GRIDCC - providing a real-time Grid for distributed instrumentation

P R Hobson, School of Engineering & Design, Brunel, UK
A. S. McGough, London e-Science Centre, Imperial College London, UK
D. Colling, High Energy Physics Group, Imperial College London, UK
On behalf of the GRIDCC Collaboration *

Abstract

The GRIDCC project is extending the use of Grid computing to include access to and control of distributed instrumentation. Access to the instruments will be via an interface to a Virtual Instrument Grid Service (VIGS). VIGS is a new concept and its design and implementation, together with middleware that can provide the appropriate Quality of Service (QoS), is a key part of the GRIDCC development plan. An overall architecture for GRIDCC has been defined and some of the application areas, which include distributed power systems, remote control of an accelerator and the remote monitoring of a large particle physics experiment, are briefly discussed.

INTRODUCTION

Recent developments in Grid technologies have concentrated on providing batch access to distributed computational and storage resources. GRIDCC is extending this to include access to and control of distributed instruments. In this paper we define an instrument to be any piece of equipment controlled through a computer interface such as telescopes, particle accelerators or power stations. Instruments work in real-time and their successful operation often requires rapid interaction with conventional computing/storage resources and/or other instruments. The control of instruments is often an interactive process. The real-time and interactive nature of instrument control provides a critical requirement for the definition of acceptable Quality of Service (QoS) constraints for interactions between the different Grid components.

The GRIDCC [1] project is a 3 year project funded by the European Union which started in September 2004. There are 10 project partners from Greece, Italy, Israel and the United Kingdom. The first complete release of the software will be during the second year of the project. The goal of GRIDCC is to build a widely distributed system that is able to remotely control and monitor complex instrumentation that ranges from a set of sensors used by geophysical stations monitoring the state of the earth to a network of small power generators supplying the European power grid. These applications introduce requirements for real-time and highly interactive operation of computing Grid resources.

GRIDCC has developed a new element for abstracting the instruments on the Grid. This Instrument Element (IE) is a unique concept to GRIDCC. It consists of a coherent collection of services which provide all the functionality to configure, partition and control the physical instrument. The GRIDCC project has also developed a skeleton architecture for using instruments within a worflow. Thus allowing scientists to automatically coordinate as much of their scientific process as possible. In the current phase the project is now developing these elements in order to meet the real-time requirements.

ARCHITECTURE

The overall architecture of GRIDCC has now been defined and is shown in Figure 1. The Virtual Control Room (VCR) is the GRIDCC user interface. It allows users to build complex workflows which are then submitted to the Execution Service, it can connect to resources directly and be used to control instruments in real time. The VCR also extends human interactions with Grid resources through its Multipurpose Collaborative Environment (MCE).

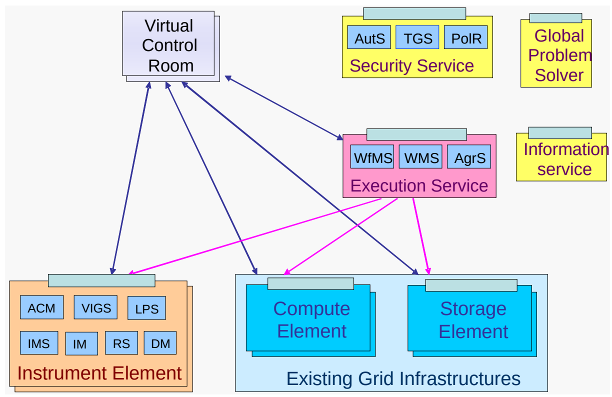

Figure 1: The Components of the GRIDCC architecture

Instrument Element (IE)

The IE is a concept unique to the GRIDCC project. It is an abstraction of the instrument (or group of instruments) into a standard interface which can be used within the rest of the GRIDCC architecture. The term instrument is used in this context to define a piece of equipment that needs to be initialized, configured, operated (start, stop, standby, resume, application specific commands), monitored or reset.

The Instrument Element is a collection of (Web) services: the Virtual Instrument Grid Service (VIGS), the Resource Service (RS) and the Local Problem Solver. The

* The GRIDCC Project `http://www.gridcc.org`, (gridcc-general@imperial.ac.uk)

VIGS concept is at the core of GRIDCC, it is shown diagrammatically in Figure 2. The Access Control Manager (ACM) will identify every user or entity (another VIGS for instance) and provide the proper authorization protocol. The Instrument Manager (IM) receives commands from the VCR (or another VIGS). These are then routed to the instruments under the control of the IM.

