LSST Case Study for GridPP

Wednesday 8th June 2016

George Beckett (Edinburgh); Joe Zuntz (Manchester); Alessandra Forti (Manchester); Marcus Ebert (Edinburgh)

Document history

15/APR/ 16	MGB	Produced first draft
21/APR/16	MGB	Updated draft in light of comments from JZ
25/APR/16	MGB	Updated draft in light of comments from ME and AF
25/APR/16	MGB	Added further info on VOs, as requested by RGM
27/APR/16	MGB	Added conclusions and options for further work, plus notes about client tool setup and support
11/MAY/16	MGB	Made further changes, based on feedback from AF and ME.
8/JUN/16	MGB	Update to description of hybrid Ganga/ Dirac approach, as advised by Ulrik Egede; noted individual effort contributions to project.

Background and motivation

GridPP is a distributed, national computing infrastructure that supports computational particle physics and, in particular, LHC science. At the time of writing, GridPP has 19 sites with a total computing capacity of ~40,000 cores, along with 40 Petabytes of disk-based and 40 Petabytes of tape-based storage capacity.

While the GridPP infrastructure primarily serves LHC science, 10% of the resources are available to researchers from outside of particle physics. This equates to 35 million core hours per year.

A number of UK physics and astronomy groups have established an initiative called UK Tier 0 to consider the possibility of a future, shared computational infrastructure for some or all of the groups' computational requirements. Within the UK Tier 0 activity, a galaxy-classification workflow has been identified as one of a number of pilots to trial use of the GridPP infrastructure (where GridPP has been chosen because of its analogue to a possible shared-infrastructure model).

The galaxy-classification workflow has been proposed by Joe Zuntz (Manchester) in his work to progress understanding of weak-lensing techniques that could be used to probe the nature of dark energy exhibited on observations from the Dark Energy Survey (DES) and—in the future—the Large Synoptic Survey Telescope (LSST).

Definition of the problem

Joe's initial workflow involves the classification of the shape of 100 million galaxies that have been imaged by DES. Each galaxy can be classified independently of the others, making the workflow *embarrassingly parallel*.

There are potentially tens of images of each of the 100 million galaxies to be classified—for example, representing imaging in different bands of the electromagnetic spectrum. These images are distributed across ~30,000 files, though with all images of a particular galaxy being held in the same file. At the outset of the pilot, the image files were held on networked storage at NERSC (California) and Brookhaven National Laboratories (New York).

The shape classification is undertaken by an application called *im3shape*, which has been developed by Joe in Python with computational kernels in C, C++, and FORTRAN. Using *im3shape*, the classification of a galaxy involves 10—20 seconds of computing time on a modern CPU.

The Porting Process

The process of porting Joe's workflow to GridPP involves seven steps:

- Enabling access for Joe to (one or more) GridPP sites;
- Setting up a client computer, from which Joe would access GridPP resources;
- Installing the *im3shape* software on GridPP;
- Staging the input (image) files onto GridPP storage;
- Running the shape-classification workflow;
- Retrieving the output (and logs) from GridPP;
- Obtaining support from GridPP operational staff,

—which are documented in turn below.

At the time of writing, a rudimentary implementation of the above steps has been completed, allowing Joe to execute the workflow on GridPP with modest manual involvement.

Obtaining access

Access to GridPP resources is controlled at the level of science groups, represented by *Virtual Organisations (VO)*. A Virtual Organisation is a grouping of users that are in some sense connected, with regard to the use of an infrastructure—for example, they may be involved in a particular project, or work at a particular institution. A Virtual Organisation is typically managed by a responsible person from the particular group—for example, a project manager or an institutional administrator—making it straightforward to determine who should and should not be a member of a particular VO. Grid resource providers can then make decisions with regard to access and resource allocation for each VO, without needing to concern themselves with the particulars of who is part of the VO. The concept of a VO is usually encountered in conjunction with a (grid) certificate, which is used to confirm the identity of a user in the context of a grid infrastructure. A certificate is analogous to username and password credentials, though is more suitable for use in a distributed (multi-institutional) infrastructure.

A suitable VO has been identified—LSST—and Joe's credentials have been added to this VO. Further, a number of GridPP sites have enabled access to their resources for the LSST VO.

Client setup

There are typically two options to set up a client computer from which to access GridPP. The usual option is to install a set of GridPP-provided client tools on one's local computer. This is relatively straightforward. However, the client tools require a specific version of the Linux operating system (at the time of writing, this is Scientific Linux/ Red Hat Linux Version 6). If the intended client computer does not run this specific operating system, then the second option–to use a virtualised, client environment (CernVM)–is better, accessed as a remote desktop session across the Internet.

For this case study, a variant of the first option was selected – in that Joe could have access to a local machine, at his institution, which had the client environment configured. The selected machine is actually intended for administrator use, rather than science: as the port progressed, Joe encountered minor problems due to lack of local disk space on the client machine. However, this was not considered a sufficient obstacle to merit migrating to a different client machine.

