Enabling Grids for E-sciencE

Information Society and Media

EGEE-II

THE GATEWAY APPROACH PROVIDING EGEE/GLITE ACCESS TO NON-STANDARD ARCHITECTURES

THE GATEWAY ARCHITECTURE AND THE MODIFICATIONS OF THE MIDDLEWARE TO PROVIDE ACCESS TO WORKER NODES ARCHITECTURES UNSUPPORTED BY THE CURRENT VERSIONS OF GLITE

Document identifier:	
Date:	**4/09/07**
Activity:	**JRA1**
Document status:	**DRAFT**
Document link:	https://edms.cern.ch/document/edmsId/version

Abstract: This paper describes the gateway architecture and the required modifications to the gLite Middleware to make available to the GRID computing machines whose hardware/software architecture is non directly supported by gLite. This work has been performed in the framework of the integration of ENEA-GRID and EGEE infrastructure.

Document Log

Issue	Date	Comment	Author/Partn
1.0	04/09/07	Initial version	Text by G. Bracco, C. Scio, A. Santoro, S.Migliori

Document Change Record

Issue	Item	Reason for Change

Table of contents

TABLE OF TABLES

1. INTRODUCTION

1.1 PURPOSE

The success of the GRID depends also on its flexibility in accommodating the computational resources available over the network. A big effort is underway to develop accepted GRID standards but in the meanwhile solutions have to be found to include into EGEE infrastructure resources based on platforms or operation systems which are not currently supported by gLite middleware. The gateway approach described in this report provides a working solution to this issue and it has been used to provide GRID access to ENEA AIX SP systems.

1.2 DOCUMENT ORGANISATION

This paper consists of an introduction describing the motivation of the gateway implementation, followed by sections dedicates respectively to the description of the architecture, the issues about authentication, the modifications required to the computing element middleware, considerations about the worker nodes, a list of open issues and finally the conclusions.

1.3 ENEA-GRID AND EGEE

ENEA (the Italian Agency for Energy, Environment and New Technologies [1]) is a funded partner in EGEE/EGEE-II projects, inside the SA1 activity. ENEA participation in EGEE is focused in integrating ENEA-GRID resources into EGEE infrastructure.

ENEA-GRID [2] has been developed since 1998 with the purpose to integrate the main computational resources inside the institution into a unique infrastructure providing an unified user environment and an homogeneous access method. ENEA computational resources are distributed over WAN (6 computing centres in Northern, Central and Southern Italy) connected by GARR, the Italian Research and Academic Network, and consist at present of about 100 hosts with a total of 650 cpu, with a variety of Operating Systems and Architectures (AIX, Linux 32/64, IRIX, MacOS X, Windows). The most relevant resource is at the moment an IBM p575 with 192 processors (AIX operating system). The installed storage is about 15 TB and the number of registered users exceeds 600.

GRID functionalities in ENEA-GRID (unique authentication, authorization, resource access and discovery) are provided using “mature”, multi-platform components that constitute ENEA-GRID middle-ware. These components are a distributed file system, AFS/OpenAFS [3], a resource manager , LSF Multicluster (Job Forwarding Model) [4] and an unified user interface based on Java and Citrix technologies [5].

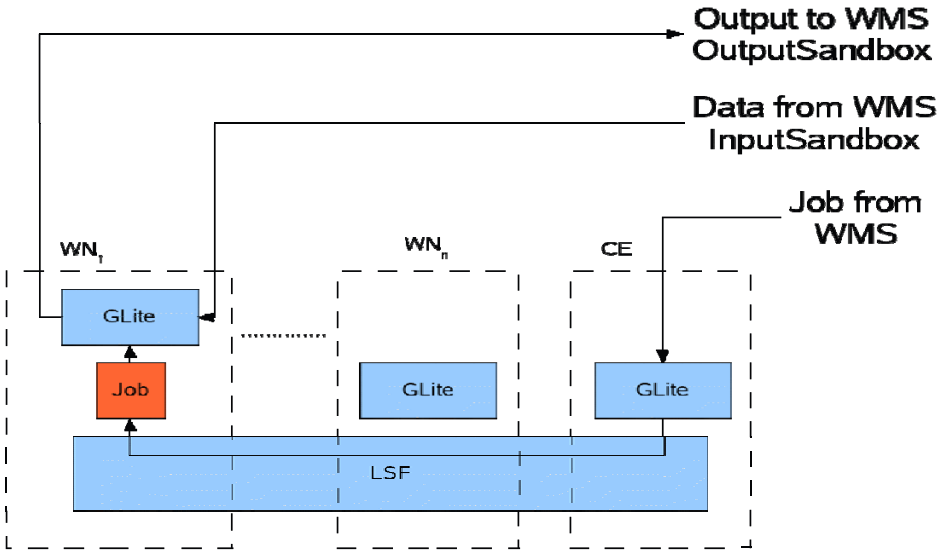
1.4 INTEGRATING ENEA-GRID AND EGEE

The goal of the integration activity between ENEA-GRID and EGEE is an implementation where the overall ENEA-GRID infrastructure is seen from the EGEE infrastructure as a single local EGEE site, namely the ENEA-INFO site. This goal can be implemented at the moment with some restrictions and the implementation has to face two main issues. The first is the unavailability of EGEE gLite middle-ware [6] for many of the platforms in ENEA-GRID and the second is the compatibility of the user management. Both these subjects are discussed in the following sections and the gateway implementation provide a solution for both of them [7,8]. The current gateway implementation is based on gLite version 3.0.1. Migration to version 3.1 is underway.

