Università degli studi di Ferrara

Dottorato di Ricerca in Matematica e Informatica

Ciclo XXIII

Coordinatore: Prof. Luisa Zanghirati

NetJobs: A new approach to network monitoring for the Grid using Grid jobs

Settore Scientifico Disciplinare INF/01

Dottorando:
Dott. Alfredo Pagano

Tutore:
Prof. Eleonora Luppi

CoTutore:
Dr. Mario Reale

Anni 2008/2010

A tutti quelli che mi hanno dato una mano
e ai miei genitori che me ne hanno date quattro

Contents

Abstract

With grid computing, the far-flung and disparate IT resources act as a single "virtual datacenter". Grid computing interfaces heterogeneous IT resources so they are available when and where we need them. Grid allows us to provision applications and allocate capacity among research and business groups that are geographically and organizationally dispersed.

Building a high availability Grid is hold as the next goal to achieve: protecting against computer failures and site failures to avoid downtime of resource and honor Service Level Agreements.

Network monitoring has a key role in this challenge.

This work is concerning the design and the prototypal implementation of a new approach to Network monitoring for the Grid based on the usage of Grid scheduled jobs. This work was carried out within the Network Support task (SA2) of the Enabling Grids for E-sciencE (EGEE) project.
This thesis is organized as follows:

Chapter 1: Grid Computing From the origins of Grid Computing to the latest projects. Conceptual framework and main features characterizing many kind of popular grids will be presented.

Chapter 2: The EGEE and EGI projects This chapter describes the Enabling Grids for E-sciencE (EGEE) project and the European Grid Infrastructure (EGI).

EGEE project (2004-2010) was the flagship Grid infrastructure project of the EU. The third and last two-year phase of the project (started on 1 May 2008) was financed with a total budget of around 47 million euro, with a further estimated 50 million euro worth of computing resources con-

tributed by the partners. A total manpower of 9,000 Person Months, of which over 4,500 Person Months has been contributed by the partners from their own funding sources.
At its close, EGEE represented a worldwide infrastructure of approximately to 200,000 CPU cores, collaboratively hosted by more than 300 centres around the world. By the end of the project, around 13 million jobs were executed on the EGEE grid each month. The new organization, EGI.eu, has then been created to continue the coordination and evolution of the European Grid Infrastructure (EGI) based on EGEE Grid.

Chapter 3: gLite Middleware Chapter three gives an overview on the gLite Grid Middleware.

gLite is the middleware stack for grid computing used by the EGEE and EGI projects within a very large variety of scientific domains. Born from the collaborative efforts of more than 80 people in 12 different academic and industrial research centers as part of the EGEE Project, gLite provides a complete set of services for building a production grid infrastructure. gLite provides a framework for building grid applications tapping into the power of distributed computing and storage resources across the Internet. The gLite services are currently adopted by more than 250 Computing Centres and used by more than 15000 researchers in Europe and around the world.

Chapter 4: Network Activity in EGEE/EGI

Grid infrastructures are distributed by nature, involving many sites, normally in different administrative domains. Individual sites are connected together by a network, which is therefore a critical part of the whole Grid infrastructure; without the network there is no Grid. Monitoring is a key component for the successful operation of any infrastructure, helping in the discovery and diagnosis of any problem which may arise. Network monitoring is able to contribute to the day-to-day operations of the Grid by helping to provide answers to specific questions from users and site administrators.

This chapter will discuss all the effort lavished by EGEE and EGI in the Grid Network domain.

Chapter 5: Grid Network Monitoring based on Grid Jobs

NetJobs is a prototype of a light weight solution for the Grid network monitoring. A job-based approach has been used in order to prove the feasibility of this non intrusive solution. It is currently configured to monitor eight production sites spread from Italy to France but this method could be applied to the vast majority of Grid sites. The prototype provides coherent RTT, MTU, number of hops and TCP achievable bandwidth tests.

Abstract

Grazie al Grid computing, risorse eterogenee e geograficamente lontane possono apparire come "datacenter virtuali". Cicli di calcolo, spazio disco, reti ad alta velocità sono disponibili senza barriere di tempo e distanze; il tutto diviene fruibile quando e dove se ne ha bisogno. Grid permette in tal modo di progettare applicazioni sia per il mondo della ricerca che per quello dell'industria, mondi spesso molto lontani.

Ottenere Grid ad alta affidabilità è il prossimo traguardo da raggiungere: far fronte a interruzioni di servizi e indisponibilità di risorse per rispettare gli impegni presi, meglio conosciuti come SLA (Service Level Agreement), è la naturale evoluzione del sistema Grid.

Il controllo della rete ha un ruolo chiave in questa sfida.

Il lavoro di questa tesi si concentra sull'analisi di strumenti per il monitoring di rete in ambito Grid ed in particolare sullo sviluppo di un nuovo software da un approccio alternativo al controllo della rete di Grid. Tale approccio consiste nell'utilizzo di Grid jobs per il controllo di rete e rappresenta una soluzione nuova e non intrusiva ad un annoso quanto mai attuale problema.

Questo lavoro è stato svolto nell'ambito del progetto di Grid Europea EGEE e più specificamente all'interno dell'unità di supporto di rete EGEE SA2.

Questo lavoro di dottorato è organizzato in 5 capitoli:

Capitolo 1: Grid Computing Origini ed evoluzioni del sistema Grid. Analisi della architettura di Grid: i blocchi da cui prende forma ed il modo in cui questi interagiscono.

Capitolo 2: I progetti EGEE ed EGI Il progetto EGEE (Enabling Grids for E-sciencE) ha rapprensentato il primo progetto di calcolo distribuito su larga scala finanziato dalla comunità europea. Nel biennio della sua terza fase (Maggio 2008 - Maggio 2010) il progetto è stato finanziato con un totale di circa 47,150,000 euro oltre a 50,000,000 euro di risorse computazionali come contributo dei singoli partner. Risorse umane per 9,010 uomo/mese, di cui oltre 4,500 uomo/mese derivanti dalle finanze dei partner del progetto.

L'infrastruttura di calcolo distribuito costruita e cresciuta con i progetti DataGrid (2002-2004), EGEE-I, -II e -III (2004-2010) viene ora supportata dalla nuova European Grid Initiative (EGI). Sarà questa organizzazione a lungo termine che coordinerà d'ora in poi le iniziative nazionali (National Grid Initiative), i veri blocchi costruttivi della griglia paneuropea

Capitolo 3: gLite Middleware Il middleware Grid è il software che si posiziona tra il sistema operativo e le applicazioni e che permette un accesso sicuro ed omogeneo alle risorse, a prescindere dalle loro specificità implementative. Il progetto EGEE ha sviluppato un gruppo di componenti che costituiscono il middleware denominato gLite. Verranno descritte le componenti principali e il modello di funzionamento del middleware stesso.

Capitolo 4: L'importanza della rete nei progetti EGEE e EGI
Le infrastrutture di Grid sono distribuite per loro stessa natura, coinvolgendo molti siti e spesso diversi domini amministrativi.

I siti sono connessi tra loro da reti informatiche, che rappresentano quindi una parte critica dell'intera infrastruttura di Grid; senza rete non può esistere una Grid. Il monitoring o controllo è un elemento chiave per un corretto funzionamento di qualsiasi infrastruttura, permette di analizzare e diagnosticare qualsiasi problema sorto. Il monitoring di rete è inoltre fondamentale nel contribuire al controllo quotidiano delle Grid venendo incontro sia agli amministratori che ai singoli utenti della Grid.

In questo capitolo sarà trattato lo sforzo profuso dal progetto EGEE e dall'infrastruttura permanente EGI nell'ambito della sinergia rete-Grid.

Capitolo 5: Netjobs: Un nuovo approccio al controllo di rete per la Grid tramite Grid jobs

NetJobs è il nome dato ad un tool efficace e poco intrusivo sviluppato per il controllo di rete tra siti Grid. NetJobs è in fase prototipale ed al momento configurato per il controllo di 8 siti Grid distribuiti tra Italia e Francia, ma la sua flessibilità e scalabilità gli permettono di gestire un alto numero di siti senza difficoltà. Il tool è in grado di eseguire misure di rete RTT, MTU, numero di hops e larghezza di banda TCP. Ne verranno descritte le varie fasi di analisi e design, implementazione e collaudo.

Chapter 1

Grid Computing

1.1 Grid Computing

Figure 1.1: The Grid

Grid computing is an emerging computing model that provides the ability to perform higher throughput computing by taking advantage of many networked computers to model a virtual computer architecture that is able to distribute process execution across a parallel infrastructure. Grids use the resources of many separate computers connected by a network (usually the Internet) to solve large-scale computation problems. Grids provide the ability to perform computations on large data sets, by breaking them down into many smaller ones, or provide the ability to perform many more com-

putations at once than would be possible on a single computer, by modeling a parallel division of labour between processes. Today resource allocation in a grid is done in accordance with SLAs (service level agreements).

1.1.1 Origins

Like the Internet, the Grid Computing evolved from the computational needs of "big science". The Internet was developed to meet the need for a common communication medium between large, federally funded computing centers. These communication links led to resource and information sharing between these centers and eventually to provide access to them for additional users. Ad hoc resource sharing 'procedures' among these original groups pointed the way toward standardization of the protocols needed to communicate between any administrative domain. The current Grid technology can be viewed as an extension or application of this framework to create a more generic resource sharing context.

Fully functional proto-Grid systems date back to the early 1970's with the Distributed Computing System [1] (DCS) project at the University of California, Irvine. David Farber was the main architect. This system was well known enough to merit coverage and a cartoon depiction in Business Week on July 14, 1973. The caption read "The ring acts as a single, highly flexible machine in which individual units can bid for jobs". In modern terminology ring = network, and units = computers, very similar to how computational capabilities are utilized on the Grid. The project's final report was published in 1977 [2] . This technology was mostly abandoned in the 1980's as the administrative and security issues involved in having machines we did not control do our computation were (and are still by some) seen as insurmountable.

The ideas of the Grid were brought together by Ian Foster, Carl Kesselman and Steve Tuecke, the so called "fathers of the Grid." They lead the effort to create the Globus Toolkit incorporating not just CPU management (e.g. cluster management and cycle scavenging) but also storage management, security provisioning, data movement, monitoring and a toolkit for developing additional services based on the same infrastructure including

agreement negotiation, notification mechanisms, trigger services and information aggregation. In short, the term Grid has much further reaching implications than the general public believes. While Globus Toolkit remains the de facto standard for building Grid solutions, a number of other tools have been built that answer some subset of services needed to create an enterprise Grid.

The remainder of this article discusses the details behind these notions.

1.1.2 Definitions of Grid computing

The term Grid computing originated in the early 1990s as a metaphor for making computer power as easy to access as an electric power grid.

Today there are many definitions of Grid computing:

- The definitive definition of a Grid is provided by Ian Foster in his article "What is the Grid? A Three Point Checklist" [3] The three points of this checklist are:

  1. Computing resources are not administered centrally.
  2. Open standards are used.
  3. Non-trivial quality of service is achieved.

- Plaszczak/Wellner define Grid technology as "the technology that enables resource virtualization, on-demand provisioning, and service (resource) sharing between organizations."

- IBM says, "Grid is the ability, using a set of open standards and protocols, to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet. A Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of resources distributed across multiple administrative domains based on the resources availability, capacity, performance, cost and users' quality-of-service requirements" [4]

- An earlier example of the notion of computing as utility was in 1965 by MIT's Fernando Corbató. Fernando and the other designers of the Multics operating system envisioned a computer facility operating "like a power company or water company".

- Buyya defines Grid as "a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements". [5]

- CERN, one of the largest users of Grid technology, talk of The Grid: "a service for sharing computer power and data storage capacity over the Internet." [6]

- Pragmatically, Grid computing is attractive to geographically distributed non-profit collaborative research efforts like the NCSA Bioinformatics Grids such as BIRN: external Grids.

- Grid computing is also attractive to large commercial enterprises with complex computation problems who aim to fully exploit their internal computing power: internal Grids.

Grids can be categorized with a three stage model of departmental Grids, enterprise Grids and global Grids. These correspond to a firm initially utilizing resources within a single group i.e. an engineering department connecting desktop machines, clusters and equipment. This progresses to enterprise Grids where non-technical staff's computing resources can be used for cycle-stealing and storage. A global Grid is a connection of enterprise and departmental Grids which can be used in a commercial or collaborative manner.

Grid computing is a subset of distributed computing [7]

1.1.3 The dream

Imagine a lot of computers, let's say several million. They are desktop PCs and workstations, mainframes and supercomputers, but also data vaults and instruments such as meteorological sensors and visualization devices.

Imagine they are situated all over the world. Obviously, they belong to many different people (students, doctors, secretaries...) and institutions (companies, universities, hospitals...).

So far we have imagined nothing new. This is pretty much what the world looks like today.

Now imagine that we connect all of these computers to the Internet. Still not much new, most of them are probably connected already.

Now imagine that we have a magic tool which makes all of them act as a single, huge and powerful computer. Now that really is different. This huge, sprawling mess of a computer is what some dreamers think "The Grid" will be.

Well, if we are a scientist, and we want to run a colleague's molecular simulation program, we would no longer need to install the program on our machine. Instead, we could just ask the Grid to run it remotely on our colleague's computer. Or if our colleague was busy, we could ask the Grid to copy the program to another computer, or set of computers, that were sitting idle somewhere on the other side of the planet, and run our program there. In fact, we wouldn't need to ask the Grid anything. It would find out for we the best place to run the program, and install it there.

And if we needed to analyze a lot of data from different computers all over the Globe, we could ask the Grid to do this. Again, the Grid could find out where the most convenient source of the data is without we specifying anything, and do the analysis on the data wherever it is.

And if we wanted to do this analysis interactively in collaboration with several colleagues around the world, the Grid would link our computers up so it felt like we were all on a local network. This would happen without we having to worry about lots of special passwords, the Grid could figure out who should be able to take part in this common activity.

1.1.4 The reality

"The Grid", as just described, is definitely a dream.

But reality is catching up fast with this dream. And as they say, fact is usually weirder than fiction.

So where the Grid might be in ten years time, and what it might do, nobody knows. One way to get an idea of what might happen, though, is to look at how the evolution of computing has naturally led to the concept of the Grid.

- **Distributed computing** Nowadays, whenever there is a problem due to lack of computing power (a complicated calculation or an applications that require more computing power than a single computer can provide) the solution is to link computer resources from across a business, a company or an academic institution. The network of computer is then used as a single, unified resource.

  This solution is called "distributed computing", and this term refers to just about any system where many computers solve a problem together. Grid computing is, in a sense, just one species of distributed computing. There are many others, a few of which are listed below.

- **Metacomputing** Metacomputing was a name coined for a particular type of distributed computing, very popular in the early 'nineties, which involved linking up supercomputer centers with what was, at the time, high speed networks.

- **Cluster Computing** Many years ago, back in the last century, scientists put some PCs together and got them to communicate. The first cluster was called Beowulf, after a Norse hero who killed a dragon.

  The dragon these scientists were trying to kill was the expensive mainframe or supercomputer. They succeeded beyond their wildest dreams. Many commercial companies now offer clusters of PCs as a standard off-the-shelf solution.

  Clusters can have different sizes. One of the big advantages of this approach is "scalability": a cluster can grow simply by adding new

PCs to it. Of course there are limits, because somehow the computers have to communicate with each other, and this starts to get pretty hairy when there are many computers. But clusters of hundreds of computers are not uncommon nowadays.

- **Peer to Peer computing** We must have heard also about "Napster", the website that used to let music fans share music files from all over the world. By downloading a piece of software onto our hard drive, we could connect to a network of other users who have downloaded the same software. Users only had to specify which information on their hard drive was public, and could access what others had made public.

  In this way computers can share files and other data directly, without going through a central server.

- **Internet computing** We may have heard about SETI@home. Based at the University of California - Berkeley, SETI@home is a virtual "supercomputer" which analyzes the data of the Arecibo radio telescope in Puerto Rico, searching for signs of extraterrestrial intelligence. Using the Internet, SETI brings together the processing power of more than 3 million personal computers from around the world, and has already used the equivalent of more than 600.000 years of PC processing power!

  SETI@home is a screen-saver program - i.e. it works without impacting normal use of the computer - and any owner of a PC can download it from the Web. The different PCs (the nodes of such Grid) work simultaneously on different parts of the problem, retrieving chunks of data from the Internet and then passing the results to the central system for post-processing. The success of SETI has inspired many other @home applications.

  SETI@home is also an example of the concept of "cycle scavenging". The term means that we rely on getting free time on computers which we do not control. For SETI@home this is based on goodwill, because

so many people are interested in the goal of the project. Clearly, cycle scavenging is not a viable strategy for every computing task.

- **Local grid computing** Nowadays, for problems that can be divided into many smaller problems, all independent of each other, the solution is to link computer resources from across a business, a company or an academic institution. The network of computer is then used as a single, unified resource.

  This solution belongs to the general class of computing called "distributed computing". Nowadays a lot of people call this solution Grid computing, although it fails by some definitions. So "local Grid computing" is one way to distinguish it. "Networks of workstations", now, is another common name for it.

  Local Grid computing makes the most of existing computer resources within an organization. Dedicated software efficiently matches the processing power required by any application with the overall availability. One popular type of software for linking computers in institutions like universities is Condor. Condor is a type of software often referred to as "middleware", because it is not the operating system - the program that runs the computer - nor is it an application program running on the computer, but it is "between" these two, making sure that the application can run optimally on several computers, by automatically checking which computers are available. No more wasting time waiting for available computing power while systems in the next office remain idle!

  Like clusters, local Grid computing is scalable - we can keep adding more PCs and workstations, within reasonable limits. Often the connection between the computers in such a system is a local area network, although it can also be via Internet. Usually the computers are geographically close together, for instance in the same building, and belong to the same administrative domain.

  Local Grid computing is limited to a well-defined group of users, a department or several departments inside a corporate firewall, or a few

trusted partners across the firewall. Also, such systems typically pool the resources of some dedicated PCs as well as others whose primary purpose is not distributed computing - in other words it involves some "cycle scavenging", at least on a local scale.

- **Grid computing** Grid computing can be seen as the evolution of local Grid computing to the global scale, made possible by the advent of very high-speed Internet connections, and of powerful computer processors that are able to run quite complex middleware in the background without disturbing the task that the computer is trying to handle.

  As Internet connect speed increases, the difference between having two PCs in the same office, the same building, the same city or the same country shrinks. And by developing sophisticated middleware which makes sure widely distributed resources are used effectively, Grid computing gives the user the impression of shrinking the distances further still. Furthermore, as the middleware gets more sophisticated, it can deal with the inevitable differences between the types of computers that are being used in a highly distributed system, which are harder to control than within one organization.

