

A Generic Campus Grid Computing Framework for Tertiary Institutions - the Case of the University of Stellenbosch

Samuel Tewelde Yigzaw

**Assignment submitted in partial fulfilment of the requirements for the degree of Master
of Philosophy (Information and Knowledge Management) at the University of
Stellenbosch**



Supervisor: Mr. Daniel F. Botha

December 2005

Declaration

I, the undersigned, hereby declare that the work contained in this assignment is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature:

Date:

Abstract

Prior to the invention of Personal Computers the scope of research activities was limited by the pre-existing capabilities of problem solving mechanisms. However, with the advent of PCs and inter-networking thereof, the new tools (hardware and software) enabled the scientific community to tackle more complex research challenges and this led to a better understanding of our environment. The development of the Internet also enabled research communities to communicate and share information in real time.

However, even the Internet has limitations of its own when it comes to the need of sharing not only information but also massive storage, processing power, huge databases and applications, expensive and delicate scientific instruments, knowledge and expertise. This led to the need for a networking system that includes these above-mentioned services, using the Internet infrastructure, semantic web technologies and pervasive computing devices, which is so called *Grid Computing*. This research study deals with a Generic Campus Grid Computing framework, which mobilizes the available idle/extra computing resources residing in the faculty-computing centres for use by the e-community on CPU-intensive or Data-intensive jobs. This unused computing capacity could be utilized for Grid computing services; hence, the already available resources could be more efficiently exploited. Besides, this could be a huge saving when compared to the cost of acquiring supercomputers by these institutions. Therefore, this research study intends to establish a simple and functional Generic Campus Grid Computing Framework at this stage, with the consent that subsequent research studies could deal with further assessment in a more detailed perspective and practical implementation thereof.

Opsomming

Voor die uitvinding van die Persoonlike Rekenaar is die omvang van navorsingsaktiwiteite beperk deur die voorafbestaande vermoëns van probleemoplossingsmeganismes. Met die verskyning van PR's en die daaropvolgende internetwerking daarvan, het die nuwe gereedskap (hardeware en sagteware) die wetenskaplike gemeenskap in staat gestel om meer komplekse navorsingsuitdagings aan te pak. Dit het gelei tot groter begrip van ons omgewing. Die ontwikkeling van die Internet het navorsingsgemeenskappe ook in staat gestel om in reële tyd te kommunikeer en inligting te deel.

Nietemin, selfs die Internet het gebreke wanneer dit kom by die behoefte om nie slegs inligting te deel nie, maar ook massiewe stoorruimte, verwerkingskrag, baie groot databasisse en toepassings, duur en delikate wetenskaplike toerusting, kennis en kundigheid. Dit het gelei tot die behoefte aan 'n netwerksisteem wat bogenoemde dienste insluit, deur gebruik te maak van Internet-infrastruktuur, semantiese web tegnologieë, en alomteenwoordige rekenaar-toestelle. Hierdie sisteem staan bekend as "Grid Computing" of te wel Rooster Komputasie. Hierdie navorsingstudie handel oor 'n Generiese Kampus Rooster Komputasie Raamwerk wat die ongebruikte, ekstra komputasiebronne, wat beskikbaar is in fakulteite se rekenaar-gebruikersareas, mobiliseer vir gebruik deur die e-gemeenskap op SVE-intensiewe of Data-intensiewe toepassings. Hierdie ongebruikte komputasie kapasiteit kan aangewend word vir Rooster komputasie dienste; gevolglik kan die beskikbare bronne dan meer effektief benut word. Verder kan dit lei tot groot besparings wanneer dit vergelyk word met die koste om superrekenaars aan te koop deur die betrokke instansies. Dus, op hierdie stadium stel hierdie navorsingstudie dit ten doel om 'n eenvoudige en funksionele Generiese Kampus Rooster Komputasie Raamwerk te skep met dien verstande dat daaropvolgende studies sou kon fokus op verdere assessering met 'n meer gedetailleerde perspektief en met praktiese implementasie.

Dedication

This thesis is dedicated to my beloved family for their invaluable moral and material support in all walks of my life.

Acknowledgements

My sincere gratitude to the Almighty God who kept me focused and let me accomplish my study successfully and in good health.

My special appreciation goes to my supervisor, Mr. D.F. Botha, for his invaluable encouragement and guidance, which enabled me to complete this research study in a good faith and contentment. His helpful guidance and advice further encourages me to open my future endeavours to contribute to this on-going field of research.

I am grateful to Dr.Martin Van Der Walt for his humble cooperation in all my requests from the department and his invaluable encouragement that helped me to smoothly complete my MPhil program.

My profound gratitude also goes to the NUFFIC and the University of Asmara for their financial support and giving me this opportunity to study for my MPhil.

Many people have helped me during the course of my study. My heartfelt gratitude goes to Mr.Andries Ruiters of the University of Groningen, for his material and expertise advice. Special thanks also to Dr.Tewelde Zerom, from the UOA, for his patience and invaluable cooperation, especially in my research study materials support. I also thank Mr.Jan Louw, the Humarga manager for his considerate cooperation in network information gathering; and Mr.Francoise of StoneTree Company, Dr.Rudiger from Bio-Informatics at UWC and his colleagues, and Ms.Heide for their invaluable support.

I am also indebted to all the staff members of the department of Information and Knowledge Management, the University of Stellenbosch, for their considerate attention and cooperation.

Last but not least, I owe a very special thanks to my beloved family for their moral and material support, which helped me realize my dream; and to my friends for their continuous support in all walks of my life.

Table of contents

DECLARATION -----	II
ABSTRACT -----	III
OPSOMMING -----	IV
DEDICATION -----	V
ACKNOWLEDGEMENTS -----	VI
LIST OF FIGURES -----	X
CHAPTER 1 -----	1
INTRODUCTION AND RESEARCH METHODOLOGY -----	1
1.1 – INTRODUCTION -----	1
1.2 - PROBLEM STATEMENT & OBJECTIVES-----	3
1.2.1 - Problem Statement-----	3
1.2.2 - Objectives -----	3
1.3 - RESEARCH METHODOLOGY AND APPROACH -----	4
1.3.1 - Literature Review and Case Studies -----	4
1.3.2 - Hands-on assessment of the University of Stellenbosch’s Intranet infrastructure and services -----	5
1.4 - SCOPE AND LIMITATIONS OF THE STUDY -----	6
1.5 - IMPACT OF THE RESEARCH -----	6
CHAPTER 2 -----	7
EARLY STAGES AND DEVELOPMENT OF THE GRID COMPUTING -----	7
2.1 – STUDY BACKGROUND-----	7
2.1.1 - Introduction and Definitions: -----	7
2.1.2 – A Brief History of Grid Computing Development-----	10
2.1.3 – How Does the Grid work? -----	13
2.2 – WHY GRID COMPUTING SYSTEMS? -----	13
2.2.1 – The Drive for Developing Grid System and Limitations of the Pre-existing System -----	13
2.2.2 - The Prospects of the New System -----	16
2.2.3 - Semantic Grids -----	18
2.3 - LATEST DEVELOPMENTS IN THE GRID RESEARCH -----	20
2.3.2 – Opportunities Promoting Grid development -----	20
2.3.3 – Potential Threats to Grid Development Efforts -----	21
2.3.4 – Properties of Future Grid Developments -----	23

CHAPTER SUMMARY -----	25
CHAPTER 3 -----	26
TYPES OF GRID ENVIRONMENTS- SCOPE AND ARCHITECTURE, AND COMPONENTS OF A GENERIC GRID FRAMEWORK -----	26
3.1 - INTRODUCTION -----	26
3.1.1 - Virtually Mobilizing Grid Resources -----	26
3.1.2 – Conceptual Layers of Grid Computing -----	27
3.2 - SCOPE AND COMPONENTS OF THE GRID COMPUTING -----	30
3.2.1 – Scope of the Grid Computing design-----	30
3.2.2 – Types and Components of Grid Computing -----	32
3.2.3 – The Layers of Grid Computing-----	37
3.3 – GRID COMPUTING ARCHITECTURE-----	38
3.3.1 – Grid over VPN-----	38
3.3.2 – Virtual Private Grid-----	39
3.3.3 – Grid Community-----	39
3.3.4 – Ad Hoc Grid-----	40
3.4 - LEVELS OF GRID COMPUTING -----	40
3.4.1 - Cluster Grid-----	40
3.4.2 - Campus Grid-----	41
3.4.3 - Global Grid-----	41
3.5 – ISSUES OF SECURITY-----	42
3.6 - POTENTIAL CHALLENGES OF THE GRID COMPUTING-----	44
CHAPTER SUMMARY -----	47
CHAPTER 4 -----	49
UNIVERSITY OF STELLENBOSCH CAMPUS-GRID FRAMEWORK AND INTER-CAMPUS GRID COMPUTING -----	49
4.1 – INTRODUCTION AND GENERAL PRINCIPLES -----	49
4.2 - POTENTIAL CAPACITY OF THE US FACULTY-COMPUTING CENTRES BASED ON THE CASE OF THE HUMARGA COMPUTING CENTRE -----	50
4.2.1 - Processing Capacity-----	50
4.2.2 - Storage Capacity-----	50
4.2.3 – Current Utilization of the Computing Resources -----	51
4.2.4 - Assessment of the Applications Used In the Humarga Computing Centre-----	52
4.3 - US GRID FRAMEWORK -----	53

4.3.1 - Computing Resource Mobilization and Components of the US C-Grid -----	53
4.3.2 – The Logical Campus Grid Architecture-----	54
4.3.3 – The Physical Campus Grid Architecture -----	57
4.3.4 - What makes this Campus-Grid architecture different?-----	59
4.4 – IMPACT OF THE US C-GRID FRAMEWORK -----	59
4.4.1 – The Prospects of Enhanced Network Services -----	59
4.4.2 – Its Impact on Inter-Campus Resource Sharing -----	61
CHAPTER SUMMARY-----	62
CHAPTER 5-----	63
SUMMARY, CONCLUSION AND RECOMMENDATIONS-----	63
5.1 – SUMMARY -----	63
5.2 – CONCLUSION-----	67
5.3 - RECOMMENDATION -----	68
REFERENCES-----	69
APPENDIX – A -----	I
APPENDIX – B -----	III
APPENDIX – C -----	IV
APPENDIX – D -----	V

List of Figures

Fig 2.1 - Computing Platform Evolution -----	12
Fig 2.2 - The Semantic Grid -----	19
Fig 3.1 - The three-layered Architecture viewed as services -----	29
Fig 3.2 – Grid System Architecture -----	31
Fig 3.3 - The layered Grid protocol architecture, compared to the Internet protocol architecture -----	37
Fig 4.1 The one month-profile for average CPU utilization of the Humarga Server (21/03/2005 – 20/04/2005), Humarga Computing Centre-----	51
Fig 4.2 - Network utilization graph in a random 24hrs period -----	52
Fig 4.3 - Logical Campus Grid Layout -----	56
Fig 4.4 - Physical Campus Grid Layout - -----	58

Chapter 1

Introduction and Research Methodology

1.1 – Introduction

In the last couple of decades, there has been a substantial change in the way we perceive and use computing resources and services. In the early stage of this development, the motive for adopting a computing technology was to facilitate a job, thereby replacing the need for relatively more labour to do it and to do it faster. In addition, it was normal to expect one's computing needs to be served by localized computing platforms and infrastructure. This was followed by a focus on improving the processing power of the computers via breakthroughs in the semiconductor technologies and the taking-up of networking components- software and hardware. These greatly changed the way jobs were done and promoted new ideas on shared computing systems- *inter-networking*. This inter-networking also opened a new era where a need for sharing information and cooperative work in processing tasks led to the development of new technology that linked different clusters and local area networks: the *Internet* that spans the globe. Information infrastructure like this, using World Wide Web (WWW) technologies, has involved a series of improvements to accommodate ever-increasing needs for improved and shared services.

However, the Internet has limitations on its own when it comes to the need of sharing not only information but also massive storage, processing power, huge databases and applications, expensive and delicate scientific instruments, knowledge and expertise. Therefore, these drove the evolution of a new networking architecture/system that includes these above-mentioned services, using the internet infrastructure and improved WWW technologies. This system is called *Grid Computing*.

Sun Microsystems, in the *Grid technology overview: Sun powers the Grid* [Online], describes the new generation inter-networking- the *Grid*, as follows:

Grid Computing delivers on the potential in the growth and abundance of network connected systems and bandwidth: computation, collaboration and communication over the Advanced Web. At the heart of Grid Computing is a computing infrastructure that provides dependable, consistent, pervasive and inexpensive access to computational capabilities. By pooling federated assets into a virtual system, a grid provides a single point of access to powerful distributed resources.

Using a Grid computing system, networked resources such as desktops, servers, super computers, storage, databases, and even expensive scientific instruments, could be virtually mobilized to deploy massive computing power wherever and whenever it is needed most. Users who comply with the requirements of membership or subscription to this Grid system can find resources quickly, use them efficiently, and scale them seamlessly.

The current status of computation is analogous in some respects to that of electricity in the early 20th century. At that time, electric power generation was possible, and new devices were being devised that depended on electric power, but the need for each user to build and operate a new generator hindered use. The truly revolutionary development was not, in fact, electricity, but the electric power grid and the associated transmission and distribution technologies. Together, these developments provided reliable, low-cost access to a standardized electricity service. Similarly, the quest for more computing power beyond what is available in the prevailing system by data intensive and/or scientific research led to the acquisition of supercomputers and high capacity servers. The evolution of Grid computing, therefore, could be considered to provide the foundation to the extended and ever increasing need for more computing power.

Generally, the Grid, as the IT infrastructure of the future, is expected to transform *computation, communication, and collaboration* between different systems, businesses and users. Over time, these could be seen in the context of grids- academic grids, enterprise grids, research grids, entertainment grids, community grids, and so on. As described in different papers and articles, Grids will become service-driven with lightweight clients accessing computing resources over the Internet. Data-centres will be safe, reliable, and available from anywhere in the world. Applications will be part of a wide spectrum of network-delivered services that include computing cycles, data processing tools, accounting and monitoring, and more.

This thesis will discuss the issues of Grid computing- the development, scope of services, its various new features and the capabilities it offers to transform *computation, communication and collaboration*, while focusing on a preliminary study on establishing a University of Stellenbosch Campus-Grid Framework, which will be referred to as *US C-Grid Framework* throughout the study. Moreover, it will discuss possible ways of inter-campus Grid links with other universities.

1.2 - Problem Statement & Objectives

1.2.1 - Problem Statement

With the increasing sizes and levels of complexity of research data, there inevitably comes the need for more computing power and storage capacity. Especially important is the case of universities and big research centres, where advanced and more complicated data are involved in different academic or other research instances. Thus far, the use of supercomputers has been able to solve the problem with the required computing power. However, the increasing need for more computing power goes beyond the capacity of supercomputers and draws a need to think of a new method of addressing the problem. Today, grid-computing system is on the verge to effectively address the problem of the demand for immense computing capacity and storage to handle the ever-complex research data in a more efficient fashion. Hence, it is clear that universities should benefit from utilizing their *idle/extra* capacities, especially with their faculties' computing centres where they keep a huge number of computers whose idle state could be predicted. Additionally, numerous universities can mobilize their computing resources and create a virtual computing environment where each partner university can benefit from such organization, and this is what Grid is all about.

However, the main questions remain: how could physically distributed computing resources be virtually organized? Which capabilities could be utilized in this virtual organization of the computing resources? And, how could a generic Campus Grid Framework be established and managed based on the US Intranet?

1.2.2 - Objectives

As described in the previous sections, a great challenge for academic and research institutions is to have the necessary computing power and storage capacity. Based on relevant research in the areas of Information and Communication Technologies, especially the new developments in the Internet and WWW technologies, and knowledge management, this study aims to achieve the following primary objective:

To develop a GCGC (Generic Campus Grid Computing) Framework for a tertiary institution that could be used to assess/evaluate how it could virtually organize its network resources to enable intra- and inter-campus computing resource sharing.

The above stated main objective will be achieved by realizing the following sub-objectives. These sub-objectives include:

- To present a literature background study of Grid computing and a summary on findings of the latest developments in the field
- To establish the salient components that define a generic Campus Grid framework
- To estimate the *idle/extra computing power* that can be mobilized from the faculty-computing centres of the University of Stellenbosch
- Estimating the capabilities that the University of Stellenbosch Campus-Grid can offer for inter-campus computing resource sharing

1.3 - Research Methodology and Approach

Based on the objectives described above, it is considered appropriate to use two different but interdependent methodologies. These methodologies comprise, firstly, a review of existing literature on 'Grid Computing', and secondly, hands-on assessment of the US Intranet infrastructure and services. Both methodologies will consist of various approaches as described in the respective sections below and the findings will be synthesized into a *GCGC Framework*

1.3.1 - Literature Review and Case Studies

As the area of study namely, Grid Computing, is relatively new and only flourished recently, it is believed that there is a dearth of published books on the subject. Therefore, the literature review and most of the work will be referring to some of the latest available books and different articles published on the field, especially by prominent computer scientists and founders and leaders in the ongoing research on the Grid Computing. The gathered information will be used to deepen the understanding of Grid Computing. Furthermore, a graphical description will be incorporated into the study clearly presenting the way Grid System works.

Additionally, a case study approach will be used to illustrate how a Grid System could be implemented in a tertiary institution. The case study will be limited to University of Stellenbosch due to its excellent proximity and the opportunity to gather the necessary information/data, as well as due to time and financial constraints of the study.

1.3.2 - Hands-on assessment of the University of Stellenbosch's Intranet infrastructure and services

The following sections describe the approaches that will be used to gather the relevant information for assessing US intranet infrastructure and services.

Estimation of the Idle computing power (for the computing centres): This will be done by collecting network information on one of the computing centres- Humarga, as the manager of Humarga has expressed cooperation and agreed to assist in giving access to collecting the information or providing the required information. The collected information will be used to determine the idle/extra cycles and storage capacities of each computer in the computing centre and for all the computers in general over a period of time. These capacities will be determined later according to the means of acquiring the information, so that the pattern of the idle/extra computing capacity could be predicted for the centre in the future.

Estimating the capabilities US Campus-Grid can offer: General assessment will be made on the services that the US Campus Grid will be able to deliver to the users. In this research study, the primary intention of establishing a generic framework for mobilizing such a computing capacity virtually may not necessarily be aiming to address a particular limitation or problem. Rather, it shall be viewed as a guide for establishing a basic and functional means of mobilizing the maximum computing and storage capacity as a starting phase so that the extra capacity could either effectively be used for internal services and/or inter-campus resource sharing. The virtually organized idle/extra capacity can also be provided for commercial use if it is not fully used otherwise yet.

From the above investigations, the necessary information will be synthesized to enable the development of a generic framework for the US Campus-Grid computing. This framework will be representing the University of Stellenbosch's virtual organization of the available computing capacity in the faculty-computing centres and ways of effectively utilizing the idle/extra computing and storage capacity of the Intranet. Note that the capacity of the other faculty-computing centres could also be estimated depending on the findings of a more detailed future study of the capacity at the Humarga computing centre.

Moreover, the research will also assess the possible scalability of the Campus-Grid network to include Inter-Campus network resource sharing.

1.4 - Scope and Limitations of the Study

This thesis project will be dealing with establishing a theoretical model for a GCGC framework from literature study and background research of the Grid computing system on the one hand, and identifying and organizing the findings of the hands-on assessment of the University of Stellenbosch Intranet resources and capacities, which are salient components of a Campus Grid framework, on the other. From the findings of both methodologies, a GCGC Framework for a tertiary institution will be established. Therefore, the scope of the research extends up to practical assessment of the University of Stellenbosch Intranet to the extent possible, as a typical tertiary institution Intranet setup. However, due to the shortage of published books in this field, most sources of the literature study will be limited to a few renowned books and articles by prominent figures/authors in the field and online publications on the Grid computing system, especially on Campus Grid.

