Enhanced Multi-Site Hybrid Simulations Using an Internet-based Grid System for Construction and Civil Engineering Facilities

Dong Kun Noh
Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
Email: dnoh@illinois.edu

***Abstract*— KOCED(Korea Construction Engineering Development) is formed to provide a shared infrastructure for construction and civil engineering facilities across Korea. Users can perform remote experiments on these distributed facilities using the KOCEDgrid, which is an Internet-based grid computing system. We present technologies to support enhanced hybrid simulation on the KOCEDgrid. A new real-time controller and viewer provide a user-friendly interface to an enhanced environment for hybrid simulation, which has been built on a new version of the grid middleware and on a new hybrid modeling tool. Users can now conduct experiments more flexibly and combine more than one piece of experimental equipment. The elimination of unnecessary communication overheads has reduced simulation times.**

***Index Terms*— Grid application, grid system, KOCEDgrid, collaboration and hybrid experiment**

I. Introduction

Grid computing achieves high throughput by combining many computers in a virtual parallel architecture across which process execution can be distributed. Grids can use the resources of many separate computers connected by a network (usually the Internet) to solve large-scale problems, and can deal with large data sets by sharing them across many processes.

KOCED (Korea Construction Engineering Development) was formed to provide a shared infrastructure for construction and civil engineering research and education facilities at different locations. The KOCEDgrid is a high-speed network service which supports shared-use collaboration by linking high-performance computers, large databases, experimental equipment, and other hardware and software resources located across the country. It connects a total of twelve universities, research institutes, and government facilities across Korea. The KOCEDgrid is designed to allow dispersed facilities to access to high-speed computing, high-capacity database, and advanced equipment. To utilize the KOCEDgrid infrastructure, KOCED is developing a range of application technologies. The KOCEDgrid will dramatically reduce the amount of equipment required at individual facilities and research centers.

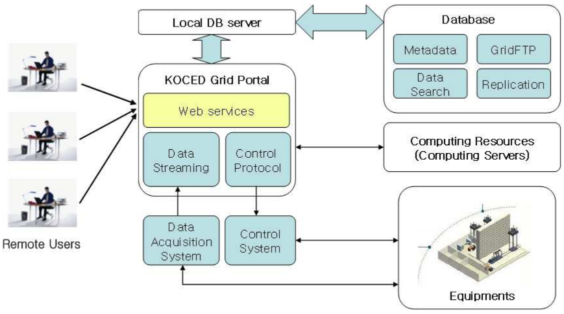


Figure 1. Remote equipments and data services on the KOCEDgrid.

The KOCED collaboration effectively integrates the large-scale experimental facilities, computing facilities, and digital libraries into a single virtual laboratory. The experimental installations include real-time hybrid simulation facilities, earthquake simulation facilities and a wind tunnel. University classrooms can also be connected to these shared research facilities to create an environment that allows remote participation in experiments.

For example, Fig 1 shows how a user can perform experiments regardless of their location, share experimental results, and collaborate remote colleagues. Experiments conducted at KOCED's testing facilities are made available to users in other locations through the KOCEDgrid portals. Implemented as a web service, these portals allow the remote monitoring of experiments and access to test results. Cameras installed in test facilities make it possible to observe and verify experiments in real time. In addition, the KOCEDgrid is connected to the NEES (Network for Earthquake Engineering Simulation) in the U. S. and to the E-Defence network in Japan. This has achieved a degree of globalization of construction experiment facilities, allowing their technologies and information to be shared across a trans-continental research community. KOCEDgrid also enables users to access its database through a range of services, as shown in Fig 1.

Multi-site online experiments, distance learning, and dispersed design are facilitated by the following KOCEDgrid services:

- Resource management services support the location and monitoring of resources which are needed by remote users, allowing them to plan their work and

This paper is based on "A Case Study: Hybrid Simulation on the KOCEDgrid System," by D. Noh, S. Yi, H. Kwon, T. Jeong, J. H. Lee and H. Shin, which appeared in the Proceedings of International Conference on Hybrid Information Technology (ICHIT), Jeju, Korea, Nov. 2006. © 2006 IEEE.

