

A Problem Solving Environment based on Grid Services: NAREGI-PSE

Hitohide Usami¹, Hiroyuki Kanazawa², and Shigeo Kawata³

- 1 National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430, Japan, usami@grid.nii.ac.jp,
WWW home page: <http://www.naregi.org/>
- 2 Fujitsu Limited, Shiodome 1-5-2, Higashi-Shinbashi, Minato-ku,
Tokyo 105-7123, Japan, kanazawa.h@jp.fujitsu.com
- 3 Utsunomiya University, 7-1-2 Yohtoh, Utsunomiya 321-8585,
Japan, kwt@cc.utsunomiya-u.ac.jp

Abstract. The National Research Grid Initiative (NAREGI) program is trying to develop NAREGI-PSE, which is a part of the NAREGI middleware. It provides services for deploying and executing large scale scientific computer simulation software on a grid's distributed and heterogeneous computer system. NAREGI-PSE has a PSE server that handles the applications residing on the distributed computers, and co-shares the application know-how, such as the source codes shared by the research community. This PSE server consists of an application pool and four PSE services. Users access the distributed computers through the NAREGI portal and the NAREGI application environment, which includes the NAREGI-PSE. The PSE server's main purposes are: (1) simple and easy execution of a user's application program in the grid environment, (2) simple and easy deployment of a user's program onto the distributed computer environment, (3) simple and easy software plug-in system into the application pool in the PSE server for software-reuse, and (4) application archive for a co-sharing application and it's know-how among the research communities, which is achieved by grid virtual organization (VO). NAREGI middleware is designed and developed based on grid services according to OGSA/WSRF frameworks.

1 Introduction

Using network connections spread out over a wide area, the large-scale grid project to integrate high performance computer resources that will be geographically dispersed is promoted in many countries. For example, TeraGrid in the United States integrates resources including more than 102 teraflops of computing capability and

more than 15 petabytes of online and archived data storage [1]. In Japan, the NAREGI program was instituted to research and develop high-performance, scalable grid middleware for a nationwide grid environment for scientific research.

In a large-scale grid environment, it is quite difficult for users to know the detailed specifications of a distributed computer system they can use. Most grid users will not write programs. Instead, they will use grid-enabled applications that make use of grid resources. These applications may be widely distributed general purpose software or applications that are used only in specific fields. In advanced scientific research, the latter is very important. Such applications are probably developed by small groups of researchers, and most researchers only use it on the grid resources.

We had to address the need for application developers and users in the field of advanced science research. Application users want to use applications with minimum knowledge of the applications and grid environment. Therefore, application developers should prepare information for the applications prior to release. NAREGI-PSE focuses on the functions to share applications, to deploy applications onto the grid, and to compose workflows using shared applications.

NAREGI-PSE consists of the following subcomponents: 1) a retriever and the application information setup, 2) the application and application information registration, 3) the compilation, test run, and comparison of the outputs with the desired results in the register phase in order to check if the program can be correctly run on the target computer systems, 4) an application search component to find the required programs from the application pool, 5) the application deployment, 6) the application and workflow search and registration in the application pool, and 7) a search for the appropriate computers based on the application information that is retrieved. This article at first briefly focuses on the NAREGI program itself, but later mostly on the NAREGI-PSE activities, including the design concepts, configuration, functions, and user scenario.

2 NAREGI Program

The NAREGI project is a five-year project that was instituted in fiscal 2003 in Japan. In 2006, the NAREGI project restarted as a new five-year project called the “NAREGI program” under the Development and Application of Advanced High-performance Supercomputer Project initiated by the Ministry of Education, Culture, Sports, Science and Technology (MEXT). NAREGI aims to research and develop high-performance, scalable grid middleware for the national scientific computational infrastructure. Such middleware will help facilitate computing centers within Japan as well as worldwide in constructing a large-scale scientific “research grid” for all areas of science and engineering, to construct a “National Research Grid” [2][3].

As a representative application area, NAREGI has adopted nanoscience, and large-scale nanoscience simulations have been performed. We assume that the future computational environment for scientific research will have a computational scale well beyond 100 teraflops and tens of thousands of users online. As such, the grid-

enabled nanoscience applications associated with NAREGI will serve as the hallmarks of the project to evaluate the effectiveness of the grid middleware that we will develop. The experimental deployment of these applications will be significant in terms of the scale of the computational requirements. It will also provide a virtual distributed computing environment with a large number of users in nanoscience and nanotechnology, and from the areas of academia as well as industry.

