

# ***VisPortal: Increasing Scientific Productivity by Simplifying Access to and Use of Remote Computational Resources***

## **Problem Statement**

Computational science is the process of modeling and simulating phenomenon with the ultimate aim of understanding complex systems. As computer platforms increase in capacity and simulations increase in capability, the data produced by computational science research projects grows ever larger and more complex. In response, resource centers, like the National Energy Research Scientific Computing center, provide unique capabilities and computing capacity. With the growth in the size and complexity of simulations, computing resources and analysis tools comes the need for researchers to master a larger number of increasingly complex systems in order to perform their work. Keeping up with rapidly evolving infrastructure can easily become a full-time job that detracts from a researcher's core mission – scientific research.

## **Approach**

Our goal is to simplify and streamline the process of using remotely located visual data analysis software tools. This discussion presents an example of an easy-to-use interface that mediates access to and use of diverse and powerful visual data analysis resources. The interface is presented via a standard web browser, which is ubiquitous and a part of every researcher's work environment. Through the web interface, a few mouse clicks are all that is needed to take advantage of powerful, remotely located software resources. The VisPortal project is the software that provides diverse services to remotely located users through their web browser. Using standard Globus-grid middleware and off-the-shelf web automation, the VisPortal hides the underlying complexity of resource selection and distributed application management. The portal automates complex workflows that would otherwise require a substantial amount of manual effort on the part of the researcher. With a few mouse clicks, a researcher can quickly perform complex tasks like creating MPEG movies, scheduling file transfers, launching components of a distributed application, and accessing specialized resources.

While portals offer considerable advantages over standalone grid applications in terms of managing client-side complexity, working within the portal paradigm has not been an easy task. While it shields the users from the underlying complexity of the grid, the same cannot be said for developers who are attempting to use portals to encapsulate the complexity of using grid-based resources. The exercise of creating a working implementation with a very controlled set of users is critically important for identifying the deficiencies in current technology and driving the direction of future middleware development.

Our portal builds upon the Grid Portal Development Toolkit (GSDK) developed by Jason Novotny. GSDK uses the Tomcat/JSP 3.x web engine and the Java CoG 1.0a to provide direct Grid service access to the Java Server Pages (JSP) engine. JSP is widely used in industry by e-commerce applications, so we have many opportunities for synergy with commercial web automation environments. Also, based on the experiences of several previous portal implementation efforts, we have found JSP to be considerably more readable and maintainable than Perl-CGI or Python-CGI methods.

At SC03, we demonstrated the VisPortal and Visapult, which is a pipelined-parallel direct volume rendering application. In the demonstration, a user first authenticates to the portal, which provides access to remote data storage and computing Grid-based resources. Next, the user interacts with VisPortal through their browser to select a data files, and to change default parameters for the Visapult application if so desired. Next, the portal brokers file transfer, if necessary, and then launches the multiple components of the distributed Visapult application. Without the portal interface, the user would be required to manually move data files, to manually launch the Visapult components on multiple machines, and to study manual pages for the Visapult applications to learn the syntax of a complex command line interface.

## Benefits

Shielding the user from complexity results in more efficient use of computational resources. Our VisPortal shows the potential use of such interfaces. The portal paradigm for encapsulating capabilities represents a leap forward for computational scientists by allowing them to focus more on their scientific research and less on the mechanics of using distributed resources.

Additionally, the portal environment provides a testbed environment for software developers where new tools targeted at specific research challenges can be quickly placed into the hands of the researchers. In this way, the portal provides a fast path to bring research results into production use.

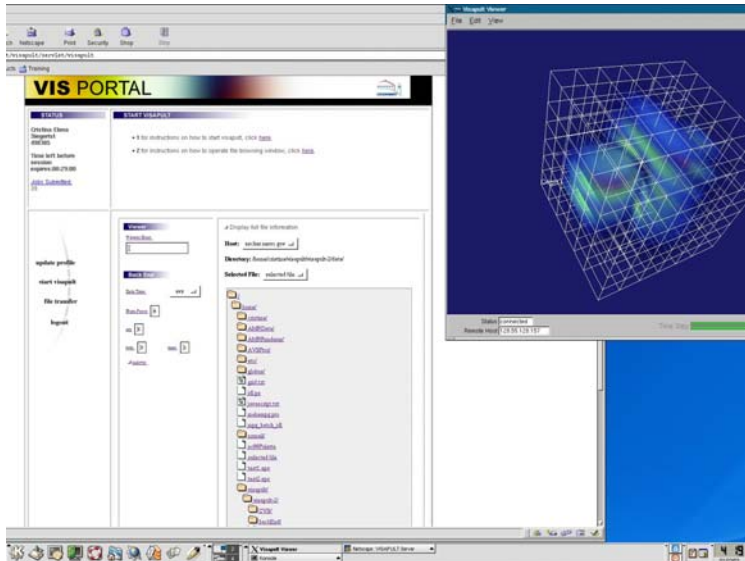


Figure 1. Using a standard web-browser interface, VisPortal presents task setup and execution for complex software tools. This figure shows the Visapult pipelined-parallel direct volume rendering application.

