PROCEEDINGS OF SCIENCE

Grid Management Using Instant Messaging

Tan Nam Beng[1]
Nanyang Polytechnic
Singapore
E-mail: tan_nam_beng@nyp.gov.sg

In this paper, we describe an implementation that utilizes an instant messaging system for providing Grid Computing submission, monitoring and management capabilities. Currently, accessing the Grid Network is usually via command line-based terminals or a web-based user interface. The user will compose the job specification via some standard job submission language. The target grid node to execute the job can be predetermined by the user or a Grid broker can assist to identify the suitable grid resource. There is presently limited intuitive means for users to assess the grid network utilization and situation before making the decision. An improved method for Grid job submission, monitoring and management is proposed.

The International Symposium on Grids and Clouds (ISGC) 2013
March 17-22, 2013
Academia Sinica, Taipei, Taiwan

[1] Speaker

PoS(ISGC 2013)023

1. Introduction

A conventional grid computing (FIG. 1) allows members to share computing resources in a network of computers in a distributed and coordinated manner [1], [2], [3]. Accessing the grid network is usually done at a user's workstation, either via a command line or web-based interface. Typically, a user of a grid node composes the job specification in accordance with a computing language and syntax. The job specification is then sent to a Resource Broker engine in a Supervisory Node for the job to be submitted to a suitable Resource Node. In addition, the job is monitored by a Job Monitor engine. Both the Job Submitter and Job Monitor engines reside in the Supervisory Node. The user is, therefore, unaware whether he/she is using a computer or data at one's resource node or that at a remote resource node. In other words, the complexity of sharing computing resources in a grid network is hidden from the users; there is no intuitive means for users to assess or monitor the grid network utilization before submitting a job. Further, there is no way to determine whether a Resource Node in the grid has failed. For example, if one of the node connectivity is temporarily affected but a job has been successfully completed, the status of the job is deemed to have failed.

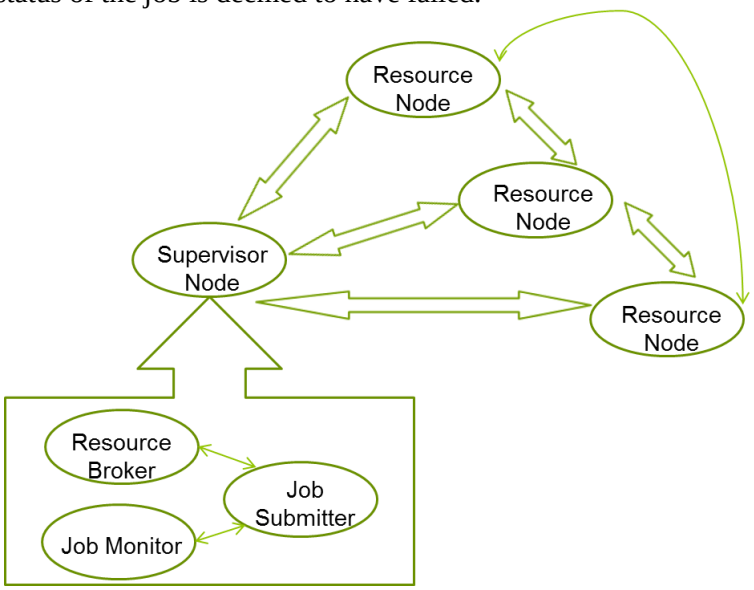

Figure 1: Grid Network

2. Instant Messaging

An Instant Messaging ("IM") system is a client-server system. The IM server tracks the presence information of IM clients that an IM client has subscriptions with. Presence information includes means, ability, capability, status, willingness and location. By publishing presence information, it is getting easier to contact people or to utilise resource. It helps the current service offering much better service according to customer's different presence status [4], [5] , [6]. The IM system thus links these IM clients together and allows the IM clients to communicate with one another. The IM Server is connected to a Directory/Authentication Server. The Directory/Authentication Server runs a Roster engine. The Roster engine manages transactions and associated roster subscription details of the IM clients connected to the IM

PoS(ISGC 2013)023

Server. The roster subscription details of the IM clients are stored in a Repository. For example, as shown in FIG. 2, IM Client A connects up with the IM Server and notifies the IM Server that it is connected. The IM Server returns a list of contacts to IM Client A together with their presence information from the Repository. The contact list contains the contact details of those clients who have subscribed to both the IM Server and the IM Client A. The IM Client A then sends out its presence information to the IM Server stating that it is "available". The IM Server updates its Repository and simultaneously sends the presence information of IM Client A to all clients that have subscribed to the IM Client A. For example, as shown in a Roster in FIG. 2, IM Client B has IM Client A in its contact list (but not IM Client C). The IM Server would send the presence information of IM Client A to IM Client B only.

