AGENT TOOLS TO SETUP GRID NODES
IN LOCAL AREA NETWORK

ABDUL RASHEED KHAN BIN YUSOF KHAN

UNIVERSITI TEKNOLOGI MALAYSIA
UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT PAPER AND COPYRIGHT

Author’s full name: ABDUL RASHEED KHAN BIN YUSOF KHAN

Date of birth: 820330-14-5573

Title: AGENT TOOLS TO SETUP GRID NODES IN LOCAL AREA NETWORK

Academic Session: 2008/2009

I declare that this thesis is classified as:

☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1972)*

☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*

☑ OPEN ACCESS  I agree that my thesis to be published as online open access (full text)

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:

1. The thesis is the property of Universiti Teknologi Malaysia.
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

__________________________
SIGNATURE

__________________________
SIGNATURE OF SUPERVISOR

820330-14-5573
(NEW IC NO. / PASSPORT NO.)

DR. MUHAMMAD SHAFIE BIN ABD LATIFF
NAME OF SUPERVISOR

Date: 30 APRIL 2009

NOTES: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction.
“I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Science (Computer Science).”

Signature : ...............................................................  
Supervisor  : DR. MUHAMMAD SHAFIE BIN ABD LATIFF  
Date       : 30 APRIL 2009
AGENT TOOLS TO SETUP GRID NODES IN LOCAL AREA NETWORK

ABDUL RASHEED KHAN BIN YUSOF KHAN

A thesis submitted in fulfilment of the requirements for the award of the degree of Bachelor of Science (Computer Science)

Faculty of Computer Science and Information System
University Technology of Malaysia

APRIL 2009
“I declare that this thesis “Agent Tools to Setup Grid Nodes in Local Area Network”
is the results of my own work except the resources as cited in the references. The
thesis has not been accepted for any degree and is not
concurrently submitted in candidature of any other degree.”

Signature : ............................................................
Name : ABDUL RASHEED KHAN BIN YUSOF KHAN
Date : 30 APRIL 2009
To my beloved parents, and family members
ACKNOWLEDGEMENT

First and foremost, I would like express my personal appreciation to my supervisor, Dr. Muhammad Shafie Bin Abd Latiff for his guidance and time. His patience and trust has encouraged me to complete this project. His efforts really mean a lot to me. Thanks for his comments and ideas that help me to improve the accuracy of this project. Next, I would like to convey my greatest gratitude for my parents and family members. Your confidence and trust in me have made a lot of difference. Thank you for always being beside me and accompany me through my life. You are always there to share my happiness and sorrow. Thanks for always been there.
The need for more computing power and memory to run application and software had attracted more researchers to find some alternative technologies in distributing the jobs across the network. To meet this requirement grid computing technology was started. The problem with grid computing environment is that the complexity of its application installation and setup. The aim of this project is to develop an agent tool that will enable complex installation and setup procedures in creating grid nodes to be simplified and time saved. By creating this agent tool, the complexity of the installation and setup of Globus 4.2.1 will be compartmentalized. The methodology used to create this application is evolutionary prototyping model due to its suitability in developing this type of application. As a result the agent tool created will help users to setup grid nodes with more efficient and productive.
ABSTRAK

# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>SUBJECT</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>ACKNOWLEDGEMENT</td>
<td>iv</td>
</tr>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>ABSTRAK</td>
<td>vi</td>
</tr>
<tr>
<td></td>
<td>TABLE OF CONTENTS</td>
<td>vii</td>
</tr>
<tr>
<td></td>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>LIST OF ABBREVIATION</td>
<td>xii</td>
</tr>
<tr>
<td></td>
<td>LIST OF APPENDIX</td>
<td>xiii</td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2 Background Problem</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.3 Importance of the Project</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1.4 Project Aim</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1.5 Project Objectives</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1.6 Project Scopes</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1.7 Summary</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>LITERATURE REVIEW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1 Introduction</td>
<td>6</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>2.2 Grid Computing</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>2.3 MYREN</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2.4 Globus Toolkit</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.5 Cent OS 5.2</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>2.6 Red Hat Enterprise Linux</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>2.7 Visualization Pipeline for Medical Dataset</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>2.8 Workflow Globus Installation and Setup</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2.9 Globus Installation process in Linux OS</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>2.10 Globus 4.2.1 Setup Process</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>2.11 Summary</td>
<td>37</td>
<td></td>
</tr>
</tbody>
</table>

3  METHODOLOGY

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction</td>
<td>38</td>
</tr>
<tr>
<td>3.2 Project Methodology</td>
<td>39</td>
</tr>
<tr>
<td>3.3 Justification of Methodology</td>
<td>39</td>
</tr>
<tr>
<td>3.4 System Development Methodology</td>
<td>40</td>
</tr>
<tr>
<td>3.4.1 Planning Stage</td>
<td>42</td>
</tr>
<tr>
<td>3.4.2 Analyzing Phase</td>
<td>42</td>
</tr>
<tr>
<td>3.4.3 Design Phase</td>
<td>42</td>
</tr>
<tr>
<td>3.4.4 Development Phase</td>
<td>43</td>
</tr>
<tr>
<td>3.4.5 Testing and Evaluation Phase</td>
<td>44</td>
</tr>
<tr>
<td>3.5 System Requirements</td>
<td>44</td>
</tr>
<tr>
<td>3.5.1 Hardware Requirements</td>
<td>44</td>
</tr>
<tr>
<td>3.5.2 Software Requirements</td>
<td>45</td>
</tr>
<tr>
<td>3.6 Summary</td>
<td>45</td>
</tr>
</tbody>
</table>

4  SYSTEM DESIGN

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>46</td>
</tr>
<tr>
<td>4.2 System Architectural Design</td>
<td>47</td>
</tr>
<tr>
<td>4.2.1 Explanation of System Architecture Design</td>
<td>48</td>
</tr>
<tr>
<td>4.3 General Design</td>
<td>48</td>
</tr>
<tr>
<td>4.4 User Interface Design</td>
<td>51</td>
</tr>
</tbody>
</table>
5 IMPLEMENTATION AND TESTING

5.1 Introduction 55
5.2 Development Environment 55
   5.2.1 Installation of Required Software 56
5.3 Coding 56
   5.3.1 Installation codes for Globus Toolkit 4.2.1 57
   5.3.2 Settings in the First machine 61
   5.3.3 Settings in the Node machine 68
5.4 System Testing 74
   5.4.1 Installation Testing 74
   5.4.2 Master Machine setup testing 76
   5.4.3 Node Machine setup testing 78
5.5 Conclusion 79

6 CONCLUSION AND DISCUSSION

6.1 Introduction 80
6.2 Achievement 80
6.3 Limitations and Challenges 81
6.4 Expectations 82
6.5 Summary 82

REFERENCE 83
APPENDIXES 84
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Malaysian Research and Education Network</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Globus Toolkit version 4 (GT4)</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>First machine setup workflow</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Node machine setup workflow</td>
<td>17</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Process flow of an application design in evolutionary prototyping.</td>
<td>41</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Basic Architecture Design</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>System Architecture Design</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Grid Architecture</td>
<td>48</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Agent tool User Interface design</td>
<td>52</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Snippets coding for installing prerequisite software before installing Globus toolkit 4.2.1</td>
<td>57</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Snippets coding for compiling and installation of Globus toolkit</td>
<td>60</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Snippets of configuring the security for first machine</td>
<td>61</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>Snippets in setting up proxy server on the first machine</td>
<td>62</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Snippets in setting up grid ftp on the first machine</td>
<td>63</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>Snippets in setting up Web service container, RFT and GRAM4</td>
<td>65</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>Snippets in setting up security in node machine</td>
<td>68</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>Snippets for Gridftp on node machine</td>
<td>70</td>
</tr>
<tr>
<td>FIGURE NO</td>
<td>TITLE</td>
<td>PAGE</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>Snippets for Web service Container and GRAM</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>RSL for node machine</td>
<td></td>
</tr>
<tr>
<td>Figure 5.10</td>
<td>Pre-requisite Installation testing and display</td>
<td>74</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>Checking the system before installation of Globus 4.2.1</td>
<td>75</td>
</tr>
<tr>
<td>Figure 5.12</td>
<td>Installation completion of Globus 4.2.1</td>
<td>75</td>
</tr>
<tr>
<td>Figure 5.13</td>
<td>Master machine security files</td>
<td>76</td>
</tr>
<tr>
<td>Figure 5.14</td>
<td>Myproxy configuration testing</td>
<td>76</td>
</tr>
<tr>
<td>Figure 5.15</td>
<td>Gridftp configuration testing</td>
<td>77</td>
</tr>
<tr>
<td>Figure 5.16</td>
<td>Test in submitting job by an application</td>
<td>77</td>
</tr>
<tr>
<td>Figure 5.17</td>
<td>Node Machine security test</td>
<td>78</td>
</tr>
<tr>
<td>Figure 5.18</td>
<td>Node Gridftp configuration testing</td>
<td>78</td>
</tr>
<tr>
<td>Figure 5.19</td>
<td>Testing in receiving and submitting a job in node machine</td>
<td>79</td>
</tr>
<tr>
<td>Figure C1</td>
<td>To Lunch the agent tool</td>
<td>90</td>
</tr>
<tr>
<td>Figure C2</td>
<td>Agent tool user Interface</td>
<td>91</td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT4</td>
<td>Globus Toolkit version 4</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MYREN</td>
<td>Malaysian Research and Education Network</td>
</tr>
<tr>
<td>OS</td>
<td>Operation System</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>RAD</td>
<td>Rapid Application Development</td>
</tr>
<tr>
<td>SDLC</td>
<td>Software Development Life Cycle</td>
</tr>
<tr>
<td>UUID</td>
<td>Universally Unique Identifier</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modelling Language</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>TITLE</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>Gantt chart for PSM 1</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>Gantt chart for PSM 2</td>
</tr>
<tr>
<td>APPENDIX C</td>
<td>User Manual</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Introduction

As time passes, human beings become more reliant to computers as the main source of information. Computers are created to operate faster so it can be used for highly calculation intensive task. Scientific researchers are the main user of computer resources where they are able to perform highly calculation-intensive tasks such as problems involving quantum mechanical physics, weather forecasting, climate research, molecular modelling, physical simulations, cryptanalysis and more.

To satisfy the need in high computing resources for this kind of task, a few innovations were created to solve the problem. One of them is the creation of Supercomputers where these computers are capable in processing billions of calculation in a split second. Although these solution is the best for performing high calculation intensive task but they are meet with a few challenges. One of them is the costs of owning the supercomputers are too expensive for most researches. Supercomputer also generates large amounts of heat and must be cooled I order to operate for a long period of time. Cooling supercomputers is a major problem. So more cost will be incurred to solve this problem.
Another way in solving highly calculation intensive task is the introduction of cluster computing. Computer cluster are a group of linked computers, working together closely so that in many respects they form a single computer. These computers are connected to each other using a fast local area network (LAN). Clusters are usually deployed to improve performance and availability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

Although computer cluster is considered as a better solution for researchers needs, they also have their own problems. One of them is that clusters need to be deployed within the same geographical area. So to deploy the clusters the need of space is crucial. These will also be considered as additional cost for researches to consider when they use this method.

In the early 1990’s another solution was thought out in solving highly calculation intensive task which is Grid Computing. Grid computing or grid clusters is a technology closely related to cluster computing. The key differences between grids and traditional clusters are that grids connect collections of computers that are geographically dispersed. In order to deploy a grid cluster the infrastructure of fast wide area network is needed. Rather than acting like one single computer, grid computing is considered as more like a computing utility. In which independent task is sent in packets to computers within the network to be processed. When the calculation is over the result will then be sent back to the node that requested it.