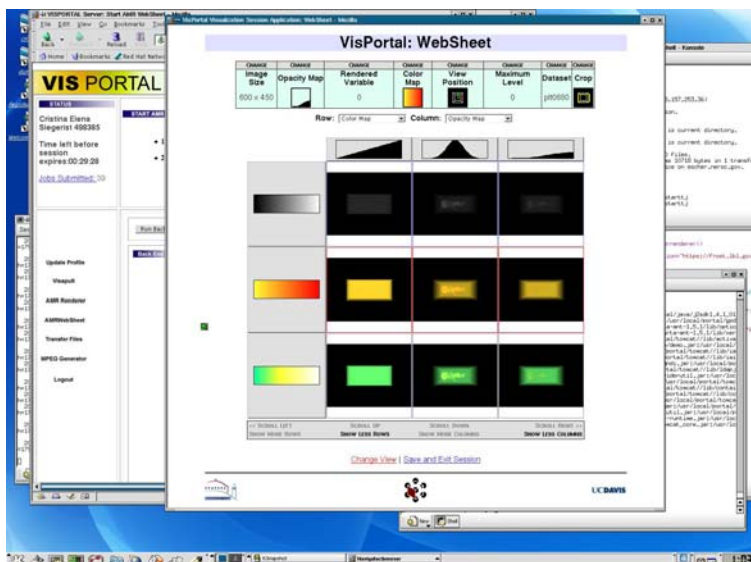


Figure 2. VisPortal automates complex workflows, thereby accelerating the rate of scientific discovery. This figure shows a portal-brokered result of a complex parameter study visualization application, the AMRWebSheet research prototype.

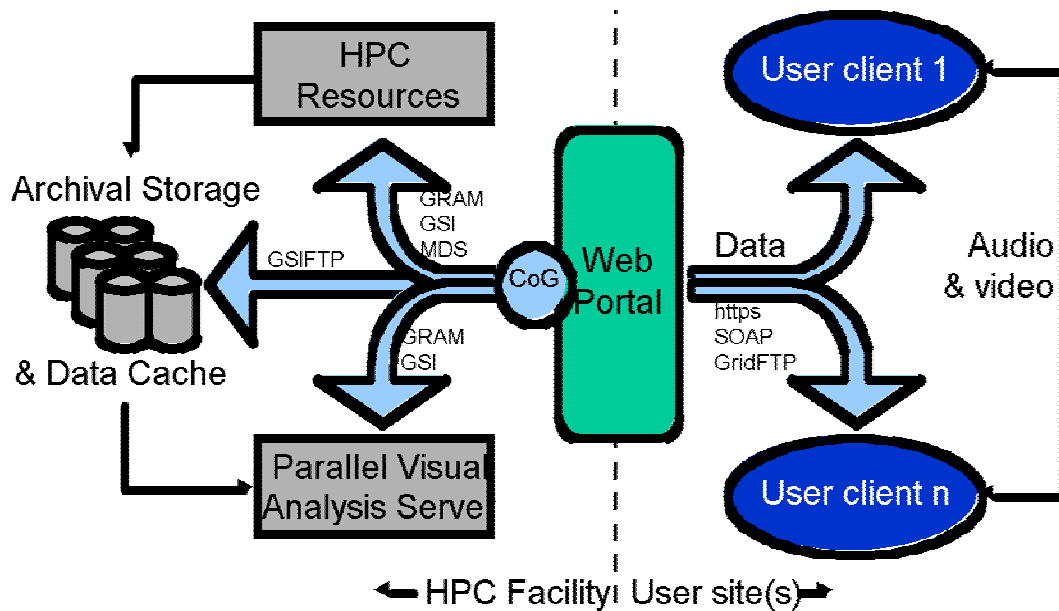


Figure 3. GPDK uses a three-layer architecture. In the client layer, a user's workstation runs a web browser that is capable of supporting Javascript, cookies and SSL. In the second layer, a GPKD-enabled web server runs on a different machine, and processes HTTP requests. In the third layer, grid-based resources are invoked from the GPKD portal, and may be resident anywhere on "the grid."

### Contact

Cristina Siegerist, LBNL/NERSC Visualization Group. [cesiegerist@lbl.gov](mailto:cesiegerist@lbl.gov), 510-495-2520.