1.5 - Impact of the Research

This preliminary study will be done consulting different articles and a case study on Campus-Grid Computing. Therefore, the key contribution of this study is expected to be providing the groundwork with basic information and methods for developing Campus-Grid for a generic tertiary institution, and especially for the University of Stellenbosch. The study will additionally provide an elaborate description and assess developments in the Grid environment. Hopefully, this study will encourage/induce further researches that will enable the practical implementation of the Campus-Grid System for the US.

Chapter 2

Early Stages and Development of the Grid Computing

2.1 – Study Background

2.1.1 - Introduction and Definitions:

As described in the introduction section of chapter one, the IT infrastructure has gone through ever-changing stages to accommodate the continuously increasing demands for more capacity and services. Hence, the Internet evolved providing global access to information in a real time, although the reliability of the information published and the manageability of the infrastructure is still questionable; and enabling virtual interconnection of enterprise networks. In its latest stages, the Internet has incorporated new developments that include semantic web technologies and pervasive computing, which provide the basis for further possible developments in the field.

The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

(Berners-Lee, Hendler & Lassila, 2001)

The World Wide Web Consortium (W3C) as quoted in Goble & Roure, *Semantic Web and Grid [online]*, describes the Semantic Web as:

an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is the idea of having data on the Web defined and linked in a way that it can be used for more effective discovery, automation, integration, and reuse across various applications. The Web can reach its full potential if it becomes a place where data can be shared and processed by automated tools as well as by people.

David De Roure (2003) also defines Pervasive Computing as follows:

Pervasive computing is the trend towards increasingly ubiquitous (another name for the movement is ubiquitous computing), connected computing devices in the environment, a trend being brought about by a convergence of advanced electronic - and particularly, wireless - technologies and the Internet. Pervasive computing devices are not personal computers as we tend to think of them, but very tiny - even invisible - devices, either mobile or embedded in almost any type of object imaginable, including cars, tools,

appliances, clothing and various consumer goods - all communicating through increasingly interconnected networks.

Another definition also asserts:

Pervasive Computing (sometimes called ubiquitous computing) is a vision of our future computing lifestyle in which computer systems seamlessly integrate into our everyday lives, providing services and information in an "anywhere, anytime" fashion. (Michael N. Huhns, 2004)

The Internet has limitations of its own when it comes to the need for sharing not only information but also massive storage, processing power, huge databases and applications, expensive and delicate scientific or otherwise laboratory instruments. Therefore, there inevitably is a need to think of a networking architecture/system that includes these above-mentioned services and much more, using the Internet infrastructure and semantic WWW technologies, the so called *Grid Computing*.

In the mid 1990s Foster and Kesselman proposed a distributed computing infrastructure for advanced science and engineering, dubbed *The Grid*. The name arose from an analogy with an electricity power grid: computing and data resources would be delivered over the Internet seamlessly, transparently and dynamically as and when needed, just like electricity.

Minoli, Daniel (2005: 281-282) described that Grid computing can be considered as:

a network of computation and supports the concept of network 'Utility Computing' with which users can get 'on demand' machine cycles off a Grid with out having to own the physical asset. He further asserted that Grid computing also supports the concept of Enterprise Grid with which organizations make more synergistic use of often-underutilized assets they already own.

Another more elaborate definition is also in Daniel Minoli (2005:355), saying:

Grid computing is a network of computation, namely tools and protocols, for coordinated resource sharing and problem solving among pooled assets. It allows coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations

The Grid was distinguished from conventional distributed computing by a focus on large-scale resource sharing, innovative science-based applications and a high performance orientation. In recent years, however, the focus has shifted from the high performance aspect

towards a broader definition involving a flexible, secure and coordinated resource sharing among dynamic collections of individuals, institutions and resources - what we refer to as *Virtual Organizations*.

A good example of conventional distributed computing is the *cluster computing*.

A cluster is comprised of multiple interconnected, independent nodes that cooperatively work together as a single unified resource; unlike Grids, cluster resources are typically owned by a single organization, as defined by Minoli, Daniel (2005:354)

Current Grid developments are transforming the prevailing concept of information sharing to a new era of knowledge sharing via the capability of the system to provide effective sharing of the vast computing power, scientific data and information, various intelligent devices, sensors and creating cooperative communities, through virtually organizing the resources, hence, creating a Virtual Organization.

The Globus project, which conducts research and development to create fundamental technologies behind the Grid, introduces the term *virtual organization (VO)* as:

a set of users in multiple network domains who wish to share some of their resources. The virtual organization might be (as is currently the norm) a group of academic institutions that wish to enhance resource sharing. (Foster, I. [et al], 2002)

Foster, Kesselman & Tueck, in Berman, Fox & Hey (2003:172), also define Virtual Organization as:

a set of individuals and/or institutions defined by such sharing rules that are necessarily highly controlled, with resource providers and consumers defining clearly and carefully just what are shared, who are allowed to share, and the conditions under which the sharing occurs. This resource sharing includes- direct access to computers, software, data and other resources as required by a range of collaborative problem solving and resource brokering strategies emerging in industry, science and engineering.

Grid computing system needs a basic enabling infrastructure, which provides the environment where different Grid applications run. Besides, it needs software using common protocols that define the mechanisms by which the users of services and the virtually organized resources negotiate, thereby addressing the fundamental issue of inter-operability of applications and services in the Grid system. This Grid *software* is often called *middleware* because it is mid-level software that provides services to users and to the resources.

Foster, Kesselman & Tueck, in Berman, Fox & Hey (2003:176), define Grid Middleware as:

the services needed to support a common set of applications in a distributed network environment.

As described in an article by David De Roure, [et al.] (2003), the development of grid computing through time can be characterized in three consecutive stages, while the change continues.

Referred to as the first generation, the early Grid efforts started as projects to link supercomputing sites, and this approach was known as Meta-computing.

In the second generation the core software for the Grid has evolved from that provided by the early vanguard offerings, such as Globus (GT1) and Legion, which were dedicated to provision of proprietary services to large and computationally intensive high performance applications, through to the more generic and open deployment of Globus (GT2) and Avaki. Alongside this core software, the second generation also saw the development of a range of accompanying tools and utilities, which were developed to provide higher-level services to both users and applications, and spans resource schedulers and brokers as well as domain specific users' interfaces and portals. Peer-to-peer techniques have also emerged during this period.

The third generation is a more holistic view of grid computing and can be perceived to address the infrastructure for e-Science rather than the enabling technology. In particular, the terms *distributed collaboration* and *virtual organization* were adopted by Foster, Kesselman, & Tueck (2003) in the *Anatomy of the Grid System*.

This chapter will discuss the issues of Grid computing- the history of Grid computing, the motive/drive for the development of the Grid system, the prospects of the new developments in this field and the inter-relationship between the Grid computing and Semantic web technologies, the opportunities, threats and the future of Grid Computing.

2.1.2 – A Brief History of Grid Computing Development

Compiled from various literature sources, this section provides a brief overview of the history of Grid computing.

Perhaps the best place to start is in the 1980s, a decade of intense research, development and deployment of hardware, software and applications for parallel computing. Parallel computing

in the 1980s focused researchers' efforts on the development of algorithms, programs and architectures that supported simultaneity.

During the 1980s, researchers from across disciplines also began to come together to attack *grand challenge* problems that included key problems in science and engineering for which large-scale computational infrastructure provided a fundamental tool to achieve new scientific discoveries. The grand challenge and multidisciplinary problem teams provided a model for collaboration that has had a tremendous impact on the way large-scale science is conducted to date. Today, interdisciplinary research has not only provided a model for collaboration but has also inspired whole disciplines (e.g. bioinformatics) that integrate formerly disparate areas of science.

In the 1990s, the US Gigabit test-bed program included a focus on distributed metropolitan-area and wide-area applications.

The first modern Grid is generally considered to be the information wide-area year (I-WAY), developed as an experimental demonstration project. In 1995, during the week-long supercomputing conference, pioneering researchers came together to aggregate a national distributed test-bed with over 17 sites networked together. Over 60 applications were developed for the conference and deployed on the I-WAY, as well as a rudimentary Grid software infrastructure to provide access, enforce security, and coordinate resources and other activities.

In the late 1990s, Grid researchers came together in the Grid Forum subsequently expanded to the Global Grid Forum (GGF), where much of the early research is now evolving into the standard base for future Grids. Recently, the GGF has been instrumental in the development of the Open Grid Services Architecture (OGSA), which integrates Globus and Web services approaches. OGSA is being developed by both the United States and European initiatives, and many other scholars from all over the world, aiming to define core services for a wide variety of areas including- Systems Management and Automation, Workload/Performance Management, Security, Availability/Service Management, Logical Resource Management, Clustering Services, Connectivity Management, Physical Resource Management.

Next-generation Grid applications, as described by Berman, Fox & Hey (2003:41) will include the following:

- *Adaptive applications (run where they can find resources satisfying their specific criteria)*

- *Real-time and on-demand applications (do something right now)*
- *Coordinated applications (dynamic programming, branch and bound) and*
- *Poly-applications (choice of resources for different components)*

The following figure depicts the evolution of the computing system from the stand alone PC to the currently evolving Grid system.

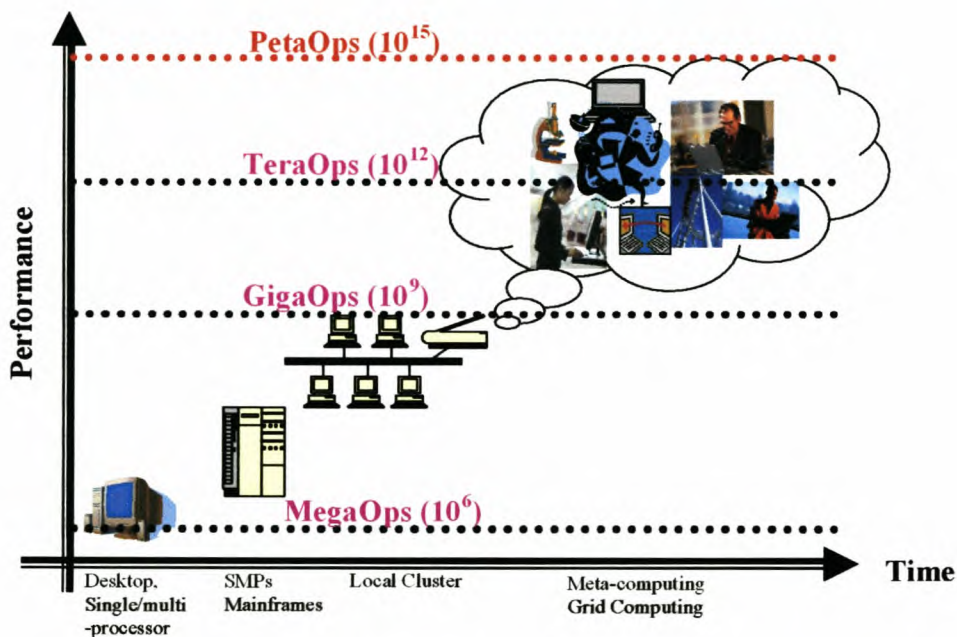


Fig 2.1 - Computing Platform Evolution, adapted from: *Some Technology Trends* (Laforenza, D. [Online])

As shown in the above figure, the performance level in *Operations per second (Ops)* increases as new computing systems evolve over time. It starts from Desktop systems with single/multi-processor systems, which function at MegaOps capacity, to Symmetric Multi-Processing (SMP) Mainframes and Local Clusters that operate up to GigaOps. The later developments in the Computing systems enabled jobs to run faster than ever with capacities beyond TeraOps and even up to PetaOps. These levels could be realized by the evolution of Meta-computing and lately the development of Grid computing, which is still in its early stage.

As Om Malik (2002) described, the prominent figure and founder of the world of Grid Computing is Mr. Ian Foster, like what Linus Torvalds, co-creator of the Linux operating

system, is to the open-source movement. Mr. Foster champions grid technology. Today, he says, grids are where the Web was in 1991 or 1992 -- more academic curiosity than commercial venture. But, just as the Internet grew from a collection of small academic networks to a 'humongous octopus' spreading its tentacles around the world, Mr. Foster predicts today's mini-Grids will grow into a huge global grid, a transcontinental processing pool engaged in all sorts of complex tasks, like designing and testing semiconductors and decoding the human genome. Applications like customer relationship management (CRM) and supply chain management (SCM) will be run from such a network.

2.1.3 – How Does the Grid work?

Considering a best describing analogy to the Grid Computing, an Electric Power Grid transparently delivers electricity from networked power stations seamlessly without the customer noticing where the electric source is from. Customers usually plug into a nearby electric outlet of comfort and immediately start consuming power from the pool of power stations- *Power Grid*, and need not care which power station is supplying the source at the particular moment. More interestingly, the electric power supplying source station may even change while the user appliance is turned on, that is, without any interruption/notice.

Similarly, Grid services are provided seamlessly to the authorized users regardless of which particular resource is providing the service at the particular moment. The user doesn't need to be concerned about by which Grid resource his/her job is processed, rather he/she expects the job to be done without any limitation of processing capacity, storage and/or memory. Moreover, the Grid resources while doing the job could encounter any problem and halt, but the process never gets lost. Technically, the Grid management system has got a fault tolerant capacity and addresses such issues like process interruption and/or malfunctioning by repeating the particular interrupted job on another resource readily available until the Grid management software (Middleware) confirms that the job submitted by the authorized user is complete, which then makes it ready to be pulled by the user.

2.2 – Why Grid Computing Systems?

2.2.1 – The Drive for Developing Grid System and Limitations of the Pre-existing System

Nowadays, many global projects are focusing on the development of Grid middleware and the mechanisms needed by the applied science community and big research institutions to

effectively exploit these infrastructures and available network resources and services. These projects are all being driven by the needs of these diverse communities involving both data and computing intensive applications, e.g. the EU Data-Grid, and NASA's Information Power Grid. (Steven Newhouse, [Online])

Foster & Kesselman (1998) discussed the following main factors that drive the need for developing Grid computing as innovations in a wide range of areas including:

- Technology improvement: Evolutionary changes in VLSI (Very-Large-Scale Integration) technology and microprocessor architecture can be expected to result in a factor of 10 increase in computational capabilities in the next five years, and a factor of 100 increase in the next ten years.
- Increase in demand-driven access to computational power: Many applications have only episodic requirements for substantial computational resources. For example, a medical diagnosis system may be run only when a cardiogram is performed, a stock market simulation only when a user re-computes retirement benefits, or a seismic simulation only after a major earthquake. If mechanisms are in place to allow reliable, instantaneous, and transparent access to high-end resources, then from the perspective of these applications it is as if those resources are dedicated to them.
- Increased utilization of idle capacity: Most low-end computers (PCs and workstations) are often idle: various studies report utilizations of around 30% in academic and commercial environments. Utilization can be increased by a factor of two, even for parallel programs, without impinging significantly on productivity. The benefit to individual users can be substantially greater.
- Greater sharing of computational results: The daily weather forecast represents such facility, which involves perhaps huge numerical operations, and shares the computational results effectively. Few other computational results or facilities are shared so effectively today, but they may be in the future as other scientific communities adopt a *big science* approach to computation. The key to more sharing may be the development of *collaboratories: centres without walls, in which the nation's researchers can perform their research without regard to geographical location- interacting with colleagues, accessing instrumentation, sharing data and computational resources, and accessing information in digital libraries.*

- New problem-solving techniques and tools: A variety of approaches can improve the efficiency or ease with which computation is applied to problem solving. For example, network-enabled solvers allow users to invoke advanced numerical solution methods without having to install sophisticated software.

Underlying each of these advances is the synergistic use of high-performance networking, computing, data sharing and advanced software to provide access to advanced computational capabilities, regardless of the location of users and resources.

Besides, the pre-existing system experiences limitations to meet the requirements of the new trends mentioned above. In parallel with this trend, IT infrastructures are experiencing a fundamental paradigm shift in moving away from scalable client/server approaches towards decentralised, scale-free and service-oriented approaches. They may consist of millions of heterogeneous networked components and billions of dependencies. The unprecedented level of complexity, instability and pervasiveness reached by today's IT systems very often leads to a situation where local or component failures can be propagated without control across the IT infrastructure and degenerate into global crashes or inconsistencies.

When confronted with such a level of complexity, the traditional computing models show limitations as they lack the capabilities, constructs and associated semantics necessary to express emergent and non-functional properties and behaviours and ensure fundamental properties such as consistency and completeness. Furthermore, the pre-existing implementation models offer very little support for managing, adapting, and reacting to complex contextual changes and failures.

Moreover, Foster, Kesselman & Tueck, in Berman, Fox & Hey (2003:172), argue that the Grid concept is indeed motivated by a real and specific problem and that there is an emerging, well-defined grid technology base that addresses significant aspects of this problem. They said:

The real and specific problem that underlies the grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.

Therefore, it became inevitable to search for a system that must address these complex and interwoven problems through intelligent and semantic systems with robust and fault tolerant capabilities. The vision of the Grid computing system is to deal with these shortcomings of the pre-existing internet technologies and accommodate the breakthroughs and innovations in the above mentioned areas of development.

2.2.2 - The Prospects of the New System

The popularity of Grids has been growing very rapidly driven by the promise that they will change dramatically the life of individuals, organisations and the society as much as the Internet has done in the past decade. By enabling knowledge and computing resources to be transparently delivered to and used by citizens and organisations as traditional utilities, Grids have the potential to give new impetus to the IT market and boost growth and competitiveness in many industrial and business sectors.

The NGG (Next Generation Grid) report 2004, by Keith Jeffery (2004) has identified new additional requirements that have arisen in the light of one more year of experience of the experts working in the Grids domain. In particular, the shortcomings of existing Grids middleware are much better understood, and despite the development directions concerning OGSA providing some integration with services-oriented architectures, it is becoming clearer that applications in the Grids environment require a greater range of services than can be provided by the combination of currently evolving Grids middleware and existing operating systems. This is particularly apparent in the area of knowledge and semantics; there is a need for semantically rich knowledge-based services in both Grid Foundations Middleware and Grid Services Middleware to improve the functionality and also to support Grid applications, which require semantic and knowledge-based support.

The Grids environment must behave intelligently using knowledge-based techniques, including semantic-rich operation and interoperation. Long-cherished computer science principles are re-examined in the light of the new requirements. There are particular implications in managing the software complexity demanded by the requirements derived from the applications envisaged; this has aspects of software theory, design, construction practice and also tools and environments to assist software development.

Recent market forecasts made by highly renowned business analysts stated that the worldwide market for IT services is expected to increase considerably in the future; in particular, process management services will grow fastest, as demand for outsourcing IT management and applications rises. Grid technologies have the potential to drive the market evolution of the IT industrial sector toward *IT services*. (Keith Jeffery. 2004)

A huge impact is expected from the implementation of the Grid computing on many activities that deal with complex and real-time issues. For example, a crisis management scenario requires that mobile workers (police, fire fighters, paramedics, environmental monitors,

military, etc.) collaborate in time-critical and dangerous situations, and have real-time access to information and knowledge in order to improve their decision-making processes under demanding circumstances.

Modern security personnel also deal with a number of difficult missions where technological and scientific tools can improve the resulting security. The scenarios are extended to the whole duration of events, before and after these events have taken place, and incorporate a variety of devices and infrastructure facilities that are needed for this, including mobile phones and telecommunications networks, PDAs- *Personal Digital Assistant*, cameras, notebooks, wireless hotspots and others. (Foster & Douglis, 2003)

As discussed in the above sections, Grid computing is at its development stage; it could just be a beginning of a miracle and take-up of disruptive technologies leading to a new era of inter-networking and knowledge sharing.