There are a few advantages in using grid computing where computer resources will able to be shared among a group of researches in different geographical area and it is more cost effective.

1.2 Background problem

MYREN is the Malaysian Research and Education Network that provide high speed broadband network for research organization in Malaysia to link up with
each other to carry out collaborative research or application. This network is a low cost alternative compared to commercial networks.

In using MYREN’s high speed bandwidth capacity it will allows for data-intensive application and sharing of computer resources more efficiently. In the existence of this high speed bandwidth capacity network between 14 universities and research institution throughout Malaysia grid computing is a possibility. In order to use this network as grid computing, a series of configuration need to be performed to a host computer or a node within the network.

Some researchers do have some problem in configuring their computer so that they will be able to use the resources in MYREN to run highly calculation intensive task application. This process also will consume time for the researches and they will encounter few problems along the way. Some will also abandon the task when they consider the task to be too complicated and time consuming.

In order to solve the problem of the complexity of setting up grid nodes, a proposal of creating an agent tool that will be able to configure a host computer or a node automatically so that the host computer will be connected to the network faster. By creating an automated agent for the researchers, this will encourage them to use the infrastructure and resources that already in place.

The agent tool created in this project will only be tested within a Local Area Network. Although it’s only tested within a Local Area Network, its result will be the same as testing it within the MYREN grid. This is because the configuration used by the agent tool is the same no matter which Network it is used on.

1.3 Importance of the Project

In creating this application the process of configuring a node in Local Area Network will become hassle free and more reliable. Researchers will only need to run the application and the configuration will then be handled by the agent tool. This will encourage more researchers who need more resources in their computation
needs to utilize the resources. High computing assets can be shared across the organization more effectively.

1.4 Project Aim

The aim of this project is to develop an agent tool application in order to simplify the creation of grid nodes in Local Area Network and also creating a setup program void of error and problems.

1.5 Project Objectives

The objectives that to be achieved in this project is as follows:

1. To develop an agent tool to setup grid nodes in Local Area Network using Globus toolkit.
2. To setup up the grid nodes in Local Area Network using Cent OS 5.2 and Red Hat 5.2 as an Operating System.
3. To run a grid application once the grid nodes is setup between the host nodes in the Local Area Network.

1.6 Project Scope

1. The agent tool is to install and setup Globus toolkit 4.2.1 within a Local area Network.
2. This agent tool will allow grid application to run within a Local Area Network once the installation and setup process is completed.
3. This agent tool is only for connection between the grids, so thread scheduling and load balancing will not be included in the agent tool for application run within the grid.

4. The agent tool will be applied on Cent OS 5.2 or Red Hat 5.2 Operating System.

1.7 Summary

In conclusion of this chapter, there are a few innovation created to handle high capacity calculation intensive task. One of the latest technologies is the creation of grid computing where high computing assets across organization could be shared for research and educational purposes. In order to utilize this resources a few complex series of configuration need to be perform on a host computer. Because of this, the creation of an agent tool that will enable simplify configuration of a host computer so that researchers will be able to connect to the grid. With the creation of this agent tool it will encourage researches to use the resources between them to perform high calculation intensive application using the MYREN grid.
CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter it will begin by the introduction on what is grid computing, where it will explain in brief what is the purpose of grid computing and compare it to other term that we already familiar with. Then to continue on with the introduction of the MYREN (Malaysian Research and Education Network) which a high speed wide area network where it connects 14 universities and research organization throughout Malaysia. In order to setup this grid nodes, the introduction of Globus toolkit as a fundamental enabling technology that will allow people share computing power, databases, and other tools securely without sacrificing local autonomy will be explained. Lastly the introduction of the operating system which is “Scientific Linux” where the agent tool application will be implemented on will be briefly introduced. So this is, in brief the flow of what will be written within the chapter.
2.2 Grid Computing

By definition Grid computing is a form of distributed computing whereby it composed of a cluster of networked computer loosely coupled acting in concert to perform very large and highly calculation intensive task. This technology has been applied to computationally-intensive scientific, mathematical and academic problem through volunteer computing. Another main character of grid computing is that it is dispersed in different geographical area with the connection of high speed bandwidth capacity wide area network. [1]

The term grid computing originated in 1990’s as a metaphor for making computer power as easy to access as an electric power grid. The idea of grid computing was thought out by three people which were Ian Foster, Carl Kesselman and Steve Tuecke which were widely considered as the “father of grid” [2]. They lead the effort in creating the Globus toolkit which was the operating system behind the grid so that additional services based on the same infrastructure could be created by other developers. While Globus Toolkit remains the standard for building grid solution, a number of other tools were also created to answer some subset of services needed to create a global grid.

There are some will say that grid computing is just like one large supercomputer but in reality these term is in contrast with the real notion of supercomputer. Supercomputer is a computer that has many processors that is connected by a local high speed computer bus [3]. While grid computing in general is a special type of parallel computing, this relies on complete computer connected to a wide area network using a conventional network interface such as the Ethernet. Grid computing also sometimes is confuse with Cluster computing, but clusters are a group of computers that are linked together to form a single computer which is connected through fast local area network [4]. Although these three technology are been used to solve the same problem the method of connection between them is different, this is what differentiate grid computing with supercomputer and computer cluster.
To continue on, there are many advantages of grid computing. The primary advantage is that each node can be purchased as commodity hardware which when combined can produce similar computing resources to a multiprocessor supercomputer, but at a lower price. Thus this arrangement is well suited for application in which multiple parallel computations can take place independently without the need to communicate intermediate result between processors. This arrangement also makes it possible to write and debug on a single conventional machine, and eliminate complication due to multiple instances of the same program running in the same shared memory and storage space at the same time.

It could be said that grid computing will the future advancement for computer network; this is shown by a well known project which is the World Community Grid. The mission of project is to create the largest public computing grid benefitting humanity. This work is built on the belief that technological innovation combined with visionary scientific research and large-scale volunteerism can change our world for the better \cite{5}.

2.3 MYREN

MYREN is an acronym for Malaysian Research and Education Network. MYREN is a high speed broadband network for research organization in Malaysia to link up to each other and carry out collaborative research and application. The features in MYREN are as follows.

1. It’s a low cost high speed broadband network alternative to commercial network.
2. With its high bandwidth capacity allows data intensive application and sharing of resources to be conducted more efficiently.
3. It also provide an experimental platform for researches to investigate, develop and test new network and internet technologies and application prior to commercial use.
4. MYREN is linked to other countries Research and Education Network (RENs) so that collaboration could take place on a global scale.

MYREN is currently connecting 14 Universities and research institutions throughout Malaysia via a nationwide IP backbone at 45Mbps capacity. The backbone covers 8 states in the east and west Malaysia. One advantage of MYREN is that it is versatile and able to support all type of traffic. Researches that are currently using this network are able to run and operate various applications such as video conferencing, digital video transport system (DVTS), data computing, visualization modeling and e-learning. That is why it is important for MYREN to continue to deliver quality performance required by the research community.

Members of MYREN are connected via respective access link, which can either be a fiber optic link or microwave link. The access connection varies according to the user requirement. MYREN connectivity is centralized at MYREN Network Operation Center (NOC) located in Cyberjaya. Below is the diagram of Universities and research institutions that are currently connected to MYREN.

![MYREN Network Diagram](http://www.myren.net.my/network/domestic)

**Figure 2.1:** Malaysian Research and Education Network

(Source: Malaysian Research and Education Network Website – “http://www.myren.net.my/network/domestic”)
MYREN is also connected to several international research communities in Asia Pacific, Europe and North America via the Trans-Eurasia Information Network 2 (TEIN2) and internet2. The connectivity coverage includes China, South Korea, Japan, Thailand, Vietnam, Philippines, Singapore, Indonesia, Australia, Europe and the United States. By connecting to the MYREN researches are able to communicate and collaborate with their counterparts abroad over the virtual work space at improve network performance and lower network latency [6].

The reason for the introduction of MYREN is that, it is hoped that the agent tool that is created for this project can be used within it. Although the agent tool is only tested within a Local Area Network, the installation and setup procedure remain the same if it was used on MYREN.

Thus this is a brief explanation of what MYREN is and how it is a crucial infrastructure for the implementation of grid computing.

2.4 Globus Toolkit

Globus Toolkit is an open source application which is a fundamental enabling technology for grid computing. The toolkit allow people to share computing power, databases and other tools securely online across corporate, institutional, and geographical boundaries without sacrificing local autonomy [7]. The toolkit includes software services and libraries for resource monitoring, discovery and management. Below is the basic overview of the latest Globus toolkit version which is version 4 (GT4).
The toolkit is package as a set of components that can be used either independently or together to develop application. The software is useful because every organization has its own unique modes of operation and collaboration between multiple organizations can be difficult because of incompatibility of resources such as data archive, computers and network. The Globus toolkit is created to remove this obstacle which will allow seamless collaboration. Its core services, interfaces and
protocols allow users to access remote resources as if they were located within their own machine room while simultaneously preserving local control over who can use resources and when.

This project has revolutionized the way science is conducted. Some example of a large scale e-science projects relying on the Globus Toolkit include the Network for Earthquake Engineering and Simulation (NEES), FusionGrid, the Earth System Grid (ESG), the NSF Middleware Initiative and its GRIDS Center, and the National Virtual Observatory. In addition, many universities have deployed campus Grids and deployments in industry are growing rapidly [7].

2.5 Cent OS 5.2

CentOS is a freely-available operating system that is based on Red Hat Enterprise Linux. This rebuild project strives to be 100% binary compatible with the upstream product and, within its mainline and updates, not to vary from that goal. Additional software archives hold later versions of such packages, along with other Free and Open Source Software RPM-based packages. CentOS stand for Community Enterprise Operating System.

Red Hat Enterprise Linux is largely composed of free and open source software, but is made available in a usable, binary form (such as on CD-ROM or DVD-ROM media or download) only to paying subscribers. As required, Red Hat releases all source code for the product publicly under the terms of the GNU General Public License and other licenses. CentOS developers use that source code to create a final product that is very similar to Red Hat Enterprise Linux; the logos must be changed, because Red Hat does not allow them to be used for redistribution. CentOS is freely available for download and use by the public, but is not maintained or supported by Red Hat. There are other distributions derived from RHEL's source as well, but they have not attained the surrounding community that CentOS has built; CentOS is generally the one most current with Red Hat's changes.
CentOS's preferred software updating tool is based on yum, although support for use of an up2date variant exists. Each may be used to download and install both additional packages and their dependencies, and also to obtain and apply periodic and special (security) updates from repositories on the CentOS Mirror Network.

CentOS refers to the source as "PNAELV" (Prominent North American Enterprise Linux Vendor), which means Red Hat, coined in response to questions raised by Red Hat's legal counsel in a letter to project members regarding possible trademark issues.

### 2.6 Red Hat Enterprise Linux

Red Hat Enterprise Linux is a Linux distribution produced by Red Hat and targeted toward the commercial market, including mainframes. Red Hat commits to supporting each version of Red Hat Enterprise Linux for 7 years after its release. All of Red Hat's official support, all of Red Hat's training and the Red Hat Certification Program centre is on the Red Hat Enterprise Linux platform. Red Hat Enterprise Linux is often abbreviated to RHEL, even if Red Hat is now attempting to discourage this.

New versions of Red Hat Enterprise Linux are released every 18 to 24 months. When Red Hat releases a new version of Red Hat Enterprise Linux, customers may upgrade to the new version at no additional charge as long as they are in possession of a current subscription (i.e. the subscription term has not yet lapsed).

Red Hat's first Enterprise offering (Red Hat Linux 6.2E) essentially consisted of a version of Red Hat Linux 6.2 with different support levels, and without separate engineering.

