

CoaxSim Grid: Building an Application Portal for a CFD Model

Byoung-Do Kim¹, Nam-gyu Kim², Jung-hyun Cho³, Eun-kyung Kim³, Joshi Fullop¹

¹National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign

²Korea Institute of Science and Technology Information, Tae-jun, Korea

³Sookmyung Women's University, Seoul, Korea

Abstract

An application portal has been developed to support the execution of a computational fluid dynamics (CFD) numerical model that simulates hydrodynamic flow instability of the coaxial injector in the liquid rocket engine. The CFD application has been integrated into a web-based portal framework so that users can utilize resources available in grid computing environment. The portal provides users with a single sign-on access point that connects grid computing resources available in the users virtual organization. The portal developed in this project is built in the framework of GridSphere and Grid Portlet package. In addition to the default services offered by the GridSphere, extra service modules have been developed in order to provide customized features for the specific needs of the numerical model. The portal development was conducted as an international collaborative research between NCSA and Korea Institute of Science and Technology Information (KISTI).

1. Introduction

Three important characteristics of a grid computing environment are scalability, accessibility and portability. Scalability represents the ability to offer a diverse set of resources including high-performance computers (HPC), large storage systems, visualization systems, and even special experimental instruments for certain research communities. In order to take advantage of these various resources, the computational application itself needs to be scalable up to a certain level that passes normal utilization of local resources. The accessibility of the grid environment could be realized by software layer. The main topic these days in cyberenvironment research would be how to provide better solution for scientists and engineers to access and utilize the grid resources for their research. Another important factor is portability, which guarantees easy of transformation of developed solution between heterogeneous computing environments. Achieving three features mentioned above could serve as an

effective measure for developing a good problem solving environment (PSE) for the grid computing application.

The web-based portal as a science gateway could be the most appropriate approach for any existing scientific application to take advantage of available grid technology. The Web browser-based portal user interfaces is a great candidate for a single sign-on access point that provides access to heterogeneous, geographically distributed resources, services, applications, and tools. By using the interface, the grid-enabled portal can offer various grid solutions to users as long as the user has access to the Internet. Integrating various grid toolkits into the web-based portal framework has been proven to be an effective mechanism for scientific research communities to use the grid resources in the form of science gateway [1, 2].

With a grid portal framework and proper toolkits, it is possible to develop a scientific portal in a shorter time period than before. The main effort is in developing customized services for the numerical application that is implemented in to the portal. Since each

application has different requirements, developing customized modular services and plug them into the basic portal frame work would be the best approach as it gives the portal high level of flexibility and portability.

In this year, NCSA has initiated an international collaborative research project with KISTI in the area of cyberenvironment; developing problem solving environment for scientific applications. KISTI has been putting great effort into cyberenvironment research with the national e-Science project for the last couple of years, and one of the outcomes is e-Science Aerospace Integrated Research Systems (eAIRS) [3]. The eAIRS is an application portal that offers specialized services for a CFD application developed by Seoul National University, Korea. The eAIRS became a target item for the collaboration project because of its similar characteristics to the requirements to be developed for the NCSA's application. While the eAIRS successfully operates for its own CFD solver, its limited portability and other technical issues led the NCSA-KISTI development team to create a new portal framework. The most noticeable difference here is that eAIRS uses its own middleware and portal framework based on GT2 while the application portal developed in this project utilizes GridSphere and Grid Portlet to ensure portability. Though the mechanism underneath the portal interface is different from each other, both portals still share some of service modules, which is a great aspect of

the collaboration. The TeraGrid community was assumed to be a virtual organization for this portal project. This paper presents an overview of the application portal as well as details of the newly developed service modules for the CFD application.