Robert Filman (2004), in his article on *Days of Miracle and Wonder*, described this era as the beginning of a new paradigm change in the inter-network concepts and functions. In his explanation:

The Internet was the disruptive technology of the nineties. Travel agents, customer-support personnel, booksellers, newspaper publishers, and countless others found the Internet making their old ways of doing business obsolete. The coming disruptive technology will be an epidemic of networks of sensors (and actuators). Devices that measure the environment, process their measurements, communicate the results, and sometimes invoke actions will be pervasive. Sensor nets will ration water, nutrients, and pesticides in agriculture; monitor and control manufacturing processes; detect and guide fire and disaster fighting; monitor structural and earthquake damage; guide autos to less-travelled roads; measure and predict the weather on Earth and other planets; route communications traffic; check and replenish inventory; monitor and optimise habitat environments; track animals; and, most invasively, monitor people's health and movements. Networks of sensors will rush an ambulance to a heart-attack victim, identify who planted the bomb in the baby carriage, and warn a rental-car company of a traffic scofflaw. Like public health, telecommunications, and the automobile before them, sensor nets will be a vehicle of social transformation.

Therefore, resources must be made available to design, build and maintain Grids that are of *high capacity*- rich in resources, *of high capability*- rich in options, *persistent*- promoting

stable infrastructure and a knowledgeable workforce, *evolutionary*- able to adapt to new technologies and uses, *usable*- accessible, robust and easy-to-use, *scalable*- growth must be a part of the design, and able to support/promote new applications.

2.2.3 - Semantic Grids

The semantic web technologies and pervasive computing are converging to address the ever-increasing complexity of the network resources, knowledge sharing and research problems through Grid computing applications.

Goble & Roure (2002) discussed the following main reasons why the Semantic Web researchers are interested in the Grid computing application to optimize the exploitation of the available computing capacity and resources sharing:

- It is a very good example of the type of application envisaged for the Semantic Web. The essence of the Grid is the power provided by large-scale integration of resources, and the scale and automation of the Grid necessitates the universally accessible platform that allows data to be shared and processed by automated tools as well as by people.
- It is a real application: the emphasis is on deployment and on high performance, and is on a large scale and has established communities of users. Such applications are essential to the uptake of the Semantic Web.
- The Grid genuinely needs Semantic Web technologies. Even at the most basic level, Grid developers acknowledge that ‘information islands’ are being created and require an interoperability solution at information level such as provided by grid middle ware at data/computation level.
- It will stress Semantic Web solutions, and it raises some specific grid-related issues, which will provide a useful challenge. Solutions to these issues are unlikely to be peculiar to grid computing. Related issues will surely be evident in other Semantic Web applications in the fullness of time.
- It is self-contained, with a well-defined community who already work with common tools and standards.
- Aspects of the Semantic Web could be applications of grid computing, for example in search, data mining, translation and multimedia information retrieval. The partnership between the Semantic Web and the Grid presents an exciting vision.

Each partner has obstacles to its progress, but each stands to benefit from the other. To be successful, the partnership requires disjoint communities to come together. If they do, we can look forward to the ‘next generation’ of the Web: one with tremendous power to enable a new paradigm in science and engineering, which is termed as *Semantic Grid*.

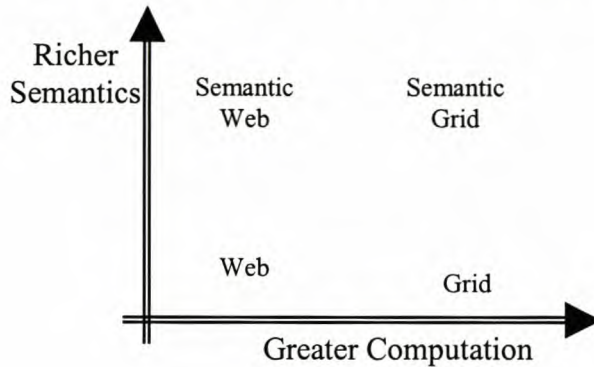


Fig 2.2 - The Semantic Grid

As Steven Newhouse described in the *Laying the Foundations for the Semantic Grid*, we may consider the movement towards a Semantic Grid at both the service and knowledge layers as essential in simplifying the effective utilisation of sophisticated distributed services. The description of these services using existing web-service protocols will enable their intelligent composition and exploitation with minimal human interaction. The transparent and optimal delivery of sophisticated computational and data services to the applied science community will be the key to the successful adoption of e-science.

Equally important is the role of sensor devices and communication units in pervasive and distributed computing. As Robert E. Filman (2004), in his article on *Lessons from System Development*, mentioned about the NASA-launched robot vehicles- the Mars Rovers, which were sent to land on and start investigating Mars, the whole system and functions of the robots well exhibit the features of pervasive and distributed computing. They send panoramic camera images through radio to scientists on the earth that present possible exploration targets. Group of scientists from different parts of the world, working round the clock, communicate back and forth with the robots to guide the movements of the robots to particular sites and use their sensors and communication units- Panoramic camera, miniature thermal-emission spectrometer, mossbauer spectrometer, alpha particle X-ray spectrometer, magnets, microscopic imager and rock-abrasion tool, together data. All the data collected was also being analysed and activities for the next day/move had to also be determined by the scientists. This whole process, therefore, involved a bunch of sensor devices and instruments

that took part in the data collection, processing, communicating between each other and with the earth station and perform automated tasks.

Generally, the development of Grid computing is also about accessibility of the bigger computing power as much as the ability to virtually organize this immense computing power.

2.3 - Latest Developments in the Grid Research

2.3.1 – Required development efforts for the Grid system

The NGG2 (Next Generation Grid) expert group in its latest assessment, as reported by Keith Jeffery (2004), considered the stage of research and development reached in this field of Grid computing and concludes that future efforts should deal with:

- development of a design for a new operating system that provides a fault-tolerant, scalable, self-healing, self-managing environment upon which Grids service middleware may ‘sit’
- development of Grids foundations middleware suitable both for enhancing existing operating systems and for inclusion within the above mentioned new operating system design
- development of Grids service middleware in a modular fashion allowing applications to utilise those services they require;
- research and development in computer science and information technology required to accomplish the above mentioned tasks, notably new models and software for transactions and messaging; for scheduling, resource management and optimisation; for trust, security and privacy; for data, information and knowledge management; for software development and deployment including mobile code; and for intelligent and appropriate user interfaces and device interfaces;
- development of novel applications that are wealth-creating or improve the quality of life, particularly in the e-business domain, but also in e-health, e-environment, e-culture, e-science, e-government

2.3.2 – Opportunities Promoting Grid development

Ample opportunities exist for the development of the next generation inter-networking, the Grid. These include:

- *Paradigm shift towards IT services:* Today's trend in the IT market in shifting revenues from the sales of products towards the provision of on-demand services creates unprecedented opportunities to develop new business models based on shared and distributed network services providing optimised capabilities to the user communities.
- *Operating system virtualisation:* The adoption of Web Service technology by the major OS vendors allows the development of distributed applications that are independent of the underlying operating system and language technologies. The convergence between Grids and Web Services therefore provides a significant opportunity to move to a model of software development and service provision where the market dominance of particular OS vendors is no longer a major economic issue.
- *Service Model for Industry:* The Grid today is used most heavily in particle physics, environmental science, life science applications, genomic research, protein folding, and medical applications, in advanced engineering R&D, in chemistry and materials science. It is expected that business, like finance or media, and many industries, such as aerospace, automotive, or entertainment will seize the opportunity to use existing IT resources more efficiently. Grids offer a new range of service models for the service industry.
- *Standardisation:* opportunity exists for the efforts to develop the Grids foundations / middleware in such a way that businesses, industries, science, healthcare, environment, culture, education etc... may have an interoperating advantage.
- *Next generation Grids:* The distinctive vision of Grids operating from the level of devices to supercomputers, to serve communities ranging from individuals to whole industries, could have a significant economic and social impact far beyond the scope of existing computing and data Grids.

2.3.3 – Potential Threats to Grid Development Efforts

There are, however, potential threats associated with these Grid research and development efforts. Keith Jeffery (2004) mentions the following:

- *Dependency on development tool support:* Support for interoperable messaging protocols (such as SOAP) depends on the tools provided by the various language and OS platform owners. While at the moment there is agreement on the overall direction of Grids middleware and Web Service evolution, disputes or changes in policy over

supported technologies could have a rapid impact on the ability of Grids developers and service providers to support particular language and operating system combinations.

- *Standards evolution:* As Grid technologies mature and become more complex, the adoption of standards (official or de facto) will be a requirement for sustained development of Grids, and only applications compatible with those standards will gain widespread adoption. It is vital that any vision for the evolution of Grids is accompanied by a clear representation of that vision to the key standards bodies and technology providers worldwide.
- *Non-acceptance and Lack of Use:* Industry may not accept the developed / developing middleware leading to a non-interoperable environment thus reducing the potential market size and the advancement of the knowledge society.

Berman, Fox & Tony (2002) also mentioned the following main threats that need to be addressed for an effective Grid Computing development:

- *Adaptive:* on the Grid, the choice of the machine, the network and other component impacts greatly the performance of the program. This variation in performance can be leveraged by systems that allow programs to adapt to the dynamic performance that can be delivered by Grid resources. Adaptive computing is an important area of Grid middleware that will require considerable research over the next decade.
- *Autonomic:* Both the nodes of the Grid and their organization must be made robust – internally fault-tolerant, as well as resilient to changes and errors in their environment. Ultimately, the Grid will need self-optimizing, self-configuring, self-healing and self-protecting components with a flexible architecture that can adapt to change.
- *Grid programming environments:* currently, efforts to develop viable programming environments for the Grid are limited to just a few forward-looking groups. In order for the Grid to be fully usable and useful, this state of affairs will need to change. It will be critical for developers and users to be able to debug programs on the Grid, monitor the performance levels of their programs on Grid resources and ensure that the appropriate libraries and environments are available on deployed resources.

2.3.4 – Properties of Future Grid Developments

Having understood all the existing opportunities, potential threats and efforts needed for the successful implementation of the Grid computing, certain corresponding properties are expected to be integrated/included in the future developments. Keith Jeffery (2004), described that a Next Generation Grid environment should have the following properties in order to satisfy the requirements of the scenarios considered:

- *pervasive*, with mobility as the cornerstone enhanced with more advanced pervasive computing facilities
- *self-managing* with the ability to handle highly dynamic and unpredictable configuration of demanders and suppliers
- *resilient* with the ability to handle highly dynamic and unpredictable configuration of the network connecting the computing nodes, and associated synchronisation of information sources
- *flexible* to handle various types of computing nodes and highly dynamic distribution of computation tasks among involved resources
- *easy to program* with a high-level, functional programming interface reusing existing software modules
- *flexible in trust* to allow business operations to work effectively and efficiently as virtual organisation and distributed collaborations
- *secure* to assure confidence in its use for business purposes

The following crucial features could be identified for the optimised performance of the Grid Computing. These include:

- self-adaptive, self-healing , self-managing and self-reconfiguring
- more sophisticated role-based security and trust between operating system instances or components
- extended in the sense of business continuity
- scale-independent
- open for interoperation – cooperating operating systems or components

- extended with the concept that the OS should be modular so that minimal configurations can be used without sacrificing interoperability
- a clear and open interface for Grids Foundations Middleware to Grids Service Middleware
- extended in the sense of context-aware geographically, temporally and role-based
- re-use of standards in operating system components to encourage interoperability and to provide a consistent interface to Grids foundations
- appropriate power consumption and code-size for the Grids entity (e.g. nano device)

Interestingly enough, two major trends of the Grid computing are mentioned by Stodghill, Heber & Lifka, in Goth, Greg (2004), which will be responsible for the new post-cluster paradigm. These are:

- The improvement of very high bandwidth networks between institutions using a *Dense Wavelength Division Multiplexing (DWDM)* network capable of transmitting up to 40 simultaneous light wavelengths (λ or waves), each with 10Gbps data transmission capacity. Such a high bandwidth network, the NLR (National Lambda Rail), has been deployed connecting Chicago and Pittsburgh, USA.
- The adoption of standardized middleware, most probably web services, enabling users in a wide occupation range to tap into High Performance Computing (HPC) clusters at widely spread locations. Lifka further adds that the advent of such web-services enabled distributed architecture will open HPC beyond scientific research and into industry.

Goth (2004) also describes that researchers in the Research Triangle Park Area, USA, have also developed a new provisioning protocol, called Just In Time (JIT), which can greatly speed up the transmission of large data blocks. JIT enables files to be sent without being converted back and forth from optics to electronics at each hop. Only the traffic control information is converted at each hop. The resulting decrease in the latency lets more applications share resources simultaneously.

From the above discussions, the Grid vision is absolutely critical to future advances of science and society. However, vision alone will not build the Grid. The promise and potential of the Grid must drive agendas for research, development and deployment over the next move.

Chapter Summary

Grid computing is a network of virtually organized computation, data and storage services where users get these Grid services *on demand* without having to own the physical assets. Hence, it helps companies to make a more synergistic use of often underutilized assets they already own. Moreover, Grid computing allows coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. A Virtual Organization (VO) was defined in the introductory section as a set of users in multiple network domains who wish to share some of their resources. This resource sharing includes direct access to computing cycles, software, data, storage and other resources as required by a range of collaborative problem solving and resource brokering strategies emerging in industry, science and engineering.

In the coming move of inter-networking, therefore, the semantic web technologies and pervasive computing are converging to address the ever-increasing complexity of network resources, knowledge sharing and complex research problem solving through Grid computing applications. This will involve different enterprise networks, research centres, sensors and sensor nets, delicate and expensive scientific instruments, and the Grid middleware to manage all the Grid resources. Therefore, this capability of the new generation inter-networking could enable a closer and better understanding of our environment; promote a favourable environment for advanced data and/or computing intensive research in science and engineering. Moreover, it could open a new vision to possibilities of new methods of sharing knowledge/expertise and collection, analysis and sharing of data across physical, material or capital boundaries. Hence, over time, this new technology will transform *computation, communication, and collaboration* between different systems, businesses and users, thereby transforming societies and the nature of science and business.

Future Grid development is expected to evolve into a technology, which is: pervasive, self managing, resilient, flexible to handle various types of components and services, easy to program, reliable and secure, easy to use, scalable and interoperable. All these are essential features, which promote Grid technology development.

Grid computing could be developed to serve a specific domain of users or to extend across the globe. Hence, the following chapter discusses the different scopes and architectures of Grid computing and the generic components it incorporates.

Chapter 3

Types of Grid Environments- Scope and Architecture, and Components of a Generic Grid Framework

3.1 - Introduction

Grid infrastructure will provide us with the ability to dynamically link together resources as an ensemble to support the execution of large-scale, resource-intensive, and distributed applications. (Berman, Fox & Hey, 2003:9)

Grid computing incorporates different components, as discussed in the respective sections below, from different perspectives and is designed to address problem areas that span a range of levels. Various types of Grid environments are also available that suite specific requirements of the user community.

This chapter presents these different patterns of Grid computing that theoretically describe the components required to establish a generic Grid computing model.

3.1.1 - Virtually Mobilizing Grid Resources

As described by Kelly, Roe & Sumitomo (2002), the idle cycles of networked PCs is increasingly being recognized as a huge and largely untapped source of computing power in virtually mobilizing the distributed, idle computing power in the Grid system. The providers of this service are the PCs whose cycles are temporarily being donated by their owners. This distributed processing capacity can be virtually organized by the Grid management middleware in two ways:

The first method, in Hwang, S. [et al.] 2003, is described where a PC is considered idle if its keyboard or mouse has not been touched for a specific period of time, in his case 2 minutes, otherwise it is regarded as busy. During this idle period, the PC's full processing capacity is donated to the Grid system until it gets busy again and the whole capacity is released back.

In this case, the client software installed in the PC continuously checks the status of the machine according to the settings and lets the machine donate its resources to the Grid system whenever it is idle. However, this system has a drawback in that a PC, which it regards as busy, does not usually utilize its full capacity. For example, a powerful Pentium IV, 2.5GHz PC, when used for playing cards or simple word processing, is regarded as busy while it actually is utilizing even less than 10% of its processing capacity. Therefore, the extra

processing capacity of any underutilized resource needs to be extracted and mobilized. The other method, however, addresses this issue as discussed next.

In the second case, many CPU scavenging software do mobilize every extra cycle which is apparently idle even while the PC is being used. In this case, the client software which harnesses the extra processing capacity checks the level of CPU utilization of the host computer and lets the PC offer/donate the extra cycles to the Grid system, which keeps the CPU work at a full capacity. Also, whenever the PC is not being utilized locally, the client software lets the whole processing capacity be donated to the Grid system. This is the most efficient way of virtually organizing the available processing capacity, where it releases the required amount of cycles from those donated as extra capacity to the local use, whenever additional capacity is needed by the donating machine/PC locally. A typical example of this enabling software could be the *GreenTea* – a pure Java based peer-to-peer Grid platform that facilitates Grid services by harnessing idle/extra computing resources on the network. (GreenTea Technologies Inc [Online])

Generally, the Grid middleware is a means of connecting the available idle computing resources on the one hand and users requiring more computing power than they can get locally on the other. Hence, virtually organizing the idle computing capacity, as discussed above, aims at polling the client machines for free idle capacity and harness/virtually organize the available power, and seamlessly synchronize to the pool of computing capacity mobilized from all the network resources in the Grid.

On the other hand, this middleware handles the users' requests for more capacity to process their jobs. It receives job requests from authorized users, allocates them to the freely available computing resources, follows up the processes till they are completed and provides the results back to the respective users. In these activities, the middleware is responsible for addressing any faults/failures by the resource, which is processing any particular job; hence, it includes fault tolerant features where any job that is reportedly failed is sent to another computing resource for proper processing.

3.1.2 – Conceptual Layers of Grid Computing

At this time, there are a number of grid applications being developed and there is a whole raft of computer technologies that provide fragments of the necessary functionality. However, there is currently a major gap between these endeavours and the vision of e-Science in which there is a high degree of easy-to-use and seamless automation and in which there are flexible collaborations and computations on a global scale. To bridge this practice–aspiration divide,

research efforts should aim to move from the current state of the art in e-Science infrastructure to the future infrastructure that is needed to support the full richness of the e-Science vision. Here the future e-Science research infrastructure is termed as the *Semantic Grid- Semantic Grid to Grid is meant to connote a similar relationship to the one that exists between the Semantic Web and the Web.*

The Semantic Grid is characterised as an open system in which users, software components and computational resources (all owned by different stakeholders) come and go on a continual basis. There should be a high degree of automation that supports flexible collaborations and computation on a global scale. Roure, Jennings & Shadbolt, in Berman, Fox & Hey, (2003:437-470)

Semantic Grid is a centre without walls, in which researchers can perform their research without regard to geographical location - interacting with colleagues, accessing instrumentation, sharing data and computational resource, and accessing information in digital libraries. We extend this view to accommodate *information appliances* in the laboratory setting, which might, for example, include electronic logbooks and other portable devices.

As described by David De Roure in Berman, Fox & Hey (2003:238), Keith G. Jeffery of CLRC introduced a three-layer grid vision in a paper presented for the UK Research Councils Strategic Review in 1999. These three conceptual layers that characterise the computing infrastructure are:

- *Data/computation* This layer deals with the way that computational resources are allocated, scheduled and executed, and the way in which data is shipped between the various processing resources. It is characterized as being able to deal with large volumes of data, providing fast networks and presenting diverse resources as a single meta-computer. The data/computation layer builds on the physical 'grid fabric', i.e. the underlying network and computer infrastructure, which may also interconnect scientific equipment. Here data is understood as uninterrupted bits and bytes.
- *Information* This layer deals with the way that information is represented, stored, accessed, shared and maintained. Here information is understood as data equipped with meaning. For example the characterization of an integer as representing the temperature of a reaction process.
- *Knowledge* This layer is concerned with the way that knowledge is acquired, used, retrieved, published and maintained to assist e-Scientists to achieve their particular

goals and objectives. Here knowledge is understood as information applied to achieve a goal, solve a problem or enact a decision. In the Business Intelligence literature, knowledge is often defined as actionable information; for example, the recognition by a plant operator where, in the current context, a reaction temperature demands a shutdown of the process.