A Resource Service instructs the IM on the number, type and location of the instruments it should control: it knows the topology of the instruments, the connection type, their identification, etc. The Info Service Proxy collects all the information (errors, state and monitor data) from the instruments and acts as local cache before sending the data to the central Information Monitoring Service (IMS). The Data Mover Manager moves the data acquired from the instruments. Automated problem solving in a Grid environment is a major feature of the project. Two levels of problem solver are proposed; one to solve problems related to the function of a given instrument and one to solve system wide problems.

Virtual Control Room (VCR)

The VCR is the application scientists interface into the GRIDCC system. It should be tailored to the scientist's needs and present information in a manner which is understandable to the scientist. It should allow interactive real-time control of instruments along with the ability to work collaboratively with other scientists. Workflows may be constructed through the VCR, though they may be presented in a tailored manner for the application scientist.

Information & Monitoring Service (IMS)

The IMS provides a repository for information collected within the architecture. It may be interrogated to discover information.

Global Problem Solver (GPS)

The GPS uses information from the IMS in order to diagnose potential situations which may lead to damage of the instrument or other elements. It is also responsible for taking whatever actions are appropriate to remedy these situations.

Security Services (SS)

The applications for which GRIDCC is designed mean that security – Authentication, Authorization and Accounting – are especially important in this project. The damage that could be caused by a security incident resulting in misuse of the components of the National Power Grid or the multi-billion Swiss Franc CMS detector is vast. Yet the security system is constrained to both to be light, so that the GRIDCC QoS requirements can be met, and to interoperate with security systems of other Grid systems. In order to satisfy these two, almost mutually contradictory, requirements GRIDCC has decided to use a split security system. When interacting with the components of other Grid projects the GSI security will be used and the user identified by their X.509 [10] proxy certificate, however when interacting with the IE the user will be identified by a Kerberos ticket. The level of security provided by both systems is equivalent. Though Kerberos [7] has a lower overhead.

Our proposed baseline security architecture has the end-user authenticated at the User Interface (Control Room) by using their system credentials, (e.g. username/password, and a X.509 Certificate issued by a trusted Certification Authority). The core system, consisting of Control Rooms (User Interfaces) and VIGSs (Computing Elements), employs a ticket-based authentication mechanism and symmetric cryptography via centralized Key Distribution Servers. Authorization is performed at the VIGS level employing local access rules. In Use Cases requiring secure and accountable interfacing to GSI based Grids (e.g. EGEE/LCG) end-users should pass GSI proxy certificates along with their control messages that trigger the GridCC EGEE/LCG interaction.

Execution Service (ES)

The ES is responsible for taking a workflow describing what the application scientist wishes to be performed and ensuring that it is executed within any QoS requirements which may exist.

Compute Element (CE)

The CE is an abstraction of a resource (or group of resources), which can perform computation, into a standard interface which can be used within the rest of the GRIDCC architecture.

Storage Element (SE)

The SE is an abstraction of a resource (or group of resources), which can store data, into a standard interface which can be used within the rest of the GRIDCC architecture.

SOFTWARE DEVELOPMENT AND STANDARDS

Much effort has already happened within the e-Science community to develop Grid middlewares. These middlewares have in general been tailored towards non interactive computational job execution, such as Condor [2, 13], Globus toolkit (versions 1 through 4) [6, 5, 14], Grid Engine [12] and LSF [11]. Rather than ignore the substantial effort made by these projects we seek here to extend them with the added functionality. Wherever possible we use existing code and community standards in developing GRIDCC.