The middleware R&D work is being conducted at the newly established Center for Grid Research and Development, hosted by the National Institute of Informatics (NII) in Tokyo, Japan. The grid-enabled nanoscience application work is under the auspices of the Center for Applications Research and Development, hosted by the Institute for Molecular Science (IMS) in Okazaki, Japan. These two centers are collaborating to establish and operate a dedicated NAREGI test bed with Japan's SuperSINET as the underlying network infrastructure. The test bed will facilitate nearly 18 teraflops of computing power distributed over nearly 3000 processors. Both the developed grid middleware and the grid-enabled nanoscience applications will be under scrutiny and expected to achieve a performance over a predetermined scale, as well as serving to test the stability and manageability of future grids hosted by the two centers and utilized by various application domains.

3 NAREGI middleware framework

The grid middleware R&D work consists of six research and development groups, as shown in Figure 1, which are referred to as "Work Packages" (WPs). WP-1 focuses on the lower- and middle-tier middleware for resource management, such as a Super Scheduler, GridVM (providing local resource controllers), and information services on the grid. WP-2 covers the basic parallel programming tools for the grid, mainly consisting of two key middleware pieces, GridRPC (for task-parallel applications) and GridMPI (for data-parallel applications). WP-3 works on the grid tools for end users, including the grid workflow, the problem solving environment (GridPSE), and the grid visualization tools. WP-4 deals with the data grid for the federation of databases spread worldwide on the grid environment, while WP-5 investigates networking, security, and user management issues for high-performance grid infrastructures, such as real-time traffic measurements, QoS provisioning, and optimal routing for VOs and robust file and data transfer protocols. Finally, WP-6 acts as a liaison with the Center for Applications Research and Development, developing application-specific middleware components in order to grid-enable large-scale nanoscience applications, including those that require the coupling of multiple applications on the grid.

3.1 Lower- and middle-tier middleware (WP-1)

The requirements for a scheduler that can handle the widely distributed computing resources of a grid environment include a high level of scalability, fault

tolerance, and collaborative scheduling functions coordinating between multiple sites. This area of research and development covers such components as a “Super Scheduler”, which can manage all scheduling over a wide area, a broker that can secure the computational resources meeting the user requirements, such as the number of CPUs, urgency, and cost, a scheduler for the cluster environment, middleware for computational resources, networks, and grids, and tools for monitoring information and managing system configurations for the various applications [4].

1) Super Scheduler: This is a meta-scheduling system for the large-scale control and management of a wide variety of resources shared by different organizations in the grid environment. The system will be aimed primarily at identifying resources that can meet requests from batch job users and allocate these resources to specific jobs.

2) GridVM (local resource controllers): This is a new grid middleware that deploys a virtual layer of computing resources in the grid environment and facilitates resource utilization, resource protection, and fault tolerance.

3) Information services: A secure, scalable resource information management service will be established for the purpose of running a large-scale, multi-discipline grid computing environment.

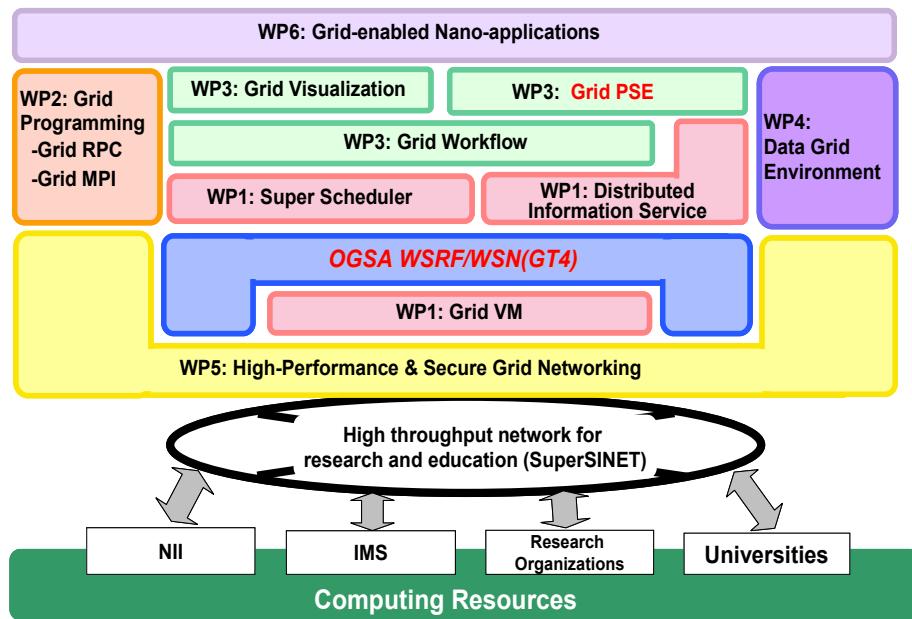


Fig. 1. NAREGI grid middleware stack

3.3 Grid programming environment (WP-3)

For this grid environment to be widely accepted by researchers, who are the end users, the grid software environment must be easy for them to use. To this end, research and development will be conducted in areas, such as the design of a workflow description language for controlling jobs on the grid, grid workflow tools for executing jobs in cooperation with the resource management mechanism, software tools for visualizing massive computational results distributed over the grid remotely with a light network load, and GridPSE to act as a software environment that can easily enable the execution, linkage, and coordination of the applications, computational modules, data, and other resources used by researchers over a wide area.

