

# The eMinerals minigrid and the National Grid Service: a user's perspective

A. Marmier

Department of Chemistry, University of Bath, Bath BA2 7AY

## Abstract

In this paper, I discuss my experience in using these two eScience resources as a computational modeller. I focus on the main difficulties I encountered both in my work and in encouraging colleagues to “take on” the grid : Access, Transferability, Support. I argue that, while being part of a VO is still almost a prerequisite for successfully running eScience, a small effort in providing clear visible documentation and sharing working examples will make it accessible to a greater part of the community.

## Introduction

The eMinerals project established its minigrid two years ago as a realistic test bed and pedagogical device as well as a practical distributed tool for the Virtual Organisation.

More recently (April 2004), the National Grid Service (NGS) was put online. The NGS managed resources (4 large Beowulf clusters) are significant and provide the UK eScience community with excellent computational power and data storage facilities.

In essence, the minigrid is, at least partially, a prototype for a much wider grid, of which the NGS is a good approximation.

In this paper, I compare the two resources from the point of view of this most elusive entity, the “consumer” of eScience, in other words the basic scientist, of which the author consider himself a reasonable approximation. My aims are to identify/illustrate the difficulties in “using” (by which I essentially mean running complex, parallel, computer simulations)

eScience resources and discuss simple ways to alleviate them and make eScience an easy and attractive option for most computational research. I mostly argue that an effective use of these eScience resources still requires a lot of direct request/or question, and that (combined with a lack of documentation for some tasks) could do a lot to put off prospective users.

## User Profile

### *Expertise*

The author's assertion that he is the kind of basic scientist likely to use the computational resources offered by the GRID cannot really be backed up “scientifically”, but is based on circumstantial evidence. After ten years (PhD and various postdoctoral positions) in the field of computational modelling of surfaces processes and quantum chemistry, he has interacted with, and observed, many similar colleagues. Of course, the average scientist is a misnomer, as his/her range of competence can vary considerably.

But for the sake of clarity let us assume the following range of (relevant to the discussion) competences:

- More or less advanced MS Windows know-how
- More or less advanced Unix/Linux know-how
- Some scripting language knowledge
- Ability to submit and monitor complex parallel jobs to various queuing systems with a provided submission scripts (i.e. NO real knowledge of the queuing system's syntax)
- Ability to install/compile parallel codes, assuming a friendly Makefile (or equivalent) AND a good documentation

### ***Requirements***

Day to day, routine submission is a technically easy task. The difficulty arises with changes: change in code (and therefore submission parameters), change in resource. Of course, the nature of eScience makes changes in resources more common, and adapting one's submission procedures more important a skill. Obviously this adaptation phase is all the more relevant and difficult the first time one is confronted to a grid-style, remote machine submission.

Anything that will ease the adaptation of submission procedures is highly desirable (on a related note, the same can be said for compilation procedures)

### **Resource description**

#### ***eMinerals minigrid***

The *eMinerals minigrid* includes compute and data resources as well as the software and middleware, which enable their use. The compute resources are both heterogeneous (Beowulf

machines, IBM SP's, Condor pools and many Linux boxes) and distributed across the sites linked with the project. The backbone is constituted of three identical 16 processors clusters located at Cambridge, Bath and UCL. The data resources include a dedicated database server to hold the metadata catalogue and multiple SRB servers.

It is built on the Globus Toolkit 2. It can be accessed either via a custom *eMinerals* portal (including integrated versions of the CCLRC Data Portal and the CCLRC HPC Portal) or by using *globus* or one of its wrapper (like *condorG*) from a Linux Box. Direct access (by *gsi-ssh*) is only allowed to the Cambridge cluster, for tests and compilation purposes. The queuing system for the three clusters, on which I will focus the discussion, is *openPBS*.

The Storage Resource Broker (SRB), version 3.2.1 (upgraded from version2) is used to facilitate data management and sharing across the VO. The SRB allows files and data stored over multiple distributed heterogeneous physical resources to be accessed through a single virtual file system.

#### ***Usage summary***

Early on, it was decided to use *condorG* (the *condor* user friendly *globus* wrapper) instead of *globus* for the remote submission. There are two main reasons for this choice. First of all, the team was familiar with the *condor* system, having been testing the University College London Condor pool, and had been exposed to the syntax of the *condor* submission scripts. Secondly, *condor* comes with its own easy to use workflow manager, *DAGman*, which proves very convenient to manage the file transfer (input and output) part as well as the job submission part of the task. Finally, the script format (as opposed to a command

line for globus) lends itself very well to the inclusion of specialist commands in the script itself.

