Reconfiguration Mechanisms for Virtual Organization using Remote Deployment of Grid Service*

Hyukho Kim[1], Yangwoo Kim[1], Pillwoo Lee[2]
[1]*Dept. of Information and Communication Engineering, Dongguk University, Korea*
[2]*Korea Institute of Science and Technology Information, Korea*
[1]*{hulegea, ywkim}@dongguk.edu*, [2]*pwlee@kisti.re.kr*

Abstract

Grid computing is becoming more and more attractive for coordinating large-scale heterogeneous resource sharing and problem solving. Although Globus toolkit version 4 which is developed based on WSRF can successfully deal with the heterogeneity of resources and the diversity of applications, the current grid environments based on Globus toolkit version 4 have several limitations regarding dynamic reconfiguration of virtual organization (VO) and remote hot deployment of grid services. This paper proposes WIVO (Web-based user Interface for Virtual Organization), the web-based user interface using web service. Using WIVO, user requested sub-VO can easily be configured on top of grid environment based on Globus toolkit version 4, and grid services can also be deployed remotely for better utilization of resources and applications. It is expected that the VO reconfiguration function as well as the remote grid service deployment function provided in WIVO can improve resource and application utilization.

1. Introduction

Grid computing [1, 2, 3] is a technology used to establish web-based Internet computing environment integrating geographically dispersed computing resources. To provide the environment where grid users can corporate safely, efficiently searching and sharing the distributed resources, the concept of VO (Virtual Organization) is used. In order to establish the environment where users can compose and use the VO easily, Grid middleware was developed.

Among them, Globus Toolkit [4, 5], an open middleware, is used most commonly and is regarded as a de facto standard of grid computing. Grid, by adopting web service which uses platform independent transport method, has proposed OGSA (Open Grid Service Architecture) [6]. Since then, a new grid standard called WSRF (Web Service Resource Framework) [7] is defined as the web service standard develops. The Globus toolkit 4 (GT4), properly embodied for WSRF, provides WSRF-based web services as well as JAVA-based hosting service for stateful resource. GT4 container operates on each host and a VO is organized through the distributed GT4 containers. A common method to execute an application on a certain host is to implement the application as a WSRF based service and deploy the embodied service to GT4 container which exists in each host. Although GT4 processes a variety of applications and different types of resources successfully using the JAVA-based hosting environment and platform independent web service protocol, it has some limitations when it comes to provide

* This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. F01-2007-000-10032-0).
Yangwoo Kim (Corresponding Author)

users with grid resources. The dynamic features of grid should adequately allocate grid resources to users who need them. In addition, current services should be provided continuously even in cases where grid resources leave VO or in case of resource instability, overload or interruption. However, the present GT4 version only can use the services deployed manually by local administrator. Moreover, for a new service, the administrator should deploy it manually on the host where the service is to be deployed and reactivate the container. Therefore, the resources existing at a VO can't be used efficiently and the users are not allowed to use resources by reorganizing them dynamically.

This paper proposes WIVO (Web-based user Interface for Virtual Organization) which is a user interface for the applications based on WSRF and for reorganization of VO that provides resources. The WIVO is also a framework used to deploy services to the containers on specific hosts. WIVO provides the following functions based on GT4 for an effective supply of resources. First, it provides the function to build a VO as well as a Sub-VO using the resources in the VO. Second, it provides the monitoring function for resources and services existing in each VO. Third, it provides the function for remote deployment of grid service to use applications and resources. WIVO proposed in this paper is implemented as a web service and can be used not only in GT4 container but also in every JAVA-based host environments based on WSRF specification.

The rest of the paper is organized as follows. In Section 2, we briefly explain the related work. Section 3 presents the overall architecture and main functionalities of WIVO. Then the implementation of WIVO is shown in Section 4. Finally, we conclude in Section 5.

2. Related work

Grid is a highly distributed and geographically dispersed computing environment, in which numerous domains could be involved. It is also highly desirable for a user to deploy its services into remote service containers for various purposes. Many key issues such as remote and hot deployment and service reconfiguration in grid computing have been extensively studied in the past years.

