European Research Network on Foundations, Software Infrastructures and Applications
for large scale distributed, GRID and Peer-to-Peer Technologies

A Network of Excellence funded by the European Commission

Authorizing Grid Resource Access and Consumption

Erik Elmroth[1],
elmroth@cs.umu.se,
Michał Jankowski[2],
Norbert Meyer[2]
{jankowsk, meyer}@man.poznan.pl,

[1] *Department of Computing Science and HPC2N*
Umeå University
901 87 Umeå, Sweden
[2] *Poznań Supercomputing and Networking Center*
ul. Noskowskiego 10
61-704 Poznań, Poland

CoreGRID Technical Report
Number TR-0161
July 4, 2008

Institute on Grid Information, Resource and Workflow Monitoring Services

CoreGRID - Network of Excellence
URL: http://www.coregrid.net

CoreGRID is a Network of Excellence funded by the European Commission under the Sixth Framework Programme

Project no. FP6-004265

Authorizing Grid Resource Access and Consumption

Erik Elmroth [1],
elmroth@cs.umu.se,
Michał Jankowski [2],
Norbert Meyer[2]
{jankowsk, meyer}@man.poznan.pl ,

[1] Department of Computing Science and HPC2N
Umeå University
901 87 Umeå, Sweden
[2] Poznań Supercomputing and Networking Center
ul. Noskowskiego 10
61-704 Poznań, Poland

CoreGRID TR-0161

July 4, 2008

Abstract

The tasks to authorize users access to Grid resources and to authorize their regulated consumption is studied and some key functionality is identified. A novel authorization infrastructure is proposed by combining the Virtual User System (VUS) for dynamically assigning local pool-accounts to Grid-users and the SweGrid Accounting System (SGAS) for Grid-wide usage logging and real-time enforcement of resource pre-allocations.

1 Introduction

The process of granting users access to Grid resources and to authorize their resource consumption in large-scale grids with thousands of users is a complex process. This contribution investigates two vital parts of this process, namely to grant users access to resources without having to á priori open individual user accounts on all Grid resources and to perform real-time regulation of the users' resource utilization based their Grid-wide pre-allocation and previous consumption.

These two problems are examined in some more detail in Section 2. One conclusion is that the authorization process is complex and varying for different usage scenarios. However, by a clear separation of concerns, it is possible to identify well-defined tasks that ideally should be handled by equally well separated components, easily used in concert with other tools, e.g., in accordance with the fundamentals of Service Oriented Architectures (SOA) in general and the Grid eco-system approach [20, 7] in particular.

In Section 3, we review two existing technologies for dynamically assigning temporary local user accounts to Grid users (VUS - The Virtual User System) and for performing real-time enforcement of Grid-wide resource pre-allocations and usage logging (SGAS - The SweGrid Accounting System). Section 4 illustrates how these two solutions can be used in concert, providing key authorization support for a very common usage scenario in academic large-scale Grid environments.

This research work is carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265).

Discussions about possible future extensions and some conclusions are given in Section 5 and and Section 6, respectively, followed by acknowledgments and references.

2 Authorization for Grid Resource Usage

Authorization of Grid resource usage include authorization of both resource access and resource consumption. Compared to traditional authorization problems, the distributed nature of grids gives additional complexity to the problem. In the following, we introduce some important aspects of these problems.

2.1 Authorizing resource access

Authentication, authorization of resource access, and closely related user management are crucial from the point of view of security of any computing system. Free services with anonymous access are rather simple ones, offering only limited functionality (like read-only data access). Also in a strictly commercial usage scenario, in which anyone who is able to pay is allowed use a complex service, needs at least identification of the user.

In the large, distributed, multi-institutional, and complex environment like the Grid, authorization and user management are even more challenging. Problems like managing a global base, mapping global user accounts to local accounts, defining authorization policies for lots of entities (users and services) arise.