In the above description, there are three basic issues that need to be considered. Firstly, all grids have some element of all three layers in them. The degree to which the various layers are important and utilized in a given application will be domain dependent. Thus, in some cases, the processing of huge volumes of data will be the dominant concern, while in others the knowledge services that are available will be the overriding issue. Secondly, this layering is a conceptual view on the system that is useful in the analysis and design phases of development, while it may not strictly apply to the implementation for reasons of efficiency. Thirdly, the service-oriented view applies at all the layers. Thus, there are services, producers, consumers and contracts at the computational layer, information layer and knowledge layer.

As shown in Fig 3.1 below, all the service layers are interlinked with each other so that either of them doesn't stand without a proper organization and/or application of the others. The E-Environment, as the user of the services, also interacts directly with each layer. Hence, the user benefits from data/computational services of the Grid, intermediate shared information, and the knowledge and expertise sharing services enabled by the Grid computing.

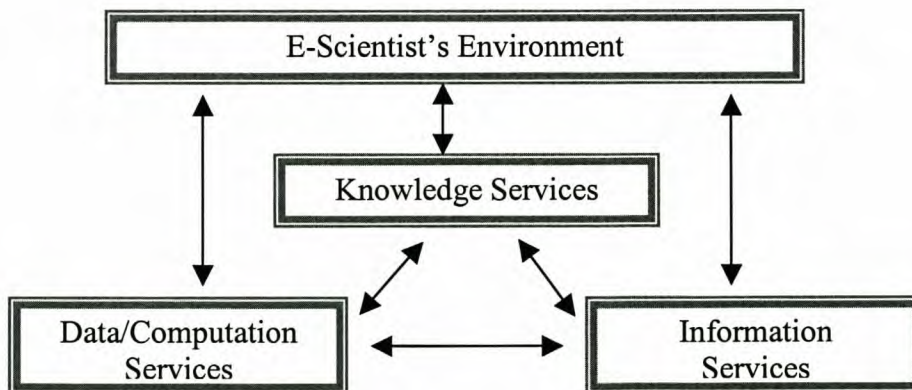


Fig 3.1 - The three-layered Architecture viewed as services Adapted from Roure, Jennings & Shadbolt, in Berman, Fox & Hey (2003:439)

3.2 - Scope and Components of the Grid Computing

3.2.1 – Scope of the Grid Computing design

With the grand aim of virtually organizing computing capacity, Grid computing can be designed to be deployed within an enterprise, that is, behind a firewall, or it can work across firewalls between enterprises and/or individual users on the net. The middleware used will also differ accordingly.

A good example for total resource aggregation and resource sharing platform for enterprise grids (Cluster/Campus Grids) is the ‘GreenTea’ software.

GreenTea software is a pure Java, peer-to-peer Grid platform that facilitates P2P computing, distributed computing, Grid computing and network computing by harnessing the idle computing resources on the network. (GreenTea Technologies Inc. [Online])

GreenTea runs on any Java enabled computing platform- Windows, UNIX, Linux, OS.X, and by using this software, all the computers in the organization can be utilized to form a large virtual supercomputer. The resources it mobilizes include both hard resources (CPU, RAM, Hard Disk, Network Bandwidth, etc.) and soft resources (files, software applications and services, etc.). Besides, GreenTea enables give-and-take sharing of computing resources bi-directionally and/or many to many.

On the other hand, this software enables users of different platforms to work together sharing their resources through the same medium. It has a strong performance because it can exploit a higher bandwidth of the intranet, probably in Gbps (Gigabits per second), for optimized performance compared to the resource sharing through the Internet infrastructure. For a more efficient performance, GreenTea needs to install client software on the user PCs.

Another scenario is discussed by Kelly, Roe & Sumitomo (2002) where Grid resources need to be mobilized across enterprises through the Internet, a more robust and flexible software is needed to manage the virtual resource organization. Efficiently performing this task is the Microsoft’s *.NET* Common Language Runtime. Unlike the GreenTea that is entirely dependent on Java, *.NET* runtime is designed to be language independent, hence, it can run applications created using any of the languages- C#, Visual C++, Visual Basic or one of the many third party *.NET* languages such as Component Pascal, Eiffel, Perl, Smalltalk, Fortran or Cobol.

Generally, the .NET is a language independent, cycle stealing package and operates over the conventional web services through http requests.

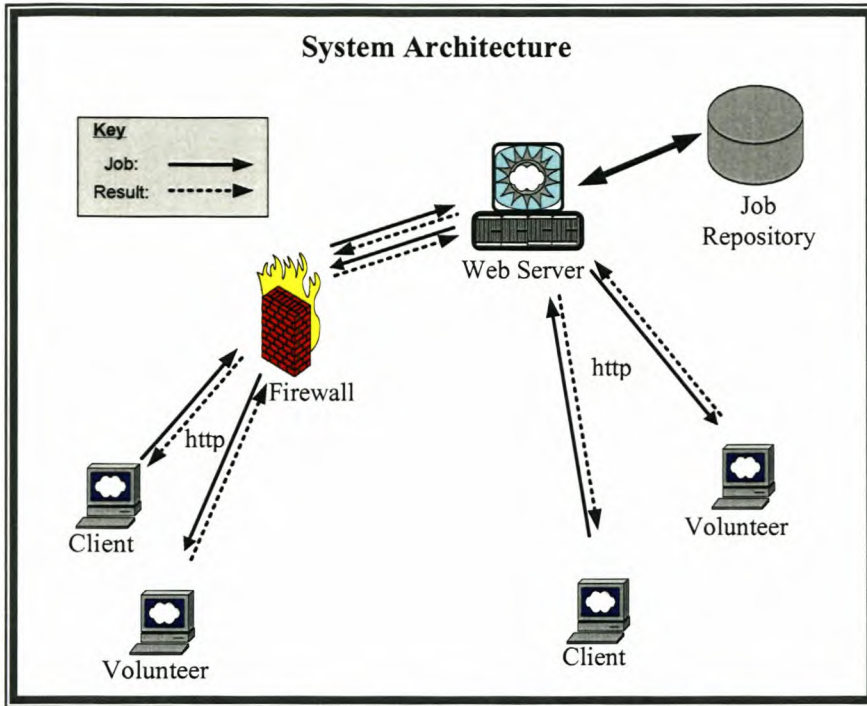


Fig 3.2 – Grid System Architecture:

A Web Server mediates the communication between Grid users and resources enabling Grid nodes behind firewalls to properly function using HTTP.

As shown in figure 3.2, clients and volunteers do not need to be connected directly peer-to-peer, hence, in case where a firewall exists, the enterprise web server mediates the communication by receiving job requests and sending back results on the client side, and letting the volunteer machines pull jobs to be processed, and receive results for each job. All communications between the clients and volunteers are made by using web services protocol, *Simple Object Access Protocol (SOAP) over HTTP*.

Note that the speed of processing a job does not necessarily depend directly on the number of volunteers, but also on the *granularity* of the tasks and the *ratio of communication to computation*, that is, the greater the communication to computation ratio, the smaller the increase in the speed of performing the task.

Both the .NET and GreenTea are cycle stealing systems and they are discussed above to represent the different systems used for that purpose. Other cycle stealing systems include:

- *Condor*: uses automatic check-pointing and task migration to load-balance the tasks amongst the fixed set of possible volunteers.
- *Piranha*: supports adaptive parallelism using a model to decouple computation from processing
- *BOINC- Berkeley Open Infrastructure for Network Computing*: used by Grid projects like Seti@home, taps every extra cycle available even while the client machine is working at under-capacity level and mobilizes it for use in the Grid system

Other Internet based cycle stealing systems also include: GIMPS, Charlotte, the Knitting factory project, CX projects and Parabon.

All these systems focus on optimized cycle stealing where CPU-intensive projects are the main concern of the Grid community. Moreover, they shield the application programs/users from the changing set of resource donating machines.

Grid Systems such as Globus and Legion, however, provide an interface through which distributed computing resources such as supercomputers, scientific instruments and archive data storage can be easily shared and used across organizational boundaries.

The main drawback of the .NET runtime is the current architecture of the middleware to work on a single server implementation, which is obviously a hindrance to scalability.

3.2.2 – Types and Components of Grid Computing

Grid computing is gaining a lot of attention in the business environment, although it was mainly used in the academic and scientific community for some time.

Initially, businesses need to develop their own ‘Intra-Grids’ in a standard framework and, over time, these internal Grids can be interconnected to form a global resource sharing environment across enterprises. Grid computing standards make this analogy of global resource sharing a reality.

When planning to develop a Grid computing for an organization, the type of Grid environment to be used will affect the level of utilization of the shared resources and decision making in general. With regard to this issue, Grids can be categorized into three categories by the type of solution they best address, as discussed in Daniel Minoli (2005:13). These are:

- *Computational Grids*: the central aim of this type is provide strong computing power and resources are set aside for this purpose. These resources are mostly high-

performance machines/servers and supercomputers, which are allocated to ‘number crunch’ data or provide coverage for other intensive workloads.

- *Scavenging Grids*: this type of Grid mostly deals with a large number of desktop PCs. These PCs are scavenged for idle cycles available and other resources. Note that the owner of the scavenged resource is granted full control over the resource and is released seamlessly whenever more power is needed for local processing.
- *Data Grids*: this type of Grid computing is designed to host data and manage the access and security of the data across multiple organizations and users. It provides a unified interface for all data repositories in an organization, and through which data can be queried, managed and secured.

Besides, wireless Grids could also be regarded as one type of Grid computing system. Wireless Grids constitute mobile, nomadic and fixed location systems, temporarily connected via *ad hoc* wireless networks, as described in Mcknight, Howison & Bradner (2004). Note that other than the fixed location wireless devices, these devices include not only mobile but also nomadic devices shifting across organizational boundaries.

There is no hard and fast rule, however, with regard to which Grid computing type to adapt as a particular appropriate Grid design can take a combination of both the types.

Grid computing system at all levels is made up of three components:

- The Grid resources which should be available for use in the virtually organized pool of computing system
- The users/clients, who are authorized to exploit the available Grid resources from the pool, and
- The Grid middleware that provides the infrastructure/medium over which both the users and the network resources communicate

The Grid resources could be fixed and be ready for use as long as they are functional, or they could be offered on voluntary basis, thus, continuously varying sets of resources are offered seamlessly by the Grid system. Although the nature and methods of generating the Grid services differ, both the fixed and voluntary services do function through a Grid Middleware that seamlessly organizes these available resources and shields the users from the changes in the set of available resources generating the service

The Users of the Grid system could also be on Ad-hoc/temporary basis, community membership or on a permanent 'Give and Take'. These different types of memberships do apply to different Grid architectures, which will be discussed in the next section (Section 3.3).

There is no a universal consensus yet on what the canon components of a Grid should be. There are different fundamental building blocks from different views- functional, physical and service views. Daniel Minoli (2005:71-100) discussed these viewpoints in more details, and here follows a summary of his presentation:

Functional View: from this point of view, the basic components include-

- *Grid Portal:* provides the user with an interface to launch applications from the virtual computing resources
- *Grid Security Infrastructure:* is a key requirement for Grid computing, which provides mechanisms for authentication, authorization, data confidentiality, data integrity and availability, especially from the users point of view
- *Broker:* This functionality searches and provides a matching Grid resource with the application launched by the authenticated user.
- *Scheduler:* this is used to coordinate the execution of jobs where the user wishes to reserve a specific resource or to insure that different jobs within the application run concurrently.
- *Data Management:* provides a reliable and secure method for moving files and data to various nodes within the Grid.
- *Job and Resource management/Grid Resource manager:* provides the services to actually launch a job to run on a particular resource, to check the job's status and to retrieve the results when the job is complete.
- *Grid Resources:* these include processors, data storage, memory, bandwidth, scientific equipment etc. These resources need to be physically (through inter-connecting networks) and logically (through the Grid support software) available to run the Grid applications properly.

At the core of the Grid system are standard protocols that interconnect all the above-mentioned functional blocks/components.

Protocols are formal descriptions of message formats and a set of rules for message exchange. (Daniel Minoli, 2005:80)

Physical View: from the physical point of view, Grid computing is a collection of networks, processors, storage and other resources as mentioned below:

- *Networks:* Networks provide the most fundamental resource for the Grid. More importantly, the recent growth in communication capacity (network bandwidth, hardware and software) makes the Grid computing practical and possible.
- *Computation:* this constitutes the next most common resource in the grid computing where distributed computing cycles are virtually mobilized to form a powerful computing pool regardless of their speed, architecture, software platform or storage apparatus they are locally attached to.
- *Storage:* this is also another most important resource where an immense storage capacity can be provided by making use of the storage on multiple system units with a unifying file system. Hence, a file or database can span several physical storage devices bypassing size restrictions often imposed by the local system.
- *Scientific Instruments:* particularly Inter-Campus Grids or Global Grids can provide shared access to expensive scientific equipment, or interconnect geographically dispersed equipment into a large overall scientific tool.
- *Software and Licenses:* Grid software applications should be enabled to run on an available processor on the Grid. This has to do with scalability and how efficiently the multiple processors in the Grid work. However, scalability has also limitations in the case of licensed software, which could be too expensive to install in a large set of processors. In this case, the particular licensed software can be installed in some selected processors and the jobs requiring this software are routed to these particular processors. Hence, this way the Grid can save significant expenses for the organization.

The Service View: from the service point of view, standard protocols and easy interfaces constitute the main components of Grid computing. A good interface design must enable smooth routing and execution of jobs as well as proper management of transmission of data and files. Besides, standardized protocols also constitute a core service component enabling all the Grid resources to communicate with each other and insure inter-operability. Generally,

a Grid computing should enable its virtually organized resources and interlinked components to provide their optimized services and live up to its promises. Hence, it should provide the Grid users with parallel processing and distributed job execution, computing power and storage space sharing, and sharing expensive scientific instruments thereby transforming communication, coordination and collaboration.

In a different Grid architecture, like that of the Scandinavian Production Grid- in Eerola, Paula [et al], 2003, the Grid system includes:

- *User Interface*: communicates with the Grid manager and queries the information system and replica catalogue
- *Information System*: consists of a dynamic set of distributed databases, coupled to computing and storage resources to provide information on specific resource status
- *Computing Clusters*: consists of a front-end node that manages several back-end nodes typically through a private closed network. The Grid middleware doesn't dictate the local batch system configuration, but adds on a component that hooks local resources into the Grid.
- *Storage element*: allows access and control based on the users' Grid certificates rather than their local identities
- *Replica Catalogue*: registers and locates data sources to be used by the Grid manager and user interfaces.

Generally, the specific Grid architecture adopted determines the components needed for a particular Grid Computing network that suits the main objectives of the particular system.

Daniel Minoli (2005:18) mentioned the fundamental components of Grid computing as follows:

- ❖ *Resource Management*: the Grid must be aware of what resources are available for different tasks
- ❖ *Security management*: the Grid needs to take care that only authorized users can access and use the available resources
- ❖ *Data management*: data must be transported, cleansed, parcelled and processed
- ❖ *Services Management*: users and applications must be able to query the grid in an effective and efficient manner

All the above-discussed components of the Grid interact in logically hierarchical layers as discussed in the next section.

3.2.3 – The Layers of Grid Computing

Foster, Kesselman & Tueck, in Berman, Fox & Hey (2003:178-185), described the various layers of the grid architecture in relation to Internet protocol architecture as follows:

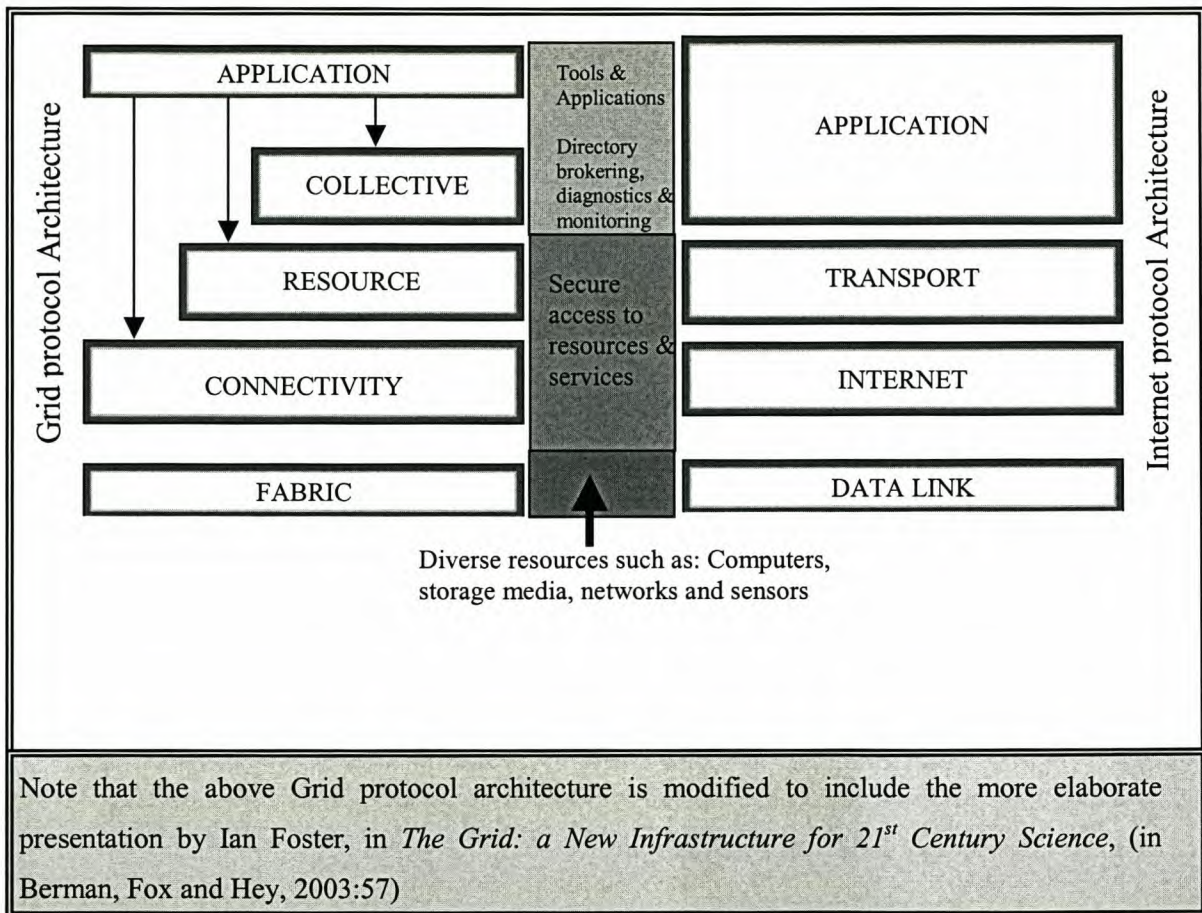


Fig 3.3 - The layered Grid protocol architecture, compared to the Internet protocol architecture

As shown in the above figure, the Grid Protocol architecture layers can be discussed as follows:

- *The Grid Fabric Layer:* provides the basic infrastructure and resources- computational resources, storage systems, catalogue, network resources and sensors, on which Grid applications are run, mediated by Grid protocols from the upper layers
- *The Connectivity Layer:* defines core communication and authentication protocols required to enable the exchange of data between Fabric layer resources and to provide a secure mechanism for verifying the identity of users and resources respectively.

- *The Resource Layer*: builds on Connectivity Layer protocols to define protocols and, APIs (Application Programming Interface) and SDKs (Software Development Kit) for the secure negotiation, initiation, monitoring, control, accounting and payment of sharing operations on individual resources.
- *The Collective Layer*: contains protocols and services that are not associated with anyone specific resource, but rather are global in nature and capture interactions across collections of resources, that is, coordinating multiple resources.
- *The Application Layer*: comprises the user applications that operate within a virtually organized environment, and they are constructed in terms of, and by calling up on, services defined at any layer.

Ian Foster, in *The Grid: a New Infrastructure for 21st Century Science*, describes that the *connectivity* and *resource* layers in the Grid Protocol Architecture correspond to *network* and *transport* layers of the Internet Protocol Architecture. From his explanation, the connectivity and resource layers are the ones, dealing with secure access to resources and services, which are similar to the functions of internet and transport layers in the internet protocol architecture, which ensure a reliable link and transport of the sessions between applications using TCP/IP protocol.

