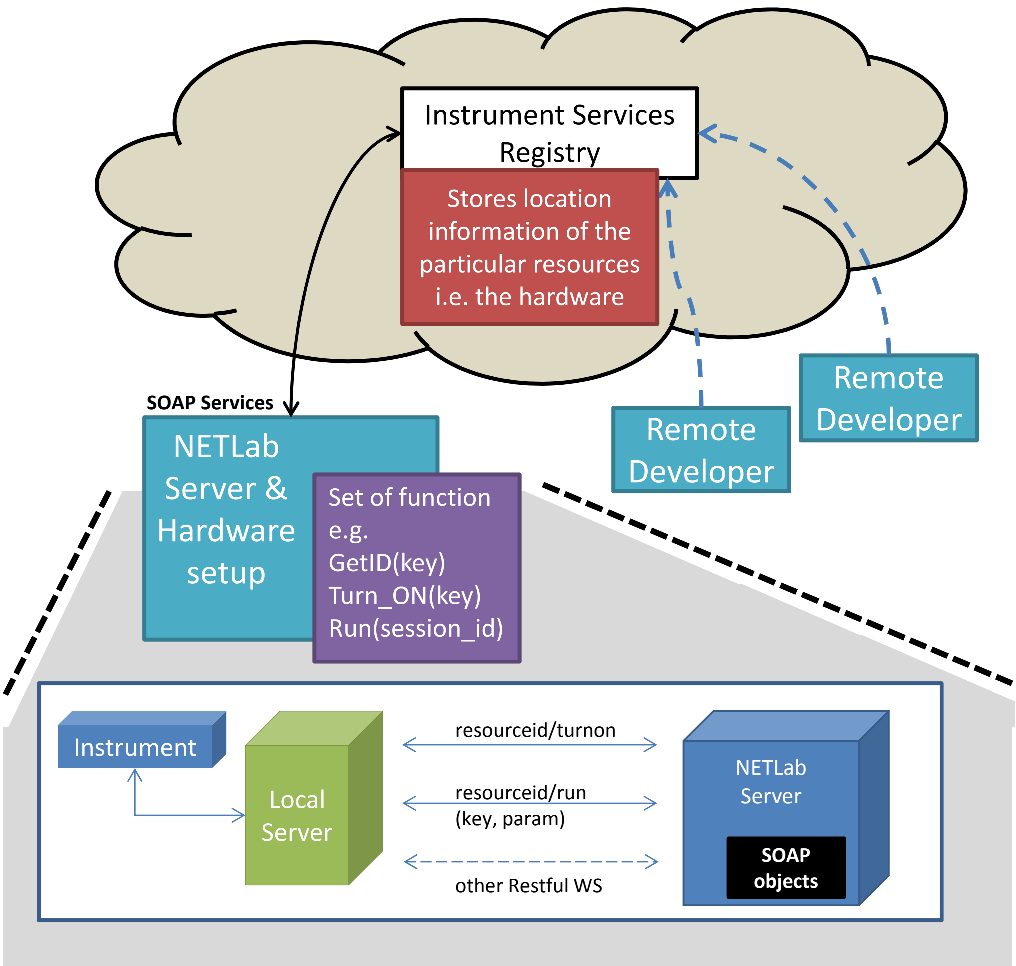

Figure 1. Cloud architecture.

