CO-ALLOCATING COMPUTE AND NETWORK RESOURCES - BANDWIDTH ON DEMAND IN THE VIOLA TESTBED

Thomas Eickermann, Lidia Westphal
Central Institute for Applied Mathematics
Research Centre Jülich,
52425 Jülich, Germany
{th.eickermann, l.kirtchakova}@fz-juelich.de

Oliver Wäldrich, Wolfgang Ziegler
Department of Bioinformatics
Fraunhofer Institute SCAI
53754 Sankt Augustin, Germany
{oliver.waeldrich, wolfgang.ziegler}@scai.fraunhofer.de

Christoph Barz, Markus Pilz
University of Bonn
D-53125 Bonn, Germany
{barz, pilz}@cs.uni-bonn.de

Abstract Distributed applications or workflows need to access and use compute, storage and network resources simultaneously or chronologically coordinated respectively. Examples are distributed multi-physics simulations that use the combined computational performance and data storage of multiple clusters. A coordinated reservation and allocation of the resources is a prerequisite for the efficient use of such resources. This contribution describes the components of a system that provides Grid users with this functionality. The Grid middleware UNICORE is extended to access a MetaScheduling Service (MSS) performing orchestration of resources of different administrative domains, using advance reservation capability of local resource management systems (RMS) - including network connections for which ARGON serves as RMS. ARGON leverages Bandwidth on Demand, a cornerstone of next generation Grid enabled optical networks rendering the network to a first class Grid resource.

Keywords: Grid, advance reservation, Bandwidth on demand, ARGON, UNICORE

1. Introduction and Overview

Advanced applications usually benefit from the existence of different, heterogeneous resources available in Grids. Being able to select among multiple resources allows the end-user to execute the individual components of his application using the most appropriate resources available. Examples of such applications are distributed multi-physics simulations where multiple resources are needed at the same time, or complex workflows where the resources are needed with some timely dependencies [11].

Additionally, having distributed applications and data, there is also a need for dedicated QoS of the network connections between the resources to support efficient execution of the applications. However, to make efficient use of the resources we need reservation mechanisms that guarantee the availability of the selected resources including the network at the time they are needed to execute application components or a component of a workflow. Without reservation there is only a best effort approach to execute applications across multiple resources without a chance of coordination. Having reservation mechanisms allows to completely planning the execution of an application or workflow if the timely dependencies are given by the user. In the VIOLA [15]project we created a UNICORE based Grid testbed on top of an optical network.

This testbed provides solutions to the problems addressed above: the orchestration of resources of different sites belonging to different administrative domains is done by a MetaScheduling Service (MSS) [16]. This service is responsible for the negotiation of agreements on resource usage with the individual local resource management systems. The agreements are made using WS-Agreement [1]developed by the GRAAP [8]working group of the Global Grid Forum [7]. The agreements made basically are Service Level Agreements on the advance reservation of the resources needed for an application or a workflow [17]. The local resource management systems finally include the advance reservation in their individual schedules. Extending this approach to network resources as done in VIOLA allows user or application driven selection and reservation of network connections with dedicated QoS based on evolving network technologies.

2. Architecture

2.1 The UNICORE Environment and Extension of the Client

The Grid-system UNICORE [13]is being developed since 1998 and is used in various projects and production environments, mainly in Europe and Japan. UNICORE is based on a three-tier architecture, consisting of (1) a Java-Client as the user-interface to the Grid, (2) server-components at the UNICORE-sites that provide the secure access of the user to the UNICORE Grid and manage the users jobs and finally (3) the target systems which execute those jobs (see Figure 1).