The first version of Red Hat Enterprise Linux to bear the name originally came onto the market as "Red Hat Linux Advanced Server". In 2003 Red Hat rebranded Red Hat Linux Advanced Server to "Red Hat Enterprise Linux AS", and
added two more variants, Red Hat Enterprise Linux ES and Red Hat Enterprise Linux WS.

Verbatim copying and redistribution of the entire Red Hat Enterprise Linux distribution is not permitted due to trademark restrictions. However, there are several redistributions of Red Hat Enterprise Linux minus trademarked features (such as logos and the name).

2.7 Visualization Pipeline for Medical Dataset

Scientific visualization is becoming increasingly important in analyzing and interpreting numerical and complex dataset. However, the dataset generated by medical detectors or simulations is growing in size and complexity. Using traditional desktop computers to interpret and analyze the dataset for this medical dataset is unreliable. This is because conventional computer will be overwhelmed with intensive processing of large dataset even with the latest development of visualization techniques. It is due to the limited memory and computational power available on the machines \cite{9}.

To overcome this problem Grid computing environment is needed for the application to run providing the necessary computational power required for visualization operations. Grid services could then be used to support remote visualization of large dataset of large dataset and to break the constraint of physical co-location of the resources by applying grid computing technology.

Based on this case study problem the creation of an agent tool is suggested so that the complexity of setting up a grid node will be reduced and the application will be able to perform hassle free.

The creation of the agent tool to setup the grid nodes in MYREN is in a way to compartmentalize the complexity is creating an application that is used to run in grid computing environment. By creating this agent tool the process of installing and
setting up the grid node will be handled so that other more important process can be
given more consideration.

This is one of the examples of grid application that is able to use the agent
tool created in this project. This show the importance of the agent tool for researches
and normal user which requires easy installation and setup procedure before using
their grid application.

### 2.8 Workflow Globus Installation and Setup

Figure 2.3 is a simplified workflow diagram of the process needed to run a
grid computing application on a machine. This is to show the complexity of the setup
process. The first process is to setup the master machine.

**Figure 2.3:** First machine setup workflow
Figure 2.3: First machine setup workflow (continued)

- Globus Installation - Building the Globus Toolkit
- Setting Up Security on First Machine
- Creating Proxy Server on the First Machine
- Setting up GridFTP
- Starting up Webservice Container
- Configuring RFT
- Setting up GRAM4
- Setting up Nodes
- Testing Grid Application
- END
After first machine installation and setup, node machine will then be installed and setup for connection to the first machine. Figure 2.4 is the workflow for installation and setup of nodes machine.

**Figure 2.4: Node machine setup workflow**
2.9 Globus Installation process in Linux Operating System

The process of installing Globus in Cent OS 5.2 is complicated for first time user to grid computing. Below are the commands that need to be performed just to install Globus toolkit. The reason that this installation process is in here is to show the complexity in installing the Globus toolkit and the automation program is worthwhile.
2.9.1 Pre-Installation

1. Java JDK 1.5.0_15

The first application that needs to be installed before installing Globus 4.2.1 is Java JDK 1.5.0. This application can be downloaded from http://java.sun.com.

```
# mkdir /usr/java
# cd /usr/java
(move the jdk-1_5_0_15-linux-i586.rpm to /usr/java)
# rpm -iv jdk-1_5_0_15-linux-i586.rpm
Preparing packages for installation...
jdk-1.5.0_15-fcs
# vi /etc/profile
(insert)
export JAVA_HOME=/usr/java/jdk1.5.0_15
export PATH=$PATH:$JAVA_HOME/bin

# source /etc/profile
# env
# export JAVA_HOME=/usr/java/jdk1.5.0_15
# export PATH=$PATH:$JAVA_HOME/bin
```

Add java to the system and change the default from Java JRE to Java JDK.

```
# /usr/sbin/alternatives --install /usr/bin/java java
/usr/java/jdk1.5.0_15/bin/java 2
# /usr/sbin/alternatives --config java

There are 2 programs which provide 'java'.
Selection Command
-----------------------------------------------
*+ 1 /usr/lib/jvm/jre-1.4.2-gcj/bin/java
  2 /usr/java/jdk1.5.0_15/bin/java

Enter to keep the current selection[+], or type selection number: 2
```
2. Apache-ant-1.7.1

The second application that needs to be installed is Apache Ant 1.7.1.

```
# wget http://apache.mirror.aussiehq.net.au/ant/binaries/apache-ant-1.7.1-bin.tar.gz
# tar -xzvf apache-ant-1.7.1-bin.tar.gz
# mv apache-ant-1.7.1 /usr/local/
# vi /etc/profile
(Added the followings)
export ANT_HOME=/usr/local/apache-ant-1.7.1
export PATH=$PATH:$ANT_HOME/bin
```

To test the installation is successful and the apache ant version, this procedure must take place.

```
# export ANT_HOME=/usr/local/apache-ant-1.7.1
# export PATH=$PATH:$ANT_HOME/bin
# source /etc/profile
# env | grep ant
...
...
# which ant
/usr/local/apache-ant-1.7.1/bin/ant
```

3. C, C++ Compiler

The third application that needs to be installed is new version C and C++ compiler. To do the following commands make sure that the Cent Os 5.2 and Red Hat Enterprise Linux is registered on RHN.
4. GNU tar

The fourth application that needs to be installed is GNU tar. This can be done by checking the version that your systems have.

```
# tar --version
.tar (GNU tar) 1.15.1
```

5. GNU Sed

The fifth application that needs to be installed is GNU Sed. This can be done by checking the version that your systems have.

```
# sed --version
GNU sed version 4.1.5
...
6. zlib

The sixth application that needs to be installed is zlib library. This can be done by checking the version that your systems have.

```
# yum list zlib
... 
Installed Packages
zlib.i386 1.2.3-3
installed
```

7. GNU make

The seventh application that needs to be installed is GNU make. This can be done by checking the version that your systems have.

```
# make --version
GNU Make 3.81
... 
```

8. sudo

The eighth application that needs to be installed is sudo. This can be done by checking the version that your systems have.

```
# sudo -V
Sudo version 1.6.8p12
... 
```
9. openssl

The ninth application that needs to be installed is GNU make.

```
# openssl version
OpenSSL 0.9.8b 04 May 2006
# yum list openssl-devel
... Available Packages
openssl-devel.i386 0.9.8b-10.el5
(base
(must install)
# yum install openssl-devel
... Installed: openssl-devel.i386 0:0.9.8b-10.el5
Dependency Installed:
e2fsprogs-devel.i386 0:1.39-15.el5
keyutils-libs-devel.i386 0:1.2-1.el5
krb5-devel.i386 0:1.6.1-25.el5
libselinux-devel.i386 0:1.33.4-5.el5
libsepol-devel.i386 0:1.15.2-1.el5
zlib-devel.i386 0:1.2.3-3
Complete!
# yum list openssl openssl-devel
... Installed Packages
openssl.i686 0.9.8b-10.el5
installed
openssl-devel.i386 0.9.8b-10.el5
installed
Available Packages
openssl.i386 0.9.8b-10.el5
(base
```

10. XML::Parser for GRAM

The tenth application that needs to be installed is XML Parser.

```
# yum list perl-xml-parser
... Available Packages
perl-XML-Parser.i386 2.34-6.1.2.2.1
(base
(must install)
# yum install perl-XML-Parser
... Installed: perl-XML-Parser.i386 0:2.34-6.1.2.2.1
Dependency Installed:
perl-Compress-Zlib.i386 0:1.42-1.fc6
perl-HTML-Parser.i386 0:3.55-1.fc6
perl-HTML-Tagset.noarch 0:3.10-2.1.1
perl-URI.noarch 0:1.35-3
perl-libwww-perl.noarch 0:5.805-1.1.1
```
The following Code is to check the installation and XML parser version.

```bash
# ldconfig -n /usr/local/lib (collect new entries)
# ldconfig (rebuild the library)
# ldconfig -p (lists all entries)
# ldconfig -p | grep expat
    libexpat.so.0 (libc6) => /lib/libexpat.so.0

# updatedb
# locate XML/Parser.pm
```

### 2.9.2 Globus 4.2.1 Installation

After the pre-installation, now the installation of Globus 4.2.1 can proceed.

```bash
# useradd globus
(globus user must perform some tasks later)
# useradd griduser

# mkdir /usr/local/globus
# chown globus:globus /usr/local/globus
# wget http://www-unix.globus.org/ftppub/gt4/4.2.1/installers/src/gt4.2.1-all-source-installer.tar.gz
# chown globus:globus gt4.2.1-all-source-installer.tar.gz
# mv gt4.2.1-all-source-installer.tar.gz /home/globus/
```
To continue with the installation. The user must be changed to globus before the setup is executed.

```bash
# su - globus
$ tar -xzvf gt4.2.1-all-source-installer.tar.gz
$ cd gt4.2.1-all-source-installer

(Make sure globus user has this in its env)
$ export JAVA_HOME=/usr/java/jdk1.5.0_15
$ export ANT_HOME=/usr/local/apache-ant-1.7.1
$ export GLOBUS_LOCATION=/usr/local/globus
$ export PATH=$ANT_HOME/bin:$JAVA_HOME/bin:$PATH

$ ./configure --prefix=/usr/local/globus
checking build system type... i686-pc-linux-gnu
checking for javac... /usr/java/jdk1.5.0_15/bin/javac
checking for ant... /usr/local/apache-ant-1.7.1/bin/ant
configure: creating ./config.status
config.status: creating Makefile

$ make
```
After letting the Globus toolkit compiled 1 to 1.5 hour the following instruction must be performed.

```bash
# su - globus
$ jobs
$ (nothing is running)
$ cd gt4.2.1-all-source-installer
$ tail build.log
Your build completed successfully. Please run make install.

(Again make sure this is in the env)
$ export GLOBUS_LOCATION=/usr/local/globus

$ make install

Creating client support scripts for GridFTP over ssh. This will allow
GridFTP clients from this installation to access sshftp:// urls.
You will still need to run the following command as 'root' to enable
this machine to *accept* sshftp connections. This will create the file /etc/grid-security/sshftp.

$GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp -server
If root access is not available, the option -nonroot may be added to enable connections as your user only. This will create the file
$HOME/.globus/sshftp.

$GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp -server -nonroot

... ... WARNING: It looks like /usr/local/globus/tmp/gram_job_state may not be on a local filesystem.
WARNING: The test for local file systems is not 100% reliable. Ignore the below if this is a false positive.
WARNING: The jobmanager requires state dir to be on a local filesystem
WARNING: Rerun the jobmanager setup script with the -state-dir=<state dir> option.
Done.
Reading gatekeeper configuration file...
Warning: Host cert file: /etc/grid-security/hostcert.pem not found. Re-run
    setup-globus-gram-job-manager after installing host cert file.
Creating system information...
Done
Done..Done
```
2.10 Globus 4.2.1 Setup process

After the installation process is completed. Now it will continue to the setup procedure of Globus 4.2.1.

2.10.1 Setting up security on your first machine

First Globus requires be authenticated and authorized. Certificates are used for this purpose. The Distinguished Name (DN) of a certificate will serve as an authenticated identity. That identity will then be authorized. In this simple tutorial, the authorization will happen in a file lookup.

```
root@elephant:~# export GLOBUS_LOCATION=/sandbox/globus/globus-4.2.1
root@elephant:~# source $GLOBUS_LOCATION/etc/globus-user-env.sh
root@elephant:~# cd ~/globus/gt4.2.1-all-source-installer
root@elephant:gt4.2.1-all-source-installer# perl gt-server-ca.pl -y
Setting up /sandbox/globus/globus-4.2.1/
Please enter a password of at least four characters for the CA:
Confirm password:
Creating a new simpleCA, logging to gt-server-ca.log...
Running setup-gsi...
Your CA hash is: 1bcdfe89
It is located at /sandbox/globus/globus-4.2.1/share/certificates/1bcdfe89.0
Your host DN is /O=Grid/OU=GlobusTest/OU=simpleCA-elephant.mcs.anl.gov/CN=host/elephant.mcs.anl.gov
The hostcert is located at /sandbox/globus/globus-4.2.1/etc/hostcert.pem
```
The last step is to copy that signed certificate into `/etc`.

