ThaiGrid: Architecture and Overview

Vara Varavithya
Department of Electrical Engineering
King Mongkut's Institute of Technology North Bangkok
E-mail: vara@hpc.ee.kmitnb.ac.th
Putchong Uthayopas
Department of Computer Engineering
Kasetsart University
E-mail: pu@ku.ac.th

ABSTRACT -- During the past decade, tremendous amounts of computing resources has been invested. The computing resources include not only hardware and software but also data communication infrastructure. Human society, as a whole, has better understanding and accustom to the benefit of information technologies. These factors drive the needs in higher degree of resource sharing. The Grid enable technologies will allow efficient computing resource sharing and collaboration among different entities. The computing resource sharing is needed to be improved significantly in terms of application domains, user interactions, security, and resource management. Deploying grid enable technologies is not a trivial task. In this paper, we present the hardware-software architecture deployed in the ThaiGrid. The ThaiGrid is the collaborative project intended to create a high performance computing testbed in Thailand. The Kasetsart University and the King Mongkut's Institute of Technology are the two universities involve in the initial implementation of the ThaiGrid. The ThaiGrid consists of three high performance computing platforms. An overview of the Grid middle-ware implemented on the ThaiGrid is presented.

Keywords -- ThaiGrid, Grid computing environment, High performance computing, Distributed environment, Wide area computing, Metacomputing.

1. Introduction

The proliferation of computer networks leads to high degree of computing resource sharing. During the past decade, tremendous amounts of computing resources has been invested. The computing resources include not only hardware and software but also data communication infrastructure. Human society, as a whole, has better understanding and accustom to the benefit of information technologies. These factors drive the needs in higher degree of resource sharing. These computing resources include computational power (computing cycles), storage space, I/O devices, etc. A large scale high performance computing system is made possible using mechanisms that provide efficient resource sharing. Analogous to the electrical power grid, a *computing grid* [1] is deployed by implementing a set of mechanisms that incorporates several geographically distributed computing facilities.

Most of the areas in computational sciences can benefit from the grid. High throughput computing, applications in computation on fluid dynamics, genetic programming, finite element applications, simulations, and large databases are examples of tentative applications that can deploy in the grid environment. The core services in grid environment include resource allocation, security, directory services, and monitoring [1].

The resource allocation is responsible for managing the sharing of computational, storage, and I/O resources which is the main component of the Grid infrastructure. The security mechanisms include user authentication to access resources, negotiate security policy among different site, and information secrecy. The directory services provide information regarding availability of resources on the grid, the location of resources, and access rules for each resources, etc.

In a large scale resource management, monitoring system is crucial in grid operation. The status of tasks, system utilization, and workload, and level of services are examples of monitoring activities. The grid enable technology will extend the use of computers into many new areas. Newly emerging applications include data center, virtual reality, collaborative research, large scale information sharing, virtual organization, resource broker. Globus toolkit. Examples of the early efforts include Globus toolkits [?], NetSolve [2], etc.

Examples of grid testbeds include I-WAY and Gusto [1, 3, 4] which has shown feasibility of constructing the large scale computing platforms. This paper presents a architecture on deploying Grid infrastructure in Thailand, so called *ThaiGrid*. There are two important aspects in the developing grid infrastructure. First, a national/regional organization is need to be established to encourage the use of grid middle-ware in the

high performance computing centers. Second, ThaiGrid needs an communication infrastructure that compose of very high speed links interconnected between the computing sites.

High performance computing research at the Kasetsart University and the King Mongkut's Institute of Technology North Bangkok has been conducted for several years. With the collaboration between two research groups, the ThaiGrid testbed is implemented and tested. The Globus toolkits were installed on three Beowulf clusters. Section 2 describes an overview of the Grid. The preliminary studies on ThaiGrid deployment is outlined in Section 3. Section 4 gives concluding remarks.

2. Grid Preliminaries

The Grid technology provide the global scale resource sharing [5]. The main function of the Grid technologies include resource allocation, access control, authentication and authorization. There are quite a few existing softwares/frameworks that implement subset of grid services. Examples include DQS [6], LSF [7], Condor [8], PVM [9], and MPI [10]. Most of the present scientific and engineering applications are designed for serial applications. An application that belongs to this class consumes lots of computation time. The user can submit their batch job to the pool.

