UMD User and Operation Working Group Recommendations – Draft

Summary and Introduction

The UMD (Universal Middleware Distribution) User and Operation Working Group has been appointed by the three main European grid infrastructures: DEISA, EGEE and NDGF. The working group consists of:

From DEISA:

- Stefan Heinzel
- Denis Girou

From EGEE:

- Steve Traylen
- Johan Montagnat

From NDGF:

- Josva Kleist
- Michael Gronager

Further from EGI_DS Roberto Barbera has contributed.

The working group has been active since February the 24th 2009, and has worked according to the following ToR:

> *Collect wishes from developers, operations and users on missing functionalities and define a roadmap for the future UMD evolution and comes up with concrete ideas for future UMD developments*
>
> *Outcome: wish list for 2 years and vision for 5 years in time for the workshop*

Summary

The group is of the opinion that it is primarily activities on common user administration, authentication, authorization and accounting and the Scientific Portals that needs to be well coordinated by the middleware activity. As for many of the other activities they need more detailed case by case study before a common ground can be distilled.

The vision on the longer scale is that all domains can use the same infrastructure services – a service should not be considered relevant for just either HPC or HTC (e.g. common accounting, identity infrastructure), and to promote use of de facto tools with add ons rather than duplicating functionality in infrastructure specific tools

1 Background

This section gives the background of the main European e-Infrastructures.

1.1 DEISA

DEISA is operating a heterogeneous HPC infrastructure currently formed by eleven European national supercomputing centres that are tightly interconnected by a dedicated high performance network. The term 'heterogeneous' refers to the variety of HPC system architectures, operating systems, batch schedulers and local file systems provided by the DEISA supercomputing centres and that is typical for an HPC ecosystem.

DEISA is structured as a layer on top of the national supercomputing services by providing generalized interfaces and services that allow to access and utilize this pool of computing resources in a consistent way and therefore more efficiently. That way, the DEISA HPC infrastructure and services combine - for users and user communities - the advantage of having access to a variety of supercomputing architectures for different demanding computing purposes with the advantage provided by consistent interfaces to these different resources and services.

Accordingly, the profile of using the DEISA HPC infrastructure and services can be regarded as being similar to the usage of a monolithic supercomputing system of single HPC centre.

1.2 EGEE

The EGEE project brings together experts from more than 50 countries with the common aim of building on recent advances in Grid technology and developing a service Grid infrastructure which is available to scientists 24 hours-a-day.

The project provides researchers in academia and business with access to a production level Grid infrastructure, independent of their geographic location. The EGEE project also focuses on attracting a wide range of new users to the Grid.

The project's main focus is:

- To expand and optimise Europe's largest production Grid infrastructure, namely EGEE, by continuous operation of the infrastructure, support for more user communities, and addition of further computational and data resources.
- To prepare the migration of the existing production European Grid from a project-based model to a sustainable federated infrastructure based on National Grid Initiatives for multi-disciplinary use.

1.3 NDGF

The Nordic Data Grid Facility, NDGF, is a collaboration between the Nordic countries (Denmark, Finland, Norway, Sweden).

The motivation for NDGF is to ensure that researchers in Nordic countries can create and participate in computational challenges of scope and size unreachable for the national research groups alone.

NDGF is a production grid facility that leverages existing, national computational resources and grid infrastructures.

To qualify for support research groups should form a virtual organization, a VO. The VO provides compute resources for sharing and NDGF operates a grid interface for the sharing of these resources.

Currently, a significant fraction of Nordic resources are accessible with ARC and gLite grid-middleware, some sites with both.

1.4 Future European e-Infrastructures

The European Grid Initiative (EGI) Design Study represents an effort to establish a sustainable grid infrastructure in Europe. Driven by the needs and requirements of the research community, it is expected to enable the next leap in research infrastructures, thereby supporting collaborative scientific discoveries in the European Research Area (ERA).

The main foundations of EGI are the National Grid Initiatives (NGI), which operate the grid infrastructures in each country. EGI will link existing NGIs and will actively support the setup and initiation of new NGIs.

The goal of the EGI Design Study (EGI_DS) is to evaluate use cases for the applicability of a coordinated effort, to identify processes and mechanisms for establishing EGI, to define the structure of a corresponding body, and ultimately to initiate the construction of the EGI organization (EGI.org). The EGI Design Study is a project funded by the European Commission's 7th Framework Programme.

