Geographical failover for the EGEE-WLCG grid collaboration tools

To cite this article: A Cavalli et al 2008 J. Phys.: Conf. Ser. 119 062022

View the article online for updates and enhancements.

Related content

- GOCDB, a topology repository for a worldwide grid infrastructure
  Gilles Mathieu, Dr Andrew Richards, Dr John Gordon et al.

- Ten years of European Grids: What have we learnt?
  Stephen Burke

- Grid computing for UK particle physics
  J Coles

Recent citations

- A lightweight high availability strategy for Atlas LCG File Catalogs
  Barbara Martelli et al

- E. Vilucchi et al
Geographical failover for the EGEE-WLCG Grid collaboration tools

Alessandro CAVALLI\textsuperscript{1,4}, Alfredo PAGANO\textsuperscript{1,5}, Osman AIDEL\textsuperscript{2,6}, Cyril L’ORPHELIN\textsuperscript{2,7}, Gilles MATHIEU\textsuperscript{2,8} and Rafal LICHWALA\textsuperscript{3,9}

\textsuperscript{1}INFN/CNAF, Bologna, Italy
\textsuperscript{2}IN2P3/CNRS Computing Centre, Lyon, France
\textsuperscript{3}PSNC, Poznan, Poland

E-mail:
\textsuperscript{4}alessandro.cavalli@cnaf.infn.it
\textsuperscript{5}alfredo.pagano@cnaf.infn.it,
\textsuperscript{6}oaidel@cc.in2p3.fr,
\textsuperscript{7}cyril.lorphelin@cc.in2p3.fr,
\textsuperscript{8}gilles.mathieu@in2p3.fr,
\textsuperscript{9}syriusz@man.poznan.pl

Abstract. Worldwide grid projects such as EGEE and WLCG need services with high availability, not only for grid usage, but also for associated operations. In particular, tools used for daily activities or operational procedures are considered to be critical. The operations activity of EGEE relies on many tools developed by teams from different countries. For each tool, only one instance was originally deployed, thus representing single points of failure. In this context, the EGEE failover problem was solved by replicating tools at different sites, using specific DNS features to automatically failover to a given service. A new domain for grid operations (gridops.org) was registered and deployed following DNS testing in a virtual machine (vm) environment using nsupdate, NS/zone configuration and fast TTLs. In addition, replication of databases, web servers and web services have been tested and configured. In this paper, we describe the technical mechanism used in our approach to replication and failover. We also describe the procedure implemented for the EGEE/WLCG CIC Operations Portal use case. Furthermore, we present the interest in failover procedures in the context of other grid projects and grid services. Future plans for improvements of the procedures are also described.
1. Failover System

1.1. Purpose and context
Maintaining high levels of access to the information across heterogeneous environments without compromising a quality user experience, can challenge any IT organization.

The Grid infrastructure deployed and maintained through the EGEE [1] and LCG [2] projects aims to provide resources and services on a 24/7 basis. Ensuring such availability implies that not only the infrastructure itself, but also the operational and management tools, are always available. As described in [3], these tools are essential in daily operations, because they provide the Grid monitoring and problem tracking features. In addition they store, manage and present the information required by site-administrators, Virtual Organizations and other specific operational task forces.

Namely, these operational tools are: EGEE Grid User and Operation Support (GGUS) [4][5], the CIC Operations Portal [6][7], the Service Availability Monitoring framework (SAM) [8], the Grid Operations Centre DataBase (GOCDB) [9], and a range of widely used tools such as SAM Admin’s Portal (SAMAP) [10], Gstat [11] or GridIce [12].

The growing need of high availability for these tools led to the creation, in January 2006, of a “failover” working group, chaired by the Grid operations team at CNAF/INFN, Italy. The mandate of this working group is to propose, implement and document well-established procedures to limit service outages for the EGEE operational tools. Among the activities of the group, the analysis of the correlations among the different tools, helped to highlight the interdependences of each service. The result of this work culminated with a detailed table named “Operations Tools Map” [13].

