A joint scientific and technological activity and study on

Grid Enabling Technologies

Sectoral report from the Grid Enabling Technologies study within the ENACTS project

Stavros C. Farantos and Stamatis Stamatiadis (FORTH)

Institute of Electronic Structure and Laser
Foundation for Research and Technology, Hellas

And

Nello Nellari and Djordje Maric (ETH-CSCS)

Swiss Center for Scientific Computing

Version 1.2
December 31, 2002

http://www.enacts.org

Table of Content

List of Figures

List of Tables

1 INTRODUCTION

1.1 Study objectives and report overview

The main objective of this study is to evaluate the current technologies for Grid computing as described in the "Grid Service Requirements" study and have been implemented in several projects and Grid testbeds around the world. Academic applications are mainly covered, with emphasis given in Molecular Sciences.

This study is the third undertaken in the frame of European Network for Advanced Computing Technology for Science project (ENACTS) after the "Grid Service Requirements" and the "High Performance Computing Roadmap". In this work we are aiming:

- To briefly review the most popular software packages for implementing a Grid. Here we examine Globus, Legion and UNICORE since most applications are related to these technologies.
- To consider the current trend of development for Grid technologies and definition of standards.
- To locate the available testbeds for computational Grids worldwide, mainly focusing on those based on Globus, Legion and UNICORE.
- To review the current applications and projects with the established software packages for the computational Grids mentioned above.
- To investigate the present status of Grid Computing for Molecular Simulations and point out the needs for a new programming design of such applications.
- To report personal experience from building a local Grid based on Globus paradigm and UNICORE.

The above targets are accomplished via searching the World Wide Web and personal experience. From the very beginning we encountered the difficulties of dealing with a new subject such as "The Grid" for which the testbeds and applications and projects running on them are continuously changed and extended. In most cases it was difficult to select information about the functions of the testbed or the application other than a general description or future plans after the completion of the project. One can realize the size of the expansion of this field by referring to the review written two years ago by David Henty on the same subject in the frame of the project, "The Direct Initiative" [1], a predecessor of ENACTS. Just three applications are mentioned there compared to numerous testbeds and applications given in Chapter 3 of this report. The same author refers to Globus and Legion Grid models but UNICORE is not mentioned at all in his review. Hence, in this study we follow an austere program focusing mainly on academic testbeds and applications, and we occasionally mention commercial applications of the Grid. Grid applications with very limited information are not presented here.

The results of the Internet search are gathered in three Tables. Each of them summarizes the testbeds and applications which use Globus, Legion and UNICORE, respectively. However, even this classification can not be unique since we found several projects overlapping and most of the testbeds have been constructed to run a specific application, i.e. information and data providers, computational Grids, etc.

In Chapter 2 we briefly review the most commonly used middleware for implementing a Grid: Globus, Legion and UNICORE. We emphasize their main characteristics and their differences and we consider next future developments for Grid technologies. In Chapter 3 we collect in the tables 5 to 7 testbeds of computational and data Grids as well as applications and projects and give short descriptions of their history and functions. This information, which is principally collected from the World Wide Web, is by no means complete. In the Appendix A the URL addresses are given which the interested reader can consult for more information. We analyze the types of applications and the categories of scientists to whom current Grids address and attempt to point out future needs. Chapter 4 deals with the current status of Grids in Molecular Sciences-Physics, Chemistry, and Biology. Chapters 5 and 6 present our personal experience on installing Globus and UNICORE in local computers. Thus, we construct small local computational Grids, which allow us to run a few applications. Finally, in Chapter 7 we summarize the main conclusions of this study.

1.2 Reference

[1] The Grid: A Critical Review of Current Status and Future Directions in Grid Technology. The Direct Initiative, David Henty, EPCC, October 2000.

2 GRID TECHNOLOGIES

2.1 Introduction

The vision of a Grid as a seamless, integrated, computational and collaborative environment embraces different categories of distributed systems. Following the classification introduced by Foster and Kesselman in [1], computational Grids are categorized into five major application classes. In summary those classes identify specific context of applications such as supercomputing applications, high-throughput computing, on-demand computing, data-intensive computing and collaborative computing.

Although each class has its own specific requirements and focus, the general principles that characterize a Grid environment such as heterogeneity, scalability and fault tolerance are common to the different application contexts. The research in Grid technologies is inspired by those principles and, so far, mainly concentrates on the definition of protocols as the basic step for enabling the interoperability between resources belonging to diverse and possibly remote organizations.

Currently, there are various software packages, tools and products that address at different levels and from different perspectives the issues of building a computational Grid. In particular, this document considers Globus [2], Legion [3] and UNICORE [4] as reference technologies in order to compare the objectives, the approaches, and the models that they propose. As it appears in this study, those technologies at the moment are standing out in the most important Grid projects and testbeds.

Finally, the information reported in the following sections serves as the basis for observing the current trends in research and development of Grid computing solutions. For a more extensive and detailed description of the technologies considered in this Chapter, you can refer to the first ENACTS report "Grid Service Requirements".

2.2 Reference Grid technologies

Currently Globus, Legion and UNICORE characterize the majority of Grid testbeds. All three products originally come from a similar context and were motivated by analogous reasons. In fact, they have been initially developed in research projects related to the HPC domain and addressed to a particular audience such as the scientific and academic community. The following paragraph looks at those technologies from a very general perspective considering objectives, concepts and architectural models, and, finally, solution characteristics.

First, it is essential to consider the final objectives that each technology aims to achieve. While Globus and Legion address the problem of building a generic computational Grid, that in a further step can be customized or enhanced in order to support a specific class of applications, UNICORE primary focus is on uniform batch job submission and monitoring even if its functionalities are not limited only to this aspect.

Second, even if from a conceptual point of view Globus and Legion share the same objective, they adopt a completely different approach to solve the problem. The objective can be summarized as building a single virtual machine for handling distributed, heterogeneous computing resources. Globus follows the toolkit approach that distinguishes between local services and global services constructed on top of the first ones. The relationship between local and global services is expressed through a layered architecture adhering to the principles of the "hourglass model" [5]. The neck of the hourglass represents a well-defined interface that provides a uniform access to diverse implementations of local services. Higher-level global services are defined in terms of this interface defined as a transparent interface since, through a structured mechanism, tools and applications can discover and control aspects of the underlying system.

On the other hand, the Legion approach consists of providing an operating system by extending the functionality offered by traditional OS for a single machine, such as a single namespace, a file system, security, process creation and management, inter-process communication, IO, resource management, to a set of heterogeneous and independently administered machines. Legion architecture is designed as an object-based metasystem of distributed and independent entities. In particular, Legion defines a set of core objects that support the basic services needed by the metasystem, such as host objects for processing resources or vault objects for persistent storages. This object oriented based architecture allows a modular design. In this case extending the functionality is achieved through the specialization of basic Legion objects. In addition, the object concept provides a natural border for setting tailored security policies.

Lastly, UNICORE stands for UNiform Interface to COmputer REsources and it can be viewed as a vertical system. Actually UNICORE is based on multi tiered architecture that defines different components for each layer, from the client application, to the Gateway, the Network Job Supervisor and to end with the Target System Interface. The common denominator between each layer is represented by the Abstract Job Object (AJO) that contains a description of actions to be performed on different UNICORE sites, regardless the characteristics of the target machines. Not only the AJO defines a work to be performed by UNICORE, but also it holds the security credentials for accessing and using the various distributed resources.

Finally, concerning the solution characteristics, UNICORE and Legion assumed a similar approach since both intend to provide a complete product, which, once installed and configured for the specific environment, can be immediately ready for being productive.

	Globus	UNICORE	Legion
Objective	Generic computational	Mainly uniform job submission and monitoring	Generic computational
Solution	Open source	Complete product Open source	Complete product
Architecture	Service Toolkit	Multi Tiered Application	OS
Implementation Model	"Hourglass model" and transparent interfaces	Abstract Job Object	Object Oriented metasystem

Table 1: Grid Technologies Summary

In particular, Avaki is a private company that provides a commercial distribution of Legion and also offers technical support and professional services for deploying the Grid software. On the other hand, Globus provides an implementation of the Global Grid Forum [6] specifications, proposals and standards following a toolkit model. This model permits to deploy only the components that are necessary for a specific objective. However, the configuration and installation of Globus components can be a heavy process depending on the customizations required to adapt the code for a specific architecture or for integrating existing applications.

Product	Operating Systems
Globus Toolkit 2.0	Linux, Solaris, IRIX, Compaq/Tru64, AIX (http://www.globus.org/gt2/install/download.html)
Avaki 2.5	Linux, Solaris, IRIX, Compaq/Tru64, AIX, Windows (http://www.avaki.com/products/productfaq.html)
UNICORE 3.6 – 4.0	All OSs depending on the availability of Java 2 Runtime Environment (JRE) and Perl (http://www.UNICORE.org)

Table 2: Operating Systems Supported

2.3 Workload management systems

Workload management systems provide an effective way of controlling and administering single or clustered machines belonging to a homogeneous administrative domain. Therefore, Grid infrastructures can benefit from their functionalities in order to access and to use efficiently local computing resources. In particular, workload management systems are characterized by traditional functions like job queuing and scheduling mechanisms, priority definition, resource monitoring and management.

Currently, there are many commercial and freely available products such as Condor, Mosix, NQS, LSF, PBS, Sun Grid Engine and LoadLeveler (see previous ENACTS report “Grid Service Requirements”). Those have different characteristics and targets, although they all perform conventional batch system functions.

In particular, Condor [7] has emerged as one of the most recurrent batch system in academic Grid testbeds. This product has the following main characteristics: it is free for research purposes, it addresses the problem of supporting a High Throughput Computing (HTC) environment, and it has a preferential relationship with Globus.

First, Condor comes with an academic license that allows the free utilization of the product for research. Second, the Condor Project focuses on mechanism and policies that support High Throughput Computing on large collection of distributively owned computing resources. Some types of problems require a computing environment that is able to provide a large amount of computational power over a long period of time. Therefore, Condor supports automatic resource location and job allocation, check pointing and the migration of processes. Specifically, all the participating computing resources are continuously monitored for determining, on the base of predefined policies, which of them can be considered available. In this way Condor defines a dynamic pool of available resources that can be used for the execution of jobs. The main requirement is that the user source code has to be linked with the Condor libraries in order to allow job migration. Finally, the integration of Condor and Globus has been implemented in the Condor-G system. In particular, Condor-G combines the inter-domain resource management protocols of the Globus Toolkit and the intra-domain resource management methods of Condor [8].

Mosix (www.mosix.org) and openMosix (www.openmosix.org), its open source spin-off, enable the "unification" of Linux systems based on x86 architecture. They are kernel modifications that turn networked Linux systems into a single computer with single process space, distributing all jobs on the available machines and migrating them between nodes to balance the load. The system is constantly monitored on the availability of nodes, the resources provided by them (CPU speed, memory), their connection speeds and their load; decisions are made on the distribution of jobs based on the collected information. The whole procedure is performed by the kernel, making it transparent to the user; the cluster acts as a virtual multiprocessor machine for most applications. Since Mosix is part of the kernel and maintains full compatibility with normal Linux, a user's programs, files, and other resources will all work as before with no changes necessary. Among the applications that cannot benefit from it are those based on threads or shared-memory programming - threads cannot be migrated by Mosix.

Product	Operating Systems
Condor 6.2.1	Linux, IRIX, Solaris, Windows (http://www.cs.wisc.edu/condor/downloads/)
PBS	Linux, FreeBSD, NetBSD, Tru64, HP-UX, AIX, IRIX, Solaris (http://www.openpbs.org/platforms.html)
Sun Grid Engine 5.3	Tru64, HP-UX, AIX, Linux, IRIX, Solaris (http://Gridengine.sunsource.net/project/Gridengine/download.html)

Table 3: Workload Management Systems free products

Product	Operating Systems
LSF 5.0	MacOS, Tru64, UNICOS, HP-UX, AIX, Linux, Windows, IRIX, Solaris (http://www.platform.com/services/support/services/platforms5.0.asp)
LoadLeveler 3.1	AIX (http://www-1.ibm.com/servers/eserver/clusters/software/)

Table 4: Workload management systems commercial products

2.4 Grid products interoperability

At the current stage, the interoperability between computational Grids based on different infrastructures is still an open issue. The main critical points are the compatibility among diverse security models and the translation of the different high level protocols used to specify actions in the Grid environment. For example, Globus defines a Resource Specification Language (RSL), while in UNICORE the Abstract Job Object (AJO) represents the neutral description of the actions to be performed in the Grid. Therefore, a translation process is necessary in order to enable the interaction of the two environments. Currently, Globus, as the most wide spread open source Grid software is considered a target environment by other Grid products. This is the case of the GRIP project where the focus is on the partial interoperability between UNICORE and Globus, intended as the ability to seamlessly submit and monitor jobs from a UNICORE environment to a Globus based Grid.

Currently, the Open Grid Service Architecture (OGSA) proposal suggests a service-oriented approach combined with the Globus toolkit model. The reception of such approach by the Grid community can be a first step for improving considerably the interoperability between different products. For more details see the next paragraphs on Web Services and OGSA.

On the other hand, the products and solutions for implementing a computational Grid have to demonstrate the community how to work with pre-existing applications such as resource managers and scheduling and queuing systems. This aspect of interoperability towards local resources is particularly important, since the participation in a Grid community should not affect significantly existing software installation. Currently, the integration of workload management systems in a Grid product such as Globus, Legion and UNICORE can require important software development efforts. Furthermore, Grid products should allow discovering specific features of the underlying batch system in order to exploit the existing functionality.

2.5 Common requirements for distributed systems

The sharing and the coordination of a dynamic set of resources belonging to different organization represent a more and more common requirement that is definitely not only limited to the academic and research community.

In particular, the necessity of more and more complex interaction between enterprises is pushing towards the definition of standards, solutions and models that address the issue of building reliable distributed systems. This trend is inevitably affecting the development of Grid technologies and the definition of Grid standards.

2.5.1 Web Services

The term Web Services [9] indicates a service-oriented approach based on XML technologies to address distributed computing. In particular, Web Services proposals are industry led and they are being defined by the W3C. They consider especially the description of software components and services, the methods for accessing their functionalities and the methods for discovering the most relevant providers. The main Web Services standards concern the Simple Object Access Protocol (SOAP) [10], the Web Service Description Language (WSDL) [11], the Web Service Inspection (WS-Inspection) [12] and the Universal Description Discovery and Integration (UDDI) [13]. Those proposals are in various stage of the W3C process.

First, SOAP enables applications to access remote applications providing an XML-based remote procedure call mechanism. SOAP is independent by the underlying transport mechanism and it can be used for exchanging any XML information.

Second, WSDL allows describing a web service in term of a XML document. In particular, service interfaces are defined in terms of message structures and operations and then bound to a concrete network protocol and a specific data-encoding format.

Third, WS-Inspection consists of an XML language, the WS Inspection Language (WSIL), and related conventions for locating service description published by a service provider. A service description is usually indicated as a URL to a WSDL or can be a reference to an entry in a UDDI registry. In particular, UDDI provides a directory-based mechanism for registered services that supports publishing, looking up and binding methods for a service.

Finally, other interesting proposals, which interest in particular the inter process relationships, are the Web Service Flow Language (WSFL) [14] and XLANG [15]. In particular, WSFL expresses workflows as services that are defined as a combination of Web Services. On the other hand, XLANG aims to support complex transaction that may involve multiple Web Services.

2.5.2 Semantic Web

The Semantic Web is an initiative of the W3C, with the goal of extending the current Web. As reported by the W3C Semantic Web Activity statement, 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. 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 [16].

The Resource Description Framework (RDF) supports the Semantic Web in much the same way that HTML is the language for the original Web. RDF provides the foundation of metadata interoperability across different resource description communities by defining a syntax specification and a schema specification [17, 18]. In particular, RDF provides a framework for describing and interchanging metadata that is based on the following concepts. Any object uniquely identifiable by a Uniform Resource Identifier (URI) is indicated as a *resource*. A *property* expresses the relationship of values associated with *resources*. A *statement* is a combination of a *resource*, a *property* and a *value*, where a *value* may be atomic in nature, such as text strings or numbers, or it may be other *resources*. Alternatively a *statement* can be indicated as a triple of subject, predicate and object. Finally, RDF defines a method for expressing *resources* into a machine-readable format based on XML. RDF design aims to provide a framework with characteristics of independence, interchangeability and scalability. First, independence since any organization can define its own *resources*. Second, the XML conversion allows the interchange of RDF *statements*. Finally, *statements* are expressed as simple triples of elements thus simplifying the handle and the look up even in large numbers.

2.5.3 Open Grid Service Architecture (OGSA)

The Open Grid Service Architecture (OGSA) [19] is an example of interaction between concepts and technologies developed in different contexts such as the Web and the Grid communities. In particular, OGSA combines the toolkit model proposed by Globus with the service-oriented approach of Web Services. As result, the next Globus Toolkit 3.0, planned for the end of 2002, will partially implement the OGSA proposal.