The paper [8] provides a definition of the Virtual Organization as a set of individuals or institutions defined by "what (resource) is shared, who is allowed to share, and the conditions under which sharing occurs". This concept allows for easier decentralization of user management (each VO is responsible for managing some group of users). On the contrary, in the classical solution each computing node must store authorization information locally for each user (e. g. in Globus grid-mapfile), an approach which is obviously not scalable and brings a nightmare of synchronization problems. However, in usage scenarios where the mapping to user accounts is not static (i.e., virtual accounts or a similar solution is used), this decentralization requires proper and safe mapping of the Grid user to a local account.

Independently of the resource owner's level of trust to the VO, he or she should retain the ultimate control of the local resource. E.,g. the resource owner should be able to define privileges for the members of VOs and to ban single, unwanted users. In order to realize this, the security policy should be combined from two sources: the VO and the provider. Another requirement is fine grained authorization [14], that allows limiting user access rights to specific resources. The authorization decision must depend on privileges granted to the user by the VO and it can possibly be over-ruled by the provider. More requirements are described in [4].

Of course, the required model of authorization depends on the application, so that, the authorization system must be quite flexible. This many different aspects of the authorization problem has lead to many different solutions and a variety of security tools and services. For instance, there is a number of services that expose the VO part of the security policies: VOMS [2], VOIS [12], CAS [17]. On the other hand, there are several tools for authorization and mapping the users: VUS (described below), LCAS/LCMAPS [1] and others.

2.2 Authorizing resource consumption

In addition to authorizing a user access to a resource, there is often a need to also authorize the actual usage consumption.

Authorization based on usage quantities is directly or indirectly an authorization based on the users ability to "pay" for the usage. In academic environments, users are commonly assigned pre-allocations (quota) of resources, often simply in terms of a certain number of computer hours (per month, per year, etc) on a particular resource or set of resources. The ability to "pay" is in this case a question of determining if the user has already spent his or her pre-allocation or if there is still quota left for running another job. In Grid economies not based on pre-allocations, e.g., based on real cost compensation, the corresponding authorization must be based on the user's real ability to pay.

These two scenarios can be generalized in a number of ways, e.g., depending on the type of Grid economy considered. Independently if the economy is based on real or virtual money, the price setting may be dynamic and the mechanism for this may, e.g., include negotiation before resource allocation [16, 3] or analysis of the fraction of resource utilization during the computation [15]. In either case, there is a need for tools to authorize the actual resource utilization before (and sometimes even during) the utilization of the resource.

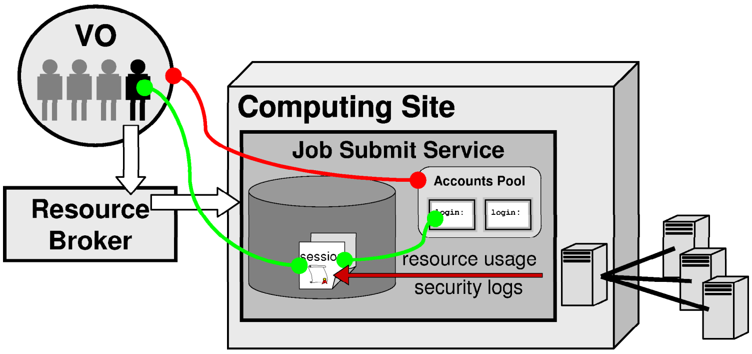

Figure 1: Architecture of the Virtual User System

There may also be a need to authorize the execution based on other aspects of the "size" of the job to be executed. Often, this is a question of how large number of processors a job may utilize, how many hours a job may run, or a combination of these two, although other parameters such as usage of memory, disk, software, etc, may be taken into account. This type of authorization is most often only a question of enforcement of basic policies defining the types of jobs that are allowed to run during different times of the day or week. However, it may also be of interest to apply different policies depending on the privileges of the user or user group. For example, it is often relevant to take into account the total number of jobs currently submitted or running by a certain user or user group. Notably, support for this type of policy enforcement is common in state-of-the-art batch systems, but for a truly distributed Grid, there is a need for similar type of support for Grid-wide policy enforcement, or more specifically, local enforcement of local and global policies based on local and Grid-wide usage information.

