STRUCTURAL SYSTEM SIMULATION USING GRID-COMPUTING FRAMEWORK

by Data Iranata[a] and Radityo Anggoro[b]

ABSTRACT

A multi-level modeling and simulation method of structural system using grid-computing framework is proposed in this paper. Two levels of parallel processing will be involved in this framework: (1) multiple locally distributed computing environments connected by the local network to form (2) a grid-based cluster-to-cluster distributed computing environment. To successfully perform the simulations, a large-scale structural system is decomposed into the simulations of a simplified global model and several detailed component models with various scales. These correlated multi-scale simulation tasks are distributed amongst clusters and connected together in a multi-level modeling and simulation method and then coordinated over the internet. This paper also presents the development of a grid-computing software framework that can support the proposed simulation approach. The architectural design of the program also allows the integration of several multi-scale models to be clients and servers under a single platform. Additionally, the comparison result between proposed method and assumed exact solution show that the proposed simulation method is appropriate to simulate the response of the structural systems.

KEYWORDS: *computer simulation; distributed computing; grid computing; internet; structural system.*

INTRODUCTION

Advances in computer technology and numerical methods have allowed the simulation of engineering problems that traditionally have been addressed via experimentation and theoretical models. Some industries have been able to design sophisticated engineered systems based solely on computer simulation. In addition, many complex phenomena, such as airplane crashes and car accidents, can already be analyzed by computer simulations. In the context of structural engineering, using computer simulation to realistically represent the behavior of structural systems in detail in various situations, such as the global response and the detailed damage of a structure during a major earthquake, is also a goal which must be achieved by engineers.

In comparison with other engineering systems, structural engineering systems such as bridges and buildings usually are large-scale and contain the effects of various structural components and materials at various scales. Therefore, to successfully perform the simulation of structural systems, the structural model must be able to capture the behaviors of the global system and all the detailed local mechanisms. Modeling the whole structural system in every detail using very fine meshes as the way to simulate the response of structural components is an approach to achieve this. However, the resulting models would be enormous and hard to handle efficiently using current computational power.

High-performance computing is the key to the investigation of problems that are computationally intensive, such as the simulation of structural engineering systems. At the dawn of parallel computing, shared memory machines dominated. In this kind of hardware architecture, communication between processes was irrelevant. Thus, few modifications of the finite element analysis algorithms were necessitated by the computational hardware. As distributed memory computers and clusters of networked workstations were introduced, communication times between processes became significant. This brought about innovations in the finite element analysis using domain decomposition algorithms by dividing tasks into a few loosely coupled subtasks to minimize the communication penalty. The well-known domain decomposition methods include sub-structuring (static condensation, dynamic reduction), parallel central difference algorithm by Hajjar and Abbel,[1] the Iterative Group Implicit (IGI) algorithm by Modak and Sotelino,[2] Finite Element Tearing and Interconnecting (FETI) by Farhat and Roux,[3] and the nonlinear sub-structuring algorithm by Chen and Archer.[4]

Presently, grid computing is being perceived as the most promising avenue to achieve further computational power. Virtually an infinite number of machines can be connected via the internet, theoretically leading to unlimited computational capabilities. However, unless each subtask is highly independent, internet communication can render impossible any possible gain in efficiency from a grid-based distributed application. Depending on the time of the day, the traffic in the internet can be such that no timely communication is possible. In addition, in all of the above domain decomposition methods, there is a limitation to the number of processors which produce efficiency. Therefore, extending those methods directly for running on a platform which consists of massive machines connected by comparatively slow, over-utilized communication channels to efficiently perform a simulation is not a trivial task. New methodologies are necessary that minimize the amount of data being communicated and the frequency of data communication.

Since decomposing a complex model and distributing the decomposed tasks to all available machines will not work in a grid-based computing environment due to the

---

[a] Assistant Professor, Department of Civil Engineering, Sepuluh Nopember Institute of Technology (ITS), ITS Campus, Sukolilo, Surabaya 60111, Indonesia.

[b] Assistant Professor, Department of Informatics, Sepuluh Nopember Institute of Technology (ITS), ITS Campus, Sukolilo, Surabaya 60111, Indonesia.

Note. The manuscript for this paper was submitted for review and possible publication on February 09, 2009; approved on June 24, 2009. Discussions open until January 2011. This paper is part of the ITS Journal of Civil Engineering, Vol. 29, No.2, November 2009. © ITS Journal of Civil Engineering, ISSN 2086-1206/2009.

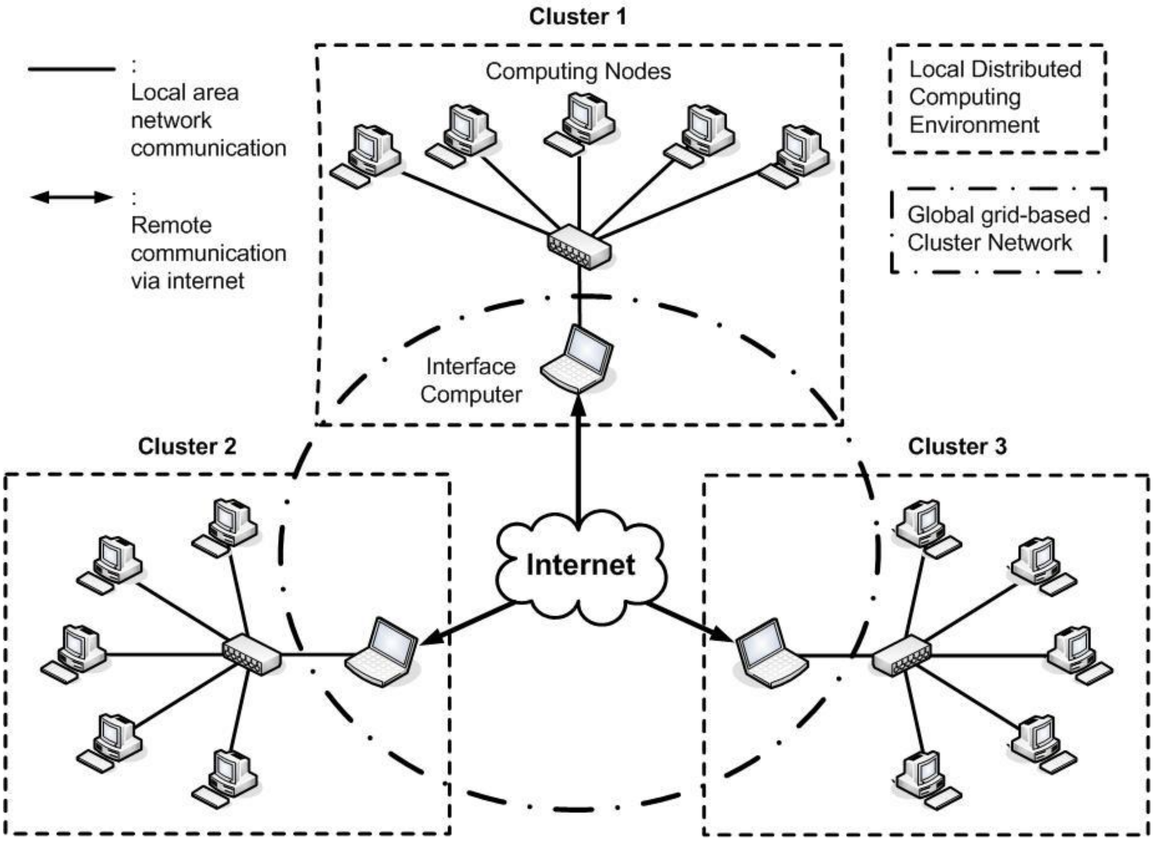


Fig. 1. Grid-based cluster-to-cluster distributed computing environment.

internet-imposed communication time obstacle, a new grid-based simulation methodology for the simulation of structural engineering systems is proposed and investigated in this paper. Two levels of parallel processing will be involved in this framework: (1) multiple locally distributed computing environments connected by the local network to form (2) a grid-based cluster-to-cluster distributed computing environment connected by the internet. To successfully perform the structural system simulation in this computing environment, a large-scale structural simulation task is decomposed into the simulations of a simplified global model and several detailed component models at various scales. These correlated multi-scale simulation tasks are distributed amongst clusters connected together via internet to form a multi-level modeling hierarchy, and then these simulations, in separated clusters, coordinate with each other over the internet to complete the computer simulation of a whole structural system.