doi:10.4304/jsw.5.7.681-688

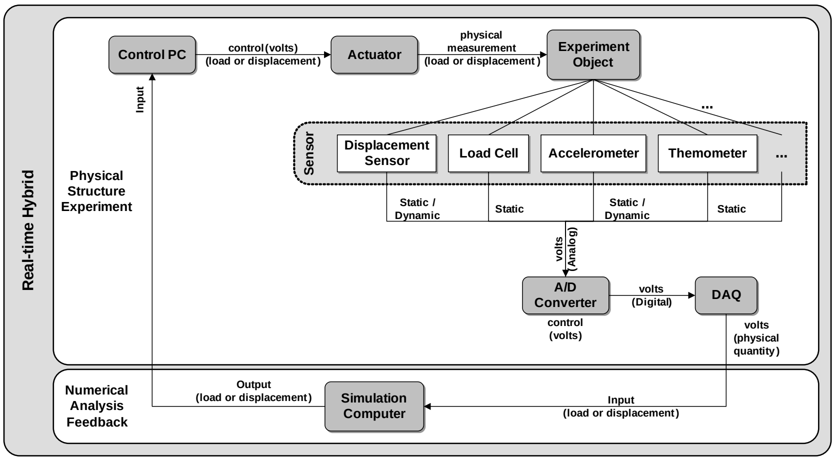


Figure 2. A hybrid simulation seamlessly combining analytical and physical models

obtain results in a consistent manner.

- Data management services support the efficient processing of searching and visualization of experimental results. The real-time transmission of high-quality images allows experiments to be observed from multiple perspectives, and high-quality video recordings are automatically stored in a data repository for future reference.
- Telepresence services provide remote access to experiments. Visual and audio data are transmitted in real time to distant users, and experimental equipment can be operated remotely if necessary. Test results can be sent to several researchers simultaneously, for comment and validation.
- Collaboration services enable distributed teams to work together. Data can be presented in various formats and accessed remotely by researchers at any time. Experimental procedures and measurement protocols can be delivered to test sites in real time.

A major topic of research on the KOCEDgrid is hybrid simulation, which offers a cost-effective alternative to large-scale laboratory testing. A hybrid simulation combines numerical simulations with simultaneous experimental tests of substructure components by integrating the dynamic equation of motion of the overall model. Components which are difficult to model numerically can be built in the laboratory, while components with predictable behaviors can be modeled on computers. Current challenges is the increasing complexity of the geographical distribution of models, which requires allowances for communication delays.

In this paper, we present technologies to support hybrid simulations on the Internet-based KOCEDgrid system. We have already provided an overview of these technologies [1]. We will now expand and elaborate on the ideas behind our case study of hybrid simulation on the KOCEDgrid system. The remainder of this paper is organized as follows. In Sections II and III, we present an example of hybrid simulation on the KOCEDgrid system and show how it uses the KOCEDgrid architecture. In Section IV, we explain how grid services have been implemented and extended to support the hybrid simulations. Section V proposes some changes to the experimental methods of hybrid simulation based on these services. Section VI summarizes our work and presents some ongoing and future extensions.

II. Overview of the Hybrid Simulation

Hybrid simulation (formerly called pseudo-dynamic or on-line computer-controlled testing) is one of the main components of the KOCED project. A hybrid simulation center is being built up at Myoungji University in Korea, which is currently providing prototype small-scale hybrid simulations. In the future, KOCED researchers will be able to implement full-scale hybrid simulations involving a range of facilities at this hybrid simulation center.

A. *Concept of the Hybrid Simulation*

Hybrid simulation seamlessly combines analytical and physical models of a structure and simulates their response in the time domain. Unlike conventional numerical modeling of an entire structure, a hybrid simulation decomposes the structure to be tested into independent substructures that can be modeled computationally or physically. The physical models may be located at different facilities and tested separately, but the results are integrated within a single computational simulation. The control systems of the physical experiments and the computer simulations communicate and influence each other's behavior over the course of the experiment. Data acquired from the physical models is analyzed and fed to the simulations. The simulations, in turn, provide input to actuators which control the contribution of the physical models to the experiment. This exchange of data takes place at each step of the simulation, as shown in Fig. 2.

Implemented in this way, a hybrid simulation allows a researcher to model the well-understood parts of a structure in a finite-element program, while parts that are highly nonlinear or hard to model can be tested in the

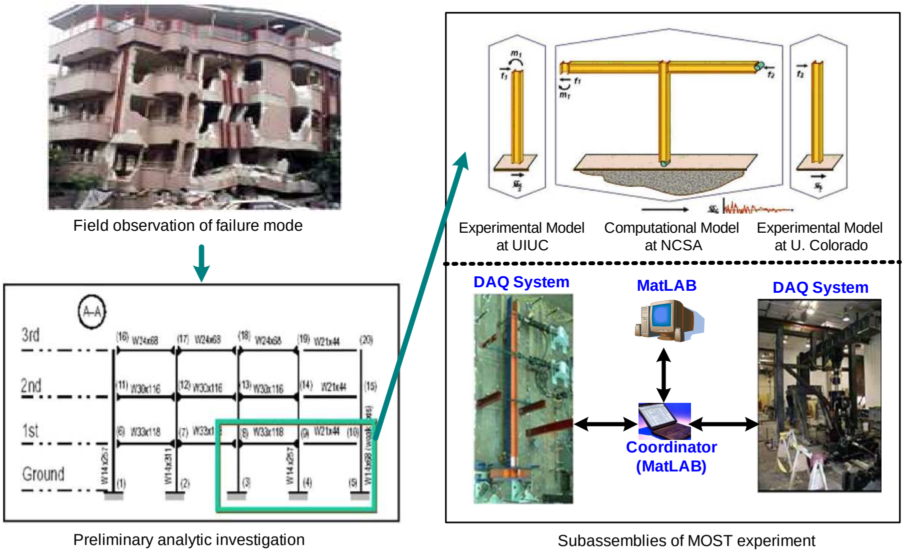


Figure 3. The subassemblies of the framework tested in the MOST experiment.

