Optimizing bulk data transfers using network measurements: a practical case

A. Ciuffoletti
Dept of Computer Science - Pisa - Italy
S. Pardi
INFN-Napoli Section - Italy
L. Merola, F. Palmieri, G. Russo
Universitá degli Studi di Napoli Federico II Complesso di M.S. Angelo Via Cintia - Napoli

November 10, 2008

Abstract

The quality of the connectivity provided by the network infrastructure of a Grid is a crucial factor to guarantee the accessibility of Grid services, schedulate efficiently processing and data transfer activity on the Grid and meet QoS expectations. Yet most Grid application do not take into consideration the expected performance of the network resources they plan to use. In this paper we describe the effective use of a Grid Monitoring framework, whose measurements are used to introduce *netwrok aware* features in a legacy application.

We use GlueDomains, a network monitoring framework oriented to Grid infrastructures that measures a small (although possibly extensible) set of network parameters. Such framework works *off the shelf* with minimal administrative effort, is reliable, and has a negligible impact on system operation. The deployment covers a Metropolitan Grid infrastructure, aimed at supporting a data intensive eScience application. We describe a real use case consisting of bulk data trasfers during the operation of the Grid for the Virgo experiment.

1 Introduction and motivation

In the recent past Grid computing has introduced several improvements as regarding the software and the implementation of new algorithms, thus creating a stabler and more performant environment. So that, in the current state, the absence of a network management strategy appears to be one of the primary limits to improve the QoS level.

Considering the impact of quality of service issues, network performance can affect dramatically the job computation time in those applications that process large quantities of remote data, and is crucial whenever an application requires data replication. Moreover, a degraded connectivity may cause job failures, as in the case of an application scheduled in a short queue on processing resources, or if unexpected latencies cause timeout events.

Currently, in the main Grid deployments the network resource is considered as *pure facility* and the middleware components act agnostically with respect to network performance parameters, for instance during job scheduling, data transfer, and data replica process. This situation is justified by the challenges introduced by the introduction of network reservation policies in heterogeneous and distributed environment, and also by the absence of a simple and functional network monitoring framework.

In essence, if we introduce an abstract view of a Grid as the integration of computational, storage and network resources, the lack of information about network performance brings the overall system to work globally below the best effort threshold.

In this paper we describe an experiment, consisting in introducing *network awareness* in a simple data transfer application, using network measurements obtained from GlueDomains [6], a network monitoring framework created to support middleware services. GlueDomains offers a set of measurements useful for general operations, and we want to investigate their use for bandwith performance previsions. It is composed of a software environment for data producer and consumer, a basic set of measurements tools and an information model for data representation. GlueDomains proposes a simple and modular environment, very easy to deploy and configure, which can be easely expanded with custom measurement tools. The framework has been integrated in the INFN-GRID middleware (based on LCG-gLite) and deployed in a real testbed on the SCoPE Grid Infrastructure [12].

The rest of this article is organized as follow: in the next section we present the applicative context and some use cases and middleware services that can take advantage of the availability of network information. Next we summarize the concepts and tools underlaying the GlueDomains framework, introducing the information model used for the publication of the performed measurements. Finally we present the GlueDomains integration in the INFN-GRID middleware and the testbed as deployed: the architecture, the physical and logical topology and the available measurements.

2 The need for network measurements

The LHC experiments of high energy physisc, as ATLAS[1], or CMS[2], are dominated by a enormous data acquisition rate. The CMS[3], for instance, plans to manage an event rate of 100Hz, corresponding to a data rate of 100MB/s that must be analyzed. An other example is the Virgo Experiment for grativational waves detection [4] that is caracterized by a data acquisiton rate of 10MB/s 24h for day[8], that must be analysed and replicated on geographical scale from experiment Tier 0 to Tier 1, and to Tier 2 upon request.

The peculiar requirements that caracterize this kind of applications are challenging the the Grid development community to improve the middleware component in order do be *network aware*. Such feature is often discussed in literature with reference to meta-scheduling algorithms [11, 15, 9]. The related works propose new resource brokering strategies that consider availability of computational, storage and network resources, whose target is the optimization of appropriate cost estimators.

Another interesting topic is related with replica optimization. File catalog services like LFC [7] represent with a logical name a set of physical file locations, in order to select the more convenient replica when needed. Many strategies are proposed [14, 10, 5] that aim to improve the use of the file catalog services by quantifying network costs.

All the above experiences confirm the favorable impact of *network awareness* in Grid system, but the successful application of such paradigm relies on the availability of network measurements; a specialized infrastructure is required, that provides information about netwrok performance. In the next section we introduce GlueDomains, and motivate its adoption as a network monitoring framework.

3 The GlueDomains network monitoring architecture

Schematically, these are the reasons why the GlueDomains network monitoring infrastructure fits our scenario:

a. it has an extremely low footprint on deployed system resources;

b. it requires the introduction of a limited number of functionalities;

c. it is easy to deploy;

d. its scalability matches the present and future sizes of our system;

e. it works unattended for long periods of time, and is fault tolerant;

f. it can be easely reconfigured, for instance in case of join of a new sensor;

g. it can be adapted introducing custom network monitoring tools and publication engines.

The above points will be motivated in the course of the following summary of GlueDomains architecture.

The network monitoring layout consists of a partitioning of the network monitoring end-points (hereafter the *Edge Services*) into *Domains*: this significantly contributes to its scalability (point (d) above). Such partitioning takes into account the fact that monitoring will in fact aggregate and report *domain-to-domain* measurements: therefore domains should be designed so to include edge points that have a uniform connectivity with the rest of the network. Such assumption significantly simplifies the design of the network monitoring layout, while matching a wide range of use cases simply considering a DNS based domain partitioning.

