Large Scale Distributed Simulation on the Grid

Georgios Theodoropoulos, Yi Zhang, Dan Chen, Rob Minson
University of Birmingham
Edgbaston, Birmingham, B15 2TT, UK
{gkt,yxz,cxd,rzm}@cs.bham.ac.uk

Stephen J. Turner, Wentong Cai
School of Computer Engineering
Nanyang Technological University
Singapore 639798
{ASSJTurner,aswtcai}@ntu.edu.sg

Yong Xie
Computing Laboratory
Oxford University
Oxford, OX1 3QD, UK
Yong.Xie@comlab.ox.ac.uk

Brian Logan
School of Computer Science & Information Technology
University of Nottingham
Nottingham NG8 1BB, UK
bsl@cs.nott.ac.uk

Abstract

The development of many complex simulation applications requires collaborative effort from researchers with different domain knowledge and expertise, possibly at different locations. These simulation systems often require huge computing resources and data sets, which may be geographically distributed. In order to support collaborative model development and to cater for the increasing complexity of such systems, it is necessary to harness distributed resources over the Internet. The emergence of Grid technologies provide exciting new opportunities for large scale distributed simulation, enabling collaboration and the use of distributed computing resources, while also facilitating access to geographically distributed data sets. This paper discusses the research challenges that must be addressed before these opportunities can be exploited and presents HLA_GRID_REPAST, *a system for executing large scale distributed simulations of agent based systems over the Grid.*

1. Introduction

In the Spring of 2003, the UK e-Science Core Programme issued a call for proposals to establish synergistic links between UK e-Science projects and leading Grid projects from around the globe. One of only four such projects is DS-Grid[1], a collaboration between the Midlands e-Science Centre (MeSC[2]) at the School of Computer Science, University of Birmingham and the Parallel and Distributed Computing Centre (PDCC[3]) at the School of Computer Engineering, Nanyang Technological University, Singapore, with participation of the School of Computer Science & Information Technology, University of Nottingham.

DS-Grid is concerned with Large Scale Distributed Simulation on the Grid. The vision of the project is a "Grid plug-and-play distributed simulation system", a distributed collaborative simulation environment where researchers with different domain knowledge and expertise, at different locations, develop, modify, assemble and execute distributed simulation components over the Grid. The simulation components reside in the Grid as Grid services which the system will automatically discover, semantically match (with each other and with the simulation goals), and return to the user. A number of important new research challenges have to be addressed before this vision is realised; addressing these challenges is the long term objective of the DS-Grid collaboration.

Building on existing complementary efforts in the UK

[1] "DS-Grid: Large Scale Distributed Simulation on the Grid", e-Science Sister Project GR/S82862/01. URL: `http://www.cs.bham.ac.uk/research/projects/dsgrid`

[2] `http://www.mesc.bham.ac.uk`

[3] `http://pdcc.ntu.edu.sg/`

and Singapore, most notably the work on service oriented middleware for HLA simulations on the Grid in Singapore (the HLA_GRID system) and the work on HLA distributed simulations of agent-based systems in Birmingham (the HLA_REPAST system [6]), the project has developed HLA_GRID_REPAST, a prototype platform for executing HLA agent-based simulation on the Grid.

In [13] we provided an overview of the HLA_GRID_REPAST system and presented some preliminary performance results. This paper discusses the research challenges in developing and executing large scale distributed simulations on the Grid, and presents a more in depth description of the HLA_GRID_REPAST system. The rest of the paper is organised as follows: Section 2 outlines the key research challenges in Grid-aware distributed simulation. Section 3 discusses HLA_GRID_REPAST, providing an overview of the HLA_GRID and HLA_REPAST systems. Section 4 concludes the paper.