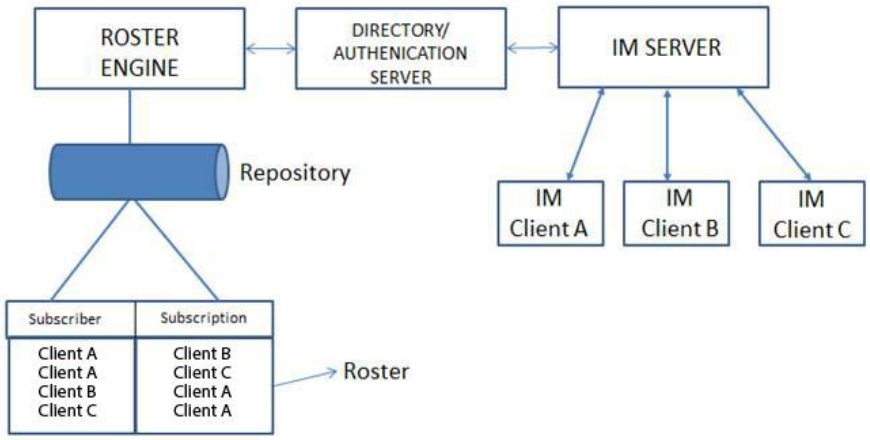

Figure 2: Instant Messaging Architecture

Conventional Instant Messenger (IM) systems, such as MSN, Yahoo, AOL, and Skype [7] and so on, can therefore provide computer users who have subscribed to and are connected on-line to such IM servers to "chat" or communicate synchronously. Some of these messaging may not be routed through the respective IM server but directly with other IM clients A, B, C by communicating through the use of Peer-to-Peer protocols after the IM server(s) has/have linked the IM clients together. In addition, these conventional IM systems can also provide asynchronous communication. Hence, an IM system can provide a suitable platform for implementing a grid computing network where the IM clients are Resource nodes or computing machines. FIG. 3 shows a conventional IM system being hosted on an internet gateway. As depicted in FIG. 3, the IM clients etc. are linked to the internet gateway through their respective Internet Service Provider (ISP) servers.

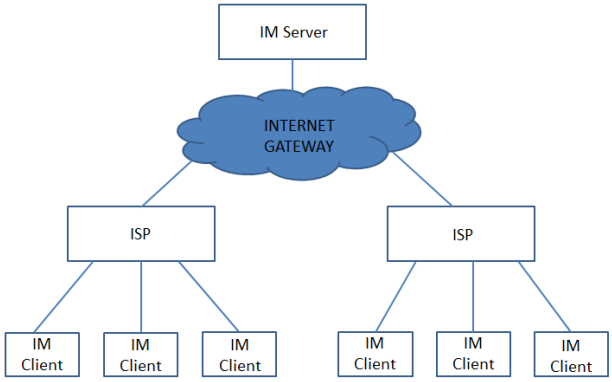

Figure 3: IM System hosted on Internet Gateway

3. Grid Management Using Instant Messaging

It can thus be seen that there exists a need for a new system for implementing grid computing submission, monitoring and scheduling using Instant Messaging in a manner that can minimize if not overcome the limitations of conventional grid computing systems.