2 Requirements

This section covers the user requirements as collected by the infrastructures. The requirements form the infrastructures reflect that the infrastructures are quite different:

- DEISA: A few very large sites of heterogeneous sizes.
- EGEE: Many homogeneous sites of various sizes.
- NDGF: Many heterogeneous sites of various sizes.

The access pattern from the users of the DEISA infrastructure is also quite different from the access pattern seem in EGEE and NDGF. DEISA users needs access to targeted single large installations using either direct terminal access or UNICORE where appropriate. The typical EGEE and NDGF user does not target their jobs to a specific resource but does instead leave it to the logic of the grid middleware (brokering) to choose an appropriate site.

This suggests that the problems tackled by the NDGF and EGEE users are of a more standardized type; the same application is installed at several sites and ran over and over again with different input data and parameters. DEISA users are typically running much more challenging codes, OpenMP and MPI are taken for granted and the direction for DEISA is towards even more challenging problems (Grand Challenge Projects) where fault tolerance inside one application running on multiple cores due to the immense number of cores will become important in few years. On the other hand DEISA is also tackling challenges from Virtual Communities. The latter falls into a category much closer to the problems of Virtual Organizations also seen in the EGEE and NDGF infrastructures. E.g. within NDGF some VOs are routinely running MPI jobs on 32-256 cores at several sites. It seems like some common ground exists in exactly this domain.

Due to some of the differences mentioned above the requirements to the middlewares used by the three infrastructures also differ and are on different levels of maturity and deployment. Below a table listing some of the user requirements is listed per infrastructure.

Area/Infrastructure	DEISA	EGEE	NDGF
Portability and interface to local services	Continuous support for multiple OS platforms and Batch systems, advanced reservation	Support for more linux distributions and platforms (esp. client side). Job priority and short job turn around time improvements. Advanced reservation and resources pre-emption.	Support for more unix flavors. Tighter coupling with the batch systems (access to more features).
Resource interface and adaptation	Better support of hybrid programming, supporting high end MPI features for fault tolerance on modern PFLOP systems	Better support for MPI, including advanced reservation for MPI nodes. Description of resources.	Better topology description for MPI jobs. Continuous support for MPI applications.
User Administration, Authentication, Authorization and Accounting	PKI, Accounting	User identification policy for grid portals.	Hiding proxies for the user. Accounting independent of access method.
User Interface	Promoting a uniform resource interface, Scientific Portals	Scientific Portals, grid client API, Interactive jobs.	Scientific Portals, VO Manager privileges
Data handling	Global File Systems, GridFTP, Staging	Transparent distributed file system. WS APIs to data management services and POSIX file access, encrypted file system support.	Improvements of data caching and staging.
Operation	Precise monitoring of all core services	Reliability. Improved logging, documentation, infosys-m/w integration, configuration. Short deadline jobs.	Instrumentation, less specialized tools.

2.1 DEISA, details on requirements

2.2 EGEE, details on requirements

Portability and interface to local services

Support of multiple systems is of high importance for EGEE users. In particular, the user client (EGEE User Interface) should be ported to different operating systems (linux, windows, macosx) to help developing grid applications in the usual users environment. Support of different OS for computing nodes is sometimes necessary to run specific software, although it is a less pressing requirement.

Job priority and job turn around time may highly impact applications with short jobs: a large number (thousands) of possibly short (minutes) jobs is encountered. The middleware must be robust enough to hold this load and it must not penalize such applications too much (e.g. by providing a fair jobs prioritization scheme or by using pilot jobs).

Some time-sensitive applications, especially parallel jobs, require advanced reservation to ensure that the code can run in a specific time window. Note that advanced reservation also addresses the problem of interactive jobs timely execution (see interactive jobs in "user interface" below). Some critical applications (e.g. risk protection) even require resources pre-emption to ensure immediate allocation of sufficient amounts of resources.

Resource interface and adaptation

Parallel execution is mandatory for many grid applications. The middleware is expected to enable parallel job submission (by specifying a number of CPU cores to allocate at submission time), at least at a site scale and provide the MPI and OpenMP environments. Some applications also require support to access multi-core and FPGA resources. Users should be able to query the system for the maximum number of concurrent processes that can be allocated on each site. In addition, it is very important to know whether the file system is shared among the nodes: the behavior of the program will differ when accessing files (e.g. if each parallel process is writing to a same file name, there is an overlap if the file system is shared while there is no collision risk if it is not shared).

User Administration, Authentication, Authorization and Accounting

A common authorization and authentication mechanism for all grid services, with single sign-on is of outmost importance.