DQS and LSF allow a user to submit batch jobs to a set of workstations in which queuing and load balancing primitives are provided. In the Condor system, serial batch jobs are submitted to idle workstations in the organizational network therefore utilizing unused resources (so called *cycle steeling*). In order to minimize annoyance to interactive jobs, a batch process is handle dynamically among different workstations using process check pointing and process migration. The majority of tasks that suitable for the software system like Condor are classified as high throughput computing applications.

In distributed parallel computations, tasks are executed by a set of intercommunicating computer systems. These systems exploit parallelism embedded in the applications and execute multiple operations in parallel. For coarse gain computation, the communication is usually carried out by means of passing messages from one node to another over the interconnection network. Several parallel programming languages have been proposed to aid the development of parallel applications. The main components of parallel programming languages include process management, process synchronization, and a set of communication primitive.

Example of the popular parallel programming language are PVM [9] and MPI [10]. MPI allow users to develop portable message passing applications. The process management task involves the subtasks spawning the processes across a set of processors and mechanisms to dynamically create or terminate processes. Synchronization mechanisms are required to coordinate the processes.

The communication primitives can be classified into two categories, unicast communication, and collective communication. In unicast communication, a message is send from a source node to a destination (one to one). Collective communication which involves a group of intercommunicating nodes is useful in several applications. Some parallel programming languages provide efficient support for these applications.

Most of high performance computing resources has been controlled by a single administrative domain. This assumption cannot be applied to the grid environment. The computing resources in the grid are geographically distributed, managed by different organizations, and governed by different policies. Therefore the resource sharing techniques that currently available is not adequate. Since computing resources are not located at the same physical site, the communication speed is limited by the propagation delay. The bandwidth of the communication trunks is much slower compared to the speed of system area network and network congestion is more likely to happen. The grid applications cannot expect the same level of communication services compared to intra-cluster communication speed or the speed of high performance system area networks.

Globus toolkits are a collection of grid enable components. The main functions of the Globus toolkits include Resource Location and Allocation [11], Communications [12], Directory Services [13], and Security subsystem [14].

Resource location and allocation are implemented with rule-based selection where suitable computing resources are allocated to specific applications in both computing power and communication performance. Nexus, communications in the Globus toolkits, defines basic interface to the communication subsystem. The higher level primitives can be constructed using Nexus services. For example, different communication methods are allowed in the same task running on different platforms.

Globus's directory services, called Metacomputing Directory Services (MDS), are based on *LDAP*. Updated information about the system configuration, CPU and network load information, and application specific information can be fetched using the MDS. One important aspect of grid environment is the access control and authentication. Globus has single sign-on capability based on PKI and SSL. The initial implementation of ThaiGrid is based on Globus toolkits. The readers are encouraged to refer to [1] for the details information on the Grid technologies.

3. The ThaiGrid Architecture

In the early stage of ThaiGrid implementation, we focus on the following application areas, High throughput computing applications, High performance parallel applications, and Information storage. Other services such as collaborative computing and teleinstrumentation will be included in the next phase of ThaiGrid implementation.

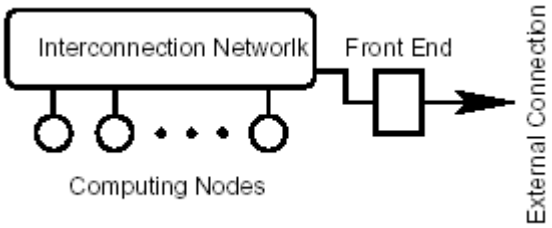

Figure 1. Basic Computing Facility.

3.1 Pool Resources

ThaiGrid will provide computing resource needed and return the results to the user in the batch mode. We target to have at least 1024 computers participated in the ThaiGrid to serve this class of applications. The grid have to aware of the heterogeneity of the computers such that a computer with Interconnection Networlk Front End Computing Nodes External Connection Figure 1. Basic Computing Facility. right architecture is selected or an application is portable to multiple platforms.

