CGSP: An Extensible and Reconfigurable Grid Framework*

Yongwei Wu[1], Song Wu[2], Huashan Yu[3], and Chunming Hu[4]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
[2] Cluster and Grid Computing Lab, School of Computer, Huazhong University of Science and Technology, Wuhan, 430074, China
[3] School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China
[4] School of Computer Science, Beihang University, Beijing, 100083, China

Abstract. ChinaGrid Support Platform (CGSP) is proposed to provide grid toolkit for ChinaGrid application developers and specific grid constructors, in order to reduce their development cost as greatly as possible. CGSP extensible and reconfigurable framework, which satisfies the expansion and autonomy requirement of ChinaGrid, is mainly discussed in the paper. In the framework, domain is presented to denote one unit which could provide grid service for end users by itself. Layered structure of domains and corresponding interactive relationship are paid much more attention. CGSP design motivation and simple execution management mechanism are also described in this paper.

1 Introduction

Grid computing [5,6,7] has emerged as an important new field by its focus on large-scale computing resource sharing and coordinated use of resources at multiple sites. It provides approaches to integrate widespread heterogeneous resources into one multi-institutional virtual organization and uniform application interface. It is important to recognize that the resources in this context include computational systems and data storage and specialized experimental facilities.

Based on existing network infrastructure, many grid computing projects have been launched, such as UK e-Science Program [12], Information Power Grid (IPG) [13], TeraGrid [14], China National Grid (CNGrid) [15]. As on national wide grid, China Education and Research Grid (ChinaGrid) [1,2] aims at constructing a public service system for Chinese education and research. Without exception, all of these grid projects paid much more attention on the middleware-the software that enables grid computing/services.

* This Work is supported by ChinaGrid project of Ministry of Education of China, Natural Science Foundation of China under Grant 60373004, 60373005, 90412006, 90412011, and National Key Basic Research Project of China under Grant 2004CB318000.

J. Cao, W. Nejdl, and M. Xu (Eds.): APPT 2005, LNCS 3756, pp. 292–300, 2005.

Grid middleware is the kernel of constructing grid effectively. Great efforts have been made to develop scalable, secure, and highly available grid platforms, which transparently shield the heterogeneities and dynamic behaviors of participants, on top of local operating systems. As one grid middleware, ChinaGrid Support Platform (CGSP) [3,4]aims to provide grid toolkits for ChinaGrid application development and specific grid construction, in order to make them completed more easily and quickly.

Based on Open Grid Service Architecture (OGSA) [9], CGSP is developed according to the Web Service Resource Framework (WSRF) [10] specification for the construction of ChinaGrid from April, 2004. In Jan. 2005, CGSP version 1.0 (CGSP1.0) was released. CGSP1.0 is developed by 5 top universities including Tsinghua University, Huazhong University of Science and Technology, Peking University, Beihang University, and Shanghai Jiao Tong University. It aims to integrate all sorts of heterogeneous resources, especially education and research resources, distributed over China Education and Research Network (CERNET), to provide transparent, high performance, reliable, secure and convenient grid services for scientific researchers and college students.

In this paper, we begin with the design principles of CGSP. Then, CGSP function modules are described simply. In the fourth and fifth parts, CGSP extensible and reconfigurable framework, and execution management across domains are put forward respectively in detail. These provide a whole vision for CGSP framework. Future work and conclusion are discussed at last.

2 Design Principles

CGSP is a grid middleware developed for the construction and evolution of ChinaGrid. The design goal is to reduce the cost of grid construction and application development greatly. In addition to supplying the grid runtime environment of ChinaGrid, CGSP offers a whole set of tools for building portal, deploying grid services and developing various grid applications.

Based on CGSP, ChinaGrid can be constructed into one tree/layered structure. Each node of the tree is a domain. Each domain has the same logic structure and consists of same function modules. They interact through CGSP information center.

In moving forword, CGSP has been guided by a set of key design principles as follows.

- Support localized requirement of ChinaGrid. In fact, it still needs a long way to implement intensive message passing computation over Internet. It makes more proportional local users to use local resources in the grid in order to avoid the reduced performance caused by limitation of network bandwidth and latency over the Internet, and improves the service efficiency of the grid as well.
- Meet the autonomy requirement of ChinaGrid. Each grid application, such as bioinformatics grid [20], image processing grid [22], or computing fluent dynamics grid [21], has its own user and resource management mechanism.