2 THE GATEWAY ARCHITECTURE

2.1 STANDARD GLITE UTILIZATION

The Computing Element (CE) machine used in a standard gLite installation and its relation with the Worker Nodes (WN) and the rest of the EGEE grid is shown in Figure 1. Basically, when the Workload Management Service (WMS) sends the job to the CE, the gLite software on the CE employs the specific dispatch manager installed on the CE to schedule jobs for the various Worker Nodes (in case of the ENEA-INFO CE the dispatcher employed is LSF Multicluster [4]. When the job is dispatched to the proper worker node (WN_1 in Figure 1), but before it is actually executed, the worker node employs the gLite software installed on itself to setup the job environment (most notably it loads from the WMS storage the files needed to run, known as the InputSandbox). Analogously, after the job execution the Worker Node employs gLite software to store on the WMS storage the output of the computation (the OutputSandbox). The problem is that this architecture is based on the assumption underlying the EGEE design that all the machines, CE and WN alike, employ the same architecture. In the current version of gLite (3.0.1) the software is written for intel-compatible hardware running Scientific Linux [12].



2.2 STANDARD GLITE AND ENEA-GRID

The ENEA-GRID differs from the EGEE grid. The basic difference is that the Worker Nodes are not necessarily the same hardware/software platform of the CE. Specifically, most of the resources made available by ENEA-GRID to the rest of EGEE are IBM machines running AIX, instead of the standard intel-compatible hardware running Scientific Linux. This implies that the gLite software cannot run on the Worker Nodes of such a cluster thus it is not possible for LSF to use the Worker Node itself to gather the job input and transmit back its output. A different method to run the gLite software must be devised.

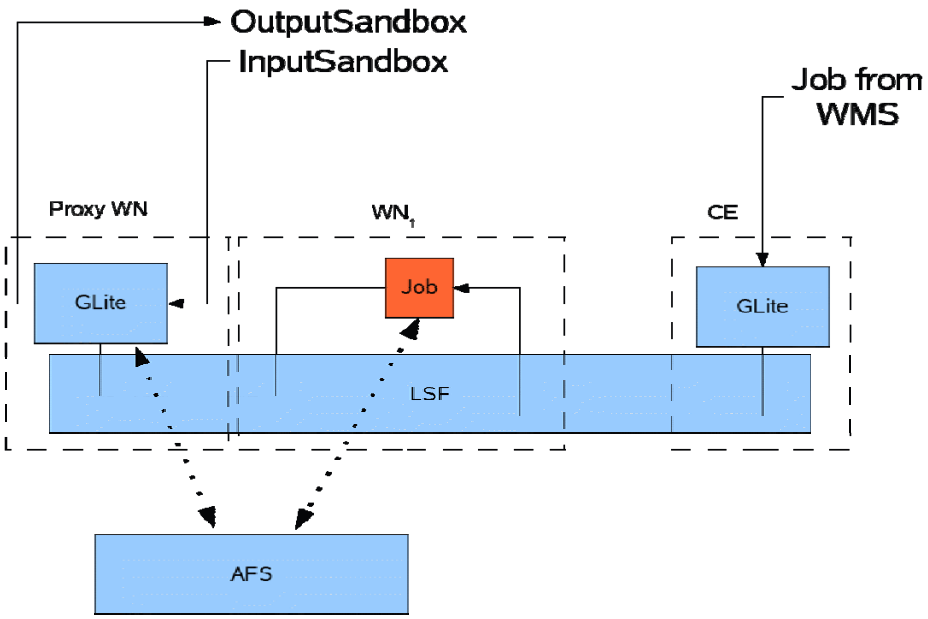
2.3 DESIGN OVERVIEW

The basic design principle of the ENEA-INFO gateway to EGEE is outlined in Figure 2, and it exploits the presence of a shared filesystem (namely AFS). When the CE receives a job from the WMS, the gLite software on the CE employs LSF to schedule jobs for the various Worker Nodes, as in the standard gLite architecture. However the worker node is not capable to run the gLite software that recovers the InputSandbox. To solve this problem the LSF configuration has been modified so that any attempt to execute gLite software on a Worker Node actually executes the command on a

specific machine, labeled Proxy Worker Node. Differently from the other Worker Nodes, this machine must be able to run gLite, and must have been installed with the gLite packages needed for a Worker Node, but does not belong to the Worker Nodes set. By redirecting the gLite command to the Proxy WN, the command is executed, and the InputSandbox is downloaded into the working directory of the Proxy WN.