Parallel computers and cluster of workstations are expensive resources. Several organizations in Thailand operate these state of the art computing platforms. An expensive system will be better utilized using Grid infrastructure. The "real" high performance applications were written using parallel programming languages. The Grid middleware will allocate necessary parallel resources on the grid to the parallel application or co-allocate several parallel resources on different sites to a single application [15]. This system will allow Thai researchers to better access to the high performance parallel systems. Amounts of information is increased in a very rapid rate which drives the demands on high availability storage. The Grid technology can serve the storage sharing among organizations in which data integrity and confidentiality are maintained. The data is transparently replicated among the participated site to enhance storage's availability.

We define a basic computing facility as in Figure 1. The computing facility can provide both computing cycles, data storage, and special I/O equipments. The frontend server responsible for the grid management which includes resource allocations, access control, etc. The grid enable middlewares are installed on these machines. The collections of the basic computing facilities are shown in Figure 2. A group of basic computing facilities are connected via wide area networks. The resource allocation and admission control are managed by resource brokers. Information on the available resource of the Grid can be accessed from the brokers.

3.2 Basic Requirements

Basic requirements in ThaiGrid includes

- Batch serial jobs submission to the ThaiGrid: Users can submitted the batch serial jobs to the ThaiGrid and the processing nodes will be allocated to specific jobs.
- Parallel jobs submission to the ThaiGrid: Users can specify the architecture and resource requirements for the parallel tasks and submit the parallel application to the grid. A single parallel platform or multiple parallel platforms can be allocated/coallocated to the task.
- Single sign on capability: Users of the ThaiGrid testbed are required to authenticate themselves only the first log-on to the Grid. Only single sign on much be adequate to access resources in multiple site without reducing security level.

3.3 The Testbed

ThaiGrid testbed consists of three eight nodes clusters and an SMP server. Two clusters, AMATA and SMILE, are resided in KU[1]. The PALM[2] cluster and SMP server are located at KMITNB. Figure 2 shows the cluster architectures participated in the ThaiGrid. PALM and SMILE are Intel-Based Clusters. The processing nodes in these clusters range from Pentium Pros to highend Pentium III. Fast Ethernet Switches operated at 100 Mbps full duplex are used as interconnection fabrics. The AMATA cluster has deployed fast AMD processors interconnected with the Myrinet network. The Myrinet network is a high bandwidth low latency system area network operated at 1.25 Gbps. Each cluster has a frontend server to provide the access control. The Globus toolkits are installed in these frontend servers to provide Grid services.

3.4 Communication Infrastructure

The performance of the ThaiGrid depends heavily on communications. The three clusters are connected to the campus network using Fast Ethernet Technology. The campus networks at both KU and KMITNB are based on 155/622 Mbps ATM technology. Both sites are linked with relatively low speed communication via ThaiSarn2[3] and the UniNet[4]. The links between the computing facilities to ThaiSarn2 are funded by NTL of NECTEC. The detail topology is presented in Figure 3. The communication latency of the intracluster networks and the campus networks is much lower than the inter-site latency. We hope that a high speed infrastructure that offers dedicated fiber optic links between participated universities in ThaiGrid in which Gbps bandwidth can be expected from the network.

[1]Details of both clusters can be found in ttp://smile.cpe.ku.ac.th.
[2]Details of the PALM cluster can be found in http://hpc.ee.kmitnb.ac.th.

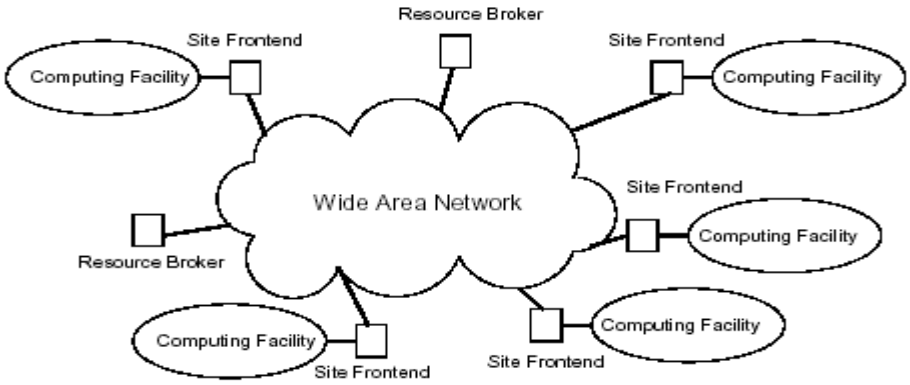

Figure 2. Basic Architecture in the ThaiGrid.

