

An e-Science Problem Solving Environment for Scientific Numerical Study

Hyejeong Kang*, Kyoung-a Yoon*, Seoyoung Kim*, Yoonhee Kim*, and Chongam Kim**

* *Department of Computer Science, Sookmyung Women's University*

** *Department of Mechanical and AeroSpace Engineering, Seoul National University*

* {hj kang, yoonka, sssyyy77, yulan}@sookmyung.ac.kr, ** chongam@snu.ac.kr

Abstract—As increasing complexity of scientific computational applications that embrace multiple physical models and require a large range of scales on computing power, demands for interoperable infrastructures steadily have been increasing. Even though heterogeneity of computational resource including large-scale computing power, storage and networks gives difficulty of management on extending resource availability in e-Science environment, researchers, especially who have limited knowledge of managing computing resource, desire to have uniform and easy-to-use access interface to those diverse computational resources. However, no problem solving environment (PSE) is capable of solving all problems in diverse application domains as their execution requirement are so various. Depending on application characteristics, it is meaningful to provide a proper PSE to solve specific application patterns for e-Science environment and manage life cycle of an application for designing, executing, analyzing, and profiling.

In this paper, we present a common and integrated framework providing a problem solving environment among multi application domains, especially, in scientific numerical study, running over e-Science environment fostering interoperability in Grid infrastructure. The proposed framework, based on UNICORE, provides seamless, secure, and intuitive access to distributed Grid resources. It also speeds up developing new numerical study with existing application patterns, which are easily shared, reusable, and extendable.

Keywords—e-Science, PSE, scientific numerical study, UNICORE

I. INTRODUCTION

In several scientific fields, new computation methods have been developed to overcome the limitation of study. Accordingly, an e-science environment should be satisfied the demand of scientist, providing more efficient e-science infrastructure that can be conducted research and solved largely increased and complicated scientific problems.

In recent years, a grid environment has been adopted in order to accomplish these necessities, which are motivated for several reasons, including technology reuse, computational performance and also, security system aspects. [1]

Nevertheless, there is no scientific computational application that can be enabled various applications

performance in different domain more efficiently, within common environment which can satisfy various requirements.

Currently, already developed framework just support e-science environment for only specific application or for analysing computational model itself. By constructing research environment which defined per application in specific e-science environment, it is difficult to conduct research efficiently and access heterogeneous resource in absence of common interface even though they have similar process. Also, as proceeding e-science development, common interface and integrated environment is required to provide various analyse techniques to research and reuse it with the demand of other application's requirement.

In this paper, we describe the design and implementation of the PSE that aims to provide numerical study environment that require seamless access into Grid resources through common interface in various domains with common process. To support common and integrated e-science environment, we developed scientific computational application based on UNICORE, providing user interface to each application using GridBean and environment for execution of three domains.

We begin with introduction about characteristic of scientific application and our target applications in Section II and UNICORE which our framework based on are described in Section III. Section IV describes how problem solving environment is developed and PSE's implementation details are explained in Section V. Finally, in Section VI some conclusions and future works are presented.

II. ANALYSIS FOR SCIENTIFIC APPLICATION

A. A Scientific Application

E-Science today, particularly scientific experiments and studies, involve distributed and heterogeneous resources. The scientific applications use different approaches with similar characteristics in a wide variety of scientific domains and typically require distributed, high-throughput and data-intensive computing. As a result, there are an increasing number of scientific applications available that require access to Grid resources in order to benefit from the wide variety of resources and services available on e-Science infrastructure.

As shown in Fig 1, e-Science framework has been used to verify scientific theory and computational techniques composed with serial or parallel code. Before the jobs are submitted, user can additionally specify jobs' properties or requirements of resources. The jobs which are created through a framework are delivered to Grid middleware system like UNICORE [4], gLite [5] and Globus [6] by using Web services, which results can be confirmed in local environment by researchers.

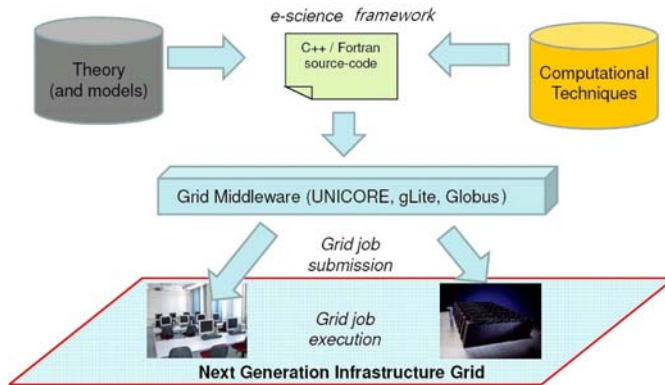


Fig 1. In e-Scientific applications as Grid jobs[7]

B. Scientific Application for numerical study

In practice, scientific computational applications for numerical study have been developed in foreign, like FLOWGRID [8], Scilab [9] with ProActive parallel suite [10], which can be executed on Grid environment as well as manage, monitor computational job. Even though they provide e-science environment, in case of FLOWGRID, is carried out targeting the specialized engineers only and limited in terms of extensibility.

