A Grid Service for Management of Multiple HLA Federate Processes

Katarzyna Rycerz[1], Marian Bubak[1,2], Maciej Malawski[1], Peter M.A. Sloot[3]

[1]Institute of Computer Science, AGH, al. Mickiewicza 30,30-059 Kraków, Poland
[2]Academic Computer Centre – CYFRONET, Nawojki 11,30-950 Kraków, Poland
[3]Faculty of Sciences, Section Computational Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{kzajac |bubak|malawski}@uci.agh.edu.pl, sloot@science.uva.nl
phone: (+48 12) 617 39 64, *fax:* (+48 12) 633 80 54

Abstract. The subject of this paper is a Grid management service called *HLA–Speaking Service* that interfaces an actual High Level Architecture (HLA) application with the Grid HLA Management System (GHMS). *HLA–Speaking Service* is responsible for execution of an application code on the site it resides and manages multiple federate processes. The design of the architecture is based on the OGSA concept that allows for modularity and compatibility with Grid Services already being developed. We present the functionality of the Service with an example of N–body simulation of dense stellar system.

Keywords HLA, Grid, distributed interactive simulations, federate management, N-body simulation

1 Introduction

Distributed simulations often require extensive computing resources. The Grid [2, 3] is a promising means of solving this problem as it offers the possibility to use resources which are not centrally controlled and are under different administrative policies. Simulations built with an implementation of the High Level Architecture (HLA) system [6] allow for merging geographically-distributed parts (called *federates*) of simulations (called *federations*) into a coherent entity. The High Level Architecture is explicitly designed as support for interactive distributed simulations, it provides various services required for that specific purpose, such as time management, useful for time-driven or event-driven interactive simulations. It also takes care of data distribution management and enables all application components to see the entire application data space in an efficient way. On the other hand, the HLA standard does not provide automatic setup of HLA distributed applications and there is no mechanism for migrating federates according to the dynamic changes of host loads or failures, which is essential for Grid applications. Therefore, there is a need for a system that would manage HLA-based simulations on the Grid. The Grid Services concept provides a good starting point for building the Grid HLA Management System (GHMS) for that purpose, as described in [12]. In this paper we focus on the HLA management service called *HLA–Speaking Service* that interfaces actual HLA application with GHMS and is responsible for execution of application code on the site it resides. The service manages multiple federate processes and is based on experience obtained from its prototype for single processes[15]. We also show how the distributed N–body simulation of a dense stellar system can benefit from the Grid by using this service. Such simulations remain a great challenge in astrophysics [17] and there is a need for a computer infrastructure that will significantly improve their performance. We believe that the Grid is a promising environment for such requirements, since it offers the possibility of accessing computational resources that have heretofore been inaccessible.

The paper is organized as follows: in Section 2 related works are presented, in Section 3 we describe the *HLA–Speaking Service*. Section 4 presents the application for the N–body simulation and shows how this application is run with GHMS. In Section 5 we present the migration overhead and improvement of performance of N-body simulation resulting from GHMS, and we conclude the paper in Section 6.

2 Related Works

Execution and management of HLA–based simulation on the Grid is a very interesting and important problem. [1] describes design and implementation of a load management system for running large-scale HLA simulations in a Grid environment based on Globus Toolkit 2. These authors also present a framework where the user can design parallel and distributed simulations without prior knowledge of HLA/RTI [13]. In [14] they presented a protocol that supports efficient checkpointing and federate migration for dynamic load balancing. In order to facilitate the usage of this protocol, an additional layer between HLA and the actual application was proposed[13]. Although this approach presents a more efficient migration protocol, it is not sufficient for porting HLA legacy code to the Grid. In [16] a framework for executing large-scale distributed simulations using Grid services was presented. Currently, this framework addresses dynamic discovery of HLA federates. The approach assumes that each federate is a simulation model encapsulated in a Grid Service. These solutions are complementary to the approach presented in this paper. In [16] there is an assumption that a user is able to build a simulation from existing Grid Services (which encapsulate functionality of HLA federates) or can wrap federates into Grid Services (the paper does not, however, describe how to wrap a federate into a Grid Service, which is a nontrivial issue). This paper describes a *HLA–Speaking* Grid Service that allows a user to simply load legacy federates and manage execution of the simulation. The useability of these approaches depends on simulation requirements.

