The Design and Implementation of Extensible Information Services

Guiyi Wei¹, Guangming Wang¹, Yao Zheng², and Wei Wang²

¹ Zhejiang Gongshang University, Hangzhou, 310035, P. R. China
weiguiyi@tom.com, huc1@mail.hz.zj.cn

² Center for Engineering and Scientific Computation, Zhejiang University,
310027, P. R. China
{yao.zheng, weiw}@zju.edu.cn

Abstract. High-performance distributed computing often requires careful selection and configuration of computers, networks, application protocols, and algorithms. In grid environment, the participators of domains provide site information by default that may be not adequate for resources manager to achieve optimal performance. In this paper, an Extensible Information Service Architecture (EISA) is described to resolve this problem. EISA supports a virtual organization participator's autonomous control of local resources. It consists of three layers, the registration and deployment layer, normalization and organization layer, and information collection layer. Dynamic information services in EISA could be deployed and extended easily. EISA improves the extensibility of grid information services and the efficiency of resources utilization.

1 Introduction

The term “Grid” denotes a distributed computing infrastructure for advanced science and engineering [1] and is distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases high-performance orientation [2]. The real and specific problem underlies the Grid concept is coordinated resources sharing in dynamic, multi-institutional virtual organizations. As Grid becoming a viable high performance computing alternative to the traditional supercomputing environment, various aspects of effective Grid resources, such as proper scheduling and efficient resource utilization scheme across the Grid can lead to improve system and a lower turn-around time for jobs.

High-performance distributed computing often requires careful selection and configuration of computers, networks, application protocols, and algorithms. These requirements do not arise in traditional distributed computing. The situation is also quite different in traditional high-performance computing where systems are usually homogeneous and hence can be configured manually [4]. In grid computing, neither defaults nor manual configuration is acceptable. Using default configurations often can not result in acceptable performance, and manual configuration requires steering of remote systems that is usually impossible. We need an information-rich approach to configuration in which decisions are made, whether at compile-time, link-time, or

run-time [5], based upon information about the structure and state of the site on which a program is to run.

In grid environment, the participators of some special domains provide site information that is discovered by default. In some cases, default information may be not adequate for the scheduler to achieve optimal result. The most important part of a good scheduler is an excellent algorithm. The input arguments and objectives of the algorithm are subject to different applications. The computation intensive applications, such as Monte-Carlo simulations, need high speed computing power. The I/O intensive applications, such as distributed data sharing system, need large capacity storage. The communication intensive applications, such as computational fluid dynamics simulations, need large bandwidth and low latency. To achieve the optimal objectives of these applications, adequate information about distributed nodes in the grid environment must be acquired. Nowadays the information is discovered and retrieved by grid information services, such as MDS, LDAP, etc., which is provided by grid middleware. Typically, the original information is generated by many different elementary programs, which are always encapsulated as web services or grid services. According to the difference of operation systems, networks, storage management and other local resource management policies, some programs may not always be resided in all of the domain sites. It is possible that the schedule result is not optimal because of the information scarcity. In this paper, an Extensible Information Service Architecture (EISA) is described to resolve above problems, which support dynamic deployment of scalable information service.

2 Related Work

The Multidisciplinary ApplicationS-oriented SIMulation and Visualization Environment (MASSIVE) project adopted EISA in its information service infrastructure. The grid created for the MASSIVE utilizes resources located at the CESC and relevant departments of ZJU. The resources at other institutions of high performance computing are to be added to it recently. The grid currently uses Globus Toolkit 2.x as the middleware to enable the resources to be accessed after authentication and authorization.

Using OGSA architecture, the MASSIVE project enables grid-based CFD and CSM by some specified services. Services for geometry pre-processing and mesh generation, migration and execution of application programs on remote platforms, collaborative visualization, and data analysis, form the basis of a globally distributed virtual organization. A typical usage would be the generation of a mesh using a mesh generation service on an IRIX platform, the solution of CFD and CSM problems with the meshes previously created on a PC cluster, and the collaborative visualization of the numerical results with equipments such as a stereo tiled display wall and a BARCO stereo projection system at the CESC.

To minimize the risks of remote execution of simulations, the MASSIVE Grid makes it possible for the user to keep the codes and data at a local machine, and to transfer them to a remote machine only when it is to be executed, and it is then erased from the remote site after execution. Services will be provided to support the secure

migration, remote execution, deletion of the application program, and the secure return of the application results to the user [6].

Visualization of large scaled scientific data sets using the Grid is an important aspect of the MASSIVE project. To make effective use of all of the devices, a powerful information service faculty is needed.

