GES 2007 German e-Science

Available online at http://www.ges2007.de

This document is under the terms of the
CC-BY-NC-ND Creative Commons Attribution

SOME RIGHTS RESERVED

Co-allocation of MPI Jobs with the VIOLA Grid MetaScheduling Framework

Th. Eickermann[1], W. Frings[1], O. Wäldrich[2], Ph. Wieder[1] and W. Ziegler[2]

[1] Research Centre Jülich, Central Institute for Applied Mathematics (ZAM),
52425 Jülich, Germany
email: {th.eickermann, w.frings, ph.wieder}@fz-juelich.de
[2] Fraunhofer Institute SCAI,
53754 Sankt Augustin, Germany
email: {oliver.waeldrich, wolfgang.ziegler}@scai.fraunhofer.de

Abstract

The co-allocation of resources for the parallel execution of distributed MPI applications in a Grid environment is a challenging task. On the hand it is mandatory to co-ordinate the usage of computational resources, like for example compute clusters, on the other hand improves the additional scheduling of network resources the overall performance. Most Grid middlewares do not include such meta-scheduling capabilities, but rely on the provision of higher-level, often domain-specific, services. In this paper we describe the integration of a meta-scheduler, namely the VIOLA MetaScheduling Service, into an existing Grid middleware to provide a framework for co-allocation of MPI jobs. For these purposes, the design and architecture of the framework are presented and, based on the MetaTrace application, the performance of the system is evaluated.

1 Motivation

The MetaTrace [16] simulation of pollutant transport in groundwater is a distributed, parallel message-passing (MPI) application. The performance metric of this application, i.e. the run time for a simulation, depends on the number of processors used to execute the application. This implies, in general, that adding computational Grid resources, like for example additional clusters, is a suitable method to increase the performance of the application and thus retrieve simulation results quicker. Optionally one could perform more detailed or larger simulations, e.g. covering a bigger area of the ground within the same simulation time.

In case the processors used to execute the application belong to multiple clusters, it has to be guaranteed that all of them are available at the same time. Since clusters are typically not idle but loaded with jobs in different states of execution, such a requirement implies additional middleware services to co-allocate the different resources for the different application parts, including the interconnecting network.

It is obvious that such a co-allocation service should not be used to schedule the different parts of solely the MetaTrace application, but any other parallel application. Moreover, it should not be restricted to only computational and network resources, but it should also allow the co-allocation of arbitrary resources types like for example scientific instruments or data. The VIOLA MetaScheduling framework we present in this paper fulfils exactly these requirements.

2 Related Work and State-of-the-Art Technologies

Currently, only a few tools exist that are capable of allocating resources across multiple sites located in different administrative domains, like for example Calana [3] or the GridWay meta-scheduler [10]. However, these approaches lack a number of capabilities that are necessary for reliable orchestration of resources for the execution of a distributed application. Most prominent is the possibility to negotiate with agents of the resource owners a common time-slot for the execution of the distributed application and - once this negotiation is completed - to fix the resource allocation with a Service Level Agreement (SLA) [13]. Moreover, these tools usually demand to also perform the resource management for the local resources following a monolithic approach that is not suitable for a heterogeneous Grid environment. The existing tools provide management of compute resources only, whereas reservation of Quality of Service for the network connections between the compute resources allocated for an application is not possible.

The standardisation of an SLA description and the negotiation of SLAs is addressed in the Grid Resource Allocation Agreement Protocol (GRAAP) working group [8] of the Open Grid Forum [17]. The working group has recently published the final draft of WS-Agreement [1], a document that specifies both a language to express SLAs and a protocol to create them. The specification is expected to become a proposed recommendation of the OGF within the second quarter of 2007.

As a start the current protocol specified by WS-Agreement to create SLAs is covering only simple scenarios with just one negotiation round between resource provider and resource consumer. However, pre-final implementations of the specification revealed and ongoing discussion anticipated that there are other relevant scenarios that require a multi-step negotiation and transaction-like behaviour. Currently a number of approaches exist to find solutions for such negotiation scenarios, some of them are currently explored by the GRAAP working group in order to extend WS-Agreement. HARC [12] is another approach which is developed at the Louisiana State University and used in the EnLIGHTened project for co-allocating computational and network resources. HARC is using Gray and Lamport's Paxos Commit Protocol [9]. The G-lambda project [6] is working on another solution for the co-allocation of computational and network resources based on the Grid Network Service (GNS). GNS is a co-allocation Web Service and supports a two-phase commit protocol.

