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

Resource Management for Future Generation Grids

U. Schwiegelshohn, R. Yahyapour
{uwe.schwiegelshohn,ramin.yahyapour}@udo.edu

Computer Engineering Institute
University of Dortmund,
44221 Dortmund, Germany

P. Wieder
ph.wieder@fz-juelich.de

Central Institute for Applied Mathematics
Research Centre Jülich
52425 Jülich, Germany

CoreGRID Technical Report
Number TR-0005
May 19, 2005

Institute on Resource Management and Scheduling

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

Resource Management for Future Generation Grids

U. Schwiegelshohn, R. Yahyapour
{uwe.schwiegelshohn,ramin.yahyapour}@udo.edu

Computer Engineering Institute
University of Dortmund,
44221 Dortmund, Germany

P. Wieder
ph.wieder@fz-juelich.de

Central Institute for Applied Mathematics
Research Centre Jülich
52425 Jülich, Germany

CoreGRID TR-0005
May 19, 2005

Abstract

This paper discusses the requirements for and functionalities of resource management systems for future generation Grids. To this end it is also necessary to review the actual scope of future Grids. Here we examine differences and similarities of current Grid systems and distinguish several Grid scenarios to highlight the different understandings of the term Grid which exist today. While we expect that a generic Grid infrastructure cannot suite all application scenarios, it would certainly be beneficial to many of them to share such an infrastructure. Instead of identifying a minimal subset of necessary Grid middleware functionalities, we postulate that Grids need a resource management system both well-designed and rich in features to be usable for a large variety of applications. This includes for example extended functionalities for information and negotiation services which can be used by automatic scheduling and brokering solutions.

1 Motivation

The term "Grid" was introduced in the late 1990s by Ian Foster and Carl Kesselmann [1], but the idea of harnessing and sharing distributed compute power can also be found in the earlier metacomputing concepts of the 1980s. However, "the Grid" ignited broad interest in scientific and public press as the next boost to computing technology in general. The Grid stands today for a new approach that promises to give a wide range of scientific areas the computing power they need. While several running Grids already exist, it has to be acknowledged that about eight years after the advent of Grids they are still far from being commonly usable and robust technological platforms which serve the needs of each and every application scenario. Instead, the term Grid is often associated with a large variety of different meanings and notions. This can easily be observed when following the vivid discussions about a common definition of a Grid [2], [3]. This situation is certainly not an optimal base to start talking about future Grids. Therefore, we use a different, more functional approach in this paper. Under the assumption that sufficient resources are generally available the functionality of Grids is mainly determined by its resource management system (RMS). Hence we will

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

discuss the requirements for and the restrictions of such a RMS in future generation Grids [4]. Nevertheless, we first need to identify the different types of Grids which are associated with different application scenarios [5]. Without any claim for completeness, we list the following scenarios:

High Performance Computing Grids can be considered as the archetype of a Grid. Due to the complexity of many scientific problems, it is frequently impossible to obtain an analytical solution. Instead search heuristics are used to solve many optimization problems and simulation is often the method of choice to determine the behavior of a complex process. These methods proceed to require more computing resources in order to solve more complex problems. Most publicly known Grid projects fall into this scenario in which different computing sites of usually scientific research labs collaborate for joint research. Here, compute- and/or data-intensive applications are executed on the participating HPC computing resources which are usually large parallel computers or cluster systems. The total number of participating sites in this Grid projects is commonly in the range of tens or hundreds, the available number of processing nodes is in the range of thousands. Most people active in Grid research originate from this community and have these kinds of Grid scenarios in mind.

Enterprise Grids represent a scenario of commercial interest in which the available IT resources within a company are better exploited and the administrative overhead is lowered by the employment of Grid technologies. The allocation of applications or business services to available hard- and software resources is maintained via Grid middleware. A common field of interest in this area is the automatic and efficient management of Web Services, e.g. their deployment and invocations on application servers or execution hosts. In comparison to HPC Grids, the resources are typically not owned by different providers and are therefore not part of different administrative domains. This eases the management problem in several aspects.

