AstroGrid-D

Deliverable 1.3

Workgroup 1: Resource Integration and Grid Support

AstroGrid-D Production Grid, Version 1[1]

Deliverable	1.3 AstroGrid-D Production Grid, Version 1
Authors	Workgroup 1: Resource Integration and Grid Support
Editors	H. Enke, T. Radke, T. Brüsemeister
Date	9.1.2007
Document Version	1.0.1
Current Version	1.0.1
Previous Versions	1.0.0, 0.1.3, 0.1.2, 0.1.0

A: Status of this Document

Published Deliverable 1.3, Version 1.0.1, describing the main components of the first version of the grid built by integration of AstroGrid-D resources and the usage of D-Grid resources.

B: Reference to project plan

This deliverable refers to a twofold goal of our project plan. First, the complete integration of all resources, which are part of the co-financing, based on Globus 4.x, and the components required to use these resources as a production grid by some of the UseCases of AGD. Second, the useability of other D-Grid resources within the frame of our middleware.

[1]This work is part of the AstroGrid-D project and D-Grid. The project is funded by the German Federal Ministry of Education and Resarch (BMBF).

C: Abstract

This document describes some of the developped procedures to integrate the AGD resources into a grid, the additional efforts to build up an efficient VO management for this grid, the use of the AGD-Information Service to overcome the shortfalls of monitoring features of Globus-MDS, basic requirements of health monitoring facilities and basic usage policies for the grid.

D: Changes History

Version	Date	Name	Brief summary
0.1.0	26.11.2006	H. Enke	Working Draft Creation
0.1.1	6.12.2006	T. Radke, T. Brüsemeister	Renamed 'Health Monitoring' into 'Service Monitoring', Spelling
0.1.2	7.12.2006	H. Enke	Added sketches for VOMRS, some clarification to chapter about clusters and workstations, references
0.1.3	23.12.2006	H. Enke	Added a chapter about execution environments, references
1.0	3.1.2007	H. Enke	Published version
1.0.1	9.1.2007	T. Radke	Finalisation of information service and execution environment chapters; Proof-reading

E:

Contents

1 Default Procedures for Resource Integration

Based on the guidelines of Deliverable 1.2[4] and the various experiences while installing GT4 (Globus Toolkit 4.x) on AGD (AstroGrid-D) resources, a set of installation instructions was developed. Since AGD resources are supposed to enable Grid-MPI with MPICH-G2, the GT4 installation has to be done from the sources. Since there was only scarce experience how to tackle this on various AGD-sites, a user guide was required. This user guide is now, together with a set of scripts aimed at an easy configuration and administration of grid resources, available as a product of AGD on our website. D-Grid has included this documentation as a link on the central support site at LRZ. The guide consists of

- documentation of compiling and configuration of prerequisites like Postgresql and iodbc, Ganglia, JDK.
- concise instructions of setting up the environment, compiling and adding patches to GT4.
- necessary administrative setup and configuration of GT4, including grid account setup.
- certification handling and policy configuration.
- a default configuration and support for local RAs.
- a complete suite of front- and backends for VO Management in AstoGrid-D.

The default configuration of the resources was developped mostly with SL (Scientific Linux) 4.x[12] as underlying OS, and thus it is usable without any changes for Red Hat and Fedora distributions. It turned out to work, without much changes, to work on SuSE-Distributions as well. This was achieved by adhering to the FHS (Filesystem Hierarchy Standard)[9]. With changes due to deviations from FHS it was used as well with a Debian distribution.

To make changes of JDK, Tomcat and even GT4 upgrades possible without reconfiguring all path and environment settings, the concept of symlinks to all those packages was introduced. This approach allows to retain all settings and only requires symlinking new versions to the appropriate places. The GT4 requirements work well with symlinks with the exception of the required sudoers file entries, where a script is employed to provide safe new entries according to the local environment.

All configuration of xinetd and init.d started services is assisted by scripts, relieving the local administrator of the painstaking task to edit a lot of configuration files by hand and picking all pertaining information from the widespread documentation of GT4.

For grid account management, the concept of grid-pools as groups and grid accounts, which are only accessible by means of GT4 based methods, as suggested in D.1.2, was implemented. The configuration of the grid-security for a local resource was eased by providing a default package of D-Grid Root-CA policies and certificates and a script, which allows a very easy configuration of the local policy, if the Institute (aka local resource provider) obtains the certificates for services and users from GridKA. A similar script for configuring with DFN Root-CA will be available as soon as DFN provides pertaining information. Additionally, the installation of security relevant procedures using the process of revocation lists was included.