At the base of this proposal there is the conviction that the creation, management and application of dynamic ensembles of resources and services have a set of fundamental common requirements regardless of the nature of the deployment contexts. In particular, the main issue is how applications can deliver specific *Quality of Services* (QoS) across a collection of resources with heterogeneous and often dynamic characteristics. At this purpose, OGSA suggests a practical solution with an architecture that supports the creation, maintenance and application of ensembles of services maintained by virtual organizations. Conceptually, the term *service* is used abstractly in OGSA for indicating a resource characterized by the set of operations it can perform. Resources can be of different types such as computational or storages resources, networks, programs and databases. Moreover, the interoperability problem is addressed in two steps: the definition of service interfaces and the protocols that can be used to invoke a particular interface.

Standard interface definitions could be sufficient to allow service virtualization. On the other hand, they are not enough to achieve service compositions. At this respect, standard semantics are required to enable proper service interactions. Therefore OGSA defines a *Grid service* as a Web service that provides a set of well-defined interfaces and follows specific conventions. The interfaces address discovery, dynamic service creation, lifetime management, notification and manageability; the conventions concern naming and upgrade ability. Currently, authorization and concurrency control are not considered, while authentication and reliable invocation are defined as service protocol bindings and thus external to the core definitions.

2.5.4 Jini and P2P computing

There are alternative and/or complementary approaches and technologies for Grid computing. Some of them like Jini [20] and Peer To Peer (P2P) computing [21] are currently subject of study and research by various working group at the GGF.

Jini [22] is a technology and at the same time an approach for writing reliable distributed software. The problems addressed are very general such as handling changes over time, reliability, and communication between programs in distributed systems. Therefore the Jini approach could be quite suitable respect to Grid computing needs like location, description and access to distributed and heterogeneous resources. Especially considering an environment where dynamic, varying resources characteristics in both time and space are the norm. Some examples and prototypes that use Jini for implementing Grid solutions are presented in [23] [24] [25].

P2P is a distributed computing model like the client/server architecture. But while the client/server is a centralized architecture, where clients request services and servers provide those services, the P2P is a decentralized architecture, where neither client nor server status exists in a network. Every entity in the network, referred to as a *peer*, has equal status, meaning that an entity can either request a service or provide a service.

The JXTA project [26] is one example for a generic framework for P2P. This framework is based on a set of XML protocols and a Java reference API. In particular the JXTA Protocols Specification describes a set of XML-protocols that provide the basic elements for P2P computing, including: *peer discovery*, *peer groups*, *discovery*, *services*, *data transfer* and *message routing*. JXTA provides a reference implementation for the Java programming language. Moreover, the corresponding implementations for C/C++, Perl and Python are also available.

The OGSAP2P Research Group at GGF intends to compare and analyze P2P grids and traditional server grids like those centered on Web Services and OGSA. Actually the two approaches have significantly different characteristics especially respect to security and trust, connectivity, and interactivity.

2.6 References

[1] Foster, I. and Kesselman, C. (eds.).
”The Grid: Blueprint for a new Computing Infrastructure”.
Morgan Kaufmann, 1999.

[2] Globus
http://www.globus.org/

[3] Legion
http://legion.virginia.edu/

[4] UNICORE
http://www.UNICORE.org/

[5] Foster, I., Kesselman, C. and Tuecke, S.
”The Anatomy of the Grid: Enabling Scalable Virtual Organizations.
International Journal of High Performance Computing Applications”,
15 (3). 200-222. 2001.

[6] Global Grid Forum
http://www.Gridforum.org/

[7] Condor
http://www.cs.wisc.edu/condor/

[8] J. Frey, T. Tannenbaum, M. Livny, I. Foster, S. Tuecke.
”Condor-G: A Computation Management Agent for Multi-Institutional Grids.
Proceedings of the Tenth International Symposium on High Performance
Distributed Computing (HPDC-10)”, IEEE Press, August 2001.

[9] Web Services Activity Statement
http://www.w3.org/2002/ws/Activity

[10] SOAP Version 1.2 Part 0: Primer
W3C Working Draft 26 June 2002
N. Mitra
http://www.w3.org/TR/soap12-part0

[11] Web Services Description Language (WSDL) 1.1
W3C Note 15 March 2001
E. Christensen, F. Curbera, G. Meredith, S. Weerawarana
http://www.w3.org/TR/wsdl

[12] Web Services Inspection Language (WS-Inspection) 1.0
K. Ballinger, P. Brittenham, A. Malhotra, W. A. Nagy, S. Pharies
http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html

[13] UDDI
http://www.uddi.org/

[14] Web Services Flow Language (WSFL) Version 1.0
F. Leymann
http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

[15] XLANG - Web Services for Business Process Design
Satish Thatte
http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm

[16] W3C Semantic Web Activity Statement
http://www.w3.org/2001/sw/Activity

[17] Resource Description Framework (RDF) Model and Syntax Specification
W3C Recommendation 22 February 1999
Ora Lassila, Ralph R. Swick, eds.
http://www.w3.org/TR/REC-rdf-syntax

[18] RDF Vocabulary Description Language 1.0: RDF Schema
W3C Working Draft 30 April 2002
Dan Brickley, R.V. Guha, eds.
http://www.w3c.org/TR/rdf-schema/

[19] Ian Foster, Carl Kesselman, Jeffrey M. Nick and Steven Tuecke.
"The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration",
http://www.globus.org/research/papers/ogsa.pdf

[20] Service Management Frameworks (JINI-RG)
http://www.ggf.org/5_ARCH/jini.htm

[21] Peer To Peer Area
http://www.ggf.org/4_GP/P2P.htm

[22] Jini Community
http://www.jini.org

[23] E. Sharakan, N. Nellari
JAM: Job and Application Manager
http://www-unix.mcs.anl.gov/gridforum/jini/JAM-WhitePaper.pdf

[24] EPCC JiniGrid
http://www.epcc.ed.ac.uk/computing/research_activities/grid/jinigrid/

[25] T. Suzumura, S. Matsuoka, H. Nakada
JiPANG: A Jini-based Computing Portal System
http://matsu-www.is.titech.ac.jp/~suzumura/jipang/

[26] JXTA Project
http://www.jxta.org

3 GRID TESTBEDS AND APPLICATIONS

3.1 Globus

Most of the computational Grid testbeds that we have found worldwide are specific service oriented and only a few of them are for general purposes. Also, some of them use more than one middleware packages. In Table I we tabulate in two independent columns the most popular testbeds and applications or projects that use Globus.

Testbed	Application/Projects
IVDGL	CrossGrid
National Technology Grid i) National Partnership for Advanced Computational Infrastructure (NPACI) ii) National Computational Science Alliance (NCSA)	DataGrid
NASA Information Power Grid (IPG)	
ASCI Distributed Resource Management (DRM)	GridLab
Globus Ubiquitous Supercomputing Testbed Organization (GUSTO)	
National Virtual Observatory (NVO)	AstroGrid
	Comb-e-chem
AstroGrid	GridPP
The European Data Grid	MyGrid
European Grid of Solar Observatories (EGSO)	Reality Grid
DiscoveryNet	
Blue Grid	Earth System Grid (ESG)
NorduGrid	Grid Physics Network (GriPhyN)
	Network for Earthquake Eng. Simulation Grid
	Particle Physics Data Grid (PPDG)
	TeraGrid

Table 5: Computational Grid Testbeds and Applications/Projects for Globus

The iVDGL, NPACI and NCSA are general purposes computational Grids, whereas Grids such as NVO, AstroGrid and EGSO have been made for astronomical applications. Finally, the Blue Grid initialized by IBM is aiming to commercial applications.

With respect to applications/projects most of them are supported by USA and EU research grants, although the e-Science program in Great Britain has generated several projects. Most of the applications involve Particle Physics projects (DataGrid), astronomy (AstroGrid), but also combinatorial chemistry applications (Comb-e-Chem) as well as earth science and meteorological (ESG) projects. For more information the reader should consult the appropriate web pages.

3.1.1 Testbeds

3.1.1.1 International Virtual Data Grid Laboratory (iVDGL)

iVDGL testbed is using Globus and Condor as middleware for its realization. It is a general-purpose facility and in fact it will integrate several current Grid projects in U.S.A. and EU.

This is currently under construction and it is going to be the first Grid covering, eventually, the whole planet with the participation of European Union, United States of America, Japan, Australia and Africa. It will provide an integrated computational resource for major scientific experiments in physics, astronomy, chemistry, biology and engineering. The iVDGL will serve as a unique resource for scientific applications requiring access to Petabytes of data and beyond. It started Sep. 25, 2001 that it is awarding $13.65M over five years to a consortium of 15 universities and four national laboratories to create the International Virtual Data Grid Laboratory (iVDGL).

The iVDGL will provide a global computing resource for several leading international experiments in physics and astronomy, including the Laser Interferometer Gravitational-wave Observatory (LIGO), the ATLAS and CMS experiments at CERN, the Sloan Digital Sky Survey (SDSS), and the proposed National Virtual Observatory (NVO). For these projects, particularly the CERN experiments whose members number in the thousands and whose data collections are expected to reach 100 Petabytes early in the next decade, the powerful global computing resources available through the iVDGL will enable new classes of data intensive algorithms that will lead to new scientific results. Other application groups affiliated with the NSF supercomputer centers and EU projects will also take advantage of its resources. Sites in Europe and the U.S. will be linked together by a multi-gigabit per second transatlantic link funded by a companion project in Europe.

3.1.1.2 National Technology Grid

The U.S.A. National Partnership for Advanced Computational Infrastructure (NPACI) and the National Computational Science Alliance (NCSA) are building the prototypes of a National Technology Grid. Both initiatives (funded via the U.S. National Science Foundation's PACI program) are deploying the Globus Toolkit across a wide range of facilities and resources, including supercomputing centers, research labs, and college and university campuses.

The goal of NSF's PACI program is to build, operate, and encourage the use of a prototypical computational Grid that will support distributed scientific and engineering applications and form the basis of a national computational infrastructure in the same way that the NSFnet formed the basis of today's Internet.

3.1.1.3 PACI

Partnerships for Advanced Computational Infrastructure (PACI) are programs of the U.S.A. National Science Foundation's Directorate for Computer and Information Science and Engineering (CISE). Started in 1997, PACI is creating the foundation for meeting the expanding need for high-end computation and information technologies required by the U.S. academic research community. PACI partners contribute to the development of a new information infrastructure by developing, applying and testing the necessary software, tools, and algorithms which contribute to the further growth of this "national Grid" of interconnected high-performance computing systems. There are two national partnerships in PACI: the National Computational Science Alliance (Alliance), led by the National Center for Supercomputing Applications in Champaign, IL; and the National Partnership for Advanced Computational Infrastructure (NPACI), led by the San Diego Supercomputer Center in San Diego.

3.1.1.4 ALLIANCE

The U.S.A. National Computational Science Alliance (Alliance) is a nationwide partnership of more than 50 academic, government and business organizations working together to prototype an advanced computational infrastructure for the new century. This infrastructure, called the Grid, is rapidly developing into a ubiquitous, pervasive, national-scale information infrastructure that links supercomputers, virtual environments, scientific instruments, large databases and research teams.

Started in 1997, the Alliance is one of two national partnerships funded by the National Science Foundation's Partnerships for Advanced Computational Infrastructure (PACI) program and receives cost sharing at partner institutions. The National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign leads the Alliance under the direction of NCSA and Alliance Director.

The Alliance includes an Enabling Technologies (ET) team, which builds the infrastructure of the Grid, and five Application Technologies (AT) teams, which drive the development of applications and tools to meet the needs of specific scientific disciplines. The Partners for Advance Computational Services (PACS) provide computing resources and training to the national high-performance computing community. The Education, Outreach and Training Partnership for Advanced Computational Infrastructure (EOT-PACI) spans both PACI partnerships and brings PACI innovations to new audiences, including schools, local, state and federal government, and underserved populations.

3.1.1.5 NASA Information Power Grid

The National Aeronautics and Space Administration is using the Globus Toolkit as a basis for the NASA Information Power Grid (IPG). The IPG joins supercomputers and storage devices owned by participating organizations into a single, seamless computing environment.

Part of a joint effort among leaders within government, academia, and industry, the IPG will help NASA scientists collaborate to solve important problems facing the world in the 21st century. Just as the web makes information anywhere available everywhere, the IPG will someday give researchers and engineers to distant supercomputing resources and data repositories whenever they need them.

3.1.1.6 ASCI Distributed Resource Management (DRM)

The Department of Energy's Lawrence Livermore, Los Alamos, and Sandia National Laboratories house the Accelerated Strategic Computing Initiative (ASCI): a key component of the U.S. nuclear program. These labs are constructing a testbed for a Distributed Resource Management (DRM) system which will manage the labs' combined computational resources. The DRM team is using a version of the Globus Toolkit that supports Kerberos security to build this testbed.

3.1.1.7 GUSTO

The GUSTO testbed served as the testbed for early versions of the Globus Toolkit. GUSTO was created by a set of partners that it is called the Globus Ubiquitous Supercomputing Testbed Organization (GUSTO). In February 2000, GUSTO included 125 sites in 23 countries, representing one of the largest computational environments ever constructed.

3.1.1.8 National Virtual Observatory (NVO)

Create and apply production Grids for data analysis in astronomy. This project describes the NVO’s scientific opportunities and Information Technology challenges. It presents an implementation strategy and management plan to create an initial federation, and a foundation for further tools and applications. The NVO will challenge the astronomical community with new opportunities for scientific discovery and it will challenge the information technology community with a visionary but achievable goal of distributed access and analysis of voluminous data collections. The scope of the effort is international.

3.1.1.9 AstroGrid

AstroGrid is a United Kingdom Grid initiative to construct, manage, and utilize large numbers of astronomical data. The aim of the UK Astro-Grid project is to focus on short term deliverables, both relevant application tools and the federation, by the data centres that manage them, of key sky survey datasets, namely: (a) SuperCOSMOS, Sloan, INT-WFC and UKIRT WFCAM; (b) XMM-Newton and Chandra; (c) and SOHO, Cluster and IMAGE. The differences between the data types involved in these federations means that each brings distinct challenges, whose solutions will shed light on generic problems to be faced by the developing global "Virtual Observatory". Initially a three year program started in April 2001 and estimated to cost £5.5M, which will add value to existing UK astronomy resources in the short term, as well as positioning the community strongly with respect to wider Grid initiatives, in astronomy and beyond. A key aim of astronomers world-wide is to stitch together essentially all such databases in a "Virtual Observatory" together with intelligent discovery agents so that the archive becomes like a second sky. This is a very ambitious aim, with many technical and organizational challenges.

3.1.1.10 European Grid of Solar Observatories (EGSO)

Build a Grid infrastructure that will allow a Virtual Observatory, unifying interfaces to astronomy databases and providing remote access as well as assimilation of data.

3.1.1.11 DiscoveryNet

DiscoveryNet is a U.K. e-Science testbed for processing data from high throughput devices in biochemistry, environmental studies, etc.

3.1.1.12 Blue Grid

This is an IBM initiative in association with the Department of Energy of US to build a Grid. It started with just two computer systems, but in the future it will include others at the national laboratories of Lawrence Berkeley, Argonne, Oak Ridge and Pacific Northwest.

The first two computers on the Science Grid is a 3,328-processor Unix system at the DOE's National Energy Research Scientific Computing Center (NERSC), which IBM designed, and a smaller and more recent 160-processor Intel system. In addition, a storage system at NERSC with 1.3 petabytes of space--about 30,000 times that of a desktop computer--will be attached.

A core group of NERSC supercomputer users will be able to tap into the system, which originally wasn't expected to become a Grid system until 2004.

IBM and others are making efforts to rework the idea as a tool for business computing as well. Hewlett-Packard, Compaq Computer, Sun Microsystems, Microsoft and others also are pushing Grid computing. IBM expects other companies' systems will be used on the Grid as well.

3.1.1.13 NordGrid

The NordGrid is the pioneering Grid project in Scandinavia. The purpose of the project is to create a Grid computing infrastructure in Nordic countries, operate a functional testbed and expose the infrastructure to end-users of different scientific communities. Project participants include universities and research centers in Denmark, Sweden, Finland and Norway. The cornerstone of the infrastructure adopted at NorduGrid is the Globus toolkit developed at Argonne National Laboratory and University of Southern California. The Globus toolkit is widely accepted as being the defacto standard for Grid computing and provides a collection of robust protocols, low level services and libraries. It is however missing several important high level services like: grid-level scheduler, grid-level authorization system, grid-level accounting and quotas, job data stagein/stageout, grid application development toolkits, user-friendly grid entry points. With the need to provide working production system, NorduGrid has developed it's own solution for the most essential parts. An early prototype implementation of the proposed architecture adopted by the NorduGrid is being tested and further developed. It consists of: Information System based on Globus Monitoring and Discovery Service, User Interface integrated with Resource Broker for submitting jobs with complex enough requirements, Grid Manager providing interface for complex job submission based on Globus Resource Allocation Manager and GridFTP protocol developed by Globus team. Aiming at simple but yet functional system capable of handling common computational problems encountered in the Nordic scientific communities we were choosing simple but still functional solutions with necessary parts implemented first.

