Implementing a Scientific Visualisation Capability within a Grid Enabled Component Framework

J. Stanton, S. Newhouse, and J. Darlington

London e-Science Centre, 180 Queen's Gate, London SW7 2BZ, UK,
lesc@ic.ac.uk, internet: http://www.lesc.ic.ac.uk*

Abstract. Collaborative scientific visualisation and computational steering are key enabling technologies within many e-science applications. Grid infrastructures provide an ideal environment for providing such a capability, and one approach is to use encapsulation within a Grid enabled component framework. This research note discusses the initial development of a collaborative visualisation component within the ICENI Grid middleware infrastructure.

1 Introduction

The delivery of collaborative scientific visualisation and computational steering of large data sets has been a goal for many years but has in practice been difficult to achieve.

As computational power has increased and high speed networks and sophisticated visualisation tools have become commonplace, so has risen the demand for infrastructures to support real-time collaborative visualisation and computational steering. To satisfy this demand the emerging Computational Grids, federations of distributed computational, storage and networking resources, must provide mechanisms to support collaborative visualisation environments that can be readily applied to scientific applications.

This research note describes the integration of such a capability within the Imperial College e-Science Networked Infrastructure (ICENI), an integrated Grid middleware for component based applications [1].

2 Deployment of Component Based Grid Applications

Effective Grid infrastructures provide a range of underlying services necessary for the efficient deployment and execution of applications. These are reviewed in detail in [2], but we can summarise a mimimum subset for the transparent use of Grid resources as follows:

* Research supported by the OST Core e-Science Programme on equipment provided by the HEFCE/JREI grant GR/R04034/01 and EPSRC equipment grant 2001

B. Monien and R. Feldmann (Eds.): Euro-Par 2002, LNCS 2400, pp. 885–888.

- registration services to provide the pool of high performance computational resources within the virtual Grid communities;
- security services to define access to these resources;
- analytic services to provide a basis for performance based deployment of applications onto these resources;
- monitoring services to support process re-deployment according to changes in network performance and the application's computational requirements.

Within the ICENI component based Grid infrastructure, the components which are composed to define applications are additionally annotated with performance data which is then used to determine the optimum deployment on available resources. Components which have large or diverse computational demands (such as for numeric computation and for visualisation) may therefore be instantiated by the Grid middleware on separate resources, or co-located if transport costs would negate any benefits that this would provide.

3 Implementing Collaborative Visualisation and Steering within ICENI

3.1 Design Architecture

Previous work had developed a prototype system for collaborative visualisation and steering within a distributed environment [3]. This has been extended by encapsulating its functionality within a co-ordinating server component that can be deployed by the ICENI grid middleware.

Users wishing to interface with an application do so by connecting to its corresponding server instance using a local visualisation client program. As each client connects the the server, a rendering process is deployed onto Grid resources to process the visualisation pipeline and transport the resulting image to the user's workstation using Chromium [4].

A data pull model is used, with each rendering process requesting new data as required to refresh its image. The server collects these requests and passes them to the data generating application. One copy of each data set requested is then returned to the server at the end of the following computation cycle, for distribution by the server to the rendering processes of the requesting clients.

Conceptually therefore the server component provides the inferface between the data generating application, the client processes and the rendering processes. In practice the transport mechanisms used for data exchange will depend on the deployment of the processes and may range from direct access for co-located components to GridFTP for high volume transfer between remote components.

The communication model for an example session involving two clients and deployment of all processes on remote resources is shown in Figure 1.

3.2 Demonstrator Application

A scientific application simulating coronal mass ejections [5] was selected to demonstrate the functional capabilities of the initial implementation. Originally