  One of the most popular middleware packages today is called Globus, and it is essentially a software toolkit for making Grids. With such middleware, the aim is to couple a wide variety of machines together effectively, including supercomputers, storage systems, data sources and special classes of devices such as scientific instruments and visualization devices.

  Grid computing focuses more on large scale sharing, which goes beyond institutional boundaries. Also, Grid computing leans more to using dedicated systems, such as scientific computer centers, rather than cycle scavenging. Finally, and what is in some ways the most challenging aspect, Grid computing aims to use resources that are not centrally controlled. The sharing is across boundaries - institutional and even national - which adds considerable complexity, while

bringing also huge potential benefits.

One definition of Grid computing, by Ian Foster, one of the persons who helped coin the term, distinguishes it from other forms of computing.

The definition is that a full-blown Grid must satisfy three criteria:

1. no central administrative control of the computers involved (that eliminates clusters and farms, and also local Grid computing)
2. Use of general-purpose protocols (that eliminates single-purpose systems such as SETI@home)
3. High quality of service (that eliminates peer-to-peer and means that Grids should not rely on cycle scavenging from individual processors, but rather on load balancing between different independent large resources, such as clusters and local Grids)

Another distinction is that a Grid could in principle have access to parallel computers, clusters, farms, local Grids, even Internet computing solutions, and would choose the appropriate tool for a given calculation. In this sense, the Grid is the most generalized, globalized form of distributed computing one can imagine.

1.1.5 The evolution

Referring to Grid computing as "The Grid" is a convenient shorthand, but it also can lead to a lot of confusion.

The reality, now and for a while to come, is that there is not one single "Grid" (as there is one single "Internet" and one single "Web"). Indeed, there are some experts who believe that there may never be one single Grid. Instead, there are many Grids evolving, some private, some public, some within one region or country, some of truly global dimensions. Some dedicated to one particular scientific problem, some all-purpose.

Compared to the Dreamers' Grid, these Grids all have very restricted capabilities for the moment. But they are gradually growing and becoming more sophisticated. And thanks to the Dreamers, there is still a lot of

enthusiasm for the same long-term vision: a scenario where the computer power and storage capacity of millions of systems across a worldwide network function as a pool that could be used by pretty much anyone who needs it.

To achieve this, complex systems of software and services must be developed. That's why many IT experts all over the world, from science and from industry, have started Grid development efforts.

From Web services to grid services

One of the areas where the interests of scientists and businessmen converge is "standards". If everybody starts making their own kind of Grid, then it becomes difficult and expensive to combine Grid technologies.

Fortunately, there is an activity underway since 2002 to define Grid standards, called the Open Grid Standards Architecture, which is supported both by a large part of the scientific community and increasingly by industry. OGSA is a spin off of the Global Grid Forum, a self-appointed organization that runs several international GGF meetings each year - the first one was in March 2001.

What OGSA is trying to do, basically, is to harmonize the work going on to develop the Globus Toolkit - primarily an academic initiative, with so-called "Web Services", which industry is pushing in order to provide a common standard for services offered over the World Wide Web. In practice, OGSA is being championed by the academic team behind the Globus Toolkit in collaboration with IBM.

The technic definition is "a software application accessible via Internet protocols using XML for messaging, description and discovery". XML (eXtended Markup Language) is a powerful computer syntax for communicating information across networks, which can be likened to a much more sophisticated version of HTML, the markup language used on websites to provide links to other sites.

So Web services use XML for communication, for describing the type of services available, and for discovering services on the Web. Examples of Web Services are stock quotes, weather reports, and anything that needs to

communicate between a website and a data producer in order to give the user some updated information.

The current view in the field is that Grid Services will in practice be just a sub-class of Web Services, but which give access to the sort of computing power that Grids enable. Maybe running a very complex analysis of our own stock portfolio or giving we a local weather report at the exact place we happen to be.

From Scientific Grids to Commercial Grids

Scientists want to make discoveries. Businessmen want to make money. So when it comes to Grid technology, their views are not identical.

Still, since scientists depend heavily on commercial IT solutions, and industry benefits by science-driven innovation, there are strong links between the Grid that scientists are dreaming about, and new types of technologies and services that many industrial companies are introducing.

Different vendors have created and marketed distributed computing systems for years, and commercial grid solutions are now appearing on the market. Most of them focus on the "enterprise" model, which provides dependable, consistent, and inexpensive access to computing resources inside a single business.

The enterprise model surely helps an enterprise to lower costs, enter new areas of development and develop better products. However the sharing arrangements are typically quite restricted and static.

In general, commercial distributed computing technologies do not address broad scientific concerns, such as the need of flexible sharing relationships among different organizations, and the need to deal with different hardware and software from different makers.

While the commercial world is mainly concentrating on solving Grids for single enterprises, the research world is setting up, testing and deploying large collaborative grid infrastructures (testbeds) that span several countries and many institutions.

1.2 How the Grid works

There are different ways to explain how the Grid works.

Conceptually, there are five big ideas that distinguish the Grid from other types of distributed computer systems - or are at least crucial to the Grid's success. The conceptual view is given by "five big ideas":

1. Resource Sharing
2. Secure Access
3. Resource Use
4. The Death of Distance
5. Open Standards

Of course, there are many big ideas behind the Grid. And of course, some of them have been around long before the name Grid appeared. Nevertheless, if we look at where the software engineers and developers who are building the Grid are spending their time and effort, then there are five big areas.

The most important is the sharing of resources on a global scale. This is the very essence of the Grid. Then, although it is hardly a novelty, security is a critical aspect of the Grid, since there must be a very high level of trust between resource providers and users, who will often never know who each other are. Sharing resources is, fundamentally, in conflict with the ever more conservative security policies being applied at individual computer centers and on individual PCs. So getting Grid security right is crucial.

If the resources can be shared securely, then the Grid really starts to pay off when it can balance the load on the resources, so that computers everywhere are used more efficiently, and queues for access to advanced computing resources can be shortened. For this to work, however, communications networks have to ensure that distance no longer matters - doing a calculation on the other side of the globe, instead of just next door, should not result in any significant reduction in speed.

Finally, underlying much of the worldwide activity on Grids these days is the issue of open standards, which are needed in order to make sure that R&D worldwide can contribute in a constructive way to the development of the Grid, and that industry will be prepared to invest in developing commercial Grid services and infrastructure.

1.2.1 Resource sharing

The First Big Idea behind the Grid is sharing of resources: We enter the Grid to use remote resources, which allows we to do things that we cannot do with the computer we own, or the computer center we normally use (if we are, say, a scientist doing sophisticated computer simulations). This is more than simple file exchange: it is direct access to remote software, computers and data. It can even give we access and control of remote sensors, telescopes and other devices that do not belong to we.

A major challenge for the implementation of the Grid come from this very simple fact: resources are owned by many different people. This means that they exist within different administrative domains, they run different software, and they are subject to different security and access control policies.

Grid philosophy is about creating a situation amongst owners of computer resources where everyone concerned sees the advantage of sharing, and there are mechanisms in place so that each resource provider feels they can trust any user who is trusted by any other resource provider. For example, when the persons in charge of a computer centre decide to share their resources on the Grid, they will normally put conditions on the use of those resources, specifying limits on which resources can be used when, and what can be done with them.

1.2.2 Secure Access

The Second Big Idea behind the Grid could be summarized as secure access, and is a direct consequence of the first big idea. Sharing resources creates some of the most challenging issues for Grid development:

- Access policy - resource providers and users must define clearly and carefully what is shared, who is allowed to share, and the conditions under which sharing occurs;

- Authentication - we need a mechanism for establishing the identity of a user or resource;

- Authorization - we need a mechanism for determining whether an operation is consistent with the defined sharing relationships.

Of course, the Grid needs an efficient way to keep track of all this information: who is authorized to use the Grid, and which resources on the Grid are they authorized to use? Who authenticates that a given user is who he says he is? What are the usage policies of the different resources?

All these things may change from day to day, so the Grid needs to be extremely flexible, and have a reliable accounting mechanism. Ultimately, the accounting will be used to decide pricing policy for the Grid. In IT security, it is common to talk about the "three A's", Authorization, Authentication and Accounting, and this is certainly true for the Grid.

The problems are not new - in a sense it is the same sort of issue that goes on behind the scences when we use our credit card in a restaurant. The difference is that the Grid requires new types of solutions to these problems. It is as though the owner of a café were to lend some of his tables to another café and the waiters would have to keep track of who gets paid what.

Behind all these issues of trust there is the underlying issue of security. We may trust the other users, but do we trust that our data and applications are protected as they flow across the Internet to other computer resources, or while they are being processed on other computers? Without adequate security, it is actually possible today for someone to use our data (confidential or otherwise) and possibly modify it on its path over the Internet - hence the warnings we get about security everytime we use our credit card on the internet. Also, without adequate security, it is possible that while our data is residing on another computer on the Grid, the owner of that computer - or some crackers - could read our data.

A lot of work is going on to find a solution to all of these issues, which really concern the whole spectrum of Information Technologies and not just the Grid. Security, for example, is being addressed by sophisticated encryption techniques both during data transmission and also during their representation/storage on external resources. New solutions for many of security issues are constantly being developed. But it is a never-ending race to stay ahead of malicious crackers.

1.2.3 Resource Use

The Third Big Idea behind the Grid, when we have got all the formalities of sharing resources sorted out, is efficient use of resources. This is where the Grid really starts to look interesting, even for someone blessed with a lot of computer resources. Because no matter how many resources we have, there will always be times when there is a queue of people waiting to use them. If we have a mechanism to allocate work efficiently and automatically among many resources, we can reduce the queues.

On the Grid, in principle, we have the information about the different jobs being submitted, and since the whole thing is running on computers, we should be able to calculate the optimal allocation of resources. The development of the middleware, the software that performs this task and in general manages activity on the Grid, is the main purpose of many of the Grid projects going on today around the world.

1.2.4 The Death of Distance

The Fourth Big Idea behind the Grid could be called the death of distance. High-speed connections between computers make a truly global Grid possible. Ten years ago, it would have been stupid to try to send large amounts of data across the globe to get it processed more quickly on other computer resources, because the time taken to transfer the data would nullify the benefit of quicker processing.

What makes the Grid possible today is the impressive development of networking technology. Pushed by the Internet economy and the widespread penetration of optical fibers in telecommunications systems, the perfor-

mance of wide area networks has been doubling every nine months or so over the last few years. Some wide area networks now operate at 155 megabits per second (Mbps), when in 1985 the US supercomputer centers were connected at 56 kilobits per second (Kbps) - that is a 3000x improvement in 15 years.

Of course, distance never really dies, because somebody will always have a problem for the Grid which makes even the fastest connections seem slow. For example, to work with colleagues across the world to analyse large amounts of data, some scientists will need even higher-speed connectivity, up to tens of gigabits per second (Gbps). Other scientists will demand ultra-low latency for their applications, so there is no delay when working with colleagues in real time on the Grid.

Still others will want to ensure "just-in-time" delivery of data across the Grid so that complicated calculations can be performed which require constant communication between processors. To avoid communication bottlenecks, Grid developers have also to figure out ways to compensate for any failure that occurs on the Grid during a calculation, be it a transmission error or a PC crash.

To meet such critical requirements, several high-performance networking issues have to be solved, which include the optimization of Transport Protocols and the development of technical solutions such as high-performance Ethernet switching.

1.2.5 Open Standards

The Fifth Big Idea behind the Grid is open standards. The idea is to convince the community of software engineers currently developing the Grid, including those from major IT companies, to set common standards for the Grid up-front, so that applications made to run on one Grid will run on all others. This may seem idealistic - after all, many software companies make their profits precisely because they do not share their standards with others. However, because the very nature of the Grid is about sharing, it is generally perceived to be in everyone's self interest to set common, open standards.

The sticky question is, whose standards should be used for the Grid?

There are dozens of projects and hundreds of software developers working worldwide on creating the Grid, each with their own views on what is a good standard. While they work, technology continues to evolve and provides new tools that need to be integrated within the Grid machinery, which may require revising the standards.

Who is in charge of choosing standards - and who can suggest revisions?

Both the Internet, and the Web have key standards such as TCP/IP and HTTP, which have been critical for the progress in these communities. These standards have been set by standards bodies, which have been created usually by some grassroots movement and evolve standards through some sort of consensual process. The IETF is a standards body for the Internet and W3C is one for the Web.

Grid-specific standards are currently being developed by the Open Grid Forum, a similar sort of standards body.

The Open Grid Forum (OGF) is a community of users, developers, and vendors leading the global standardization effort for grid computing. The OGF community consists of thousands of individuals in industry and research, representing over 400 organizations in more than 50 countries. Together they work to accelerate adoption of grid computing worldwide because we believe grids will lead to new discoveries, new opportunities, and better business practices.

The work of OGF is carried out though community-initiated working groups, which develop standards and specifications in cooperation with other leading standards organizations, software vendors, and users. OGF is funded through its Organizational Members, including technology companies and academic and government research institutions. OGF hosts several events each year to further develop grid-related specifications and use cases and to share best practices.

The Open Grid Forum accelerates grid adoption to enable business value and scientific discovery by providing an open forum for grid innovation and developing open standards for grid software interoperability.

Even now, given that Grid computing is still in its infancy, there is

an extraordinary level agreement on core technologies. Essentially all major Grid projects are being built on protocols and services provided by the Globus Toolkit, an open-source infrastructure that provides many of the basic services needed to construct Grid applications, such as security, resource discovery, resource management and data access.

1.3 Grid blocks

The Grid architecture identifies the fundamental components of the Grid, describes their purpose and function, and indicates how these components should interact with one another.

The Grid depends on underling hardware , from the computers and communications networks that underlie the Grid to the software for doing all sorts of complex calculations that will run on the Grid. Of all these components, though, the essence of the Grid - what really makes the whole thing possible - is the software that enables the user to access computers distributed over the network. This software is called "middleware", because it is distinct from the operating systems software that makes the computers run and also different from the applications software that solves a particular problem for a user (a weather forecasting programme, for example). The middleware is conceptually in between these two types of software - hence its name.

The objective of the middleware is to get the applications to run on the right computers, wherever they may be on the Grid, in an efficient and reliable way. More generally speaking, the middleware's task is to organize and integrate the disparate computational resources of the Grid into a coherent whole.

The development of middleware is the main purpose of many of the Grid research and development projects currently underway around the globe. Grid middleware is already enabling working prototype Grids, which are often referred to as testbeds, because they are mainly being used for demonstration purposes rather than as a reliable resource

1.3.1 Grid architecture

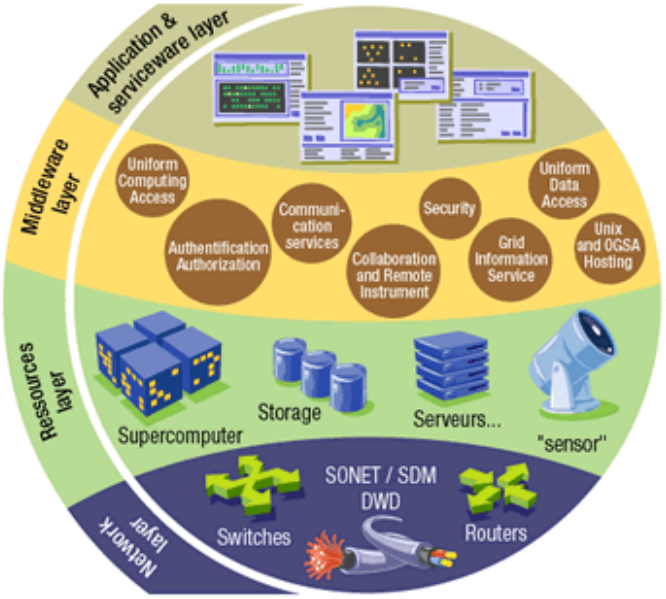

Figure 1.2: Grid layer

The architecture of the Grid is often described in terms of "layers", each providing a specific function. In general, the higher layers are focussed on the user (user-centric, in the jargon), whereas the lower layers are more focussed on computers and networks (hardware-centric).

The architecture of the Grid is often described in terms of "layers", each providing a specific function. In general, the higher layers are focussed on the user (user-centric, in the jargon), whereas the lower layers are more focussed on computers and networks (hardware-centric).

At the base of everything, the bottom layer is the network, which assures the connectivity for the resources in the Grid. On top of it lies the resource layer, made up of the actual resources that are part of the Grid, such as computers, storage systems, electronic data catalogues, and even sensors such as telescopes or other instruments, which can be connected directly to the network.

The middleware layer provides the tools that enable the various elements (servers, storage, networks, etc.) to participate in a unified Grid environment. The middleware layer can be thought of as the intelligence that brings

the various elements together - the "brain" of the Grid.

The highest layer of the structure is the application layer, which includes all different user applications (science, engineering, business, financial), portals and development toolkits supporting the applications. This is the layer that users of the grid will "see".

In most common Grid architectures, the application layer also provides the so-called serviceware, the sort of general management functions such as measuring the amount a particular user employs the Grid, billing for this use (assuming a commercial model), and generally keeping accounts of who is providing resources and who is using them - an important activity when sharing the resources of a variety of institutions amongst large numbers of different users. (The serviceware is in the top layer, because it is something the user actually interacts with, whereas the middleware is a "hidden" layer that the user should not have to worry about.)

There are other ways to describe this layered structure. For example, experts like to use the term fabric for all the physical infrastructure of the Grid, including computers and the communication network. Within the middleware layer, distinctions can be made between a layer of resource and connectivity protocols, and a higher layer of collective services.

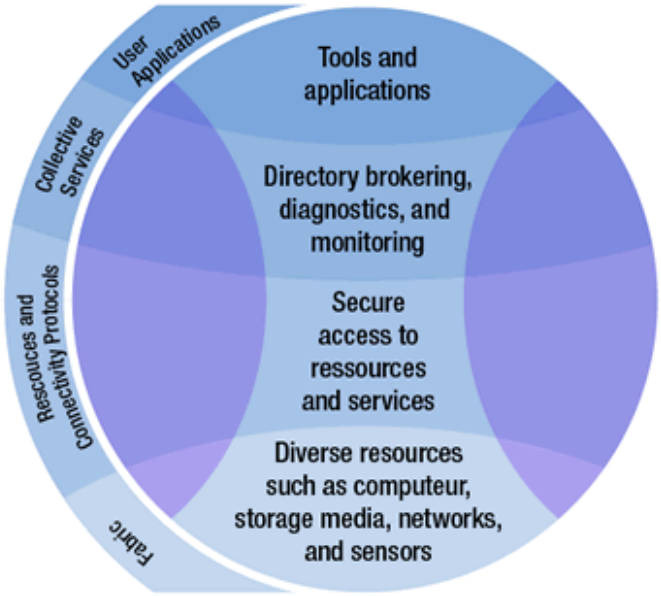

Figure 1.3: Grid fabric

Resource and connectivity protocols handle all "Grid specific" network transactions between different computers and other resources on the Grid. Remember that the network used by the Grid is the Internet, the same network used by the Web and by many other services such as e-mail. A myriad of transactions is going on at any instant on the Internet, and computers that are actively contributing to the Grid have to be able to recognize those messages that are relevant to them, and filter out the rest. This is done with communication protocols, which let the resources speak to each other, enabling exchange of data, and authentication protocols, which provide secure mechanisms for verifying the identity of both users and resources.

