A Simple Virtual Organisation Model and Practical Implementation

Lyle J. Winton

School of Physics
The University of Melbourne,
Victoria 3010, Australia.
Email: winton@physics.unimelb.edu.au

Abstract

The development of Grid middleware, such as the Globus Toolkit version 2, reached a level of maturity and stability in which it was possible to create widely distributed resource Grids. Within the last few years various experiences have arisen from the construction of such Grids and so called "testbeds". The purpose of this paper is to highlight some of the problems, propose some simple solutions, and to report on the development of prototype implementations. The focus of this paper is on solutions that can be coordinated using an information system representing Virtual Organisations.

Keywords: Computing, Grid, Virtual Organisations, Globus

1 Virtual Organisations

It has been proposed in the defining work by Foster, Kesselman, and Tuecke that "the real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisations". They go on to state that the conditions or rules of this resource sharing must be clearly negotiated and defined between the resource providers and consumers. Further, that the virtual organisation (VO) is defined as the set of individuals and institutions for which these conditions or rules apply. In our experience this is an accurate definition. It could be argued that in some cases, such as international research collaborations, the VO can be well defined apart from the sharing of resources. However, there are often individuals not directly associated with the collaboration that require access to the collaborative resources. Such individuals can include technical and computing support staff associated with home institutions of the collaboration members or staff of the resource providers.

A point to keep in mind is that the virtual organisation is only one half of the picture. A typical VO will have access to many facilities which are not owned and managed by the VO. These facilities are also organisations with security and policy concerns which may span multiple user communities or VOs. A Grid is a complex network of these organisations, VOs and resource facilities. Any VO model and associated software, therefore, must provide the ability for both VO and facility to respect and manage security, policies, and priorities.

The following requirements for VOs have been identified from existing projects (EU DataGrid WP6 2002, Foster, Kesselman, & Tuecke 2001).

- Users may be member of any number of VOs.
- A resource can participate in one or more VOs.
- Users may have any number of roles within a given VO.
- VOs must be able to specify membership policy and user authorisation.
- A users VO membership must remain confidential.
- A resource owner be able to allow authorisation by VO and VO role membership.
- It must be possible to assign job priorities within resources.
- It should be possible to list resources and actions to which a VO member or role has access.
- It should be possible to list resources to which a VO member or role has access to carry out specific actions.
- It should be possible to determine if a VO member and role has access to a certain resource and authorisation to carry out specific actions.
- Authorisation decisions must be consistent within a VO.
- It must be possible to disable a users VO authorisation.
- The VO must be able to specify security requirements on any resource for specific roles.
- A user must be able to select and deselect VOs and roles.

An information system representing a VO does not necessarily have to meet all these VO requirements. Some requirements can be met by providing additional systems to which VO members have access. One such system is the Globus monitoring and discovery system or MDS (Czajkowski et al. 2001).

Much work has gone into the development of Grid middleware towards providing access to resources (Foster & Kesselman 1997, Foster & Kesselman 1998, Ghiselli et al. 2002, P.Eerola et al. 2003, GriPhyN

We would like to acknowledge that efforts reported in this paper would not be possible without funding from the Victorian Partnership for Advanced Computing (VPAC) and the Australian Research Council (ARC). We would also like to thank VPAC, the Australian Partnership for Advanced Computing (APAC), the Australian Centre for Advanced Computing and Communications (AC3), the Australian National University Storage Facility (ANUSF), and the Melbourne Advanced Research Computing centre (ARC) for the use of their facilities.

2003). Within these efforts the term virtual organisations is frequently used only to represent a group of individuals or resources (Gullapalli 2001, GT3 web 2002, Iamnitchi 2000, GroupMan web 2003, GridPP web 2004). In a few cases, VO information systems have been used to represent these collections of individuals (Ghiselli et al. 2002, P.Eerola et al. 2003) allowing the development of tools to help coordinate resource sharing. Such developments range from tools simplifying the configuration of resources (EU DataGrid web 2004, NorduGrid web 2004, GridPP web 2004) to authentication systems (Alfieri et al. 2003, Foster et al. 2003). However, task priorities are still often managed entirely using local resource configuration. In a typical situation various roles are mapped to local resource projects or to accounts with varying quality of service requirements and priorities. System policies enforced at some facilities can inflate the administrative overheads of managing groups of users and cause configuration difficulties. The simple virtual organisation model proposed herein attempts to address these deficiencies and enable the development of tools for rapid Grid deployment and configuration.

1.1 Existing Implementations

1.1.1 VOMS

The Virtual Organization Membership Service (VOMS) is an authentication system built on top of existing Globus GSI security (Alfieri et al. 2003). It has three components: a server containing user group and role information; a client for the generation of user credentials (Grid proxies) that contain additional role information; and an administration interface to the server. The system relies on extensions inserted into the users proxy certificates. These extensions are non-critical and allow the continued use of Grid resources that do not support VOMS.

The VOMS server is effectively an information system representing the VO as a complex hierarchy of users, groups, and subgroups. Users can also be characterised by roles under which they can operate at the VO group level.

The VOMS project view authorisation information as divided into two categories: information regarding the users relationship to the VO, including groups and roles; and information regarding what VO users, groups, and roles are allowed to do at provided resources. They propose that the first category of information is best managed by the VO and the second category is best kept at the local resource.

