Globus: An Infrastructure for Resource Sharing

Every day millions of people click on hyperlinks without knowing the source of the information they are about to view. Insulated from the physical location of the data, the user just clicks and the protocols do the rest. The Globus Toolkit is designed to the same kind of thing only instead of linking users to data, Globus will link users to computational resources. Just as you do not know (or care) where your electricity was generated, so it may be with your computing cycles.

Grid computing holds great promise for the HPC cluster community. It is not intended to replace clusters nor will it turn every corporate LAN into a Top 500 cluster. Indeed, clusters may very well become the powerplants of the computational Grid.

In the article "What is a Grid?" (See Resources, pg. 50), Ian Foster of the Globus Alliance defined three criteria for a grid:

1 A grid must coordinate resources that are not subject to centralized control.
2 A grid must use standard, open, general-purpose protocols and interfaces.
3 A grid must deliver nontrivial qualities of service (e.g., relating to response time, throughput, availability, and security) for co-allocating multiple resource types to meet complex user demands.

Such a system gives users the ability to make resources available to others and to access others' resources. These resources can include data archives, computers, instrumentation, and networks.

The Grid has grown through a strategy of open source and open standards, similar to that of the Linux operating system, and distinct from proprietary attempts at resource-sharing software. This strategy encourages broader, more rapid adoption and leads to greater technical innovation, as the open source community provides continual enhancements to the product.

Nevertheless, realizing the full promise of the Grid requires solutions to fundamental issues of authentication, authorization, resource discovery, resource access, and most notably, incompatibility of resources and policies for managing them. A wide variety of projects and products offer services intended to address these issues. Here, we focus on the Globus Toolkit that underlies these various systems and that provides the basis for most deployments of Grid technology.

The Globus Toolkit: De Facto Standard in Grid Middleware

The Globus Toolkit was conceived to remove obstacles that prevent seamless collaboration. Its core services, interfaces, and protocols allow users to access remote resources as if these resources were located within the users' own machine room, while simultaneously preserving local control over who can use resources and when.

Included in the Globus Toolkit are software development kits to help programmers design and operate their own Grid applications. The current version, GT3, implements the protocols, APIs, and Grid services required for addressing security, data transfer, remote job submission, and discovery of resources and services. GT3 is based on industry standard Web services technologies and the Open Grid Services Infrastructure (OGSI), a technical specification recently approved by the Global Grid Forum, the international community and standards organization that defines technical standards for Grid computing OGSI specifies a set of "service primitives" that establish a nucleus of behavior common to all Grid/Web services that can be leveraged by meta- and system-level services. (Grid services are Web services that conform to a specific set of conventions.) OGSI is part of a broader, emerging Open Grid Services Architecture being designed to facilitate creation both of applications and of the infrastructure such applications require.

Let's take a look at the major parts of the Globus Toolkit. The toolkit provides, in addition to its OGSI core, a set of modular, complementary components: GRAM (Globus Resource Allocation and Management), which implements a resource management protocol; MDS (Monitoring and Discovery Service), which implements an information services protocol; and GridFTP, which implements a data transfer protocol. Each one uses the GSI (Grid Security Infrastructure) protocol to ensure the security of Globus Toolkit-initiated authentication and communication processes.