Failover is one basic fault-tolerance function of mission-critical systems that are providing a constantly accessible service. Its purpose is to redirect requests from the failing system to a backup, that mimics the operations of the first one. The whole process is supposed to happen automatically and transparently to the end user. Failover or switchover solutions are widely used whenever and wherever high availability is needed. Adopting such an approach is neither new nor original, yet, the real challenges of this work are its wide scope, the number of teams involved, and the geographically distributed nature of both the Grid and the related operational tools.
The level of availability requested from LCG/EGEE was assessed in 99% of the time, corresponding specifically to no more than 87 hours of downtime in a given year. The achievement of such a level of service can be troubled by failures or even planned events, as shown in Table 1 statistic (source: [14]).

| Planned                      | Outage Event Type         | Distribution |
|------------------------------|---------------------------|--------------|
| Software - Upgrade/Configuration | 22%                      |
| Hardware - Upgrade/Configuration | 9%                       |

| Unplanned                          | Outage Event Type         | Distribution |
|------------------------------------|---------------------------|--------------|
| Software Failure - Control Plane   | 15%                      |
| Software Failure - Data Plane      | 5%                       |
| Software Failure – Other           | 5%                       |
| Hardware Failure - I/O Cards       | 7%                       |
| Control/System                     | 7%                       |
| Link Failure                       | 20%                      |
| Power Outage                       | 1%                       |
| Other/Unknown                      | 9%                       |
| Total                              | 100%                     |

Even if more related to networking services failures, this list is interesting to evaluate the problems from the site perspective. Sites in most cases are working to remove the single points of failures (SPOF), therefore our EGEE failover activity is not concentrated on this local task; it is left to the sites’ good sense and method. On the contrary, the activity is focused on the remaining slice of probability that local measures are not successful, not strong enough or that, in some extreme case (of very big outages), they cannot provide a remedy at all. We thought that with a geographical failover approach we can compensate this portion of failures.

With this goal, we have investigated the following three technical solutions: DNS (Domain Name System) name re-mapping, GSLB (Global Server Load Balancing) and a failover approach based on BGP (Border Gateway Protocol) [14][16].

GSLB is an available technique to monitor the health of a set of services, with the purpose to switch the users to different data centres depending from heavy load, failure or better round trip time. It can be provided by additional software on top of the most common DNS servers, as well as by dedicated hardware available from several IT companies. BGP is the core routing protocol in use among Internet Service Providers, but unfortunately it is indicated to provide failover and load balancing for large networks and entire data centres. In fact it is technically not possible to use BGP to advertise single hosts. That is: solutions based on BGP failover and GSLB are too complex, too expensive or not oriented to our target.

The choice fell on a standard DNS approach, as discussed in the following section.
2. The adopted Technical Solution

2.1. DNS based Geographical Failover

2.1.1. Introduction

The benefits of working with DNS can be summarized in three points:

- DNS is well documented and with a quick learning curve
- Almost every institute partner of EGEE project has expertise on DNS
- DNS is known as a very stable and successful distributed service

The DNS approach consists of mapping the service name to one or more destinations and update this mapping whenever some of the destination services are detected to be in failure. The mechanism is better clarified by taking, for example the CIC portal, one of the EGEE operational tools.

The CIC portal master instance is physically in France, while the backup replica is maintained in Italy. These instances are reachable respectively as \texttt{cic.gridops.org} (IP1) and \texttt{cic2.gridops.org} (IP2). When the main instance is no longer reachable, acting on DNS, it is possible to point \texttt{cic.gridops.org} to IP2. If well performed, this operation guarantees the service continuity and in the same time it is completely transparent to the final user. It is implied that this mechanism needs, as an essential component, a proper data-synchronization procedure.

