JENNIFER M. SCHOPF AND KEITH R. JACKSON

So You Want to Set Up a Grid

Building a Grid is no simple task -- it takes planning and coordination. This column discusses some rules of thumb to consider while setting up your own Grid.

Grids are defined by three criteria:

1. A Grid must coordinate resources that are not subject to centralized control and that cross organizational boundaries.
2. A Grid must use standard, open, general-purpose protocols and interfaces.
3. A Grid must deliver nontrivial qualities of service.

Meeting the first criterion will be a recurring theme of this column. Grids have an added layer of complexity on top of "simple" clusters, and sites will have existing policies that must be worked around instead of relying on a centralized control that simply rewrites local policies.

We will start by addressing multisite policies and team issues to be resolved before you begin construction of your Grid. Then we'll give some guidelines on how to define a software stack, what to do about security infrastructure, how to verify your Grid is fully up and running, and how to address user support for your organization.

Before You Begin

Before starting to set up a Grid, everyone involved should be clear about the goal for resource usage. For example, an organization may intend to use the resources for running a specific application, or as a platform for testing software scalability issues, or both. By defining success metrics at the outset, the people involved can avoid "mission creep" and misunderstanding of expectations.

With multiple sites, the problems of setting up a single machine or cluster within one administrative domain can be multiplied tenfold. The number of people who need to be involved in every decision increases, the policy for each site can be different, and global and local policies must be reconciled.

As everyone has experienced, communication is hard. And when you start to consider communication among different sites, possibly from different parts of the United States or even from different countries, these issues can be multiplied by large differences in communication styles. Recognizing that what you think is "chiming in with a helpful suggestion" may be regarded by one of your colleagues as a rude interruption is vital to maintaining a good working relationship.

Critical to the success of a Grid is establishing teams to address specific issues regarding Grid creation and maintenance. Johnston suggests forming at least two teams: an engineering working group to implement the deployment at each site, and an application specialist team. The engineering team should contain members from each site involved in the Grid and should have well-defined liaisons with the local system administrators and network administrators for all the resources to be used in the Grid. The application specialist team should contain people familiar with both Grid middleware and end-user applications; this team will act as an interface between the users and the administrators (who often speak very different languages).

Due to the cross-organizational nature of Grids, it is frequently impossible to maintain central control. Individual sites may be required to maintain local autonomy and control over their resources. This means that Grid middleware must support reconciling local and global policies across a Grid. For example, a site may define user names based on some local mapping. At one site, a username for a researcher might be the alphanumeric u11270, while at a second it could be the researcher's last name. In this case, the different local policies are reconciled by having a grid-map file that matches a certificate to a specific local mapping.

Other policies that will need to be reconciled include usage agreements, cross-site charging policies, security policies, and information-sharing policies. These should be addressed as early as possible in Grid creation to avoid conflicts later on.

Setting Up Your Grid

Once initial policy concerns have been addressed, it is time to define a software stack, establish a security infrastructure, verify your Grid functionality, and address user support for your organization.

Common Software Stack

A usable Grid requires that a common environment be defined for the users. One important element is the definition of a *software stack*, that is, the list of software and versions a user can expect to find on a resource, in three different catego-

ries: middleware components, user software, and environment variables for consistency.

Middleware components must address four basic functionalities: resource management, information services, data management, and security. Many Grids use the Globus Toolkit(tm) to provide these functionalities. Other software packages - such as CondorG, for easier resource management interfaces, and Ganglia, for more detailed monitoring - have well-defined interfaces to the Globus Toolkit and are commonly co-deployed.

User software includes software for development environments (compilers, debuggers), application-specific libraries (BLAS and GMP for mathematical libraries, for example), and system tools and libraries (open ssh, glibc, etc.). This software should be defined down to the version needed for compatibility between resources.

Users will have a much easer time switching between resources if their home environments are set up in such a way that differences in paths are hidden from them. Common ways to do this include using softenv, using modules, or publishing software locations in a Grid information system such as the Globus Toolkit Monitoring and Discovery System. Each of these approaches enables a local site to add a level of indirection to permit local control over where packages are installed, while at the same time allowing for a consistent global policy at the user level.

Several projects have started grouping together sets of software as a first approach to defining common software stacks. The Globus Toolkit distribution contains the main Grid middleware components. Several projects, such as the GriPhyN Virtual Data Toolkit (VDT) and the Grids Center NMI, release distributes binaries of Globus, Condor, the Network Weather Service, MyProxy and some other related tools that have undergone additional compatibility testing and support. TeraGrid is also defining a software stack - but for all levels of the hierarchy, not just middleware tools.