3.1.2 Applications/Projects

3.1.2.1 CrossGrid

The primary objective of the CrossGrid Project is to further extend the Grid environment to a new category of applications of great practical importance, and into eleven (11) new European countries.

The main objective concerning the targeted applications is to elaborate the methodology for development of, and generic architecture of, large-scale Grid-enabled applications for simulation and visualization that require real time response. The challenges refer to the distribution of source data, simulation and visualization tasks, virtual time management, simulation/visualization rollback due to user actions, and platform independent VR visualization. Examples of these applications are: interactive simulation and visualization for surgical procedures, flooding crisis team decision support systems, distributed data analysis in high-energy physics, air pollution combined with weather forecasting. A visualization engine should be developed and optimized for these applications.

To enable efficient development of this category of applications for the Grid environment, new tools for verification of parallel source code, performance prediction, performance evaluation and monitoring are needed. This, in turn, requires extension of the Grid by new components for application-performance monitoring, efficient distributed data access, and specific resource management. Users should be able to run their applications on the Grid in an easy and transparent way, without needing to know details of the Grid structure and operation. CrossGrid is developing user-friendly portals and mobile personalized environments and is integrating new components into the Grid and application development tools.

The elaborated methodology, generic application architecture, programming environment, and new Grid services will be validated and tested thoroughly on the CrossGrid testbeds. This will result in further extension of the Grid across Europe. In the CrossGrid development the researchers will exploit all the available achievements of DataGrid, EuroGrid and other related projects in a way that enables their interoperability. CrossGrid will closely collaborate with DataGrid. Develop, implement, and exploit new Grid components for interactive compute and data intensive applications. This is a EU support project.

3.1.2.2 DataGrid

The European DataGrid is a project funded by the European Union with the aim of setting up a computational and data-intensive Grid of resources for the analysis of data coming from scientific exploration. Next generation science will require co-ordinated resource sharing, collaborative processing and analysis of huge amounts of data produced and stored by many scientific laboratories belonging to several institutions.

The main goal of the DataGrid initiative is to develop and test the technological infrastructure that will enable the implementation of scientific “co laboratories” where researchers and scientists will perform their activities regardless of geographical location. It will also allow interaction with colleagues from sites all over the world as well as the sharing of data and instruments on a scale previously unattempted. The project will devise and develop scalable software solutions and testbeds in order to handle many PetaBytes of distributed data, tens of thousand of computing resources (processors, disks, etc.), and thousands of simultaneous users from multiple research institutions.

The DataGrid initiative is led by CERN, the European Organization for Nuclear Research, together with five other main partners and fifteen associated partners. The project brings together the following European leading research agencies: the European Space Agency (ESA), France's Centre National de la Recherche Scientifique (CNRS), Italy's Istituto Nazionale di Fisica Nucleare (INFN), the Dutch National Institute for Nuclear Physics and High Energy Physics (NIKHEF) and UK's Particle Physics and Astronomy Research Council (PPARC). The fifteen associated partners come from the Czech Republic, Finland, France, Germany, Hungary, Italy, the Netherlands, Spain, Sweden and the United Kingdom.

DataGrid is an ambitious project. Its development benefits from many different kinds of technology and expertise. The project spans three years, from 2001 to 2003, with over 200 scientists and researchers involved. The EU funding amount to 9.8 million euros.

The first and main challenge facing the project is the sharing of huge amounts of distributed data over the network infrastructure that is currently available. The DataGrid project does not start from scratch in this challenge: it relies upon emerging computational GRID technologies that are expected to make feasible the creation of a giant computational environment out of a distributed collection of files, databases, computers, scientific instruments and devices. The DataGrid project is divided into twelve Work Packages distributed over four Working Groups: Testbed and Infrastructure, Applications, Computational & DataGrid Middleware, Management and Dissemination.

3.1.2.3 GridLab

Grid technologies and applications. A EU support project.

1. *Co-development of Infrastructure and Applications:*

A balanced program is proposed with co-development of a range of Grid applications (based on Cactus, the leading, widely used Grid-enabled open source application framework, and Triana, a dataflow framework used in gravitational wave research) alongside infrastructure development, working on transatlantic testbeds of varied supercomputers and clusters. This practical approach ensures that the developed software truly enables easy and efficient use of Grid resources in a real environment. GridLab will maintain and upgrade the testbeds through deployment of new infrastructure and large-scale application technologies as they are developed. All deliverables will be immediately prototyped and continuously field tested by several user communities. The focus on specific application frameworks allows one immediately to create working Grid applications to gain experience for more generic components developed during the project.

2. *Dynamic Grid Computing:*

GridLab will develop capabilities for simulation and visualization codes to be self aware of the changing Grid environment, and to be able to fully exploit dynamic resources for fundamentally new and innovative applications scenarios. For example, the applications themselves will possess the capability to migrate from site to site during the execution, both in whole or in part, to spawn related tasks, and to acquire/release additional resources demanded by both the changing availabilities of Grid resources, and the needs of the applications themselves.

GridLab will unify these elements in developing innovative, practical, Grid computing technologies, which will then be quickly and easily adopted and exploited by applications from many different research and engineering fields. Specific key objectives of this project are to: Design and develop a Grid Application Toolkit (GAT), to provide core, easy to use functionality through a carefully constructed set of generic APIs for both simulation codes and Grid software. The GAT will contain independent modules for handling many different aspects of Grid programming, including simulation, performance and Grid monitoring, resource brokering and selecting, performance prediction, interaction with information servers, security, notification, collaboration, data handling, remote visualization, and remote application steering. Simultaneously enhance real applications for the Grid, implementing new dynamic simulation scenarios using the GAT. Both Cactus and Triana will be extended to integrate and exploit GAT elements, making Grid Computing easily exploitable by a wide range of applications. The simulation driven, compute intensive applications are fundamentally different from the highly data driven applications in many other Grid projects (e.g., DataGrid, GriPhyN, EuroGrid). Develop and test Grid infrastructure/applications on real testbeds, constructed by linking heterogeneous collections of supercomputers and other resources spanning Europe and the USA, using and extending existing testbeds. Interoperability with different testbeds will be ensured by also using production testbeds in the USA, driving international high-speed network connectivity. Testing will be carried out by the project and by several large, closely related user communities, including an EU Astrophysics Network, and various multidisciplinary US funded collaborations.

User Scenarios

The end technology developed through this project will enable scenarios, such as the following hypothetical examples, to become reality.

1. *Gravitational Wave Detection and Analysis:*

The gravitational wave detector network, including GEO600 in Germany, collects a TByte of data each day, which must be searched using different algorithms for possible events such as black hole or neutron star collisions, or pulsar signals. Routine real time analysis of gravitational wave data from the Hanover detector identifies a burst event, but this standard analysis reveals no information about the burst location. To obtain the location, desperately required by astrophysicists for turning their telescopes to view the event before it fades, a large series of templates must be cross-correlated against the detector data. An Italian astrophysicist accesses the GEO600, and using the performance tool finds that 3 TFlops/s is needed to analyze the 100GB of raw data in the required hour. Local resources are insufficient, so using the brokering tool, she locates the fastest available machines around the world. She selects five suitable machines, and with scheduling and data management tools, data is moved, executables created and the analysis starts.

In an Amsterdam bar twenty minutes later, an SMS message from the portal's notification tool, informs her that one machine is overloaded, breaking the runtime contract. She connects with her PDA to the portal, and instructs the migration tool to move this part of the analysis to a different machine. Within the specified hour, a second SMS message tells her analysis is finished, and the resulting data is now on her local machine. Using this location data, observatories are able to find and view an exceptionally strong gamma-ray burst, characteristic of a collision of neutron stars.

2. *Numerical Relativity:*

A single simulation of an astrophysical event, e.g. black hole or neutron star collisions, ideally requires over TByte and TFlop resources, not yet available on a single machine. Learning more about the detected burst requires cross-correlating detector data with custom wave templates from full-scale neutron star simulations. Sufficiently accurate templates require running large-scale simulations too big to fit on any current supercomputer. German members of international numerical relativity collaboration are tasked with creating collision templates for ten different neutron star mass combinations. They access the web-based Simulation Portal, selecting required code modules and building parameter files with the code composition tool. The performance prediction tool estimates that each simulation requires 1024 GigaBytes of memory and 100,000 GigaFlops, with an additional 500,000 GigaFlops required for processing data to create signal templates.

The brokering tool finds that no single machine in the Simulation Testbed can supply enough memory, but locates two machines which can be connected to form a large enough virtual supercomputer, the dynamic Grid monitoring tool indicates an acceptable bandwidth between them. The scheduling tool stages the five runs to appropriate queues on the machines, and the first simulation starts. The time-consuming task of creating templates is handled by spawning simulations to smaller machines dynamically located by the broker, at each time step data is streamed to a series of networked computers for analysis, creating a simulation vector using available machines on the Grid. Collaborators around the world connect to the portal using networked workstations, home PCs and modems, as well as the latest wireless PDAs and mobile phones. They are able to use various remote access tools to visualize data, monitor performance and simulation properties, and interactively steer the simulation.

3.1.2.4 ASTROGRID

AstroGrid is a £5M project aimed at building a data-Grid for UK astronomy, which will form the UK contribution to a global Virtual Observatory. There are seven institutions involved in the AstroGrid consortium:

- School of Computer Science, Queens University of Belfast
- Institute of Astronomy, University of Cambridge
- Institute for Astronomy, University of Edinburgh
- Dept of Physics and Astronomy, University of Leicester
- Mullard Space Science Laboratory, University of London
- Jodrell Bank Observatory, University of Manchester
- Space Data Division, Rutherford Appleton Laboratory

3.1.2.5 Comb-e-chem

Support combinatorial synthesis of new compounds by combining structure and property data sources within a Grid-based information-and knowledge-sharing environment. This is a U.K. e-Science Program.

The goal of the project is to develop an e-Science testbed that integrates existing structure and property data sources, and augments them within a Grid-based information and knowledge environment. The synthesis of new chemical compounds by combinatorial methods provides major opportunities for the generation of large volumes of new chemical knowledge and is the principal drive behind the project. An extensive range of primary data needs to be accumulated, integrated and relationships modeled, so that maximum knowledge can be derived. The service-based Grid-computing infrastructure extends to devices in the laboratory and involves enriched systems, (including multimedia and live metadata), full support for provenance and innovative techniques for automation throughout the environment. The results of the project will impact on the design of materials through the prediction of properties and the identification of suitable compounds in a variety of applications.

3.1.2.6 GRIDPP

GridPP is a collaboration of Particle Physicists and Computing Scientists from the UK and CERN, who are building a Grid for Particle Physics.

There are three main developments within GridPP:

- Grid-enabled applications; and
- Grid software (middleware);
- Provision of computing infrastructure in the UK and CERN.

GridPP will enable testing of a prototype Grid of significant scale, providing resources for the LHC experiments ALICE, ATLAS, CMS and LHCb, the US-based experiments BaBar, CDF and D0, and lattice theorists from UKQCD.

3.1.2.7 MYGRID

To date Grid development has focused on the basic issues of storage, computation and resource management needed to make a global scientific community information services and tools accessible in a high performance environment. However, from the e-Science viewpoint, the purpose of the Grid is to deliver a collaborative and supportive environment that enables geographically distributed scientists to achieve research goals more effectively, while enabling their results to be used in developments elsewhere. MyGrid will integrate and extend technologies to provide an advanced e-Science environment. It will design, prototype and demonstrate a testbed that: supports the scientific process of experimental investigation, evidence accumulation and result assimilation; supports the scientist's use of the community's information; and directly supports scientific collaboration at the core of research, allowing dynamic groupings to tackle emergent research problems. The objective is to present the necessary infrastructure within a Grid environment as an e-Scientist's Workbench that actively supports the scientific lifecycle.

The particular focus of MyGrid is on data-intensive e-Science and the provision of a distributed environment that supports the in silico experimental process. The application is bioinformatics, specifically post-genomic functional analysis, where the building of value-added repositories and their use in day-to-day research will only be truly viable when scientists have efficient tools that allow them seamlessly to: link together databases and analytical tools, extract relevant information from free texts, and harness available computational resources for CPU-intensive tasks.

The MyGrid project aims to build a demonstrator infrastructure that supports a personalised problem-solving environment for an e-Scientist. The vision is of a "lab book" environment where the e-Scientist can construct in silico experiments, and find and adapt others, store partial results in local data repositories and have their own view on public repositories, and be better informed as to the provenance and the currency of the tools and data directly relevant to their experimental space. The Grid becomes egocentrically based around the Scientist – MyGrid.

3.1.2.8 RealityGrid

Enable the realistic modeling of complex condensed matter systems at the molecular and mesoscale levels. This is a UK project.

RealityGrid proposes to extend the concept of a Virtual Reality centre across the Grid and links it to massive computational resources at high performance computing centres and experimental facilities. Using Grid technology to closely couple high throughput experimentation and visualization, RealityGrid will move the current bottleneck out of the hardware and back to the human mind. A twin-track approach will be employed within RealityGrid: a "fast track" will use currently available Grid middleware to construct a working Grid, while a "deep track" will involve computer science teams in harnessing leading-edge research to create a robust and flexible problem-solving environment in which to embed the RealityGrid.

RealityGrid is a collaboration between distinguished teams of physical scientists, computer scientists and software engineers. To meet its objectives, it will utilise a computing environment built around the UK's most advanced computing technology and infrastructure. The developers of RealityGrid are now looking for a large number of applicants to join this exciting project, which also has direct links into various existing scientific programmes in all of these centres.

The collaborating universities, each of which is already exceptionally well resourced, comprise:

- Queen Mary, University of London (Centre for Computational Science and Department of Chemistry);
- University of Edinburgh (Department of Physics and Astronomy and Edinburgh Parallel Computing Centre);
- University of Manchester (Manchester Computing and Department of Computer Science);

- Imperial College, University of London (Department of Computing);
- Loughborough University (Department of Computer Science and Department of Mathematical Sciences); and
- University of Oxford (Department of Materials).

Four of these universities are now UK e-Science Centres:

- National e-Science Centre (University of Edinburgh);
- London e-Science Centre (University of London);
- e-Science North West (University of Manchester); and
- Oxford e-Science Centre (University of Oxford).

The collaborating organisations include:

- The Computation for Science Consortium (which operates the UK's national supercomputing facility CSAR in Manchester),
- Schlumberger Cambridge Research Ltd,
- Edward Jenner Institute for Vaccine Research,
- Silicon Graphics Inc.,
- Advanced Visual Systems Inc., and
- Fujitsu Ltd.

Through Schlumberger Cambridge Research Ltd. It is expected to leverage off the Cambridge e-Science Centre. The project also has a strong international dimension, involving collaborations with several US Universities and National Laboratories as well as European supercomputing centres, a substantial budget for travel and funds to host three Visiting Fellows from overseas. There is also an International Advisory Board that will ensure the project is judged against world-class standards.

3.1.2.9 Earth System Grid (ESG)

Delivery and analysis of large climate model datasets for the climate research community.

The Earth System Grid II (ESG) is a new research project sponsored by the U.S. DOE Office of Science under the auspices of the Scientific Discovery through Advanced Computing program (SciDAC). The primary goal of ESG is to address the formidable challenges associated with enabling analysis of and knowledge development from global Earth System models. Through a combination of Grid technologies and emerging community technology, distributed federations of supercomputers and large-scale data & analysis servers will provide a seamless and powerful environment that enables the next generation of climate research.

Project Abstract: High-resolution, long-duration simulations performed with advanced DOE SciDAC/NCAR climate models will produce tens of petabytes of output. To be useful, this output must be made available to global change impacts researchers nationwide, both at national laboratories and at universities, other research laboratories, and other institutions. To this end, it is proposed to create a new Earth System Grid, ESG-II - a virtual collaborative environment that links distributed centers, users, models, and data. ESG-II will provide scientists with virtual proximity to the distributed data and resources that they require to perform their research. The creation of this environment will significantly increase the scientific productivity of U.S. climate researchers by turning climate datasets into community resources. In creating ESG-II, a range of Grid and collaboration technologies will be integrated and extended, including the DODS remote access protocols for environmental data, Globus Toolkit technologies for authentication, resource discovery, and resource access, and Data Grid technologies developed in other projects. ESG-II will develop new technologies for (1) creating and operating "filtering servers" capable of performing sophisticated analyses, and (2) delivering results to users. In so doing, it will simultaneously contribute to climate science and advance the state of the art in collaboratory technology. The results are expected to be useful to numerous other DOE projects. The three-year R&D program will be undertaken by a talented and experienced team of computer scientists at five laboratories (ANL, LBNL, LLNL, NCAR, ORNL) and one university (ISI), working in close collaboration with climate scientists at several sites.