2.1.2. The new “gridops.org” domain

The domain names of existing EGEE operational tools were individually registered by their respective institute. A scenario with non homogeneous names as \texttt{cic.in2p3.fr} for the CIC portal, \texttt{goc.grid.sinica.edu.tw} for Gstat, needs an additional layer. This layer was provided by the registration of a new domain: “gridops.org”.

This stratagem allows:

- an easier renaming of all the involved services;
- a transparent alias level upon the real service names and IPs;
- the possibility to quickly remap these aliases thanks to the very short TTLs.

The new domain has been registered by the INFN-CNAF institute, and managed on a CNAF DNS server, while a slave server has been provided and set up by the GRNET institute [17], which is also involved in the EGEE project activities. The table 2 clarifies the adopted schema. The complete list is available on the failover web pages [18].

| Primary instances | Backup instances |
|-------------------|------------------|
| cic.gridops.org (In2p3, France) | cic2.gridops.org (INFN, Italy) |
| goc.gridops.org (RAL, UK) | goc2.gridops.org (INFN, Italy) |
| gstat.gridops.org (Sinica, Taiwan) | gstat2.gridops.org (INFN, Italy) |
| … | … |

The DNS “zones”, where the configuration of every domain name is located, can be managed directly on the DNS server machine or through \texttt{nsupdate} program. \texttt{Nsupdate}, is an utility available for the BIND DNS software [19], it supports cryptographic authentication allowing the dynamic and remote insertion and deletion of DNS records. It is available on main Linux distributions, as well as the other main unix clones and Windows.

2.1.3. The use of the new domain name

From the table 2, consider two possible conditions:

1. the primary and backup instances are both available. They can be queried by the users;
2. the primary instance service fails, but the backup instance service is available. In this case the DNS is operated to point the primary instance name to the backup service.

The switch operated in condition 2, is performed changing the CNAME DNS record to point to another host name, which corresponds to the backup service (see RFC 1034 and RFC 1035 for the DNS basic concepts, that cannot be discussed here).

This new domain approach has created some technical issues. First of all, absolute hyperlinks are incompatible with a DNS remapping. A partial rewriting of the EGEE operational tools was a constraint but this was solved in a short amount of time. Furthermore, the services running on SSL using an X509 certificate are subject to produce warnings on every web browser, if contacted by a name which is not the same host name the certificate has been produced for. This has been solved by producing certificates with alternative names, adding the names under gridops.org (most Certification Authorities should be able to provide this (as INFN CA [20], for example). Another technical issue could be for example the one of HTTP sessions being broken by a DNS switch, but our scenario is not as critical as the case of bank, e-commerce, medical or military procedures, so this problem has less priority for us.

2.1.4. DNS downsides
The biggest downside of this approach is related to the caching issue. The TTL (Time To Live) tells a DNS server how long it is allowed to cache a DNS record. Caching is a standard feature in the DNS. When one DNS server queries another DNS server for a record, it is allowed to store the answer in its local memory cache for the number of seconds specified by the TTL value. Only after the TTL has expired will the caching server repeat the query to get a fresh copy of the DNS record from the authoritative server. For the needs of this failover activity, the TTL of gridops.org DNS records has been configured to 60 seconds. It means that our wish is to be able to switch from a failing system to its replica in no more than this amount of time. But this may be not always as easy as desired.

It may happen that some name servers do not honor all TTLs of resource records that are cached, nevertheless this is more frequent for commercial providers than within our research networks.

The caching issue must also be considered at browser level as well as Operating System level. In general, by default, the three main operation systems, that is Microsoft Windows, Mac OS and Linux, are not affected by TTL caching issues [21] [22].

Concerning the web browsers, the behavior between the widely used Microsoft Internet Explorer and the open source Mozilla Firefox is quite different. In detail, Mozilla Firefox is very conservative with its value of 60 seconds [23], while Microsoft Internet Explorer, with a 30 minutes value, attempts to save many more queries [24]. This of course only affects users who have visited the web site immediately before a DNS update. New visitors, or returning visitors who closed their browser or waited more than 30 minutes since their last visit, are not affected. Furthermore, it is possible to adjust the length of time Microsoft Explorer caches DNS records, by updating a registry setting on the client machine. But this is of course neither practical nor user-friendly. Thirty minutes is a considerable time if related with service availability. Nonetheless, considering our goal of a 99% uptime/year and considering the downtime statistics of the last two years, this can be tolerated at this implementation stage.