The GRIDCC architecture implementation is thus being developed within gLite [4] the Grid middleware devel-

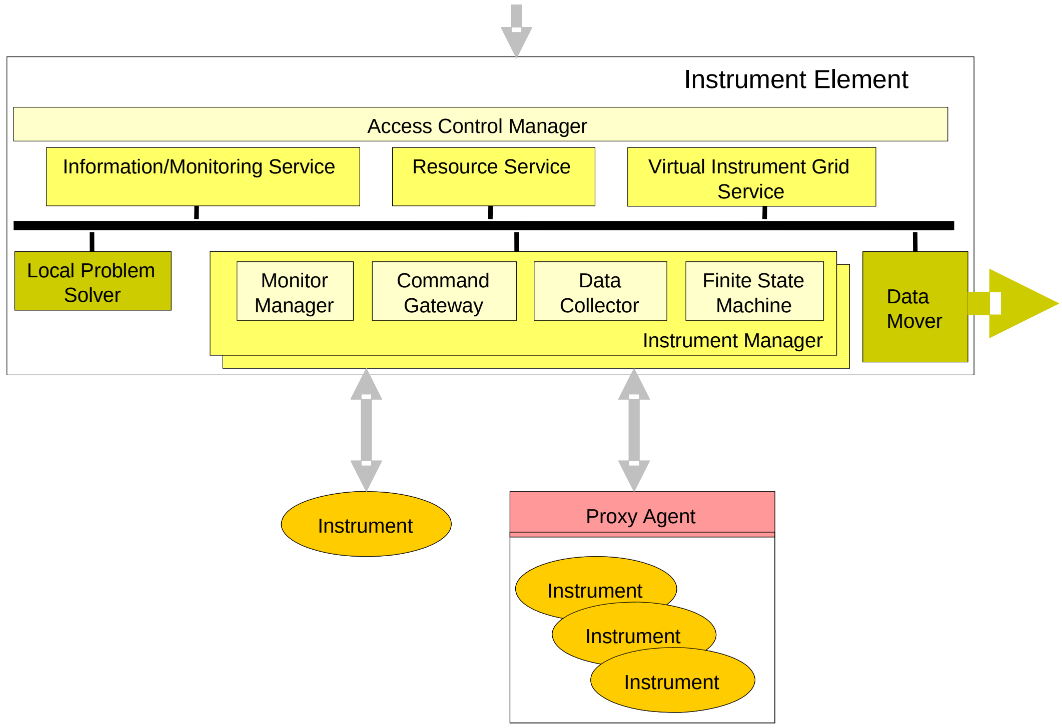

Figure 2: A Schematic of the VIGS concept. The terms are defined in the text

oped within the European community from the EGEE [3] project. Other potential candidates that were identified included the Globus toolkit version 4 and the OMII service infrastructure [8]. Although all of these architectures are at approximately the same level of maturity the decision to go with gLite was based around the accessibility to the developers as there is an overlap of members of the GRIDCC and gLite development team.

The Grid community has, over the last few years, been moving rapidly between different underlying middleware architectures. These have included such technologies as JINI [15], JXTA [16], OGSI [9] and many proprietary protocols. Stability is now being approached within the Grid community with most developers moving towards Web Service [19] oriented middlewares. The increased use of firewalls between sites on the Internet has caused significant problems when using many of the original mddlewares (such as JINI). As JINI was developed for use in a local environment without firewalls it uses a large range of ports when communicating between instances. This has caused major problems when deploying between administrative domains. This problem has been resolved by the development of JINI2. However, it was felt that Web Services provided a better solution.

There are many workflow description languages available for use within the Grid. These include BPEL [17], WS-Choreography [18] and YAWL [20]. Again with the intention of leveraging good quality implementations the choice was made to use BPEL as this appears to be the more favored option with a number of commercial and open source engines becoming available.

THE APPLICATIONS

A variety of applications have been considered within the GRIDCC project and their use cases have had a strong bearing on the design and implementation of the software. These range from meteorology, geo-hazards, neurophysiology, and power grids to accelerator control and high energy physics experiment control. While it is hoped that many of these applications will use the GRIDCC software the project itself has chosen 3 applications to take through to completion. These are:

Control and monitor of a High-Energy Physics (HEP) experiment

This application involves three basic elements: a very high number of instruments, a large number of concurrent ”users” (sometimes in the form of automatic monitoring tasks) and a high-rate of incoming data. These characteristics result in stress-tests for the monitoring of data streams, the network(s) involved, the QoS capabilities (prioritizing the data stream against normal traffic), as well as the capability to respond to different types of error conditions arising from normal operation. This latter element is very important for the analysis of what should result in a system stop and what should not and has to take place under the stringent real-time requirements of the experiment’s data-taking process.