CGSP makes it easier to construct various independent ChinaGrid applications with their own management protocols and mechanisms. Each application grid of ChinaGrid could be a solely domain which could provide specific service for its users by itself. At the same time, they could interact with each other through CGSP high level interactive regulation.
- Scalability of CGSP satisfies the demand of expansion of ChinaGrid. The tree or layered structure of CGSP ensures that ChinaGrid can link more and more universities. Actually our goal is to link up to 100 universities in China in the near future.
- Flexibility of CGSP makes it easier to rebuild the ChinaGrid. Layered structure of CGSP could be easily reconfigurable and rooted by any domain through converting its original parent domain to its child domain.
- Reconfiguration of the tree structure guarantees the integrity and uniformity of a grid system through constructing a global monitoring engine started from any node of the ChinaGrid tree.
- Different from normal grid middleware efforts (GT series [11,8], OMII [16], TinghuaGrid [17] et. al.), CGSP is a platform. Not only does it include the grid running components, such as portal, service container, service monitoring and discovery, file delivery and transformation, but it also provides the grid developing tools, such as programming API, portal constructing tool, service deploying and packaging tools and so on.

3 CGSP Function Modules

CGSP is a collection of cooperative software components. It contains several software modules which can run independently to support each step of development process, execution process, system installation process, and system management. In addition to supplying the components for building grid platform to reduce development cost of ChinaGrid applications, CGSP also offers a whole set of tools for developing and deploying various grid applications.

CGSP logically consists of 6 components showed in Fig. 1.

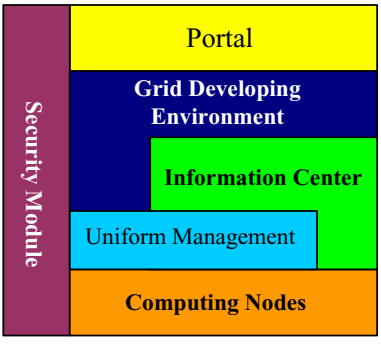

Fig. 1. Logic Diagram of CGSP

- **Portal**: Portal is a web based user interface for defining, submitting, monitoring jobs, and managing personal data, viewing resource information as well.
- **Grid Developing Environment/Toolkit (GDE)**: GDE provides a set of toolkits for grid application's construction and development. They are portal developing tool, service packaging tool, grid-enabled parallel programming tool [19], job definition tool etc.
- **Information Center (IC)**: Information Center, or Service Manager, manages the relatively static information of resources and services in the grid. It provides real-time grid information service for other CGSP function modules under a global uniform information framework. Dynamic information, such as status of grid job, is obtained from uniform management module.
- **Uniform Management(UM)**: UM aims to make heterogeneous resources, computing tools over grid, grid users, grid jobs and job operations managed in a uniform view. It consists of following four items.
  - **Service Container (SC)** is deployed to act as a runtime environment for the installing, deploying, running and monitoring of grid service in the specific node. It also provides support for real-time grid resource monitoring.
  - **Data Management (DM)** mainly provides data delivery and transformation for CGSP. It also implements one global file view and makes all sorts of grid data accessible transparently by grid end-users.
  - **Job Management (JM)** is responsible for the scheduling and monitoring of grid jobs. JM plays a key role in the execution management of grid job. It deploys and submits jobs to the service container of real computing node and starts up them through general running service of CGSP.
  - **Domain Management** ensures the autonomy of each domain of ChinaGrid with focus on user management, log and accounting, and interactive call with other domains for the user identity mapping.
  - **Grid Monitoring** It gathers status data from all the CGSP components and reports the results in structured and standardized documents through WSRF service to correlative modules or users. At the same time, it could notify the ChinaGrid system modules, administrators or users in time when the status is changed.
- **Grid Security (GS)**: GS is in charge of user identity authentication, identity mapping between different domains, service and resource authorization, and secure message passing.
- **Computing Nodes (CNs)**: CN provides real computing power for grid services. It could be a cluster, or PC server or workstation.

