NORDUGRID-MANUAL-21
28/3/2015

ARIS and EGIIS

Installation, Configuration and Usage Manual

O. Smirnova, M. Ellert, D. Johansson

Abstract

In order to create a Grid infrastructure using ARC-enabled computing resources, information description and aggregation services need to be deployed. ARIS is coupled to a computing resource and collects information about it. EGIIS keeps a list of ARIS instances, and eventually, of other EGIIS instances lower down in hierarchy. Top-level EGIIS instances thus serve as an entry point to the Grid, allowing to discover all the resources.

1 Introduction

In a distributed computing world, discovery of necessary resources and services is a key to enabling functional infrastructures.

In order to be discovered and consequently used, a Grid resource or service must accomplish these two tasks:

1. advertise itself to potential clients in a standard manner,
2. expose relevant information about itself via a standard interface.

In ARC, these functionalities are implemented in the **ARIS** component (Section 3).

Given the potentially very large number of Grid resources and services advertising themselves, a hierarchical structure of organising information published by them is necessary in order to avoid bottlenecks. Such an hierarchy is often implemented by using information aggregating or indexing services, kind of "super-peers". Such an indexing service in ARC is implemented in **EGIIS** (Section 4).

While ARIS is *coupled* to a resource, EGIIS is an *independent* service. A typical Grid resource owner always has to deploy ARIS*. EGIIS servers, on the other hand, are normally deployed by the overall Grid infrastructure operators.

A system effectively created by ARIS and EGIIS services is called the *ARC Information System*. Being based on OpenLDAP [7], it can be accessed in a standard manner by a variety of LDAP clients, giving a full overview of the infrastructure resources.

The Information System's main tasks are:

- **Resource Description:** Characterization of Grid resources by publishing (via ARIS) their static, semi-static and dynamic properties (e.g. information about active Grid jobs or user quotas are presented as dynamic properties of each individual resource).

  ARC clients are relying on the resource description functionality of the Information System during their matchmaking and brokering process. ARC Grid monitoring and job status queries also rely on resource description.

- **Resource aggregation:** Individual resources are connected to an "information mesh" by dynamically registering to one or more information index services (EGIIS). The information index services are responsible for the resource aggregation, maintaining dynamic lists of available Grid resources. Furthermore, the index services are connected to each other following an hierarchical topological order. The resulting structure is the "information mesh" of the Grid.

  ARC Grid clients performing resource discovery scan the "information mesh" utilizing its topological structure in order to find available Grid resources. Therefore resource discovery is delicately coupled to the topological structure of the "information mesh" and thus to the resource aggregation process.

Implementation details and ARC information schema description can be found elsewhere [6].

2 Overview of ARC LDAP Infosys components

The ARC middleware implements a dynamic LDAP-based distributed information system via a set of coupled resource lists (index services) and local LDAP databases. The system consists of three main components:

1. the ARC Resource Information Service (*ARIS*),
2. the Enhanced Grid Information Indexing Service (*EGIIS*),
3. and the Registration Processes (*RP*)

*Without ARIS, a resource is still functional, but is not a Grid resource

ARIS instances are responsible for resource (e.g. computing or storage) description and characterization. The local information is generated on the resource, and it can be cached. Upon client requests it is presented via LDAP interface.

The main task of EGIIS services is to maintain a dynamic list of resources (LDAP URLs of the ARISes) and other index services. The index services thus are coupled together, implementing a specific topology.

Both ARIS and EGIIs services make use of Registration Processes running locally on the resources in order to publish themselves in one or more of the resource lists maintained by EGIIS services. Registrations are always initiated by the registrants (bottom-up model).

ARIS has to be present at each Grid resource, and is therefore essentially an integral part of a Grid service, like for example a Compute Element. EGIIS is a stand-alone service and is not coupled to any resource. While EGIIS is not needed for any particular service to function as such, there is no Grid without EGIIS: EGIIS instances aggregate information and thus create a coupled infrastructure from individual resources.

ARC Grid clients such as monitors, Web portals or user interfaces perform two type of queries:

1. During the resource discovery process clients query EGIIS services in order to collect list of LDAP contact URLs of ARIS services describing Grid-connected resources.
2. During a direct resource query the clients directly contact each ARIS by making use of the obtained LDAP contact URLs.

Both type of queries are carried out and served via LDAP protocol.