2 Research Challenges in Large Scale Simulation on the Grid

Modelling and simulation is an essential tool in many areas of science and engineering, for example, for analysing natural phenomena or predicting the behaviour of new systems being designed. The development of such complex simulation applications usually requires collaborative effort from researchers with different domain knowledge and expertise, possibly at different locations. Furthermore, these simulation systems often require huge computing resources and the data sets required by the simulation may also be geographically distributed (e.g. in a supply chain simulation involving different companies, the most up to date data will be in the individual companies). In order to support collaborative model development and to cater for the increasing complexity of such systems, it is necessary to harness distributed resources over the Internet. The last decade has witnessed an explosion of interest and innovation in the field of large scale distributed simulation. This activity is mainly centred on the High Level Architecture (HLA), an IEEE standard to facilitate interoperability among simulations and promote reuse of simulation models.

The emergence of Grid technologies provide exciting new opportunities for large scale distributed simulation, enabling collaboration and the use of distributed computing resources, while also facilitating access to geographically distributed data sets. In the last few years, there has been an increasing interest in taking advantage of Grid technologies to execute HLA simulations over the Internet.

A seminal work in this direction is the Extensible Modeling & Simulation Framework[4] (XMSF), a collaborative initiative to develop a web enabled RTI [9]. Within the XMSF framework, multiple federates reside as web services on a WAN and the Federation's FOM is mapped to an XML tagset, allowing interoperation with other distributed applications supported by web services. The federates communicate using the Simple Object Access Protocol (SOAP) and the Blocks Extensible Exchange Protocol (BEEP).

Another important initiative is HLA_GRID, originally developed at the Nanyang Technological University in Singapore [5, 12, 14]. In HLA_GRID, federates can be instantiated as Grid services which are used to facilitate communication between Grid service federates and the RTI.

The HLA is a mature technology which offers enormous potential for large scale distributed simulation on the Grid. However, a number of key research challenges must be overcome before this potential can be realised. The following sections discuss some of these challenges.

2.1 Model Discovery and Semantic Matching

While the FOM and RTI provide interoperability at the communication level, there is little support for interoperability at the semantic level; as a result, simulation development lead times are often on the order of months, since the interpretation of data by each federate must be fully understood and carefully checked for consistency by the simulation developer. Thus, there is a clear need for automated federate discovery and configuration in a Grid environment.

The configuration and reuse of software components is a key problem in several fields including agent-based systems, software engineering and Web services [11]. Of these, the agents community has made the most progress in automating component configuration, but work tends to be domain specific and there are no well established standards and languages. Component-based software engineering has made substantial progress in modelling components (e.g. CORBA, .NET and Java Beans) and supporting the human designer. Components are usually well defined at the syntactic level (using languages like UML), but semantic annotations are very limited.

Fully automated configuration is limited to specific application domains where simple forms of component retrieval are also supported (e.g. exact-matching). Automatic configuration and adaptation of software components is currently the focus of research in autonomic computing (e.g. self managing and self healing systems). Web and Grid services support scalable inter-operation of heterogeneous software across a wide variety of networked platforms and a rapid integration cycle and offer the possibility of automatic composition of components obtained by querying web service registries. The discovery and automatic composition of web services has been addressed through both

[4] http://www.movesinstitute.org/xmsf/xmsf.html

syntactic and semantic frameworks. Syntactic frameworks model services as Business Processes without semantic annotations and use standardised technologies for description (e.g. WSDL, SOAP, UDDI) and (manual) configuration of services (e.g. the orchestration languages XLANG, WFSL, BPEL4WS). The component structure is typically process based and is described as a workflow which consists of activities with data and control flow. Semantic web technologies extend these frameworks with machine understandable conceptual descriptions of web service "capabilities" (semantic annotations) to support increased automation of all activities within the web services model.

For automated federate discovery and configuration, appropriate ontologies and languages must be defined to represent the metadata of federates and orchestrate simulations by matching user requirements with appropriate federates. Each federate can manifest itself as a Grid service for use by the simulator. The ontologies should represent descriptions, classifications of descriptions, and constraints related to valid configurations of simulations and should support both searching for simulation models and semantic matching of component models with simulation goals. To achieve this, ontologies for the description of the internal behaviour of federates are required, as well as appropriate reasoning and matchmaking engines. Although there have been efforts to describe the inner mechanisms of composite or stateful services, such as federates, so far these have been limited [11]. XMSF advocates the use of MDA for meaningful inter-operation of federates.