Getting access to the grid is often considered as a cryptic and long road for new comers. X509-based PKIs are fine in setting up large scale trust infrastructures but users should be shielded as much as possible from the internal manipulations related to certificate delivering and client-side management.

Portals play an important role for helping users to discover and access to grid resources. In some communities, some services are operated openly and anonymously. Service certificates are needed for high level interfaces such as web portals that offer grid access to users with a proper identification policy (e.g. anonymous access to a restricted set of algorithms). In other cases, strict access control (VOMS-based, ACL/RBAC-based) to resources and data is needed. The user identification policy behind a given portal certificate must be easily obtained and completely documented. (E.g. truly anonymous access, access with a known email address behind, access with username/password, etc).

User Interface

The programmatic interface to grid middleware is also critical for all application or high-level services development. Three programming interfaces can be considered: APIs, command line interfaces and graphical portals. APIs is by far the most important as commands lines and portals can be easily derived from APIs. The difficulty here is in getting a coherent API covering all middleware services (heterogeneity of middlewares easily lead to heterogeneous API formats and languages). Similarly, clear, coherent and documented error codes need to be returned from the different services in case of failure. The most important APIs today are Web Services/WS-*, Java and C/C++.

It must be possible to execute interactive jobs (jobs with a communication between the execution host and the user interface). The communication may be shell-based or application-specific (it should be possible to open a socket to transfer interactive feedback according to the application protocol). The middleware has to ensure that the interactive jobs are started at a precise time (the user has to be available when the interactive application starts).

Data handling

EGEE applications have many requirements concerning data manipulation, the most fundamental being a complete transparency of the distributed storage resources through a virtual global file hierarchy. This requirement is partly achieved by the file catalogs today but the users are still very exposed to multiple storage resources (when uploading files to the grid, replicating files and managing storage space on a per-resource basis). For the ease of porting applications, a POSIX-like data access mechanism is also of high importance. Data security is critical for many applications and an ACL/RBAC -based access control system (exploiting user DN as ACL items) and data encryption capabilities (on-disk and on-network) are required.

Applications often manipulate metadata stored in relational databases. A grid-compliant metadata management system is needed, including secured access (ACL-based) to the relational databases is required. Federation of multiple databases is of high interest for some applications.

It must be possible to specify data required by a job. The job submission mechanism should ensure that the data is accessible without further work once the job is started (e.g. automatic data replication on a SE accessible by the given protocol, local copy of data accessible by POSIX file access, transparent access to remote data through a POSIX-like interface, etc).

Operation

Reliability of grid middleware and resources is probably the highest priority requirement. Applications are very often facing non-negligible error rates (usually a combination of middleware errors and configuration issues) that lead to application specific development for improved reliability.

Fast, easy site installation and verification procedures for encouraging new and small sites joining the grid are needed. Testing and Monitoring of Grid Components with automatic alerts and middleware black-listing is important to improve reliability.

The ability to deploy and maintain multiple versions of application software to several grid sites coherently is required to ease the deployment of application services.

Users often encounter difficulty in identifying the up-to-date and complete documentation of the middleware components and their interface. The access to the information systems to identify the infrastructure capabilities is also of high interest (storage space, jobs supported, resources available, software available...).

Sites supporting Short Deadline Jobs (SDJs) are of high interest for many purposes (applications with very short tasks, testing and demonstration needs) but very few sites are configured accordingly.

Operational support is needed for several licensing models. Matlab is used in many different applications in particular. This support may include multiple-level access control mechanisms (per-site, per-installed software...).

2.3 NDGF, details on requirements

Portability and interface to local services

The resources abstracted by the middleware within the NDGF infrastructure spans several different OS'es, a large number of different batch systems and also quite different site hardware setups. It is vital for NDGF that the current level of portability in OS systems, in different batch systems and in flexibility when it comes to different hardware configurations are kept in ARC and/ or the relevant UMD components. Further, even tighter coupling between the batch system, the hardware and the grid abstraction layer is needed, especially for better topology publication, for improved accounting flexibility and support for flexible prioritization of jobs.

The current level of client portability is considered sufficient, and optional support for e.g. Windows and Mac OS X is considered interesting but not a high priority.

Resource interface and adaptation

As hardware setup is becoming increasingly complex, with multiple levels of parallelism, from multiple cores to different types of cluster interconnects, it is important that the grid abstraction layer moves from being an abstraction layer for multiple single core resources to become an abstraction layer for multiple multi core and multi node resources with exposure of the interconnect type and topology. Single core job should be considered a rare use case, and the more complex multi core/node job the normal and favoured job request. Again a closer link between the batch system / scheduler and the grid abstraction layer is the way forward.