RESEARCH SIGNIFICANCE

This paper presents a multi-level modeling and simulation method of structural system using grid-based cluster-to-cluster distributed computing environment. The architectural design of the program framework allows the integration of several multi-scale models as clients and servers under a single platform. Such integration will facilitate a more realistic simulation of a structural system.

GRID-COMPUTING FRAMEWORK

A simple one-level grid computing environment can be described as a massive collection of heterogeneous machines connected by comparatively slow, over-utilized communication channels. Applications for running in this kind of environment must assume any communication is over the internet, no matter whether the machine to be communicated with is local or remote in fact. Thus, this kind of framework is only suitable for distributed computing problems for which each subtask is independent or highly independent, so that time for internet communication among massive machines is not necessary or is insignificant. For existing distributed finite element analysis algorithms, communication time is significant and there is a limitation to the number of processors which can achieve efficiency. Therefore, there is no way to extend existing algorithms to be applicable in a simple one-level grid computing environment.

For reducing the internet communication bottleneck by organizing the hardware configuration of a grid-based environment, a cluster-to-cluster distributed computing framework, as shown conceptually in Fig. 1, is proposed. This framework involves two levels of parallel processing: (1) multiple locally distributed computing environments connected by the local network to form (2) a grid-based cluster-to-cluster distributed computing environment connected by the internet. In fact, each of the clusters in the framework can be a distributed memory supercomputer, a shared memory supercomputer, or just a personal computer.

Most communications are between computing nodes within a cluster like a traditional cluster computing environment. Only the messages required to be exchanged between clusters are communicated between the interface computers over the internet. Due to the introduction of two-level parallelism, the modeling approach and computational procedure for structural simulations must be revised from the architectural level, to be applicable in the proposed cluster-to-cluster distributed computing environment. The new simulation method must minimize the amount of data being communicated and the frequency of data communication over the internet, and keep most of the required communications within the same local area network.

MULTI-LEVEL MODELING AND SIMULATION METHOD

In the simulation of structural engineering systems, finite element analyses are often carried out to simulate the behavior of the system as a whole. In this type of analysis, simplistic models are often used for structural components, such as beam and column. Although the global responses of structural systems can be simulated in this way, the detailed responses of their components cannot be obtained using simplistic models. On the other hand, much research has been conducted in the analysis of structural components. In this type of analysis, the structural components are modeled in very fine mesh to get detailed responses. However, these models are in isolation with no consideration of the relationship between their behavior and that of the rest of the system. Therefore, there is a need for the integration of these two levels of knowledge under a single platform. Such integration will permit the realization of more realistic simulation of structural engineering systems.

Modeling the whole structural system in every detail using very fine meshes as the way to simulate the responses of structural components is an approach to achieve this. However, the resulting models would be enormous and hard to handle efficiently using current computational power and analysis techniques. Another

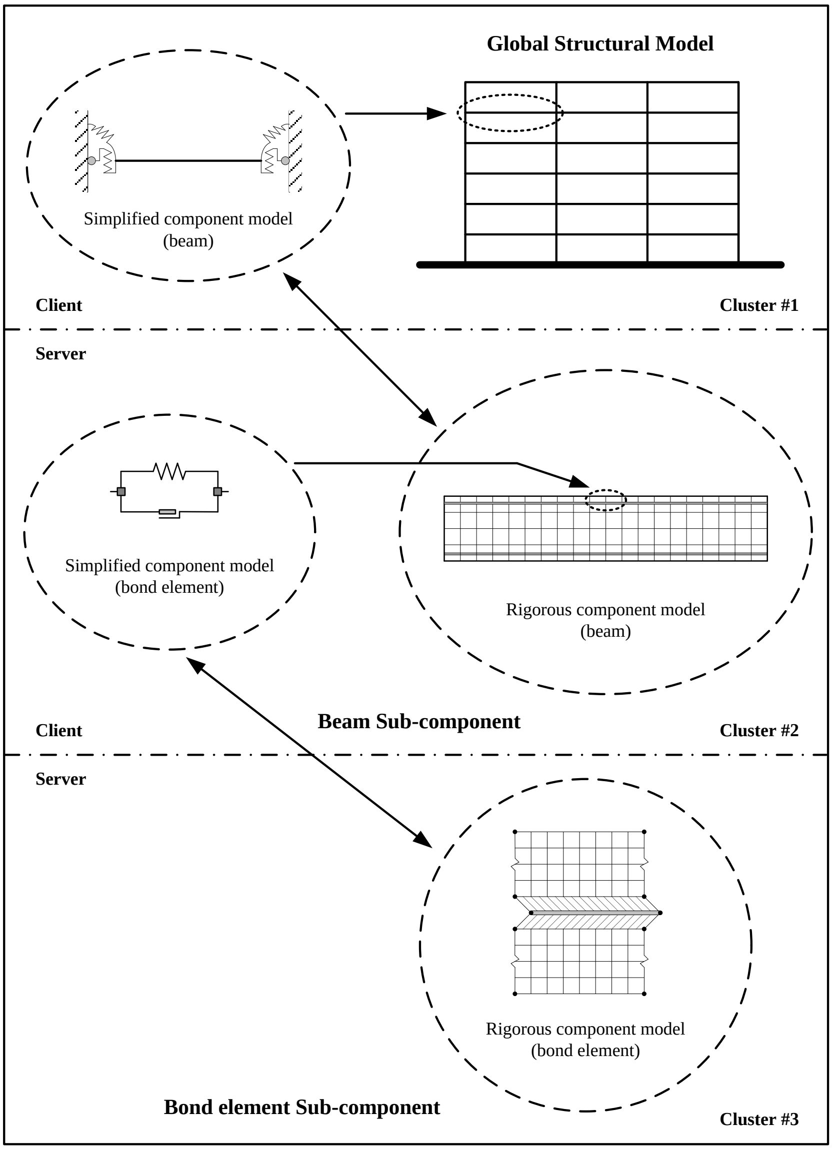


Fig. 2. Multi-level modeling and simulation method of structural system.

possible approach is proposed by using independent models of various scales for simulating the global system of a structure and its detailed components in the ways traditional structural analyses are conducted, and then trying to integrate and correlate the simulations of these multi-scale models to achieve realistic simulation of the whole structure. In this way, a detailed simulation of the whole system can be decomposed into several detailed simulations of its individual components.

To achieve this, a structural component must be modeled twice at two different scales. A simplified model to reside in a macro/global system to obtain the forces applied by the rest of the system, and a rigorous model to be analyzed in isolation to obtain its detailed responses and behavior. The two separated models, in fact, represent the same structural component in the real system, thus synchronization is required during the simulation process through communication and coordination. In addition, a rigorous model of a component can contain the simplified models of other components to result in a multi-level modeling architecture, which is shown conceptually in Fig. 2.