For GT4-WebMDS[11], enhancement of the default Globus information provider by Ganglia[10] is required for all AGD resources. The default MDS has a quite unsatisfying structure, so we

developed an XSLT transformation providing a webpage which summarizes most of the interesting information, extracting this from MDS and providing links to the original MDS information. AGD MDS information is centralized by an AGD-MDS server, which in turn provides all information to the D-Grid MDS-server at LRZ.

The user guide is currently available in German language only, a translation will be available soon. The deviations for SuSE and Debian distributions will be documented and included.

2 VO Management

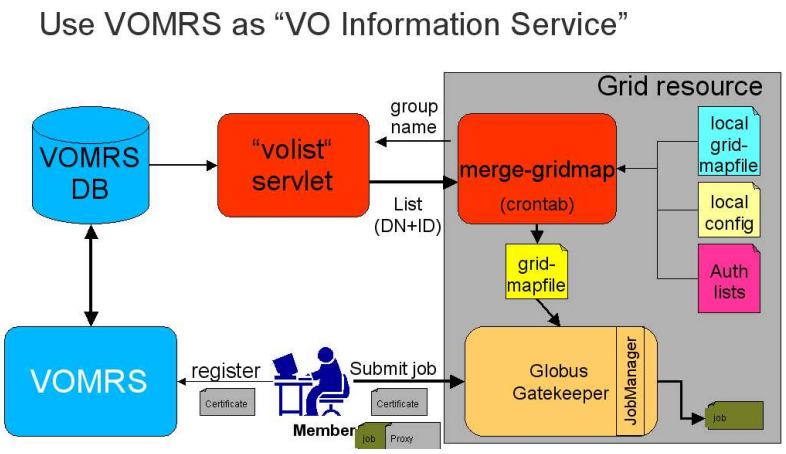
Enabling VOs (Virtual Organizations) as a means for sharing resources is one of the key concepts of the grid development. The VO concept introduces the change from a purely local management of accounts on local resources to a grid based management of grid resources. Authentication of users and services is based on a chain of trust, which has to be built up. Although the middleware allows arbitrary Root-CAs, even only locally valid CAs, a nationwide deployment of a grid across all communities is only possible by establishing one Root-CA for each country. For historical reasons currently Germany favors two of these Root-CAs. These Root-CAs are responsible for each issued certificate and take care of a secure procedure for generating certificates. The verification of identities of users and services (eg. hosts), is the counterpart to providing certificates as a means for authentication. With a growing number of users, this part remains managable only by delegating the task to local trustees, called RAs (Registration Authorities).

Each partner institute within AstroGrid-D has established an RA. To support these RAs with their task of signing all certificate requests, we developed a configuration and a set of supporting scripts for generating cert-requests, verifying them and installing obtained certificates (especially for services) according to local configuration settings.

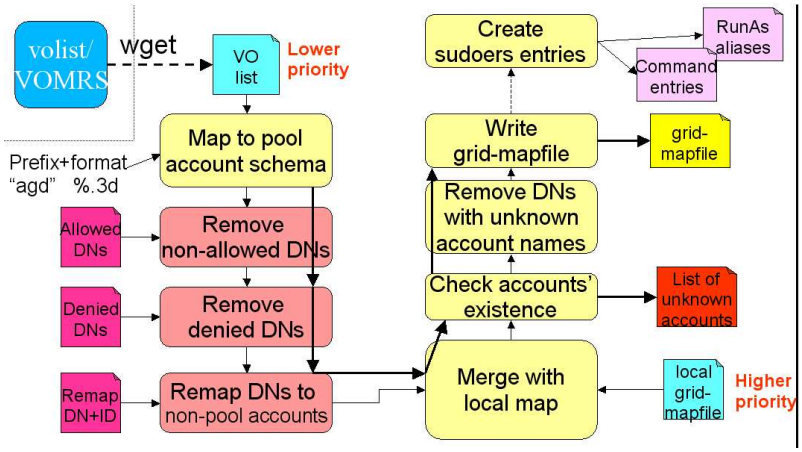
Institute	**CA**	**RA**	**Name**
AEI	GermanGrid, GridKA	AEI	Christa Haussmann-Jamin
AIP	GermanGrid, GridKA	AIP	Harry Enke
MPA	GridGerman, DFN	MPA	Rainer Sigl
MPE	GridGerman, DFN	MPE	Rainer Sigl
TUM	GridGerman, DFN	LRZ	Anton Frank
ZAH	GermanGrid, GridKA	ZAH	Peter Schwekendiek
ZIB	GridGerman, DFN	ZIB	Thomas Steinke
LRZ	GridGerman, DFN	LRZ	Anton Frank
UP	GermanGrid, GridKA	UP	Klemens Kittan
RZG	GridGerman, DFN	RZG	Johannes Reetz

While implementing this procedure we worked closely with GridKA to ensure the validity of all software components developed. With these RAs in place, AstroGrid-D VO has of today more certified users than the Kern-D-Grid of the DGI.

