G2G: A Meta-Grid Framework for the Convergence of P2P and Grids

Wu-Chun Chung, National Tsing Hua University, Taiwan

Chin-Jung Hsu, National Tsing Hua University, Taiwan

Yi-Hsiang Lin, National Tsing Hua University, Taiwan

Kuan-Chou Lai, National Taichung University, Taiwan

Yeh-Ching Chung, National Tsing Hua University, Taiwan

ABSTRACT

Grid systems integrate distributed resources to form self-organization and self-management autonomies. With the widespread development of grid systems around the world, grid collaboration for large-scale computing has become a prevalent research topic. In this paper, the authors propose a meta-grid framework, named the Grid-to-Grid (G2G) framework, to harmonize autonomic grids in realizing a grid federation. The G2G framework is a decentralized management framework that is built on top of existing autonomic grid systems. This paper further adopts a super-peer network in a separate layer to coordinate distributed grid systems. A super-peer overlay network is constructed for communication among super-peers, thus enabling collaboration among grid systems. This study proposes the G2G framework for use in a Grid-to-Grid federation and implements a preliminary system as a demonstration. Experimental results show that the proposed meta-grid framework can improve system performance with little overhead.

Keywords: *Convergence, Grid Computing, Grid-to-Grid (G2G), Peer-to-Peer (P2P), Super-Peer*

INTRODUCTION

A grid computing system is a distributed computing system for solving complex or high-performance computing problems as encountered in bioinformatics, healthcare systems, ecosystems, and even experiments involving the use of the Large Hadron Collider. In such a computing environment, a virtual organization is a self-organization and self-management group which shares the computing resources (Foster, Kesselman, & Tuecke, 2001). Grid systems employ the middleware as the abstract interface to integrate large-scale distributed computing resources. Therefore, the aggregated capability for distributed computing and data accessing can be improved by integrating geographical distributed resources.

Various organizations, institutions, and private communities around the world adopt

DOI: 10.4018/jghpc.2010070101

centralized or hierarchical architectures to develop grid systems based on the Open Grid Service Architecture (OGSA) (Foster, Kesselman, Nick, & Tuecke, 2002). Most grid systems, though, cannot be integrated for collaborative computation. This is why grid collaboration for large-scale computing has become a prevalent research topic. The integration of distinctly autonomic grid systems into a grid federation is one prospective approach. What poses a challenge to realizing grid federations, however, is how to harmonize various grid systems without bringing a heavy burden on existing grid infrastructure.

In order to coordinate multiple diverse grid systems, a grid system requires a mechanism for achieving the cross-grid convergence of diversely autonomic grid communities. There are two ways to accomplish cross-grid integration. One is to enhance the grid middleware by modifying the original mechanism in existing grid systems, while the other is to develop a meta-grid framework on top of existent grid systems. The former burdens an existent grid system with lots of efforts to harmonize with other grid systems; moreover, there is no mature cross-grid middleware for integrating with distinct grid systems. On the other hand, the latter increases the extra overhead for the existent grid systems. In this paper, we present a meta-grid framework, named the Grid-to-Grid (G2G) framework, to form a federation consisting of multiple institutional grid systems. The G2G framework can harmonize autonomic grid systems and achieve cross-grid collaborative computing with seamless modification for existing grid systems.

Past studies have applied decentralized approaches to exploit the system scalability of grids in the development of the grid management architecture. In general, a centralized or hierarchical architecture is not suitable for large-scale grids (Mastroianni, Talia, & Verta, 2007). Reasons for this include the potential bottleneck at root, the scalability limit of a grid, and load imbalance problems. Integrating grid systems with the P2P paradigm can improve the scalability of a grid federation. Therefore, we attempt to exploit a decentralized G2G framework to realize synergy between P2P networks and existing grid systems. In consideration of the scalability and the efficiency of the grid management architecture, this study utilizes a super-peer network (Yang & Garcia-Molina, 2003) to develop our G2G framework for the coordination of multiple autonomic grid systems. Each super-peer represents an autonomic grid system in our G2G system. To achieve a decentralized G2G framework, the super-peer network adopts an overlay network for communication among super-peers in different grid systems through a federation of wide-area grids.

This paper introduces a meta-grid framework of the G2G system based on a super-peer network. We also present a preliminary implementation of the proposed G2G framework and develop a Grid-to-Grid network based on the overlay network in which each grid system communicates and negotiates with other grid systems. The remainder of this paper is organized as follows. Section 2 discusses related works. In Section 3, we present the overview of the G2G framework and the implementation of a G2G prototype. The experimental results of the G2G system are shown in Section 4. We conclude this paper with the future work in Section 5.

RELATED WORKS

Scalability in large-scale grid systems has posed research challenges in recent years. There are some studies (Ranjan, Harwood, & Buyya, 2008; Trunfio et al., 2007) that discuss the adoption of the P2P technique to improve the scalability of grid systems. Some similarities and differences between P2P computing and grid computing were presented in the literature (Foster & Iamnitchi, 2003; Talia & Trunfio, 2003). Several previous studies aimed to improve the centralized-based grid infrastructure by using the P2P technique.

A decentralized event-based object middleware, DERMI (Pairot, Garcia, & Skarmeta, 2004), is proposed to favor the scalability problem of a centralized model. A DHT-based DERMI prototype uses Pastry (Rowstron & Druschel, 2001) as its underlying network topology and adopts Scribe (Castro, Druschel, Kermarrec, & Rowstron, 2002) as its publish/subscribe notification service which is built on top of Pastry. However, from a view point of the middleware layer, DERMI can facilitate the wide-area grid. From another view point, our study aims to coexist with the existence of grid systems without modifying the middleware.

The integration of a distributed event brokering system with the JXTA technology has been proposed in the literature (Fox, Pallickara, & Rao, 2005) to enable Peer-to-Peer Grids. In a previous study (Pallickara & Fox, 2003), the authors utilize the NaradaBrokering based on the hierarchical structure in the broker network. By integrating the NaradaBrokering with JXTA, services are mediated by the broker middleware or by the P2P interactions between machines on the edge of Internet. NaradaBrokering aims to present a unified environment for grid computing with a P2P interaction. In addition, the overhead will be costly for NaradaBrokering to maintain the broker network by the hierarchical topology in a dynamically changed network.

