Distributed Web-based Simulation with Server Application Approach using Callback Mechanism

Wonbo Shim, Young Hae Lee, Sook Han Kim

Department of Industrial Engineering,
Hanyang University
wbshim@pis.hanyang.ac.kr

Abstract

In this paper, we introduce the concept of web-based simulation and some reviews on the distributed simulation on the World Wide Web (WWW) and classify the features of current web-based simulation under the distributed way. And then we suggest the structure of distributed web-based simulation which can achieve parallel simulation and reduce simulation run time and show how the callback mechanism works to enable the distributions of jobs to clients with push service in Remote Method Invocation (RMI). Finally we present a prototype of distributed web-based simulation.

1. Introduction

Web-based simulation has been introduced very recently. Currently, the development of the information and communication technology is dominated by the Internet. Therefore web-based simulation and its distributed environment become more and more attractive.

In this paper we consider the classification of current researches on distributed web-based simulation and try to test the capability of the distributed web-based simulation using server application approach with Remote Method Invocation (RMI) on the real-world web.

2. Web-Based Simulation

Web-based simulation can be defined as the integration of Internet technology with the field of computer simulation to perform more efficient simulations.

A simulation requires a model and a model requires computing hardware to execute the model. Some parts of a model can be executed on other machines and other parts of a model are executed on user's machine where the user's web browser is located. To build a simulation model, which will executes on the users machine, requires that clients access the users web-server and cause a program to execute. The idea of Java for web-based simulation is to bundle an applet and send it to the clients computer to execute.

Fishwick [17] offers his perspective on the issue of web-based simulation, and identifies many potential impacts of web technologies on simulation, with particular attention given to three areas: (1) education and training, (2) publication, and (3) simulation programs. Extending Fishwicks categories, a review of the current literature base suggests five areas of focus: (1) Simulation as hypermedia, (2) Simulation research methodology, (3) Web-based access to simulation programs, (4) Distributed modeling and simulation, (5) Simulation of the WWW [3].

3. Researches on Web-Based Distributed & Parallel Simulation

The purpose of a distributed and parallel simulation is to reduce simulation run time with distributing simulation job over multiple processes and enable real time interoperability with each simulation machine. This can be achieved essentially by using parallelism which implements the distributed processes on the asynchronous discrete event simulation (DES) in a synchronous way.

To apply this simulation way to web-based simulation, there are many simulation tool which is developed to implement simulation process on the web. In recently, RMI technique in Java is introduced for the distributed simulation over the web. Of course, CORBA and HLA is also used for the distributed web-based simulation.

The WWW is a truly distributed environment. This distributed environment facilitates providing tools to small companies. A review of the literature shows three types of web-based discrete-event simulation tools (adapted, revised, and expanded to four types from Lorenz, Dorwarth et al. 1997) [8]: (1) Server Hosted Simulation, (2) Client Executed Simulation, (3) Hybrid Client/Server Simulation.

These categories are also fairly true on the distributed simulation under the web. There exsit several Java-based distributed simulation platforms (Page et al. 1997). These platforms, however, focus almost exclusively on distributed modeling or parallel execution of a single simulation replication rather than full-scale experimentation and optimization. We divide the researches on the distributed web-based simulations into the two major categories: (1) To run the simulation on the existing structured parallel discrete event simulation (PDES) support system in a stand-alone way (see Figure 1), (2) To run the simulation in a distributed way over the web (see Figure 2).

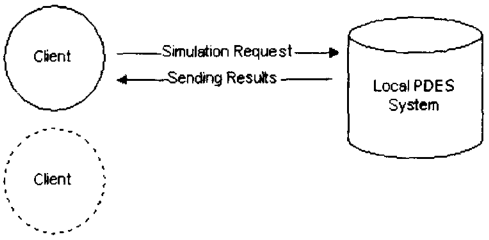

Figure 1: Server hosted distributed web-based simulation

The way of implementing the first category, which most approaches hava followed so far, have the following merits:

(1) This way is easy to construct simulation environment. It just needs to connect the local system which supports PDES and has simulation engines for the clients request.