Tremendous computational power is needed to perform such a simulation. The proposed multi-level modeling and simulation method can perfectly fit in and run in the proposed cluster-to-cluster distributed computing environment by distributing the global system model and all the rigorous component models in a hierarchy amongst clusters and interacting over the internet. Only the information required for synchronizing corresponding simplified component models and rigorous component models must be communicated over the internet. Thus, the communication time obstacle is removed at the architectural level by the proposed method with the proposed computing environment. The introduction of a simplified component model allows the state-update and state-query tasks of global analysis to be performed locally within the same (client) cluster. When the state of a structural component is desired by the analysis, its local simplified component model is used. Likewise, to update the state of a structural component, its local simplified component model is updated. In essence, a simplified component model acts locally as agent to the global analysis in the client cluster on behalf of the server-based rigorous component model. Any pair of corresponding models (simplified and rigorous) only needs to sporadically exchange a small amount of information over the internet for synchronization. In this way, the internet communication overhead can be avoided.

To successfully perform the simulation of structural systems by the proposed multi-level hierarchical modeling approach, there are two requirements to be achieved. First, the analysis algorithm must be able to handle the transition between distinct scales by translating a component's global behavior obtained from its macro-level model to actions on its micro-level model and a component's localized effects obtained from its micro-level model to the behavior of its macro-level model in the global system. Thus, mechanisms are required for extracting the values of displacements and stresses in a simplified component model, and then appropriately translating these quantities to the known values in the analysis of the corresponding rigorous component model, and vice versa.

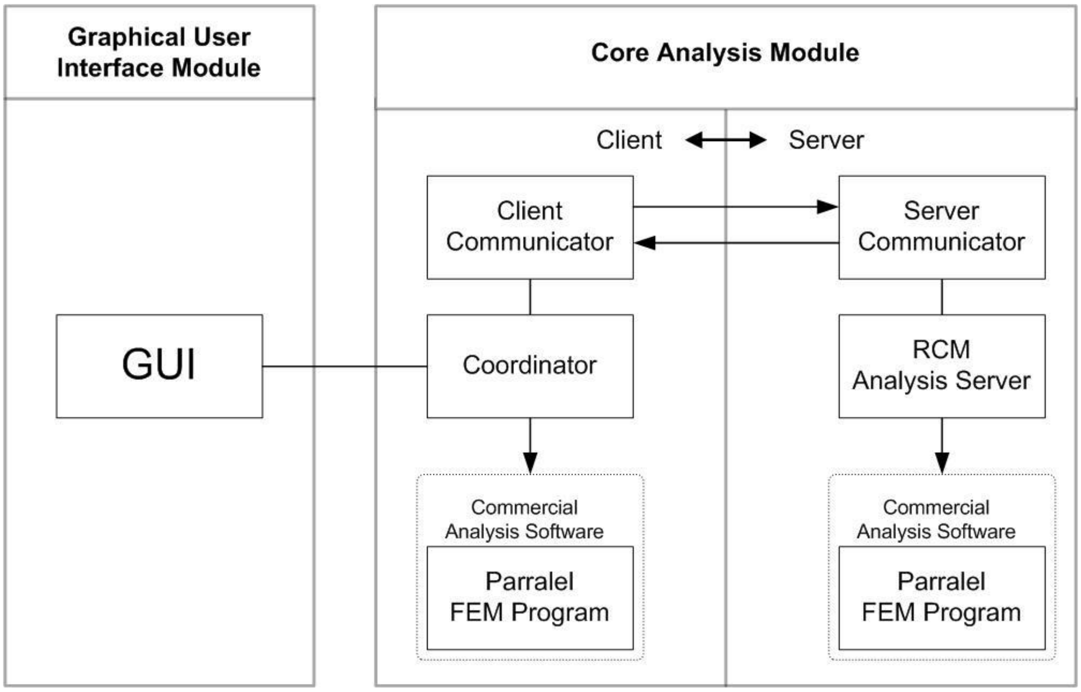


Fig. 3. Software framework of the proposed grid-based simulation.

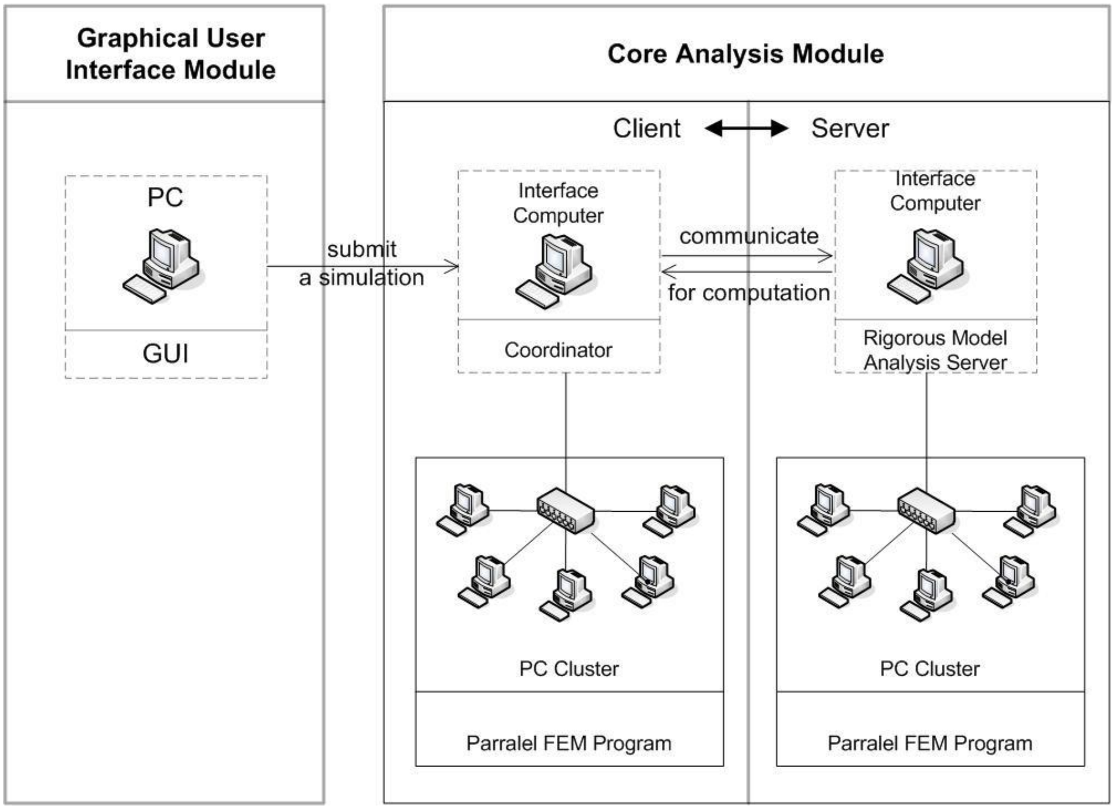


Fig. 4. Hardware configuration of the proposed grid-based simulation.

