Using Web Services for Development and Evaluation of Parallel and Distributed Simulation

Taejong Yoo[a], Daehong Kim[a], Hyunbo Cho[a,*], Enver Yücesan[b]

[a]*Department of Industrial Engineering, Pohang University of Science and Technology, San 31 Hyoja, Pohang 790-784, South Korea*
E-mail: hcho@postech.ac.kr
Tel.: +82-54-2792204
Fax: +82-54-2792870
[b]*Technology and Operations Management Area, INSEAD, Boulevard de Constance 77305, Fontainebleau Cedex, France*

Abstrac *Parallel and distributed simulation is concerned with the efficient execution of large-scale discrete event simulation models on multiprocessors and distributed platforms. After the development of WWW, many efforts in the parallel and distributed simulation have been made for modeling, particularly building simulation languages and creating model libraries that can be assembled and executed over WWW. However, web-based parallel and distributed simulation is restricted by heterogeneous computing environments. Recently, the advent of XML and web services technology has made these efforts enter upon a new phase. Especially, the web services as a distributed information technology have demonstrated powerful capabilities for scalable interoperation of heterogeneous systems. This paper aims to develop and evaluate the parallel and distributed simulation using the web services technology. In particular, a prototype multi-pass simulation framework is implemented using Java-based web services technology. It focuses on the efficiency of multi-pass simulation used for optimization through the distribution of simulation replication to several simulation servers. The development of parallel and distributed simulation using web services will help solve efficiently large-scale problems and also guarantee interoperability among heterogeneous networked systems.*

Keywords*: Web services, Parallel and distributed simulation, Simulation optimization*

Introduction

The modeling and analysis of man-made systems such as communication networks, traffic systems, and manufacturing facilities has been studied for more reliable and well-formed representation. Discrete-event simulation (DES), one of the most suitable tools for the representation of real world systems (Yücesan *et al.*, 2001), concerns the modeling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time (Law and Kelton, 2000). A common goal of DES is to choose the best among a set of competing alternatives. To obtain the good statistical estimates for each alternative, simulation must perform some amount of replications. Consequently, if complexity of simulation model is high or there are numerous alternatives, a great number of simulation replications must be required.

One way to speed up the execution of large-scale simulation is to distribute the simulation runs to several different processors and conduct the simulation experiments in a parallel fashion. The focus of parallel and distributed simulation research has historically been centered on parallel and distributed implementations of sequential simulation models that result in speedup of model execution at runtime (Page *et al.*, 1997). After the development of World Wide Web (WWW), many efforts have been made for modeling, particularly building simulation languages and creating model libraries that can be assembled and executed over WWW. However, the web-based parallel and distributed simulation is restricted by heterogeneous computing environments. Recently, the advent of XML and web services technology has resolved those difficulties. Especially, the web services have demonstrated the scalable interoperation of heterogeneous systems.

The objective of the paper is to develop and evaluate the parallel and distributed simulation using the web services technology. The detailed objectives are (1) to improve the statistical efficiency of simulation optimization via parallel and distributed simulation, and (2) to implement the prototype of the interoperable parallel and distributed simulation via the web services technology.

The remainder of the paper is organized as follows. Section 2 presents a survey of the relevant literature. Section 3 proposes the framework of web services-based parallel and distributed simulation. The prototype implementation is described in Section 4. Finally, Section 5 provides the current research efforts and future works.

Related Work

Web-based Parallel and Distributed Simulation

The topic of web-based simulation spans a variety of areas within the field (Page *et al.*, 1997). Web-based simulation requires the convergence of simulation methodology and WWW technology. Web-based parallel and distributed simulation can have the added feature of code mobility afforded by such network programming languages as Java or network package as Parallel Virtual Machine (PVM) (Douglass and Malloy, 1994). Especially, Java-based simulation-support libraries are emerging that permit the creation of simulation programs as Java applications and applets. Among these are Simkit, JavaSim, JSIM and Simjava. One of the primary advantages of these packages is that they permit network-based simulation models to be developed.

Web Services

The fundamental concept of the web services is to integrate software applications as services. Web services allow the applications to communicate with other applications using open standards. Figure 1 shows that the model has the three basic roles: service provider, service requester, and service broker (Tsalgatidou and Pilioura, 2002).