3 Environment, MetaScheduling Service, and exemplary MPI-Application

In this section we present the Grid environment which has been set up to execute the experiments and we describe the MetaScheduling Service used for co-allocation of compute and network resources. Furthermore the MetaTrace application is introduced which has been deployed in the aforementioned Grid environment to evaluate the performance of the complete system when using co-allocation (as reported in Section 4).

3.1 Grid Environment

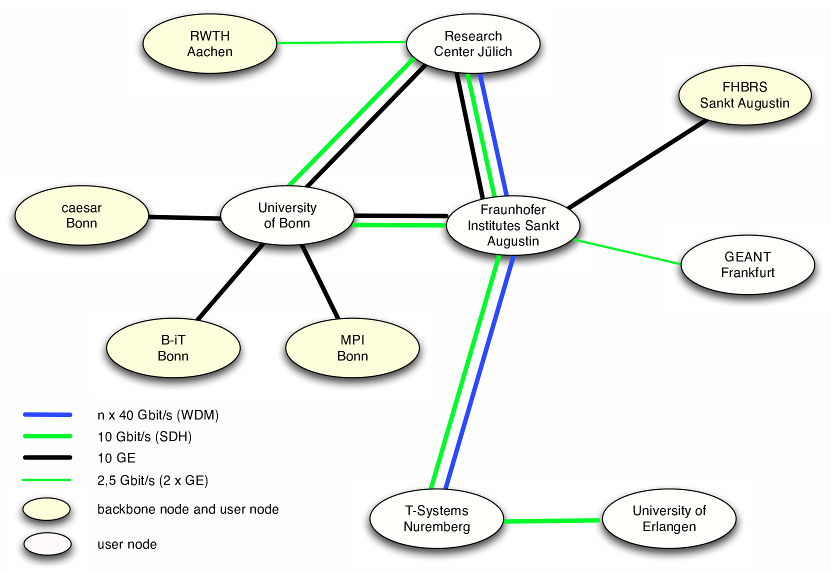

Figure 1: The VIOLA testbed as used for the experiments

D-Grid [4] is the German national effort to create a sustainable infrastructure for e-Science. D-Grid is co-funded by the Federal Ministry of Education and Research and the participating partners from academia, research, and industry. D-Grid supports and provides bundles for three middleware systems: gLite [7], Globus Toolkit 4 [11], and UNICORE 5 [5]. The “Vertically Integrated Optical Testbed for Large Application in DFN” project (VIOLA) [21] is another German project that focuses on using new optical network technologies for Grid environments in a dedicated testbed. After having been made compliant to the D-Grid infrastructure requirements, the resources of the VIOLA testbed (see Fig. 1) have been made available to D-Grid users in autumn 2006.

As the VIOLA project is based on UNICORE, we use this Grid middleware for the experiments described in Section 3.3 and 4. However, the simulation application could also be deployed in a Globus Toolkit 4 environment by replacing the MetaMPI library with mpich-g2. Please note that the UNICORE version we use is not Web Services-compliant, but a redesign towards more open standards is almost completed and will be available mid of 2007 [15].

The groundwater pollution simulation described below was executed in the VIOLA multi-cluster testbed, using three PC clusters. The clusters are located at the Research Centre Jülich, the University of Applied Sciences Bonn-Rhein-Sieg, and the Fraunhofer Institute SCAI. The clusters are interconnected with a 10 GBit/s optical network. All nodes of the clusters are connected with 1 GBit/s links to the local switches that are, in turn, connected to the backbone.

3.2 Co-allocation using the MetaScheduling Service

To solve the problem of co-allocating the usage of resources in Grids, the VIOLA MetaScheduling Service [22] has been developed. It is able to negotiate with the local scheduling systems to find and to reserve a common time slot to execute the application. Additionally, the Quality of Service of the inter-cluster node network connections is negotiated and respective reservations are made using a dedicated resource management system for the network resources.

To use the MetaScheduling Service along with the application, the framework pictured in Fig. 2 has been realised using the UNICORE [20] Grid middleware for job submission, monitoring, and control. Within this framework a user describes the distribution of the parallel MetaTrace application and the requested resources using the UNICORE client, while the remaining tasks like allocation and reservation of resources are executed automatically and are completely transparent for the user.