3.5 Middleware

The application domain under Grid concept is very large. The major components of the Grid technology are middlewares. The layers of middlewares help enabling the Grid requirements. Figure 4 shows the middleware architecture deployed in the ThaiGrid. As we target the scientific applications, a set of scientific subroutines and parallel libraries are included. At the beginning implementation of ThaiGrid, the Globus toolkits is selected as a Gridware.

SSL and LDAP software are also required for Globus to operate correctly. The Globus version 1.14 is installed to all sites in the ThaiGrid. The security mechanisms used in the Globus toolkits are based on public key technology. Two types of certificate were obtained from Globus[5], gate-keeper certificates and user certificates. The gate-keeper certificates were stored at the frontend servers. The users gain accesses to the Grid by registering their certificated to the frontend servers belong to multiple sites. With this security mechanism, only one sign on is required to access all the site in ThaiGrid for period of 12 hours. We are working on establishing an organization to issue the certificates for Thai Grid Community.

MPICH with G2 device enable is selected as the test application library. Globus and related software suite installation guides can be browsed from http://smile.cpe.ku.ac.th/thaigrid. The firewall and router configuration modification requests were sent to open some specific ports for Globus communication.

It can be concluded from the preliminary experiments in [16] that, in order to implement the ThaiGrid, a major investment in network infrastructure among universities is required. We have successfully use the Globus toolkits and test the single sign on capability. The MPI application can now run on the ThaiGrid Testbed.

[5]http://www.globus.org

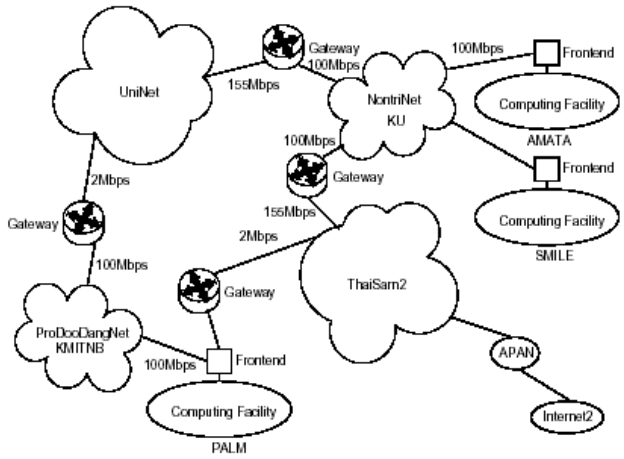

Figure 3. The ThaiGrid testbed topology. All clusters are connected to the campus networks via 100 Mbps Ethernet. Both sites have links that connect to the UniNet other and Internet Service Providers.

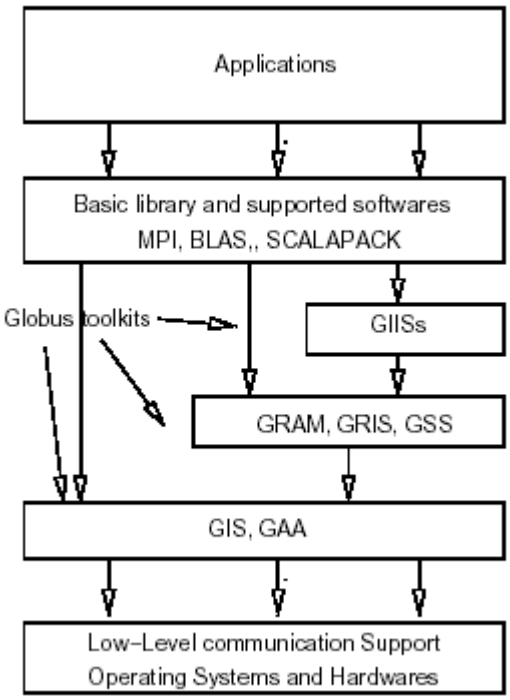

Figure 4. ThaiGrid Middleware Architecture.