The collective services are also based on protocols: information protocols, which obtain information about the structure and state of the resources on the Grid, and management protocols which negotiate access to resources in a uniform way. The services include:

- keeping directories of available resources updated at all times,
- brokering resources (which like stock broking, is about negotiating between those who want to "buy" resources and those who want to "sell")
- monitoring and diagnosing problems on the Grid
- replicating key data so that multiple copies are available at different locations for ease of use
- providing membership/policy services for keeping track on the Grid of who is allowed to do what, when.

In all schemes, the topmost layer is the applications layer. Applications rely on all the other layers below them in order to run on the Grid. To take a fairly concrete example, consider a user application that needs to analyze data contained in several independent files. It will have to:

- obtain the necessary authentication credentials to open the files (resource and connectivity protocols)

- query an information system and replica catalogue to determine where copies of the files in question can currently be found on the Grid, as well as where computational resources to do the data analysis are most conveniently located (collective services)

- submit requests to the fabric - the appropriate computers, storage systems, and networks - to extract the data, initiate computations, and provide the results (resource and connectivity protocols)

- monitor the progress of the various computations and data transfers, notifying the user when the analysis is complete, and detecting and responding to failure conditions (collective services).

In order to do all of the above, it is clear that an application that a user may have written to run on a stand-alone PC will have to be adapted in order to invoke all the right services and use all the right protocols. Just like the "webifying" of applications - where users have to adapt a stand-alone application to run on a web browser, so too the Grid will require users to invest some effort into "gridifying" their applications. So there is no free lunch, not even on the Grid!

However, once gridified, thousands of people will be able to use the same application and run it trouble-free on the Grid using the middleware layers to adapt in a seamless way to the changing circumstances of the fabric.

1.3.2 Underlying hardware

Networks link together all resources belonging to the Grid, located in the different institutions around the world, and allow them to be handled as a single, huge machine.

There are different kinds of networks available these days, characterized by their size (local, national and international) and performance in terms of "throughput", the amount of data transferred from one place to another in a specific amount of time. Typically, throughput is measured in kbps (kilo bits per second), Mbps (M for mega, a million) or Gbps (G for giga, a billion).

Taking advantage of ultra-fast networks is one of the Big Ideas of the Grid, which distinguishes it from previous generations of distributed computing. Such networks allow the use of globally distributed resources in an integrated and data-intensive fashion, and ultimately may let the Grid support parallel applications, which require a lot of communication between processors, even in cases where those processors are physically quite far apart, by reducing signal latency (the delay that builds up as data are transmitted over the Internet) to a minimum.

At present, Grid testbeds are built on high-performance networks, such as the intra-European GEANT network or the UK SuperJanet network, which exhibit roughly 10Gbps performance on the network "backbone". The term backbone is commonly used for the highest speed links in the network which link major "nodes" - major resources on the Grid such as national computing centres.

The next level down from the network backbone is the network links joining individual institutions to nodes on the backbone. Performance of these is typically about 1Gbps. A further level down is the 10 to 100Mbps desktop-to-institution network links.

As well as the speed of the network, the power of the Grid is also determined by performance of the computing resources available at nodes on the network. The major nodes will be high-performance computing resources such as large clusters of computers or even dedicated supercomputers.

To have an idea of what "high-performance" means, consider that an ordinary PC in 2003 was rated at a few Gigaflop/sec. A flop is a floating point operation, which is a basic computational operation - like adding two numbers together - used to characterize computational speed. A Gigaflop is therefore a billion flops.

In 1989, the world fastest supercomputer, called ACPMAPS, could manage 50 Gigaflops. By the summer of 2003, the fastest PC's on the market (MACG4) could do better then this.

The Japanese NEC Earth Simulator machine, reckoned to be the most powerful non-military computer in the world, has already been used for large-scale climate modelling, reaches 40 teraflops/sec, in other words about

1000x the fastest PC. It has 640 eight-processor nodes and offers 10 terabytes of memory and 700 terabytes of disk space. The HPC2500 Fujitsu new massively parallel scalar supercomputer, with its 16 384 processors, reaches 85 teraflops/sec peak performance.

At the other end of the computational scale, wireless connectivity to the Grid will enhance the performances of wired networks, allowing the integration into the Grid of smaller and smaller devices, such as PDAs (Personal Digital Assistant), mobile phones and even, perhaps, some "embedded processors", the sort of processors that takes care of our car engine these days. Although the processing power and storage capacity of such processors is modest, the sheer number of them means that their total impact on Grid performance could one day be very significant (PC precessors represent only 2% of all processors in the world, illustrating the numerical importance of embedded processors).

1.3.3 Middleware

Key to success of Grid computing is the development of the "middleware", the software that organizes and integrates the disparate computational facilities belonging to a Grid. Its main role is to automate all the "machine to machine" (M2M) negotiations required to interlace the computing and storage resources and the network into a single, seamless computational "fabric".

A key ingredient for the middleware is metadata. This is essentially "data about data". Metadata play a crucial role as they contain all information about, for example, how, when and by whom a particular set of data was collected, how the data is formatted, and where in the world it is stored - sometimes at several locations.

The middleware is made of many software programmes. For one single Grid project, the European Data Grid project, over 300'000 lines of computer code have been written by some 150 software engineers, which gives a sense of the scale of the endeavour.

Some of these programmes act as "agents" and others as "brokers", bargaining the exchange of resources automatically on behalf of Grid users and

Grid resource providers.

Individual agents continuously present metadata about users, data and resources. Brokers undertake the M2M negotiations required for user authentication and authorization and then strike 'deals' for the access to, and payment for, specific data and resources. When the deal is set, a broker schedules the computational activities and oversees the data transfers required for the particular task to be undertaken. At the same time, special network 'housekeeping' agents optimize network routings and monitor the quality of service.

And of course, all this occurs in a fraction of the time that it would take humans sitting at computer terminals to do the same thing manually.

1.3.4 Globus toolkit

Practically all major Grid projects are being built on protocols and services provided by the Globus Toolkit, a software "work-in-progress" which is being developed by the Globus Alliance, which involves primarily Ian Foster's team at Argonne National Laboratory and Carl Kesselman's team at the University of Southern California in Los Angeles.

The toolkit provides a set of software tools to implement the basic services and capabilities required to construct a computational Grid, such as security, resource location, resource management, and communications.

Globus includes programs such as:

- GRAM (Globus Resource Allocation Manager), which figures out how to convert a request for resources into commands that local computers can understand

- GSI (Grid Security Infrastructure), which provides authentication of the user and works out that person's access rights

- MDS (Monitoring and Discovery Service) to collect information about resource (processing capacity, bandwidth capacity, type of storage, etc)

- GRIS (Grid Resource Information Service) to query resources for their current configuration, capabilities, and status
- GIIS (Grid Index Information Service) which coordinates arbitrary GRIS services
- GridFTP which provides a high-performance, secure and robust data transfer mechanism
- The Replica Catalog, a catalog that allows other Globus tools to look up where on the Grid other replicas of a given dataset can be found
- The Replica Management system, which ties together the Replica Catalog and GridFTP technologies, allowing applications to create and manage replicas of large datasets.

Many of the protocols and functions defined by the Globus Toolkit are similar to protocols that exist in networking and storage today, albeit optimized for Grid-specific deployments.

There are two main reasons for the strength and popularity of the Globus toolkit:

1. The Grid will have to support a wide variety of applications that have been created according to different programming paradigms. Rather than providing a uniform programming model for Grid applications, the Globus Toolkit has an "object-oriented approach", providing a bag of services from which developers of specific applications can choose what best suits them to meet their own particular needs. The tools can be introduced one at a time into existing software programs to make them increasingly "Grid-enabled". For example, an application can exploit Globus features mentioned above such as GRAM for resource management or GRIS for information services, without having to necessarily use the Globus security or replica management systems.
2. Like the WWW and the Linux operating system, the creators of the Globus Toolkit are making the software available under an "open-

source" licensing agreement. This allows others to use the software freely, and add any improvement they make to it.

1.3.5 Testbeds

What developers call "testbeds" are dedicated Grids, which are implemented and deployed to test middleware and applications developments. They are "real Grids", whose limit is mainly the restricted access, limited to small groups of developers and scientists during limited periods of time.

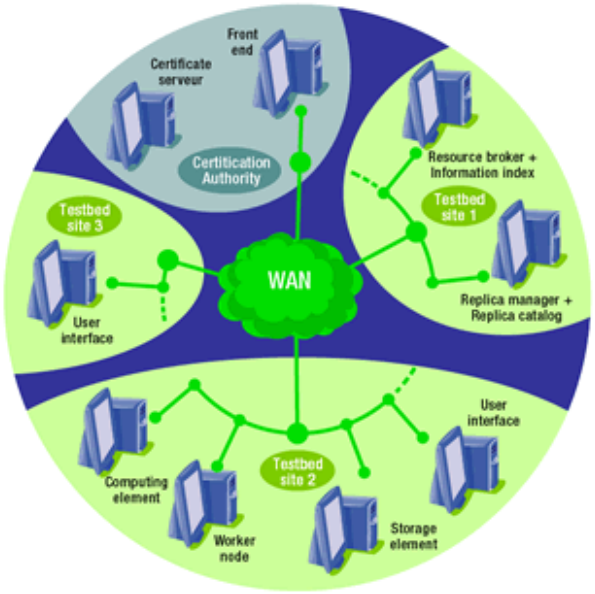

Figure 1.4: Grid WAN

A testbed is made up of one or more nodes - computer centres contributing resources to the testbed. Each node contains a certain number of computers, which may be playing different roles.

There are many testbeds either running or under construction around the world. But to provide a concrete example, we describe here the testbed of the EU EGEE projects. This testbed consisted of approximately 250 resource centres world-wide, providing some 40.000 CPUs and several Petabytes of storage. The machines linked on this testbed played one (or more, if possible) of the following different roles:

- Resource Broker, the module that receives users' requests and queries

the Information Index to find suitable resources.

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

Chapter 2

From EGEE to EGI project

2.1 EGEE project

Figure 2.1: EGEE project

The Enabling Grids for E-sciencE (EGEE) project, in its three phases: EGEE-I, -II and -III (2004-2010), was funded by the European Commission with the aim to build, on recent advances in grid technology, a service grid infrastructure which was available to scientists 24 hours-a-day.

The EGEE project officially ended on April 30 2010 but a new organisa-

tion (EGI.eu) has been created to continue the coordination and evolution of the European Grid Infrastructure (EGI) with the EGEE Grid forming the foundation.

This session will describe the EGEE project. Even if recently ended, EGEE has represented the main Academic Grid European Infrastructure. EGI (described in more details in the next session) still used the same middleware and computing infrastructure of EGEE.

The EGEE project has provided researchers in academia and business with access to a production level Grid infrastructure, independent of their geographic location.

The project, attracting a wide range of new users to the Grid, was primarily concentrated on three core areas:

- Build a consistent, robust and secure Grid network
- Continuously improve and maintain the middleware in order to deliver a reliable service to users
- Attract new users from industry as well as science and ensure they receive the high standard of training and support they need

Expanding from originally two scientific fields, high energy physics and life sciences, EGEE has integrated applications from many other scientific fields, ranging from geology to computational chemistry. Generally, the EGEE Grid infrastructure was ideal for any scientific research especially where the time and resources needed for running the applications are considered impractical when using traditional IT infrastructures.

2.1.1 Results

The Enabling Grids for E-sciencE (EGEE) project was the flagship Grid infrastructure project of the EU. The third and last two-year phase of the project (started on 1 May 2008) was financed with a total budget of cca. 47,150,000 euro, with a further estimated 50,000,000 euro worth of computing resources contributed by the partners. A total manpower of 9,010

Person Months, of which over 4,500 Person Months contributed by the partners from their own funding sources.

The EGEE's results can be summarized below:

- A Grid infrastructure spanning about 250 sites across 50 countries
- An infrastructure of more than 68,000 CPU available to users 24 hours a day, 7 days a week,
- More than 20 Petabytes (20 million Gigabytes) of storage.
- Sustained and regular workloads of 150K jobs/day, reaching up to 188K jobs/day
- Massive data transfers $>$ 1.5 GB/s
- User Support including:
    1. A single access point for support, a portal with well structured information and updated documentation;
    2. knowledgeable experts;
    3. correct, complete and responsive support
    4. tools to help resolve problems.
- Security and Policy, including:
    1. Authentication (Use of GSI, X.509 certificates generally issued by national certification authorities)
    2. Agreed network of trust (International Grid Trust Federation (IGTF), EUGridPMA, APGridPMA, TAGPMA)
    3. All EGEE sites will usually trust all IGTF root CAs

Having such resources available changes the way scientific research takes place. The end use depends on the users' needs: large storage capacity, the bandwidth that the infrastructure provides, or the sheer computing power available.

The EGEE Grid was built on the EU Research Network GÉANT and exploited Grid expertise generated by many EU, national and international Grid projects to date.

2.1.2 Beneficiaries

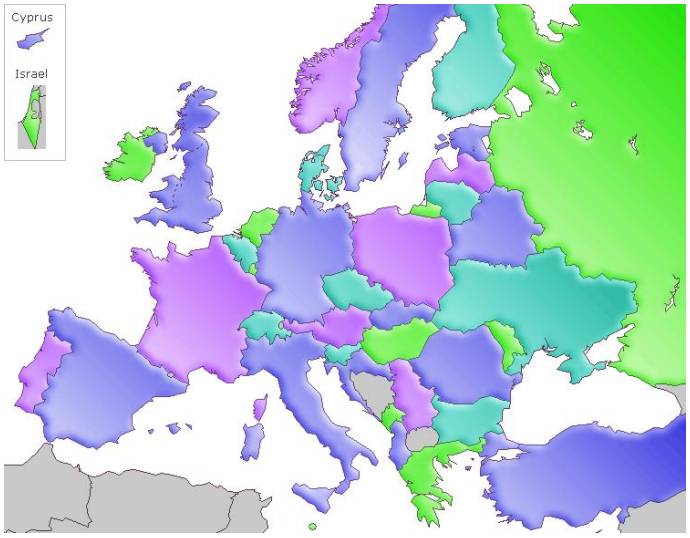

Figure 2.2: Countries involved in EGEE

The EGEE consortium consisted of 42 beneficiaries, both academic and business. All EC co-funded countries have grouped their academic partners on a national level via Joint Research Units or National Grid Initiatives so that the 42 beneficiaries represent a total of more than 120 partners. This has a structuring effect on the Grid communities across the European Research Area and is an important milestone for the planning of a sustainable Grid Infrastructure model. Beneficiaries are organized in regional federations, covering:

- Asia Pacific (Australia, Japan, Korea, Taiwan)
- Benelux (Belgium, the Netherlands)
- Central Europe (Austria, Croatia, Czech Republic, Hungary, Poland, Slovakia, Slovenia)
- France

- Germany/Switzerland
- Italy
- Nordic countries (Finland, Sweden, Norway)
- South East Europe (Bulgaria, Cyprus, Greece, Israel, Romania, Serbia, Turkey)
- South West Europe (Portugal, Spain)
- Russia
- United Kingdom/Ireland
- USA

Collaboration with additional countries in the Asia Pacific region (China, Brunei, Indonesia, Malaysia, Philippines, Singapore, Thailand, Vietnam) and in the Commonwealth of Independent States (Armenia, Ukraine, Uzbekistan) is also foreseen.

2.1.3 Infrastructure

The EGEE grid infrastructure consisted of a set of middleware services deployed on a worldwide collection of computational and storage resources, plus the services and support structures put in place to operate them:

- **The Production Service** infrastructure is a large multi-science Grid infrastructure, federating some 250 resource centres world-wide, providing some 40.000 CPUs and several Petabytes of storage. This infrastructure is used on a daily basis by several thousands of scientists federated in over 200 Virtual Organizations on a daily basis. This is a stable, well-supported infrastructure, running the latest released versions of the gLite middleware.
- **The Pre-Production Service (PPS)** provided access to grid services in preview to interested users, in order to test, evaluate and give feedback to changes and new features of the middleware. In addition

to that, the pre-production extended the middleware certification activity, helping to evaluate deployment procedures, [inter]operability and basic functionality of the software against operational scenarios reflecting real production conditions.

- The EGEE **Network Operations Centre (ENOC)** which catered for the network operational coordination between EGEE and the network providers (GEANT2 /NRENs). This is complemented by the training infrastructure and the certification test-beds as well as the needed support structures and policy groups.

2.1.4 Middleware

Middleware is a crucial component of any Grid infrastructure as it provides the 'glue' to link the hardware resources within the Grid. The gLite middleware binds the EGEE and EGI resources into a single infrastructure to provide seamless access for the project's user communities.

The EGEE infrastructure is based on a Grid Middleware stack called gLite, which is integrated, certified and distributed by the project itself.

Figure 2.3: Glite Middleware

A large fraction of the services included in the gLite distribution are maintained and further enhanced by the Middleware Engineering Activity, whose goal is to provide a reference open source implementation of selected Grid services satisfying the requirements of both users and administrators, in terms of functionality, performance and manageability.

The available services in the gLite distribution can be broadly classified in two categories:

- Grid Foundation Middleware, covering the security infrastructure, information, monitoring and accounting systems, access to computing and storage resources, providing the basis for a consistent and dependable production infrastructure;

- Higher-level Grid Middleware, including services for job management, data catalogs and data replication, providing applications with end-to-end solutions. In order to favour interoperability with other Grid infrastructures, the interfaces of the services are, wherever possible, compliant with established standards, primarily defined by the Open Grid Forum. With its experience in developing production strength services, EGEE was also committed to contribute to the standardization process through the OGF-Europe project.