Two deficiencies exist within the VOMS system. The first is that no certificate authority (CA) information is associated with this VO model. All resource managers must negotiate and install CA certificates that are required by participating VOs outside of the VOMS system. However, the EU DataGrid community feel that managing trusted groups of CAs is a much simpler problem than that of managing VOs and users (Alfieri et al. 2003). The primary mechanism for the prevention of invalid or compromises authorisation, certificate revocation lists, must also be managed outside of the VOMS system.

The second deficiency of the VOMS system lies in the separation of authorisation information into two categories. While it is clear that users relationships within the VO should be managed by the VO, it is not clear that information regarding member and group usage of resources is better managed at the local resources. Resource providers will wish to control negotiated security policies and priorities at the local facility. However, VO will undoubtedly wish to manage internal user, group, and role priorities independently of the resource providers. VO priorities may need to change rapidly, for example with periodic deadlines. This may be difficult if changes require renegotiation with many resource providers.

1.1.2 CAS

The Community Authorization Service (CAS) consists of a server containing VO information regarding CAs, users, groups, other servers, and resources (Foster et al. 2003). Policy statements within the CAS server determine what actions users and groups can perform on resources. The server works by users requesting authentication for a particular capability with their normal credentials. The server then returns a signed policy assertion which is then embedded into a new user credential (Grid proxy).

The CAS project argues that the solution to many problems of VO and resource policy enforcement can be solved by resource owners allocating blocks of resources to communities (or VOs) and allowing the community to manage fine-grained access control within the allocation (Pearlman et al. 2002). The user effectively inherits the combination of rights granted to the community by the resource provider and rights assigned to the user by the community. The community can then implement policies to distribute access to resource blocks amongst members and groups. The distribution can then be tuned to manage task priorities for the members and groups (or roles). In this respect, the CAS service goes a step beyond VOMS by allowing both VOs and resource providers the ability to manage access policies and priorities. However, the granularity of VO resource management remains at the level of allocating dedicated or shared resource blocks.

One complication with the CAS system is that resources must implement a policy evaluation API to extract the CAS assertion carried with user credentials. As the CAS server has full access to VO resources, this is required to restrict the user's access to resources. The carried community policies must be supported by the local resources. There are reported difficulties with the integration of this system and existing Grid technologies (Alfieri et al. 2003).

The initial CAS prototype worked by the server returning a delegated credential for the requested capability. The delegated credential was that of the CAS server, so the CAS server trusted the user credentials and resource providers trusted the CAS server credentials. One of the benefits of this initial prototype was that it eliminated the need to support every users CA certificate at each resource. Resource providers needed only trust the CAS servers CA. This, however, increased the potential damage that can be done in the event that a CAS certificate is compromised, and led to difficulties in determining actual user credentials. Subsequent versions of CAS, like the VOMS system, now require resources managers to manually install CA certificates associated with participating VOs.

2 Experiences

2.1 Grid 2003 HPC Challenge

At the joint Super Computing 2003 and Grid 2003 conferences the University of Melbourne GridBus Lab, in collaboration with other departments and other institutions, attempted to construct the largest testbed participating in the 2003 HPC (high performance computing) challenge (Buyya 2003). The testbed grew from several resources situated at universities and facilities within Australia, to reach 218

resources in 50 locations across 21 countries.

We have experienced numerous difficulties with constructing Grids. Some of these arise from facility architecture, security infrastructure, and social difficulties associated with different organisations sharing or providing access to facilities. Some of the difficulties are listed here together with solutions that were implemented or might be implemented in the future.

Difficulty	Solutions. Possibilities?
Middleware installation too time consuming for administrators.	Repackage with simple installer. Package managers?
Network access control list and firewall requirements.	Education and advanced warning to network administrators.
Installation validation.	Testing scripts.
User application installation.	Done by hand. Package managers?
Configuration of user and CA information for growing testbeds.	Virtual organisation information systems?

Provided a common middleware solution can be implemented, only the last two problems are of ongoing concern for a growing testbed. Updating user and certification configuration during the construction of the HPC Challenge testbed was a time consuming and problematic task. Manual configuration inevitably led to errors which in some circumstances rendered resources temporarily unusable. The automation of this process was recognised as a desirable feature.

2.2 Belle Experiment Production Grid

The Belle Experiment was constructed to investigate one of the fundamental violations of symmetry in nature, charge-parity violation. This asymmetry may in part help explain the matter – antimatter imbalance observed within the universe. The experiment is situated at the KEK B-factory in Tsukuba, Japan, and is the work of a collaboration of 400 physicists from 50 institutions around the world. The University of Melbourne proposed the use of Grid solutions to help facilitate collaboration and the use of distributed data and CPU resources.

In 2003 it became apparent that the ever increasing rate of data taken from the experiment would quickly result in a CPU shortage. The largest drain on these resources is the generation of simulated (Monte Carlo) data necessary for determining efficiencies, experimental, and systematic uncertainties. Typically three times more simulated data than experimental data is required. The Australian members of the collaboration agreed to take part in the production of 4×10^9 Monte Carlo events required for data analysis in 2004. This production is currently run over a total allocation of approximately 200 Pentium 4, 2GHz equivalent, from several Australian computing facilities (VPAC, APAC, AC3, ANUSF, and Melbourne University ARC). More than half of these facilities are accessible through Grid middleware. While much of the production has been performed using traditional methods of access, much has been learnt from the varying security policies of each facility. The quickest method of allowing access to facility resources for groups of people would be to provide a shared account. The ability to map groups of people to a single account is available in existing Grid middleware such as Globus GSI (Butler et al. 2000). However, for security and accounting purposes many facilities forbid the use of shared or role based accounts.