Globus Toolkit which is the most famous grid middleware has supported service-oriented grid computing based on OGSA since version 3. But even in the updated release version 4, remote and hot service deployment is not supported. Grid service is actually built on Web service, and extended to include functions such as state and life cycle management. For Web services, several middleware, such as Apache Axis [20], JBOSS [21], and Microsoft .NET [22], have partly implemented dynamic service deployment. However, Web services are normally stateless, so middlewares based on Web service cannot be applied to grid environments. Also, most of them only consider local deployment.

Several researches have dealt with dynamic deployment mechanisms for grid services. PlanetLab [8] is a large-scale distributed platform which integrates Internet-wide resources to obtain a massive computing power. Although PlanetLab has been used as a testbed for computer networking and distributed systems research, it can be run only on Linux platforms and it needs a special configuration. Moreover, PlanetLab doesn’t support any systematic mechanism for deploying a new service or migration of existing services. Weissman et al. [9] proposed a new Grid architecture that supports dynamic deployment and undeployment of services according to the service requests. Smith et al. [10] proposed a Service based Ad Hoc Grid that provides peer-to-peer based node discovery, automatic node property assessment, and hot deployment of services for running systems. However, these solutions have some restriction that they provide dynamic service deployment mechanisms under OGSI-based Grid environments only. Open Grid Service Infrastructure (OGSI) is an extension of Web Services specification for implementing OGSA [13]. Since a new type of Web service called Grid service is introduced supporting stateful services, OGSI which does not comply with Web services standards is deprecated in favor of WSRF.

Recently, HAND (Highly Available Dynamic Deployment) [11] suggested a dynamic service deployment mechanism for GT4 container. However HAND needs a certain function of modifying the internal data of GT4 container with a special privilege. Moreover, the HAND implementation cannot be applicable to other system environments generally. DynaGrid [12] provides the GT4 based dynamic service deployment and migration of stateful resources under Grid environment. So DynaGrid can build a Grid environment by providing a function of dynamic service deployment under any Java-based host environments. DynaGrid, however, cannot support the reconfiguration mechanism of VO. These and a few other projects [14, 15] are the main dynamic deployment efforts for grid applications. Some of them clearly are not intended for a WSRF-enabled service-oriented architecture. Moreover, although some have implemented a service-oriented dynamic deployment, they do not address dynamic features of grid environment establishment and interoperability for other systems.

WIVO effectively provides the dynamic services deployment, and the dynamic VO reconfiguration based on user security level. In addition, we designed and implemented the User Interface for Grid Service Development in the previous work. So users can develop Grid Services using UI4GSD [16], and the implemented services can directly be deployed into any GT4 containers in the VO using WIVO. As a result, our approach can maximize the resource availability as well as the services usage.

3. Web-based User Interface for VO

3.1. Overall Architecture

As you can see in Figure 1, the overall architecture of WIVO is divided into three components. First, it is the VOManager(VO Management) which manages VO creation and deletion, Sub-VO creation and deletion, etc. Second, it is the MonitorManager(Service Deploy Status Monitoring and Resource Status Monitoring) which monitors the status of resources and grid services. Third, it is the ServiceDeployer(Service Management and Remote Deployment Service) which manages deployment and undeployment of grid services.

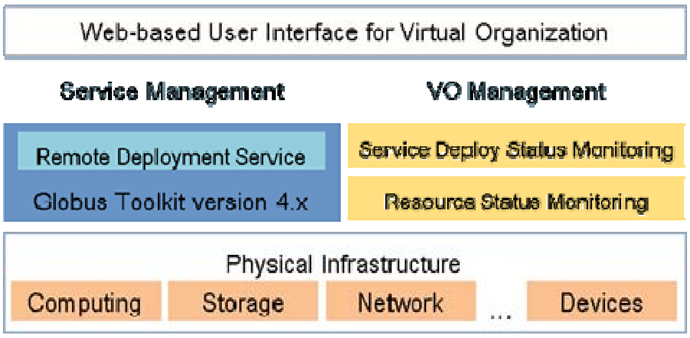

Figure 1. Overall WIVO Architecture

VOManager manages the existing VOs, and a new VO creation as well as participation to an existing VO when users access a VO using the interface provided. VOManager also conducts the function of deleting a VO when a user leaves the VO or the VO is no longer needed. VOManager also conducts the function which builds and deletes a SubVO based on services in the nodes in the existing VOs.