laboratory. Because the testing specimens can be built in full size or can be deformed quasi-statically, hybrid simulation technique faces fewer capacity restrictions than conventional laboratory testing. It is also possible to avoid physically modeling geometric nonlinearities, awkward three-dimensional effects, multiple support excitations and soil-structure interactions by incorporating them into the analytical model. Furthermore, the experimental and analytical subassemblies can be geographically distributed, allowing researchers to take advantage of the unique capabilities available in different laboratories [2].

While there are many advantages of hybrid simulation, the method faces important challenges, which include: errors introduced in modeling, numerical simulation and experimental testing; the need to estimate bounds on cumulative errors; problems in accurately controlling the actuators that impose displacements or forces; similitude law considerations; and network delays.

B. An Example Hybrid Simulation

The MOST (Multi-sites Online Simulation Test) experiment [3] of a hybrid simulation which uses a steel framework to model the effects of an earthquake on a single-story building. Fig. 3 shows the structure being tested and the three subassemblies of the MOST framework. The structure represents a two-bay single-story steel frame, which mignt be part of a multistory building. MOST is based on a method called multi-site pseudo-dynamic substructure (MS-PSDS) testing [4], in which the structure to be tested is divided into a number of substructures, each of which is physically tested or numerically simulated at the same time at a different location. A simulation coordinator controls the overall experiment and communicates with the test sites and simulation computers.

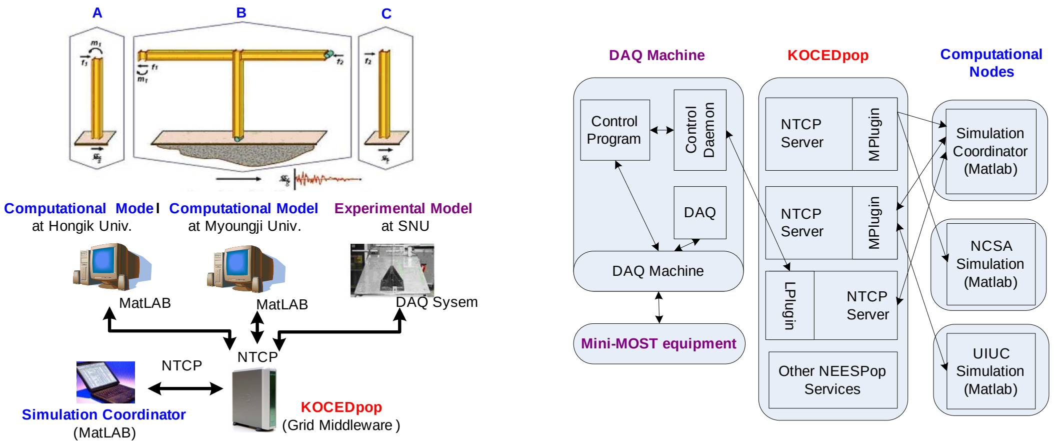
Following the success of the MOST experiment, there was seen to be a need for a smaller, more portable, less expensive experiment that could be installed in an average laboratory to demonstrate the capability of major NEESgrid service components. The mini-MOST experiment consists of a tabletop-sized piece of equipment that emulates the University of Colorado portion of MOST, together with two computational nodes that run on MatLAB numerical simulations. As the software for this new experiment is similar to that used in the MOST experiment, and provides the same services and level of functionality, mini-MOST is a good way for students and researchers to become familiar with the NEESgrid software before conducting a full-scale experiment. More detailed information on mini-MOST and NEESgrid can be found elsewhere [5] [6].

III. Hybrid Simulation on the KOCEDgrid

In order to test the potential of hybrid simulation on the KOCEDgrid, we implemented the mini-MOST experiment using modified NEESgrid software, and developed a collaborative environment for hybrid simulation.

