Middleware2003
--
How to harness the Grid with OGSA - Tutorial Proposal

Wolfgang Emmerich
University College London
Department of Computer Science
Gower Street
London WC1E 6BT
+44 20 7679 4413
W.Emmerich@cs.ucl.ac.uk
(contact person)

Ben Butchart
University College London
Department of Computer Science
Gower Street
London WC1E 6BT
+44 207 679 3723
b.butchart@cs.ucl.ac.uk

Clovis Chapman
University College London
Department of Computer Science
Gower Street
London WC1E 6BT
+44 207 679 7758
c.chapman@cs.ucl.ac.uk

Aims

The aim of this tutorial is to enhance awareness and exploitation of existing and emerging Grid middleware. In particular we promote the leverage of Web Service technologies to build distributed, resource-sharing applications using the Open Grid Services Architecture (OGSA). We share our experience of building a production quality e-Science application using the Globus Toolkit 3.0 (GT3) [1] to encourage wider adoption and further development of this and other OGSA based middleware. We also aim to highlight the limitations and future challenges for such middleware based on our experience working with early manifestations of the OGSA specification.

Learning Objectives

The tutorial will provide a concise overview of Grid middleware, the motivation and principles that drive Grid technologies and the characteristics that make Grid middleware distinct from other paradigms such as J2EE and .NET. Participants will learn how the Open Grid Service Architecture (OGSA) extends existing Grid paradigms to harness emerging Web service technologies such as SOAP and WSDL. The tutorial will enable participants to create simple OGSA services using the Globus Toolkit 3.0 (GT3) implementation and deploy these services over a distributed network. Participants will learn how legacy code such as Fortran binaries can be refactored and integrated into a Grid network by wrapping code in discrete OGSA services. Participants will learn about the practical and political problems encountered when developing Grid software in a network owned and administered by different departments and/or organizations.

Duration

We propose material lasting 3 hours for this tutorial.

Audience

We anticipate that the tutorial will attract two audiences: participants already familiar with Web services middleware who are interested to find out about Grid technologies and participants familiar with previous Grid standards wanting to learn about OGSA and the Globus Toolkit 3.0. Therefore we assume no knowledge of either Web Services or the Grid but do expect participants to have a basic understanding of fundamental middleware technologies such as XML, component based design, RPC and Java. We also think that the tutorial will interest delegates who are considering deployment of an OGSA implementation and would like to learn from our experience of developing a fully productive application using the Globus Toolkit.

Tutorial Contents

What is the Grid? An overview of Grid middleware

The basic idea of Grid computing is based on the familiar system of distributing high voltage electricity across large network connecting electricity providers with suppliers and consumers, constantly maintaining a balance between supply and demand and ensuring an agreed quality of service. Given the success of such grid systems for providing efficient distribution and exploitation of electricity and other utilities, the notion of applying a similar model to networks connecting various computing resources, which typically are neither efficiently distributed nor fully exploited, is compelling. But there are many different interpretations of how the grid analogy applies to computing systems and little consensus on what functionality and services a computing grid should provide. The various manifestations of grid middleware reveal very different goals and priorities depending on factors such as whether the software is geared towards shared data storage, CPU sharing, scientific applications vs. business applications or intranet vs. internet networks. Using our experience from the many grid projects currently in progress at UCL we discuss the characteristic functionality that we associate with grid applications and that make grid middleware distinct from other paradigms in distributed computing.

The Open Grid Services Architecture

The Open Grid Services Architecture [2] is an attempt to integrate Web service technologies with existing Grid standards so that developers of grid applications can benefit from the broad commercial support for Web service standards. The OGSA specification borrows from the Web Service toolkit mechanisms for service description and discovery (WSDL, WSIL) [3][4], automatic generation of client and server code from descriptions, binding to network protocols and support for other emerging higher level standards built on core Web service technologies (WSFL, BPEL) [5]. At the same time OGSA leverages experience from the Grid community in supporting resource management, information discovery, security and notification. In contrast to other Grid models, the OGSA approach is service oriented, where the focus is on the high level functionality that resources support rather than the physical resources themselves. This is achieved by providing a uniform interface and

semantics for accessing functionality encapsulated in transient service instances hosted by physical resources in the environment. This environment is characterized as a *virtual organization*, a constantly fluctuating ensemble of resources and services that various communities and individuals tap into to meet both their immediate and enduring computing needs [6].