Network monitoring is carried out through a number of network monitoring *tools*, whose location ensures that collected network measurements are significant; the hosts that support the activity of network monitoring tools are called *Theodolites*. One simplifying hypotheses is that they are co-located with one of the Edge Services in each Domain. Such option is valid since the network monitoring activity has a low footprint, and reduces the number of hardware devices dedicated to network monitoring (see point (b) above).

New tools can be easely added to those shipped with the GlueDomains package: one simply needs to create a new Perl class wrapping the tool in such a way that data are packaged in a hash that complies with an internal schema, derived from OGF Network Monitoring Working Group ones (see point (g) above).

The activity of the theodolites is coordinated by a centralized database : each theodolite periodically fetches the description of its activity from the database, possibly modifying its activity. The overall activity of the network monitoring infrastructure is represented by a number of network monitoring sessions. The UML description of a session is outlined in figure 1, and is rooted on the **Theodolite**.

Each **Theodolite** is composed of several monitoring **Sessions** subclassed into **Periodic** and **OnDemand**. The attributes of **Periodic** and **On demand** sessions allow the control of the monitoring activity: tool specific configurations are an attribute of a **Session**.

Note that each session addresses an instance of a tool running on the theodolite: unlike other more complex network monitoring architectures, we do not indicate the result of the monitoring activity, but its operational description.

The reconfiguration of the whole monitoring infrastructure does not need human intervention on theodolites, and is carried out simply uploading a new content in the database (see point (f above). For the same reason, the deployment of the network monitoring infrastructure is straightforward: the administrator of a theodolite that wants to join the monitoring infrastructure installs the package, and obtains the credentials that the theodolite will use to fetch the description of its activity (see point (c) above).

The task of the designer of the network monitoring activity is simplified by the provision of a tool that uses XML definitions of the monitoring tasks of each theodolite in order to maintain the MySQL database (see point (f) above).

Database updates are considered infrequent events, typically related to a change in the membership of theodolites. The database server offers a web service interface to the theodolites, in order to make the access to the database more flexible and is therefore transparent to the technology used to implement the database (see point (g) above). Fault tolerance features of the theodolite cope with transient failures of the server (see point (e) above).

The software component that manages the data transfer to the publication engine is another pluggable entity: data are presented to that plugin as Perl packed data on a pipe. The plugin is in charge to flush the pipe periodically

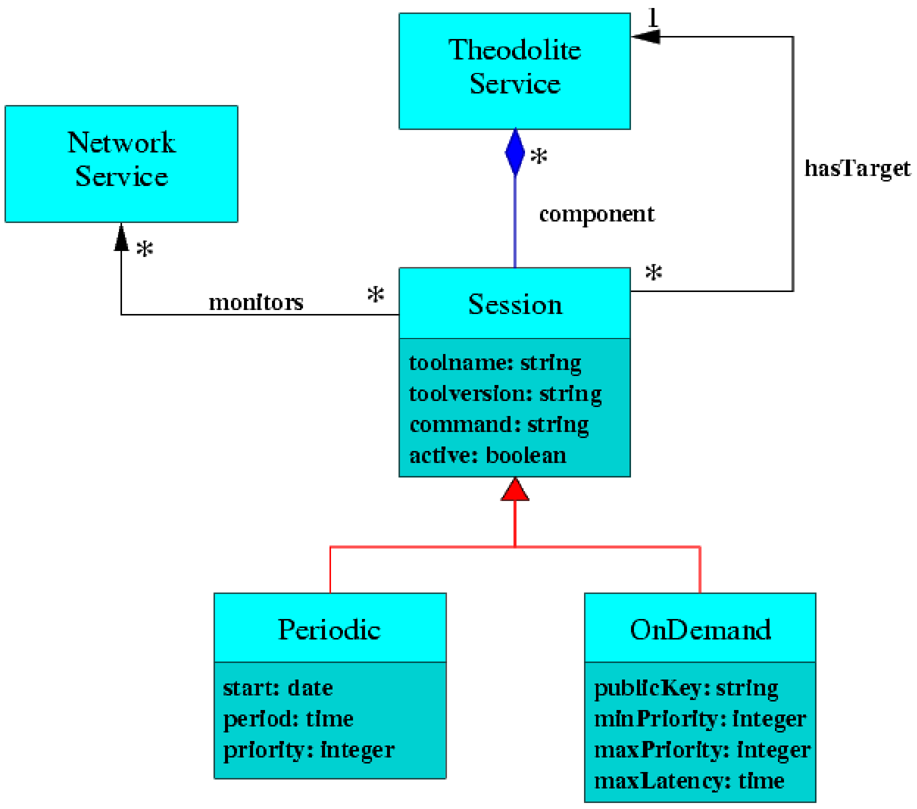


Figure 1: The UML diagram of the monitoring database

and forward the data to the publication engine of choice. Thus the theodolite, per se, has a very low footprint, since it does not support any complex operation or storage on collected data (see point (a) above). The plugin should operate according to the same principles.

In figure 2 we sketch a functional view according to the above description:

- the *topology database*, that stores and makes available to users the description of the Grid partitioning (e.g. which Domain contains a certain computing service);

- the *monitoring database*, that describes the planned monitoring activity in the Grid;

- the *production engine*, that stores and makes available the characteristics of Grid services and fabric (e.g., packet loss rate between computing and storage services);

- the *producers*, the Theodolites, that query the *monitoring database* and the *topology database* to configure their network monitoring activity. Observations are published through the *production engine*;

- the *consumers*, that find the domain they belong to, as well as those of other services of interest, querying the *topology database*. Observations are retrieved using the *production engine*.