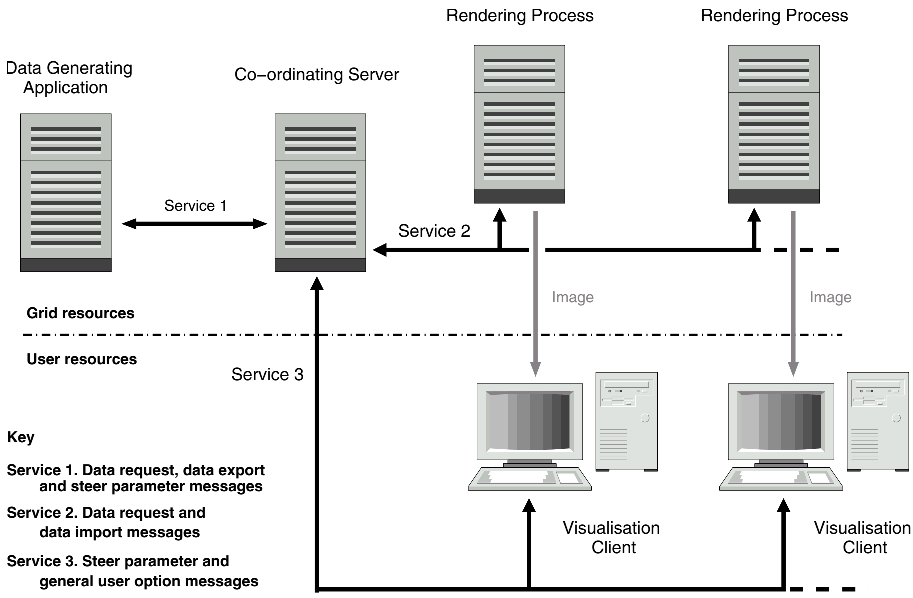

Fig. 1. Process and Communication Model for Collaborative Visualisation within the ICENI Framework

structured as a sequential Fortran program capable of resolutions necessary for 2-D batch mode analysis on a high performance UNIX workstation, it has recently become practical to execute the code on high-end commodity PC's. Results are output to file for subsequent visualisation with a typical run of 600 cycles capable of generating a data file of 160 MB.

A simple restructuring of this code was carried out to facilitate the insertion of the library routines necessary to set up the control and data paths to the co-ordinating server. A rendering process capable of generating a sequence of contoured colour maps from a corresponding sequence of arbitrary two or three dimensional scalar arrays was defined using the VTK library [6].

Three scalar arrays were identified for export, consisting of magnetic field data which would generate images representing the motion of magnetic flux tubes within a magnetised plasma. A single steerable parameter was identified defining the computational time step, which would provide a means to balance the accuracy of the simulation with its rate of development (and stability).

3.3 Operational Results

Proof of concept trials have been carried out using resources within the London e-Science Centre [7]. The data generating application was executed on a 24 processor Sun E6800 machine, while the co-ordinating server and rendering processes executed on a 20 processor Linux cluster. The visualisation clients were established on desktop PC's. These tests demonstrated the capability of multiple clients to visualise the same or different data sets concurrently and to steer the data generating application.

Simple quantitative tests have also been carried out to determine the overhead effects within the data generating application. In the absence of any visualisation or steer activity, basic system overhead increased the time for each cycle of the application by less than 0.001%. Monitoring and updating steerable parameters increased cycle time by 0.14%. Export of a requested data set was subject to wide variation but on average increased cycle time by 3.1%. This compared with 1.5% for exporting a corresponding data set to file, however the data pull approach used ensures that data is only exported when specifically requested by a rendering process.

4 Conclusions and Further Work

Computational Grids provide an ideal environment within which to provide the collaborative visualisation service essential to many e-science applications.

ICENI, a Grid enabled component based application framework, has been developed as a means of encapsulating domain specific knowledge together with performance data into components for optimised deployment onto Grid resources. The functional requirements of a collaborative visualisation and steering capability can similarly be componentised for integration within a user's application.

An initial implementation of such a component has been developed and has demonstrated the basic viability of this approach with a simple sequential application. This implementation has used a simple token based mechanism to control read access to data and read/write access to the steerable parameters. Key areas for further work are therefore to integrate the access control mechanisms with the policy based mechanisms of the ICENI framework, and to provide support for interaction models appropriate to more complex applications.

References

1. Furmento, N., Mayer, A., McGough, S., Newhouse, S., Darlington, J.: Optimisation of Component-based Applications within a Grid Environment. SuperComputing 2001, `http://www.icpc.doc.ic.ac.uk/components/papers/index.html`
2. Foster, I., Kesselmann, C. (Editors): The Grid: Blueprint for a New Computing Infrastructure. Morgan Kauffmann, 1998
3. Stanton, J.: Enabling Computational Steering through Visualization in a Distributed Environment. Master's Thesis, Imperial College, Department of Computing, `http://www.doc.ic.ac.uk/~jks100/msc`
4. Sourceforge: Project: The Chromium Project, `http://sourceforge.net/projects/chromium/`
5. Cargill, P., Chen, J., Spicer, D., Zalasak, S.: Magnetohydrodynamic Simulations of the Motion of Magnetic Flux Tubes through a Magnetised Plasma, Journal of Geophysical research, Vol 101, March 1996
6. Kitware Inc.: The Visualisation Toolkit, `http://public.kitware.com/VTK/`
7. London e-Science Centre: Resources, `http://www.lesc.ic.ac.uk/resources/index.html`