WebLab: A Generic Architecture for Remote Laboratories

Awkash Agrawal, Sanjay Srivastava
Dhirubhai Ambani Institute of Information & Communication Technology
Gandhinagar, Gujarat, India 382007
awkash_agrawal@daiict.ac.in, sanjay_srivastava@daiict.ac.in

Abstract

Remote laboratories have earlier been proposed to allow sharing and better utilization of experiment resources. However, wide -scale acceptability depends on, among other things, a framework to manage all the tasks associated with a laboratory. The need for such a laboratory is even more important in developing countries where infrastructural resources are limited and improving the resource utilization is important. We propose a Web based 3-tier architecture (WebLab) that provides shared batch mode access to the experiments over low-bandwidth networks to maximize the laboratory utilization. A generic experiment is modeled as a set of inputs, outputs, and constraints. A laboratory is modeled as a collection of experiments sharing a common administrative and access policy. A lab/experiment registration toolkit is designed to capture the metadata for the labs and experiments with minimal programming involved. This allows for a rapid and standardized integration of the experiments with the WebLab. A proof of concept lab is implemented on this architecture.

Index Terms – Remote laboratory, Registration, Distributed Application, WebLab server

1. Introduction

Laboratories play an important part in education in general, and higher education in Science and Engineering in particular. One can make a basic assumption that lab resources should be universally available to the higher education students. However, this assumption is sadly untrue for a majority of universities in developing countries which find it difficult to generate sufficient funds and manpower to create adequate laboratory infrastructure. Further, even in the universities that have the infrastructure available, it has been found that the lab utilization remains relatively low. We believe that many such countries and university systems will immensely benefit if this twin conundrum of inadequate infrastructure at one end and low resource utilization at the other end can be feasibly solved. There are obvious limitations associated with expanding the physical laboratory infrastructure. However, with the increased penetration of Internet, network connectivity among academic campuses and relatively easy availability of personal computers offers an alternate approach to the problem. In our studies, we have found that most of the Indian technical universities have reasonable PC infrastructure and they also have at least low bandwidth internet connectivity for the students. Hence in addition to creating physical lab resources which entail large investments, one may leverage the existing PC-based labs and internet connectivity to share the physical labs among students across universities. Simulation based laboratories and remote laboratories have of course been created in the past with varying degree of success. As simulations are based on mathematical model they can't fully capture all the nuances of a physical experiment. This limitation becomes even more apparent for complex experiments at advanced undergraduate and graduate level labs. Labs which can be accessed remotely over Internet have the advantage that they can be shared by the stakeholders across geographical divide and the lab utilization can be kept high by appropriate time management. However, apart from the computer-interfacing cost of the experiments, each of the experiments needs to be Internet-enabled. This makes the large scale upgrade of existing lab facilities difficult. Also, the large scale integration of different lab facilities at different universities requires standardization of the architecture We have proposed and built a 3-level web-based remote lab architecture that provides a generic support for accessing the experiments over the Internet. The proposed architecture is designed to simplify the task of integrating laboratories and the experiments with the WebLab and provides common interface for managing users, laboratories, and experiments. In particular, the task of adding new labs and experiments to the WebLab infrastructure is achieved using a process

0-7695-3059-1/07 $25.00 © 2007 IEEE
DOI 10.1109/ADCOM.2007.71

requiring minimal programming experience by the lab administrators. User access is through standard web-browser interface. A prototype laboratory has been integrated with the WebLab as a proof of concept.

The rest of paper is organized as follows: Related Work, Problem Definition, WebLab Architecture, implementation Issues, and Conclusion