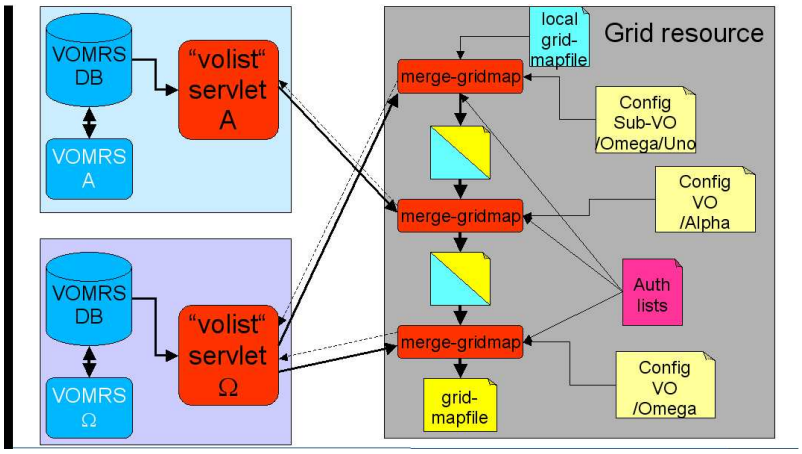
The management of these certified users based on grid principles requires to provide a means for all local RAs to manage their local certified users in a central repository, from which all resource providers obtain the information for their gridmap files. With D.1.2[4] we decided to use a 1:1 mapping of users to grid accounts locally and against the idea of a n:1 mapping (i.e. for grid usage n users are mapped to a single shared account to employ the grid services). This decision required to develop a different set of tools than those provided by VOMS[4][2], which anyway is not usable in GT 4.x-based grid environments[7]. Since gLite[15] is one of the three middleware stacks for D-Grid, we nevertheless wanted to stay compatible with VOMS. Our central repository for AGD certificates is built on VOMRS[3], which is an enhanced version of the VOMS server, the gLite central repository. Both of these are able to exchange all necessary information about grid users, thus achieving compatibility between VOMS and VOMRS.



Managing the process of locally updating the gridmap file and generating necessary grid accounts is done with VOMS by using a different means, disabling the gridmap file. Since we decided to keep the gridmap file as the central means for authentication, as it is the default GT4 procedure, a java servlet was created to allow a simple HTTP based fetch of the required information for the gridmap file from the central repository. For generating a new gridmap file from the information of the central server, new users have to be generated on demand and invalidated users have to be removed from the local gridmap file. An implementation of the update process which only obtains the information from the central repository (as it's done in VOMS) would break the principle that local resource providers should stay in full control over the mapping of grid users to local users in the gridmap file to be generated. The solution for VOMRS was implemented as a merging process where, through a set of configuration scripts, the local resource provider is able to control the gridmap file update, even if it's running a cron job.



The separation of managing VO via a server (VOMRS) and local handling of gridmap-file and user management allows for an easy integration of many different VOMRS-servers, so that community-managed VOMRS servers can easily be included. In addition, the local user/gridmap process allows for the adoption of site-specific naming conventions.



The implementation of these procedures is part of the already mentioned "globus-helper" package[7] on our AGD website.

3 Information Service for Grid production jobs

The requirements for submitting a job to the grid, as derived from our Use-Cases collection, expose a couple of insufficiencies of MDS[2]:

- even with Ganglia as an additional information provider for MDS, neither geographical location nor sufficient hardware information or the local software environment is available.
- information about local resource management like queuing systems is very shallow, i.e. which different queues are configured and available.
- information about available bandwidth connecting compute and hardware resources is missing.
- runtime information of grid jobs, required to monitor resource usage, is not available.
- only the individual user is provided with information about duration of his job.

