CROWN: A Service-oriented Grid Middleware System: Experience and Applications

Jinpeng Huai, Chunming Hu, Tianyu Wo and Jianxin Li
Beihang University
{huaijp,hucm,woty,lijx}@buaa.edu.cn

Abstract

Grid computing has emerged as a new paradigm of distributed computing technology on large-scale resource sharing and coordinated problem solving. Based on a proposed Web service-based grid architecture, we have designed a service grid middleware system called CROWN which aims to promote the utilization of valuable resources and cooperation of researchers nationwide and world-wide. To address the issues of CROWN resource management, we proposed some key technologies including trustworthy remote and hot service deployment, overlay-based distributed resource organization, resource scheduling and load balance, and federation-based virtual organization management. A status of the wide-area CROWN testbed is also introduced in this paper. Three typical applications including AREM, MDP and gViz are deployed on the CROWN testbed. Experience of CROWN testbed deployment and application development shows that the middleware can support the typical scenarios such as computing-intensive applications and data-intensive applications etc.

1. Introduction

Grid computing has been an attractive distributed computing paradigm over wide-area network, promising to enable resource sharing and collaborating across multiple domains [1-5].

Nowadays, service-oriented architecture (SOA) becomes an important trend in building a distributed computing environment for wide area network, which helps the merging of Grid and Web services. Recently, Open Grid Service Architecture [2] and Web Service Resource Framework (www.globus.org/wsrf/) were proposed and have become one of the fundamental technologies in Grid competing. SOA and related standardization work provide an important methodology to the research and application of Grid technology. First, the resources are encapsulated into services with standardized interfaces, supporting the unified service management protocol, which helps to solve the problem caused by the heterogeneity of resources; second, the resources are utilized through a service discovery and dynamic binding procedure, which helps to set up a loosely coupled computing environment. But the current resource management mechanism is not enough for all the Grid application scenarios because of the distributed and autonomic resource environment, and the existing security mechanism cannot provide features like privacy protection and dynamic trust relationship establishment, which embarrass the further application of Grid technology.

Actually, not only the Grid computing, but also the Peer-to-Peer computing and the ubiquitous computing try to explore the Internet-oriented distributed computing paradigm. The common issue of these computing paradigms is how to use the capability of resources efficiently in a trustworthy and coordinated way in an open and dynamic network environment. As we know, Internet (especially the wireless mobile network) is growing rapidly, while it is deficient in the effective and secure mechanism to manage resources, especially when the resource environment and relationship between different autonomic systems are changing constantly. At this point, three basic problems such as cooperability, manageability and trustworthiness are proposed. The cooperability problem is how to make the resources in different domains work in a coordinated way to solve one big user's problem. The manageability problem is how to manage heterogamous resources and integrate the resources on demands in a huge network environment, which is a basic condition of building an Internet-oriented distributed computing environment. The trust-worthiness problem is how to set up the reliable trust

978-0-7695-3132-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ISORC.2008.79

relationship between cross domain resources when they are sharing and collaborating.

In the year 2004 the Natural Science Foundation Committee of China (NSFC) announced the Network-based e-Science Environment Program. It's one of the important Grid initiatives in China. The main goal of this program is to build up a virtual science and experiment environment to enable the wide-area research corporation such as large-scale computing and distributed data processing. CROWN[1] is the brief name for China Research and Development environment Over Wide-area Network. Its output may fail into three parts: the middleware set to build a service-oriented Grid system, the testbed to enable the evaluation and verification of Grid related technologies, and the applications.

CROWN service grid middleware is the kernel to build an application service grid. Basic features of CROWN are listed as follows. First, it adopts an OGSA/WSRF compatible architecture[2]; second, considering the application requirements and the limitation of security architecture of OGSA/WSRF, more focus is put on the Grid resource management and dynamic management mechanism in the design stage, and a new security architecture with distributed access control mechanism and trust management is proposed to support the resource sharing and collaborating in a loosely coupled environment.

