LEE LIMING

Lessons Learned from Grid Deployment

In a previous column ("So You Want to Set Up a Grid," February 2004), Jennifer Schopf and Keith Jackson discussed the general steps for building a Grid. Over the past two years, several very large Grid deployment activities have been completed, including Grid3 [*See this month's feature — Ed.*], TeraGrid, and NEESgrid. These activities highlighted challenges that arise when deploying Grid applications and produced a number of methods and tools for overcoming those challenges. This month, we'll focus on four of these challenges and point to some of the solutions identified by these projects.

The National Science Foundation has supported work to identify and solve challenges in deploying Grid systems and applications (which they call "cyberinfrastructure"). The NSF Middleware Initiative (NMI) program sponsors a team of leading Grid technology institutions called the GRIDS Center to research, integrate, develop, and support Grid technologies so that future Grid projects have an easier time deploying systems and applications. Several authors of this column are GRIDS Center team members — including myself.

Before we dive into the four lessons and the associated tools and methods, it's important to recognize that two types of Grids exist in the modern vernacular. *Enterprise Grids* are systems and applications within a single organization that are intended to improve the efficiency of the organization's use of IT resources. *Inter-organizational Grids* are systems and applications that span more than one organization and are intended to facilitate cooperation and resource sharing among those organizations. Lessons 1 and 2 apply to both types. Lessons 3 and 4 apply mainly to inter-organizational Grids.

Lesson 1: Grid Deployment is a Software Integration Exercise

In the February column, Schopf and Jackson discussed several of the integration issues needed to build a working Grid: identifying a common software stack, defining and implementing a security infrastructure, defining and testing Grid functionality. Like any other system or application, a Grid has functional requirements: capabilities needed to support business or operational processes. To deliver those capabilities, application or system developers must:

1. review existing Grid components, such as the components in the Globus Toolkit,
2. identify the components needed to meet the system requirements,
3. develop any missing pieces, and
4. integrate the components into a single application or system.

The GRIDS Center has begun developing an inventory of existing Grid components. This inventory is called the "Grid Ecosystem," referring to the ecological concept of interrelated species. The Grid Ecosystem is a checklist that Grid application developers can use to ensure that they are not reinventing an existing component or missing an opportunity for cost savings. Many of these components are included in the NSF-sponsored NMI software suite, which means that the NMI program helps to maintain the software, ensure that it works well with other NMI components, provide technical support for the software, and provide downloadable distributions of the software. Other components are not yet part of the NMI suite.

Because the Grid Ecosystem includes so many components and because the cost of integration is significant, the GRIDS Center is also forming a list of "integrated solutions," or combinations of components that have been integrated by recent deployment projects to provide specific capabilities. This list currently includes the PURSE (Portal-based User Registration System) system for user registration and credential management, the LDR (Lightweight Data Replication) system for managing replicated data to optimize access time, and a system for monitoring operational system components to ensure that failures are detected and responded to rapidly.

Both the GRIDS Center and the Globus Alliance (*www.globus.org*) offer consultation services for Grid deployment projects in science and engineering, and several commercial organizations (e.g., IBM, HP, Sun, Oracle, Platform) offer consultation services for businesses contemplating Grid deployments. This consultation typically involves reviewing the project's requirements, identifying relevant existing components, and planning the development and integration activity.

Lesson 2: Multiplatform Support is a Big Deal

The TeraGrid and NEESgrid deployment activities were, for the most part, uniplatform deployments, requiring participants to use a single agreed-on platform. Grid3 allowed a small number of platforms, all of

them Unix-based. The next phase of the TeraGrid project requires supporting several additional platforms, and NEESgrid will most likely need additional platform support as international and industrial collaborations are formed. Because Grid systems and applications are typically formed by integrating several components from different sources (see Lesson 1), ensuring that all of the components work properly, both individually and collectively, on multiple platforms is a challenge.

In another recent column ("Testing in a Grid Environment," June 2004), GRIDS Center team member Charles Bacon described steps for establishing a testing infrastructure for Grid software. When multiple platforms are added to the mix, the complexity of the problem grows quickly, and many deployment projects are overwhelmed by the need not only to fix platform compatibility issues but also to conduct the testing required to identify the issues in the first place.

The GRIDS Center is responding to this challenge by operating a software "build and test" facility at the University of Wisconsin-Madison. This facility includes a heterogeneous pool of systems running many different platforms and a scheduler for automatically building software components and integrated systems and running associated test suites on the software. Deployment projects can supply their software stack and a set of tests, and the build and test facility will automatically schedule the software builds and test runs on a variety of platforms, reporting the results to the deployment team. The facility can automatically check out software from online repositories, allowing deployment teams to fix problems and then quickly retest the software without generating a new distribution for the test facility.