A. The Mini-MOST Experiment on the KOCEDgrid

Fig. 4(a) shows the subassemblies of the mini-MOST framework, consisting of two MatLAB codes and one experimental device. To model these subassemblies, we use four major components: KOCEDpop, computational nodes, a data acquisition (DAQ) system, and the mini-MOST equipment shown in Fig. 4(b). KOCEDpop contains the NTCP (NEESgrid teleoperations control protocol) server which communicates with the DAQ system and the computational node to exchange system status and control information. KOCEDpop also provides services such as a data repository and data streaming. Computational nodes provide the resources on which MatLAB codes can be executed, and the DAQ system controls the mini-MOST equipment and acquires measurements.



(a) Subassemblies of the mini -MOST structure

(b) Block diagram of the mini -MOST simulation

Figure 4. Architecture of the mini-MOST experiment on the KOCEDgrid.

The mini-MOST experiment is managed by the simulation coordinator, which is a MatLAB code that repeatedly issues a set of NTCP proposals, which request actions based on the current state of the simulation measured in terms of quantities such as the strength of a seismic wave and the reaction force in each component. The coordinator collects information on the resulting state of all the substructures, and uses it to determine the next set of commands to send. The simulation coordinator interacts with the DAQ system at SNU, and with MatLab simulation codes at Myoungji and Hongik by means of NTCP protocols.

Each MatLAB code displays experimental results after the simulation is complete. The structural response to the forces that result from the interaction of the seismic wave input and the computational node are presented graphically.

The video and sensor data collected by the DAQ system during 15 seconds of earthquake (1500 steps) are recorded and archived in the comprehensive collaborative framework (CHEF) on KOCEDpop. They can be recalled at any time through the KOCED portal site for further collaborative analysis of the experiment.

The overall objective of this mini-MOST experiment was to show the capability of the various KOCEDgrid service components and extensions within a hybrid simulation. This work also provides a platform for education, training, and outreach.

B. Architecture of the KOCEDgrid for Hybrid Simulation

When a project manager or researcher wants to perform a hybrid experiment using resources on the KOCEDgrid, he or she requests the KOCED administrator to acknowledge the experiment through the KOCED portal, which manages authentication, authorization and collaboration.

Having received the approval of the KOCED authorities, the user begins the experiment through the hybrid portal, which is effectively the user's interface with the experiment. It provides the user with a metadata editor to configure their experiment, control and monitoring application to support telepresence operations, the KRAS (KOCED Resource Allocation Service) to allocate the required resources, and other facilities.

In particular, KRAS provides a grid resource allocation management function in the KOCEDgrid. It arbitrates in the scheduling of resources on the KOCEDgrid, and allocates appropriate facilities, experimental devices and computing systems to the project.

All datums and meta-datums which are created during the setup and execution of an experiment are managed by the data management service. This provides a replay function which allows a researcher to analyze results in detail, and the KOCED Data Registry Service (KDRS) which register the data in the KOCED data center.

The KOCED data center is a repository of completed experiments. All results of experiments on the hybrid portal are moved to the data center after a given period. All users of the KOCEDgrid can search and consult these results through a KOCED portal or a hybrid portal site.

IV. Grid Service Implementation of an Enhanced Hybrid Simulation on the KOCEDgrid

In the current configuration, a hybrid simulation does not have a user-friendly interface. A user can only see the results after the experiment is complete, and these results are in the form of MatLAB data on the local system or web data on CHEF [7], as both. We have therefore developed a real-time remote controller and viewer to extend the user's view and to control the remote equipment. We have also ported the framework of the hybrid simulation to the new version of the grid middleware and to the advanced simulation tool.

A. Real-time Remote Controller and Viewer

When performing a hybrid experiment at a remote site, a user needs a controller for the remote equipment, a

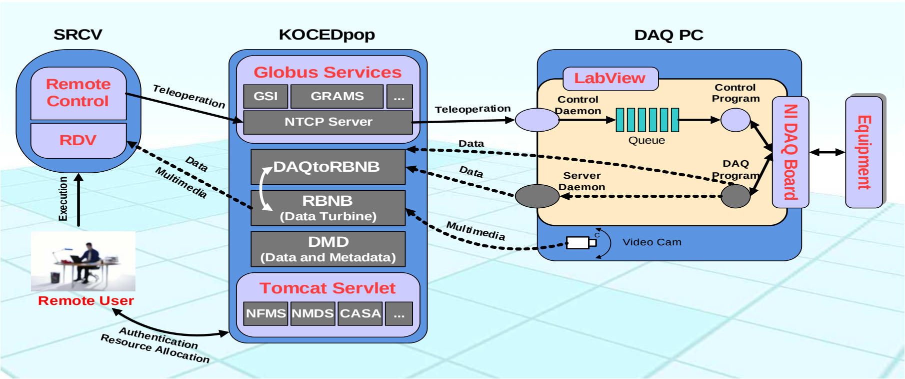


Figure 5. Control and data flow between the SRCV, KOCEDpop, and the DAQ system.

real-time data viewer, and an experimental monitor. These functions can be performed by the SNU remote controller and viewer (SRCV) over the KOCEDgrid.