As explained above, GlueDomains architecture is somewhat open: only the interface of some of its components is in fact specified.

In order to adapt such architecture to our purposes, we had to customize the three pluggable components: monitoring database, tools, and publication engine.

As for the monitoring database we reused an existing implementation based on a MySQL database. Although such database might be replicated, mainly for performance reasons, the scale of our experiment did not justify database replication.

The domain database is implemented using a set of rules that extract the domain from the DNS. Therefore there is not a real database, but a set of agreed syntactical rules.

We used GridICE for data publication, an MDS based tool extensively used in the INFN network. Therefore we reused the plugin implemented for the deployment of GlueDomains over part of the INFN network.

In figure 3 we see the relationships between the various software module, distributed on three physical entities: the database, the theodolite, and the publication engine.

According with such modular approach, the architecture of the *network monitoring host* is divided into a GlueDomains part, and an MDS specific part: in figure 3 they are separated by a dotted line, the GlueDomains part being on the left side.

The *GlueDomains* side is composed of a hierarchy of processes:

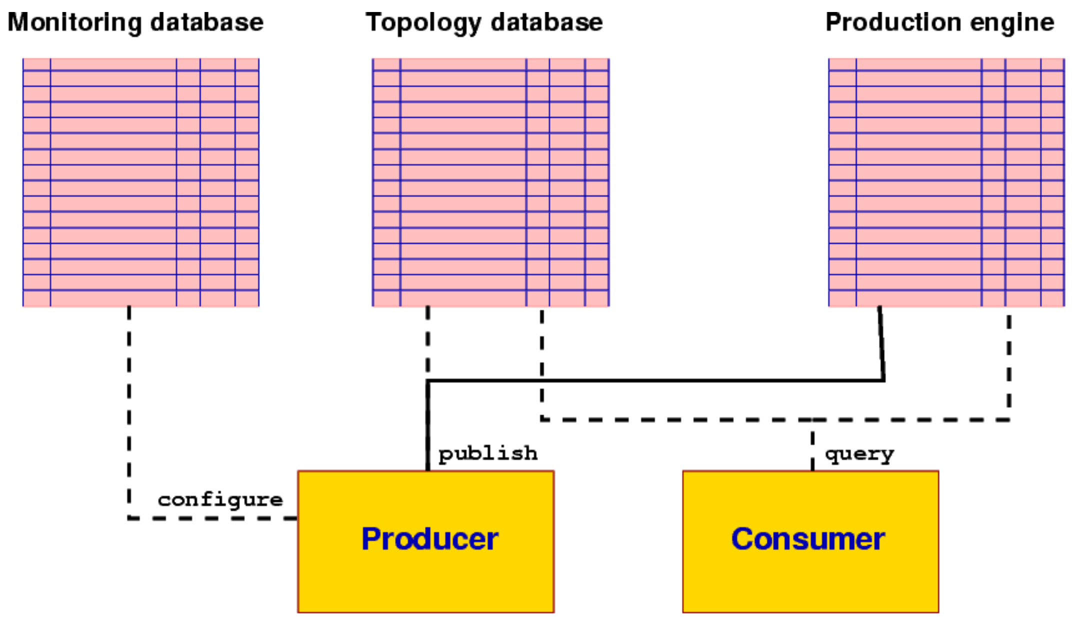


Figure 2: Modular architecture of a GIS: dashed lines represent read-only access

GlueDomains is a daemon process that controls the whole monitoring activity of the host. It spawns the processes that implement the *theodolite services*. The description of the *theodolite services* is obtained querying the *monitoring database* hosted by the *GlueDomains* server, each time a theodolite service is spawned. The query returns the list of all theodolite services that are associated with any of the IP addresses of the host.

Theodolite is a process that implements a *theodolite service*. It spawns — and re-spawns when needed — all monitoring sessions associated with a *theodolite service*. The description of all *sessions* associated with the *theodolite service* is retrieved from the *monitoring database*. The identifier of the monitored *Network Service* is retrieved from the *Topology Database*, given the identifier of the *theodolite* and of the *target* associated with the *session*. The *theodolite* may interact with the GIS adapter to initialize the publication of the observations for those *Network Services*.

Session is a process that implements a monitoring session. All parameters that configure a specific session are passed from the theodolite process, so that the session should not need to access the *monitoring* or *topology* databases. The session interacts with the GIS adapter to record the observations in the *production engine*.

We distinguish two kinds of *controls structures* for a session:

- a *periodic* control, like the usual pinger;