A P2PGrid platform based on a two-layer model for integrating P2P computing into the grid is presented in a previous study (Cao, Liu, & Xu, 2007). All grid services are provided in the grid layer while the grid entities or common PCs become peers to negotiate with each other in the P2P layer. The P2PGrid tries to provide a solution for integrating the grid computing environment with the P2P computing ones. In this study, JXTA is adopted for developing JXTA Agents to create peers, deal with dynamics of peer groups, and communicate with peers. Jobs are submitted and dispatched to workers which are organized by the above-mentioned peers on the underlying P2P computing network. By the implementation of the P2PGrid platform, peers on the edge of Internet are able to consume grid services without maintaining grid middleware packages. Adopting a separate layer from an existing grid system is a great benefit because that the original behaves of the grid layer can be preserved without modifications, and that the modification of the P2P manner will not affect the efficiency of the grid layer.

In our study, we present a decentralized meta-grid framework on top of existing autonomic grid systems. The autonomic grids are coordinated based on the super-peer network to form a Grid-to-Grid collaborative computing environment. A super-peer is able to provide/consume the grid services to/from other super-peers in remote grid systems. Super-peers are organized based on an unstructured super-peer overlay to negotiate with each other. By adopting a separate layer, the G2G framework can integrate with existing grid systems without modifying the original mechanisms and policies. In addition to the support of computation services and data services, we also propose a solution for the verification of accessing remote resources by considering the security issues in the Grid-to-Grid environment.

G2G FRAMEWORK AND PROTOTYPE

Currently, most grid systems are deployed according to centralized or hierarchical management approaches. However, these approaches have poor performance in terms of scalability, resiliency, and load-balancing for managing distributed resources (Mastroianni et al., 2007). Centralization and hierarchy are the weaknesses of deploying large multi-institutional grid systems, let alone in the widely internetworking system. In general, the performance by adopting the super-peer model is more efficient and convenient than that without adopting the super-peer model in large-scale computing environments.

In our G2G framework, we utilize the super-peer network to coordinate existing grid systems. The P2P technique is applied for the communication between super-peers. In this section, we describe the concept of the G2G system before introducing the meta-grid framework. At the end of this section, we present a preliminary prototype of the G2G system.

Super-Peer Based G2G System

The super-peer network is proposed to combine the efficiency of the centralized search as well as the features in terms of autonomy, load balance, and robustness of the distributed search. A super-peer is a node that acts both as a centralized server to a set of ordinary nodes and as a coequality to negotiate with other super-peers. In order to achieve the seamless integration of the grids in the G2G system, this study adopts the super-peer network on top of the existing grid systems, and harmonizes existing autonomous grids with each other without rebuilding/modifying any grid system.

Each super-peer in our G2G system acts as a coordinator which is responsible for coordinating a local autonomic grid system and negotiating with other super-peers in remote grid systems. For example, after obtaining a request for task execution, the super-peer first checks whether the request can be processed locally; otherwise, the request will be forwarded to other grid systems by cooperating with other super-peers. For the sake of simplicity, we currently consider only one super-peer deployed on top of the local grid systems. Similar examples have been proposed in past studies (Mastroianni, Talia, & Verta, 2005). Since there are multiple autonomic grid systems in the G2G system, we utilize an unstructured overlay to facilitate the federation of super-peers. In this way, the Grid-to-Grid interactions among distinct grid systems are based on the super-peer network by way of the P2P overlay. Based on a P2P overlay, each grid can easily join the G2G system and supply its resources and services to other grid systems. The resource utilization can be improved after applying the G2G system.

The basic concept and architecture of the G2G system are shown in Figure 1 and Figure 2. The G2G system mainly consists of the Cross-Grid layer and the Local-Grid layer. The Local-Grid layer consists of some autonomic grid systems which are built by the grid middleware to collaborate distributed resources. In the Cross-Grid layer, the G2G layer is responsible for coordinating the super-peers in autonomic grids. These super-peers not only take charge of integrating the autonomic grids by the developed common interfaces but also deal with the negotiation between grid systems in the G2G layer. The Meta-Grid interface is responsible to bridge the Cross-Grid layer and the Local-Grid layer. Using these common interfaces, the Cross-Grid layer can acquire the resources and services from the Local-Grid layer without knowing the policies, mechanisms, or algorithms in the Local-Grid layer. Since the Cross-Grid layer and the Local-Grid layer are independent, the Cross-Grid layer does not need to be modified when the mechanisms in the Local-Grid layer are modified or replaced.

G2G Framework

In this study, a meta-grid framework of the G2G system is proposed for a federation of multiple autonomic grid systems as shown in Figure 3. The G2G framework aims to support the seamless integration of the computing services and the data accessing services in the autonomous grid systems. Therefore, the super-peer in the Grid-to-Grid layer consists of seven components to bridge the Cross-Grid layer and the Local-Grid layer: *the interactive interface*, *the security management*, *the network management*, *the task management*, *the data management*, *the resource management* and *the information service*.

The *task management* component handles job computation, and the *data management* component is responsible for integrating the distributed storage systems. The *network management* component handles the network topology and the G2G interaction between distinct grids. The *resource management* manages the distributed resources in grid systems according to the resource status supported by the *information service* component. The *interactive interface* component deals with the login process for users, and the *security management* component is in charge of the authorization of using grids. When each component would like to negotiate with local grid system, the Meta-Grid Interface exchanges the information. By cooperating