1) Grid workflow: A visual tool for seamlessly preparing, submitting, and querying distributed jobs running on remote computing resources. It explicitly handles programs and data, and is independent of the specific Grid middleware. Complex workflow descriptions, such as loops and conditional branches, are supported for nanoscience applications. Graphically described workflow jobs are converted to an enhanced workflow language based on Grid Services Flow Language (GSFL), which may be a common interface with other systems such as the PSE.

2) GridPSE: A Problem Solving Environment (PSE) for a scientific grid that facilitates the development and execution of application programs in nanoscience technology applications and other areas, without detailed computer-related knowledge or skills on the grid from the user's side. NAREGI-PSE is the software product name of this sub-project and the details for it are described in Section 4.

3) Grid visualization: A real-time, post-processing visualization system for nanosimulation, capable of reducing network loads that may interfere with smooth visualization, through flexible distribution of visualization tasks in the grid environment. This system is also characterized by having functions for large-scale parallel visualization, visualization for coupled simulation, and collaboration.

3.4 Data Grid environment (WP-4)

The research subjects are data grid fundamental technology, search control technology for a database federation, and metadata-based information integration for heterogeneous data resources. These technologies are under research and development for the federation of numerous databases spread throughout the Internet on the grid environment. The technologies include the Data Grid fundamental technology for managing and querying data resources using the WSRF-based OGSA infrastructure, search control technology (preventing combinatorial explosions caused by searching across many databases), and information integration technology

with metadata that mediates heterogeneous data resources. Their developments are carried out in cooperation with the other grid environment.

3.5 High-performance and secure grid networking (WP-5)

This section discusses the research and development of high-performance and secure grid networking in the NAREGI project. In the last decade, network infrastructures have become a very complicated combination of different sub-infrastructures, with wide ranges of throughput, delay, error rate, and jitter supported by different technologies, such as Multi-Protocol Label Switching (MPLS), differentiated services (diffserv), and optical networking. Because of the nature of distributed computing, the performance of grid computing may be considerably degraded by certain conditions of underlying networks, such as poor bandwidth, long delay, or temporal failures. Thus, we should be aware of the network resources as well as the computing resources [5]. The goal of the project's high-performance and secure grid networking subgroup is to develop a reliable, easy-to-use, high-performance, secure networking infrastructure for grid computing by taking into consideration the requirements of various applications; that is, the goal is to develop a high-performance “managed network”. For this purpose, we have set up the following three subgroups:

- 1) Network function infrastructure: In developing high-performance networks, over-provisioning and strict static reservation of the network resources are potential solutions, but neither are scalable nor cost-effective. Therefore, our goal for the network function infrastructure is to develop a system of measurement, management, and control for adaptively using and assigning network resources in order to avoid resource conflicts and cost-effectively maintain the network quality for grid computing.
- 2). Communication protocol infrastructure: The currently deployed version of TCP cannot detect network congestion until a packet loss occurs, so that many packets will be discarded. As either the network bandwidth or the router buffer size increases, the number of lost packets grows, significantly degrading the TCP throughput. Therefore, our goal for the communication protocol infrastructure is to develop a communication protocol optimized for grid computing and a method of evaluating the network performance.
- 3). Grid security infrastructure: Security problems in grid computing may occur in accessing the distributed resources over the network. Our goals for the grid security infrastructure are to develop a security model for grid computing based on PKI and to implement authentication and VO management across multiple organizations [6].

3.6 Grid-Enabled Nano Applications (WP-6)

WP6 is developing application-specific middleware components to grid-enable large-scale nanoscience applications, including those that require coupling of multiple applications on the grid. One example of such applications is multi-scale

simulation, where each application component utilizes different mathematical and physical modeling approaches and cooperates on spatially or temporally different calculations. To advance such multi-scale applications, and more generally, multiple applications, users have wasted a lot of effort in developing custom codes and decomposing original codes for semantic-level communication between heterogeneous scientific application components.