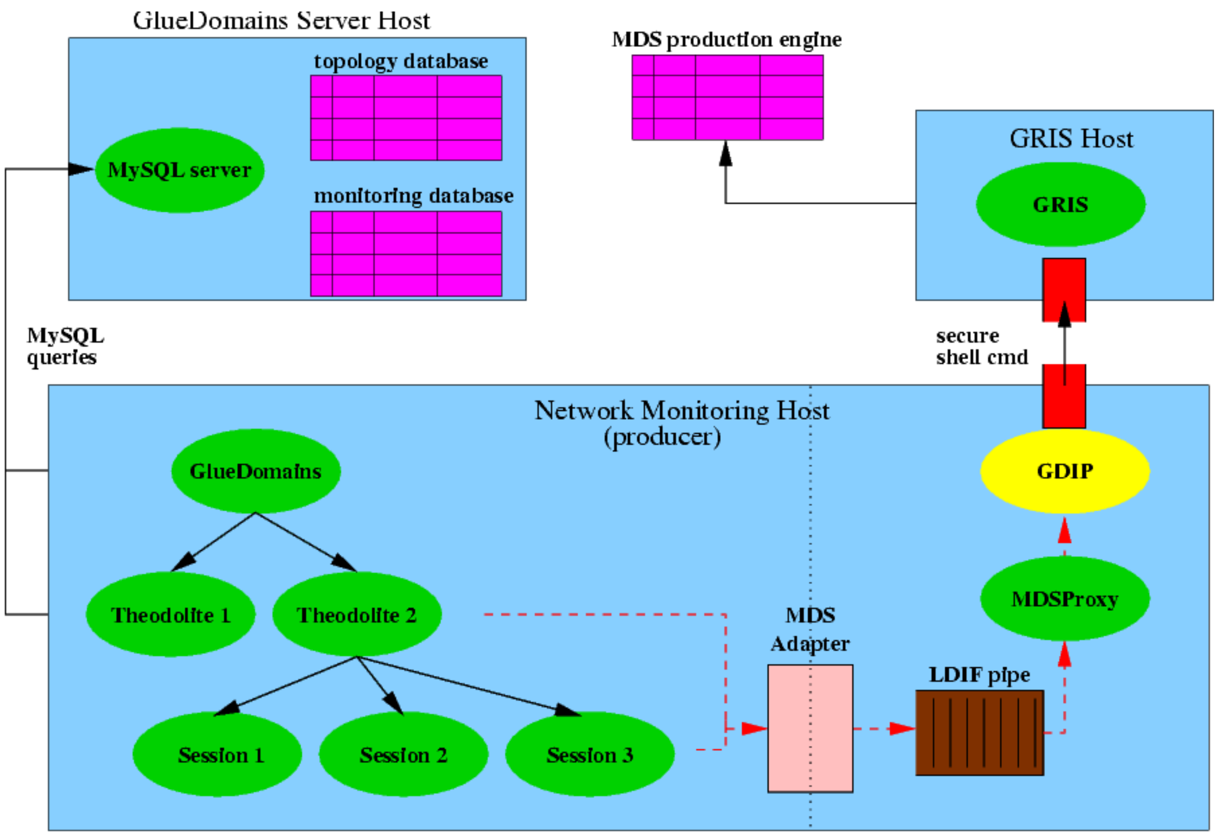


Figure 3: Prototype architecture (with MDS adapter)

- an *on demand* control, which receives a trigger from another process: for instance, an `iperf` server process, which bounces packets from a client. Such kind of control may also help the GRID-wide scheduling of monitoring activities, which is useful for expensive tests, like those used for bandwidth measurements.

The right part of the *network monitoring host* in figure 3 is MDS-specific. The design of the R-GMA specific part is simpler, and is omitted.

A flow of LDIF entries is generated by the functions provided by the MDS adaptor. Such flow is re-ordered and buffered by a *MDSproxy* process, which runs as a daemon. Periodically, the buffer is flushed by the GRIS host using the `GDIP` command, called through `ssh`.

3.1 Comparison with other Grid Information Systems

To show the potential of such architecture we compare it with the internal structure of the Network Weather Service [16], one of the more complete *Grid Information Systems. Producers* correspond to **sensor** hosts, each characterized by certain monitoring skills, that include network monitoring. The *monitoring database* consists of **nwsControl** objects, that define NWS **cliques** of sensors that perform mutual monitoring. The *production engine* consists of NWS **memories** that store data, and NWS **forecasters** that process this data to produce answers that match *consumer*'s needs, that are extrapolated from measurement **series** stored by **memories**. The system lacks a real *topology database*, which is in part implemented by **cliques**, in part relies on the mapping from **sensors** to IP addresses — from which we can infer that two sensors are in the same DNS domain, whatever this may mean. All functionalities and data are tightly packaged in a monolithic product, that can be controlled using simple Unix commands. Such a monolythic structure is the ral limit of NWS, which otherwise is consodered as a sort of paradigm in Network Monitoring Architecture.

Other *Grid Information Systems* tend to privilege the *production engine*, the database that contains the observations. This component is indeed critical for the availability of collected data, but does not help in the configuration of the measurement tools. Such tools for a complex architecture, and their activity needs to be carefully coordinated, in order to optimize the measurement activity, especially when it is based on active tools.

The NPM architecture [13] (developed as part of the gLite infrastructure by the European EGEE project) partially cope with this problem, by indexing the available data: access to data, stored in a relational database, is by way of a *mediator*, that is in charge of locating and preprocessing the data. In this way data, although collected in an uncoordinated way, is made available in an in a organized way.