Global Grids have been foreseen since the early publications about Grids. Here, the vision of a single unified Grid infrastructure is drawn, comprising all kinds of resources from single desktop machines to large scale HPC machines, which are connected through a global Grid network. Systems like Seti@Home or P2P file sharing communities have been given as early examples for such large-scale Grids in which millions of individuals share and access their resources. This scenario resembles the long-term goal of Grid research in which Grids have established a common technological infrastructure similar to the Internet of today.

Grids for Ambient Intelligence/Ubiquitous Computing are also considered in different projects. While the connection of arbitrary electronic devices, like mobile phones, PDAs, household appliances or multimedia components, does not seem directly related to Grids, the core problems which are tackled are very similar in both domains [6]. In this scenario resources have to be discovered, allocated and used in a dynamic environment; security has to be maintained between distributed resources which have to interact on behalf of users.

At this time, one cannot clearly judge which functional scope a future Grid will actually have. At least we can subsume that the presented scenarios have many similar problems to address which are subject of current research. We also note that these research activities are pursued by different research communities which do not always consider their work Grid-related, like for instance the work on Service Oriented Architectures (SOA, [7]) or the work of the Ambient Intelligence community. The scope of the Grid-to-be will almost certainly be limited in terms of the supported application scenarios, and it will be a great challenge to identify the functional boundaries of future generation Grids. Nevertheless, the concept of Grids fills several gaps in the current IT infrastructure. It is expected that the different approaches towards future Grids will merge into a common solution suitable for different application scenarios. In extension to current research on existing Grids, it might be reasonable to take a broader stance on the Grid scope for future generation Grids. In the following we deduce and discuss the implications of a broader Grid view on the resource management problem.

2 Resources in Future Generation Grids

Assuming that future Grids are not limited to small- or medium-sized HPC Grids, we have to identify the relevant resources for future systems. The following is an example collection of Grid resources:

- Compute (nodes, processors etc.)
- Storage (space, location etc.)

- Data (availability, location etc.)
- Network (bandwidth, delay etc.)
- Software (components, licenses etc.)
- Services (functionality, ability etc.)

Most of these resources are already commonly recognized as Grid-relevant like computers and data. Note that the inclusion of data naturally requires also the consideration of storage as a managed resource type. Based on the distributed nature of compute and data resources networks are important for data management as well as for communication requirements during the execution of computational tasks [8]. Moreover, interactive applications have been identified as relevant for Grids, for example visualization tasks with a high demand for bandwidth or the setup of video or multimedia streams. Such use cases make the management of network links and the possibility to request quality of service agreements for these links interesting aspects when considering networks as Grid resources. Furthermore, software components have recently been also considered as dynamic resources in Grids. This includes the identification of available software components as well as the management of software deployment and the life cycle management of software instantiations [9]. While many of these resources are already integrated into current Grids, the actual outreach of the resource notion for future Grids is not yet clear. Although current research often takes the stand that arbitrary services can be managed Grid resources [10], it can be assumed that there will be actual limits of what might be a reasonable Grid resource. Regarding the scope of future generation Grids (see also Section 1) we do not define a set of resources to be taken into account, but demand that resources to be integrated provide the means for automatic management and access sharing. From analyzing the resource types it is clear that the management of some resources is less complex while other types require more extensive coordination or orchestration efforts to be effective. For instance, the combination of hardware with software or data resources is a challenging task as the availability of the different resource types must be coordinated. Reviewing the selection of the Grid scenarios in Section 1, the key criteria for classification is the relation among the resource providers and the users of these Grids. In a general scenario, the resources belong to different and independent resource providers, who claim administrative autonomy and usually have different usage policies. Typically, these policies are enforced by the local resource management systems. As a result, these Grid configurations have a high degree of heterogeneity in the technological features of the local RMS as well as the enforced policies. This leads to major differences between Grid resource management systems and the local resource management systems, as outlined below: The Grid RMS has to deal with many heterogeneous resources in a highly dynamic environment while it has no exclusive control over any resource. In contrast, the local RMS typically manages only one or a few resource types in a static configuration. These resources reside within a single administrative domain in which the RMS has exclusive control over the resources.