Figure 1. Basic concept of the meta-grid framework

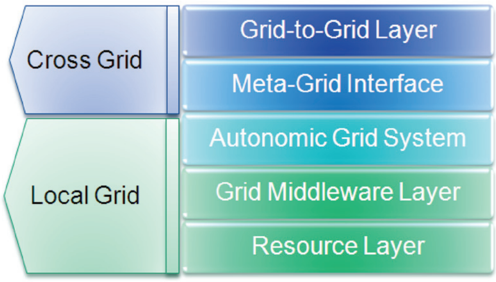
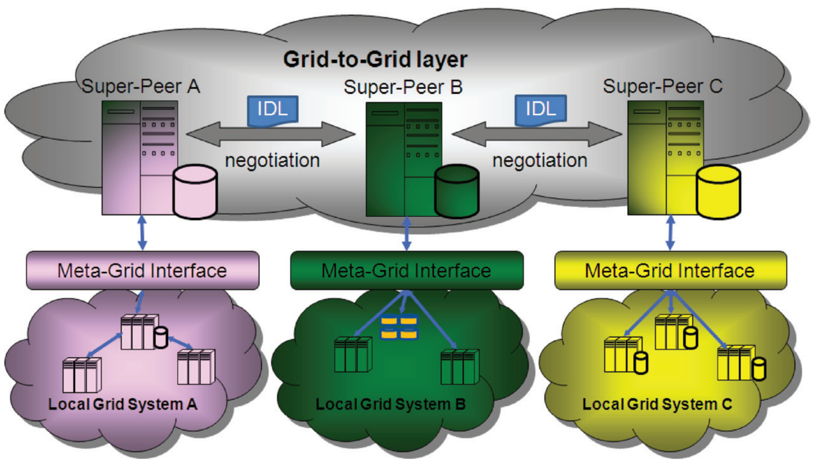


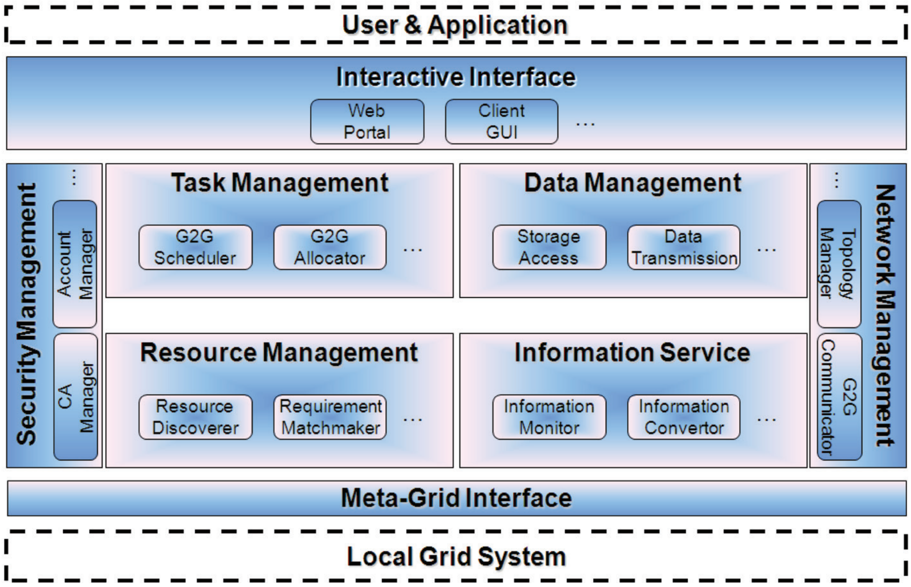
Figure 2. Architectural overview of the G2G system based on a super-peer network



these components in the G2G framework, we can apply grid applications on this framework.

We use a simple example to describe the cooperating procedure. When a user wants to login into the G2G system, the web portal will call the SecurityMgmt.loginManager() to handle the login process. This function will deliver the user's login information to the local security service through the Meta-Grid Interface. If the login process is successful, the user can edit jobs on the web portal or upload the data related to the job computation. When a user wants to submit a job for execution, the TaskMgmt.g2gScheduler() will receive the request from the web portal and call the ResourceMgmt.resourceDiscoverer() to locate the desired resources. The discovery process will firstly call the localResourceDiscovery() to check whether local resources are available by using the InformationService.informationMonitor(). If there are available resources in the local grid system, the TaskMgmt.g2gAllocator() will submit the job for local execution through the Meta-Grid Interface. In other cases, if no resources in the local grid system are available, the discovery process will call the remoteResourceDiscovery() to exploit remotely available resources by using the NetworkMgmt.g2gCommunicator(). If the remote login process is successful by using the SecurityMgmt.caManager() and the SecurityMgmt.accountManager(), the discovery process of the remote grid system is similar as above. The detail notions of developing a G2G prototype are shown in next subsection.

Figure 3. Framework overview of components in the Grid-to-Grid layer



G2G Prototype

This study uses JAVA to develop the proposed G2G framework. The developed components of the super-peer are deployed on top of each autonomic grid system to form the Grid-to-Grid federation environment. The super-peers communicate with each other by using a P2P overlay. In this subsection, we describe the implementation and the cooperation of all components in the G2G computing system.

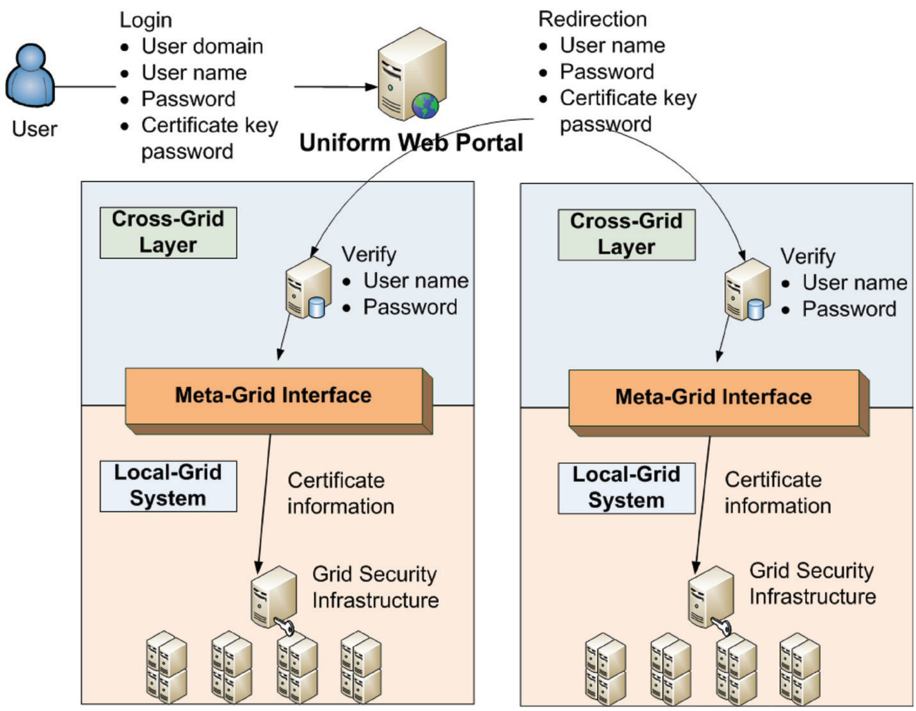
Portal and single sign-on. In general, a friendly interactive interface is important for users while using the grids. Therefore, this study develops a uniform web portal for users to easily enter a grid system and to utilize the authorized resources and services. There are two important functions for developing a uniform web portal: Single Sign-On (SSO), and workflow operation.