Under the framework of CROWN project, we also created the CROWN testbed, integrating 41 high performance servers or clusters distributed among 11 institutes in 5 cities (by April 2007). They are logically arranged in 16 domains of 5 regions by using the CROWN middleware. The testing environment is growing continuously and it becomes much similar to the real production environment. The CROWN testbed will eventually evolve into a wide-area Grid environment both for research and production.

CROWN is now becoming one of the important e-Science infrastructures in China. We have developed and deployed a series of applications from different disciplines, which include Advanced Regional Eta-coordinate numerical prediction Model (AREM), Massive Multimedia Data Processing Platform (MDP), gViz[3] for visualizing the temperature field of blood flow, Scientific Data Grid (SDG) and Digital Sky Survey Retrieval (DSSR) for virtual observatory. These applications are used as test cases to verify the technologies in CROWN.

In this paper, we introduce the layered architecture of CROWN, and analyse some key issues faced by resource management of the CROWN testbed and present some novel technologies for resource encapsulation, resource organization, resource scheduling and virtual organization (VO) management.

The rest of this paper is organized as follows. Section 2 presents the design and implementation of CROWN, the service oriented Grid middleware. In section 3, some key issues of resource management in CROWN are introduced. Section 4 presents the successful applications, with CROWN testbed, a wide-area testing environment using CROWN middleware. Section 5 concludes our work.

2. Architecture and Design of CROWN

During past several years, many key issues in grid computing have been under intensive studies. One of them is the architecture for grid. Early computational grids employed the layered architecture with an "hour-glass model" [4]. Recently, with the evolution of Web services, the service-oriented architecture has become a significant trend for grid computing, with OGSA/ WSRF as the de facto standards [6, 7]. CROWN has adopted the service-oriented architecture, connecting large amount of services deployed in universities and institutes.

The architecture of CROWN middleware is shown in Figure 1. This middleware can be classified into three layers. At the resource layer, it can integrate several existing heterogeneous physical resources. A rich set of software components and tools are provided in the middleware layer to support the development and running of grid services in the environment. An application layer is built based on the middleware layer of CROWN providing grid applications for multi-discipline e-Science research.

The grid service container, called NodeServer, is a key component of this middleware, and it is deployed on each of the grid hosts. All the services are encapsulated as certain types of resources, with some of them providing application specific functions and others providing general services, such as grid information services (GIS). Before a computer becomes a Node Server (NS), it must be installed with CROWN middleware. The service container is the core component in CROWN middleware, which provides a runtime environment for various services. Each NS usually belongs to a security domain. Every domain has at least one *RLDS* (*Resource Locating & Description Service*) to provide information services, and *RLDS* maintains the dynamic information of available services. By querying RLDS for information of available grid services, the *Scheduler* can select proper resource to execute the job and solve the problem on behalf of the grid user. CROWN Designer is an IDE for service developer. The CROWN Monitor is to trace the system status in a global view and help to analyze the system running behaviors. A full-fledged grid security solution is also provided in CROWN which contains message level security, authentication and authorization, credential management, identity mapping for heterogeneous security infrastructure, and automated trust negotiation.