The framework is designed to flexibly integrate arbitrary resources and to minimise the effort being integrated into different Grid middlewares. To obtain the former it provides an Adapter, a generic interface to the respective (local) resource manager. Currently Adapters for various batch systems and the VIOLA network reservation system exist. To achieve the later, the UNICORE Client (and clients in general), the MetaScheduling Service, and the Adapter communicate using the WS-Agreement protocol and SLAs [14]. This implies that the user specifies the duration of the meta-job and additionally - for each subsystem - reservation characteristics like the number of nodes of a cluster or the bandwidth of the connections between nodes within the UNICORE Client. The client sends the job description to the MetaScheduling Service using the WS-Agreement protocol. Based on the information in the SLA the MetaScheduling Service starts the resource negotiation process. Although WS-Agreement provides the foundation to reserve resources, the co-allocation demand makes the execution of a negotiation protocol at meta-scheduler level necessary:

1. The MetaScheduling Service queries the adapters of the selected local systems to get the earliest time the requested resources will be available.

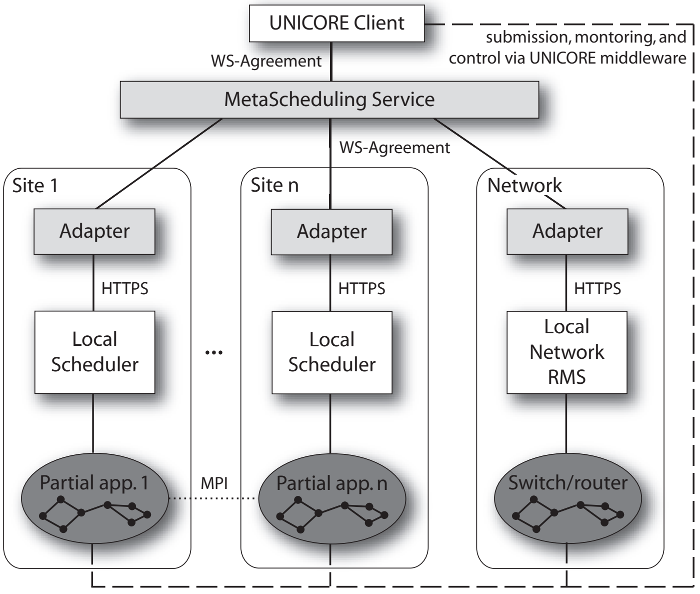

Figure 2: The VIOLA MetaScheduling framework

2. The adapters acquire previews of the resource availability from the individual scheduling systems. Such a preview comprises a list of time frames during which the requested QoS (e.g. a fixed number of nodes) can be provided. It is possible that the preview contains only one entry or even zero entries if the resource is fully booked within the preview's time frame. Based on the preview the adapter calculates the possible start-time.
3. The possible start times are sent back to the MetaScheduling Service.
4. If the individual start times do not allow the co-allocation of the resources, the MetaScheduling Service uses the latest possible start time as the earliest start time for the next scheduling iteration. The process is repeated from step 1. until a common time frame is found or the end of the preview period for all the local systems is reached. The latter case generates an error condition.
5. In case the individual start times match, the MetaScheduling Service checks the scheduled start times for each reservation by asking the local schedulers for the properties of the reservation. This step is necessary because in the meantime new reservations may have been submitted by other users or processes to the local schedulers, preventing the scheduling of the reservation at the requested time.
6. If the MetaScheduling Service detects one or more reservations that are not scheduled at the requested time, all reservations will be cancelled. The latest effective start time of all reservations will be used as the earliest start time to repeat the process beginning with step 1.

7. If all reservations are scheduled for the appropriate time the co-allocation of the resources has been completed.
8. The IDs of the MetaScheduling Service and the local reservations are added to the agreement and a reference to the agreement is sent back to the UNICORE Client.

3.3 Exemplary MPI-Application: MetaTrace

The MetaTrace application is composed of two parts, Trace and Partrace. Trace is a parallel MPI application implemented in Fortran90 that calculates the flow of groundwater. The results, especially a time-dependent vector-field that describes the water flow is transmitted to Partrace, a parallel MPI application implemented in C++. Partrace computes the dynamics of particles solved in the water or deposited in the ground. A transfer of data takes place after every simulation time-step. The duration of a time-step and the amount of transferred data depends on the parameters of the simulation and the number and performance of the involved processors. Typical values are 200 MB transfers every 10 to 15 seconds. The transfer uses MPI send and receive calls and is performed in parallel, using several nodes and network adapters of the clusters.