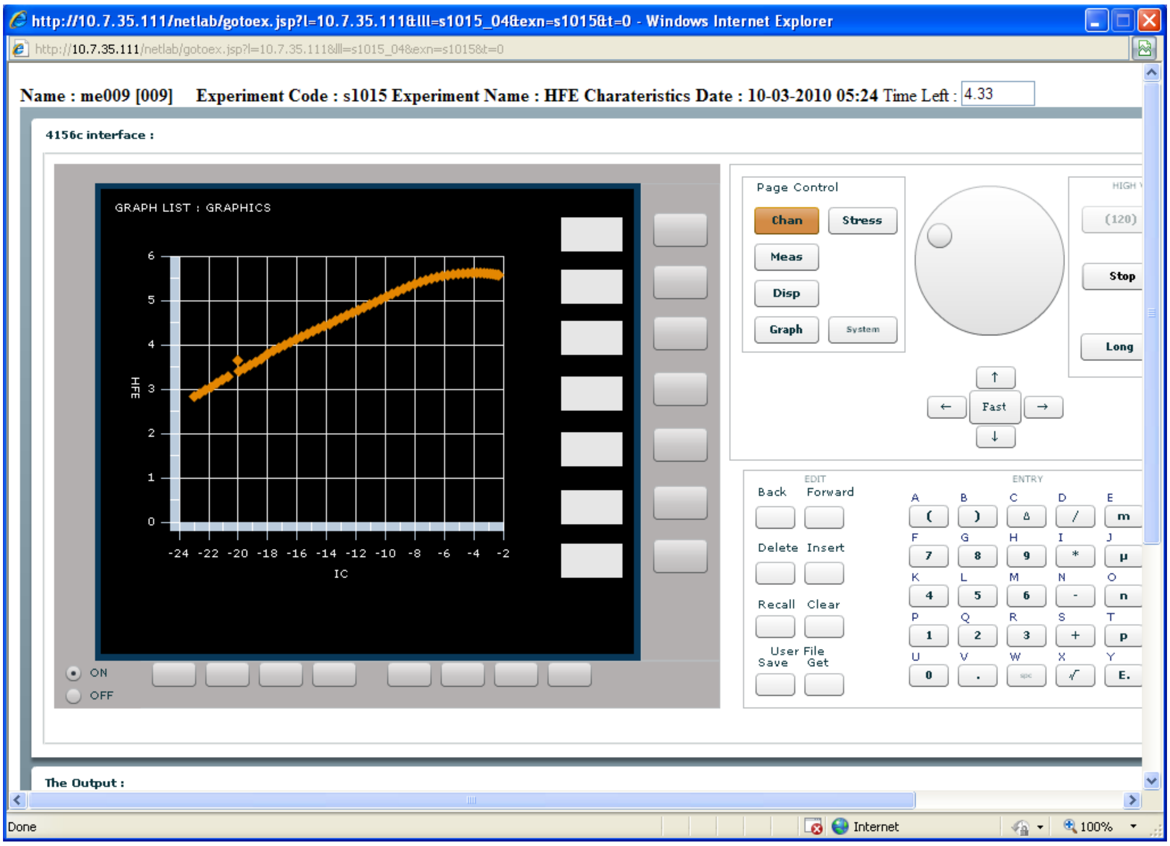

Figure 2. User interface.

III. CLOUD ARCHITECTURE

This system of using web services to represent instruments and run experiments and exchange data over different platforms enables users to combine more than one experiment to make new experiments. A typical example is the output of BJT characterization experiment being combined with BJT SPICE parameters extraction experiment. The two experiments are separate objects, but can be used to design a common experiment according to the remote user's needs. Hence a new platform can be built for technology CAD (TCAD) Laboratory that is customized to the user requirements. The remote developers can search the registry [8] and find out the available resources and use them (see figure 1).

A. *Scheduling :*

Since the resources cannot be used simultaneously by different users nor it can be replicated temporarily, a proper scheduling mechanism is required at the instrument service provider end also. In this case the queuing algorithm is used at each instrument location where the requests are processed one after another as they arrive.

B. *Other issues with a cloud based model*

Cloud Computing is one of the main interests of businesses and researchers [9, 10]. From a management perspective, cloud computing present some challenges such as Quality of Service (QoS) from the service invoked/used by the consumer. Service Level Agreements (SLAs) solves this problem by joining service computing with acceptable levels of safety and service types so that customers can know about the system and recover if there is some problem. Such issues must be addressed while developing the remote laboratory services.

One of the biggest advantages of this technology is that it can support on-demand needs of users and provide the developers with a vast array of instruments to design different kinds of experiments. This gives access to an enormous amount of computing and storage resources also. The access to all these resources is not the only advantage of Grid technologies, the integrated security features are as well which prohibit the unauthorized access to the resources and instruments located within the Grid [11].

Use of Cloud or Grid technologies has some disadvantages also as these technologies are still under development. The developers willing to use their functionalities are required to have a lot of expertise about the underlying technologies and proper ways to use them.

C. *Comparison with Traditional Model*

In general, the students work on an experiment in a remote laboratory. The experiment uses one or more instruments to get the results. Thus traditionally remote laboratory modules are developed with a view where a cluster of standalone experiments are used to make the entire laboratory system.