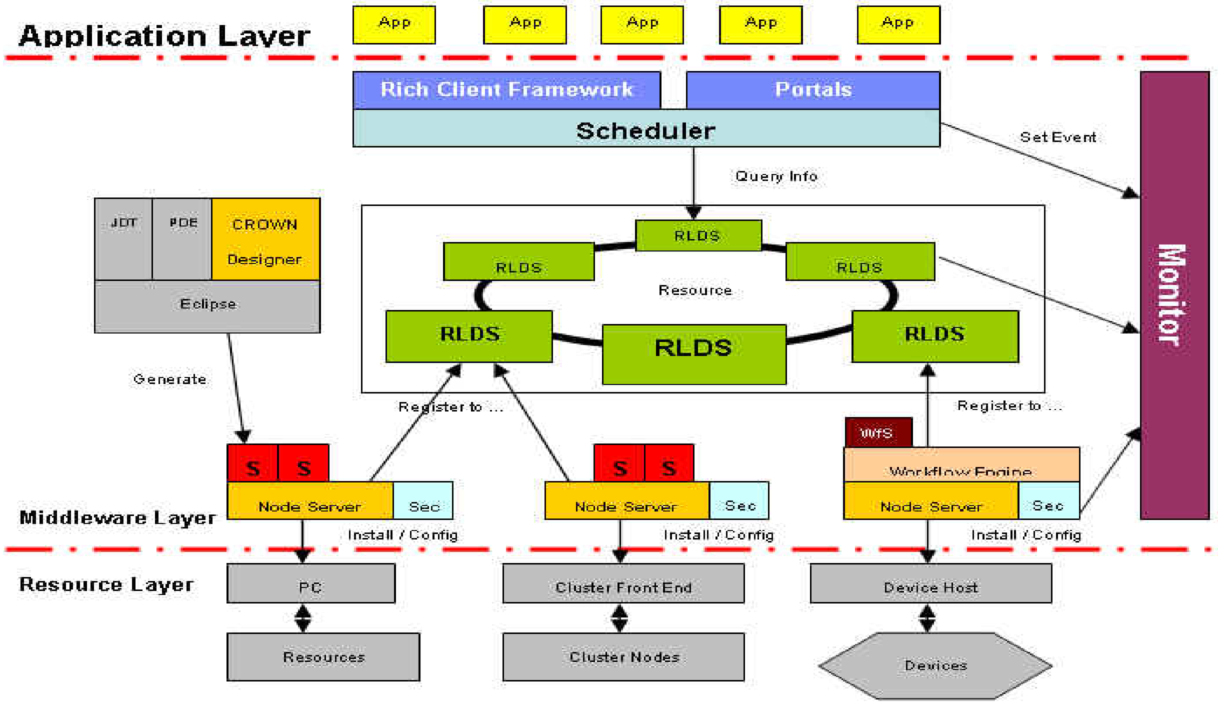

Fig 1: Architecture of CROWN middleware

3. Key Issues of Resource Management

In this section we discuss several key issues of resource management in a service oriented grid environment, and some key technologies involved in CROWN.

3.1. Resource Encapsulation and Deployment

It is an important issue to have heterogeneous resources integrated together to form a unified resource view. In a SOA environment resources such as software functions are often encapsulated as services and allow user access in a standardized way. Services can be moved across the grid nodes to have resource utilized effectively as well as for fault tolerant purpose. However, it is a challenge to have services deployed on the fly securely especially in an open environment.

In CROWN we developed a mechanism of Remote and Hot Deploy with Trust (ROST)[6]. Traditionally, the remote service deployment is supported in a cold fashion, which means the service runtime environment needs to be restarted when deploying a new service. This interrupts running services and may cause a huge overhead both in performance and management. So it is important to have a hot service deployment feature in the runtime platform. To achieve this, an archive format called GAR file (Grid Archive) is proposed to encapsulate all the necessary files and configurations for a grid service. GAR file can be moved to target service container through SOAP/HTTP/FTP protocols. Target service container receives the GAR file and uncompresses it to update the container information without stopping the container. Security issues are guaranteed through the trust negotiation using ATN (autonomic trust negotiation) technique. ATN is a new approach to access control in an open environment, which, in particular, successfully protects sensitive information while negotiating a trust relationship. With ATN, any individual can be fully autonomous. Two individuals belongs to different security domains may setup a trust relationship by exchanging credentials according to respective policies. With the availability of remote and hot service deployment, many applications will benefit, such as load balancing, job migration and so on.

3.2. Resource Organization

To organize individual services to from a resource consuming environment is another important function that the grid middleware should provided. As shown in Figure 2, a hierarchical resource organization is used in CROWN as a fundamental layout.