The SRCV has been written in Java to provide platform independence, and implemented using remote control and telepresence components [8] from the NEESgrid software package: we implemented the remote controller on NTCP (Network Transferred Control Protocol) [9], and the real-time data viewer using RDV (the Real-time Data Viewer) [10] and RBNB (the Ring Buffered Network Bus) [11]. The use of the NEESgrid security infrastructure [12] enables the SRCV to maintain a reliable authorization and security policy that is consistent with other NEESgrid software components.

Fig. 5 shows the control and data flow between the SRCV, the DAQ system of the remote equipment, and NEESgrid components such as the RDV, the RBNB and the DAQtoRBNB transfer software. This figure also shows the flow of control messages: the SRCV sends control messages, such as stepper motor displacements, to the NTCP server, which is part of the NEESgrid software, using NTCP, then the NTCP server re-sends them to the corresponding DAQ system. The Labview control daemon in the DAQ system waits for these control messages, interprets them, and then transfers them to the control program which manages the equipment.

When data is created by the sensors on the equipment, the DAQ program gathers the results and transfers them directly, or via the server daemon, to the NEESgrid DAQtoRBNB software in real time. DAQtoRBNB, which is also written in Java, is the intermediary between the DAQ and the RBNB. Real-time multimedia data streams are also transferred from the DAQ system to the RBNB, which manages data arriving in real time from many sources on multiple channels. Eventually, SRCV processes the data received from the RBNB for presentation to the user. This real-time monitoring function of the SRCV uses the RDV application from the NEESgrid software package.

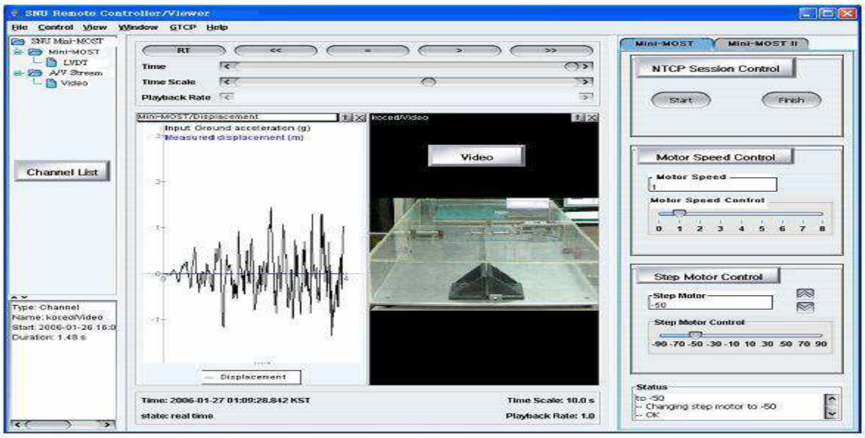
The SRCV presents the user with a combination of software and experimental models which is at least as closely coupled as they are in a real experimental environment. The SRCV also enables overall control and analysis of hybrid experiments, while detecting equipment errors and dealing with them quickly.

Fig. 6 illustrates the user interface to the SRCV. The channel list on the left-hand side shows all the channels through which various types of data are collected in real time. The middle panel displays information about the selected channel. This data is pre-processed to make it more comprehensible and it is displayed in understandable formats: a multimedia view of the remote equipment is provided together with the data values. The control panel managing the remote equipment is on the right-hand side of the interface. This panel provides NTCP session control, motor speed control, stepper motor control, and a status box showing NTCP control messages.

B. Migration to a New Version of the Grid Middleware

OGSA (Open Grid Services Architecture) is an architecture proposal based on the integration of grid and Web service concepts and technologies. The OGSI (Open Grid Service Infrastructure) specification is a companion to OGSA that defines the interfaces and protocols to be used between the various services in a grid environment. So as to provide the interoperability between grids designed using OGSA. One well-known example of the implementation of OGSA/OGSI is the Globus Toolkit maintained by the Globus Alliance. GT3 (the Globus Toolkit Version 3) is the complete reference implementation of OGSA/OGSI, which provides a common set of methods to access various grid services using open standards such as SOAP, WSDL, UDDI, and XML. It also specifies how to find, identify, and utilize new grid services as they become available.

More recently however, the WSRF (Web Service Resource Framework) has been proposed to integrate OGSI-based developments in web service (WS) architectures, dealing in particular with addressing. GT4 (Globus Toolkit Version 4) is based on WSRF, and includes the following new developments:



Type: Channel
Name: koced/Video
Start: 2006-01-26 16:0
Duration: 1.48 s

Time: 2006-01-27 01:09:28.842 KST
state: real time
Time Scale: 10.0 s
Playback Rate: 1.0

Motor Speed: 1
Step Motor: -50

```
to -50
-- Changing step motor to -50
-- OK
```

Figure 6. The user interface of the SNU remote controller and viewer (SRCV).