In order to execute MetaTrace distributed on several clusters in the VIOLA test-bed, MetaMPICH [19], a Meta-computing enabled MPI-implementation of the RWTH Aachen, has been used. MetaMPICH is based on MPICH 1.2. It has been enhanced by RWTH to support multiple communication devices for a single application: e.g. a fast interconnect inside a cluster or supercomputer, and TCP/IP between clusters. This allows to make optimal use of the available bandwidth, both inside and between the systems that compose the meta-computer.

4 Performance Evaluation

We already evaluated the performance of the MetaScheduling Service itself and the influence of negotiation on the application's performance in previous work [22]. With respect to the MPI application presented here we have been primarily interested in the performance increase due to the reservation of dedicated network bandwidth. Therefore we compared application runs without reserving the network between the compute nodes to runs with different bandwidths exclusively assigned to the MPI application. We observed a significant increase in the data rates between the different nodes and consequently a better performance of the application itself.

For our tests the MetaTrace application was distributed on two or three clusters of the VIOLA Grid. In these test scenarios Partrace was executed on the Cray XD1 in the Research Centre Jülich. The SMP clusters at the Research Centre caesar and University of Applied Science Bonn-Rhein-Sieg (FH BRS) were used for Trace. In the case of three clusters, Trace itself was also distributed to clusters at caesar and the FH BRS.

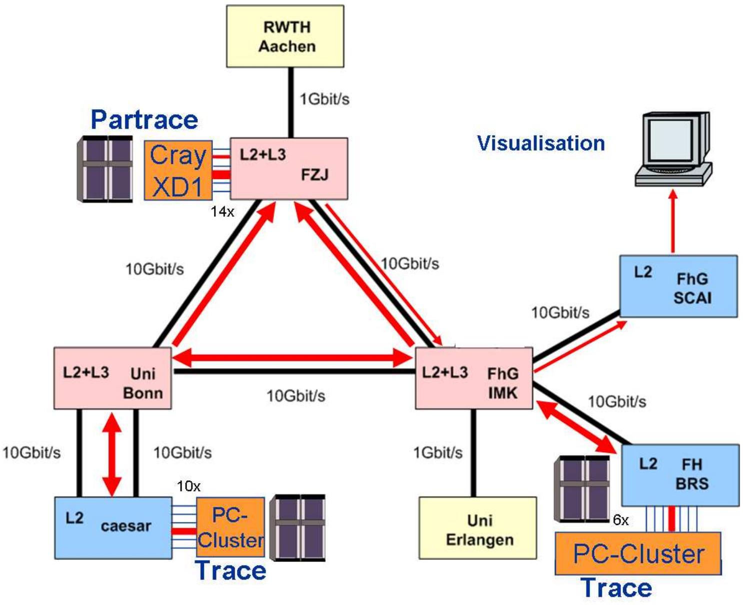

Figure 3: Setup of the MetaTrace experiments in the VIOLA testbed

In this configuration every cluster is directly connected via 10 Gigabit Ethernet to every other cluster (Fig. 3). The clusters are attached with different numbers of Gigabit Ethernet (GE) adapters to the test-bed: 34 adapters at caesar, 10 at the Cray XD1, 6 at the FH BRS. A general observation was that the application could use up to about 500 Mbit/s per GE adapter if sufficient bandwidth was available in the testbed. Fig. 4 and Fig. 5 illustrate this. They show the bandwidth utilised by the application, if no other traffic was present in the test-bed. In Fig. 4, the 6 GE adapters of the cluster at FH BRS limit the throughput to about 3.3 GBit/s. In Fig. 5, the 10 GE adapters of the Cray XD1 limit the throughput to about 5.2 GBit/s.

In our experiments, the distributed application was run with different bandwidth reserved in the network for exclusive use by the application, ranging from 1 GBit/s to 10 GBit/s. In all cases, the application performance was completely independent of the amount of additional traffic in the testbed, showing that the reserved bandwidth was in fact available solely for the application. As long as the reserved bandwidth was above 5 GBit/s for three clusters, the overall application performance was also not affected by the reservation. With lower bandwidth reservations, the utilised bandwidth is of course limited by the reservation. In that case the time that the application spends with communication increases with decreasing bandwidth. For example, a reduction of the reserved bandwidth from 10 GBit/s to 1 GBit/s increases the communication time per time-step from 0.3 sec to about 1.9 sec. This means that the fraction of the overall application runtime, that is spent with communication increases from 3% to 17%. Without bandwidth reservation, the application performance becomes unpredictable and varies with the amount and nature of additional traffic in the testbed.