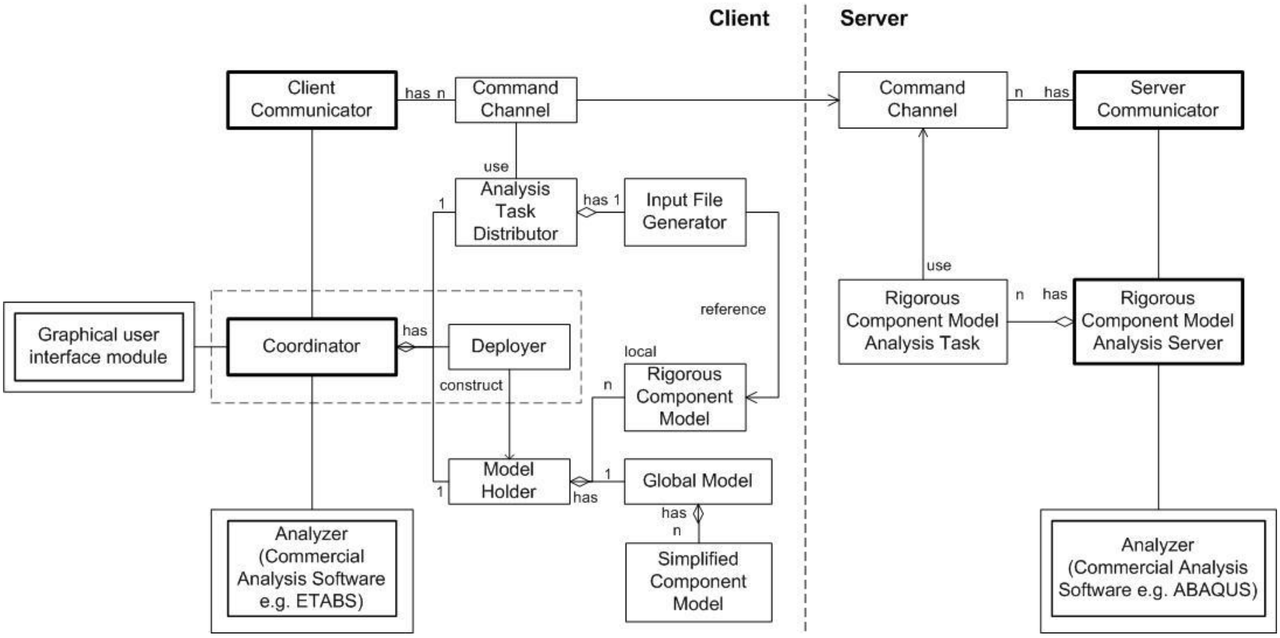


Fig. 5. Core analysis modul.

Second, a simplified component model must accurately describe the global behavior of the component. Otherwise, the resulting global actions on it would be invalid. Many researchers have developed simplified models that can generally achieve the goal for various structural components, such as the beam component by D'ambrisi and Fillippou,[5] column component by Chen and Iranata,[6] and beam-column component by Lowes and Altoonash.[7] However, in some occasions, the original simplified component model may become invalid to represent the behavior of its corresponding rigorous component model. For this reason, a rigorous model must check the state of its corresponding simplified model periodically, and calibrate or update it when necessary.

SYSTEM DESIGN AND IMPLEMENTATION

System Architecture

The proposed cluster-to-cluster distributed computing environment will require a modified client-server module

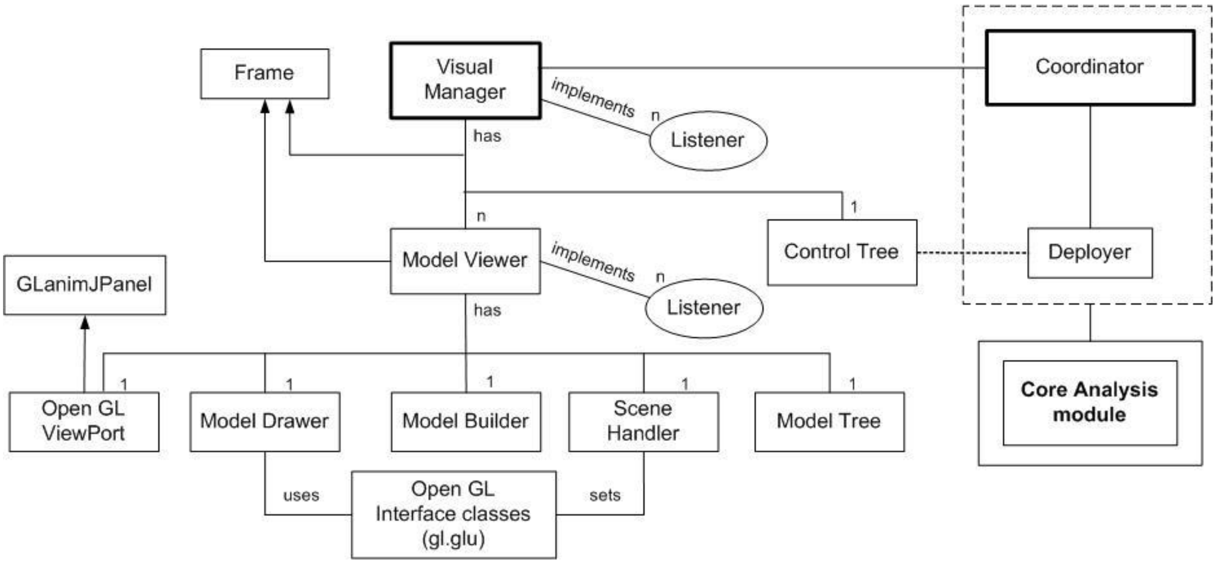


Fig. 6. Graphical user interface modul.

with a formatted architecture to perform the proposed multi-level modeling and simulation. The software framework consists of two modules, as shown in Fig. 3. They are the core analysis module and the graphical user interface module. Both modules are implemented using Java, a platform-independent, net infrastructure language, so that the system can run on any (interface) computer connected to the internet.

The hardware configuration is shown in Fig. 4. In the same way, the hardware configuration is divided into two parts. The graphical user interface module is installed on a personal computer (PC), and the core analysis module is orchestrated by one client cluster and at least one server cluster. As the figure shows, in the core analysis module, each cluster uses an interface computer to communicate, and the analyzing work is thereby administered to its own cluster system.

Core Analysis Module

The core analysis module is a client-server-based distributed system for handling all the information translation, internet communication, and synchronization coordination between a simplified component on the client and a rigorous component on the server. The object model of the core analysis module is shown in Fig. 5. The interactions between the client and server are archived by the Command Channel objects, which are created by the Client Communicator class and Server Communicator class. On the Client side, the Coordinator class has three subclasses to handle the simulation processes. The Deployer class reads the simulation configurations and stores the data of the digital models into the Model Holder class. The Model Holder class stores a global model, simplified component models and rigorous component models. At this point, the stored object of the rigorous component model provides several functions to enable other objects to access their information.

The Analysis Task Distributor class submits the analysis tasks to the assigned clusters and waits for their analyzed results to return from the server side. Basically, a client can distribute the analysis tasks to more than one server. On the other hand, a server is able to receive more then one analysis task at a time. Therefore, on the server side, the Rigorous Component Model Analysis Server class can create several Rigorous Component Model Analysis Task objects, which control the analysis program to analyze the task, and queue specific tasks for execution. The Analyzer classes, which handle the integrated analysis programs, have a synchronization property allowing them to establish the queuing function of the Rigorous Component Model Analysis Server class.

Graphical User Interface Module

The graphical user interface module provides a 3D graphical and interactive interface to users for creating, managing, and viewing simulation results from a multi-level model, which is distributed amongst clusters, from the top-most interface computer. The object model of the graphical user interface module is shown in Fig. 6. The Visual Manager class has a Control Tree object. And it can create many Model Viewer objects depending on how many models need to be displayed.

The Control Tree class accesses the Deployer class sub-program in the Coordinator class of the core analysis module. It obtains the simulation configurations stored in the Deployer class and shows them on the management window. The Model Viewer class uses the Model Drawer class, the Model Builder class and the Scene Handler class to create and interactively manipulate the visualized objects of the model. These three classes adapt the GL4Java API[8] to handle the virtual scenes.

Integration of Commercial Analysis Software

There are many popular commercial structural analysis software products available, such as ABAQUS and LS_DYNA. Because of their outstanding performance and the many different features that they offer, many engineers and researchers from various disciplines regularly use them as analyzing tools. In the past decade, with increasing demands for computing power, these software venders have used the parallelization technology to improve the computational capacity and efficiency of their products. Therefore, considering the benefits of the software analysis performance, and its general familiarity to the public, we have chosen to use ABAQUS for each cluster of the proposed grid-based computing environment. In other words, if a cluster system has the ABAQUS software program installed, it can easily become one of the analysis servers of the proposed platform, simply by installing the server-side program of the core analysis module, which will be introduced in the later section. Therefore, those who use the proposed platform can create new system or component models through the pre-process interface of ABAQUS. After the model has been analyzed, the post-process interface can then show the visualized results.