2.1 Operational overview

ARC information system consists of a few different components that are to be split over a few different machines:

1. ARIS – is installed by system administrators that are responsible for the clusters that are connected to the Grid. Most importantly, it runs on the ARC Computing Element and is responsible for publishing information about the resource (cluster). More information about ARIS is available in Section 3. This service has to be configured to register to one or more EGIIS instances.
2. EGIIS – is normally installed as a standalone service by an infrastructure operator. It can co-exist with any other service; when co-deployed with ARIS, it will make use of the same LDAP database instances. This service is responsible for aggregating information from multiple ARISes and other EGIISes, and can register this information to a higher level EGIIS. EGIISes can create a hierarchical topology of any depth. More information about EGIIS can be found in Section 4.
3. Client – is typically installed on the user's machine by users themselves, and can either be a generic OpenLDAP client for presenting information in a human readable form, or an ARC client that is responsible for job submission or status querying. The standard ARC client does not output LDAP information directly, instead it tailors it to the task that was asked of it. For example, if a job status was queried, only information about that job will be shown to the user.

2.2 Overview of ARC LDAP Infosys schemas

ARC information system currently can present information in three different formats, or schemas. These can be enabled simultaneously. The schemas are:

1. NorduGrid ARC schema – this is the NorduGrid default schema, described in detail in this document. It was inspired by Globus MDS, but has been improved a lot over the years and due to incompatible changes was moved into the NorduGrid LDAP namespace. If you want standard NorduGrid clients to submit jobs to your resource, you want to publish this schema.
2. Glue 1.2 – This is the schema that is used by gLite [3]. Currently, gLite support Glue 1.3 schema, but Glue 1.2 is sufficient to be compatible. If you configure ARC to publish information in the Glue 1.2

format, you will first produce data in NorduGrid-ARC schema which will then be translated to Glue 1.2 (see technical documentation [6] for mapping details). If you want to allow gLite clients to submit to your resource, you want to publish this schema. Please note, that you will also need to hook in your ARC cluster into the gLite information system in order to get this interoperability to work. Some information about this is available in Appendix A.

3. Glue 2.0 – This is the schema that will become the common schema for the EMI [1]. This schema can be published both through LDAP and XML interfaces of ARC Compute Element.

3 ARIS

ARIS is the information service that is installed on the ARC Compute Element. It publishes via LDAP interface information about the local computing cluster, like: operating system, amount of main memory, computer architecture, information about running and finished jobs, users allowed to run and trusted certificate authorities. The information can be published in either NorduGrid-ARC schema, Glue 1.2 schema or Glue 2.0 schema.

The NorduGrid-ARC schema is the main ARC schema, which needs to be published in order for an ARC client to submit jobs to the resource.

The Glue 1.2 schema is the main schema of gLite, if you want to make your cluster compatible with gLite clients, then you will want to enable this schema. Please take a look at Appendix A.

The ARIS component of the information system is responsible for generating the dynamic state information, implementing the first-level caching of the local information and providing the requested Grid information to the clients through the LDAP protocol. ARIS is basically nothing more but a specially populated and customized OpenLDAP database.

The dynamic resource state information is generated on the resource. Small and efficient programs, called information providers, are used to collect local state information from the batch system, from the local Grid layer (e.g. A-REX [5], Grid Manager or GridFTP server [4]) or from the local operating system (e.g. information available in the `/proc` area). Currently, ARC is capable interfacing to the following batch systems (or local resource management system LRMS in the ARC terminology): UNIX fork, the PBS-family (OpenPBS, PBS-Pro, Torque), Condor, Sun Grid Engine, IBM LoadLeveler and SLURM.

The output of the information providers (generated in LDIF format) is used to populate the local LDAP tree. This OpenLDAP back-end implements two things: it is capable caching the providers output and upon client query request it triggers the information providers unless the data is already available in its cache. The caching feature of the OpenLDAP back-end provides protection against overloading the local resource by continuously triggering the information providers.

The default information stored in ARIS follows the NorduGrid-ARC information model. Technical documentation [6] gives a detailed account of the ARC information model.

3.1 Comparison with Globus MDS-1

The original ARC information system was based on Globus MDS-1. Like the current system, this was a service based on OpenLDAP, but with custom back-end modules. These back-end modules were not portable and could only be used with a specific version of the `slapd` server. The MDS GRIS module used for publishing of resource information is now replaced by the BDII (see Section 3.2), which offers the same functionality in a much more portable solution, without changing the interface for querying the data. Clients developed against the MDS GRIS back-end do not have to be changed to work with the new BDII based system.