3 HLA–Speaking Service

The role of HLA–Speaking service in GHMS is illustrated in Fig.1. The group

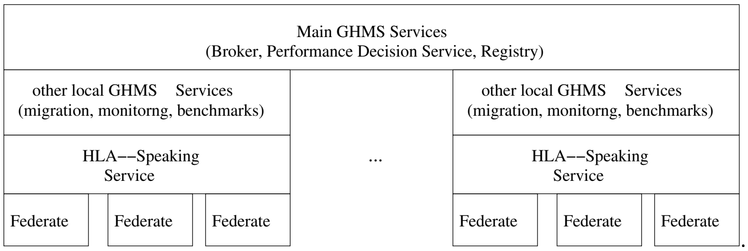
Main GHMS Services
(Broker, Performance Decision Service, Registry)
other local GHMS Services
(migration, monitorng, benchmarks)
other local GHMS Services
(migration, monitorng, benchmarks)
HLA--Speaking
Service
...
HLA--Speaking
Service
Federate
Federate
Federate
Federate
Federate
Federate

Fig. 1. The *HLA–Speaking Service* in the context of GHMS architecture

of main GHMS services consists of a *Broker Service* which coordinates management of the simulation, a *Performance Decision Service* which decides when the performance of any of the federates is not satisfactory and therefore migration is required, and a *Registry Service* which stores information about the location of local services. On each Grid site supporting HLA there are local services for performing migration commands on behalf of the *Broker Service* as well as for monitoring of federates and benchmarking. The *HLA–Speaking Service* is one of the local services interfacing federates running on its site to the GHMS system.

3.1 Multiple processes support

The *HLA Speaking Service* encapsulates various HLA implementations that are thought of as resources found in a Grid environment. The service is one of the most important parts of GHMS, as it manages the execution of the federates on a particular site. The service starts the execution of a federate code on its site, saves the application state and restores it from a checkpoint file. An HLA-specific API is hidden in the RTI library functionality and is not exposed as a service interface. Experience gained with *HLA Speaking Service* for single processes[15] has enabled us to design support for multiple federates (see Fig. 2). In this approach, we designed the control federation for forwarding requests from the Grid Service layer to the actual user federates. The Grid Service layer can request saving and restoring specific processes. The user code communicates with the control federate by means of a Grid HLA Controler Library (GHCL). The user federates communicate with each other also using RTI, but this takes place in the scope of user-defined federations.

The operations of the *HLA–Speaking Service* PortType are modified to support multiple federates and use the Resource Specification Language (RSL) from Globus GRAM [4] for description of federate processes to be run and include functions for running, saving and restoring of HLA federates.

3.2 Migration scenario

The scenario of migration is shown in Fig 3. If there is a need to migrate, the *GHMS* informs the *Broker Service* which decides that the HLA federate from

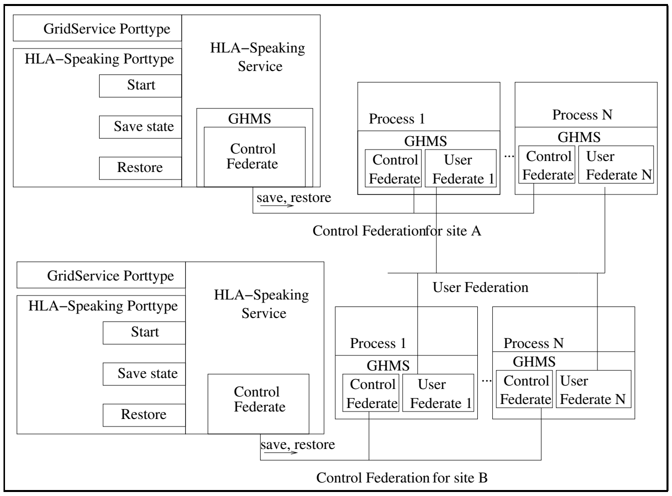

Fig. 2. Conceptual view of *HLA Speaking Service* for multiple federate processes