3.1.2.10 GRIPHYN

The GriPhyN (Grid Physics Network) project brings together an outstanding team of information technology (IT) researchers and experimental physicists to provide the IT advances required to enable Petabyte-scale data intensive science in the 21st century. Driving the project are unprecedented requirements for geographically dispersed extraction of complex scientific information from very large collections of measured data. To meet these requirements, which arise initially from the four physics experiments involved in this project but will also be fundamental to science and commerce in the 21st century, the GriPhyN team will pursue IT advances centered on the creation of Petascale Virtual Data Grids (PVDG) that meet the data-intensive computational needs of a diverse community of thousands of scientists spread across the globe.

Their team is composed of seven IT research groups and members of four NSF-funded frontier physics experiments. They believe that only an integrated research effort will provide the coordination and tight feedback from prototypes and tests that will enable both communities to meet their goals. The four physics experiments are about to enter a new era of exploration of the fundamental forces of nature and the structure of the universe. The CMS and ATLAS experiments at the Large Hadron Collider will search for the origins of mass and probe matter at the smallest length scales; LIGO (Laser Interferometer Gravitational-wave Observatory) will detect the gravitational waves of pulsars, supernovae and in-spiraling binary stars; and SDSS (Sloan Digital Sky Survey) will carry out an automated sky survey enabling systematic studies of stars, galaxies, nebulae, and large-scale structure.

The data analysis for these experiments presents enormous IT challenges. Communities of thousands of scientists, distributed globally and served by networks of varying bandwidths, need to extract small signals from enormous backgrounds via computationally demanding analyses of datasets that will grow from the 100 Terabyte to the 100 Petabyte scale over the next decade. The computing and storage resources required will be distributed, for both technical and strategic reasons, across national centers, regional centers, university computing centers, and individual desktops. The scale of this task, far outpaces our current ability to manage and process data in a distributed environment, requiring fundamental advances in many areas of computer science.

To meet these challenges, GriPhyN will pursue an aggressive program of fundamental IT research focused on realizing the concept of Virtual Data. Virtual Data encompasses the definition and delivery to a large community of a (potentially unlimited) virtual space of data products derived from experimental data. In this virtual data space, requests can be satisfied via direct access and/or computation, with local and global resource management, policy, and security constraints determining the strategy used. Overcoming this challenge and realizing the Virtual Data concept requires advances in three major areas in which GriPhyN will target IT advances:

Virtual data technologies. Advances are required in information models and in new methods of cataloging, characterizing, validating, and archiving software components to implement virtual data manipulations.

Policy-driven request planning and scheduling of networked data and computational resources. Mechanisms for representing and enforcing both local and global policy constraints and new policy-aware resource discovery techniques are required.

Management of transactions and task-execution across national-scale and worldwide virtual organizations. New mechanisms are needed to meet user requirements for performance, reliability, and cost. Agent computing will be important to permit the Grid to balance user requirements and Grid throughput, with fault tolerance.

GriPhyN is primarily focused on achieving the fundamental IT advances required to create PVDGs, but will also work synergistically on creating PVDG software systems for community use, and applying PVDG technologies to enable distributed, collaborative analysis of data. In the process, a new generation of interdisciplinary scientists with expertise in this critical area will be educated. These goals are being pursued by an exceptional team that includes computer scientists with substantial expertise in key technology areas as well as members of the four experiments.

In order to apply these advances to the experimental data analysis problems, GriPhyN will package them in a multi-faceted, domain-independent Virtual Data Toolkit, and use this toolkit to prototype the PVDGs and support the CMS, ATLAS, LIGO, and SDSS analysis tasks. This combination of IT advances, toolkit development, and PVDG development will deliver new data analysis capabilities to the entire research community, to educators, and to students, enabling revolutionary discoveries in both fundamental computer science and physics disciplines. The challenges addressed by this program are not unique to physics, but are also encountered in biology (e.g., the human genome project), medicine (e.g., the human brain project), environment (e.g., the Earth Observing

System), and many other areas. GriPhyN's results and resources thus could drive future scientific advances in these disciplines.

3.1.2.11 Network for Earthquake Engineering Simulation Grid

The aim is to create and apply a production Grid for earthquake engineering. NEESGrid is an integrated network that will link earthquake engineering research sites across USA, provide data storage, facilities and repositories, and offer remote access to the latest research tools. It was established in 2001 by a consortium of institutions led by NCSA, University of Illinois at Urbana-Champaign.

3.1.2.12 Particle Physics Data Grid (PPDG)

The Particle Physics Data Grid collaboration was formed in 1999 because its members were keenly aware of the need for Data Grid services to enable the worldwide distributed computing model of current and future high-energy and nuclear physics experiments. Initially funded from the NGI initiative and later from the DOE MICS and HENP programs, it has provided an opportunity for early development of the Data Grid architecture as well as evaluating some prototype Grid middleware.

With the current SciDAC project (Particle Physics Data Grid Collaboratory Pilot), PPDG will develop, acquire and deliver vitally needed Grid-enabled tools for data-intensive requirements of particle and nuclear physics. Novel mechanisms and policies will be vertically integrated with Grid middleware and experiment-specific applications and computing resources to form effective end-to-end capabilities. PPDG is a collaboration of computer scientists with a strong record in distributed computing and Grid technology, and physicists with leading roles in the software and network infrastructures for major high-energy and nuclear experiments. Together they have the experience, knowledge and vision in the scientific disciplines and technologies required to bring Grid-enabled data manipulation and analysis capabilities to the desk of every physicist. They have outlined a three-year program taking full advantage of the strong driving force provided by now running physics experiments, ongoing Computer Science projects and recent advances in Grid technology. Their goals and plans are ultimately guided by the immediate, medium-term and longer-term needs and perspectives of the physics experiments, some of which will run for at least a decade from 2006 and by the research and development agenda of the CS projects involved in PPDG and other Grid-oriented efforts.

The three real data intensive computing applications areas covered by the project are:

- High Energy Physics (HEP), led by CERN,
- Biology and Medical Image processing, led by CNRS (France),
- Earth Observations (EO) led by the European Space Agency.

3.1.2.13 TeraGrid

TeraGrid is a multi-year effort to build and deploy the world's largest, fastest, most comprehensive, distributed infrastructure for open scientific research. When completed, it will include 13.6 teraflops Linux Cluster computing power distributed at the four TeraGrid sites: the National Center for Supercomputing Applications (NCSA), the San Diego Supercomputer Center (SDSC), Argonne National Laboratory, and the California Institute of Technology (Caltech).

Other TeraGrid resources will include facilities capable of managing and storing more than 450 terabytes of data, high-resolution visualization environments, and toolkits for Grid computing. All of these components will be tightly integrated into an information infrastructure and connected by the fastest network ever deployed—16 times faster than today's fastest network. TeraGrid is funded by a $53 million award from the NSF.

NCSA at the University of Illinois at Urbana-Champaign will provide the bulk of the computing power for the TeraGrid. When completed, NCSA's TeraGrid computing system will consist of IBM Linux clusters powered by the next generation of Intel® Itanium™ processors, code named McKinley. NCSA's TeraGrid cluster will have a peak performance of 8 teraflops, combining the systems funded through the TeraGrid award (6 teraflops) and other NCSA cluster systems (2 teraflops).

3.2 Legion

The most popular testbeds and applications or projects based on Legion are listed in the following table.

Testbed	Application/Projects
NPACI-net	Centurion
Centurion	NPACI
	NASA
	DoD

Table 6: Computational Grid Testbeds and Applications/Projects for Legion

3.2.1 Testbeds

3.2.1.1 NPACI-net

NPACI-net is a Legion testbed based at the University of Virginia. It runs on a wide variety of architectures (SGI, rs6000, solaris, etc.) and at any given point contains a portion of the Centurion testbed as well as various other machines.

3.2.1.2 Centurion

This is a testbed at University of Virginia and consists of the following machines

Vital statistics:

- 128 x 533 MHz DEC Alphas, 128 processors, 32 GB RAM, 768 BG disk
- 128 x Dual 400 MHz Pentium2, 256 processors, 32 GB RAM, 1 TB disk

The Centurion testbed has the following characteristics:

- 240 + GFlops of peak performance
- Total of 64 GB RAM
- More than 1.7 TB disk
- 100 Mb switched Ethernet
- 1.28 Gb Myrinet

3.2.2 Applications/Projects

The following projects are run in the Centurion testbed.

3.2.2.1 Hawley MHD code simulation

An astronomical code used for simulating gas accretion disks.

3.2.2.2 MM5

A mesoscale weather modeling code used for both research into weather prediction programs and for operational predictions for organizations such as the United States Air Force.

3.2.2.3 Charmm

Chemistry at Harvard Molecular Mechanics (CHARMM, http://yuri.harvard.edu/). A program for macromolecular dynamics and mechanics, designed to investigate the structure and dynamics of large molecules. The software in this particular application was developed by Charles Brooks at the Scripps Research Institute.

3.2.2.4 Axially Symmetric Direct Simulation Monte Carlo code

Developed by G.A. Bird and modified for the Directed Vapor Deposition research at the University of Virginia's Intelligent Processing of Materials Laboratory.

3.2.2.5 Ocean Benchmark: Shallow Water Ocean Simulator

Code provided by Northrop-Grumman (NAVO), contact: Steve Piacsek, NAVO Results of running this application on Centurion.

3.3 UNICORE

The most popular testbeds and applications or projects based on UNICORE are listed in the following table.

Testbed	Application/Projects
EuroGrid	BioGrid
	MeteoGrid
	CAEGrid

Table 7: Computational Grid Testbeds and Applications/Projects for UNICORE

3.3.1 EUROGRID-Application Testbed for European GRID computing

The EUROGRID testbed is a shared cost Research and Technology Development project (RTD) granted by the European Commission (grant no. IST 20247). It is part of the Information Society Technologies Programme (IST). The grant period is Nov 1, 2000 till Oct 31, 2003. The EUROGRID project will demonstrate the use of GRIDs in selected scientific and industrial communities, address the specific requirements of these communities, and highlight the benefits of using GRIDs.

The objectives of the EUROGRID project are:

- To establish a European GRID network of leading High Performance Computing centres from different European countries.
- To operate and support the EUROGRID software infrastructure. The EUROGRID software will use the existing Internet network and will offer seamless and secure access for the EUROGRID users.
- To develop important GRID software components and to integrate them into EUROGRID (fast file transfer, resource broker, interface for coupled applications and interactive access).
- To demonstrate distributed simulation codes from different application areas (Biomolecular simulations, Weather prediction, Coupled CAE simulations, Structural analysis, Real-time data processing).
- To contribute to the international GRID development and to liase with the leading international GRID projects.
- To productise the EUROGRID software components. After project end the EUROGRID software will be available as supported product.

This HPC research Grid will be used as a testbed for the development of distributed applications, and as an arena for co-operative work among EUROGRID partners. The ultimate objective is the integration of full production systems in the Grid, and a close collaboration among different sites in solving major scientific challenges, using computational resources distributed on a European scale.

EUROGRID will concentrate on the development of general distributed. The distributed applications that will most likely have the biggest added value for scientific research are those concerned with multi-physics simulations or relying on complex computational chains, based on the coupling of different software modules. These distributed applications demand the coherent and complementary use of different computational resources, and map very naturally to a heterogeneous supercomputing environment.

Sequential coupling of software modules can be achieved from the start, through the present functionality of UNICORE. More ambitious applications involving real-time interaction of software modules will be developed using portable and well established standards like MPI2, CORBA, etc…This work will also address some issues related to the quality of services of the HPC research Grid, like Internet bandwidth, administration and operation issues, and security issues.

3.3.2 Application/Projects

3.3.2.1 Bio-GRID

The Bio-GRID is developing an access portal for biomolecular modeling resources. Bio-GRID is also developing interfaces to enable chemists and biologists to be able to submit work to HPC facilities. This task focuses on development of various interfaces to biomolecular applications and databases. The main task of Bio-GRID is to integrate selected applications with UNICORE infrastructure and provide easy tools for non-experts in high performance computing. The toolsets and user interfaces for both simulations and visualization of biomolecules will be developed.

As final step the computation portal will be established and tested with various biomolecular applications at ICM and a number of biomolecular projects and users. Well-established and widely used packages from different areas of biomolecular research, eg. molecular dynamics or quantum chemistry, are used (details are given in Chapter 4). The scalar, vector and parallel codes are considered. The user is also provided with interfaces to the various databases, usually accessible by the Internet connections and spreaded over the world. This allows to simplify access to databases and to integrate them with simulations.

3.3.2.2 Meteo GRID

A GUI will be developed and integrated in the EUROGRID environment. This GUI will allow an easy control of the LM run, e.g. regarding the model domain (anywhere on the globe), mesh size (usually between 25 and 2 km), forecast length (up to 48 hours), initial date and time, and the output of result files.

The production of a LM forecast for a specified region consists of the following steps:

1. A topographical data set for the specified region (LM domain) has to be derived from high resolution (1 km x 1 km) global data sets available at the DWD. The parameters which have to be computed as averages over model Grid boxes (e. g. 7 km x 7 km) include: Terrain height, land fraction, soil type, roughness length, plant cover, leaf area index and root depth. This data set will be computed only once for a given region and stored for later use.

2. Those data from the global model GME which cover the LM domain are extracted from the ORACLE data base of the DWD for the specified initial date and time. These data serve as initial data of the LM. Forecast fields of the GME at hourly steps will be retrieved, too. They will be used as lateral boundary conditions of the LM. The resolution of the GME is currently 60 km with 31 layers; in 2002, the mesh size will be reduced to 30 km with 35 layers.

3. The GME data have to be transferred from the DWD the HPC centre running the LM; the amount of data will be up to 100 MByte (up to 400 MByte after 2002), and the transfer should be completed in 15 to 30 minutes.

4. The interface program GME2LM that interpolates the GME data to the LM Grid has to run on the HPC in question. It creates the initial and lateral boundary data on the LM Grid. These data will be stored on disk at the HPC; the required space is in the order of 12 to 20 GByte. Later versions of Meteo-Grid will allow for site-specific observational input data as well.

5. As soon as the initial data plus two lateral boundary data sets are ready, the LM will run on the HPC and create result files at (usually hourly intervals, i. e. every 60 to 75 sec wallclock time). The typical forecast range of detailed regional numerical weather prediction runs is 48 hours.

6. These result files consist of GRIB code data (Gridded Binary data, a standard 8-bit byte code developed by the WMO, the World Meteorological Organization, Geneva). The files have to be transmitted back to the system of the user via high-speed links because the usual amount is between 2 and 4 GByte (for a 48-h forecast). The data are split into 49 files (0h, 1h, 2h, ..., 47h, 48h), each containing one forecast step; each file has a size between 40 to 80 MByte. The GRIB code data can be easily visualized by standard public domain graphic packages like GRADS or VIS5D on workstations or PCs.

Steps 2. to 6. have to be performed in parallel for real-time applications. Thus distributed computing via the Internet, controlled by the EUROGRID software, has to be realized. The whole sequence of jobs has to be initialized via the Web-based GUI and the user should be informed about the progress of each component and receive possible error messages in case of hard- or software failures.

The implementation of the relocatable weather prediction code has to be accompanied by a careful evaluation of the meteorological validity of the LM in any region of the globe for a wide range of mesh sizes between 25 and 2 km. The standard procedure of verification of model results consists of a comparison of the LM forecast to all observations available in the region. For this purpose, DWD has developed a workstation-based interactive Meteorological Application and Presentation system (MAP) that allows overlying for any region on the globe the available observations (including satellite images) and the LM forecast fields. MAP requires a medium-sized workstation for storing the huge amount of forecast and observational data as well as a rapid display of these data.

The relocatable weather prediction model LM and its preprocessor GME2LM will be installed at the HPC centres of CNRS-IDRIS and UIB as well as on the HPC system of the DWD. The LM is coded in standard Fortran90 and uses MPI for message passing. Thus in principle the code is highly portable. But there is usually a number of installation dependent parameters like compiler options or library versions which have to be adapted for optimal performance under real-time conditions for the HPC system in question. For real-time applications of the LM, some mechanisms must be provided by the HPC and EUROGRID to guarantee the execution of the LM suite at high operational priority. This has to be demonstrated in a multi-user environment. The functionality of the LM-GUI for the definition and launching of LM forecasts will be evaluated by test users who have no experience in HPC. Their experience and suggestions will be the basis for further development of the LM-GUI. A typical 48-hour prediction run of the LM for any region of the globe should not take longer than two hours including the preparation of the topographical data file. During the whole model suite, the user should be informed about the current status of the LM run and the expected time for finishing the task.