MonitorManager monitors the status of hosts in all VOs and SubVOs, and provides the integrated information to the user. It monitors not only the host status in VOs, but also the deployment status of grid services. When user needs grid resources, MonitorManager provides the current status information to users so that the maximum utilization of grid resources is possible.

The remote deployment and undeployment of grid services are not yet supported by GT4 container. However, those services must be supported for an efficient use of the grid resources. In this paper, this function is conducted by ServiceDeployer. ServiceDeployer plays a role of deploying and undeploying grid services at local and remote hosts. As a result, WIVO supports an efficient environment by providing those services for better utilization of grid resources.

3.2. Grid Service Development and Deployment

Before deploying a Grid service into GT4 container, we have to make the Grid service first. For that use, we designed and implemented an easy and efficient user interface (UI4GSD, User Interface for Grid Service Development) for development of grid services based on GT4 in the previous research [16]. The general method of developing a grid service requires the typical five step process. Each step requires professional knowledge and many complex processing. However, UI4GSD doesn't require a user to learn the expert knowledge on grid services, web services, Java programming skills, etc. Simply by following the steps guided in the menu and providing some input through the GUI provided, one can easily generate a grid service from a Java application. Even more, the UI4GSD has a simple and intuitive user interface and it can be installed on various environments. Therefore, the UI4GSD provides an easy and convenient development environment for developing grid services so that the efficiency and the convenience in developing grid services can be improved. As a result, user can implement grid services with UI4GSD, and then user can deploy or undeploy grid services into GT4 container using WIVO. The deployment and undeployment process of the grid service with WIVO is identified in Figure 2.

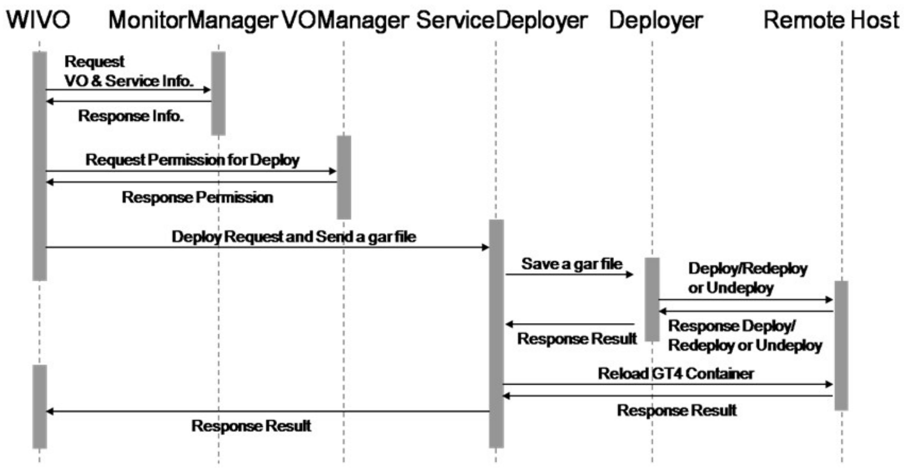

Figure 2. Grid Service Deployment Process

The first step of grid service deployment or undeployment is to request the status information of VO and grid service through MonitorManager. Then the all of the deployed services in a host can be identified and, a host to deploy a grid service can be selected. As a next step, the access permission of that host is confirmed by VOManager before a service can be deployed or undeployed. If a right permission is acquired, a service deployment or service undeployment operation can be done by ServiceDeployer. ServiceDeployer inserts the service information into DBMS for service management, and then it requests an actual service deployment to Deployer. After checking if the requested work is service deployment or service undeployment, Deployer calls RemoteDeployService. Then a grid service deployment or undeployment operation is conducted. Once the requested work is over, it returns the result.

3.3. Monitoring Service

Underlying resources form the physical infrastructure for components and applications in the upper layer. The status of underlying resources has a great impact on the whole grid systems. It is important to design and implement a service to monitor resource status and provide it to schedulers and other components. Resources in grid environment are in the form of services, and each host in a grid system has a service container installed. Therefore, in order to monitor the resource status, we need to deploy a resource monitoring service that we developed into the container. Then we can use the resources and services efficiently by monitoring their status. The resource monitoring service and the grid service monitoring service are implemented as a WSRF-compliant service so that it can be in any VOs based on GT4.