3 Requirements for Grid Resource Management

Several services or functionalities are required with respect to the resource management of Grid systems. The following set is a collection which has been identified in the Grid Scheduling Architecture Research Group (GSA-RG) of the Global Grid Forum [11]:

Resource Discovery: It is necessary to discover a resource which fulfills specific constraints or fits certain parameters.

Access to Resource Information: In addition to discovering resources, a common requirement is access to available information about a resource. This includes static as well as dynamic information.

Status Monitoring: Prior and during job execution it is necessary to monitor the condition of a resource or a job. The resource management framework should support event notification to facilitate reactive measures by the respective services.

Brokering/Scheduling: It is essential for Grids that a user does not need to manually coordinate the access to resources. To this end, efficient Grid functions are required which automatically select and schedule resource allocation for jobs.

SLA/Reservation Management: While some users might cope with resources that handle jobs in a best effort fashion, there are application scenarios in which additional information about the resource allocation is necessary. This might include reservations and precise agreements about resource-specific Quality of Service levels.

Execution Management/Provisioning: Clearly the actual execution and management of a job or the required provisioning of a resource is the main task of a Grid RMS system. This includes functionalities to cancel pending jobs or to claim planned allocations.

Accounting and Billing: In many situations, accountability for the resource consumption is required. Especially the inclusion of cost considerations for establishing business models for Grids requires functions for economic services. This includes information about the resource consumptions as well as financial management of budgets, payments and refunding.

We will discuss these functionalities in more detail in Section 4. However, the full stack of middleware functions for Grid resource management will not be necessary for all kinds of Grid systems. Therefore, we can distinguish between two cases of Grid systems with respect to their requirements on resource management capabilities: **Case 1** are **Specialized Grids** for dedicated purposes. These Grids do not expose the full set of above requirements. Moreover, due to a single or limited application domain typically high efficiency is required for the execution. Analogous, a specialized Grid resource management systems is established. This high efficiency comes usually at the price of a higher development support. The RMS is adapted to the specific application, its workflow and the available resource configuration. Thus the interfaces to the resources and the middleware are built according to the given requirements caused by the application scenario. While the Grid RMS is highly specialized, the handling for the user is often easier as the know-how of the application domain has been built into the system. **Case 2** is a **Generic Grid Middleware** which has to cope with the complete set of the requirements above to support an applicability. Here, the Grid RMS is open for many different application scenarios. In comparison to the specialized Grids generic interfaces are required that can be adapted to many front- and backends. However, the generic nature of this approach comes at the price of additionally overhead for providing information about the application. For instance, more information about a particular job has to be provided to the middleware, such as a workflow description, scheduling objectives, policies and constraints. The application-specific knowledge cannot be built into the middleware, and therefore must be provided at the frontend level. In this case the consideration of security requirements is an integral aspect which is more difficult to solve. In fact, security is also an issue in specialized Grids, but on a broader and generic scale a wide variety of security levels and policies exist and must be considered in the middleware. It is possible to hide the additional RMS complexity of generic Grid infrastructures from the users or their applications by specialized components which might be built on top of a generic middleware. Nevertheless, it can be concluded that in general a generic Grid middleware will carry additional overhead with less efficiency at the expense of broader applicability. Current research is mostly focusing on Case 1 in which solutions are built for a dedicated Grid scenario in mind. As mentioned before, these systems are usually more efficient and will therefore remain the favourite solution for many application domains. That is, Case 1 will not become obsolete if corresponding requirements and conditions exist. However, for creating future generation Grids suitable solutions are required for Case 2. If this case is not adequately addressed by Grid research, parallel other activities e.g.by the Web Service community or the Ubiquitous Computing research might provide solutions. While this is generally a positive development, it has to be acknowledged that the key requirements are very similar for many of these scenarios and that rivaling approaches might not yield any added value. Actually, we might have to deal with many different solutions for specific domains without any convergence to a common infrastructure. For instance, instead of creating a specialized customized Grid middleware, a majority of projects could benefit from an early implementation if based on a reliable and robust generic Grid RMS infrastructure that is used by a broad community. Therefore, we argue that future generation Grids should consider a broad application scenario combined with a generic RMS middleware, even at the possible expense of lower efficiency.