2.1.5 EGEE Activities

The work being carried out within EGEE is organised into eleven "activities", which come under three main areas:

- Networking Activities (NA) which are the management and coordination of all the commmunication aspects of the project

- Specific Service Activities (SA) are the support, operation and management of the Grid as well as the provision of network resources

- Research Activities (JRA) concentrate on Grid research and development

2.1.6 Networking support

In EGEE-III, the objective of the Networking Support Activity (EGEE-III SA2) was to play a key role in:

- The networking related activities in collaboration with other EGEE activities like NA4, JRA1, SA1 and SA3 (ETICS - eInfrastructure for Testing, Integration and Configuration of Software);

- The network operation centre (ENOC) integrated with EGEE-III GGUS user support system;
- The management of the relationships between EGEE-III and network providers GEANT2/NRENs (National research and Education networks)) with the strengthening of links between the "Grid" people and the "networks/GEANT2/NRENs";
- Fostering the usage of advanced network services especially with the automation (to the extent possible) of the process for network services provisioning to EGEE-III (using AMPS - Advance Multi-domain Provisioning System);
- The introduction of new services provided by the European research networking community to EGEE-III, such as the creation of an hybrid network not only based on IP services;
- Collaboration on network-related subjects with other projects;
- Enabling the Grid to be ready for IPv6 : gLite tests (JRA1 ETICS), validation process (SA3);
- Network services and operational interfaces (LCG Large hadron collider Computing Grid / LHCOPN - Large Hadron Collider Optical private network). To achieve these goals the activity was divided into several subtasks shared among involved partners:

EGEE Networking Activities (NA)

EGEE Networking Activities (NA)

28 per cent of the funding is going towards the Networking Activities which are divided into five different areas:

- Networking Activity 1 (NA1): the overall management of the project.
- Networking Activity 2 (NA2): Information Dissemination and Outreach and includes tasks such as running the external website, organising conferences and managing the distribution of publications.

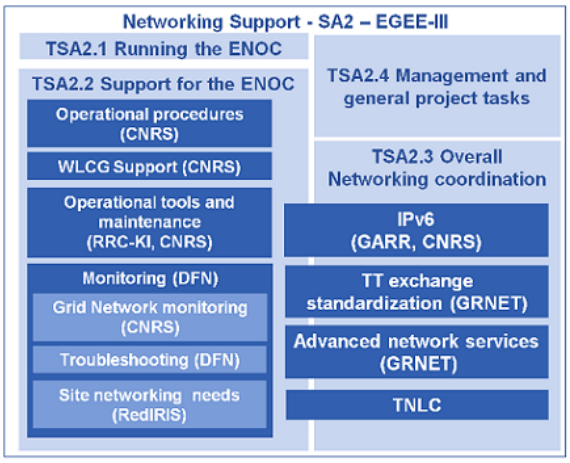

Figure 2.4: Network support in EGEE SA2

- Networking Activity 3 (NA3): User Training and Induction and includes tasks such as organising on-site training and producing training and course material.

- Networking Activity 4 (NA4): Application Identification and Support and includes tasks such as supporting pilot applications and identifying new users.

- Networking Activity 5 (NA5): Policy and International Cooperation and includes tasks such as liaising with parties interested in the EGEE project on an international level.

EGEE Specific Service Activities (SA)

48 per cent of the funding went towards the Specific Service Activities which are divided into two different areas:

- Specific Service Activity 1 (SA1): European Grid Support, Operation and Management and includes tasks such as grid monitoring and control and resource and user support.

- Specific Service Activity 2 (SA2): Network Resource Provision and includes tasks such as policies and service level agreements.

EGEE Joint Research Activities (JRA)

24 per cent of the funding went going toward the Joint Research Activities which are divided into four different areas:

- Joint Research Activity 1 (JRA1): Middleware Re-engineering and Integration and includes tasks such as re-engineering existing middleware, integrating middleware, testing and validation.
- Joint Research Activity 2 (JRA2): Quality Assurance and includes tasks such as ensuring that processes, products and operation services conform to project requirements, standards and procedures.
- Joint Research Activity 3 (JRA3): Security and includes tasks such as developing security frameworks and policies and designing security mechanisms.
- Joint Research Activity 4 (JRA4): Network Service Development and includes tasks such as developing interfaces to the network and advance reservations of network connectivity in terms of bandwidth, duration and quality of service.

2.1.7 Applications on EGEE

The Enabling Grids for E-sciencE (EGEE) project began by working with two scientific groups, High Energy Physics (HEP) and Biomedicine, but as it has progressed into its second phase it has grown to support research domains in areas as diverse as multimedia, astrophysics, archaeology, and computational chemistry. Researchers form Virtual Organisations (VOs), allowing them to collaborate, to share resources, and to access common datasets via the EGEE grid infrastructure. Below is an overview of the application domains currently supported by EGEE.

- High Energy Physics (HEP) applications

  The HEP community was one of the two pilot user domains for EGEE and remains a major user of the infrastructure, providing vital input

that allowed EGEE and nowadays EGI to ensure it provides a user-orientated service. The original EGEE HEP community was formed from the experiments of the Large Hadron Collider (LHC), currently under construction at CERN (European Organization for Nuclear Research) near Geneva, Switzerland. These four experiments, ALICE, ATLAS, CMS, and LHCb, are estimated to produce some 15 petabytes per year when the collider starts up 2007. These data are managed and processed using the EGEE infrastructure.

Other international HEP experiments are also making use of the EGEE infrastructure, including the BaBar (the B and B-bar experiment), CDF (Collider Detector at Fermilab) and DØ experiments using particle accelerators in the USA, and the ZEUS and H1 experiments using the HERA collider at the DESY laboratory in Germany

- Biomedical applications

  Applications in the biomedical field have been included in the EGEE project from the outset and are now exploiting the infrastructure in a sustained production mode. The biomedical community benefits from the Grid by enabling remote collaboration on shared datasets as well as carrying out high throughput calculations. The applications cover the fields of medical imaging, bioinformatics and drug discovery, with 23 individual applications deployed or being ported to the EGEE infrastructure.

  Notable among the biomedical sector applications is the WISDOM application, which has carried out a number of high profile drug discovery calculations. These verify the EGEE infrastructure's ability to perform large, complex tasks and its usefulness as a tool in the fight against diseases such as malaria and avian flu.

  Another important project is DECIDE. The aim of DECIDE (Diagnostic Enhancement of Confidence by an International Distributed Environment) is to design, implement, and validate a GRID-based e-Infrastructure building upon neuGRID and relying on the Pan-European backbone GEANT and the NRENs. Over this e-Infrastructure,

a service will be provided for the computer-aided extraction of diagnostic markers for Alzheimer's disease and schizophrenia from medical images.

Project activities will start on 1 September 2010.

- Astro(-particle)

  Physics applications The two major VOs in this domain, Planck and MAGIC, share problems of computation involving large-scale data acquisition, simulation, data storage, and data retrieval. The Planck satellite of the European Space Agency (ESA) was launched in 2008 and aims to map the microwave sky with an unprecedented combination of sky and frequency coverage, accuracy, stability and sensitivity. The MAGIC application simulates the behaviour of air showers in the atmosphere, originated by high energetic primary cosmic rays. These simulations are needed to analyse the data of the MAGIC telescope, located in the Canary Islands, to study the origin and the properties of high energy gamma rays.

- Earth Science Research (ESR) applications

  Earth Science covers a large range of topics related to the solid earth, atmosphere, ocean and their interfaces as well as planet atmospheres and cores. Recently, members of the ESR Virtual Organisation have worked on rapid earthquake analysis, helping the scientific community to better understand these devastating natural disasters. Geophysics applications The Geophysics domain is closely related to the Earth Sciences domain and supports EGEODE (Expanding GEOsciences on DEmand), EGEE's first industrial application. EGEODE was initiated by the private company CCG (Compagnie Générale de Géophysique). It allows academic researchers to use the company's Geocluster software on the EGEE infrastructure.

- Fusion applications

  The capability of Grids for meeting the needs of the fusion community has been demonstrated. Several applications are already running on

the EGEE infrastructure: massive ray tracing to estimate the trajectory of a microwave beam in plasma; kinetic transport and optimisation of special magnetic confinement fusion devices (stellerators). Several computational tasks related to the ITER (International Thermonuclear Experimental Reactor) project were successfully ported to the EGEE infrastructure

- Computational Chemistry applications

  The main user in the field of computational chemistry is the GEMS a-priori molecular simulator. Several applications have already been ported to the Grid and have been run in production to calculate observables for chemical reactions, simulate the molecular dynamics of complex systems, and calculate the electronic structure of molecules, molecular aggregates, liquids and solids.

- Finance & Multimedia applications

  The multimedia domain is currently in testing through EGEE's GILDA Grid testbed. The financial applications involve work with the Abdus Salam International Centre for Theoretical Physics, which is implementing a national Italian Grid infrastructure for financial and economic research in the framework of the Egrid project, funded by the Italian Ministry for Education and Research.

2.2 EGI project

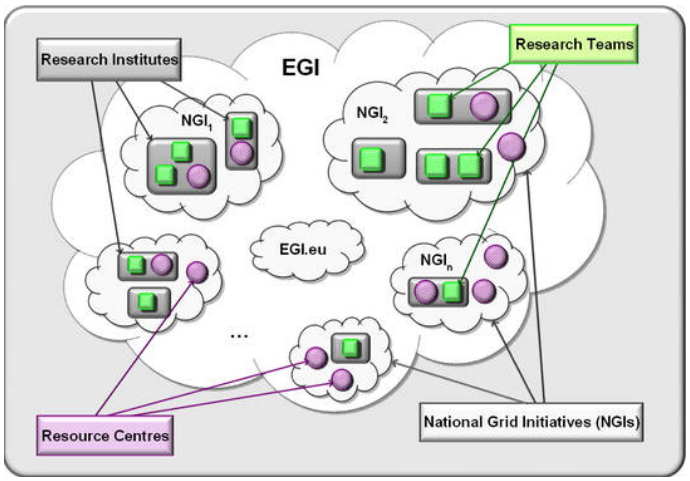

Figure 2.5: EGI

As said in the previous session, the Enabling Grids for E-sciencE project is no longer active. The distributed computing infrastructure built and nurtured by the projects DataGrid (2002-2004), EGEE-I, -II and -III (2004-2010) is now supported by the European Grid Infrastructure.

This transition is an important step in ensuring the European research community has access to a distributed computing infrastructure to maintain its leadership position in research and support its work in global collaborations for many years to come.

The new organization (EGI.eu), with its headquarter in Amsterdam, has been created to continue the coordination and evolution of the European Grid Infrastructure (EGI) with the EGEE Grid forming the foundation.

EGI is a long-term organization, not dependent on short-term funding cycles and it coordinates National Grid Initiatives, which form the country-wide building blocks of the pan-European Grid.

EGI is not a simple continuation of EGEE and other grid projects. Most existing grid infrastructure projects include in their Consortia specific national Resource Providers or Research Institutions and naturally satisfy mostly their specific requirements. In contrast, the EGI model for sustainability is built on each member state's establishment of its own NGI which

will be responsible for the provision and operation of a national grid infrastructure satisfying all the Resource Providers and Research Institutions of its country, and for representing these in the EGI Council and in the relations with EGI.org and the other NGIs. Some structuring of the national grid infrastructure efforts has begun with EGEE III with the constitution of Joint Research Units (JRUs) which need to be leveraged by EGI.

The EGI service offer is organised in a non-hierarchical NGI-based environment. It is governed by the subsidiarity principle, meaning that tasks which are efficiently fulfilled at the national or regional level should be performed at that level by the NGIs.

In its role of coordinating grid activities between European NGIs EGI.eu will:

- Operate a secure integrated production grid infrastructure that seamlessly federates resources from providers around Europe
- Coordinate the support of the research communities using the European infrastructure coordinated by EGI.eu
- Work with software providers within Europe and worldwide to provide high-quality innovative software solutions that deliver the capability required by our user communities
- Ensure the development of EGI.eu through the coordination and participation in collaborative research projects that bring innovation to European Distributed Computing Infrastructures (DCIs)

2.2.1 National Grid Initiatives

EGI is composed of a small central coordinating body (called the EGI.eu) and National Grid Initiatives (NGIs) performing the following tasks:

1. Authentication of individual users as the people they claim to be.
2. Allocation of project or discipline collaboration members to VOs where resources are shared.

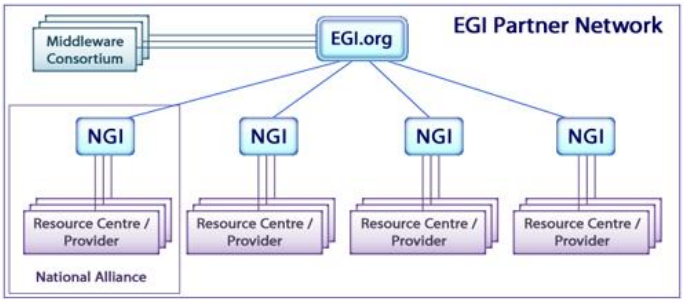

Figure 2.6: NGI

3. Allocation of computing resources to those VOs which VO members will be authorized to use.

4. Authorization of VOs to run computing jobs, store and retrieve data on individual computing resources (machines, data centres, facilities, etc.).

5. Distribution and scheduling of computing jobs, workflows, data retrieval and access requests to authorized computing resources.

6. Monitoring of the jobs submitted, processed, and the data stored by individuals.

7. Accounting of users and VO in their allocations and usage of computing resources.

8. Reporting to each NGI of their allocation of resources to VOs, and the use of those resources by individual users, in order to enable the NGI and the national funding bodies to account for the use of funds in terms of the research results produced by VOs.

9. Coordinated management of software updates and hardware upgrades while maintaining a continuous service.

The NGI in each member state needs to support these functions so that it can interact with EGI. The activities of the NGIs are not limited to the tasks that each NGI performs at national level to maintain its infrastructure, but extend to international tasks that allow the sharing of IT resources in a

robust and transparent way and the support of the international application communities. The characteristics of the NGIs can be identified as follows: Each NGI should:

- be the only recognized national body in a country with a single point-of-contact representing all institutions and research communities related to a national grid infrastructure;
- have the capacity to sign the statutes of EGI.org, either directly or through a legal entity representing it;
- have a sustainable structure, or be represented by a sustainable legal structure in order to commit to EGI.org in the long term;
- mobilise national funding and resources and be able to commit to EGI.org financially, i.e. to pay EGI.org membership fees and if there is a demand for such services in the NGI request and pay for EGI.org services;
- ensure the operation of a national e-Infrastructure to an agreed level of service and its integration into EGI;
- support user communities providing general services to the applications and fostering the grid usage for new communities;
- adhere to EGI policies and quality criteria.

Chapter 3

gLite Middleware

3.1 gLite Middleware

gLite (pronounced "gee-lite") is the middleware stack for grid computing used by the the EGEE and EGI projects and a very large variety of scientific domains. Born from the collaborative efforts of more than 80 people in 12 different academic and industrial research centers as part of the EGEE Project, gLite provides a complete set of services for building a production grid infrastructure. gLite provides a framework for building grid applications tapping into the power of distributed computing and storage resources across the Internet. The gLite services are currently adopted by more than 250 Computing Centres and used by more than 15000 researchers in Europe and around the world (Taiwan, Latin America etc.). [12]

3.2 History

After prototyping phases in 2004 and 2005, convergence with the LCG-2 distribution was reached in May 2006 when gLite 3.0 was released and became the official middleware of the EGEE project.

3.3 Middleware description

The gLite middleware itself is a complex system with interconnected parts, interacting over the network. This includes as the middleware to store

a data (Storage Element (SE)) as cluster resources (Worker Nodes, Local Resource Management System).

Every gLite instance has Computing Element as a frontend for job submission. All connections need to pass a generic interface to the cluster (Grid Gate).

Information Service (IS) or "site BDII" provides informations about the Grid resources. These informations can be used as for monitoring and accounting as to permit to the WMS/RB to find the best resource where to run grid jobs.

Many gLite implementations use Globus Monitoring & Discovery Service (MDS) for resource discovery and to publish the resource status.

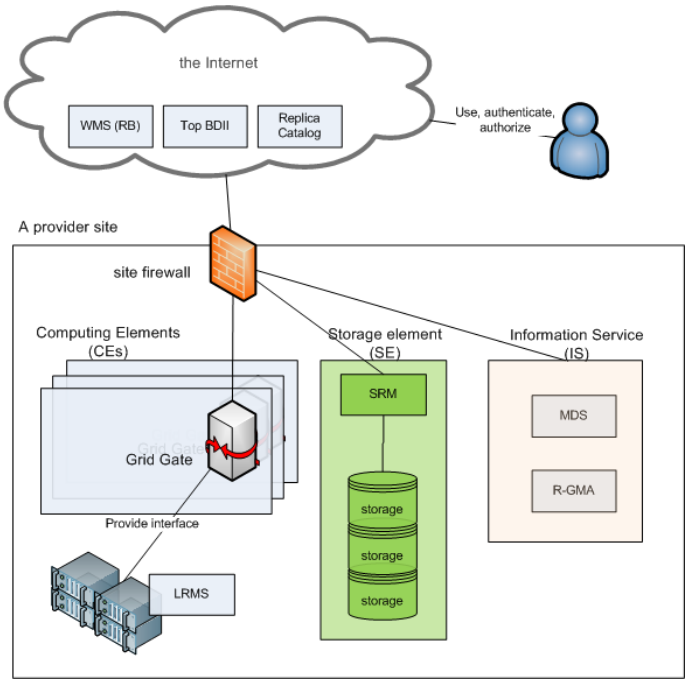

Figure 3.1: Glite Architecture

3.3.1 User Interface

The access point to the gLite Grid is the User Interface (UI). This can be any machine where users have a personal account and where their user certificate is installed. From a UI, a user can be authenticated and authorized to use the WLCG/EGEE resources, and can access the functionalities offered by

the Information, Workload and Data management systems. It provides CLI tools to perform some basic Grid operations:

- list all the resources suitable to execute a given job
- submit jobs for execution
- cancel jobs
- retrieve the output of finished jobs
- show the status of submitted jobs
- retrieve the logging and bookkeeping information of jobs
- copy, replicate and delete files from the Grid
- retrieve the status of different resources from the Information System

3.3.2 Computing element

A Computing Element (CE), in Grid terminology, is some set of computing resources localized at a site (i.e. a cluster, a computing farm). A CE includes a Grid Gate (GG) [13], which acts as a generic interface to the cluster; a Local Resource Management System (LRMS) (sometimes called batch system), and the cluster itself, a collection of Worker Nodes (WNs), the nodes where the jobs are run.

There are two GG implementations in gLite 3.1: the LCG CE, developed by EDG and used in LCG-2 , and the gLite CE, developed by EGEE. Sites can choose what to install, and some of them provide both types. The GG is responsible for accepting jobs and dispatching them for execution on the WNs via the LRMS.

In gLite 3.1 the supported LRMS types are OpenPBS/PBSPro, Platform LSF, Maui/Torque, BQS and Condor, and Sun Grid Engine. [14]

3.3.3 Storage element

A Storage Element (SE) provides uniform access to data storage resources. The Storage Element may control simple disk servers, large disk arrays or tape-based Mass Storage Systems (MSS). Most WLCG/EGEE sites provide at least one SE.

Storage Elements can support different data access protocols and interfaces. Simply speaking, GSIFTP (a GSI-secure FTP) is the protocol for whole-file transfers, while local and remote file access is performed using RFIO or gsidcap.

