Development of Three-dimensional Virtual Plant Vibration Simulator on Grid Computing Environment ITBL-IS/AEGIS*


**Center for computational science and e-systems, Japan Atomic Energy Agency, 6-9-3 Higashi-Ueno Taito-ku Tokyo 110-0015, Japan
E-mail: {suzuki.yoshio, nishida.akemi, araya.fumimasa, kushida.noriyuki, akutsu.taku, teshima.naoya, nakajima.kohei, hayashi.sachiko, aoyagi.tetsuo, nakajima.norihiro}@jaea.go.jp

***HTTR Operation Section, Department of HTTR, O-arai Research and Development Center, Japan Atomic Energy Agency, 4002 Narita-cho, O-arai-machi, Higashi-ibaraki-gun, Ibaraki 311-1393, Japan
E-mail: kondo.makoto@jaea.go.jp

Abstract
Center for computational science and e-systems of Japan Atomic Energy Agency is carrying out R&D in the area of extra large-scale simulation technologies for solving nuclear plant structures in its entirety. Specifically, we focus on establishing a virtual plant vibration simulator on inter-connected supercomputers intended for seismic response analysis of a whole nuclear plant. The simulation of a whole plant is a very difficult task because an extremely large dataset must be processed. To overcome this difficulty, we have proposed and implemented a necessary simulation framework and computing platform. The computing platform enables an extra large-scale whole nuclear plant simulation to be carried out on a grid computing platform called ITBL-IS, Information Technology Based Laboratory Infrastructure and AEGIS, Atomic Energy Grid Infrastructure. The simulation framework based on the computing platform has been applied to a linear elastic analysis of the reactor pressure vessel and cooling systems of the nuclear research facility, HTTR. The simulation framework opens a possibility of new simulation technologies for building a whole virtual nuclear plant in computers for virtual experiments.

Key words: Nuclear Engineering, Numerical Analysis, Structural Analysis, Seismic Analysis, Virtual Plant Vibration Simulator, Grid Computing, ITBL, AEGIS, HTTR

1. Introduction

CCSE/JAEA, Center for computational science and e-systems of Japan Atomic Energy Agency is conducting R&Ds for extra large-scale simulation technologies of whole nuclear plants using state-of-the-art computational and IT technologies. Specifically we focus on establishing a virtual plant vibration simulator on inter-connected supercomputers, for seismic response analysis of a whole nuclear plant (see Fig. 1). In order to achieve a highly accurate simulation, we need to consider how connecting conditions among parts composing a plant affect integrated behaviors (stresses and deformations etc.) of a whole.
A nuclear plant is generally composed of a gigantic number of parts. A simulation of a whole plant becomes a very difficult task because an extremely large dataset must be processed. To overcome this difficulty, we have established:

1. a simulation framework which allows model data preparation to be carried out in a part-wise manner, and allows the connecting condition of parts to be taken into consideration in an integrated simulation of a whole plant.
2. a computing platform which enables an extra large-scale simulation to be carried out on a grid computing platform called ITBL-IS, Information Technology Based Laboratory Infrastructure and AEGIS, Atomic Energy Grid Infrastructure.

The simulation framework using the computing framework has been applied to an elastic analysis of the reactor pressure vessel and cooling systems of the nuclear research facility, “HTTR: High Temperature engineering Test Reactor” at the O-arai Research and Development Center of Japan Atomic Energy Agency. The simulation framework taking advantage of grid distributed computing techniques showed early success in an extra large-scale simulation of major parts of the nuclear plant. Figure 2 shows the overview of the achieved simulation framework on the computing platform.

In this paper, we focus on the simulation framework and computing platform in detail.
2. Simulation framework of whole plant

2.1 Methodology of structural analysis for assembled structures

We have proposed a methodology of structural analysis for assembled structures ranging up to the size of a nuclear plant\(^{(1)}\) to estimate (1) the behavior by taking into account the heterogeneity conditions of assembled structures, (2) interactions and relative mechanical responses among connected parts, and (3) the multi-dynamic behaviors in an assembly. A methodology of structural analysis for assembly by taking account of heterogeneity conditions, for example, contact, abrasion, etc., is to treat an assembly by parts. To solve a complex structure, each part should be analyzed under heterogeneous condition, otherwise, part needs to be solved under each own necessary condition. In order to manage heterogeneity conditions in assembled structures, the method needs to treat the structure as part by part. The assembly is treated by providing boundary condition data for parts. By preparing input data one by one for analysis, and gathering them, a complex object is easily generated and practically solved in distributed manner.