3.2 BDII

The Berkeley Database Information Index (BDII) [2] consists of a standard LDAP database which is updated by an external process. The update process obtains LDIF records from a number of sources and merges them. It then compares this to the contents of the database and creates an LDIF file of the differences. This is then used to update the database.

3.3 Information providers

The information providers are responsible for creating the LDIF records that the BDII update process uses to update the database. By running the information providers in the background they do not delay the response time of the BDII LDAP server. When deployed as part of an ARIS installation, BDII is configured to use the ARC back-ends [8] as its infoprovider, which generates an LDIF describing the resource using one or more of the schemas NorduGrid ARC, Glue-1.2 and Glue-2.0, depending on configuration.

3.4 Security considerations

ARIS is implemented via an LDAP database which implies the security and confidentiality capabilities of the system.

OpenLDAP [7] contains two methods for specifying access control. The first is static, i.e. you define the rights in configuration files. From an operational point of view, the problem of this method is that needs a server restart at every security configuration change. The second method for access control, called as ACI (Access Control Information), inserts access control information inside the directory itself by augmenting every LDAP entry with a dynamically modifiable ACL. Unfortunately the ACI method is still considered to be experimental.

The current ARC setup makes use of the static LDAP access control, the trees are configured to be fully readable by anybody: ARC provides anonymous read access to every information stored in the local trees.

4 EGIIS

The EGIIS is a replacement for the old Globus MDS GIIS back-end used to implement an information index. This system uses a separate information index server process to store the registrations and generate replies to queries about them and uses the `slapd` shell back-end to communicate between the `slapd` server and the information index server. Since no custom back-end is needed, this solution is portable and can be used together with many different `slapd` versions.

The MDS GIIS back-end did not comply fully to the LDAP specification and for certain queries it returned results that are not complying with the specification. In order to minimize the impact on clients that were developed expecting this non-standard behavior, the EGIIS tweaks the behavior of the `slapd` to allow it to, for most cases, return the same type of non-compliant replies the MDS GIIS module did, thereby providing a backward compatible interface for clients.

The EGIIS does not implement the full set of features the MDS GIIS back-end had, but only those that are used by the ARC information system. In particular, it only stores the registration information of registered clients and does not cache copies of their information trees.

5 Deployment scenarios

5.1 A testbed resource running an own EGIIS

In this scenario the ARIS and EGIIS are served by the same slapd server on different base DNs. The ARIS normally publishes its information under the base

```
Mds-Vo-name=local, o=Grid
```

whereas the EGIIS publishes under

```
Mds-Vo-name=<name>, o=Grid
```

where the `<name>` is the name of the index.

The ARIS is configured in the [infosys] section of the arc.conf. The information providers that collect the information also make use of the configuration of the resources about which they collect information, e.g. the information provider that collects information about the A-REX uses the A-REX configuration.

An EGIIS is configured using an [infosys/index/...] section in the arc.conf. If you want the ARIS to register to the EGIIS on the same machine you need to add an [infosys/cluster/registration/...] section too.

5.2 A resource registering to several EGIISes

To register an ARIS to an EGIIS an [infosys/cluster/registration/...] section in arc.conf is used. You can have as many such sections as you like to register at multiple EGIISes.

```
[infosys/cluster/registration/SwedenUppsala]
targethostname="grid.tsl.uu.se"
targetsuffix="Mds-Vo-name=Sweden, o=Grid"

[infosys/cluster/registration/SwedenLund]
targethostname="quark.hep.lu.se"
targetsuffix="Mds-Vo-name=Sweden, o=Grid"
```

5.3 EGIIS accepting registrations from external resources

To configure an EGIIS an [infosys/index/...] section in arc.conf is used. You can run many indices in the same server by using multiple entries with different names.

```
[infosys/index/Sweden]
name="Sweden"
allowreg="*.uu.se:2135"
allowreg="*.lu.se:2135"
allowreg="*.liu.se:2135"
allowreg="*.umu.se:2135"
allowreg="*.chalmers.se:2135"
allowreg="*.swegrid.se:2135"

[infosys/index/Test]
name="Test"
allowreg="grid.tsl.uu.se:2135/*, Mds-Vo-name=local, o=Grid"
```

5.4 EGIIS registering to another EGIIS

To register an EGIIS to another EGIIS an [infosys/index/.../registration/...] section in arc.conf is used. You can have as many such sections as you like to register at multiple EGIISes.

```
[infosys/index/Sweden/registration/NorduGrid1]
targethostname="index1.nordugrid.org"
targetsuffix="Mds-Vo-name=NorduGrid, o=Grid"
registrantsuffix="Mds-Vo-name=Sweden, o=Grid"

[infosys/index/Sweden/registration/NorduGrid2]
targethostname="index2.nordugrid.org"
targetsuffix="Mds-Vo-name=NorduGrid, o=Grid"
registrantsuffix="Mds-Vo-name=Sweden, o=Grid"
```

6 Deployment hints

The ARIS can be installed by installing the *noridugrid-arc-aris* package. The ARIS is configured in the [infosys] section of the arc.conf file.