To conclude, there are a number of different aspects to cover when authorizing resource consumption. The fact that these authorization aspects can be required in a large number of different combinations is a good argument for developing these authorization mechanism as small separate components, designed for use in concert.

3 Existing Technologies

In the following, we introduce two existing systems, designed to solve some of the authorization problems presented in Section 2.

3.1 The Virtual User System (VUS)

The Virtual User System (VUS) [12], [13], [11], shown on Fig. 1, is an extension of the service that runs users' jobs (e. g., Globus GRAM, gLite Computing Element, etc.). VUS allows jobs to execute without having a personal user account on a site. Instead of personal user accounts for each Grid user, there is a pool of so called virtual accounts dynamically allocated to the users. The user-account mapping mechanism assures that only one user is mapped to a particular account at any given time, so that, the jobs of different users are properly isolated. An unused account may be automatically allocated to another user. This allows minimizing overhead related to creating and maintaining user accounts.

The history of account allocations is stored in the VUS database. The database can store any standard and non standard accounting data and any logging information in the global user context. In that way, tracking user of activities is possible, despite that the user is not permanently mapped to any local account. This is of course crucial from the security point of view.

There are groups of virtual accounts on each computing site. Each group consists of accounts with different privileges, so the fine grain authorization is achieved by selecting an appropriate group by the authorization module of VUS. The authorization module is pluggable and may be easily extended by implementing new plugins. The list of authorization plugins is configurable by the administrator. The mechanism is designed to make it possible to combine authorization policies of a VO and of a resource provider. For example, some plugin may authorize based on the VO membership of the user. The VO manager decides on membership and the resource manager decides if the VO is

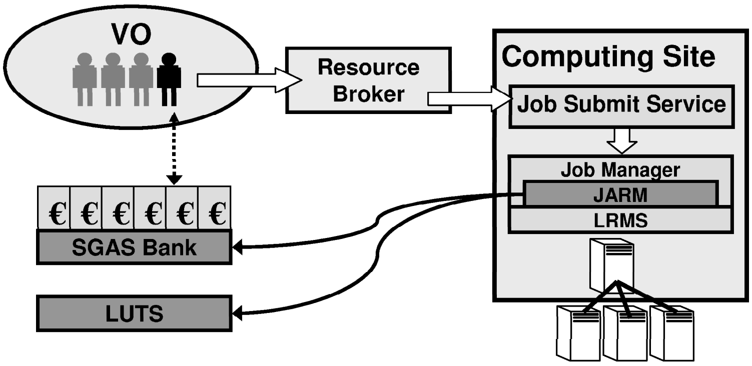

Figure 2: Architecture of the SweGrid Accounting System

authorized and which group is associated to the VO. Another plugin may refuse users appearing on a list of banned users. In this way, the resource provider has enough administrative power while part of this power and many of the administrative work is delegated to VO manager.

VUS is designed for authorization of resource access. It both makes a decision about the authorization (says if the user is allowed or not) and enforces this decision (by assigning the user an account with pre-defined privileges). Note, that the "resource" is in fact the computing site as the whole, access to its "sub-resources" (queue, CPU, disc space, files, etc.) may be limited only by setting privileges and quota limitations to the local account. This approach does not take into account any Grid-level limitations like global access control list or global quota. Such an approach is connected to the localization of VUS in the Grid architecture and the difference between authorization of resource access and consumption.

3.2 The SweGrid Accounting System (SGAS)

The SweGrid Accounting System (SGAS) [9, 19], shown on Fig. 3, includes a set of tools designed for capacity allocation between user groups in collaborative Grid environments by coordinating the enforcement of Grid-wide usage limits. In a typical usage scenario, users or projects are granted Grid-wide usage limits (resource pre-allocations), e.g., by a VO allocation authority. The pre-allocations are to be used Grid-wide, with no pre-defined usage limit on individual resources.