Although such approach helps introducing a structure in collected data, it does not help in avoiding, for instance, the collection of redundant or useless data. In order to do this, one has to introduce some sort of management of the

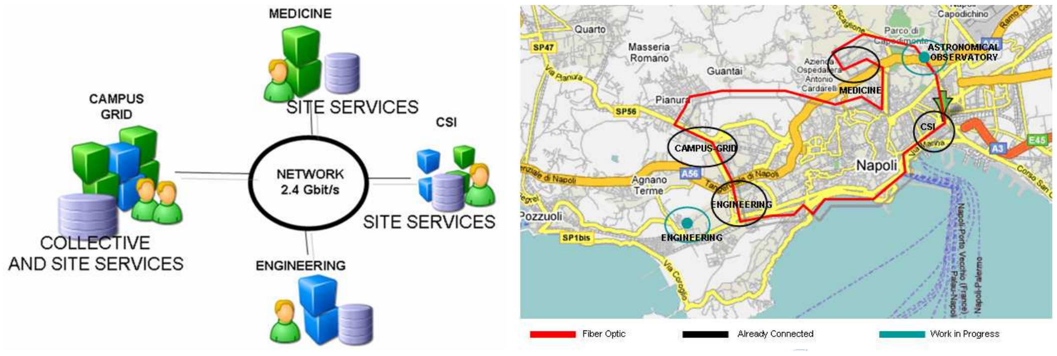


Figure 4: At left the current fiber optic MAN ring of the University Federico II. In the rigth picture a map of the city of Naples with the main SCoPE sites. In green the departments not still connected at the Federico II netowrk infrastructure

overall monitoring activity. The NWS architecture introduces a specialized component to this purpose, that exactly represents such functionality EXPLAIN.

The GlueDomains *server* reproduces a similar function, although its design improves performance and reliability with respect to the the NWS XXXX. We summarized he functionality of the server at page 4.

One distinguishing feature of GlueDomains is that it does not contain a publication engine of its own, but relies on an already existing one: we considered that data publication as a separated issue, that is preferably implemented orthogonally, possibly by a different team. Therefore the theodolite implementation offers an interface for a specialized plugin, tailored for a certain type of publication engine. In our testbed we used a GridICE plugin.

4 A testbed deployement on Metropoltan Area Grid

4.1 The SCoPE project

The S.Co.P.E. (Italian acronymic for high Performance, Cooperative and distributed System for scientific Elaboration) [12]is a research project that aims at developing several applications in the field of fundamental research, which is one of its strategic objectives. The main spin off is the implementation of an open and multidisciplinar Grid infrastructure between the departments of University Federico II, distributed in Metropolitan scale in the city of Naples.

The S.Co.P.E. Architecture provides the unification of all the main computational and storage resources already available in the sites that participate at the project, using the Grid Paradigm supported by INFN-GRID distribution.

The computing infrastructure connectivity is supported by the metropolitan

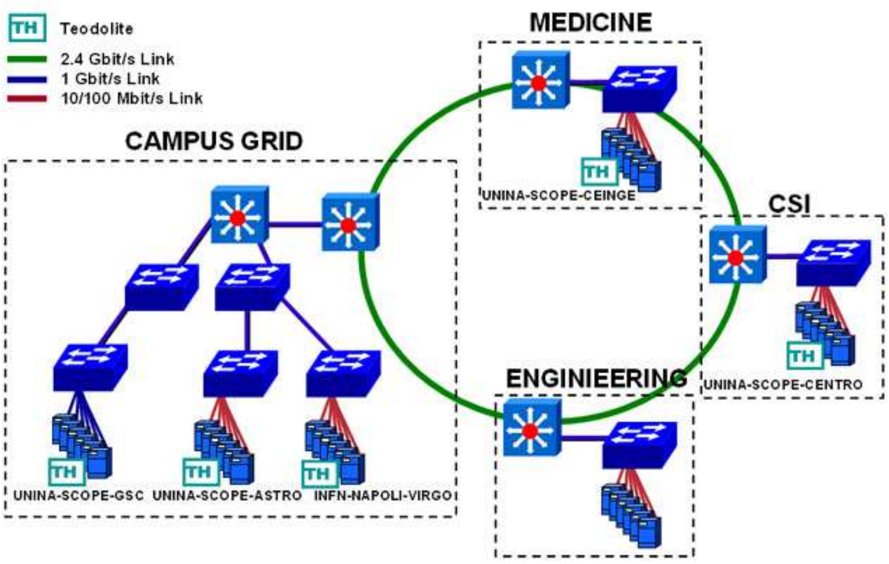


Figure 5: The testbed network topology with the link speed and the theodolite locations

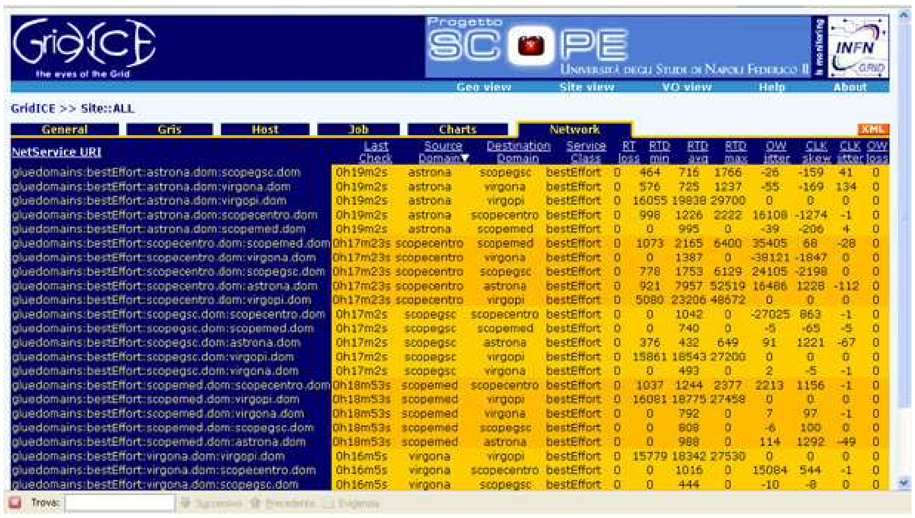
fiber optic ring that connects at 2.4Gb/s the 4 main sites of the University: Medicine, Engineering, C.S.I. (Center of Information Services) and the Campus of Monte Sant'Angelo, that accommodates the science faculty departments and are already organized in a Grid infrastructure over a fiber optic LAN at campus level (THE CAMPUS GRID). The MAN will be extended by the end of 2008 with a wireless bridge towards the Astronomical Observatory of Capodimonte and a wired link to the detached engineering site (see figure 4).