After properly configuring the ARIS it can be started and stopped using the `nordugrid-arc-ldap-infosys` start-up script.

```
service nordugrid-arc-ldap-infosys start

service nordugrid-arc-ldap-infosys stop
```

The default setting is to run the ARC information system on port 2135 (which is the IANA assigned port for "Grid Resource Information Server". If you want the informations system to be reachable from the outside you need to open this port in your firewall.

On systems with SELinux enabled, you will need to allow `slapd` to use the configured port, and to use the configured location for storing its Berkeley database files and its PID and lock files. If you are using the default configuration, this is done automatically for you by the RPM post-install scriptlet which contains the lines:

```
semanage port -a -t ldap_port_t -p tcp 2135
semanage fcontext -a -t slapd_db_t "/var/lib/arc/bdii/db(/.*)?"
semanage fcontext -a -t slapd_var_run_t "/var/run/arc/bdii/db(/.*)?"
```

If you configure ARIS to use a different port or different directories, you need to manage the SELinux configuration manually.

You can install EGIIS by installing the *nordugrid-arc-egiis* package.

The EGIIS is started using the same `nordugrid-arc-ldap-infosys` start-up script as the ARIS.

The *nordugrid-arc-egiis* package contains a `nordugrid-arc-egiis.pp` SELinux module that allows `slapd` to communicate with the information index server using the `arc-infoindex-relay` program via its shell back-end. The RPM post-installation scriptlet enables this module automatically.

A Glue 1.2 with ARC, gLite interoperability

The main reason for publishing ARC information in the Glue schema is for interoperability with gLite. In order to do this, knowledge about both the ARC information system and the gLite information system is required.

In ARC, you have a local (on ARC CE) information system called ARIS. In gLite, this is translated to a Resource-BDII.

In ARC, the ARIS registers to one or more EGIIS, while in gLite, the Resource-BDII registers to a single Site-BDII. ARC can be configured to act as a Site-BDII as well by configuring it in `arc.conf`.

In ARC, the topmost level is another EGIIS. In gLite, it is a Top-BDII. You can not set up a Top-BDII with ARC software.

In ARC, Information System can have infinite number of levels of hierarchy; in gLite there are only three.

What you are likely to want to do is to set up ARC to act as a Resource-BDII and as a Site-BDII. And then register your Site-BDII to a gLite top-BDII.

B List of Configuration Variables

Mandatory configuration variables are marked with an asterisk. Sections names and default values may contain substitutions of the form `$`*`var`* where *var* is an identifier. Section names containing such substitutions introduce a documentation schema for sections introduced by matching titles. In a default value specification, a substitution `$`*`var`* refer either to a configuration variable *var* or to the corresponding variable matched by the enclosing section name.

B.1 Section `[infosys]`

`debug`
: Whether to enable debug messages, positive enables, non-positive disables. Default: `0`.

`slapd_loglevel`
: Level of logging to enable for `slapd` Default: `0`.

`hostname`
: The FQDN of the server host. Default: ‘`hostname -f`‘.

`slapd_hostnamebind`
: The host names on which `slapd` should listen. Default: `*`.

`threads`
: The number of slapd threads. Default: `32`.

`providerlog`
: Log file of the info providers. Default: `/var/log/arc/infoprovider.log`.

`db_archive`
: Path to the `db_archive` executable. Default: `/usr/sbin/slapd_db_archive`.

`db_checkpoint`
: Path to the `db_checkpoint` executable. Default: `/usr/sbin/slapd_db_checkpoint`.

`slapd_cron_checkpoint`
: Whether to enable cron job to checkpoint infosys data. Default: `disable`.

`user`
: User name of the slapd process. Default: `ldap` or `openldap` if available.

`bdii_location`
: Prefix of the BDII installation. Default: `/usr`.

`giis_location`
: Prefix of the GIIS installation. Default: `$ARC_LOCATION`.

`bdii_update_cmd`
: Path to the `bdii-update` executable. Default: `${bdii_location}/sbin/bdii-update`.