2.2 FIESTA: Finite Element Structural analysis for Assembly

FIESTA, Finite Element Structural analysis for Assembly, intended for structural analysis of a nuclear power plant by assemblage of parts, computes parts and/or components in parallel, one by one, on distributed computers. Parts solved in distributed computers are gathered to form an entire structure. A mesh connection algorithm for different densities and geometries among parts is introduced, since mesh generations are carried out independently in part-wise manner. There are some popular algorithms to connect different meshes, such as Lagrange multiplier method, penalty method, and so on. As a first step, the penalty method was implemented and tested to assemble meshes. FIESTA proceeds the parallel and distributed computing as shown in Fig. 3. Each part data is individually prepared with neglecting adjustment between part boundaries, but the mesh density of each part is generated with design intention. Every mesh data is gathered to determine shared boundaries among parts. Part data is divided in shared boundary nodes and elements, and part own data. Each part data is treated by the domain decomposition method for parallel computing. And each part data is placed on the grid computing environment for distributed computing, by taking into account of the computer resources, load balance in computers, size of part data, and size of shared boundaries data.

![Fig. 3 Procedure of assembly analysis using FIESTA](image)

FIESTA includes three codes:

a) CONNECT code: extract information of interface of parts.
b) PartFIESTA code: decompose mesh for parallel computing.

c) FIESTA Solver code: execute parallel and distributed finite element analyses.

In order to carry out FIESTA, three input files are necessary for each part, 1) mesh data to execute finite element procedures, 2) part network data to specify the neighbor parts, 3) mesh data on the inter-boundaries. Since the parts are inter-connected, a neighbor list is necessary to efficiently search inter-connections. Using these three files, CONNECT searches appropriate finite element mesh contacts among parts and writes out a single file which defines the computational domain and inter-connections. This is the “Unify Parts” segment illustrated in Fig. 3. Then, PartFIESTA carries out the domain decomposition using this unified model. The resulting files are used to execute FIESTA solver. The flow of procedures is simply illustrated in Fig. 4.

![Fig. 3 Schematic flow of simulation using FIESTA](image)

2.3 Application of FIESTA to HTTR

In this study, we applied FIESTA to HTTR, which was digitalized in a part-wise manner. It is composed of the reactor installed inside the pressure vessel, cooling systems (the pressurized and auxiliary water air cooler) and three pipes interconnecting these major components (see Fig. 5).

![Fig. 5 The model of the whole nuclear power plant HTTR.](image)
The integrated seismic responses of the plant in entirety are analyzed. The dataset in the computation consists of finite element meshes (226,120,983 tetrahedrons, 49,071,318 nodes), input earthquake waves, and result data for visualization (2300GB). Number of meshes and nodes of each component are shown in Table 1. As input earthquake waves, we used the 20 second real data from the El Centro earthquake. The reactor pressure vessel, cooling systems, and three pipes are subject to an assembly structure analysis. Based on the physical nature of the wave propagation between parts that have a big size difference\(^2\), the simulation procedure is structured into a hierarchy of two levels: The primary level is the simulation of the major components: the reactor pressure vessel and cooling systems, whereas three pipes inter-connecting those components are considered as the secondary level. The output vibration data from the first level simulation (referred to as secondary vibration) is used as the input seismic data for the second level simulation. Since the major components can be thought as rigid boundaries into which no wave transmission occurs, the reaction from the pipes i.e., the small part is neglected. Figure 6 shows the hierarchy for seismic simulation of the whole nuclear power plant.