All the peripheral sites share their resources by offering the basic Grid site services of the INFN-GRID middleware: Computing Element, Worker Node and Storage Element based on DPM. The collective services, are centralized in a new computing center open in the Campus Grid that offer Services Discovery, Resource Brokering, File Catalog, Grid Monitoring and graphical user interface.

Most of the research topics covered by the University Federico II involve the use of different software packages and applications, each exhibiting distinguished requirements on the Grid services. This framework promotes the research on Grid computing, and offers an excellent and complete testbed in which new Grid services can be deployed and evaluated.

4.2 The testbed deployment

The GlueDomains testbed, has been deployed on five sites of the SCoPE infrastructure, interconnected in a switched network with different end-to-end performances, as shown in the figure 5.



NetService URI	Last Check	Source Domain▼	Destination Domain	Service Class	RT loss	RTD min	RTD avg	RTD max	OW jitter	CLK skew	CLK jitter	OW loss
gluedomains:bestEffort:astrona.dom:scopegsc.dom	0h19m2s	astrona	scopegsc	bestEffort	0	464	716	1766	-26	-159	41	0
gluedomains:bestEffort:astrona.dom:virgona.dom	0h19m2s	astrona	virgona	bestEffort	0	576	725	1237	-55	-169	134	0
gluedomains:bestEffort:astrona.dom:virgopi.dom	0h19m2s	astrona	virgopi	bestEffort	0	16055	19838	29700	0	0	0	0
gluedomains:bestEffort:astrona.dom:scopecentro.dom	0h19m2s	astrona	scopecentro	bestEffort	0	998	1226	2222	16108	-1274	-1	0
gluedomains:bestEffort:astrona.dom:scopemed.dom	0h19m2s	astrona	scopemed	bestEffort	0	0	995	0	-39	-206	4	0
gluedomains:bestEffort:scopecentro.dom:scopemed.dom	0h17m23s	scopecentro	scopemed	bestEffort	0	1073	2165	6400	35405	68	-28	0
gluedomains:bestEffort:scopecentro.dom:virgona.dom	0h17m23s	scopecentro	virgona	bestEffort	0	0	1387	0	-38121	-1847	0	0
gluedomains:bestEffort:scopecentro.dom:scopegsc.dom	0h17m23s	scopecentro	scopegsc	bestEffort	0	778	1753	6129	24105	-2198	0	0
gluedomains:bestEffort:scopecentro.dom:astrona.dom	0h17m23s	scopecentro	astrona	bestEffort	0	921	7957	52519	16486	1228	-112	0
gluedomains:bestEffort:scopecentro.dom:virgopi.dom	0h17m23s	scopecentro	virgopi	bestEffort	0	5080	23206	48672	0	0	0	0
gluedomains:bestEffort:scopegsc.dom:scopecentro.dom	0h17m2s	scopegsc	scopecentro	bestEffort	0	0	1042	0	-27025	863	-1	0
gluedomains:bestEffort:scopegsc.dom:scopemed.dom	0h17m2s	scopegsc	scopemed	bestEffort	0	0	740	0	-5	-65	-5	0
gluedomains:bestEffort:scopegsc.dom:astrona.dom	0h17m2s	scopegsc	astrona	bestEffort	0	376	432	649	91	1221	-67	0
gluedomains:bestEffort:scopegsc.dom:virgopi.dom	0h17m2s	scopegsc	virgopi	bestEffort	0	15861	18543	27200	0	0	0	0
gluedomains:bestEffort:scopegsc.dom:virgona.dom	0h17m2s	scopegsc	virgona	bestEffort	0	0	493	0	2	-5	-1	0
gluedomains:bestEffort:scopemed.dom:scopecentro.dom	0h18m53s	scopemed	scopecentro	bestEffort	0	1037	1244	2377	2213	1156	-1	0
gluedomains:bestEffort:scopemed.dom:virgopi.dom	0h18m53s	scopemed	virgopi	bestEffort	0	16081	18775	27458	0	0	0	0
gluedomains:bestEffort:scopemed.dom:virgona.dom	0h18m53s	scopemed	virgona	bestEffort	0	0	792	0	7	97	-1	0
gluedomains:bestEffort:scopemed.dom:scopegsc.dom	0h18m53s	scopemed	scopegsc	bestEffort	0	0	808	0	-6	100	0	0
gluedomains:bestEffort:scopemed.dom:astrona.dom	0h18m53s	scopemed	astrona	bestEffort	0	0	988	0	114	1292	-49	0
gluedomains:bestEffort:virgona.dom:virgopi.dom	0h16m5s	virgona	virgopi	bestEffort	0	15779	18342	27530	0	0	0	0
gluedomains:bestEffort:virgona.dom:scopecentro.dom	0h16m5s	virgona	scopecentro	bestEffort	0	0	1016	0	15084	544	-1	0
gluedomains:bestEffort:virgona.dom:scopegsc.dom	0h16m5s	virgona	scopegsc	bestEffort	0	0	444	0	-10	-8	0	0

Figure 6: The SCoPE GridICE web page show the measurement performed among all the possible couple of theodolites

The GlueDomains Server component is supported by a dedicated node of the collective services farm, placed in the Campus Grid site. This server manages the central database used to configure the measurement topology and the network monitoring activity of each theodolite.