The working directory of each grid user is maintained into AFS, and is shared among all the Worker Nodes and the Proxy WN, thus downloading a file into the working directory of the Proxy WN makes it available to all the other Worker Nodes as well. Now the job on the WN_1 can run since its InputSandbox has been correctly downloaded into its working directory. When the job generates output files the OutputSandbox is sent back to the WMS storage by using the same method.

2.4 LOADBALANCING ISSUES



In the above architecture, the Proxy WN may become a bottleneck since its task is to perform requests coming from many Worker Nodes. In case the performance of the Proxy WN is a concern, it is straightforward to allocate a pool of different Proxy WN, and distribute the load equally among them. The detailed method is described in Section 7.

2.5 SYNCHRONIZATION ISSUES

The adoption of a shared file system also raises synchronization issues. Specifically, it is clear from Figure 2 that the job running on WN_1 may write something on the working directory and then activate a gLite software located on the Proxy WN. Also, the job might activate a gLite software located on the Proxy WN to fetch data from other parts of the grid, and then read the data from WN1. Both scenarios share a common assumption that the cache on the machine that is reading the new data is up-to-date with the data just written by other machines. This problem is transparently managed by AFS, since any time a modified file is closed AFS automatically flush that file from the cache of all the machines that previously stored it.

2.6 FEATURES

Note that this installation approach does not require to install gLite software on any Worker Node (except, of course, on the Proxy Worker Node). Also, as will be described in detail in Section 7, the specific way used by ENEA-INFO does not require any additional software/modification on the Worker Nodes, since all the additional software is maintained on

AFS.
Also, ENEA-INFO employs AFS, but theoretically any shared file system mechanism can be employed if proper attention to the synchronization issue described in Section 2.3.2 is given. Analogously, ENEA-INFO employs the CE implementation know as LCG-CE, at the version 3.0.14.
Finally, in the case of AFS, the native quota management and data security guarantee the reliability of the user data space committed to the GRID pool account user.

3 AUTHENTICATION AND AUTHORIZATION ISSUES

On a standard EGEE site GRID users are mapped to the local UNIX users by the LCAS/LCMAPS [9] components of gLite middle-ware while ENEA-GRID users are managed using AFS resources and Kerberos 5 authentication so a compatibility solution has to be found.

LCAS/LCMAPS packages provide some integration for AFS users but not in a sufficient way for this implementation, so that a patched version of the packages has been implemented as described in [13, 14].

Moreover EGEE authentication is based on X509 certificates, which have been extended to incorporate Virtual Organization information using VOMS system [10]. While AFS and X509 compatibility is managed by the standard gssklog package [11] a development was required to add support also to VO extension, as described in [13].

Specfically, in the gLite standard design once the proxy certificate has been recognized (by LCAS), the CE gatekeeper employs LCMAPS to spawn a process with the privileges of the grid user, if the grid user is specifically known to the CE. In case the user is not specifically recognized by the CE, but belongs to an authorized Virtual Organization (VO), the gatekeeper spawns a process with the privileges of one of many accounts dedicated to manage the requests from the specific VO (such accounts are known as a “pool” accounts). In both cases, to work properly when the working directory resides on AFS, the process spawned by the gatekeeper, in addition to the privileges of the user must also have the AFS token to write on its working directory on AFS. If the process does not receive the token access to its working directory is forbidden by AFS.

Therefore, it is necessary for any thread that schedules jobs as a certain grid user (either a specific user or a pool user) to acquire the AFS token for that user. In order to acquire that token, LCMAPS invokes the gssklog command which examines the certificate of the incoming user and, if allowed, issues the proper AFS token to the calling process. Finally the token is forwarded to the Worker Nodes (that needs it to actually run jobs) by proper mechanisms built inside LSF.

The gssklog command requires the gssklogd service to be active on a predefined server [11,13], which is another main component of the gateway architecture.

4 MODIFICATIONS ON THE COMPUTING ELEMENT

4.1 YAIM

4.1.1 PROBLEMS

The YAIM [15] package is used by the gLite middleware for an easy installation of the components required by each GRID element. In the case of the CE the presence of AFS leads to several issues when yaim runs to install the gLite middleware.

For example, although NFS is available on the cluster, it is not the aim to use it on ENEA-INFO site since AFS is already available. Also the EGEE grid users do not need to be maintained on each machine by the gLite software as they are already maintained by AFS and Kerberos 5.

4.1.2 SOLUTION

Some function scripts of YAIM have been modified to face the above problems.

4.1.3 DETAILS OF PATCHED SCRIPTS

- **CONFIG_NFS_SW_DIR_SERVER, CONFIG_NFS_SW_DIR_CLIENT**

  In the "/opt/gLite/yaim/functions/local" directory, the yaim function scripts "config_nfs_sw_dir_server" and "config_nfs_sw_dir_client" attempt to configure NFS. However, as observed in Section 4.1 the ENEA-INFO configuration does not make use of NFS, thus to avoid an unwanted activation of NFS both scripts have been substituted as follows (example for "config_nfs_sw_dir_client". The same script with the name changed goes for "config_nfs_sw_dir_server"):

```
function config_nfs_sw_dir_client () {
  return 0
}
```

- **CONFIG_USERS**

  Still in the "/opt/gLite/yaim/functions/local" directory, the yaim function scripts "config_users" takes care to create groups and users for the grid accounts. However the users are already configured globally on AFS, which means that they must not be created on each machine. Thus line 82-86 of script (that take care of creating the grid user accounts) have been commented.