- Introduction of the WS-Resource concept.
- Better separation of function and exploitation of other WS specifications.
- A broader view of notification, which is a general WS requirement upon which state change notification can be built.

Since GT3 is now at the end of its life, we are migrating all the grid services used in KOCEDpop from GT3 to GT4 [13]. In particular, hybrid simulation uses a control protocol called NTCP for network control. NTCP is now supported by NEESit. However, NTCP was based on GT3, which has been supplanted by GT4. Therefore we ported NTCP to GTCP, which is the WSRF version of NTCP. NTCP and GTCP are very similar: there are some naming differences but the core code is a straight port. This has the advantage of simplicity and compatibility. GTCP exposes two interfaces: a WSRF-compliant service interface used by clients to control remote instruments and simulations, and a plugin interface to facilitate the integration of new back-ends (physical or computational simulation platforms) to the GTCP server. The plugin interface is a Java interface definition that includes methods for each platform-specific action; and a new platform can be introduced by writing a class that implements its interface definition. Following the migration to GT4, hybrid experiments on the KOCEDgrid work well with existing WS-based experimental models. Moreover, GTCP is managed not by NEESit but by the Globus Alliance, thereby increasing the number of hybrid experiments that use GTCP.

Hybrid simulations have commonly used RBNB [11] as a data transfer component. However, its weak point is that it cannot be integrated with the grid security infrastructure (GSI). Moreover, as it is not open-source, we can not enhance its functional design. Therefore, we have adopted the Narada brokering system [14], which is both open-source and also able to integrate with GSI. It is a reliable event brokering system based on an effective publish/subscribe model, and provides all the streaming functions which RBNB can offer. The use of Narada brokering offers following advantages:

- Delivery guarantees are satisfied in the presence of individual or multiple broker and link failures.
- After disconnects an entity can retrieve events missed in the interim.
- Delivery guarantees are satisfied when storage has recovered from a failure.
- Compensation for events that are lost, duplicated or re-ordered in transit over individual links.

C. Migration to OpenSees

Because there is a shortage of modeling tools in the MatLAB environment, we have ported the hybrid experimental environment to OpenSees (Open System for Earthquake Engineering Simulation) [15] which can support fast and sophisticated hybrid simulation.

OpenSees is a software framework for computational simulation of structural and geotechnical systems. It provides a wide range of material and element models for nonlinear analysis, together with solution methods and data processing procedures. OpenSees is open-source software with open-API (Application Program Interface) for researchers to extend it for their own experiment. It is also designed to support parallel processing, which makes it very suitable for implementation on a grid.

OpenSees is object -oriented software based on the finite-element method and is implemented using the C++ programming language. Due to the object-oriented design approach, the OpenSees software architecture is very modular and transparent. Developers and users in different fields can develop and modify the modules that relate to their own area of expertise with relatively little need to refer other modules.

Now we explain how we modeled the mini-MOST experiment in the OpenSees framework, which also analyzes and records the experimental results. The structure of A and B in the mini-MOST experiment are modeled with the finite element software provided by OpenSees, while the structure of C is modeled as an experimental element. The experimental element class was introduced to facilitate the use of OpenSees as a numerical simulation and analysis framework in combination with experimental methods. This class acts as an interface between the OpenSees core and OpenFresco (Open Framework for Experimental Setup and Control) [16]. Since the added experimental elements are simply another type of finite elements which happen to represent the physical components under test, no further modifications to OpenSees were necessary.

OpenFresco was developed to facilitate geographically distributed testing using OpenSees. Of course, it supports the NTCP protocol to support and communicate with MatLab-based experiments. The NTCPExpSite class in OpenFresco allows OpenFresco with to communicate with any server on the KOCEDgrid that uses NTCP. Therefore, the DAQ system at the remote equipment, which includes an NTCP server, can communicate with the OpenSees model through the NTCPExPSite class. The ExpSetup class is defined to be responsible for transforming the experimental element degrees of freedom into actuator degrees of freedom, utilizing the geometry and the kinematics of the loading system.

V. Experimental Variations for the Enhanced Hybrid Simulation on the KOCEDgrid

To give an experimenter more choice, we have increased the number of experimental types of model in the hybrid simulation. Additionally, we have optimized the simulation code to reduce the run-time of the mini-MOST experiment. In its original form, it takes $3\frac{1}{2}$ hours to simulate 15 seconds of earthquake.