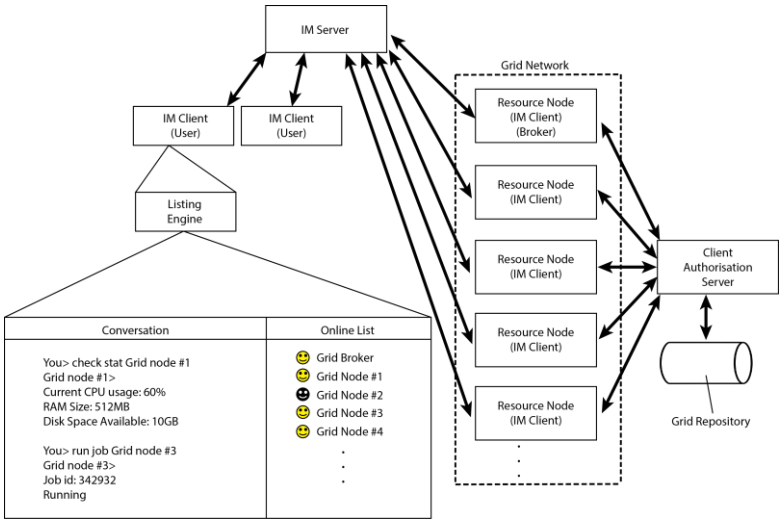

Figure 4: IM System integrated with Grid Computing

Above shows an architecture of an Instant Messenger (IM) system forming a grid computing. The IM system has an Instant Messaging (IM) server, an Authentication server connected to the IM server, a Repository connected to the Authentication server and Resource nodes (IM clients) connected to the IM server. Each Resource node includes a terminal for a user to send IM text messages to the IM server and other Resource nodes users. The Repository includes a Roster,

PoS(ISGC 2013)023

which contains all the Resource nodes' (IM clients') subscriptions to the IM server and other Resource nodes (IM) clients. In above scenario, all the Resource nodes (IM Clients) and IM Clients (users) have subscription with each other. Each Resource node(IM Client) is connected to the IM system via a background service process and does not require human intervention; the background service process connects a Resource node(IM client) to the IM System each time the computing machine is started. In the proposed IM system, once an IM client (user) has successfully signed-on to the IM system, the user is allowed to submit jobs to one or more Resource nodes. In IM System, each IM client (user) is required to register with the IM system; the registration data are stored in the Client Authorisation server, for example in a Grid Repository. A user may first have to enrol onto the grid which may require authentication for security purposes. The user positively establishes his identity with a Certificate Authority (CA). Each IM client (user) installs listing engine into one's computer. The listing engine provides a contact list showing the status of all the available or active Resource nodes that are on-line by using the log-in data kept in the Repository. An IM client is then able to submit a job request to the grid computing system via the IM system by selecting a Resource node or group of Resource nodes from the contact. Typically, the listing engine is installed when sign-on or registration to the IM system is successfully completed. The listing engine may prompt an IM client that a new Resource node is added to the IM system after a new sign-in or registration is made. An example of a dialogue prompt from the listing engine is shown in FIG. 5. An IM Client(user) who is on-line can either choose to allow the new Resource node to be added to one's contact list or not. If the user's choice is positive, data in the Repository or Client Authorisation Server is updated and the user's contact list would indicate the newly added Resource node as "available" or "on-line". If the user's choice is negative, data in the Repository or Client Authorisation Server is not updated and the user's contact list would not list the newly added Resource node or would indicate it as "not available".

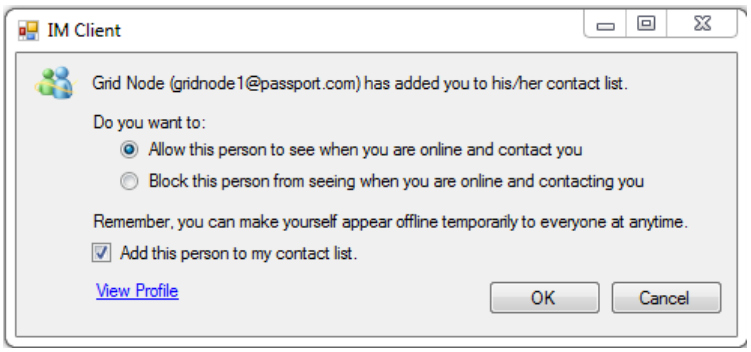

Figure 5: Adding Grid Nodes

The contact list in diagram shows the status of the utilisation and job status of each resource node in the IM System. The status of utilization of each available resource node is indicated by the CPU usage, RAM size and available data storage or disc capacity whereas the status of a job at each node is indicated by a job identification number if it is processing a job. In addition, the status of each Resource node is colour coded. For example, when a Resource node is "available" or online, it is indicated by a green smiley icon. If a Resource node is not available or offline, it is indicated by a grey icon. In addition, a Resource node is also identified if it is a Resource broker. A resource node can be identified by its node numbering and broker label.

PoS(ISGC 2013)023

Each IM client is allowed to be additionally given the brokering function. When a group of resource nodes are selected to execute a job, a predetermined Resource node in the group acts as a supervisory broker. An advantage of the IM system is that a Resource node having brokering function is able to broker a job it is handling or assigned to other Resource nodes. A job specification includes an executable application and defines the computing resources, the number of Resource nodes required to execute a job, the input data files and a job expiry time. The computing resource definition includes the operating system, processor type, RAM, data storage capacity and so on.