Figure 3 presents the architecture of resource monitoring service and grid service monitoring service. The resource status we are concerned with includes both static and dynamic information. The static information involves OS type, middleware type, file system, IP address, host name, CPU information, memory size, hard disk size, etc, while the dynamic information includes available memory, CPU load and available disk space. And the grid service monitoring service monitors all grid services in each host.

The information of service status includes the service deployment status (deploy or undeploy), service owner, and service deployer, etc.

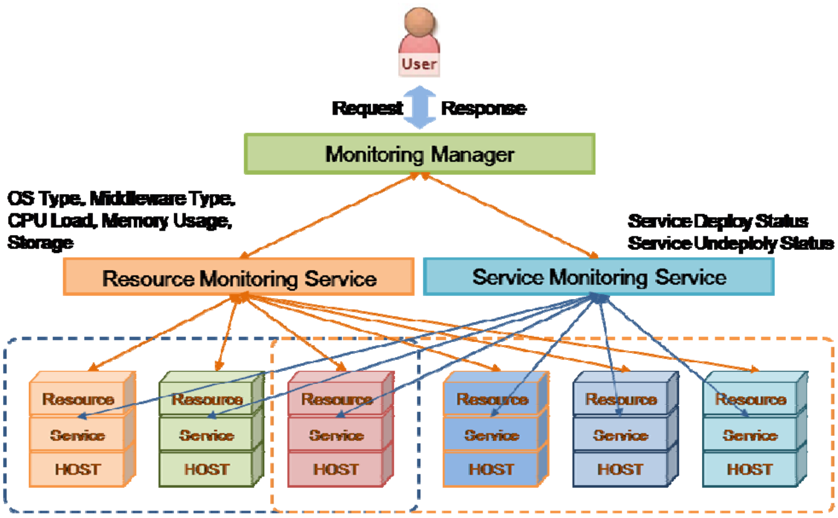

Figure 3. Monitoring Service Architecture

Even though there are some good monitoring tools available like Ganglia [17] and NWS (Network Weather Service) [18, 19], our tool is not focused on performance but focused on providing integrated monitoring services for both resources and grid services to improve utilization efficiency.

3.4. VO Reconfiguration

The grid user generally uses the grid resources in VO according to the security level and/or user level after joining VO. So, if a user wants to use the grid resource in another VO, the user has to join another. However, the security configuration and the joining process involved are very difficult and complicated. So we developed the VO management interface for efficient use and management of grid resources and services in the VO. Also in order to increase the utilization of the grid resources and services, we proposed a method for reconfiguring a VO based on the existing VO. There are three VO reconfiguration methods.

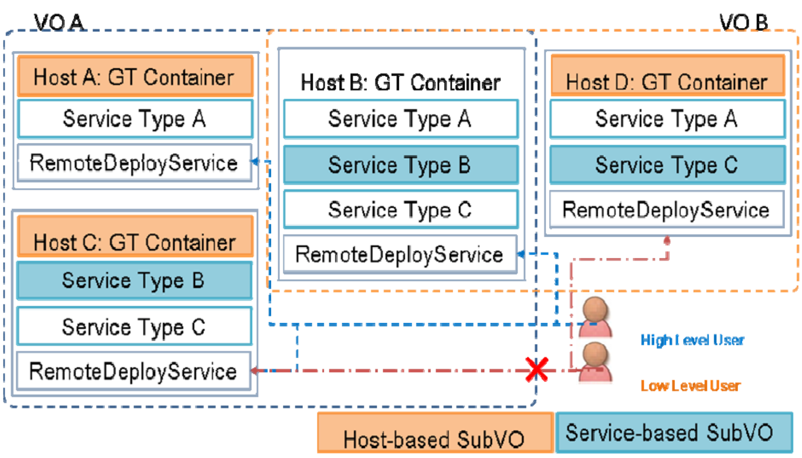

Figure 4. VO Reconfiguration Mechanism

VO reconfiguration methods shown in Figure 4 are based on service, host, and both (hybrid).

- Service based Sub-VO reconfiguration method

This method makes a Sub-VO based on services in VOs. If security level and user level are high, the user can use services in other VOs other than the VO he is participating in. However, a low level user can only use services existing in his VO. This method has an advantage that user can select services of his need.