Instead of trying to plug all this information into MDS, which is oriented towards catering to the needs of the GRAM service in the first place and would potentially break this functionality, we decided to use the Stellaris (AstroGrid-D Information Service) as an independent means to collect all of the information needed and develop interfaces for this service to provide the grid users and services with such information. The Stellaris, developed by WG2[3], developed to handle the metadata, proved capable of handling this task as well[5].

Therefore, an XSLT transformation from MDS-XSL to Stellaris-RDF was developed by WG1, in a first stage for the information which is included into MDS using the GLUECE 1.1 schema. Although GLUECE 1.1 is already outdated and a more complete GLUECE 1.2 version is available, the GT 4.x implementation is still based on the old schema. Currently we work on substituting the newer schema into the GT4, but it will need to be verified that there is no hidden compatibility issue involved. The MDS-to-Stellaris transformation already has a direct use with our GridResource-Map, originally written only to demonstrate the capability of the RDF implementation of Stellaris, which delivers now translated MDS information on top of a Google map, showing the geographicall deployment of all available AstroGrid-D resources[4].

For compute and storage resources, the geographical location may be of minor impact, but with the task of including other resources such as robotic telescopes, this feature becomes very important. Encouraged by the successful XSLT transformation of the GLUECE schema to RDF, a first implementation of an XSLT transformation from RTML (Robotic Telescope Markup Language) to RDF was already achieved[5].

Since very different information providers can be plugged into Stellaris very easily, which in turn offers a unified query interface, this approach seems a viable alternative to enhancing the Globus MDS service, and avoids the generation of potential compatibility issues.

Extracting sufficient information from Stellaris via commandline tools is currently being worked on, since the first stage of production use of the AGD grid is based on user selection of appropriate resources.

[2] see WG5, Deliverable 5.1[6]
[3] see Deliverable 2.1 and 2.2
[4] This is as well adopted by D-Grid already.
[5] see WG5, Deliverable 5.3

4 Environment for Grid Jobs

For a real production grid, additional information about the available compilers, software libraries, binaries, queue settings of the local resource manager and other, hardware-related information is crucial with respect to running an executable for a given Grid resource[6]. There are three alternatives to consider:

- the grid job comes with a statically linked executable
- the information about the resource environment is sufficient for preparing the grid job to safely execute on the resource
- the grid job provides all the necessary environment via the staging-in process

4.1 Statically linked executables

A statically linked resource may run on different OS, which share a fairly broad set of characteristics. This is supported on many OS of the Unix flavor with the overhead of emulating the environment for the general type of binary or calling an interpreter for the binary, e.g. with the Linux OS[16] Because the approach does not work across different hardware architectures, it is of limited value and restricted to relatively simple programs of serial nature. However, as there are currently only a few different architectures and OS included in the D-Grid, it is a valid approach to solving the problem of insufficient information about resources. It is not suitable for any MPI application or jobs with special hardware requirements.

4.2 Packaged Grid Jobs

A packed grid job is prepared to create all the necessary environment, libraries and binaries on the target machine by means of either generating all libraries, binaries and environment settings in advance or through a workflow (e.g. a script), which executes all necessary steps of compiling, installing and setting up the environment before the actual job is started. While this would guarantee the best performance of the actual job, the penalty for generating the proper environment is huge, and the workflow is prone to failures in the preparational stages. This approach might be suitable for jobs which are using java-based executables or in – rare occasions – if special hardware features are involved, eg. GRAPE boards.

4.3 Tuned Grid Jobs

A tuned grid job is prepared on sufficient knowledge about the environment of the resource it will be executed on. Although AstroGrid-D gathered a fairly large amount of use cases and derived requirements for their execution environment, this is only a first approximation. To enhance the flexibility of the grid environment, the user must be able to query for special information pertaining to his job from every resource. One simple method for this is to use the gsissh facility to interactively

[6] WG1 already addressed this in the first questionnaire about resource integration. A first idea of an XML-Schema was then implemented which may be a starting point.

log into each resource and gather the information on his own. But this is not suitable for any automated mechanism such as a Grid brokering service. A standardized and extensible schema for gathering such information from each resource and storing this metadata in the information service is needed. Then, with suitable query tools, either from the commandline of with a GUI, a user can retrieve the metadata and select the resources based on this information.

For any job at the current stage of integration of grid resources it will be a mix of the aforementioned approaches. Some information about the resources is known, and so a grid job using the static approach could prepare different statically linked executables for different hardware architectures or OS with substantial differences in their software ennvironment. For organizing the job execution packaging of components into a tar-file for staging is a widely used practise.