4. Implementing IM in Grid Management

The following section provides a brief description on the implementation of the system. Every node in the grid network is to be installed with the IM Client software. This includes the grid broker in the grid network. The IM Client may be implemented by using freely available IM libraries (e.g. http://jmsn.sourceforge.net/, http://hamsam.sourceforge.net/) [8], [9], [10]. Every time the grid node is available for grid usage, it will have to use the IM Client to connect to the IM Server. This is to ensure the usability of the nodes by the users. When the grid node is not available, it will be shown as offline at the IM client. The users have to register before being exposed to the nodes available in the grid network. Registrations may be in the form of email or online registration. Once registered, the users' registration information will be stored in a repository in a Client Authentication Server. The Client Authentication Server may or may not be part of the grid network. Registration information may contain the following (but not limited to): the validity of the registration, the type of access and rights the users are entitled to and number of jobs the users may submit in a day. The process of authentication will be done by the Grid node when the user is adding the grid nodes into his IM and when the user is communicating with the Grid node. The Grid node will contact the Client Authentication Server to have the user request (of adding or of job submitting) validated. When a user failed to be validated, an error message will be sent to the user. Similarly when a user is validated correctly, a message informing him that his job has been assigned (to a grid node) will be sent. The adding process of the grid nodes over at the user sides may be done automatically via some scripts or manually. Typically a user will usually perform queries to check how busy the grid is, to see how the submitted jobs are progressing and to look for resources on the grid. Thus, prior to submitting job, user can check the status of specific grid nodes by issuing the following commands:

a. check processor capacity
b. check storage capacity
c. check memory capacity

PoS(ISGC 2013)023

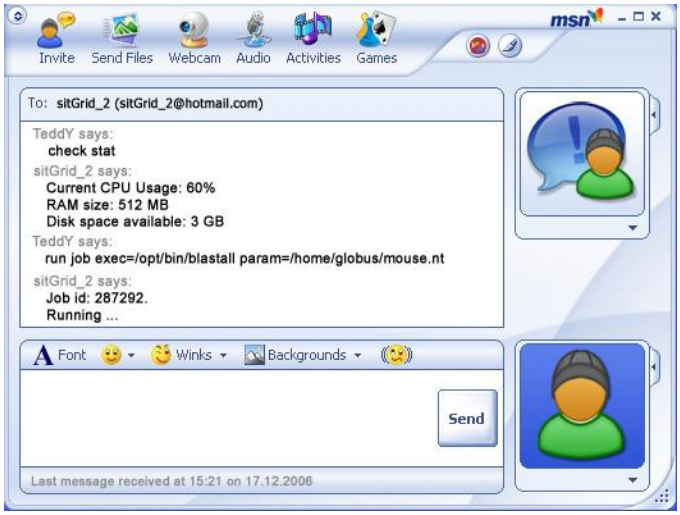

Figure 6: Check node status

There are two methods available to submit a job:

1. User can select a particular grid node and submit the job to it, or
2. User can submit the job to a designated grid broker and asks the broker to assign the job according to the capability of the broker.

Various commands will be implemented to create a list of syntax which may be used by the user in his communication with the grid nodes. The list may contain the following commands or more:

Submit job

- specifying criteria for selection of node (for Grid broker)
- specifying requirement for number of nodes to execute the job
- specifying requirement for minimum processor/memory to execute job

When a job is assigned to a grid node, the grid node will translate the text input made by the user in the IM Client into a Grid-understood job. As there will be fixed command syntaxes which users have to follow when interfacing with the Grid nodes over IM Client, the translation may be done automatically. After translation, the grid node will then start processing the job.

When the job is completed, users will be informed by a push (prompt) message by either the grid node which executed the job or the grid broker, depending on who is in charge of the job. The result of the job may be returned in the form of text messages or files. Text messages may be presented in the form of messages to the user. Files may be presented in the form of file transfer to the user.

FIG. 7 shows an IM window of the supervisory broker transferring the result of the completed job.

PoS(ISGC 2013)023

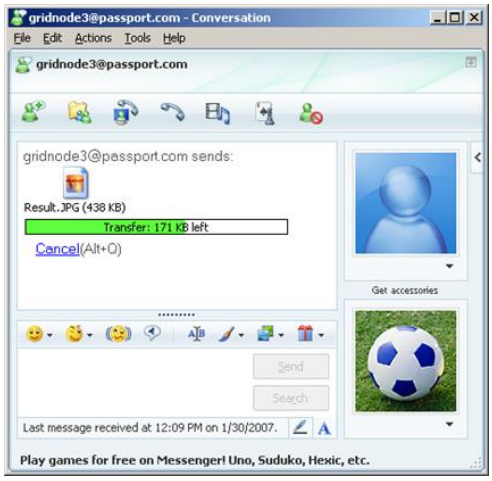

Figure 7: Transferring completed Job

FIG. 8 shows a window of the requestor IM client(user) acknowledging receipt of the completed job result from the supervisory broker (gridnode 3). Once the completed job result is received by the requestor IM client(user), the process is ended.