- Host based Sub-VO reconfiguration method

This method is similar to the service based method. However, the host based method has an advantage of being capable to use the entire host resources. This method also can deploy the needed grid service into the selected host. Similarly to service-based method, according to security level and user level, the available resources are different.

- Hybrid based Sub-VO reconfiguration method

This method can build Sub-VO based on hosts and services. User can build a Sub-VO with hosts and services according to security level and user level. This method can use the host's resources and the needed services, which is the advantage comparing to the other reconfiguration methods mentioned above.

4. Implementation

The proposed web-based user interface implementation is shown in Figure 5. The overall composition of the user interface is subdivided into three areas. First, the top left side in Figure 5 shows user information. As one can see, it shows user ID, host name, user's VO name, and user usage level. Second, the bottom left side shows overall VO list. Depending on the user level, the VO list is presented differently. If the user level is high, that user can identify all the VOs, but if the level is low, then the user can only identify the VO that he has participated in. The right hand side shows the available resources and services.

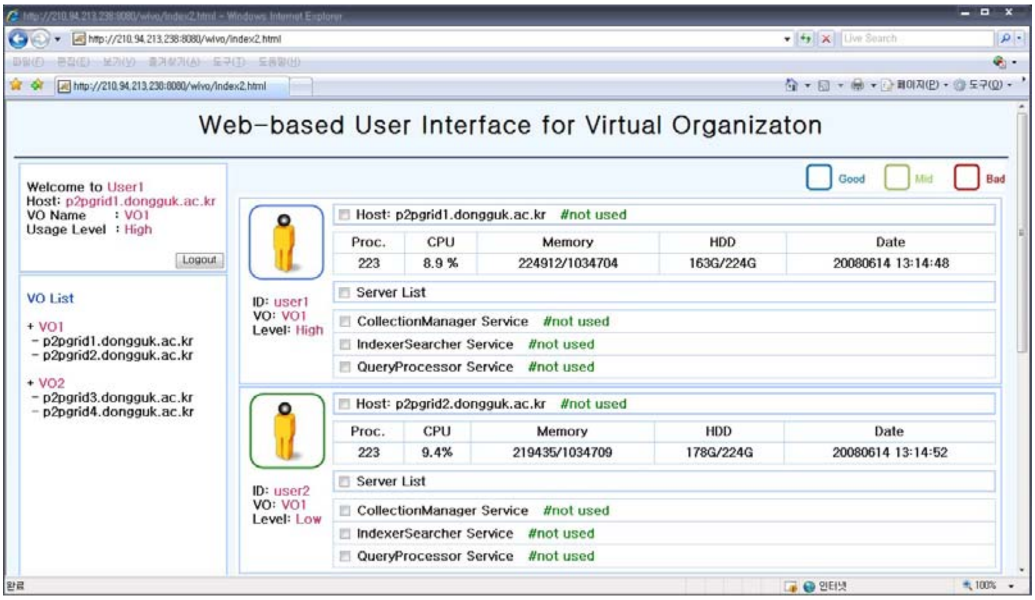

Figure 5. WIVO Implementation

If the user logs into the system through a normal process, the user can see the interface given according to his user level and the security level. Figure 5 is user interface for a user whose user level is high. Currently, as shown in Figure 5, WIVO has two VOs such as ‘VO1’ and ‘VO2’, each with two hosts. And the right side in Figure 5 presents the status information that shows who is using what resources and the services. Also a user can create a new Sub-VO based on the available resources and services, by clicking the checkbox button.

5. Application Use Case: Grid-IR Service Model with WIVO

This is an application use case showing how the proposed WIVO can be applied to Grid-IR(Information Retrieval) service model. In order to understand Grid-IR application and its’ requirements, we define this application use case through the Grid-IR services: Collection Manager service, Indexer Searcher service, and Query Processor service [23, 24].

5.1. Grid-IR System Model

GIR applies the tools of grid computing to IR to provide a common infrastructure for distributed IR. It also brings the capabilities of IR to grid computing. The basic idea of GIR is to define an IR system in terms of three functional components, implemented as grid services: Collection Manager service (CM), Indexing/Searching service (IS), and Query Processing service (QP). These services are autonomous, and being grid services, they are distributed. Since they can be created dynamically and in any combination of Virtual Organizations, they can be used to create new IR systems or link existing ones together in an interoperable network of IR services. Figure 6 shows the overall architecture of Grid-IR System.

