Computational Data and Job Management for CCP1

CCP1 is a Collaborative Computational Project broadly centred around the computation of the electronic structure of molecules [1]. The project involves over 30 research scientists from UK academia, industry and government laboratories. The members form a Working Group which meets to review the project and plan future activities.

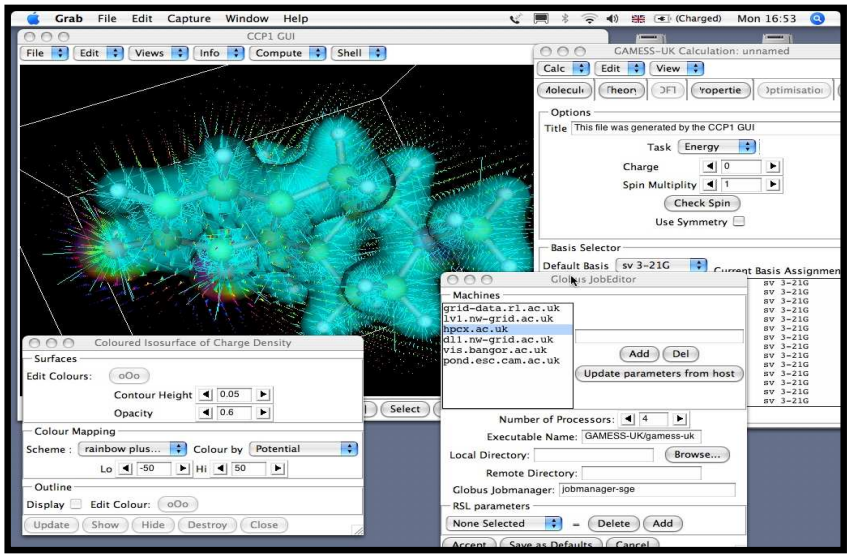

Figure 1: The CCP1GUI on Mac OSX

The members of CCP1 and their collaborators develop and maintain their own codes, such as GAMESS-UK and MOLPRO, but are also active users of a range of other codes (e.g. Molcas and NWChem). Many of these codes do not have their own graphical interface; something which is essential to enable new users to quickly adopt a code and visualise the results generated by their calculations. The CCP1GUI [2] was therefore developed to provide a single interface that could work with a range of codes and provide a familiar environment to visualise potentially highly complex data.

The CCP1GUI is developed primarily at STFC Daresbury laboratory, but is hosted on Sourceforge and is freely available to everyone.

Written in Python, it is quicker to develop than programs written in lower-level languages and will also run on all modern operating systems. Additionally, Python has the advantage that users can adapt and script the program for their own requirements.

Through our collaborations with the NW-GRID [3] and the UK's National Grid Service [4], we have considerable experience of working with computational grids. Although providing standardised access to powerful resources, grid technology is still rather complicated to use, and this is proving a significant barrier to its adoption. Taking the principles behind the CCP1GUI a step further, we have recently extended it to be able to submit jobs to different computational grids. The current version of the CCP1GUI can submit jobs to resources running Globus (via the GROWL [5] toolkit), to a Nordugrid

resource [6] or to the eMinerals infrastructure [7].

These developments allow users of the CCP1GUI to focus their attention on the calculation they are interested in, rather then the details behind using different computational codes on a range of local and remote resources.

As it has interfaces to several different codes, the CCP1GUI must be able to translate data between them. Exchanging data between codes is an awkward, but increasingly common problem, as scientists often make use of several applications in close cooperation to address complex scientific problems. There are many examples in the biological domain, where the use of hybrid methods, such as QM/ MM, are not uncommon. Therefore, the advent of Grid computing and software for specifying and executing workflows promises exciting opportunities. Unfortunately, it is very difficult to realise these opportunities; a critical lack of data standards hinders the automatic exchange of information between applications. Although workflows can be specified and executed, bespoke converters and wrappers need to be developed to deal with the transformation of any data that must be exchanged. The development of such converters is a time consuming and complex process, often involving several expert application developers. The same problem arises in the development of common tools for analysing and visualising the results from simulations, such as with the CCP1GUI. In the worst case, specific software components must be developed to handle the data generated from each application. Again, this requires a detailed understanding of each application's data model. Recent efforts to address this problem focus on the development of common data models, as exemplified by an increasing number of XML based standards for scientific data. However, there are a number of difficulties in finalising these standards, both technical and social.

The AgentX framework aims to support the exchange of information between scientific applications without a need to complete the standardisation process. The framework allows information to be extracted from different data sources (such as XML and HDF5 documents and relational databases) through a series of queries based on terms in an ontology. AgentX includes a library that can be used directly by scientific applications and that presents an API for extracting data based on its meaning. This frees the scientific application developer from having to understand the interfaces and underlying data models of specific data sources. In summary, the AgentX framework aims to facilitate the inter-operability of applications that form part of complex workflows and to promote the development of common tools that can be used to analyse and post process scientific data.

References

[1] CCP1: http://www.ccp1.ac.uk/
[2] CCP1GUI: http://www.cse.scitech.ac.uk/ccg/software/ccp1gui/
[3] NW-GRID: http://www.nw-grid.ac.uk
[4] NGS: http://www.grid-support.ac.uk
[5] GROWL: http://www.growl.org.uk
[6] Nordugrid: http://www.nordugrid.org
[7] eMinerals: http://www.eminerals.org