3.3.2.3 CAE GRID

This project focuses on industrial CAE applications. It aims at providing services to HPC customers that require huge computing power to solve their engineering problems. The major partners in this work-package are Debis SystemHaus and EADS Corporate Research centre. They will start by using the present functionality of UNICORE and will investigate two important aspect of industrial use of CAE, namely, code coupling and ASP-type services.

Code coupling or multi-physics optimization is a key point to improve system design leading by reducing the prototyping and testing costs. General support for the coupling of applications will be integrated in the EUROGRID system.

The ASP-type services will aim at designing application specific user interfaces for job submission so that the complexity of using HPC systems will be hidden to the application engineers. At the same time, this work-package will also address some issues related to the accurate prediction of cost estimates for CAE simulations enabling the user to choose for continuing or aborting a calculation.

4 GRID MOLECULAR SIMULATIONS

4.1 Introduction

Molecular science is one area where its progress closely related to the current technology of computers. Particularly, molecular simulations are important in designing new materials, pharmaceuticals, and generally, in Biological Physics and Chemistry. Molecular Computational Sciences deal with electronic structure calculations and Molecular Dynamics. The latter may be divided to quantum and classical dynamics depending on the type of equations of motion that are solved. Monte Carlo methods, both classical and quantum, together with stochastic dynamics, which involve the solution of Langevin equation, are also used in molecular simulations. A brief introduction into Quantum Chemistry concepts and main techniques was given in the case study of the second ENACTS report “High Performance Computing Technology Roadmap’’. We believe that ENACTS reports are not the place to develop further the concepts and techniques of Molecular Computational Sciences and the reader should consult the proper literature [1]. However, it is useful to keep in mind that electronic structure calculations are memory and storage bound, whereas classical dynamics are mainly CPU bound problems. Quantum Dynamics are both memory and CPU bound.

Molecular Computational Sciences face two challenges in twenty first century. The first has to do with the size of the systems; the study of molecular systems from two atoms to nanostructures and macroscopic states. The second challenge is related to time and the dynamics of the system; to cover phenomena of femtosecond durations to macro scales of seconds and minutes, time scales where some biological actions occur.

Physical properties of materials are the result of processes operating at various space and time scales, ranging respectively from nanometers to meters and from femtoseconds to seconds. At thermodynamic equilibrium, values for quantities of interest (free energy and its derivatives) are computed analytically or numerically, as space and/or time averages of the appropriate microscopic observable within the framework of statistical thermodynamics.

However, the processing and applications of materials occur often under strong gradients of pressure and temperature, i.e. very far from equilibrium. As a consequence, material properties become "history" dependent and highly non-linear. It is worth mentioning that the non-linearity is often due to the collective behavior of structural equilibrium (point defects) and metastable (dislocations, grain boundaries) defects.

Numerical simulation provides a means for studying the properties of materials under such non-equilibrium conditions. Starting from physically founded cohesion models or from an ab-initio description of atomic bonding, atomic scale simulation techniques (Molecular Dynamics and Monte Carlo) give insight to the behavior of the materials studied. This can help improving the quality of materials for technological applications and possibly to the development of new materials and/or fabrication processes. Unfortunately, the scales of most properties relevant for applications are fundamentally different from these of atomistic simulations. The challenge therefore for numerical simulations consists in bridging the gaps of space and time scales that separate atomic and macroscopic material properties.

Grid (distributing) computing could be the solution in bridging the gap in time and space. As a matter of fact, computational quantum chemistry was among the first disciplines to try to exploit the idea of networking computers. In early nineties Hartree-Fock calculations were distributed across continents to evaluate the performance in such remotely distributed cluster of computers. A review of the link of quantum chemistry to metacomputing was published by Hans Peter Luthi [2] in the year 2000. It is interesting that the early experiments proved that, i) on a network of computers a computation can be performed faster than on any single machine, and ii) a computation can be accelerated further by introducing heterogeneity in the network. However, in the following years there was no significant progress in this subject and the conclusion of this review was that whereas computational quantum chemistry was among the pioneers in metacomputing, other areas of HPC have taken the lead. As we shall see in the next sections, this pessimistic conclusion has not changed too much two years later.

4.2 What has been done

4.2.1 Quantum Chemistry

There are efficient parallelized codes for many body perturbation theory and Density Functional Theory, but not for other popular methods that estimate the electron correlation energy [2]. There is also progress in collaborative computational Grids for sharing a monitor to discuss and act on three-dimensional objects [3]. It is also worth mentioning an XML application to define a chemistry markup language, CML, for drawing and manipulating the three dimensional structures of molecules (http://www.xml-cml.org/)

4.2.2 Atom-Diatom collisions

It turns out that one of the first applications of Molecular Dynamics in distributing supercomputing was a computation in quantum chemical reaction dynamics by Mark Wu and Aron Kuppermann from Caltech [4]. The project involved the calculation of reaction cross sections with the 3D-REACT code running on the CASA gigabit network. This was an appropriate application for a heterogeneous computer, since parts of the program were suitable for vector machines, whereas other parts on massively parallel machines such as the Intel Delta and Paragon.

4.2.3 BioGrid

Another type of applications is via portals. Standard software installed in specific machines are available in the EuroGrid testbed. An example is the BioGrid run under UNICORE. The user can employ several packages from different areas of biomolecular research, eg. molecular dynamics or quantum chemistry. The codes are written for scalar, vector and parallel machines. The user is provided with interfaces to the various databases, usually accessible by the internet connections and spread over the world. This allows to simplify access to databases and to integrate them with simulations.

Software avaliable in BioGrid is:

- *Quantum Chemistry:*
  Gaussian98, Gaussian94, Gamess, Turbomole, CPMD, ADF, Molcas, Molpro
- *Molecular Dynamics:*
  CPMD, Amber, Charmm, Gromos96

4.2.4 Charmm

Molecular applications are also found in Centurion testbed. The user of this computational Grid can use Chemistry at Harvard Molecular Mechanics (CHARMM, http://yuri.harvard.edu/). A program for macromolecular dynamics and mechanics, designed to investigate the structure and dynamics of large molecules. The software in this particular application was developed by Martin Karplus at Harvard and Charles Brooks at the Scripps Research Institute.

4.2.5 Folding@Home

In order to use a worldwide distributed computing environment of thousands or even millions of heterogeneous processors such as the Grid, communications among these processors should be minimum. There are not many molecular simulations using the Grid environment such as to allow us to point out the strategy one should adopt in writing codes for molecular applications. A practical rule is to allow each processor to work independently, even though calculations are repeated, and only if something important happens to one of them, then they communicate. A. F. Voter [5] has shown that a linear speedup can be obtained in calculating the first transition rate constants by integrating replica of the system independently on M processors.

A successful application of this type of molecular calculations is Folding@Home (http://folding.stanford.edu/science.html) developed by V. S. Pande [6] and his collaborators at the department of Chemistry of Stanford University. Folding@Home is a distributed computing project that studies protein folding, misfolding, aggregation, and related diseases. They use novel computational methods and large scale distributed computing, to simulate timescales thousands to millions of times longer than previously achieved. This has allowed to simulate folding for the first time, and to now direct this approach to examine folding related disease.

A set of parallel replicas of a single simulation can be statistically coupled to closely approximate long trajectories. This produces nearly linear speedup over a single simulation. The investigators with the Folding@Home have simulated the folding mechanism and accurately predicted the folding rate of several fast folding proteins and polymers, including a non-biological helix, polypeptide alpha helices, a beta hairpin, and a three helix bundle protein from a villin headpiece [7].

4.2.6 DMMVLBCN

Other related literature with respect to distributed Molecular Modeling over Very-Low-Bandwidth computer networks can be found in the article of W. Ware [8].

4.3 What can be done

Future use of computational Grids in molecular sciences for simulating very long time and large space will be based in two important remarks:

As Ian Foster remarks [9], the average period during which speed or capacity doubles or more or less equivalently, halves in price, for networks, storage, and computing power are 9, 12, and 18 months, respectively. Thus, it is expected that distributing computing which is now prohibitive because of communication cost will become more attractive as the networking speed improves. Theoretically, communication can become unlimited and free, and then the use of geographically remote computers and other resources will be as practical as using the local resources. The improvement in networking will affect simulations in macroscopic scale.

The second observation is related to bridging the gap in time. In very long time simulations we are searching for rare events, for example in the protein folding problem we are sampling for very long times a particular minimum and rarely the trajectory jumps over a transition state to a new minimum in the potential energy landscape. This can be exploited by doing most of the work with local resources and rarely we communicate with the outside world to pass new information, as in Foldinf@Home method.

In this section we argue, that domain decomposition methods associated with Variable Order Finite Difference to compute the derivatives, which appear in the relevant equations of Molecular Sciences, may offer a paradigm for Grid computing which is in accord to Foster's remarks.

Most of the parallelized codes are based on domain decomposition methods which require continues exchange of information between neighboring processors. In a series of articles [10] we have shown how finite difference methods can be applied to solve the time dependent Schroedinger equation. To achieve high accuracy one needs to evaluate the second spatial derivatives at high order. However, one can employ low order finite difference methods if he/she uses a large number of Grid points. Low order derivatives means less bytes to exchange between neighboring processors, and thus less communication cost. The figure below shows schematically how to distribute the Grid points among the processors in one dimension. In this figure the red circles show the data that must be exchanged with the neighbor processors, K-1 and K+1. The higher the order of Finite Difference approximation the more data must be exchanged. However, we repeat that low order Finite Difference with a dense mesh achieve the same accuracy. Thus, by using Finite Difference methods we obtain an efficient parallelization scheme compared to other schemes that require the complete transfer of the matrix. However, it is understandable that implementation of parallelized codes in a Grid environment is not enough to achieve our goal, high computational performance. The frequency of exchanging data should be as small as possible something that is indeed the case for a certain kind of problems. Such an example is molecules in intense Laser fields [11]. For the full six dimensional problem hundreds of Gbytes of memory are needed. In these systems the electron is extended in space and a large mesh is required. Thus, by appropriately distributing the computational domains we do most of the work with local computers and limited information is exchanged with remotely distributed computers.

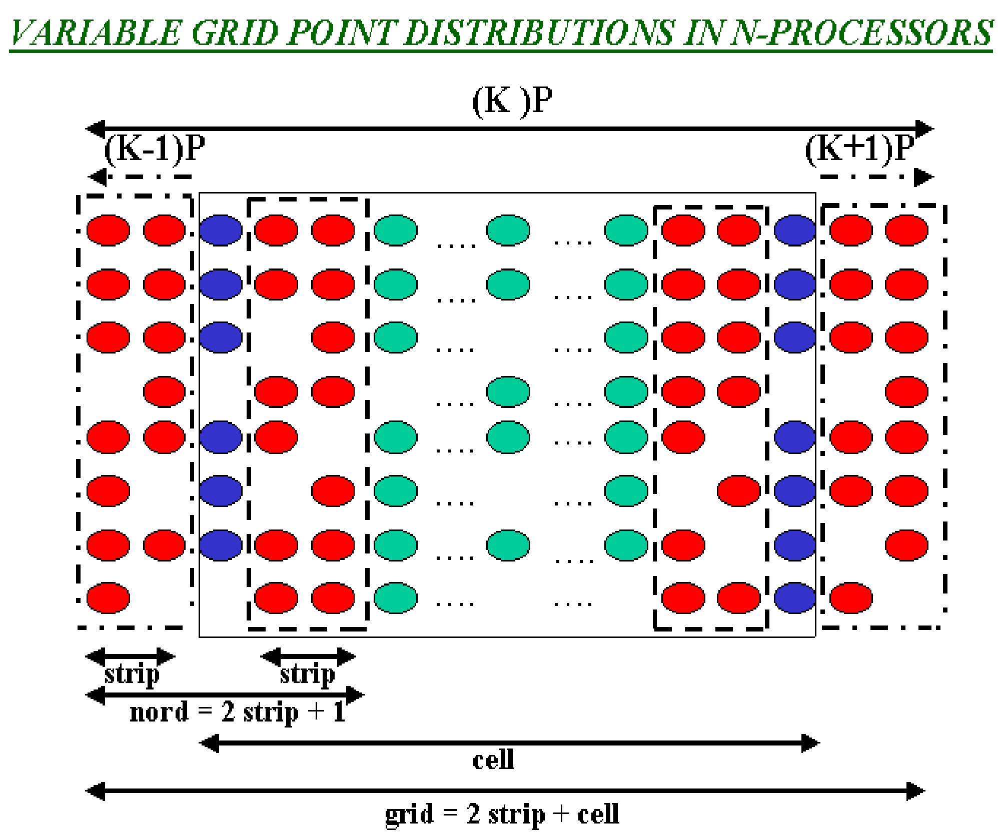

Figure 1: Variable Grid Point Distribution in n-processors.

In the above picture a cell denotes the data that are kept in the Kth processor. Strips are the data that are exchanged to and from the neighboring, K-1 and K+1, processors.

4.4 Implementation

In order to test the above ideas as well as to get personal experience in installing and running a computational Grid we created a small homogeneous testbed which consists of a cluster of PCs with 16 processors and several other computers spread among FORTH, University of Crete and Trinity College at Dublin. All of them are using Linux as an operating system for the PCs and Globus 2 for deploying the Grid. Because of the administration and security problems CSCS could not join, thus revealing one of the substantial difficulties in developing Grid computing.

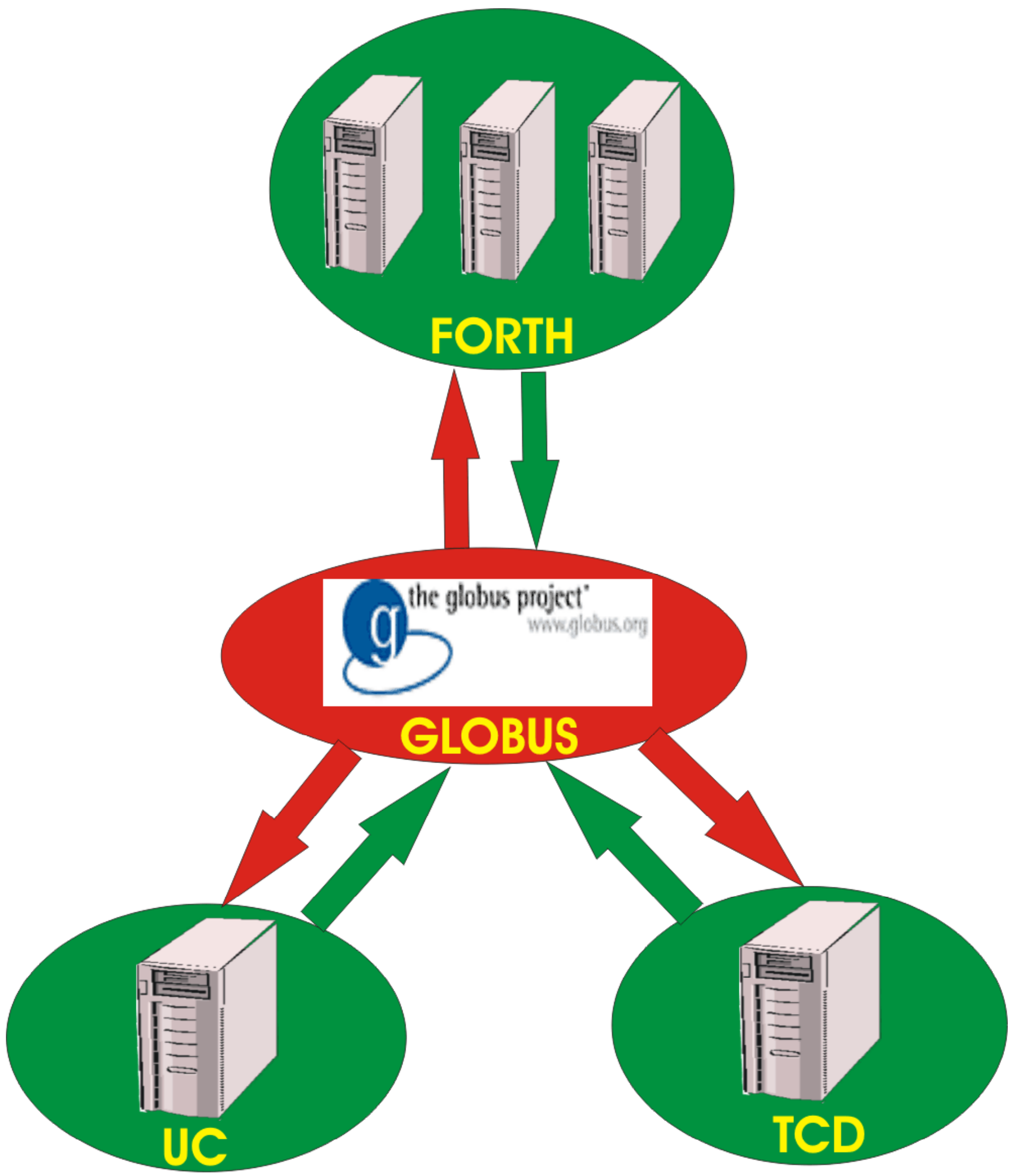

Figure 2: The first stage of ENACTS Grid.