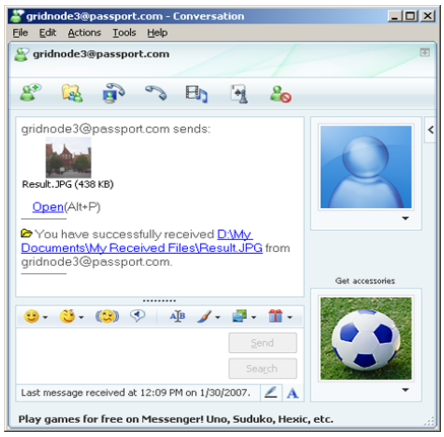

Figure 8 : Receiving completed job

The figure showed a typical job submission process flow:

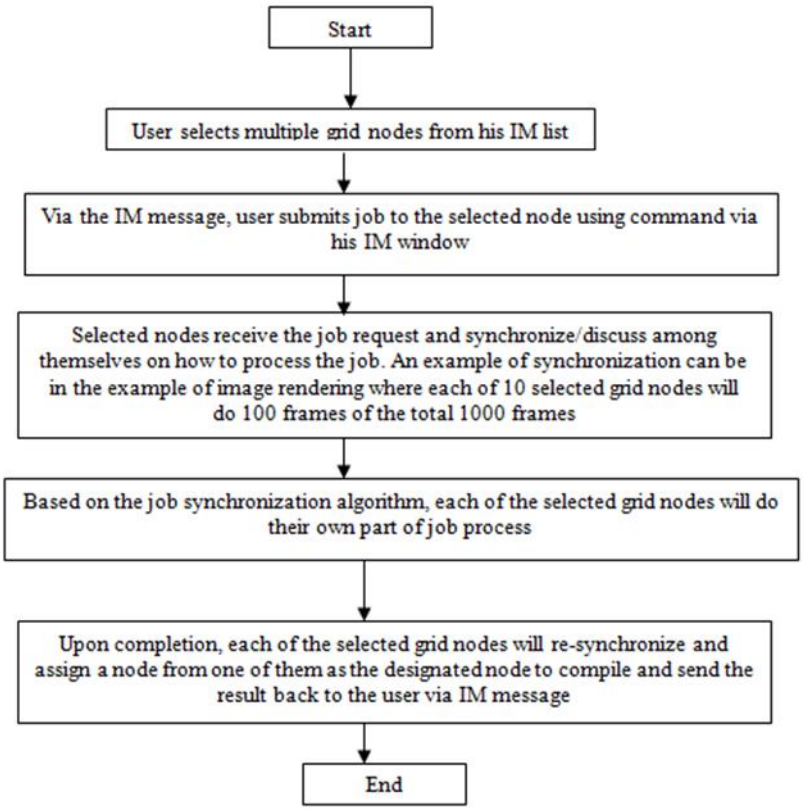

Figure 9: Job Submission Flow

An IM client (user) selects a group of resource nodes from one's IM contact list. In the next step, the user submits one's job request via one's IM window, for example, by attaching the job specification and submitting it to the group of selected Resource nodes for execution. The group of selected Resource nodes receives the job request and a selected Resource node, designated as a supervisory broker, splits the job request into smaller job units according to the number of nodes specified in the job specification and/or number of selected Resource nodes. For example, when 10 Resource nodes are selected to perform a job involving image rendering of 1000 frames, each selected Resource node would then render 100 frames of the images. The job units are then brokered out among the selected Resource nodes for processing. FIG. 10 shows a IM window of a selected Resource node participating in a job execution. When a job is completed, the supervisory broker consolidates all the completed job units together and sends the completed job result back to the IM client (user) that issued the job.