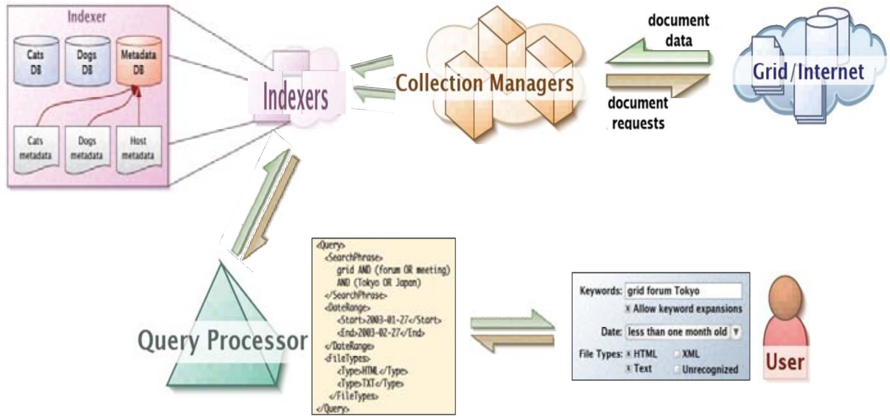

Figure 6. Grid-IR Architecture

5.2. Grid-IR Services: CM, IS, QP

This section provides an overview of the Grid-IR interfaces and the relationships among them. When we refer to a “Grid-IR service,” we mean a grid service that implements one or more Grid-IR interfaces. And each Grid-IR service implements a single Grid-IR interface.

The Collection Manager (CM) is concerned with collecting and managing source documents intended to be indexed, searched, and retrieved via one or more Grid-IR services. It does this by retrieving specified documents from various locations (remote or local), preprocessing and managing them in a local store, and providing them to clients according to specified rules. “Client” here normally refers to the Index/Search service, which uses CM’s as the sources of documents that will be indexed and searched.

The Index/Search interface (IS) conducts jobs which are assemble or indexing of documents from the variety of existing sources. The IS also creates and manages data structures needed to provide searching capabilities. (These data structures are normally referred to as an “index.”) The IS also exposes a “search interface,” which is a set of functions that allow searching of the indexed document collection.

The Query Processor (QP), the third architectural component of Grid-IR, is responsible for managing queries and result sets (i.e. response sets from searching operations). As mentioned earlier, QP provides a search interface identical to that of IS. In other words, for purposes of searching, QP is a “virtual IS.” It has no indexing capability of its own (and provides no indexing interface) but serves as a search only gateway to one or more IS’s.

5.3 Grid-IR Service Model with WIVO

This section describes an example showing how the proposed WIVO can be applied to Grid-IR System. As mentioned earlier, Grid-IR system can be constructed by using the services such as CM, IS, and QP.

After a user joins in WIVO, the user can select the resources and services that he/she needs. The selected resources and services are presented in the bottom left of Figure 7. The first host (p2pgrid1.dongguk.ac.kr) is the one that user is logged in. The user

('user1') has the 'Collection Manager service' and 'Indexer Searcher service' in user1's host. Additionally user1 uses the 'Query Processor service' in other host (p2pgrid2.dongguk.ac.kr).