Indeed, in order to hide the gory details of the SRB transfer, a wrapper for the workflow was developed. It takes as its argument a condorG script but also recognises SRB specific instructions, as can be seen in Fig. 1.

```
Universe = globus
Globusscheduler =
  lake.bath.ac.uk/jobmanager-pbs
Executable =
  /home/arnaud/bin/vasp-lam-intel
Notification = NEVER

Environment = LAMRSH=ssh -x
GlobusRSL = (job_type=mpi)
  (queue=workq)(count=2)(mpi_type
    =lam-intel)

Sdir =
  /home/amr.eminerals/run/TST.VASP3
Sget = INCAR KPOINTS POSCAR
  POTCAR
Sput = *
SRBHome =
  /home/srbusr/SRB3_3_1/utilities/bin

Output = job.out
Log = job.log
Error = job.error

Queue
```

Fig. 1: My\_condor\_submit script example. The non-CondorG commands are italicised.

Assuming that the four required input files (INCAR, KPOINTS, POSCAR and POCAR) have been uploaded on the SRB in the location pointed at by *Sdir*, this script is sufficient to transfer the inputs, submit the job to the PBS queue and then retrieve the wanted outputs, in this case everything.

In fact, my\_condor\_submit creates a DAG launching three condorG

scripts. The first and last creates two perl scripts and launch them on the remote machine, these are in charge of the transfer from and to the SRB respectively. The second effectively submits the main job to the PBS manager.

Further comments need to be made on a subtlety concerning the MPI libraries. As various codes are more or less adapted to various MPI flavour, two were implemented, MPI and LAM. Therefore some environment variables need to be set up by the user, and the MPI flavour need to be chosen in the RSL line, with the non-standard mpi-type option.

### *Support*

User support is essentially achieved by informal exchanges (a practical solution given the size of the VO), either via email, certificate protected wiki or AccessGrid, but a formal helpdesk is also implemented. A public FAQ is also accessible via the eMinerals website.

### *National Grid Service*

The NGS (funded by JISC and CCLRC) is comprised of two data clusters (located at Manchester and CCLRC) and two compute clusters (Oxford and Leeds). These compute clusters, with 64 dual processors are excellent machines (queuing system PBSpro). CSAR and HPCx are also part of the core NGS. Recently, machines at Cardiff (Condor) and Bristol (SGE workload manager) have joined it, but as I have not tested them yet, I will only cursorily mention them.

### *Usage summary*

While eScience tools (SRB, globus) are enabled, many of the more traditional approaches (direct login to the machines via gsi-ssh, direct file transfer via gsi-ftp) are still supported. As far as I am

aware, the only compulsory eScience approach implemented concerns the use of X.509 certificates. All the examples for remote submissions are given as globus commands.

As far as I can tell, the only MPI library supported is MPICH, which actually simplifies things.

### *Support*

Direct user support is excellent and provided by a dedicated mailing list, and a helpdesk. Documentation is somehow lacking, but this is partially alleviated by the quality of the helpdesk.

### *To summarise the above*

The two resources share many commonalities:

- require a globus/gsi-ssh enabled desktop (or access to such a remote machine)
- use of certificates, and my-proxy
- emphasis on direct support

In practice they are very different:

- the NGS seems to promote the direct use of Globus
- eMinerals promotes condorG
- the NGS allows the use on non-eScience protocols
- eMinerals strongly discourages these and put the emphasis on “the eScience way”

## **Discussion**

In this part, I will only discuss what is not easy to use. I must point out that a lot of the eScience tools are very user friendly! Of course, a few hurdles are enough to create an impression of difficulty and discourage potential newcomers.

## **Access**

Simply accessing either the minigrid or the NGS is already a surprisingly difficult task, significantly more so than accessing a standard (i.e. non grid resource).

First of all, there is the requirement of obtaining a certificate. While this is now well documented, there can be complexities involved for establishments without RA. Furthermore, it is not always possible to obtain certificates for short-term visitors, for instance. The eMinerals policy of issuing internal certificates, only accepted on its minigrid, is a way to circumvent problems, but still requires interaction and request.