SGAS provides a Bank [6] where bank accounts are used to keep the balance of the usage corresponding to the pre-allocation. The accounting is done in terms of an abstract currency called "Grid Credits". For improved flexibility, the process of determining the cost for using a particular resource is separated from the Bank, i.e., the Bank is totally agnostic to the mapping between resource usage and Grid Credits.

The Bank provides support for reserving Grid Credits before actually charging the account. The typical usage for this feature is for the resource to reserve a sufficiently large amount in the Bank before a job is allowed to start. After job completion, the exact amount (no larger than the reserved amount) is charged. Using this feature, resources can employ real-time quota enforcement by denying resource access if the account does not have sufficient amount of Grid Credits for the reservation. However, the resource can also be configured for soft (as opposed to strict) enforcement, by allowing jobs to run without sufficient Grid Credits, e.g., with lower priority.

For very large grids or for minimizing the risk of reducing the resource utilization due to total Bank outage, the Bank can be separated into different branches. This feature is facilitated through an implementation of a general Name Service for Web Services, which is a the key component, giving the virtual Bank very close to perfect scaling.

SGAS also includes a logging and usage tracking service (LUTS), for storing information about each individual resource utilization. The information is stored in the Usage Record format proposed by OGF-UR [21]. The logging of usage data is not time critical for the job execution and can for improved efficiency be done in batches for multiple completed jobs at regular intervals or when a certain number of jobs are completed.

The Bank and the LUTS are independent services developed using the GT4 (although they can be used from resources running any middleware, given that they can access GT4 services). For integration on particular resources, SGAS provides a Job Account Reservation Manager (JARM) as a single point of integration. The JARM performs all

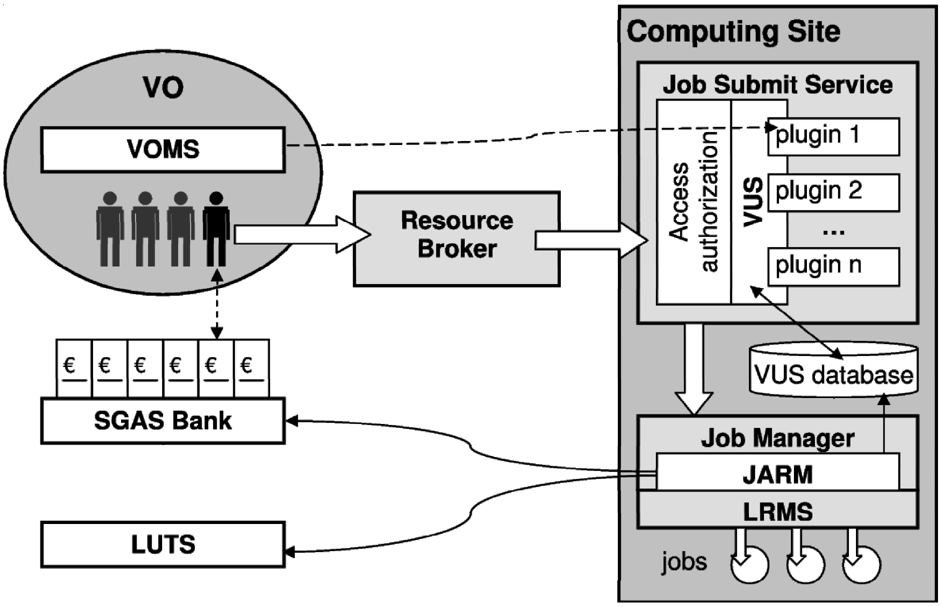

Figure 3: Architecture of the Approach

communication with the Bank (i.e., bank account reservations and charging) and the LUTS (logging of usage data). The JARM needs to be customized to the Grid resource manager used on the resource. It has already been integrated in the GT4 GRAM and the NorduGrid/ARC Grid Manager [9]. SGAS is included in the Globus Toolkit 4 as a tech preview component. For independent accounting system comparisons, praising the SGAS approach, see [18, 10].