3. Use cases
3.1. CIC Portal
The CIC Operations Portal [6] groups together, on a single web interface, several tools that are essential in EGEE, for the operational management, Grid monitoring and trouble-shooting. For this reason some recovery procedure was needed to be provided as soon as possible: planned or unexpected service outages could break the continuity of several daily activities.
As described in [7], the CIC portal is based on three distinct components:

- The web module, hosted on a web cluster, and mainly consisting of PHP code, html pages and CSS style sheets;
- The database (Oracle);
- The data processing system, based on the Lavoisier web service [25] [25].

The failover concern played a great role in the way this architecture has been designed: the clear separation between the different modules implies indeed an easier replication work. All modules master instances are hosted and running at IN2P3/CNRS computing centre, located in Lyon, France, while replicas have been set up at CNAF/INFN, Bologna, Italy.

The first occasion where this solution was actually used was one big service interruption for maintenance at the IN2P3 Lyon computing centre, planned for December 2006. A switchover to the backup instance, operated by system administrators on both sides, eventually reduced this outage from several days, as originally planned, to a few hours. The newborn CNAF instance provided the service for one entire week. Afterwards, the CIC portal users were benefitting from this failover setup in two other cases:

- because of a lately planned Oracle maintenance at IN2P3, between 01/02/2007 and 13/02/2007;
- to avoid possible interruptions during a hardware maintenance to come at IN2P3, between 5/06/2007 and 28/06/2007.

3.1.1. Web module Replication.

Two prerequisites for the replication work to proceed smoothly were:

- give a well defined structure to the code tree and provide it through a versioning system;
- make the code portable, whilst avoiding having too many dependencies as much as possible.

The replication started by building Apache httpd server and PHP5 at CNAF, and configuring them by taking all the requirements from the original installation, including optional packages such as Oracle instantaneous, and all the other needed libraries.

New X509 server certificates have been requested from the two involved Certification Authorities, with the special option of the “certificate subject alternative names”, which enables them to be used for SSL/https connections on the new “failover” service names: cic.gridops.org and cic2.gridops.org.

Upon this basic service layer, the portal PHP code has been deployed.

The code, which resides on a CVS repository, is periodically downloaded by a daily cron job. A special CVS tag named “production” identifies the current stable release. Moreover, the files that need local parameter customization are downloaded as templates, then properly parsed and filled by the update script.

3.1.2. Data processing system (Lavoisier) replication.

Lavoisier installation [26] requires Java, Apache ANT and the Globus Toolkit 4 Java WebServices core. On the top of this, a Lavoisier parallel instance has been installed following the official instructions. One role of this component is to periodically get data from external sources, therefore replicating the same identical configuration made it a perfectly interchangeable backup.

3.1.3. Database replication

The database layer represents the most challenging part. The backend is based on Oracle, a database solution with a steeper learning curve, especially concerning its high availability features. The need to build up some production-quality replica in a reasonable amount of time led us to start with an intermediate goal: a manual export and import of the database contents.

As long as the amount of data on this database is still within the range of tens of MegaBytes, this kind of dump transfer is always possible. The exported data has been transferred via http and verified by file integrity checker tools, before being applied on the destination instance. We have established a
complete, documented procedure, which involved at least 2 persons (one at each side). This
operational procedure includes the certification of the wholesome data integrity and the coherence
between the two sites, but still needs a short interruption of service. Consequently, a real
synchronization solution between the two databases is under study, as exposed in section 4.