4. Conclusions and Future Works

The Grid enable technologies will allow efficient computing resource sharing and collaboration among different entities. The computing resource sharing is needed to be improved significantly in terms of application domains, user interactions, security, and resource management. In this paper, we present the hardware-software architecture deployed in the ThaiGrid. The basic computing facility is defined and can be used as a basis in implementing Grid node. The ThaiGrid testbed is implemented on three clusters at Kasetsart University and King Mongkut's Institute of Technology North Bangkok. We have enumerated a minimum set of middlewares required to achieve basic requirements of the ThaiGrid. The Globus toolkits together with scientific subroutines and parallel libraries were installed and tested on the ThaiGrid testbed.

The most important aspect of the Grid is collaborations among the users. We would like to encourage the participation from universities and organizations. The central organization to issue certificate for ThaiGrid is needed to be established for Thailand. The future works on ThaiGrid includes implementing full set of services for high performance computing, testing the teleconference applications and developing monitoring system for grid environment.

Acknowledgments

Special Thanks to Pitsanu Lousangfa, Supakit Prueksaaroon, and Somsak Sriprayoonsakul who have played major roles in implementing the ThaiGrid testbed.

References

[1] I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, 1999.

[2] H. Casanova and J. Dongarra, "NetSolve: A network server for solving computational science problems," *International Jouranl of Supercomputer Applications and High Performance Computing*, vol. 11, no. 3, pp. 212–223, 1997.

[3] I. Foster, J. Geisler, W. Nickless, W. Smith, and S. Tuecke, "Software infrastructure for the I-WAY metacomputing experiment," 1998. to appear.

[4] I. Foster, J. Geisler, and S. Tuecke, "MPI on the I-WAY: A wide-area, multimethod implementation of the Message Passing Interface," pp. 10–17, IEEE Computer Society Press, 1996.

[5] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal in Supercomputing Applications*, 2001 (to appear).

[6] "DQS 3.1.3 user guide." Supercomputer Computations Research Institute, Florida State University, Tallahassee, March 1996.

[7] S. Zhou, "Load sharing in large-scale heterogeneous distributed systems," in *Proceedings of the Workshop on Cluster Computing*, 1992.

[8] M. Litzkow, M. Livney, and M. Mutka, "Condor– a hunter for ifle workstations," in *Proceeding of the 8th International Conference on Distributed Computing Systems*, pp. 104–111, 1988.

[9] J. Dongarra, G. Geist, R. Manchek, and V. Sunderam, "Integrated PVM framework supports heterogeneous network computing," *Computers in Physics*, April 1993.

[10] J. J. Dongarra, S. W. Otto, M. Snir, and D. Walker, "A message passing standard for MPP and workstations," *Communications of the ACM*, vol. 39, pp. 84–90, July 1996.

[11] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, "A resource management architecture for metacomputing systems," in *The 4th Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 62–82, Springer- Verlag LNCS 1459, 1998.

[12] I. Foster, C. Kesselman, and S. Tuecke, "The Nexus approach to integrating multithreading and communication," *Journal of Parallel and Distributed Computing*, vol. 37, pp. 70–82, 1996.

[13] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke, "A directory service for configuring highperformance distributed computations," in *Proc. 6th IEEE Symp. on High Pe rformance Distributed Computing*, pp. 365–375, IEEE Computer Society Press, 1997.

[14] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A security architecture for computational grids," in *ACM Conference on Computers and Security*, pp. 83–91, ACM Press, 1998.

[15] K. Czajkowski, I. Foster, and C. Kesselman, "Co-allocation services for computational grids," in *Proc. 8th IEEE Symp. On High Performance Distributed Computing*, IEEE Computer Society Press, 1999.

[16] V. Varavithya and P. Uthayopas, "Thaigrid: Preliminary studies on thai's high performance computing infrastructure," in *ANSCSE*, 2001.