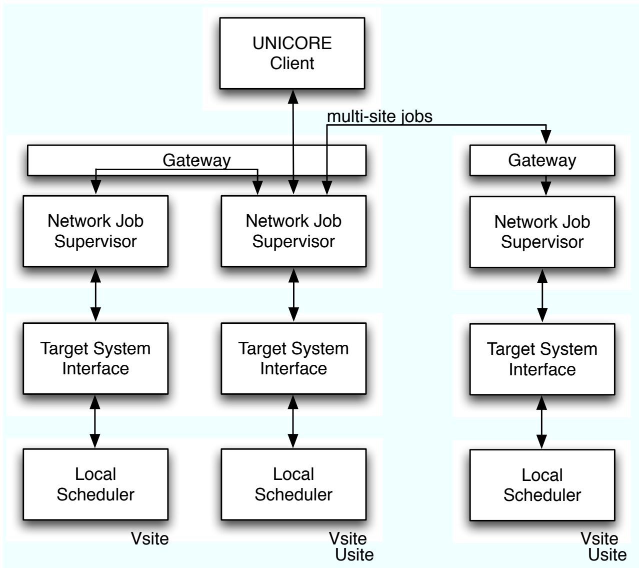

Figure 1. UNICORE Architecture

The standard UNICORE software offers extended workflow support. UNICORE jobs are composed of subjobs that can be executed on the same or different resources (called vsites). Dependencies between those subjobs can be specified, forcing them to be executed in a particular order. In addition to that, conditional execution and control statements allow to build loops of subjobs. However, UNICORE has no build-in capabilities to make advance reservations or to provide synchronous access to distributed resources.

Within VIOLA, this feature has been added via a UNICORE Client-plugin that accesses an external MetaScheduling Service. The plugin provides a GUI that lets the user specify his job including the number of processes to run on which target systems and the required bandwidth between them. Based on this information, the client requests a reservation from the MSS. Once the reservation has been made by the MSS, it is processed like any other UNICORE job. A job may consist of a number of subjobs one for each target system that is requested. Users can retrieve output, monitor or cancel the job.

In the current version, the plugin is tailored to the needs of distributed simulations using the metacomputing-enabled MPI-implementation MetaMPICH [3]. Using it, the user not only specifies the resources needed but also further MetaMPICH-related information allowing the plugin to perform additional tasks, as e.g. distributing the different types of MetaMPICH tasks (compute tasks, network router tasks, I/O server tasks) onto the requested cluster nodes based on various policies and generating a MetaMPICH configuration file.

The plugin is designed and implemented in a modular fashion, allowing easy adaptation to other types of distributed application, not based on MetaMPICH. An example under consideration is the distributed simulation of crystal growth in the VIOLA application TechSim. Here, two MPI-applications are coupled via MpCCI [9]using plain TCP/IP sockets.

2.2 The MetaScheduling Service (MSS)

Once the MSS receives the agreement proposal with the necessary information on resources and QoS needed for an application from the UNICORE client it starts to negotiate with the local Resource Management Systems (RMS) of these resources (see Figure 2). The negotiation has four main phases:

1 querying the local RMS of the selected systems for free slots to execute the application within a preview period

2 determining a common time slot (this is done in parallel for all RMS)

3 if such a time-slot exists, perform a reservation request of this slot on behalf of the user.
otherwise
restart the query with a later start time of the preview period

4 check whether the reservation was made for the correct time slot on all systems (because local job requests might interfere),
if yes
we are done;
otherwise
restart the query with a later start time of the preview period.