Single Sign-On (SSO) is adopted for users to access the grids with only-once login. Each user can utilize grid resources/services after the successful verification through the proxy server and the security management. This study proposes a uniform web portal on top of each autonomous grid system. A redirection mechanism is also developed in the uniform web portal as shown in Figure 4.

When a user logins the G2G system from this uniform web portal, the portal will determine which grid system the user should be entered according to the user's login information. The candidate grid system will verify whether the user's login information is valid or not. If the login is successful, the portal will deliver the user's login information to the local security service through the Meta-Grid interface. If the certificate of the user is also valid, the login process is successful and complete. Otherwise, it will be a failure. Since the login process is accomplished through the integration of original grid systems, if the local grid supports the SSO mechanism, the G2G layer can also sustain the SSO mechanism.

On the other hand, the workflow operation in the G2G system supports the task submission. The workflow structure in the G2G system is similar to the M-Task structure (Rauber & Runger, 2005). Each workflow is composed of

Figure 4. Redirection mechanism for users to login on the uniform web portal



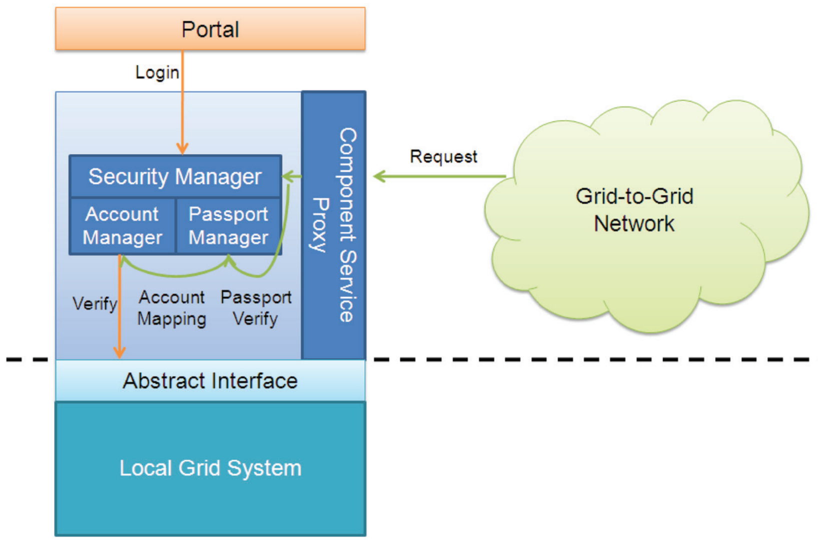
multiple stages and each stage is composed of multiple jobs. Jobs between distinct stages may be dependent; however, jobs in the same stage are all independent, that is, all jobs within the same stage can be scheduled and allocated for simultaneous execution.

Our study also develops a workflow editor in the uniform web portal. According to the resource status obtained from the information service component, users can edit the tasks on the portal and also specify desired resource requirements. We adopt an XML-based structure language to describe the task information and support the resource discovery by multiple attributes with the range query. After the task submission, the edited workflow can be transformed into the XML-based form and can be stored in the database for users to lookup, cancel, or refine their tasks at anytime.

Security service. Grid authentication and authorization are key services in grid security management. A previous study (Foster, Kesselman, Tsudik, & Tuecke, 1998) has defined the Grid Security Infrastructure (GSI) for the legal utilization of grid services. In the G2G system, the security management component deals with not only the certificate authorization locally, but also the admission request from remote grid systems. The secrecy and privacy mechanisms in the G2G system have to guarantee the original legal services in the local grid systems and accept the permission of utilizing local resources/services for other remote grid systems. The security management component includes two primary functions as shown in Figure 5: the passport manager for the authorized privilege and the account manager for the legal account management.

The passport manager takes care of the passport registration and the verification in the

Figure 5. Certificate authentication and authorization for our G2G system



G2G system. A passport represents the admission or verification of the request from remote grid systems. If one grid system tries to access resources in another grid system, it must get a visaed passport from the target grid system before accessing the resources. This study develops a distributed passport-interchange-mechanism in the G2G system. According to the maintenance of neighborhood relationship, each grid system can request a remote resource/service from its neighbors or neighbors' neighbors by forwarding the resource/service request along the overlay network. After discovering the available resource/service in remote grids, the requester will receive the visaed passports from the granted grid system; and then, the requester can submit tasks to the granted remote grids with legal permission.

On the other hand, the account manager is responsible for the management of legal accounts. In this study, the function of the account authentication is adopted for a "local account" to login the grids. A local account indicates an original user account in the local grid system. Once an account requests for a login from the portal, the portal asks the account manager to verify its identification. Another important aspect of the account manager is the account mapping mechanism. Account mapping is used to handle requests issued by foreign users from remote grid systems. Every grid system which wants to use the resources in other grids must register to the granted grid system before accessing those resources. The register process acquires a passport and gets a temporary account. Once the register process is completed, every request with the visaed passport from remote grid systems will be treated as a local user account through the account mapping mechanism.

Data service. Data management is responsible for data maintenance and high performance transmission in the data grid (Chervenak, Foster, Kesselman, Salisbury, & Tuecke, 2000). In this study, the data service of our G2G system supports specific APIs for the transparent accesses of existent data storage in each local grid system and for the data transmission among different autonomic grid systems. The abstract APIs is responsible to contact with a storage system in a local grid system or a general file system. Data accesses between the Cross-Grid layer and the Local-Grid layer adopt the well-defined data operations; otherwise, the data accesses from one grid to another grid adopt the G2G communication mechanism through super-peers.

