THE GRIDCC PROJECT PROVIDING A REAL-TIME GRID FOR FOR DISTRIBUTED INSTRUMENTATION

R.Pugliese[1], F.Asnicar[1], L.Del Cano[1], C.Scafuri[1] , G.Maron[2]
on behalf of the GRIDCC Collaboration
[1] *Sincrotrone Trieste SCpA, Trieste, ITALY,* [2] *INFN, Legnaro, ITALY*

ABSTRACT

The GRIDCC project will extend 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 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.
The goal of GRIDCC [1] 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. The project is currently in its initial stages and work is primarily concentrated in the areas of workflow and architechture definition, refining the use-cases for the applications and the testing and evaluation of existing Grid middleware to determine whether it meets our requirements.
The talk will present the overall architecture of the system and current status of the GRIDCC project. From the outset the GRIDCC project team analysed use cases from a wide range of different applications. The applications are: the High-Energy Physics experiment CMS [2], the particle accelerator ELETTRA [3], monitoring of distributed electrical power generators [4] meteorology, analysis of neurophysiological data for migraine detection, and a geophysical monitoring network. The first three of these are regarded as core applications and are currently being taken forward to demonstration on actual Grid test-beds.

INTRODUCTION

The GRIDCC 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. 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.

GRIDCC is developing these definitions, where appropriate, building on work that is being carried out in the various standards organisations. In order to meet these defined QoS constraints it is often necessary to provide mechanisms for reserving the instruments, and subsequently for the compute and storage elements they will require. GRIDCC will run a representative set of applications in order to validate the software produced by the project. These applications will control and monitor different types of instrument, acquire data from these instruments and then analyse this data on existing Grid test-beds. This will demonstrate integration of this project with existing Grid projects. The project will deploy applications with a significant social impact (control of a network of power grids generators, meteorology, geo-hazards monitoring, analysis of neuro-physiological data), engineering impact (distributed telecommunications measurements) and scientific impact (control and monitor of a high energy physics experiment, far remote operation of an accelerator facility). To perform an experiment an application scientist will inevitably need to perform more than one task. They may wish to perform an experiment using a piece of equipment, analyse the results of this experiment and then potentially perform further experiments using the instrument in light of this analysis. This is but one simple example of what an application scientist may wish to perform. Thus there is a need to provide support for *workflows*, defined as the execution and management of sets of dependent and potentially heterogeneous tasks. The term "task" is used to indicate a general piece of work while a "job" refers specifically to a computational task. This must be powerful enough to deal with the myriad of possible workflows that the application scientist may wish to describe, but also presented in a manner to the scientist which is intuitive and simple for them to work with. It should not be assumed that the application scientist is a competent software engineer nor should they be forced to try to be such.

Application scientists should be presented with a similar interface to the Grid and the instruments to that which they are already familiar. For this the GRIDCC project is developing a Virtual Control Room (VCR) environment which will present to the scientist a tailored interface both to the Grid and the instruments. One of the barriers to placing instruments onto the Grid in the past has been the potential that inexperienced users may (unintentially) damage the instrument by miss-use. There is thus a strong requirement to protect the instrument through appropriate checks and measures. These include local problem solvers which are capable of diagnosing situations which would cause damage to a local instrument. This could be monitoring the temperature of motors on a telescope and closing it down in the case of overheating. Likewise there is a need for global problem solvers. These are required to diagnose problems which cannot be diagnosed or remedied by a single instrument. This could be monitoring the number of power stations that are online and preventing more from coming online when this would cause too much current to pass through part of the power grid.

The goal is to build a widely distributed system that is able to remotely control and monitor complex instruments that range 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 new applications introduce requirements for real-time and highly interactive operation of Grid resources. To achieve this goal we have three main objectives:

To develop generic Grid middleware, based on existing building blocks (Grid Services) which will allow the remote control and monitoring of instruments such as distributed systems.

To incorporate the new middleware into a few significant applications that will allow the software to be validated both in terms of functionality and quality of service aspects: European Power Grid, Meteorology, Analysis of Neurophysiological data, RemoteOperation of an Accelerator Facility, High Energy Phisics.

To disseminate widely the new software technology, the results of the evaluation on the test-beds and to encourage a wide range of enterprises to evaluate and adopt our Grid oriented approach to real-time control and monitoring of remote instrumentation.