PoS(ISGC 2013)023

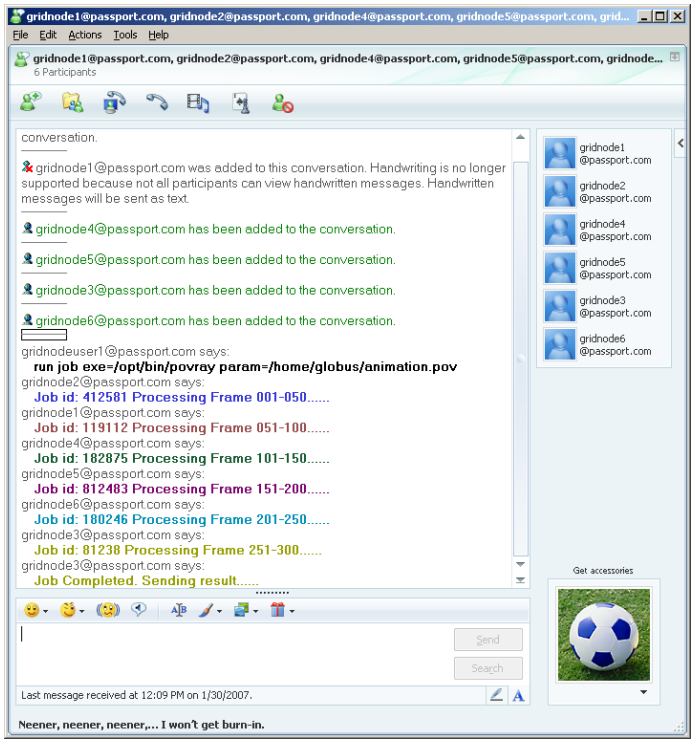

Figure 10 : Resource nodes participating in Job Execution

5. Related Works

With the advent and proliferation of non-secured instant messaging (IM) via the Internet, information is sent within the organization and between organizations. This poses a problem especially when it is used to send confidential business messages as they are open to the prying eyes of eavesdroppers. One method used to solve the confidentiality problem is to use Public Key Infrastructure (PKI). This requires users to own and install digital certificates issued by the Certificate Authorities (CA). IM is peer-to-peer by nature and the method involve a third party to maintain information, e.g. certificate information required for encryption / decryption of messages. Our institution is researching in this area that relates to a cryptographic code installed in end-user stations for secure Instant Messaging. A passphrase is generated to invoke the secured channel. The passphrase may be used to generate a symmetric key for encryption. The passphrase can be split into sub-passphrases, which can be sent to a recipient via separate communication channels. The sub-passphrases can then be assembled to obtain the complete passphrase for generating the symmetric key for decryption. The solution provides more efficient security algorithm and key management/distribution [11]

PoS(ISGC 2013)023

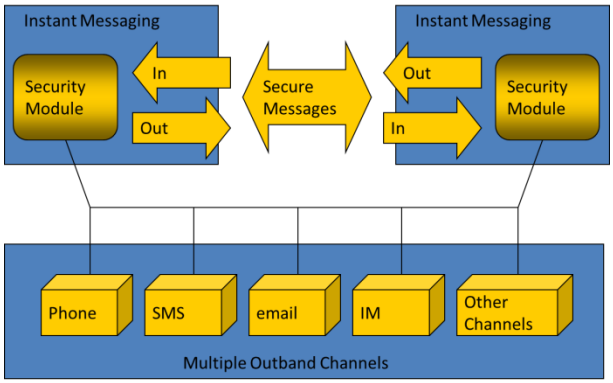

Figure 11 Secured Messaging using Out-band Authentication

References

[1] OGF – Open Grid Forum. Online. http://www.ogf.org.

[2] Globus Toolkit. Online. http://www.globus.org/toolkit/.

[3] Bart Jacob, Michael Brown, Kentaro Fukui, Nihar Trivedi, *Intoduction to Grid Computing*, IBM Red Book,December 2005

[4] Linan Zheng, Dr Stephan Rupp *Instant Messaging: Architectures and Concepts* , June 2005

[5] MDay, J Rosenberg, H.Sugano *A Model for Presence and Instant Messaging*, RFC 2778, , Feb 2000

[6] Peter Saint-Andre, Kevin Smith & Remko Troncon, *XMPP: The Definitive Guide*, 2009

[7] Tim Van Lokven, *Review and Comparision of Instant Messaging Protocol*, January 2011

[8] Online: http://taverna.sourceforge.net

[9] XMPP toolkit (http://xmpp.org/xmpp-software/libraries/)

[10] IM SDK (http://www.interactiveni.com/sdk/)

[11] Teo Yong King, *Secure Messaging using Outband Mode Authentication*, Oct 2007

PoS(ISGC 2013)023