There is no need to mention that different commercial analysis software programs have their own unique and specific input file format. Therefore, some rules should be established for identifying the objects with the proposed software system, in order to enable the platform to recognize the data format of different analysis software programs in a dynamic and automatic fashion, without needing to modify the code of the system.

There are two interface requirements that need to be established before the commercial analysis software can be integrated into the proposed analysis platform. First, the system should be able to recognize the content of the input file of the commercial analysis software and convert it to the corresponding object types of the proposed software system. These objects types include nodes, elements, forces, boundary conditions, etc. This step is necessary because the global model and the rigorous component models of the whole simulation problem will be displayed on the graphical user interface of this system. The second requirement is that the system should be able to read the results from each of the models studied by the commercial analysis software. Therefore, after the global model is completed, the nodal displacements in the global model can then be extracted from the simplified component model. These extracted values of the displacements are then translated to become the initial boundary conditions of the corresponding rigorous component model.

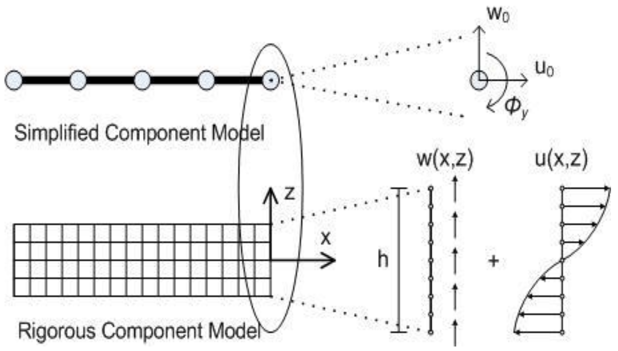


Fig. 7. The extracting and translating mechanism for nodal displacement.

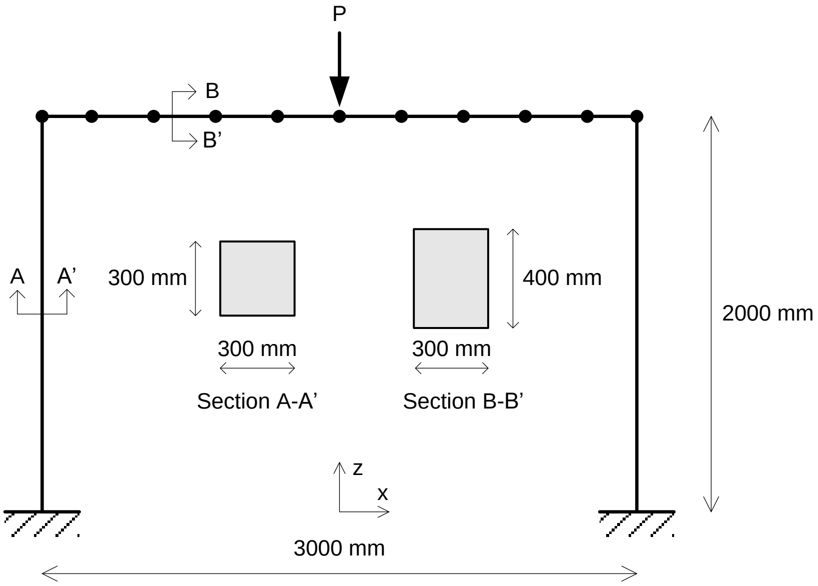


Fig. 8. A simple portal frame of reinforced concrete structure.

The Synchronization of the Multi-Level Hierarchical Models

In order to synchronize the multi-level hierarchical analysis of structural systems with the proposed modeling approach, the analysis algorithm must be able to handle the transition between distinct scales by translating a component's global behavior, obtained from its macro-level model, to actions on its micro-level model. Likewise, a component's localized effects obtained from its micro-level model must be coordinated with the behavior of the macro-level model in the global system. Thus, mechanisms are required for extracting the values of displacements in a simplified component model, and then appropriately translating these quantities to the known values in the analysis of the corresponding rigorous component model.

One of the ways to achieve the extracting and translating mechanism is using beam theory, as presented by Ghugal and Shimpi.[9] The third order beam theory including the transverse shear strain and nonlinear axial stress as proposed by Krishnamurty[10] is adopted for this study. In this theory, the parabolic transverse shear stress distribution through the cross section of the beam can be obtained using constitutive relations. The displacement field of the theory is as follows:

$$u(x,z) = u_0 + z\left[\phi_y + \frac{4}{3}\left(\frac{z}{h}\right)^2\left(\phi_y + \frac{dw}{dx}\right)\right] \qquad (1)$$

$$w(x,z) \cong w(x) \qquad (2)$$

Where: u and w are the nodal displacement at the x and z direction, respectively; ϕ_y is the rotation of the cross section of the beam at the neutral layer in the y direction, and h is the height of the beam.

Fig. 7 shows the extracting and translating mechanism

Table 1. Comparison results of stress (S11) component.

Section	S11 (MPa)		Difference (%)	Average Difference (%)
	Proposed method	Assumed exact solution		
a-a	0.214537	0.218613	-1.8645	**1.4364**
	0.174549	0.172570	1.1468	
	0.132376	0.129971	1.8504	
	0.100514	0.102853	-2.2736	
	0.071948	0.075449	-4.6401	
	0.035664	0.034018	**4.8386**	
	-0.012539	-0.012102	3.6117	
	-0.083751	-0.085201	-1.7016	
	-0.285338	-0.286330	-0.3465	
b-b	0.390498	0.385965	1.1745	
	0.289118	0.279953	3.2738	
	0.194533	0.190468	2.1342	
	0.141548	0.135030	4.8271	
	0.101053	0.099308	1.7570	
	0.053071	0.053240	-0.3163	
	-0.017291	-0.016613	4.0775	
	-0.164069	-0.165119	-0.6359	
	-0.658225	-0.656149	0.3164	
c-c	-0.712487	-0.712574	-0.0122	
	-0.574473	-0.574600	-0.0221	
	-0.396119	-0.396272	-0.0386	
	-0.221023	-0.221184	-0.0728	
	-0.045641	-0.045800	-0.3474	
	0.129845	0.129686	0.1226	
	0.305139	0.304973	0.0544	
	0.483476	0.483290	0.0385	
	0.621156	0.620503	0.1052	
d-d	-2.056910	-2.085180	-1.3558	
	-1.427080	-1.420140	0.4887	
	-0.943963	-0.934936	0.9655	
	-0.490245	-0.486144	0.8436	
	-0.044393	-0.044540	-0.3291	
	0.401531	0.397147	1.1039	
	0.855851	0.846506	1.1039	
	1.340510	1.333260	0.5438	
	2.057640	1.990420	3.3772	

from single nodal displacement in the simplified component model into several nodal displacements in the corresponding plane in the rigorous component model. This figure illustrates the mechanism in the two-dimensional plane. Each single nodal of simplified component model has three deformation components, they are: vertical displacement, horizontal displacement, and rotation. Those displacements should be extracted and translated into the corresponding plane of the rigorous component model. Since it is modeled as the continuum element, the nodal displacement of the rigorous component model only consists of vertical and horizontal displacement. The vertical displacement of the simplified component model can be directly translated into several nodal displacements in the corresponding plane of the rigorous component model since it is not a function of the z-axis. But for the horizontal displacement which is function of the z-axis, the single nodal displacement of the simplified component model should be extracted by using the third order beam theory equation into several nodal displacements and translated into corresponding plane of rigorous component model.