We illustrate OGSA design features with reference to a fully productive e-Science application that we have modified to operate in a Grid environment using the Globus Toolkit 3.0. Computational chemists developed the original application combining functionality from several Fortran programs to create a process to predict stable crystal forms from an initial molecular structure. The process is extremely hungry for CPU time and even a simple analysis currently takes six to eight weeks to run. Apart from the limitation of running code on a single processor, scientists have to manually copy and reformat files so different parts of the process can run on platforms most suitable for a particular calculation. This lack of automation and resource coordination frequently results in errors forcing a total or partial rerun of the analysis. We show how an OGSA implementation facilitates the distribution of such applications across a large network of heterogeneous platforms radically improving performance of the system through parallel CPU capacity, coordinated resource management and automation of the business process.

The Globus 3.0 Toolkit (alpha version)

The Globus Toolkit 3.0 (GT3) [1] is a high quality open source implementation of the OGSA specification. The GT3 core design provides a lightweight hosting environment that transparently manages the underlying network infrastructure and transport protocols allowing users to focus on the functionality of the application rather than task of distributing the application across a heterogeneous network of resources. To achieve this, the implementation exploits existing state-of-the-art Web Service technologies enabling services to expose themselves through WSDL interface and providing automatic generation of client and server code stubs. This greatly simplifies the process of creating services that can operate in a Grid environment. In addition to standard Web service functionality the programming model offers a transport independent security infrastructure, lightweight discovery and inspection, dynamic deployment and soft-state management of stateful service instances.

The Globus Toolkit 3.0 will also incorporate many high-level services built on features present in the previous versions of the toolkit. We will briefly cover the following:

- The GRAM: The Grid Resource Allocation Manager API allows jobs, in the form of executable files, to be run remotely and provides the necessary means for submitting, monitoring and terminating a job.
- Data Grid tools: a set of features designed to deal with the management of large amounts of data, including Grid FTP for high-throughput file transfer or the replica catalogue to hold file management metadata.
- The Grid Security Infrastructure and OGSA Security Architecture.

Integrating legacy code into OGSA services

Often users will want to integrate existing code into OGSA services to take advantage of the capacity the Grid offers for running applications in parallel across many processors or distributing data across a large network of data storage devices. The GT3 delegation model provides OGSA functionality without having to modify legacy code. Higher-level services can be built on top of the delegation framework to provide configurable grid-enabled components that facilitate the integration of different kinds of legacy code into a Grid environment [7]. Using a Web Service compliant business process language such as the Business Process Execution Language [5], users can specify the interactions between these high level services to create powerful distributed application logic that is instantly deployable on any OGSA based Grid network.

With reference to our running example, the crystal polymorph prediction application, we will illustrate how the GT3 delegation model is applied to wrap legacy code so that functionality is distributed across the Grid network without having to modify the original Fortran code.

Summary and Conclusion

As a conclusion to this tutorial, we will share lessons learnt from our experience developing with OGSA and highlight the limitations as well as the benefits of deploying OGSA middleware. In particular, we will examine the behaviour of the crystal polymorph prediction system operating in a Grid environment and consider how effectively the implementation exploits available resources. We also discuss practical and political considerations that arise from "real world" Grid environments where technical arguments are often compromised by the needs and preferences of different users, organizations and domain administrators.

Conduct of tutorial

Delivery

The tutorial will consist mostly of a talk support by a Powerpoint presentation. We also intend to illustrate some concepts with live software demonstrations: the first outlining the process of creating a simple OGSA service using the Globus Toolkit 3.0 (GT3) from simple interface description through to stub generation and wrapping an implementation using the delegation model; the second demonstrating the crystal polymorph application running in a simulated Grid environment. This will enable participants to experience Grid middleware from a developer's perspective and lend some reality to the concepts and mechanisms the tutorial covers.

AVA and requirements

The tutorial and demos will be delivered from a single laptop PC that we will provide. We will require a projection facility powerful enough for the room and anticipated size of the audience.