Because not everyone is comfortable with relying on an external testing facility, and because the UWM facility does not plan to increase its capacity forever as new deployment projects request its help, it is the intent of the GRIDS Center to make the software that operates the build and test facility available to the public. When this happens, companies and other organizations will be able to build their own build and test facilities without having to start from scratch.

Lesson 3: Consistent Installations are Hard to Coordinate

Enterprise Grids sometimes have the luxury of a single deployment team with a unified management structure. Inter-organizational Grids do not have this luxury, and they must contend with the reality that different groups, more often than not, do the same things in different ways. In the February column, Schopf and Jackson pointed out the critical role that a common software stack plays in ensuring that all sites within a Grid use the same software. Of course, one can install and configure the same software in different ways, so the stack must define not only the software to be installed but also the installation methods and the configuration settings needed to keep the Grid working.

One way to overcome this challenge is to reduce the installation problem to its simplest form and minimize the number of choices that can be made at installation time. The Rocks Cluster Distribution is a software suite for clusters developed at the San Diego Supercomputer Center. The suite is currently used to manage more than 250 clusters around the world, totaling more than 17,000 CPUs. The Rocks base distribution (a bootable CD-ROM) includes basic operating system and cluster management tools, but additional "Rolls" (CDs) provide specialized software for typical cluster uses (visualization, compute farms, storage farms, etc.) and even a few Grid applications. For example, the BIRN (Biomedical Informatics Research Network) system uses Rocks as its distribution mechanism, and all BIRN clusters are built by installing the Rocks base distribution and then applying Grid and BIRN-specific "Rolls." Updates to the software require that a new CD-ROM be distributed and each cluster reloaded from the CD. By greatly reducing the complexity of system administration, BIRN administrators are assured that their clusters will meet the minimum requirements for participation in the BIRN system.

The Grid3 deployment activity takes a different approach, using a tool called PACMAN (PACkage MANager) to simply the installation and configuration process. Grid3 system architects install the Grid3 software on a number of centralized systems, known as "caches." These caches include both the software and the configuration settings needed to ensure a consistent installation. Local system administrators use the PACMAN client to connect to a cache and automatically download, install, and configure the Grid3 software. When systemwide updates are needed, the caches are updated and local systems automatically update from the caches. This method leaves basic cluster maintenance to the local system administrators while at the same time offering them an easy way to install and configure the required Grid3 software.

Lesson 4: Service Level Agreements are Good; Enforcing Them is Better

A service level agreement (SLA) is a definition of the services that a specific system will provide. An SLA for a Web hosting service, for example, might specify the availability of a

Web server, an email server, an administration service, a status monitoring service, and a usage monitoring and logging service. Failure to provide any of these services may be viewed as a breach of contract on the part of the hosting service. The NEESgrid system defined an informal SLA that required certain services to be provided by every site participating in the NEES collaboration. Grid3 and TeraGrid each defined more formal SLAs that each site had to conform to or else suffer sanctions from partner institutions or funders.

While it is important to define SLAs for the systems that make up a Grid system, it is even more important to have mechanisms that test compliance with the SLA and also monitor and record compliance over time.

FIGURE ONE Example of the Inca monitoring system

The TeraGrid project developed an SLA monitoring system called Inca, which is freely available for use in other Grid deployment projects. Inca has enabled the TeraGrid deployment teams to measure their sites against a predefined SLA and to demonstrate compliance at specified levels over time. The TeraGrid SLA covers issues such as software packages that must be installed and configured in certain ways, services that must be running on predefined ports, accounts that must be created, environment variables that must be set, and directories and files that must be present on the system. Inca comprises a test script interface that defines how a test script is invoked and how it reports success or failure, a harness for running tests automatically, and a hierarchical reporting system for collecting test reports and producing aggregate reports at multiple levels of detail. The terms of the SLA are defined by the specific tests that Inca is configured to run. An archive service logs the test results as time passes. The result is a set of Web-accessible reports that shows, at a glance, which elements of a Grid are operating in compliance with an SLA and which elements are not.

Tools like Inca, PACMAN, and Rocks are proving to be key elements in successful deployments of Grid applications and systems. Though each new production deployment raises new issues, work done in earlier projects, and clearinghouses like the GRIDS Center, are helping new teams to avoid the pitfalls of the past.

Resources

Grid3	www.ivdgl.org/grid3
NEESgrid	www.nees.org
TeraGrid	www.teragrid.org
GRIDS Center	www.grids-center.org
The Grid Ecosystem	www.grids-center.org/ecosystem
PURSE	www.grids-center.org/solutions/purse
GRIDS Build and Test Facility	www.grids-enter.org/services/buildtest
PACMAN	physics.bu.edu/~youssef/pacman
Rocks	www.rocksclusters.org
Inca	tech.teragrid.org/inca

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 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.

Lee Liming is manager of the Distributed Systems Laboratory (DSL) at Argonne National Laboratory, part of the Globus Alliance.