2. The CFD Application

The application implemented into the portal is a computational fluid dynamics (CFD) numerical model for the coaxial injector flow in the liquid rocket engine. This model has been developed through collaboration research work between NCSA and School of Aeronautics and Astronautics Engineering at Purdue University, West Lafayette, IN. The purpose of this numerical model is to investigate hydrodynamic instability of the two-phase jet flow (gas and liquid) in the coaxial type of the injector that is normally used in the Space Shuttle Main Engine (SSME) and other large scale space-launching rockets. In the fuel injection systems of the liquid rocket engine, gaseous fuel and liquid oxidizer are injected through the nozzle into the combustion chamber of the rocket engine. Due to many disturbance factors such as acoustic pressure variation from the chamber or intrinsic hydrodynamic instability in the flow itself, the atomization mechanism of the fuel injection promotes combustion instability during rocket engine operation.

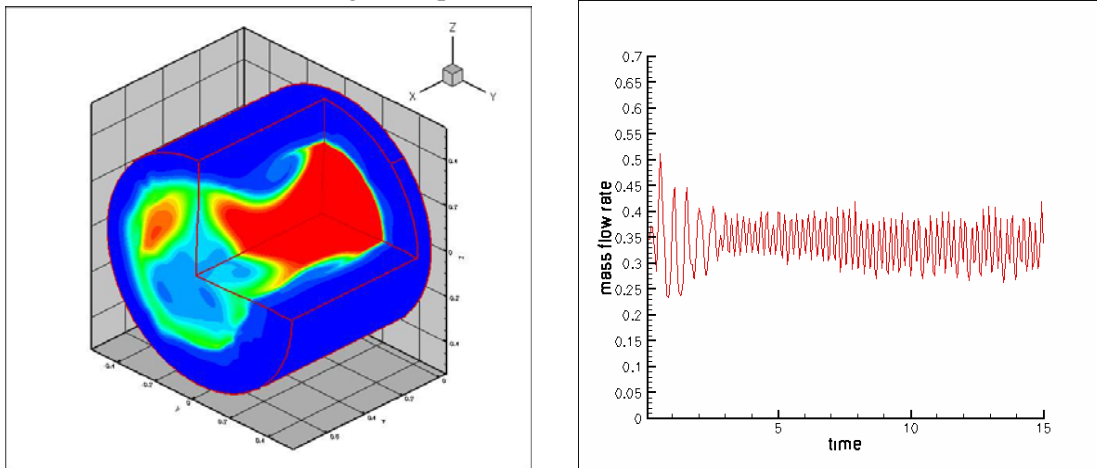


Figure 1. Cross-sectional view of liquid jet density contour in a recessed region of the coaxial injector and frequency analysis of the jet pulsation.

The numerical model in this paper is to simulate the coaxial jet flow and to investigate the atomization mechanism of the coaxial jet in order to obtain a better understanding of combustion phenomena in liquid rockets [4, 5, 6]. Figure 1 shows flow instability inside the injector nozzle and jet pulsation at the exit.

The numerical model consists of three major parts; pre-processing, main solver, and post-processing, which is a common structure for engineering numerical models. Mesh generation, input parameter setup, and code-run configuration compose the pre-processing step. The main solver is the core of the model, and solves flow physics in the computational domain. The solver produces a large amount of data in multiple formats. This data get visualized in the post-processing step. The workflow described here is widely employed procedures especially in CFD numerical models. Figure 2 shows the workflow explained in this section. Due to the short time length for the collaboration project, the portal development focused on integrating the CFD solver and automating post-processes. The pre-process procedure will be added on to the current product later on.

The code is parallelized using Message Passing Interface (MPI), and runs on a large number of processors. It is also possible to control the size and frequency of the output data files as well as the number of parameters to be printed. For production runs, it normally produces data for density, pressure,

and velocities in three-dimensional Cartesian coordinate with respect to time and space. The model is also capable of check-pointing, allowing users to check for numerical divergence or other numerical problems. Typical runs of the application takes 3 to 5 days with 16 to 32 processors depending on simulation requirements. The data size of the output also varies, but typical execution produces 10 to 20 GB for each run.

The application portal that we have developed integrates the numerical model into the GridSphere portal framework and enables users to control all the features mentioned above. In addition to the basic functions that are provided by the GridSphere and Grid Portlets, extra service modules for the specific needs of the model have been developed by means of portlet programming. As the portal puts complex service layers and resources behind the web-based interface, an experienced user can run the code on various resources available in his/her virtual organization. The next section will discuss the architecture of the application portal.