4.4 Information Service Schemata

While there are already schemata as GLUECE built into Globus, no schema for information about compilers, libraries, queue settings, MPI facilities (including different drivers for ethernet, myrinet, infiniband, etc.) are known. These schemata have to be developed and implemented. With Stellaris as an efficient mechanism for storing and retrieving the information already in place, the implementation of suitable commadline tools and GUIs was started. With respect to the broker mechanism, a suitable method to extract this information should not be too hard to find.

Here are some snippets of the xml-tags we already worked with, which are required for the environment. This is only meant as a starting point for more complete environment information.

```
<resource id="gavoX.aip.de">
   <cpu-capabilities>
      <flags>
         <sse value="1">
         <sse2 value="0">
         <3dnow value="1">
         <3dnowext value="1">
         <mmxext value="1">
      </flags>
   </cpu-capabilities>
   <software>
      <!-- software that matches the hardware specified above -->
      <compiler name="GNU" version="3.2.2" path="/usr"/>
      <compiler name="Intel" version="9.0" path="/opt/intel/9.0"/>
      <!-- if type="workstation|cluster" zero or more performance libraries tags -->
      <library name="MKL" version="7.0" path="/opt/intel/mkl70"/>
      <library name="GSL" version="?" path="/usr"/>
      <library name="fftw2" version="?" path="/opt/lib/fftw2"/>
      <!-- if type="workstation|cluster" zero or more mpi tags -->
      <mpi name="MPICH" version="1.2.5" path="/opt/mpich"/>
      <mpi name="MPICH-G2" version="1.2.7" path="/usr/remote/mpich-G2-gcc32dbg"/>
      <mpi name="SCALI" version="?" path="/opt/scali"/>
      <mpi name="LAM" version="6.5.6" path="/usr/local"/>
      <!-- if type="cluster" one or more batchsystem tags -->
```

```
      <batchsystem name="openPBS" version="2.3.16" path="/usr/local">
         <queue name="debug" size="4"/>
         <queue name="default" size="64"/>
      </batchsystem>
   </software>
   <grid>
      <grid-software>
         <!-- if type="cluster|workstation" zero or more globus tags -->
         <middleware type="Globus Toolkit" version="4.0.1">
            <path value="/usr/remote/globus/current">
            <tcp_port_range start="20000" end="25000"/>
         </middleware>
      </grid-software>
      <grid-security>
         <!-- defined one or more accepted certificates -->
         <accepted ca="GermanGrid" authority="GridKA-CA" type="X509"/>
         <!-- defined zero or more denied certificates -->
         <denied ca="GridGerman" authority="DFN" type="X509"/>
      </grid-security>
      <virtual-organisation>
         <vo name="AstroGrid-D">
            <!-- specify the usage policy for the resource -->
            <!-- example shared="4" for available nodes -->
            <!-- example time="0.01" for 1% of all cpu cycles -->
            <!-- example time="120:00:00" for specific number of cpu hours -->
            <policy shared="..." time="..."/>
         </vo>
      </virtual-organisation>
   </grid>
</resource>
```

5 Grid Service Monitoring

We call a process Grid Service Monitoring, which regularly checks

- the availability of all default grid services (GSIftp, GSIssh, GSIscp, GRAM, gatekeeper, MDS)
- the regular update of gridmap files and revocation lists for certificates
- for central services running on special hosts (AGD-MDS, Stellaris) their availability

to ensure a healthy and reliable grid infrastructure. This is only the first stage, later we want to include more and detailed information about the performance of these services and resources. This information is uploaded to the Stellaris, from where a web-interface draws an actual status and via a commandline tool it is usable to verify quickly possible sources of failure when a job submission fails.

A first, incomplete, implementation is already done, including the upload process to Stellaris. An improved version of the web-interface is being worked on. For several UseCases this information will be included into the job submission setup.

6 Grid Usage Policies

With the production stage, the definition of usage policies becomes an important part of the work. In this early stage, without suitable brokering and monitoring facilities, we defined a basic set of policy statements to be followed by all AstroGrid-D users and resource providers:

Policy Statement1
Each member of VO AstroGrid-D is, by virtue of being part of this VO, able to use whichever grid resource is included in our production grid. For grid resources with PBS queues no lower priority than for local users should be introduced by local resource providers.