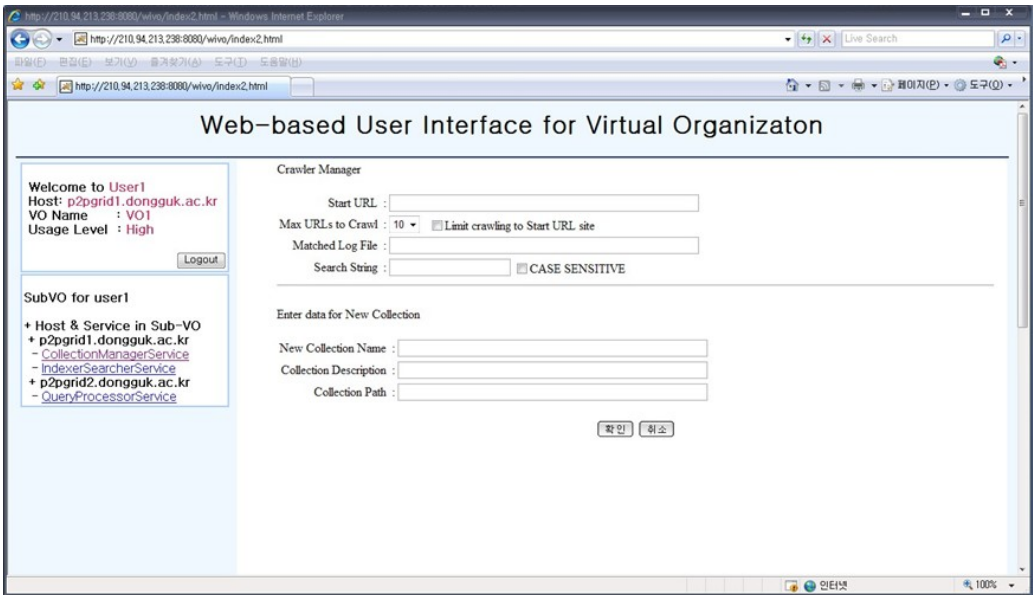

Figure 7. Grid-IR Service Model with WIVO

Each Grid-IR service processes their jobs, and returns the query result to the user. The work flow of Grid-IR service model which is built with the selected resources and services through WIVO system is as follows.

- User joins the WIVO system.
- If user authentication process is completed, user can select the resource and service according to user's level.
- If the resource and service selection is completed, user can get them by clicking the 'Submit' button.
- The acquired resource and service can be presented in the bottom left side in Figure 7.
- User can collect the needed data through CM service.
- If new collection is created, user can make index files through IS service.
- If index files are created, user submits a query to the index through QP service.
- User confirms the query result.
- User returns resources and services in sub-VO to system if job is completed and no longer needed.

6. Conclusion and Future Work

Grid uses VO for management and integration of heterogeneous resources. However VO needs the unified management interface because VO is built based on heterogeneous resources. Although GT4-based VO management service is presently being supported, it has some limitations in building sub-VO and in deploying grid services remotely. Therefore we designed and implemented the user interface based on web technologies for easy management of VO and for the efficient use of grid resources.

So grid user can build a new VO or join an existing VO using web-based user interface. As a result, the resource and grid services in a VO can be managed efficiently. Moreover, grid users have to know the status of grid resources and services, so that they can use them adequately. Thus, we implemented the status monitoring function for the grid resource as well as for the grid services in a VO. Finally, we also developed remote deployment (undeployment) function for grid services which is not supported in GT4. Using this function, we can easily deploy/undeploy services remotely to any hosts we want. As a result, the proposed web-based unified interface provides the better and convenience mechanism for VO management and can use the grid resources and grid services much more efficiently.

Although the proposed web-based interface has several advantages on the use aspect of grid resources, still there are more areas for further improvements in the future. First, we have to develop the web-based UI4GSD which is developed by Java Swing API currently. This will enable us to develop grid services on the web, and deploy them into the host directly, again, on the web. Second, it has to be able to monitor the operation status of deployed services in GT4 container. Once the operation status and the resource status are monitored, the work can be migrated to the adequate host for better execution, even in case of system failure. Consequently, it will improve reliability and efficiency in executing grid services, and furthermore, it will contribute to the development of the grid technique and industry as well.

References

[1] I. Foster, "What is the Grid? A Three Point Checklist", GRID Today, July 20, 2002.

[2] I. Foster and C. Kesselman, TheGrid2: Blueprint for a New Computing Infrastructure, Morgan Kaufmnn, 2nd edition, November 18, 2003.

[3] Fran Berman, Geoffrey Fox, and Anthony J.G. Hey, Grid Computing: Making The Global Infrastructure a Reality, John Wiley & Sons, April 8, 2003.

[4] I. Foster, “Globus Toolkit Version 4: Software for Service-Oriented Systems”, FIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2006.

[5] Ian Foster, A Globus Toolkit Primer (Draft), http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf, April 26, 2005.