4 Grid Resource Management Middleware

Following the previous consideration towards a broader scope of future Grid systems, we continue to discuss the required functionalities in more detail. Instead of identifying a minimal subset of functions which would probably render the resulting middleware inadequate for most application scenarios, we have to carefully identify the relevant set of requirements to make the Grid middleware usable for many scenarios. However, this set should not be too extensive to be implemented into a reliable and robust middleware. This tradeoff is difficult to achieve and should be subject of further discussion in the community. The following selection is a summary of the current discussion in the previously mentioned GGF research group on Grid Scheduling Architecture.

4.1 Resource Discovery

The discovery of existing resources in a Grid does not necessarily indicate that a resource is actually available or suitable for a specific task or for the requesting user. The actual decision about accessing a resource will probably require several steps in the resource selection or scheduling process. Therefore, we first consider only the discovery process. We assume that in a first step it is necessary to identify resources that are principally available in the Grid. The Grid middleware must offer a fast mechanism to identify the resources which fit a given description. The actual selection or scheduling will be based on additional information retrieval or negotiation. While for Grids with a small or limited number of participating sites an index or directory information service is feasible, it becomes a more complex problem to identify which resources are currently available in a Grid. For future generation Grids a flexible and scalable discovery service is required. There are existing technologies e.g. from the peer-to-peer world that work well for large-scale systems. However, for smaller environments other approaches, like the mentioned index and directory services, may be more efficient. A coherent interface which supports both approaches might be reasonable. Nevertheless, it is important to identify resources which meet the requirements for a task in the best possible way. In general it is not sufficient to discover all the resources of a specific type, for the result set might be too large and not really useful. This implies that models for a flexibly parameterized search are required to get a prioritized and small set of "good" resources. Therefore, a flexible way is necessary to limit and steer the search towards the anticipated results. For resource brokerage or scheduling it might be necessary to re-iterate and modify the search criteria to improve the list of available resources.

4.2 Access to Resource Information

Besides the pure discovery of resources covered in the previous paragraph, it will be necessary to access additional information about resources. Some of this information is static or at least valid for a relatively long time. Such information may be suitable for caching or storing in remote information services. Some other information is highly dynamic and therefore not suitable for deferring to remote services. That is, this information must be updated or requested frequently from clients. Therefore, for future Grids the middleware should provide a coherent access to this static and dynamic information. It also has to be considered that this information may not only consist of flat values, but one might need to access more complex and potentially time-variant data collections as well. A typical example is access to information about planned or forecasted future events, like known future resource reservations or forecasted resource allocations. Similarly, there might be need to access information about past events. Some system may utilize such past information to make predictions about future events. It can be debated whether such information is closer related to a monitoring service. Nevertheless, it has to be acknowledged that access to this extensive information is important for many application scenarios and for many scheduling and brokering services. Most of current Grid information services do not support such information. Moreover, many existing services are not suitable for a scalable Grid scenario with many resources and corresponding information sources. Other aspects to consider are security and privacy for the information system in future generation Grids. On one hand, access to information should be as fast and efficient as possible, that is, information should be cached and distributed. On the other hand, access to this information may be restricted to certain users and require secure authentication, i.e. public information may differ from information provided to a specific user and some information may only be available from directly contacting the resources or services within its administrative domain. Therefore a coherent interface to Grid information is necessary which should support caching whenever possible, e.g. if the information does not change for a sufficiently long period. But it should include mechanisms and paradigms to retrieve secured or dynamic information whenever necessary. In all cases, the unified information services for Grids should not be limited to specific resource types but support all kinds of different resources. For instance, data, network and software resources are due to their obvious differences often and probably unnecessarily not handled in the same manner as other resource types. Here it could be discussed whether this is actually necessary or if a more flexible and generic information service could decrease the complexity.