3.7 Services Architecture of NAREGI Middleware

NAREGI middleware is designed and developed based on OGSA/WSRF. WSRF is a messaging model and provides the ability to model state-full resources in a framework of web services. WSRF defines the conventions for managing “state”, so that applications can reliably share dynamically changing information. Major NAREGI middleware components are developed as grid services using GT4 [7] based on WSRF. The NAREGI middleware services architecture is shown in Figure 2. Each component works as an independent service and communicates with the others according to the grid service access protocol.

The top layer is the services for the end user, and this layer is usually provided as a portal for corresponding to the research community realized by VO. The second layer is the gateway services, and this layer provides several services building science gateway in TeraGrid. The third layer is the core grid services, and this layer provides core grid services, such as resource management, security services, and data services. The fourth and fifth layers are the physical resource infrastructure, which includes the network resources. NAREGI-PSE tightly co-works with workflow services and other layer services, such as information service and Super Scheduler, and is located in the gateway services layer.

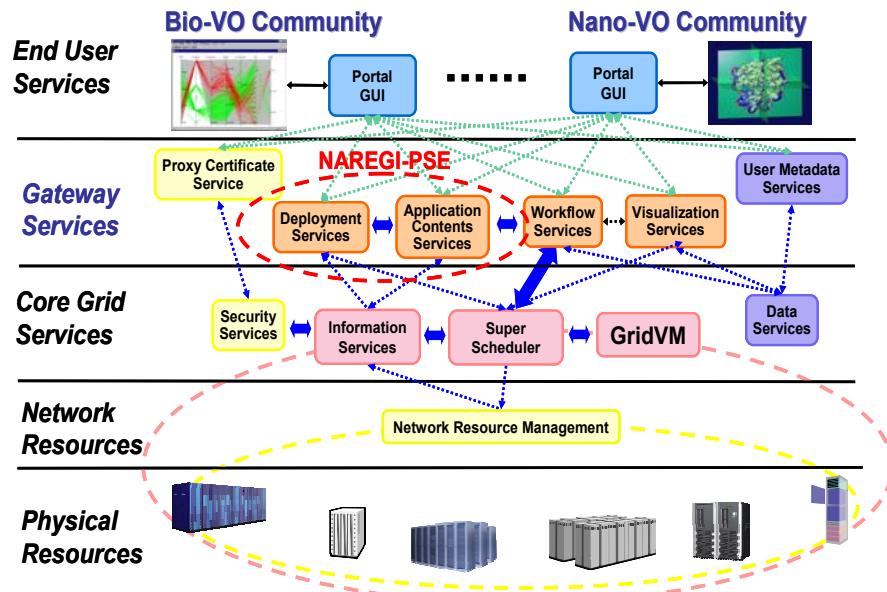


Fig. 2. NAREGI middleware service architecture

The workflow services are a visual tool for seamlessly preparing and submitting distributed jobs running on remote computing resources. It handles programs and data explicitly, and is independent of the specific Grid middleware. Complex workflow descriptions, such as loops and conditional branches, are supported for nanoscience applications. Graphically described workflow jobs are converted to an enhanced workflow language based on a workflow modeling language (WFML), which is a common interface with other systems, such as the PSE. The user imports the application information from an application contents service (ACS) in NAREGI-PSE and builds the application scenario using that application information on the workflow graphical user interface (GUI).

The information service contains various kinds of information; CPU, memory, OS, job queue, account, usage record, etc. These contents are aggregated and hierarchically accumulated in relational databases. A common information model (CIM) based schema is used to describe the resources in the computing grid [8]. The access control information for the applications in VO is supplied by the information service.

The Super Scheduler is a scheduling system for large-scale control and management of a wide variety of resources shared by different organizations in the grid environment. The system will be aimed primarily at identifying resources that can meet requests from batch job users and allocating these resources to specific jobs.

4 NAREGI-PSE

NAREGI-PSE is a part of the results of a grid application environment (WP3) group activity. NAREGI-PSE facilitates the deployment and execution of application programs in nanoscience technology applications and other areas, without detailed computer-related knowledge or skills for the grid on the user's side. [9-18]

4.1 NAREGI-PSE Design Concept

The design and development concepts of NAREGI-PSE are as follows:

- (1) Provide a framework to distributed users' applications on a grid
 - Users can register, deploy, and retrieve applications by using NAREGI-PSE for real-time collaborations.
 - Application developers distribute and share their applications with research community members.
 - Application users find it easy to use the latest research applications without a compilation or test run.
- (2) Focus on a legacy application

- Deploy application binaries for specific target machines.
 - Compile source programs, if needed.
- (3) Design and development policy
- Simple and easy execution of users' application programs on distributed computers.
 - Simple and easy deployment of users' programs onto the distributed computer environment.
 - Simple and easy software plug-in into the application pool in the PSE server to provide a software-reuse environment. In this mechanism the PSE server supports users to reuse previous software for their own purposes.
 - A problem solving scenario written by the workflow stored in the application pool in the same way as for a single application. End users easily reuse the stored workflows developed by themselves or by others for a new problem solving scenario.