3. Portal Architecture

The application portal employs a standard three-tiered architecture; resource layer, grid service layer, and portal interface. The architecture consists of several major elements such as a standardized portlet container based on the JSR-168 standard,

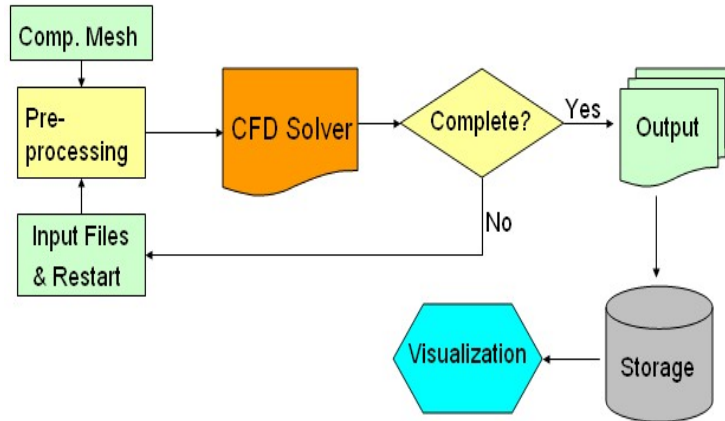


Figure 2. Workflow diagram of the coaxial injector flow modeling application

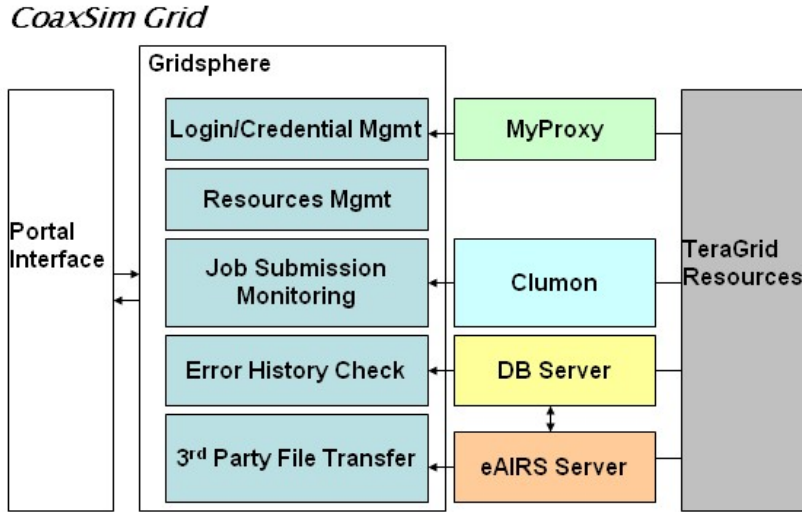


Figure 3. Three-tiered architecture of the CoaxSim Grid Portal

grid security, file transfer and remote execution based on the Globus toolkit. The GridSphere framework with the Grid Portlet also employs the three-tiered architecture and provides the basic functionality mentioned above. The application portal developed in this project is based on the GridSphere framework 2.01 and Grid Portlet 1.3.0. The GridSphere, a portlet container and a collection of core services and advanced user interface library, makes developing a portal easier by employing a portlet programming approach [7, 8]. Pre-WS Globus Toolkit 4 serves as base middleware in this architecture while it still maintains the GT2 compatibility.

Within the GridSphere framework, a series of customized service portlets have been developed for the specific needs of the coaxial injector application. In addition to the new services, some of the basic features have been improved as well in order to increase the efficiency and ease of use.

Figure 3 shows a diagram of the application portal architecture. The middle section illustrates relations between grid services and 3rd party service modules that are required to operate the services. Further explanation on service development and its mechanism will be given in the next sections.