3.3 – Grid Computing Architecture

Different Grid computing architectures are developed to suit different computing needs. The types of network resources needed to be shared and the composition of users/clients determine which Grid architecture to adopt. Some of these are discussed in the following sections, as adapted from Esparza, O. et al [Online].

3.3.1 – Grid over VPN

GoVPN is a possible solution for building a virtual grid environment with dedicated resources. In GoVPN, first a VPN is created and then a Grid is deployed over it. The private environment is created at the network level so that users outside this environment do not have network connectivity to the resources. *Virtual Private Networks* (VPNs) are discrete network entities configured and operated over a shared network infrastructure. (Chris Metz, 2003).

There are several mechanisms that VPNs use to enable private data going through a public network; the most important among them are authentication, privacy and tunnelling. IPSec,

TLS (Transport Layer Security) and SSH (Secured Shell) are ways of performing these mechanisms.

GoVPN has several advantages: it can provide warranted QoS at the network level, it relies on existing technologies and it allows seamless integration of legacy applications. However, GoVPN also has some drawbacks. Allowing the Grid level to become aware of changes in the network level and vice versa is tedious because Grid and network level are independently deployed. This fact limits GoVPN to pretty static environments. Moreover, each level (Grid and network) performs its own security services, leading to either duplicated schemes and/or mismatching security policies.

3.3.2 – Virtual Private Grid

This architecture provides a solution for building private grid environments comprising both network and grid levels. In this sense, VPG can be understood as a middleware interacting with these two layers. Hence, the middleware has to include two modules: one for managing network resources and another for controlling security.

In VPG each node is considered a member of a multicast group. When the virtual Grid environment has to be created, a secure multicast group also has to be set up. A node will be considered a member of the group if it knows the common secret shared by all members of the group- called *weak authentication*. The VPG-SA (VPG-Security Architecture) has the responsibility of delivering the secret key to every initial member.

The only way of performing this action is by means of a unicast connection with each one of the initial nodes of the virtual environment. As long as this action only takes place once, bandwidth and computation issues are not very relevant.

Multicast along with logical key tree based schemes can be used to improve the performance of security management in the Grid.

3.3.3 – Grid Community

Grid Community serves groups of communicating individuals or communities- such as Earth Science, Medical Science and Bioinformatics communities, High School Classes, etc. sharing resources implemented as Grid Services.

Grid Community (GC) and *Ad-hoc Grid* (AG) both create the private grid environment using only grid-level mechanisms. In GC each site grants coarse-grained access of its resources to a community account requiring the existence of a Community Authentication Server (CAS).

AG deploys a similar scheme but allows more spontaneous and short-lived collaborations. However, privacy and quality of service at the network level are not addressed by any of them.

As described in Ian Foster [et al] (2002), in Grid Community, resource owners grant access to blocks of virtual resources to a community as a whole, and let the community itself manage fine-grained access control within that framework using CAS servers.

3.3.4 – Ad Hoc Grid

The above discussed grid architectures support various applications with diverse scope and requirements, but they fail to support sporadic collaborations in the absence of a central regulating authority. Ad hoc grid architecture is, therefore, introduced motivated by the need to support such applications.

Ad hoc Grids offer a structure-, technology-, and control-independent Grid solution. Structural independence reflects the ability to self-organize among its participant peers. Technology independence reflects the ability to support multiple Grid protocols and technologies. Control independence embodies the ability to support administrative functionality without any central coordination. Applications changing members, policies, and requirements are well suited for ad hoc Grids. (Amin, Laszewski & Mikler, 2004)

Note that any Grid computing design implementation can incorporate more than one of the above-mentioned architectures, which best suit to the specific requirements of the institution/organization.

Apart from the choice of architecture, the development of a Grid computing environment is also determined by the size of the Grid itself. Therefore, different features and security considerations are involved at different levels, in terms of scope, as discussed in the following sections.

3.4 - Levels of Grid Computing

Sun Microsystems (2003) describes three levels of Grid computing- Cluster Grid, Campus Grid and Global Grid.

3.4.1 - Cluster Grid

Cluster Grids are the most popular and simplest forms of Grid Computing. Meeting the needs of most organizations, Cluster Grids consist of one or more systems working together to provide a single point of access to users. Typically used by a team of users such as a single

project or a department, a Cluster Grid can be used to support both high throughput and high performance jobs. The users of this Grid can share resources within the cluster through the Grid server, which connects all machines within the cluster. The class of software at the heart of the cluster Grid is the distributed resource management (DRM) system. Intranet transmission links or other high-quality, high-throughput, high-performance communication services are used to interconnect these nodal computing resources with the Grid.

3.4.2 - Campus Grid

Campus Grids enable multiple projects or departments to share computing resources in a cooperative way. Campus Grids may consist of dispersed workstations and servers, as well as centralized resources located in multiple administrative domains, in departments, or across the enterprise. Each machine is connected to its cluster grid server and the grid servers are interconnected. If a process requires more processing power, storage or applications than what's available on its cluster grid, then it is allocated to the much larger processing/computing capacity available on the campus grid. Generally at a cluster level, the access to the Grid resources may not require special security procedures or usage policy. However, interconnecting these clusters into campus/enterprise Grid needs a special attention with regard to harmonizing the policies of all the clusters and insuring security across the clusters. The issue of security becomes even more serious as the domain of the Grid computing system expands, that is, the *Global Grid*.

3.4.3 - Global Grid

When application needs exceed the capacity of a Campus Grid, organizations can tap partner resources through a *Global Grid*. Designed to support and address the needs of multiple sites and organizations, Global Grids provide the power of distributed resources to users anywhere in the world for computing and collaboration. They can be used by individuals or organizations sending overflow work to a grid provider, or by multiple companies working together and sharing data - crossing organizational boundaries with ease, hence, it can also be regarded as a collection of Campus Grids. Sun Microsystems, in the *Building a global compute Grid* [Online], also defines Global Grid as:

collections of enterprise and cluster grids as well as other geographically distributed resources, all of which have agreed upon global usage policies and protocols to enable resource sharing.

Generally, at the core of the Grid computing network is the Cluster Grid where the actual job is requested and, in most Grid architectures, processed. Then the Campus Grid is developed,

interconnecting these cluster Grids taking into consideration the new security issues and diverse resources. Global Grid can also be formed interconnecting Campus Grids and should incorporate advanced security systems and reliable accessibility and inter-operability issues. Therefore, it is important to understand how the Cluster Grid logically works. Note that individual nodes can also subscribe to the Grid computing at any level.

With regard to the nature of the Grid computing applications, *Platform Computing*, as referred in *Platform in the news archive* [Online], expects that a three-phased approach will take place for massive application of grid computing. These are:

- *Enterprise Grids* - These involve commercial implementation of production grids within major corporations having global presence and requiring a great need for resource access.
- *Partner Grids* - they cover project collaborations between organizations of similar industries or interest areas for reaching common objectives. Examples are: Life Sciences organizations and SETI@home.

SETI@home - Search for Extraterrestrial Intelligence. It is a distributed computing project for Internet-connected home computers. Its purpose is to analyze data incoming from the *Arecibo* radio telescope searching for possible evidence of radio transmissions. With over 5 million users worldwide, the project is the most successful example of distributed computing to date.

- *Service Grids* – refer to Grid computing available as utility or service in the open market. Ordinary users demanding grid services as a utility will drive this third phase.

3.5 – Issues of Security

Grid computing needs to address critical issues of security for reliability of its applications and services, while running on the public networks. At this early stage of the Grid computing, it is obvious to encounter security threats and loop-holes. However, it is crucial for the success of the Grid development that some fundamental security issues need to be addressed along with the development of the Grid system.

Yaacob & Iqbay (2003) mentioned that a typical Grid computing framework consists of authentication, sharing, coordination and synchronization.

- *Authentication*: is a process of establishing identity of the Grid user. This determines the right of the user to share/use Grid resources. Hence, the authorized user can have access to all available resources for which he is authorized.
- *Sharing*: this involves access to the authorized resources; hence, machines that are authorized do share the corresponding Grid resources over the Internet/Intranet. These Grid resources distributed all over the network medium need to be registered for their availability and then maintained and updated dynamically during computation.
- *Coordination and Synchronization*: this process manages dependencies between activities. A program which needs to run multiple machines due to lack of the required resource locally must be divided into independent sub-programs, and these sub-programs need to be coordinated and synchronized to complete the task in proper order.

Besides, the *Grid security architecture* should deal with the following fundamental principles, as discussed by Foster, I. et al, 2003 [Online]:

- The Grid security architecture must grant access to resources distributed across multiple administrative domains
- The inter-domain security solution used for Grids must be able to interoperate with the local security solutions encountered in individuals' domains, enforcing the support for multi-domain trust and policy
- A user should be able to authenticate once and subsequently initiate computations that acquire resources without the need for further user authentication (single sign-on)
- Multi-domain support and single sign-on will need the usage of a common way of expressing the identity of a principal such as a user or a resource, that is, a uniform credentials/certification infrastructure
- The Grid usually runs over a less-trusted network, thus security and data privacy should be assured at the Grid level. Currently, Grid toolkits provide some specific security features like- resource access control, user authentication, creation of a multicast channel, and hop-by-hop encryption of communication links instead of end-to-end security controls.

The encryption uses common key by all the nodes in the virtual Grid environment. The key should be known by all the nodes in the Grid and must be updated whenever the composition of the Grid resources changes. This proved a perfect forward and backward secrecy.

Grid Computing usually uses hop-to-hop encryption using encryption key, which is only understood by authorized members of the Grid system. Hence, any communication throughout the Grid is encrypted and requires any machine a key to decrypt and understand the content.

Generally, a good security algorithm should be adapted to minimize the bandwidth consumption during the security key updates, which could use key-tree structure, analogous to actual tree structure where the leaves stand for Grid Nodes and all the branches connected to the stem are the link-architecture between the nodes and the main Grid Middleware (management software).

3.6 - Potential Challenges of the Grid Computing

The fundamental research challenges in the Grid computing, as discussed by Keith Jeffery (2004), include:

- An increased user-centric focus on how applications of next generation Grids are manifested to the user via pervasive computing devices
- The additional information representation requirements for context and personal awareness, supporting proactive behaviour
- On-demand and timely presentation of information, requiring dynamic composition and negotiation of services. This creates challenges for negotiation, orchestration and scheduling
- Pre-emptive behaviour by Grids, which is related to autonomous behaviour, and contrasts with the traditional view of Grids as a batch processing system
- Valuable information representation on small devices, synthesis of knowledge models on wireless devices for ubiquitous use

Other main challenges/problems that must be solved to enable Grid development before it can be used extensively and live up to its promises include:

- *The Dynamic Nature of Applications*: latest developments in the nature of applications, being dynamic and extensive, posed a great challenge that demands dramatic changes and new ways of using computing that still require further research.
- *Programming Models and Tools*: these need to address the complex nature of Grid architecture and enable efficient interaction between the Grid users and shared

resources available. They need to adapt to changes in the networking environment and should provide latency tolerant and fault tolerant solutions, which require further research and refinement.

- *Resource Management*: sharing of resources in the dynamic and increasingly complex application environment introduces challenging resource management problems that require continuous research to accommodate changes to meet rigorous end-to-end performance requirements across multiple computational resources connected by heterogeneous, shared networks
- *Security*: Grid security systems need to adapt the local security policy of the shared resource over the general Grid security for the specific process, and they also have to address issues of license and accounting
- *Instrumentation and Performance Analysis*: Appropriate instrumentation, measurement and analysis tools and methods should be developed to gauge the performance and identify/address the particular problem area and be able to provide more reliable future Grid services
- *End Systems*: more research and development is required to come up with new end systems that are bigger and more complex and are able to work in a high-performance networking driven by future Grid computing architectures and latest network technologies.
- *Network Protocols and Infrastructure*: major advances and innovations in network communications and services to transport, route and managing the network traffic are required to enable the future Grid applications that are meant to work with high bandwidth and meticulous performance guarantee.

Especially the wireless Grid services area is posing the main challenge on the Grid research and development. New Grid applications are focused more on distributed services. Much of the data for these services are collected through small, self-contained intelligent devices that combine limited sensing and computation capabilities with wireless communications. For example, in many crisis management situations the commanders and the security personnel, along with medical teams, have to operate outside their facilities and all the relative built-in equipment. These scenarios have some specific requirements that have to be taken into consideration.

Enhancing of senses for the mobile users- the enhancement of senses mainly corresponds to:

- Optimised pictures, videos, and stereo sound analysis: Victims in a crisis situation often carry mobile devices that can be incorporated into the crisis management infrastructure (e.g. providing images and video clips of the surrounding area, being used to instruct victims or rescue personnel) that will optimise communication in stressful environments
- Dynamic access to information from co-operating mobile devices of the team members or victims for crucial real-time information, e.g. spatial and temporal coordinates of the victims and team members.

Intelligent decision support for both the individuals and the control centre- mobile units collect and process information locally at a preliminary level. Mostly, however, they are supposed to provide the information to a central unit in order to post-process it, perform simulation of the situation, extract useful information and statistics, and in the sequel to predict actions so as to support decisions for the pro-active handling of the emergency situation.

Distributed mobile ad hoc network- these critical situations require that the portable devices should operate independently from the central station. In cases where the central station has been damaged or is temporarily unavailable, the mobile devices should be able to co-operate so as to perform a set of services, thus providing tolerance in some emergency cases.

Access to remote databases- this aspect refers to the ability of the mobile staff to have access to remote information distributed over the world. For instance, in the case of a sports event involving tens of thousands of people, access to the medical records of each victim, regardless of where the victim comes from, will have a crucial impact on the results of the healthcare operation. This ability is especially needed in cases where time-critical situations must be handled and any delay through information mediators may influence the result of the actions.

At this time, there are a number of grid applications being developed and there is a whole raft of computer technologies that provide fragments of the necessary functionality. However, there is currently a major gap between these endeavours and the vision of e-Science in which there is a high degree of easy-to-use and seamless automation and in which there are flexible collaborations and computations on a global scale. To bridge this practice–aspiration divide, research efforts should aim to move from the current state of the art in e-Science

infrastructure to the future infrastructure that is needed to support the full richness of the e-Science vision.

Generally, Grid computing development faces many challenges in technology requirements, effective middleware development, security issues and meeting the ever-increasing demands for its services. The flexibility and efficiency of the Grid middleware has, however, the greatest potential to ease the potential threats and challenges along its development lines, and this deserves more attention for new ways of addressing problems.

Cal Robbens, a computer science professor at Virginia Polytechnic Institute, in Goth (2004), widely discussed the main challenges of the Grid development and concluded that:

The real hard problem with the Grid- the long-haul kind of thing is not only its bandwidth, but there is also a much harder software problem to coordinate that staff.

Chapter Summary

Grid computing virtually organizes distributed computing resources as an ensemble to support the efficient execution of large-scale, resource-intensive, and distributed applications in a dynamic environment.

Grid computing infrastructure can be characterised by three conceptual layers, which include : the *data/computation* layer that deals with the way computational resources are allocated, scheduled and executed, and in which data is shipped between the various processing resources ; the *information* layer that deals with the way information is represented, stored, accessed, shared and maintained ; and the *knowledge* layer that is concerned with the way knowledge is acquired, used, retrieved, published and maintained to assist the e-science environment. All these layers supplement each other to meet the service requirements by the e-environment, which span across all these different layers.

Moreover, Grid computing can be developed to work within an enterprise, that is, behind a firewall or across enterprises and in global scale involving resources across firewalls. Different Grid software tools/platforms are used in each case; hence, *GreenTea* makes a good example in the first case, and *Microsoft .NET* in the latter.

Three types of Grids are discussed in this chapter, which are categorized according to the type of solution they best address. These are: *Computational Grids* that provide strong computing power from resources that are set aside for this purpose, which could include: clusters, super

computers, dedicated data centres, etc.; *Scavenging Grids* that deal with a large number of desktop PCs for a pool of computing power- CPU cycles, storage and memory; and *Data Grids* that host data and manage the access and security of the data across multiple organizations and users. *Wireless Grids* are also progressively getting integrated into the comprehensive Grid computing, which constitute one type that deals with wireless devices/sensors connected via ad-hoc wireless networks.

With regard to the issue of security, Grid security architecture should follow fundamental principles, which include: synchronizing the general Grid policies with those of individual domains; allowing a single-sign-on access for authorised users to using different Grid resources.

Generally, as it is new and still in its development phase, Grid computing faces many challenges in technology requirements, effective middleware development, and increased focus on user-centric next-generation Grid applications, security issues and meeting the ever-increasing demands for its services. Therefore, research and development in this field needs to focus on addressing the fundamental problem areas in laying the core Grid infrastructure, which leaves the system less vulnerable to threats and creates a favourable environment for developing efficient middleware.

Chapter 4

University of Stellenbosch Campus-Grid Framework and Inter-Campus Grid Computing

4.1 – Introduction and General Principles

In establishing the US C-Grid, the computing resources to be virtually organized could be considered as partially dedicated because they do only donate their extra capacity to the campus-Grid. Besides, most of them are not supposed to run applications that use Grid resources beyond those locally available. This is because the computing centre PCs are used by students for applications that usually do not require a big computing power beyond the capacity of the local machine, and they are only used temporarily, hence, less concern to privacy issues on the PCs. Moreover, the PCs are mostly underutilized even during the *busy* hours, and most of them are idle during nights and weekends.

Therefore, the Grid architecture should consider the special features of the way computing capacity is harnessed, that is, virtually organizing the dedicated idle/extra capacities of all the computing centre PCs, while granting those PCs a full control of the resources they donate.

The main benefit of the Campus Grid is to the University at large, saving huge amount of money that would, otherwise, be spent to acquire supercomputers. The faculty computing centres could also benefit from the services they dedicate through some accounting system by charging the users of the services.

To make appropriate use of these available idle/extra resources, a proper Grid architecture should be adopted that spans all the campus computing resources.

The general guidelines for building this generic Campus-Grid framework are:

- to establish a Campus-Grid computing with something simple that works
- to address potential points of failure, paving the way for a more comprehensive C-Grid computing
- to give resource owners full control over their resources
- to ensure that this system will use the existing intranet infrastructure with some upgrades of necessary network devices, and eventually be compatible with different versions of Grid middleware available in the market

- to leave installation details (method, Operating System, configuration and so on) up to system administrators
- to impose as few restrictions of site configurations as possible- for example, let clusters select the amount of resources they dedicate to the Grid; permit computing nodes on private as well as public networks to access the C-Grid based on the access rules

In assessing the capacity and possible architecture of the US C-Grid computing, this research study depends on information from one of the faculty-computing centres, Humarga. All the network information discussed in the next sections will be based upon findings from this computing centre.

4.2 - Potential capacity of the US Faculty-Computing Centres based on the case of the Humarga Computing Centre

4.2.1 - Processing Capacity

Humarga, one of the five main faculty computing labs, has got 374 PCs, each one having a CPU of 2.4GHz. These will add up to a total of 897.6GHz – the arithmetic sum of all the processors. However, CPU scavenging Grid software usually mobilizes the distributed processors into a Grid of a factor of the arithmetic sum, 0.8 in the case of GreenTea Grid middleware- as per email-interview with the GreenTea developers from China. Hence, an average of 718GHz ($897.6\text{GHz} * 0.8$) can be made available for the C-Grid computing only from one faculty computing centre, Humarga. This is calculated in the case of an absolute idle state of all the computers. Although this is an ideal state, all the workstations are usually highly underutilized, and section 4.2.2 describes the utilization issue with regard to the capacities mentioned above.