Most storage resources are managed by a Storage Resource Manager (SRM), a middleware service providing capabilities like transparent file migration from disk to tape, file pinning, space reservation, etc. However, different SEs may support different versions of the SRM protocol and the capabilities can vary.

There is a number of SRM implementations in use, with varying capabilities. The Disk Pool Manager (DPM) is used for fairly small SEs with disk-based storage only, while CASTOR is designed to manage large-scale MSS, with front-end disks and back-end tape storage. dCache is targeted at both MSS and large-scale disk array storage systems. Other SRM implementations are in development, and the SRM protocol specification itself is also evolving.

Classic SEs, which do not have an SRM interface, provide a simple disk-based storage model. They are in the process of being phased out.

3.3.4 Information service

The Information Service (IS) provides information about the WLCG/EGEE Grid resources and their status. This information is essential for the operation of the whole Grid, as it is via the IS that resources are discovered. The published information is also used for monitoring and accounting purposes.

Much of the data published to the IS conforms to the GLUE Schema [15], which defines a common conceptual data model to be used for Grid resource monitoring and discovery.

The Information System that is used in gLite 3.1 inherits its main concepts from the Globus Monitoring and Discovery Service (MDS) [16]. However, the GRIS and GIIS in MDS has been replaced by the Berkeley Database Information Index which is essentially an OpenLDAP server that is updated by an external process.

3.3.5 Workload management

The purpose of the Workload Management System (WMS) [17] is to accept user jobs, to assign them to the most appropriate Computing Element, to record their status and retrieve their output. The Resource Broker (RB) is the machine where the WMS services run.

Jobs to be submitted are described using the Job Description Language (JDL), which specifies, for example, which executable to run and its parameters, files to be moved to and from the Worker Node on which the job is run, input Grid files needed, and any requirements on the CE and the Worker Node.

The choice of CE to which the job is sent is made in a process called match-making, which first selects, among all available CEs, those which fulfill the requirements expressed by the user and which are close to specified input Grid files. It then chooses the CE with the highest rank, a quantity derived from the CE status information which expresses the goodness of a CE (typically a function of the numbers of running and queued jobs).

The RB locates the Grid input files specified in the job description using a service called the Data Location Interface (DLI), which provides a generic interface to a file catalogue. In this way, the Resource Broker can talk to file catalogues other than LFC (provided that they have a DLI interface).

The most recent implementation of the WMS from EGEE allows not only the submission of single jobs, but also collections of jobs (possibly with dependencies between them) in a much more efficient way then the old LCG-2 WMS, and has many other new options.

Finally, the Logging and Bookkeeping service (LB) [18] tracks jobs managed by the WMS. It collects events from many WMS components and records the status and history of the job.

3.3.6 Security

The gLite user community is grouped into Virtual Organisations (VOs) [13]. A user must join a VO supported by the infrastructure running gLite to be authenticated and authorized to using grid resources.

The Grid Security Infrastructure (GSI) in WLCG/EGEE enables secure authentication and communication over an open network [19]. GSI is based on public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol, with extensions for single sign-on and delegation.

In order to authenticate himself, a user needs to have a digital X.509 certificate issued by a Certification Authority (CA) trusted by the infrastructure running the middleware.

The authorisation of a user on a specific Grid resource can be done in two different ways. The first is simpler, and relies on the grid-mapfile mechanism. The second way relies on the Virtual Organisation Membership Service (VOMS) and the LCAS/LCMAPS mechanism, which allow for a more detailed definition of user privileges.

The gLite user community is grouped into Virtual Organisations (VOs) [13]. A user must join a VO supported by the infrastructure running gLite to be authenticated and authorized to using grid resources.

The Grid Security Infrastructure (GSI) in WLCG/EGEE enables secure authentication and communication over an open network [19]. GSI is based on public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol, with extensions for single sign-on and delegation.

In order to authenticate himself, a user needs to have a digital X.509 certificate issued by a Certification Authority (CA) trusted by the infrastructure running the middleware.

The authorisation of a user on a specific Grid resource can be done in two different ways. The first is simpler, and relies on the grid-mapfile mechanism. The second way relies on the Virtual Organisation Membership Service (VOMS) and the LCAS/LCMAPS mechanism, which allow for a more detailed definition of user privileges.

3.4 gLite job submission chain scheme

This section briefly describes what happens when a user submits a job to the WLCG/EGEE/EGI Grid to process some data, and explains how the different components interact.

The following figure illustrates the process that takes place when a job is submitted to the Grid. The individual steps are as follows:

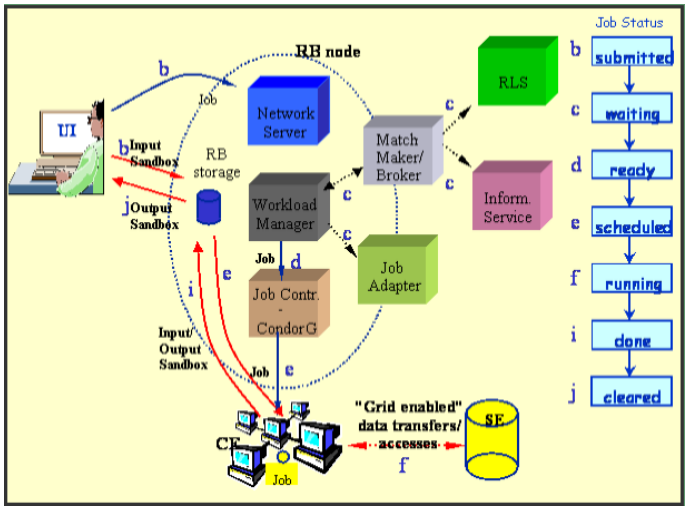

Figure 3.2: Glite Job Flow

1. After obtaining a digital certificate from a trusted Certification Authority, registering in a VO and obtaining an account on a User Interface, the user is ready to use the WLCG/EGEE Grid. He logs in to the UI and creates a proxy certificate to authenticate himself in subsequent secure interactions.

2. The user submits a job from the UI to the gLite WMS. In the job description one or more files to be copied from the UI to the WN can be specified, and these are initially copied to the gLite WMS. This set of files is called the Input Sandbox. An event is logged in the LB and the status of the job is SUBMITTED.

3. The WMS looks for the best available CE to execute the job. To do so, it interrogates the Information Supermarket (ISM), an internal cache

of information which in the current system is read from the BDII, to determine the status of computational and storage resources, and the File Catalogue to find the location of any required input files. Another event is logged in the LB and the status of the job is WAITING.

4. The gLite WMS prepares the job for submission, creating a wrapper script that will be passed, together with other parameters, to the selected CE. An event is logged in the LB and the status of the job is READY.

5. The CE receives the request and sends the job for execution to the local LRMS. An event is logged in the LB and the status of the job is SCHEDULED.

6. The LRMS handles the execution of jobs on the local Worker Nodes. The Input Sandbox files are copied from the gLite WMS to an available WN where the job is executed. An event is logged in the LB and the status of the job is RUNNING.

7. While the job runs, Grid files can be directly accessed from a SE or after copying them to the local filesystem on the WN with the Data Management tools.

8. The job can produce new output files which can be uploaded to the Grid and made available for other Grid users to use. This can be achieved using the Data Management tools described later. Uploading a file to the Grid means copying it to a Storage Element and registering it in a file catalogue.

9. If the job ends without errors, the output (not large data files, but just small output files specified by the user in the so called Output Sandbox) is transferred back to the gLite WMS node. An event is logged in the LB and the status of the job is DONE.

10. At this point, the user can retrieve the output of his job to the UI. An event is logged in the LB and the status of the job is CLEARED

11. Queries for the job status can be addressed to the LB from the UI. Also, from the UI it is possible to query the BDII for the status of the resources.

12. If the site to which the job is sent is unable to accept or run it, the job may be automatically resubmitted to another CE that satisfies the user requirements. After a maximum allowed number of resubmissions is reached, the job will be marked as aborted. Users can get information about the history of a job by querying the LB service.

Chapter 4

Network Activity in EGEE and EGI

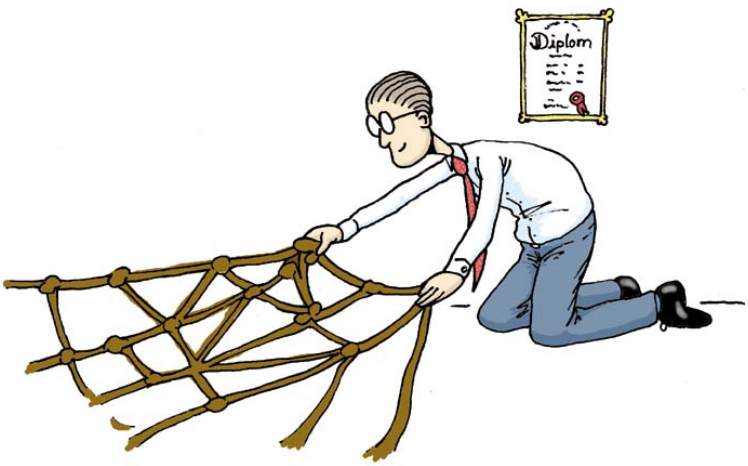

Figure 4.1: Networking

Whatever you do on a grid you need to use the network.

Availability and performance are therefore crucial for grid projects and demands on the network depend on the type of grid applications. This is why EGI, the major European Grid Initiative, and its predecessor project EGEE, both had a network activity task. The network activity provides an interface between European Grid Infrastructure and network providers.

In EGEE the network support activity of the project, SA2, was in charge of dealing with everything related to networks (troubles, operations, monitoring, etc.).

4.1 EGEE SA2 Network Activity

The purpose of the Network Resource Provision activity was to ensure the Enabling Grids for E-science in Europe (EGEE) project access to the appropriate networking services provided by GEANT and the National Research and Education Networks across Europe. The relationship between the project and the network providers was managed by the Network Resource Provision team, via a formal body called the Technical Network Liaison Board. The Network Resource Provision team ensured that all user requirements were met in terms of network capacity and service class. One of the main objectives was to ensure the provisioning of a high bandwidth network offering guaranteed performance and virtual private network capability to the end users. The Network Resource Provision team performed aggregate modelling, derived the Service Level specifications for network provision, created Service Level Agreements with the network providers and monitor the Service Level Agreements against demand (aggregate traffic) and supply (network performance).

The EGEE project used the European research networks to connect the providers of computing, storage, instrumentation and applications resources with users in Grid Virtual Organisations. This process was overseen by SA2, dealing with all the issues related to the network infrastructure that underlies the EGEE Grid, both those arising within the project and those between the project and outside groups and organisations. The latter consisted of relationships with the other project activities on network issues (for instance applications requirements with NA4, network monitoring with SA1). Moreover, SA2 also took care of the relations with related projects (see Related Projects information sheet) concerning network cross-activities, such as collaboration with the EUChinaGrid project on IPv6 compliance of EGEE's gLite middleware. SA2 acted as the interface between EGEE and the network infrastructure. This role was two-fold: first, SA2 acted as a technical interface to build and manage the collaboration with the network providers. The Technical Network Liaison Committee (TNLC) was one of the places where the exchanges between EGEE and the networking community took

place. SA2 pushed for the adoption of network Service Level Agreements (SLAs) in both the Grid and the network community, to provide Grid users with the best network service they could expect from today's network. Second, through the EGEE Network Operations Centre (ENOC), SA2 acted as a day-to-day operational interface between EGEE and the underlying network providers. The ENOC, as an end-to-end coordination unit, was the unique point of contact for all the network related operational issues between EGEE and the pan-European network GÉANT2. It was the interlocutor for GÉANT2/NRENs to contact EGEE about network troubles and interface with EGEE's network support unit. As such, the ENOC was also responsible of the operations of the LCG optical private network. SA2 is built on the experience acquired by the three main partners (CNRS, GRNET and RRC-KI) during the first phase of EGEE, and now there was extended to some of the National Research and Education Networks (DFN, GARR) involved in EGEE. The networking community was further represented in EGEE through the participation of DANTE, which is owned by a consortium of the European NRENs.

In EGEE-III, the objective of the Networking Support Activity (EGEE-III SA2) was to play a key role in:

- The networking related activities in collaboration with other EGEE activities like NA4, JRA1, SA1 and SA3 (ETICS - eInfrastructure for Testing, Integration and Configuration of Software);
- The network operation centre (ENOC) integrated with EGEE-III GGUS user support system;
- The management of the relationships between EGEE-III and network providers GEANT2/NRENs (National research and Education networks)) with the strengthening of the links between the "Grid" people and the "networks/GEANT2/NRENs";
- Fostering the usage of advanced network services especially with the automation (to the extent possible) of the process for network services provisioning to EGEE-III (using AMPS - Advance Multi-domain Provisioning System);

- The introduction of new services provided by the European research networking community to EGEE-III, such as the creation of an hybrid optical IP network not only based on (IP services); Collaboration on network-related subjects with other projects;

- Enabling the Grid to be ready for IPv6 : gLite tests (JRA1 involving ETICS), validation process (SA3); Network services and operational interfaces (LCG Large hadron collider Computing Grid / LHCOPN - Large Hadron Collider Optical private network).

To achieve these goals the activity was divided into several subtasks shared among involved partners:

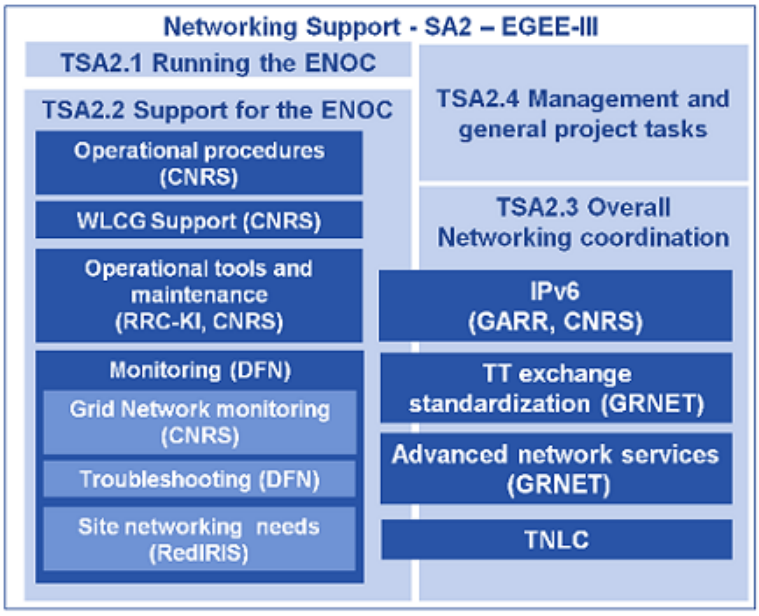

Figure 4.2: Network Support in EGEE SA2

4.2 EGI and Network Activity

Within EGI project, GARR, the Italian National Research and Education Network, has been assigned the global task (O-E-12) for the coordination of network support.

4.2.1 Role of GARR, the Italian National Research and Education Network

GARR has been involved in EGEE from the very beginning. In particular, in EGEE II GARR has directly contributed to the coding and the standardization work related to XML schema of the exchanged Network trouble tickets and the exploitation of the gLite middleware using IPv6. In EGEE III GARR was involved in the IPv6 task, coordinating the testing of gLite using IPv6 and the IPv6 support enforcement activities (tutorials, etc.) and further testing the IPv6 compliance of gLite components, when they were made available. GARR was also involved in the task on monitoring for the grid, contributing to the set up of the grid-jobs based prototype, beta testing it and continuously providing feedback to the core team of developers. GARR has a long experience in managing and monitoring high capacity networks (dating back to the early 90's). In addition, GARR belongs to the international network of the NRENs worldwide and works in close collaboration with GEANT and DANTE. GARR is also highly involved in the prototypal deployment of the GEANT3 PerfSONAR monitoring suite for the LHCOPN network.

The foreseen activities for the EGI project are the following ones:

- Initial assessment of the network support within individual NGIs belonging to EGI
- Gathering of expectations, available manpower information, familiar useful tools currently in place for Network monitoring and troubleshooting, from the individual NGIs
- Follow up of the development , consolidation to full quality and reliability and deployment - (previous general consensus within the NGI community) - of the prototypal tools for Network Monitoring and troubleshooting developed within the EGEE III SA2 (network support) activity, namely the PerfSONAR-Lite TSS and the approach to Network Monitoring based on Grid jobs.

- Possible further exploitation of additional tools for network monitoring and troubleshooting serving the NGIs and NRENs communities
- Define, jointly with the VRC/VO user community, a subset of the Grid sites belonging to the EGI global infrastructure made up by the NGIs, to be periodically monitored on a scheduling basis
- Design and implement a solution for a workflow to exchange information about network faults and scheduled downtimes
- Organize - through a set of established communication channels within the NRENs and DANTE - a unique contact point for the EGI community for all end-to-end performance investigation required issues (PERT)
- Liaise the EGI and NGI communities with the NRENs and DANTE, to exploit synergies on tools and their development, to agree on priorities and a general, agreed and shared model for the network support for EGI

4.2.2 Specific tools developed

This section gives now a high-level summary on the use of network monitoring for the Grid and the tools produced and used by the EGEE project

The tools described are:

- e2emonit and netmon2rgma
- ENOC - EGEE Network Operation Centre
- NPM - Network Performance Monitoring
- perfSONAR-Lite TSS (based on perfSONAR)
- Grid Jobs based Network Monitoring or NetJobs

e2emonit, netmon2rgma

e2emonit [20] is a collection of tools for providing end-to-end network measurement data, developed within the EDG and EGEE-I JRA4 projects. It is based on a set of wrapper scripts, written in Perl, which run the measurement tools themselves, and process their output, storing it for later consumption. The measurement tools included in e2emonit are:

- ping [21]
- iperf [22]
- udpmon [23]

producing a number of different metrics: Round Trip Time (two-way delay), two-way packet loss, TCP achievable bandwidth, UDP achievable bandwidth, one way delay variation and one-way packet loss. It was soon recognised in EGEE-II that these scripts were rather fragile for deployment onto a large production infrastructure, so that work was performed to improve their robustness. The focus of this work was on providing comprehensive unit and system tests for the scripts, which in turn allowed several faults to be discovered and fixed. For further details see the NPM Savannah list of issues [24]. The work also included the migration of the build process of e2emonit to the ETICS platform [25]. As such they were the first Perl based project to make significant use of ETICS, and were able to provide detailed feedback to the ETICS developers enabling several issues to be identified and fixed. The outcome of this process is a set of e2emonit packages which have been built and tested on several different computer platforms, and are fit for deployment.

Netmon2rgma is the part of e2emonit which taking the network measurements from the Perl wrapper scripts and storing them for later access in an R-GMA [26] database. R-GMA provides a "virtual database" to netmon2rgma, taking care of the data transport and storage requirements of e2emonit. During the course of the project netmon2rgma was migrated to ETICS along with the rest of the e2emonit components.

