EVE – A Grid enabled data analysis environment for neutron scattering experiments

Ronald Fowler, Anjan Pakhira, Lakshmi Sastry, Toby Perring*

e-Science Centre, CCLRC Rutherford Appleton Laboratory, UK.
*ISIS Department, CCLRC, RAL, UK.

Abstract

The ISIS facility at RAL produces data on the structure of condensed matter through the scattering of neutrons. The amount of data generated in a single experiment is set to increase greatly as more sophisticated instruments come on line. Analysis of this data and fitting to theoretical models is a task that will require access to large data storage and compute resources in a timely manner for experimentalists to assess the data in the course of an experiment. Grid computing is being used in this project to provide users with fast interactive data analysis. A MATLAB based client is used to give a familiar and flexible interface to the scientists, while the Globus toolkit is used to access remote resources. Visualization processing will be performed on the remote resource.

1. Data analysis for ISIS

The ISIS facility at the Rutherford Appleton Laboratory [1] is the world's most powerful pulsed spallation neutron source. It provides beams of neutrons and muons that enable scientists to probe the structure and dynamics of condensed matter on scales ranging from the sub-atomic to the macro-molecular. The scientific domain covers soft condensed matter, biomolecular sciences and advanced material science.

In studying condensed matter, the user normally needs to visualise data obtained from an experiment together with some representation of the underlying structure or physical phenomenon that gave rise to the data. In addition, there are many techniques for simulation and modelling which nowadays 'interact' with neutron scattering experiments – Monte Carlo, molecular dynamics, *ab-initio* methods (such as those based on density functional theory) and finite element modelling etc. The most common use is simply to compare to an experiment as an aid to analysis and interpretation. These simulations can also be used to enhance the performance and analysis of the experimental results, even to plan them and to predict results.

As ever more ambitious structural and dynamical studies are tackled and greater volumes of data are collected in ever decreasing times, it is clear that the ability to visualise data and results in appropriate forms will be a key element in extracting information from the datasets. This includes the notion of "experimental steering", where the ability to visualise data and results during an experiment can have a significant impact upon the aims and outcome of the experiment. The issue here is whether a Grid/Web Services based methodology will deliver a near real time data analysis environment to the scientist's desktop.

2. Excitations Visualisation Environment (EVE)

The Grid is mainly recognised for its ability to support secure and dynamic access to compute and data resources. The e-Science challenge for the Excitations Visualisation Environment (EVE) project is to extend these aspects of the Grid for real time data analysis and visualisation, as shown in Figure 1.

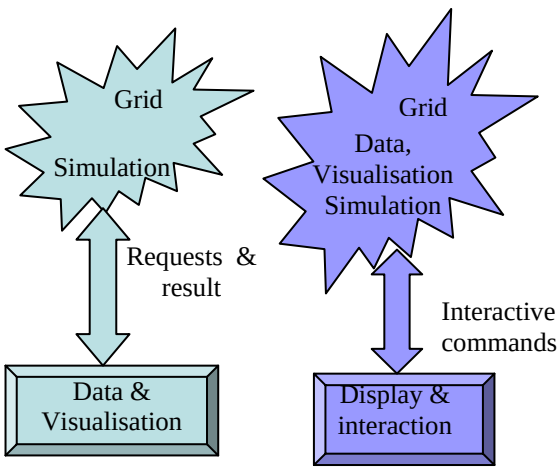


Figure 1: Conceptual overview of Grid enabled application visualisation environment

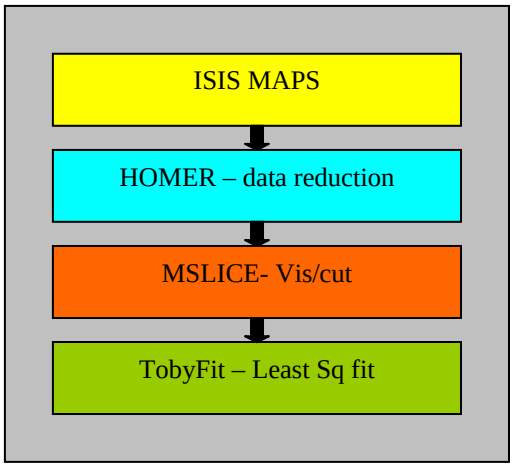


Figure 2: ISIS data analysis suite.

EVE consists of specialized application visualisation software tools based on Grid/Web Services and are developed around existing simulation and modelling software which often themselves require considerable computing power.

In a typical neutron scattering experiment data is gathered from an array of detectors that are positioned around the target sample. These record the number of particles scattered into a given solid angle and their energy distribution. This data is processed through a suite of programs, as shown in Figure 2. An instrument such as MAPS generates the data from an experiment. Homer is responsible for processing of the raw data from the electronics and writing this to formatted files. Mslice is a data visualisation program that also allows 1D cuts and 2D slices of the 4D data to be selected and saved to file. Finally TobyFit [2] is used to do non-linear least squares fitting of theoretical models to the data. This last program is the most computationally expensive part of the analysis.