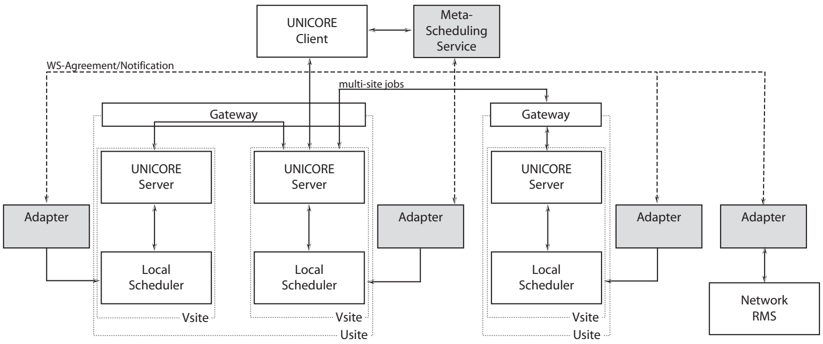

Figure 2. Architecture of the VIOLA MetaScheduling Environment

If no common time-slot within the local RMSs specific look-ahead times can be identified, an error is reported to the user. The pseudo-code of the co-allocation algorithm is depicted in Listing 1. The successful negotiation and reservation is sent back as agreement to the UNICORE client which then processes the job as usual. When the job starts at the negotiated common starting time the MSS collects the IP addresses of the participating machines (this information may not be available at an earlier time as the local scheduling system might assign the job to different nodes than planned at the time of submission) and communicates them to the network RMS which in turn is then able to manage the end-to-end connections with the requested QoS.

```
set n = number of requested resources
set resources[1..n] = requested resources
set properties[1..n] = requested property per resource # number of nodes, bandwidth,
                                                       # time, ...
set freeSlots[1..n] = null                             # start time of free slots
set endOfPreviewWindow = false
set nextStartupTime = currentTime+someMinutes          # the starting point when
                                                       # looking for free slots
while (endOfPreviewWindow = false) do {

  for 1..n do in parallel {
    freeSlots[i] = ResourceAvailableAt(resources[i], properties[i], nextStartupTime)
  }

  for 1..n do {
    set needNext = false
    if (nextStartupTime != freeSlots[i]) then {
      if (freeSlots[i] != null) then {
        if (nextStartupTime < freeSlot[i]) then {
          set nextStartupTime = freeSlots[i]
          set needNext = true
        }
      } else {
        set endOfPreviewWindow = true
      }
    }
  }
}

if ((needNext = false) & (endOfPreviewWindow = false)) then return
  freeSlots[1] else return "no common slot found"
```

Listing 1: Pseudo code of the common timeslot negotiation algorithm

2.3 Advance Network Reservation

Taking a look at the Grid as a geographically distributed set of resources comprising computing and storage for users and their applications, the connecting network infrastructure becomes important. While sites are usually connected by IP best effort technologies, the coordination of high performance resources like meta-computing brings new requirements and challenges to the network.

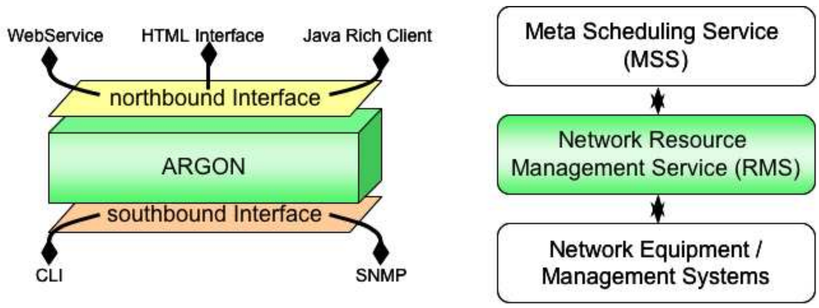

Figure 3. North- and sourthbound Interfaces of ARGON

A sites Internet connectivity is usually tailored to the bandwidth demands of the well-known interactive Internet applications like e-mail and web traffic. It is assumed that coupling clusters to efficiently use computing and storage resources from multiple sites requires high bandwidth (e.g. in terms of multiple Gbit/s) and low delay (e.g. as low as possible) connections with virtually exclusively usage characteristics. The idea of QoS in the network domain has been apparent for many years [10], [4]. In addition, the VIOLA project provides an in advance reservation interface which allows to connect sites on demand with high speed, low delay connections.

These premium connectivity services can be invoked by the Meta-Scheduling Service in order to provide on demand the required network QoS for multi-site jobs. The following section presents a brief overview of the developed network RMS ARGON [2](Allocation and Reservation in Grid-enabled Optic Networks) including the advance reservation capable interface for the Grid application layer offering connectivity services with a specified QoS on top of the optical network between the Grid sites in the VIOLA network. Figure 3 shows the north- and southbound interfaces of ARGON.

