Supporting SLA Negotiation for Grid-based Medical Simulation Services

Siegfried Benkner[1], Gerhard Engelbrecht[1], Stuart E. Middleton[2], and Mike Surridge[2]

[1] Institute of Scientific Computing, University of Vienna, Austria,
[2] IT Innovation Centre, University of Southampton, UK

Extended Abstract

1 Introduction

The EU Project GEMSS has developed a service-oriented Grid infrastructure that supports the provision of advanced medical simulation services over the Internet. In order to enable the use of Grid services in a clinical environment, predictability of service response times is of paramount importance. To address this issue, we have developed a flexible Quality of Service (QoS) infrastructure for providing explicit response time guarantees for simulation services which are executed remotely on some Grid host. The GEMSS Grid infrastructure adopts a reservation based approach to QoS coupled with application specific performance models, advance reservation mechanisms, and client-driven negotiation of service level agreements (SLAs).

The GEMSS Grid infrastructure has been designed to work in a commercial environment where clients want to be able to choose from several service providers before agreeing to book a specific resource. As a consequence, GEMSS supports flexible pricing models for individual services, and a macroscopic negotiation, based on a FIPA reverse English auction protocol [4], where a client can choose the best offer from a set of competing service providers. A corresponding SLA contract is signed and exchanged to commit both parties before job execution occurs. Besides explicitly negotiable QoS guarantees, the GEMSS infrastructure provides implicit QoS by realizing highest security levels and providing support for error recovery.

Using the GEMSS Grid middleware, six Grid-enabled medical simulation applications [3] have been developed, which can be characterized by a relatively small numbers of time consuming jobs requiring powerful parallel computers in order to meet the tight time-constraints usually required during clinical procedures.

This extended abstract presents an overview of the GEMSS Grid infrastructure, implemented and tested at sites in several EU countries, and outlines the main issues involved in the negotiation of service level agreements between competing service providers.

2 GEMSS Grid Infrastructure

The GEMSS Grid architecture adopts a service-oriented approach based on standard Web services. GEMSS services are defined via WSDL and securely accessed using SOAP messages. Supported in this architecture are a number of security mechanisms such as PKI, HTTPS, WS Security and an end-to-end security protocol for separate encryption of sensitive portions of the transferred data.

A GEMSS Grid comprises multiple Grid clients and Grid service providers, one or more service registries and a certificate authority. Client applications utilize components from the GEMSS client infrastructure, which is based on a pluggable component framework, to perform service discovery, QoS negotiation and job submission and handle the creation of service input data and visualization of service output data.

The GEMSS service provision environment enables service providers to expose simulation applications available on clusters or other HPC hardware as QoS-enabled services that can be accessed on demand over the Internet. All GEMSS services provide support for quality of service negotiation, job execution, job monitoring, and error recovery.

When a Grid job is required to be run, the client usually initiates negotiations with a set of service providers. The quality of service negotiation is then run to request bids from all interested service providers who can run the clients job; this usually results in a QoS contract being agreed with a single service provider. The client then uploads the job input data to the service provider, starts the job, monitors its progress, and finally downloads the results.

3 SLA Negotiation

In GEMSS we use simple agent technology to enable the client to negotiate the best QoS agreement from a set of service providers. Our macroscopic negotiation is thus between a single client and many service providers. The GEMSS service provider infrastructure employs a QoS manager to perform microscopic negotiation and work out the best reservation possible to offer to the client. QoS guarantees are expressed in the form of a service level agreement following the Web Service Level Aggreement specification (WSLA) [**?**], which define the agreed constraints for individual jobs in the form of SLA parameters. Due to the time-critical requirements of the GEMSS applications, services are usually configured to support WSLA parameters for specifying the exact begin and end time of a job. Price may also be defined as an SLA parameter, allowing a client to see the cost demanded by a service provider.

3.1 Micro QoS Negotiation

The micro QoS infrastructure is centered on the QoS manager which provides a high level interface for QoS negotiation to clients and utilizes a compute resource manager, an application performance model and a chosen pricing model.

The performance model is used to compute the estimated run time and other performance relevant data for a service request. It takes as input a request descriptor and a machine descriptor and returns a performance descriptor. The request descriptor, supplied by the client during QoS negotiation, contains application specific meta-data about a specific service request. For example, in the case of an image reconstruction service, request parameters typically include image size and required accuracy. The machine descriptor, supplied by the service provider, specifies the resources that could be offered for an application service. The performance descriptor returned by the performance model usually contains the estimated execution time and other parameters like number of processors used to execute a job, required memory, and required disk space. In the case of a parallel MPI job, the performance model, which is usually parameterized with the number of processors, may be executed repeatedly until the time constraints set by the client are met.