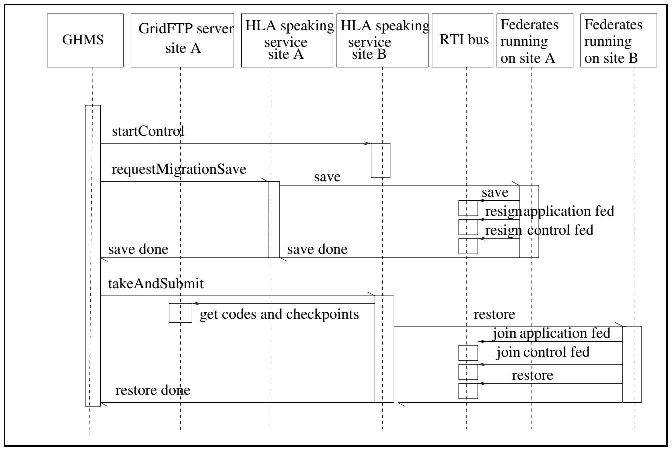

Fig. 3. Sequence diagram describing federate migration for multiple federates

site A has to be migrated to site B and triggers the *Migration Service* to perform this migration. The *Migration Service* creates the *HLA Speaking Service* on the new site, then asks for creation of a control federate that creates the control federation. When this is done, the *GHMS* asks the *HLA–Speaking Service* on site A to save the state. The *HLA Speaking Service* forwards this request with the identifiers of federates that have to be saved to all federates it controls by means of GHCL. The federates invoke an HLA API for federation saving [6] which suspends the other federates. The federates know (by their identifiers) if there is a request to save the state of their federations and resign from them Subsequently, user data is saved in a checkpoint file and the appropriate federates resign from the federations defined by the user (application federations) as well as from the control federation. Next, the *GHMS* invokes the `takeAndSubmit` operation of the *HLA-Speaking Service* on site B which triggers it to fetch the user codes and checkpoint files from site A to site B and restore federates from the available checkpoint files. The federates join the control federation as well as user-defined (application) federations and invoke the GHCL library which restores user–specific data and uses the HLA API [6] for restoring the application federation RTI internal state.

4 Feasibility study with N—body simulation

The N-body simulation is part of the virtual laboratory called Gravitylab [7] which is a powerful software environment for experiments concerning the dynamics of dense stellar systems. The algorithm of the simulation uses a direct method, where the gravitational force acting on a particle is computed by summing up the contributions from all other particles according to the Newton's law

$$\boldsymbol{F_i} = m_i * \boldsymbol{a_i} = -Gm_i \sum_{j=1,j\neq i}^{j=N} \frac{m_j(\boldsymbol{r_i} - \boldsymbol{r_j})}{|\boldsymbol{r_i} - \boldsymbol{r_j}|^3}$$

The numerical method is based on the Hermite [8] scheme, where higher-order derivatives are explicitly computed in order to construct interpolation polynomials of the force. The performance analysis of different parallelization schemes for direct code used in the simulation of astrophysical stellar systems for different computer architectures, including the Grid, can be found in [5].

The application consists of two modules: the MPI N–body simulation module and the integrated visualization–interaction module. The modules communicate using the HLA RTI bus which connects visualization/interaction federate with the MPI root process of the simulation federate. Those federates form the application federation. Additionally, HLA is used for control of MPI simulation processes on the Grid site as described in Section 3.1. Therefore, all MPI processes join the control federation together with the control federate in the HLA–speaking service. For connection with the control federate, all processes use the GridHLAController library.

5 Results

We have performed two different experiments to show the overhead of various migration stages and its impact on simulation performance. The experiments were performed on the DutchGrid DAS2 testbed infrastructure and at CYFRONET, as shown in Table 1. Because the bandwidth available for testing was broad (10Gbps), communication did not play an important role and calculations were the most time–consuming part of the execution. In order to create conditions in which migration would be useful, we increased the load of the Grid site where the simulation was executed (Amsterdam) by submitting nonrelated, computationally–intensive jobs. Next, we imitated a Resource Broker and migrated the simulation to another site which was not overloaded (Delft). The experiments were performed at night in order to avoid interference with other users and repeated 10 times to check if they are reproducible.