2.2 Resource Management

A large-scale simulation executing in a distributed environment may need computing resources from many different organisations and the availability of these computing resources may change during the execution of the simulation. To meet the real-time requirements demanded by interactive simulations and the performance requirements demanded by analytical simulations, sophisticated resource management mechanisms are required.

HLA does not provide any support for resource management and dynamic balancing of simulation load and mechanisms for resource management, load monitoring, dynamic load balancing, check-pointing and migration in a Grid environment are therefore required.

Load balancing of distributed simulations has been studied extensively. However, most existing work assumes parallel/distributed platforms with negligible interprocessor communication delays. Their feedback loop is based mainly on processor load and they tend to largely ignore communication latencies. In a Grid, however, communication delays are substantial and can be the decisive factor of the simulation performance. This issue is increasingly receiving the attention on researchers in distributed simulations (e.g. [4]).

Mechanisms for federate migration have been developed, e.g. [2]. For decision making, models are required which take into account locality of and the communication patterns between the federates for their migration in the Grid. This would also require mechanisms for monitoring and evaluating the progress of the simulation against some baseline cost models and assessing the benefits of work migration. Interaction with the various HLA services such as federation management, ownership management and time management must be considered.

The heterogeneity of resources on the Grid will require careful evaluation of trade-offs related to communication, compute capacity and appropriateness of specific HLA federates (for instance, a less desirable federate in terms of its properties may provide significantly better performance).

2.3 Support for Collaborative Model Development

To conduct simulation experiments easily over geographically distributed resources from different organisations, mechanisms that can deploy coordinated, secured simulation executions are required. The HLA does not provide any support for collaborative development of simulation components and applications and hence, new Grid-aware collaborative environments for distributed simulation must be developed. There already exists considerable interest in such Advanced Collaborative Environments in the Grid community[5].

Collaborative Environments for HLA simulations on the Grid could integrate support for capturing requirements, model discovery and matching, workflow specification and coordinated startup of federates at each site. For collaborative model development, it is also essential to create Grid services to facilitate graphic and/or text-based user interactions.

3 The HLA_Grid_RePast System

HLA_GRID_REPAST is designed for executing distributed, large scale simulations of agent-based systems over Grids. It integrates HLA_REPAST and HLA_GRID and acts as a middleware to glue simulation code written in REPAST with the HLA.

3.1 HLA_Grid

HLA_GRID is a framework designed to extend the HLA to the Grid. In particular, it focuses on improving the