One of the more apparent problems is that each facility has it's own non-trivial account application procedure. If a new researcher were to require access to these resources they would need to follow the procedure for each and every facility. While most facilities have simplified the process by allocating resources based on project and then associating accounts to the project, the process for applying for a new account is generally manual and requires intervention from several people (project leaders and administrators). For a Grid the size of the current collection of resources this is a small issue, but clearly this would become a major problem for a larger Grid. A possible solution might be for the VO to provide an account application process where the superset of all required information and approvals for all sites are collected at the one time. This is yet to be investigated.

3 A Simple Model

An initial starting point for a simple VO model is the collection of all information necessary for authorisation with any resource. Authorisation of the entire VO, specific groups or roles, and individual users may be required.

- User identification (certificate identifier)
- Service identification (certificate identifier)
- Groups and Roles within the organisation (user or service collections)
- Trusted user certifying bodies (certificate authority information)
- Trusted resource certifying bodies (certificate authority information)
- Untrusted certifying bodies
- Untrusted identities (users and certificates)

One possible certification structure is that a VO is certified by a single certificate authority (CA) and certificate domain. (In the case of the widely used GSI infrastructure (Butler et al. 2000) the certificate domain can be defined as the organisational component of the certificate subject.) Members of a VO can then be identified by the CA and certificate domain. A problem arises when a single user becomes a member of multiple VOs as they must manage multiple certificates. Another certification structure is where one certificate is used to identify a person, and the person may belong to one or many domains. The issue of adding a person to a VO then becomes more complex as the VO may have to then trust additional CAs and configure all resources accordingly. A popular certification structure is to have a world wide network of trusted CAs for e-Research, usually one for each country (LCG web 2004). The creation of a new CA from a joining country would then require the world wide deployment of configuration for that CA. The proposed model allows the VO to specify to resource facilities their trusted CA list, those used to certify VO members.

As an addition to the model a simple indication of VO priorities can be included.

- User priorities
- Groups and Roles priorities
- Default VO priorities

This allows for the assignment of priorities globally, to individuals, and to groups of individuals. Overlapping groups must also be taken into account as individuals can have varying priorities associated with multiple roles within the virtual organisation. Role-based priorities can be specified by associating a priority to role objects within the VO structure, and

allowing the user to submit an optional role identifier (object distinguishing name) with each job. When determining a job's priority the role object can be queried to determine if the user has the right to run jobs under that role.

An information system based on this model was implemented using an LDAP database. This system was designed to extend implementations of several similar efforts (Ghiselli et al. 2002, P.Eerola et al. 2003, GridPP web 2004) to ensure compatibility with tools that may be used in existing Grid deployments. Four LDAP object classes were used in this implementation, all in common use. All objects were grouped using the *organizationUnit* class which is not essential to this model. Individual users and services, entities requiring authorised access, were represented using the *organizationalPerson* class found in the LDAPv3 schema (RFC2256 1997). Groups or roles were represented using the *groupOfNames* object class also found in the LDAPv3 schema, the *member* attributes referring to individuals (*organizationalPerson* objects) within the VO structure. CA information necessary for configuration was implemented using the *document* object class found in the Cosine and Internet X.500 schema (RFC1274 1991). In the simplest case this allows resource administrators to identify which configuration files are needed for each CA.

Logical groupings of users, services, and CAs within a VO can be facilitated using the inherent tree structure of LDAP. The VO structure can be represented as a hierarchy of *organizationUnit*s. For example, a VO manager may wish to group members by institution for easy reference. In this case each institution can be represented by an *organizationUnit* under which member *organizationalPerson*s are kept. These *organizationUnit* objects can also be treated as groups or roles for authorisation and defining priorities in addition to *groupOfNames* objects. A hierarchy of *organizationUnit* objects can be implemented to represent complex group and subgroup structures found in some organisations.

The *description* attribute of *organizationUnit*, *organizationalPerson*, and *groupOfNames* objects is used to store additional information not in the LDAPv3 schema. A user or service certificate identifier (Grid certificate subject) is stored in the *description* attribute preceded by the string "subject=". The user, service, and role priorities are stored in an additional *description* attribute as an integer preceded by the string "priority=". A default VO priority, inherited by users, services, and roles without specific priority, can be defined using the *description* attribute of the VO's parent *organizationUnit* object. Two attributes are used in CA configuration file *document* objects, the *documentIdentifier* representing the preferred name of the file, and the *documentLocation* specifying the URL from where the document can be obtained. CA *document*s with a *description* attribute value of "delete" can allow the VO to specify CAs that are not longer required or trusted.

As an alternative VO members can be represented by *inetOrgPerson* objects (RFC2798 2000) and their Grid certificate stored in the attribute *userCertificate*. The users Grid certificate subject is extracted directly from the certificate.