4. CoaxSim Grid Portal

Figure 4 shows front page of the CoaxSim Grid portal that has a brief explanation about the portal development project. The basic features such as user account management, grid credential retrieval, resource registration, file management, and job submission are provided by the GridSphere. Once a user logs into the portal, the first service tab is 'Profile' that includes account and service management. In the next tab 'Grid Services', the user has a choice of proxy server for the credential retrieval; *myproxy.ncsa.uiuc.edu* or *myproxy.teragrid.org* can be used as long as the resources are registered to the portal. The time length of the credential activation can be configured through the credential management portlet. If the user is an administrator of the portal, then he or she can register any resources available in the virtual organization. The resource management page under the grid service tab provides a feature that users can examine those registered resources. The MDS 2 (Monitoring and Discovery System) service offers information regarding available services and jobs on the machine, however, the MDS 2 is not at production level and the information offered is very limited at this point. Once MDS 4 is deployed onto the TeraGrid systems, it is expected to provide more useful information of the resources. Another basic feature in the grid service tab is file management. It allows users to upload

and download files and move data around between local workstation and the grid resources. The user interface for the file management service is mediocre at best, but the effort of developing another file management module was not feasible at the given time frame. The next tab, named 'Simulations', handles job submission and status check-up for the jobs that have been submitted through the portal. Jobs are submitted by Globus Resource Allocation Manager (GRAM) service, and users are asked to input information required for running the job. Once the job is submitted by GRAM, users can review the job status in the job monitoring page. The GRAM service simply shows submitted, pending, active, and done status. In the CoaxSim Grid portal, an extra feature displays detailed job status information from the local scheduler. The implementation of the feature will be explained in the next section.

The functions explained so far in this section are mainly the given features by GridSphere or modified version of them. In the 'Data' tab, newly developed features that are specifically customized for the coaxial

injector model have been implemented. First, the 'Real Time Code Check' page provides users with a function that displays calculated results of critical parameters in the middle of job execution. Once the user selects a job from the list, the page displays plots of residual value from the matrix solver and total computation/communication time up to that point in the code execution. This feature is extremely useful to the user when the code runs over a long period of time since the residual value provides users with information on numerical behavior of the code. The time plot also gives the user an idea of how long the code will run, and also how far the code has progressed.

The second function is data management. While the GridSphere provides a basic file management portlet, it still requires manual labor from the user if he/she wants to move the output files to a local workstation. The data management function developed for this portal offers an automatic file transfer capability between the computational resources and long-term storage unit (Mass Storage System at NCSA in this case).

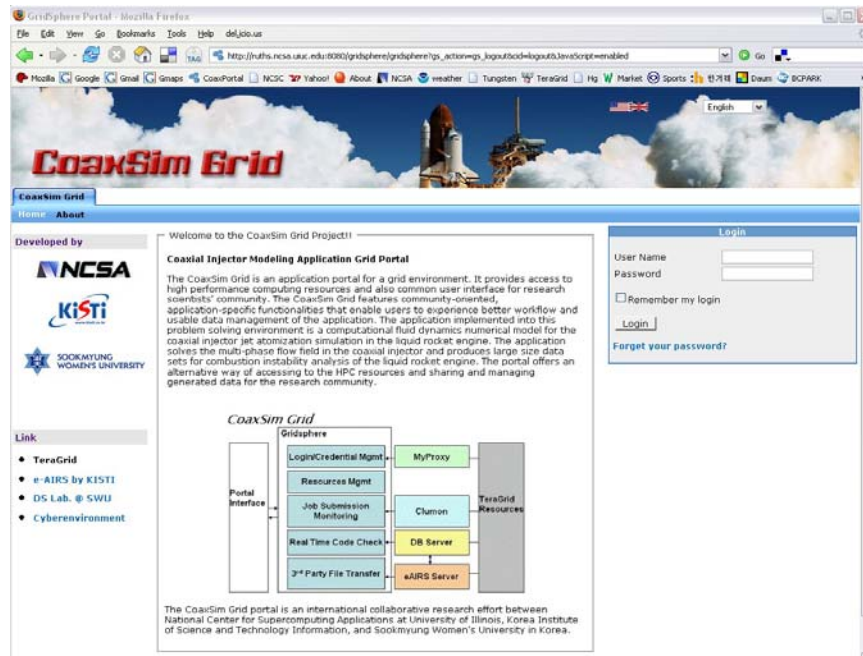


Figure 4. Front page of the CoaxSim Grid Portal

It also provides direct file download from the computational resources to a local workstation. Both single file and multiple files download are possible allowing users flexible file management. Copies of saved files are extracted from the mass storage system but the original data files remain available in the storage system after the download.