(2) The performance of simulation is already certificated and safe.

(3) The client is not necessary to concern the implementation of simulation process.

(4) It is not a problem that the speed of current Internet is physically slow and unsteady.

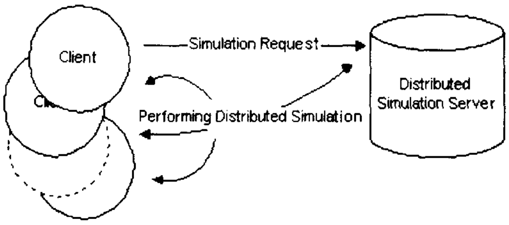
Client
Simulation Request
Distributed Simulation Server
Performing Distributed Simulation

Figure 2: Client/Server Distributed Simulation through the Web

On the other hand, A literature [5] show experiments of simulation on the web for the second category of distributed web-based simulation. This paper tests a capability of implementing simulation over the web which is performed both a server side and client side. Every machines which is participated in this simulation job have simulation engine. Of course, there are some problems at present. The dynamic web environment raises questions on such crucial issues as network congestion, communication reliability, and network security. The first concern is therefore capability rather than run-time performance. Issues of portability, maintainability, and conformance to standards are crucial in demonstrating the feasibility of PDES over the web. The second concern is execution speed, which will be addressed both by designing intelligent distribution algorithms and by taking advantage of emerging techniques aimed at speeding up communication over Internet.

4. Feature of Java and Requirements for Distributed Simulation on the Web

A research have identified the following requirements for the development language and environment for the distributed web-based simulation: (1) model, (2) dynamic loading, (3) multiple excution threads for concurrency, and (4) ability to distribute model instances over multiple processors and to allow those instances to exchange information (network support) [13].

A few major elements in Java for the above requirements are explained in the followings.

4.1 RMI

Remote method invocation (RMI) is the action of invoking a method of a remote interface on a remote object. Most importantly, a method invocation on a remote object has the same syntax as a method invocation on a local object.

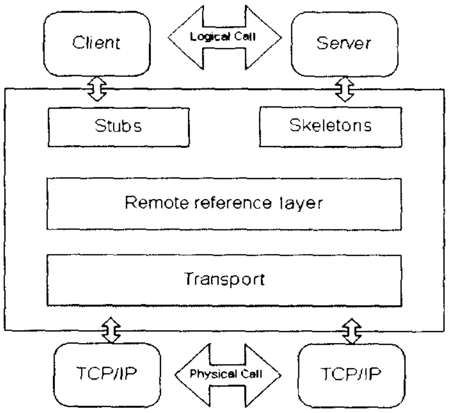

Figure 3: Structure of RMI

Unlike High Level Architecture (HLA) and Common Object Request Broker (CORBA), RMI does not define its object interface language separately from the implementing language. Rather, RMI uses Java's own interface syntax as its object interface language (Javasoft, 1997; Farley, 1997). This considerably simplifies application design and programming, since it is in one rather than two languages.

4.2 Callback

Sometimes, it is more efficient that calling a argument function or objects, possibly at a later time when some event occurs in the discrete event simulation. Callback can be defined as a method which possibly perform asynchronous calls between client and sever [2]. For callback, it is necessary to constitute client's interfaces into distribution objects and then to register client's object reference to server's distribution objects. Figure 4 shows the difference between general synchronous call and asynchronous call by callback. As shown in this figure, there is no need for clients to wait for immediate response form the server, but they can carry out their own ongoing job.

The purpose of using a callback is to provide the ability of push technology in Java RMI for the simulation server and the flexibility in the operation of simulation.