2. Related Work

Various surveys, design and implementation of remote laboratories have been reported since early 90's. A good review of the existing labs can be found in [9]. It discusses the pros and cons of real, remote, and virtual (simulation based) laboratories. References [1], [2], [3] and [7] provide framework for remote lab based on client-server architecture. Reference [1] is based on the web kernel idea that manages all the user-device tasks. Reference [2] uses a scheduler to allot exclusive time slots to the users. Entire experiment resources are exclusively available to the user during this slot and the experiment runs in the interactive mode. Reference [3] proposes a remote lab system on a Java based, client-server model whose functionality is primarily for interfacing with the user and controlling of the actual lab experiments. References [5] and [10] propose remote laboratories for engineering experiments which is based on the product LabView [4] of National instruments. Reference [11] gives details of PEARL, a framework for remote laboratories. It's distinguishing feature is access of remote experimentation facility for physically disabled students. Another project that has implemented remote lab is iLab/Icampus project at MIT [12]. iLab allows a student to perform the microelectronics lab experiment remotely. iLab/Weblab of MIT is now based on service broker architecture resembling 3-tier web architecture. The first tier is the student's client application that either runs as an applet or as a downloaded application on the student's workstation. Second tier is the service broker and the third tier is the Weblab server itself. Cyberlab [13] of Stanford University mainly provides remote access to optics laboratory. The domain of our interest, namely, remote lab access over low bandwidth network, very high demand-supply ratio, and a generic architecture independent of any specific class of laboratories is not sufficiently addressed in the work that we have surveyed. Bandwidth limitation, which is the major concern, has not been addressed in any of the implementations. Most of the solutions are specific to a restricted domain whether it is of control engineering, or physics laboratory experiments etc. Management of the experimental data & results generated by the experiments has been weakly addressed, since the emphasis has been more on how to run the experiments remotely. Some of the implementations require client software to be installed, which may hinder open access of the remote lab. Lastly, there is a lack of standard interface and toolkit for integration of a generic experiment to the remote lab infrastructure. It is important to allow rapid integration of labs and experiments from different university systems with minimal programming requirement by the administrators.

3. Problem Definition

We wish to create a remote lab architecture that provides all the functionality of a laboratory over Internet. It should provide a seamless view of all the laboratories registered with it along with the experiments under its control. It should provide a consistent user interface for experiment, user and data management. It should primarily provide batch mode access to experiments and optimize the lab utilization within the constraints of lab access policy. In sum, the system should model the commonalities of all the experiments and represent it to the system in a generic way. This allows a lab administrator to plug in the experiment into the WebLab system by specifying the generic structure of the experiment In this context, we formally define an experiment as a collection of a

Set of inputs

$$X=\{x_1, x_2, x_3, \dots x_n\} \qquad (1)$$

Set of constraints

$$C=\{g_1, g_2, g_3, \dots g_m\} \qquad (2)$$

$$g_i = g_i(x_1, x_2, x_3, \dots x_n)$$

Where g_i is a constraint function.

Set of Output

$$O = \{o_1, o_2, o_3, \dots o_l\} \qquad (3)$$

$$o_i = f_i(x_1, x_2, x_3, \dots x_n)$$

Where f_i specifies output function.

All except the output may be null. Constraints can be further classified into two types Data Constraints & Process constraints. Data constraints are associated with the individual input elements such as datatype, range etc. Process constraints are specified as a general functional of inputs. Even though the definition is quite general, it allows us to model most of the initial and boundary conditions for a given experiment and creates a form-based experiment registration process that obviates the need of complex programming during the experiment deployment.

4. WebLab Architecture

The proposed WebLab architecture is tailored to optimize laboratory resource utilization under high demand-resource ratio. There are three different roles we have identified for the users of the WebLab: WebLab administrator who is responsible for overall administration of WebLab server, lab administrators who are responsible for the management of lab servers & experiment devices, and students who can perform experiments. Figure 1 shows the architecture of the WebLab. It is 3-tier architecture with components distributed across the network It provides the seamless view of experiment across various laboratories to the user who is performing experiment. 3-Tier architecture has been chosen consciously keeping following things in consideration. Starting with allowing the hookup of new experiments into WebLab toolkit in a centralized manner, yet management and control of the experiments are with the individual labs. Secondly Centralized authorization, authentication and registration of users but scheduling and prioritization of user experiments is in the hands of individuals lab owners. Generic definitions of experiment are with WebLab server and finer details are with individual lab server. There are some other choices available such as web services architecture but some problems have been identified with them. For example with the case of web-services the overhead involved in the translation of payload to the device specific format is substantial and can affect the performance. Secondly the device controller (Controlling software for experiment apparatus) should have capability to understand the semantics of XML to do the necessary translation to a device specific format. Thirdly is as there is no legacy code base available for experiment control.Considering the above points, There are four parts to it

- User, depending on the role, performs an experiment, or registers a new experiment, or creates and manages user account.

- Weblab Server provides two broad functionalities that of an Experiment manager and Resource manager. As an Experiment manager it is responsible for authentication of user and input collection, constraint verification, scheduling & result collection of experiment.As a Resource manager it acts as creator and maintainer of Lab registry, Access registry and Experiment registry. These registries hold the information pertaining to the lab, access policy and individual experiments. These registries are created by the process of registration which is discussed in detail in the next section. All the information that is required by the experiment manager to perform various functions from user authentication, input collection, constraint checking to result collection and display is provided by these registries. Constraint checking (initial and boundary conditions) for the experiments is normally performed by the experiment device. However, we have shifted this functionality to the WebLab server. This allows for quick turn around time. For the user if there are flaws in the input data as the WebLab server flags the error. This is quite significant in the batch-mode operation where the experiment may be scheduled to run at a later time. Further, the experiment utilization also improves as the experiment device is guaranteed to get proper input. Secondly it also ensures the safety of the experiment devices due to improper input data. It also provides two level safeties for experiment apparatus, one at WebLab server side and other at device controller side if any.