When a communication between grids is necessary for data transmission, the super-peer takes care of the negotiation and communication with other super-peers in the G2G layer. We use the account manager to manage the foreign data files in this case. When the data files are accessed from remote grid systems, these data files can be stored in the local storage system and then be mapped to local owners. After data mapping and account handling, the foreign data file can be accessed by local users.

In this study, we focus on collaborating with the computing grid system and the data grid system, and omit developing efficient policy for data replication. Those issues about data coherence and parallel downloading between distinct grid systems remain as a future work.

Information service. The main responsibilities of the information service include the resource indexing and monitoring for capturing the resource status in a grid system. Traditional Grid Information Service (Czajkowski, Fitzgerald, Foster, & Kesselman, 2001; Fitzgerald et al., 1997) generally adopts the centralized or hierarchical architecture. Such architectures for the information service are hard to directly apply to the G2G system because of the single point of failure problem. To alleviate the failure problem, this study develops an information service for crossing the inter-grid systems on top of the existent information monitoring system.

Our information service consists of two mechanisms: the *information monitor* and the *information convertor*. Each grid system in the Local-Grid layer is responsible for monitoring the local information. The super-peer queries the local information by using the proposed information monitor to negotiate with the local grid system. On the other hand, we propose an Information Description Language (IDL) to negotiate the information with the local-grid system and exchange the information between cross-grid systems. The IDL is a XML-based structure, which is shown as Figure 6 and Figure 7, for describing the grid information such as the workflow submission, the job requirement, the resource information, and so on. We also develop the information convertor to transform diverse XML-based information into our IDL format, and vice versa. By using the information convertor, each grid system of our G2G system can extract the information from the local-grid or remote-grid description.

By the IDL, information can be negotiated from Cross-Grid layer to Local-Grid layer and messages can be exchanged between distinct grid systems in the G2G layer. A workflow or a complicate job requirement with multi-attribute range query can be supported in the distributed resource discovery.

Network management. This study proposes a Grid-to-Grid overlay based on the super-peer network. In the G2G system, the super-peer in each autonomic grid system takes responsible for the negotiation and communication with other super-peers over the G2G network. The decentralized overlay network is adopted to construct the neighborhood relationship and to forward a request between super-peers.

There are two main management functions in the G2G network management: the *topology manager* and the *G2G communicator*. The topology manager maintains the overlay network for the neighborhood relationship or routing information between super-peers. As one super-peer tries to join the G2G network, the topology manger guides the new super-peer to join this G2G system, and then the super-peer will forward a request to build its routing table after successfully joining into the G2G network.

Figure 8 depicts the procedure for a new grid system to join the G2G system. Every

Figure 6. Example for the task submission and the job requirement in the IDL structure

```
<TASK>
    <WORKFLOW NAME="Workflow Name">
        <STAGE NAME="Stage Name" ORDER="1">
            <JOB NAME="Job Name">
                <PROGRAM NAME="Matrix Multiplication" ARG="1024" NP="4" TYPE="MPI" />
                <INPUT_FILES>
                    <INPUT_FILE DATA_ID="Data ID for the Program" />
                </INPUT_FILES>
                <REQUIREMENTS>
                <!-- One of the following requirement tag should be matched for executing the job. -->
                    <REQUIREMENT>
                    <!-- All the required information should be matched with the following conditions. -->
                        <INFORMATION_TYPE NAME="Information Name" VALUE="Information Value" VALUETYPE="Value Type"
                                          UNIT="Value Unit" OPERATOR="AND, NOT" RELATION="EQUAL, GREAT, LESS" />
                    </REQUIREMENT>
                    <REQUIREMENT>
                        <!--
                          -- For example, the job requires the desired resource with CPU is greater than 2GHz,
                          -- and the machine type is not x86_64.
                        -->
                        <INFORMATION_TYPE NAME="2" OPERATOR="1" RELATION="2" UNIT="1" VALUE="2000" VALUETYPE="6" />
                        <INFORMATION_TYPE NAME="21" OPERATOR="2" RELATION="1" UNIT="0" VALUE="x86_64" VALUETYPE="1" />
                    </REQUIREMENT>
```

Figure 7. Example for resource information in the IDL structure

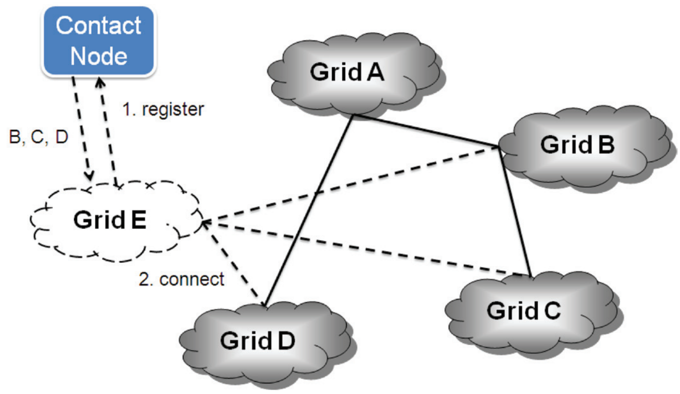
```
<RESOURCE>
    <NODE ADDRESS="Super-peer Address">
        <CLUSTER ADDRESS="Cluster Address">
            <HOST ADDRESS="Host Address 1" REPORTED="Last-record TIMESTAMP">
                <RESOURCE_STATE NAME="Attribute NAME" VALUE="Attribute VALUE" TYPE="Value TYPE" UNIT="Value UNIT" />
            </HOST>
            <HOST ADDRESS="Host Address 2" REPORTED="Last-record TIMESTAMP">
                <!-- For example, a resource with the CPU is 3200 MHz and the machine type is x86. -->
                <RESOURCE_STATE NAME="2" VALUE="3200" TYPE="6" UNIT="1" />
                <RESOURCE_STATE NAME="21" VALUE="x86" TYPE="1" UNIT="0" />
            </HOST>
```

autonomic grid system acts as a super-peer to be a member of the super-peer network. When a grid system wants to join the G2G system, it registers to a contact node and gets a list of grid systems selected by the contact node in random. After obtaining the random list, the new grid system measures the network latency or the bandwidth with all the candidate members. After the measurement process, the new grid system selects members with better performance to register as its neighbors. This join procedure for a new member is similar to the previous work (Mastroianni et al., 2005). We implement this mechanism for a new super-peer to join the G2G network, and enhance the construction of an unstructured overlay network.