  Still, although the users must not be added to the specific machine, they must be added to the proper additional groups, thus two more modifications are required. The first modification involves modifying the line:

```
sort -t: -k2,2 $USERS_CONF | join -v 2 -t: -1 1 -2 2 $PASSWD - > $MISSING
```

  This line would return only the grid accounts that have not been created on the system, because it assumes that if the user exists the proper setup has already been performed.

  In ENEA-INFO configuration all users have already been created (on AFS) but they have yet to be configured, thus the line has been changed to return all the grid accounts, as follows:

```
sort -t: -k2,2 $USERS_CONF | join -a 1 -t: -1 1 -2 2 - $PASSWD > $MISSING
```

The second modification consists in actually adding those accounts to any additional group, thus instead of the commented lines we use the following ones:

```
if   [ "${additionalgroups// /,}""bar" != "bar" ]
then
    usermod -G "${additionalgroups// /,}" $user
fi
```

Another operation carried out by this script is to insert into the cron service a periodic invocation of the "cleanup-grid-accounts.sh" script which removes from the home directories the stale temporary directories.

Since the stale home directories are on AFS, and thus shared among all the machines, the script should run only on one machine. Thus the following lines have been added after line 101, which prevents any machine different from the CE to run the "cleanup-grid-accounts.sh" script:

```
if [ `hostname` != "egce.frascati.enea.it" ]; then
  return 0
fi
```

4.2 INTERACTION WITH GSSKLOG

4.2.1 PROBLEM

As described in Section 3 the gssklogd server resides on a different machine than the CE. However, the CE must notify gssklogd about the user mappings, otherwise the gssklogd is not able to authorize the user to access.

4.2.2 SOLUTION

Part of the configuration files of gssklogd consists in information resident on the CE (specifically the certificates contained in the "`/etc/grid-security/certificates/`" "`/etc/grid-security/vomsdir/`" directories and the user mapping instructions contained in the "`/etc/grid-security/grid-mapfile`" file). Therefore a script `update_enea_context`, to be executed hourly, has been added to the cron of the CE.
The task of the script is to copy the information contained in the above files/directories into AFS, where they can be properly read by gssklogd.
It is to be noted that although the script runs with root privileges, it would not have the authorization to write on access those files, since it has not acquired an AFS token. Therefore the first action of the script is to acquire the AFS token. This is done by generating a kerberos keytab for the user "egeegssk" (a system user authorized to work in the aforementioned directories) and invoking the K5START utility as in the following line:

```
k5start -qtU -f /var/keytabs/egeegssk.keytab
```

The kstart package [16] is a classical way employed in AFS to obtain a token from kerberos keytabs.

4.3 THE JOB MANAGER

4.3.1 PROBLEMS

The Globus Job Manager employed by the LCG-CE assumes that the underlying file system is a normal UNIX filesystem. However, in the ENEA-INFO case the underlying file system is the AFS shared filesystem and the grid users directories are shared among the machines. Also, the Job Manager, when issuing a job to a Worker Node, exports the CE environment variables to the worker node. In the ENEA-INFO context this may lead to conflicts, because the environment of the CE does not necessarily match the environment of the worker node.

4.3.2 SOLUTIONS

Slight modifications have been applied to the scripts “lsf.pm” and "cleanup-grid-accounts.sh".

4.3.3 DETAILS OF PATCHED SCRIPT

- **LSF.PM**

  The “lsf.pm” script in the directory “/opt/globus/lib/perl/Globus/GRAM/JobManager/” is responsible to generate scripts that wrap the job to be submitted, and actually submit them to LSF. One of the tasks accomplished by the generated script is to set some environment variables. However, the environment variable GLOBUS_LOCATION is set into the script submitted to LSF (and thus executed on a Worker Node) to the value it assumes on the CE, because of the underlying assumption that the CE and the WN share the same environment. However for the reasons explained in Section 2, this assumption is not correct in the ENEA-INFO environment, thus we add another modification to the “lsf.pm” script.

  Immediately after line 204 we add a new line:

  ```
  print JOB "GLOBUS_LOCATION=\"/afs/enea.it/project/eneaegee/system/eneaaddon/egui/opt/globus\";
  export GLOBUS_LOCATION";
  ```

  which basically overrides a previous setting of the GLOBUS_LOCATION environment variable with the setting needed by our infrastructure.