5. Service Modules Development

5.1 Grid credential retrieval module

The credential and resource management function is provided by Grid Portlet in the Gridsphere framework. Once the MyProxy server is registered in the *resources.xml* by administrator, the user can create session credential using X.509 proxy certificate. Since typical run time of the application is usually longer than normal credential life-cycle, users had to reissue the credential once the original one is expired. The MyProxy team has recently developed a solution to the problem while they were working on EU

DataGrid project. At start of a session, users store their long-lived credentials in a dedicated MyProxy server and delegate short-lived credentials to their jobs. When job's credential nears expiration, the Workload Management System retrieves a new short-lived credential from the MyProxy server on the user's behalf and uses it to refresh the job's credential [9]. The CoaxSim Grid portal's credential module uses this feature in order to avoid issues due to credential expiration.

5.2 Job submission and status check module

The features for job submission from GridSphere satisfy the basic needs for running this application, but the user interface was found to be not efficient enough. We have integrated the GRAM service selection page into the resource selection and job submission page, and redesigned the RSL scripting page for a better user interface.

The screenshot displays the 'Job Monitoring Portlet' interface. At the top, there are tabs for 'Profile', 'Grid Services', 'Simulations', and 'Data'. Below these, there are sub-tabs for 'Job Submission' and 'Job Monitoring'. The main content area is titled 'Job Monitoring Portlet' and includes a 'TeraGrid Resources Monitoring Link' pointing to 'CLUMON'. A 'Job List' table shows three jobs with their IDs, names, end times, and statuses. Below this, 'Job Information for : teragrid_gram_job_200506140712' is displayed, followed by 'Services Information', 'Job Variables', and 'Hosts Information' tables.

| Job ID | Job Name | Job End Time | Job Status |
|--|----------------|----------------|------------|
| teragrid_gram_job_200506140712 | coax_test_demo | 20060614000422 | DONE |
| https://tg-login3.ncsa.teragrid.org:46556/1838/1150288319/coax_test_03 | coax_test_03 | 20060614073120 | DONE |
| 4e0a21071b300395:17de8eb:10bf052dfb6:-7e00 | coax_test | 20060620163348 | DONE |

| user | started | account | jobname | queue | status | server | submitted |
|--------|--------------------|---------|---------|-------|---------|-----------------------------|--------------------|
| baibes | Sat May 20 2:38:18 | lqp | mgac | dque | Running | tg-master.ncsa.teragrid.org | Sat May 20 2:38:18 |

| var1 | var3 | var2 |
|------|------|------|
| abc | ghi | def |

| Mem_Free | OS | CPU_count | hostname | Physical_Memory | Mem_Used | Load | CPU_speed |
|---------------|-------|-----------|----------|-----------------|---------------|------|-----------|
| 2,049,638,400 | Linux | 2 | tg-c234 | 4,199,301,120 | 2,149,662,720 | 0.00 | 1,296 |
| 2,068,239,104 | Linux | 2 | tg-c233 | 4,199,301,120 | 2,111,052,016 | 0.71 | 1,296 |
| 2,079,457,280 | Linux | 2 | tg-c232 | 4,199,301,120 | 2,119,843,840 | 0.69 | 1,296 |
| 2,094,465,024 | Linux | 2 | tg-c231 | 4,199,301,120 | 2,104,836,096 | 0.69 | 1,296 |
| 2,081,947,648 | Linux | 2 | tg-c230 | 4,199,301,120 | 2,117,353,472 | 0.69 | 1,296 |
| 2,552,217,600 | Linux | 2 | tg-c229 | 4,204,347,392 | 1,652,129,792 | 0.71 | 1,296 |

Figure 5. Job Monitoring page with a link to CluMon and information from the PBS scheduler.