GRAM is based on a layered architecture where high-level global resource management services are built atop local organizations' resource allocation services. A standardized GRAM interface gives access to a variety of local resource management tools that a site might have in place, such as Load Sharing Facility (LSF), Network Queu-
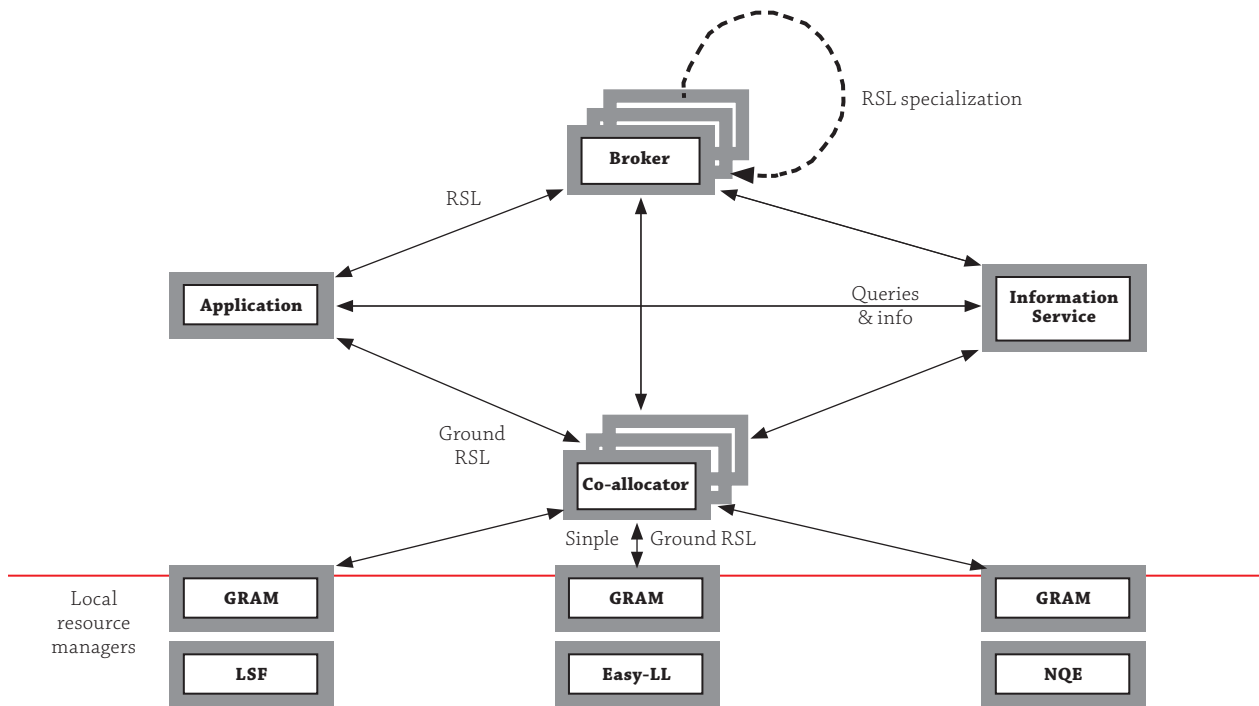
ing Environment (NQE), LoadLeveler, and Condor. Figure 1 shows the three primary resource allocation and management components: an extensible Resource Specification Language (RSL), an interface to local resource management tools (GRAM itself), and a co-allocator (DUROC).

MDS provides tools to enable discovery and querying of system components These tools include the Grid Index Information Service (GIIS), which knits together arbitrary information sources to present a coherent system image that the user can explore with other Grid applications.

GridFTP is a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks. Its protocol is based on the popular FTP protocol for Internet file transfers. To meet the heavy data-transfer requirements of Grid users, GridFTP includes additional features and extensions defined in Internet Engineering Task Force (IETF) RFCs. GridFTP benefits from strong GSI

security, data channels that are authenticated and reusable, the ability to perform parallel transfers and to resume partial or interrupted transfers, third-party (direct server-to-server) transfers, and command pipelining. GSI enables secure authentication and communication over an open network. This public-key encrypted security infrastructure permits mutual authentication across and among distributed sites, with single sign-on capability. No centrally managed security system is required, and the Grid maintains the integrity of its members' local policies. The Globus Toolkit's implementation of GSI adheres to the Generic Security Service API, a standard promoted by the IETF.

A typical series of Grid operations proceeds as follows. The user obtains necessary authentication credentials from one or more sites. After being authenticated, the user queries information services to discover available resources and locate required input

files. Next, the user submits requests for computation, for transferring data, for reserving storage or bandwidth, and so on. The Grid monitors progress of these requests and sends the user notification when they succeed, fail, or get delayed.