We also present the G2G communicator to take care of the network communication and the message negotiation. In order to communicate with different autonomic systems, we not only apply the IDL to describe the exchanged information but also design an application-level request format for message transmission. Every communication is accomplished by using the socket connection. The communication in-between two grid systems can be divided into sender- and receiver-modules. For the sender module, all the requests will be transformed into a predefined request format, and then the requests are sent to remote super-peers in serial. For the receiver module, the remote super-peer de-serializes all the received requests and forwards to the corresponding components.

Task management with resource discovery. The task management is in charge of the task submission through the interactive interface. A task consists of jobs executed in sequential or in parallel. In the G2G system, tasks are not only submitted from local users, but probably are requested from remote grid systems. Depending on available grid resources, the G2G scheduler and G2G allocator need to consider the job

Figure 8. Procedure for a new grid system to join the G2G network



execution among the intra-grid submission or the inter-grid submission.

In the task management module, we adopt a workflow structure to organize jobs in a predefined order for execution. The workflow structure is constructed by stages. The jobs in one stage must wait for execution until all jobs in previous stage are completed because of the stage-by-stage approach to avoid destroying the job dependence in different stages. We also develop the workflow manager and the job manager to handle requested tasks. After a task is submitted to the workflow manager, the manager schedules the order of jobs and decides where to execute these jobs. The decision of migrating the executable jobs to a local grid or a remote grid depends on the system performance or the resource requirement.

On the other hand, each job has its resource requirement for execution. This study also applies a resource discovery mechanism (Mastroianni et al., 2005) to explore the distributed resources status over the Grid-to-Grid overlay network, and supports a matchmaking policy to provide candidate resources satisfied the specified requirements. After a task is submitted to the waiting queue for execution, the G2G scheduler picks one of queuing jobs according to the First-Come-First-Served (FCFS) policy, and then checks whether local resources are sufficient or under loading at first. The decision of where to execute a job depends on not only whether local grid system is over loading, but also whether local resources are satisfied with requirements through information service. If the local grid system is not busy and there are sufficient available resources, the job will be submitted to the local grid system to be executed. Otherwise, the job manager will ask the distributed resource discovery module to search available resources over the overlay network. If there are sufficient resources in other remote grid systems through the super-peer network, the job will be migrated to the remote grid system for execution. Otherwise, this job will be queued in the waiting queue for available resources.

EXPERIMENTAL EVALUATION

In this section, a simple experiment is conducted to evaluate the system performance of the proposed framework. At first, we describe the experimental environment in our evaluation. Then, we show the experimental results in terms of the average turnaround time and the extra overhead occurring from a separate layer by adopting the G2G system.

Experimental Environment

In the experimental environment, two autonomic grid systems based on the framework of

Taiwan UniGrid (Shih et al., 2008) are applied. In order to limit the impact of the heterogeneous environment, we utilize the same cluster in each autonomic grid system for execution. One autonomic grid system contains a cluster with 8 higher computational power CPUs (i.e., the grid with higher computational power.) The other autonomic grid system contains a cluster with 32 lower computing power CPUs (i.e., the grid with rich resources). In each local grid system, the SRB (Baru, Moore, Rajasekar, & Wan, 1998) is adopted to be the local data storage system; and the ganglia information monitoring system (Massie, Chun, & Culler, 2004) is utilized to monitor the local information. We deploy the proposed super-peer network on top of each autonomic grid system to form a Grid-to-Grid federation environment.

We use a matrix multiplication program as the benchmark. Each job in the experiment is a parallel program written by MPI with C. The matrix size is 1024x1024. The number of required processors for each job is set to be 4. The ratio of communication to computation of the test program is about 1 to 100. Each job is submitted per 10 seconds; and the total number of submitted jobs varies from 5 to 20 in two cases to evaluate the system performance. In case 1, the same jobs are submitted to each autonomic grid system for four rounds. The number of jobs in each round is 5, 10, 15, or 20, respectively. For this case, all the jobs are only executed in the local grid system. In case 2, the same jobs of each round are submitted to each autonomic grid system with the G2G federation environment. For this case, each job will be executed in the local grid or the remote grid according to the decision made by the above-mentioned G2G scheduler and G2G allocator.

We estimate the average turnaround time of each job for completing the computation and also measure the average overhead of each job with regard to the time consumed from the Cross-Grid layer to the Local-Grid layer. The job turnaround time is defined as the time period from the time when a job is submitted to the waiting queue for processing in the Cross-Grid layer, to the time when the submitted job is completed in the Local-Grid layer. The overhead of migrating a job is defined as the time period from the time when a job is submitted to the Cross-Grid layer, to the time when the submitted job is decided to which local-grid system for execution. By definition, the job turnaround time includes the execution time and the migrating overhead. If the cost of overhead takes too much time in the Cross-Grid layer, the overall turnaround time for executing a job will be increasing. The G2G system is expected to reduce the average turnaround time of each job with little overhead when there is over-loading or there are no available resources in the local grid system.

Experimental Results

Figure 9 depicts the experimental results in terms of the average turnaround time. The results show that each grid system requires longer turnaround time to complete a job when there are more jobs submitted for execution without the help from remote grid systems. As a result, the grid with higher computational power takes the longer time to complete a job once there is getting more jobs for execution. This is because a job is queued for a long time if local resources are often busy in the grid system with fewer resources. The longer time a job is queued, the more time the turnaround time will be consumed. However, the system performance can be improved by using the proposed meta-grid framework. When there are no available resources in a local grid system, the job will be migrated to a remote grid system with available resources for execution. On the other hand, Figure 10 and Figure 11 depict the ratio of average overhead compared with the average turnaround time. The experimental results show that the average overhead in all cases is light. The reason is that the average overhead will be reduced if a job can be allocated to available resources for execution as soon as possible. Hence, the average turnaround time of completing a job will be reduced as well.