There is a special version of Message Passing Interface program for Globus, MPI-G2 (http://www-unix.mcs.anl.gov/mpi/mpich/docs/mpichman-globus2/node14.htm) which allows our code on the high order Finite Difference method to run in a distributed computer system. We have not planned to carry out a complete investigation of the efficiency of these computations in the frame of ENACTS project. That would require the optimization of our codes. Instead, our modest targets were to get first, experience in installing Globus, and second, to learn how to use it in our everyday work with computers. Overall, we got a positive opinion. The installation of Globus is straightforward for an experienced programmer. It turns out, that with the latest versions of Globus things are simplified even more, but some efforts should be put to improve the documentation. Although, most functions of Globus can be performed with the old methods, e.g. ftp for file transfer; the sign up procedure of Globus to use the Grid for a certain period of time is very convenient. We would recommend using Globus as a working environment even if there are no distributed jobs to run. There are just a few commands to learn and you can start immediate your work. Of course, we would advice one to write scripts in RSL (Resource Specification Language) when he has to distribute a computation and transfer input/output files to/from remote computers to his local master computer. Our specific application is not the proper example for improving timing. The calculations slowed down considerably when the Irish site was also included in the Grid. Nevertheless, we met no difficulties in running jobs and collecting the results.

4.5 References

[1] “Computational Biochemistry and Biophysics”, O.M. Becker, A.D. MacKerell, Jr., B. Roux and M. Watanabe, Marcel Dekker, Inc., New York, 2001.

[2] “Metacomputing, an emerging technology?”, Hans Peter Luthi, Computer Physics Communications , 128, 326, 2000.

[3] P. F. Flukiger, S. Portmann, H. P. Luthi, Lecture Notes in Compt. Sci. 1615, 11, 1999.

[4] Mark Wu and Aron Kuppermann, in “The Grid: the blueprint for a new computing infrastructure”, Ian Foster and Carl Kesselman, Morhan Kaufmann Publishers, Inc., San Francisco, 1999, pp. 62-65.

[5] “Parallel replica method for dynamics of infrequent events”, A. F. Voter, Phys. Rev. B, 57, R13985, 1998.

[6] “Screen savers of the world unite”, M. Shirts and V. S. Pande, Science, 290, 1903, 2000.
“Mathematical analysis of coupled parallel simulations”, M. Shirts and V. S. Pande, Phy. Rev. Lett., 86, 4983, 2001.

[7] “Atomistic protein folding simulations on the hundreds of microsecond timescale using worldwide distributing computing”, V. S. Pande, I. Baker, J. Chapman, S. Elmer, S. Khaliq, S. Larson, Young Min Rhee, M. Shirts, Ch. Snow, E. J. Sorin, and B. Zagrovic, to be published.
“ -Hairpin folding simulations in atomistic detail using an implicit solvent

model", B. Zagrovic, E. J. Sorin and V. S. Pande, J. Mol. Biology, 313, 151, 2001.

[8] "Distributed Molecular Modeling over very-low-bandwidth computer networks", W. Ware, http://world.std.com/~wware/

[9] "The Grid: A new infrastructure for 21st century science", Ian Foster, Physics Today,

[10] "High Order Finite Difference Algorithms for Solving the Schroedinger Equation in Molecular Dynamics. I", Raul Guantes and Stavros C. Farantos, J. Chem. Phys., 111, 10827, 1999.
"High Order Finite Difference Algorithms for Solving the Schroedinger Equation in Molecular Dynamics. II. Periodic variables", Raul Guantes and Stavros C. Farantos, J. Chem. Phys., 113, 10429, 2000.

[11] "Efficient grid treatment of the ionization dynamics of laser-driven H_2^+",

Daniel Dundas, Physical Review A, 65, 023408, 2002.

5 GLOBUS PRACTICES

5.1 Globus Components

The Globus Project, an initiative of University of Southern California's Information Sciences Institute and Argonne National Laboratory, provides software tools and defines protocols to enable the development of computational grids and grid-based applications. These tools are collectively referred to as the Globus Toolkit. It is an open architecture, open source project, under active development and continual evolution.

The Globus Toolkit is a set of tools providing fundamental grid services:

- Resource management
- Communication
- Security
- Information
- Fault tolerance
- Data management.

These are modular and can operate independently of each other, enabling the administrators and the developers of grid-based applications to choose the tools they will install and use. Parts of the Globus Toolkit can coexist with or replace mechanisms provided in the "traditional" computing environment, such as ftp and rsh.

Below, we will present the tools, focusing on those relevant to installation and usage of Globus. It should be noted that, as the Globus Toolkit is in development, new services are introduced or implementation of old ones are changing. We attempt to give an overview of the core tools at version 2; admittedly their status may change dramatically in newer versions.

5.1.1 Resource Management Service

The Globus Toolkit implements this service by providing the *Globus Resource Allocation Manager* (GRAM). This tool performs resource location and allocation as well as process management (creation, monitoring, termination). It provides a uniform interface to the local resource management tools and schedulers (e.g. LSF, NQE, LoadLeveler, Condor) that may be deployed in a site. The user issues generic resource requests using the *Globus Resource Specification Language* (RSL) that are translated by GRAM into commands targeted to the local system.

Resource management is provided mainly by two programs: the gatekeeper and the jobmanager.

The gatekeeper acts as the interface towards the user. It runs on each machine where Globus is installed, normally as a daemon. It authenticates the incoming request using the

Grid Security Infrastructure (GSI), and then matches the user to a local user id; it is thus required to have an account on all machines whose resources are requested, but not necessarily different for each remote user.

The inbound request specifies a particular local service to launch, usually a jobmanager process running under the remote user id. On successful authentication, the jobmanager parses the request and handles the allocation of the requested resources, the execution of the job and the deallocation of the resources used, on termination of the job. It also communicates with the local user by issuing and returning a contact string; via this string, the user can inquire the status of the submitted job, cancel a pending job or terminate a running one.

Globus provides a co-allocation service via the *Dynamically Updated Request Online Co-allocator* (DUROC). It enables the user to submit a request for resources located on multiple machines and handles their synchronization. That is, multiple resources are allocated simultaneously for a single job and used in parallel. Pending requests can be edited dynamically.

5.1.2 Resource Specification Language (RSL)

RSL is an extensible language through which GRAM components communicate with other components in the system; the user can express resource requests in it. The requests can be explicit, targeting an executable for example, to a certain number of nodes, in a specific machine, in a specific queue. The user can also express abstract requests towards a broker for the location and allocation of certain resources. Automatic brokers are not available yet, but they can and should be developed to accept an abstract request, locate sites offering the required resources and construct explicit RSL commands.

For example:

```
(* this is a comment *)
&(executable="/home/jdoe/myprogram")
(stdin=/home/jdoe/input)
(stdout=/home/jdoe/output)
(jobType=mpi)
(count=2)
```

this code submits 2 processes of the executable /home/jdoe/myprogram, connecting the files /home/jdoe/input, /home/jdoe/output as input and output. It is assumed that all files exist at the remote site, the program specified is actually an executable for the targeted architecture, and the jobmanager there, has been configured to spawn mpi processes.

This RSL script is submitted to a specific host via the globusrun command (see the section on “Usage”).

For a full specification of RSL see www-fp.globus.org/gram/rsl.html; for the RSL parameters targeted to GRAM see www-fp.globus.org/gram/gram_rsl_parameters.html.

5.1.3 Communication service

The Nexus and, more recently, the globus_io libraries are the communication components of the Globus Toolkit. According to the documentation:

"Nexus is a portable library providing the multithreaded communication facilities required to implement advanced languages, libraries, and applications in heterogeneous parallel and distributed computing environments. Its interface provides a global memory model via interprocessor references; asynchronous events; and is thread safe when used in conjunction with the Globus thread library. Its implementation supports multiple communication protocols and resource characterization mechanisms that allow automatic selection of optimal protocols."

"The globus_io library is motivated by the desire to provide a uniform I/O interface to stream and datagram style communications. The goals in doing this are to

- Provide a robust way to describe, apply, and query connection properties. These include the standard socket options (socket buffer sizes, etc), as well as additional attributes. These include security attributes and, eventually, Quality of Service attributes.
- Provide a service to support non-blocking I/O and handle asynchronous file and network events.
- Provide a simple and portable way to implement communication protocols. Globus components such as GASS (Global Access to Secondary Storage, see below) and GRAM can use this to redefine their control message protocol in terms of TCP messages, instead of Nexus Remote Service Requests."

Nexus and globus_io are not designed as user protocols; they rather provide the building blocks for such. As they are targeted to developers and not to users or administrators, a detailed presentation of them is out of the scope of this report.

5.1.4 Security Service

The Globus Toolkit uses the *Grid Security Infrastructure* (GSI) to authenticate and encrypt via the *Secure Sockets Layer* (SSL) protocol all communication and resource use. Features of this system include

- A "single sign-on" for each user of the grid: the authentication is handled by a proxy set up by each user; the proxy requests user credentials at creation, and, during its user-specified lifetime, handles automatically all authentication requests. Apart from the convenience it brings, a proxy enables the user to submit unattended batch jobs.
- The circumvention of a centrally managed security system; the GSI provides a common interface of the diverse local security solutions, make feasible the integration of sites with varying degree of security provisions.

Globus security is based on the ITU X.509v3 authentication protocol. Two alphanumeric strings, the public and the private keys, are generated for each user, optionally protected by a password. A certificate request based on the public key and the identity of user is sent to a Certificate Authority (CA) in order to validate it. A CA can be an independent third party or organization running certificate-issuing software. The validation process creates a certificate encompassing the public key sent, the user identity (possibly checked), the distinguished name of the CA, the date of issue, the expiration date, a serial number etc., encrypted with the private key of the CA. Setting up a proxy by a user means that a proxy certificate is generated and is signed by the user certificate. The proxy certificate is sent to the gatekeeper whenever an authentication is required for the user. Assuming that the gatekeeper is configured to trust the signing CA, it checks whether the certificate is valid; if it is, it maps the remote user to a local one.

Proxies provide a convenient alternative to constantly entering passwords, but are also less secure than the user's normal security credential.

Associated with the gatekeeper there is a certificate, signed also by a CA; it is sent back to the client whenever authentication is requested (for example, prior to retrieving a staged file or delivering stdout/stderr); this enables the proxy to verify it communicates with the intended system and not an imposter.

5.1.5 Information Service

The *Metacomputing Directory Service* (MDS) (encountered in its User Guide as the Monitoring and Discovery Service) is the part of the Globus Toolkit that implements this service.

An absolutely essential part of a Grid infrastructure is a mechanism to locate, classify and monitor the available resources (compute nodes, storage space, network, instruments, etc.); a complex task given their diversity, heterogeneity and the dynamic behavior of them. It is vital for an application to have information on the resources constantly, on which the initial targeting and the subsequent adaptation will be based.

Information services in the Globus Toolkit are designed to provide:

- Access to static and dynamic information regarding system components
- A basis for configuration and adaption in heterogeneous, dynamic environments
- Uniform, flexible access to information
- Scalable, efficient access to dynamic data
- Access to multiple information sources
- Decentralized maintenance.

The MDS uses the *Lightweight Directory Access Protocol* (LDAP) as a uniform interface to posting and retrieving such information: not only can it query the system state from a rich variety of system components, it also enables the construction of a uniform namespace for resource information across a grid spanning many organizations with different management and security concerns.

Two basic components are provided by MDS: the *Grid Resource Information Service* (GRIS) and the *Grid Index Information Service* (GIIS).

A GRIS, running on each site provides information on the resources available on this site: the platform type, the architecture, the operating system, the type, characteristics and number of CPUs, type and size of memory and storage space, etc. They can be processed and filtered based on constraints set out by the user. GIIS, on the other hand, works as "yellow pages": it combines arbitrary GRIS services providing a mechanism to locate specific resources in the grid as a whole. A GRIS registers itself with a GIIS, making available to the aggregate directory services its information.

5.1.6 Fault tolerance Service

Earlier versions of the Globus Toolkit included the *Heartbeat Monitor* (HBM) that allows system administrators or ordinary users to detect failure of processes.

The HBM performs periodic inquiry, checking the status of a registered process. The registration consists of a specific function call made by the application, in order to notify the local HBM daemon.

In the current version of Globus, there exist no documentation on how to enable or use this service.

Nexus communication library also provides fault detection mechanism.

5.1.7 Data management

Globus facilitates the staging of executables and input files through *Global Access to Secondary Storage* (GASS). The user can specify in the request the files that should be transfered to the remote site; the jobmanager will retrieve them prior to execution of the program. If the request was submitted interactively, the jobmanager will send the output files to the originating site automatically; in a batch submission, the user should retrieve them manually.

All files are accessed via a *Universal Resource Locator* (URL). Four types of remote file access are allowed [1]:

- Shared read-only access to a file
- Shared write access where the last write dominates
- Shared append-only access to a file
- Unshared unrestricted read/write access

The latest release of the Globus Toolkit includes many new components (GridFTP, Replica Catalog, Replica Management) intended for use in building Data Grid environments.

5.2 Installation and Configuration

In this section, we will give instructions on setting up a computational grid using the generic Globus Toolkit at version 2.2. By no means should they replace the procedure given in www.globus.org/gt2.2/install.html, which is the primary source of information. We will follow it closely, clarifying and, in cases, augmenting it. Note that if you want to participate in a already set-up grid, you, should, in general, follow the instructions given by its administrators; these include installation of probably modified versions of Globus software, certificates from a specific CA, etc.

Note that the Globus Toolkit has been tested, according to the Globus homepage, on UNIX machines; a Windows XP/2000 port is under development. Recently, alpha versions of Linux packages in .rpm and .deb formats were made available for testing by third parties. We have no experience with them.

5.2.1 Set-up of Own Certificate Authority

The Globus Toolkit is configured to trust certificates issued by its own CA (Globus CA) providing no instructions on using alternative CAs or setting up local ones. In our view, this policy seriously hampers the deployment of test or local grids as it is time-consuming and imposes unnecessarily strict security. Recently, the Globus Project released the Globus Simple CA Package (www.globus.org/security/simple-ca.html), a tool to provide a convenient way of setting up a certificate authority. It requires that Globus Toolkit is already installed and therefore, it is still necessary to acquire few credentials from Globus CA. We feel that it should be independent of Globus. Our approach uses the publicly available openSSL package at version 0.9.6c. We managed to set up our own CA using the following (UNIX) commands:

```
openssl req -x509 -days 1000 -new -keyout CA.key -out
CA.cert
```

This command creates a new CA 1024-bit RSA private key (CA.key) and certificate (CA.cert) with lifetime 1000 days.

It requests a pass phrase to protect the key, and the CA's Distinguished Name; it comprises the country, the city, the organization, the name of the CA administrator, etc.

```
openssl x509 -hash -noout -subject -in CA.cert
```

The command above provides:

1. The hash used in forming an index to allow certificates in a directory to be looked up by subject name. You should copy CA.cert to "hash".0, that is, assuming the hash is "36a89c1c" you issue the command:

   cp CA.cert 36a89c1c.0

   This file will be used during installation of Globus.

2. The DN of the Certification Authority. It should be sent to any grid administrators wishing to trust your CA along with the certificate above.

These commands set up our own CA. In order to sign a certificate request received by a user (e.g. in the file req.pem), with lifetime, say, 100 days, the administrator should issue the command:

```
openssl x509 -req -CAkey CA.key -CA CA.cert -CAcreateserial
-days 100 -in req.pem \
 -out cert.pem
```

The file cert.pem contains the certificate and should be sent back to the user. The administrator should verify the identity of the user, prior to validation.

5.2.2 Globus Toolkit installation

You should have root permission to install Globus; it is recommended by the developers that the build commands should be issued by an ordinary user, say, with username "globus", and the placement and initialization of the package should be done by root. It is probably a good precaution security-wise, but it is somewhat restrictive and unnecessary for a test grid. Our installation was performed totally by root.

You should also have a fairly updated OS on the machine you plan to install Globus; for specific requirements consult the globus homepage. In certain Linux systems (Redhat 7.0, Suse 7.1 and probably other distributions), for some reason the installation process can't recognize the right threads package. In that case, update your glibc-* packages using rpm-files from RedHat 7.1 and it will recognize it automatically.

The administrator should download from www.globus.org/gt2.2/download.html the client and server source packages as well as the Grid packaging toolkit (gpt). There exist binary ones for certain architectures, but problems (missing modules etc.) were reported for them in the past. Our installation was based on the source packages. The SDK packages are need only if other packages will be built on top of Globus (such as MPICH-G2). Note that the current version of globus-enabled MPICH (v1.2.4) is not compatible with Globus Toolkit at version 2.2 which is the most recent; you will have to use the previous version of the toolkit in order to install mpich-g2.