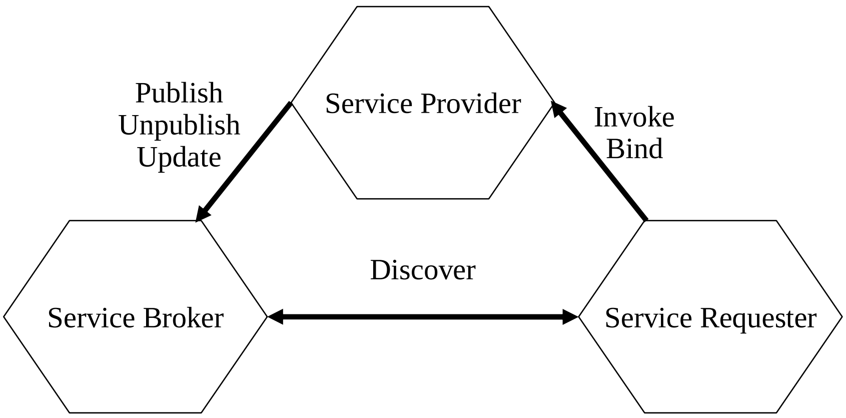

Figure 1. Web services model.

The model must have the standard technologies: communication protocol (Simple Object Access Protocol, SOAP), service description (Web Services Description Language, WSDL), and service discovery (Universal Description, Discovery and Integration, UDDI) (Wang *et al*, 2004). Web services can communicate with each other even if they are running on different operating systems or are written in different languages.

Framework of Parallel and Distributed Simulation with Web Services

In the present work, the web services-based parallel and distributed simulation is proposed to improve the efficiency of simulation optimization. The proposed simulation is based on the ranking and selection procedures as a simulation optimization. Namely, there are several alternatives according to the parameter settings of the simulation model. This simulation supports to find the best alternative. The framework of the proposed simulation is shown in Figure 2. The web server has a simulation optimization module that can be accessed via a web browser. The simulation coordinator calculates the number of replication for each alternative and requests each simulation server the simulation results. A simulation server has the core simulation engine, which receives the alternative index, the parameter settings of simulation model, and the number of replications from the simulation coordinator. It then performs the simulation and sends the results to the simulation coordinator. This process is repeated until the correct selection level is satisfied.

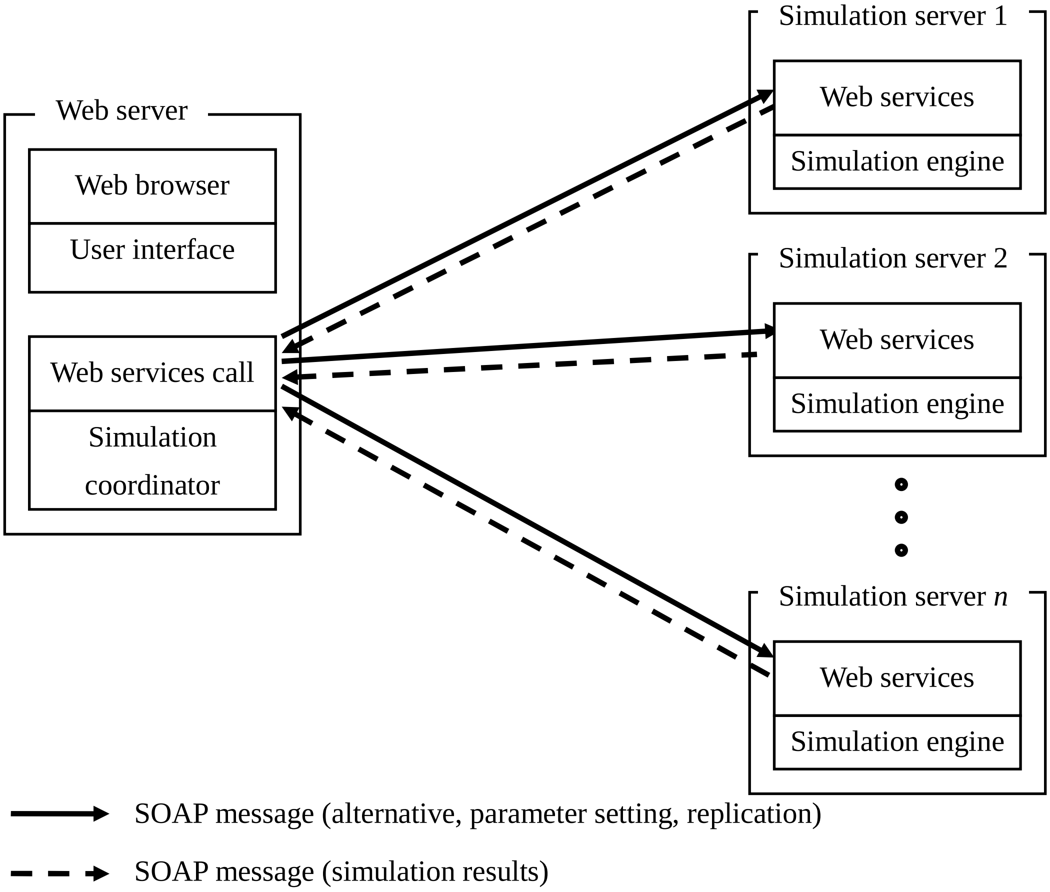

Figure 2. Architecture for parallel and distributed simulation with web services.