A. Increasing the Number of Experimental Models

In a hybrid simulation, it is not necessary to construct and test a model of the complete structure when major damage or unpredictable behavior is only expected in a few structural elements. Therefore, in the Mini-MOST experiment running on the NEESgrid [6], only one experimental node (the mini-MOST device) is used, and two computational nodes emulate the behavior of the other structures. However, the beam of A in the mini-MOST experiment (Fig. 4) is also the key element which is expected to exhibit unpredictable behavior or major damage. Thus, it is appropriate to model it experimentally.

Our new hybrid simulation model is now made up of two experiments and one computational model. A Labview-based mini-MOST device at SNU, an experimental structure (a modified mini-MOST device) at Hongik University and a MatLAB-based computational model at Myoungji University are all connected through the grid network using NTCP. The MatLAB-based simulation coordinator is responsible for integrating the experiment, and the SRCV controls the overall experiment.

Having replaced a computational model with an experimental model, we expect some improvement in accuracy because the experimental model can reflect non-linear physical properties such as solidity or elasticity more exactly.

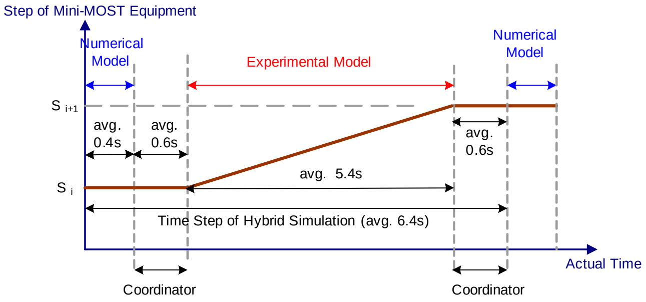


Figure 7. Time sequence for one load step of the mini-MOST experiment with one physical device.

However, using more experimental models increases the time required to perform an experiment, because it usually takes much longer to get results from an experimental model. Fig. 7 tracks the actual simulation time for one load step of the mini-MOST experiment with one physical device. We can see that the mini-MOST device takes longer than either the numerical models or the coordinator, by a factor of nearly ten. Therefore, we need to consider the tradeoff between accuracy and execution time carefully when designing a hybrid simulation.

On the KOCEDgrid, there is a need to vary the combination of physical devices and numerical codes according to various experimental requirements and environments. Thus, KOCED aims to provide a hybrid simulation with a flexible combination of physical devices and computational nodes, because it can increase the scalability of hybrid simulations, and makes more efficient use of the available physical devices.

B. Increasing the Speed of the Hybrid Simulation

In the mini-MOST experiment on the NEESgrid, it takes 1500 steps to simulate 15 seconds of earthquake, and computation takes 8 seconds per step, giving an elapsed time of about $3\frac{1}{2}$ hours on the NEESgrid. But the mini-MOST experiment at SNU only takes 5 seconds per step, and less than 2 hours to perform a full simulation, since we used a KOCEDpop server with a higher specification than the recommended one of the NEESgrid [5].

However, 5 seconds per step still seems too slow, even though it includes the computational overhead at each site and the network delay. Therefore, we analyzed the coordinating code which controls the overall flow of an experiment, which was originally written for reliability and robustness.

Fig. 8 shows the overall flow of a typical hybrid simulation. Firstly, connections are established between the computational and experimental models, and each model is initialized. Then stiffness matrices are built through basic connection tests. Finally, the actual pseudo-dynamic structure is simulated, with a loop through the simulation code at each step. In each loop, the simulation coordinator

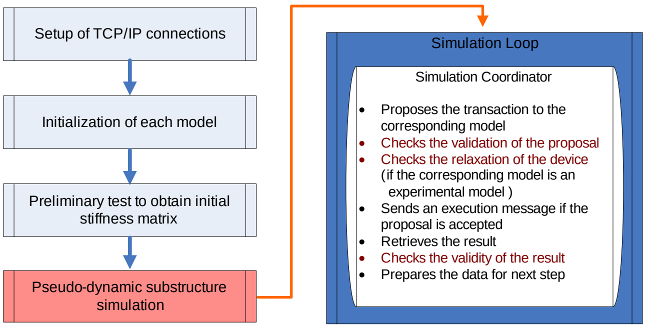

Setup of TCP/IP connections
Initialization of each model
Preliminary test to obtain initial stiffness matrix
Pseudo-dynamic substructure simulation
Simulation Loop
Simulation Coordinator
Proposes the transaction to the corresponding model
Checks the validation of the proposal
Checks the relaxation of the device (if the corresponding model is an experimental model)
Sends an execution message if the proposal is accepted
Retrieves the result
Checks the validity of the result
Prepares the data for next step

Figure 8. Analysis of the coordination code for the hybrid simulation.

executes a set of tasks related to the computational and experimental models. It proposes the next transaction to the appropriate model, and sends an execution message if the proposal is accepted. If the model is experimental, the coordinator checks the relaxation of the physical device before sending the execution message. On receiving the result, it assessed its validity using check routines, and if it is acceptable it then goes on to the next step.