### Tab. 1 Number of meshes and nodes of each component

<table>
<thead>
<tr>
<th>Components</th>
<th>Number of meshes (tetrahedrons)</th>
<th>Number of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactor pressure vessel</td>
<td>126,713,831</td>
<td>26,047,774</td>
</tr>
<tr>
<td>Pressurized water air cooler</td>
<td>63,079,845</td>
<td>13,810,729</td>
</tr>
<tr>
<td>Auxiliary water cooler</td>
<td>14,962,732</td>
<td>3,364,149</td>
</tr>
<tr>
<td>Pipe1</td>
<td>1,895,042</td>
<td>622,090</td>
</tr>
<tr>
<td>Pipe2</td>
<td>7,331,618</td>
<td>1,830,427</td>
</tr>
<tr>
<td>Pipe3</td>
<td>12,137,915</td>
<td>3,396,149</td>
</tr>
<tr>
<td>Total</td>
<td>226,120,983</td>
<td>49,071,318</td>
</tr>
</tbody>
</table>

Fig. 6  The hierarchy for seismic simulation of the whole nuclear power plant. R and C represent the reactor pressure vessel and the cooling systems, respectively.

### 3. Computing platform: ITBL-IS

#### 3.1 Overview of ITBL project

ITBL-IS (Information Technology Based Laboratory Infrastructure) was developed and has been managed by CCSE/ JAEA in the ITBL project\(^3\). The ITBL project is a national project placed as one of the e-Japan Priority Policy Program to realize the e-Japan Strategy which sets goals to make Japan the world's most advanced IT nation. The objective of the ITBL project is to establish a virtual research environment in which researchers in various disciplines can collaboratively develop highly sophisticated simulation systems by fully utilizing computer resources located in high-speed network. The ITBL project was launched at April 2001 by six institutes: the National Institute for Materials Science (NIMS), the National Research Institute for Earth Science and Disaster Prevention (NIED), Japan Aerospace Exploration Agency (JAXA), the Institute of Physical and Chemical Research (known as RIKEN), Japan Science and Technology Agency (JST), and Japan Atomic Energy Agency (JAEA) and has been carried out to March 2006 as 5 years’ plan. At the end of the project, 681 users from 90 institutions participate in this project.
3.2 Functions of ITBL-IS

ITBL-IS consists of three layers; the communication and security service layer, the primitive service layer, and the application developer service layer. We briefly explain these layers.

A) Communication and security service layer

The ITBL environment has the authentication mechanism that enables user’s authentication with a digital certificate and the communication mechanism that enables secure communication through the Internet with HTTPS protocol.

A-a) Authentication mechanism

When users access the ITBL environment, users are authenticated by using the digital certificate. Users can access computers with single sign-on by comparing an ID of a user certificate with a user ID of the computers in the ITBL environment.

A-b) Communication mechanism

The establishment of the ITBL environment needs to connect heterogeneous computers in various research institutions through the Internet. To reduce the computer load, ITBL servers which consist of the certification server and control server are introduced. In addition, to minimize the modification of the firewall, HTTPS protocol which is a secure Web protocol is used. We call this communication infrastructure STARPC (Seamless Thinking Aid Remote Procedure Call).

B) Primitive service layer

Four basic functions, which are system monitoring, process management, task scheduling, and data managing, are prepared to implement the application service layer.

B-a) System monitoring

This is for getting information of network resources and computer resources.

B-b) Process management

This is for starting and stopping programs, monitoring status of program execution, and controlling job submissions.

B-c) Task scheduling

This is for allocating job executions based on users’ preferences on computers by using information obtained with system information monitoring.

B-d) Data management

This provides management of file information, copy and remove of files, and rename of files.

C) Application developer service layer

This provides users with the functions to develop their applications based on the functions 3.1 and 3.2. It includes distributed programming, visualization, and community management.

C-a) Distributed programming

Users can operate files or directories (copying, deleting, renaming and so on) for heterogeneous computers as a same manner (File Manager). Users can visually define the relationship for data and job execution using workflow (Task Mapping Editor: TME)\(^4\). Users can show resource information such as the number of users for each computer, network workload, computers’ spec, queues, and scheduling (Resource Information Service: RIS). Furthermore, users can generate their application using MPI library for heterogeneous computers (Seamless Thinking Aid Message Passing Interface: STAMPI) and tools which provide file editing, compiling, execution of programs, X-window application environment.