`bdii_debug_level`
: Log level for BDII. Default: `ERROR`.

`bdii_tmp_dir`
: Temporary directory used by BDII. Default: `/var/tmp/arc/bdii`.

`bdii_var_dir`
: Directory containing architecture dependent data for BDII. Default: `/var/lib/arc/bdii` or `/var/run/arc/bdii`.

`bdii_run_dir`
: Directory containing runtime state for BDII. Default: `/var/run/arc/bdii` or `$bdii_var_dir`.

`bdii_log_dir`
: Directory where to place BDII log files. Default: `/var/log/arc/bdii`.

`port`
: Port number on which BDII shall listen. Default: `2135`.

`bdii_db_config`
: Location of the Berkeley database configuration file. Default: `/etc/bdii/DB_CONFIG`.

`bdii_database`
: LDAP database backend to use for BDII. Default: `hdb`.

`bdii_archive_size`
: The number of BDII updates to log. Default: `0`.

`infosys_compat`
: Whether to enable backward compatibility. Default: `disable`.

`infosys_nordugrid`
: Whether to publish LDAP records in the NorduGrid schema. Default: `enable`.

`infosys_glue12`
: Whether to publish LDAP records in the GLUE-1.2 schema. Default: `disable`.

`infosys_glue2_ldap`
: Whether to publish LDAP records in the GLUE-2 schema. Default: `disable`.

`provider_timeout`
: Number of seconds to wait before killing information provider processes. Default: `300`.

`bdii_breathe_time`
: Period in seconds between updates of the BDII. Default: `10` or `120` (compat).

`bdii_read_timeout`
: The amount of time to wait until an information is assumed to have timed out. Default: max collection cycle time or `300` (compat).

`bdii_delete_delay`
: Period in seconds before performing a delete. Default: `0`.

`bdii_update_pid_file`
: File in which to store the `bdii-update` PID. Default: `$bdii_run_dir/bdii-update.pid`.

`slapd_pid_file`
: File in which to store the `slapd` PID. Default: `$bdii_run_dir/db/slapd.pid`.

`registrationlog`
: Registration log file. Default: `/var/log/arc/inforegistration.log`.

`slapd`
: Path of the `slapd` executable. Default: based on `$PATH`.

`bdii_conf`
: Path of the generated BDII configuration file. Default: `/var/run/arc/infosys/bdii.conf`.

`giis_fifo`
: Path of the generated GIIS configuration file. Default: `/var/run/arc/infosys/giis-fifo`.

B.2 Section `[cluster]`

`cachetime`
: Age in seconds after which cluster entries are considered outdated. Default: `$bdii_breathe_time`.

B.3 Section `[queue/$queue]`

where `$queue` is the queue name.

`cachetime`
: Age in seconds after which queue entries are considered outdated. Default: `$bdii_breathe_time`.

B.4 Section `[se/$se]`

where `$se` identifies the storage element.

`cachetime`
: Age in seconds after which SE entries are considered outdated. Default: `$bdii_breathe_time` or max update cycle.

B.5 Section `[infosys/cluster/registration/$p]`

where `$p` is the VO name.

`targetsuffix`
: LDAP suffix of the target. Default: `Mds-Vo-name=$p,o=Grid`.

`targethostname`*
: Host name of the target.

`targetport`
: Port number of the target. Default: `2135`.

`registranthostname`
: Host name of the registrant. Default: `$hostname`.

`registrantport`
: Port number of the registrant. Default: `$port`.

`registrantsuffix`
: LDAP suffix of the registrant. Default: `nordugrid-cluster-name=$hn,Mds-Vo-name=local,o=Grid`.

`regperiod`
: Registration period to publish for cluster registration. Default: `120`.

`ttl`
: TTL to publish for cluster registration. Default: $2 \cdot$ `$regperiod`.

`timeout`
: Time limit to publish for cluster registration. Default: `45`.

`cachettl`
Cache TTL to publish for cluster registration. Default: `0`.

`sizelimit`
LDAP query result size limit to publish for cluster registration. Default: `0`.

`bindmethod`
Authentication method to publish for cluster registration. Default: `ANONYM-ONLY`.

B.6 Section `[infosys/se/$seentry/registration/$p]`

where `$seentry` is the name of the storage element and `$p` is the VO name.

`targetsuffix`
LDAP suffix for target. Default: `Mds-Vo-name=$p,o=Grid`.

`targethostname`*
Host name for target.

`targetport`
Port number for target. Default: `2135`.