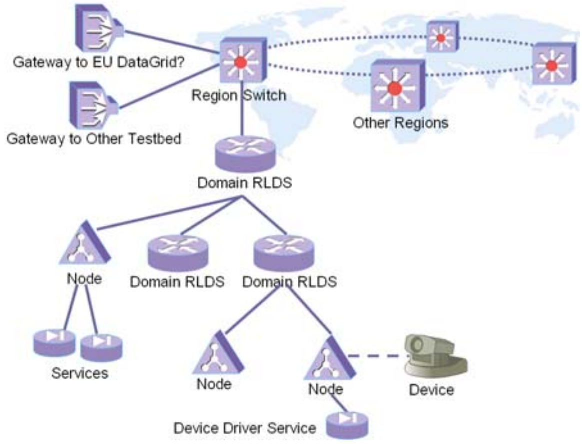

Fig 2: CROWN resource organization

To improve the performance of resource locating and discovery, two overlay based technologies are proposed.

A service club (S-Club) overlay is built over the existing mesh network of GISs, so that same type of services are organized into a service club. An example of such a club overlay is shown in Figure 3, where nodes C, D, E, and G form a club. A search request could be forwarded to the corresponding club first such that search response time and overhead can also be reduced if the desired result is available in the club.

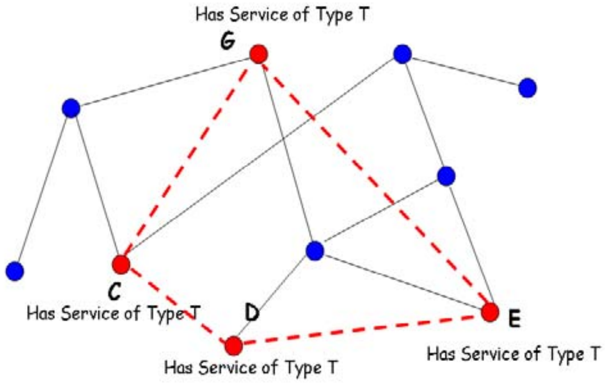

Fig 3: An example of service club

There is some overhead in maintaining the S-Club overlay. So we have to be careful on the trade-off between the potential benefit and the cost incurred. We assume that any search request is firstly sent to a GIS close to the user. On receiving a search request for a specific service type, the GIS checks locally whether there has been a club for this type. If yes, the GIS forwards the request to the club, which will be flooded within the club only. If there is no club for this type, however, the GIS floods the request throughout the mesh network. When a new GIS joins the GIS network, it has no idea what clubs are there. But since it has at least one neighbour in the underlying mesh network, it can ask one of its neighbors for the information of existing clubs. Namely, it simply copies the information of clubs from its neighbour.

Compared with the previous approaches, S-Club adopts a fully decentralized architecture with an unstructured topology. Each GIS keeps the information of services registered to it. To improve the performance, overlay is constructed dynamically and may be changed constantly with the self-organizations of service clubs.

Besides S-Club, there is a RCT (Resource Category Tree) for the third layer's resource management. Computational resources are usually described by a set of attribute-value pairs. Among all attributes of a computational resource, one or several attributes is chosen to characterize the resource capacity of meeting application resource requirements as primary attributes (PA). An overlay called RCT (Resource Category Tree) is used to organize computational resources based on PAs.

Grid applications can be characterized by their requirements for computational resources, e.g. computing intensive and data intensive applications. In turn, categorizing computational resources based on certain resource characteristics that can meet application resource requirements. By doing so, resource discovery is performed on specific resource categories efficiently. For example, resources with huge storage can better serve a data intensive application, thus they can be organized together based on an overlay structure.

Furthermore, according to the observation, the values of most resource attributes are numerical, e.g. values of disk size. And attributes whose values are not numerical can be converted to be numerical through certain mathematical methods. Based on this consideration, RCT adopts an AVL tree (or balanced binary search tree) overlay structure to organize resources with similar characteristics. The attribute that can best describe the characteristic of resources organized by an RCT is named a primary attribute or PA. Figure 4 is an example of RCT. The chosen PA is available memory size, and the value domain of available memory ranges from 0MB to 1000MB.