```bash
root@elephant:~# mkdir /etc/grid-security
root@elephant:~# mv $GLOBUS_LOCATION/etc/host*.pem /etc/grid-security/
```

After that make the containercerts owned by Globus.

```bash
root@elephant:~# cd /etc/grid-security
root@elephant:/etc/grid-security# cp hostcert.pem containercert.pem
root@elephant:/etc/grid-security# cp hostkey.pem containerkey.pem
root@elephant:/etc/grid-security# chown globus:globus container*.pem
root@elephant:/etc/grid-security# ls -l *
```

```diff
-rw-r--r-- 1 globus globus 2724 2008-06-16 14:26 containercert.pem
-r-------- 1 globus globus  887 2008-06-16 14:26 containerkey.pem
-rw-r--r-- 1 root root 2724 2008-06-16 14:26 hostcert.pem
-rw-r--r-- 1 root root  887 2008-06-16 14:26 hostkey.pem
```

2.10.2 Creating Proxy server

After the security configuration, MyProxy server needs to be configured on elephant, following the instructions at configuring MyProxy. This will be used to store user's certificates.

```bash
root@elephant:~# export GLOBUS_LOCATION=/sandbox/globus/globus-4.2.1/
root@elephant:~# cp $GLOBUS_LOCATION/share/myproxy/myproxy-server.config /etc
root@elephant:~# vim /etc/myproxy-server.config
root@elephant:~# diff /etc/myproxy-server.config
```
SGLOBUS_LOCATION/share/myproxy/myproxy-server.config
15,21c15,21
< accepted_credentials "*"
< authorized_retrievers "*"
< default_retrievers "*"
< authorized_renewers "*"
< default_renewers "none"
< authorized_key_retrievers "*"
< default_key_retrievers "none"
---
> #accepted_credentials "*"
> #authorized_retrievers "*"
> #default_retrievers "*"
> #authorized_renewers "*"
> #default_renewers "none"
> #authorized_key_retrievers "*"
> #default_key_retrievers "none"

root@elephant:~# cat
SGLOBUS_LOCATION/share/myproxy/etc.services.modifications >> /etc/services
root@elephant:~# tail /etc/services
binkp 24554/tcp  # binkp fidonet protocol
asp 27374/tcp  # Address Search Protocol
asp 27374/udp
dircproxy 57000/tcp  # Detachable IRC Proxy
tfido 60177/tcp  # fidonet EMSI over telnet
fido 60179/tcp  # fidonet EMSI over TCP
# Local services
myproxy-server 7512/tcp  # Myproxy server

root@elephant:~# cp
SGLOBUS_LOCATION/share/myproxy/etc.xinetd.myproxy
/etc/xinetd.d/myproxy
root@elephant:~# vim /etc/xinetd.d/myproxy
root@elephant:~# cat /etc/xinetd.d/myproxy

service myproxy-server {
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    server = /sandbox/globus/globus-4.2.1/sbin/myproxy-server
    env = GLOBUS_LOCATION=/sandbox/globus/globus-4.2.1
    LD_LIBRARY_PATH=/sandbox/globus/globus-4.2.1/lib
    disable = no
}

root@elephant:~# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
root@elephant:~# netstat -an | grep 7512
tcp 0 0 0.0.0.0:7512 0.0.0.0:* LISTEN
2.10.3 Set up GridFTP

After MyProxy Configuration, the next process is to setup gridftp for globus.

```
root@elephant:/etc/grid-security# vim /etc/xinetd.d/gridftp
root@elephant:/etc/grid-security# cat /etc/xinetd.d/gridftp
service gsiftp
{
    instances               = 100
    socket_type             = stream
    wait                    = no
    user                    = root
    env                     +=
        GLOBUS_LOCATION=/sandbox/globus/globus-4.2.1
    env                     +=
        LD_LIBRARY_PATH=/sandbox/globus/globus-4.2.1/lib
    server                  = /sandbox/globus/globus-4.2.1/sbin/globus-gridftp-server
    server_args             = -i
    log_on_success          += DURATION
    disable                 = no
}
root@elephant:/etc/grid-security# vim /etc/services
root@elephant:/etc/grid-security# tail /etc/services
vboxd           20012/udp
binkp           24554/tcp                       # binkp fidonet
protocol
asp             27374/tcp                       # Address Search Protocol
asp             27374/udp
dircproxy       57000/tcp                       # Detachable IRC Proxy
tfido           60177/tcp                       # fidonet EMSI over telnet
fido            60179/tcp                       # fidonet EMSI over TCP
# Local services
myproxy-server  7512/tcp                        # Myproxy server
gsiftp          2811/tcp
root@elephant:/etc/grid-security# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
root@elephant:/etc/grid-security# netstat -an | grep 2811
tcp        0      0 0.0.0.0:2811            0.0.0.0:* LISTEN
```
Now the gridftp server is waiting for a request, client will be able to transfer a file

```
bacon@elephant $ myproxy-logon -s elephant
Enter MyProxy pass phrase: ********
A credential has been received for user bacon in /tmp/x509up_u1817.
bacon@elephant $ globus-url-copy
gsiftp://elephant.mcs.anl.gov/etc/group
file:///tmp/bacon.test.copy
bacon@elephant $ diff /tmp/bacon.test.copy /etc/group
```

2.10.4 Starting web service container

Now the setup for /etc/init.d entry for the webservices container can be performed. The details about the container can be found at Java WS Core Admin Guide.

```
root@elephant:~# cp $GLOBUS_LOCATION/etc/init.d/globus-ws-java-container /etc/init.d
```

```
globus@elephant:~$ /etc/init.d/globus-ws-java-container start
Starting Globus container. PID: 29985
```

```
bacon@elephant $ globus-check-remote-environment -s
https://localhost:8443
### Remote Endpoint Version Information ###
Axis Version on remote endpoint https://localhost:8443:
Apache Axis version: 1.4
Built on Mar 01, 2007 (10:42:15 CST)
Java WS Core Version on remote endpoint https://localhost:8443: 4.2.1
```
2.10.5 Configuring RFT

The use of globus-crft command is to start a reliable file transfer. It takes an input file whose syntax is one pair of URLs per line. It will use RFT to manage the transfer of all the URLs in the transfer file.

```
bacon@elephant $ cat transfer
gsiftp://elephant.mcs.anl.gov/etc/group
gsiftp://elephant.mcs.anl.gov/tmp/asdf
bacon@elephant $ globus-crft -ez -f transfer
Creating the RFT service.
Starting the RFT service.
Waiting for the RFT transfers to complete.
Transferred 1 of 1   | Status: Done

bacon@elephant $ diff /etc/group /tmp/asdf
bacon@elephant $
```

2.10.6 Setting up GRAM4

Now that GridFTP and RFT is working, GRAM can be setup for resource management.

```
root@elephant:~# visudo
root@elephant:~# cat /etc/sudoers
Runas_Alias GLOBUSUSERS = ALL, !root;
globus ALL=(GLOBUSUSERS) NOPASSWD: /sandbox/globus/globus-4.2.1/libexec/globus-gridmap-and-execute
-g /etc/grid-security/grid-mapfile /sandbox/globus/globus-4.2.1/libexec/globus-job-manager-script.pl *
globus ALL=(GLOBUSUSERS) NOPASSWD: /sandbox/globus/globus-4.2.1/libexec/globus-gridmap-and-execute
-g /etc/grid-security/grid-mapfile /sandbox/globus/globus-4.2.1/libexec/globus-gram-local-proxy-tool *
```
2.10.7 Setting up second machine: Security

After the first machine is configured, it will continue on the configuration on a second machine. This machine will become a node. The security setting for the second machine is as follows.

globus@cognito:~$ export GLOBUS_LOCATION=/usr/local/globus-4.2.1
globus@cognito:~$ scp -r elephant:/sandbox/globus/globus-
4.2.1/share/certificates $GLOBUS_LOCATION/share

root@elephant:~# myproxy-admin-addservice -c "cognito.mcs.anl.gov" -l cognito
2.10.8 Setting up second machine: GridFTP

GridFTP setup on the second machine is identical to the first. The list of command are as follows

```
root@cognito:/etc/grid-security# export GLOBUS_LOCATION=/usr/local/globus-4.2.1
root@cognito:/etc/grid-security# source $GLOBUS_LOCATION/globus-user-env.sh
root@cognito:/etc/grid-security# myproxy-retrieve -s elephant -k cognito.mcs.anl.gov -l cognito
Enter MyProxy pass phrase:******
Credentials for bacon have been stored in /etc/grid-security/hostcert.pem and /etc/grid-security/hostkey.pem.
rroot@cognito:/etc/grid-security# cd /etc/grid-security
root@cognito:/etc/grid-security# cp hostcert.pem containercert.pem
root@cognito:/etc/grid-security# cp hostkey.pem containerkey.pem
root@cognito:/etc/grid-security# chown globus:globus container*.pem
root@cognito:/etc/grid-security# ls -l *.pem
-rw------- 1 root root 912 2008-06-19 13:50 containercert.pem
-rw------- 1 root root 887 2008-06-19 13:50 containerkey.pem
-rw------- 1 root root 912 2008-06-19 13:45 hostcert.pem
-rw------- 1 root root 887 2008-06-19 13:45 hostkey.pem
root@cognito:/etc/grid-security# myproxy-destroy -s elephant -k cognito.mcs.anl.gov -l cognito
MyProxy credential 'cognito.mcs.anl.gov' for user cognito was successfully removed.

root@cognito:/etc/grid-security# vim grid-mapfile
root@cognito:/etc/grid-security# cat grid-mapfile
"/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.mcs.anl.gov/CN=Charles Bacon" bacon
```

Reloading internet superserver configuration: xinetd.
The GridFTP will then be tested. The command is as follows

```
cognito % setenv GLOBUS_LOCATION /usr/local/globus-4.2.1
cognito % source $GLOBUS_LOCATION/etc/globus-user-env.csh
cognito % myproxy-logon -s elephant
Enter MyProxy pass phrase: *******
A credential has been received for user bacon in /tmp/x509up_u1817.
cognito % globus-url-copy gsiftp://cognito.mcs.anl.gov/etc/group \
  gsiftp://elephant.mcs.anl.gov/tmp/from-cognito
```

2.10.9 Setting up second machine: Webservices

Setting up the container on the second machine is a lot like the first. The list of command is as follows.