The management of such a VO information system can occur using existing tools for the addition, modification, and removal of LDAP database records. There are several tools available for the management of LDAP databases (Winton 2004, Gawor 2001, Miao 2004).

This proposed model of the VO information system, unlike other models, does not focus on the discovery of resources accessible to the VO. Systems for the publishing, collation, and discovery of resource information have existed for some time. For example, the Globus 2 toolkit includes the Grid Information Index Service (GIIS), an LDAP based service for the discovery of resources (Czajkowski et al. 2001). Such systems have proved sufficient for a number of applications (Venugopal et al. 2004, P.Eerola et al. 2003) and there are several efforts to extend such systems (P.Eerola et al. 2003, BDII web 2004, Yu et al. 2004).

This structure allows for the assignment of priorities globally, to individuals, and potentially overlapping groups of individuals. However, some Grid middleware implementations such as Globus do not allow for overlapping groups where individuals can have varying priorities associated with multiple roles within the virtual organisation. Role-based priorities can be taken into account by associating a priority to *groupOfNames* and *organizationUnit* objects with the VO structure, and allowing the user to submit an optional role identifier (distinguished name) with each job. When determining a job's priority the *groupOfNames* or *organizationUnit* can be queried to determine if the user has the right to run jobs under that role.

3.1 Resource Configuration Manager

To realise the usefulness of a VO information system, facilities must be able to interpret the VO information and configure resource accordingly. A resource configuration manager can be developed to help automate this task. To be useful, a configuration manager should allow resource administrators to implement a wide range of commonly encountered resource usage policies. It must also allow the administrator to make final decisions in situations that could potentially compromise the resource.

3.1.1 Implementation

The GridMgr (Grid Manager) tool was developed as an implementation of a resource configuration manager for the Globus toolkit (Winton 2004). It was specifically developed to meet the resource policies of facilities encountered during the deployment of the HPC Challenge and the Belle production Grid. The concept and original source for this tool was taken from the NorduGrid's "nordugridmap" script (NorduGrid web 2004), a modified version of the European DataGrid's "mkgridmap" script (EU DataGrid web 2004). This tool is currently in use at several sites within Australia.

A number of resource usage policies are available in the GridMgr implementation.

- Mapping users and groups within a VO to shared accounts.
- Manually mapping individuals within a VO to individual accounts.
- VO users and groups can be mapped to a range of accounts (eg. "grid001" through "grid200"), providing limited security separation for users. As subsequent access by the same user may require common files and data, this mapping is performed using a repeatable hashing algorithm on the user's Grid subject.
- Restriction of VO mappings to specific local user groups, preventing access to jobs and data between VOs on the same resource.
- Mapping users to individual accounts by matching their certificate name to the local account

full name usually found within password file comments. As on many systems an account full name can be modified by the user, name changes are monitored to notify the administrator of possible exploitation. Any name change that leads to an account map will cause a notification and must be explicitly approved by the administrator.

- Denial of access to entire VOs, VO groups, individual users, and to Grid subjects matching a given expression.

Grid certificate subjects are periodically extracted from one or more registered VOs, mapped to local accounts via the above policies as configured, and written to the Globus *grid-mapfile* allowing authorised access. A static *local-grid-mapfile* is incorporated into this file to allow administrators to manually add Grid subject mappings. For resources wholly owned by the VO, accounts can be generated as required. A number of security requirements can be enforced before a subject is placed in the *grid-mapfile* :

- valid full name matching
- full name matches only one account
- no shared accounts (optional)
- no new or changed system account full name (optional)
- valid account group (optional)
- non-root account (optional)
- non-existent account
- not denied by subject pattern match
- not denied by VO, group, or individual.

Failed security requirements are reported in the *grid-mapfile* as comments and some requirements cause email notification. For example, a new or changed account full name matching a VO individual will cause email notification and can require administrator approval.

CA certificates and configuration files are also periodically downloaded from locations specified within the VO information system. Existing files are first tested then overwritten in the event of a change. New or modified CA certificates are tested for validity before installation, as an invalid CA certificate can disable a Globus installation. New or modified CA certificates can cause email notification and can be temporarily disabled pending administrator approval. Specific CA certificates can be permanently disabled. Files marked for deletion are removed if not required by any VO register on the resource. As a number of CAs store their configuration file in archive formats, the download and extraction from "gzip" compressed "tar" archives and "zipped" archives is supported.

CA certificate revocation lists (CRL) can be maintained in two ways. The first is by registering CRLs as a configuration file within the VO information system. The second follows the method used by the European DataGrid (Ghiselli et al. 2002) of installing files containing the download location of the CRLs. These files have the same name as the CA certificate but with the extension **.crl_url*. Periodically an attempt is made to download CRL files from the locations specified in any installed **.crl_url* files. To ensure a high level of security all CRL files must be updated frequently. The GridMgr tool allows the download of Grid subjects, CA files, and CRL files as separate operations giving the resource administrator the freedom to choose their frequency.

An additional feature was added to alert resource administrators when the host certificates are about to expire. While this does not strictly deal with VO configuration it does help ensure continued authentication and access for VO members.