From this rule follows that each local provider is supposed to regularly update the gridmap file according to the contents of the central repository. Since the local provider nevertheless stays in control of his resources, he still is able to exclude certain users temporarily or permanently from using his resource.

Policy Statement 2
Deviations from this policy, i.e. exclusion of users, have to be made public immediately as well to the excluded user as to the VO Management body, which is currently constituted by the VO administrators. The VO Management body informs the project management and the Steering Board of AstroGrid-D.

This rule introduces the requirement of exposing the deviation from the basic policy to the VO and providing the reason for the decision by the local resource provider.

Policy Statement 3
Each VO member must make considerate use of the available grid resources and is required to cooperate with a local resource provider, if there is an issue raised with a job he submitted to this resource.

This rule is aimed at introducing a cooperative behavior within the VO members and towards the local resource providers. For example, if a job requires only moderate hardware resources, it should not be submitted to a cluster with specialized hardware like the GRAPE cluster. Or, a job which only requires 5 CPUs must not be submitted to a queue with 32 nodes. To register such behavior, WG1 will open up a module in our issue tracking system, directed at managing such issues.

Policy Statement 4
If an over-commitment of available resources occurs, the priority of jobs is to be decided based on a) the amount of committed resources of the institutes between the conflicting partners, b) on the importance of the jobs in the frame of the project plan.

This rule is a first attempt for conflict resolution. Within the project, the project management and the architecture group will try to resolve a conflict based on this rule, and the Steering Board has to decide finally, if no resolution is found.

In later stages of the production grid some of these rules will be handled by the resource broker.

7 Clusters and Workstations

While clusters and their local resource management systems seem the natural resources for the grid, the inclusion of workstations is nevertheless useful. The main difference is the local resource manager: for clusters, there is a queueing/scheduling system (e.g. SGE or Torque alias PBS)) available, while workstations do not run such managing system. This difference has to be handled carefully when including workstations into the grid infrastructure. As one result from our AstroGrid-D use case enquiry, we know that a considerable number of applications will run as task-farming jobs with moderate requirements on hardware and runtime. Especially these jobs benefit from the availability of workstations for job submission. Since these workstations primarily are meant to serve an individual user, this priority has to be ensured. Therefore another policy statement is required.

Policy Statement 5
If a workstation is available as a grid resource, the jobs of the local owner/user have a higher priority than grid jobs.

This rule is necessary for inclusion of workstations into the grid. The implementation of this policy requires to implement a means to suspend grid jobs, if the local owner of the resource requires complete availability of the workstations exclusively. An implementation of this policy is available, which consists of a suspend facility available to the user at any time, and a re-enabling process, which runs as a cronjob.

References

[1] Globus Firewall Requirements, http://www.globus.org/toolkit/security/firewalls/Globus\%20Firewall\%20Requirements-7.pdf

[2] VOMS: Virtual Organisation Membership Service, http://edg-wp2.web.cern.ch/edg-wp2/security/voms/voms.html

[3] VOMRS: Virtual Organisation Membership Registration Service, http://wwwserver2.fnal.gov/www/docs/vox/voxconv/Output/voxTOC.html

[4] Deliverable 1.2 (AGD-WG1), http://www.gac-grid.org/project-documents/deliverables/wp1/ResourceIntegration-D_1_2_1.0.0.pdf

[5] Deliverable 2.1 (AGD-WG2), http://www.gac-grid.org/project-documents/deliverables/wp2/Information_Service_D21.pdf

[6] Deliverable 5.1 (AGD-WG5), http://www.gac-grid.org/project-documents/deliverables/wp5/D5.1.pdf

[7] Globus helper package (AGD-WG1), http://www.gac-grid.org/project-products/Software/Globus-Helper.html

[8] Support Guides (AGD-WG1), http://www.gac-grid.org/project-products/grid-support.html

[9] (Linux) Filesystem Hierarchical Standard, http://www.pathname.com/fhs/pub/fhs-2.3.pdf

[10] Ganglia, http://ganglia.sourceforge.net/

[11] GlobusMDS, http://www.gac-grid.org/project-products/grid-status.html

[12] Scientific Linux https://www.scientificlinux.org/

[13] GLUE Schema, http://glueschema.forge.cnaf.infn.it/

[14] GLUECE: Compute Element, http://www.globus.org/toolkit/docs/4.0/info/key/gluerp.html

[15] gLite: EGEE Lightweight Middleware for Grid Computing, http://glite.web.cern.ch/glite/

[16] Binary Formats: Linux Kernel Documentation, binfmt_misc.txt.