```
root@cognito:~# cp $GLOBUS_LOCATION/etc/init.d/globus-ws-java-container /etc/init.d

globus@cognito:~ $ /etc/init.d/globus-ws-java-container start
Starting Globus container. PID: 19745
```

2.10.10 Setting up your second machine: GRAM4

The last step is to setup the GRAM service for Globus on the second machine. The command is as follows.
cognito % vim a.rsl
cognito % cat a.rsl
cognito % cat a.rsl

<job>
  <executable>my_echo</executable>
  <directory>${GLOBUS_USER_HOME}</directory>
  <argument>Hello</argument>
  <argument>World!</argument>
  <stdout>${GLOBUS_USER_HOME}/stdout</stdout>
  <stderr>${GLOBUS_USER_HOME}/stderr</stderr>
  <fileStageIn>
    <transfer>
      <sourceUrl>gsiftp://cognito.mcs.anl.gov:2811/bin/echo</sourceUrl>
      <destinationUrl>file:///${GLOBUS_USER_HOME}/my_echo</destinationUrl>
    </transfer>
    </fileStageIn>
    <fileCleanUp>
      <deletion>
        <file>file:///${GLOBUS_USER_HOME}/my_echo</file>
      </deletion>
      </fileCleanUp>
    </job>

cognito % globusrun-ws -submit -S -f a.rsl
Delegating user credentials...Done.
Submitting job...Done.
Job ID: uuid:1223d7e6-3e35-11dd-a209-003048241085
Termination time: 06/19/3008 19:22 GMT
Current job state: StageIn
Current job state: Active
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
Cleaning up any delegated credentials...Done.
cognito % cat ~/stdout
Hello World!
cognito % ls ~/my_echo
ls: /home/bacon/my_echo: No such file or directory
2.11 Summary

In this chapter, discussion was made on grid computing and the possible implementation of this agent tool when it’s completed. To show the complexity of the procedure of installing and setting up process, the full documentation were presented in this chapter so that the comparison could be made. By going through the process, user could see the appeal of the agent tool where it simplifies the complexity of installing and setting up Globus 4.2.1.
CHAPTER 3

METHODOLOGY

3.1 Introduction

Methodology is the combination of paradigm, method, policy, procedure, rules, standard, technique, equipment, and programming languages that are used to analyze the needs and the system design that is under development. The suitable method has to be chosen as it plays an important role in the beginning of the project because this will make the development of the project will proceed smoothly. Besides that, with systematic and with the correct methodology, it will produce a software that has the properties of being easy to maintain, less dependency and easy to use.

In the development of a system, there are a few methodology than can be use or to choose from. There are Prototype Model, Spiral Model, Waterfall Model, Evolutionary Development model, Systems Development Life Cycle (SDLC), Top-down and bottom-up design and others more. To make a decision in choosing a methodology, it should be suitable to the system that is being developed. The reason is every methodology implements different method, approach and technique that differs from each other.
3.2 Project Methodology

As a whole, every system software development need to go through phases that is known as the Systems Development Life Cycle (SDLC) and must be build accordingly to a fix development methodology. There are a few methodologies that have been mention that can be used if suitable but as a conclusion Evolutionary Prototyping model is the most suitable model for this project to run.

3.3 Justification in Choosing Evolutionary Prototyping Model

The methodology, Evolutionary Prototyping Model is chosen to develop this system because the phases of development in this methodology are suitable with the system development that takes place. These are the reason why the Evolutionary Prototyping Model is chosen and its advantages:

1. The development phase with the approach of the Evolutionary Prototyping methodology can be repeated until the prototype can be build to fulfill the specification of the user’s needs. The repetition of this phase is important because they will make sure that the system will deliver only what the user needs.
2. The approach of this methodology is suitable if the specification needs that is acquire from the beginning of the development is not complete. For an example, the users maybe did not know what the best interface is for themselves. The modification of this system can be done immediately before the system is really implemented.
3. The time for development require is short, because every irregularities from the specification needs can be detected immediately and repair. This characteristic is important because time that is given for this system development is short and limited.
4. By using this methodology it gives the advantage to the development that is not clear with the technical solution that is used. For example, when in the development process cannot identify the suitable algorithm or technique that can be use for the start. With this approach, the technical solution can be flexible and amendable.

3.4 System Development Methodology

In the Evolution prototype approach, the specification of system framework is built according to the user need of the system. Based on the system framework that is build, a version of it will be planed and build. This version is called prototype. Then, these prototypes will be used in testing and for evaluation. After that these prototype will be revise again and again to meet the users’ specification. Then at the end these prototype that is approve will be expand further to be a perfect system. The phases that is involve in the development of this system with the approach that is used by the Evolutionary prototype is the planning phase, Analyzing phase, Design Phase, Development phase, and the evaluation and testing phase. The Figure 3.1 below shows the model for Evolutionary Prototyping Model.
Figure 3.1: Process flow of an application design in evolutionary prototyping.
3.4.1 Planning Stage

The planning phase is the first phase to follow in order to build this system. Also, a few activities have been done to determine the objectives and set the goal of the system, the project scope, the system resource that is the hardware and the software that is used, and the work time planning table. The objectives and the goal of the system and the scope that is identified and are explained in detailed in Chapter 1. Then, the work planning that involves separation and compilation of tasks in the development of the system is also planned. Please refer the work time planning table in the form of Gantt chart that attach in the Attachment B for more detailed explanation.

3.4.2 Analyzing Phase

The purpose of this phase is to investigate the system needs. In this phase, a deeper research on the problem face on the existing system, the users requirement and the system requirement is done using the information that is acquire in the Planning phase. Next, the result of the analysis on the requirements is recorded and is translate into a model that is the system frameworks that will be explain in chapter 4. Besides that, the hardware and software that is use in the development of this system will also be analyzed. Characteristic and the compatibility of the software and hardware used is also being research. The justification of choosing the hardware and software is being explain in section 3.5.1 and 3.5.2
3.4.3 Design phase

In the design phase, it will explain how the system will solve the problem that is identified. The purpose of the activities in this phase is to translate the functions in the specification requirements to system modules.

The design of the system is based on the specification requirements that are identified in the Analyzing phase. The aspects in the design phase are the system architecture design, module design and user interface design. This design is build by using the Unified Modelling Language (UML) and is represent by use case model, and sequence diagram. The system architecture design will show the process for the whole system and the module design will explain the separate modules and the function of it. Last but not lease, the user interface design where it specify the screen of the mobile phone device and information that needs to be shown to the user. There will be more detailed explanation on the system design in Chapter 4.

3.4.4 Development phase

The development phase will change or turn the detailed design into programming codes. Prototypes will build based on the design that is produced in the previous phases. These prototypes will then undergo a further expansion to produce a genuine system.

First, devices and software that is needed in the project is assembled and reconfigure. Then the prototype will be created by coding with the program designing and logic functions that is based on the specification requirements.
3.4.5 Testing and Evaluation Phase

In this phase, the program that is produce is checked, tested and approve with the use of demo versions in real-time and then compare with results that shown in the simulations. If the real-time demo does not run as what in the simulation does, modification will be made on the prototype.

Testing and evaluation may not only apply on the program but also applies to the specification requirements and system design as well. After testing is made, if the prototype is not up to the specification requirements standards the prototype can be fixed or upgraded. The development prototype phase will repeat until it fulfills the specification requirement of the system. Last, when the prototype meets the specification requirements it will be used to develop a perfect system.

3.5 System requirements

To develop this system hardware and software that are relevant to the system requirements is used. Hardware and software that is needed had been identified during the planning phase and is analyzed as system requirements. The hardware and software that is used for development will be discussed in the following section:

3.5.1 Hardware Requirements

The hardware that is used to develop this system is:

1. Intel Pentium IV 1.5GHz Processor and above
2. 512 MB DDR2 RAM and above
3. 60 GB of hard disk free space
4. Keyboard and mouse.

The hardware that is mentioned above is used because of the compatibility in building the system. Personal Computer with the specification that is mentioned can run or support the development of this system. With the high processor and bigger RAM the simulation of the system can run smoothly.

### 3.5.2 Software Requirements.

The software that is used to develop this system is:

1. Cent OS 5.2 Linux operating system
   a. Cent OS 5.2 is used to develop this system because of its inherited stability and flexibility.
2. Red Hat Enterprise Linux
   a. Another Linux operating system is used to make the agent tool more universal.
3. Globus Toolkit version 4 (GT4)

### 3.6 Summary

As a conclusion for this chapter the methodology that is used for this system is the Evolutionary prototype methodology because of its flexibility in handling the project. Then the specification of the system is also being verified and discussed here.
4.1 Introduction

For the early stage of application development, the specification requirements of the application is being identify and it is important to use as a reference in building the application. The functions of the application are modelled in the System design phase. The design of this system helps to identify the functions that is developed throughout the system development timeframe, fulfils the user’s requirements. Designs that are produced will be used as reference for further development of the system. Some of the designs that are use as a reference are the basic architecture design and system architecture design. In this chapter also the agent tool interface design is introduced. The interface design in this chapter will explain the functionality of the option within the interface and what the result of this option once executed.
4.2 System Architecture Design

**Figure 4.1:** Basic Architecture Design

Master or First machines that manage resources to solve a common task within the grid.

**Figure 4.2:** System Architecture Design

Node Machine that are connected to the Local Area Network as a resource to the grid.
4.2.1 Explanation of system architecture design

From Figure 4.1 it shows how the system architecture design looks like in a basic overview. The agent tool is an application that will install appropriate application and connectivity setting on a master computer that will handle the management of resources from client computers that is connected to it. Once a master computer is set up, then client computer can be assign and appropriate application and setting can be installed on these other computer.

Once the agent tool is set up on all the resources or computer, the master computer could then allocate adequate resources for program. A scheduling service will be used to decide what job runs where and when.

4.3 General design

![Diagram]

**Figure 4.3:** Grid Architecture
Creating a complete grid system requires a wide variety of protocols services and software development kits. There are different components that make up a grid system and are categorized according to their function and purpose. The agent tool proposed will allow this architecture to function appropriately so that application could then be implemented within the grid created.

The grid architecture can be expressed in terms of a layered diagram shown in Figure 4.3. The description of each layer from the bottom up will be explained below.

**4.3.1 Fabric**

This layer refers to the resources which are actually going to be shared in the grid system, such as individual computers, clusters, supercomputers, network storage, database, etc.

**4.3.2 Connectivity**

The connectivity layer refers to all the protocols that will allow the resources to communicate. Fundamental internet protocol such as TCP/IP, HTTP, DNS, etc. fall into this layer. Grid system also need secure communication, so this layer also includes protocols that will response to the security challenges faced in a grid system.
4.3.3 Resources

This layer refers to all the services and protocols that enable the management of individual resources. This management might include tasks such as initiating resources, monitoring them and accounting them. At this layer it will not concern with the global interactions between resources in the grid, only with individual resources.

4.3.4 Collective

This layer encompasses the services and protocols that deal with managing multiple resources. Building on the services provided in the Resource layer, this layer will allow a bunch of resources to work together to solve a common task. The following is a sample of the type of services commonly found in this layer.

1. Resource registries: To allow discoveries of resources in a virtual organization and to query their properties.
2. Allocation and scheduling services: When a program to run on a grid system, the user don’t have to decide where exactly on the grid it will run. An allocation service will use the resource directory to discover a resource adequate for the program and will allocate that resource for the job.
3. Monitoring service: Allow the user to monitor all the resources are working properly.
4.3.5 Application

This layer refers to the actual application that will be running on a grid system. Notice how these applications are not required to interact directly with the collective services. They are also free to access the Resources and Connectivity services directly whenever required to.

4.4 User Interface design

User interface design is an important part of the agent tool created on this program. The reason is that, the interface will make the installation and setup process much easier and the user could choose which functionality that they want. Figure 4.4 is the interface design for the agent tool.
From Figure 4.4, there are five functions to choose from the agent tool. To choose this function the user need to use the UP/DOWN arrow to select the functionality and press ENTER to select the process. The reason for the simplicity of the interface is so that no additional applications need to be installed just for the interface. By using this interface any computer and Linux operating system have the library needed for the interface and could use it without having any problem.

The user could also use the numbering that is assigned to each function as a shortcut without using the arrow button. This will make it easier to use the agent tool. From Figure 4.4 there are five functions that can be considered from. This functionality will be explained as follows.
1. Install Globus 4.2.1
   a. This function is design to install the necessary pre requisite application for Globus and installing Globus 4.2.1 when the pre requisite is meet.
   b. The function must be used on the first / master machine and also the node machine.

2. Master machine setup
   a. This functionality is design to provide the procedure to setup a master machine before connecting it to a node.
   b. This function must be use on the first / master machine only.

3. Node machine setup
   a. This functionality is used to provide the necessary procedure to setup a node one a master machine is assign. After a master machine is assign the procedure could be run on the node machine.
   b. This function must be used on the node machine after the setup of the first machine.

4. Add node
   a. This function is used on the master machine every time a node is added to the grid.
   b. This function must only be used on the master machine.

5. Exit Program
   a. This function is used when the user wanted to exit the program.

By using this interface user who doesn’t have much experience on using Linux operating system could easily use it without the need to remember the appropriate command to use it.
4.5 Conclusion

In conclusion, the architecture design of the agent tool is discussed in this chapter. Besides that the basic and system architecture design and its explanation is properly stated in this chapter. Other than that, a basic view of the agent tool is displayed and its functionality is described as a rough idea how the application looks like. These are the topic that is already been discussed within this chapter.
CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

In this chapter, the implementation of the system will be discussed. In this phase the requirements and environment that needs to be fulfilled for the project system is realized. These include installation of the required software, and testing are been made also to ensure that this project do not fail. Any error or bugs that are found in the program will be debugged to ensure the system does work.

5.2 Development Environment

In this section, the installation of the required software is discussed.
5.2.1 Installation of Required Software

In developing the agent tool for installing and setting up the grid, shell script is used. The reason for choosing this development software is that in every Linux operating system this development software is already preinstalled and its is compatible with the operating system itself. So there is no need to install any foreign software in developing the agent tool.

A shell script is a script written for the shell, or command line interpreter, of an operating system. It is often considered a simple domain-specific programming language. Typical operations performed by shell scripts include file manipulation, program execution, and printing text.

For Cent OS 5.2, The Bourne shell, or sh, was used. The reason for using this shell script is that it’s a popular default shell for Unix accounts. The binary program of the Bourne shell or a compatible program is located at `/bin/sh` on most Unix systems, and is still the default shell for the root superuser on many current Unix implementations.

5.3 Coding

In this section some of the important coding of shell script will be displayed to show the process of installing and setup for grid computing environment. It will begin with installation and setup for the first machine codes then continue with the process of installation and setup for node machine.
5.3.1 Installation codes for Globus Toolkit 4.2.1

Below is the coding which is written in shell script in installing Globus toolkit 4.2.1 in Cent OS 5.2. This coding is design to check the system whether they have the required pre-installation software before installing Globus toolkit. All the required application is kept in rpm so it will be easy for the program to install the application in Cent OS. Figure 5.1 and Figure 5.2 is coding for installation process of the Globus toolkit and it’s prerequisite. The reason for the 2 module is so that this application can be run on separate user.