4.2 NAREGI-PSE Configuration

The configuration of NAREGI-PSE is shown in Figure 3. NAREGI-PSE serves four WSRF-based grid services and a client GUI.

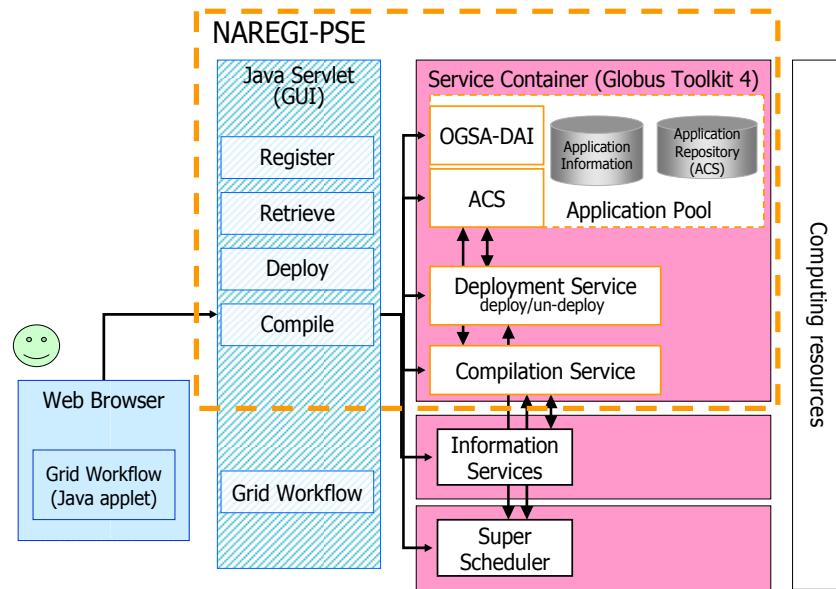


Fig. 3. NAREGI-PSE System Configuration

The NAREGI-PSE client GUI delivers a user interface to users, who work on distributed computer systems. The deployment service provides the function where they can transfer files of an application to remote computing resources, and execute a “post-procedure” to configure and/or examine a deployment. The compilation service provides the service that transfers the source files of an application to remote computing resources, and executes a script compilation, then collects and stores the binary files to ACS if the script compilation succeeds.

The application pool is implemented by using the application repository (AR) interface and application archive (AA) format from the ACS specifications. In addition, the application information utilizes OGSA-DAI and PostgreSQL.

4.3 Application pool

The application pool consists of two databases, the application information database and the ACS database as shown Figure 4. The application information database contains the meta-information related to the application, including the resource requirements (JSDL), and the ACS database contains the application entities, including the source codes.

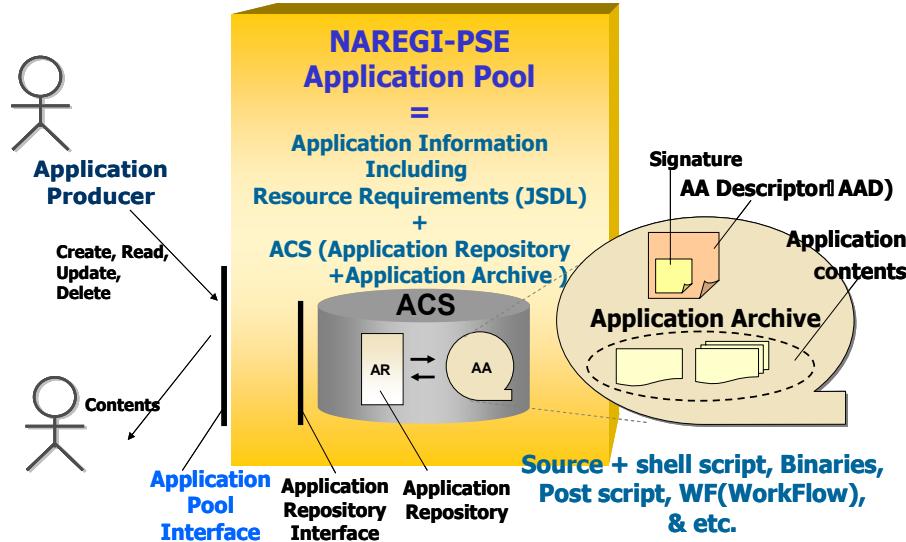


Fig. 4. Structure of Application Pool

(1) Application Contents Service (ACS)

ACS is a set of requirements and specifications that is being worked on by the Application Contents Service Working Group (ACS-WG) [19] within the Global Grid Forum (GGF) [20], which is an international standards and community body focusing on Grid technologies. The ACS specification documents define a

standardized way to manage and handle a grid application as a deployable logical unit, so as to maintain consistency, reduce management overhead, and enable automation throughout the lifetime of an application.