An overview of the procedure followed is given below. It is a summary, with clarifications, of the installation instructions which are available on the globus homepage. This information is definitely in evolution; the installation procedure changed dramatically from major version 1 to version 2 of the toolkit, and it will probably change in version 3.

1. Declare where to install Globus via the environment variable *GLOBUS_LOCATION*; define *GPT_LOCATION* to indicate where to install gpt (usually in the same directory).
2. Install gpt:

   gzip –dc gpt-2.2.2-src.tar.gz | tar xf -
   cd gpt-2.2.2 && ./build_gpt && cd ..

3. Install the necessary client and/or server packages according to the Globus installation instructions, that is, for each bundle run

${GPT_LOCATION}/sbin/globus-build -install-only <bundle> <flavors>

On successful installation run

. $GLOBUS_LOCATION/etc/globus-user-env.sh

$GPT_LOCATION/sbin/gpt-postinstall

4. Set up GSI by issuing the command

   $GLOBUS_LOCATION/setup/globus/setup-gsi

   This command offers to modify a file; accept it. You should then provide the base Distinguished Name for the user and gatekeeper's certificates. For example for the users belonging to IESL (the Organizational Unit), a part of FORTH (the Organization), located in Greece (the Country), their DN should include:

   C=gr,O=forth,OU=iesl (1)

   The same applies to the machines at this site. Therefore, the string above should be the reply at the prompts when 1, 2 are pressed. You can also include in this string other name-value pairs: L=(location), ST=(state), etc. You exit via (q).

5. Install credentials of CA. The grid administrator can configure Globus to trust additional CA's other than the default (Globus CA). For the CA created in §5.2.1, the CA certificate file "*36a89c1c.0*" should be saved in the directory */etc/grid-security/certificates*. The administrator should create in the same directory the file "*36a89c1c.signing_policy*" containing the following:

   access_id_CA X509 '<CA DN>'

   pos_rights globus CA:sign

   cond_subjects globus ' "<BASE DN>/" '*

   where <CA DN> should be replaced with the string (without any quotes) representing the CA Distinguished Name (see §5.2.1), and <BASE DN> should be the string in (1). If you chose different base names for host and users above then you should substitute ' "<BASE USER DN>/*" "<BASE HOST DN>/*:" '. This file defines which CA (<CA DN>) is trusted for signing which names ("<BASE DN>"). See for more on the format of these the file /etc/grid-security/certificates/42864e48.signing_policy, which is the signing policy of the default CA (Globus CA).

6. (Optionally) replace all instances of "anonymousbind yes" with "anonymousbind no"in the file ${GLOBUS_LOCATION}/etc/grid-info-slapd.conf. It prohibits anonymous MDS requests.

7. Create certificate requests for gatekeeper and (optionally) LDAP:

   *${GLOBUS_LOCATION}/bin/grid-cert-request –service host *

–host <HOSTNAME>

$GLOBUS_LOCATION/bin/grid-cert-request –service ldap \
–host <HOSTNAME>

<HOSTNAME> should be the Fully Qualified Domain Name (FQDN) of the site. Make sure that it corresponds to a valid DNS entry; the site should be accessible by this name.

Send /etc/grid-security/host/hostcert_request.pem and /etc/grid-security/ldap/ldapcert_request.pem to the CA for signing. The host certificate should be then saved at /etc/grid-security/hostcert.pem; this file must be readable only by root. The ldap certificate should be saved at etc/grid-security/ldap/ldapcert.pem.

8. Install the start-up scripts for the gatekeeper and gsiftp, modifying certain system files; define certain environment variables. Notice that at least the tcp ports 2119, 2811 should be accessible through a firewall. This procedure for a Linux system is as follows:

- Append to /etc/services the following lines:

```
gsigatekeeper   2119/tcp        # Globus Gatekeeper
gsiftp          2811/tcp        # GsiFTP
```

- If your Linux distribution is xinetd-based, write

```
service gsigatekeeper
        {
     socket_type  = stream
     protocol     = tcp
     wait         = no
     user         = root
     env          = LD_LIBRARY_PATH=${GLOBUS_LOCATION}/lib
     server       = ${GLOBUS_LOCATION}/sbin/globus-gatekeeper
     server_args  = -conf \
${GLOBUS_LOCATION}/etc/globus-gatekeeper.conf
     disable      = no
   }
```

in the file /etc/xinetd.d/gsigatekeeper, and save in /etc/xinetd.d/gsiftp the following:

```
service gsiftp
    {
     instances    = 1000
     socket_type  = stream
     wait         = no
     user         = root
     env          = LD_LIBRARY_PATH=${GLOBUS_LOCATION}/lib
     server       = ${GLOBUS_LOCATION}/sbin/in.ftpd
     server_args    = -l -a -G ${GLOBUS_LOCATION}
```

```
    log_on_success += DURATION USERID
    log_on_failure += USERID
    nice           = 10
    disable        = no
}
```

Otherwise, that is, if your distribution is inetd-based, put the following in /etc/inetd.conf:

```
gsigatekeeper stream tcp nowait root \
/usr/bin/env env LD_LIBRARY_PATH=${GLOBUS_LOCATION}/lib \
${GLOBUS_LOCATION}/sbin/globus-gatekeeper \
-conf ${GLOBUS_LOCATION}/etc/globus-gatekeeper.conf

gsiftp stream tcp nowait root \
/usr/bin/env env LD_LIBRARY_PATH=${GLOBUS_LOCATION}/lib \
${GLOBUS_LOCATION}/sbin/in.ftpd -l -a -G ${GLOBUS_LOCATION}
```

- Copy to the appropriate startup directories the file `${GLOBUS_LOCATION}/sbin/globus-mds`. Also, copy to the file/directory which is sourced on logon the files `${GLOBUS_LOCATION}/etc/globus-user-env.[c]sh`. They define certain environment variables.
- Start the globus-mds and inetd/xinetd services.

If everything went ok the computational grid is now set up. By default, the gatekeeper daemon and the gsiftp server are up, and the GRIS is registered to a GIIS running on the machine.

On a machine where at least the client packages are installed, a user must create a certificate request that will be subsequently signed by one of the trusted CA's:

The user should run grid-cert-request; this command will construct a private key and a certificate request getting the user name via the 'finger' command (or the ${GLOBUS_SH_FINGER} if it is defined) and using the user Base DN defined during setup. By default, the request is saved in ${HOME}/.globus/usercert_request.pem which should be sent to the CA. The certificate received from the CA should be saved in ${HOME}/.globus/usercert.pem. The output of

openssl x509 -noout -subject -in $HOME/.globus/usercert.pem

(the ID of the user) should be sent to the administrator of the grid the user wishes to gain access to.

On receipt of the identity of a remote user, the grid administrator should place it in /etc/grid-security/grid-mapfile mapping it to a local user (say 'jdoe') as follows:

```
"/C=gr/O=forth/OU=iesl/CN=John Doe"  jdoe
```

Multiple mappings to the same local user are allowed.

5.2.3 Usage

The basic commands a user needs are:

- grid-proxy-init.
  It constructs the proxy that responds to authentication requests.
- globusrun.
  This is the main program that parses RSL commands. For example, to run on <host> (-r), the commands in the file "commands.rsl" (-f) and direct the output to stdout (-s), use:

  globusrun -s -r <host> -f commands.rsl

- globus-job-run.
  It is the simple, easy to use, interface to globusrun. Based on its arguments, it constructs an RSL request passing it to globusrun. For example:

  globus-job-run <host> <executable>

  runs on <host> the (local on <host>) <executable>. It supports staging of files.
- globus-job-submit. It is the batch interface to globusrun. globus-job-submit <host> <executable> returns a contact string through which globus-job-status, globus-job-get-output, globus-job-clean can query the status of a job, collect the output or terminate it. Currently, it does not support staging of files.
- grid-info-host-search. It is the command to query a GRIS server. grid-info-host-search -p <port> <hostname> <LDAP filter>
- grid-info-search. It is the command to query a GIIS server.
- globus-url-copy. It is the basic command to copy files: globus-url-copy <source URL> <dest URL> The URL prefixes handled are: gsiftp://, file://, http://, https://.

For details on them or additional commands, consult the globus documentation.

5.2.4 References

[1] J. Bester, I. Foster, C. Kesselman, J. Tedesco, S. Tuecke. GASS: A Data Movement and Access Service for Wide Area Computing Systems. Sixth Workshop on I/O in Parallel and Distributed Systems, 1999.

6 UNICORE PRACTICES

6.1 Overview

The following paragraphs consider the installation and the configuration of a UNICORE Grid, for that reason it is assumed that the reader is already familiar with the concepts, the architecture and the components of this product.

The information reported in this section is related to the version 3.6.x of UNICORE.

There is a complete different approach between UNICORE and for example the Globus toolkit model. In particular, rather than defining basic services and protocols for enabling the building of global services, UNICORE provides a complete solution that allows seamless, secure, and intuitive submission, monitoring and controlling of jobs for different systems at multiple sites.

Therefore, in building such a Grid you can consider two contexts of deployment. The first is related to the server side components that a computing site has to install and configure in order to participate in the Grid. The second is connected to the client side application that let a user accesses and makes use of the resources available in the Grid.

The deployment contexts are considered separately in the next sections. For more information on general concepts and technical details you can refer to Part II Chapter 6 of the "Grid Service Requirement" document.

6.2 UNICORE installation: servers side components

The next picture shows an example of a multiple server side deployment where two Gateway components represent different organizations as UNICORE sites (Usite). Then those Gateways are responsible to interact with a set of Network Job Supervisor (NJS) processes. NJSs correspond to virtual site (Vsite) in the overall Grid environment. At the lowest level, the Target System Interface assures the communication with the underlying batch system running on the target machine.

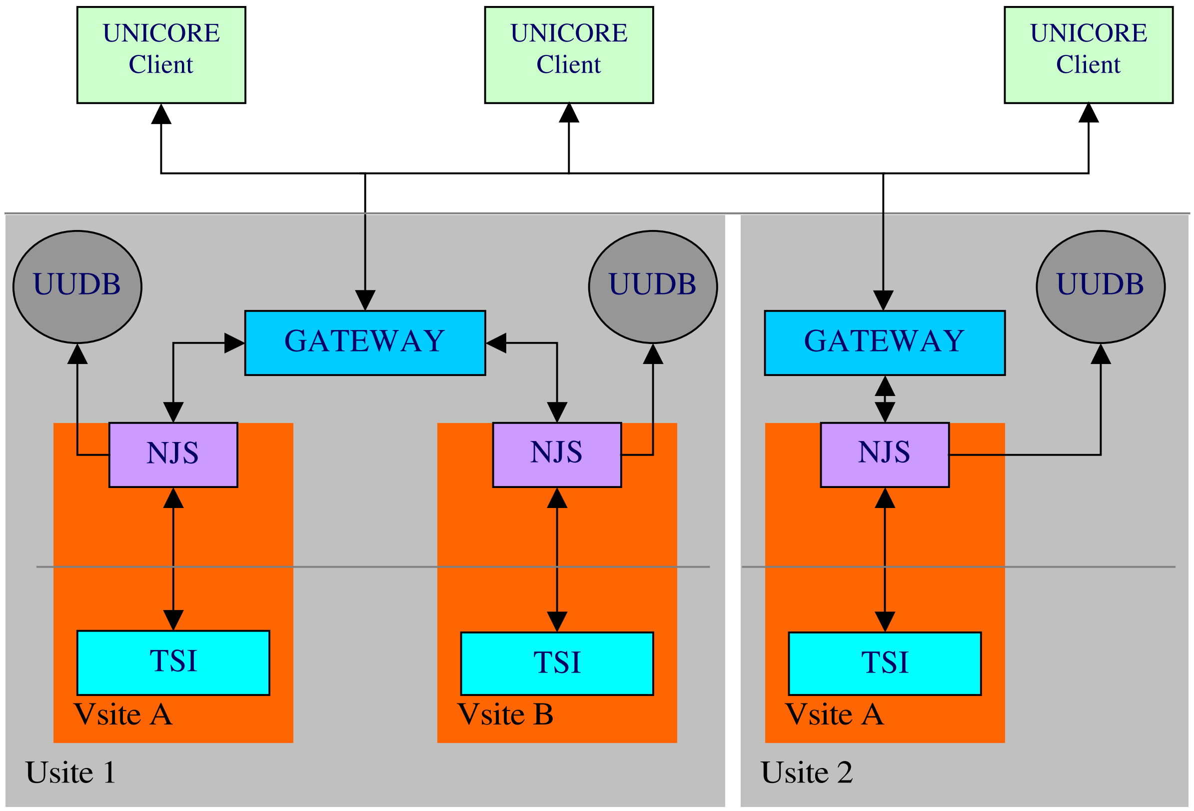

Figure 3: UNICORE Deployment Example

6.2.1 Gateway

6.2.1.1 Requirements

The Gateway is implemented as a Java application that has the following requirements: a Java 2 Runtime Environment installation, the availability of the Java Secure Socket Extension (JSSE) and the Abstract Job Object (AJO) libraries. For running this application needs a Public Key Infrastructure (PKI) certificate and its private key from an established Certificate Authority (CA) that determines the identity of the Gateway itself. Moreover a set of public certificates should be provided in order to define all the Certificate Authorities accepted by this Usite.

6.2.1.2 Objectives

In summary, the Gateway is responsible to filter all the connections from outside, to assure secure communications between the server side and the outside world, to provide information on the available resources represented by the set of Vsites and, finally, to manage the connection with the respective Vsites. Particularly, the Gateway works as a point of contact for all UNICORE connections addressed to the server side. The communication is done through the Secure Socket Layer (SSL) and only connections belonging to clients that present a valid certificate are accepted. A requirement for a valid certificate is that the issuing CA is included in the list of accepted CAs for the Gateway and the certificate is not included in the Certificate Revocation List (CRL), in case this optional checking is enabled in the Gateway's configuration.

6.2.1.3 Configuration

The Gateway application is configured through two files that are read at start up. The first one defines the general properties of the application such as the name, the port number, the certificate and the list of accepted CAs.

Optional parameters for consulting CRLs can be included too. Those parameters concern the activation of this service, the default validity period of a CRL and the extension to a CRL's validity period when the update of the CRL fails.

Finally, the application provides a logging mechanism that can be configured by specifying appropriate properties.

The second file defines the list of Vsite managed by the Gateway. Those Vsite are described by the following properties: the Vsite name, the machine address, the port number. Optional properties are: the type for the Vsite, the AJO library version and the way for transferring data from client to the Vsite. It is important to notice that in the current version of UNICORE a NJS process represents a Vsite, but in the future alternative implementation would also be possible.

6.2.2 Network Job Supervisor

6.2.2.1 Requirements

The Network Job Supervisor is used to represent a Vsite in the UNICORE Grid. Like the Gateway, it is implemented as a Java application with the same requirements: a Java 2 Runtime Environment installation, the availability of the Java Secure Socket Extension (JSSE) and the Abstract Job Object (AJO) libraries. A PKI certificate and its private key from a Certificate Authority are also needed in order to use SSL for communicating with the Gateway.

6.2.2.2 Objectives

The role of the NJS consist of elaborating the AJO coming from clients and other Vsites, forwarding sub jobs to the respective Gateways, supporting file transfer, presenting local resource information to the Gateway and supplying jobs status and job outputs.

In particular, the serialized AJO is analyzed by the NJS that performs the mapping of the user certificate to the id relative to the target system and the translation of the part of the AJO that has to be executed by this Vsite into a format suitable for the underlying system. This last process is indicated as incarnation and the Incarnation Data Base (IDB) defines the translation rules. Finally, a special repository indicated as UNICORE User Data Base (UUDB) is used to perform the id mapping for the final target system.

6.2.2.3 Configuration

The NJS uses the following resources during start up and processing: a configuration file for NJS properties, the IDB, the UUDB, a directory for the log information, a directory for storing the state and a set of scripts for starting, monitoring and administering the application.

The set of NJS properties includes the Vsite name, the path to the IDB, the path to UUDB, the port for listening to Gateway requests and the Gateway machine. Other properties concern the security settings that are required in order to establish SSL connection for consigning sub AJOs and to validate the certificates of remote Gateways.

The NJS can be administered through a set of scripts that are able to stop, pause and restart the application, or to list information on selected actions.

6.2.3 Incarnation Database

The execution of an action to a target system is defined as a two steps process where the NJS first convert the action terms into site and vendor specific commands that, in a second step, are passed for the execution to the Target System Interface (TSI). Moreover, the NJS needs also to execute common commands like creating a new Uspace, removing an existing Uspace, retrieving error and output streams or the status for submitted jobs. Since the NJS is designed as a generic application, it needs to be customized with site-specific information. This initialization process is achieved through the information contained in the Incarnation Database (IDB).

The IDB is organized as text document with different sections. Each section is recognized by a keyword name, ended by the END keyword and has its own set of valid keywords and subsections.

Main sections are the general, the execution TSI and the storage TSI sections. While other sections describe incarnation for execution tasks, import and exporting files, clean up working spaces and so on.