The Growing Grid Community

Just as the Web began in an effort to facilitate collaboration among researchers, the Grid's initial motivation was to advance the sharing of computational resources for science and engineering. The Globus Alliance's precursor organization, the Globus Project, started in 1995 with funding from DARPA and grew to its present international scope through investments by the U.S. Department of Energy, the National Science Foundation, and NASA. Large-scale research deployments are under way in the United States, Europe, and

Asia-Pacific for applications in high-energy physics, earthquake engineering and simulation, fusion energy, and biomedical imaging.

Since 2001, Globus Toolkit sponsors have included such corporations as IBM and Microsoft Research. The private sector has gravitated to the Globus Toolkit as the platform for a wide range of commercial services and applications. Companies committed to using the toolkit as their standard Grid software include Avaki, Cray, Entropia, Hewlett-Packard, IBM, Oracle, Platform Computing, Silicon Graphics, Inc., Sun Microsystems, and Veridian. The NSF Middleware Initiative's GRIDS Center (see Resources) has compiled a Grid Projects and Deployments System with descriptions of organizations worldwide that are using the Globus Toolkit and related technologies for research and commerce.

For users and developers seeking to participate in the Grid community, public conferences of the Global Grid Forum (GGF) and Globus Alliance offer excellent opportunities for enrichment. GGF meets quarterly, three times in the United States and once abroad, to facilitate discussion of proposed standards.

And the GlobusWORLD annual conference (Jan. 20-23, 2004 in San Francisco) features sessions and workshops focusing on Grid applications of interest to users from the industry and academic sectors.

In future issues, we will address specific Grid components and capabilities in greater detail, with the goal of providing in-depth technical details that ClusterWorld readers can use.

Tom Garritano is project manager of the GRIDS Center, part of the NSF Middleware Initiative. He can be contacted at contact@grids-center.org

---

```
$ ./taskmaster hostfile 5
10 10

Spawning host threads

Host lucifer thread running.
Host caine thread running.
Host uriel thread running.
Host abel thread running.
Host archangel thread run-
ning.

rand[0] = 0.840188
rand[1] = 0.394383
rand[2] = 0.700976
rand[3] = 0.809676
rand[4] = 0.561380
rand[5] = 0.224983
rand[6] = 0.916458
rand[7] = 0.133982
rand[8] = 0.274746
rand[9] = 0.046468

Results:
nhosts  nrands  delay  time
   5       10      10     30
```

Much better. Now five hosts take 30 seconds to do 100 seconds worth of work. This might turn out to be worthwhile after all!

Conclusion

Building a cluster (or discovering that your existing LAN *is* a cluster) is apparently pretty easy, really.

Using a fairly simple "master" script and a "worker" application we can clearly do work in parallel and can already see a significant speedup (a factor of three using five hosts) which should be quite reproducible on just about any LAN.

Between now and next month, you can play with taskmaster and task and see if you can discover settings that yield really good parallel speedup (where running on 5 hosts completes in close to 1/5 the time of one host).

In future columns we'll explore themes like Amdahl's Law and parallel speedup, parallel librar-ies, "the standard Linux cluster design," and more, using this basic cluster (and taskmaster/task) as a starting point.

We'll also learn better ways of installing and managing a cluster, how (and when) to go beyond the simple NOW-style cluster and into a grid or a Beowulf, how to compare shelved towers, rack-mounts, and different processor types and networks. Our goal will be to achieve a sufficient level of experience that you are ready to be "handed off" to my fellow columnists, whose dedicated columns will take you from being a neophyte to a perfect master in the specific areas of clustering that benefit you most.

Hope to see you there.

Robert Brown, Ph.D, is has written extensively about Linux clusters. You can find his work and much more on his home page at http://www.phy.duke.edu/~rgb.