4 CGSP Framework Architecture

CGSP consists of a set of well-assorted software packages. Base on it, ChinaGrid can be constructed into a layered tree showed in Fig. 2 (left). Each node of the

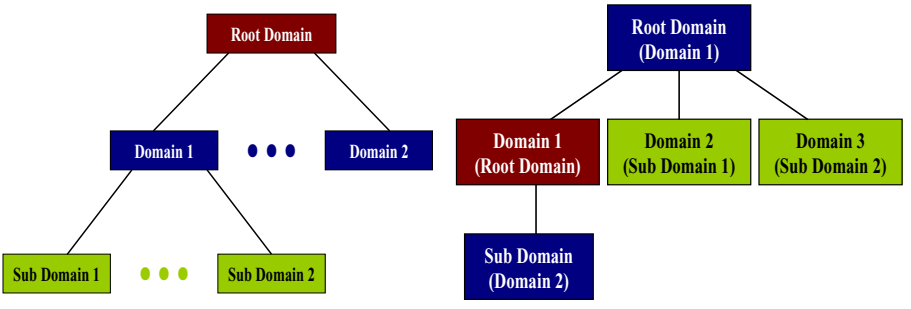

Fig. 2. Tree Structure of ChinaGrid Based on CGSP

tree is a domain. Each domain, maybe root domain, domain or sub-domain, has its own Portal, GDE, IC, UM, GS and CNs. It could provide independent grid service for end users by itself. Each domain can be a specific application grid (bioinformatics grid, image processing grid). It can also be a region grid (Shanghai Grid, Tsinghua Grid). All domains could share one certificate authorization (CA) center, and can also have their own CAs.

In Fig. 2, Root Domain, Domain and Sub-Domain are only used to note their parent-child relationship. The tree Fig. 2 (left) can be reconfigurable easily into another tree as Fig. 2 (right). Each domain could become the root domain of the CGSP if its original parent domain is converted to its child domain same as showed in Fig. 2. Root Domain in the left becomes the Domain 1 in the right. It is clear that the cost of such a reconfiguration is very low with a little effort.

Each domain has the same internal logic architecture and consists of 6 same function modules showed in Fig. 1. At the same time, the module interactive relationships in one domain can also be gotten from Fig. 1. Same as Portal, GDE must get the support from UM and IC, but no direct interaction with CNs. As the grid middleware kernel, UM and IC are called each other and interactive with the real CNs. GS always is closed to all components of CGSP.