THE GRIDCC ARCHITECTURE

In this section we present the overall architecture for the GRIDCC project. The architecture is composed of 8 service groups, see Figure 1, each of which may be decomposed into a number of individual services. These services are outlined below:

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 Service (IS): The IS 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 IS 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): These services are pervasive throughout the GRIDCC architecture and provide the mechanisms for performing Authentication and Authorisation of users.

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.

Instrument Element (IE): The IE 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.

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.

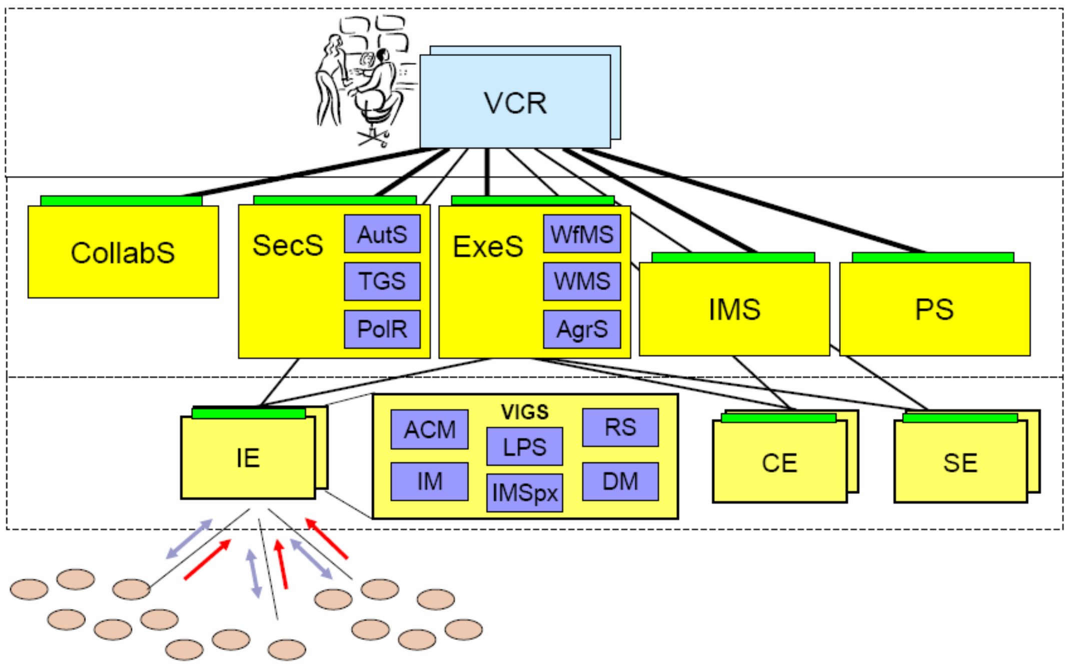

Figure 1: The GRIDCC Architecture

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[5], Globus toolkit (versions 1 through 4)[6], Grid Engine[7] and LSF[8]. 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[9] the Grid middleware developed within the European community from the EGEE[10] project. Other potential candidates that were identified included the Globus toolkit version 4 and the OMII service infrastructure[11]. 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. In the following we will descibe in more detail some of the components of the GRIDCC architecture.

THE VIRTUAL CONTROL ROOM

The Virtual Control Room 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 extents human interaction with Grid resources through its Multipurpose Collaborative Environment (MCE). The VCR is an instance of the MCE for a specific GRIDCC application. The VCR should be intuitively easy to use for the application scientist and present the required functionality in a manner which the scientist is familiar. Web-based portals for the delivery of scientific information, experimental and computational services using a Grid infrastructure have already proven effective in meeting these requirements. We thus use Grid portal technology to implement the VCR. The MCE is being built from components defined with public interfaces, semantics and standard behaviors. The MCE provides a set of collaborative services such as chat, audio/video conferencing, instant messaging as appropriate to aid control of an application. The Client, which provides the Human Computer Interaction (HCI) component, is being implemented via a Web browser running on a client computer or PDA and will present portlets to the user. The user has to log on to the portal, a step often referred to as authentication. This will activate the user's current "context”. Users can configure the portal to their own personal "view" and this will be stored when they log off. The user can select the portlets they require for a particular job and to customise these.