4 VUS-SGAS Integration

This section describes how VUS and SGAS can be used in concert for coordinated authorization of resource access and resource consumption. The architecture of our approach is illustrated on Fig. 3. The modules presented in the figure are described below together with an outline of the job submission and execution process.

In the typical Grid scenario, the user is a member of some Virtual Organization. This means in practice, that some facilities (services) for authorization of the VO members are needed for the VO (database of its members, authorization service, etc.).

The figure includes a VOMS [2] for authorization of resource access, although any other service providing similar functionality may be used. In order to authenticate based on VO membership and the role of the user, the VOMS signs a proxy certificate for the user, which is used in the authorization process on the resource.

Another service is needed in order to support authorization of resource consumption, such as an SGAS Bank. The Bank is initiated with pre-allocated usage quotas by a Bank administrator, and it provides information about the quotas remaining based on the Grid-wide resource usage of the Bank account holder, as described in Section 3.

The job submission process is initiated by the user submitting a job, typically via a resource broker, that selects the most appropriate resource for the job at a given time. On the resource, the job is received and managed by a Globus GRAM. In our architecture, the SGAS JARM intercepts the call, and requests Bank account reservation (Hold) of an appropriate amount of Grid Credits. The amount of credits for the reservation is determined from the user's specified maximum execution time. The conversion from execution time to Grid credits is done by the JARM based on the resource cost plan. If the request is successful, the job specification is passed to the GRAM, otherwise it is filtered out, i.e., defined as unauthorized to the requested resource consumption. Notably, the JARM may be configured to make use of local policies for execution of jobs when quota is insufficient. Typically, it may allow such jobs to run with lower priority, possibly subject to the amount of account overdraft.

The Globus GRAM consists of two modules: the job submission service (Gatekeeper in pre-WS GRAM or MJFS in WS GRAM) and the job manager. The first module is responsible mainly for authentication, authorization (of access), mapping the global user identity (distinguished name) to a local one (system account) and then starting an appropriate job manager. These actions take place only once, while the job is submitted. This module is agnostic to job-related information and independent of the local resource management system (LRMS, typically a batch system).

The VUS sub-module, integrated in the job submission service, replaces the standard Globus authorization, as described in Section 3. The VUS may have a number of plugins. One plugin should make use of access authorization policy defined by the VO of the user (by analyzing the VOMS proxy certificate). Any other plugins authorizing access to the resource may be applied according to the local security policy. The last plugin maps the user to an account and stores the mapping in the VUS database. Note, that the account is allocated to the user provided that both types of the authorization have been successful, so that a user account at the resource is not wasted if the job is not allowed to run, e.g.,due to Bank account overdraft.

The job specification is passed to the job manager, which is LRMS specific and able to analyze the job description. Once started, the job manager controls the job through communication with the batch system. After job completion, JARM collects the accounting data, calculates the actual cost and charges the user's Bank account.

The VUS database is capable of storing both standard and non standard accounting metrics and allows defining static prices for a resource unit. The JARM makes use of this price information together with the user-specified predicted maximum execution time and the LRMS accounting data when estimating the expected cost for the execution and for computing the real cost of the job, respectively. Any accounting or logging information may be stored in the database during the execution or anytime later.

The SGAS LUTS may be used for Grid-wide tracking of resource usage. It allows secure publication of the accounting data in the format of OGF Usage Records [21]. The JARM enters this information in the LUTS upon retrieval from the LRMS or after retrieved it from the VUS database for collecting the usage data. Notably, the logging in the LUTS need not be done in real time. On the contrary, it may for performance reasons be done in batches for multiple jobs at a time.