Compared with traditional AVL, each node of RCT manages a range of values, instead of a single value. Each node only needs to maintain its connection with direct child nodes and parent, and operations like registration, updating and query can start from any node. Unlike in traditional AVL structure, higher-level nodes of RCT are not required to maintain more information or bear more load than those in lower levels, which provide the basis for RCT to scale easily.

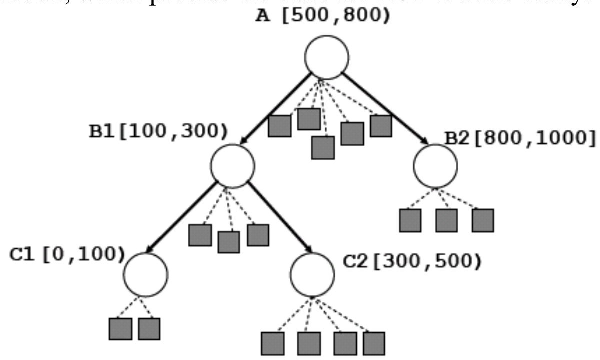

Fig 4: An Example of RCT

3.3. Resource Scheduling

CROWN provides a meta-scheduling service which queues and schedules user's jobs according to a set of predefined strategies. It refers to the RLDS to get current service deployment information and performs matchmaking. Users can also expend these strategies by implementing a policy service provider interface (SPI). The scheduler supports two types of job, POSIX application invocation and grid service invocation. Job submission description language (JSDL) is used to describe the basic information of a job as well as the QoS requirements and security demands. A Basic Execution Service (BES) is implemented as a standard job submission and query interface.

To balance the load among service grid nodes we proposed a Bulletin-Board Based Cooperative Load Balance (BBCLB) strategy by using several bulletin-boards service (BBS) as load intermediates. A modified thresholds based load transfer algorithm has been applied with a non-preemptive selection policy. Grid nodes can exchange queuing jobs via BBS according to their load status. BBS is also used to distribute load information so that idle nodes can pull extra jobs from busy ones directly. The performance evaluations have shown that our strategy can effectively balance the load of service invocation, and improve the system throughput.

3.4. Virtual Organization Management

Grids are becoming a large-scale distributed computing environment, where a potentially unbounded number of users and services without pre-existing trust relationships may be involved. It has been a fundamental but challenging problem to dynamically build mutual trust between service requester and provider coming from different organizations in an open grid environment. In CROWN, new security architecture, as shown in Figure 5 is designed to provide a fine-grained and extensible framework enabling trust federation and trust negotiation for resource sharing and collaboration in an open grid environment.

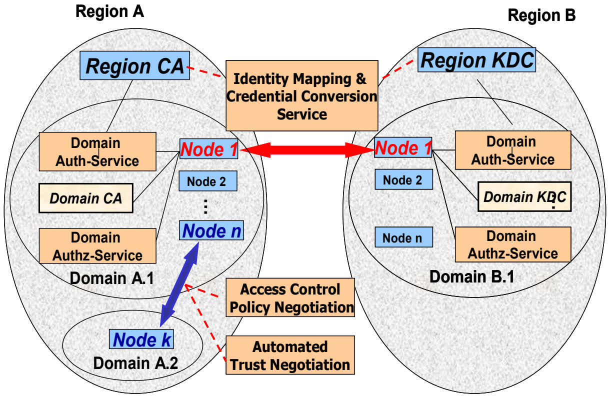

Fig 5: Architecture of CROWN Security

In this architecture, a series of security services such as authentication service, authorization service and trust management service are designed and developed to bridge the trust between heterogeneous security infrastructures such as PKI or Kerberos. The basic functions of this services and components are summarized as follows:

- Secure communication mechanism: We present a policy-based fine-grained secure communication mechanism based on message level security, such as encryption/decryption, signing/verification etc. Moreover, other new security functionalities can easily be extended in this architecture due to its flexibility. It is highly flexible though configuration, which facilitates administrators to specify fine-gained security policies for any service.
- Access control mechanisms: The domain authentication and authorization services are deployed in every domain to enable access control with high configuration, and the user or resource attributes are specified with SAML and the access control policy is specified with XACML to achieve fine-grained resource access control.
- Trust federation mechanisms: In a multi-party collaboration, users in one region may have fundamental problem accessing services provided by other region because they have different authentication method as well as different format for user credential, such as X.509 certificate and Kerberos ticket. An identity mapping and credential

conversion services is an essential enabling mechanism to establish profound collaboration among multi-parties domains.

