Grid Computing Initiative at UI:
A Preliminary Result

Riri Fitri Sari, Kalamullah Ramli, Bagio Budiardjo
e-mail: {riri, k.ramli, bbudi@ee.ui.ac.id}
Center for Information and Communication Engineering Research
Electrical Engineering Department – Faculty of Engineering
University of Indonesia
Kampus Baru UI, Depok 16431, Indonesia

Abstract
This paper reviews the Grid Computing and its architecture. The Grid computing paradigm has been a major global research issue in distributed computing since its initiation in 1995. The aim of this research is to provide a Grid Computing environment based on Globus and .net or Java technologies which can be used in the frontier research on the application of the leading edge technology to provide a shared working environment which can use resources available at different network domain as a single resource. This technology can be used for the e-science, such as the Virtual Reality technology and other resource intensive technologies.

In this paper we show our preliminary results on making our own grid computing environment using Java and Axis Web Services. We also show our results and performance evaluations on the application of the prominent Grid Computing platform called Globus. The result shows that this new approach to distributed computing will enable us to be part of the global grid computing initiative as well as envisioned us to build an Indonesian Grid Portal in the near future.

1. Introduction

During the last decade, Grid computing has emerged as a new field in the computer and networking technology. The definition of a Grid computing is a software and hardware infrastructures that provide reliable, consistent, and pervasive access to high performance computing infrastructure. Different from the conventional distributed system, Grid computing infrastructure will provide us with the ability to dynamically link together resources as an ensemble to support the execution of large-scale, resource intensive, and distributed application [1]. The most important aspect in the Grid architecture is the concept of protocol architecture which defines how the distributed system element interact each other.

The distributed computing technology has undergone a considerable development in the last decade. Scientist all over the world have spent a lot of efforts to utilize distributed computing resources. Due to the fact that information access and information processing sometimes need large-scale and powerful resources, particularly for computationally complex problems, some efforts to build new infrastructures from the existing standard has been done by developing web services.

Some grid computing platform use web services technology. Web service is a software system which is designed to support interaction between two machines on the network. Web service has an interface described by Web Service Directory Language (WSDL) format. Other systems interact with web service by means of having the descriptors on the WSDL using Simple Object Access Protocol (SOAP), sends messages using Hyper Text Transfer Protocol (HTTP) using an eXtended Markup Language (XML) form.

Web service architecture defines a distributed system model which focuses on services which interact by exchanging XML documents [6]. One basic element of the web service architecture is the definition of syntax for information exchange. Some efforts to augment this base architecture with additional rule have been considered in order to enable services interaction such as sophisticated behavior as authentication, transaction, and reliable messaging.

Web Services are the basis for grid services, web services are another distributed computing technology in addition to CORBA, RMI, EJB, etc. They allow programmers to create client and server. Grid computing uses the web services technology due to the fact that web services are platform-independent and language-
independent. They use standard XML languages. This means that client program can be programmed in C++ and running under Windows, while the Web Service is programmed in Java and running under Linux.

Most Web Services use HTTP for transmitting messages (such as the service request and response). This is a major advantage when built an Internet-scale application, since most of the Internet's proxies and firewalls allow HTTP traffic. Figure 1 illustrates a simplified view of web service (the framework) architecture and execution sequence.

![Figure 1. Simplified grid (web service) execution sequence](image)

In this paper we presented the preliminary results of the Grid Computing initiative at the University of Indonesia (UI). The aim of our research is to provide a Grid Computing environment based on Globus and .net technologies which can be used in the frontier research on the application of the leading edge technology to provide a shared working environment which can use resources available at different network domain as a single resource which can be used by single user logon.

2. Grid Computing

Different from conventional distributed computing system, grid computing focuses on sharing of large scale resources, innovative applications which required high performance computing support to solve scientific problems. Some multinationals company such as IBM has developed their industry-specific Grid computing initiatives, which focuses on operating systems heterogeneity, and largely used Linux operating system as the focus of their grid strategy, mainly for toolkit hosting. Sun Microsystems has also produced an edition of Grid Engine 5.3 software for Linux which has been distributed through SuSE LinuxAG. On the other hand the .net strategy from Microsoft has also adopted the grid computing principles. Many standard and technology on Grid computing are developed worldwide.

Grid computing has a high level architectural description as depicted in Figure 2. This architecture layering affects the design and implementation of applications in Grid Computing environment [3, 14, 16, 17]. Figure 2 also shows the comparison between Grid protocol and Internet protocol architecture.