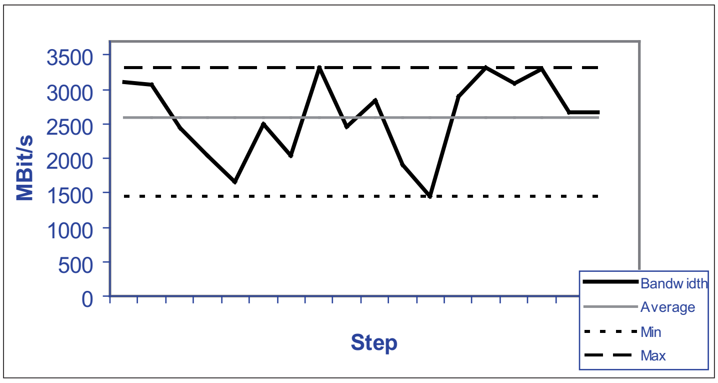

Figure 4: MetaTrace results running MPI jobs on two clusters

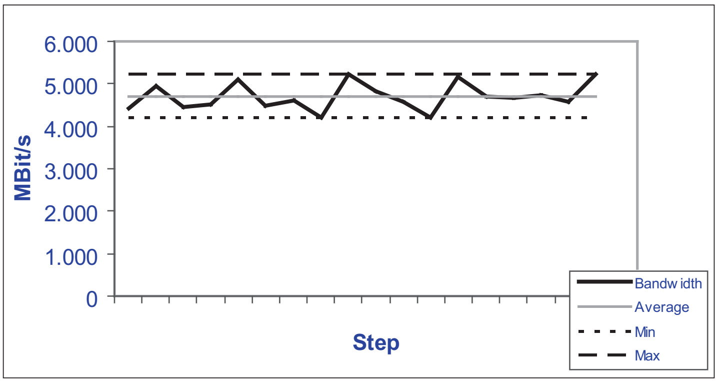

Figure 5: MetaTrace results running MPI jobs on three clusters

5 Conclusions

In this paper we presented a Grid framework to co-allocate MPI jobs. We integrated the VIOLA MetaScheduling Service and the UNICORE middleware to run the MetaTrace application on the VIOLA testbed. This framework has been implemented and successfully demonstrated at the IST 2006 in Helsinki and the CoreGRID Industrial Conference 2006 in Sophia Antipolis.

The framework is further developed within a number of projects. The overall goal is to evolve the framework towards a workflow meta-scheduler with support for arbitrary types of resources. First results of experiments made [23] proved true the expectation that a reduction of turnaround times is achievable when doing advance reservation of resources for a workflow. The achievements observed were gains in turnaround time of up to 45% depending on the number of CPUs and the time needed to execute the components of a workflow.

As part of the scientific work done by the CoreGRID Institute on Resource Management and Scheduling, the framework and the Intelligent (Grid) Scheduling System (ISS) are integrated to provide scheduling solutions for the Swiss-Grid [2]. Moreover, the MetaScheduling Service is the foundation for a co-allocation service of the Phosphorus [18] project, aiming at facilitating communication among Grid middleware services, network resource provisioning systems, and a GMPLS control plane for Grids. Last but not least it is planned to integrate the results of the MetaScheduling Service development into a Grid Scheduling solution for the D-Grid.

Acknowledgements

The work reported in this paper is funded by the German Federal Ministry of Education and Research through the VIOLA project under grant #01AK605L. This paper also includes work carried out jointly within the CoreGRID Network of Excellence funded by the European Commission's IST programme under grant #004265.

References

1. A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. WS-Agreement - Web Services Agreement Specification, final version, March 20, 2007. <https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.graap-wg/docman.root.current_drafts/doc6091/48>.
2. K. Cristiano, R. Gruber, V. Keller, P. Kuonnen, S. Maffioletti, N. Nellari, M.-C. Sawley, M. Spada, T.-M. Tran, O. Wäldrich, Ph. Wieder, and W. Ziegler. Integration of ISS into the VIOLA Meta-scheduling Environment. In *Proc. of the 2nd CoreGRID Integration Workshop*, volume 4 of *CoreGRID Series*. Springer, 2006. To appear.
3. M. Dalheimer, F.-J. Pfreund, and P. Merz. Calana - A General-purpose Agent-based Grid Scheduler. In *Proc. of the 14th IEEE International Symposium on High Performance Distributed Computing (HPDC-14)*. IEEE Computer Society Press, 2005.
4. D-Grid Initiative. Web site. 14 Aug 2006 <http://www.d-grid.de/index.php?id=1&L=1>.