- Lab server also performs two different roles, that of an experiment manager and of a device manager. As an experiment manager it performs input output operations and status management. As a device manager, it is responsible for device registry creation and maintenance. Device registry holds the necessary information for making experiment devices remotely available. This includes information like the network address of the device hosting the experiment, necessary software installation for communicating with lab server at the experiment device controller etc. Device registry holds the necessary information to make experiment remotely available. device registry also holds the information for dynamic binding of request to perform experiment. Requests are not directly coming to Lab-Server, It comes to it via It WebLab server, thus hiding the identity of Lab where it is performed and securing the Lab setups. At the same time hiding the details of how this binding takes place.

- Experiment device controller is responsible for controlling the actual experiment. It sends and receives data and control signals to and from experiment apparatus, transferring data to and from Lab server. Even though our architecture has explicit support only for the batch mode experiments for the reasons stated earlier, it is possible to provide a limited level of interaction in this model. One can break an experiment into a sequence of experiment stages. At the end of each stage the output can be examined by the user and if required, modified. The new input then gets fed into the next stage. Scheduler can ensure that all the stages of the experiment are scheduled on the same device in sequence. This ensures that the device remains in a consistent state. However, this limited interaction remains non-realtime in nature. In any case, our assumption of low bandwidth Internet connectivity doesn't permit realtime experiments with strict time bounds or low-delay interactivity. A limited interactivity experiment of this type may potentially reduce the device utilization significantly as a new experiment can not be scheduled on the device till all the stages are complete. For this reason, only a limited set of privileged users should be allowed to use this mode of operation. Video streaming is also not the part of first version of the architecture since it consumes large amount of bandwidth. Many of the video streaming experiments are also realtime in nature and hence not considered in the proposed architecture. Next we discuss the scheduling and registration processes of the WebLab.

A. Scheduling

Scheduling of our system maintains a persistent queue for the experiments submitted in respective labs. As the experiments are performed in batch mode, a persistent state scheduler is designed. Scheduler maintains a persistent queue for the submitted experiment input. As the queue is persistent in nature there is no loss of data between the failures, hence user-supplied data is safe. It is a priority based. Scheduling, that takes into account the laboratory access policy set by the lab administrators. We anticipate that every lab will designate a set of privileged users who will have priority over ordinary users in performing experiments in that lab. Even though more elaborate access policies can certainly be implemented, we have currently implemented a 2-level of priorities to wait for the submission of experiment parameters to the system till it gets free. Rather the user can submit inputs of the experiment any time when he/she is desires the same time design do not allow user to capture the resources indefinitely which are assumed to be scarce and sharable in nature.

B. Registration

Our WebLab architecture models a distributed heterogeneous system. It is envisaged that many disparate laboratories from many different universities will offer experiments through WebLab. It is assumed that many of the laboratories may not have the complex programming expertise typically required to integrate experiments to the WebLab system. In such a scenario, it is important to develop a mechanism that simplifies the task of experiment integration with the WebLab having no or minimal programming expertise required at the individual labs. Registration captures the soft state of a particular experiment. This essentially means capturing sufficient metadata corresponding to the experiment, so that it can be made Internet accessible. This includes metadata for input parameters, output parameters, and constraints whether it is process constraint or data constraints, the signature of the functions which has to be used in constraints specification. It may also include metadata for documentation, manuals, analysis tools etc, necessary software at controller required for communication with the Lab server and for translation of input stream to device specific format. Simple constraints that are arithmetic or logical expressions with primitive operators are directly supported. Complex constraints that may be in the form of boundary value equations or differential equations etc requires certain amount of programming. All the operators which are not directly supported must be implemented as methods of a constraint class. These methods are stored on the WebLab server and loaded dynamically when the constraints are to be evaluated for a given set of input parameters. Lab registration captures the relevant information about the lab being integrated with the WebLab. This includes the lab server network address, list of experiments available in the lab, and the access control policy for the lab. WebLab uses this information to bind a user request to an experiment device in the lab. It also uses this information for prioritizing the user requests. Device registration captures the relevant information of the device controllers for individual experiments. This includes the address and network interface of the device controller and the necessary software for the translation of experiment input/output values to native format. Based on the metadata captured in the registration process, a UI generator generates HTML based forms for collecting input data and delivering the result.