'Correct selection' is defined as the event that the selected *best* alternative is actually the best alternative. Therefore, the probability of correct selection can be defined as follows, assuming that the performance measure needs to be minimized:

$$P\{CS\} = P\{\mu_{best}(N_{best}) < \mu_i(N_i)\}, \text{ for all } i \neq best \quad \textbf{(1)}$$

where N_i is the number of simulation replications for alternative i and μ_i is the mean of the performance measure of alternative i.

Since Equation (1) is not easy to compute, the approximate probability of correct selection (APCS) can be estimated and used as a lower bound for the probability of correct selection (Chen, 1996). Note that the computation of the APCS is simply a product of pair-wise comparison probabilities through the application of the Bonferroni inequality.

$$P\{CS\} \geq \prod_{i \neq best}^{M} P(\mu_{best} < \mu_i) \equiv APCS \quad \textbf{(2)}$$

where M is the sample size of alternatives.

Implementation of Parallel and Distributed Simulation with Web Services

This section shows the description of the simulation model and the implementation of the major components in the proposed simulation: (1) simulation server, and (2) simulation coordinator. Three simulation servers are applied to this prototype.

Simulation Model

A prototype shop floor with multi-pass scheduling is applied as a simulation model. The assumptions, parameter settings, and alternatives are as follows:

- Assumptions
  1) Performance criterion is the average flow time.
  2) The number of different part types in the shop floor is 5.
  3) The number of AGV in the shop floor is 1.
  4) The setup time of each machine is constant.
  5) Machine breakdowns are not considered.
- Parameter settings
  1) The number of each machine can be selected between 1 and 5.
  2) The variances of the processing time: i.e. 0.05, 0.1, 0.15... × processing time.
  3) The variances of the transportation time: i.e. 0.05, 0.1, 0.15... × transportation time.
- Alternatives: 3 dispatching strategies × 4 releasing strategies = 12 alternatives
  1) The number of dispatching strategies is 3: EDD, FIFO, and STT.
  2) The number of part releasing strategies is 4: SPT, EDD, FIFO, and STT.

The proposed prototype receives the parameter settings of the simulation model from users and finds the best alternative among the twelve alternatives.

Simulation Server

One of the objectives in the paper is to demonstrate the interoperability of parallel and distributed simulation platforms. The typical development tools of the web services are Java-axis and .Net framework.

- Java-axis applied to the two simulation servers.
  1) Environment: J2sdk1.4.7, Tomcat4.1, and Axis1.2.
  2) Language: Java.
  3) The type of core simulation engine: stand-alone application.
- .Net framework applied to the one simulation server.
  1) Environment: .Net framework1.1.
  2) Language: VB ASP.Net.
  3) The type of core simulation engine: Com object.

The inputs to the simulation server are the alternative index, parameter settings (the number of each machine types, the variance of processing and transportation times), and replication number.

Simulation Coordinator

The simulation coordinator requests each simulation server to perform simulation. The simulation coordinator was implemented with the Java Server Page (JSP) and each simulation must be performed asynchronously. The AsyncCall method supported by “org.apache.axis.client.async” package makes it possible to asynchronously call the Java–axis based web services.

```
Service service1 = new Service();
Call call1 = (Call)service1.createCall();
call1.setTargetEndpointAddress(new URL(address));
call1.setOperationName(new QName("http://soapinterop.org/", "record"));
AsyncCall ac1 = new AsyncCall(call1);
IAsyncResult result1 = ac1.invoke(new Object[]{rule[l], m11, m22, m33, pv1, tv1, num[l]});
```

In .Net based web services, there is no direct method for asynchronously call in JSP. A Java proxy class generator is adopted to create a Java proxy class based on the WSDL specifications of any target web services. The generated Java proxy class supports both asynchronous and synchronous communication methods for calling .NET web services and implements a callback mechanism to support asynchronous communications (Verma, 2003).

```
Service1Proxy sp = new Service1Proxy(address);
IWebServiceCallback callback = new simulcallback();
sp.simulAsync(ruled[m+4], m11d, m22d, m33d, pv1d, tv1d, numd[m+4],callback);
result21[] = (float[])callback.getresponse();
```

After receiving the results of each simulation server, the simulation coordinator performs the ranking and selection procedure and decides whether more simulation is needed.