5. D. Erwin (ed.). UNICORE Plus Final Report. Technical report, Research Center Jülich, Germany, 2003. ISBN 3-00-011592-7.
6. G-lambda Project. Web site. 29 Mar 2007 <http://www.g-lambda.net/wordpress/>.
7. gLite - Ligthweight Middleware for Grid Computing. Web site. 14 Mar 2007 <http://glite.web.cern.ch/glite/>.
8. Grid Resousource Allocation Agreement Protocol Working Group. Web site. 29 Mar 2007 <http://www.ogf.org/gf/group_info/view.php?group=graap-wg>.
9. J. Gray and L. Lamport. Consensus on transaction commit. Technical report MSR-TR-2003-96, Microsoft Research, 2004. <http://research.microsoft.com/research/pubs/view.aspx?tr_id=701>.
10. GridWay Metascheduler, 2006. Web site. 20 Dec 2006 <http://www.gridway.org/>.
11. Globus Toolkit 4. Web site. 14 Mar 2007 <http://www.globus.org/toolkit/>.
12. HARC: The Highly-Available Resource Co-allocator. Web site. 29 Mar 2007 <http://www.cct.lsu.edu/ maclaren/HARC/>.
13. J. J. Lee and R. Ben-Natan. *Integrating Service Level Agreements*, chapter 1. Wiley Publising, Inc., 2002.
14. H. Ludwig, T. Nakata, O. Wäldrich, Ph. Wieder, and W. Ziegler. Reliable Orchestration of Resources using WS-Agreement. In *Proc. of the 2006 International Conference on High Performance Computing and Communications (HPCC-06)*, volume 4208 of *LNCS*, pages 753–762. Springer, 2006.
15. R. Menday. The Web Services Architecture and the UNICORE Gateway. In *AICT-ICIW '06. Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet*, page 134. IEEE Computer Society, February 19–25, 2006.
16. VIOLA MetaTrace, 2006. Web site. 20 Dec 2006 <http://www.viola-testbed.de/content/index.php?id=metatrace>.
17. Open Grid Forum. Web site. 29 Mar 2007 <http://www.ogf.org>.
18. Phosphorus, 2006. Web site. 20 Dec 2006: <http://www.ist-phosphorus.eu/>.
19. M. Pöppe, S. Schuch, and T. Bemmerl. A Message Passing Interface Library for Inhomogeneous Coupled Clusters. In *Proceedings of the CAC Workshop at the IPDPS03,*, 2003.
20. A. Streit, D. Erwin, Th. Lippert, D. Mallmann, R. Menday, M. Rambadt, M. Riedel, M. Romberg, B. Schuller, and Ph. Wieder. UNICORE - From Project Results to Production Grids. In L. Gandinetti, editor, *Grid Computing: The new Frontier of High Performance Processing*, number 14 in Advances in Parallel Computing, pages 357–376. Elsevier, 2005.
21. VIOLA – Vertically Integrated Optical Testbed for Large Application in DFN, 2006. Web site. 29 Mar 2006 <http://www.viola-testbed.de/>.
22. O. Wäldrich, Ph.Wieder, and W. Ziegler. A Meta-scheduling Service for Co-allocating Arbitrary Types of Resources. In R. Wyrzykowski, J. Dongarra, N. Meyer, and J. Wasniewski, editors, *Proc. of the Sixth International Conference on Parallel Processing and Applied Mathematics (PPAM 2005)*, volume 3911 of *LNCS*, pages 782–791. Springer, 2006.
23. P. Wieder, Oliver Wäldrich, Ramin Yahyapour, and Wolfgang Ziegler. Improving Workflow Execution through SLA-based Advance Reservation. In S. Gorlach, M. Bubak, and T. Priol, editors, *Integrated Research in Grid Computing, CoreGRID Integration Workshop 2006, Krakow, Poland*, pages 333–344. Academic Computer Centre CYFRONET AGH, 2006. ISBN: 83-915141-6-1.