Parallel applications optimized per site should be considered the normal use case and general MPI support only fall back. Further, support for optimization and compilation of applications at the resource frontends should be strived further.

User Administration, Authentication, Authorization and Accounting

The community should move towards more de facto standards for handling AAAA. E.g. for Authentication the current federated approaches based on institutional IdPs should be integrated further, and in general work done by other projects should be integrated and used. It should be as simple to authenticate on the grid as it is today e.g. to buy books at amazon. On the European scale an initiative from TERENA on a common federated id service should be the basis. This initiatives also scales from portals to api and client use.

When it comes to user administration and accounting it is important that use is granted and accounted for per id / group / role independently of the access method. I.e. it is the same system that handles ssh based access and grid VO-based access.

User Interface

The current client api from ARC is capable of submitting jobs to several different grids and is an excellent tool for creating portals as well as integration grid into applications.

A better, more standardized, scheme for separation of privileges between VOs, and between managers and users within a VO is needed. The effort in communicating smaller changes in the setup to single sites will not scale for multiple VOs at a large number of sites and hence some convention is needed here.

Data handling

The current data handing in ARC is considered sufficient for the next years. It includes staging data from SEs before job start and staging data back after the job has finished, and setting the appropriate ACLs on the data. Further, the registration in suitable catalogues is today adequately supported. It is important to continue to utilize the data handling systems already optimized at the sites, like e.g. various cluster file systems in the job work flow. Further, it is important to contribute to emerging de facto standards like NFS 4.1 when it comes to local and wide area POSIX file access.

Operation

The current high stability and reliability of the middleware used at the infrastructure operated by NDGF is considered highly important and should not be jeopardized by any new feature addition or protocol change. Further, a move towards isolating the critical added value per component and add this e.g. as a plugin to widely used open source tools could further improve the stability and minimize the development and deployment costs.

Tools for measuring the current state of the single components, like throughput, job frequency i.e. service instrumentation would be beneficial for the operation team. Again, this should preferably be done as plugins to existing de facto service instrumentation tools rather as a standalone new project.

Finally, the possibility for the operation team to issue high priority debug jobs with short turn around time, as well as better access to the site logs could ease the task of ensuring smooth infrastructure wide operation.

3 Common feature list

This section gives the common list of features requested by the users over the next 2 years. Reliability is a long-going and prominent requirement both for the short and long term.

It is important to stress again that as the scope of the infrastructures are somewhat different and some of the commonalities apparent from the table in Section 2 might still be different in scope though similar in theme. E.g. better exposition of resource topology and features (MPI) means for DEISA to explore and setup new possibilities for users to use MPI on massively parallel systems, for the homogeneous EGEE infrastructure it means a deployment project to enable MPI at all sites. For NDGF MPI is seen mainly as an added feature per specific application and hence it is again some other improvements that are strived for.

However, still a continuous push for uniform interfaces to the resources that exposes the internal capabilities and topologies of the resource better is common for the infrastructures. Further, support for a great variety of OS'es and OS versions is important, as is tighter integration with the local batch systems. A concise set of clearly specified APIs is also very important for applications and high-level middleware development.

Another important infrastructure activity, potentially crossing all three infrastructures is the administration of users, their authentication, authorization and the need for an accounting system independent of the access method, but hierarchical and federated supporting most optimally the administrative domains in a common European grid infrastructure.

Data management is a very common requirement, shared by several infrastructures. It covers first system-wide virtual file hierarchy as well as transparent data storage and access with a high level of security (access control, encryption). Data transfer capabilities between infrastructures is needed to ensure inter-operability between NDGF, DEISA and EGEE.

The Scientific Portals is also mentioned by all the infrastructures as important for the users, further they have the potential for enabling a cross infrastructure use hidden for the users.

4 Conclusions and Visions

This section concludes on the document and gives the future 5 year vision.

Mainly activities on common user administration, authentication, authorization and accounting and the Scientific Portals are activities that should be well coordinated by the middleware activity. As for many of the other activities they need more detailed case by case study before a common ground can be distilled.

Putting the vision in bullet form, the group would like to see:

- All infrastructure services are for all domains – not just either HPC or HTC (e.g. common accounting, identity infrastructure).
- Promote use of de facto tools with add ons rather than duplicating functionality in infrastructure specific tools.
- Exploiting the differences of the infrastructures through VO/VC targeted scientific portals enabling HTC/HPC for a series of application for different VO/VCs.