Likewise, the Scilab has limitation of overall development and verification of application's specific numerical codes.

In addition, other existing research environment has also limitation to analyze various numerical techniques because numerical equations of each domain have strong interdependency and are also complicated.

Therefore, for numerical study, efficient research environment is essential to verify scientific computational equations in diverse domains and to prompt progress of developing new numerical techniques with existing application patterns.

In this paper, our targeted applications are aerospace (Hyperbolic Conservation Laws), mathematical science (Multi-phase) and computational science (Turbulence), which have similar computational techniques with different domains.

On developing a PSE environment for numerical applications, various requirements should be meted.

- Support research execution in diverse environment: it can be executed in personal computer or grid environment.
- Support independent computation environment on domains.
- Support to execute serial code and parallel code based on FORTRAN.

- Support pre-process and post-process module: pre-process for generating input file of numerical analysis and post-process for visualizing analyzed results.
- Support to access various resources in grid environment.
- Support extendable architecture.

For meeting needs described above, we developed a common and integrated Problem solving environment (PSE) providing access to heterogeneous resources as well as enabling to analyze numerical model in three domains based on UNICORE.

III. UNICORE

UNICORE (Uniform Interface to COmputing REsources) is a Grid computing technology providing grid software that combines resources of supercomputer centres and makes them available through the Internet. It hides unnecessary details from the users and provides single sign-on with well-established security mechanism relying on X.509 certificates and SSL (Secure Socket Layer). Within the UNICORE environment the user has a convenient way of using distributed computing resources without having to learn site or system specifics by seamless way.

Also the UNICORE support various clients for jobs creation, submission and monitoring, both graphical and command-line oriented. The graphical UNICORE Rich Client referred to as URC used in our framework offers graphical editors for setting up job descriptions. Instead of having to edit text-based job descriptions, the user is provided high level interfaces which are tailored to the applications what he wants to execute on remote systems. And these graphical interfaces are developed using GridBean which provides pluggable interface.

Our framework is specifically supported in the UNICORE Rich client (URC) as adding scientific domain-specific plug-in.

The UNICORE server, called UNICORE/X in UNICORE6, is composed with several components, which include gateway, execution engine (XNJS), databases (IDB, XUADB) storing execution and authorization information, user working directory (USPACE), etc. All services and components in server are Web services (WS-RF, SOAP/HTTPS), which make clients have to submit their job to server through web.

In its recent version, UNICORE6 follows the latest standards from the Grid and Web services world and offers a rich set of features to its users. It conforms to the Open Grid Service Architecture (OGSA) and several open Grid standards as mandated by the Open Grid Forum (OGF). The Compliance of UNICORE6 with the open standards is an important means of fostering interoperability with other Grid middleware as presented above, thus it widens the horizon of users and application developers.

Besides, the advantage of adopting open standards in UNICORE6 allows for the seamless use of UNICORE components by other technologies.

IV. DESIGN OF FRAMEWORK

A. GridBean

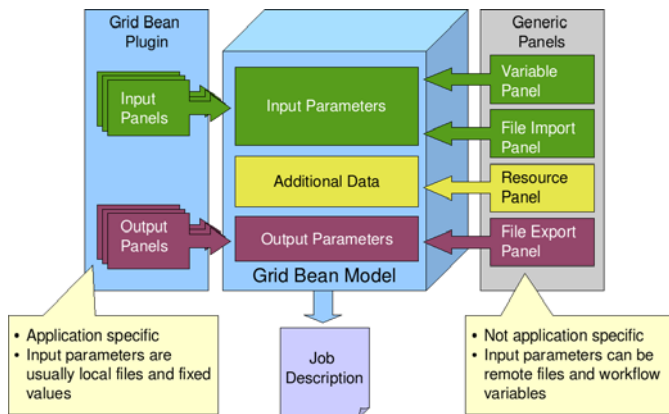


Fig 2. GridBean's Structure[11]

The GridBean [11] is one of the Grid Programming Environment (GPE, extended environment for grid client and based on High-level grid API) [12]. It generates job specification and provides pluggable GUI. By using GridBean, not a predecessor UNICORE Plug-in, our PSE supports grid standards such as JSDL, BPEL and WS-RF and it is possible to develop clients independent from URC as well as UNICORE clients.