3.2. SAMAP

SAMAP (an acronym for SAM Admin's Page) is a web-based monitoring tool for SAM framework
job submission. Its purpose is to send tests to all kinds of Grid resources, check the results and publish
them on the SAM (see section 3.5). Created and initially hosted in Poznan Supercomputing and
Networking Centre (PSNC), Poznan, Poland, after many positive opinions and use, it became an
official EGEE operations tool. Then, it has been integrated into CIC Portal and currently it is
commonly used by all the EGEE Grid operators and managers.

In the following sections an overview of the project has been provided, while more information and
updates can be found on the wiki [27]. Finally, a Savannah project has been created [28] to accept
requests and bug submissions by the users and the EGEE project staff.

There are two main components responsible for SAMAP functionality:
- Web portal part, implemented as a set of PHP scripts; it is connecting to the Grid User Interface part, to send user's actions, and to the GOCDB, to fetch a list of available sites, Grid users and their roles; it is responsible for end-users recognising, matching their roles with the target Grid sites and displaying the list of available sites for SAM job submission;
- User Interface part, implemented as a set of PHP and shell scripts integrated with the SAM-client command line tool installed on the same machine; it is responsible for retrieving and executing actions from the portal part of SAMAP;

The portal part is running on the web server machine, and it is currently integrated into the CIC Portal. The UI part is installed on a dedicated machine. Some dedicated Workload Management System (WMS) servers are needed, specifically prepared for SAMAP purposes. The failover of these external requirements are separately provided.

For the failover needs, SAMAP has been installed in two independent instances. The main instance is located in CC-IN2P3 centre and the backup one is located in INFN-CNAF centre. Both instances are using two independent WMS servers located in INFN-CNAF (main) and CERN (backup), so the end-user can use alternative WMS server for job submission in case one of them is down. A central CVS repository is used for code and configuration synchronization, with the use of a “production” tag.

Using DNS based geographical failover described above we can easily switch from one to the other independent SAMAP instance and it is done almost transparently to the end user. The only side effect is that all the SAM test jobs submitted on the one SAMAP instance are no longer available on the second instance after failover switch. Anyway, due to non-production character of the SAM test jobs (but only for Grid services monitoring and testing purposes), and the fact that they can be easily resubmitted again for a given site, this side effect is not relevant.

3.3. GSTAT, GRIDICE

GSTAT [11] is the first service that the failover activity has taken care of. The reason is that its
mechanism is simply made of python scripts, producing HTML, images and RRD [29] files. And the
user interaction is read-only. But besides these good points, the amount of data collected from the
information system of a big Grid can be quite large, leading to the need to just install another instance
with only the code and the configuration synchronized. Most of the information gathered by GSTAT,
if the periodical query succeeds, is almost identical between two servers. The exceptions that might
happen are not worrying, because they give only a slightly different snapshot of a site. On the other
hand, the availability, and the response time of a site, can also give opposite values. But if this happens, it means that some instability issue is probably occurring, and having two different views, from distant points of the network, can help when trying to diagnose site problems. For this reason, the requirement to synchronize the monitoring data has not been considered anymore.

The code and configuration changes are not frequent, so the method is a semi-automatic CVS update under operator control.

GRIDICE [12], although it is more complex than GSTAT as monitoring system, and also involving a Postgres DB, fits in the same line of reasoning, thus it has been only duplicated, and pointed to the same grid information system.

3.4. GOCDB
As it appears from the tools map [8], the GOCDB is a highly needed source of information for all the other systems. Because of this central role the EGEE community induced the development team to work strictly coupled with the failover working group to realize a backup replica. Although not yet in place, a DB failover for the Oracle back-end between the main and a backup site in UK is under development. In parallel, a separate web front-end has been provided in Germany by the ITWM [30] Fraunhofer institute, connected to a CNAF DB which hosts a periodic dump of the GOCDB content, ready for a first level of disaster recovering.