`registranthostname`
Host name of registrant. Default: `$hostname`.

`registrantport`
Port number of registrant. Default: `$port`.

`registrantsuffix`
LDAP suffix of registrant. Default: `nordugrid-se-name=$sename:$hn,Mds-Vo-name=local,o=Grid`.

`regperiod`
Registration period to publish for SE registration. Default: `120`.

`ttl`
TTL to publish for SE registration. Default: $2 \cdot$ `$regperiod`.

`timeout`
Timeout to publish for SE registration. Default: `45`.

`cachettl`
Cache TTL to publish for SE registration. Default: `0`.

`sizelimit`
LDAP query result size limit to publish for SE registration. Default: `0`.

`bindmethod`
Authentication method to publish for SE registration. Default: `ANONYM-ONLY`.

B.7 Section `[infosys/index/$vo/registration/$r]`

where `$vo` is the index VO name and `$r` is the target VO.

`allowreg`

`Default: A pattern of external resources from which to accept registrations. This may be repeated.`

Mds-Vo-name=$r,o=Grid Default: `targetsuffix`. LDAP suffix of target.

`targethostname`*
Host name of target.

`targetport`
: Port number of target. Default: `2135`.

`registranthostname`
: Host name of registrant. Default: `$hostname`.

`registrantport`
: Port number of registrant. Default: `$port`.

`registrantsuffix`
: LDAP suffix of registrant. Default: `Mds-Vo-name=$indexname,o=Grid`.

`regperiod`
: Registration period to publish. Default: `120`.

`ttl`
: TTL to publish. Default: $2 \cdot$ `$regperiod`.

`timeout`
: Timeout to publish. Default: `120`.

`cachettl`
: Cache TTL to publish. Default: `0`.

`sizelimit`
: LDAP query result size limit to publish. Default: `0`.

`bindmethod`
: Authentication method to publish. Default: `ANONYM-ONLY`.

B.8 Section `[infosys/site/$site_bdii]`

`unique_id`
: A string identifying the site. Default: `$site_bdii`.

`url`*
: The URL of the BDII for the site.

B.9 Section `[infosys/glue12]`

`resource_location`*
: Free-form string describing the location of the resource.

`resource_latitude`*
: The latitude of the resource as a decimal number.

`resource_longitude`*
: The longitude of the resource as a decimal number.

`cpu_scaling_reference_si00`*
: CPU scaling according to SI00 benchmark.

`processor_other_description`*
: Description of the CPU as specified in the GLUE-1.2 standard.

`glue_site_web`*
: URL of the website belonging to the institution providing the resource.

`glue_site_unique_id`*
: A unique identification of the site also used in GOCDB.

`provide_glue_site_info`
: Whether to publish GLUE site info. Default: `true`.

Acknowledgements

This work was supported in parts by: the Nordunet 2 program, the Nordic DataGrid Facility, the EU KnowARC project (Contract nr. 032691), the EU EMI project (Grant agreement nr. 261611) and the Swedish Research council via the eSSENCE strategic research program.

References

[1] European Middleware Initiative (EMI). URL `http://www.eu-emi.eu`. Web site, 2010.

[2] Berkeley Database Information Index V5. URL `https://twiki.cern.ch/twiki/bin/view/EGEE/BDII`. Web site.

[3] gLite, Lightweight Middleware for Grid Computing. URL `http://glite.web.cern.ch/glite/`. Web site.

[4] D. Cameron A. Konstantinov. *The NorduGrid Grid Manager And GridFTP Server: Description And Administrator's Manual*. The NorduGrid Collaboration. URL `http://www.nordugrid.org/documents/GM.pdf`. NORDUGRID-TECH-2.

[5] A. Konstantinov. *The ARC Computational Job Management Module – A-REX*. The NorduGrid Collaboration. URL `http://www.nordugrid.org/documents/a-rex.pdf`. NORDUGRID-TECH-14.

[6] B. Kónya. *The NorduGrid/ARC Information System*. The NorduGrid Collaboration. URL `http://www.nordugrid.org/documents/arc_infosys.pdf`. NORDUGRID-TECH-4.

[7] M. Smith and T. A. Howes. *LDAP : Programming Directory-Enabled Applications with Lightweigt Directory Access Protocol*. Macmillan, 1997.

[8] A. Taga and Th. Frågåt. *ARC batch system back-end interface guide with support for GLUE2*. The NorduGrid Collaboration. URL `http://www.nordugrid.org/documents/Backends-arc1.pdf`. NORDUGRID-TECH-18.