4.2.2 - Storage Capacity

Each PC in the Humarga computing centre has a hard disk storage capacity of 20GB, while the current file storage system at this computing centre is being provided by a single Dell PowerEdge server with 5 discs in a RAID array (Redundant Array of Inexpensive Disks), and has a capacity of $74\text{GB} * 5 = 370\text{GB}$. This is big enough for the current demand for storage space. However, a storage capacity of more than 5.6TB ($15\text{GB} * 374\text{PCs}$, reserving 5GB storage space on each workstation for local use) can be mobilized and made available for the Grid services. Therefore, a distributed storage system with redundancy can be configured to utilize this idle storage capacity residing in every workstation.

Currently, the Humarga's Dell PowerEdge file server is providing a file storage service to various departments within the faculty. However, the major users of the proposed US C-Grid service will be departments of the university that use huge data-intensive or CPU-intensive research activities as well as others from within the university and/or other research centres and partner universities. Hence, a much bigger demand for storage is also going to complement these jobs that require immense processing capacity from the Grid.

4.2.3 – Current Utilization of the Computing Resources

The CPU utilization of each PC in the computing centre could be measured using a network monitoring software. In this case, therefore, a demo version of *Solarwinds Engineers Edition V-8* network monitoring software is used to observe the CPU and bandwidth utilization of the Humarga computing centre. Samples of the captured network information are attached in the Appendix Section at the end of the study.

The following graph shows the average CPU utilization of the Humarga Server as a percentage of its total capacity:

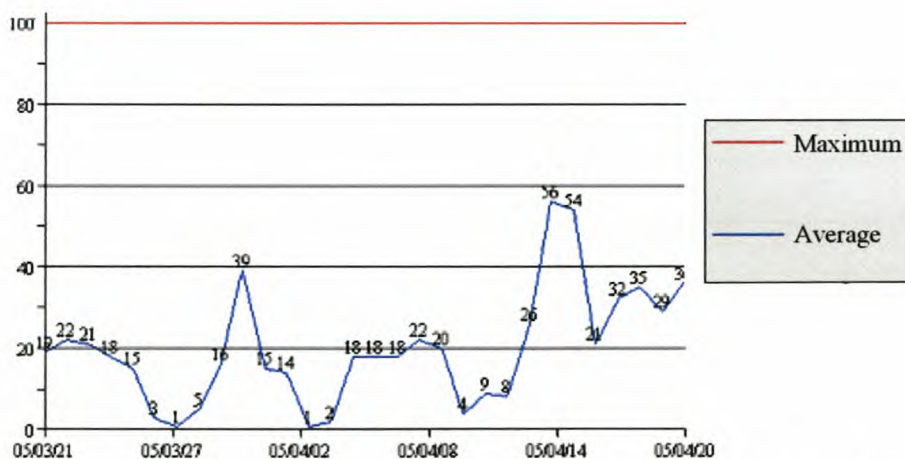


Fig 4.1 - A one month-profile for average CPU utilization of the Humarga Server (21/03/2005 – 20/04/2005), Humarga Computing Centre

From the graphical network statistics shown in Fig 4.1 over a period of one month, the average CPU utilization of the Humarga Server is 19.9%, with a few peak levels up to 56% and a few lower extremes of 1%. Similar pattern of CPU utilization could be observed using the Solarwinds network monitoring software from most of the workstations in the centre. Logically, therefore, a workstation with a CPU speed of 2.4GHz is using 19.9% of its total capacity, that is, 0.48GHz on average. Hence the rest is just an extra capacity (1.92GHz per each workstation) that could be mobilized using Grid computing. Exact figures, however, could be calculated and provided in subsequent research studies on the US C-Grid computing.

Note that Grid middleware can be configured to utilize the CPU up to a specific percentage of its full capacity for a reason that all PCs might not efficiently work at full capacity round the clock, which in practice could result in problems with CPU fan, thereby overheating CPU that might end up crashing/halt.

The network traffic to and from each computer, as well as the centre in general could be measured and presented as in the following example

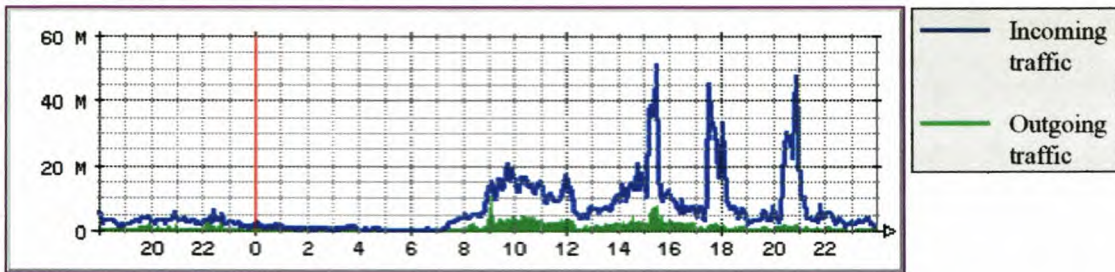


Fig 4.2 - Network utilization graph in a random week-day: 24hrs period, shown in the latter section of the graph, for the Humarga computing centre in total.

During the peak network utilization hours, only about 0.04% of the available one Gigabit/second intranet bandwidth, the capacity at which Humarga is connected to the rest of the US Intranet, is used for incoming network traffic. Incoming traffic constitutes browsing the internet, receiving email and downloading files from the internet among others. On the other hand, the outgoing traffic is almost negligible throughout the period, which refers to uploading files to the internet and sending emails/files. As shown in the above graph, the incoming traffic ranges from close to 0% at night hours (0:00 – 7:00) to slightly above 0.04% of the one Gigabit bandwidth during the peak hours of the day. This is mostly because the workstation is used for applications that run locally, if at all.

4.2.4 - Assessment of the Applications Used In the Humarga Computing Centre

Different applications are used in Humarga, including ArcGIS, ArcView, Adobe Creative Suite, Macromedia Studio MX, Finale, Qbase, Libronix, SPSS. These applications are installed in every workstation so that students can use them locally without adding traffic to the network. On the other hand, the C-Grid is intended for providing the idle/excess capacity to specific department and/or external commercial or Inter-Campus Grid service sharing. Hence, students run those applications they need locally, and leave more network bandwidth free for the Grid service.

According to the proposed US C-Grid architecture, all workstations are supposed to only process jobs and store data that are managed by the Grid middleware, but not to send jobs to the Grid for processing.

4.3 - US Grid Framework

This study focuses on the Campus Grid framework, inter-connecting clusters of computing power, which are virtually organized from each faculty computing centre only; it does not consider the idle/extra capacities residing in the individual PCs elsewhere in the campus. This is for the reason that at this initial stage, there should be a simple and functional Campus Grid computing, and over time it could be extended to include all the other Grid resources within the University. Additionally, the computing centre PCs are less vulnerable to privacy as they belong to no specific student and their idle/extra capacity can easily be predicted and organized virtually.

4.3.1 - Computing Resource Mobilization and Components of the US C-Grid

Varieties of services can be mobilized for the US C-Grid, the main resource being the distributed computing power that can be virtually mobilized from all the Faculty computing centres. Other Grid resources include: shared applications, expensive lab equipment, and storage services.

Distributed computing power: As discussed in the previous section (4.2), one of the faculty-computing centres, Humarga, of the US C-Grid can mobilize a processing power of up to 718GHz. Along with the other four computing centres inter-connected through the Grid middleware; the US C-Grid can make an immense computing power available for the Grid processing services. In this regard a CPU scavenging Grid middleware harnesses the idle/extra processing capacity within the clusters, while a computational Grid middleware provides the inter-connection among these clusters' processing power and the processing thereof.

Storage Services: The storage capacity of all the PCs virtually mobilized into the US C-Grid can also provide an immense storage capacity to the Grid users regardless of where their data is kept. The issue of security and accessibility of the storage media is addressed by the Campus Grid middleware. In this case, mirroring of the stored data plays the central role for a reliable access of the data so that it remains accessible even when some storage media are out of use for several reasons including failure and/or power outage.

Expensive Scientific/Lab Equipment: Apparently the expensive lab equipment usually belong to the corresponding department and in the same location, therefore, sharing these equipment mostly exists across universities or different Campus Grids between corresponding departments. However, a Professor who wants to use this machine in the lab from his desktop can use the Grid service, which in turn uses the pool of computing capacity to facilitate the process and his needs for extra computing capacity beyond that of his desktop PC, in general.

Shared Applications: the US C-Grid can also provide some expensive software applications including those which are used for analysis and interpretation of results from shared lab equipment; thus only the result file will need to be sent to the Grid node that requested the job instead of having to install the application locally and do the analysis. Simulation and some expensive statistical packages can also be shared over the Grid computing, saving costs and time.

Some major Grid middleware are discussed in the previous chapters including: the GreenTea, Boinc, Microsoft.NET, CORBA, etc. However, in a real work of developing an applicable Campus Grid computing, a thorough analysis and comparison should be made between these available middleware to identify the best feasible one for the specific campus intranet. The same also applies for the accounting software that is more functional to address the issue of service control and compensation.

Therefore, a GCGC framework, in this regard, constitutes: a feasible scavenging Grid middleware at the cluster level, an appropriate computational Grid middleware across the clusters and Grid client software in every Grid-node, the central cluster management servers, and on the shared scientific/lab instruments.

The US C-Grid services could be virtually mobilized in a more effective way as discussed in the next sections- *Logical and Physical Campus Grid Architectures*.

4.3.2 – The Logical Campus Grid Architecture

Within the domain of the Computing centres, the machine-cycle inefficiencies can be addressed by virtual servers or re-hosting- example, VMWare, MS Virtual PC, Virtual Server, LPARs from IBM and Partitions from SUN and HP, which do not require a Grid infrastructure. However, Grid computing emphasizes more on geographically distributed, multi-organizational, utility based, and networking reliant methods, whereas, Clustering and re-hosting have a more, but not exclusively, data centre-focused, single organization-oriented approach.

In the US C-Grid computing, the computing centres do only provide idle/extra computing capacity, and do not usually make use of Grid services beyond their local resources. Therefore, an efficient cluster system could be used to mobilize local cluster resources, and the cluster as a whole could be integrated to the campus Grid along with the other clusters-computing centres. Hence, Grid client middleware can be installed on the server where the cluster system software resides, and the whole cluster is considered as a huge supercomputing Grid node. Therefore, the whole campus-Grid system incorporates both computational and scavenging Grid types at different levels.

In this regard, the cluster managers should have full control over their respective clusters and may need to run commercial services under the regulations of the central IT department, when idle/extra capacity exists. The Campus Grid may also provide its idle/extra computing capacity for inter-campus cooperation and/or for commercial external services, provided that a higher bandwidth link is established between the different campuses or universities for a more efficient use of the services. A good example regarding an inter-institutional Grid computing infrastructure is presented by Stodghill, Heber & Lifka, in Greg Goth (2004).

Based on the discussions above and literature study, the logical layout of Campus Grid Computing could be presented as in the following figure, which depicts the different components and the logical interactions between them.

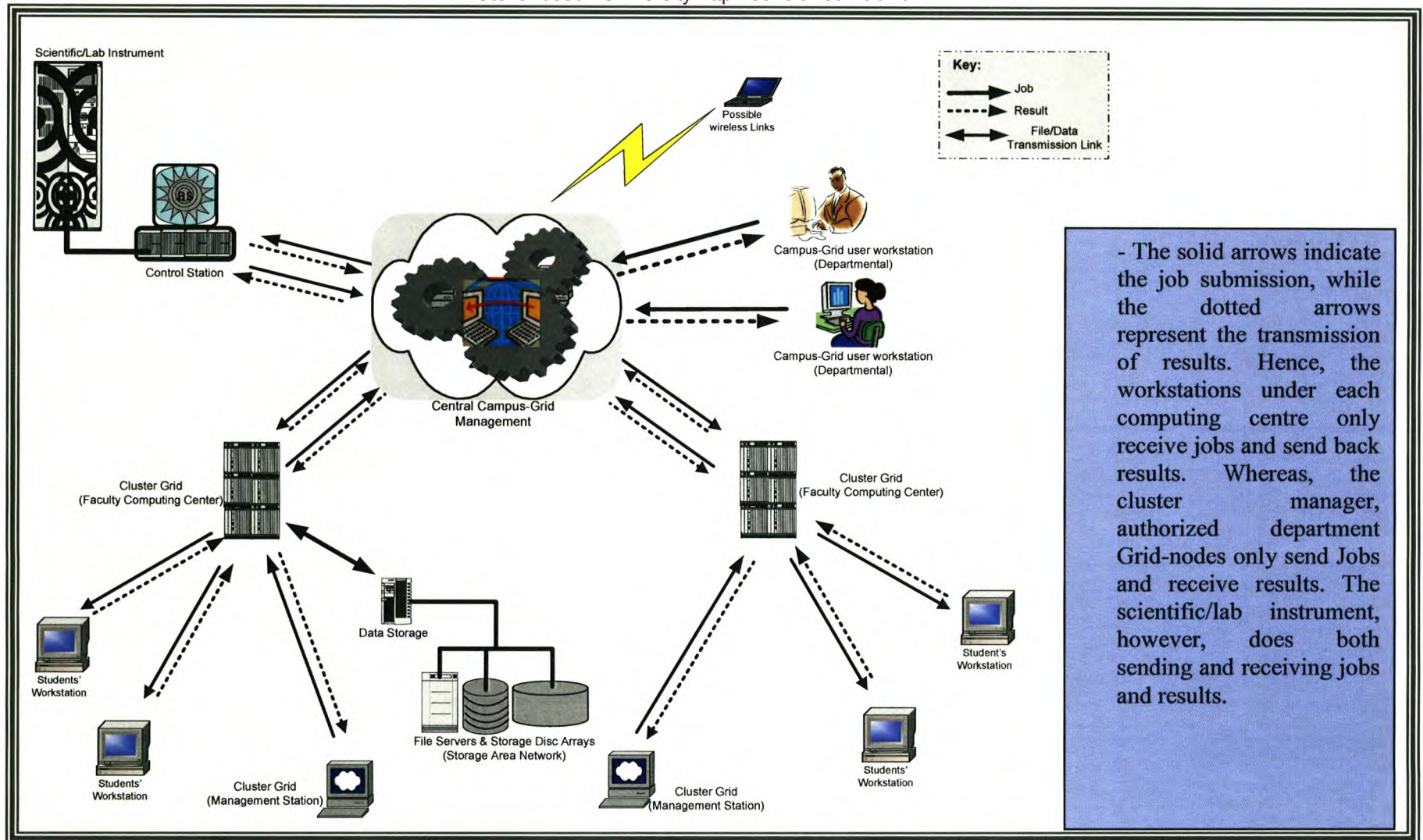


Fig 4.3 - Logical Campus Grid Layout:

a Generic layout where Grid middleware's client software are installed only on the *cluster management servers*, the *scientific/lab instrument operating computer* and *Grid user nodes* from the departments, which use the Grid computing pool.

As shown in Fig 4.3, the clusters provide only computing power (CPU, Memory, Storage) to the Campus grid, while some expensive scientific/lab instruments can be shared in the Grid through their operating computers. The Grid-Client software is, therefore, installed in the cluster server/management machine, in the shared scientific/lab instruments' operator PCs and in the authorized Grid nodes, which form a *Computational Grid* type, and a *Scavenging Grid* type is implemented to form each cluster.

Hence, the authorized specific departments' Grid-nodes use the Grid services as per their needs. On the other hand, these mobilized computing capacities from the clusters could be integrated into an accounting system and could be compensated for the services they provide as per the university's specific rules, and it should be the responsibility of the IT department.

4.3.3 – The Physical Campus Grid Architecture

The major components of the Physical C-Grid architecture are:

- *Computational Resources*: the faculty computing centre Ethernets (Humarga, Farga, Narga, Firga, Girga),
- *Shared expensive scientific/lab equipment*: particularly applicable in Inter-Campus or Global Grids,
- *User Nodes*: the Campus Grid Nodes from the specific departments that need to use this pool of computing power,
- *Storage*: Collective storage capacity from all the faculty computing centres and/or special storage systems can be virtually organized and made accessible to any authorized Grid node.
- *Software and Licenses*: Expensive and/or licensed software applications can be run from a few workstations by assigning jobs that require those applications to these workstations from all around the Grid, saving costs of purchasing more licenses and software for the campus.
- *The Network Infrastructure*: this provides the most fundamental communication infrastructure interconnecting all the physical components of the Campus Grid computing. Recent developments in high-bandwidth network equipment further promote the development of Grids. Hence, the campus Grid can use appropriate network devices such as Routers and Switches, capable of supporting high-bandwidth connectivity.

Depicting the most common components of the Campus Grid computing, the physical layout could look like the Fig 4.4 below- a more descriptive network diagram:

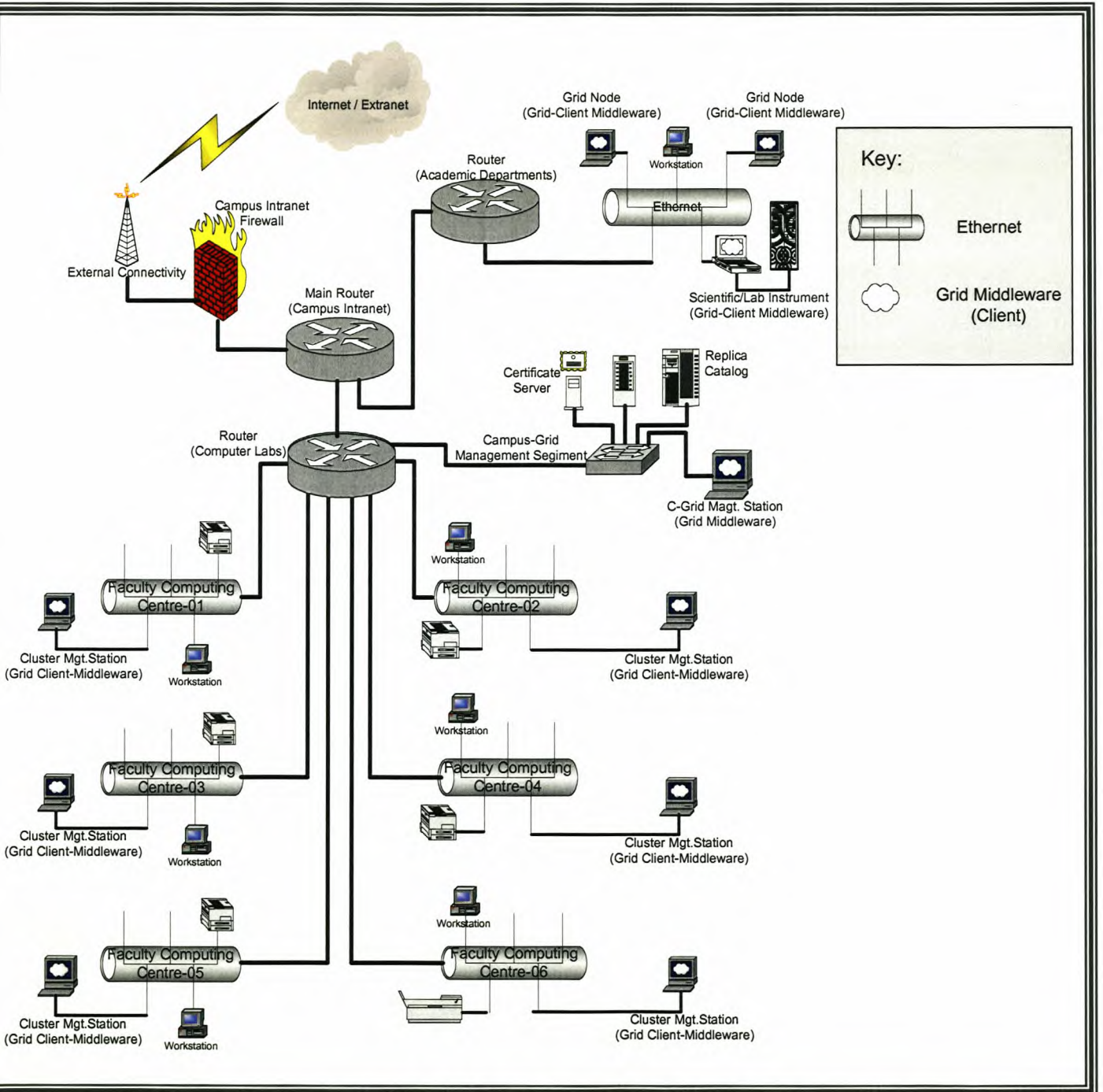


Fig 4.4 - Physical Campus Grid Layout:

A generic Layout consisting of several *faculty-computing centres*, a *Campus-Grid management unit* connected to this network segment's router, and an *Academic Departments' network segment* representing many more segments, which could contain *Grid Nodes*.