Software installation

The recommended mechanism through which to install user software on GridPP is the CERN-VM File System (CVMFS), which can easily be accessed from any GridPP (or, more generally, WLCG) site. However, there is a delay of several hours between software being uploaded to CVMFS and it then

being visible to GridPP sites. This makes it problematic to use CVMFS during software development or porting—when the software package can change frequently. Because of this, Joe has developed a workflow that copies the *im3shape* software to sites as part of the staging in of data for compute jobs.

Staging input files

A large portion of the image files (held at NESRC and BNL) have been copied onto suitable GridPP storage elements. The process for doing this was refined over several iterations, eliminating initial problems related to: inserted files not being registered in the GridPP file catalogue; and lack of clarity as to which storage element was best suited to host particular files. A scripted process has been developed to support transfer of files directly from NERSC into GridPP Storage Elements, with destinations being selected from available Storage Elements to share demand and reduce the load on any individual Element. For BNL, a more typical transfer approach has been implemented, whereby files are first read out to an intermediary server and then inserted into GridPP from there.

Running a workflow

Two interfaces to GridPP were considered for handling input/ output data and for submitting/ managing compute jobs, called Dirac and Ganga. Dirac is a client toolkit with functionality for staging data into and out of storage elements, for submitting and monitoring compute jobs, plus other relevant housekeeping tasks. It had been proven to work reliably with very large job numbers for the LHCb experiment. Ganga is a higher-level tool that simplifies the execution of common grid workflows. It has interfaces to several client toolkits, including the Dirac toolkit, plus includes a native low-level client toolkit.

At the outset, Ganga was favoured, but using direct job submission (rather than high-level submission functionality)—that is, using the native client toolkit. This approach suffers a limitation for long-running jobs, due to a limit on maximum job lifetime (more specifically, the lifetime of the short-lived proxy certificate that is created to authorise the job). To avoid this limitation, the workflow was then ported to Dirac (which has a proxy renewal mechanism to allow longer job lifetimes). A job (shell) script was developed using the Dirac command-line interface, which could successfully submit collections of jobs to GridPP compute elements. A Dirac web client could then be used to identify failed jobs and to resubmit them.

Later on in the pilot, an alternative, hybrid approach was proposed using Ganga as an interface to the Dirac client—with individual galaxy-classification jobs being collected into hierarchies of work (called Ganga Tasks) that were sized to fit to available computing resources and job-lifetime windows. The Ganga/ Dirac hybrid approach also allowed the workflow to be initially tested on local resources before migrating to GridPP, allowed collections of jobs to be submitted to GridPP compute elements, and automated the detection and resubmission of failed jobs. Further, the hierarchical organisation of work dramatically reduced the number of jobs that needed to be managed individually, making it a more attractive option for Joe.

Retrieving results

For *im3shape*, each galaxy-classification job produced a modest amount (a few kilobytes) of text output that were straightforward to stage out of compute elements at the end of the computation. Recovery of output results was achievable using standard Dirac and/ or Ganga client commands. Further, the hybrid approach (described above) provided a flexible way to handle and merge the output data.

Accessing support

The usual support process was not evaluated as part of this pilot: as a GridPP incubator project, Joe was paired with a specific GridPP expert (Alessandra Forti) for the duration of the activity. To progress beyond the incubator stage, Joe will obtain support from GridPP by lodging enquiries with the GGUS ticketing system.

Effort and timescales

The porting activity ran during the second half of 2015 and the first quarter of 2016, though the majority of porting work was undertaken in the last three months. During this time, Joe estimates he invested around 14 days of effort for his contribution to the activity. This was complemented by around 25 days of effort from the GridPP team.

Conclusions and further work

This pilot is considered to be a success in that, at the completion, Joe was able to progress his research – to classify the shape of galaxies in the DES survey – effectively using GridPP resources and significantly faster than was the case previously (for example, using highly contended HPC resources at NERSC). Following on from the pilot, Joe has proposed to continue using GridPP to support his scientific work as a typical non-LHC user.

The time to complete the port (estimated to be around 4 months) is longer than would usually be acceptable for a typical LSST researcher, though the effort involved on the part of the researcher (around 1—2 weeks) seems appropriate. The effort required from the GridPP team (25 days) is more than could be reasonably expected in general. It is hoped that the experience from this pilot will both reduce the effort requirements and the time for a future porting activity.

For LSST science, several important requirements have been intentionally avoided in this first pilot, though should be introduced in a follow-on activity. First, much of the analysis undertaken in relation to LSST (and other telescopes) involves access to third-party catalogues (usually presented as relational databases). The setup of database client tools on GridPP and the connection from a GridPP site to a remote catalogue are both potential obstacles to be overcome. Second, astronomy analysis tasks often produce significant volumes of output, which then needs to be deposited in third-party catalogues or repositories. Both the capacity to hold voluminous outputs on GridPP and to efficiently stage them to a remote repository are aspects of the workflow that would warrant further investigation.

Acknowledgements

Thanks are due to GridPP engineers, for their significant assistance in completing this pilot. This work was supported by the European Research Council in the form of a Starting Grant with number 240672 and by The Science & Technology Facilities Council (STFC), grant number ST/N002512/1.