One important requirement of the fitting process is that it needs to be highly interactive, as the user has to decide models to use, which parameters to vary, initial guesses to use, etc. It will also benefit from advanced visualisation interaction which will allow the user to directly manipulate parameters and visualise the resultant output.

3. Computational aspects

3.1 Numerical methods

TobyFit uses a standard non-linear least squares fitting algorithm based on the Levenberg Marquardt algorithm. The scientist provides functions that represent a parameterized form of the expected scattering function and background models. The aim of the software is to determine the best fit model parameters and check how accurate the model is. The calculation of the scattering is expensive since many multidimensional integrals must be evaluated, and this done using a Monte-Carlo method. Derivatives in the fitting process are determined numerically. Fitting is performed in an interactive manner to allow the best set of models to be selected. Good initial guesses for parameters are determined more quickly by first fitting to subsets of the total data.

3.2 Data requirements

Current experiments produce data files of perhaps tens of megabytes which can be dealt with on a local workstation. Newer instruments such as MERLIN, may produce output of 20 gigabytes or more. Such amounts of data would push the storage limits of workstations and the associated transfer and analysis times would be prohibitive. Such requirements are best dealt with on a central compute resources such as the Beowulf clusters provided by the National Grid Service[3].

3.3 Visualization requirements

The current software provides basic 1D graphs and 2D colour maps of the data. While these will continue to be useful when working with greatly increased data sets, it will be necessary to generate the data for these on the remote cluster and only return the minimum required data for visualization on the client.

Since the experimental data and the calculated fit to it are actually four dimensional, it is desirable to develop techniques to display the quality of the final fit. Another part of the EVE project is to develop effective ways of displaying this to the scientist. Again the main part of the computation for this will be performed on the remote resource, with the minimal data returned to the local client.

3.4 Parallelisation of the fitting process

TobyFit was developed as a serial code in Fortran 77. To exploit cluster technology on the grid it was necessary to make this parallel. To maximise portability this has been done using calls to the MPI library. OpenMP parallelisation is also being investigated, for shared memory machines, and may also be combined with MPI on certain architectures, such as those used in the NGS clusters.

The nature of the computations in the data fitting process is such that the balance of computation to communication is reasonably good if the problem is partitioned at the level of individual pixels of data.

Obtaining an efficient load balance across processors is made more difficult because of the way in which the Monte-Carlo integration is performed. An error estimate is used in the evaluation of each integral and the integration terminated when the requested level of accuracy is reached. Because the number of steps taken can vary greatly across the pixels within a given data file, and this distribution is very non-uniform, a simple mapping of pixels to processors gives poor load balance. This can be seen in Figure 3, where we show the speed up obtained on an NGS compute cluster using a simple mined mapping of pixels. Also shown in the figure is the speed up obtained in a simulation calculation, where the same number of Monte-Carlo (MC) steps is used at each point. The good scaling of both results as the number of processors increases indicates that communication is not a bottleneck, even for the very small test problems used in this case. The under performance that occurs when using variable numbers of MC steps will be addressed with a more efficient partitioning strategy.

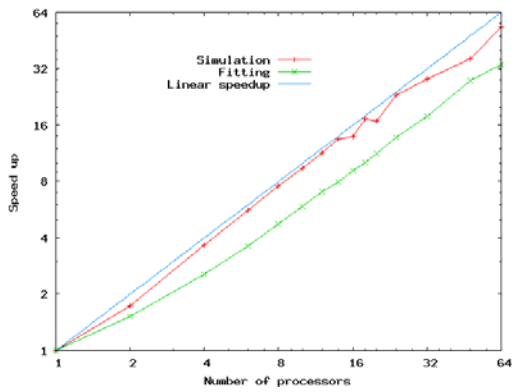


Figure 3: Speed up as a function of the number of processors used (log-log plot) for an NGS compute cluster.

As well making the fitting code parallel it is necessary to make it operate in a batch mode. This enables the front end client to build the commands to perform the fitting or simulation operation that is required, and then submit the complete operation to a suitable grid resource. The result of the operation will then be returned to the client via grid ftp.

4. Interactive simulation within a grid framework

4.1 Architecture of the EVE system

The current grid service offered on the NGS machines is based on Globus Toolkit version 2.4. Hence it is necessary to formulate the access to these resources in terms of these operations rather than as pure grid services. The interaction between the client GUI and the application visualization server can make use of web/grid services, and will provide a method to use GT2.4 methods to the backend. An outline of the architecture to be used in the EVE project is shown in Figure 4.