- **CLEANUP-GRID-ACCOUNTS.SH**

  The job manager employs the script "cleanup-grid-accounts.sh" in the directory “/opt/lcg/sbin/” to perform garbage collection of the files left by previous runs of the Job Manager. The script however assumes that the directories to clean are not in afs, which is not the case for the ENEA-INFO installation. Thus ENEA-INFO had to add a line at the start of the script to acquire the AFS token in the same way described in section 4.2.2:

  ```
  k5start -qtU -f /var/keytabs/egeegssk.keytab
  ```

  where the egeegssk user is also allowed to modify the home directories of the grid users (see Section 7). Also, line 196 of the script has been changed from:

  ```
  sysdirs='/(afs|bin|boot|dev|etc|initrd|lib|proc|root|sbin|usr)/
  '
  ```

  to:

  ```
  sysdirs='/(bin|boot|dev|etc|initrd|lib|proc|root|sbin|usr)/'
  ```

  essentially telling the script to cleanup also directories that reside under AFS.

4.3.4 BUG FIX

As a side note, the script “lsf.pm” is not able to properly handle environment variables containing blanks, which are employed in the ENEA-INFO environment. Specifically, when the script is generated, the “lsf.pm” is not able to properly recognize whether some environment variables have blank separators inside in which case it does not generate a correct script.

To solve this bug lines 203 and 204 of the script “lsf.pm” have been changed from:

```
print JOB $tuple->[0], '=', $tuple->[1],
 '; export ', $tuple->[0], "\n";
```

to:

```
print JOB $tuple->[0], '="', $tuple->[1],
 '"; export ', $tuple->[0], "\n";
```

4.4 LCMAPS MODIFICATION

In order to be able to work on the EGEE grid accounts home directories (both the pool and static user accounts), the CE software needs to acquire an AFS token. This is done through the LCMAPS service. However in this respect the existing LCMAPS service supports only statically defined users and it does not work for pool account users. Thus the LCMAPS service has been modified as described in [13].

4.5 THE INFORMATION SYSTEM

4.5.1 PROBLEMS

The script "lcg-info-dynamic-lsf" of the Information System (IS) is the interface between the IS and the underlying LSF. Its task is to collect data for the Information System by calling the proper LSF commands. However it assumes a specific configuration of LSF, which is not employed in the ENEA-INFO environment.

Specifically the LSF configuration employed by ENEA-INFO exhibits the following problems, in respect to the current version of the IS script:

1. In the LSF configuration each node is described by a customizable “Type” field (among others). The script "lcg-info-dynamic-lsf" assumes that in the LSF is configured such that the "Type" field of the CE node is equal to the "Type" field of the Worker Nodes. This is not the case of the ENEA-INFO environment, since the Worker Nodes may be radically different from the CE node. The Worker Nodes dedicated to EGEE are a subset of the total nodes of ENEA-INFO and share the same configuration, while the CE and proxy nodes are dedicated to the EGEE project and as such are identified by a different type.

2. The "lcg-info-dynamic-lsf" assumes that the output of the LSF command "bhosts" returns only the machine names without the domain portion. However, the ENEA-INFO LSF environment spans across multiple domains, which forces ENEA to configure LSF to identify the machines by their fully qualified domain name, not by their simple machine name. This means that “bhosts” reply with the full domain names of the nodes while the script expects a name without domain portions, which causes script failures.

4.5.2 SOLUTION

Three lines of the "lcg-info-dynamic-lsf" script have been modified to correct the issues.

4.5.3 DETAILS OF PATCHED SCRIPTS

The three points of the "lcg-info-dynamic-lsf" script that have been modified are the following:

1. Line 430: the original pattern matching instruction "next if (! /(^[a-zA-Z][0-9A-Za-z_\-]+)\s+(\w+)/);" has been modified as "next if (! /(^[a-zA-Z][0-9A-Za-z_\-\.]+)\s+(\w+)/);" to include fully qualified domain names. This fixes problem 2) described above.

2. Line 438: the instruction "next if ($hoststype{$hostname} ne $CELSFType);" that verifies whether the type of the CE matches the type of the Worker Nodes has been commented. This partially fixes problem 1) described above.

3. Line 866: the instruction "next if (!defined $hoststype{$hostname} || $hoststype{$hostname} ne $CELSFType);" that verifies whether the type of the CE matches the type of the Worker Nodes has been commented. This, along with the modification above, fixes problem 1) described above.

5 THE WORKER NODES

The worker nodes are unmodified, and no gLite middleware has been installed on the local disk. It is assumed instead that AFS and LSF are installed on them. Proper configuration of LSF (see Section 7) allows to execute some software resident on AFS which in its turn execute software on the proxy node.

6 THE PROXY WORKER NODE

The proxy worker node is the only machine of the cluster that actually has the worker node software installed on it (in addition to AFS and LSF). However, there has been no need to modify such a software in any way. Only the standard Worker Node installation has been necessary.

Note that the proxy worker node must not be included into the list of the worker nodes maintained in the proper file (typically the “wn.conf”file) otherwise it might receive jobs intended for a different architecture.