To be specific, as described in Fig 2, GridBean consists of GridBean Plug-in and GridBean Model. GridBean Plug-in enables to develop GUI customized in application. And GridBean Model makes job description files, JSDL or BPEL, by defining parameters and in/output files. Generic Panels in Figure's right side are provided for in/output file selection, resource selection and variables checking that basic features which are common to all applications.

GridBean Plug-in provides API which can develop standalone client based on java swing and portal client based on portlet. In case of standalone client developed by GridBean Plug-in, each graphical component is bound to a data model element, so it has a need to link between components and GridBean Model. For this reason, GridBean Plug-in has `linkComponent()` method, `validator()` method for checking input before writing to the model and `translator()` method for saving input parameter to model and changing data format.

B. Framework Scenario for numerical study

Three applications, as presented above in section II, aerospace (HCL), mathematical science (Multi-phase) and computational science (Turbulence), require common processes as pre-process, simulation process, and post-process.

Fig 3 depicts scenario about using our problem solving environment. First, user installs our PSE program on PC and then selects a numerical study's method to perform a job. In pre-process step, user can generate or upload input file that is proper to numerical study has been selected. When it is finished, in simulation process step, user configures solving conditions such as experiment method. And user selects resource to submit the job and then JSDL file which specified by GridBean Model about job definition and job attributes will be submitted and executed at the previously selected resource. If job execution is finished, client notices it to user and user

can download output files. Graphical view is also offered for output files which require visualization.

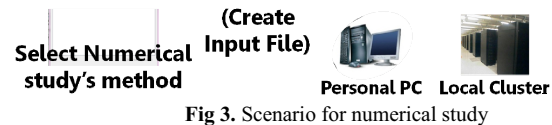


Fig 3. Scenario for numerical study

C. PSE Framework Architecture

Fig 4 shows framework for architecture based on UNICORE. In order to access to grid resources provided by each application provider, user selects the target system, after user generates a job which will be carried out in resources using PSE framework interface based on URC. If user once selects the target system, the job with user information is authenticated by the Gateway and forwarded to the Web services interfaces offered by the UNICORE/X server.

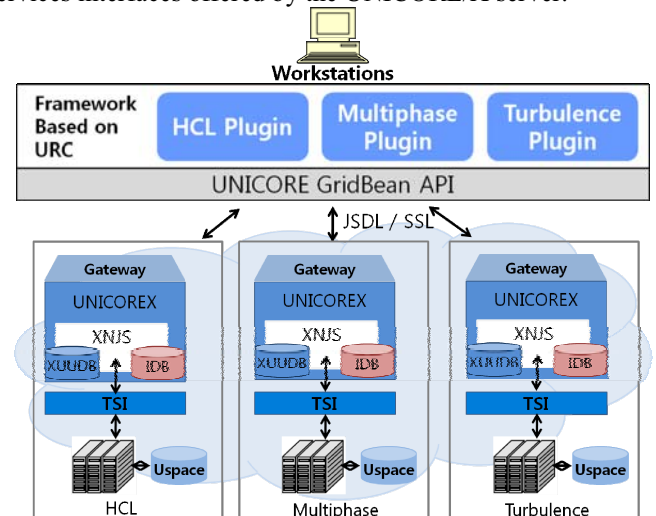


Fig 4. PSE Framework Architecture

And submitted abstract job by user is delivered to server's controlling entity called XNJS which provides set of services like job management, storage management and data transfer services. It communicates with the XUADB that is storing information of user authentication. If user has the authority to access grid resources, the XNJS maps the job concreted by IDB to the desired Target Systems, submits the job and monitors its progress.

Finally, the TSI takes the concrete job from the XNJS and executes it on the target system. Every job has its own a

temporary working directory named Uspace which stores data related to the specific job.

V. IMPLEMENTATION OF FRAMEWORK

To implement user interface of PSE in three other domains, we developed application's specific plug-in per application using GridBean that is presented above (IV). Within specific plug-in, components are organized dynamically by providing function to control event of the experiment. For specific example of user interface, to execute the 1D Euler numerical study of aerospace field is shown in Fig 5. Moreover, with the aim of integrated workbench framework, applications in three other domains can be accessed through uniform manner within PSE.

Basically user interface of framework consists of one or more plug-in modules and one data model depends on each of numerical studies' method. But there are a lot of methods and parameters which need to experiment.