A simple portal frame of reinforced concrete structure, as shown in Fig. 8, is used to help analyze the accuracy of the proposed simulation method. A downward concentrated load is applied to the middle span of the beam segment of the global frame model. Furthermore, the beam component is picked and analyze as rigorous component model. Thus, the nodal displacement one beam simplified component model extracted and translated into several nodal in the corresponding plane of the beam rigorous component model. The analysis results obtained using the proposed simulation method is then compared to the assumed exact solution, obtained by analyzing a detailed portal frame model, as shown in Fig. 9. To check the accuracy level, the stress (S11) component of four section of the beam component is taken, e.g. section a-a, b-b, c-c, and d-d. The comparison results of each section are presented in Table 1. The comparison results show that maximum and average difference between proposed method and assumed exact solution is only about 4.84 and 1.44 percent, respectively. Therefore, based on the comparison results, it assesses the proposed simulation method is appropriate to simulate the response of the structural systems.

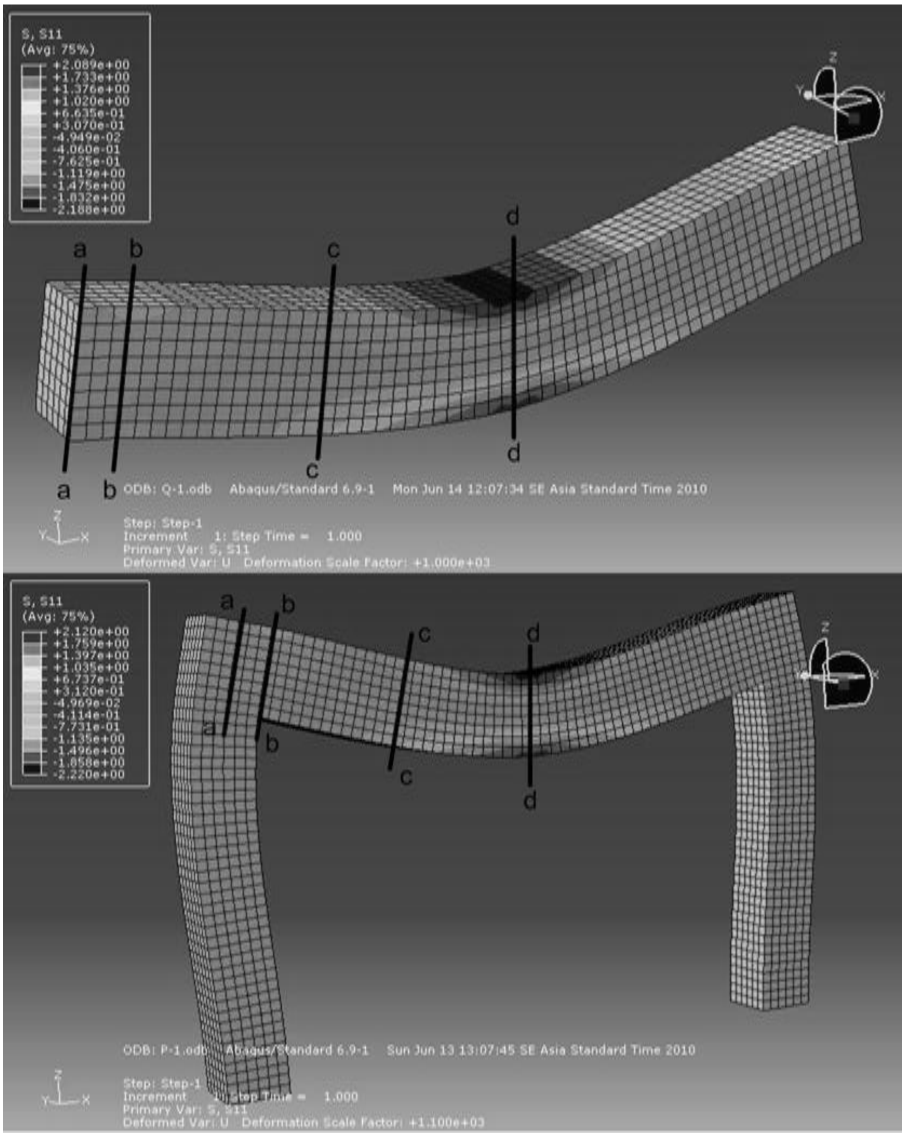


Fig. 9. Comparison result between proposed simulation results (top) and assumed exact solution (bottom).

Table 2. Reinforced concrete material properties.

Material properties	Symbol	Value
Modulus elasticity	E	23500 MPa
Poisson's ratio	υ	0.18
Density	w	2400 kg/m^3

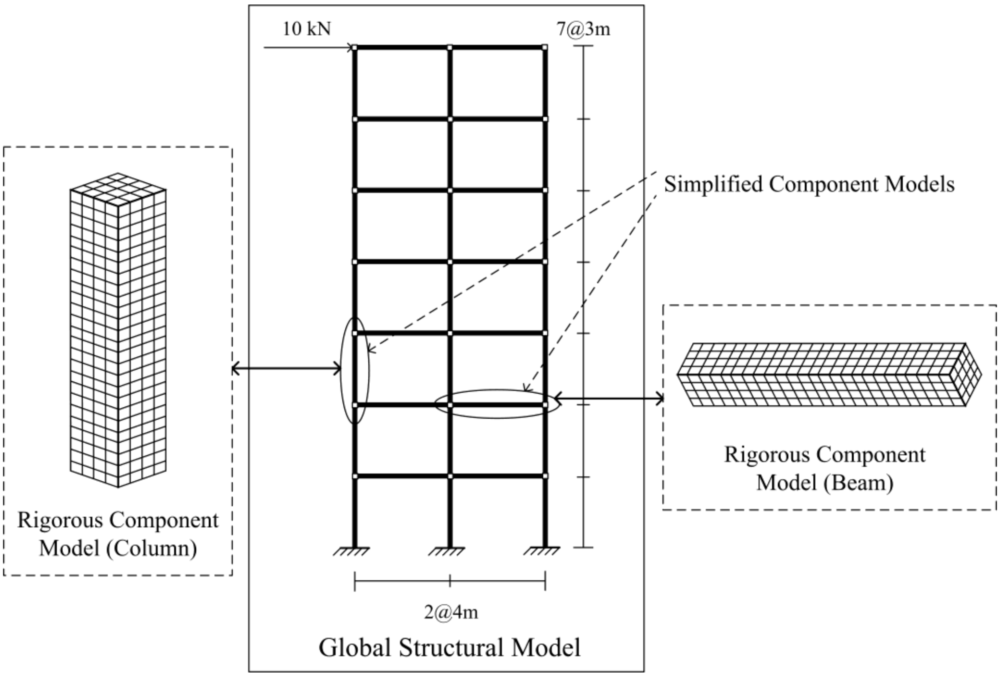
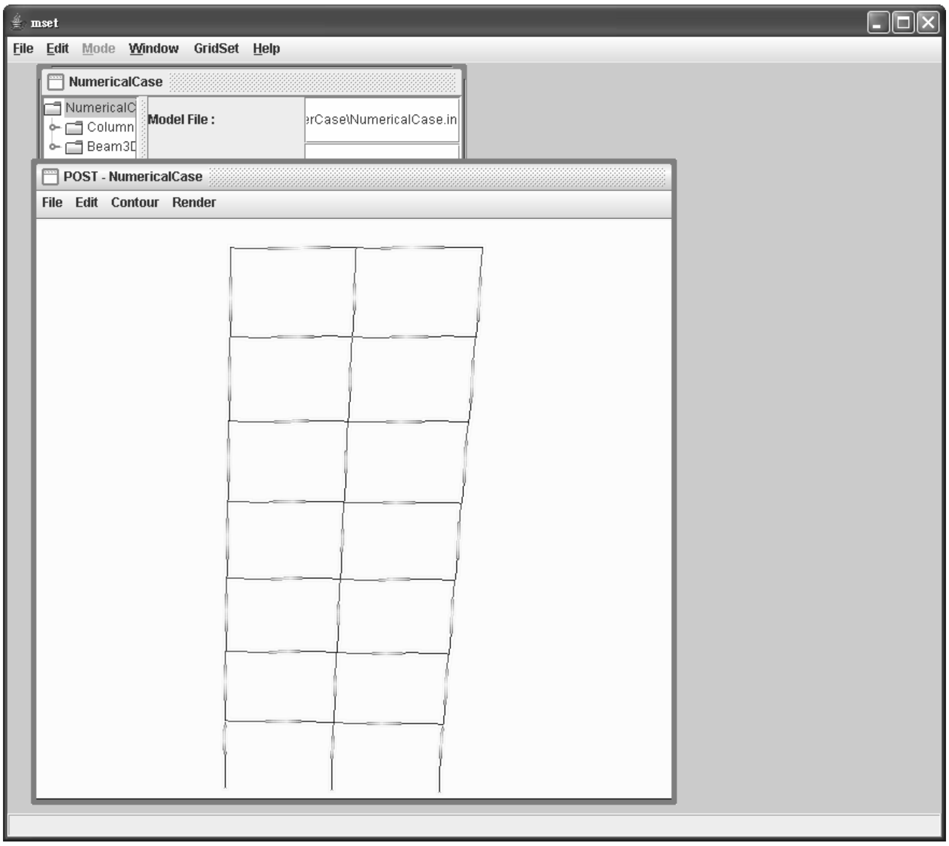