In the Job Monitoring page, we added a couple of new features in order to give more job-specific information to the users. When the job is submitted by the GRAM service, it is sent to the local scheduler, such as PBS, and the GRAM generates a job ID for its own use. Unfortunately, relating globally scheduled jobs to jobs scheduled to run on a computational resource with its own scheduler has been a problem since the inception of global grids. Since GRAM service only returns very simple job status message, users are not able to get any information regarding how the job is processed on the local machine. An indirect method of solving this problem that we have employed is to assign some key (a unique job ID in the RSL script in this case) to the global job at submission time, and pass that key along to the local scheduler as a job attribute. Then on the information query side of things, that key could be used to query and ascertain the local job name. CluMon, a cluster monitoring system, developed by NCSA [10] has been utilized in this mechanism. The CluMon was developed to give an overview of the current state of the

computational resources at a glance. It delivers information from both the scheduler and hosts to users through web interface. To convey this information to the portal user, CluMon was slightly modified to recognize a global ID request and use that key to cross-reference the local job name and return the job info to the requester. The only technical bug is that the local scheduler (PBS based) currently has a bug where it is not returning all of the job information on a remote query, as opposed to a query from the local machine. Once this is fixed, the mechanism should work as designed. Figure 5 shows the job monitoring page that displays job-related information extracted from CluMon.

5.3 Real time code checkup module

This module has been developed in order to allow users to check the behavior of the code during run time. At every time step, the code writes values of physical parameters in a single file that are critical for determining the convergence of the code. The portal can pull this file whenever users want, and display the plots that are shown in Figure 6.

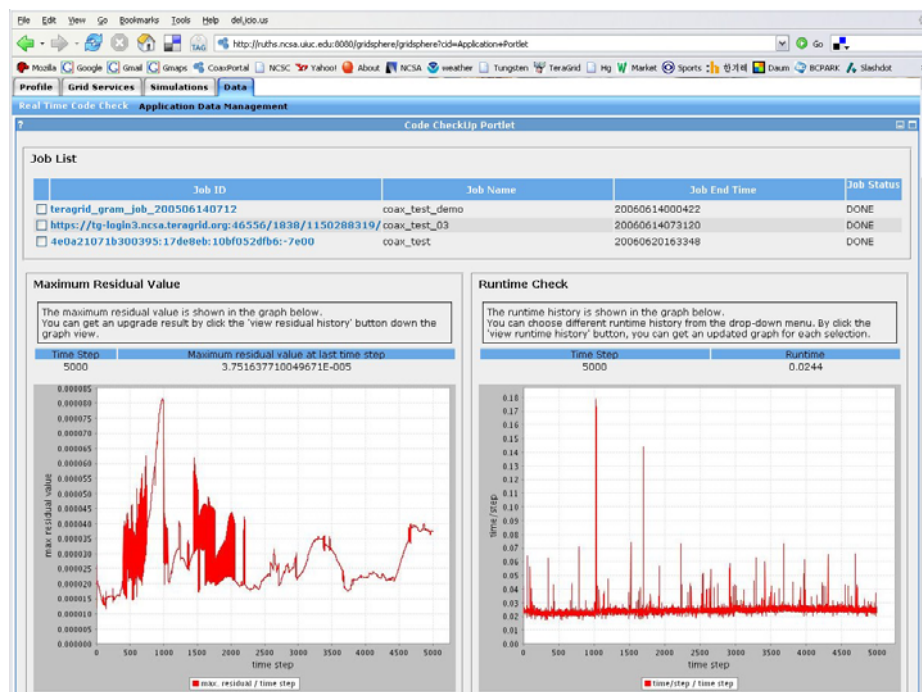


Figure 6. Real Time Code Check-up page with residual value plot and computational time per time step plot.