4.3 Status Monitoring

Since we already discussed the requirements for information services which should provide access to information about resources and jobs, it is foreseeable that also the monitoring of an expected resource or job status will be an integral part of Grids. Due to the dynamic nature of Grids, frequent and unexpected changes might require immediate attention by the user or a Grid scheduler. To this end, these changes must be reliably detected and signalled. It can be debated whether these functions are part of a general Grid information service. For better ease of use it will be necessary that monitoring and event notification are provided as core services which do not require high overhead for an application. For future generation Grids it will be important to have a generic and coherent interface for all kinds of status monitoring. This should include, but is not limited to, the monitoring of resource conditions, pending resource agreement, current schedules, job execution, conformance of allocations to service level agreements, and workflow status.

4.4 Execution Management/Provisioning

Obviously a main functionality for a Grid resource management system is the actual execution of a job and/or the provisioning of the resources. Current Grid systems usually support at least the submission of a computational job to the remote RMS. Though, depending on the resource type and nature of the job, the actual provisioning of resources and access to these resources may be different. Again, coherent and generic execution management functions are needed to claim a resource allocation based on an agreement, whether the agreement is the provisioning of a network link with dedicated SLA, or the agreement is a temporarily granted access to a data resource.

4.5 SLA Management/Agreement Management

In previous sections we considered job execution and provisioning of resource allocations as key applications of Grids. While some usage scenarios can do with simple job submission paradigms in which the actual time of execution and quality of service is not planned in advance, there will be many applications which require more precise information or guarantees about execution and allocation. Thus, agreement-based resource planning will be an integral part of Grids. In an agreement which is created before the actual job execution and resource provisioning, all necessary information is exchanged. Such an agreement can be as specific about a job execution as requested or very unspecific if this is sufficient for a particular job. That is, a simple job submission to a remote queuing system without any advance information about the actual job execution time can also be written in an agreement. In the same way, more specific service-level agreements can be made if the service or resource provider supports this.

4.6 Brokering/Scheduling

For a broad proliferation of Grids it will be essential to have a flexible and automatic brokering and scheduling system. Due to the potential high number of available resources and the heterogeneity of the access policies, the resource selection for a job cannot be executed manually by a user. Moreover, there is a need to specify in detail the requirements, preferences or objectives for a job to allow the brokering and scheduling system to automatically select and plan a suitable resource allocation for a task. First, we have to consider that a flexible job description is necessary, which includes information about the job specifics. In addition we will need information about what is required for the job to be executed. While many current Grid systems deal with simple jobs, future Grids have to be able to deal with more complex jobs which include workflows and require co-allocation of several resources with corresponding dependencies. In principle a Grid scheduler should be able to plan a whole workflow in advance if the dependencies are static and the complete information about the resource requirements is accessible. According to the different access and scheduling policies that might exist in a Grid, it cannot be expected that a single Grid scheduling strategy will suit all needs. Instead, it must be taken into account that a Grid will have many different schedulers with varying features and strategies which are optimized for certain applications. For instance, the end-user may provide an individual Grid scheduler. Therefore, it will be essential for Grid RMS middleware to offer well-defined interfaces and protocols to all functions and information a Grid scheduler needs. Such a Grid scheduling architecture will facilitate the implementation of different application-specific Grid schedulers. Considering the broad scope of Grid resources, future research has to analyze how specific management functions for some types of resources can be included in the general brokering and scheduling process. Especially data and network resources are often associated with dedicated management and information services [12], [13]. Regardless of this association current approaches do usually not provide adequate means to integrate such resources into the job brokerage and scheduling process, although the management tasks are very to those of other resources. For instance, the access to planned data transfers or network reservations would be of high interest to a Grid scheduler. Here, consistent and coherent interfaces seem to be necessary for a better integration of the management of these different resources. Ideally, a job on a Grid should be able to request data resources and network links from a scheduler in the same fashion as it would request software components or CPU power. Above we discussed the individual policies established by the providers. To maintain the control at the local RMS systems during the Grid scheduling process, it will be necessary that different scheduling instances interact with each other. One potential approach for future generation Grids to support such interactions will be the provision of negotiation interfaces which permit the creation of agreements [14], [15]. Due to the complexity of the scheduling task in a large-scale Grid environment, such negotiations can take a long time as many iterations of probably parallel negotiations with different resource providers might be necessary. Therefore, any access to information that can support this procedure will be helpful. This can include tentative information about availability, potential quality of service or costs. There might also be the need to combine agreements from different providers to fulfill the requirements of complex job workflows or co-allocation tasks. It is crucial to understand that the scheduling objectives for Grids may differ from well-known criteria in parallel computer scheduling. Due to the availability of many different resources with potentially different costs, the user might have more complex scheduling objectives for his Grid jobs. The simple minimization of response time might be substituted by a complex objective function which includes priorities for costs, quality of service or time preferences. To support Grid schedulers which can deal with these requirements, it will be necessary to support according information and functions in negotiation and agreement interfaces [16], [17]. For instance, it might be necessary to inform local RMS systems about certain preferences to generate better agreements during the negotiations.