```bash
#!/bin/sh

#This script is to install Globus and it's dependencies
#by: Abdul Rasheed Khan
#filename: intgt01

exporting_path() {
    export JAVA_HOME=/usr/java/jdk1.5.0_15
    export ANT_HOME=/usr/local/apache-ant-1.7.1
    export GLOBUS_LOCATION=/usr/local/globus
    export PATH=$ANT_HOME/bin:$JAVA_HOME/bin:$PATH
}
```

**Figure 5.1:** Snippets coding for installing prerequisite software before installing Globus toolkit 4.2.1
#creating java jdk folder
if [ -d /usr/java ]
then
    echo "Directory /usr/java/ exists"
else
    mkdir /usr/java/
fi

#creating globus folder
useradd globus
if [ -d /usr/local/globus ]
then
    echo "Directory /usr/local/globus exists"
else
    mkdir /usr/local/globus
fi

chown globus:globus /usr/local/globus

#Creating griduser
useradd griduser

#moving installation file
if [ -f /usr/local/apache-ant-1.7.1-bin.tar.gz ]
then
    echo "Apache-ant installer exists"
else
    cp apache-ant-1.7.1-bin.tar.gz /usr/local/
fi

if [ -f /usr/java/jdk-1.5.0_15-fcs.i586.rpm ]
then
    echo "Java JDK installer exists"
else
    cp jdk-1.5.0_15-fcs.i586.rpm /usr/java/
fi

if [ -f /root/perl-XML-Parser-2.34-6.1.2.2.1.i386.rpm ]
then
    echo "XML-Parser installer exists"
else
    cp perl-XML-Parser-2.34-6.1.2.2.1.i386.rpm /root/
fi

if [ -f /home/globus/gt4.2.1-all-source-installer.tar.gz ]
then
    echo "Globus installer exists"
else
    cp gt4.2.1-all-source-installer.tar.gz /home/globus/
fi

**Figure 5.1**: Snippets coding for installing prerequisite software before installing
Globus toolkit 4.2.1 (Continued)
# Moving Scripts

cp intgt02 /usr/local/bin/ -f

# Installing Java JDK

echo "## Installing Java JDK 1.5.0_15 ##"
cd /usr/java
rpm -ivh jdk-1.5.0_15-fcs.i586.rpm
rm jdk-1.5.0_15-fcs.i586.rpm

JAVA_HOME=/usr/java/jdk1.5.0_15
export JAVA_HOME
PATH=$PATH:$JAVA_HOME/bin

export JAVA_HOME=/usr/java/jdk1.5.0_15
export PATH=$PATH:$JAVA_HOME/bin

/usr/sbin/alternatives --install /usr/bin/java java
/usr/java/jdk1.5.0_15/bin/java 2
/usr/sbin/alternatives --config java

which java
ls -l /usr/bin/java
ls -l /etc/alternatives/java

# Installing Apache-ant

echo "## Installing Apache-ant 1.7.1 ##"

cd /usr/local

if [ -d /usr/local/apache-ant-1.7.1 ]
then
echo "Apache-ant 1.7.1 exists"
else
tar xzf apache-ant-1.7.1-bin.tar.gz
fi

export ANT_HOME=/usr/local/apache-ant-1.7.1
export PATH=$PATH:$ANT_HOME/bin

rm apache-ant-1.7.1-bin.tar.gz
which ant

exporting_path

# Installing XML-Parser

echo "## Installing XML-Parser ##"

cd /root
rpm -ivh perl-XML-Parser-2.34-6.1.2.2.1.i386.rpm
rm perl-XML-Parser-2.34-6.1.2.2.1.i386.rpm

---

**Figure 5.1:** Snippets coding for installing prerequisite software before installing Globus toolkit 4.2.1 (Continued)
Figure 5.1: Snippets coding for installing prerequisite software before installing Globus toolkit 4.2.1 (Continued)

```bash
#!/bin/sh

#This Shell script is to install Globus in globus folder
#by: Abdul Rasheed Khan
#filename: intgt02

if [ -d /home/globus/gt4.2.1-all-source-installer ]
then
    echo "Globus installer folder exists"
else
    echo "Globus installer folder exists"
    tar xzf gt4.2.1-all-source-installer.tar.gz
fi

cd gt4.2.1-all-source-installer

export JAVA_HOME=/usr/java/jdk1.5.0_15
export ANT_HOME=/usr/local/apache-ant-1.7.1
export GLOBUS_LOCATION=/usr/local/globus
export PATH=$ANT_HOME/bin:$JAVA_HOME/bin:$PATH

./configure --prefix=/usr/local/globus
make
make install

exit 0
```

Figure 5.2: Snippets coding for compiling and installation of Globus toolkit
5.3.2 Settings in the First machine

The next coding is the most important part of the application which is the security setting for the first machine and its nodes in the grid. Figure 5.3 shows the snippets of the coding for setting up the security in first machine.

```
#!/bin/sh

#This Shell script is to Setup Security for Globus
#by: Abdul Rasheed Khan
#filename: sec

export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

cp myproxy /usr/local/bin/ -f
cp fmgftp /usr/local/bin/ -f
cp fmplogin /usr/local/bin/ -f
cp fmwsc /usr/local/bin/ -f
cp fmcstart /usr/local/bin/ -f
cp cuinter /usr/local/bin/ -f
cp rft /usr/local/bin/ -f
cp gram /usr/local/bin/ -f

cd ~globus/gt4.2.1-all-source-installer
perl gt-server-ca.pl -y
mkdir /etc/grid-security
mv $GLOBUS_LOCATION/etc/host*.pem /etc/grid-security/

cd /etc/grid-security
cp hostcert.pem containercert.pem
cp hostkey.pem containerkey.pem
chown globus:globus container*.pem
ls -l *.pem
#myproxy setup

cd ~
myproxy
exit 0
```

Figure 5.3: Snippets of configuring the security for first machine
After security setting, it will continue with the Myproxy setting. This will be used to store our user's certificates. Recall that so far we have made a host certificate, but we don't have any certificates for end users yet.

```
#!/bin/sh

#This Shell script is to create proxy server
#by: Abdul Rasheed Khan
#filename: myproxy

cd
export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

cp /root/GT4/myproxy-server.config /etc
diff /etc/myproxy-server.config
$GLOBUS_LOCATION/share/myproxy/myproxy-server.config
cat
$GLOBUS_LOCATION/share/myproxy/etc.services.modifications >>
/etc/services
tail /etc/services

cd /etc/xinetd.d/
proxyser='
service myproxy-server

{:
  socket_type = stream
  protocol = tcp
  wait = no
  user = root
  server = /usr/local/globus/sbin/myproxy-server
  env = GLOBUS_LOCATION=/usr/local/globus
  LD_LIBRARY_PATH=/usr/local/globus/lib
  disable = no
}
'
echo -e $proxyser >myproxy
cat myproxy

cd
/etc/init.d/xinetd reload
sleep 5
netstat -an | grep 7512

echo 'My proxy Setting Done'
```

Figure 5.4: Snippets in setting up proxy server on the first machine
After proxy server is setup upped on the first machine it will continue with the setting for grid file transfer protocols (gridftp). Figure 5.5 is a part of the coding for gridftp settings.

```
#!/bin/sh
#This Shell script is to create gridftp
#by: Abdul Rasheed Khan
#filename: gftp

cd
export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh
```

**Figure 5.5:** Snippets in setting up grid ftp on the first machine
Figure 5.5: Snippets in setting up grid ftp on the first machine (Continued)
Last coding snippets are the settings for web service container, RFT and GRAM4 for the first machine. Figure 5.6 is the part of snippets for setting these services.
Figure 5.6: Snippets in setting up Web service container, RFT and GRAM4 (Continued)
#!/bin/sh
#This Shell script is to sample clients/services to interaction with the container
#by: Abdul Rasheed Khan
#filename: cuinter

cd
export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

globus-check-remote-environment -s https://localhost:8443
sleep 5
exit 0

#!/bin/sh

#This Shell script is to configure RFT
#by: Abdul Rasheed Khan
#filename: rft

echo 'Testing RFT'
cd
export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

trans='gsiftp://hostname.fsksm.utm.my/etc/group
gsiftp://hostname.fsksm.utm.my/tmp/asdf'
echo -e $trans >transfer
cat transfer
echo 'Enter hostname:'
read thost
sed -e "s/hostname/\$hostname/g" transfer >transfer2
mv transfer2 transfer -f
cat transfer
globus-crft -ez -f transfer
diff /etc/group /tmp/asdf
exit 0

Figure 5.6: Snippets in setting up Web service container, RFT and GRAM4 (Continued)
5.3.3 Settings in the Node machine

For the node machine the setup process is different compared to the first machine. This is because; the node will need to link with the first machine for it required setting. In Figure 5.6 is the security setting for the node machine.