Some facilities have strict security policies that forbid the use of shared accounts, that is each individual must have a separate account. While the developed GridMgr tool has allowed for such a situation with the ability to map users to local accounts, a potential security problem should be noted. If any resource providing access to a VO uses shared accounts, it is possible for a malicious user to obtain another individual's credentials (short term proxy) using that resource. Once the credential has been obtained they may then access the individual's account on any resource within the VO, including those with separate accounts. In summary, the "no shared accounts" security policy only makes sense if all resources accessible by a VO follow this policy. For a VO to enforce this policy they need only allow access to their VO information to facilities that adhere to this policy. However, a resource attempting to enforce this policy relies entirely on the VO also enforcing the policy for all of it's accessible facilities.

3.2 VO Managed Job Queueing Services

In traditional cluster computing, jobs are placed in a resource queue and the local queue manager determines priorities and executes jobs when resources become available. Effectively jobs are *pulled* from the queue to free resources. It has been one of the achievements of Globus and other Grid projects to provide common access to these resource queues.

From the Grid perspective one problem with this mode of operation is that local resource priorities cannot be determined until jobs are placed in the resource's queue. Even then priorities are often hard to determine. For a heavily utilised resources queue times can be quite long. When dealing with a Grid of resources, a job sitting in a local queue has the potential to be executed elsewhere. So an intelligent Grid job manager might submit a job to an appropriate resource queue only to remove it later if more appropriate resources become available. Alternatively the job might be submitted to multiple resource queues and removed once execution starts at one of the resources.

Viewing this problem from a different angle, the traditional cluster computing mechanism of *pulling* jobs from a queue is broken. Most Grid middleware, such as Globus, relies on the *pushing* of jobs to local resource queues where they are eventually *pulled* from the queue to free resources. Again, the problem becomes apparent when dealing with heavily utilised resource queues. The resource queue may never appear free so determining when to push jobs becomes a complex issue of estimating queue times.

An alternative mechanism is to allow job consumers to *pull* jobs when resources become available or queue times become short. This can eliminate the problem of determining each resources priority for jobs and estimating queue times. An independent job queueing service can act as a location from which jobs can be consumed or pulled. The priority of jobs could be managed independently of resource priorities by allowing certain jobs to be consumed first. In practise, however, this may not be desirable as facilities must allocate resource fractions or guarantee quality of service to multiple virtual organisations. Hence, the priority of jobs must be determined between the queueing service and the resource consumer. In the Grid context such a system will allow for the management of priorities for both VOs and resource facilities.

Additionally, a VO can internally manage the allocation of resource fractions with a granularity finer than whole resource blocks.

3.2.1 Prototype

A prototype VO managed job queueing service was developed as a web service with simple authentication based on proxy subject. The primary goal of the prototype was as a proof of concept. The following functionality was incorporated:

- submission of jobs with user defined priority and optional role;
- determination of VO defined priority for user or role;
- determination of overall priority for job;
- deletion of jobs by submitting user;
- listing of all jobs for submitting user;
- pull the job of highest priority;
- pull the job of next highest priority;
- reservation of jobs (exclusive resource access);
- release of jobs (allow other resources access);
- flagging of jobs as failed or completed;
- retrieval of job state;
- (un)registration of VO information servers.

In a typical scenario, users submit jobs to the queue then the jobs are associated overall priorities based on a combination of user, VO, and role priorities. Resource job consumers extract the jobs of highest priority from multiple queueing services, determine the local priority for the jobs, then reserve the jobs of highest priorities executing them on the local resource. Once execution completes the jobs are then flagged as either failed or completed. The prototype was used to effectively simulate a scenario of several VOs with multiple users submitting multiple jobs of varying user and VO priority.

In order to test the queueing service prototype a resource level job consumer was required. A simple consumer was built including all functionality required for the simulation of a typical cluster resource. The following functionality was included:

- extraction of jobs from multiple VO queues;
- extraction of VO job priorities;
- conversion of VO priorities to the resource's local priority for each job;
- reservation of the jobs of highest priority;
- allocation of reserved jobs to simulated free CPU;
- simulated job completion by freeing CPU (after some random period);
- and the marking of jobs as complete in the VO queue.

The conversion of VO priorities to the resource's local priority was performed using the following algorithm. For each VO queue a minimum and maximum local priority is defined, P_{min} and P_{max}. The local priority P_{local} is then calculated using the VO's job priority P_{job}.

$$P_{local} = P_{min} + (P_{max} - P_{min}) \left(1 - \frac{1}{P_{job}/50+1}\right)$$

VO job priorities below 0 are translated to 0. This calculation then maps VO priorities below 50 to the lower half of the local priority range, VO priorities below 100 to the lower two thirds of the range, and VO priorities below 200 to the lower 80% of the range. In this way the VO priority, regardless of how large, can be attenuated to within the local priority range while preserving comparative job priorities for the same VO.

A single VO may operate multiple job queues in parallel, with users submitting to any queue. The relative priority of jobs across all queues is preserved as priorities are extracted from a common VO information service. Provided resources administrators attribute the same local priority range for all queues within a VO, jobs from any queue will have comparative priorities. A VO may scale the number of queues to accommodate increasing job quantities and to provide fewer points of failure.