The right plot is matrix residual value with respect to time and the left plot is showing computational or communication time with respect to accumulated time or each time step. This graph generation is done by utilizing freely available java graphic API, JFreeChart. This module was previously developed for the e-AIRS project at KISTI, and was implemented here with a slight modification. The displaying parameters can always be replaced with other choices depending on the users requirements. This service is particularly useful to the most of numerical application developers especially where the application has a check-pointing capability because it shows numerical behavior of the code in real time during execution.

5.4 Post-processing data management module

The GridSphere provides a default file management portlet. Once the user download and install the GridSphere and Portlet package, the user can use the feature

out of the box. However, users of a specific numerical model usually want to have customized services for the output file management because data management for the post processing requires specific procedures that are suitable for an automated process. The coaxial injector model produces hundreds of output files from each run, and the size of the data easily goes beyond of normal size of disk quota of users. The CoaxSim portal utilizes 3rd party file transfer mechanism that was developed with KISTI's eAIRS project. Once the eAIRS file transfer module using gridFTP protocol is activated, then the eAIRS server keeps remotely checking the data production from the CFD solver on the computational machine. The files are transferred to long-term storage units that are already registered to the portal. At the same time, meta data for the transferred files are saved at separate DB server. When portal server inquires about the information on the output file on behalf of the user, the DB server parses corresponding meta data to the portal server for display.

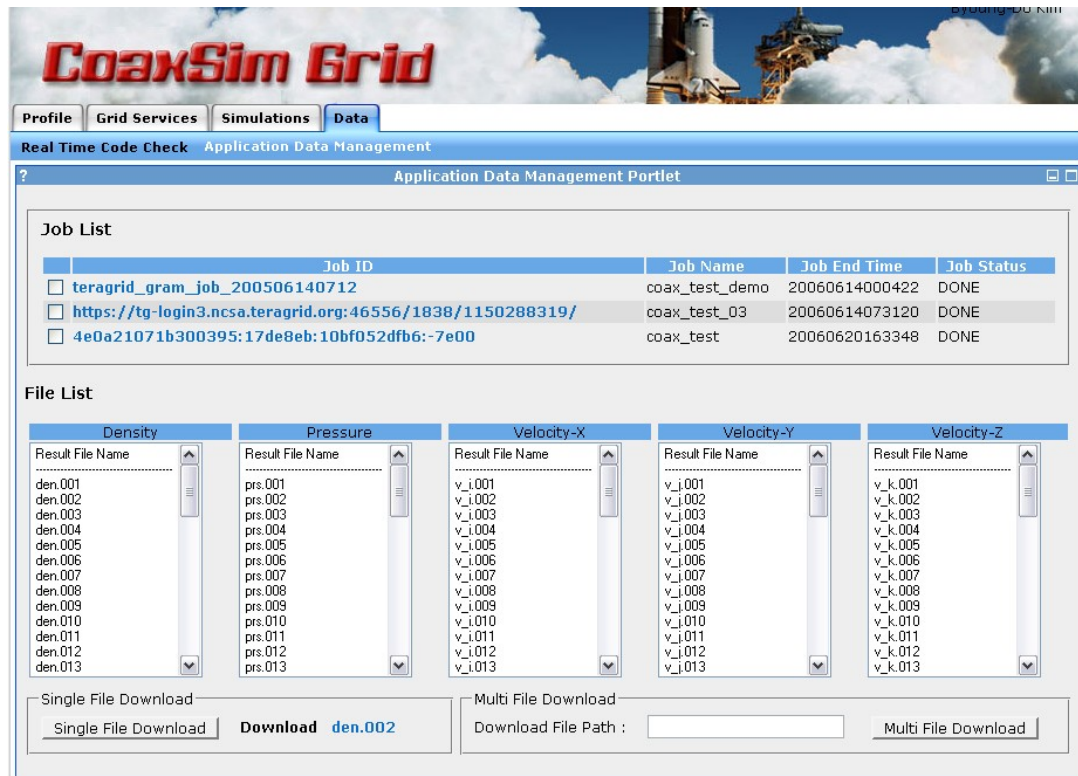


Figure 7. Data Management page for output file browsing and single/multiple file download

The actual file transfer from the storage system to the local workstation occurs only when the user decides to download the file to the local workstation. This mechanism saves time and avoids network performance bottlenecks since only the meta data are needed before the actual file transfer. The e-AIRS server enables the 3rd party file transfer activity between the computational resource and storage system and also between the storage systems to local workstation by using gridFTP protocol.