As the job is completed and the temporary user account is no longer used (for some configurable time), it may be released and mapped to another user later on. The accounts are released on demand - when there is no free account for the user, the procedure of releasing accounts is started. This procedure may also be performed periodically.

It may be important in some situations and for some users, to grant access independently of their remaining Grid-wide quota, and to allow the job to run without charging the cost to the Grid-wide bank. For example, a resource provider may want to give access free of charge to the users from his own organization. Such a feature may be implemented either by a modification of the JARM or by including a VUS plugin that can chose between a JARM-enabled and a JARM-free job submission. Notably, such modification would not have to affect the logging in the VUS of local resource usage.

5 Limitations and Future Extensions

For truly large-scale grids, the SGAS LUTS could benefit from being distributed in the same way as the Bank. Such a development is straight-forward, utilizing the same generic Name Service as for the distributed Bank.

In some situation, it may also be beneficial to have bank accounts organized hierarchically. For example, if a group is given a resource pre-allocation from an outside authority, it could be beneficial for the group to have instruments to easily partition and even re-distribute this grant among the group members. Currently, such arrangements can be realized by having the external authority performing the resource allocation on a per-user basis. For improved flexibility for the groups and reduced burden on the external authority, a hierarchical bank structure, e.g., based on the hierarchical share-policy trees in [5], would be beneficial. For this usage scenario, the hierarchical Bank would also need support for delegating (to "group leaders") the right to perform certain account administration tasks.

The scope of this paper does not include the problem of synchronization of the authorization between the broker and the computing site. In other words, the broker should be able to state if the user will be able to run the job on the site before the job is actually submitted there. For that purpose, it should perform exactly the same authorization. In general, the broker cannot read the site configuration, but it may "ping" the site to check if the user is allowed to access the resource. For the broker to assure that jobs will not be denied due to Bank quota overdrafts, the Bank reservation process can be modified so that the broker performs the reservation and only leaves to the resource to perform the post execution charging and logging of the job. The general design of SGAS already allows this usage scenario, although it has so far not been used in practice, since the process where the bank performs both reservation and charging makes the accounting operations more transparent in the job submission process.

6 Concluding Remarks

We have presented a novel approach to Grid job authorization for large-scale grids with thousands of users. By combining established technologies for dynamically assigning Grid users virtual user accounts on individual computers (with (VUS)) and Grid wide accounting and resource allocation enforcement (with SGAS) we obtain concerted authorization for resource access and resource consumption. Notably, despite the focus on large-scale grids, the proposed solution leaves the resource owner with ultimate control over the resource. We also remark that the solution is highly flexible and allows for policy customization, also for rare cases where there is not need for access control (e.g., applications allowing anonymous access) or control of the degree of consumption (e.g., for users with unlimited, flat-rate).

7 Acknowledgements

This work has been supported by the EU CoreGRID NE (FP6-004265), EU BalticGrid NoE (FP6-026715) and The Swedish Research Council (VR) under contract 621-2005-3667. The research has been conducted using the resources of the High Performance Computing Center North (HPC2N).

References

[1] http://www.dutchgrid.nl/datagrid/wp4/lcmaps/.

[2] R. Alfieri, R. Cecchini, V. Ciaschini, L. Dell'Agnello, A. Frohner, A. Gianoli, K.Lentey, and F.Spataro. VOMS: an Authorization System for Virtual Organizations. In *1^{st} European Across Grids Conference*, Santiago de Compostela, February 13–14 2003.

[3] R. Buyya and S. Vazhkudai. Compute power market: Towards a market-oriented grid. In *The First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001)*, May 2001.

[4] J. Denemark, M. Jankowski, K. Ales, L. Matyska, N. Meyer, M. Ruda, and P. Wolniewicz. Best practices of user account management with virtual organization based access to grid. In Roman Wyrzykowski, Jack Dongarra, Norbert Meyer, and Jerzy Wasniewski, editors, *Parallel Processing and Applied Mathematics*, volume 3911 of *LNCS*, pages 633–642. Springer-Verlag, 2006.