Fig. 10. The multi-level model of the numerical study.



Model File : rCase\NumericalCase.in

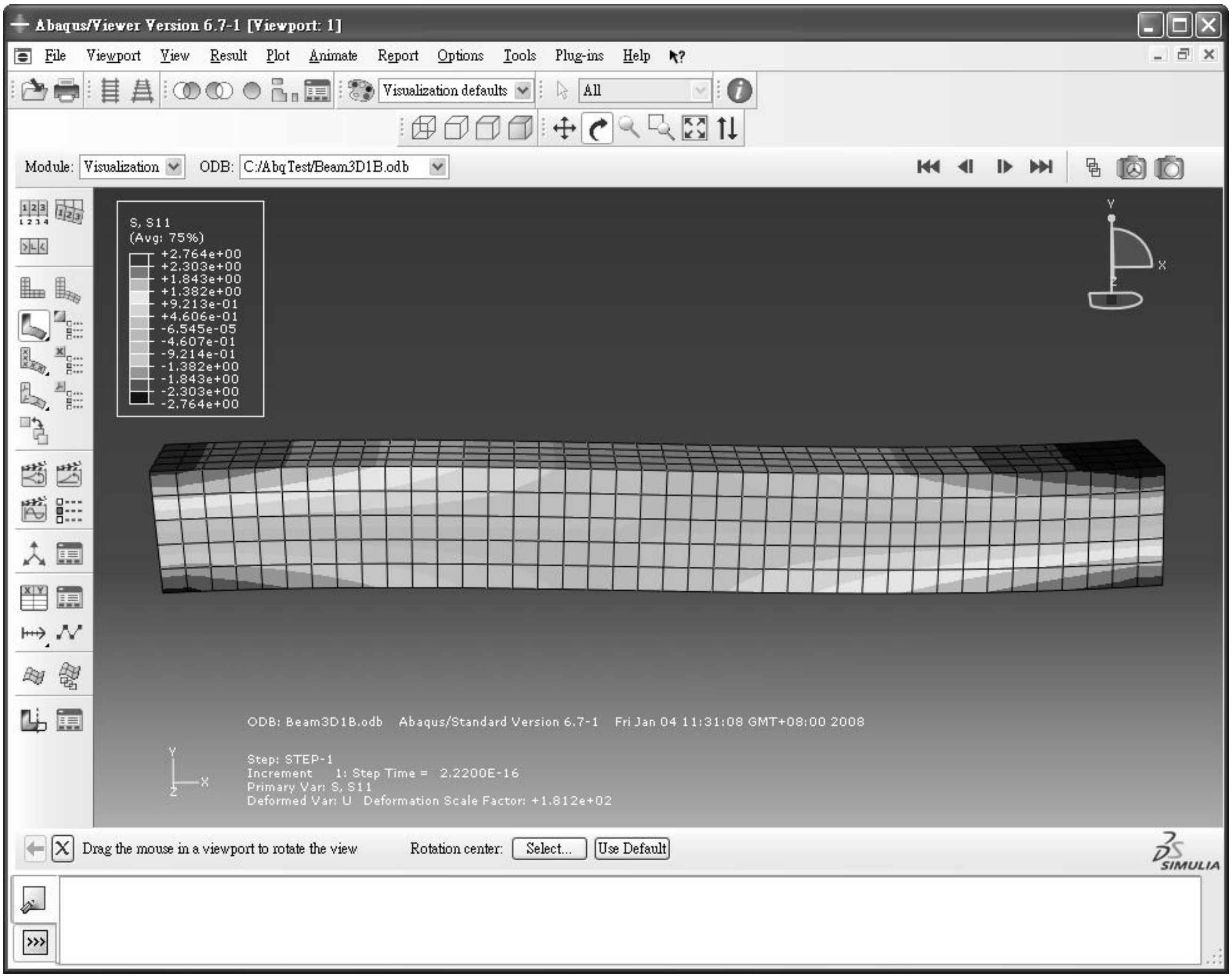
Fig. 11. Simulation results of the global structural model.

APPLICATION OF THE PROPOSED METHOD

The simulation of a simple two-level hierarchical model of a frame structure is presented to demonstrate the proposed system. As shown in Fig. 10, the global model is a two dimensional 7-story frame of reinforced concrete structure comprised of beam, column, and beam-column joint elements. Then a lateral load is applied as a point load at the left roof corner. The material nonlinearity is neglected. Hence, the structural response is limited to linear stage. The detail material properties are presented in Table 2. To obtain the detail response, two structural components (beam and column), have been selected to be analyzed in detail using the rigorous models.

The simulated result of the global structural model is presented in Fig. 11. This model is performed using Java, a platform-independent, net infrastructure language, so that the system can run on any computer interface that connected to the internet. The output of the global structural model is only nodal displacement on the frame element.

Furthermore, the simulation result of selected beam and column component are shown in Fig. 12 and Fig. 13, respectively. The nodal displacements of the simplified model in the global system are extracted and translated into the nodal displacements on the corresponding planes of the rigorous model, shown as known boundary conditions, to obtain the detailed responses of each component. Since the structural response is limited to the linear response, the tension and compression stress almost has symmetry values. For the beam component, the maximum and minimum stress (S11) are +2.76 MPa and -2.74 MPa, respectively. And for the column component, maximum and minimum stress (S11) are +8.97 MPa and -8.36 MPa, respectively. Based on the simulation results, it assesses the proposed simulation method is appropriate to simulate the response of the structural systems.