![Figure 2. Comparison between Grid architecture and IP architecture](image)

Fabric layer on Grid architecture provide resources, for example, computing resources, replacement system, catalog, network resources, and sensor. Connectivity layer defines the core communication and authentication protocol that is required for network transaction on grid computing. Communication protocol allows data exchange between resources on the fabric layer. Authentication protocol is built on the communication services to provide cryptography security mechanism to check user identity and resources. Resources layer is developed on the top of connectivity layer. This layer defines a protocol for secure negotiation, initiation, monitoring, control, accounting, and the payment of shared operation on the
individual resources. Resources protocol layer focuses on individual resources and ignores the issues on global system. The last layer is the user application that operates virtual organization environment. Figure 3 illustrates Grid application architecture from a programmer’s view.

![Diagram](image)

Figure 3. API implemented by *Software Development Kits (SDK)* [17]

Each layer on all protocol provides access to some useful layer such as resources management, data access, and resources discovery. On each layer, application Programming Interface (API) must be defined for appropriate services.

3. **Globus Grid Computing Framework**

In order to achieve our goals, we formed a Grid computing testbed based on Globus Toolkit which is the defacto standard for Grid computing environment. Grid computing is defined as the hardware and software infrastructure which provides a dependable access, consistent, and can be use any time any where (pervasive), inexpensive, and having a high computational power.

In this section we describe our Grid computing platform implementation using Globus Toolkit. Globus toolkit is a set of components that implements basic services for security, resources location, resources management, communication, etc. Each component defines an interface where the higher level use it to invoke component mechanism and provide implementation that use appropriate operating low level for implementing mechanism on a different environment. This toolkit distinguishes a simple local service and a complex global service that build on the top of local services. Some principal components of Globus toolkit are resources management, communication, information, and security.

Globus is a layered architecture in which high level global services is built on the top of the core local services [15]. On this basic layer architecture, Globus Resources Allocation Manager (GRAM) provides local component for resources management. Each GRAM is responsible for set of resources that operate under the same allocation site-specific policy which are applied by a local resources management system such as, Load Sharing Facility (LSF) or Condor [15].

Communication services on Globus toolkit are provided by nexus communication library. Nexus defines a low level API communication which is used to support a wide range from library and high level communication language. It refers to a different programming model such as message passing on Message Passing Interface (MPI); Remote Procedure Call on C++; and striped transfer on Parallel Application Workspace (PAWs). The Nexus communication services are used extensively in implementing the other Globus module.

The dynamic conditions from Grid environment mean that toolkit component, programming tool, and application must adapt their behavior in responding to the changing of structure and system condition. Globus Metacomputing Directory Service (MDS) is designed to support adaptation process that have been mentioned before which provides enough information on system component that can always be used. MDS store and make information which can be accessed such as architecture type, operation system version, and the amount of memory available in the computer, bandwidth and network latency, protocol communication that can be used, and mapping between IP address and network technology.
4. Results and Discussion
4.1 Globus framework
In this section we presented the result of our Grid Computing infrastructure establishment. First, for the Globus based grid computing framework, we show the result of the execution. The Globus toolkit was installed in our Linux Fedora based System. It can run properly with the existence of supporting components, i.e. Java SDK version 1.4.2, Apache Ant version 1.6, and Junit version 3.8.1 for Grid Service [4, 5, 15]. We extensively tested the functionality and services of Globus in forming a Grid environment.

Figure 4 shows the result of the command execution of

```
[elektro@root gt3]$ bin/globus-service-browser
```

to test the services installed on the container using the graphical user interface of Globus. Other services in Globus are also running well.

Figure 4. The output of service browser Globus command

Grid services are built upon Web services. In addition to that, Open Grid Service Infrastructure (OGSI) defines the additional mechanisms for creating and managing Grid services. Some concepts, such as naming, service data, notification, and life cycle, are part of the key features defined by OGSI. Naming on Grid service is based on web service structure, addressed by Uniform Resource Identifier (URI). URI addressing system on Grid service is called Grid Service Handler (GSH). In order to meet the requirements to communicate with the service, GSH must be resolved to a Grid Service Reference (GSR). Each GSH must be unique and point to a Grid service instance. There cannot be two Grid service instances with the same GSH. A GSH can be thought of as a permanent network pointer to a particular Grid service instance. GSR contains all information needed by the client to communicate with Grid service. Globus toolkit provides mechanism called Handler Resolver to support client resolving GSH to GSR. But since Globus toolkit uses SOAP connection to communicate with Grid service, the GSR is a Web Services Description Language (WSDL) file. WSDL [7] describes interface on web service that will be used, such as what methods are used by the service.