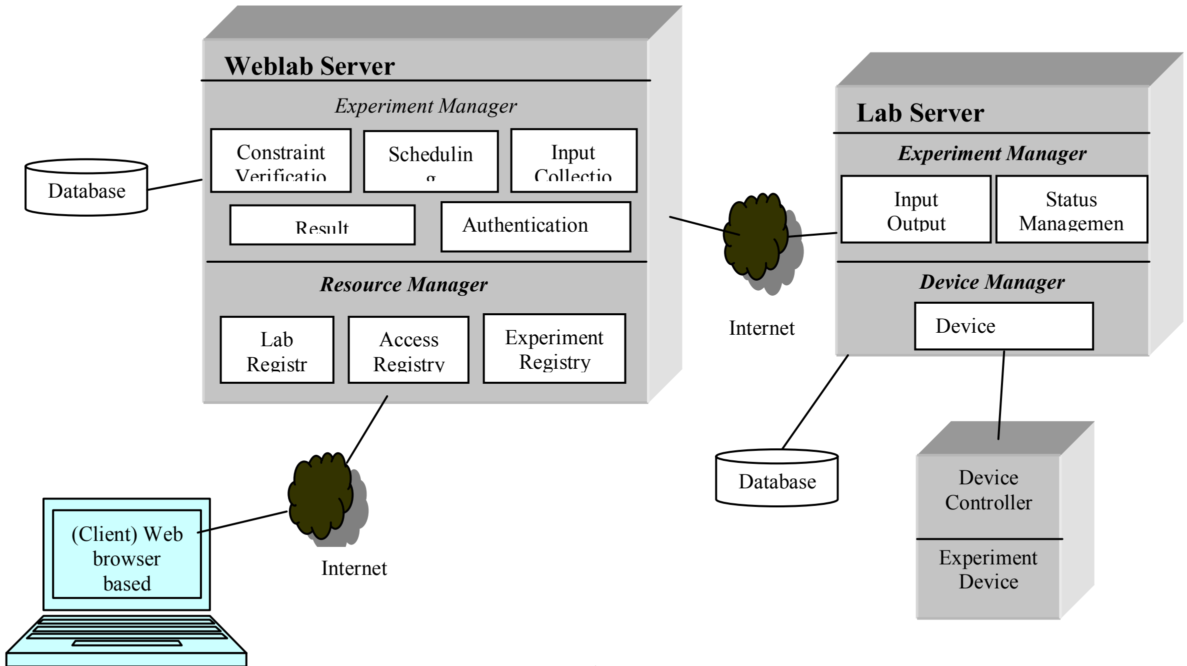

Figure 1
WebLab Architecture

5. Implementation Details

Various aspects have been analyzed before coming to the decision of implementation platform.. We wished to provide a browser based user interface so that no separate client-side software needed to be installed. At present, we do not provide data analysis tools, so the user functionality is limited to experiment selection, providing the input parameters, selection of appropriate output filter and downloading the result. Similarly, the browser-based lab administrator interface provides registration of experiments and lab management functions. We needed a platform that could tie together the various pieces running different hardware and OS. Finally we wanted to choose a platform to minimize the cost of overall implementation and optimally utilize the resources.J2EE platform with IBM Websphere application server version 5.0 has been chosen for the purpose of front end, lab and experiment management. IBM WebSphere application server has full support for the Java SDK 1.4 & J2EE 1.4.A new administrative model based on the Java Management Extensions (JMX) framework and an XML-based configuration repository. A Web-based administrative console provides a GUI interface for administration. An interface based on the Bean Scripting Framework, wsadmin, has been provided for administration through scripts. There is a tight coupling between the development environment and the application server, which makes the application development much easier. It is providing web-based administrative console, which make the task of deployment and administration very easy. It provides JMS support with inbuilt message queue that can be used if the persistent requirements are not stringent. As this queue looses the data if application server is down.