```bash
#!/bin/sh

# This Shell script is to run job
# by: Abdul Rasheed Khan
# filename: gram

cd
export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

globusrun-ws -submit -c /bin/true
echo $?
sleep 3

globusrun-ws -submit -c /bin/false
echo $?
sleep 3

exit 0
```

Figure 5.6: Snippets in setting up Web service container, RFT and GRAM4

(Continued)

```bash
#!/bin/sh

# This Shell script is to setup security in Node Machine
# by: Abdul Rasheed Khan
# filename: nsec

Globus toolkit Node Installation
```

Figure 5.7: Snippets in setting up security in node machine
cd  
export GLOBUS_LOCATION=/usr/local/globus  
source $GLOBUS_LOCATION/etc/globus-user-env.sh

# moving script  
cd /root/GT4/  
cp ngftp /usr/local/bin/ -f  
cp nplogin /usr/local/bin/ -f  
cp nwsc /usr/local/bin/ -f  
cp ncstart /usr/local/bin/ -f  
cp sjob /usr/local/bin/ -f

echo 'Enter Master Machine Hostname:'  
read mmhost

cscp -r $mmhost:/usr/local/globus/share/certificates  
$GLOBUS_LOCATION/share

echo 'Enter Hostname:'  
read nhost

myproxy-retrieve -s $mmhost -k $nhost.fsksm.utm.my -l $nhost  
cd /etc/grid-security  
cp hostcert.pem containercert.pem  
cp hostkey.pem containerkey.pem  
chown globus:globus container*.pem  
ls -l *.pem

myproxy-destroy -s $mmhost -k $nhost.fsksm.utm.my -l $nhost  
mapfile='"/O=Grid/OU=GlobusTest/OU=simpleCA-hostname.fsksm.utm.my/OU=fsksm.utm.my/CN=griduser" griduser'  
echo -e $mapfile >grid-mapfile  
se -e "s/hostname/$mmhost/g" grid-mapfile >grid-mapfile2  
mv grid-mapfile2 grid-mapfile -f  
cat grid-mapfile  

cd ~  
ngftp  

exit 0

Figure 5.7: Snippets in setting up security in node machine (Continued)
After the setting for security, it will continue with the gridftp setting for the node machine. Figure 5.8 will show the coding for setting up gridftp in the node machine.

```
#!/bin/sh

#Globus toolkit Node Installation
#This Shell script is to setup gridftp in Node Machine
#by: Abdul Rasheed Khan
#filename: gridftp

cd
export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

cd /etc/xinetd.d/
ftpfile='

{instances               = 100
 socket_type             = stream
 wait                    = no
 user                    = root
 nenv                       +=
 GLOBUS_LOCATION=/usr/local/globus
 nenv                       +=
 LD_LIBRARY_PATH=/usr/local/globus/lib
 nserver                   = /usr/local/globus/sbin/globus-gridftp-server
 nserver_args             = -i
 nlog_on_success          += DURATION
 ndisable                 = no
 nenv                       +=
 GLOBUS_TCP_PORT_RANGE=40000,41000
}
'

echo -e $ftpfile >gridftp

cat gridftp

cd /etc/

sed '$a\gsiftp        2811/tcp' services > new_services

mv new_services services -f

tail /etc/services
```

Figure 5.8: Snippets for Gridftp on node machine
Last coding snippets are the settings for web service container, RFT and GRAM4 for the node machine. Figure 5.9 is the part of snippets for setting these services.
#!/bin/sh

# This Shell script is to Setup Webservice Container
# by: Abdul Rasheed Khan
# filename: wsc

cd
export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

cp $GLOBUS_LOCATION/etc/init.d/globus-ws-java-container /etc/init.d
sleep 5
su - globus -c ncstart

# copying sudoers file to /etc/
# changing the sudoers file

cd

# copy /root/GT4/sudoers to /etc/ -f

# submit a staging job.
# This job will copy the /bin/echo command from cognito to a file called $HOME/my_echo.
# Then it runs it with some arguments, and captures the stderr/stdout.
# Finally, it will clean up the my_echo file when execution is done.

su - griduser -c sjob

# /etc/init.d/globus-ws-java-container start

exit 0

#!/bin/sh

# This Shell script is to start container
# by: Abdul Rasheed Khan
# filename: cstart

export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

/etc/init.d/globus-ws-java-container start
/etc/init.d/globus-ws-java-container restart

---

**Figure 5.9:** Snippets for Web service Container and GRAM RSL for node machine

(Cotinued)
#!/bin/sh

# This Shell script is submitting job
# by: Abdul Rasheed Khan
# filename: sjob

cd
export GLOBUS_LOCATION=/usr/local/globus
source $GLOBUS_LOCATION/etc/globus-user-env.sh

cd
jobfile='
<job>
  <executable>my_echo</executable>
  <directory>${GLOBUS_USER_HOME}</directory>
  <argument>Hello</argument>
  <argument>World!</argument>
  <stdout>${GLOBUS_USER_HOME}/stdout</stdout>
  <stderr>${GLOBUS_USER_HOME}/stderr</stderr>
  <fileStageIn>
    <transfer>
      <sourceUrl>gsiftp://jati.fsksm.utm.my:2811/bin/echo</sourceUrl>
      <destinationUrl>file:///gLOBUS_USER_HOME/my_echo</destinationUrl>
    </transfer>
    <fileCleanUp>
      <deletion>
        <file>file:///gLOBUS_USER_HOME/my_echo</file>
      </deletion>
    </fileCleanUp>
  </fileStageIn>

  echo -e $jobfile >a.rsl
  cat a.rsl

  sleep 5

  globusrun-ws -submit -S -f a.rsl
  cat ~/stdout
  ls ~/my_echo

  exit 0

Figure 5.9: Snippets for Web service Container and GRAM RSL for node machine
(Cotinued)
5.4 System Testing

To ensure that the system is working accordingly to the setting required for installation and setup process for Globus 4.2.1, within the agent tool there is already a testing process that is taken place.

5.4.1 Installation Testing

The first testing process is the installation of pre-requisite application that, the agent tool will display as shown in Figure 5.10 when the pre-requisite application is installed.

```
##Installing Java JDK 1.5.0 15#
Preparing...                                       [100%]
1:jdk                                             [100%]

There are 2 programs which provide `java`

Selection Command

* 1 /usr/lib/jvm/jre-1.4.2-gcj/bin/java
  + 2 /usr/java/jdk1.5.0_15/bin/java

Enter to keep the current selection[+], or type selection number: 2
/usr/java/jdk1.5.0_15/bin/java
-rwxrwxrwx 1 root root 22 Jan 18 17:24 /usr/bin/java -> /etc/alternatives/java
-rwxrwxrwx 1 root root 30 Apr 24 04:56 /etc/alternatives/java -> /usr/java/jdk1.5.0_15/bin/java
##Installing Apache-ant 1.7.1#
/usr/local/apache-ant-1.7.1/bin/ant
##Installing XML-Parser##
warning: perl-XML-Parser-2.34-6.1.2.1.1.1386.rpm: Header V3 DSA signature: MD5KEY, key ID 4df2a6fd2
Preparing...                                       [100%]
1:perl-XML-Parser                                   [100%]
```

Figure 5.10: Pre-requisite Installation testing and display

From Figure 5.10, it shows that the necessary application is installed according to the Globus 4.2.1 setting requirement.
After the pre-requisite installation is completed, Globus 4.2.1 is now ready to be installed. Figure 5.11 will be displayed to show that the entire pre installation configuration is within its parameter and there is no warning from the Globus Installation pack.

Figure 5.11: Checking the system before installation of Globus 4.2.1

Once the installation of Globus 4.2.1 is completed, Figure 5.12 will be displayed to show that the installation is successfully installed in the machine.

Figure 5.12: Installation completion of Globus 4.2.1
5.4.2 Master Machine Setup Testing

For master machine setup test, the first thing to test is the security certificate that setup security process created. Figure 5.13 show the necessary file for the security for a master node.

```
Running setup-gsi...
Your CA hash is: 4B5c15c0
It is located at /usr/local/globus/share/certificates/4B5c15c0.pem
Your host DN is /O-Grid/OU=GlobusTest/OU=simpleCA-cengal.fskm.utm.my/CN=host/cengal.fskm.utm.my
The hostcert is located at /usr/local/globus/etc/hostcert.pem
-rw-r--r-- 1 globus globus 2719 Apr 24 04:54 containercert.pem
-r-------- 1 globus globus 887 Apr 24 04:54 containerkey.pem
-rw-r--r-- 1 root root 2719 Apr 24 04:54 hostcert.pem
-rw-r--r-- 1 root root 1398 Apr 24 04:54 hostcert_request.pem
-r-------- 1 root root 887 Apr 24 04:54 hostkey.pem
```

**Figure 5.13:** Master machine security files

After the security process is setup, the process will continue with the testing of myproxy settings. The testing in Figure 5.14 will show that the configuration is done correctly and the server is listening from other nodes in the grid.

```
# Local services
myproxy-server 7512/tcp
service myproxy-server
{
    socket_type = stream
    protocol = TCP
    wait = no
    user = root
    server = /usr/local/globus/sbin/myproxy-server
    env = GLOBUS_LOCATION=/usr/local/globus LD_LIBRARY_PATH=/usr/local/globus/lib
disable = no
}

Reloading configuration: [ OK ]
tcp 0 0 0.0.0.0:7512 0.0.0.0:* LISTEN
```

**Figure 5.14:** Myproxy configuration testing
To continue on, gridftp configuration will be tested next. Figure 5.15 will show that the configuration is done correctly and the gridftp server is listening from other nodes within the grid.

![Figure 5.15: Gridftp configuration testing](image)

The last test will be to check whether the configuration of all the grid application is correct and Globus will be able to send a job. Figure 5.16 will display that the services is working and job could be submitted by an application.

![Figure 5.16: Test in submitting job by an application](image)
5.4.3 Node Machine Setup Testing

For node machine setup test, the first test is to show that all necessary security file is copied from the master machine so that the grid connection between master and node could take place. Figure 5.17 is a test that shows the necessary file is copied successfully.

```
Enter Master Machine Hostname: cengal
password: 485c15c6
    cengal
grid-security.conf.485c15c6 100% 948  0.9KB/s  00:08
grid-host.ssh.conf.485c15c6 100% 2718 2.7KB/s  00:00
grid-user.ssh.conf.485c15c6 100% 2830 2.8KB/s  00:00
485c15c6 signing policy 100% 1355  1.3KB/s  00:00
Enter Hostname: jatl
Enter MyProxy pass phrase: 
Credentials for jatlj have been stored in /etc/grid-security/hostcert.pem and etc/grid-security/hostkey.pem.
-rw------- 1 globus globus 904 Apr 24 05:58 containercert.pem
-rw------- 1 globus globus 887 Apr 24 05:58 containercert.pem
-rw------- 1 root root 904 Apr 24 05:58 hostcert.pem
-rw------- 1 root root 887 Apr 24 05:58 hostkey.pem
```

**Figure 5.17:** Node Machine security test

After the security test, gridftp for the node machine will be tested. Figure 5.18 will show that the configuration is done correctly and the gridftp server is listening from master machine and other nodes within the grid.