4.3.4 - What makes this Campus-Grid architecture different?

In standard Grid computing architecture every member of the Grid donates its idle/extra computing capacity and is eligible to use Grid services as per the specific rules and guidelines. However, the Campus Grid for tertiary institutions as being dealt within this study virtually organizes idle/extra capacities from each computer in the faculty-computing centres, integrates them into a campus Grid, and provides this pool of computing capacity to specific departments, which require immense computing power for inter-campus cooperation and/or for commercial use.

Therefore, according to Daniel Minoli's (2005:13) definition, this campus Grid framework incorporates a Scavenging Grid system at the cluster level and a Computational Grid system across clusters (at the campus level).

Besides, this C-Grid computing mobilizes computing resources, especially CPU Cycles, Storage and Memory, from the faculty computing centres that give computing services to students. This makes them relatively easily predictable for their idle/extra capacities and safer and secured because they are usually used for temporary students' activities- word processing, browsing and some specific software applications.

Note that this Generic Campus Grid framework is presented to establish a basic and functional Grid computing system, which can be extended to include many more resources from each department and administration offices when needed. Moreover, the fact that it is not primarily established for commercial use, but for facilitating research and laboratory experiments, also makes the establishment of the framework different from C-Grid computing frameworks in other contexts.

4.4 – Impact of the US C-Grid Framework

4.4.1 – The Prospects of Enhanced Network Services

The US C-Grid harnesses huge computing capacity (CPU, Memory, Storage), which is apparently idle/extra to the local process. Therefore, this Grid computing could provide a wide range of potential benefits to the University of Stellenbosch, and it is up to the US to exploit these opportunities. The main benefits include:

- Better utilization of underused IT resources: Mobilized, apparently underutilized IT resources include- Processors, Storage, Memory and Bandwidth. The institution at

large would benefit from cost savings, where supercomputers and dedicated file servers would be needed otherwise.

- Reliable access to immense computing power: the campus Grid provides easier access to a pool of computing capacity from a user's desktop, rather than relying on a central supercomputing centre.
- More reliable computing: this is usually attributed to the effectiveness of the Grid middleware used, which manages the processes and provides automatic graceful recovery of jobs from failures due to hardware/processor problems. Hence, this relies more on software technology than expensive hardware.
- Parallelization of processing: the Grid computing provides the capability of partitioning jobs into independently running 'micro-jobs' and lets them run simultaneously, which does the job amazingly faster than the traditional one- running a the job in a single or a few faster processors.
- Resource balancing: the Grid system schedules jobs to run in relatively idle processors. It can even replace lowest priority jobs that are running with highest priority jobs that are in a queue, in cases where the Grid resources are fully utilized.
- Simplified management of IT resources: the feature of Grid middleware providing a uniform method to handle heterogeneous systems makes the management of IT easier.
- Creates virtual resources and virtual organizations for collaboration that enables knowledge and expertise sharing
- Access to the plethora of IT resources: This comprises special scientific/lab equipment, software applications that are expensive or licensed, and expertise in highly specialized areas. This is usually applicable in inter-campus/global Grid computing where one institution owns some expensive or specialized resources and these need to be shared among many other institutions which are interconnected via the Grid computing system.
- Provides immense computing capacity, which can even be used for commercial services.

4.4.2 – Its Impact on Inter-Campus Resource Sharing

Being effectively established to live up to its promises, the US C-Grid can provide its Grid services to external partner institutions commercially and/or on the basis of mutual cooperation agreements. This can be discussed from two perspectives- potential resources for inter-campus Grid sharing and limitations.

Potential Capacity of the US C-Grid: The major Grid resource that the US C-Grid could offer is the Computational- CPU Cycles, resource. Provided that the enabling infrastructure is established, therefore, other partner universities and research centres can make use of the extra processing capacity provided by the US C-Grid. Besides, it can also enable sharing immense storage services and some expensive scientific/lab instruments across the campus boundaries. Hence, the inter-campus/global Grid can potentially create scientific tools consisting of a set of interlinked instruments from geographically disparate locations, provided that the other campuses also establish Grid computing links with the US C-Grid.

Moreover, the inter-campus Grid link can also enable research communities of a specific interest/field to exchange knowledge and information across university campuses and research institutions.

Limitations: The main limiting factor in inter-campus Grid is the bandwidth capacity of the link-infrastructure. Grid computing requires a higher possible bandwidth network to operate on because it involves a huge data transmission when running data-intensive and/or CPU-intensive jobs. Therefore, the link capacity is a bottleneck for such links across C-Grids. This network superhighway, however long it may take, shall be dealt with to enable future Grid resource sharing across institutions. Best practices can be learned from experiences of Inter-campus high bandwidth networking projects in the USA and Europe, discussed in the previous sections of the thesis.

Moreover, all the other potential partner institutions should also launch such C-Grid initiatives in a nationally coordinated manner and be prepared to be part of and help develop the national, inter-campus high-bandwidth networking infrastructure project. The awareness of Grid computing advantages and the prospective IT advancements would, therefore, be promoted by every campus- tertiary institution and research centres. The current level of awareness and the efficiency of available IT resources utilization in the tertiary institutions is the other impeding factor to the issue.

Chapter Summary

This chapter presents a case study of C-Grid framework on the US intranet based on one of its faculty-computing centres, Humarga.

Computing resources in this case are supposed to donate their idle/extra capacity to the campus Grid while serving the local user on a temporary basis. Hence, they could be considered as partially dedicated resources. These PCs are usually underutilized, if not idle, and local processes usually do not require extra resources from the campus Grid. Besides, the computing centre workstations are chosen to be mobilized to form the C-Grid for a reason that they are less vulnerable to privacy as they belong to no specific user/student, and that their idle/extra capacity can be relatively easily predicted.

The core principle of this study is to start a campus Grid computing with something simple that works and that poses as few restrictions as possible to the cluster Grid in every computing centre.

The US C-Grid mainly comprises mobilized distributed processing power, distributed storage services and memories, shared scientific/lab instruments, and shared expensive applications.

Well designed and established, the C-Grid computing would render rich potential benefits to the Grid Users. These include: access to the plethora of IT resources, better utilization of underused IT resources, reliable access to immense computing power, a more reliable computing, parallelization of processing, resource balancing, simplified management of IT resources, creating virtual resources and virtual organizations for collaboration that enables knowledge and expertise sharing, and providing immense computing capacity, which can even be used for commercial services.

In the bigger sphere of inter-campus Grid networking, the US C-Grid could potentially provide its extra CPU cycles for use by the partner institutions and research centres. Moreover, it can provide an immense pool of storage capacity and some expensive scientific instruments across campus boundaries. Hence, the inter-campus Grid computing can also enable research communities of specific interest/field to exchange knowledge and information across campuses and research institutions. However, the main limiting factor in the inter-campus Grid is the bandwidth capacity of the link-infrastructure because it involves a huge data transmission when running data-intensive and/or CPU-intensive jobs. Therefore, best practices can be learned from experiences of Inter-campus high bandwidth networking projects in the USA- *NLR*, and the European *Data-Grid*.

Chapter 5

Summary, Conclusion and Recommendations

5.1 – Summary

This research provides a thorough literature study of Grid computing in general and Campus Grid in particular; it presents a Generic Campus Grid Computing Framework, based on preliminary network information from the University of Stellenbosch Intranet. Moreover, this study discusses the salient components of the GCGC framework, estimates the idle capacity residing in the faculty computing centres- assessing the case of one of US faculty computing centres, and presents the range of services the Campus Grid could potentially provide for users from departments of the US and for inter-campus Grid resource sharing. A brief summary that recapitulates the key points is provided at the end of each chapter.

The main focus of the research study is to assess how the next generation inter-networking- the Grid Computing, is paving its way in transforming *Computation, Communication and Collaboration* to the next level, although continuous change is inevitable at all levels. In this regard, next generation scientific exploration requires computing power and storage that no single institution alone is able to afford. Additionally, easy access to distributed data is required to improve the sharing of results among scientific communities spread around the world. The proposed solution to these challenges is to enable different institutions working in the same scientific field, to put their computing, storage and data resources together in order to achieve the required performance and scale, hence, Grid computing emerges to enable this collaboration between different e-environments.

The summary of this research study could be presented in terms of the following questions:

- Why do we need Grid computing? Basically, Grids can enhance human creativity by, for example, increasing the aggregate and peak computational performance available to important applications and allowing the coupling of geographically separated people, hence knowledge, and computers to support collaborative performances. Moreover, Grid applications are able to provide dependable, consistent, and pervasive access to high-end computational capabilities.
- What does Grid mobilize and how does it work? Grid computing environments differ in terms of technologies used, languages they use and in terms of how objects are treated. In

most Grid computing services, the Grid software divides a job into multiple smaller and discrete tasks that are then distributed to the various computing nodes in the Grid. It should be understood, however, that not all computational jobs can be broken up into small, independent tasks; besides the idle/extra CPU capacity, Registered Grid computing nodes can also lend data, memory and storage spaces.

- Can Grid Computing be applied on an existing network infrastructure? Usually not, because Grid computing, especially scavenging Grids, require each workstation to be capable of working at full capacity and for longer possible hours. Already existing networks may not be entirely composed of such a capability, as they were not intended for Grid. Therefore, establishing Grid computing needs to be accompanied by a thorough assessment of the composition of the already existing network and deal with the required changes and improvements. In addition, the capacity of the bandwidth of the network infrastructure is vital for the Grid deployment. Hence, in most cases the cabling and/or network devices need to be replaced to allow larger data transfer and faster connection capabilities. The greater challenge is with regard to inter-campus connection that has to rely on the existing limited bandwidth Internet/VPN infrastructure, unless a high-bandwidth national/across-institutional network infrastructure is established.
- What types of applications will Grid computing be used for? The following types of Grid applications are discussed in some depth:
 - Distributed supercomputing, in which many grid resources are used to solve very large problems;
 - High throughput, in which grid resources are used to solve large numbers of small tasks;
 - On demand, in which grids are used to meet peak needs for computational resources;
 - Data intensive, in which the focus is on coupling distributed data resources; and
 - Collaborative, in which grids are used to connect people, hence, knowledge and all important Grid resources.
- Who will use grids? Depending on the need for the particular Grid development and setup, the users of the Grid could be:
 - a national Grid, serving national government bodies

- a private Grid, serving specific domain, like a health maintenance organization
- a community Grid, serving scientific collaboration and communities of interest
- a public Grid, supporting a market for computational services.
- an ad hoc Grid, serving wireless Grid users and temporary Grid computing membership

Also in broader categories:

- a Cluster Grid, serving a single department/section of an organization
 - a Campus Grid, serving group of departments/whole organization
 - a Global Grid, serving multiple organizations across the globe
- How will Campus Grids be used in tertiary institutions? Campus Grid organizes the available idle/excess computing capacity residing especially in the huge collections of PCs in the faculty computing centres, and creates a virtual computing pool. This *Campus-Grid* computing pool can be used by departments that run data-intensive/CPU-intensive jobs or used for inter-campus Grid services sharing. It can also be outsourced for commercial services.
- What problems must be solved to enable Grid development? Section 3-6 of the research study provides an overview of the challenges referring to the focus, representation, requirements, dynamic composition, and a bigger issue regarding the use of wireless devices which are increasingly more important for the ubiquity of the Grid services. Hence, these remain to be addressed before Grids can be fully applicable on a large scale.
- What are the potential services Grids can provide? The aggregation of computing power can bring a number of benefits:
- Increases the throughput of users' jobs by maximizing resource utilization
 - Increases the range of complementary hardware available, for example, computing clusters, large shared memory servers, extra storage capacity required, etc.
 - Provides a *Grid supercomputing power* that can provide a platform for grand challenge applications
 - Provides the tight integration of geographically and functionally disparate databases

- Provides the catering for a huge, dynamic data set and the processing thereof.

Moreover, Grid environments incorporate properties that are desirable for the effective functioning of the system, and these include: reliability, security and trust across multiple administrative domains, persistence, scalability, open to wide user communities, pervasive and ubiquitous, transparent, easy to use and program, and based on standards for software and protocols.

Generally, Grid Computing is not an alternative to Internet as many people presume from the expression that *Grid is the next generation Internet*. It is the future of Inter-networking service being built on the Internet protocols and services that boosts the usability of the Internet infrastructure, enabling the creation and use of computation and data enriched environments. Additionally, it provides a conducive, virtual environment promoting the sharing of knowledge and scientific tools. This is to say, Grid computing and resources, especially in the Global scale, reside on the Internet and are provided over the Internet. Hence, the Internet and its protocols form the Infrastructure for Grid computing.

5.2 – Conclusion

After the advances made in distributed system design, collaborative environments, high performance computing and high throughput computing, the Grid is the logical next step. Now that cost cutting has a major impact on information services, Grid computing further enables designers and engineers to crunch through their ever-increasing, compute-intensive jobs – without the need for a budget to acquire supercomputers. This Grid computing combines idle/extra computing power of its over-powered workstations and servers, while integrating this with other valuable resources located elsewhere in the pool, to provide powerful scientific research tools for a particular purpose or problem solving. The major limitation with the prevailing computing resources, however, is that each workstation/Grid node should not be demanded to work at full CPU capacity round the clock, as it could result in over-heating of the processors, and failure of other hardware components.

Moreover, the perception that the Grid computing is a source of free computing power-*cycles*, and shared resources, is also far from reality. Rather, the Grid computing is a source of an immense computing power- *cycles, data and tools*, as it provides all these resources under a controlled and coordinated access and sharing policy. In this regard, resource owners tend to enforce policies that constrain access according to their group membership, and the central Grid policy should consider the local policy of the resource to be shared or used. Therefore, cost information and accounting systems need to be integrated to address the issue of *fair-sharing*.

Initially, when the PC was first invented, it had a much larger computing capacity than needed. However, the demand for even larger computing power, caused by a rapid growth in its applications, quickly outstripped its capacity. Similarly the Grid computing, currently under development, provides an immense pool of computing power. But in the near future, the need for even more computing capacity will, in the same way, inevitably overtake its current capacity, and will drive continuous innovation resulting in evolution of new computing systems. Keeping the above in mind, two further questions arise. Firstly, *will the demand for such immense capacity be satisfied by the foreseen technology, and the services it is supposed to provide?* And secondly: *How far will the technology improvements and the Grid infrastructure development influence the complexity and nature of scientific research and data?* Such core concerns will continue to dictate further research in the field of Grid computing, and induce continuous innovations.

5.3 - Recommendation

Grid computing system is currently in its early stages, and numerous research and developments on this field are currently underway. An abundance of articles and papers are being published, while there are other, largely untapped, areas that still need thorough assessment and research.

Some of the present studies deal with hypothetical assumptions on Grid computing, while many others propose different methods and applications for the Grid computing development. The interdisciplinary nature of Grid Computing research and development requires the complementary and coordinated modular development efforts from experts in communication infrastructure, database design, information architecture and management, and computer science and engineering. At the core of the development are also the users of the e-environment against whose requirements the proposed Grid services are configured and fine-tuned. Consequently, since most of the research projects are being made on specific problem areas and fields in Grid computing, further investigation needs to be done on the coordination of the potential solutions and methods that addresses its requirements. Furthermore, such a research is not only about improving processing speed, or easier data access and sharing, but also about inventing new methods and strategies to maximize the utilization of enterprise resources, and the sharing across institutional boundaries. Hence, concurrent with the research on Grid computing, attention should be given to the re-organization and inventing of new methods in handling research data and management of the shared resources, as enabled by the Grid services.

Forth-flowing from this research, a further study could be made on a similar topic, accompanied by a practical implementation of the fundamental Grid infrastructure that can then be used by the CPU-Intensive and/or data-intensive jobs. This could develop a less complicated and functional Campus-Grid Computing System. With regard to inter-campus Grid computing, however, subsequent research could provoke *cross-institutional* and/or national Grid Computing projects, which would avoid the limiting VPN-link capacity between the institutions.

References

- Agoston, T. C., Ueda, T. & Nishimura, Y. *Pervasive computing in a networked world* [Online]. Available: http://www.isoc.org/inet2000/cdproceedings/3a/3a_1.htm [Feb 2005]
- Amin, K., Laszewski, G. & Mikler, A.R.. 2004. *Towards an architecture for Ad Hoc Grids* [Online]. Available: <http://www-unix.mcs.anl.gov/~laszewsk/papers/vonLaszewski-adhoc-adcom2004.pdf> [Feb, 2005]
- Berman, F., Fox, G. & Hey, T. 2003. *Grid computing: making the global infrastructure a reality*. England: John Wiley & Sons Ltd.
- Berman, F., Fox, G. & Hey, T. 2003. *The Grid: past, present, future*. In *Grid computing: making the global infrastructure a reality/* edited by Berman, Fox & Hey. England: John Wiley & Sons Ltd. P.9-50.
- Berners-Lee, T., Hendler, J. & Lassila, O. 2001. *The Semantic Web [online]*. Scientific American. Available: http://www-sop.inria.fr/acacia/personnel/Fabien.Gandon/lecture/licence_travaux_etude2002/TheSemanticWeb/ [May 2005]
- CPU scavenging: Free Encyclopaedia* [Online]. Available: <http://encyclopedia.thefreedictionary.com/CPU+scavenging>[Feb 2005]
- Data Grids* [Online]. Available: <http://web.datagrid.cnr.it/LearnMore/index.jsp> [Feb 2005]
- Eerola, P., [et al]. 2003. Building a production Grid in Scandinavia. *IEEE Internet Computing*, Vol.7, No.4: 27-35
- Esparza, O. [et al]. *Security issues in virtual Grid environments* [CD-ROM]. Computational Science – ICCG 2004, 4th International conference, June 2004. Poland: Krakow
- Filman, R. E. 2003. Semantic services. *IEEE Internet Computing*, Vol.7, No.4:4-6
- Filman, R. E. Lessons from system development. *IEEE Internet Computing*, Vol.8, No.1: 4-6

Filman, R. E. 2004. Days of miracle and wonder. *IEEE Internet Computing*, Vol.8, No.3: 4-6

Foster, I. 2003. The Grid: a new infrastructure for 21st century science. In *Grid Computing: making the global infrastructure a reality*/ edited by Berman, Fox & Hey. England: John Wiley & Sons Ltd. P.51-64

Foster, I. [et al] 2002. *A community authorization service for group collaboration* [Online]. Available: http://www.globus.org/research/papers/CAS_2002_Revised.pdf [Feb 2005]

Foster, I. [et al.] 2003. *Security for Grid services* [Online]. Available: <http://www.globus.org/Security/GSI3/GT3-Security-HPDC.pdf> [Feb 2005]

Foster, I. & Douglass, F. 2003. The Grid grows up. *IEEE Internet Computing*, Vol.7, No.4: 24-26

Foster, I. & Kesselman, C. 1998. *Computational Grids* [Online]. Available: <http://www.globus.org/research/papers/chapter2.pdf> [Jan 2005]

Foster, I., Kesselman, C. & Tueck, S. 2003. The anatomy of the Grid. In *Grid Computing: making the global infrastructure a reality*/ edited by Berman, Fox & Hey. England: John Wiley & Sons Ltd. P.171- 198

Gaynor, M. [et al]. 2004. Integrating wireless sensor networks with the Grid. *IEEE Internet Computing*, Vol.8, No.4: 32-39

Goble, C. & Roure, D.D. *Semantic Web & Grid Computing* [Online]. Available: <http://www.semanticgrid.org/documents/swgc/swgc-final.pdf> [Feb 2005]

Goth, G. 2004. Is high-performance computing entering a new era? *IEEE Internet Computing*, Vol.8, No.2: 9-11

Graupner, S. [et al]. 2003. Service centric globally distributed computing. *IEEE Internet Computing*, Vol.7, No.4: 36-43

GreenTea Technologies Inc. [Online]. Available: <http://www.GreenTeaTech.com> [Feb 2005]

Grid Computing: Free Encyclopaedia [Online]. Available:
<http://encyclopedia.thefreedictionary.com/Grid+computing> [Feb 2005]

Grid technology overview: Sun powers the Grid [Online]. Available:
<http://www.sun.com/software/grid/overview.html> [Feb 2005]

Grid: the competitive advantage. Aug 2004 [Online]. Available:
http://www.sun.com/solutions/documents/articles/grid_adv_aa.xml?null [Feb 2005]

History of computing hardware: Free Encyclopaedia [Online]. Available:
<http://encyclopedia.thefreedictionary.com/History+of+computing+hardware>

History of the Internet: Free Encyclopaedia [Online]. Available:
<http://encyclopedia.thefreedictionary.com/History+of+the+Internet>

Huhns, M. (ed). 2004. Intelligent agents meeting the semantic web in smart spaces. *IEEE Internet Computing*, Vol.8, No.6: 69-79

Hwang, S. [et al] 2003. *An idle compute cycle prediction service for computational Grids*. Korea: Seoul

IBM Grid Computing –Gridlines [Online]. Available: <http://www-1.ibm.com/grid/gridlines/January2004/feature/teamwork.shtml> [Feb 2005]

Jeffery, K. 2004 (ed). *Next generation Grids 2: requirements and options for European Grids research 2005-2010 and beyond, expert group report* [online]. Available:
http://www.semanticgrid.org/docs/ngg2_eg_final.pdf [Jan 2005]

Kelly, W., Roe, P. & Sumitomo, J. 2002. *G2: A Grid middleware for cycle donation using .NET* [Online]. Available: <http://g2.fit.qut.edu.au/G2/HomePage/PDPTA02.doc> [Feb, 2005]

Laforenza, D. *Some Technology Trends* [online]. Available:
http://miles.cnuce.cnr.it/~lafo/domenico/APD2003/L_2.pdf [May 2005]

McKnight, L.w., Howison, J. & Bradner, S. 2004. Wireless Grids: distributed resource sharing by mobile, nomadic and fixed devices. *IEEE Internet Computing*, Vol.8, No.4: 24-31

Malik, O. 2002. *Distributed computing Grid networks will put idle computing power to work -- but when?* Red Herring Magazine, October 11, 2002

Metz, C. 2003. The latest in virtual private networks: Part I. *IEEE Internet Computing*, Vol.7, No.1: 87-91

Metz, C. 2004. The latest in virtual private networks: Part II. *IEEE Internet Computing*, Vol.8, No.3: 60-65

Minoli, D. 2005. *A networking approach to Grid computing*. New Jersey: John Wiley & Sons Inc.