Speakers' Profile

Ben Butchart and Clovis Chapman are both Research Fellows at University College London and have experience both in industry and academic research working with middleware technologies such as J2EE, JXTA, XML and Web Services. They are currently involved in a project to re-engineer an existing computational chemistry application to operate in a Grid environment using early releases of the Globus Toolkit 3.0. This has provided valuable practical experience of deploying an OGSA application in a production environment.

Dr. Wolfgang Emmerich is a Senior Lecturer at University College London and heads the Software Engineering Research Group. His group has conducted numerous tutorials on middleware topics at conferences around the world including:

- Component Technologies: Java Beans, COM, CORBA, RMI, EJB and the CORBA Component Model. 24th Int. Conference on Software Engineering, Orlando, Florida, 2002 [8]
- Mobile Computing Middleware. Middleware 2001. Heidelberg, Germany, 2001 [9]
- Distributed Objects. 21st Int. Conference on Software Engineering, Los Angeles, 1999 [10]
- Recent Developments in OMG/CORBA. 20th Int. Conference of Software Engineering, Kyoto, Japan, 1998 [11]

Schedule

The purpose of this section is show that the material is suitable for a 3 hours tutorial:

Introduction and Outline	5 min
What is the Grid ?	10 min
Overview of Grid middleware	15 min
Open Grid Service Architecture	20 min
Case Study: potential of Grid for improving the crystal prediction application	10 min
Globus Tookit 3.0 Core Design	15 min
Case Study: example creating an OGSA service (live demo)	15 min
Break	30 min
Wrapping Legacy code	10 min
Case Study: example wrapping a Fortran binary	10 min
Higher Level Globus Toolkit Features	15 min
Case Study: live demo of the crystal polymorph prediction tool	15 min
Case Study: analysis of system behaviour and resource usage	15 min
Lessons learnt and future challenges	10 min
Conclusion and Summary	5 min

References:

[1] Zhang, L., Chung, J., Zhou, Q. Introduction to a Grid architecture and toolkit for building Grid solutions, IBM, 2002, http://www-106.ibm.com/developerworks/webservices/library/ws-grid1/

[2] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C.,Vanderbilt, P. Grid Service Specification, Global Grid Forum, 2002 www.gridforum.org/ogsi-wg/drafts/draft-ggf-ogsi-gridservice-04_2002-10-04.pdf

[3] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. Web Services Description Language (WSDL) 1.1, W3C, 2001, www.w3.org/TR/wsdl

[4] Ballinger, K,. Brittenham, P., Malhotra, A., Nagy, W,. Pharies, S. Web Services Inspection Language Specification (WS-Inpection) 1.0, IBM and Microsoft, 2001 http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html

[5] Curbera, F., Goland, Y., Klein, J., Leymann, F., Roller, D., Thatte, S., Weerawarana, S. Business Process Execution Language for Web Services, Version 1.0 BEA Systems, IBM, Microsoft, 2002 http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbiz2k2/html/bpel1-0.asp

[6] Foster, I., Kesselman, C., Nick, J. and Tuecke, S The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project, 2002, www.globus.org/research/papers/ogsa.pdf

[7] Sandholm, T., Seed, R., Gawor, J. OGSI Technology Preview Core – A Grid Service Container Framework (Draft), Globus Project, 2002 http://www.globus.org/ogsa/releases/TechPreview/OGSITPCore.pdf

[8] W. Emmerich and N. Kaveh. Component Technologies: Java Beans, COM, CORBA, RMI, EJB and the CORBA Component Model. Proc. of the 24th Int. Conference on Software Engineering, Orlando, Florida. pp. 691-692. ACM Press. 2002.

[9] L. Capra, W. Emmerich and C. Mascolo Exploiting Reflection and Metadata to build Mobile Computing Middleware. In Proc. Workshop on Mobile Computing Middleware. Heidelberg, Germany, November 2001

[10] W. Emmerich and N. Roodyn. Distributed Objects. Tutorial Summary. In: Proc. of the 21st Int. Conference on Software Engineering, Los Angeles. pp. 665-666. IEEE Computer Society Press. 1999

[11] W. Emmerich. Recent Developments in OMG/CORBA . Tutorial Summary. Proc. of the 20th Int. Conference of Software Engineering, Kyoto, Japan. Volume II, pp. 276--277. IEEE Computer Society Press. 1998