ENOC Network Operation

The ENOC (EGEE Network Operations Centre) acted as a single point of contact between EGEE and the NRENs, see Figure 4.3
It was designed in close collaboration among many parties (DANTE, NRENs, EGEE operation activity SA1) [32] and has been fully implemented since the end of EGEE-II. The service has run without any interruption since the end of EGEE.

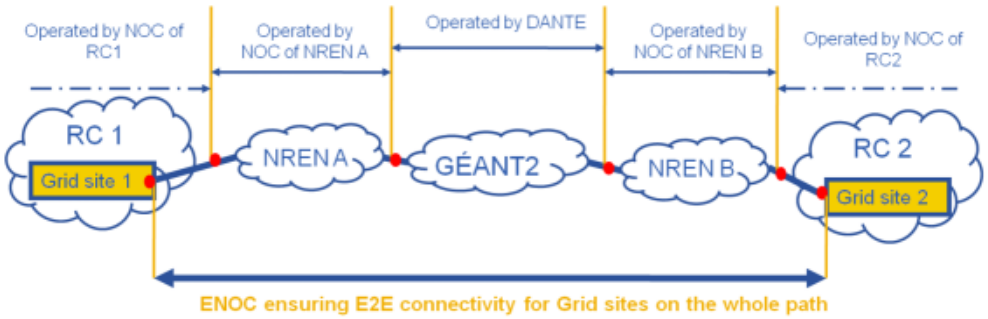

Figure 4.3: ENOC and EGEE, GEANT2 and NRENs

The goal of the ENOC was to provide an interface between Grid and network providers. Considering limitations in network provisioning and monitoring, the main roles were:

- To process information from network providers and to monitor the network Grid (problems, scheduled downtimes, etc.). The ENOC received network trouble tickets from network providers, parsed and converted them to a standard format [33]. They were then analysed and relevant ones are made available for site managers and Grid operators, see figure 4.4

- To provide user support by following network issues raised by users (mainly through GGUS).

- To provide network support to Grid operations around network issues

The ENOC dealt with network problems troubleshooting, notifications from the NRENs, network Service Level Agreement (SLA) enforcement and

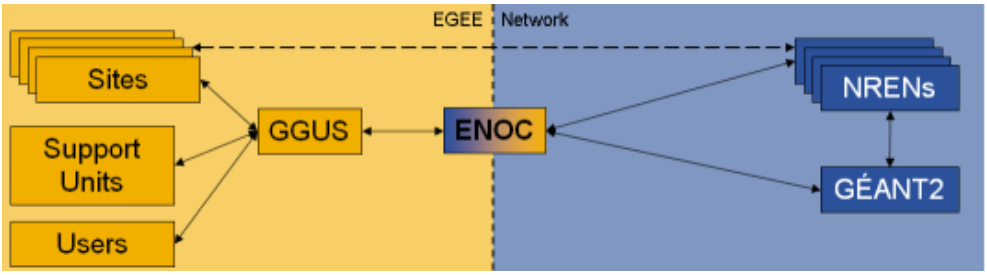

Figure 4.4: ENOC between GGUS and NRENs/GEANT

monitoring and network usage reporting. A network support unit in the Global Grid User Support (GGUS) of EGEE to provide coordinated user support across Grid and network services.

The automatic assessment of a trouble ticket's impact on the Grid has been very difficult to achieve. SA2 originally developed a statistic correlation method during EGEE-III that unfortunately did not provide enough accurate results, but did show that the NRENs must provide to EGEE more accurate tickets. SA2 wrote an Internet RFC draft [28] in order to foster standardization in this domain

The most important tools developed for the ENOC were:

- DownCollector: A tool assessing and reporting the connectivity for Grid sites [29].
- Network Operation Database schema: A database describing network infrastructure and site interconnection.
- ASPDrawer: LHCOPN monitoring providing a high level view of service available for the Grid [30]. This tool was tailored for the particular case of the LHCOPN, a dedicated network. This tool was used until October 2009.
- TTdrawlight [31]: A tool listing and mapping network trouble tickets on to a network map to show outages. Nevertheless, this tool still needs improvement.
- PerfSONAR-Lite TSS (described in the next session)

- NPM: Network Performance Monitoring (described in the next session)

Figure 4.5: ENOC alarm system

Network monitoring quickly amerged as a key requirement for the ENOC. A basic tool highlithing connectivity issues was released and provided very interesting results. Furthermore results were published to Grid operators through the CIC portal and through the Nagios monitoring prototype for Grid sites.

NPM: Network Performance Monitoring

During the course of EGEE-I, tools to allow uniform access to network measurement data from a heterogeneous set of measurement frameworks were developed by JRA4, as described in [34]. In EGEE-II this work has been continued by the Network Performance Monitoring (NPM) task of SA1, with more emphasis on the operational aspects.

Once realized that many different tools which provide network measurement data (e2emonit, netmon2rgma) were already available, the NPM task decided to focus on provide access to data collected by these existing tools. The situation is illustrated by Figure 4.6, which shows different end users using NPM developed services to access network data collected by a range of monitoring frameworks.

As Figure 4.6 shows, there are likely to be many consumers of the data, using some kind of client tool, requiring access to many sources of network data. It was therefore decided to develop middleware providing a single point of contact for clients, in order to simplify usage and configuration. The key middleware component is the Mediator, a webservice which

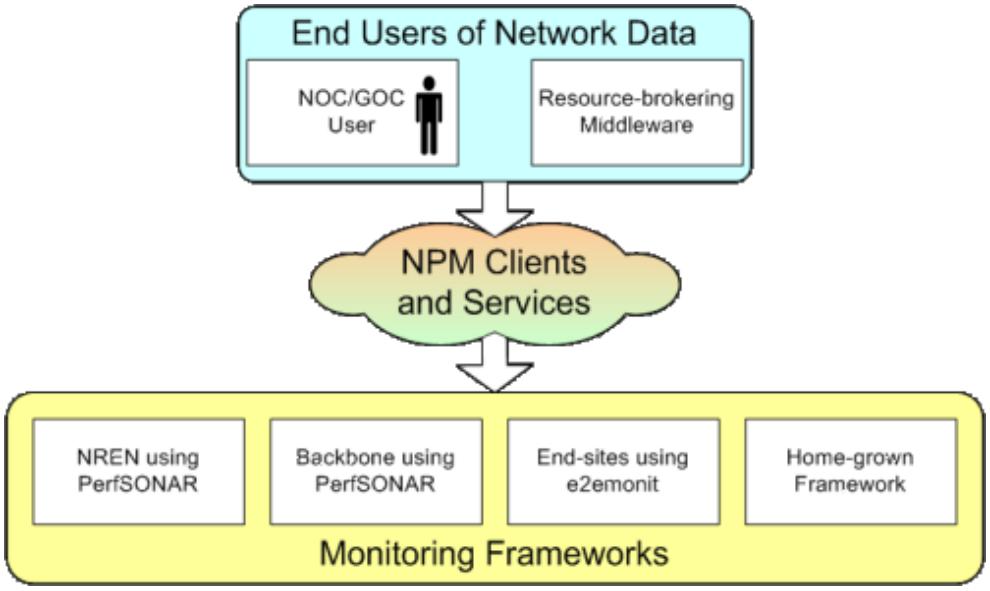

Figure 4.6: NPM Usage Scenario

currently supports two different measurement frameworks, the perfSONAR Measurement Archive for access to passive router utilisation data and the NPM provided NMWG4RGMA for access to e2emonit end-to-end performance measurements.

An overview of the NPM architecture is shown in Figure 4.7. The diagram shows the different components provided by EGEE NPM, and their interaction through the passing of NM-WG compliant XML messages.

Metric Availability

Taken together perfSONAR and e2emonit are able to provide the following metrics to users through the diagnostic tool:

Metric	Tool
Passive Link Utilisation	perfSONAR
Round Trip Time (Two-way delay)	ping
Two-way packet loss	ping
TCP achievable bandwidth	iperf
UDP achievable bandwidth	udpmon
One-way packet loss	udpmon
One-way delay variation	udpmon

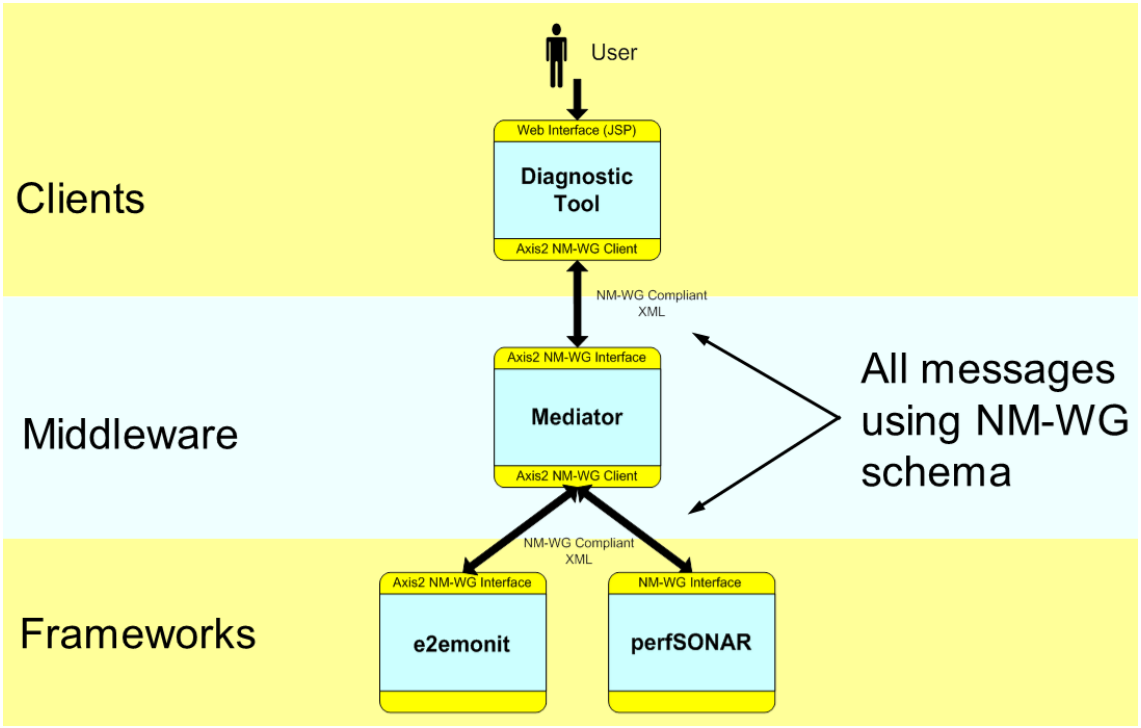

Figure 4.7: NPM Architecture

perfSONAR-Lite TSS and perfSONAR

This section describes the perfSONAR-Lite TSS and perfSONAR tools and the underlying mechanisms and protocols.

perfSONAR-Lite TSS

perfSONAR-Lite TSS represents another EGEE-III network troubleshooting tools developed in order to facilitate and speed up network problem solving for EGEE clients.

As a subcontractor of DFN, the University of Erlangen has developed software (perfSONAR-Lite Troubleshooting Service) for investigating throughput (tool BWCTL), packet run times (ping), paths in the network (traceroute), as well as port scans and DNS configurations. This is an easy-to-install variant of the perfSONAR monitoring software (perfSONAR is described in the next session).

Via a central web-server authorized Grid clients can request measurements between sites using tools such as traceroute, ping, portscan, dns-lookup or bandwidth measurements with BWCTL. Unlike already exist-

ing approaches, this EGEE-III network troubleshooting solution offers on-demand tests and measurements that can be run in limited time intervals over specific link connections without any permanent background measurements. The implementation is based on a platform independent plugin architecture in connection with a common core perfSONAR interface. Measurement requests and results are made available via the central web-server with only a lightweight client set up at each Grid site.

perfSONAR

perfSONAR is a framework and a set of Webservices protocols that enables network performance information to be gathered and exchanged in a multi-domain, federated environment.

It has been developed by the GN2/GN3 project but here described because another EGEE tool, perfSONAR-Lite TSS, was buil on it.

The goal of perfSONAR is to enable ubiquitous gathering and sharing of this performance information to simplify management of advanced networks, facilitate cross-domain troubleshooting and to allow next-generation applications to tailor their execution to the state of the network. This system has been designed to accommodate easy extensibility for new network metrics and to facilitate the automatic processing of these metrics as much as possible. perfSONAR is a joint project started by several national R&E networks and other interested partners. The complete set of participants is available from the perfSONAR web site [35]. The aim of this project is to create an interoperable framework to be gathered and exchanged in a multi-domain, heterogeneous, federated manner. perfSONAR is targeting a wide range of use cases.
For example current use cases include:

- collection and publication of latency data
- collection and publication of achievable bandwidth results
- publication of utilization data
- publication of network topology data

- diagnosing performance issues

While perfSONAR is currently focused on publication of network metrics, it is designed to be flexible enough to handle new metrics from technologies such as middleware or host monitoring. One can envision a number of future, higher-level services that will use the perfSONAR data in interesting ways. For example, data transfer middleware could use perfSONAR to locate the best replica/copy of a file to request, or to help determine the optimal network protocol to use for a given link. Network engineers could use perfSONAR to help automate the detection of large bulk data flows that may require special handling, such as tagging the flow as high- or low-priority, depending on its source or destination. Finally, network researchers will find perfSONAR-enabled networks a convenient source of performance and topology information. A focus of the perfSONAR project has been to define standard schemas and data models for network performance information. Development of actual, interoperable implementations has followed the Internet Engineering Task Force (IETF) spirit of multiple working interoperable implementations. There are at least 10 different organizations producing perfSONAR-compliant software implementations as of today.

The Major perfSONAR Services can be summarized as below and represented in Figure 4.8

- Measurement Point Service (MP): Creates and/or publishes monitoring information related to active and passive measurements
- Measurement Archive Service (MA): Stores and publishes monitoring information retrieved from Measurement Point Services
- Lookup Service (LS): Registers all participating services and their capabilities
- Authentication Service (AS): Manages domain-level access to services via tokens
- Transformation Service (TS): Offers custom data manipulation of existing archived measurements

- Resource Protector Service (RPS): Manages granular details regarding system resource consumption
- Toplogy Service (TS): Offers topological information on networks

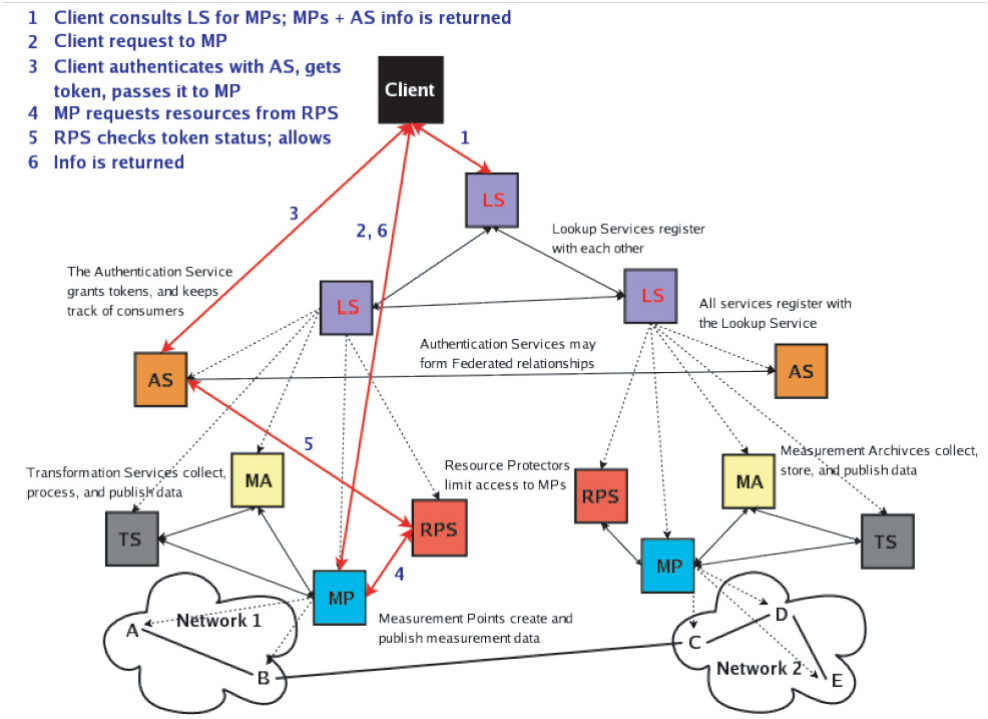

Figure 4.8: PerfSONAR system components

Grid Jobs based Network Monitoring or NetJobs

This software was developed based on the paradigm: "Monitoring the GRID network using the GRID itself"

This project represents the main part of this PhD work and will be deeply analyzed in the following chapter.

Chapter 5

Grid Network Monitoring based on Grid Jobs

5.1 Introduction

During the life of EGEE project was never addressed a network monitoring solution at the grid level. The SA2 group tried to fill this gap designing and carrying out two different tools:

- A perfSonar-based solution designed for network troubleshooting knows as perfSONAR-Lite TSS
- A complementary solution for site-to-site continuous monitoring

A site-to-site continuous monitoring tool was prototyped by the SA2 group and focused on the idea of monitoring the grid using grid jobs. The name of this tool is NetJobs and the main developers are Etienne Double from CNRS UREC institute and I.

Being deeply involved in the design part and GUI (Graphical User Interface) development, NetJobs represent the main part of my PhD activity.

NetJobs has been realized following five different phases, below described:

1. Description of the actors and their corresponding requirements regarding a network monitoring system
2. Technical considerations

3. NetJobs: a GRID Network Monitoring based on Grid Jobs

4. NetJobs: Architecture overview

5. Netjobs: A prototyped system to demonstrate as theorized

5.2 Actors and their requirements

Actors who could benefit from a network monitoring system at the grid level can be categorized into three different groups: ROCs (Regional Operation Centers) and Sites, Applications and Middleware, Grid Users. Each group has various expectation from such a system.

5.2.1 ROC and Sites

The main requirements of importance for ROCs and sites are the following ones:

- The monitoring tools MUST be able to provide end-to-end performance data of network paths important to applications.
- The monitoring frameworks MUST generate alerts when performance falls outside a preset range.
- The alerts generated MUST be accessible in a known location.
- Each alert SHOULD have associated with it details of how and why it was generated.
- The measurements MUST be available for both IPv4 and IPv6 protocols.
- Historical data MUST be available for all metrics.
- On demand measurements MUST be possible for all metrics.
- Alerts on either a target value or a target rate MUST be available for all metrics.

