QuakeSim Grid Services on Columbia

Marlon Pierce and Yili Gong for the QuakeSim team
(PI: Andrea Donnellan, JPL)

Abstract

This short paper discusses our plans and requirements for Grid services on Columbia and possibly other NASA computing resources. We are working to develop a general purpose Grid provisioning system that can be used to manage long-running "high throughput" collections of simulations across NASA, NSF TeraGrid, and international Grid installations. The results of these simulations will be made available through data services accessible through both human and programming interfaces.

Scientific Use Cases

The GeoFEST application is a parallelized finite element method for simulating forces produced by crustal faults using realistic material models. GeoFEST's most prominent current usage is to calculate of surface deformations that can be compared to InSAR satellite observations, allowing for the refinement of fault models. Work is also underway to use GeoFEST as a source for Green's functions that can be incorporated into UC-Davis's Virtual California simulation of interacting fault systems. GeoFEST itself uses pre- and post-processing codes that may also require high performance computing.

Comment [MSOffice1]: Jay could give more /better detail here if necessary. Also, I don't have an estimate for the total number of hours required to actually do all of these simulations.

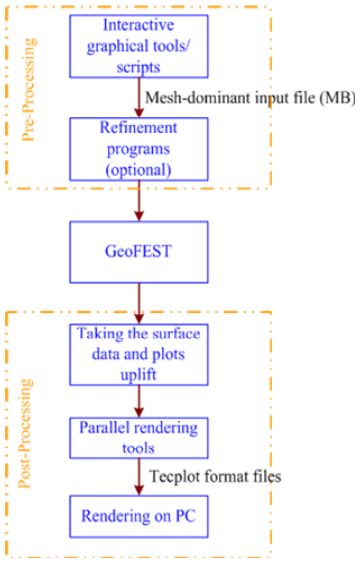

Figure 1 illustrates the GeoFEST workflow sequence, which includes several steps for pre-processing (creating and refining finite element meshes) and post-processing (for visualization) as well as GeoFEST runs themselves.

Currently, QuakeSim is designed for Web portal users to set up and launch individual GeoFEST runs. However, the two use cases described above can be run ahead of time, allowing portal users (and other Grid applications in workflows) to query for results of pre-run GeoFEST calculations rather than launching new calculations on demand.

In short, we want to run GeoFEST to obtain both surface displacements and Green's functions for all faults in the western United States. These results will be stored in catalog Web services.

Proposed Grid Infrastructure

We view the GeoFEST case as an important example of a more general problem: many computational results are general in nature and can be done ahead of time. The general issues are planning/provisioning resources to execute the workflow, managing the simulations, and storing and cataloging the results. Standard Grid technologies (Globus, MyProxy, Condor-G) can be used for this purpose. The GeoFEST workflow shown in Figure 1 is relatively simple, making it a good test case for our planned system before we attempt more complicated scenarios. The general architecture is shown in Figure 2.

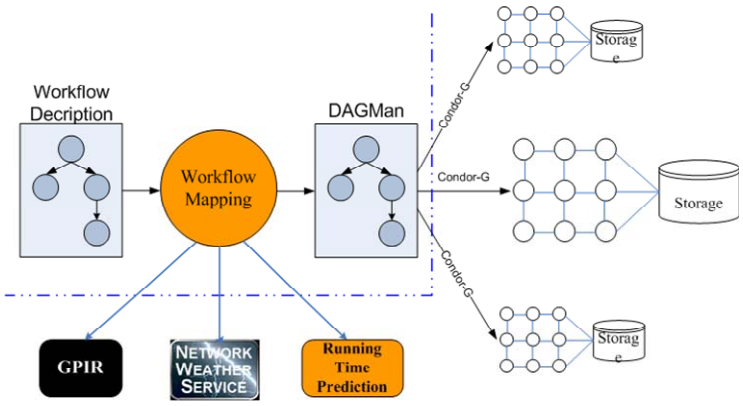

Figure 2 shows the general provisioning architecture.

The basic components of the system are a workflow management system (initially, Condor's DAGMan), Grid clients (Condor-G), Grid services (Globus), information services (such as the TeraGrid's GPIR and the Network Weather Service's Batch Queue Prediction Service). The provisioning and managing service uses these components to find best available computing resources for a particular GeoFEST run, launches and monitors the application on the Grid resource using Condor-G, and moves the results (along with metadata descriptions) to a storage service. The actual job execution is launched by Globus GRAM's job manager, which maps to the local queuing system (PBS, typically). Note that the Grid resource chosen can be in one of

several different Grid installations (NASA, TeraGrid, etc), so the provisioning system will also need to acquire the appropriate credential from a MyProxy server.

Note again that this problem can be viewed as a very large "parameter sweep" style problem, where the parameters are all fault models from the QuakeTables fault database and other sources.

One important simplification that we make is that we do not attempt to support an arbitrary number of Grid users of the system in Figure 2. Scientists would not directly interact with the provisioning and execution services and would not access computing centers' resources directly. They would use the results stored in data services. Thus we only need to obtain a few QuakeSim "system" accounts and allocations for any particular computing system that we wish to use.

Requirements for NASA

Because the system must work with several different Grids, we try to keep the requirements as generic as possible.

- Globus: NASA would need to install the Globus 4.0 toolkit and run GRAM and GridFTP services. Either the "pre-Web Service" or "Web Service" versions of these services are acceptable. These would need to be accessible by a few external but well-defined clients (i.e. we can give you specific IP addresses and port numbers).
- Job Manager Support: the Globus job manager for Columbia's queuing system (PBS?) would need to be checked and tweaked.
- Certificate Authority: NASA would need to set up a standard Globus Certificate Authority and use the standard Globus security configurations (i.e., a grid-map-file is used to convert certificate Distinguished Names into local UNIX accounts).
- One Time Password Support: the system would be run by a QuakeSim developer, who would get a Grid credential using a one-time password. This operation can be done manually by the developer. This credential would then be stored in a MyProxy server and can be accessed without developer intervention for a set amount of time (by default, one week). A standard MyProxy server can be used, and it can be run either by NASA or by the QuakeSim team.
- System information services (such as Network Weather Service Batch Queue Predictor) need to be installed and accessible from outside NASA. Alternatives information services include GPIR (from TACC and the OGCE project) and Globus MDS.

Note other parts of the system shown in Figure 2 (such as Condor-G) do not need to be installed by NASA. Condor-G acts as an external client to the requested Globus services.