EJBs have been used for the purpose of implementation of WebLab server and Lab server and MDB is used for scheduler. Communication between the experiment apparatus and the Lab server is through sockets. Results and experiment parameters are maintained in Oracle 9i database. XML is used for specifying the data constraints. For process constraints, functions used in the constraints are dynamically loaded in to the system as and when required, while the expression representing the process constraints are stored in postfix form. A user interface generator has been developed for the purpose of generating user interface for the experiments. This is used for input collection from the user. During the process of registration, lab administrator provides sufficient metadata corresponding to the experiment parameters so that HTML forms for supplying input corresponding to the experiment is generated dynamically based on that metadata. At the same time we have implemented a UI generator for the registration to simplify the registration process. We also wanted to keep the device controller as light as possible. We do not make any assumption about the experiment controller which may or may not be PC-based. We do assume that socket library is available on the device controller to communicate between the experiment device that is performing the experiment and Lab server, which is a dedicated server for a particular lab. Since sockets are widely supported on a variety of PC/non-PC platforms, it gives us the flexibility in the communication with the Lab Server A prototype implementation of the lab with full implementation of WebLab server has been done. Some of the experiments of computer networks & microcontrollers lab have been taken for integration with the WebLab as a proof of concept. Currently these experiments are made available primarily for performance and usability testing.

6. Conclusion

We have presented a 3-tier web-based architecture for remote lab. It provides support for rapid integration of new experiments to the remote lab and gives a seamless view of the experiments across labs. Shared, batch mode operation and browser-based interface allows for the best resource utilization under low-bandwidth connectivity.

7. References

[1] Deniz D.Z., Bulancak A. , Ozcan G, "A novel approach to remote laboratories*" Frontiers in Education, 2003. 33rd Annual* Vol. No. 1, pp. T3E-8 - T3E-12, 5-8 Nov. 2003

[2] Overstreet, J.W. Tzes, "A. Internet -based client/server virtual instrument designs for real-time remote-access control engineering laboratory", *American Control Conference, 1999. Proceedings of the 1999*, Vol. No. 2, pp. 1472 – 1476, 2-4 June 1999

[3] Sam Hsu, Bassem Alhalabi, and Mohammad Ilyas, "A Java-based Remote Laboratory for Distance Education" *International Conference on Engineering Education (ICEE) 2000 Taipei*, Taiwan, MD2-1, August 14-18, 2000,

[4] "Distance Learning Remote Laboratories using LabView", white paper, http://zone.ni.com/devzone/conceptd.nsf/webmain/7BD0B01FCF3CF61A86256B510059F0FB/$File/WP2238.pdf

[5] Nedic, Z Machotka, J. Nafalski, "A. Remote laboratories versus virtual and real laboratories" *Frontiers in Education, 2003. 33rd Annua*l, Vol. No. 1, pp. T3E-1 - T3E-6, 5-8 Nov. 2003

[6] Esche, S.k Chassapis, C. Nazalewicz, J.W. Hromin et. al., "A Scalable system architecture for remote experimentation", *Frontiers in Education.2002. FIE 2002. 32nd Annual*, Vol. No. 1, pp. T2E-1 - T2E-6, 6-9 Nov. 2002

[7] C. Rohrig and A. Jochheim, "Java-based framework for remote access to laboratory experiments*", IFAC/IEEE Symposium on Advances in Control Education*, Gold Coast, Australia, 2000.

[8] Benmohamed, H. Leleve A, Prevot P, "Remote laboratories: new technology and standard based architecture", *International conference on Information and Communication Technologies from Theory to Applications, Proceedings. 2004*, pp. 101– 102, 19-23 April 2004

[9] Callaghan, M.J. Harkin, J. McGinnity, T.M. Maguire L.P, "An Internet based methodology for remotely accessed embedded systems", *IEEE International Conference on Systems, Man and Cybernetics, 2002*, Vol. No. 6, pp. 157 –162, 6-9 Oct. 2002

[10] Gustavsson. I., "Remote laboratory experiments in electrical engineering education", *Proceedings of the Fourth IEEE International Caracas Conference on Devices, Circuits and Systems, 2002* pp.1025-1 - 1025-5, 17-19 April 2002

[11] Tilman Schafer, Jean Marc Seigneur, Alexis Donelly, "PEARL: A Generic Architecture for Live Experiments in a Remote Laboratory", *International Conference on Simulation and Multimedia in Engineering Education*, Orlando, January 2003

[12] "The challenge of building internet accessible labs", draft,http://icampus.mit.edu/iLabs/Architecture/downloads/protectedfiles/ILabChallenge.doc

[13] Lambertus Hesselink, Dharmarus Rizal, Eric Bjornson, "CyberLab: Remote access to laboratories through the world-wide-web", *International Conference on Advances in Infrastructure for Electronic Business, Science, and education on the Internet* L'Aquila, July 31 - August 6, 2000