```
# Local services
ssh2 2011/tcp
Retrieving configuration: [ OK ]
tcp 0 0 0.0.0.0:2011 0.0.0.*: LISTEN
```

**Figure 5.18:** Node Gridftp configuration testing
The last test will be to check whether the configuration of all the grid application in the node machine is correct and Globus will be able to send and receive a job. Figure 5.19 will display that the services is working and job could be received and submitted by an application.

![Image of terminal output]

**Figure 5.19:** Testing in receiving and submitting a job in node machine

### 5.5 Conclusion

As a conclusion, this chapter discussed the implementation and testing of the agent tools. Implementation can be described as the realization system from the design and analysis phrases. From the coding and result of the testing, it shows that the agent tool achieve its requirement and can be implemented in a local area network for grid computing installation and setup purposes.
CHAPTER 6

CONCLUSION AND DISCUSSION

6.1 Introduction

In this chapter the summarization of all the chapters which is from Chapter 1 to Chapter 5 will be done. In this chapter also there will be an inclusion of achievement, challenges of the project and also expectation from it. The discussion in this chapter will help in understanding and familiarizing the user with the agent tool. By doing this discussion process, it will help recognize the achievement and also steps that must be taken to complete a project with this kind of magnitude.

6.2 Achievements

In this project, the background problem is successfully identified in Chapter 1. Then based on that background problem objectives and the system requirements are created to counter the problem faced in the background problem. Next, scopes
are created to specify the field or zone that the system will serve. Besides that, the modules and the function of the system is identify clearly in this documentation. Also by reading articles and papers for the project, the knowledge gain from it is quite exciting. This is because the grid computing environment is considered a new technology where not many people have considered doing it. This new technology also is not taught in the curriculum which makes it interesting and new. It hoped that by the end of the project new information and knowledge could be gather and could be applied in daily life.

6.3 Limitation and Challenges

There are a few limitation and challenges faced while completing this report. Below are few challenges that were faced during the duration of the project report.

i. Topic understanding

Grid computing is something new which just stars to gain its popularity among computer scientist when crating their application. Because of this most resources that was gain from research on the topic was quite vague. This is why the topic is harder to understand and a lack of example has hampered he project understanding.

ii. Time allocation

The time taken to understand the topic took quite sometime which hampered the project continuity. Also whit the work load during normal class time makes the project a bit harder to be completed. Alas, the leniency of the lectures and help from them had made possible to complete the project within the allocated time.
iii. System design

Agent tools to connect to MYREN grid has not been created before, because of this here are no existing software that can be simulated to complete the application. So most of the design is created after much consideration and taught.

6.4 Expectation

The first expectation for this project is to complete this application and make it fully functional and then fulfill the system requirement and scope that were mentioned in Chapter 1. Besides that, the program will benefit application that need to use grid computing environment to execute successfully.

The second expectation is to expand the application further so that it can support more platform than what have plan in the documentation. Furthermore, to distribute the application outside and as an open source to attract people in using it or to further develop this application so that the system will evolve into something better.

6.5 Summary

As a conclusion, the agent tools created in this project is able to atomize the installation and setup process in creating a grid computing environment in a local area network. By atomizing the process, the setup process will become significantly faster and void of error that will usually come from lack of experience in setting up a grid environment.
REFERENCES

APPENDIX A:

PSM 1 GANTT CHART
<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial Project Planning</td>
<td>8 days</td>
</tr>
<tr>
<td>2</td>
<td>Find organization</td>
<td>1 day</td>
</tr>
<tr>
<td>3</td>
<td>Identify Problem Background</td>
<td>4 days</td>
</tr>
<tr>
<td>4</td>
<td>Identify Objectives and Solution</td>
<td>2 days</td>
</tr>
<tr>
<td>5</td>
<td>Identify, Equipments and Development Tools</td>
<td>2 days</td>
</tr>
<tr>
<td>6</td>
<td>Background Study</td>
<td>7 days</td>
</tr>
<tr>
<td>7</td>
<td>Problem Background Of Existing System &amp; Case Study</td>
<td>3 days</td>
</tr>
<tr>
<td>8</td>
<td>Technology, Equipment and Development Tools Review</td>
<td>3 days</td>
</tr>
<tr>
<td>9</td>
<td>Hardware and Software Specification</td>
<td>3 days</td>
</tr>
<tr>
<td>10</td>
<td>Methodology</td>
<td>16 days</td>
</tr>
<tr>
<td>11</td>
<td>Identify suitable methodology to use</td>
<td>3 days</td>
</tr>
<tr>
<td>12</td>
<td>Choose methodology and describes each of the phases for the system development</td>
<td>12 days</td>
</tr>
<tr>
<td>13</td>
<td>Analysis &amp; Design</td>
<td>17 days</td>
</tr>
<tr>
<td>14</td>
<td>Gather requirements from organisation</td>
<td>5 days</td>
</tr>
<tr>
<td>15</td>
<td>Model the requirements using UML</td>
<td>7 days</td>
</tr>
<tr>
<td>16</td>
<td>Design the architecture and database</td>
<td>7 days</td>
</tr>
<tr>
<td>17</td>
<td>Conclusion &amp; Discussion</td>
<td>3 days</td>
</tr>
<tr>
<td>18</td>
<td>Summary of project achievement</td>
<td>1 day</td>
</tr>
<tr>
<td>19</td>
<td>Future projects planning &amp; conclusion</td>
<td>2 days</td>
</tr>
<tr>
<td>20</td>
<td>Draft Report Writing</td>
<td>11 days</td>
</tr>
<tr>
<td>21</td>
<td>Submit Draft to Supervisor</td>
<td>1 day</td>
</tr>
<tr>
<td>22</td>
<td>Supervisor evaluate drafts and suggestions</td>
<td>5 days</td>
</tr>
<tr>
<td>23</td>
<td>MOPH Draft Report</td>
<td>5 days</td>
</tr>
<tr>
<td>24</td>
<td>Submit Modified Draft</td>
<td>5 days</td>
</tr>
<tr>
<td>25</td>
<td>Presentation &amp; Final Modification</td>
<td>21 days</td>
</tr>
<tr>
<td>26</td>
<td>Prepare slides for presentation</td>
<td>7 days</td>
</tr>
<tr>
<td>27</td>
<td>Project presentation</td>
<td>3 days</td>
</tr>
<tr>
<td>28</td>
<td>Make final draft correction</td>
<td>5 days</td>
</tr>
<tr>
<td>29</td>
<td>Supervised approval of Final Draft</td>
<td>3 days</td>
</tr>
<tr>
<td>30</td>
<td>Submit Final Modified Report</td>
<td>1 day</td>
</tr>
</tbody>
</table>
APPENDIX B:

PSM 2 GANTT CHART
<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Early Implementation Planning</td>
<td>8 days</td>
</tr>
<tr>
<td>2</td>
<td>Amend PDMI</td>
<td>1 day</td>
</tr>
<tr>
<td>3</td>
<td>Address system design</td>
<td>3 days</td>
</tr>
<tr>
<td>4</td>
<td>Gather data for sampling</td>
<td>2 days</td>
</tr>
<tr>
<td>5</td>
<td>Total coding on the sample data</td>
<td>3 days</td>
</tr>
<tr>
<td>6</td>
<td><strong>Implementation Phase</strong></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Design Clinic &amp; Win System Interface</td>
<td>14 days</td>
</tr>
<tr>
<td>8</td>
<td>Coding on a Media DSS</td>
<td>24 days</td>
</tr>
<tr>
<td>9</td>
<td>Coding on MedialPain Clinic System</td>
<td>40 days</td>
</tr>
<tr>
<td>10</td>
<td><strong>Testing Phase</strong></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Testing and modification on system design and database</td>
<td>7 days</td>
</tr>
<tr>
<td>12</td>
<td>Testing and modification on system modules</td>
<td>10 days</td>
</tr>
<tr>
<td>13</td>
<td>Testing and modification on system interface</td>
<td>10 days</td>
</tr>
<tr>
<td>14</td>
<td>Full Report Writing</td>
<td>55 days</td>
</tr>
<tr>
<td>15</td>
<td>Write full report draft</td>
<td>40 days</td>
</tr>
<tr>
<td>16</td>
<td>Submit full report draft to supervisor</td>
<td>1 day</td>
</tr>
<tr>
<td>17</td>
<td>Supervisor evaluates report and suggest enhancements</td>
<td>7 days</td>
</tr>
<tr>
<td>18</td>
<td>Modify final report</td>
<td>5 days</td>
</tr>
<tr>
<td>19</td>
<td>Submit complete final report</td>
<td>5 days</td>
</tr>
<tr>
<td>20</td>
<td><strong>Presentation &amp; Final Modification</strong></td>
<td>22 days</td>
</tr>
<tr>
<td>21</td>
<td>Prepare slides for presentation</td>
<td>7 days</td>
</tr>
<tr>
<td>22</td>
<td>Make presentation</td>
<td>5 days</td>
</tr>
<tr>
<td>23</td>
<td>Make final correction to report</td>
<td>5 days</td>
</tr>
<tr>
<td>24</td>
<td>Supervisor approved on final report</td>
<td>3 days</td>
</tr>
<tr>
<td>25</td>
<td>Submit complete thesis</td>
<td>1 day</td>
</tr>
</tbody>
</table>
1.0 Agent Tool Folder

For the Agent tool application for installing and setting up Globus 4.2.1 it is contained in a folder. The name of the folder is GT4. This folder is used to store all the necessary script and application for installing Globus 4.2.1. The characteristic of the folder is as follows.

1. Size of the folder : 191 MB
2. Name of the folder : GT4
3. The folder contains 25 files which will be listed as follows
   a. Four application file
      i. JDK 1.5.0_15
      ii. Apache Ant 1.7.1
      iii. Perl XML Parser 2.34-6
      iv. Globus installer 4.2.1
   b. 19 shell script.
   c. 2 configuration file.

To use the application, first the folder must be moved to the root folder on the root user (administrator account). The location should be “/root/GT4”. Once the folder is moved to this location, you can start using the agent tool.
2.0 To run the agent tool.

After moving the folder to the appropriate directory now the agent tool can be used. The first step is to open a terminal and type in the following command shown in Figure C1. The command is

```bash
# cd /root/GT4
#./install
```

![Figure C1: To Lunch the agent tool](image)

Once the command is inputted on the terminal the agent tool will launch and the user interface will be displayed.
3.0 Agent Tool User Interface

After the agent tool is launch, the user interface of the program will be displayed shown in Figure C2.

![Figure C2: Agent tool user Interface](image)

From Figure C2, there are five functions to choose from the agent tool. To choose this function the user need to use the UP/DOWN arrow to select the functionality and press ENTER to select the process. The user could also use the number in RED as a shortcut to the function that is needed.

From Figure C2 there are five functions that can be considered from. This functionality will be explained as follows.
1. Install Globus 4.2.1
   a. This function is design to install the necessary pre requisite application for Globus and installing Globus 4.2.1 when the pre requisite is meet.
   b. The function must be used on the first / master machine and also the node machine.
2. Master machine setup
   a. This functionality is design to provide the procedure to setup a master machine before connecting it to a node.
   b. This function must be use on the first / master machine only.
3. Node machine setup
   a. This functionality is used to provide the necessary procedure to setup a node one a master machine is assign. After a master machine is assign the procedure could be run on the node machine.
   b. This function must be used on the node machine after the setup of the first machine.
4. Add node
   a. This function is used on the master machine every time a node is added to the grid.
   b. This function must only be used on the master machine.
5. Exit Program
   a. This function is used when the user wanted to exit the program.
4.0 Quickstart

Following steps is the best procedure need to be taken when creating a grid within a Local Area Network. This step will make it easier for the user to perform the setup.

1. Launch the agent tool on the master/first machine.
2. Choose 1. Install Globus 4.2.1
3. The whole process will take around 1 to 1.5 hour depending on your computer capabilities. Be patient.
4. Once the installation is completed, the user interface will appear again automatically.
5. Choose 2. For Master machine Setup.
6. There will be a couple of input need to be entered which is the hostname and password.
7. Once the setup is completed the interface will appear.
9. This function will ask the user to enter the hostname of the node. Once its entered the node can then connect to the master machine.
10. Launch the agent tool on the node machine.
11. Choose 1. Install Globus 4.2.1
12. The whole process will take around 1 to 1.5 hour depending on your computer capabilities. Be patient.
13. Once the installation is completed, the user interface will appear again automatically.
15. There will be a couple of input need to be entered which is the hostname and password.
16. Once the node machine setup is completed, you can start using your grid application.
17. You can now exit the program.