7 ADDITIONS ON AFS

7.1 REDIRECTION TOWARDS THE PROXY WN

As shown in Figure 2 when the CE wants to execute a job, it actually sends it via LSF to one of the WNs. However part of the commands that are embedded into the job are globus/edg commands required to move files to/from the resource broker. Such commands are part of the gLite middleware and cannot be found on the Worker Nodes, which have no middleware software residing on them. Thus, a proper directory of AFS stores a set of shell scripts which have the same names of the gLite middleware commands but they merely invoke LSF to execute the corresponding command on one of the proxy worker node. The general structure of a wrapper is the following:

```
#!/bin/bash
lsrun -R <proxy-resource> <directory-on-proxy>/<command-name> $*
```

Note that the "`-R <proxy-resource>`" allows to address scalability issues since it executes the job on a node that exports the resource `<proxy-resource>`.

If care is taken to assign the `<proxy-resource>` to all proxy worker nodes (and only to them) LSF will automatically take care also of load balancing. Up to now ENEA-INFO has not experienced such issues, thus a single proxy node is currently employed. In case of need this mechanism allows seamless extension to a generic number of proxy worker nodes.

The wrappers have been specified for the following commands:

In directory "/opt/lcg/bin":

lcg-aa, lcg-infosites, lcg-ManageSoftware, lcg-sd, lfc-entergrpmap, lfc-modifyusrmap, lcg-cp, lcg-is-search, lcg-ManageVOTag, lcg-uf, lfc-enterusrmap, lfc-rename, lcg-cr, lcg-job-monitor, lcg-mon-wn, lcg-version, lfc-getacl, lfc-rm lcg-del, lcg-job-status, lcg-ra, lcg-wn-os, lfc-ln, lfc-rmgrpmap lcg-fetch, lcg-la, lcg-rep, lfc-chmod, lfc-ls, lfc-rmusrmap, lcg-gt, lcg-lg, lcg-replica-manager, lfc-chown, lfc-mkdir, lfc-setacl, lcg-info, lcg-lr, lcg-rf, lfc-delcomment, lfc-modifygrpmap, lfc-setcomment

In directory "/opt/edg/bin":

edg-brokerinfo, edg-gridftp-ls, edg-gridftp-rename, edg-gridftp-rmdir, edg-gridftp-exists, edg-gridftp-mkdir, edg-gridftp-rm, edg-wl-logev

In directory "/opt/edg/libexec":

gLite_dgas_ceServiceClient

In directory "/opt/gLite/bin":

gLite-brokerinfo, gLite-gridftp-rename, gLite-version, ldapdelete, ldapsearch, gLite-gridftp-exists, gLite-gridftp-rm, globus-url-copy, ldapmodify, gLite-gridftp-ls, gLite-gridftp-size, grid-proxy-info, ldapmodrdn, voms-proxy-info, gLite-gridftp-mkdir, gLite-url-copy, ldapadd, ldappasswd

In directory "/opt/globus/bin":

globus-url-copy, grid-proxy-info, ldapadd, ldapdelete, ldapmodify, ldapmodrdn, ldappasswd, ldapsearch

In directory "/opt/lcg/sbin":

castor, newacct, nsrename, rfrm, stageqry, vmgrdeletedenmap, vmgrgettag, cpdskdsk, nschclass, nsrm, rfstat, stagestat, vmgrdeletedgnmap, vmgrlistdenmap, Cupvadd, nschmod, nssetchecksum, rtstat, stageupdc, vmgrdeletelibrary, vmgrlistdgnmap, Cupvcheck, nschown, nssetcomment, showqueues, stagewrt, vmgrdeletemodel, vmgrlistlibrary, Cupvdelete, nsdelcomment, nsshutdown, stagealloc, sysreq, vmgrdeletepool, vmgrlistmodel, Cupvlist, nsdeleteclass, nstouch, stagecat, tpconfig, vmgrdeletetape, vmgrlistpool, Cupvmodify, nsenterclass, rep, stagechng, tpmstat, vmgrdeltag, vmgrlisttape, Cupvshutdown, nsfind, rfcat, stageclr, tpread, vmgrenterdenmap, vmgrmodifylibrary, dumptape, nslistclass, rfchmod, stageget, tprstat, vmgrenterdgnmap, vmgrmodifypool, infd, nslisttape, rfcp, stagein, tpsrv_in_prod, vmgrenterlibrary, vmgrmodifytape, msgd, nsls, rfdir, stageout, tpstat, vmgrentermodel, vmgrsettag, msgi, nsmkdir, rfmkdir, stageping, tpusage, vmgrenterpool, vmgrshutdown, msgr, nsmodifyclass, rfrename, stageput, tpwrite, vmgrentertape.

7.2 SETTING UP CORRECT ENVIRONMENT ON WORKER NODES

In addition, there is the need to tell the worker node that it has to look for such commands into AFS. This is accomplished by the jobstarter script of LSF. The jobstarter is the script that LSF runs in order to execute the batch jobs. The configuration of LSF has been tailored so that the worker nodes use a specific jobstarter.

The code of the jobstarter is the following:

```
for i in
/afs/enea.it/project/eneaegee/system/config/egee/profile.d/*.sh ; do

      if [ -r "$i" ]; then
        source $i
      fi
    done

$*
```

The main goal of this jobstarter is to load the environment variables that contain the paths of the globus commands, and to take care that such paths are set to the proper paths on AFS. The only exception to this rule is listed in Section 4.3.1 since one of those environment variables (GLOBUS_LOCATION) is set by the CE instead than by the WN and modifying the CE script "lsf.pm" was needed to take care of this issue.

