Grid Service Information Discovery

Joint Workshop: EGEE/OSG/NorduGrid

Copenhagen, Denmark

December 12-14, 2006

Trip Report by Steven C. Timm, Fermilab

Introduction:

This workshop grew out of discussions between the staff of EGEE (Enabling Grids for E-sciencE), NorduGrid, and OSG (Open Science Grid). All of these grids are looking for a common and simple Service Discovery process, and all agree there is a need to have one that can interoperate across grids. The workshop's agenda was to review experience with current deployments, determine the high-level requirements for a new system, and plan how it might be accomplished.

Part of my motivation to be at the workshop was to understand the implications to FermiGrid and also alert the developers to the existence of job-forwarding services such as ours. FermiGrid has a stake in the outcome of any service discovery protocol since we would likely have to write an emulation for it, given the implementation of our site Globus job gateway.

To give an idea of the challenge, the current size of EGEE has 200 production sites, each with compute element, storage element, BDII, and monitoring services at least. They may run other services as well. Currently there are 60 Virtual Organizations (VO's). The anticipated scale of EGEE in 3 years is expected to be over 1000 sites, 20 different services, 200 VO's, 10 roles and groups apiece, 40 x 10^6 pieces of metadata.

The workshop was held at the Nordic DataGrid Facility in Copenhagen, Denmark, and representatives of all three grids attended as well as some key middleware developers. The list of participants follows:

Laurence Field, (CERN/EGEE), Steve Fisher (RAL/EGEE), Antony Wilson (RAL/EGEE), Arumugam Paventhan (RAL/EGEE), Markus Schulz (CERN/EGEE), Steven Timm (Fermilab/OSG), Shaowen Wang (UIowa/OSG), Sergio Andreozzi (INFN/OMII), Weijian Fang (U. of Southampton/OMII), Oxana Smirnova (Lund/NorduGrid), Balazs Konya (Lund/ARC/NorduGrid), Laura Perlman (ISI/Globus), Andriana Iamnitchi (USF), Lydia Prieto (USF).

All the participants in the workshop agreed on a common statement for the directions that the grids should jointly take in developing the next generation service discovery. This is reproduced at the end of this report. This report will cover the highlights of the discussion which led to the decisions that were made, drawing on my own notes as well as the notes of the meeting scribe, Laurence Field.

Existing Services:

The first day of the meeting was spent in going over existing service discovery and information services that are now deployed in various grids. The goal was to identify similarities between them and identify known problems of the current services.

Globus Monitoring and Data Service (MDS2) has been used in LCG, OSG, and NorduGrid. It is now deprecated by Globus Alliance. The initial implementation used the GRIS (Grid Resource Information Service) and the GIIS (Grid Index Information Service). This was a distributed service with one GRIS per resource and one GIIS per site. In all its various implementations it suffered from scalability and stability problems. GIIS was used to cache the information, but the total information is huge, 19MB currently for EGEE, 35MB from NorduGrid. NorduGrid had an intelligent client for MDS2 which had a timeout for misbehaving sites. This is part of the ARC (Advanced Resource Connector) middleware.

BDII (Berkeley Database Information Index) is now used as the front end to the information system, replacing the GIIS and eventually the GRIS as well. It begins with a registry of sites and caches information every two minutes. In LCG implementation there is a BDII at every site. The site BDII grows in size with the number of other BDII's

that are querying it. OSG will have a single BDII for whole OSG and advertise to it with CEMon (Computing Element Monitor). Again the caches are large.

Globus MDS4 (web service) doesn't run information providers on demand, just on a fixed interval. It has central index servers and it is possible to make custom indexes, with periodic caching. The data is stored in an XML database and queries are done using Xpath. The one problem that was mentioned is that the XML database takes up a lot of space in memory. Globus toolkit includes Java and C interfaces, a web based client, and a command line client. Largest known implementation is the TeraGrid. Globus developers have written some information providers. Modular Information Providers for the Open Science Grid have been written in beta version at U of Iowa as well.