Newhouse, S. *Laying the foundations for the semantic Grid* [Online]. Available: <http://www-icpc.doc.ic.ac.uk/components/> [Feb 2005]

Osterle, H. A process-oriented framework for efficient Intranet management [Online]. Available: http://www.isoc.org/inet99/proceedings/1d/1d_4.htm [Feb 2005]

Pervasive Computing [Online]. Available: http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci759337,00.html [Feb 2005]

Platform in the news archive [Online]. Available: <http://www.platform.com/newsevents/inthenews/archive.asp?year=2004> [Jan 2005]

Roure, D.D. 2003. *The Semantic-Pervasive-Grid Triangle* [Online]. Available: <http://www.semanticgrid.org/pervasive> [Feb 2005]

Roure, D.D., Jennings, N.R. & Shadbolt, N.R.. 2003. The Semantic Grid: a future e-science infrastructure. In *Grid Computing: making the global infrastructure a reality*/ edited by Berman, Fox & Hey. England: John Wiley & Sons Ltd. P.437-470
<http://www.semanticgrid.org/documents/semgrid-journal/semgrid-journal.pdf>

Roure, D.D., [et al.] 2003. The evolution of the Grid. In *Grid computing: making the Global infrastructure a reality*/ edited by Berman, Fox & Hey. England: John Wiley & Sons Ltd. P.65-100

Rowley, D. 1998. *Five intranet management problems you can solve with a change management system* [Online]. Available:
<http://itmanagement.earthweb.com/erp/article.php/601521>

Squeezing more from your IT dollars [Online]. May 2004. Available:
http://www.sun.com/solutions/documents/articles/MF_squeezing_AA.xml [Feb 2005]

Sun Microsystems inc. *Campus Grid overview* [Online]. Available:
<http://apstc.sun.com.sg/old/campusgrid.php> [Feb 2005]

Sun Microsystems inc. 2002. *Introduction to Cluster Grids- Part 1* [Online]. Available:
<http://www.sun.com/blueprints/0802/816-7444-10.pdf> [Jan 2005]

Sun Microsystems Inc. 2003. *Building a global compute Grid* [Online]. Available:
<http://wwws.sun.com/software/grid/whitepaper.edge.pdf> [Jan 2005]

Sun Microsystems Inc. 2003. *Computing at the edge* [Online]. Available:
<http://wwws.sun.com/software/grid/whitepaper.edge.pdf> [Feb 2005]

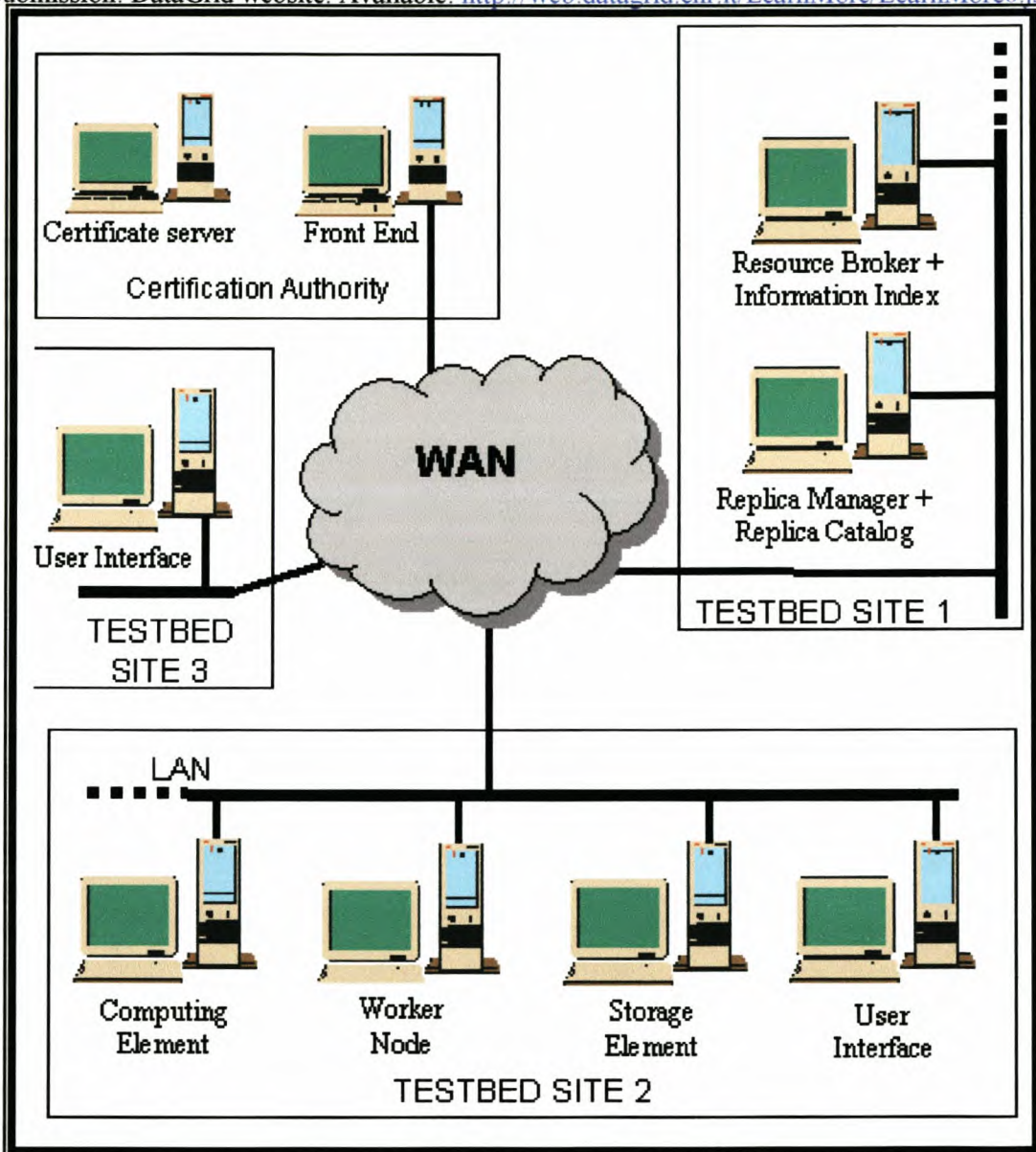
Super Computer: Free Encyclopedia [Online]. Available:
<http://encyclopedia.thefreedictionary.com/Supercomputer/> [Feb 2005]

The Globus Alliance. *Grid test-beds* [Online]. Available:
<http://www.globus.org/research/test-beds.html> [Feb 2005]

The guide to computing literature [Online]. Available:
<http://portal.acm.org/results.cfm?query=%2Bauthor%3AP115430&querydisp=author%3AIan%20Foster&coll=GUIDE&dl=GUIDE&CFID=33456069&CFTOKEN=29059325> [Jan 2005]

Appendix - A:

Job Submission: DataGrid website. Available: <http://web.datagrid.cnr.it/LearnMore/LearnMore8.jsp>



The picture above describes a Grid test-bed with three sites and a Certification Authority.

In support of the test-beds, a *Certification Authority* is needed, which is made up of two machines:

- a web front-end which receives requests and delivers certificates once they are issued and signed;
- a secure server, possibly disconnected from the network, which signs certificates.

A certificate is required for each user to prove his/her identity when he/she submits a job request to the test-bed.

- *A test-bed is made up of one or more sites, three sites in the case of the above example. Each site contains a certain number of machines, each one playing a different role. Each role is implemented by one or more middleware modules that*

have to be installed on the machine. Below is a list modules as shown in the above picture and their respective roles.

- The *Resource Broker* is the module that receives users' requests and queries the Information Index to find suitable resources.
- The *Information Index*, which can reside on the same machine as the *Resource Broker*, keeps information about the available resources.
- The *Replica Manager* is used to coordinate file replication across the test-bed from one *Storage Element* to another. This is useful for data redundancy but also to move data closer to the machines which will perform computation.
- The *Replica Catalogue*, which can reside on the same machine as the *Replica Manager*, keeps information about file replicas. A logical file can be associated to one or more physical files which are replicas of the same data. Thus a logical file name can refer to one or more physical file names.
- The *Computing Element* is the module which receives job requests and delivers them to the *Worker Nodes*, which will perform the real work. The Computing Element provides an interface to local batch queuing systems. It manages one or more *Worker Nodes*. A *Worker Node* can also be installed on the same machine as the *Computing Element*.
- The *Worker Node* is the module installed on the machines which will process input data.
- The *Storage Element* is the module installed on the machines which will provide storage space to the test-bed. It provides a uniform interface to different Storage Systems.

The *User Interface* is the module that allows users to access all the Data-Grid service (Job submission, Data Management, Information Management, etc.)

Appendix – B : CPU Utilization Info, captured using a CPU Gauge software

The figure shows four screenshots of the 'Advanced CPU Load' software interface, each displaying a table of CPU utilization data for different IP addresses. The data is as follows:

IP Address	Status	% Load	Max CPU load today
146.232.53.140	●	1 %	1 % at 3:44 PM
146.232.53.142	●	1 %	1 % at 3:45 PM
146.232.53.143	●	6 %	20 % at 4:18 PM

IP Address	Status	% Load	Max CPU load today
146.232.53.151	●	3 %	7 % at 3:51 PM
146.232.53.152	●	4 %	9 % at 5:04 PM
146.232.53.153	●	17 %	17 % at 5:07 PM

IP Address	Status	% Load	Max CPU load today
146.232.53.170	●	2 %	2 % at 5:07 PM
146.232.53.172	●	3 %	3 % at 5:05 PM
146.232.53.174	●	1 %	2 % at 5:04 PM

IP Address	Status	% Load	Max CPU load today
146.232.53.162	●	5 %	29 % at 5:05 PM
146.232.53.164	●	6 %	6 % at 5:07 PM
146.232.53.166	●	2 %	2 % at 5:07 PM

CPU utilization data captured from Humarga: current level of utilization – PCs running at a highly under-capacity levels

The figure shows four screenshots of the 'Advanced CPU Load' software interface, each displaying a table of CPU utilization data for different IP addresses. In all cases, the CPU load is at 100%.

IP Address	Status	% Load	Max CPU load today
146.232.53.142	●	100 %	100 % at 1:26 PM
146.232.53.143	●	100 %	100 % at 1:27 PM
146.232.53.144	●	100 %	100 % at 1:27 PM

IP Address	Status	% Load	Max CPU load today
146.232.53.151	●	100 %	100 % at 1:31 PM
146.232.53.152	●	100 %	100 % at 1:31 PM
146.232.53.153	●	100 %	100 % at 1:32 PM

IP Address	Status	% Load	Max CPU load today
146.232.53.145	●	100 %	100 % at 1:28 PM
146.232.53.146	●	100 %	100 % at 1:28 PM
146.232.53.147	●	100 %	100 % at 1:29 PM







IP Address	Status	% Load	Max CPU load today
146.232.53.154	●	100 %	100 % at 1:33 PM
146.232.53.155	●	100 %	100 % at 1:33 PM
146.232.53.156	●	100 %	100 % at 1:33 PM

CPU utilization data captured from Humarga when the full capacity is mobilized using a CPU-Scavenging software.







Network Information gathered using Solarwinds Engineers Edition V-8 Network Monitoring Software (Demo Version)

Appendix – C : Network Utilization Info, captured using a Bandwidth Gauge software

Bandwidth Gauges [Bandwidth Gauge- 129-155--01]

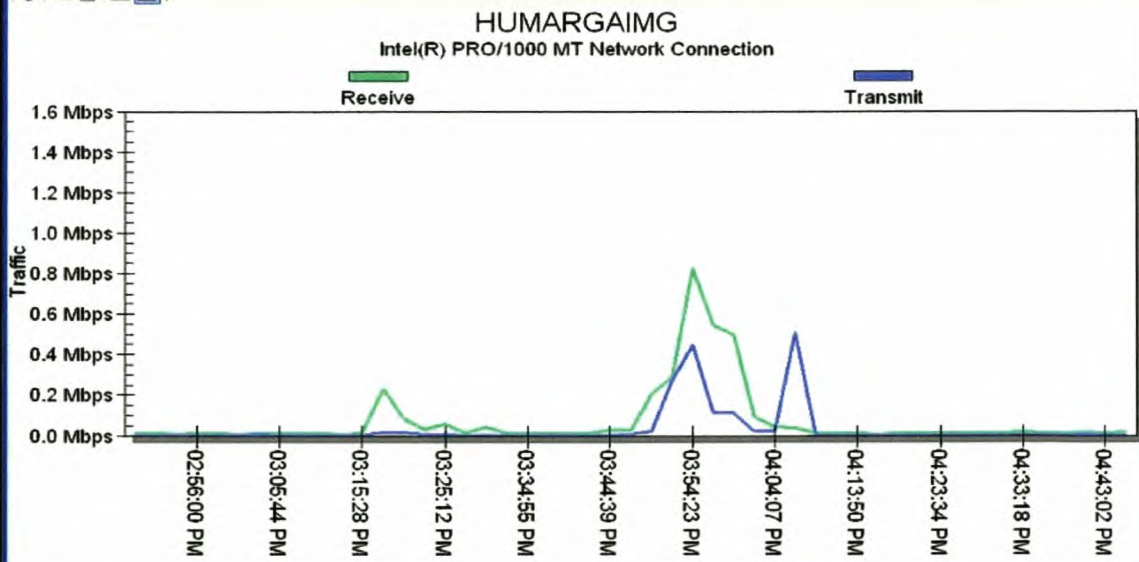
HUMARGAIMG		HUMARGAIMG		HUMARGAIMG	
Intel(R) PRO/1000 MT Network Connection		Intel(R) PRO/1000 MT Network Connection		Intel(R) PRO/1000 MT Network Connection	
					
	Receive		Transmit		Transmit
Current % Util	0.01 %	0.00 %	Current % Util	0.01 %	0.00 %
Max % Util	0.35 %	4.73 %	Max % Util	0.40 %	1.33 %
Min % Util	0.00 %	0.00 %	Min % Util	0.00 %	0.00 %
Avg % Util	0.02 %	0.03 %	Avg % Util	0.02 %	0.02 %

Bandwidth Gauges [Bandwidth Gauge- 156-176--01]

HUMARGAIMG		HUMARGAIMG		HUMARGAIMG	
Intel(R) PRO/1000 MT Network Connection		Intel(R) PRO/1000 MT Network Connection		Intel(R) PRO/1000 MT Network Connection	
					
	Receive		Transmit		Transmit
Current % Util	0.01 %	0.00 %	Current % Util	0.07 %	0.01 %
Max % Util	0.83 %	0.51 %	Max % Util	0.12 %	0.25 %
Min % Util	0.01 %	0.00 %	Min % Util	0.01 %	0.00 %
Avg % Util	0.06 %	0.03 %	Avg % Util	0.03 %	0.02 %

Current levels of network traffic generated by each PC: Bandwidth Utilization

Historical Statistics [HUMARGAIMG Intel(R) PRO/1000 MT Network Connection]



Current levels of network traffic generated by each PC: Graphic, historical Statistics of a random PC

Network Information gathered using *Solarwinds Engineers Edition V-8* Network Monitoring Software (Demo Version)

Appendix – D

A Historical network traffic statistics

HUMARGAIMG				
Intel(R) PRO/1000 MT Network Connection				
Time Stamp	Recv Bps	Xmit Bps	Recv % Utilization	Xmit % Utilization
18:03:17	9.7 Kbps	241 bps	0.01 %	0.00 %
18:02:11	11.9 Kbps	266 bps	0.01 %	0.00 %
18:01:05	8569 bps	206 bps	0.01 %	0.00 %
17:59:59	12.7 Kbps	242 bps	0.01 %	0.00 %
17:58:53	12.4 Kbps	335 bps	0.01 %	0.00 %
17:57:47	9.7 Kbps	232 bps	0.01 %	0.00 %
17:56:41	11.5 Kbps	337 bps	0.01 %	0.00 %
17:55:35	9402 bps	224 bps	0.01 %	0.00 %
17:54:29	10.0 Kbps	189 bps	0.01 %	0.00 %
17:53:23	12.1 Kbps	367 bps	0.01 %	0.00 %
17:52:17	8997 bps	155 bps	0.01 %	0.00 %
17:51:11	10.0 Kbps	259 bps	0.01 %	0.00 %
17:50:05	9.7 Kbps	380 bps	0.01 %	0.00 %
17:49:00	32.8 Kbps	1.11 Mbps	0.03 %	1.11 %
17:47:53	8633 bps	232 bps	0.01 %	0.00 %
17:46:47	9.9 Kbps	147 bps	0.01 %	0.00 %
17:45:41	12.4 Kbps	276 bps	0.01 %	0.00 %
17:44:35	9444 bps	160 bps	0.01 %	0.00 %
17:43:29	11.2 Kbps	253 bps	0.01 %	0.00 %

Current levels of network traffic generated by a random PC: Tabular, historical Statistics of a random PC

Network Information gathered using *Solarwinds Engineers Edition V-8* Network Monitoring Software (Demo Version)