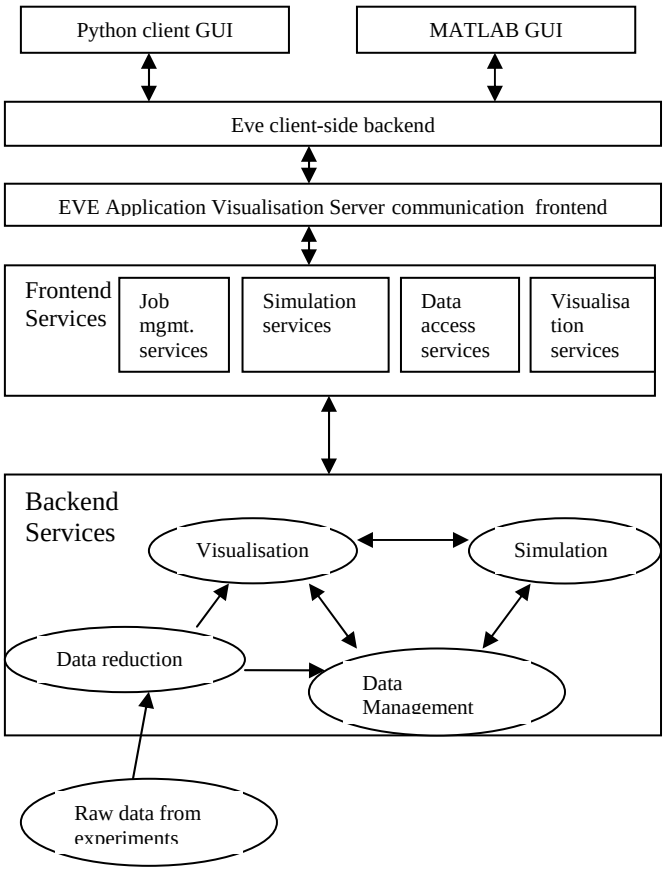


Figure 4: EVE architectural overview

The client interface is currently been developed in MATLAB since this tool is widely used by the instrument scientists at ISIS. In future a similar interface will be built using Python which offers greater flexibility.

The backend services will run on one or more grid resources, such as the NGS clusters. The data will typically reside on a server at ISIS

and will be moved using gridftp to the selected grid resource.

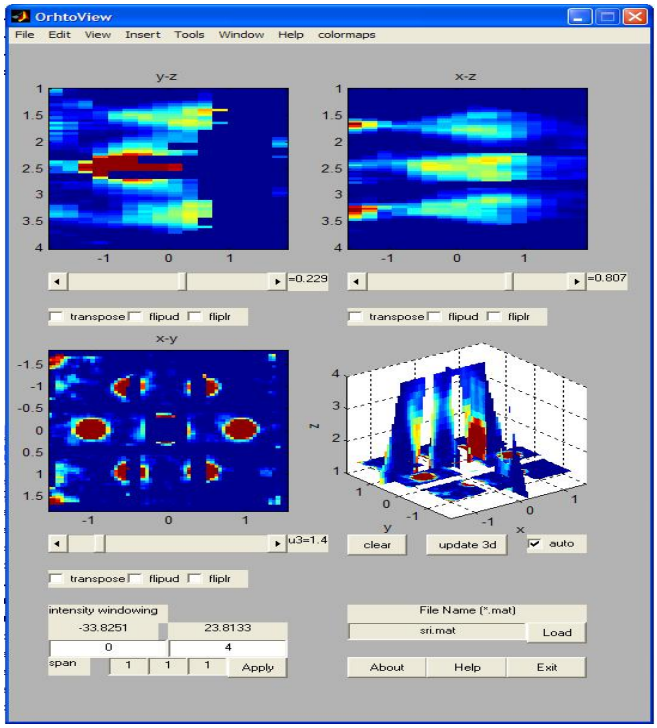
Since substantial grid resources will be used, and the experimental data may have a high level of confidentiality, it is important to make these operations secure. This will be achieved by requiring the user to have a valid X509 proxy certificate on the visualization server. In future the user certificate could be stored on a MyProxy server and only accessed from the visualization server. The Geodise Computational Toolbix for MATLAB [4] could be used to manage certificates and access to Globus on the client.

4.2 Resource brokering

A user will have access to a number of grid resources on which the backend analysis code can be run. Typically this may the four NGS clusters plus any other machines they have accounts on. The MDS will provide information on the current status of the batch queues of each of the available resources and this will be used to select most likely candidate to run a given request. In case the MDS information is out of date, or unavailable, the EVE frontend will resort to trying to run the job on each cluster in turn until a suitable target is found.

4.3 The MATLAB user interface

A typical example of the MATLAB interface to be used is shown in Figure 5. This shows the analysis of 2D slices of experimental data within the existing MSLICE component of the data analysis suite.



5. Conclusions

The EVE project is aimed at bring useful grid based computing directly to the scientists in their analysis of ever increasing amounts of experimental data. By working closely with the application developers and end users we hope to ensure that these tools make a real contribution to the work of ISIS. They should also have applications in similar scientific data analysis applications.

6. References

[1] http://www.isis.rl.ac.uk
[2] "Tobyfit Version 2.0: Least squares fitting to single crystal data on HET, MARI and MAPS", T.G.Perring, 2000. Available at http://www.isis.rl.ac.uk/excitations/documents/tobyfit.pdf.
[3] http://www.ngs.ac.uk
[4] The Geodise MATLAB toolbox: http://www.geodise.org/toolboxes/generic/toolbox_matlab.htm