[6] I. Foster, C. Kesselman, J. Nick, S. Tuecke, “The physiology of the grid: an open grid services architecture for distributed systems integration”, Open Grid Service Infrastructure WG, GGF, June 2002.

[7] OASIS Web Services Resource Framework (WSRF) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf.

[8] L. Peterson, T. Anderson, D. Culler, T. Roscoe, “A blueprint for introducing disruptive technology into the internet”, in: Proceedings of ACM HotNets-I Workshop, 2002.

[9] J. Weissman, S. Kim, D. England, “A framework for dynamic service adaptation in the grid: next generation software program progress report”, in: Proceeding of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2005.

[10] M. Smith, T. Friese, B. Freisleben, “Towards a service-oriented Ad Hoc grid”, in: Proceedings of the 3rd International Symposium on Parallel and Distributed Computing/the 3rd International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks (ISPDC/HeteroPar), 2004, pp. 201–.208.

[11] L. Qi, H. Jin, I. Foster, J. Gawor, “HAND: highly available dynamic deployment infrastructure for globus toolkit 4”, http://www.globus.org/alliance/publications/papers/HAND-Submitted.pdf.

[12] Eun-Kyu Byun, Jin-Soo Kim, “DynaGrid: A dynamic service deployment and resource migration framework for WSRF-compliant applications”, Parallel Computing 33 (2007) 328–.338

[13] S. Tuecke, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling, P. Vanderbilt, Open Grid Service Infrastructure(OGSI) version 1.0, http://globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf, June 27, 2003.

[14] P. Watson and C. Fowler, "An Architecture for the Dynamic Deployment of Web Services on a Grid or the Internet", Technical Report Series, CS-TR-890, University of Newcastle upon Tyne.

[15] Hailong Sun, Wantao Liu, Tianyu Wo, and Chunming Hu, "CROWN Node Server: An Enhanced Grid Service Container Based on GT4 WSRF Core", Grid and Cooperative Computing Workshops, 2006. GCCW '06 Fifth International Conference, Page(s):510–517, Oct. 2006.

[16] HyukHo Kim, Yangwoo P. Kim, Pilwoo Lee, "Design and Implementation of User Interface for Grid Service Development based-on Globus Toolkit4", ICCSA 2008 Joint Workshop, December 13-14, 2007.

[17] M. L. Massie, B. N. Chun, and D. E. Culler, "The Ganglia Distributed Monitoring System: Design, Implementation and Experience", Parallel Computing, Vol. 30, Issue 7, July 2004.

[18] R. Wolski. "Dynamically forecasting network performance using the network weather service". Cluster Computing, 1:119-132,1998.

[19] R. Wolski, N. Spring, and J. Hayes. "The network weather service: A distributed resource performance forecasting service for metacomputing". Future Generation Computer Systems, 15(5-6):757-768, 1999

[20] Apache Axis, website: http://ws.apache.org/axis/.

[21] M. Fleury and F. Reverbel, "The JBoss Extensible Server", presented at ACM/IFIP/USENIX International Middleware Conference, 2003.

[22] Microsoft.NET, website: http://microsoft.com/net/.

[23] Grid Information Retrieval Research Group, http://www.gir-wg.org/

[24] N. Nassar, K. Gamiel, and J. Morris, "Grid Information Retrieval Architecture", http://www.gir-wg.org/papers/Grid_Information_Retrieval_Architecture.pdf, September 2003.

Authors

Hyukho Kim

He received the M.S degrees from Dongguk University in 2003. Since 2003, he has been with Dongguk University as a Ph. D student. His current interest areas are "Grid Computing", "Cloud Computing" "P2P Computing" and "Parallel and Distributed Computing".

Yangwoo Kim

He received the M.S and Ph.D degrees from Syracuse Univ. in 1986 and 1992 respectively. Since 1996, he has been with Dongguk University as a professor. His current research interest areas are "Grid Computing", "Cloud Computing" "P2P Computing" and "Parallel and Distributed processing".

Pillwoo Lee

He received the M.S and Ph.D degrees from Univ. of Tsukuba in 1991 and 1998 respectively. Since 2000 he has been Korea Institute of Science and Technology Information(KISTI) as a Principal Researcher. His current research interests are "Grid Computing", "Grid Middleware" and "Parallel and distributed processing"