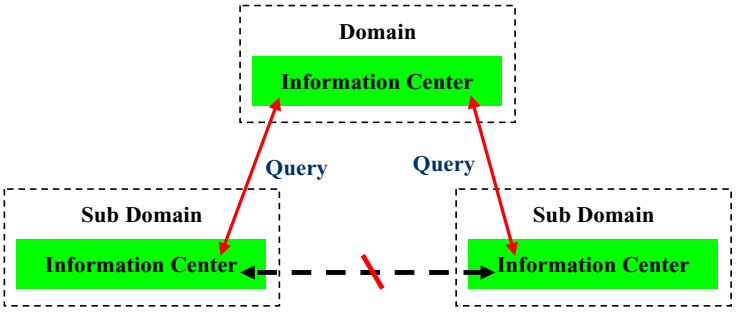

Fig. 3. Relationship between ICs in different domains

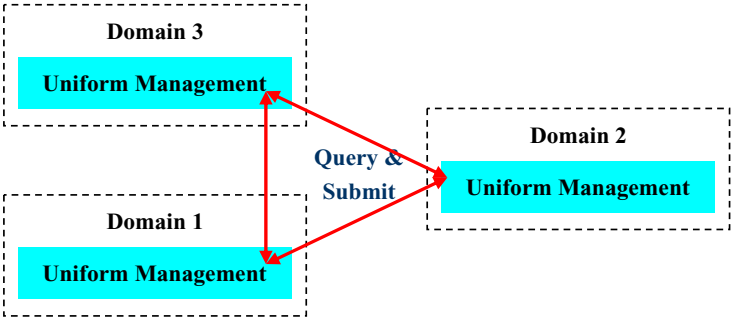

Fig. 4. Relationship between UMs in different domains

Different domains could access each other. But the interactive calls only happen between UMs and ICs of different domains. Fig. 3 shows the IC module relationship of CGSP. In order to see clearly and easily portrayed, Portal, GPE, Gs, and CNs are ignored in the figure. First, we can find that IC in domain has a bidirectional call relationship with the IC in sub domain. But, there is no direct relationship between ICs of different sub domains. That is to say, one IC only has a bidirectional relationship with its parent ICs or its child ICs and has no any relationship with its brother, grandson or grandfather ICs. From each IC, all resource and service information of the whole grid can be gotten through querying the whole tree. In fact, the tree structure of CGSP is held and determined totally by the IC module.

Different from the relationship between ICs, we can find that each UM could call UMs of any other domains directly from the Fig. 4. But which UM will be called by the UM? This is determined by the IC. That is to say, one UM wants to get the support from another UM in other domains, but it does not know which domain could supply such support. So, it must ask help from its own IC. Its IC will search one domain which could provide this requirement through querying its parent IC or child IC. We know that the ICs in a grid are constructed into a tree structure. So, if there is one domain that could provide such support, it can be found through querying the tree at least. But any UM could only call its own IC (in the same domain) and has no interactive relationship with other ICs (in other domains). Fig. 4 shows such relationship.

5 Execution Management Across Domains

Let's see the execution management of job which needs to be submitted from one domain to another one. That is to say that such job can not be completed by its own domain. It will be submitted to another domain to execute.

Fig. 5 shows such a job execution flow step by step. Before submitting job to the job manager, user must upload the input data required by the job to the personal data space in the data manager. Then, job manager will query the available computing nodes from the IC when it gets a computing job from

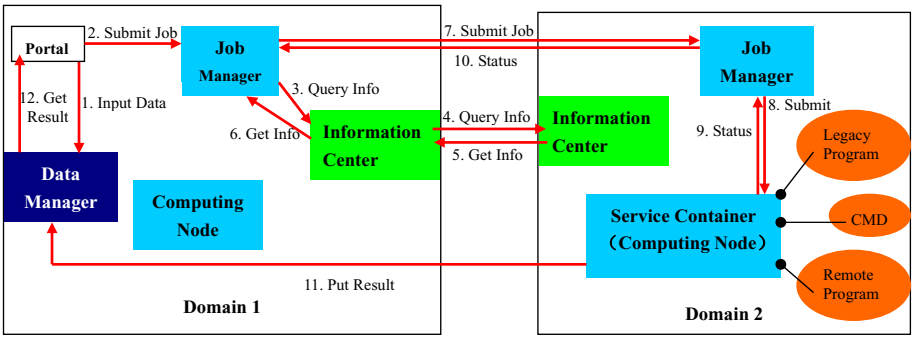

Fig. 5. Job Execution Flow across Domains

Portal. If its own domain can not complete the job, the IC will query other ICs of the grid. And then the job manager will submit the job to the job manager of another domain, which could complete the job. At last, the job will be sent to the service container of one real computing node and begins to execute it.

After the job is started up, the service container will report job status to the original job manager through local job manager in time. When the job is completed, the service container will put the computing results to the personal data area of the data manager of the original domain, which in turn sends the results to user through Portal at last.

There is a problem about user identity mapping during accessing and executing across domains in CGSP. Each domain has its own user management, so user identity should be converted when one job manager submits jobs or report job status to another job manager (such step 7 and 10 in the Fig. 5). The domain manager is in charge of the user identity mapping between different domains in CGSP. But for the interaction of ICs between different domains, it is another story. ICs complete the interaction through system user, who can only access its parent of child domain ICs

6 Conclusion and Future Work

Deferent from the other grid middleware efforts, CGSP supplies a grid platform to satisfy all kinds of requirements of grid constructors and grid application developers. The cost of grid application is reduced greatly based on the CGSP. Besides runtime environment, it also provides developing tools, such as portal constructing tool, programming environment, service packaging tools and so on.

Furthermore, CGSP extensible and reconfigurable framework makes nation-wide grid reality. Especially in China, there are many universities and research institutes. How to guarantee their respective autonomy is the key design principles. At the same time, the integrity of grid is also involved into the CGSP extensible framework.