ARGON is designed to provide a set of network related services to the Grid community, e.g. advance reservations can be requested by the upper layer (e.g. MSS). This includes the instantaneous setup of network connections if the requested resources are available for the specified span of time. At this level ARGON tries to hide the details of the network technologies, i.e. the user or application specifies QoS requirements for a service and describes the service endpoints.

ARGON maps abstract premium connectivity services to specific layer 2 and layer 3 network services via MPLS as well as point-to-point connectivity services via GMPLS. Beside the details of a single service, a set of services can be bundled in a single request for reservation. Hence, a reservation may consist of several services with chronological dependencies which may themselves consist of several connections as a basis for the service. Consequently, the whole reservation can be regarded as a transaction: All services contained are accepted, rejected or postponed as a whole. This also applies for malleable reservations where the overall service of the reservation can be stretched or compressed in the same way. The idea of malleable reservations is sketched in Figure 4. A data amount has to be transferred and according to the present resource allocation and reservation parameters, ARGON can choose an appropriate duration and capacity frame to schedule the service.

In order to allow for automated resource coordination and provisioning, the northbound interface is implemented as a WebService and accessible via SOAP [12]. The interface currently consists of five message types for reservation of resources, cancellation of reservations, query of reservation related information, availability information and the binding of additional information for provisioning purposes. Availability information and binding of provisioning information are especially important for the co-allocation of resources via the MSS. The availability request helps to find a common time slot for cluster and network resources.

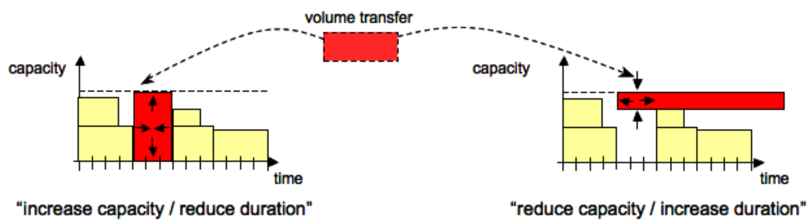

Figure 4. Malleable Reservations

A late binding of provisioning information allows for the MetaScheduling Service and the local scheduling systems respectively appointing the cluster nodes used for a reservation just in time before the provisioning. At the time of reservation only the service endpoint (e.g. provider or consumer edge router), but not the identity of the cluster nodes needs to be known. The provisioning information may consist of ports of the router to which the cluster nodes are attached and/or IP addresses.

The southbound interface of ARGON to the network components uses standard network management protocols if available to initiate MPLS/GMPLS based signalling to control both the MPLS and the GMPLS domain. At the time of writing, the primary interfaces to the network equipment use either a Command Line Interface (CLI) which is not only vendor specific but also version dependent and SNMP if possible. It is also envisioned to integrate vendor specific management interfaces that support XML message transfer with a higher layer of abstraction. In the context of MPLS two services are favoured by ARGON: A layer 3 based tunnel service and VPLS. The layer 3 based tunnel service utilizes MPLS traffic engineered point-to-point tunnels which convey IP packets.

One of the next topics for the network RMS ARGON includes the challenge of multi-domain reservations (east-/westbound interface). This topic includes the interaction between multiple ARGON systems or other network RMS like UCLP [14], G-lamda [6]and DRAC [5]which provide similar ideas to build next generations Grid enabled optical networks.

The core of ARGON utilizes the network topology information to compute the possible paths in the network to realize and plan the requested services in advance. Although the network equipment in the VIOLA testbed allows for traffic engineering, in on demand and in advance reservations must be handled by ARGON. Protocols used for traffic engineering like OSPF-TE and RSVP-TE provide means for instantaneous path computation and signalling within the network components. Preplanning of future capacity usage is therefore left to the core of ARGON which supervises the resource usage in the underlying network layers like MPLS and GMPLS.

3. Outlook

The current version of the VIOLA Grid testbed expects the user to describe the resource demands of his application using the UNICORE client and do a pre-selection of resources satisfying this demand. However, we are working on several other projects to have applications providing this information. Annotating applications with the knowledge about their requirements will allow to make the resource pre-selection process more automatic and disburden the user from this task.