R-GMA (Relational Grid Monitoring Architecture) is part of gLite/EGEE and uses a relational database to store the various information from grid schemas, from whence it can be retrieved via SQL queries. It is possible to authenticate with grid certificates and configure the views to define what parts of a table any given user can see. However, there are reports of connection timeouts and too much load on the registry, and it has difficulty handling the conditions of large numbers of jobs in the system simultaneously. Improvements have been proposed to deal with these and other problems.

Grimoires: Grid RegIstry with Metadata Oriented Interface: Robustness, Efficiency, Security. This is an Open Middleware Infrastructure Institute-UK (OMII-UK) funded project from the University of Southampton. Grimoires can deploy as a web services container within OMII, GT4, and Tomcat/Axis frameworks. The service registry is based on UDDI (Universal Description Discovery and Integration) framework but extends the framework.

Service Discovery in gLite: gLite includes a Service Discovery plugins to the R-GMA service mentioned above as well as for the BDII. There is a command line interface and also an API. The Simple API for Grid Applications (SAGA) is currently using this API for service discovery since it has no native one of its own.

NAREGI (National Research Grid Initiative) uses CIMOM, the CIM Object Manager, which distributes information about compute elements based on the Common Information Model, an emerging industry standard. This is then aggregated to a relational database and implemented as a grid service by use of Globus service OGSA-DAI (Open Grid Services Architecture Data Access and Integration). Each site is referred to as a Cell Domain.

One of the main outcomes of this discussion was the surprising similarities in each system. Most used an index which contained information of the site level interface. The system would then pull the information from the site level. On a conceptual point of view the site could be represented by a database and an interface. One of the main differences between the different systems is how grid level caching is handled. Some common use cases of the systems were walked through to demonstrate the kind of information that needed to be found, and the frequency of its update.

As far as security is concerned, most of the above systems have the capacity to use GSI authentication to retrieve the information, but do not do so due to the extra overhead that this places on the information process. All seem to think it is desirable and will eventually have to be re-implemented.

Service Discovery and Query:

The second day of the meeting focused on what the definition of Service Discovery is and what functionality it ought to have. To do this, we need to define the questions which Service Discovery can ask and the answer which is returned. There is general agreement that the questions which could be asked were those which are generic for all services, and the answer would be one of two handles, the Service Access Point and the Information End Point.

Closely related to this is the question of what a service is and what a resource is. All resources are made visible to the grid via services but in general there can be a many to many relationship between services and resources. For instance, a single Compute Element resource might have an OSG gatekeeper service and an LCG gatekeeper service. Likewise you could have a site gateway service which is associated with more than one Compute Element resource. It follows that resources will have to have unique identifiers as well, and resource discovery can be viewed as a reverse lookup operation to service discovery.

There is a need for a common Service Discovery API. S. Fisher of Rutherford presented the work his group intends to do on this API. They proposed that this work be done within the SAGA (Simple API for Grid Applications) activity of the Open Grid Forum. SAGA currently does not have a Service Discovery API. Once the API is developed, a plugin specification would also be needed to enable the API to be used with multiple grid

systems. A set of plugins was developed as part of the OGF gin-info grid interoperability activity which could be adapted for this use.

It was agreed that an information provider interface would be useful so that the developers of the underlying systems, e.g. batch systems and storage systems, can maintain the provider. The interface should be simple. A command that produces XML on standard out would suffice. This interface could be used both at the provider and the plugin level. Whatever interface is designed would need to produce the information that is defined by the schema and caching should be employed to protect against overloading the underlying resource.

Service Query:

Once it was agreed that service discovery would be a top-level activity, discussion then turned to the question of how to obtain information about a service that is service-specific. This is sometimes referred to as service selection or resource selection but for the terms of this meeting we decided to refer to it as service query. It may involve polling the services themselves or a site-level database such as a BDII. Some services have their own API (e.g. VOMS= Virtual Organization Management System) others don't, (e.g. FTS= File Transfer System).

Issues involved in service-specific queries are (a) defining a schema for each service type (b) identifying which attributes would be considered static and which are dynamic, and (c) determining whether the information returned by the query can be reduced to key-value pairs. For some services the GLUE schema is sufficient, for others new schema will have to be defined. "Static" attributes are considered to be those that stay the same for 6 hours or more. Schema design needs to plan for moving the dynamic values around with more frequency. It was thought that some of the service-specific queries cannot be reduced to key-value pairs.

