EXPERIENCES WITH DATA INDEXING SERVICES SUPPORTED BY THE NORDUGRID ARC MIDDLEWARE

Dr. J. L. Nielsen*, Prof. J. R. Hansen, Dr. A. Wäänäänen, Niels Bohr Institute, Copenhagen, Denmark
Dr. A. Konstantinov, Vilnius University, Lithuania and University of Oslo, Norway
Prof. F. Ould-Saada, University of Oslo, Norway
Prof. P. Eerola, Dr. B. Kónya, Dr. O. Smirnova, Lund University, Sweden
Prof. T. Ekelöf, Uppsala University, Sweden
Dr. M. Ellert, CERN, Geneva, Switzerland

Abstract

The NorduGrid middleware, ARC, has integrated support for querying and registering to Data Indexing services such as the Globus Replica Catalog and Globus Replica Location Server. This support allows one to use these indexing services for for example brokering during job-submission, automatic registration of files and many other things. This integrated support is complemented by a set of command-line tools for registering to and querying these Data Indexing services.

In this article we will describe experiences with these indexing services both from a daily work point of view and in production environments such as the Atlas Data-Challenges 1 and 2. We will describe the advantages of such Data Indexing services as well as their shortcomings. Finally we will present a proposal for an extended Smart Indexing Service which should deal with the shortcomings described. Such an indexing service is being designed at the moment.

NORDUGRID AND THE ARC MIDDLEWARE

The NorduGrid project was initiated in 2001 by universities in the Nordic countries with the goal of developing a Grid-architecture meeting the requirements of scientists in the Nordic countries. Starting out as a grid-project with emphasis on batch processing suitable for problems encountered in High Energy Physics, it has now developed into a rather generic Grid system used by a growing number of scientists in all fields of science. The NorduGrid architecture is a light-weight, non-invasive and dynamic one, but robust and scalable enough to meet the most challenging tasks in science.

ARC (Advanced Resource Connector) is the Grid-middleware developed by the NorduGrid project. It is based on the Globus libraries and API but contains original architecture solutions, services and implementations. It now supports one of the largest Grid production system with 10 countries, more than 40 different sites, more than 5000 CPU's and about 30 TB of storage.

* langgard@nbi.dk

Features of the ARC middleware

ARC is currently built on top of Globus Toolkit 2 but with core services replaced. The alternate services developed by NorduGrid is

- The Grid Manager that
  1. Checks user credentials and authorization.
  2. Handles jobs locally on clusters (interfaces to the LRMS).
  3. Handles stage-in and stage-out of files automatically.
- Information System based on Globus MDS but with an extended NorduGrid schema.
- Lightweight User-interface with a builtin resource broker.
- The xRSL (extended RSL) job-description language.
- The Grid Monitor for live cluster- and job-monitoring — available at `http://www.nordugrid.org/monitor`.

For a thorough description of the NorduGrid ARC middleware, see [1].

INDEXING SERVICES

NorduGrid has the standard Globus Indexing Services deployed. The Globus Replica Catalog [2] has been deployed since spring 2002 and been used as the primary indexing server for about two years. It was in particular used during ATLAS Data-Challenge 1 to keep track of datafiles stored on various NorduGrid Storage Elements.

The Globus Replica Location Service [3] has been deployed and used since the beginning of 2004 and has been used as the primary indexing service in the ATLAS Data-Challenge 2. A detailed record of the NorduGrid involvement in ATLAS Data-Challenge 2 is given in [4].

In the following we will describe the features of these indexing services and the builtin ARC support for both.

The Globus Replica Catalog

The Globus Replica Catalog (RC) basically stores mappings between logical filenames and physical files stored on storage servers somewhere. The default implementation of the RC works against a standard LDAP server but the API is general so that other implementations for example based on SQL are possible.

The Globus Replica Location Service

As the RC, the Globus Replica Location Service (RLS) provides mappings between logical filenames and physical files. These mappings are stored in Local Replica Catalogs (LRCs) that send information about the mappings to Replica Location Indices (RLIs). Each RLI thus contains mappings from a logical filename to LRCs. When a client wants to find physical files corresponding to a logical filename, it first contacts the RLIs it knows about. These will respond with a possible set of LRCs where the logical filename can be found. The client then continues to query the LRCs and obtains the physical filenames from these. This architecture allows to setup the RLS as a distributed system with no single point of failure giving better performance and scalability.