4.7 Accounting and Billing

It is anticipated that future Grids provide support for business models including means for corresponding cost and budget management on RMS level. As there should be support for automatic resource selection and allocation based on accounting and billing information, this data must be secured through appropriate techniques like confidentiality, integrity and non-repudiation. For instance, a Grid scheduler might need information about the available budget of a user to allocate a specific resource. This situation may occur during the negotiation and reservation of suitable resources for a job, if the budget has to be allocated prior to the execution of a job. Similarly, functions are needed to refund budget if an negotiated level of service is not provided and re-scheduling is necessary. Scenarios like these emphasize the financial and legal outreach of Grid resource usage and stress the need for reliable and accountable data which can be used to verify transactions.

5 Conclusion

In this paper, we discussed the scope of next generation Grids. While the scope of these Grids is not yet known in terms of supported applications and resources, several assumptions have been made to extend the scope compared to current Grid research. In addition we examined the implications of an extended approach to the Grid resource management infrastructure and provided a compilation of RMS functions necessary to realize such extensions. Although this compilation serves as a starting point for further research, we lay no claim to completeness of the selected and discussed topics. Moreover, the goal of this work is not to propose yet another new Grid middleware. In this area there are currently several activities including those in the Global Grid Forum or the Globus Toolkit approach [18], [10]. Especially the efforts specifying the Open Grid Services Architecture (OGSA, [19]) are intended to identify and specify relevant Grid Services and their interactions. The variety of the different approaches implies future work to compare the different activities' results to obtain compatibility of future generation Grid systems. Apart from the Grid community's efforts there are additional approaches to Grids in other research communities such as Ambient Intelligence/Ubiquitous Computing. Without analyzing the different efforts, it can be seen that many resource management requirements are very similar. Therefore we argue for the identification and adaption of core interfaces and services across different communities and various application domains. Furthermore, the scalability of systems will be a key issue for broad Grid proliferation. The interfaces to core services should remain consistent, independent of the size of a Grid: a single global Grid system, an Enterprise Grid or even a Personal/Ambient Grid should provide comparable functionality and response time. A lot of similar requirements can also be found in the security area, where a common and flexible security model is crucial for many systems. The realization of different implementations while using similar approaches can be seen as redundant and time consuming. Overall, the future Grid resource management system should be transparent to the user and provide a generic and pervasive architecture allowing different inter-operable implementations. An appropriate design should therefore support the reusability of Grid solutions in other application fields, too; otherwise, the current concept of Grids may fail. Such an approach may also prevent an often reported phenomenon in projects which suffer from re-inventing similar services and functions. However, other research disciplines might deliver their own rivaling infrastructures, ending with a multitude of different systems without any convergence towards a flexible and generic infrastructure as Grids have been meant to be. To reach the goal of a generic, inter-operable and re-usable Grid resource management we – as many others in the Grid community – consider a fast agreement on widely accepted standards for architectures, services, patterns, protocols and resources as crucial.

References

[1] I. Foster and C. Kesselman, eds. *The GRID: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1998.

[2] I. Foster. What is the Grid? A Three Point Checklist. *GRIDToday* **1**(6), July 21, 2002.