In order to monitor the network performance of our Grid, five theodolites are installed in the following sites:

- UNINA-SCOPE-GSC in the Campus Grid Site
- UNINA-SCOPE-ASTRO in the Campus Grid Site
- INFN-NAPOLI-VIRGO in the Campus Grid Site
- UNINA-SCOPE-CENTRO in the Center Site
- UNINA-SCOPE-CEINGE in the Medicine Site

They are installed in the Storage Elements and configured in a full mesh, making available Rondtrip Time, Packet Loss Rate and One-Way Jitter for each domain-to-domain path. The data produced by the GlueDomains sensors are periodically uploaded and published through the GridICE web interface (see figure 6) which is used also to check theodolite operation.

5 The data transfer problem

The SCoPE infrastructure exports the storage via an SRM interface. Each file, distributed on several storage elements, is registered in a central logical file catalog (LFC) and the applications access to the physical replicas by querying this service. In absence of a replica optimization service [14], when an user or an application asks for a file, addressed using its *logical name*, the LFC catalog returns a physical location selected *randomly* among the different replicas that correspond at the same logical name: this way is clearly sub-optimal. In the rest of this paper we show some prelimnary results about a real use case, and we compare the performances obtained using the *random* strategy applied by LFC with respect to the use of a cost driven selection obtained from data provided by GlueDomains.

5.1 The Framework

The VIRGO comunity at University Federico II is in charge of analyzing the data produced by the Virgo interferometer, located in Cascina (PISA). To this purpose, fresh data continuously flow from the interferometer to the Virgo Grid. Data acquisition is followed by a synchronization phase, in which raw data are replicated in the different storage elements of the SCoPE infrastructure, in order to optimize their parallel processing on distinct computing resources.

The network traffic created during data synchronization and distribution affects the general Grid performance, which reflects on the network performance observed by the other users of the Grid.

In [11] a meta-scheduler network-aware is introduced, called DIANA, for data intensive applications. The scheduling algorithm used by DIANA is based on the estimate of the cost of computation and of data transfers. This algorithm, defined in the framework of the CMS experiment, provides a cost function that can be used to select site from data set are downloaded during network congestion.

The model proposed in the DIANA framework estimates the network cost using very basic network measurements (Round Trip Time (RTT), Jitter, Packet Loss and the Maximum Segment Size(MSS)), through the following equation:

$$NetCost \propto \frac{Losses}{Bandwidth}$$

where

$$Losses = RTT \times W1 + Loss \times W2 + Jitter \times W3$$

and

$$Bandwidth < \frac{MSS}{RTT} \times \frac{1}{\sqrt{Loss}}$$

This model as been used to perform some preliminary tests, with the purpose of evaluating the effective use of network measurements as provided by GlueDomains.

```
lcg-cp --verbose lfn:/grid/scope/gluetest.txt srm://grid002.ceinge.unina.it:8446/
dpm/ceinge.unina.it/home/scope/gluetest97.txt

Using grid catalog type: lfc
Using grid catalog : scopelfc01.dsf.unina.it
VO name: (null)
Source URL: lfn:/grid/scope/gluetest.txt
File size: 1246076580
Source URL for copy: gsiftp://scopese01.dsf.unina.it/scopese01.dsf.unina.it:/scope/
scope/2007-12-02/gluetest.txt.1530.0
Destination URL: srm://grid002.ceinge.unina.it:8446/dpm/ceinge.unina
.it/home/scope/gluetest97.txt
# streams: 1
# set timeout to  0 (seconds)
   1241186304 bytes   1744.02 KB/sec avg   2598.40 KB/sec inst
Transfer took 701940 ms
```

Figure 7: Network performance without replica selection

5.2 Experimental results

The experiment consists in downloading in the storage element of the Medicine site UNINA-SCOPE-CEINGE a set of 1.2Gb size files which is replicated in the sites INFN-VIRGO-NAPOLI, UNINA-SCOPE-GSC and UNINA-SCOPE-CENTRO.

The files are registered in the SCoPE logical file catalog: this allows to have an single namespace for all the replicas distributed along different storage elements.

The tests are deliberately run during an intensive sincronization activity among UNINA-SCOPE-GSC, UNINA-SCOPE-CENTRO and UNINA-SCOPE-ASTRO sites, and unfold in two phases:

- Replication of 10 files on the UNINA-SCOPE-CEINGE site by using the lcg-utils tools and the logical file name.
- Replication of 10 files on the UNINA-SCOPE-CEINGE site by selecting the Strage using the above *NetCost*.

During the first test sequence, the `lcg-cp` command, after querying the lfc catalog, start downloading data from `scopese01.dsf.unina.it`, which frequently happens to be the busiest host. Therefore the global performance degrades and each file takes about 700s to be replicated on the UNINA-SCOPE-CEINGE site, with an average rate of 1.800 KB/sec vs the theoretical 12.000 KB/s. Figure 7 reports a typical session.

In the second sequence of tests, data are downloaded using the SURL address and selecting the location by computing the network cost in advance. Starting from the measurements provided by the UNINA-SCOPE-CEINGE theodolite, right before the download we can estimate the Network Cost. A typical session is in figure 8.