[5] E. Elmroth and P. Gardfjäll. Design and Evaluation of a Decentralized System for Grid-wide Fairshare Scheduling. In *e-Science 2005: First International Conference on e-Science and Grid Computing*, pages 221–229, Washington, DC, USA, 2005. IEEE Computer Society.

[6] E. Elmroth, P. Gardfjäll, O. Mulmo, and T. Sandholm. An OGSA-Based Bank Service for Grid Accounting Systems. In *Applied Parallel Computing*, Lecture Notes in Computer Science, pages 1051–1060. Springer-Verlag, 2006.

[7] E. Elmroth, F. Hernandez, J. Tordsson, and P-O. Östberg. Designing service-based resource management tools for a healthy Grid ecosystem. In *Parallel Processing and Applied Mathematics*, volume 4967 of *Lecture Notes in Computer Science*, pages 259–270. Springer-Verlag, 2008.

[8] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications*, 15(3), 2001.

[9] P. Gardfjäll, E. Elmroth, L. Johnsson, O. Mulmo, and T. Sandholm. Scalable Grid-wide capacity allocation with the SweGrid Accounting System (SGAS). *Concurrency Computat.: Pract. Exper.*, To appear, 2008.

[10] M. Göhner, M. Waldburger, F. Gubler, G.D. Rodosek, and B. Stiller. An Accounting Model for Dynamic Virtual Organizations. Technical Report No. 2006.11, University of Zürich, Department of Informatics, November 2006.

[11] M. Jankowski and N. Meyer. Dynamic User Management in the BalticGrid Project. In Paul Cunningham and Miriam Cunningham, editors, *Expanding the Knowledge Economy: Issues, Applications, Case Studies*, volume 4 of *Information anc Communication Technologies and the Knowledge Economy*, pages 1401–1406, Amsterdam, 2007. IOS Press.

[12] M. Jankowski, P. Wolniewicz, and N. Meyer. Virtual User System for Globus based grids. In *Cracow Grid Workshop '04 Proceedings*, Cracow, 2004.

[13] M. Jankowski, P. Wolniewicz, and N. Meyer. Practical Experiences with User Account Management in Clusterix. In *Cracow Grid Workshop '05 Proceedings*, Cracow, 2005.

[14] K. Keahey, V. Welch, S. Lang, B. Liu, and S. Meder. Fine-grain authorization policies in the grid: design and implementation. In *1^{st} International Workshop on Middleware for Grid Computing*, 2003.

[15] K. Lai, L. Rasmusson, E. Adar, S. Sorkin, L. Zhang, and B. A. Huberman. Tycoon: an Implemention of a Distributed Market-Based Resource Allocation System. Technical report, HP Labs, Palo Alto, CA, USA, December 2004.

[16] J. Li and R. Yahyapour. Negotiation strategies for grid scheduling. In *Advances in Grid and Pervasive Computing*, volume 3947 of *Lecture Notes in Computer Science*, pages 42–52. Springer-Verlag, 2006.

[17] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. pages 50–59, 2002.

[18] C-P. Rückemann, W. Müller, and G. von Voigt. Comparison of Grid Accounting Concepts for D-Grid. In *Proc. Cracow Grid Workshop 06, Cracow*, October 2006.

[19] T. Sandholm, P. Gardfjäll, E. Elmroth, L. Johnsson, and O.Mulmo. A service-oriented approach to enforce Grid resource allocations. *International Journal of Cooperative Information Systems*, 15(3):439–459, 2006.

[20] The Globus Project. An "ecosystem" of Grid components.
http://www.globus.org/grid_software/ecology.php. September 2007.

[21] Usage Record WG (UR-WG). https://forge.gridforum.org/projects/ur-wg/, January 2008.