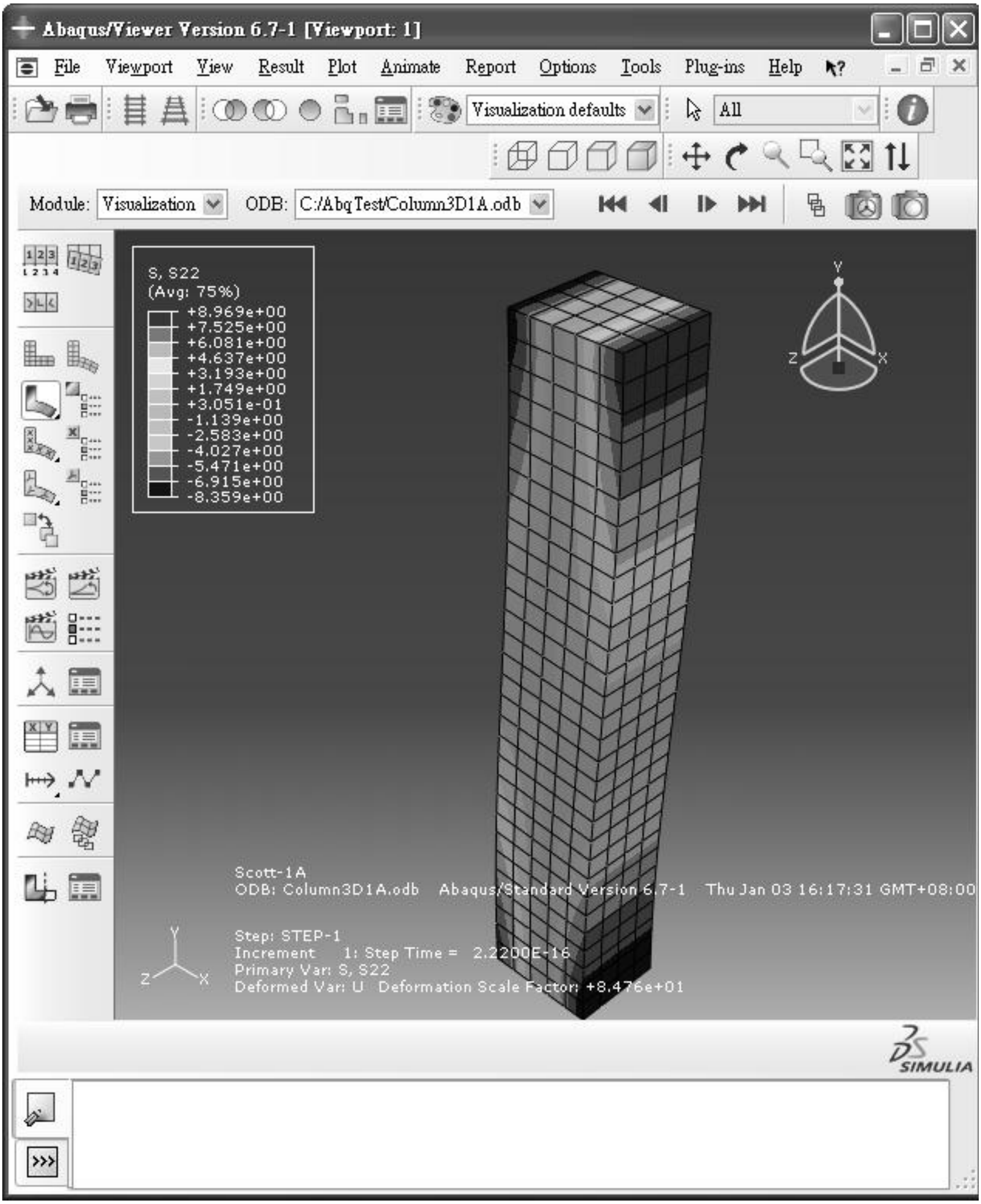
```
S, S11
(Avg: 75%)
  +2.764e+00
  +2.303e+00
  +1.843e+00
  +1.382e+00
  +9.213e-01
  +4.606e-01
  -6.545e-05
  -4.607e-01
  -9.214e-01
  -1.382e+00
  -1.843e+00
  -2.303e+00
  -2.764e+00
```

```
ODB: Beam3D1B.odb   Abaqus/Standard Version 6.7-1   Fri Jan 04 11:31:08 GMT+08:00 2008

Step: STEP-1
Increment     1: Step Time =   2.2200E-16
Primary Var: S, S11
Deformed Var: U   Deformation Scale Factor: +1.812e+02
```

Drag the mouse in a viewport to rotate the view

Fig. 12. Simulation results of the beam component.



Visualization defaults

All

Module: Visualization

ODB: C:/AbqTest/Column3D1A.odb

```
Scott-1A
ODB: Column3D1A.odb    Abaqus/Standard Version 6.7-1    Thu Jan 03 16:17:31 GMT+08:00

Step: STEP-1
Increment      1: Step Time =   2.2200E-16
Primary Var: S, S22
Deformed Var: U   Deformation Scale Factor: +8.476e+01
```

Fig. 13. Simulation results of the bar component.

CONCLUSIONS

This paper presents a multi-level modeling and simulation method of structural system using grid-based cluster-to-cluster distributed computing environment. The purpose of this study is to utilize the idle and available computational resources on the internet for providing the computing power needed for processing large-scale structural simulations. However, slow internet communication is expected to be a significant bottleneck. To conquer this internet-imposed obstacle, the grid computing environment is first organized as a two-level parallel platform, which first utilizes local cluster computing, and then remote, cluster-to-cluster computing. A hierarchical modeling approach and computational procedures for the proposed cluster-to-cluster computing environment have been added to streamline the process and avoid excessive internet communication. To fulfill the proposed concept, a prototype software system has been designed and implemented to perform the proposed multi-level modeling and simulation in a cluster-to-cluster distributed computing environment. Furthermore, to check the accuracy level, the simulation results of the proposed method is compared to the assumed exact solution. The comparison results show that maximum and average difference between proposed method and assumed exact solution is only about 4.84 and 1.44 percent, respectively. Therefore, based on the comparison results, it assesses the proposed simulation method is appropriate to simulate the response of the structural systems.

REFERENCES

1. Hajjar, J. F.; and Abel, J. F., “Parallel Processing of Central Difference Transient Analysis for Three-Dimensional Nonlinear Framed Structures,” *Communications in Applied Numerical Methods*, V. 5, No. 1, January 1989, pp. 39-46.
2. Modak, S.; and Sotelino, E. D., “The Iterative Group Implicit Algorithm for Parallel Transient Finite Element Analysis,” *International Journal for Numerical Methods in Engineering*, V. 47, No. 4, February 2000, pp. 869-885.
3. Farhat, C.; and Roux, F. X., “A Method of Finite Element Tearing and Interconnecting and Its Parallel Solution Algorithm,” *International Journal for Numerical Methods in Engineering*, V. 32, No. 6, October 1991, pp. 1205-1227.
4. Chen, H. M.; and Archer, G. C., “New Domain Decomposition Algorithm for Nonlinear Substructures,” *ASCE Journal of Computing in Civil Engineering*, V. 19, No. 2, April 2005, pp. 148-159.
5. D’Ambrisi, A.; and Fillippou, F.C., “Modeling of Cyclic Shear Behavior in RC Members,” *ASCE Journal of Structural Engineering*, V. 125, No. 10, October 1999, pp. 1143-1150.
6. Chen, H. M.; and Iranata, D., “Realistic Simulation of Structural Experiments by the Coupling of a Simplified Component Model and a Rigorous Component Model,” *Structural Engineering and Mechanics Structural Engineering and Mechanics*, V. 30, No. 5, November 2008, pp. 619-645.
7. Lowes, L.N.; and Altoontash, A., “Modeling of Reinforced Concrete Beam-Column Joints Subjected to Cyclic Loading,” *ASCE Journal of Structural Engineering*, V. 129, No. 12, December 2003, pp. 1686-1697.
8. GL4Java, “Jausoft GL4Java Home-page.” <http://www.jausoft.com/gl4java.html> (July, 1, 2004).
9. Ghugal, Y. M.; and Shimpi, R. P., “A Review of Refined Shear Deformation Theories for Isotropic and Anisotropic Laminated Beam,” *Journal of Reinforced Plastics and Composites*, V. 20, No. 3, February 2001, pp. 255-272.
10. Krishnamurty, A.V., “Toward a Consistent Beam Theory,” *American Institute of Aeronautics and Astronautics Journal*, V. 22, No. 6, June 1984, pp. 811-816.