[5] http://calder.ncsa.uiuc.edu/ACE-grid/

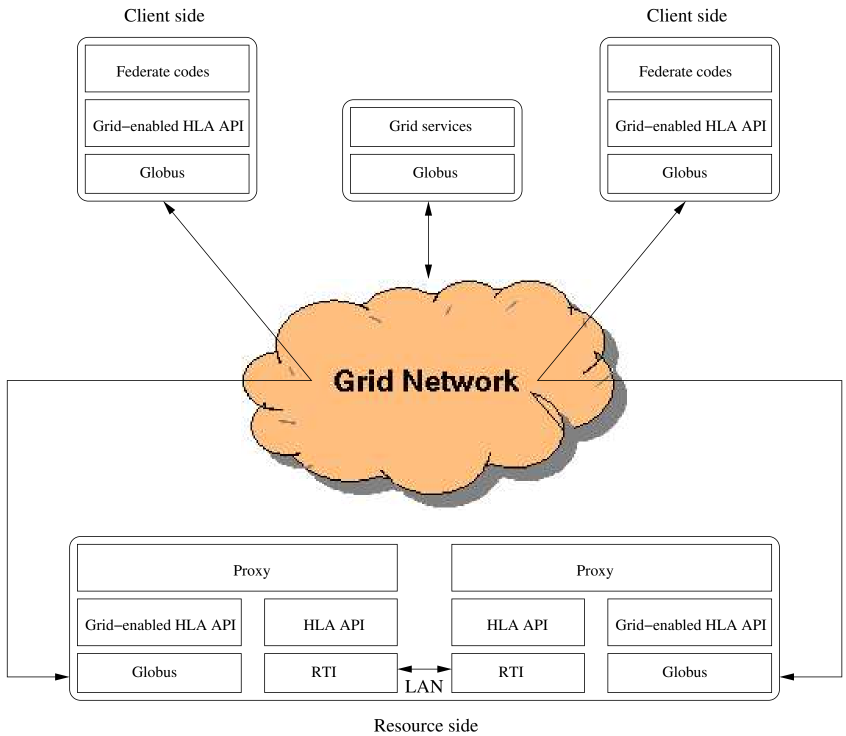

Figure 1. Architecture of Proxy-based HLA simulation on the Grid.

interoperability and composability of HLA-compliant simulation components.

The framework, which is illustrated in figure 1, achieves interoperability between different simulators (federates) by using a Federate-Proxy-RTI architecture, in which different participants (clients) in the same simulation run their federate codes at their local sites, and the RTIEXEC and FEDEXEC are executed on the remote resource. A new *proxy* entity is introduced to act on behalf of the client's federate code and communicate with the proxies of other clients through the RTI. Proxies are executed at remote grid resources. Federate codes and their respective proxies communicate with each other through Grid services and a Grid-enabled HLA library, which provides the standard HLA API to the federate codes, is implemented to translate the communications into Grid services invocations.

HLA_GRID includes additional Grid services to support the creation of the RTI, discovery of federations, etc. The framework hides the heterogeneity of the simulators, execution platforms, and how the simulators communicate with the RTI. All interfaces used in the HLA_GRID comply with the standard HLA interface specification.

A prototype of HLA_GRID has been implemented in Java using the Globus Toolkit version 3. The prototype includes standard HLA/RTI APIs to support Federation, Time, Object, Declaration and Ownership Management. The prototype has been tested using the benchmark programs in the DMSO RTI package. More details of HLA_GRID's design, implementation and experiments can be found in [5, 12, 14].

3.2 HLA_RePast

The REPAST system [3] is a Java-based toolkit for the development of lightweight agents and agent models. It was developed at the University of Chicago's Social Science Research Computing division and is derived from the Swarm simulation toolkit. It has become a popular and influential toolkit, assessed by [10] as the most effective development platform currently available for large-scale simulations of social phenomena.

The system provides an inter-dependent collection of tools and structures which are generally useful for the simulation of agents, and a sequential discrete event simulation kernel for the execution of the model.

HLA_REPAST is a middleware layer which enables the execution of a federation of multiple interacting instances of REPAST models within the HLA as depicted in figure 2.

The main task of the HLA_REPAST middleware is to detect the occurrence of events in the REPAST model and consistently, reliably and transparently communicate them to the RTI. To achieve this, HLA_REPAST provides mechanisms for mapping REPAST state-transitions to RTI events (via the `PublicObject` scheme described below), for