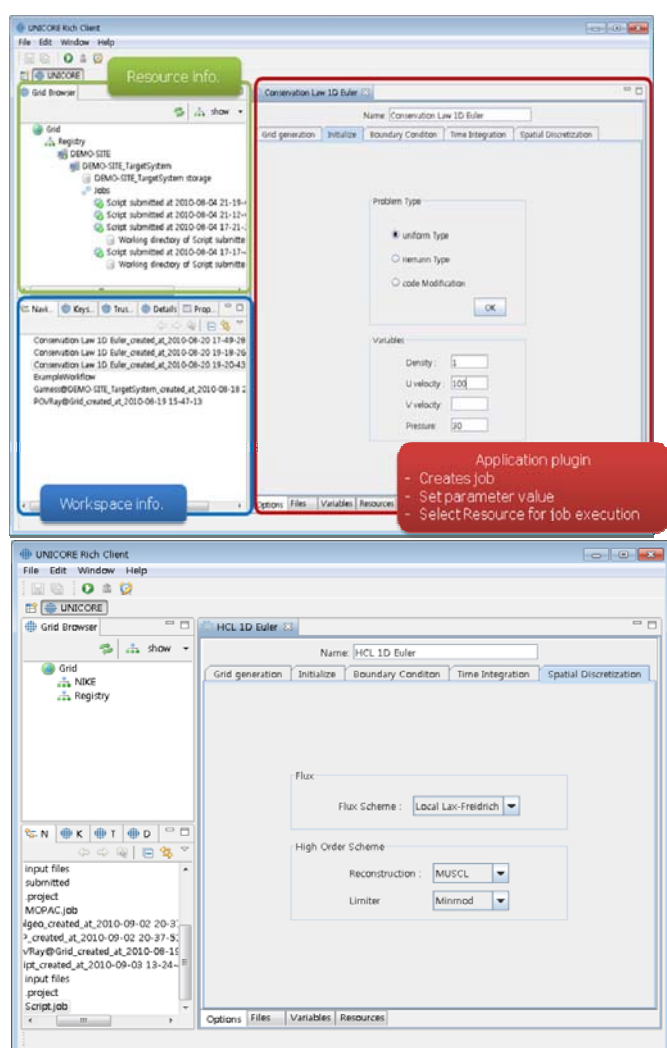


Fig 5. User Interface of Hyperbolic Conservation Law

For user's convenience, we developed interface to be organized dynamically by controlling event about the experiment method selection. To dynamically configure

parameters when selected experimental methods in each application, we classified parameters commonly applied and additional other parameters instead of defining a set of parameters in each experimental method. Defining parameter set like this, makes the management parameters more efficient and accurate.

In case of middleware, we used most of UNICORE's, and we modified IDB's *simpleidb* file which saved application's information such as name, version, location of executable file and parameter information required for job execution. Also, in order to provide resources among authorized user only, we generated authorization key for each application and its information is managed through XUDB.

VI. CONCLUSIONS

In this paper, we proposed a common and integrated PSE for scientific numerical study. This framework based on UNICORE provides common interface to access grid resources easily and offers plug-in optimized for applications.

To provide PSE to application appropriately, we developed scientific program for numerical analysis through adequate understanding problem of application.

Supporting various applications that are in diverse domain which have similarities in their experiment, in common environment is important and expected to improve their experiments' efficiency, convenience and reusability of technologies.

In the future we intend to extend our framework for supporting more diverse applications which have similar scenario to our target applications. Also we will support more complex experiments and more diverse numerical study method.

ACKNOWLEDGMENT

This work was supported by the National Institute for Mathematical Sciences (NIMS) grant funded by the Korean government (No. A21001)

REFERENCES

- [1] Benedyczak, K., - Users' UNICORE Security Guide - December, 2008
- [2] http://www.fz-juelich.de/jsc/hgfgroup/show_attach.php?pubid=32
- [3] Namgyu Kim, Ju Beom Kim, June H. Lee, S.B. Lim, "The design of the e-Science Service Framework for Applications", KOREAN SOCIETY FOR INTERNET INFORMATION 2007 Fall Conference vol. 8 /1, pp. 373~378, 2007
- [4] UNICORE, www.unicore.eu
- [5] gLite, <http://glite.web.cern.ch/glite/>
- [6] Globus, <http://www.globus.org/>
- [7] <http://www.isgtw.org/?pid=1001251>
- [8] FLOWGRID, <http://www.unizar.es/flowgrid/>
- [9] Scilab, <http://www.scilab.org/>
- [10] ProActive parallel suite, <http://proactive.inria.fr/>
- [11] Development of GridBeans
GPE Project Page Rebecca Breu, Sandra Bergmann, "Integration of Applications in UNICORE 6"