Far Remote Operation of an Accelerator

This application also involves large numbers of instruments (mainly sensors). It is different from HEP Control and Monitor in (a) the rate of incoming data: it is far smaller that the data in HEP but of potentially far higher importance which introduces new requirements on the reaction times to alarms (b) the level of the human-machine interface, which again is higher than that in HEP.

Power Grids

In electrical utility networks (or power grids), the introduction of very large numbers of 'embedded' power generators often using renewable energy sources, creates a severe challenge for utility companies. Existing computer software technology for monitoring and control is not scalable and cannot provide a solution for the many thousands of generators that are anticipated. GRIDCC technology would allow the generators to participate in a Virtual Organization, and consequently to be monitored and scheduled in a cost-effective manner.

A specific testbed application will be built and demonstrated within the GRIDCC project by means of computer simulation and emulation. Existing software at Brunel University will allow the real-time simulation of a representative power network and the associated generators. New software will be created to interface the generator simulations to the GRIDCC environment. Distributed generator scheduling algorithms will be modified to utilize GRIDCC technology. The test bed will demonstrate the performance of the emulated system under various conditions, ranging from light power system loading (where energy economics is most important) to power system emergency conditions (where overloaded power circuits necessitate co-ordinated generator control to avoid power black-outs).

CURRENT STATUS OF THE PROJECT

The overall architecture of GRIDCC has now been determined. We have developed a Instrument Element and a skeleton workflow architecture. Progress has been made in defining an overall framework for the problem solver. Where possible we we have been mindful of the need to inter-operate with other Grid projects and to adhere to the standards that are emerging in the world of Web Services. As the baseline architecture is now available the project has now moved onto a phase of evaluation and hardening. We seek to make the architecture capable of meeting the real time constraints that the use of instruments dictates. Thus the performance of the architecture is being evaluated. The tasks that the users are submitting are also being evaluated. This will allow for the timely deployment of these tasks onto resources such that the real time requirements can be met.

REFERENCES

[1] The gridcc project. `http://www.gridcc.org`.

[2] Condor Team. Condor Project Homepage. `http://www.cs.wisc.edu/condor`.

[3] EGEE. Enabling Grids for E-science. `http://public.eu-egee.org/`.

[4] EGEE. EGEE Middleware Architecture, EU Deliverable DJRA1.4. `https://edms.cern.ch/document/594698/`, july 2005.

[5] I. Foster and C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, Summer 1997.

[6] I. Foster and C. Kesselman. The Globus Project: A Status Report. In *IPPS/SPDP '98 Heterogeneous Computing Workshop*, pages 4–18, 1998.

[7] John T. Kohl, B. Clifford Neuman, and Theodore Y. T'so. *Distributed Open Systems*, chapter The Evolution of the Kerberos Authentication System, pages 78–94. IEEE Computer Society Press, 1994.

[8] OMII Project. Open Middleware Infrastructure Institute. `http://www.omii.ac.uk`.

[9] Open Grid Services Infrastructure (OGSI) v1.0. `http://forge.gridforum.org/projects/ggf-editor/document/draft-ogsi-service1/en/1`.

[10] OpenSSL. X509. `http://www.openssl.org/docs/apps/x509.html`.

[11] Platform Computing. Project LSF. `http://www.platform.com/products/LSF`.

[12] Sun Microsystems. Project SGE. `http://wwws.sun.com/software/gridware/`.

[13] Thain, D., *et al.*,. Condor and the Grid. In F. Berman, A. J. G. Hey, and G. Fox, editor, *Grid Computing: Making The Global Infrastructure a Reality*. John Wiley, 2003.

[14] The Globus Project. `http://www.globus.org/`.

[15] The Jini project team. Jini Network Technology. `http://wwws.sun.com/software/jini/`.

[16] The JXTA project team. Project JXTA. `http://www.jxta.org/`.

[17] The OASIS BPEL team. BPEL4WS. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel`.

[18] The W3C WS-Choriography team. WS Choriography Model Overview. `http://www.w3.org/TR/2004/WD-ws-chor-model-20040324/`.

[19] W3C. Web Service. `http://www.w3.org/TR/ws-arch/`.

[20] W.M.P. van der Aalst and L. Aldred and M. Dumas and A.H.M. ter Hofstede. Design and Implementation of the YAWL system. In *Proceedings of The 16th International Conference on Advanced Information Systems Engineering (CAiSE 04)*, Riga, Latvia, june 2004. Springer Verlag.