In summary, a grid system with G2G federation can not only reduce the cost of overhead

Figure 9. Average turnaround time for completing jobs

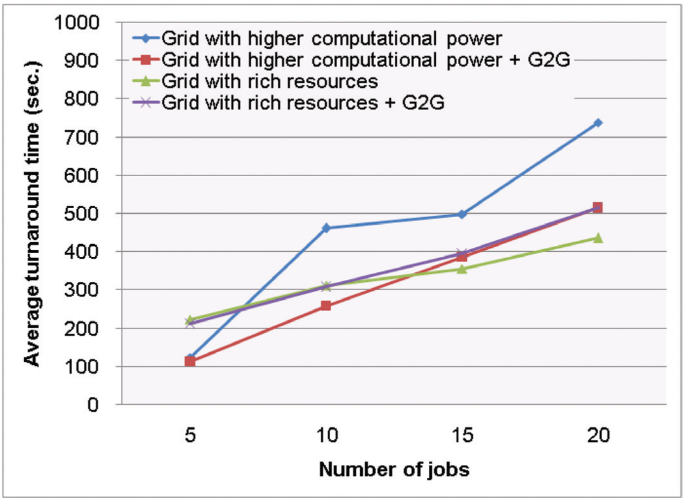
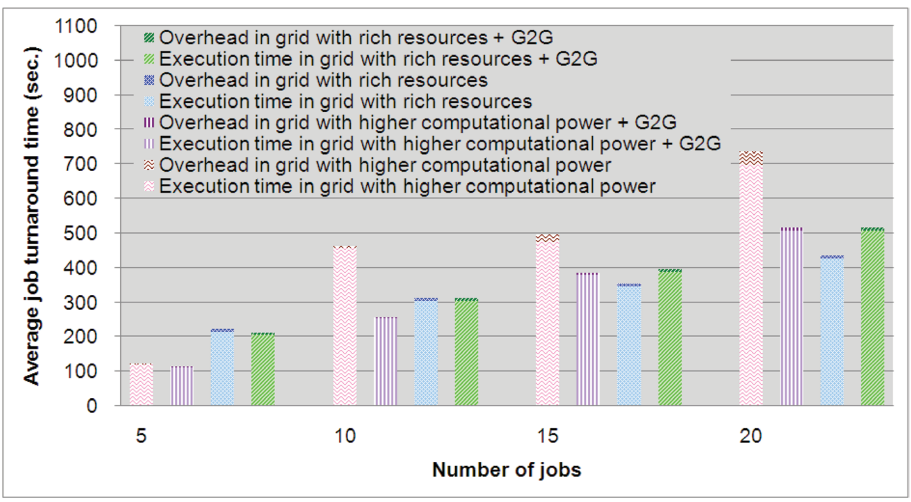


Figure 10. Distribution of average turnaround time for completing jobs

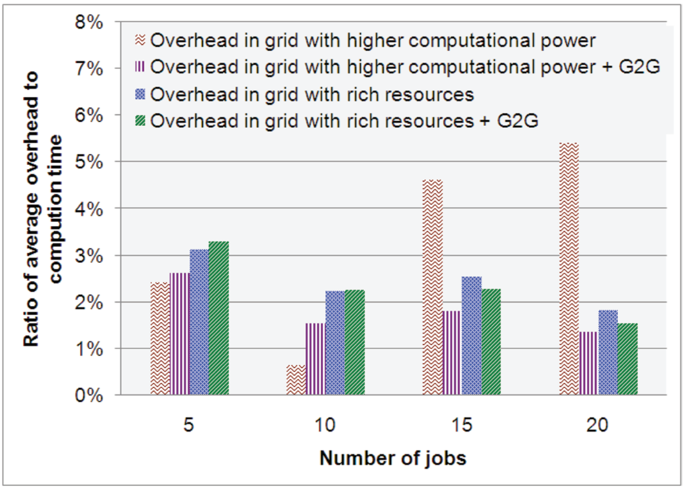


when there are not available resources for execution, but also reduce the average turnaround time with little overhead. The reason is that some jobs submitting to the grid with higher computational power can be migrated to the grid with rich resources for execution. Hence, the grid with high computational power can achieve lower average turnaround time and the grid with rich resources can achieve higher resource utilization. From the experimental results, the average turnaround time can be reduced up to about 45% and the average overhead in all cases is less than 6%.

CONCLUSION AND FUTURE WORKS

Integrating the grid computing with P2P technique can improve the scalability of the large-

Figure 11. Ratio of average overhead to computation time for completing jobs



scale grid system. This study proposes a meta-grid framework, named the G2G framework, for the Grid-to-Grid federation of autonomic grid systems without modifying the original mechanisms and policies. Based on the super-peer network, we adopt a separate layer built on top of existing grid systems to develop the Grid-to-Grid collaborative computing environment. Each super-peer represents a grid system. A super-peer in the G2G system is responsible for coordinating an internally autonomic grid system and for communicating with other super-peers. The overlay network among super-peers is constructed via the unstructured approach.

A grid system is deployed with the capacity of the super-peer for coordinating the G2G system. With well-defined APIs, the G2G system resembles an abstract layer that is separate from the existing grid systems. In our G2G system, an existing grid system can be upgraded its G2G capability without upsetting original mechanisms. We not only take care of the support of computation services and data services, but also consider a solution for the grid security across different grid systems. To evaluate the performance of the G2G system, we implemented a preliminary system to show that the proposed system not only is workable but also improves the system performance.

In the future work, we will refine the IDL format to follow the standard description language, and also study on the efficiency of grid security across diverse grid systems. We also intend to integrate with more autonomic grid systems and enhance the G2G framework to the Service-Oriented Architecture (SOA) in order to develop a service-oriented G2G computing system.

REFERENCES

Baru, C., Moore, R., Rajasekar, A., & Wan, M. (1998). *The SDSC Storage Resource Broker*. Paper presented at the Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research.