THE INSTRUMENT ELEMENT

The Instrument Element (IE) is a concept unique to GRIDCC. The term IE describes a set of services that provide the needed interface and implementation that enables the remote control and monitoring of physical instruments. 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 IE is viewed by users as a set of Web Services.Web Services are an excellent choice when there is a need to provide a common language to the cross domain collaboration and to hide the internal implementation details of accessing specific instruments, which is the case in the IE. The interface of the IE consists of three basic components: Virtual Instrument Grid Service; Resource Service; Information and Monitoring Service.

The above are generally public services that can be accessed by anyone, provided that they have the appropriate credentials which are checked by the *Access Control Manager*.

The IE is comprised of more components not directly accessible by external entities but indirectly through the edge services described above. They implement the operational functions and comprise the core engine of the IE. These components are the following: Instrument Manager(s) (IM); Local Problem Solver.

In Figure 2 we can see where these components are placed and how they interact together in a single installation. Specific details for the functionality of each of the components are described below.

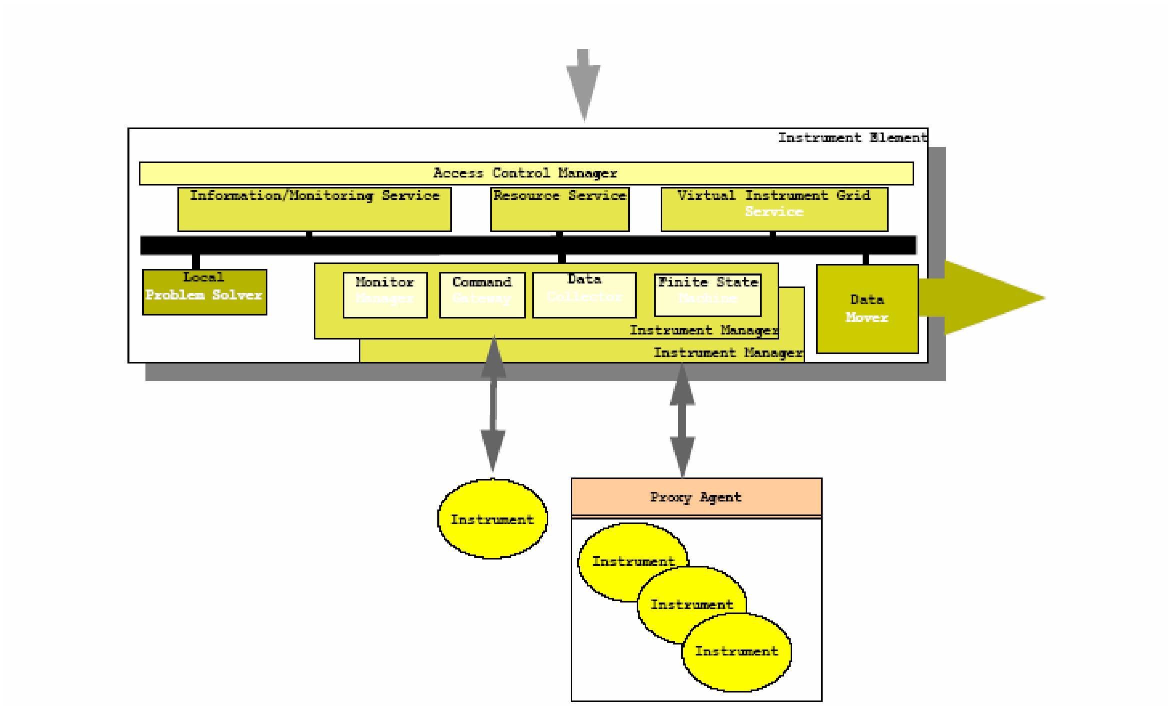

Figure 2: The internal architecture of the Instrument Element

The IM are the parts of the instrument element that perform the actual communication with the instruments. They act as protocol adapters that implement the instrument specific protocols for accessing its functions and reading its status. Since the instruments are heterogeneous in nature there is a need to support many IM's in the same container, one instance for each logical set of instruments.
Each IM is composed of four subcomponents:

- The *command gateway* accepts control requests from the Virtual Instrument Grid Service and forwards them to the instrument;
- The *monitor manager* is responsible for collecting monitoring data (status, errors, etc.) and provide it to the IS and/or the Local Problem Solver in a common format;
- The *data collector* component gathers the flow of data (if any) from the instruments. It then passes this data to the *Data Mover* which provides the interface with any external storage or processing elements;
- The *finite state machine* reflects the collective state of instruments controlled by the IM. It receives the status from the individual instruments and updates its state accordingly.