ARC support for indexing services

The ARC Grid Manager and User-interface has builtin support for both the RC and the RLS. If for example the user specifies a logical filename formed as an `rc://` (in case of the RC) or an `rls://` (in case of the RLS) pseudo-URL, the User-interface will

- query the specified indexing service for the size and locations of the corresponding physical files.
- use the information obtained in the resource brokering.

Furthermore when a job has been submitted with the specification of such a pseudo-URL, the Grid Manager on the chosen cluster will

- query the specified indexing service for the physical filename.
- download the physical file to the cluster — or if the file already exists in the local cluster's cache, create a symbolic link to the session-directory of the job.
- possibly register in the indexing service the downloaded physical file as being available in the local cluster cache. This can later be used by the User-interface during brokering.

Additionally the Grid Manager can at the end of a job automatically

- obtain a list of available Storage Elements by querying the indexing service.
- upload output-files to one of these Storage Elements.
- register the output-files in the indexing service with given logical and physical filenames.

Further information about this can be obtained from [5].

Command Line Tools

ARC has integrated support for the RC and the RLS through a set of data-management command-line tools. These tools makes daily task operations on the indexing services much simpler for the average user.

The command-line tools include

- `ngls` — lists physical files and their attributes. The user can either specify a logical or physical filename.
- `ngcopy/ngrequest` — copy and replicate files registered in indexing services. Again the user can either specify a logical filename or physical filename. Multi-threaded and 3rd party transfers are supported as well.
- `ngremove` — can delete physical files and logical records in one operation.

EXPERIENCES

In this section we will describe some of the experiences with the above mentioned indexing services. Emphasis will be on experiences obtained during NorduGrid production in the ATLAS Data-Challenges.

Using the RC in ATLAS Data-Challenge 1

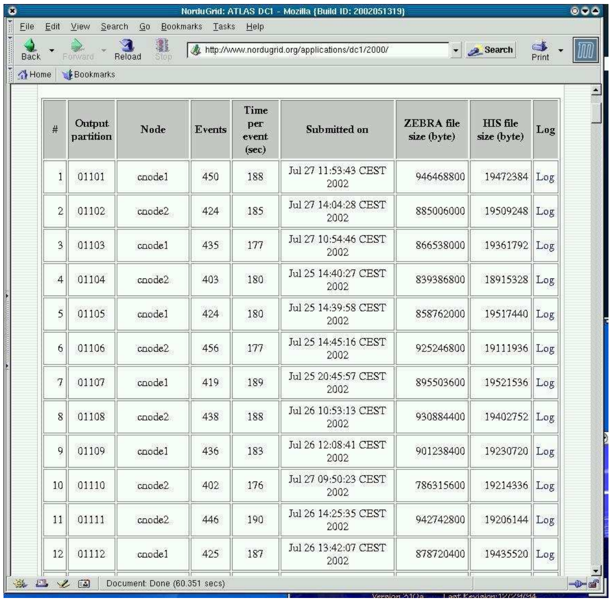
NorduGrid participated in ATLAS Data-Challenge 1 and contributed with about 5 percent of the total worldwide production. We used the RC as the primary indexing service during the whole of this Data-Challenge. A snapshot of the Replica Catalog Browser that was used during the production is shown in figure 1. See [6] for a thorough account of this Data-Challenge on NorduGrid.

Unfortunately during this Data-Challenge, the RC crashed at random times and had to be restarted often. In the end, we had to run a script that checked whether the RC was responding and restart it if it was not. This was not a completely satisfactory solution.

In the default installation, the RC requires cleartext passwords for authentication which is hardly optimal. However, this was patched by NorduGrid to allow for ordinary GSI-certificate-based authentication.

The RC supports file-collections and putting arbitrary attributes on logical files which is essential. But despite its support for collections, the RC is not very flexible since a file can only belong to one collection at a time.

Furthermore the RC command-line tools are not very easy or logical to work with.