[3] J. Schopf and B. Nitzberg. Grids: Top Ten Questions. *Scientific Programming, special issue on Grid Computing*. **10**(2):103-111, Aug., 2002.

[4] K. Jeffery, et al. *Next Generation Grids 2 – Requirements and Options for European Grids Research 2005-2010 and Beyond*. July, 2004. Online: ftp://ftp.cordis.lu/pub/ist/docs/ngg2_eg_final.pdf.

[5] J. Nabrzyski, J. Schopf and J. Weglarz, eds. *Grid Resource Management - State of the Art and Future Trends*. Kluwer Academic Publishers, 2004.

[6] N. Davies, A. Friday and O. Storz. Exploring the Grid's Potential for Ubiquitous Computing. *Pervasive Computing* **3**(2):74-75, Apr.-June, 2004.

[7] D. Kane. Service-Oriented Architectures. In *Loosely Coupled – The Missing Pieces of Web Services*, pages 91-112, RDS Press, 2003.

[8] H. Zhang, K. Keahey and B. Allcock. Providing Data Transfer with QoS as Agreement-Based Service. In *Proc. of the IEEE International Conference on Services Computing (SCC 2004)* (D. Feitelson and L. Rudolph, eds.), IEEE, 2004.

[9] G. von Laszewski, E. Blau, M. Bletzinger, J. Gawor, P. Lane, S. Martin and M. Russell. Software, Component, and Service Deployment in Computational Grids. In *Component Deployment: Proc. of the First International IFIP/ACM Working Conference on Component Deployment*. Volume 2370 of Lecture Notes in Computer Science, pages 244-256, Springer, 2002.

[10] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling and S. Tuecke. *From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution*. Technical report, OASIS Web Services Resource Framework (WSRF) TC, March, 2004.

[11] The Global Grid Forum, Grid Scheduling Architecture Research Group (GSA-RG). Web site, 2005. Online: https://forge.gridforum.org/projects/gsa-rg/.

[12] V. Sander, W. Allcock, P. CongDuc, I. Monga, P. Padala, M. Tana, F. Travostino, J. Crowcroft, M. Gaynor, D. B. Hoang, P. Vicat-Blanc Primet and M. Welzl. *Networking Issues for Grid Infrastructure*. Grid Forum Document GFD.37, Global Grid Forum, 2004.

[13] M. Antonioletti, M. Atkinson, R. Baxter, A. Borley, N.P. Chue Hong, et. al. OGSA-DAI Status Report and Future Directions. In *Proc. of UK AHM 2004* (EPSRC, ed.), Sep., 2004.

[14] U. Schwiegelshohn and R. Yahyapour. Attributes for Communication Between Grid Scheduling Instances. In *Grid Resource Management - State of the Art and Future Trends*, pages 41-52, (J. Nabrzyski, J. Schopf and J. Weglarz, eds.), Kluwer Academic Publishers, 2003.

[15] U. Schwiegelshohn and R. Yahyapour- *Attributes for Communication Between Grid Scheduling Instances*. Grid Forum Document GFD.6, Global Grid Forum, 2001.

[16] R. Buyya, D. Abramson, J. Giddy and H. Stockinger. Economic Models for Resource Management and Scheduling in Grid Computing. *Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE)*, 2002. Accepted for publication.

[17] C. Ernemann and R. Yahyapour. Applying Economic Scheduling Methods to Grid Environments. In *Grid Resource Management - State of the Art and Future Trends*, (J. Nabrzyski, J. Schopf and J. Weglarz, eds.), pages 491-506, Kluwer Academic Publishers, 2003.

[18] I. Foster and C. Kesselman. Globus: A metacomputing Infrastructure Toolkit. *The International Journal of Supercomputer Applications and High Performance Computing 11*(2):115-128, 1997.

[19] I. Foster, C. Kesselmann, J.M. Nick and S. Tuecke. The Pysiology of the Grid. In *Grid Computing* (F. Berman, and G.C. Fox and A.J.G. Hey, eds.), pages 217-249, John Wiley & Sons Ltd, 2003.