

# SunDCG

Sun Data and Compute Grids  
- Going Global with Globus and Grid Engine –  
[www.epcc.ed.ac.uk/sungrid/](http://www.epcc.ed.ac.uk/sungrid/)

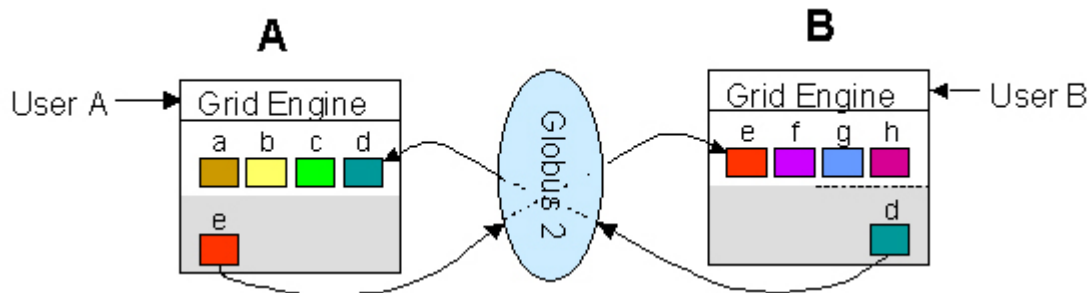


Figure 1: By configuring queue e to use the Transfer-queue over Globus (TOG) software, Enterprise A can access resources at enterprise B. Similarly, enterprise B can access resources at enterprise A by configuring queue d to use the TOG software.

## Overview

The SunDCG project aims to develop an industry-strength, fully Globus-enabled compute and data scheduler based around Grid Engine, Globus plus a wide variety of data technologies.

The project started in February 2002 and will run until January 2004. The partners are the National e-Science Centre, represented in this project by EPCC, and Sun Microsystems.

Sun Grid Engine is a distributed resource management system that allows the efficient use of compute resources within an organisation. However there is some desirable Grid functionality that Grid Engine does not provide.

Sun classifies Grids at three different levels:

Cluster Grid - a single team or project and their associated resources.

Enterprise Grid - multiple teams and projects but within a single organisation, facilitating collaboration of resources across the enterprise.

Global Grid - linked Enterprise and Cluster Grids, providing collaboration amongst organisations.

## Grid Engine

Grid Engine meets the first two levels by allowing a user to transparently make use of any number of compute resources within an organisation. However, Grid Engine, alone does not yet meet the third level.

The Globus Toolkit is essentially a Grid API for connecting distributed compute and instrument resources. Integration with Globus allows Grid Engine to meet this third level. That is, it allows collaboration amongst enterprises. The project aims to develop a scheduler based on Grid Engine that allows user jobs to be scheduled across a global grid and allow these jobs to have access to their necessary data sources. Globus will be used as the Grid API to provide secure communications.

As a first step, a compute global grid scheduler has been developed. This integrates Grid Engine V5.3 and Globus Toolkit V2.2.x to allow access to remote resources.

This integration is achieved by use of the Transfer-queue Over Globus (TOG) software developed by the project.

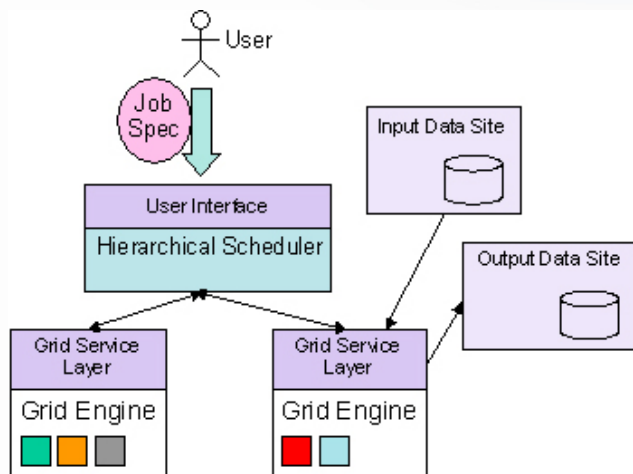
Figure 1, above, illustrates how an enterprise can access remote compute resources at a collaborating enterprise.

This is achieved by configuring a queue on a local Grid Engine to use the TOG software. The TOG software provides secure job submission and control functionality between the enterprises. The software enables an enterprise to schedule jobs for execution on remote resources when local resources are busy. Data and executables can be transferred over to the remote resource with subsequent transfer of results back to the local installation.

This global grid scheduler uses the Grid Engine interface for job submission and control. Remote system administrators still have full control of their resources.

The TOG software has been used to create a compute grid between the universities of Glasgow and Edinburgh. Researchers at the Glasgow site of the National e-Science Centre have been able to access compute resources at EPCC using a Grid Engine installation configured with the TOG software.

TOG is also being used to set up a biomedical e-Science demonstration using the new SRIF network linking three sites within the University of Edinburgh – EPCC, the Scottish Centre for Genomic Technology and Informatics (GTI) at the New Royal Infirmary of Edinburgh and the MRC Human Genetics Unit (HGU) at the Western General Hospital.



Following the development of the TOG software, the project has investigated the integration of access to data sources via data grid technologies such as OGSA-DAI, GridFTP and SRB.

The next step for the project team is to develop a hierarchical scheduler that scales better in a grid environment and enables access to remote data sources via data grid technologies.

As illustrated in Figure 2, below left, this will use an OGSA-compliant Grid Service interface to Grid Engine.

The hierarchical scheduler will query child Grid Engine installations at collaborating sites to determine if they are able to run a user's job.

The scheduler will then place the user's job at the site that best matches the following criteria.

- It is capable of running the job.
- It has the lowest load of the available sites.
- It has the best access to the required data sources.
- For those user jobs that are not data grid aware, a data component will handle the transfer of data between sites.

### Further information

For more information on the project and its deliverables please access the project web site at

<http://www.epcc.ed.ac.uk/sungrid/>