For what concerns the required network performance metrics, it MUST be possible to obtain measurements of the following metrics:

- RTT (Round Trip Time)
- Packet Loss
- Capacity
- MTU (Maximum Transmission Unit)
- OWD (One Way Delay) - if possible
- IPDV (IP Packet Delay Variation) - if possible

Specifically, the latter two (OWD and IPDV) are very difficult and expensive to get and those metrics seems to have only a low impact on EGEE traffic which is mainly TCP that is why we added the mention - if possible - to the ROC and site requirements.

5.2.2 Application and Middleware

The EGEE grid is underlying various applications, and its middleware is working as an interface between these applications and the sites. This section describes the needs of the applications and of the middleware concerning the monitoring system.

Grid traffic classification in terms of QoS

We classified the following kinds of traffic for the EGEE grid according to the RFC 4594:

Category of applications running on the grid:

1. Internal grid middleware communication (messages between nodes, for example VO authentication between UI and VOMS, or scheduling messages between WMS and CE, or registration of resources on a BDIIĚ.) Corresponding category in the RFC
2. Data transfers (GridFTP, RFIO)

3. SAM, Network monitoring (ENOC)

Which represent respectively:

1. Standard Grid middleware intercommunication processes

2. High-Throughput Data Trasfers

3. OAM (Operations, Administration, and Management)

According to the RFC [28] data transfers (GridFTP, RFIO) may be classified in the category "Low-Priority-Data" because they are tolerant to packet loss (being TCP-based) but the performance would be impacted. We want transfers to be fast, so High-Throughput Data may be better choice.

Ideally it would be good to differentiate between synchronous and asynchronous transfers of data. The asynchronous transfers (via FTS for example) don't have a user (or a job) waiting for the data and a lower-quality treatment of that data would have few unwanted side effects. Synchronous transfers, however, mean that a user (or a job) is waiting for the data and a lower-quality service would probably mean wasted CPU, etc. Therefore it might be better to allow the services to mark the quality of service desired. They would set a DSCP value of AF11, AF12 or AF13 (sub-categories of "High-Throughput Data"), or maybe, in particular case where latency is critical, AF21, AF22, AF23 (sub-categories of "Low-Latency Data"). Because of the amount of work needed for this kind of differentiation, this was kept out of this PhD activity. For common gLite-based applications, no other kind of traffic has been determined. However, another use case to be considered is the one related to Grid projects with real-time or quasi-real-time applications and which are using the EGEE infrastructure and its gLite middleware. The DORII project [36] is an example. Its real-time environment leads high networking needs. This link between EGEE and DORII will probably become stronger as we move towards EGI and consequently it would be probably wise to have a common network monitoring solution.

Middleware specific needs for a monitoring system

The middleware could obviously take benefit from a network monitoring infrastructure. It is reported here two use cases:

1. The knowledge of the availability or a degraded performance to join a site could be used by the File Transfer System (FTS) component. The FTS could thus modify the priority of these data transfers that it manages.

2. The job submission could take into consideration the network performance into its decision process for the choice of the site that will process the job (see EGEE DJRA4.7).

5.2.3 GRID Users

A grid user may also be interested in a monitoring system, for troubleshooting needs. For example if a job failed during the night, a user may want to retrieve the network state at this time. Data archival is therefore a major requirement. Having the option to be able to easily export archived data in some format (ASCII, Excel, XML, etc.) is also highly desirable, in order to subsequently being able to easy manage them. Another possible case of interest for a user is to investigate the reasons for a slow data transfer while transferring data among grid sites. Note: PerfSONAR Lite TSS complements this approach for the on-demand network probing needs of the grid users.

5.3 Technical considerations

This part summarizes the technical topics that should be kept into account in the design phase of a network monitoring system.

5.3.1 Incremental process and adaptability

A monitoring system should be implemented in an incremental way, starting from the most important features: it would allow taking benefit from it already in the short term. The system should be able to adapt itself to

the moving needs. For example, a given site could decide to host a new VO in future, and therefore need monitoring information for paths which were previously not considered relevant. Also, for new experiments, some additional metrics could be required. Two solutions exist to ensure this adaptability:

- Either the system adapts itself automatically
- The users have a way to request adaptations

5.3.2 Acceptable policies for GRID sites

The monitoring system must:

- Be secure
- Not be too demanding/intrusive regarding network and computing resources
- Preferably be easy to deploy

5.3.3 Metrics

We discuss in this section the metrics we consider relevant for grid infrastructures. Since network providers usually provide per-link metrics already, the metrics we are describing should be measured end-to-end (i.e. site-to-site). We plan to have a reasonable set of metrics implemented first (possibly the easiest first), and the others following, in an incremental way, in order to get the first results as soon as possible. For example, it would be good point to get at least one metric related to latency: preferably a measurement of type "One-Way-Delay", or, if not possible, RTT. It would also be a good point to get at least one metric related to bandwidth: achievable bandwidth or available bandwidth. The capacity might be registered at a later stage. The packet losses or route changes could also be measured in a later step, because, looking at the latency and bandwidth data only, this kind of problem should already be detected. The MTU could also be interesting although not absolutely necessary. Some metrics are less interesting in a grid context.

Jitter (the deviation between the time when a device is expected to issue a message and when the message is actually transmitted), for example, is not really interesting for common gLite-based applications because the grid is currently not providing any services able to handle real-time applications. It would only be interesting for related projects like DORII (A real-time environment leads high networking needs). Anyway, depending on the user feedback, the system could be improved to include more metrics.

5.3.4 Metrics evaluation

The following table reports the required metrics and provides some relevant evaluations on them, including the way how to acquire them. Some metrics are difficult to acquire, as explained by the table, due to various reasons (i.e. they require an infrastructural overhead, for example).

Metric	Comments	Difficulty
One-way-delay and Round-Trip-Time	OWD requires at least NTP synchronization. OWD is a metric that seems not impacting a lot TCP transfer which is the most common usage of the network by EGEE applications. RTT is easier to implement in a first step and the variation of RTT can also provide instructive results.	Easy (RTT) to Medium (OWD)
Packet Loss	Although some tools like "ping" display "Packet loss" information, losses usually occur in a seldom way, and consequently active monitoring is not well adapted to this measurement.	Easy (in a passive monitoring context)
Bandwidth Capacity	The Capacity is usually measured by retrieving SNMP data from routers, which can hardly be envisioned at the grid level: for a site-to-site path, it would require to know the path used by the network packets and to aggregate the data from all the routers along the path. We could, instead, use one of the tools available to estimate it (nettimer for example).	Difficult
Achievable bandwidth	The end-to-end Achievable bandwidth (iperf, netperf, etc.) is the most obtrusive bandwidth measurement because the link is flooded with network packets, which has the side effect of reducing the bandwidth of other applications along the path (at least in a best effort context). Therefore we may not collect this metric.	Medium
Available bandwidth	The end-to-end Available bandwidth could be measured by tools which are able to provide an estimation of this metric without flooding the link.	Difficult
MTU	Traceroute can return this value. It can also be guessed by using several ping probes.	Easy
Topology changes	It could be detected by variations in traceroute or hop count results. It is not easy to implement a traceroute metric which would work in all cases, because firewalls often block some related packets.	Easy (hops) to Difficult (full route data)

5.3.5 Time considerations

Due to costs considerations, being compatible with our requirement time resolution, the pragmatic choice is to use NTP.

5.3.6 Accuracy

As mentioned in ref [37], applications using network measurements require a given level of accuracy from these measurements and also assume that all the measurements provided have been validated. Recommending specific values for accuracy, which are generic enough to be acceptable to any application, isn't practical and not advisable. However, we suggests that when network measurement values are provided, their accuracy should be included as well. Applications can then choose to believe the network measurements or disregard them based on their accuracy values. Taking into account the cost of a time synchronization system, NTP appears as a pragmatic choice that covers the majority of EGEE use cases.

5.3.7 Directions

In ref [37] it is stated that is essential, unless specified for, all the metrics to have an associated direction. No a priority assumptions should be made on the direction being both forward and backward. Nevertheless achieving to have metric in both directions will increase a lot the complexity of the monitoring tool deployment.

5.3.8 Running probes on heterogeneous hardware

Some specific measurements may require specific hardware. For example, if one needs a precise One Way Delay measurement, a GPS antenna is required. This implies requirement for a GPS card and patched OS. On the contrary, if less precision is needed, other solutions based on NTP synchronization could be enough. Consequently, considering costs, and even if it would be appreciable to get all probes running on the same proven hardware architecture, in all EGEE sites, the pragmatic choice is to allow heterogeneous hardware.

5.3.9 Metrics aggregation

An analysis carried out within SA2 (Domenico Vicinanza, DANTE) explored this point and exposed the conclusions at EGEE's Technical Network Liaison Committee of March 2010 [38]: depending on the metrics, some may be aggregated (with some limitations, like the RTT), some others not, for example the achievable bandwidth. As a general rule, the advice is to avoid aggregation.

5.3.10 Site paths to monitor

Monitoring all site-to-site possible links would be obviously too much: there are about 300 sites, which would mean 300*299/2=44850 links. As a consequence we have to choose which site-to-site paths the system should monitor. Our conclusion is that the system should NOT choose the paths itself, but, instead, provide an interface to the user for this. Through this interface, the user would be able to select the site-to-site paths which are important for him, fill a request form, and after validation by the system administrator, these additional paths would be monitored.

5.3.11 Frequency of measurements

The frequency of measurements should not be chosen too high, in order to avoid high intrusiveness in the network. This parameter should be matched to existing systems, and, optionally give the user a reasonable time interval range to chose from.

5.3.12 Synchronisation

Two measurements running at the same time could negatively effect each other. This is especially true for bandwidth measurements. In order to avoid this, the system should schedule the measurements in order to avoid that a given probe is involved at the same time in two bandwidth tests. Note: This is theoretically not enough. For example, if we consider 4 sites A, B, C and D, the paths A<->B and C<->D might share the same subsection

of network segments. However, regarding the complexity of such topic, the handling of this problematic case will not be treated.

5.3.13 Archiving

The system should preferably archive all values in a data storage solution, like a database. Depending on the frequency of measurements and the number of metrics could be required too much disk space. In this case a consolidation mechanism (aggregation of older values) like RRDTool should be used.

5.3.14 Security

The access to the tool should be secured since it could be used to generate a DOS attack. If the tool requires a client and server connection, this connection should be secured ideally by a mutual authentication system based on crypt mechanism. Moreover, malfunctioning of the alarming system could generate lot of messages that could influence heavily grid any operations. The privacy of the data is also a concern and the result should be available only to the appropriate persons. Authorization schema should be then defined accurately. [43]

5.3.15 Usability

Last but not least a network monitoring system should be easy to query. A user interface that presents information graphically, typically with draggable windows, buttons, and icons is preferable to a pure textual interface.

A GUI (Graphic User Interface) should be designed following these main items:

- Visual design
- Functionality
- User-friendliness
- Consistency

- Efficiency
- Performance
- Navigation
- Feedback
- Standard compliant

5.4 NetJobs: a GRID Network Monitoring based on Grid Jobs

5.4.1 Preliminary work

As a preliminary work concerning the idea of sending grid jobs to monitoring the GRID network, we have tried to get a map of the EGEE grid at level 3, by sending traceroute commands among sites. These first results were compiled in the map (May 2009) The interactive map in Figure 5.1 was generated by sending a job to the sites in the dteam VO. This job tried to determine the network path (using the traceroute command) to all other sites in the dteam VO.

On this map are only represented the sites which satisfy all the following items:

- The site is registered in the dteam VO
- The site contains at least one CE
- The CE was accessible at the time of the test

Each network segment was shown if at least one of the related traceroute probes succeeded. If several routers or network segments were found with the same GPS coordinates, only one was shown. GPS coordinates were obtained from the site http://www.geoiptool.com (they seem to be correct in most cases).

Figure 5.1: EGEE Jobmaps by traceroute

5.4.2 Advantages and limitation

A monitoring system based on grid jobs has some advantages and some limitations.

Among advantages we remember:

- No deployment work is needed in all grid sites
- Grid coverage is maximum
- The system could benefit from grid services, for example:
  - Storing results on a SE
  - Benefiting from the built-in authentication between nodes
- Considering that the aim of our monitoring system is to measure network performance between grid nodes, a grid node is the best place for executing a probe

The technical difficulties are:

- Lack of full control on the probe environment. For example:
  - It is not possible to run network monitoring tools as root (a grid job does not have such privileges)
  - The location (in the network architecture) and hardware of the probe cannot be chosen (it will be the one of the Worker Node)
- A robust handling of the middleware must be implemented in order to avoid that problems of the middleware impact the behavior of the system

Being aware of the fact that the active network grid monitoring is intrusive and bandwidth consuming, the tests are scheduled in order to avoid disturbing sites and grid behaviour. We benefited from the experience of the LHCOPN monitoring system in order to determine the period of the tests, and the amount of data sent for each bandwidth measurement.

5.4.3 NetJobs Architecture overview

The basic architecture is shown by Figure 5.2. A user connects to a web front end which displays monitoring data recorded in a database. The same front end may be used to connect to various databases. This is useful if a user is member of several projects and each of them implements this monitoring system internally. A database is filled in by one or more monitoring servers. Each of the monitoring servers is managing a subset of grid jobs. The grid jobs are collecting the network monitoring metrics. Having several servers registering in the same database allows getting the expected scalability (when there are many jobs to manage).

Initialization phase

Figure 5.3 shows a simplified view of the job initialization phase:

In a first step, the Central Monitoring Server Program (CMSP) submits jobs to each site. In a second step, when a given job is running, it connects back to the CMSP. This socket connection will remain active and will be the base of the communication between the CMSP and each of these jobs.

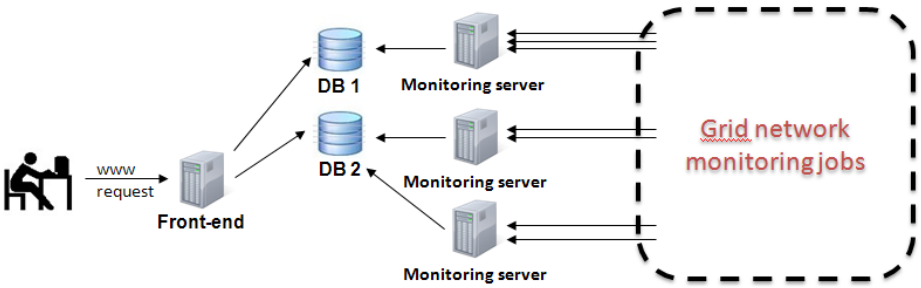

Figure 5.2: NetJobs Architecture

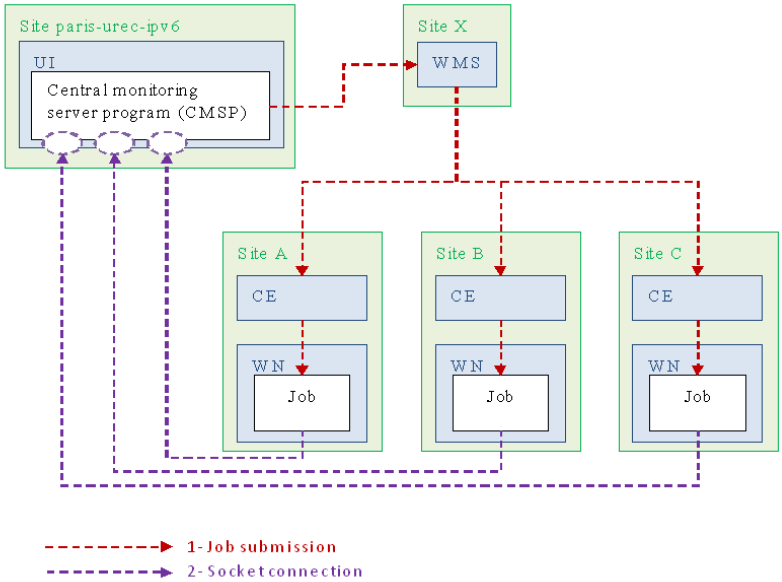

Figure 5.3: NetJobs Schema

The main reason for this design is the important delay between the job submission and the job execution There are many resources involved for each job submission. For this reasons it was not efficient enough to start one job each time a probe is needed. The fact that the socket connection is initiated by the job and not by the CMSP allows avoiding the security risk of having one listening socket opened in each site. Additionally an authentication mechanism is implemented in order to verify the identity of both endpoints.

A job cannot run forever because of the limits set at the GlueCE object level in the Computing Element (clusters). Mainly, if a job is still running after the delay given by GlueCEPolicyMaxWallClockTime, it will be aborted. In order to avoid artificially generating a high number of aborted jobs, the monitoring job should stop before before the GlueCEPolicyMaxWallClockTime expires. Anyway, the CMSP is able to detect if the socket connection is closed and to stop the job itself. At this time it will submit a new monitoring job to the same site. There will be a little shift while between the time the job is submitted and the time it is run on the WN. Because of that, there will be 2 jobs permanently running at each site, so that when one stops, the other one is still able to handle the scheduled probes.

Test phases

The monitoring tool designed can perform different tests. Will be described here two tests:

1. A latecy test: RTT (Round Trip Time)

2. A bandwidth test: GridFTP data transfer

Round Trip Time test Figure 5.4 shows a simplified view of a round-trip-time test to be run from site A to site B:

The test follows this scenario:

1. The CMSP contacts the job running at Site A and requests it to run a RTT test to Site B

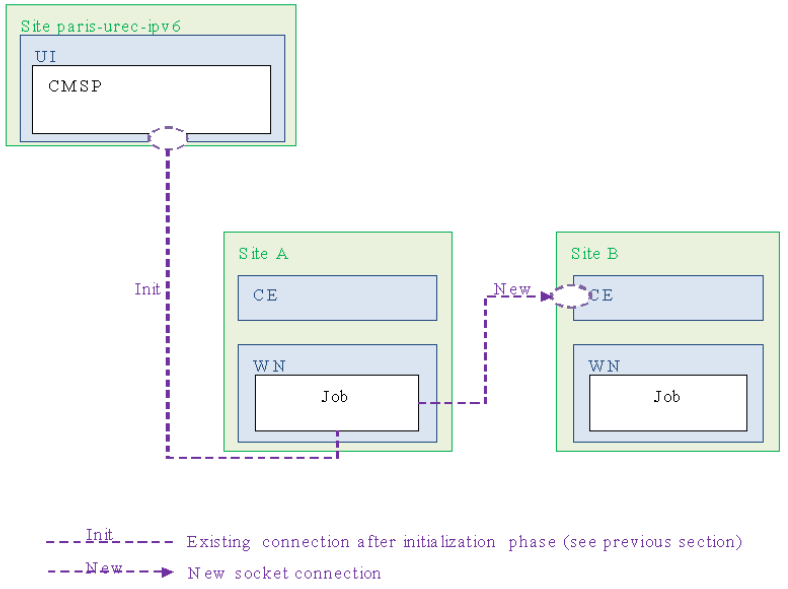

Figure 5.4: NetJobs RTT test

2. The Job at site A connects to the TCP port of a CE at site B, measures the time the connect() call takes, and closes the connection (see next sections for explanations)

3. The job at site A notify the result of the test to the CMSP

This kind of test should be run every few minutes.