The ACS architecture is described as follows. The ACS provides a repository for grid applications. The application producer creates application archive (AA) instances accompanied with the meta-information describing the contents. The AA is a logical bundle of files that is used for provisioning and executing a task in a Grid system. Such information may include, but is not limited to, the application executable code, configuration data necessary for initial deployment of the application, and the deployment descriptor documents.

An AA consists of an Application Archive Descriptor (AAD) and zero or more application contents. The ACS repository will parse and make use of the AAD. The AAD may contain information, such as the identification, access constraint policy, and information associated with the structure, of the AA contents. In order to achieve application contents retrieval, the ACS repository needs to know the meta-information of the application content files, such as the names and types of the files. They should be provided by the Application Provider and be included in the AAD.

(2) Application Archive

The Application Archive of a single application contains the following materials:

- Source files (optional)
- An executable file (required, but it may be generated by a compilation procedure.)
- A compilation procedure (optional)
- A deployment procedure (required)
- Files required for executing the application (optional)

An application stored in the Application Repository will be a part of a complex application, using the grid workflow. A workflow consists of some applications that can be registered in NAREGI-PSE, too. The Application Archive of a workflow contains a workflow file described in WFXML. The resource requirements of applications in NAREGI-PSE are described based on the Job Submission Description Language (JSIDL). The application with a different resource requirement is stored as another application archive, as shown in Figure 5.

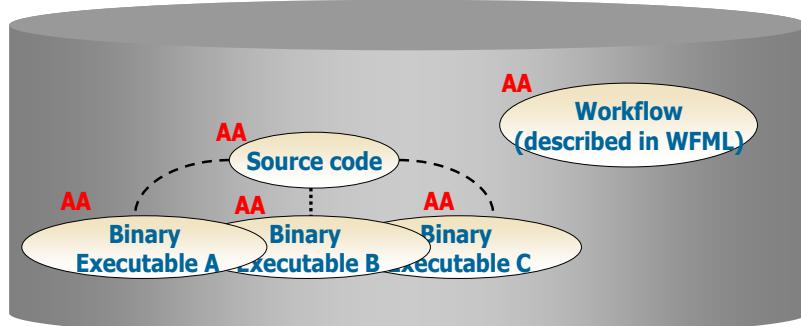


Fig. 5. Structure of Application Archives

4.4 Compilation and Deployment Services

The following two services are the key services in NAREGI-PSE.

(1) The compilation service

The compilation service provides the remote compilation function, according to the following procedures.

- Retrieve and select an application to compile.
- Display the candidate set of the computing resources from the resource requirements of the application.
- Select (a) computing resource(s) for the compilation.
- Transfer files to the selected computing resources and execute the compilation.
- If the compilation succeeds, the resulting files are stored in the application repository.

(2) The deployment service

The deployment service provides the deployment function, according to the following procedures.

- Retrieve and select a application to deploy.
- Display the candidate set of computing resources from the resource requirement of the application.
- Select (a) computing resource(s) for the deployment.
- Transfer files to the selected computing resources and execute the “post-procedure”
- If the post-procedure succeeds, the deployment information is stored in NAREGI-information service (NAREGI-IS).

4.5 NAREGI-PSE Using Scenario

NAREGI-PSE assumes two types of users, one is the application developer and the other is the application user. The application user just uses applications that were developed by other researchers or by his/herself.

(1) Application developer

In Figure 6, an application developer registers his/her application with the PSE server and tells that the application deploys to the execution server. This information is put into NAREGI-IS. Developed applications are deployed and registered to the application pool in the following way.