This conservative coding style, using a check and go method, is very robust, but it is only necessary if the integrity of the atomic operations is a very important issue. In hybrid simulations for general civil engineering, such a high level of reliability is not required. Therefore, the experiment could be accelerated by minimizing the checks on each transaction. This can save 35% of the total time. It might even be beneficial to allow the coordinator to choose the appropriate tradeoff between reliability and execution speed for a particular experiment.

VI. Conclusions

The KOCEDgrid is a high-speed network service which supports collaboration by linking high-performance computers, large databases, experimental equipment, and other construction engineering resources located across Korea. We have presented some recent developments in technologies used by KOCEDgrid to support hybrid simulation. A new real-time controller and viewer provides a user-friendly interface for hybrid simulation, so that an experiment on a remote site can be easily monitored and controlled throughout an entire simulation. KOCEDgrid has been migrated to GT4 and now uses GTCP to communicate. We have developed GTCP plug-ins for hybrid simulation models. To give users a more flexible simulation environment, we have extended our model to include computational nodes written in OpenSees. The new OpenSees hybrid simulation model was successfully built on new KOCEDgrid services based on GT4 and GTCP, and enables users to conduct experiments more flexibly with more than one piece of experimental equipment. It also eliminates unnecessary communication overheads caused by MatLAB and NTCP, and makes real-time hybrid simulations possible. KOCED will continue to evolve to provide a more accurate, a more efficient, and a more user-friendly hybrid simulation environment. Other interesting research topics such as the security of hybrid simulations on KOCEDgrid systems form our ongoing work.

References

[1] D. Noh, S. Yi, H. Kwon, J. Lee, T. Jeong and H. Shin, A case study: hybrid simulation on the KOCEDgrid system, 2006, IEEE International Conference on Hybrid Information Technology, Vol. 1, 203-209.

[2] S. Campbell and B. Stojadinovic, A system for simultaneous pseudodynamic testing of multiple substructures, 1998, 6th US National Conference on Earthquake Engineering.

[3] B.F. Spencer, A. Elnashai, N. Nakata, H. Saliem and G. Yang, The MOST experiment: earthquake engineering on the grid, 2004, NEESgrid.

[4] E. Watanabe, K. Sugira, K. Nagata, T. Yamaguchi and K. Niwa, Multi-phase interaction testing systems by means of the Internet, 1999, 1st International Conference on Advances in Structural Engineering and Mechanics, 43-54.

[5] C. Kesselman, R. Butler, I. Foster, J. Futrelle, D. Marcusiu, S. Gulipalli, L. Pearlman and C. Severance, NEESgrid system architecture, 2003, NEESgrid.

[6] N. Nakata, G. Yang and B. F. Spencer, System requirements for mini-MOST experiment, Nov. 2004, UIUC.

[7] Comprehensive collaborative framework (CHEF) project website. Online at http://chefproject.org

[8] C. Kesselman, L. Pearlman and G. Mehta, Design for NEESgrid telepresence referral and streaming data services, 2003, NEESgrid.

[9] L. Pearlman, M. D'Arcy, E. Johnson, C. Kesselman and P. Plaszczak, NEESgrid teleoperation control protocol (NTCP), 2003, NEESgrid.

[10] Real-time data viewer (RDV) website.
Online at http://it.nees.org/software/rdv/index.php

[11] Ring buffered network bus (RBNB) website.
Online at http://rbnb.creare.com/rbnb/WP/WebTurbine.html

[12] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, A security architecture for computational grids, 1998, 5th ACM Conference on Computer and Communications Security, 83-91.

[13] Globus Toolkit version 4.0 website.
Online at http://www.globus.org/toolkit/docs/4.0

[14] G. Fox and S. Pallickara, Deploying the Narada Brokering substrate in aiding efficient web and grid service interactions, 2005, Proceedings of the IEEE, Special Issue on Grid Computing, 93(3), 564-577.

[15] Open system for earthquake engineering simulation (OpenSees) website.
Online at http://opensees.berkeley.edu

[16] Y. Takahashi and G. L. Fenves, Software framework for distributed experimental-computational simulation of structural systems, 2006, Earthquake Engineering and Structural Dynamics, 35(3), 267-291.

Dong Kun Noh received B.S., M.S., and Ph.D. degrees in EECS from Seoul National University, Seoul, Korea in 2000, 2002, and 2007, respectively. He is currently a postdoctoral researcher in Dept. of Computer Science at the University of Illinois at Urbana-Champaign. His current research interests are in the field of mobile computing and communication with emphasis on energy management, QoS control and service discovery architecture on wireless ad-hoc and sensor network.