Cao, J., Liu, F. B., & Xu, C.-Z. (2007). P2PGrid: Integrating P2P Networks into the Grid Environment. *Concurrency and Computation*, *19*(7), 1023–1046. doi:10.1002/cpe.1096

Castro, M., Druschel, P., Kermarrec, A.-M., & Rowstron, A. I. T. (2002). Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure. *IEEE Journal on Selected Areas in Communications*, *20*(8), 1489–1499. doi:10.1109/JSAC.2002.803069

Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., & Tuecke, S. (2000). The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications*, *23*(3), 187–200. doi:10.1006/jnca.2000.0110

Czajkowski, K., Fitzgerald, S., Foster, I., & Kesselman, C. (2001). *Grid Information Services for Distributed Resource Sharing*. Paper presented at the Proceedings of the10th IEEE International Symposium on High Performance Distributed Computing, New York.

Fitzgerald, S., Foster, I., Kesselman, C., von Laszewski, G., Smith, W., & Tuecke, S. (1997). *A Directory Service for Conturing High-Performance Distributed Computations*. Paper presented at the Proceedings of The Sixth IEEE International Symposium on High Performance Distributed Computing.

Foster, I., & Iamnitchi, A. (2003). On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In *Peer-to-Peer Systems II* (pp. 118-128).

Foster, I., Kesselman, C., Nick, J. M., & Tuecke, S. (2002). *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Retrieved from http://www.globus.org/alliance/publications/papers/ogsa.pdf

Foster, I., Kesselman, C., Tsudik, G., & Tuecke, S. (1998). *A Security Architecture for Computational Grids*. Paper presented at the Proceedings of the 5th ACM conference on Computer and communications security.

Foster, I., Kesselman, C., & Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, *15*(3), 200–222. doi:10.1177/109434200101500302

Fox, G., Pallickara, S., & Rao, X. (2005). Towards Enabling Peer-to-Peer Grids. *Concurrency and Computation*, *17*(7-8), 1109–1131. doi:10.1002/cpe.863

Massie, M. L., Chun, B. N., & Culler, D. E. (2004). The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. *Parallel Computing*, *30*(7), 817–840. doi:10.1016/j.parco.2004.04.001

Mastroianni, C., Talia, D., & Verta, O. (2005). A Super-Peer Model for Resource Discovery Services in Large-Scale Grids. *Future Generation Computer Systems*, *21*(8), 1235–1248. doi:10.1016/j.future.2005.06.001

Mastroianni, C., Talia, D., & Verta, O. (2007). *Evaluating Resource Discovery Protocols for Hierarchical and Super-Peer Grid Information Systems*. Paper presented at the Proceedings of the 15th Euromicro International Conference on Parallel, Distributed and Network-Based Processing.

Pairot, C., Garcia, P., & Skarmeta, A. F. G. (2004). *DERMI: A Decentralized Peer-to-Peer Event-Based Object Middleware*. Paper presented at the Proceedings of the 24th International Conference on Distributed Computing Systems.

Pallickara, S., & Fox, G. (2003). NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids. In . *Proceedings of Middleware*, *2003*, (pp.998–999).

Ranjan, R., Harwood, A., & Buyya, R. (2008). Peer-to-Peer-Based Resource Discovery in Global Grids: A Tutorial. *IEEE Communications Surveys & Tutorials*, *10*(2), 6–33. doi:10.1109/COMST.2008.4564477

Rauber, T., & Runger, G. (2005). *M-Task-Programming for Heterogeneous Systems and Grid Environments*. Paper presented at the 19th IEEE International Parallel and Distributed Processing Symposium.

Rowstron, A., & Druschel, P. (2001). Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Proceedings of Middleware*, *2001*, (pp. 329–350).

Shih, P.-C., Chen, H.-M., Chung, Y.-C., Wang, C.-M., Chang, R.-S., Hsu, C.-H., et al. (2008). *Middleware of Taiwan UniGrid*. Paper presented at the 2008 ACM symposium on Applied computing.

Talia, D., & Trunfio, P. (2003). Toward a Synergy Between P2P and Grids. *IEEE Internet Computing*, *7*(4), 96–95. doi:10.1109/MIC.2003.1215667

Trunfio, P., Talia, D., Papadakis, H., Fragopoulou, P., Mordacchini, M., & Pennanen, M. (2007). Peer-to-Peer Resource Discovery in Grids: Models and Systems. *Future Generation Computer Systems*, *23*(7), 864–878. doi:10.1016/j.future.2006.12.003

Yang, B., & Garcia-Molina, H. (2003). *Designing a Super-Peer Network*. Paper presented at the 19th International Conference on Data Engineering.

Wu-Chun Chung received his B.S. degree in computer and information science from Aletheia University in 2004 and his M.S. degree in computer science and information engineering from National Dong Hwa University in 2006. He is currently pursuing his Ph.D. degree in the department of computer science at National Tsing Hua University. His research interests include grid computing, distributed systems, peer-to-peer computing, and wireless networks.

Chin-Jung Hsu received his B.S. degree in computer science and information engineering from Fu-Jen Catholic University in 2007. He is currently pursuing his M.S. degree in the department of institute of information systems and applications at National Tsing Hua University. His research interests include grid computing, distributed systems, peer-to-peer computing, and cloud computing.

Yi-Hsiang Lin received his B.S. degree in information management from National University of Kaohsiung in 2007. He is currently pursuing his M.S. degree in the department of institute of information systems and applications at National Tsing Hua University. His research interests include grid computing, distributed systems, and peer-to-peer computing.

Kuan-Chou Lai received his MS degree in computer science and information engineering from the National Cheng Kung University in 1991, and the PhD degree in computer science and information engineering from the National Chiao Tung University in 1996. Currently, he is an associate professor in the Department of Computer and Information Science at the National Taichung University. His research interests include parallel processing, heterogeneous computing, system architecture, P2P computing, grid computing, and cloud computing. He is a member of the IEEE and the IEEE Computer Society.

Yeh-Ching Chung received a B.S. degree in Information Engineering from Chung Yuan Christian University in 1983, and the M.S. and Ph.D. degrees in Computer and Information Science from Syracuse University in 1988 and 1992, respectively. He joined the Department of Information Engineering at Feng Chia University as an associate professor in 1992 and became a full professor in 1999. From 1998 to 2001, he was the chairman of the department. In 2002, he joined the Department of Computer Science at National Tsing Hua University as a full professor. His research interests include parallel and distributed processing, cluster systems, grid computing, multi-core tool chain design, and multi-core embedded systems. He is a member of the IEEE computer society and ACM.