The main section of the IDB defines properties that apply to the whole Vsite and to all task incarnations like the path to the Uspace or the resource broker provided by this Vsite.

The execution TSI section contains information used by the NJS for incarnation of execution tasks. This section describes the configuration of the target machine as number of nodes, processors, software resources for available applications and libraries, resource capacity for memory, processors and cpu time, finally it includes a sub section for queues configuration and priority.

Lastly, the storage TSI section defines a target system that works as an archive server.

Consequently, it is able to execute file transfer and commands on file systems not shared with the home and Uspace file system.

6.2.4 UNICORE User Database

The UUDB holds a mapping from certificates to user identifiers on the target system that are trusted by the NJS. Therefore, the UUDB can be shared by different NJSs in case users have the same login name on the target machines. Furthermore, certificates from other NJSs can be included in the UUDB too.

This is a method for authenticating external NJS in case of a multiple sites job submission.

6.2.5 Target System Interface

The TSI represents a machine that concretely executes a job. It is implemented as a Perl program that interacts with the batch system of the target machine, and, in most cases, it is based on the specific queuing application if the system provides one.

Currently there are different versions available customized for the particular architecture and queuing system. On the other hand, adapting this code for a specific environment should not be too complicated because the TSI functions are very limited. Actually, those functions comprise files management in the Uspace, job submission and controlling and retrieving status for submitted jobs.

6.3 UNICORE installation: client application

6.3.1 Requirements

The UNICORE client is implemented as a Java application whose requirements are a Java 2 Runtime Environment installation, the availability of JSSE, Xerces and AJO libraries, a valid user certificate and a signer certificate from the Certificate Authority. Respectively, the JSSE implements the SSL and Xerces is used for parsing XML documents.

6.3.2 Objectives

The application provides a graphical environment for accessing and making use of the resources available in a UNICORE Grid.

In particular, the most important functions supported concerns the management of user options, especially what is related to security settings, the preparation of UNICORE jobs and the monitoring and controlling of submitted jobs.

6.3.3 Job Preparation Agent

One of the main objectives of UNICORE is to provide a uniform job description that can be understood and executed by different Grid sites. The UNICORE client provides a graphical interface for preparing such a description, hereafter referred as UNICORE job. A UNICORE job is expressed using the AJO library and is organized as a directed acyclic graph consisting of job groups and tasks. Each job has a primary job group that defines the top level Vsite as the final destination for the submission. A job group may contain other job groups or tasks.

Secondary job groups may run on a different Vsite and the tasks are executed on the Vsite defined by the responsible job group.

The client application includes a Job Preparation Agent as a graphical tool for constructing a UNICORE job from job groups and a set of predefined tasks. In the default configuration those tasks are import, export, transfer, file operation, script and command.

Other important aspects in the construction of a UNICORE jobs are the definition of resource requirements and the dependencies between job groups and tasks. At this purpose, two editors are provided by the application: the resource editor and the dependencies editor. First, the resource editor allows the user to define application requirements in terms of computing capabilities such as main memory, number of nodes, CPU time and priority, or in terms of disk space. Second, the dependencies editor establishes the order for the execution of the tasks defined by job groups.

Resource definition and tasks dependencies complete the description of a UNICORE job that can be ready for submission in case the selected Vsites fulfill the requirements. On the other hand, the UNICORE client provides an inspection function that may be used to check the consistency of a UNICORE job and, eventually, to see error details of the job graph.

6.3.4 Job Monitoring and Control

Once a UNICORE job has been submitted to a Vsite, the user can monitor and control its execution through the Job Monitoring and Control (JMC) component provided by the UNICORE client. In relation to the monitoring feature, the JMC reports the current status for each single job group and task that compose the UNICORE job. In particular, a job is considered completed once the top-level job group is finished. Any time during the job life it is possible to retrieve error, output streams and output files marked for being exported to the client local space as soon as those data are generated. Finally, the JMC defines commands for controlling the job execution such as deleting, aborting, holding and resuming a job. In particular, deleting a job is limited to finished jobs. As result of this action every data produced are removed from remote sites. On the other hand, an aborting command removes the job from the UNICORE system without giving the user the possibility to retrieve any information.

Holding and resuming a job are only possible on target systems that support these actions.

6.3.5 Configuration

The UNICORE client needs a valid certificate in PKCS12 format and a copy of the CA signer certificates. The user certificate follows the X.509 V3 standard and contains personal data, a private key, a public key and a copy of the CA signer certificate. In particular, each request from the client is signed with the user private key, and then the UNICORE site is responsible to check the integrity of the request using the public key.

All user and CA certificates are stored locally on the client machine in a password-protected database.

The UNICORE client provides the necessary means for managing the local database such as adding and removing user and CA certificates, changing the protection password, exporting the client public key or checking the CRL.

6.4 UNICORE configuration and adaptation

6.4.1 Application Service Provider

In certain Grid environments not only is essential sharing distributed computing and storage resources, but also it can be useful to access and use a specific set of applications and codes. At this purpose, UNICORE provides an infrastructure that can be customized and configured in order to support Application Service Provider (ASP) features, particularly related to the batch execution of high performance codes. The UNICORE solution requires the configuration of the server side components together with the enhancement of the client application.

6.4.2 Vsite configuration

The Vsite properties are defined through the IDB. In particular, the execution TSI section can contain a list of software resources. Currently, the IDB specification expects two different types of software resources: application and context. The application type is used to represent executable codes, while the context type can indicate libraries, configuration files or execution contexts. At the moment there are three pre-defined software resources that correspond to diverse execution contexts: time on for measured tasks, debug for running in debugging mode and profile for profiling sessions. Other than the type, a software resource is characterized by the name, the version and a description.

The execution section of the IDB lists only the available software resources, and then the run section completes this information by defining how to invoke a specific application and in which context the application will run. Therefore, the information of the run section is specific for the target system. On the other hand, software resource declarations must be agreed by the community in order to support the same application on different Vsites.

6.4.3 Plug-in mechanism

The UNICORE client exploits the dynamic features of the Java runtime environment for loading at run-time the code that implements the JPA and the JMC for a specific application. This mechanism allows extending the standard functionality through external modules and without modifying the main code application. Those modules are indicated as UNICORE client plugins and they are organized as signed jar archives. Even script and command tasks, which come with the standard distribution of the UNICORE client, are implemented as plugins.

6.5 Bibliography

[1] J. Almond, D. Snelling
UNICORE: Secure and Uniform Access to Distributed Resources via the World Wide Web
A White Paper, October 1998

[2] M. Romberg
UNICORE: Beyond Web-based Job-Submission
Proceedings of the 42nd Cray User Group Conference,
May 22-26,2000, Noordwijk

6.6 Acronym list

- AJO: Abstract Job Object
- ASP: Application Service
- CA: Certificate Authority
- CRL: Certificate Revocation List
- NJS: Network Job Supervisor
- IDB: Incarnation DataBase
- JRE: Java Runtime Environment
- JMC: Job Monitoring and Control
- JPA: Job Preparation Agent
- JSSE: Java Secure Socket Extension
- PKI: Public Key Infrastructure
- SSL: Secure Socket Layer
- TSI: Target System Interface
- UUDB: UNICORE User DataBase
- Usite: UNICORE Site (Site providing services to UNICORE)

- Vsite: Virtual Site (A computing resource belonging to a Usite)
- Uspace: UNICORE Space (A dedicated file space for UNICORE jobs)

7 CONCLUSIONS

7.1 Study objectives summary and conclusions

The study reported in this document is organized around six main points: a general analysis on the principal Grid software packages, the current development of Grid technologies, the location of computational Grid testbeds, the reviewing of the applications/projects running on those testbeds, the status of Grid computing for Molecular Simulations and a summary of our practical experiences on implementing a computational Grid with Globus and UNICORE.

The analysis on Grid software packages in Chapter 2 considers particularly Globus, Legion and UNICORE, since at the moment those products characterize most of the current Grid deployments and the most relevant Grid projects. This part aims to provide complementary information with respect to the first ENACTS report "Grid Service Requirements". Actually, the first ENACTS report offers an extensive presentation of available Grid technologies, solutions and products, and we often refer to this document for the technical details and the presentation of less popular Grid products.

Currently the majority of projects and Grid initiatives are based on Globus. Moreover, Globus seams to be dominant also in Grid testbeds, especially in academic testbeds and Grid deployments built on Linux machines and Linux clusters. However, the impression is that the current phase corresponds to an early phase for Grid computing characterized by a difficult interoperability between different Grid technologies, a complex installation and configuration process for Grid products, and finally a very limited number of end users. At the moment there is a growing consensus around the OGSA proposal based on Web Services standards and specifications. In a short time new OGSA compliant Grid products will be available, such as the next Globus 3.0 that will partially implement the OGSA proposal. The expectation is that the following phase of Grid computing, characterized by an agreement on a well-defined set of standards and specifications, will enable an easier interaction between different technologies and components, thus enabling the competition and possibly also the specialization of the various Grid products. The final objective is to provide a set of services to the user community for an effective, seamless and easy access to various, distributed and heterogeneous resources. The success of Grid computing will be measured on the benefits that a final user can get in a day by day work by using Grid technologies and products.

The lists presented in this report of the current testbeds and the applications/projects run on them by no means are supposed to be exhaustive. We know that several others exist which may have been just started. However, the following conclusions can be drawn from the internet search we have carried out:

1. Globus as a Grid deployment technology dominates in both sides of the Atlantic Ocean.

2. Most of the applications concern particle physics, astronomy, environmental and meteorological projects and in less extent biological and chemical applications.

It is interesting to compare the above lists with a previous review the reference of which is given in the Introduction. The explosive growth of testbeds and applications in Grid computing is remarkable. It is worth mentioning the very recent announcement of Mathematica for its new version suitable for a grid environment, the GridMathematica (http://www.wolfram.com/products/gridmathematica/). Mathematica is a popular software package for symbolic and numerical calculations as well as graphics.

Special attention was given for the Grid applications in the field of Molecular Sciences. Although, quantum chemistry has always kept a pioneer role in applying new computer technologies, it turns out that in the Grid case there is no much progress. This is because distributing computing requires new numerical algorithms and new ways of programming. At present, there are just a few examples which apply the established Grid deployments, but we believe in the near future we shall see them expanding. We may think that parallelized codes are ready for running in a Grid. This is not true since the employment of thousands or even millions of computers requires different programming approaches. At present only ploblems with trivial ('embarrassing') parallelization can be treated in a Grid.

The last two chapters of the document describe our experiences in installing Globus and UNICORE. Those chapters intend to give an idea how the models and concepts, expressed in the second chapter, are then implemented in reality. Therefore, the intention is to complement this study with practical experiences rather than provide a user guide for installing Globus or UNICORE.

Appendix A Web Bibliography

ENACTS

- ENACTS Project at EPCC
  http://www.epcc.ed.ac.uk/enacts/

ARTICLES

- LAM / MPI Parallel Computing
  http://www.mpi.nd.edu/lam/
- The Numerical Algorithms Group Ltd
  http://www.nag.co.uk/
- Welcome to Linux.Corel.Com
  http://linux.corel.com/
- NIST Chemistry WebBook
  http://webbook.nist.gov/chemistry/
- Ian Foster's home page
  http://www-fp.mcs.anl.gov/~foster/
- The Grid
  http://www.mkp.com/books_catalog/catalog.asp?ISBN=1-55860-475-8
- Globus: Technical Papers
  http://www.globus.org/research/papers.html#anatomy
- CORDIS: IST: Future and Emerging Technologies: Global Computing
  http://www.cordis.lu/ist/fetgc.htm

MIDDLEWARE & TECHNOLOGIES

- The Globus Project
  http://www.globus.org/
- Condor Project Homepage
  http://www.cs.wisc.edu/condor/
- Legion: A Worldwide Virtual Computer
  http://legion.virginia.edu/
- UNICORE
  http://www.UNICORE.de/
- SOAP Version 1.2 Part 0:
  Primer W3C Working Draft 26 June 2002
  N. Mitra
  http://www.w3.org/TR/soap12-part0
- Web Services Description Language (WSDL) 1.1
  W3C Note 15 March 2001

E. Christensen, F. Curbera, G. Meredith, S. Weerawarana
http://www.w3.org/TR/wsdl

- Web Services Inspection Language (WS-Inspection) 1.0
  K. Ballinger, P. Brittenham, A. Malhotra, W. A. Nagy, S. Pharies
  http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html
- UDDI
  http://www.uddi.org/
- Web Services Flow Language (WSFL) Version 1.0
  F. Leymann
  http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf
- XLANG - Web Services for Business Process Design
  Satish Thatte
  http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm
- Resource Description Framework (RDF) Model and Syntax Specification
  W3C Recommendation 22 February 1999
  Ora Lassila, Ralph R. Swick, eds.
  http://www.w3.org/TR/REC-rdf-syntax
- RDF Vocabulary Description Language 1.0: RDF Schema
  W3C Working Draft 30 April 2002
  Dan Brickley, R.V. Guha, eds.
  http://www.w3c.org/TR/rdf-schema/
- Jini Community
  http://www.jini.org
- JXTA Project
  http://www.jxta.org/

TESTBEDS

- Globus Testbeds
  http://www.globus.org/research/testbeds.html
- EUROGRID Project
  http://www.euroGrid.org/
- EUROGRID Project WP1: Bio GRID
  http://www.euroGrid.org/wp1.html
- Legion Testbeds
  http://legion.virginia.edu/testbeds.html

GRIDS

- International Virtual Data Grid Laboratory
  http://www.ivdgl.org/
- BioGRID resources
  http://bioGrid.icm.edu.pl/resources/index.php3

- About TeraGrid
  http://www.ncsa.uiuc.edu/About/TeraGrid/
- About The Alliance
  http://www.ncsa.uiuc.edu/About/Alliance/
- About NSF PACI
  http://www.ncsa.uiuc.edu/About/PACI/
- Global Grid Forum
  http://www.globalGridforum.org/L_About/about.htm
- EUROGRID Project
  http://www.euroGrid.org/
- Rajkumar Buyya's Home Page
  http://www.buyya.com/ecoGrid
- The DataGrid Project
  http://www.eu-datagrid.org/
- Legion: High Performance
  http://legion.virginia.edu/overview_high.html
- Welcome to MCS
  http://www-fp.mcs.anl.gov/division/welcome/default.asp
- NEOS Server for Optimization
  http://www-neos.mcs.anl.gov/
- GriPhyN -- Grid Physics Network
  http://www.griphyn.org/
- NEESGrid Web Site Homepage
  http://www.neesGrid.org/
- CCLRC - Daresbury Laboratory - Home Page
  http://www.dl.ac.uk/
- Energy Dept., IBM to unveil Science Grid - Tech News - CNET.com
  http://news.com.com/2100-1001-866452.html
- Grid Computing Info Centre: GRIDInfoware
  http://www.Gridcomputing.com/
- Particle Physics Data Grid
  http://ppdg.net/
- NEESGrid Web Site Homepage
  http://www.neesGrid.org/
- Nature Web Matters: Internet Computing and the Emerging Grid
  http://www.nature.com/nature/webmatters/Grid/Grid.html
- Welcome to ApGrid
  http://www.apGrid.org/
- GridPP - The UK Grid for Particle Physics
  http://www.Gridpp.ac.uk/
- AstroGrid Portal: Home
  http://www.astroGrid.ac.uk/
- US National Virtual Observatory
  http://us-vo.org/

- Grid Computing Planet
  http://www.Gridcomputingplanet.com/
- Avaki Corporation - About Grid Compouting
  http://www.avaki.com/products/aboutGrid.html
- US ATLAS Grid Activities
  http://www.usatlas.bnl.gov/computing/Grid/
- Initiative GRID-FRANCE
  http://Grid-france.in2p3.fr/index-e.html
- Globus and EDG-deployment on fs2
  http://www.dutchGrid.nl/Admin/das2-uva/index.html
- MyGrid
  http://myGrid.man.ac.uk/
- UK e-Science (GRID) core programme
  http://www.escience-Grid.org.uk/
- DIET – Introduction
  http://www.ens-lyon.fr/%7Edesprez/DIET/diet_intro.html
- The Grid Resource Broker
  http://sara.unile.it/grb/grb.html
- DutchGrid Platform for large-scale distributed computing in The Netherlands
  http://www.dutchGrid.nl/
- Scandinavian Grd
  http://www.nordugrid.org/

APPLICATIONS

- Grid Application and Deployment Projects
  http://www-fp.mcs.anl.gov/~foster/Grid-projects/
- Centurion Home Page
  http://legion.virginia.edu/centurion/Centurion.html
- Centurion Applications
  http://legion.virginia.edu/centurion/Applications.html
- BioGRID
  http://bioGrid.icm.edu.pl/
- MetaNEOS Project
  http://www-unix.mcs.anl.gov/metaneos/
- NetSolve: Overview
  http://icl.cs.utk.edu/netsolve/overview/