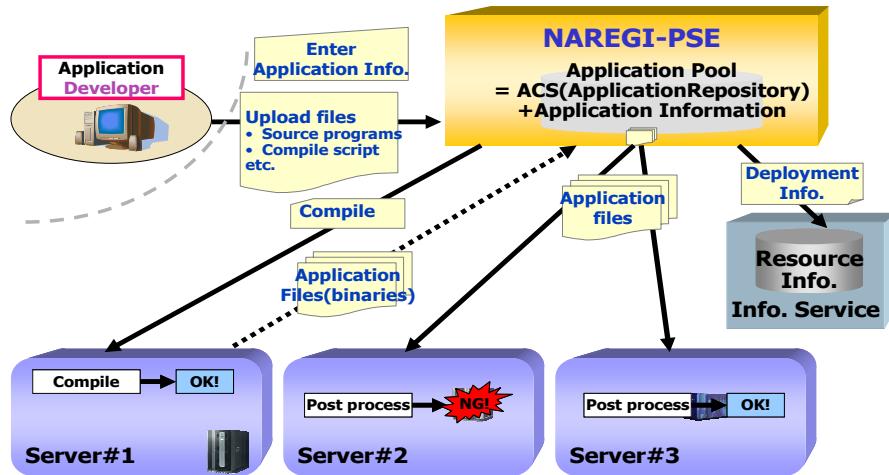


Fig. 6. NAREGI-PSE Usage Scenario (I)

- Upload to PSE server:
 - Upload files (e.g., source code/executables, compile script, post-process script, initial input files, etc.) to the PSE application pool.
 - Upload information (e.g., description, system requirements, etc.) associated with the uploading application to the application pool.
- Compilation (if needed):
 - Select an application and then select a server for the compilation matching to a resource requirement.
 - The PSE transfers the necessary files (e.g., source code) from the application pool to the compile server.
 - The PSE compiles and verifies them on the server.
 - The PSE transfers files (e.g., executable) from the compile server to the application pool.
- Deployment
 - Select an application and servers that meet the system requirements for the application deployment.
 - The PSE transfers the executables in the application pool to the selected servers.
 - Executes a post-process defined by user to configure and/or verify deployment on each server. (Optional)
- PSE registers information on the deployed servers to the information service

(2) Application User

Figure 7 shows the application user executing the registered application in NAREGI-PSE. The application executing processes are as follows.

- Application Retrieval:
 - Retrieve the application using GUI
 - Import the information from the selected application (system requirements - JSDL, etc.) from the application pool to make a workflow icon of the Grid Workflow
- Execution:
 - Compose a workflow job from the registered workflow icon.
 - The PSE submits a job to the Super Scheduler.
 - The Super Scheduler dispatches the resources referring the resource information provided by Information Service.

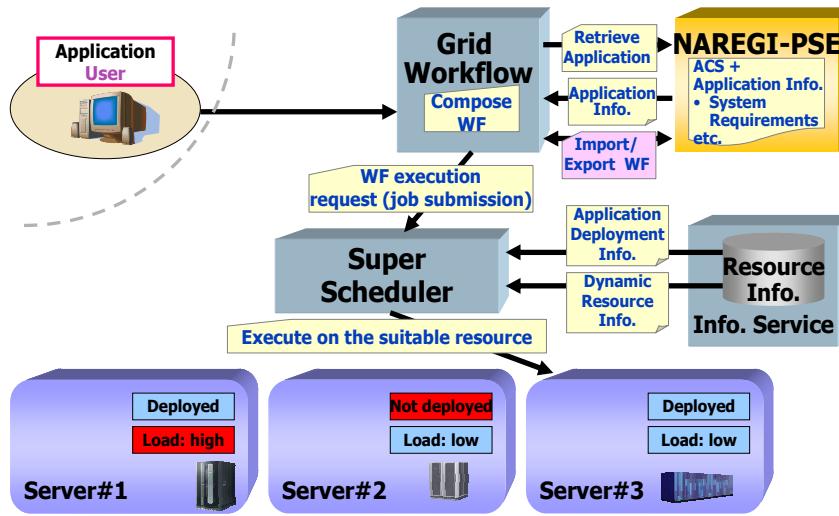


Fig. 7. NAREGI-PSE Usage Scenario (II)

5 Conclusions

The NAREGI program is now in the middle of its fourth year. The beta version of NAREGI middleware was released in May 2006. The program is now fully engaged in software development for delivering the final version of a NAREGI-integrated (NAREGI V1.0) Grid middleware at the end of the next fiscal year. The research projects under the NAREGI program have made considerable progress in the development the a prototype software, and some of them have already produced

preliminary results, including GridRPC, GridMPI, and NAREGI-Certification Authority (NAREGI-CA). NAREGI-PSE also has developed a beta version and delivered it to IMS and several other user sites. The role and viability of PSE in the distributed computer system have been demonstrated. The PSE server provides a smooth and flexible environment in the HPC on distributed computers, and encapsulates the complex information of distributed systems, so that on the PSE server users can perform HPC as if distributed computers are under the users' hands. The PSE server may open a new environment for the HPC world.