3 EISA

The grid participator provides its local static resources information and dynamical status of devices' activities with core resource information service supported by the adopted grid middleware. MDS [4] is the information service provided by Globus Toolkit in grid environment.

The variety of resources leads to the variety of Information Providers (IP). EISA operates over the wide area and supports a VO participator's autonomous control of local resources. It consists of three layers, a registration and deployment layer, normalization and organization layer, and information collection layer (as depicted in figure 1).

Fig. 1. The EISA Architecture

The registration and deployment layer, which provides Virtual Organization interaction, is interconnected with a Grid Information Index Service (GIIS) server. The normalization and organization layer consists of two components, information normalization module and IPs management module. At the information normalization module, information is collected from native or temporarily deployed IPs, and normalized into annotated resource information that is tailored to a client's needs. At the IP management module each IP is migrated, configured and activated within its

control. The information collection layer is the IPs' container, which is extensible and capable of adopting external information discovery programs and encapsulating them into grid services.

3.1 The Registration and Deployment Layer

The registration and deployment layer provides mechanisms to support interaction among GIIS servers, local host and EISA clients. It can also enforce the site's security policies. Logically the registration and deployment layer is depicted in Figure 2, where a range of clients interacts with it to request resource information and deploy new IPs. The registration and deployment layer register the site's contact details, and the categories of resource they manage to the GIIS servers of the VO. Clients locate the site by searching GIIS registry. Thereafter, clients directly interact with registered sites, query it, and subscribe for reservation and notification when conditions changed.

Fig. 2. Logical representation of the registration and deployment layer

The EISA is based on a common Grid Security Infrastructure (GSI) that provide security functionality. GSI defines a common credential format based on X.509 identity certificates and a common protocol based on transport layer security (TLS, SSL) [7]. The registration and deployment layer security can be configured with GSI to restrict client access to a site, based on VO. In some cases, there may be additional security requirement for special information retrieving. A site policy may decree that the requests from clients that do not satisfied are blocked, or the requests are only permitted to retrieve cached information for a subset of resources.

3.2 The Normalization and Organization Layer

The information normalization module and IPs management module are two major components of the layer.

3.2.1 Normalization

Normalization is the process of transforming raw data from diverse different IPs into information that is well defined and described in a consistent and standard format (as

depicted in figure 3). Normalization is used to provide a homogeneous view of resources [3]. The raw data from IPs may be in many different formats. They could be: (1) name-value pairs of device's attributes and its value; (2) encoded in binary, ASCII, Unicode, or other formats; (3) represented using comma-delimited fields, or a metadata language like XML; (4) Context requiring knowledge to understand it.

XML naming schemas provide a way to enforce normalization requirements. A naming schema is an abstract description of the name, unit, meaning, organization and relation of attributes that compose a given entity within a domain of reference [3]. For example, the GLUE [8] naming schema provides a common conceptual model used for grid resource monitoring and discovery. GLUE provides common terms of reference and semantics that define how computing, storage and network resources, and their attributes, should be labeled, described and related.

Fig. 3. The normalization process

The normalization process includes four steps described as follow.

1. Discover raw data of the resources.
2. Encapsulate raw data into class style.
3. Transform information in class style to XML format and migrate them into a single XML document.
4. Register the migrated information to reservations and cache it.

3.2.2 Information Provider Management

The IP management module controls the deployment of each IP. At this module, we define a deployment in this way:

$$D < P, IN, OUT, S > \quad (1)$$

Here:

- **D** is the name of the deployment that is a character string and identifies itself exclusively.
- **P** is the name of information provider. P includes the executable program, its requirements of running environment (such as OS type, OS version, and necessary library), and its version. P is given in XML style. It can be denoted as:

$$P = (\text{exe_name}, \text{env}, \text{ver}) \quad (2)$$

- **IN** is the input parameters set of P. It includes parameters name, parameters data type, and query rules. It can be denoted as:

$$\text{IN} = (<\text{para_name}, \text{data_type}>, <\text{query_rules}>) \quad (3)$$

- **OUT** is the output raw data of P. Each data is stored by a couple of value as <attribute, value>. For example, <FreeMemory, 54.6> means that there is 54.6 megabytes free RAM memory.
- **S** is the collection of semantics, knowledge, and transformation rules.

At IP management module, management jobs are performed using a serial of instructions. Each instruction consists of an *operator* and an *operand*. Nowadays, *Operator* includes *import*, *upgrade*, *remove*, *register*, *activate*, and *deactivate*. *Operand* is the name of a deployment.