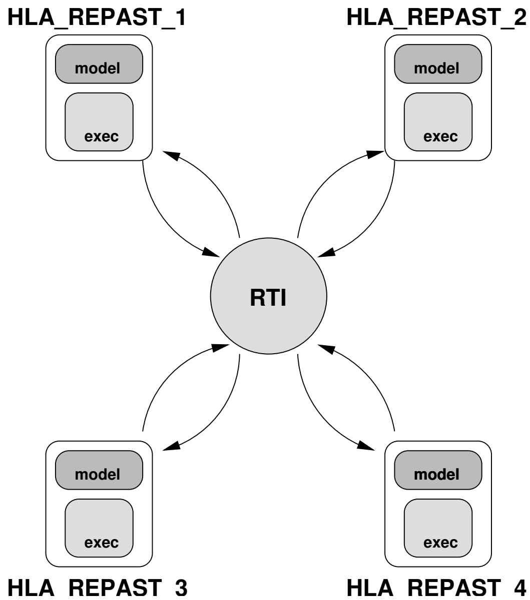

Figure 2. An HLA_REPAST Federation

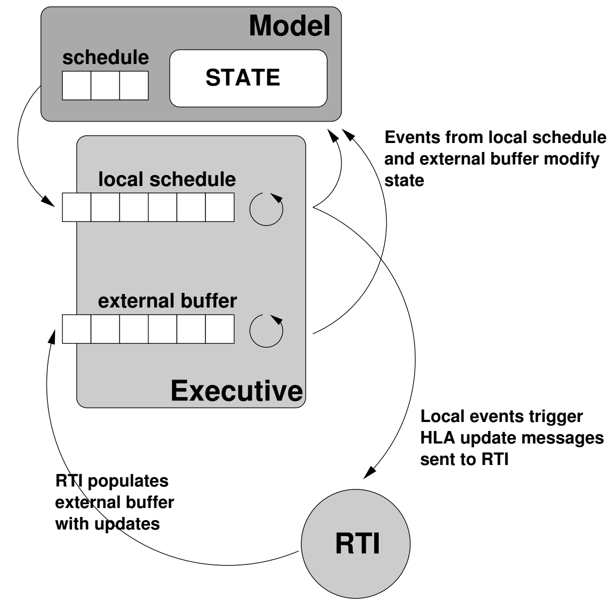

Figure 3. The HLA_REPAST Model-Executive Interface (A Single HLA_REPAST Federate)

conflict resolution and for integrating the REPAST scheduling system with the RTI (as depicted in figure 3). More information about the HLA_REPAST system can be found in [6].

3.3 Architecture of HLA_GRID_RePast

Figure 4 presents a conceptual view of the federate executive based on HLA_GRID_REPAST. Each federate executive basically consists of two major parts, one on the client side and another on the proxy side, which usually run on different machines. The client side contains the following modules: the REPAST agent-based simulation model, the HLA_REPAST middleware, the Client RTI Ambassador and the Client Federate Ambassador Service from HLA_GRID. These components usually run on a local machine (from the simulation model's point of view).

On the proxy side, there are the Proxy RTI Ambassador Service and the Proxy Federate Ambassador which interact with the real RTI hosted by a remote machine. Both the Proxy RTI Ambassador Service and the Clien Federate Ambassador Service are implemented as GRID services on different sides, shown as the round rectangle boxes in Figure 4. The modules on both sides are coupled together to form a single federate executive.

Before the simulation starts, the Proxy RTI Ambassador Services should have been started, and each federate is associated with a Proxy RTI Ambassador Service. The identity of each federate's Proxy RTI Ambassador Service will be passed to the corresponding Client RTI Ambassador. When the simulation starts, the Client Federate Ambassador Service of each federate will be initialised and registered with the corresponding Proxy RTI Ambassador Service.

During the simulation, the events and state changes generated by the simulation model will be translated into the corresponding RTI function calls by the underlying HLA_REPAST middleware. Then the RTI calls will be passed to the Client RTI Ambassador, and it will translate these function calls once again into Grid service invocations to access the remote Proxy RTI Ambassador Service on the proxy side.

Finally, the Proxy RTI Ambassador Service will interact with the real RTI by executing the real RTI calls with respect to the client side. The return values will be sent back as the return value of the GRID service call while the runtime exceptions of the RTI calls will be sent back by means of Apache Axis faults, which will be described later. On the other hand, callbacks from the RTI to a federate are translated into invocations of the Client Federate Ambassador Service by the Proxy Federate Ambassador. The Client Federate Ambassador Service will convey these callbacks to the HLA_REPAST middleware, and the latter will then convert the federate calls into REPAST events and put them into the event scheduler.