Finally, we regard Grid as one of the fundamental technologies of the IT infrastructure in the 21st century, and expect that the results of the NAREGI program will greatly advance research and development in the scientific fields, improve Japan's international competitiveness, and have a major economic impact.

Acknowledgments

This work was partly supported by a grant from the Ministry of Education, Sports, Culture, Science and Technology (MEXT) of Japan through the NAREGI (National Research Grid Initiative) Program. We would like to express special thanks to Prof. Kenichi Miura (NII), Prof. Mutsumi Aoyagi (Kyushu University), Prof. Yoshio Tago (Kanazawa University), and other members of the NAREGI project for their valuable discussions and useful comments.

References

- [1] TeraGrid: <http://www.teragrid.org/>
- [2] S. Matsuoka, S. Shimojo, M. Aoyagi, S. Sekiguchi, H. Usami, and K. Miura, "Japanese computational grid research project: NAREGI", Proceedings of the IEEE, Vol. 93, NO. 3, pp. 522-533, March 2005.
- [3] NAREGI: National Research Grid Initiative
<http://www.naregi.org/> [National Institute of Informatics]
<http://nanogrid.ims.ac.jp/nanogrid/> [Institute for Molecular Science]
- [4] I. Foster and C. Kesselman, "The GRID Blueprint for a New Computing Infrastructure," Morgan Kaufman Publishers, 1998
- [5] Y. Kitatsuji and K. Yamazaki, "A Distributed Real-time Tool for IP-Flow Measurement," In Proc. of the 2004 Int'l Symposium on Applications and the Internet, pp. 91-98, Tokyo, January 2004
- [6] Common Component Architecture Forum, <http://www.cca-forum.org/>
- [7] Globus Toolkit: <http://www-unix.globus.org/toolkit/>
- [8] G. Wasson and M. Humphrey, "Toward Explicit Policy Management for Virtual Organization," IEEE 4th International Workshop on Policies for Distributed Systems and Networks, pp. 173-182, 2003

- [9] E. Gallopoulos, E. N. Houstis, and J. R. Rice, "Computer as thinker/doer: Problem solving environments for computational science," IEEE Comp. Sci. and Eng., Vol. 1, 1994, pp. 11-23.
- [10] E. N. Houstis, E. Gallopoulos, J. R. Rice, and R. Bramley, eds., "Enabling Technologies for Computational Science," (Kluwer Academic Publishers, Amsterdam, 2000).
- [11] K. Schuchardt, B. Didier, and G. Black, "Ecce: A Problem Solving Environments Evolution Toward Grid Services and a Web Architecture," Available at <http://citeseer.nj.nec.com/459199.html>
- [12] Y. Umetani, M. Tsuji, K. Iwasawa, and H. Hirayama, "DEQSOL: A Numerical Simulation Language for Vector/Parallel Processors," Proc. of IFIP WG2.5 Working Conference on Problem Solving Environments for Scientific Computing, Sophia Antipolis, France1, 1987, pp. 147-162.
- [13] C. Konno, M. Yamabe, M. Saji, N. Sagawa et al., "Automatic Code Generation Method of DEQSOL," Journal of Information Processing, Information Processing Society of Japan, Vol. 11, No. 1, pp. 15-21.
- [14] Y. Umetani, C. Konno, and T. Ohta, "Visual PDEQSOL: A Visual and Interactive Environments for Numerical Simulation," Proc. of IFIP TC2/WG2.5 Working Conference on Programming Environments for High-Level Scientific Problem Solving, Karlsruhe, Germany, 1991, pp. 259-267
- [15] S. Kawata, K. Iijima, C. Boonmee, and Y. Manabe, "Computer-assisted scientific-computation/simulation-software-development system - including a visualization system -," IFIP Transaction, Vol. A-48, 1994, pp. 145-153.
- [16] C. Boonmee and S. Kawata, "Computer-Assisted Simulation Environment for Partial-Differential-Equation Problem: 1. Data Structure and Steering of Problem Solving Process," Trans. of Jpn. Soc. Comp. Eng. and Sci., Paper No. 19980001, 1998
- [17] C. Boonmee and S. Kawata, "Computer-Assisted Simulation Environment for Partial-Differential-Equation Problem: 2. Visualization and Steering of Problem Solving Process," Trans. of Jpn. Soc. Comp. Eng. and Sci., Paper No. 19980002, 1998.
- [18] CACTUS: <http://www.cactuscode.org/>
- [19] ACS-WG: <https://forge.gridforum.org/projects/acs-wg/>
- [20] GGF: Global Grid Forum <http://www.ogf.org/>