3.2.2 Beyond Prototype

To more evenly distribute VO priorities over the local resource priority range (specific for each VO), an active range for VO priorities could be specified. VO priorities within this active range could then be scaled to fit within the central 80% of the resource's local priority range. VO priorities below and above this range could be attenuated within the lowest and upper most 10% of the local priority range, preserving comparative priorities within a VO. Negative VO priorities could also be handled with such an algorithm. However, there may be some advantages to having a fixed VO priority scaling algorithm for all resources, as in the prototype. The consequences of a VO reorganising their priorities can be more easily predicted.

A fairshare algorithm for priority scaling could be a more sensible choice to help facilities better manage resource allocation schemes. In such an algorithm VO priorities could be dynamically scaled and attenuated in an attempt to establish and maintain a target usage fraction. For instance, a facility may wish to allocate a target of 20% of their resource to a specific VO. Initially, or if few jobs have been submitted, the VO's usage would be below target so the local job priorities are scaled higher in comparison with other jobs. A VO may eventually reach a usage fraction greater than their allocated target, as facilities may wish to allow access to idle resources regardless of allocation fractions. If the VO's usage goes above target the local job priorities are scaled lower in comparison with jobs from other VOs.

A more serious problem with the prototype is that fixed job priorities can lead to job lock out. Higher priority jobs can be continually submitted and will always be executed in preference to lower priority jobs. This could be remedied by gradually increasing job priority with queue time. Another solution would be to implement a fairshare like algorithm on the queue itself. In such an algorithm the user's priority within the VO would represent a target queue usage. Queue usage could be calculated as the number of jobs recently executed through the queue within a specific time period.

The resource job consumer can be trivially extended beyond use for simulation towards managing jobs on real resources. The Globus toolkit is a likely choice for a generic interface providing resource usage information through MDS and a job submission and monitoring service through GRAM. The difficult task, however, is the integration of resource management and scheduling which must lie between the VO queues and the resources. While it is entirely appropriate for a free resource to request jobs, the requesting resource may not be the best choice for the

jobs themselves. At worst, the requesting resource may be entirely inappropriate for specific jobs due to insufficient hardware resources, mismatched software environments, repeated job failure, expensive or slow network connectivity to data centres. In part, this can be solved by the requesting resource inspecting the job description, however this does not allow for more complex user or VO decisions. For instance, a user or VO may have a preferred set of resources for their jobs due to network topology, but would be prepared to run elsewhere if resources are scarce. It would be problematic for this information to be held and decisions to occur at the resource level. Another instance may be where a user or VO has tools to identify non-trivial job failure, such as problems caused by variations in operating systems or numerical libraries. In this case the user or VO may make a decision after monitoring the output of several jobs to prevent further submission to the resource.

For more complex job requirements the VO queue could manage job descriptions of a higher level than resource specification languages. The queue could be used to manage workflow descriptions or directives such as those in Nimrod/G (Abramson et al. 2002). This leads to the further problem of prioritising jobs based on description. Complex jobs can have varying resource usage which may be difficult to predict. In fact, the expertise in predicting resource usage for each application usually lies within the virtual organisation. This further highlights the need for complex user or VO decisions.

In the prototype design of the European Data Grid middleware (Ghiselli et al. 2002) the need for a workload management service was highlighted. This service includes a resource manager for the matching of job descriptions with resources. In their design the management of user and VO priorities are contained wholly within this service. However, they argue the need for this to accommodate a "distributed organisation" of community based schedulers which are in some way coordinated and which also allows for "allocation fairness". A mechanism for this coordination would be for the work load manager services to extract jobs from VO managed queues. The scheduling, in this situation, can then be coordinated using job state management and guided by the priorities originally specified within the VO information service.

3.3 Simulations

The VO managed job queueing service prototype was tested in a simulation together with the previously mentioned resource level job consumer. The simulation included 3 VOs each with 10 users of varying VO priority. The VOs maintained one queueing service each. All VOs have access to 10 simulated resources, each consisting of a clusters of 10 to 50 nodes or job execution slots. Each resource allocated a different local priority range for each VO queue. The ranges were of at least 10 and randomly allocated between 1 and 100. Each user periodically submitted between 10 and 50 jobs at once but only when they had no jobs in the queue. Each job took between 1 and 10 minutes to complete on any node.

This particular scenario led to an average stable state of 30 jobs for each of 30 users. These were shared across 10 hosts with an average of 30 nodes. The average job load for this situation is 3 times more than the number of available nodes, an over utilised resource scenario. Resource usage reached saturation within 4 minutes of the simulation commencing. The state of the queue was frozen after 15 minutes for evaluation.

In order to evaluate the effect of priority assignment of jobs within a VO, the time in which a job was unallocated (queue time) was compared with the job priority (Fig. 1). In order to reduce the contribution of queue load, the queue time was divided by the queue size at time of submission. While there is a possible trend towards jobs with higher priorities obtaining shorter queue times, it is not obvious. The comparison, however, is made difficult due to differing host priorities for the same VO. If the comparison is made for a specific resource the trend is clearer (Fig. 2).

