DISTRIBUTED SYSTEMS DEPARTMENT

While Grid middleware is becoming more widespread, few applications are able to easily take advantage of Grid infrastructure such as that provided by the DOE Science Grid. Currently it is still too difficult for scientists to interact with the Grid and develop and troubleshoot Grid applications. The focus of the Distributed Systems Department at the U.S Department of Energy's Lawrence Berkeley National Laboratory is to develop tools to enable scientists to map their workflow onto the Grid, easily prototype and develop Grid applications, and monitor and troubleshoot Grid usage.

We are demonstrating a set of next-generation Grid tools that address many of these issues. A prototype workflow manager that allows scientists to graphically describe their workflow and have it implemented on the Grid provides a useful interface to the Grid. Several Grid tools developed using our Python toolkits and instrumented with the NetLogger package are being used to demonstrate the ability to dynamically monitor a Grid application and troubleshoot faults. In addition, fine-grained authorization decisions are being made with the Akenti authorization service to control access to Grid resources.

The DOE Science Grid is providing an advanced distributed computing infrastructure on a scale that supports the level of scientific computing necessary for DOE to accomplish its scientific missions. The Science Grid is a collaboration between Lawrence Berkeley National Lab, Argonne National Lab, Oak Ridge National Lab, Pacific Northwest National Lab, and DOE's National Energy Research Scientific Computing Center.

Tools for Rapid Deployment of Grid Applications

The ubiquitous adoption of Grid middleware offers the promise of creating science applications by gluing together multiple components. What is required to make this vision a reality is a suite of tools that will enable this easy composition. We are currently developing such tools based on the Python programming language for both the 2.x and 3.x versions of the Globus Toolkit™. Python is ideally suited for use as a "glue" language, as it provides a nice combination of easy to read syntax, high-level constructs, support for interacting with C/C++ and Fortran, and has widespread use in the scientific community. We are demonstrating an application built using these tools that captures the workflow of a standard computational chemistry application and provides the ability to troubleshoot and monitor the underlying components.

Grid Troubleshooting

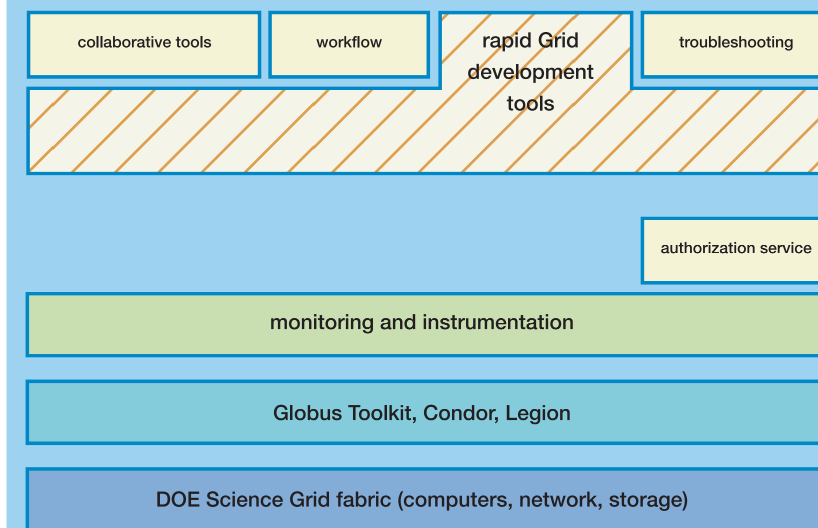
One of the central challenges of Grid computing today is that, in practice, Grid applications are prone to mysterious failures and inexplicable performance bottlenecks. This problem is compounded by the layers of software which are designed to shield Grid users from the gory details of a large and complex system. These layers are essential for usability, but may also hide the very feedback needed to diagnose poor performance or failure.

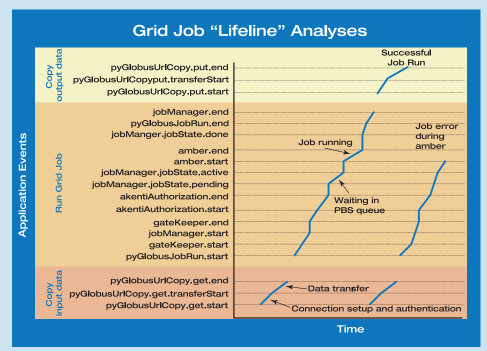
For example, assume a Grid job has been submitted to a resource broker, which uses a reliable file transfer service to copy several files to the site where the job will run, and then runs the job. This particular process should normally take 15 minutes to complete, but two hours have passed and the job has not yet completed. Determining what, if anything, is wrong is difficult and requires a great deal of monitoring data. Is the job still running or did one of the software components crash? Is the network particularly congested? Is the CPU particularly loaded? Is there a disk problem? Was a software library containing a bug installed somewhere?.