At present, four ChinaGrid applications, Bioinformatics Grid, Image Grid, Computational Fluid Dynamics Grid and Mass Information Grid have completed the initial building over the CGSP and almost 30 most famous universities in China, as the members of ChinaGrid program, are building their campus grid on the basis of CGSP.

In the future, WS core 4.0 from Globus Toolkit will be replanted into CGSP. Job Submission Description language (JSDL) will be used to replace current Grid Job Description Language (GJDL). At the same time, open source ActiveBPEL [24] Engine will be used to implement the workflow component of job manager in CGSP. OGSA-DAI [23] based integration of heterogeneous database is one of the most important new functions which will be implemented in CGSP version 2. Media service, grid monitoring, accounting and management will be also strengthened in the near future.

Acknowledgement

We wish to express our sincere appreciation to the Prof. Hai Jin, Prof. Weimin Zheng, Prof. Xiaomin Li, Dr. Xiaowu Chen and other ChinaGrid experts for extending their generous support for the successful conduct of the CGSP. Special thanks to CGSP working group members for the CGSP success.

References

1. ChinaGrid, http://www.chinagrid.edu.cn.
2. H. Jin, *ChinaGrid: Making Grid Computing a Reality*, Proceedings of ICADL 2004, Lecture Notes of Computer Science, (2004), 3334, 13-24
3. ChinaGrid Support Platform, http://www.chinagrid.edu.cn/CGSP.
4. CGSP Work Group, *Design Specification of ChinaGrid Support Platform*, Tsinghua University Press, Beijing, China, 2004
5. I. Foster, C. Kesselman, S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organization*, International J. Supercomputer Applications, 15(3), (2001)
6. I. Foster, C. Kesselman, *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, J. Nick, S. Tuecke, (2002)
7. Baraglia, R., Laforenza, D., Lagana, A., *A Web-based Metacomputing Problem-Solving Environment for Complex Applications*, Proceedings of Grid Computing 2000, (2000), 111-122
8. I. Foster, C. Kesselman, *Globus: A Metacomputing Infrastructure Toolkit*, International J. Supercomputer Application, (1997), 11(2), 115-128
9. Open Grid Services Architecture (OGSA), https://forge.gridforum.org/projects/ogsa-wg, or http://www.globus.org/ogsa/
10. Web Service Resource Framework (WSRF), http://www.globus.org/wsrf/ and http://www.ggf.org/documents/GFD.30.pdf
11. Globous Toolkits, http://www.globus.org.
12. UK e-Science Program, http://www.rcuk.ac.uk/escience/;
13. NASA Information Power Grid, http://www.ipg.nasa.gov/;
14. TeraGrid, http://www.teragrid.org/;
15. China National Grid, http://www.cngrid.cn;

16. Open Middleware Infraxtructure Institute, http://www.omii.ac.uk/
17. Dazheng Huang, Fei Xie, Guangwen Yang, *T.G.: a Market-oriented Computing System with Fine-grained Parallelism*, 9th Workshop on Job Scheduling Strategies for Parallel Processing Seattle, Washington, (2002)
18. Tuecke, S., Czajkowski, K., Foster, I. , et.al.: *Open Grid Services Infrastructure (OGSI) Version 1.0*, Global Grid Forum Draft Recommendation. (2003).
19. Yongwei Wu, Guangwen Yang, Qing Wang, Weiming Zheng, *Coarse-grained Distributed Parallel Programming Interface for Grid Computing*, Lecture Notes in Computer Science, (2004), 3032, 255-258; Expanded Version is accepted by International Journal of Grid and Utility Computing;
20. ChinaGrid Bioinformatics Grid, http://166.111.68.168/bioinfo/tools/index.jsp
21. ChinaGrid Computational Fluid Dynamics (CFD) Grid, http://grid.sjtu.edu.cn:7080/grid/
22. ChinaGrid Image Processing Grid, http://grid.hust.edu.cn/ImageGrid/
23. http://www.ogsadai.org.uk/
24. The Open Source BPEL Enging, http://www.activebpel.org/