3.3 The Information Collection Layer

The information collection layer consists of IPs for querying local resources, performing reservation, concurrent handling, and retrieval of historical data, and special information access policies (as depicted in figure 4).

Fig. 4. The information collection layer

A key element of the information collection layer is the IP pool that is used for interacting with a variety of IPs and maintaining a reservation refresher with the help of time rules. Clients can query an arbitrary resource in a standard way and retrieve information described in a standard format. Clients do not require prior knowledge of

resources; IPs can be queried to return details of naming schemas that are appropriate to the selected resource.

Figure 4 shows the steps that occur when a client queries a resource:

- Step 1: When query come, information collection layer retrieve the reservation.
- Step 2: Return information hold in reservation. If it fulfills requirement, results will be returned to the upper layer. Otherwise it continues retrieval with step 3.
- Step 3-5: Run proper IPs to perform discovering. Then IPs return results.
- Steps 6: Perform reservations.

4 Implementation

Besides the elementary information, EISA should provide additional information of running program and their result. For example, network bandwidth, network data flow, remote mapped user, remote mapped working directory and visualization devices information are additional that should not be provided by default information service of VO sites. The programs have been developed for discovering information above, encapsulated into OGSA grid services, and can be deployed in the EISA.

The implementation of these additional information services should fulfill the following requirements.

- The IP's output format should be compatible to LDIF of LDAP.
- EISA uses security component of Globus Toolkit. It is also compatible with OGSA standard, so that they can be integrated seamlessly into Globus-based grid application system.
- User-defined IPs should not conflict with those existing services, or not it will work improperly.
- New IPs should be flexible enough. Both the information and the way they integrated into EISA system can be customized.

Each attribute item in a resource object is stored by a couple of values as <attribute, value>. All information is encapsulated as <node_id, res_info>, where the "node_id" is a host name or IP address of the host, and the "res_info" is an instance of class "MdsResult". The C++ class definition of "MdsResult" is shown below:

```
typedef map<const string , string> InfoMap;
class MdsResult {
    map <string , InfoMap *> entry
    public:
        string getValue(string node, string attr); }
class VGridMds {
    MdsResult result;
    LDAPServer * server;
    public:
        search();
        search(string filter, int scope)
        search(string basedn, string filter, int scope)
        search(string basedn, string filter, string[]
attrToReturn, int scope)
```

```

connect();
disconnect();
update();
getResult(); }

```

5 Conclusions

Resource information service is a vital part of any Grid software. The variety of resources leads to the variety of Information Providers. This paper describes an Extensible Information Service Architecture that operates over the wide area and supports a VO participant's autonomous control of local resources. EISA consists of three layers, a registration and deployment layer, normalization and organization layer, and information collection layer. It implements dynamic deployments of information services for additional information search with high-level extensibility and security.

Acknowledgements

The authors wish to thank the National Natural Science Foundation of China for the National Science Fund for Distinguished Young Scholars under grant Number 60225009. We would like to thank the Center for Engineering and Scientific Computation, Zhejiang University, for its computational resources, with which the research project has been carried out.

References

1. I. Foster, and C. Kesselman (eds.): *the Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, (1999)
2. Ian Foster, Carl Kesselman, Steven Tuecke: the Anatomy of the Grid: Enabling Scalable Virtual Organizations, *Int. J. Supercomputing Applications*, 15(3), (2001)
3. Mark Baker, Garry Smith: Ubiquitous Grid Resource Monitoring, *Proceedings of the UK e-Science All Hands Meeting 2004*, 61-68, (2004)
4. Steven Fitzgerald, Ian Foster, Carl Kesselman, et al.: A Directory Service for Configuring High-Performance Distributed Computations, *Proceedings of the 6th International Symposium on High Performance Distributed Computing (HPDC '97)*, p.365, (1997)
5. D. Reed, C. Elford, T. Madhyastha, E. Smirni, et al.: The Next Frontier: Interactive and Closed Loop Performance Steering, *Proceedings of the 1996 ICPP Workshop on Challenges for Parallel Processing*, 20-31, (1996)
6. David W. Walker, Jonathan P. Giddy, Nigel P. Weatherill, Jason W. Jones, Alan Gould, David Rowse, and Michael Turner: GECEM: Grid-Enabled Computational Electromagnetics, *Proceedings of UK e-Science All Hands Meeting*, 2003
7. V. Welch, F. Siebenlist, I. Foster, et al.: Security for Grid Services, *Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, IEEE Press, 2003
8. GLUE-Schema, <http://www.hicb.org/glue/glueschema/schema.htm>, December 2004