Security Infrastructure

Once a software stack has been defined, security issues must be addressed. These include trust issues such as CA acceptance, identification/authorization policy, gridmap file management, and accounting and allocation agreements.

Grid environments with a common Globus Toolkit deployment typically use Public Key Infrastructure-based tools and have certificates and a certificate authority (CA). Once you have decided on a CA, it is important to settle on policies for identifying users and resources. Many sites have divergent requirements on how a user must prove his or her identity before accessing resources. Some sites may issue you an identity certificate based on your email address, while others may require users to present, in person, a drivers license or passport before the certificate is issued. Typically all sites in a Grid must meet the highest identification policies required by any one site.

Once the identity policy issues are understood, certificates must be issued from the defined certificate authority for all hosts, services, and users. In addition, the mapping of user certificate to local login must be maintained throughout the Grid. In Globus Toolkit-based Grids, this is done through the use of a gridmap file that must be maintained in a consistent way on each of the resources.

Firewalls are another important issue. Some details are given at (I<www-fp.globus.org/security/v2.0/firewalls.html>), and these will also be discussed in an upcoming I<On The Grid> column.

Grid Functionality

After establishing the basic software infrastructure, you should verify that the installation is functioning properly. This verification should be done first on each site in isolation and then across sites. One set of verification tests is available at the Globus Toolkit website at I<www.globus.org/toolkit/testing>. Similar tests should be run for your other services. For example, MPI has a nice set of verification tests.

User Support

After your Grid has been established and everything is up and running, it is time to support the users - as they use it in ways you never imagined! In our experience, several simple mechanisms can ease the pain of this process.

- Establish Web pages and a quick-start guide. Several questions will be asked over and over: How do I get an account? What software is supported? Which commands do I use to run a job? We recommend (at least initially) simply adapting a quick-start guide from a project similar to yours.

- Set up mailing lists. Though simple, a mailing list of members from your sites, both systems- and application-oriented, can serve as a front line of defense for many user concerns.

- Establish a trouble ticket system for your project. Because of the cross-site nature of any Grid, having a cross-site trouble ticket

system that gets rerouted as necessary can strongly increase the level of usability of your Grid. Freeware such as Bugzilla is used extensively for trouble ticketing in many Grid projects.

First Steps to Success

Setting up a Grid, as we've seen, involves numerous issues stemming from the fact that different sites will have different local policies that must be reconciled with global control. In this column we have discussed many of those issues and have provided guidelines for setting up teams to run the Grid, defining a software stack, handling security infrastructure, and addressing user support for your organization.

You may need additional infrastructure as your Grid scales. We have not discussed accounting or shared allocations, higher-level schedulers, network monitoring between sites, or user-level portals. You may need many of these for a successful Grid, and we expect that these and others will be addressed in future columns.

Resources

"What Is The Grid"
I. Foster, www.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf

"Implementing Production Grids for Science and Engineering":
W. Johnston et al., , in The Grid: Blueprint for a New Computing Infrastructure (Second Edition), ed. Foster and Kesselman, 2003

Softenv Keys:
www-unix.mcs.anl.gov/systems/software/softenv/softenv-intro.html

Globus Toolkit is a registered trademark held by the University of Chicago.

This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38 with the University of Chicago and under Contract DE-AC03-76SF0098 with the University of California; by the National Science Foundation; by the NASA Information Power Grid program; and by IBM.

Jennifer M. Schopf is a researcher at Argonne National Laboratory and part of the Globus Alliance, with a focus on monitoring and performance. She can be reached at jms@mcs.anl.gov.

Keith R. Jackson is a scientist at the Lawrence Berkeley Lab where he leads the DOE Science Grid Engineering team. He can be reached at krjackson@lbl.gov.

Rexource box overflow to the right

Modules:
I<www.hlrn.de/doc/modules/>
Globus Toolkit Monitoring and Discovery System:
I<www.globus.org/mds>
GriPhyN Virtual Data Toolkit (VDT):
I<www.lsc-group.phys.uwm.edu/vdt>
Grids Center NMI release:
I<www.nsf-middleware.org/NMIR3>
TeraGrid:
I<teragrid.org>
Security and Crediential Management on the Grid:
I< S. Lang, S. Meder, ClusterWorld, January 2004>)
Globus Toolkit Firewall Requirements:
(I<V. Welch, www-fp.globus.org/security/v2.0/firewalls.html>)
Globus Tookit Installation Verification:
I<www.globus.org/toolkit/testing>
MPI verification tests:
I<www-unix.mcs.anl.gov/mpi/mpi-test/tsuite.html>