Experiment Results

To demonstrate the statistical efficiency of simulation optimization, a simple experiment is conducted in the Internet environment. Figure 3 depicts the run time performances on different number of servers applied to the different correct selection. The gap of simulation run time increases geometrically fast with the increase of correct selection.

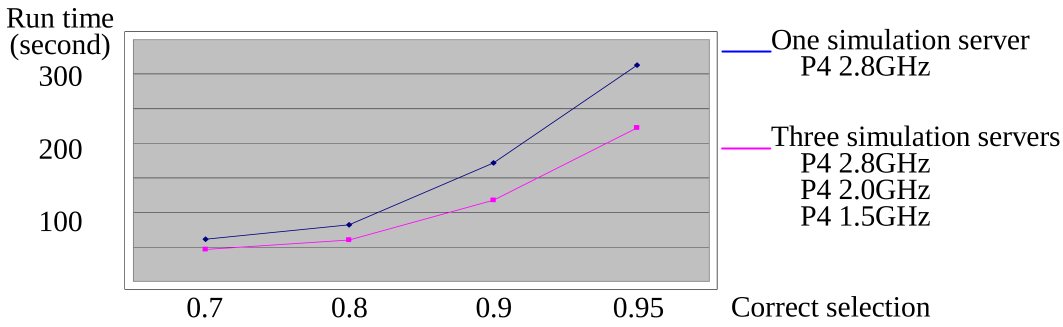

Figure 3. Run time comparison between one simulation server and three simulation servers.

To determine whether the experimental results establish that the simulation time is significantly different between the use of one simulation server and the three simulation servers, an analysis of variance (ANOVA) was performed. The following hypothesis was tested at a significance level of 0.05:

H_0: Total simulation time$_{\text{with one simulation server}}$ = Total simulation time$_{\text{with three simulation servers}}$

The experimental result and the ANOVA result demonstrate that the simulation time of three simulation servers is significantly faster than that of one simulation server as shown in Table 1.

Table 1. Experimental result and the ANOVA test for the simulation time.

Correct selection	The number of simulation server	The number of samples	Mean(second)	Variance	F-value	P-value
0.7	1	30	60.6	833.1	5.641	0.021
	3	30	45.8	366.3		
0.8	1	30	81.7	2578.5	4.074	0.048
	3	30	59.6	1136.6		
0.9	1	30	171.8	12535.6	6.012	0.017
	3	30	117.7	2553.1		
0.95	1	30	319.4	27946.3	6.289	0.014
	3	30	230.3	11206.3		

Conclusion

In this paper, the framework of parallel and distributed simulation using web services technology was introduced for the statistical efficiency of simulation optimization. As shown in the simple performance testing experiment, the simulation time of the three simulation servers is much faster than that of a one server. Using .Net and Java-based web services technology, interoperable parallel and distributed simulation servers can be developed under the heterogeneous networks systems. In the future, the prototype can be improved in such a way that simulation replications can be dynamically allocated according to the capability of simulation server. On the other hand, the interoperable feature of the prototype can be applied to the coordination and the operation of heterogeneous simulation modules such as HLA/RTI.

Reference

1. Chen, C. H. (1996). A lower bound for the correct subset-selection probability and its application to discrete-event system simulations. *IEEE Transactions on Automatic Control*, 41(8), 1227-1231.
2. Douglass, J. T. & Malloy, B. A. (1994). Using a shot clock to design an efficient parallel distributed simulation. *Proceedings of the 1994 Winter Simulation Conference*.
3. Law, A. M. & Kelton, W. D. (2000). *Simulation modeling and analysis*. Singapore: McGraw-Hill Book Co.
4. Page, E. H., Moose, R. L. Jr., & Griffin, S. P. (1997). Web-based simulation in simjava using remote method invocation. *Proceeding of the 1997 Winter Simulation Conference*.
5. Swisher, J. R., Hyden, P. D., Jacobson, S. H., & Schruben, L. W. (2004). A survey of recent advances in discrete input parameter discrete-event simulation optimization. *IIE Transactions*, 36, 591–600.
6. Tsalgatidou, A. & Pilioura, T. (2002). An overview of standards and related technology in web services. *Distributed and Parallel Database*, 12, 135-162.
7. Verma, M. (2003). Platform interoperability: Sun ONE and Microsoft .NET, achieving asynchronous communications. *Technical Article*, Second Foundation Inc.
8. Wang, H., Huang, J. Z., Qu, Y., & Xie, J. (2004). Web services: problems and future directions. *Web Semantics: Science, Service and Agents on the WWW*, 1, 309-320.
9. Yücesan, E., Luo, Y. C., Chen, C. H., & Lee, I. S. (2001). Distributed web-based simulation experiments for optimization. *Simulation Practice and Theory*, 9, 73-90.