7.3 CENTRALIZED ACCOUNTS ON AFS

As described in Section 2 the home directories of EGEE grid accounts (both static users and pool accounts) reside on AFS. Therefore each machine involved in grid (the CE, the SE, worker nodes, proxy worker nodes) must share the same users, with the same UID.

In the standard EGEE installation the usernames and UIDs of grid and pool accounts are listed into a file named "ig-users.conf", which guarantees that all the machines share the same users and UIDs.

In case AFS is employed (as in ENEA-INFO) there is an additional concern. Specifically, the home directories are shared and reside on AFS, thus not only the usernames and UIDs on each machine must be aligned among them, but the UIDs must also be aligned with those employed by the AFS home directories.

The solution adopted by ENEA-INFO consists in having a central management of the /etc/passwd file, storing it on AFS and modifying it on each machine everytime a new user is added/removed. As already observed in the description of the "config_users" script in Section 4.1.2, the information stored in ig-users.conf is not used to create users, but only to create the groups they belong to.

7.4 CLEANUP ON AFS

Another effect of the home directories of EGEE grid accounts residing on afs is that the cleanup-grid-accounts.sh script (running on the CE) must have AFS permissions to modify those directories.

This is accomplished by creating a fictitious user "egeegssk" and setting the home directories so that "egeegssk" has "write" permissions on them, as described in section 4.2.2.

8 OPEN ISSUES

8.1 MANAGING MULTIPLE ARCHITECTURES WITH A SINGLE CE

Currently the setup described in this document can manage architectures different from Intel-compatible Linux. Still, it is not yet possible for a single CE to control machines of multiple architectures. The same structure of the Information System expects a single CE to manage machines belonging only to one type of hardware/software architecture. Since the ENEA-INFO site offers to the rest of EGEE two kind of processors (Intel-compatible and PowerPC), the current setup of ENEA-INFO involves the presence of two CEs, one for the Intel machines, and the other for the PowerPCs. This means that the same user, belonging to a specific VO, will be assigned to one specific pool account on one CE (e.g. enea001) and, in general, a different one on the other CE (e.g. enea003).

If the two CE had no relation with each other this would not be a concern, but the working directory of the worker nodes resides on AFS and is shared among the CEs. To avoid the presence of different users on the same working directory, and to allow a better monitoring at LSF level, it is in the interest of ENEA-INFO that users of the grid are mapped on the same pool account for both CEs.

To achieve this goal, the directory "/etc/grid-security/gridmapdir" has been shared among the two CEs (through NFS). The directory contains the mapping of the grid users to the pool accounts, and sharing it ensures that the mapping is shared among the CEs. However this is a method must be implemented manually, and a more automated method to share user mapping should be devised.

8.2 INEFFICIENT GLUE SCHEMA INFORMATION

Another issue in providing access to IBM SP worker nodes employing AIX 5.2 is related to the information system. In the GlueSchema employed by the information system, the hardware architecture of the Worker Nodes is specified by the GlueHostProcessor attributes:

- GlueHostProcessorVendor
- GlueHostProcessorClockSpeed
- GlueHostProcessorModel

Analogously, the Operating System of the worker nodes is specified by the GlueHostOperatingSystem attributes:

- GlueHostOperatingSystemName
- GlueHostOperatingSystemRelease
- GlueHostOperatingSystemVersion

Currently we have defined these variables as follows:

- GlueHostProcessorVendor=IBM
- GlueHostProcessorClockSpeed=1500
- GlueHostProcessorModel=G4
- GlueHostOperatingSystemName=AIX
- GlueHostOperatingSystemRelease=5.2
- GlueHostOperatingSystemVersion=AIX

the issue is that the standards for the definitions of these variables are not clearly defined for architectures outside the intel-compatible domain.

The Gstat wiki site (http://goc.grid.sinica.edu.tw/gocwiki/How_to_publish_the_OS_name) provides an updatable list for the Operating System, which for the moment we have updated adding the operating system entries detailed above. However there is not an analogous page for the GlueHostProcessor attributes. Moreover, checking at another page of the Gstat wiki site (http://goc.grid.sinica.edu.tw/gocwiki/How_to_publish_my_machine_architecture) it would appear that the hardware architecture should be exported by defining the (currently almost unused) GlueHostArchitecturePlatformType attribute. The problem is that the same wiki says that the architecture name should be obtained by using the "uname -m" command, but on AIX the command gives as result "00373D6A4C00" instead of an intellegible architecture name like "i686".

Another concern is whether the information available is sufficient for a user friendly selection of the resource. Currently, a user submitting a generic Linux i386 job should specify inside its JDL a complex boolean string spanning at least three different values for GlueHostArchitecturePlatformType (namely "i386", "i686" and "x86_64"), countless values for GlueHostOperatingSystemName (e.g. "CentOS", "Debian", "Scientific Linux" and all the different Linux flavors) and a fair amount of values for GlueHostProcessorModel (e.g. "Xeon", "PIV" and "OPTERON", among others).