C-b) Visualization

Users can use a visualization system (AVS/ITBL)\(^5\) in cooperation with commercial visualization software, AVS/Express\(^6\). This system enables to display and to manipulate result of visualization on the Web. It also allows users to read any datasets on a computer located in the ITBL environment. It also can treat large scale datasets by parallelization of
the visualization process (Parallel Support Toolkit: PST). In cooperation with commercial software AVS/Express, PST enables the parallel visualization on supercomputer where MPI library is used for parallelization.

C-c) Community management

It provides some essential functions that users need to have a virtual community for their researches. It consists of the cabinet in which users can share datasets with their members, the meeting room that users can use as a communication tool to discuss their opinions, and the video meeting with images, voices, and documentaries among multiple places.

3.3 Client API of ITBL-IS

Users have developed various grid-enabled applications using ITBL-IS. These applications are roughly classified to four categories: parameter survey, substitution of computer, combination of different architecture, and virtual scalable parallel as shown in Fig. 7. According to the classification of applications, we have defined the client API (application programming interface) in hierarchy (Fig. 8). Some of them have been implemented on ITBL-IS. Grid middleware are roughly classified to two categories. One is Globus-based middleware and the other is the pre-existing system. Examples of grid environment based on Globus are NAREGI, EGEE, TeraGrid, and so on. Examples of the pre-existing system are ITBL, UNICORE, ABAKU, DIET, and so on. In the definition, we have considered common functions in these grid middleware. APIs have been realized on Globus, UNICORE, and so on. Our client API is based on those APIs. SAGA has been researched in order to define APIs for grid applications. When implementing our API completely, its information would be useful.

Fig. 7 Classification of ITBL applications.

In our definition, the client API has three levels: low level, middle level, and high level. The low level API has basic functions. The middle level API is for connection between the high level API and the low level API. The high level API enables to develop grid-enabled applications efficiently. We implemented the basic API in the low level and independent API in middle level. These APIs are currently available for Globus, NAREGI, ITBL and UNICORE grid middleware. The heterogeneous computing API has also been developed to easily realize the grid-enabled application of structural analysis for nuclear plant size of assembled structures.

We briefly explain these APIs.

1) Basic API (authentication API and communication API)

These APIs are the base for all of the functions. It establishes a communication route from a user terminal to the grid environment. When an application on a user terminal uses
computer resources, this function enables a user to access the grid environment with the authentication and encrypted communication. It is possible for users to use streaming and RPC communications among a user terminal and backend computers.

2) Independent API
   2-1) Job control API
       These offers the functions to operate workflow: Execution of a script of workflow in which job execution sequences defined is described, cancellation of execution, and receiving the current status of workflow.

   2-2) File control API
       These offer the following functions: File management on a user terminal and computers, directories displaying, file copying, file I/O, and so on.

   2-3) Resource information API
       These offer the functions to operate resource information: receiving the information of computer resources and queue information, which include names of queue, current status, the number of jobs and so on.

   2-4) Grid connection API
       These consist of a set of the previous APIs: Currently the connection between Globus, NAREGI, ITBL, UNICORE\(^{(16)}\) and DIET\(^{(17)}\) has been established.

3) Integrated API
   3-1) Heterogeneous computing API
       These offer the functions to generate the script for a job execution and submit a job. Usually the job on a grid environment is executed using a script written by an individual language. While job control API with script description enables to specify their detail grid function, users must understand every language when using every grid environment. Using these APIs, users can set up information for job execution in each grid environment with common interface. Currently, APIs are available for TME script of ITBL and WFML script of NAREGI.

4. Computing platform: AEGIS

CCSE/JAEA has launched R&D of grid middleware AEGIS for establishing computational infrastructure for atomic energy field by inheriting the knowledge and skills in ITBL-IS\(^{(18)}\) after the ITBL project. (AEGIS originally stands for Athena’s shield in Greek Myth.) The R&D aims at establishing predictable science and engineering to guarantee security and safety of atomic energy and preserve environment by constructing an environment of real-scale nuclear facility with computer aided research and development, computer aided engineering, computer aided science, and network computing system which enables those synchronicity (see Fig. 9).
The computer-Aided Research & Development (CARD) aims to realize an environment for seamless thinking aid which includes problem solving environment, research administration, large-scale visualization, and large data sharing. The computer-Aided Engineering (CAEN) aims to establish multi experimental tools, which includes 3D virtual vibrator, 3D virtual operator, 3D virtual generator, and 3D transition predictor. The computer Aided Science (CASC) aims to promote the atomic energy research & investigation which includes theory-verification simulator, physical experiment simulator, deductive analysis simulator, and phenomena-prediction simulator.