4.3 Object serialization

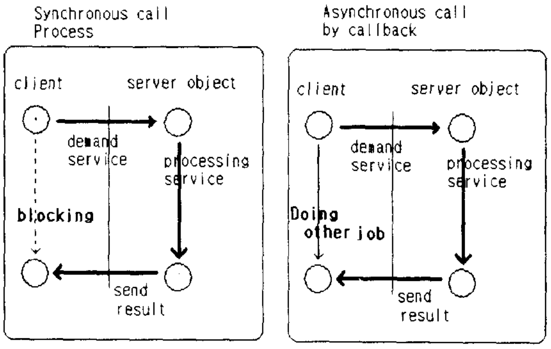

Figure 4: Synchronous call and Asynchronous call

Java (version 1.1 and above) has added an interesting feature called object serialization that allows you to take any object that implements the 'serializable' interface and turn it into a sequence of bytes that can later be restored fully into the original object. This is even true across a network, which means that the serialization mechanism automatically compensates for differences in operating system. Object serialization was added to the language to support two major features: RMI and Java Beans [14].

5. Proposed Structure of Distributed Web-Based Simulation

The web-based simulation should use the concept of distribution in order to cover disadvantages of existing stand-alone environment and make simulation time short. So the proposed web-based simulation uses application for the server as a host to make clients ease connection and performance through server utility and multithreads. Some of characteristics of the proposed web-based simulation under distributed environment is as follow.

(1) Extension of web-server for distributed environment by using application.

- Simulation multithread helps that multi clients simulate at the same time.
- When a client simulates models, server application controls state of event in simulation

(2) Communication of client-server and

client-client under distributed environment is conducted by sending object unit.

(3) Web-based simulation software realizes applet for clients and application for server.

(4) Communication and object call between clients are conducted by modules (IP Address, Java Database connectivity (JDBC), RMI).

Figure 5 is the structure of proposed distributed web-based simulation in this research. Sending Message between client and server is performed by object units. The followings are performing steps.

Step 1. Client connects server and requests a simulation job.

Step 2. Client downloads applet as modeling object.

Step 3. Client inputs simulation parameter in main frame for constructing simulation model and then submits it.

Step 4. When client ask for simulation, it can invoke simulation objects for simulation and communication from the server.

Step 5. Server announces to the other connected client machine over the web and notifies to participate in this simulation job.

Step 6. Server pushes a simulation model to each clients and then clients perform their own undertaken simulation job with RMI.

Step 7. When the clients end its runung simulation, they send results to server and then server alarms the end of simulation to each client, and sends results to all client that needs simulation results.

Step 8. Results from each client and server are collected to server. And server analyses collected results, sends simulation result to each client, and save simulation models and results in database.

Step 9. Client disconnects.

In Figure 5, the role of sever is much greater than the role of server in previous researches, and communications among clients are possible by server. Whenever clients connect to server, server application makes server socket by thread and memorizes the IP address of connected client and registers clients object reference. Server allocates simulation loads to the client according to this address or reference of the client.

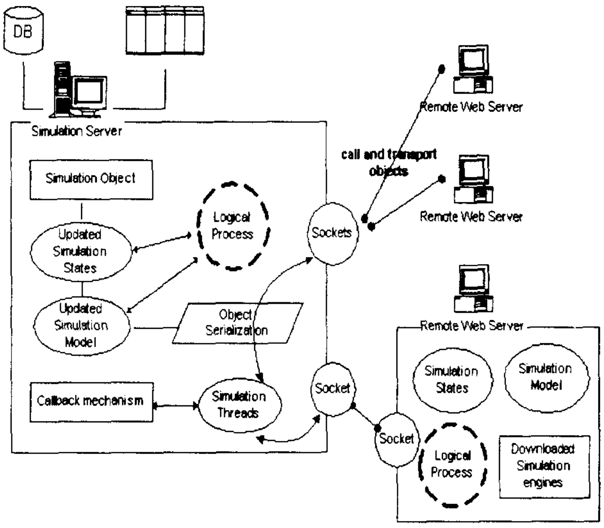

Figure 5 Proposed functions between server and clients

Figure 6 is the structure for a server application which is composed of the following seven objects [21]. These objects in server have their own roles and are invoked the required object by clinets whenever it is needed.