6. Summary and Discussion

CoaxSim Grid, an application portal for a CFD numerical solver has been developed through the collaborative project between NCSA and KISTI. The CoaxSim Grid portal developed in this project utilizes basic portal features by the GridSphere framework and Grid Portlet package from GridLab. The main focus of the development of the CoaxSim Grid was to provide customized services for the coaxial injector modeling application. Rather than asking users to get accustomed to what is given with current grid technology, proactively developing users need and the requirements of the application in the context of the grid portal could bridge the gap between the engineering researchers and computer science technology.

The architecture presented in this paper is based on the concept that the portal server is a container of service clients that are designed according to the portlet component model. The use of standardized portlet component model enables fast development of customized portal for a given application. It also allows plug-in type of service module development, which guarantees flexibility and portability of the portal contents.

By having unified user interface for using all types of grid computing resources in the TeraGrid community, the CoaxSim portal could accelerate the pace of research production of user groups when it is positioned as a research community oriented problem solving environment. Challenges are still there, however, because it still requires high level of understanding of grid

technology and lots of programming work for services development. Keeping up with the rapidly changing the standards in the grid computing technology is another barrier for the engineering and science research groups.

The CoaxSim Grid portal in this project is an example of overcoming those obstacles through collaboration between computer science group and engineering application group. Though there are still many areas to work on in order to create more sophisticated grid computing solution for generic scientific applications, the CoaxSim Grid project has shown a possible way of reusable grid services development using currently available grid technologies.

Acknowledgement

The authors would like to acknowledge supercomputer time provided by the National Center for Supercomputing Applications and financial support from Korea Institute of Science and Technology Information.

References

- [1] M.P. Thomas, J. Burruss, L. Cinquini, G. Fox, D. Gannon, L. Gilbert, G. von Laszewski, K. Jackson, D. Middleton, R. Moore, M. Pierce, B. Plale, A. Rajasekar, R. Regno, E. Roberts, D. Schissel, A. Seth and W. Schroeder, 'Grid Portal Architecture for Scientific Applications', *Journal of Physics, Conference Series* 5. Accepted for publication
- [2] M.P. Thomas, M. Dahan, K. Mueller, S. Mock, C. Mills and R. Regno, 'Application Portals: Practice and Experience', *Grid Computing Environments: Special Issue of Concurrency and Computation: Practice and Experience* 2002 V.14:1427-1444.
- [3] e-AIRS project at <http://obiwan.kisti.re.kr/escience/eairs>
- [4] B. Kim, S. D. Heister "Numerical Modeling of Hydrodynamic Instability of Swirl Coaxial Injectors in a Recessed

Region”, 42th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Sacramento, CA, 2006

[5] B. Kim, S. D. Heister, S. H. Collicott “Three-Dimensional Flow Simulations in the Recessed Region of a Coaxial Injector”, *Journal of Propulsion and Power*, Vol. 21, No. 4, pp. 728-742, 2005

[6] B. Kim, S. D. Heister “Effect of Chamber Pressure Variation on High-Frequency Hydrodynamic Instability of Shear Coaxial Injector”, 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Fort Lauderdale, FL, 2004

[7] J. Novotny, M. Russell and O. Wehrens, ‘GridSphere: An Advanced Portal Framework’ at <http://www.gridisphere.org/gridisphere/wp4/Documents/France/gridisphere.pdf>

[8] J. Novotny, M. Russell and O. Wehrens, ‘GridSphere: A Portal Framework For Building Collaborations’ at <http://www.gridisphere.org/gridisphere/wp4/Documents/RioBabyRio/gridisphere.pdf>

[9] D. Kouril and J. Basney ‘A Credential Renewal Service for Long-Running Jobs’

[10] CluMon at <http://clumon.ncsa.uiuc.edu/>