Modifications have been made to both HLA_GRID and HLA_REPAST in order to integrate them into HLA_GRID_REPAST. Two major modifications are concerned with object encoding and remote exception han-

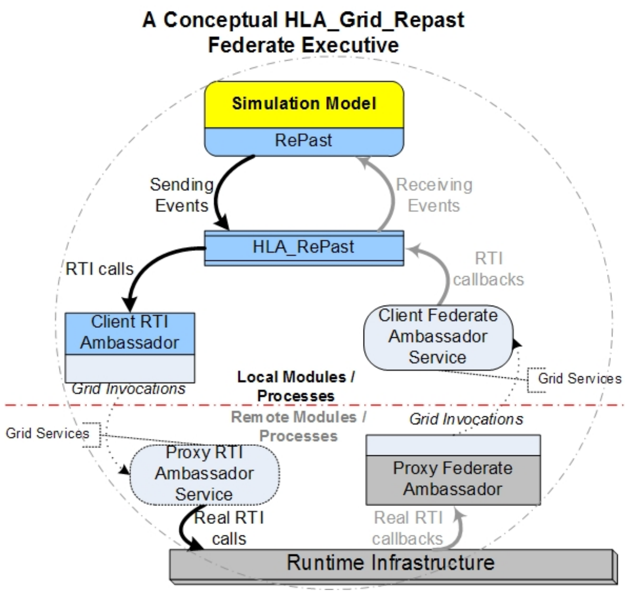

Figure 4. Structure of HLA_GRID_REPAST

dling. The former is required by the SOAP protocol while the latter is designed for improving performance.

3.3.1 Object Encoding

The object encoding scheme has been modified to cooperate with HLA_GRID. HLA_REPAST uses a *PublicObject* scheme to translate Java expressions into HLA function calls [6]. Java objects in the PublicObject scheme may be transferred between federates during a simulation. In HLA_REPAST, the Java objects are encoded into Java Byte arrays. However, Java Byte arrays cannot be used directly within HLA_GRID, which, like other grid services, uses SOAP as the communication protocol between clients and servers. SOAP is based on XML which does not support binary data. The problem can be solved by modifying HLA_GRID to support either the *SOAP With Attachments* (SwA) proposal [1] or the *WS-Attachments* proposal [7]. Such an approach would require modifications to the Globus Toolkit and would affect the interoperability of HLA_GRID_REPAST. Instead of modifying the Globus Toolkit, a new object encoding scheme for Java objects is employed in HLA_GRID_REPAST as follows. First, the Java object is encoded into a Java byte array as in HLA_REPAST. Then, the base64 encoding scheme (from the Apache project) is used to encode the Java byte array into a Java String Object. Finally, a Java Byte array is generated from the Java String and passed to HLA_GRID for encapsulation in a SOAP message.

3.3.2 Exception Handling

In addition, remote exception handling has been introduced in HLA_GRID_REPAST. In HLA_GRID_REPAST, Java exceptions generated in Grid service calls are packed into Apache Axis faults and passed back to the client side. The RTI implementation used in HLA_GRID_REPAST is the DMSO RTI NG 1.3. In this implementation, Java exceptions work as return values of RTI function calls in many situations. HLA_REPAST makes heavy use of such functions. In many cases, the exceptions (return values) are discarded immediately without further handling. This behaviour causes very small overhead when the simulation federates and the RTI ambassadors execute on the same machine. This is not the case in HLA_GRID, however, where the federate simulations and the RTI ambassadors are in different machines connected via a Grid. In this case, exceptions raised by RTI calls waste substantial network bandwidth. To address this issue, in HLA_GRID such exceptions are registered with the RTI Ambassador Services after connection between the client simulation and RTI Ambassador Service is established through the RTI Proxy Ambassador. When RTI calls generate exceptions, the RTI Ambassador Service will only pass exceptions that are not registered back to the client simulation. In this way, registered RTI call exceptions are handled remotely at the proxy side. The performance advantage of remote exception handling is substantial, and is demonstrated in the next section.

3.4 Evaluation of HLA_GRID_REPAST

To evaluate the robustness and performance of HLA_GRID_REPAST we have developed an agent-based federation using TILEWORLD [8] as a test case. Tileworld is a well established testbed for agents. It consists of a grid-like environment consisting of tiles, holes and obstacles, and on or more agents which attempt to score as many points as possible by pushing tiles to fill in the holes (figure 5). For HLA_GRID_REPAST, a TILEWORLD simulation developed for HLA_REPAST has been used, as described in [6]. In that implementation, the environment of Tileworld containing the tiles, holes and obstacles of the model is simulated by a single federate while the TILEWORLD agents are simulated by one or more agent federates. In general, the simulation will contain one environment federate and at least one agent federate.