Simulation object: This object contains basic modules which are needed to perform simulation. As described in the right part of Figure 6, three types of parameters, which could be input on the web (direct input, design style input using GUI, input by code) exist. These parameters are changed to codes in required style to perform simulation through interpreter and connected to simulation library and use engines to make outputs.

Random number and random variates object: This object generates many probability distributions such as exponential, normal, binomial and so on.

Statistics (collection) object: related with gathering and analysis output data from simulation run.

Report object: Server and clients use this object when they save or open outputs. If a client calls this object, sever sends required output within the object.

Database object: Output files saved in JDBC are reused through this object. In order to use this file client should get the permission from application in server.

Network communication object: This object is used for the case of information exchanges needed between server and client

Security Manager Object: This object is important aspect of RMI because runing remote objects involves a potential security risk [10].

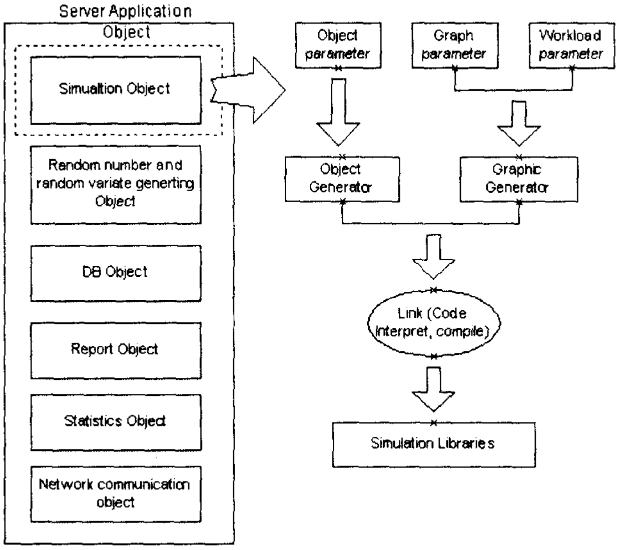

Figure 6: Structure for server application

The simple socket is used to connect and RMI is used to transfer data between clients or between client and server.

A client has very simple structure (see Figure 7) compared with server. It is inefficient to have many objects because it takes a long time to download many objects. Therefore a client requests objects (client uses this object to call the required object from server) and network communication object for information exchange with server and calls only needed object.

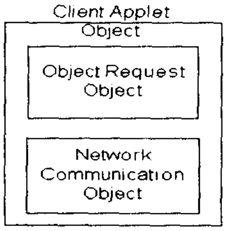

Figure 7: Structure for client applet

6. Conclusions

In this paper, we introduce the structure of distributed web-based simulation and show some simulation objects on the client/server structure and then describe in detail each procedure involving the push technology realized by callback mechanism.

Another paper could be written on the software implementation of the experiment about how callback meachnism is applied into more efficient implementation of distributed web-based simulation with overcoming the gaps of physical networking time between the actual speed of current Internet and the speed of logical processing in a simulation machine. We may possibly make use of these gaps for the unsynchronized PDES [7].

In this paper we have not studied closely related to HLA or CORBA as a distributed platform. It mainly concerns about reusability of object-oriented programs on the web-based distributed environment. But in the future, it will also consider interoperability and follow the approach of HLA and CORBA.

7. References

[1] Ernest H. page, The MITRE Corporation, "Web-based simulation in simjava using remote method invocation", Proceedings of the 1997 Winter Simulation Conference, pp. 468-474, Atlanta, GA, 7-10

[2] Jae Hyun Park, "Technique of Java's asynchronous distribution call" in Microsoftware, Infoage, 1998. 8.