The Network cost shows that the best storage element from which to download the replica is INFN-NAPOLI-VIRGO, since the SCOPE-UNINA-GSC NetCost is sensitive to the relevant packet loss rate. The data transfer rate shows

SITE	MSS	RTT (sec)	JITTER	LOSS	NetCost
INFN-NAPOLI-VIRGO	1024	0.890	174	0	0.15
SCOPE-UNINA-ASTRO	1024	2.993	142	6250	1447
SCOPE-UNINA-CENTRO	1024	2.442	2314	0	5.5
SCOPE-UNINA-GSC	1024	2.308	12	40186	18163

Figure 8: Available measurements before the download

```
lcg-cp --verbose srm://virgo-se.na.infn.it:8446/dpm/na.infn.it/home/scope
/gluetest.txt srm://grid002.ceinge.unina.it:8446/dpm/ceinge.unina.it/home/scope/gluetest94.txt

Source URL: srm://virgo-se.na.infn.it:8446/dpm/na.infn.it/home/scope/gluetest.txt
File size: 1246076580
Source URL for copy: gsiftp://virgo-se.na.infn.it/virgo-se.na.infn.it:/flatfiles/SE00/virgo/scope/
2007-12-02/gluetest.txt.20.0
Destination URL: srm://grid002.ceinge.unina.it:8446/dpm/ceinge.unina.it/home/scope/gluetest94.txt
# streams: 1
# set timeout to  0 (seconds)
   1196621824 bytes  11129.29 KB/sec avg  11200.00 KB/sec inst
Transfer took 111090 ms
```

Figure 9: Network performance without replica selection based on network performance

a significant improvement in terms of the average transfer time. In table 9 we show the output of a typical lcg-cp session, and in figure 10 we summarize the results of the performed tests time needed to copy the file from ??

6 Conclusions and future work

It is a widespread sentiment that the introduction *network awareness* in the implementation of Grid services promises important performance improvements, for which an effective, reliable network monitoring system is just a premise. We have successfully applied such concept to the management of a typical eScience task, in the frame of a relevant scientific experiment, Virgo.

From such experience we conclude that one of the characteristics of the network monitoring infrastructure is its flexibility and reliability; its deployment and maintenance costs should be negligible compared with the management of the target activity. Measurements should be presented with an application

TEST	TOTAL TIME	TOTAL SIZE	Time Mean/file	BAND
TEST SERIE 1	116 min	12 GB	701 sec	14Mbs
TEST SERIE 2	20 min	12GB	120 sec	81Mbs

Figure 10: Comparison between transfer performance

friendly interface, to simplify the upgrade of legacy applications.

In this paper we targeted the use of simple network measurements, with the purpose of verifying the effectiveness of simple cost estimators. The results we obtained are encouraging, and we plan to investigate the possibility of acquiring bandwith measurements, in order to improve decision accuracy.

References

[1] ATLAS web page. http://atlas.web.cern.ch/Atlas/index.html.

[2] CMA production. http://cmsdoc.cern.ch/cms/production.

[3] *Data analysis techniques for high-energy physics.* Cambridge University Press.

[4] Webpage of the Virgo experiment. http://www.virgo.infn.it.

[5] Design of a replica optimisation framework. Technical Report DataGrid-02-TED-021215, DATAGRID Project, December 2002.

[6] Sergio Andreozzi, Augusto Ciuffoletti, Antonia Ghiselli, and Cristina Vistoli. Monitoring the connectivity of a grid. In *2nd Workshop on Middleware for Grid Computing*, pages 47–51, Toronto, Canada 2004.

[7] J-P. Baud, J. Casey, S. Lemaitre, C. Nicholson, D. Smith, and G. Stewart. Lcg data management: From edg to egee. In *Proceedings of "UK eScience All Hands Meeting"*, Nottingham, UK, 2005.

[8] P. Canitrot, L. Milano, and A. Vicer. Computational costs for coalescing binaries detection in VIRGO using matched filters. Technical Report VIR-NOT-PIS-1390-149, Virgo Project.

[9] Hai Jin, Xuanhua Shi, Weizhong Qiang, and Deqing Zou. An adaptive meta-scheduler for data-intensive applications. *International Journal of Grid and Utility Computing*, 1(1):32 – 37, 2005.

[10] P. Kunszt, E. Laure, H. Stockinger, and K. Stockinger. Advanced replica management with Reptor. In *Lecture Notes in Computer Science*, volume 3019. Springer Berlin / Heidelberg, 2004.

[11] R. McClatchey, A. Anjum, H. Stockinger, I. Ali, A. Willers, and M. Thomas. Data intensive and network aware (DIANA) grid scheduling. *Journal of Grid Computing*, 5(1):43–64, 2007.

[12] S. Pardi, L. Merola, G. Marrucci, and G. Russo. The SCoPE project. In *3rd International Conference on Cybernetics and Information Technologies*, Orlando, Florida, USA, July 2006.

[13] Alistair Phipps. Network performance monitoring architecture. Technical Report EGEE-JRA4-TEC-606702-NPM NMWG Model Design, JRA4 Design Team, September 2005.

[14] K. Stockinger, H. Stockinger, L. Dutka, R. Slota, D. Nikolow, and J. Kitowski. Access cost estimation for unified grid storage systems. In IEEE Computer Society Press, editor, *4th Intl. Workshop Grid2003*, Phoenix/Arizona, USA, 2003.

[15] B. Volckaert, P. Thysebaert, M. De Leenheer, F. F. De Turck, B. Dhoedt, and P. Demeester. Network aware scheduling in grids. In *Proc. of the 9th European Conference on Networks and Optical Communications*, Eindhoven, The Netherlands, Jun 2004.

[16] R. Wolski, N.T. Spring, and J. Hayes. The Network Weather Service: a distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems,*, 15(5):757–768, October 1999.