Experimental results have been obtained by executing simulations over a Grid in a LAN environment and a WAN environment between the UK (Birmingham) and Singapore as shown in figure 6. The performance results obtained have been presented and discussed in [13] and they are briefly summarised here.

Our experiments in a LAN environment with one agent federate simulating different number of agents

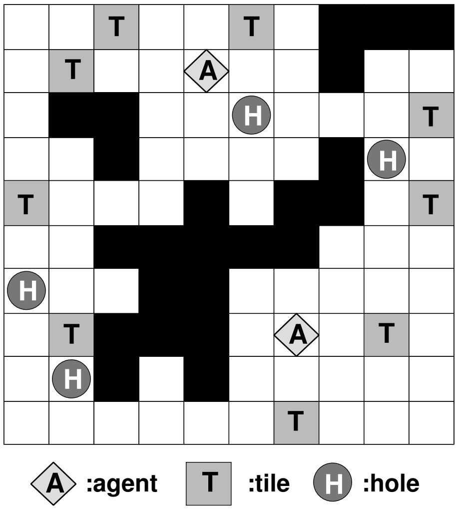

Figure 5. A 20x20 TILEWORLD

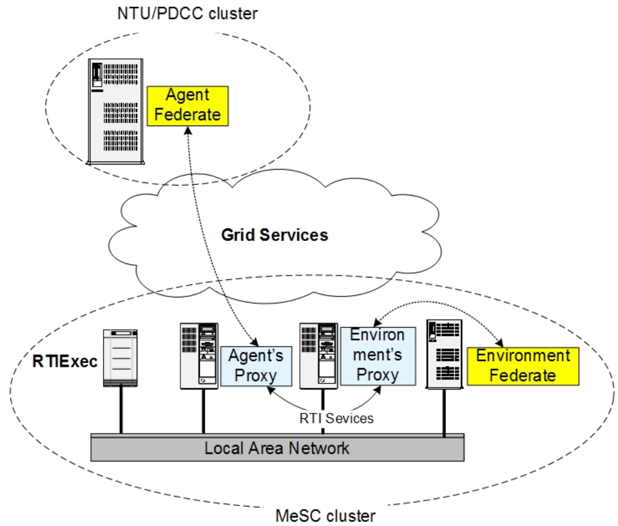

Figure 6. TILEWORLD **Configuration for** HLA_GRID_REPAST

showed that the performance of HLA_GRID_REPAST with local exception handling is lower compared with HLA_GRID_REPAST with remote exception handling. When the number of agents is small (less than 12 in our experiments), HLA_GRID_REPAST exhibits a higher overhead compared to HLA_REPAST. As the number of agents increases and computation load becomes the dominant factor, the total simulation time for HLA_GRID_REPAST is relatively close to that of HLA_REPAST. The distribution of 128 agents in multiple federates (1 to 16) in HLA_GRID_REPAST yields speedup patterns similar to HLA_REPAST with the relative overhead slightly increasing as the cross Grid communication increases (i.e. as the number of federates increases).

Experiments in a WAN environment have shown that HLA_GRID_REPAST performs much worse than HLA_REPAST compared with the performance difference in a LAN environment. Further experiments are currently being performed to further analyse the communication costs in the WAN environment.

4 Summary

The emergence of Grid technologies provide exciting new opportunities for large scale distributed simulation, enabling collaboration and the use of distributed computing resources, while also facilitating access to geographically distributed data sets. In the last few years, there has been an increasing interest in taking advantage of Grid technologies to execute HLA simulations over the Internet. Contributing to this effort, the DS-Grid e-Science project envisages the development of a distributed collaborative simulation environment where researchers with different domain knowledge and expertise, at different locations, develop, modify, assemble and execute distributed simulation components over the Grid. The required components reside in the Grid as Grid services which the system will automatically discover, semantically match (with each other and with the simulation goals) and return to the user.