- Trust management and negotiation: During the dynamic trust establishment between two unknown nodes located in different security domains, the sensitive credentials or access control policies may be disclosed. An automated trust negotiation service is employed to build mutual trust on the fly by disclosing access control policies and credentials iteratively.

4. CROWN Testbed and Applications

To validate the effectiveness and performance of the CROWN middleware, a testbed integrates more than 40 high performance servers or clusters distributed among 11 institutes in 5 cities was established. By installing CROWN on each node, we logically arranged them into 16 domains of 5 regions. The testbed is deployed in a wide-area environment which shows important characteristics of grid system such as dynamic, heterogeneity, distribution and autonomy. By using such a testbed, we can study grid systems behavior in real world and verify the technologies and algorithms of resource sharing and collaboration.

Currently 11 different applications are deployed into CROWN and more than 20000 requests are processed per year. CROWN is now becoming an important e-Science infrastructure in China. In this paper we present 3 typical applications of CROWN: AREM, MDP and gViz.

4.1. AREM

AREM, short for Advanced Regional Eta-coordinate numerical prediction Model, is a tool to study and refine the numerical prediction model of weather and climate. Several numerical models are worked out by meteorologists during their research and production work. Typically these models use the raw weather data from the national meteorology authority as input, and simulate the weather transformation according to the laws of atmospheric physics and fluid dynamics. The output can be used as a prediction result of the future weather. The simulations based on complex numerical calculations need large quantity of computing power and storage capacities. By using the resource organization, job scheduling technologies provided by CROWN, we successfully developed AREM research system. We encapsulated the Fortran complier, visualization tools (GrADS) and the simulation framework of AREM as services, and a unified raw weather data center is also deployed. Meteorologists can submit simulation jobs to the system and refine the numerical models according to the results. Since the jobs are executed by using the resources provided by CROWN testbed, the execution procedure can be parallel, the execution time can be much reduced and the efficiency of weather system research and prediction model refinement is improved.

4.2. Multimedia Data Processing

Large amount of storage capability and computing power is needed when performing multimedia data processing, such as content recognition of voice or video. Traditionally a centralized processing model is applied and pieces of data are collected and processed in a single point. When the input data increase, this method provides little scalability especially for the real-time applications. We combine the service grid technologies with the massive data processing and implement the MDP platform for multimedia data processing. MDP has been deployed into CROWN and provides service since 2005. We encapsulate the related algorithms into services and deploy then on many grid nodes. Users can provide many ways of multimedia data and submit jobs to the grid scheduling system. After analyzing the work load of grid nodes, available resources can be found automatically and data can be processed by invoking corresponding services. Since the platform is deployed in a wide-area environment, we also introduced the trust management and negotiation mechanisms. These technologies protect the user data and make the processing trustworthy.

4.3. gViz

CROWN interoperates with other grid middleware so that large amount resource can be integrated. Our testbed also links to some famous grid systems. gViz is a visualization tool used to show the computation result in grid. It takes several data sources as input and generates picture and HTML pages automatically by using a set of services. We deployed gViz application both in CROWN and White Rose Grid (WRG) which is a part of UK National Grid Service (NGS). A demo is presented in the UK e-Science All Hands Meeting 2005, different gViz services are running on the inter-continental platform simultaneously. They interact to generate the final visualization result. The application gives a good example of resource collaboration of heterogeneous platforms.

5. Conclusion

We draw the conclusion with following lessons learned in CROWN R&D.