If the user didn't specify anything, the jobs would risk to be sent to incompatible architectures as AIX (and thus fail). On the other hand, if the user specified simpler strings the jobs would be sent to a much smaller subset of /hardware/software architectures than those that could actually process it.

It would seem advisable to define additional GlueSchema attributes (or new values for existing attributes) that might help in selecting wide subsets of different architectures that can still accept the same jobs.

8.3 RGMA

Another currently unsolved issue is the use of RGMA. To work properly RGMA requires to have a client running on the worker nodes. since there is no software installed on worker nodes, the RGMA clients cannot be run.

Thus, the wrapper of “rgma-client-check” in directory “/opt/gLite/bin/” is different from those presented in Section 7 and contains instead a simple “echo” command informing that RGMA is not active on ENEA-INFO. Specifically the wrapper is written as follows:

```
#!/bin/bash
echo Sorry, no RGMA on ENEA-INFO site.
echo For more informations please visit
http://www.afs.enea.it/project/eneaegee/
exit 0
```

8.4 GRID-ICE

A fourth open issue, similar to the one above is the use of GRID-ICE. Grid ICE is a monitoring tool, whose components are distributed over the site, including the worker nodes.

Since the ENEA-INFO Worker nodes run no gLite software, the output of GRID-ICE from the worker nodes would be senseless, thus GRID-ICE on the worker nodes has been disabled by setting the configuration flag “`GRIDICE_MON_WN=no`” in the site configuration file (“my-site-info.def”). However most of the information required by GRID-ICE are still made available by LSF. Therefore it is probably possible to solve this the problem by implementing a service that collects the same information from LSF and writes them into a file that is subsequently accessed by GRID-ICE.

9 CONCLUSION

The report has described in details the architecture and the implementation of the gateway solution built on the main components of ENEA-GRID middle-ware, which is based on very mature and reliable software, namely the AFS distributed file system and LSF Multicluster. The key element of the architecture is a set of Linux proxy machines, running standard gLite middle-ware, which support the communication between the non standard worker nodes and the EGEE infrastructure.

The gateway provides a flexible and affordable solution for the access in principle to all the platforms and operating systems available in ENEA-GRID and has been finalized to the case of the AIX SP system, but tests have also been performed for Altix IA64, IRIX, MacOS X and Solaris.

This result can be used to expand the EGEE GRID capability by including a wider range of resources but also, on the other hand, to take advantage on the maturity of the gLite grid services to offer a working GRID solution to communities have been up to now discouraged by the middle-ware rigidity.

In the last year the ENEA-INFO EGEE site has been certified in the gateway configuration for AIX resources and it is open to production jobs. The site supports at present several VOs (COMPCHEM, EGRID, FUSION) and the experimentation with applications is underway.

10 REFERENCE/AMENDMENT/TERMINOLOGY

10.1 REFERENCES

Table 1: Table of references

[1]	http://www.enea.it
[2]	**"ENEA-GRID: a production quality grid infrastructure"**, GARR Meeting "Armonizzazione delle strutture di rete e delle griglie computazionali", CRUI, Roma, 15 July 2004, www.garr.it/incontro_griglie.htm
[3]	http://www.openafs.org
[4]	http://www.platform.com
[5]	http://www.citrix.com
[6]	http://www.glite.org
[7]	G. Bracco et al. **"Integration of ENEA-GRID multiplatform resources in EGEE"**, SA1 Open Session "Access to other Platforms" , 4° EGEE Conference, Pisa 24-26 october 2005; http://indico.cern.ch/sessionDisplay.py?sessionId=22&slotId=0&confId=a0514#2005-10-27
[8]	G. Bracco et al. **"Implementing GRID interoperability",** AFS Best Practice Workshop 2006, University of Michigan, Ann Arbor, 12-16/6/2006, http://pmw.org/afsbpw06/
[9]	LCAS/LCMAPS packages, http://www.nikhef.nl/grid/lcaslcmaps
[10]	VOMS package, http://infnforge.cnaf.infn.it/voms
[11]	gssklog package, D. Engert, ftp://achilles.ctd.anl.gov/pub/DEE/gssklog-0.11.tar
[12]	https://www.scientificlinux.org/
[13]	G Bracco et al,.**"AFS Pool Account Users",** EGEE Technical Note EGEE-TR-2006-006
[14]	G. Bracco et al. EGEE,06 Conference, Geneva (Switzerland) 25-29/9/2006, **"AFS pool account users and GRID interoperability"**, http://indico.cern.ch/getFile.py/access?contribId=203&sessionId=120&resId=0&materialId=slides&confId=1504
[15]	YAIM, https://twiki.cern.ch/twiki/bin/view/EGEE/YAIM
[16]	KSTART, http://www.eyrie.org/~eagle/software/kstart/

10.2 DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGEE "Document Management Procedure" will be followed: http://egee-jra2.web.cern.ch/EGEE-JRA2/Procedures/DocManagmtProcedure/DocMngmt.htm.