#	Output partition	Node	Events	Time per event (sec)	Submitted on	ZEBRA file size (byte)	HIS file size (byte)	Log
1	01101	cnode1	450	188	Jul 27 11:53:43 CEST 2002	946468800	19472384	Log
2	01102	cnode2	424	185	Jul 27 14:04:28 CEST 2002	885006000	19509248	Log
3	01103	cnode1	435	177	Jul 27 10:54:46 CEST 2002	866538000	19361792	Log
4	01104	cnode2	403	180	Jul 25 14:40:27 CEST 2002	839386800	18915328	Log
5	01105	cnode1	424	180	Jul 25 14:39:58 CEST 2002	858762000	19517440	Log
6	01106	cnode2	456	177	Jul 25 14:45:16 CEST 2002	925246800	19111936	Log
7	01107	cnode1	419	189	Jul 25 20:45:57 CEST 2002	895503600	19521536	Log
8	01108	cnode2	438	188	Jul 26 10:53:13 CEST 2002	930884400	19402752	Log
9	01109	cnode1	436	183	Jul 26 12:08:41 CEST 2002	901238400	19230720	Log
10	01110	cnode2	402	176	Jul 27 09:50:23 CEST 2002	786315600	19214336	Log
11	01111	cnode2	446	190	Jul 26 14:25:35 CEST 2002	942742800	19206144	Log
12	01112	cnode1	425	187	Jul 26 13:42:07 CEST 2002	878720400	19435520	Log

Figure 1: A snapshot of the Replica Catalog Browser used during Data-Challenge 1

Using the RLS in ATLAS Data-Challenge 2

In the ATLAS Data-Challenge 2 production, we used Globus RLS as the primary indexing service. The RLS has a reasonably logical client API which makes building RLS-client-interfaces easy. We used this in the Dulcinea executor as desribed in [4].

RLS by default uses GSI for authentication but can also be setup to allow anonymous queries. In our experience, the RLS is generally very responsive — answering even very complex wildcard and attribute queries in a matter of seconds.

However, during the production RLS suffered from some of the same diseases as the RC. The RLS would freeze at random times several times a day not responding to queries anymore. Therefore we had to construct, as for the RC, an automatic restart-script that pinged the RLS and if it did not respond, kill it and restart it. Recently a patch has been released by the RLS-developers that should solve this problem. After applying this patch, our RLS has now been stable for more than a month.

RLS has no builtin support for logical collections but instead assumes that this is provided by an additional metadata service. Such a metadata service do not exist at present time. Therefore the use of special file-attributes is the only way to support collections within RLS. However, this approach is somewhat artificial and not entirely satisfactory since the notion of collections becomes very VO-specific. It would be much nicer to have a common notion of a collection in RLS.

Also the RLS command-line interface, `globus-rls-cli`, as for the RC, is not very logical to work with — often requiring guesswork to get the commands right.

To sum up, neither the RC nor the RLS has been satisfactory neither during production or in daily work and lacks some of the functionality one would expect to see and need to have in an indexing service.

FUTURE INDEXING SERVICES

Neither of the indexing services described do satisfy the requirements that an indexing service needs to satisfy to be able to cope with the amount of data that is produced when LHC start.

Therefore NorduGrid is at the moment designing a new indexing service, a so-called Smart Indexing Service (SIS). The following is a non-exhaustive list of design-requirements:

- It must have GSI authentication.
- It must have support for logical collections.
    1. with no restrictions on how many collections each file can be in.
    2. with support for sub-collections and empty collections as well.
- It must have support for arbitrary file-attributes.
- It must have fine-grained access-control — with the possibility of giving read- and write-access to collections based on a user's certificate.
- The access-restrictions of the physical files on the Storage Elements should be reflected in the indexing server.

Such a SIS would naturally work together with the Smart Storage Element developed within NorduGrid [7] and thus would consistute a second step towards a fully functional data-management system meeting the requirements imposed by LHC-data-analysis.

REFERENCES

[1] Paula Eerola et. al., "Building a Production Grid in Scandinavia", IEEE Internet Computing, July/August 2003.

[2] http://www-fp.globus.org/datagrid/replica-catalog.html

[3] Ann L. Chervenak, Naveen Palavalli, Shishir Bharathi, Carl Kesselman, Robert Schwartzkopf, "Performance and Scalability of a Replica Location Service", Proceedings of the International Symposium on High Performance Distributed Computing Conference (HPDC-13), June 2004.

[4] Mattias Ellert et. al., "Performance of the NorduGrid ARC and the Dulcinea Executor in ATLAS Data-Challenge 2", CHEP 2004, Interlaken, contribution no. 499.

[5] Aleksandr Konstantinov, "The NorduGrid Grid Manager and GridFTP server: Description and Administrator's manual". http://www.nordugrid.org/papers.html.

[6] Paula Eerola et. al., "ATLAS Data-Challenge 1 on NorduGrid", in Proc. of the CHEP'03, PSN: MOCT011 (2003)

[7] Aleksandr Konstantinov et. al., "Data management services of NorduGrid", CHEP 2004, Interlaken, contribution no. 294.