The communication of the MSS with the other components of the system is based on WS-Agreement. WS-Agreement version 1 does not support re-negotiation of agreements already accepted an extended version with richer negotiation capabilities is under preparation. Once this version becomes available we will switch to the new version. In the FP6 funded project UniGrids, a WS-based version of UNICORE is under development. A tighter integration of the MSS into the UNICORE system is delayed until UNICORE/GS will be available.

Acknowledgments

Some of 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. Web Services Agreement Specification (WS-Agreement), March 2007. 15 Mar 2007 <https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.graap-wg/docman.root.current_drafts/doc6091/>.

[2] ARGON - Allocation and Reservation in Grid-enabled Optic Networks. VIOLA Project Report, March 2006 <http://www.viola-testbed.de/>.

[3] B. Bierbaum, C. Clauss, Th. Eickermann, L. Kirtchakova, A. Krechel, S. Springstubbe, O. Wäldrich, Ph.Wieder, and W. Ziegler. Reliable Orchestration of distributed MPI-Applications in a UNICORE-based Grid with MetaMPICH and MetaScheduling. In *Proc. of the EuroPVM/MPI 2006*, volume 4192 of *Lecture Notes in Computer Science*, pages 174 – 183. Springer, 2006.

[4] S. Chen and K. Nahrstedt. An Overview of Quality-of-Service Routing for Next Generation High-Speed Networks: Problems and Solutions. In *IEEE Network, Special Issue on Transmission and Distribution of Digital Video*, volume 12, No 6, pages 64–79. IEEE Communications Society, 1998.

[5] Dynamic Resource Allocation Controller (DRAC). March 2007 <http://www.nortel.com/drac/>.

[6] G-lambda Project. March 2007 <http://www.g-lambda.net/>.

[7] GGF – The Open Grid Forum. Mar 2007 <http://www.ogf.org>.

[8] Grid Resource Allocation Agreement Protocol Working Group. Mar 2007 <https://forge.gridforum.org/projects/graap-wg/>.

[9] MPCCI - Multidisciplinary Simulations through Code-Coupling. Mar 2007 <http://www.scai.fraunhofer.de/mpcci.html/>.

[10] P. Paul and S. V. Raghavan. Survey of QoS Routing. In *Proceedings of the 15th international conference on Computer communication*, Mumbai, Maharashtra, India, 2000.

[11] G. Quecke and W. Ziegler. MeSch – An Approach to Resource Management in a Distributed Environment. In *Proc. of 1st IEEE/ACM International Workshop on Grid Computing (Grid 2000)*, volume 1971 of *Lecture Notes in Computer Science*, pages 47–54. Springer, 2000.

[12] Simple Object Access Protocol Specification. SOAP Specification version 1.2. Web site, 2007. Online: <http://www.w3.org/TR/soap12/>.

[13] 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. Grandinetti, editor, *Grid Computing: The New Frontiers of High Performance Processing, Advances in Parallel Computing 14*. Elsevier, 2005.

[14] User-controlled LightPaths. March 2007 <http://www.canarie.ca/canet4/uclp/>.

[15] VIOLA – Vertically Integrated Optical Testbed for Large Application in DFN. Mar 2007 <http://www.viola-testbed.de/>.

[16] O. Wäldrich, Ph.Wieder, and W. Ziegler. A Meta-scheduling Service for Co-allocating Arbitrary Types of Resources. In *Proc. of the Second Grid Resource Management Workshop (GRMWS'05) in conjunction with the Sixth International Conference on Parallel Processing and Applied Mathematics (PPAM 2005)*, volume 3911 of *Lecture Notes in Computer Science*, pages 782–791, Poznan, Poland, September 11–14, 2006. Springer.

[17] Ph. Wieder, O. Wäldrich, R. Yahyapour, and W. Ziegler. Improving Workflow execution through SLA-based Advance Reservation. In *Intregrated Research in Grid Computing, CoreGRID Integration Workshop*, pages 333 – 344, Krakow, Poland, 2006. ISBN: 83-915141-6-1.