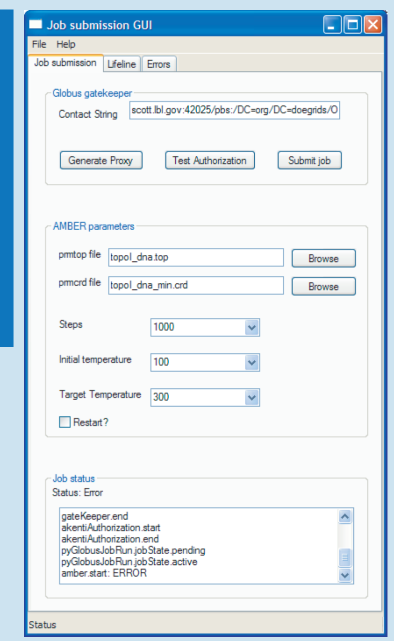
Monitoring, defined as the measurement and publication of the state of a Grid component at a particular point in time, provides the information to help track down the current status of the job and locate problems. To be effective, monitoring must be "end-to-end", meaning that all components between the application endpoints must be monitored. This includes software (e.g., applications, services, middleware, operating systems), end-host hardware (e.g., CPUs, disks, memory, network interface), and networks (e.g., routers, switches, or end-to-end paths). The monitoring framework must provide accurate, detailed, and adaptive monitoring of all distributed computing components, including the network. Analysis tools can use this monitoring data for real-time analysis, anomaly identification, and fault recovery.

End-to-end monitoring data is needed to troubleshoot Grid applications. In our demo we will show how end-to-end monitoring data and monitoring data analysis tools can be used to locate the source of failures in a Grid Environment.

Tools for Next Generation Grid Applications







Grid Authorization services

Strong, but easy to use access control of Grid resources and services is essential to scientific collaboration on the Grid. Akenti is a general purpose policy-based authorization service that can be called at various points by a resource provider to make fine-grained decisions about access. Policy can be determined and signed by multiple stakeholders for a resource, enabling access policy to be local to a resource site or handled centrally by a virtual organization. Akenti can be used to return policy decisions to a policy enforcement point, or can return a signed authorization assertion to a client, who can in turn present it to one or more access points.

Tools for Grid Workflow Management

We are developing a collaborative workflow description and tracking tool for use in Grid environments. The introduction of the BPEL4WS workflow language has led to the availability of a language and workflow engines suitable for prototype implementations. Our collaborative workflow tool is based on BPEL4WS and uses a Condor-G SOAP interface for job submission and tracking and the capabilities of DAGMan to execute Directed Acyclic Graphs (DAG) of Grid jobs.

We are focusing on the role of scientific workflow management in facilitating daily activities between collaborating researchers, rather than just traditional job control workflow. To accomplish this goal, workflow management is being integrated into the collaborative environment in which the researchers work. The integration of computing and collaborative interaction is best achieved through direct interactions between the workflow management tool and other collaborative tools. For example, workflow exceptions, such as disk quota or credential problems, that occur within a managed workflow can interactively notify users through the secure messaging tool. Or, as part of a normal data reduction workflow, intermediate results can be posted to a common web server with the appropriate URL delivered as an asynchronous note to those responsible for monitoring the analysis process.

Collaboration Technologies and Tools

The Pervasive Collaborative Computing Environment (PCCE) is providing lightweight, non-intrusive, and flexible ways for collaborators to stay in touch and work together. The goal of the PCCE is to leverage the Grid middleware environment including Grid security, authorization and information services, in developing tools that support the day-to-day connectivity and underlying needs of collaborators. It allows participants to securely locate each other, use asynchronous and synchronous messaging, share documents, share progress, share applications and hold videoconferences.

In a typical scientific collaboration, there are many different locations where data would naturally be stored. We are developing a lightweight, scalable, and secure file-sharing system that will enable scientists to store and manage their files on local storage facilities while sharing them with remote participants. The underlying infrastructure that will enable this is the secure and reliable group communication system. The end product will be a scalable and secure peer-to-peer file-sharing system that leverages off of the capabilities of a group communication system and X.509 identity certificate based authentication and authorization.