The compute resource manager provides an interface to the scheduler for obtaining information about the actual availability of computing resources and is utilized by the QoS manager in order to check and create temporary reservations during QoS negotiation. In this context the compute resource manager generates a resource descriptor containing details about temporarily reserved resources, which is used by the prcing model to determine the price for a service request.

The QoS manager relies on heuristics that consider the outcome of the performance model, the availability of resources, and the pricing model to decide whether the clients QoS constraints can be fulfilled. The QoS manager returns a corresponding QoS offer to the client and performs an advance reservation of the required computing resources via the compute resource manager. Once a QoS contract is established by the QoS manager, the resource manager performs advance reservation of the required computing resources. Currently a GEMSS resource manager is available for two scheduling systems which provide support for advance reservation, the Maui scheduler [6] and COSY [2].

3.2 Macro QoS negotiation

The macroscopic QoS negotiation between a client and multiple service providers is based on the FIPA reverse English auction protocol [4]. The client first discovers a set of available services by querying a GEMSS registry. Once a set of suitable service providers is discovered they are each invited to join the auction protocol. The client application specifies the auction criteria within a QoS request, in our case the acceptable min/max values for both the start and end time of the job and the acceptable price range, which is sent to each service provider via a call for proposals message. Each service provider then starts a micro QoS negotiation and comes back with a temporary reservation encoded in a WSLA document. These are collected by the client when the proposal deadline is reached and scored using a simple dot-product scoring algorithm based on the vector of QoS values. Several rounds of bidding can occur but ultimately a single service providers WSLA is accepted and the client moves to the job execution phase.

Clients and service providers employ a relatively low level of trust in the auction. A service provider only makes a temporary reservation that will expire if the client takes too long to make a decision. Likewise service providers will be dropped from the auction if they fail to make a bid in time. Before any medical input data, which will be sensitive in nature, is sent to a winning service provider the WSLAs will be signed and exchanged. We also expect that the client will only deal with service providers that have signed a written legal contract to enable the processing of the client's medical data.

4 Conclusions

We evaluated the GEMSS QoS infrastructure by conducting a series of mirconegotiation tests for a 3D medical image reconstruction service. The macro-negotiation infrastructure was used to run these tests between a client application in the UK and three service providers in Austria, each using a different pricing model. The micro-negotiation tests were performed on different clusters running the MAUI scheduler. The results of this evaluation provide support to our view that the GEMSS Grid provides guarantees to clients regarding quality of service within a realistic economic model, and the basic legal and security framework needed to provide a platform for future exploitation. In the final paper the image reconstruction service, the associated performance model and micro negotiation evaluation will be presented in more detail.

References

1. S. Benkner, G. Berti, G. Engelbrecht, J. Fingberg, G. Kohring, S. E. Middleton, R. Schmidt. *"GEMSS: Grid-infrastructure for Medical Service Provision"*, Methods of Information in Medicine, Vol. 44/4, 2005.
2. J. Cao, F. Zimmermann. *"Queue Scheduling and Advance Reservations with COSY"*, Proceedings of the International Parallel and Distributed Processing Symposium, Santa Fe, New Mexico, 2004
3. D. M. Jones, J. W. Fenner, G. Berti, F. Kruggel, R. A. Mehrem, W. Backfrieder, R. Moore, A. Geltmeier. *"The GEMSS Grid: An evolving HPC Environment for Medical Applications"*, HealthGrid 2004, Clermont-Ferrand, France.
4. FIPA, The Foundation of Intelligent Physical Agents, IEEE FIPA Standards Committee, http://www.fipa.org/
5. The GEMSS Project: Grid-Enabled Medical Simulation Services, EU IST Project, IST-2001-37153, http://www.gemss.de/
   S. Middleton et al: GEMSS: Privacy and security for a Medical Grid. In Proceedings of HealthGrid 2004)
6. Maui Cluster Scheduler. http://www.clusterresources.com/products/maui/
7. A. Roy, V. Sander. Advance Reservation API, GGF Scheduling Working Group, 2002. http://www.ggf.org/documents/GFD/GFD-E.5.pdf
8. Web Service Level Agreement (WSLA) Language Specification. http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf, IBM 2003.