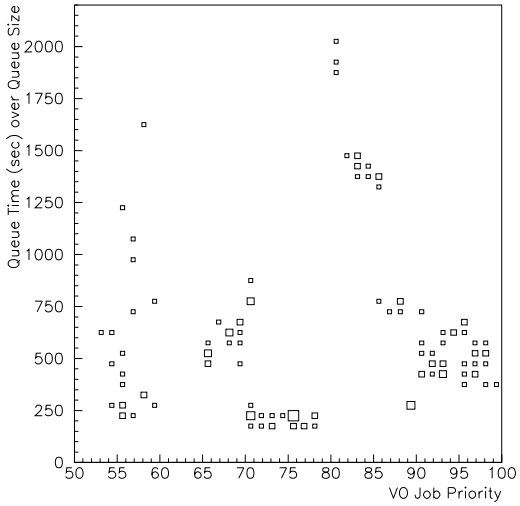

Figure 1: Queue time divided by queue size compared with a specific VO's priorities.

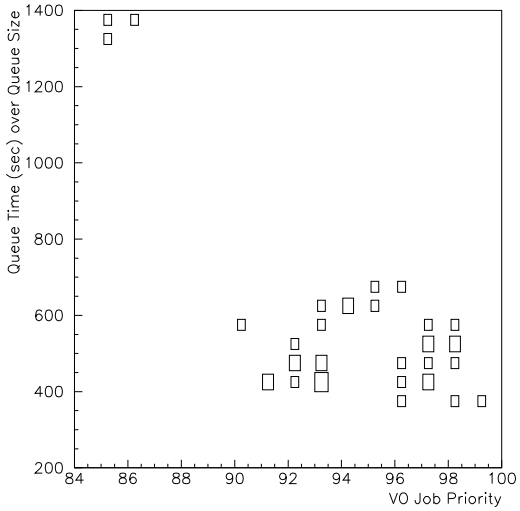

Figure 2: Queue time divided by queue size compared with a specific VO's priorities on a specific resource.

Another possible complication is the previously mentioned prototype problem of "job lock out". Looking closely at figure 1 there are no completed jobs with VO priority lower than 50. The lock out problem becomes obvious if we compare the incomplete job count with VO priority (Fig. 3).

To evaluate the effects of local resource priority range, the queue time divided by queue size was compared with the average local priority (Fig. 4). The trend towards jobs of higher mean priority obtaining shorter queue times is clear. It is observed that only two VOs obtained access to the resource. The third VO, with a local priority range lower than the two observed in the figure (mean priority of 27.5) was effectively locked out.

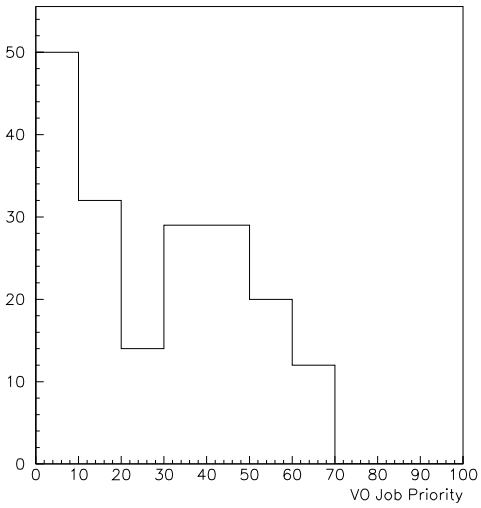

Figure 3: Incomplete job count at 10 minutes by VO priorities.

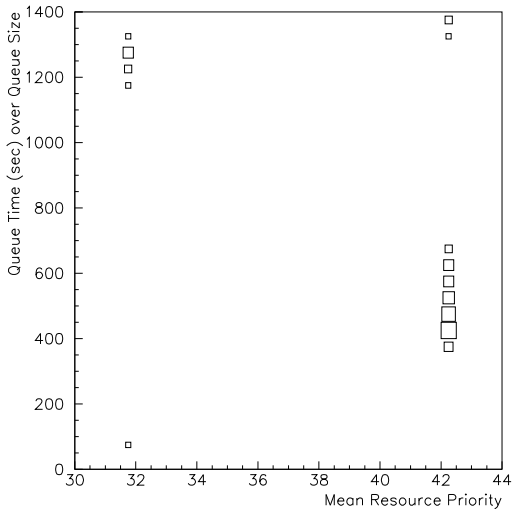

Figure 4: Queue time divided by queue size compared with a specific resource's mean VO priority.

An underutilised resource situation could be simulated by increasing the number of the resource nodes or decreasing the number of jobs. This is not a useful simulation, however, as all jobs are virtually guaranteed resource allocation immediately.

4 Summary

The VO information system model proposed has proved sufficient for the development of a tool aiding in the deployment and configuration of Globus resources. The developed tool, GridMgr, allows for a number of facility security policies in an effort to further facilitate the use of resources owned and managed by organisations external to the VO. A prototype VO managed queueing system was also developed to help coordinate the use of resources according to user and role based priorities specified within the VO information system. The usefulness of the prototype is supported by simulation. A number of problems for future investigation arose from experiences with the prototype. To address issues of job lock out and resource allocation fractions, dynamic priority assignment should be considered including fairshare like algorithms. The need to incorporate complex VO and user requirements is also an area for future investigation.

References

Foster, I., Kesselman, C. & Tuecke, S. (2001), 'The Anatomy of the Grid: Enabling Scalable Virtual Organizations', *in* Intl J. Supercomputer Applications, (2001.

Foster, I., Kesselman, C., Nick, J. & Tuecke, S. (2002), 'The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration', Global Grid Forum, Jun 2002.