The safety, certainty, and reliability of large-scale nuclear facilities and supercomputers for atomic energy research needed to be maintained for the Internet, and the communication mechanism developed in ITBL was improved for increased Internet communication security using more reliable communication mechanism. Here, the double authentication with an IP address of client terminal and a certificate was implemented. The communication speed was improved by reducing thread number needed for data transfer and modifying the construction of a web server. To improve the security of accessing JAEA's Local Area Network (LAN) from the Internet, we implemented the use of hardware token with a digital certificate.

By establishing the simulation framework on the computing platform ITBL-IS and AEGIS, especially using the client API, early success in the extra large-scale simulation of the major parts of the nuclear plant has been realized.

5. Application of computing platform to simulation framework

The client API has been used to establish a grid-enabled application of structural analysis for nuclear plant size of assembled structures. Its architecture is briefly shown in Fig. 10. The application consists of four functions to control the developed simulation framework: 1) determination of input files and job submission, 2) transfer of input files, 3) job submission, and 4) visualization.

1) Determination of input files and job submission

After the connection to the grid environment ('Basic API' is used), a user selects input files. Input files are registered using the ‘new data’ dialog. It is needed to select programs corresponding to input files and to specify directories. In this dialog, a user can seamlessly select directories of a client terminal and supercomputers. When input files are selected, programs are automatically determined. Then, a user selects job queues. This function is realized using 'File control API'.

2) Transfer of input files

After the selection of input files and job queues, a user can submit jobs using ‘run’
button. Input files are transferred to work directories on supercomputers. Here, 'File control API' is used.

3) Job submission

After the transfer of input files, jobs are submitted and executed. When jobs are finished without any errors, output files are transferred and saved. Here, 'Job control API' is used.

4) Visualization

The visualization function is implemented using 'File control API' and 'Job control API'. Currently, the visualization of input mesh data and output data is available. Here, the PST is used to generate a visualization image.

6. A prototype experiment of assembly structural analysis of nuclear power plant

The six components: the reactor pressure vessel, the pressurized water air cooler, the auxiliary water air cooler, pipe1, pipe2, and pipe3 shown in Table 1 were solved using 512, 256, 128, 64, 64, 128 CPUs, respectively. Supercomputers involved in the simulation include FUJITSU PRIMEPOWER, SGI Altix3700Bx2, and HITACHI SR11000.

Figure 11 shows the visualized image of the whole analysis and the time evolution of the deformation of the reactor pressure vessel. The responses at local places, such as connections of parts that have a high possibility of damage or failure, are revealed (Fig. 12 (a) and (b)).
Fig. 12  Local seismic responses along three axial directions x, y and z, respectively, at the connections of the pipe with (a) the pressurized water air cooler and (b) the auxiliary water air cooler. The direction x, y and z is shown in Fig. 5.

7. Conclusion

The simulation framework developed has been applied to an elastic analysis of the reactor pressure vessel and cooling systems of the nuclear research facility, HTTR. The simulation framework taking advantage of grid distributed computing techniques showed early success in the extra large-scale simulation of the major parts of the nuclear plant, and opens a possibility of new simulation technologies for building a whole virtual nuclear plant in computers for virtual experiments.

Acknowledgments

This research was partially supported by Ministry of Education, Culture, Sports, Science and Technology (MEXT) under Japanese national project “Development and Applications of Advanced High-Performance Supercomputer”. Present study includes the result of “Development of physical model describing the dynamic interaction characteristics of component connections for the analysis of an entire nuclear facility” entrusted to “Japan Atomic Energy Agency” by MEXT.

References


(6) Upson, C., Faulhaber, T.A., Jr., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz,


(8) http://www.naregi.org/index_e.html