In the proposed new approach, the system consists of several instruments that are used for different experiments which in turn make up the laboratory module. From the student perspective, the interface and experiment should essentially remain the same. The cloud based approach may be used to increase the efficiency of the system such as flexible experiment setup, resource sharing among institutions etc.

IV. Case Study

The case study presented below is DC current gain characterization experiment for a bipolar transistor. This experiment was modified from the earlier client-server architecture. The instrument used is Agilent 4156C. The Net Beans-based Java SOAP web service was implemented with its methods. The user interface of the instrument is created using Adobe Flex 3.5 (see figure 2). The interface closely resembles the actual instrument because it gives a finer impression to the user about the performance of the real instruments.

SOAP based services view the system as a collection of objects and their functions. In the NETLab the objects are the instruments and functions are various operations that can be done with the instrument. The services are created in Java using the Netbeans environment. Here the instrument name is 4156C and the service class is named in4156ojs containing the following methods:

a. getName (key): gives the name of the instrument.

b. TurnON (key): This method turns on the instrument. Generally some time is required before the instrument is ready for execution after it has been turned ON. Hence a delay is associated with the function. It calls the internal web service. Similarly TurnOFF (key) turns of the instrument.

c. RunExperimentHFE (key, parameters): It first checks whether the instrument is ready or not. If the instrument is not ready then it returns error message. Otherwise, it checks if any other request is already under process before calling the web service. It then calls the internal web services for HFE experiment with the parameters that are passed and forwards the result as an XML string.

d. isOFF(key) : Returns false if the instrument is on and returns true if the instrument is off or being turned on, as many instrument require some time to become fully active.

Conclusions

Applications of web services has been discussed that can help in operating remote laboratories from remote locations. By sharing resources between laboratories, the number of experiments can be increased along with user satisfaction and learning outcomes. Detail of instrument and their web implementation, advantages and disadvantages have been discussed. The contention between where to use different technologies to implement web services - RESTful and SOAP with respect to a remote laboratory is presented.

References

[1] J. A. del Alamo et al., The MIT Microelectronics WebLab: a Web-Enabled Remote Laboratory for Microelectronic Device Characterization. See http://ilab.mit.edu.

[2] M. E. Auer and W. Gallent, The Remote Electronic Lab, as a Part of the Telelearning Concept at the Carinthia Tech Institute, Proc. of the ICL2000, Villach/Austria, 28./29.09.2000.

[3] D. Ursutiu, NI ELVIS and LabVIEW: Creative Thinking in Engineering Education, 2009 Annual ASEE Global Colloquium on Engineering Education Budapest, Hungary October 12– 15, 2009.

[4] F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore, Editors, Remote Instrumentation Services on the e-Infrastructure : Applications and Tools, ISBN: 978-1-4419-5573-9, Springer Science+Business Media, LLC New York, 2011.

[5] A. Maiti, S. Mahata and C. K. Maiti, Common interface platform for development of remote laboratories, Remote Engineering and Virtual Instrumentation (REV) 2012, 9th International Conference on, pp.1-4, 4-6 July 2012, doi: 10.1109/REV.2012.6293144.

[6] A. Maiti, Remotely-Triggered Semiconductor Devices Characterization Laboratory in Cloud Environment, TALE 2012, Aug. 21-23, 2012.

[7] A. Maiti, A Hybrid Algorithm for Time Scheduling in Remotely Triggered Online Laboratories, IEEE EDUCON 2011, pp 926-930, 4-6 April, Amman, Jordan, 2011.

[8] OASIS: Web Services Resources Framework (WSRF 1.2). http://www.oasis-open.org/committees/wsrf/.

[9] R. Fielding and R. N. Taylor, Principled design of the modern web architecture, ACM Transactions on Internet Technology, vol. 2, pp. 115–150, 2002.

[10] C. Riva and M. Laitkorpi, Designing web-based mobile services with REST, Proc. Standard Performance Evaluation Corporation (SPEC) Benchmark Workshop, Vienna, Austria, January 2009.

[11] D. Ursutiu et al., WEB Instruments, IEEE EDUCON 2010, Madrid, pp. 585-590,2010.