3.5. SAM
This is the heaviest and most worrying work in progress in the EGEE-LCG failover activity. SAM [8] is made of several components, of both standard and Grid middleware kind. Moreover it is heavily and frequently updated and queried, both from users and other services, and its Oracle DB has a grow rate in the order of $10^5$ rows/day, which makes it more delicate than any other to properly replicate and switch.

A team from the Polish CYFRONET [31] institute, exploiting the good experience collected on a local instance of SAM provided to the regional grid initiative, will try to start up a collaboration with CERN LCG-3D project, to set up the needed Oracle Streams features which are the base of a SAM replication.

A test run on the currently available resources at CYFRONET has shown that the needed computing power can be provided.

4. Future plans and improvements

4.1. Oracle replication
In order to automate the Oracle back-end synchronization, avoiding the manual transferring of data, we are testing different solutions offered by Oracle Database 10g suite [32]. Streams, DataGuard and Materialized Views represent three effective technologies but very different among them.

A Materialized View is a complete or partial copy (replica) of a target table from a single point in time. It can be called manually or automatically: when it happens, a snapshot of the DB is delivered to one or more replicas. The Materialized View approach seems to be the fastest solution to develop, but the replicated DB will never be perfectly synchronized. If something is wrong between two snapshots, all the performed modifications will be lost. Besides, Materialized Views can only replicate data (not procedures, indexes, etc) and the replication is always one-way.

Oracle Streams represent a complex system of propagation and management of data transactions. It is driven by events, like triggers. The main concerns with Streams are the tuning and administration of these events as well as the management of the correct cause-effect criteria. For a 2-way or a N-ways replication, streams are the only solution because they avoid incidental conflicts that the materialized approach cannot solve.
Oracle Data Guard is another possibility to face planned or unplanned outage. This automated software solution for disaster recovery requires one or more stand-by instances that will be kept updated with the master DB. The transactional synchronization with the master DB is achieved through the archived redo logs, which contain all the executed operations on the primary DB. Finally, the modus operandi for applying the redo logs is a tunable process according to either security, performance or availability.

As all these methods have pros and cons, a deep investigation is under way in order to answer the specific questions: "How up-to-date does the copy of the data need to be? How expensive is it to keep it updated?"

The application of these features to the Oracle based grid tools treated on this document, is following different paths. This depends from the weight of the DB and from the amount of manpower that can be dedicated. For the CIC portal it is possible that the task will be moved on the web code itself. The GOCDB staff, on the other hand, are internally studying a Streams based replication, to replace the current first approach of a manual dump. For the SAM heavy DB, it is likely that the CERN LCG-3D knowledge of Streams will be exploited.

4.2. Distributed monitoring system

The failover process consists of three phases: the first step is to detect failure, the second step to identify and activate the standby resource, and the last step for the standby application to go active.

For the time being, the failure detection delay can range from a few seconds, when the failure is detected by human loosing the application connection, to a few minutes, thanks to the monitoring systems locally provided by the computing centres. Unfortunately, the procedures that follow the detection take much more time, because they are influenced by a human factor: if one human operator is ready and on duty, he can easily operate a switch, but it is not granted that this is done in a very short time. To correct this weakness, we are working on an automatic switchover capability based on Nagios software.

Nagios is an host, service and network monitoring software, well known and stable Open Source solution. What we are doing is designing and developing a distributed matrix of Nagios instances, which can continuously exchange information about the health status of the different monitoring tools that are under control.

The idea is based on the following main points:

- a set of distributed Nagios servers;
- the exchange of tests results between the servers;
- strict tests, executed more times to avoid false-positive cases;
- quorum-based verdict, inspired by the Byzantine Generals Problem [33], leading to a minimum of four monitoring instances, and the need to reach a quorum at least on three to take a decision.

