eServices Infrastructure for Collaborative Science

Principal Investigators:

Katarzyna Keahey, ANL

Keith R. Jackson, LBNL

Summary

The eServices Infrastructure for Collaborative Science Project is developing technology and services designed to address the needs of the Science and Technology Computing (STC) community in the context Grid Services. By providing a unifying standards-based architecture for computational Grids, Grid services provide a platform for the development of next generation grids. This project supports this effort by contributing towards developing general Grid services infrastructure as well as developing higher-level services. Our main focus however is to make Grid services accessible to scientific applications. We therefore investigate performance issues, develop language bindings needed by the STC community such as C/C++ and Python, and develop application-level services and functionality.

The emergence of Grid services and the supporting architecture is a milestone in the evolution of computational "Grids": systems allowing for flexible, secure, and coordinated resource sharing among dynamic collections of individuals and institutions. It represents an evolution of the current Grid technologies leading towards a Grid system architecture based on an integration of Grid and Web services concepts and technologies. A standard technology base allows us to provide a unifying framework for the description, discovery, and invocation of services in the Grid. Conversely, it also allows technologies developed within the Grid community to provide added value to Web services, by addressing federation, security, reliable invocation, and lifetime/state management issues not supported by current Web services technologies. Thus, the advantages of the framework, flow in two directions: developers of collaboratory applications gain access to emerging Web services technologies while at the same time, the industrial Web services community gains access to the advanced technologies developed by the Grid community.

From the perspective of Grid users providing a unifying architecture for the Grid significantly lowers the entry barrier and development costs of building collaborative, distributed applications. It enables service developers to rely on well-defined standards in order to provide their services, and it enables users to access these services in standard ways. This enhances the collaborative work paradigm through opening the doors for new inter-collaboratory relationships. For this reason, we believe that this effort stands to be of critical importance to the widespread acceptance of Grids.

Grid services address the needs and challenges prevalent in both e-business and e-science. While these two communities share a common vision, and a common base in terms of needs related to Grid computing, they also face challenges unique to their specific needs. This project is dedicated specifically to investigating and satisfying the unique needs arising in the context of Scientific and Technical Computing (STC).

In addition to ensuring that the needs of STC community are reflected in emerging Grid services standards, we provide implementations of the corresponding infrastructure using tools and languages that the scientific community relies on, such as C/C++ and Python. We will investigate performance issues of critical importance to STC; our focus here is on communication protocols and optimal infrastructure implementation. Finally, we invest in application-level infrastructures and services development to make the Grids easily accessible to application scientists. A thrust in higher-level services development allows us to support and bind these efforts.

The current research and development efforts focus primarily on producing efficient and reliable implementations of Grid services infrastructure. Based on community feedback, and in order to facilitate transitions for established user communities we also initiated efforts providing backwards compatibility with previously existing Grid infrastructures. Accordingly, this project has contributed the C and Python client side implementations of the Open Grid Services Infrastructure (OGSI), allowing for secure interactions, and fully compatible with existing implementations of OGSI servers. In addition, we provided the Globus Toolkit 2 (GT2) GRAM client interface, familiar to many Grid users, implemented on the base of these bindings. This code has been made accessible to the community as part of the regular GT release cycle. .
Experience gained while building both the Python and C bindings, convinced us that more work was required at the basic Web Service level. Much of the required tooling was not available. To meet this need in Python we built tools to support the automatic generation of bindings to Web Services defined in WSDL. This includes support for generating Python classes from XML Schema types. In a similar thrust, the C/C++ development joined an open source project to provide the requisite Web services tooling. In addition, we implemented elements of the Web Service Security standards to allow for the usage of message level security. This is important to allow compatibility with the ongoing security work in OGSA and the commercial web service community. We also developed basic support in both C and Python, for server side functionality. This will allow the development of OGSI services in C/C++ and Python that can leverage off the sophisticated security, lifecycle management, and notification services that are an integral part of the OGSA standard. It will also be of vital importance to allowing legacy scientific codes to be exposed to the Grid as OGSI components.

The recently released Web Services Resource Framework (WS-RF) standards are the successor to OGSI. Although they are for the most part semantically equivalent, a port to this new standard will still require a refactoring of our code bases. In the next year, we will develop full WS-RF implementations in Python and C/C++.

Working with application communities we also initiated work on application services that should be of broad interest to the STC. Of primary importance is the investigation into remotely manageable dynamic accounts or sandboxes and agreement-based systems such as WS-Agreement. We will continue this work in the future and integrate it with the developing Grid services infrastructure.

For further information on this subject contact:
Kate Keahey,
Argonne National Laboratory
http://www.mcs.anl.org/eservices
Phone: 630-252-1673
keahey@mcs.anl.gov