To transfer instruments from one IE to another, all that is required is to remove the IM instance from the original IE and instantiate a new one on the new IE. The other services (edge and core) react to this action accordingly. In general, each IM is collaborating locally with the interfacing services to translate external requests for control or monitoring to requests that the instruments understand and can react to The Local Problem Solver component has been introduced to add both problem solving and prediction capabilities inside the IE. It is positioned in the core of the IE and can collect alarms, errors, warnings, messages from all the IM's (more specifically from the monitor managers as described above), and to propose wherever possible appropriate recovery actions or raise alarms.
The local problem solver interacts with the information and monitor service in order to make available its status and alarms, warnings or other information to be processed by a larger scale system, like the Global Problem Solver to external entities. The Access Control Manager component is responsible for checking user credentials and deciding whether a request should be processed by the IE (this process is referred as authentication and authorization). Once it is decided to proceed with the request it forwards it to one of

the edge services. The Virtual Instrument Grid Service (VIGS) is the interface of the IE that is responsible for parsing control requests that target one or multiple instruments and forwarding them to the corresponding IM, with the collaboration of the Resource Service. This service is very critical since it has *real time requirements* and should react according to the control requests that arrive from external entities.

The Resource Service (RS) provides a catalog of all the instruments managed by the IE. With this service one can obtain specific information about the instruments themselves, learn about topology settings and about configuration parameters for each instrument. This information can then be used to prepare requests for setting or controlling the instruments in a specific manner and send to the instrument through the VIGS. Moreover, the RS deals with the logical partitioning of a set of instruments by a user with the objective of operating only a subgroup of them. This use case arises when an IE is handling a set of instruments that must be controlled by a single user or group to avoid conflicts with other requests. The instruments are then locked by the RS and accept requests only from that user / group. In turn when the need for the partition no longer required, the instruments are unlocked and can accept requests from all users. More sophisticated allocation and advanced reservation schemes can be added using the Execution Service. The Information and Monitoring Service uses a publish / subscribe system to disseminate monitoring data to the interested partners. More specifically each instrument, through the monitor manager publishes monitoring information to the Information and Monitoring Service arranged in topics. The local problem solver uses the same approach to publicise each state and corresponding data. The external entities, in turn, subscribe themselves as listeners to the topics and can receive synchronously or asynchronously the flow of monitoring data from the IE. The topics can be organized in subjects of general nature, specific to one or a set of instruments, in severity levels (info, warn, error) etc. The Information and Monitoring Service (IMS) can act as a standalone module, where listeners have to subscribe to it locally, or be part of a larger scale distributed system where the publishers and subscribers can register themselves in their local IMS and interact with all the IMSes in the Grid transparently.

CONCLUSIONS

The GRIDCC project is integrating instrumentation into traditional computational/storage Grids. The project started in September 2004 and the first full release of the GRIDCC software will be during the second year of the project. The software produced will be validated on at least three of representative applications.

REFERENCES

[1] http://www.gridcc.org
[2] http://cmsdoc.cern.ch/cms/TRIDAS/Temp/CMS_DAQ_TDR.pdf
[3] http://www.elettra.trieste.it/info/index.html
[4] M Irving, G Taylor, P Hobson, *IEEE Power & Energy Magazine* March/April 2004, pp 40 - 44
[5] The Condor Project, http://www.cs.wisc.edu/condor/
[6] The Globus Alliance, http://www.globus.org/
[7] The gridengine project,http://gridengine.sunsource. net/
[8] Platform LSF, http://www.platform.com/products/LSF
[9] gLite - Lightweight Middleware for Grid Computing, http://glite.web.cern.ch/glite/
[10] Enabling Grids for E-science, http://public.euegee.org/
[11] The Open Middleware Infrastructure Institute, http://www.omii.ac.uk