Operating System	RedHat Enterprise Linux Advanced Server, version 3		
Network	10 Gbps (DAS2) + 155 Mbps (DAS2-Cyfronet)		
Role	**Name**	**CPU**	**RAM**
Migration source	DAS2 Nikhef	Pentium III 1 GHz	1 GB
Migration dest	DAS2 Delft	Pentium III 2GHz	2 GB
Interation federate	Cyfronet Krakow	2.40 GHz Xeon	1 GB
RTIexec	Cyfronet Krakow	2.40 GHz Xeon	1 GB

Table 1. Grid testbed infrastructure.

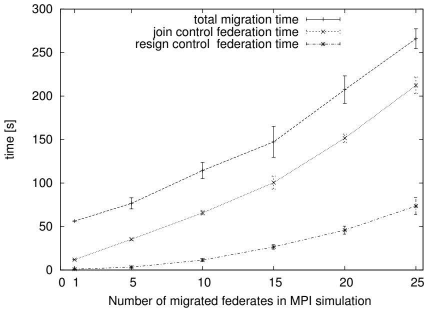

Fig. 4. The time of migration stages dependent on MPI federate number

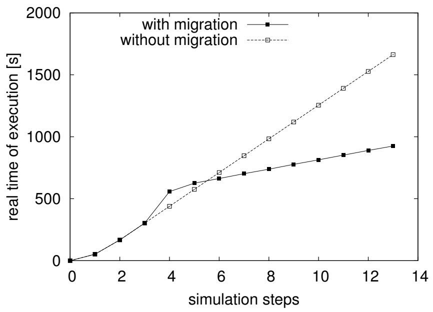

Fig. 5. Impact of migration on performance of N-body simulation

Migration time. Fig.4 shows migration time as a function of the number of MPI processes in the federation. For our tests, we used GT v3.2 and HLA RTI 1.3v5. The number of stars used in the simulation was 5000. The results were obtained as an average of 10 measurements. The error bars indicate estimated standard deviation denoted by σ. The time includes phases as shown in Fig.3 in Section3.2. The overhead of particular phases is shown in Fig.4 (activities dependent on the number of MPI processes) and Tab.2 (activities independent of the number of MPI processes). As can be observed, the largest overhead in the case of the tested application is for rejoining the control federation. This is due to the fact that in the RTI implementation we used, every federate has to open a TCP connection to every other federate in the control federation. Reliable multicast [9] would significantly decrease this overhead. Other activities, such as saving and restoring the state of the federation application, joining and resigning

	$\bar{x}$ [s]	σ [s]
save time	6.4	1.3
application resign time	3.7	0.5
transfer time	0.4	0.05
GRAM	2.5	0.1
PBS waiting time	8.5	0.7
application join time	12.9	1.9
restore time	8.2	0.8

Table 2. Execution time of various migration stages

the federation application, transferring files (GridFTP overhead), GRAM and PBS induce overhead which does not depend on the number of MPI federates since only one root MPI process is actually joining the application federation in order to communicate with the visualization federate, as shown in Tab.2.

Impact of migration on performance As in the previous experiment, for our tests we used GT v3.2 and HLA RTI 1.3v5. The simulation consisted of 20 MPI processes calculating 3D position and velocity of 24000 stars. Fig 5 shows the time as a function of number of interactive steps in the loop with a human (for the first 13 steps) In each step, the simulation calculates data and sends them to the visualization and interaction module using HLA. The dashed line shows what would happen if the network stayed busy for some time and the responses took longer. In this situation it is better to spend some time on migration to the other site, which offers better response times as shown by a solid line in the figure. Fig. 5 shows that the user can gain access time between steps 5 and 6 independently of the time lost on migration(performed between steps 3 and 4).

6 Summary and Conclusions

This paper presented the *HLA–Speaking service* which enables efficient management of the execution of HLA code on the Grid. The service is a part of the GHMS system described in [12] and it is used for running, saving and restoring one or more federate processes on the Grid site on which it resides. We have also presented a sample application managed by our service – N–body simulation of a dense stellar system. Our solution gives the ability to not only apply the HLA standard on the Grid, but also to easily adapt legacy HLA applications to the Grid environment in a robust and efficient way.