An unfortunate feature of essential eScience middleware (globus and gsi-ssh to name them) is that they are difficult to install, and at this stage, beyond the ability of the research scientist otherwise reasonably proficient in Unix and other traditional tools. Which means that either these members of a VO with related expertise can install these softwares on their colleague’s desktop, or that a suitable remote machine can be provided. This second solution (the only one for non-VO linked individuals) requires separate request for an account.

Additionally, SRB accounts and rights must be set up separately by a different authority again.

In practice, this means that it is difficult to access eScience outside a VO.

This being said, the problem of accessibility is well known, and I understand that actions are being taken to make the installation of grid middleware much simpler.

## **Usage/Transferability**

In order to run a complex parallel code on a new resource successfully, a

scientist needs to submit it to a batch queuing system, which does involve transmitting parameters either via a command line, or more often via a submission scripts. There exist quite a few different batch queuing systems (LoadLeveler, OpenPBS, condor...) with possibly different implementation, and subtleties and idiosyncrasies. As already mentioned, to my knowledge few research scientist are routinely writing their own submission scripts from scratch, as these are usually inherited in an almost ready-to-go form and modified as little as possible so that to ensure successful submission. The mode of acquisition of these scripts vary:

- they are sometimes provided by the system manager in the root directory of each user
- they are often obtained from sympathetic and knowledgeable colleagues
- they are very often “borrowed” by having a look in a working directory of an accessible account! (despite the moral ambiguity of such a practice).

Due to the nature of eScience, and to stricter access right to files, this last approach is not permissible. This point is actually quite important, as it adds to the burden of the user and forces him/her to have to learn a new skill, submission script syntax, on top of the genuinely eScience new paradigms. This is further compounded by the fact that there are many variants and implementations of queuing systems.

Examples of scripts can be found for some NGS resources, albeit not all, and they are relatively well hidden. Whether or not this is a deliberate decision to favour the adoption of remote submission (via `globusrun`) I cannot tell. It seems the case, as examples of

`globus` command line submission are certainly more abundant in the NGS documentation.

Nevertheless, once an example of a relevant submission script has been obtained, the traditional, basic and widespread “submission script know-how” is sufficient for using the NGS resources (by direct login via `gsi-ssh`).

Secondly, and more importantly, I have found that the task of translating a batch queuing system specific scripts to a form acceptable by Globus or CondorG (essentially the environment and RSL parameters) to be of major difficulty, and one that is poorly documented, as most of the few `globus` examples given are for simple, non parallel tasks.

I must add that the `my_condor_submit` wrapper was relatively easily portable from the eMinerals minigrid to the compute machines of the NGS. The only difficulty was that the path for the `Scommands` was hard-coded for the eMinerals clusters but had to be acceptable as an option (see Fig. 1, `SRBHome`).

### *eScience Scope*

Considering the difficulty of adapting one’s submission protocols to different eScience resources, it is not surprising that, given the option of using a resource “the old-fashioned way”(simply by direct login via `gsi-ssh`), one could be tempted not to bother with `globus` or CondorG. To the best of my knowledge, nothing directly perceivable (beyond technological knowledge) is gained by using the eScience option instead: codes are not faster if submitted via `globus`. This is unfortunate, and might slow down the early adoption of eScience good practices. There should be a mechanism

to ensure that there is an incentive in using eScience tools to remotely submit jobs (be it access to a better queue, or better priorities or something else), and this bias towards “the eScience way” should be publicised to favour awareness and adoption.

## Conclusion

I first postulate that in order to use a resource, most users would rather like to have to ask as few questions as possible, and to have to deal with as few as possible authorities.

With this in mind, I think that I have identified three main limitations to the adoption of eScience methodologies by computationally oriented researchers, especially those who do not benefit from the support of a VO:

- First of all the high number of request to different organisation needed just to start
- Secondly, the lack of a very good, highly visible documentation on how to install globus (as this is a *sine qua non* step)
- Finally, the complexity and subtlety of translating into submission scripts into the RSL syntax

While I have no simple solution to the first problem, and am confident that the second will be solved in the near future, I can suggest that a well-organised repository of working condorG or globus scripts would be immensely useful to the community.

Finally, at this stage, I found that being part of a VO is practically mandatory in order to access and use eScience resources, but that I are not very far from “off the shelves” solutions, by providing visible, user focused documentations and rich lists of working examples.

## Acknowledgements

This work was funded by NERC via the grant NER/T/S/2001/00855.