Once a verdict has been reached on the Nagios instances, a DNS switch and the proper changes are performed, to activate the backup instance for the service supposed to be out of service. The last prerequisite of this action is that a positive verdict can also be reached on the status of the backup service.
References

[1] The Enable Grids for E-sciencE (EGEE) project: http://www.eu-egee.org/
[2] The LHC Computing Grid (LCG) project: http://lcg.web.cern.ch/lcg/
[3] H.Cordier, G.Mathieu, J.Novak, P.Nyczyk, F.Schaer, M.Schulz, M.H.Tsai: Grid Operations: the evolution of operational model over the first year in Proceedings of Computing in High Energy and Nuclear Physics (CHEP06), Mumbai, India, February 2006
[4] GGUS portal: http://ggus.org
[5] T.Antoni, F.Donno, H.Dres, G.Mathieu, P.Strange, D.Spence, M.H.Tsai, M.Verlato: Global Grid User Support: the model and experience in the Worldwide LHC Computing Grid in Proceedings of Computing in High Energy and Nuclear Physics (CHEP06), Mumbai, India, February 2006
[6] CIC portal: http://cic.gridops.org
[7] O.Aidel, A.Cavalli, H.Cordier, C.L’Orphelin, G.mathieu, A.Pagano, S.Reynaud: CIC Portal: a Collaborative and Scalable Integration Platform for High Availability Grid Operations in Proceedings of the 8th IEEE/ACM International Conference on Grid Computing (Grid2007), Austin, USA, September 2007
[8] SAM: http://goc.grid.sinica.edu.tw/gocwiki/Service_Availability_Monitoring_Environment
[9] GOCDB portal: https://goc.gridops.org/
[10] SAM admin’s page: https://cic.gridops.org/samadmin/
[11] Gstat monitoring pages: http://goc.grid.sinica.edu.tw/gstat/
[12] Gridlce: http://gridlceforge.cnaf.infn.it
[13] OTM: http://goc.grid.sinica.edu.tw/gocwiki/Failover_mechanisms/OptoolsMap
[14] Reliable IP Nodes, A Prerequisite to Profitable IP Services: http://www.nspllc.com/NewPages/Reliable_IP_Nodes.pdf
[15] GSLB/BGP: http://www.networkcomputing.com/showitem.html?docid=1605ws1
[16] GSLB “page of shame”: http://www.tenereillo.com/GSLBPageOfShame.htm
[17] GRNET homepage: http://www.grnet.gr
[18] EGEE Failover page: http://www.gridops.org
[19] ISC homepage, the home of BIND: http://www.isc.org
[20] INFN CA documentation: http://security.fi.infn.it/CA/en/docs/
[21] How to Disable Client-Side DNS Caching in Windows XP and Windows Server 2003: http://support.microsoft.com/kb/318803/en-us
[22] Mac OS X lookupd man page: http://developer.apple.com/documentation/Darwin/Reference/ManPages/man8/lookupd.8.html
[23] mozilla.org Networking Preferences: http://www.mozilla.org/quality/networking/docs/netprefs.html
[24] How Internet Explorer uses the cache for DNS host entries: http://support.microsoft.com/kb/263558/en-us
[25] S.Reynaud, G.Mathieu, P.Girard, F.Hernandez, O.Aidel: Lavoisier: A Data Aggregation and Unification Service in Proceedings of Computing in High Energy and Nuclear Physics (CHEP06), Mumbai, India, February 2006
[26] Lavoisier documentation: http://grid.in2p3.fr/lavoisier
[27] SAMAP Wiki page: http://goc.grid.sinica.edu.tw/gocwiki/SAM_Admin_Page_developments
[28] SAMAP Savannah project: https://savannah.cern.ch/projects/samap/
[29] RRD: http://oss.oetiker.ch/rrdtool
[30] ITWM: http://www.itwm.fraunhofer.de/en/central/index/
[31] CYFRONET: http://www.cyfronet.pl
[32] Oracle documentation: http://www.oracle.com/technology/documentation/index.html
[33] Lamport L, Shostak R, Pease M 1982 The Byzantine Generals Problem ACM Transactions on Programming Languages and Systems Vol. 4 No. 3 382-401