- Service oriented architecture is proved to be a good choice in heterogeneous resource integration.
- Layered middleware can reduce developer's work significantly and is helpful in interface standardization and interoperability between different grid platforms.
- Security and trust are very important in grid and should be processed carefully. It's always a synthetic implementation involving different middleware layers.
- Applications can be benefit from the underlying SOA grid infrastructure in that they can reuse existing services and simply arrange the interaction of services. Also they can make use of more resources to improve their performance.

We are also looking at other important features of grid middleware. Among them a wide-area data management service is under developing. Also we are trying to improve the quality and interoperability of CROWN. And to fulfill the requirement from real application and network environment, several fault tolerant models are implementing in CROWN.

Acknowledgements

Part of this work is supported by grants from the China Natural Science Foundation (No.90412011, 60703056), China 863 High-tech Program (Project No. 2006AA01A106), China 973 Fundamental R&D Program (No. 2005CB321803) and National Natural Science Funds for Distinguished Young Scholar (No. 60525209). We would also like to thank Hailong Sun, Li Lin, Liang Zhong, Qin Li and other members in CROWN team for their contributions. We also thank the team in University of Leeds and all our collaborators for their suggestions and help in developing CROWN.

References

[1] CROWN Project, http://www.crown.org.cn

[2] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The physiology of the Grid: An Open Grid Services Architecture for distributed systems integration", http://www.globus.org/research/papers/ogsa.pdf

[3] Ken Brodlie, David Duce, Julian Gallop, Musbah Sagar, Jeremy Walton, Jason Wood. "Visualization in Grid Computing Environments". Proceedings of IEEE Visualization 2004, pp155-162.2004

[4] Hailong Sun, Wantao Liu, Tianyu Wo, Chunming Hu," CROWN Node Server: An Enhanced Grid Service Container Based on GT4 WSRF Core", Fifth International Conference on Grid and Cooperative Computing Workshops pp. 510-517, 2006

[5] Jinpeng Huai, Tianyu Wo, and Yunhao Liu, "Resource Management and Organization in CROWN Grid," in Proceedings of the 1st international conference on Scalable information systems, 2006.

[6] Jinpeng Huai, Hailong Sun, Chunming Hu, Yanmin Zhu, Yunhao Liu, Jianxin Li," ROST: Remote and hot service deployment with trustworthiness in CROWN Grid", Future Generation Computer Systems archive Volume 23 , Issue 6 (July 2007)

[7] J. Frey and T. Tannenbaum, "Condor-G: A computation Management Agent for multi-Institutional Grids," Journal of Cluster Computing, vol. 5, pp. 237, 2002.

[8] W. Hong, M. Lim, E. Kim, J. Lee, and H. Park, "GAIS: Grid Advanced Information Service based on P2P Mechanism," in proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC-13), 2004, pp. 276-277.

[9] A. Iamnitchi, I. Foster, and D. C. Nurmi, "A Peer-to-Peer Approach to Resource Location in Grid Environments," in proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11), 2002.

[10] H. Sun, Y. Zhu, C. Hu, J. Huai, Y. Liu, and J. Li, "Early Experience of Remote and Hot Service Deployment with Trustworthiness in CROWN Grid," presented at Advanced Parallel Processing Technologies, 6th InternationalWorkshop,APPT 2005, 2005.

[11] J. Basney, M. Humphrey, and V. Welch, "The MyProxy Online Credential Repository," Software: Practice and Experience, vol. 35, pp. 801-816, 2005.

[12] S. Bajaj, D. Box, and D. Chappell, "Web Services Policy Framework," 2005.

[13] T. M. Simon Godik, "OASIS eXtensible Access Control Markup Language (XACML)," 2003.

[14] R. Housley, W. Ford, T. Polk, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," 1999.

[15] S. Anderson, J. Bohren, and T. Boubez, "Web Services Secure Conversation Language," 2005.

[16] J. Linn, "Generic Security Service Application Program Interface, Version 2," 1997.

[17] S. Anderson, J. Bohren, and T. Boubez, "Web Services Trust Language," 2005.

[18] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)," 2005.