However a number of important new research challenges have to be addressed before this vision is realised, including support for collaborative development of simulation applications, advanced model and service discovery mechanisms, novel resource management, mechanisms for fault-tolerant, coordinated, secured simulation executions, and load balancing mechanisms to meet the different requirements of the different models.

As a testbed to achieve DS-Grid's objectives, this paper has presented HLA_GRID_REPAST, a prototype platform for executing large scale agent-based simulations over Grids.

5 Acknowledgements

A number of people have helped in many different ways to make this work possible. In particular we would like to thank Ben Stone, Steve Pillinger, Jim Nichols, Irene Goh, Tan Joo Sing and Mike Lees. This work is funded by the UK e-Science Programme.

References

[1] J. Barton, S. Thatte, and H. F. Nielsen. SOAP messages with attachments. W3C Note, Dec. 2000.

[2] W. Cai, Z. Yuan, M. Y. H. Low, and S. Turner. Federate migration in HLA-based simulation. *Future Generation Computer Systems*, 21(1):87–95, Jan. 2005.

[3] N. Collier. RePast: An extensible framework for agent simulation. `http://repast.sourceforge.net/docs/repast\_intro\_final.doc`, June 2003.

[4] K. Iskra, G. Albada, and P. Sloot. Time Warp cancellations optimisations on high latency networks. In *Proceeding of the 7th IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'03)*, pages 128–137, 2003.

[5] W. Jie, T. Zang, Z. Lei, W. Cai, S. Turner, and L. Wang. Constructing an OGSA-based Grid computing platform. In *International Conference on Scientific and Engineering Computation (IC-SEC 2002)*, pages 738–741, 2002.

[6] R. Minson and G. Theodoropoulos. Distributing RePast agent-based simulations with HLA. In *European Simulation Interoperability Workshop 2004*, pages 04E–SIW–046, June 2004.

[7] H. F. Nielsen, E. Christensen, and J. Farrell. Specification: Ws-attachments. `http://www-106.ibm.com/developerworks/webservices/library/ws-attach.html`, June 2002.

[8] M. E. Pollack and M. Ringuette. Introducing the Tileworld: Experimentally evaluating agent architectures. In *National Conference on Artificial Intellegence*, pages 183–189, 1990.

[9] J. M. Pullen, R. Brunton, D. Brutzman, D. Drake, M. Hieb, K. Morse, and A. Tolk. Using Web services to integrate heterogeneous simulations in a Grid environment. *Future Generation Computer Systems*, 21:97–106, 2005.

[10] R. Tobias and C. Hofmann. Evaluation of free Java libraries for social-scientific agent-based simulation. *Journal of Artificial Societies and Social Simulation*, 7(1), Jan. 2004.

[11] S. van Splunter, N. J. E. Wijngaards, F. M. T. Brazier, and D. Richards. Automated component-based configuration: Promises and fallacies. In *Proceedings of the Adaptive Agents and Multi-Agent Systems workshop at the AISB 2004 Symposium*, pages 130–135, 2004.

[12] Y. Xie, Y. M. Teo, W. Cai, and S. Turner. Service provisioning for HLA-based distributed simulation on the Grid. In *Proceedings of the Nineteenth ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation (PADS 2005) (formerly called Parallel and Distributed Simulation)*, pages 282–291, Monterey, CA, USA, June 2005.

[13] Y. Zhang, G. Theodoropoulos, R. Minson, S. Turner, W. Cai, X. Yong, and B. Logan. Grid-aware large scale distributed simulation of agent-based systems. In *5rd European Simulator Interoperability Workshop (Euro-SIW)*, pages 05E–SIW–047, Toulouse, France, June 2005.

[14] W. Zong, Y. Wang, W. Cai, and S. Turner. Grid services and service discovery for HLA-based distributed simulations. In *The 8th IEEE International Symposium on Distributed Simulation and Real Time Applications (DSRT 2004)*, pages 116–124, 2004.