Buyya, R. (2003), Global Data-Intensive Grid Collaboration', regarding SC2003 HPC Challenge. http://gridbus.cs.mu.oz.au/sc2003/

P.Eerola et al. (2003), 'The NorduGrid architecture and tools'. Proceedings of 2003 Conference for Computing in High Energy and Nuclear Physics, CHEP03, La Jolla, Mar 2003.

Foster, I. & Kesselman, C. (1997), 'Globus: A Metacomputing Infrastructure Toolkit'. Intl J. Supercomputer Applications, 11(2), pp. 115-128, 1997.

Foster, I. & Kesselman, C. (1998), 'The Globus Project: A Status Report'. Proceedings of IPPS/SPDP 98 Heterogeneous Computing Workshop, pp. 4-18, 1998.

Ghiselli et al. (2002), 'DataGrid Prototype 1', TERENA Networking Conference, Jun 2002.

GriPhyN (2003), 'GriPhyN Annual Report - 2002 through 2003', GriPhyN2003-21, Jul 2003.

Alfieri et al. (2003), 'Managing Dynamic User Communities in a Grid of Autonomous Resources', Presented at 2003 Conference for Computing in High Energy and Nuclear Physics, CHEP03, La Jolla, Mar 2003.

EU DataGrid WP6 (2002), 'VOMS vs EDG Security Requirements', EU DataGrid, Work Package 6, 2002.

Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C. (2001), 'Grid Information Services for Distributed Resource Sharing'. Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, Aug 2001.

Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J., Welch. V. (2000), 'A National-Scale Authentication Infrastructure'. IEEE Computer, 33(12), pp. 60-66, 2000.

Abramson, D., Buyya, R., & Giddy, J. (2002), 'A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker', Future Generation Computer Systems (FGCS) Journal, Volume 18, Issue 8, pp. 1061-1074, Elsevier Science, The Netherlands, Oct 2002.

NorduGrid web page (2004), 'Description of the NorduGrid Virtual Organization'. http://www.nordugrid.org/NorduGridVO/vo-description.html

EU DataGrid web page (2004), 'mkgridmap Authorisation Utility'. http://cvs.infn.it/cgi-bin/cvsweb.cgi/Auth/edg-mkgridmap/

Gullapalli, S. (2001), 'The GraDS MacroGrid', Presented at 2001 Globus Retreat, San Francisco, Aug 2001. http://www.globus.org/about/events/retreat01/presentations/retreat01gullapalliTalk.ppt

GT3 web page (2002), 'Grid Services in Action: A Prototype of the GT3 Core', Demonstrated at the 4th Global Grid Forum, Toronto, Feb 2002. http://www.globus.org/ogsa/releases/ TechPreview/GT3CoreIntro.pdf

Iamnitchi, A. (2000), 'Resource Discovery in Virtual Organizations', Presented at 2000 Globus Retreat, Pittsburgh, Jul 2000. http://www.globus.org/retreat00/presentations/adriana_info/

GroupMan web page (2003), 'The Caltech Virtual Organization Group Manager', http://groupman.sourceforge.net/

GridPP web page (2004), 'GridPP Virtual Organisation Authorisation Server', http://www.gridpp.ac.uk/vo/

Yu, J., Venugopal, S., & Buyya, R. (2004), 'A Market-Oriented Grid Directory Service for Publication and Discovery of Grid Service Providers and their Services', Journal of Supercomputing, Kluwer Academic Publishers, USA, Feb 2004.

Alfieri et al. (2003), 'VOMS: an Authorization System for Virtual Organizations', Presented at the 1st European Across Grids Conference, Santiago de Compostela, Feb 2003. http://grid-auth.infn.it/

Foster, I., Kesselman, C., Pearlman, L., Tuecke, S., Welch, V. (2003), 'The Community Authorization Service: Status and future', Presented at 2003 Conference for Computing in High Energy and Nuclear Physics, CHEP03, La Jolla, Mar 2003.

Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S. (2002), 'A Community Authorization Service for Group Collaboration', *Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002.

LCG web page (2004), 'LHC Computing Grid Digital Certificate Authorities'. http://lcg-registrar.cern.ch/pki_certificates.html

RFC2798, 'Definition of the inetOrgPerson LDAP Object Class', Network Working Group, IETF, Apr 2000.

RFC2256, Wahl, M., 'A Summary of the X.500(96) User Schema for use with LDAPv3'. Network Working Group, IETF, Dec 1997.

RFC1274, Barker, P., & Kille, S., 'The COSINE and Internet X.500 Schema'. Network Working Group, IETF, Nov 1991.

Venugopal, S., Buyya, R., & Winton, L. (2004), 'A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids', 5th International Middleware Conference, Oct 2004.

BDII web page (2004), 'Berkeley Database Information Index (BDII)'. http://grid.desy.de/testbed/EDG/BDII.html

Winton, L. (2004), 'The Grid Manager utility (GridMgr)'. http://epp.ph.unimelb.edu.au/twiki/bin/view/EPP/GridToolsGridMgr

Gawor, J. (2001), 'LDAP Browser/Editor'. http://www.iit.edu/ gawojar/ldap/

Miao, T. (2004), 'LDAP Explorer web interface', http://igloo.its.unimelb.edu.au/LDAPExplorer/