4.2 Java based Grid computing environment results
In addition to Globus-based environment, we built a Java based grid computing application based on Tomcat servlet container and the Axis Simple Object Access Protocol (SOAP) implementation to demonstrate a simple-to-use framework that requests the following minimal grid requirements [11, 12, 13]:

- Machine independence, achieved through the use of Java, Tomcat, and Axis.
- Secure, scalable infrastructure, achieved through the use of SOAP-based Web services for client-server communication.
- Abstracted task execution.

In order to build an application, first, we built server-side SOAP service using Tomcat and Axis. Then a connection stubs to support client-side use of the SOAP service must be created. Thereafter we could build
the main client application. Figure 5 illustrates a typical web service invocation, in which in step 4, the connection stubs to the server are created.

![Figure 5. Typical Web Service Invocation](image)

The SOAP service provides a way for the grid computing application to run services from the SOAP server. The first step in providing the SOAP service is to set up the SOAP infrastructure. We chose Tomcat as the servlet container/HTTP server because it is an open source project and proves to be extremely reliable and easy to use [8]. Axis [9] is used as the SOAP services provider because it too is open source, supports an easy-to-use drag-and-drop service installer, and comes with a tool that creates SOAP client-side stubs from WSDL files [10]. On successful installation of Tomcat 4.1.30 and Axis 1.1, then we wrote the SOAP service class called MatServer. This service contains methods for the matrix application.

The Axis Java Web service (JWS) deployment scheme eases the deployment of the new service. With Axis, we can write a simple service (i.e. own GridConnection class), change the .java file suffix to .jws, then drag and drop it into Tomcat's Axis webapp directory. Axis discovers the new .jws file, compiles it, and makes it available as a SOAP endpoint. Once the new SOAP service is deployed, we had to generate the client-side code to use the service. In this step, we used the Axis WSDL2Java command-line tool to generate the Java files that provide the stubs necessary to invoke the SOAP service. This tool completely eliminates the process of hand-coding the classes required to implement the SOAP interface's client side. The WSDL2Java tool requires a WSDL file. Axis can generate WSDL viewable by a Web browser for any service locally deployed. After saving the WSDL file, we can continue with the stub creation. We saved the WSDL as MatServer.wsdl, and then run the command:

```java
java org.apache.axis.wsdl.WSDL2Java -o . -p <package_name> MatServer.wsdl
```

The main client application is responsible for getting inputs data from users and makes it available to be sent. We tested the grid computing framework by starting the Tomcat/Axis SOAP server, then running the main application. Figure 6 illustrates a successful execution of the grid client application in the grid computing framework to process matrix.

![Figure 6. An execution of a grid client application to process matrix.](image)

Grid computing concept focused on infrastructure that allows users to access a great scale and potential resources. Furthermore, Grid computing describes an infrastructure which is flexible, secure, and coordinated dynamic resource sharing. Globus toolkit is a toolkit that contained many components that allowing a machine to fulfill criteria for Grid computing. With a Globus toolkit installed on the machine, machine can easily being part of a Grid network. Some advantage of becoming part of Grid network is the wide access to the resources that will get and can be accessed by UNIX shell script and API programming. A user who has authority to the network based on Grid will get many facilities. A user can run application
using globus command. Using globus, computation can be performed by other machine which have resources to run application which could be any many within the grid. Then the user will receive the output resulted by that machine. The selection of machine that will run the application requested by the user is based on LDAP, which is used as resources discovery system. We can install services which often used or needed by user to add Globus reliability. The more services installed to the container, the more resources can be accessed. This Grid network based environment does not have a static server. This means that each machine on the Grid network sometimes can be a client and sometime can be a server.

5. Conclusion

We have presented and discussed our preliminary results on building a grid computing infrastructure which is based on Globus. Another Grid computing framework based on Java was also discussed. Grid network with Globus toolkit allows the creation of web service communication without having to consider different environment and platforms on each machine. Using web services it is apparent that we could built a large scale Grid network with large computational power, in which user only have to go through a single sign-on authentication system.

In the long term it is expected that we will establish a grid computing system which consists of different administrative domain to be shared to facilitate intensive computation in providing solution to complex problem which need a high computing resources. It is expected that eventually we could contribute to the international grid computing community to solve some e-science problems. E-science is defined as the sciences supported by Internet-enabled distributed global collaboration, which process a large number of data, large-size computational resources, and high performance visualization. It could be interpreted as grid computing application and data, middleware development, and access to hardware necessary to support new informatics techniques, such as virtual reality.

Reference

1. F. Berman, G. C. Fox, A. Hey, Grid Computing Making the Global Infrastructure Reality, John Willey and sons, November 2003