For the service query interface, we considered extending the Service Discovery API, making a common and separate query interface, or making a service-specific API for each service. The complexity of some of the queries precludes extending the Service Discovery API, which relies on key value pairs. Service-specific API's would be very sensitive to schema changes. The conclusion is that a common query interface would be required, however it is not clear what this should be or how it can be agreed on.

The final part of the meeting discussed bootstrapping, namely where does the service discovery get the initial information it needs. The earlier investigation of the common features of the grids shows that each grid has a top level aggregation service. The end points of these aggregators need to be passed to the configuration of the Service Discovery API so that it can find these aggregators. One possibility is that the VO should know these, as they have already negotiated with the infrastructure to gain access. Grid infrastructures should look at the VO's they support to understand which Grids they need to interoperate with and they can get the endpoints from the VO. An agreed format may need to be worked out.

Consensus Statement:

1) We agree that there is a need for a common way to do Service Discovery
2) A Service Discovery API is needed and this work will be done in the SAGA working group within OGF.
3) This will need a generic description of the Service and this will be defined with the GLUE working group within OGF.
4) The gin-info group could help to provide the required plugins.
5) There is a need for Service-specific schema. This will also be defined in the GLUE working group within OGF
6) A common information provider interface would be helpful. It was decided that this would just be a command which returned XML. There is no need to make this an official standard but it would help if this recommendation was adopted so that information providers could be shared and possibly developed by the developers of the underlying system which is being queried
7) It would be a good idea to set up a community repository which can be used to share plugins etc.
8) In principle it was agreed that a common query API would be required, however it wasn't clear how this would be decided.

Next Steps:

In the Open Grid Forum 19, currently taking place in Chapel Hill, NC, there is a session devoted to the Service Discovery API within the SAGA framework. Various funding proposals have been submitted both to European funding agencies and to US funding agencies. One European group is seeking funding to work on the above-mentioned Service Discovery API within SAGA; another US group is seeking funding to work on

Modular Information Providers for use in Globus MDS4. The Europeans expected an answer in January on whether their proposal would be approved.

As mentioned in the trip report filed by the University of South Florida researchers, they have laid out a plan of getting real usage traces from real user behavior on the grid. These traces would not be limited to information systems but include all the various resources that are used in a typical grid use case.

Informal discussions with NorduGrid staff present at the meeting indicated a willingness to continue exploring ways to collaborate with OSG on information services. It is unlikely that the next step in this collaboration will be known until some of the work on the Service Discovery API is actually done within the OGF. It is my understanding that NorduGrid is planning to base whatever their next software release will be on the common API when it comes out.

Although the consensus statement above doesn't explicitly reflect it, most of the participants agreed that not much could be done on the harder problem of service and resource selection until the service discovery API is first defined in the OGF. That is why #8 above refers to a common query API being required but not being unclear about its implementation. This common query API refers to that process, not to service discovery.

Useful Web Links

MDS2 http://www.globus.org/toolkit/mds
BDII https://twiki.cern.ch/twiki/bin/view/EGEE/BDII
MDS4 http://www.globus.org/toolkit/mds4
R-GMA http://www.r-gma.org
Grimoires http://www.grimoires.org
gLite http://glite.web.cern.ch/glite
NAREGI http://www.naregi.org/index_e.html
CIM http://www.dmtf.org/standards/cim
OGSA-DAI http://www.ogsadai.org.uk/
SAGA https:/forge.gridforum.org/projects/saga-rg
GLUE http://glueschema.forge.cnaf.infn.it/
EGEE Service Discovery presentation:
https://twiki.cern.ch/twiki/bin/view/EGEE/ServiceDiscovery/2006-09-05-EGEE-SD.pdf
EGEE: http://www.eu-egee.org

NorduGrid: http://www.nordugrid.org
OSG: http://www.opensciencegrid.org
TeraGrid MDS:
http://mds.teragrid.org:8080/webmds/webmds?info=indexinfo&xsl=tg_gluesummaryxsl