[3] Ernest H. Page, "The rise of web-based simulation: Implications for the high level architecture", Proceedings of the 1998 Winter Simulation Conference, pp. 1663-1668, Dec. 1998

[4] Steffen Straburger, Thomas Schulze, Ulrich Klein, James O. Henriksen, "Internet-based simulation using off-the-shelf simulation tool and HLA", Proceedings of the 1998 Winter Simulation Conference, pp. 1669-1676, Dec. 1998

[5] Enver Yücesan, Chun Hung Chen, Insup Lee, "Web-based simulation experiments", Proceedings of the 1998 Winter Simulation Conference, pp. 1649-1654, Dec. 1998

[6] Wayne J. Davis, Xu Chen, Andrew Brook, "Implementing on-line simulation upon the world-wide-web", Proceedings of the 1998 Winter Simulation Conference, pp. 87-95, Dec. 1998

[7] Dhananjai Madhava Rao, Narayanan V. Thonduglam, Radharamanan Radhakrishnan, Philip A. Wilsey, "Unsynchronized parallel discrete event simulation", Proceedings of the 1998 Winter Simulation Conference, pp. 1563-1570, Dec. 1998

[8] Larry Whitman, Brian Huff, Senthil Palaniswamy, "Commercial simulation over the web", Proceedings of the 1998 Winter Simulation Conference, pp. 335-339, Dec. 1998

[9] Reuben Pasquini, Vernon Rego, "Efficient process interaction with threads in parallel discrete event simulation", Proceedings of the 1998 Winter Simulation Conference, pp. 451-458, Dec. 1998

[10] Arnold Buss, Leroy Jackson, "Distributed simulation modeling: A comparison of HLA, CORBA, and RMI", Proceedings of the 1998 Winter Simulation Conference, pp. 819-825, Dec. 1998

[11] Richard M. Fujimoto, "Parallel and Distributed Simulation" in Handbook of Simulation, edited by Jerry Banks, 1998 John Wiley & Sons. Inc.

[12] Kalyan S. Perumalla, Richard M. Fujimoto, "JANE An architecture for interactive parallel simulations", Proceedings of the 1999 International Conference Web-based Modeling and Simulation, pp. 83-88, Jan 17-20, 1999

[13] Bruno R. Preiss and Ka Wing Carey Wan, "The parsimony project: A distributed simulation testbed in Java", Proceedings of the 1999 International Conference Web-based Modeling and Simulation, pp. 89-94, Jan 17-20, 1999

[14] Bruce Eckel. "The Java IO System" in Think In Java, 1998, Prentice Hall

[15] Buss, Arnold H. and Kirk A. Stock, "Discrete event simulation on the world wide web using Java", Proceedings of the 1996 Winter Simulation Conference. pp.780-785, Dec. 1996

[16] Cubert, Robert M. And Paul A. Fishwick, "Toward a web-based modeling methodology and repository", 1998 Western Multiconference on Computer Simulation, SCS, San Diego, 1998

[17] Fishwick, Paul A., "Web-based simulation: some personal observations", Proceedings of the 1996 Winter Simulation Conference, pp.772-779, Dec. 1996

[18] Iazella, Giuseppe and Andrea D'Ambrogio, "A web-based environment for the reuse of simulation models", Proceedings of the 1998 Western Multiconference on Computer Simulation, SCS, San Diego, 1998

[19] McNab, R. and F. W. Howell, "Using Java for discrete event simulation", Proceedings of Twelvth UK Computer and Telecommunications Performance Engineering Workshop (UKPEW), Univ. of Edinburgh, pp. 219-228, 1996

[20] Erik Berglund and Henrik Eriksson, "Distributed Interactive Simulation for Group-Distance Exercise on the web", International Conference on Web-based Modeling and Simulation, January 11-14, 1998, San Diego, California

[21] Jeffrey A. Joines and Stephen D. Roberts, "Object-Oriented Simulation" in Handbook of Simulation, Edited by Jerry Banks, 1998, John Wiley & Sons, Inc.