The reasons for choosing this design is that ICMP is often blocked over the network; therefore we have chosen to use TCP in order to compute the RTT ("Round Trip Time").

A TCP connection follows these steps:

- The client sends SYN to the server
- The server acknowledge by returning SYN/ACK
- The client confirms by ACK

Because of this, from the time the SYN message is sent, the SYN/ACQ message will be received after a time corresponding to a "Round Trip Time" (since a "round trip" of packets is involved). The last step is considered instantaneous because the client does not wait for any response. Therefore

we can estimate that the RTT is equivalent to the time a connect() function call takes. Note: We confirmed this by comparing values obtained by this technique to values obtained by ping, and they were very similar.

The second step was to find, at a given site, a TCP port on a gLite node which could be opened from outside the site. We found that the TCP port of the CE job queue was a good candidate, since we could easily read it in the BDII. For example if the queue is ce-4.dir.garr.it:2119/jobmanager-lcgpbs-dteam, we can make the RTT test to the port 2119 of ce-4.dir.garr.it.

There are other metrics collected by this same test Actually, several connections are performed, in order to also collect:

- The MTU (by reading the IP MTU socket option)
- The hop count (by an iterative method using connection attempts with various values of IP TTL)

GRIDFTP Bandwidth test Figure 5.5 a simplified view of an active GridFTP bandwidth test to be run from site A to site B

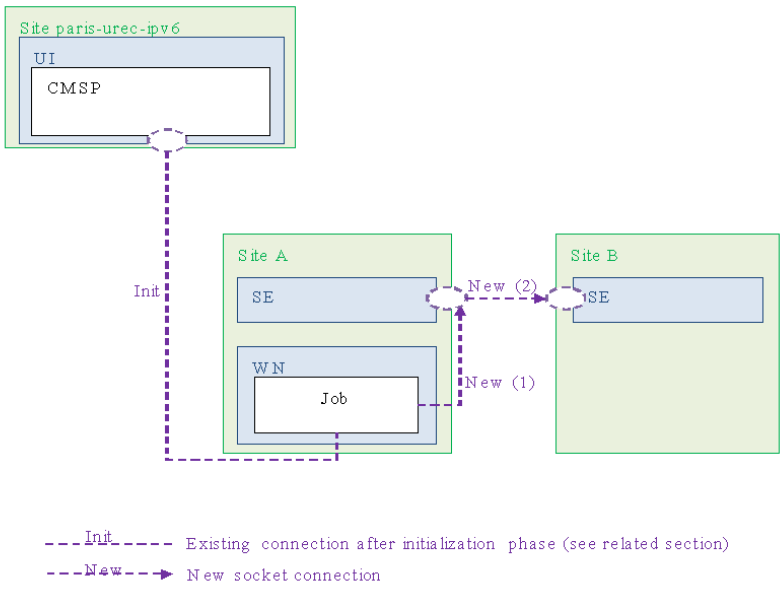

Figure 5.5: NetJobs BWT test

The test follows this scenario:

- The CMSP contacts the job running at Site B and requests it to open a TCP port
- The Job at site B opens a port P and notify it to the CMSP
- The CMSP contacts the job running at Site A and requests it to start a transfer A to B
- The Job at site A request this transfer and measures the time it takes
- The job at site A notify the result of the test to the CMSP

For a given site pair, this test should be run a few times a day and send data for around 10 seconds each time.

Reliability, Compatibility and Scalability of the System

The management of gLite jobs is complex. The system has to ensure that, in a given site, when a job stops, another one is ready to continue fulfilling the server requests. However, in order to avoid generating much load on the grid, the server cannot send too many job requests. This management has been improved little by little and now seems to be satisfactory. However:

- This management of gLite jobs should still be monitored in order to ensure that its behaviour is satisfactory during a longer period of time
- The gLite parameter GlueCEPolicyMaxWallClockTime must be reasonably high. Otherwise, in the case of sporadic high submission delays or gLite failures, there might be some periods of time where no job is listening.

A reasonable rule is: GlueCEPolicyMaxWallClockTime $>$ 6h; there is no upper limit: the higher, the better. The source code of the server is still young and some problems may still appear in the future. Some of the problems may be due to gLite bugs; for example a gLite bug caused failures when passing to summer time (march 27 to 28, 2010, see [36]).
Concerning the jobs reliability, gLite failures may only occur at job submission time. Once the job is running, the only failures we can expect are bugs

in the job code itself.
The system must be compatible with all worker nodes which provide a Python >= 2.3 interpreter. The source code is developed using Python 2.3, but, if a given worker node provides a later version, backward compatibility should apply. Anyway if an issue is detected, the code could be adapted. Since some of the gLite code is using Python, this requirement should not be a problem. If, however, jobs are not running at a given site, the site administrator should provide a temporary remote access to a worker node in order to solve the problem.

Compatibility of measurements methods

The RTT/MTU/hops test and the active gridFTP test are compatible to all sites. The passive gridFTP test is only compatible with Storage Elements which allow remote GridFTP access to their GridFTP log files. It seems that this is true for DPM-based SE only.
It is possible to start several instances of the server, each of them managing a respective set of sites. This ensures a good scalability. [43]

5.4.4 Proof of concept

NetJobs has been engineered and developed in two part: a backend, the core, written in python and bash scripting language and a frontend, coded in php and AJAX.

Since Jan 2010 8 GRID sites were involved in a testbed to prove the tools on the road. Site were chosen in a multi domain and international contest to reflect a real use case as much as possible.

The 8 GRID sites, showed in Figure 5.6 are:

1. Paris Urec CNRS
2. IN2P3 Lyon
3. INFN-CNAF
4. INFN-ROMA1

5. INFN-ROMA-CMS

6. GRISU-ENEA-GRID

7. INFN-BARI

8. INFN-CATANIA

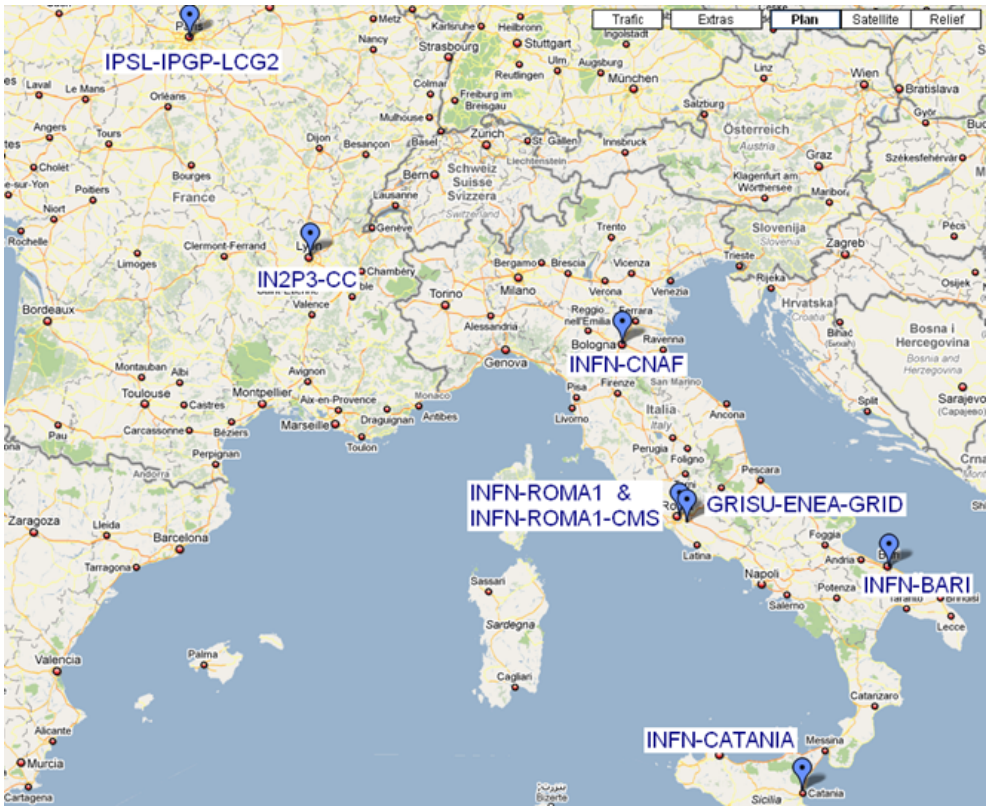

Figure 5.6: Sites Involved in NetJobs

Data are collected from the probes running at the 8 sites and displayed and graphed through dynamic plots. Thanks to the frontend showed in Figure 5.7 the tool can be easily queried by any user.

Among the features we remember:

- queries per site or glite job ID
- different DBs associated to different sites
- dynamic plots based on all or selected data

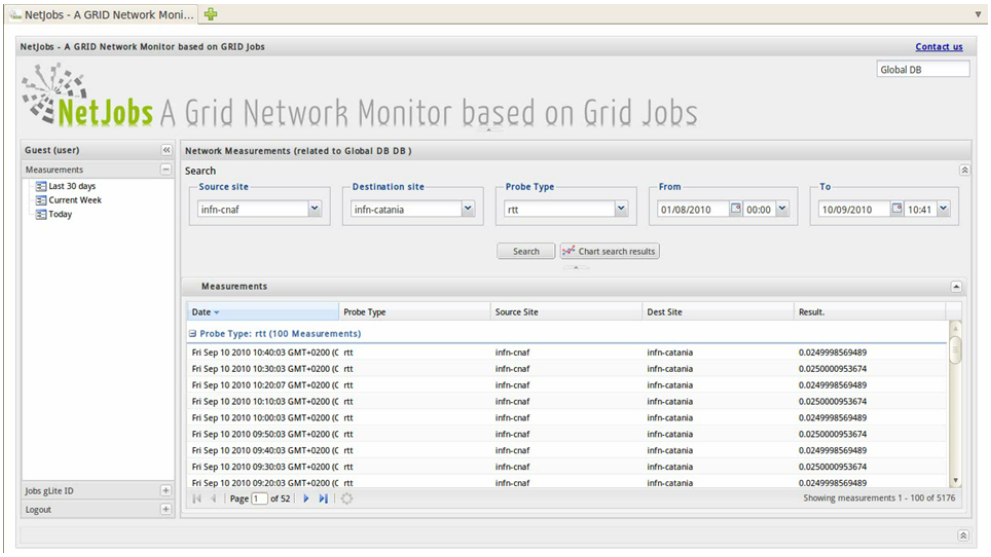

Figure 5.7: NetJobs User Interface

- time range selection available

This work has been presented during three international technical workshops, receiving encouraging evaluations and feedbacks:

- LHCOPN meeting, Bologna - Dec 19th, 2009
- 4th TNLC (Technical Network Liaison Committee) EGEEIII, Lyon - Feb 24th, 2010
- EGI TF, Amsterdam - Sept 15th, 2010

Has been observed as Netjobs can help users and grid site administrators to analyze a site-to-site network path, troubleshooting any problems.

5.4.5 Conclusion and further work

Despite Netjobs has proven to be a good prototype, there is room for improving it. Stability on large environment and a proper alarm system based on instant message and email are the two high priority working items.

Glossary

A

API Application Program Interface;

B

BDII Berkeley Database Information Index;

C

CA Certification Authority;
CE Computing Element;
CERN European Laboratory for Particle Physics;
CM Computer Model;
CMS Compact Muon Solenoid;
CNAF INFNs National Center for Telematics and Informatics;
CNRS National Center for Scientific Research, France;
CP Charge Parity;
CPU Central Process Unit;
CRM Cluster Resource Manager;
CVS Concurrent Version System;

E

EDG European DataGrid;
EGEE Enabling Grids for E-Science in Europe;
EGI European Grid Initiative;
EMI European Middleware Initiative;
ESM Experiment Software Manager;
EU European Union;

F

FTP File Transfer Protocol;

G

GeV Giga electron Volt;
GIIS Grid Index Information Server;
gLite Lightweight Middleware for Grid Computing;
Globus Globus Toolkit Grid Middleware (middleware stack);
GLUE Grid Laboratory for a Uniform Environment;
GRAM Globus Resource Allocation Manager;
GRIS Grid Resource Information Service;
GSI Grid Security Infrastructure;
GUI Graphical User Interface;
GUID Grid Unique ID;

H

HA High Availability;
HEP High Energy Physics;
HPC High-performance computing;
HTC High-throughput computing;
HTTP Hyper Text Transfer Protocol;

I

ICMP Internet Control Message Protocol;
ID Identifier;
IN2P3 Institut Nacional de Physique Nuclèaire et de Physique des Particules, France;
INFN Istituto Nazionale di Fisica Nucleare;
IP Internet Protocol;
IS Information Service;
ISO International Standard Organization;

J

JA Job Adapter;
JC Job Controller;
JCS Job Control Service;
JDL Job Description Language;

L

LAN Local Area Network;
LB Logging and Bookkeeping Service;
LCFG Local ConFiGuration System;
LCG LHC Computing Grid;
LDAP Lightweight Directory Access Protocol;
LFN Logical File Name;
LHC Large Hadron Collider;
LHCb Large Hadron Collider beauty experiment;
LM Log Monitor;
LSF Load Sharing Facility;

M

MAC Media Access Control;
MB Match-Maker Broker;
MDS Monitoring and Discovery Service;
MW Middleware;

N

NGI National Grid Initiatives ;
NOC Network Operations Center;
NS Network Server;
NTP Network Time Protocol;

P

PERL Pratical Extraction and Report Language;
PFN Physical File name;
PHP Hypertext Preprocessor;
PID Process IDentifier;

R

RA Registration Authority;
RAL Rutherford Appleton Laboratory;
RAM Random Access Memory;
RB Resource Broker;
RC Replica Catalog;
RLS Replica Location Service;
RM Replica Manager;
ROC Regional Operation Center;
RPC Remote Procedure Call;
RPC Resistive Plate Chamber;
RPM RedHat Package Manager;

S

SC Super Computing;
SDK Software Development Kit;
SE Storage Element;
SP Simulation Production;
SNMP Simple Network Management Protocol ;
SSH Secure SHell;

T

TCP Transmission Control Protocol;

U

UDP User Datagram Protocol;
UI User Interface;
URL Universal Resource Locator;

V

VO Virtual Organization;

W

WAN Wide Area Network;
WLCG The Worldwide LHC Computing Grid;
WM Workload Manager;
WMS Workload Management System;
WN Worker Node;
WP Work Package;
WPn Work Package number;

Bibliography

[1] David J. Farber; K. Larson (Sept 1970). *"The Architecture of a Distributed Computer System - An Informal Description". Technical Report Number 11, University of California, Irvine.c*

[2] Mockapetris, Paul V.; David J. Farber (1977). *"The Distributed Computer System (DCS): Its Final Structure". Technical Report, University of California, Irvine.*

[3] What is the Grid? A Three Point Checklist
http://www-fp.mcs.anl.gov/ foster/Articles/WhatIsTheGrid.pdf

[4] IBM Solutions Grid for Business Partners: Helping IBM Business Partners to Grid-enable applications for the next phase of e-business on demand.

[5] A Gentle Introduction to Grid Computing and Technologies
http://www.buyya.com/papers/GridIntro-CSI2005.pdf

[6] The Grid Café - What is Grid?
http://gridcafe.web.cern.ch/gridcafe

[7] Wikipedia the free encyclopedia that anyone can edit
(http://en.wikipedia.org/wiki/Grid_ computing)

[8] The Enabling Grids for E-sciencE (EGEE) project
http://public.eu-egee.org

[9] The Enabling Grids for E-sciencE (EGEE2) project, the second two-year phase started on 1 April 2006

[10] The INFN Grid project - *http://grid.infn.it*

[11] Introduction to virtualization
http://wiki.openvz.org/Introduction_ to_ virtualization

[12] gLite documentation http://glite.web.cern.ch/glite/documentation/default.asp

[13] Foster, Kesselman, Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, Int. J. High Performance Computing Applicat., 2001

[14] CESGA Experience with the Grid Engine batch system

[15] GLUE Working Group (GLUE)

[16] OGF MDS 2.2 Features in the Globus Toolkit 2.2 Release

[17] F Pacini, EGEE User's Guide, WMS Service, DATAMAT, 2005

[18] EGEE User's Guide, Service Logging and Bookkeeping (L&B), CESNET, 2005

[19] The Globus Toolkit 4.0, Overview of the Grid Security Infrastructure

[20] e2emonit http://www.egee-npm.org/e2emonit/

[21] The Story of the PING Program

[22] The TCP/UDP Bandwidth Measurement Tool http://dast.nlanr.net/Projects/Iperf/

[23] UDPmon Home Page http://www.hep.manchester.ac.uk/u/rich/net/

[24] EGEE NPM Savannah Bugs https://savannah.cern.ch/bugs/?group=jra4npm

[25] ETICS - The Grid Quality Process

[26] R-GMA http://www.r-gma.org/

[27] Trouble ticket data model https://edms.cern.ch/document/866697/1

[28] "The Network Trouble Ticket Data Model" Internet Draft http://tools.ietf.org/html/draft-dzis-nwg-nttdm-01

[29] DownCollector https://ccenoc.in2p3.fr/DownCollector/

[30] ASPDrawer https://ccenoc.in2p3.fr/ASPDrawer/

[31] TTdrawlight - Dashboard of network troubles and tickets provided by the ENOC https://ccenoc.in2p3.fr/TTdrawlight/

[32] MSA2.3: Operational interface between EGEE and GEANT/NRENs https://edms.cern.ch/document/565449/2

[33] Trouble ticket data model https://edms.cern.ch/document/866697/1

[34] DJRA4.7: Report on Network Monitoring https://edms.cern.ch/document/695235

[35] Performance focused Service Oriented Network monitoring ARchitecture. http://www.perfsonar.net

[36] The DORII project http://www.dorii.eu

[37] Network Performance Metrics Update https://edms.cern.ch/document/475908/2

[38] Minutes of the TNLC of March 2010: https://edms.cern.ch/document/1064397/1

[39] GGUS ticket https://gus.fzk.de/ws/ticketinfo.phpticket=56877

[40] LHCOPN meeting - Bologna, Dec 19th, 2009

[41] 4th TNLC (Technical Network Liaison Committee) EGEEIII Feb 24th, 2010

[42] EGI TF, Amsterdam - Sept 15th, 2010

[43] Network monitoring for EGEE and beyond, https://edms.cern.ch/document/1001777/1

[44] Job Based Network Monitoring Technical, Analysis https://edms.cern.ch/document/1011901

List of Figures