Acknowledgments The authors would like to thank Piotr Nowakowski for useful remarks. This research is partly funded by the EU IST Project CoreGRID, the Polish State Committee for Scientific Research SPUB-M grant, and by the Dutch Virtual Laboratory for e-Science project (www.vl-e.nl).

References

1. W. Cai, S.J. Turner, and H. Zhao. A Load Management System for Running HLA-based Distributed Simulations over the Grid. In S.J. Turner and S.J.E. Tay-

lor, editors, *Proceedings of the sixth IEEE International Workshop on Distributed Simulation and Real Time Applications (DS-RT 2002)*, pages 7–14, Fort Worth, Texas, October 2002. IEEE Computer Society.

2. I. Foster. What is the Grid? A three checkpoints list. *GridToday Daily News And Information For The Global Grid Community*, 1(6), July 2002.
3. I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Open Grid Service Infrastructure WG, Global Grid Forum*, June 2002. `http://www.globus.org/research/papers.html`.
4. Globus Project Home Page. `http://www.globus.org/`.
5. A. Gualandris, S.P. Zwart, and A. Tirado-Ramos. Performance analysis of direct N-body algorithms on highly distributed systems. *IEEE Transactions on Parallel and Distributed Systems.* Submitted, available at `http://arxiv.org/abs/astro-ph/0412206`.
6. HLA specification. `http://www.sisostds.org/stdsdev/hla/`.
7. P. Hut and J. Makino. The art of computational science. `www.artcompsci.org/`.
8. J. Makino and S.J. Aarseth. On a Hermite integrator with Ahmad-Cohen scheme for gravitational many-body problems. *PASJ*, 44:141–151, April 1992.
9. D.M. Moen, J.M. Pullen, and F. Zhao. Implementation of host-based overlay multicast to support of web based services for rt-dvs. In *8th IEEE International Workshop on Distributed Simulation and Real-Time Applications (DS-RT 2004), 21-23 October 2004, Budapest, Hungary*, pages 4–11. IEEE Computer Society, 2004.
10. MPI Forum Home Page. `http://www.mpi-forum.org`.
11. RSL – Globus Resource Specification Language v1.0. `http://www.globus.org/gram/rsl_spec1.html`.
12. K. Rycerz, M. Bubak, M. Malawski, and P. Sloot. A Framework for HLA-Based Interactive Simulations on the Grid. *SIMULATION*, 81(1):67–76, 2005.
13. Z. Yuan, W. Cai, and M.Y.H. Low. A Framework for Executing Parallel Simulation using RTI. In S.J. Turner and S.J.E. Taylor, editors, *Proceedings Seventh IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2003)*, pages 20–28, Delft, The Netherlands, October 2003, IEEE.
14. Z. Yuan, W. Cai, M.Y.H. Low, and S. J. Turner. Federate Migration in HLA-Based Simulation. In M. Bubak, G. D. van Albada, P. M.A. Sloot, and J. J. Dongarra, editors, *Computational Science - ICCS 2004*, volume 3038 of *Lecture Notes in Computer Science*, pages 856–864, Krakow, Poland, June 2004. Springer-Verlag.
15. K. Zając, M. Bubak, M. Malawski, and P.M.A Sloot. Towards a Grid Management System for HLA-Based Interactive Simulations. In S.J. Turner and S.J.E. Taylor, editor, *Proceedings Seventh IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2003)*, pages 4–11, Delft, The Netherlands, October 2003. IEEE Computer Society.
16. W. Zong, Y. Wang, W. Cai, and S.J. Turner. Grid Services and Service Discovery for HLA-Based Distributed Simulation. In *8th IEEE International Workshop on Distributed Simulation and Real-Time Applications (DS-RT 2004), 21-23 October 2004, Budapest, Hungary*, pages 116–124. IEEE Computer Society, 2004.
17. S.P. Zwart, H. Baumgardt, P. Hut, J. Makino, and S. L. W. Mcmillan. Formation of massive black holes through runaway collisions in dense young star clusters. *Nature*, 428:724–726, April 2004.