COL/NW/03

Parallel Analytical Visualization Grid Service

FY 2005 Proposal to the NOAA HPCC Program

September 13, 2004

| Title Page | Proposed Project | Budget Page |

Principal Investigator: **Phyllis J. Stabeno**

Line Organization: OAR/PMEL
Routing Code: R/PMEL
Address:

NOAA/PMEL/OE2
7600 Sand Point Way NE
Seattle, WA 98115

Phone: (206) 526-6453
Fax: (206) 526-6485
E-mail Address: Phyllis.J.Stabeno@noaa.gov

Christopher Moore	Albert J. Hermann	Paul Hyder
Christopher.Moore@noaa.gov	Albert.J.Hermann@noaa.gov	Paul.Hyder@noaa.gov

Proposal Theme: **Technologies for Collaboration, Visualization, or Analysis**, with aspects of **Advanced Networking Technologies**

Phyllis J. Stabeno	William E. Sites	Cynthia L. Loitsch	Eddie N. Bernard
Principal Investigator	Division Leader	Program Support Officer	Director
PMEL	OERD/PMEL	PMEL	PMEL

Parallel Analytical Visualization Grid Service

Proposal for FY 2005 HPCC Funding

Prepared by: Phyllis Stabeno and Christopher Moore

Executive Summary:

We propose to develop a 3D visualization application that runs as a Grid service on IPV6. The service will do parallel composite rendering (creating different parts of the visualization on different computational nodes on the Grid) using the VTK client-server framework and the MPEG2 MPI library built on top of the Globus toolkit. Globus is standard middleware for OGSA applications. Our application, leveraged from previously-funded HPCC projects[1], will be implemented on the NOAAGrid and increases the functionality of the NOAAGrid as a cyberinfrastructure workflow system. With this addition to the NOAAGrid, users will be able to use a web page to submit model runs for Grid distributed computing and also use distributed Grid resources to do analysis and visualization.

Problem Statement:

The easy access to low-cost, high-performance, network-aware computers has had a great impact on the way scientists conduct their research. Atmospheric and oceanographic modeling and analysis has improved, but the scientist is burdened by the increasing size of the data collected and the size of model results due to the large, dense grids on which models are being run.. In the immediate future, we anticipate collecting data at the rate of terabytes per day from many classes of applications, such as climate simulations running on teraflop class computers and experimental data produced by larger and more sensitive oceanographic instrument arrays.

Of course, the generation of data is not an end in itself. Rather, one generates or acquires data in order to obtain scientific insight. While there is a role for data analysis and reduction, understanding is achieved when the user interprets results - usually in the form of a visualization - and the field of scientific visualization has emerged as a challenging and important research area in its own right. Advances in algorithms, computer architecture, and visualization systems continue with the goal of allowing ever more sophisticated analyses of larger datasets. Desktop systems capable of interactive exploration of Gigabyte datasets represent the state of the art at the present time. High-resolution and immersive (virtual reality) displays, advanced analyses such as feature detection and tracking, and Terabyte (or even Petabyte) datasets on distributed systems are now commonplace.

---

1 "Development of a Prototype NOAA Grid", (FY04 - Govett, Schaffer, Gross and Moore),
"Personal Access Grid nodes for Immersive Collaboration", (FY04 - Stabeno, Hermann and Moore), and
"JavaGIS", (FY04 - Vance, Moore, Kennedy and Payton)

In a time when computational and data resources are distributed around the globe, users need to interact with these resources and each other easily and efficiently. A computational Grid, by definition, represents a connection of distributed resourced that can be used regardless of the user's location. Analyses, model runs, and scientific visualization are all part of a workflow process that must deal with new challenges: for example, security, wide area networks, heterogeneity, and unreliable components. New approaches to system architecture can allow us to address these challenges in a principled fashion. In particular, infrastructure being developed for emerging national-scale computations Grids can be leveraged to both simplify and enhance the robustness, performance, and portability of scientific visualization systems.

This project addresses many of the HPCC program objectives, including utilizing new infrastructure technology in the form of the Grid and Globus middleware, as well as increasing efficiency by offering advanced visualization to all NOAA labs as a Grid service.

Proposed Solution:

In FY04, PMEL, FSL, and GFDL were linked in a HPCC-funded prototype Grid called the NOAAGrid. Distributed-memory clusters at GFDL and FSL, and individual user nodes at PMEL were connected, and coupled parallel ocean-atmosphere model runs were performed to demonstrated the feasibility of performing weather-prediction or climate research model runs on this computational Grid. As a follow-on to that project, we plan on increasing the NOAAGrid's usefulness by taking it one step closer to a true cyberinfrastructure system - adding grid services to do high-end visualization and analyses of the resulting large data set.

I. Grid Services

A. MPICH-G2: A Grid-Enabled MPI

The NOAAGrid is patterned after the NSF-funded TeraGrid, and is based on middleware called the Globus Toolkit® (www.globus.org). The distributed-memory models typically used by NOAA researchers rely on a message-passing interface known as MPI. The NOAAGrid uses MPICH-G2, a complete implementation of the MPI-1 standard that uses Globus Toolkit services to support efficient and transparent execution in heterogeneous Grid environments. MPICH-G2 selects the most efficient communication method possible between any two processes, using vendor-supplied MPI if available, or Globus communication (Globus IO) with the option to utilize *parallel sockets* via Globus' GridFTP.

B. GridFTP - Parallel Data Transfer

Applications that transfer large blocks of data from one process to another over a high-bandwidth network (e.g., visualization data) may instruct MPICH-G2 to use these parallel sockets for communication between processes through the use of existing MPI idioms (the researcher need not adjust his code - this benefit is "plug and play"). GridFTP is based on the

RFC 959 FTP protocol, leveraging the ubiquity of the standard, adding backward-compatible extensions to allow for such functions as reliable restarts, performance monitoring, and coordinated multi-host transfers. This ability makes the Grid a perfect platform for offering a parallel visualization system, which must transfer large amounts of either geometry or raster data, back and forth between remote clusters and users' desktops.

II. Parallel Composite Rendering

A. High-Resolution Remote Rendering

A parallel visualization server prototype will facilitate the generation of isosurfaces from terabyte-sized datasets generated from a Grid-launched model run. These images can be saved to a file for later viewing, or streamed in near-realtime to an immersive display system, or a user's desktop using GridFTP. The technique used is referred to as composite rendering. In parallel composite rendering, the client requests the grid service for an isosurface - the data is loaded and isosurfaced in parallel, then the rendered subpieces and zbuffer are sent to the compositing component to be stitched back together on the clients' display (figure 1).

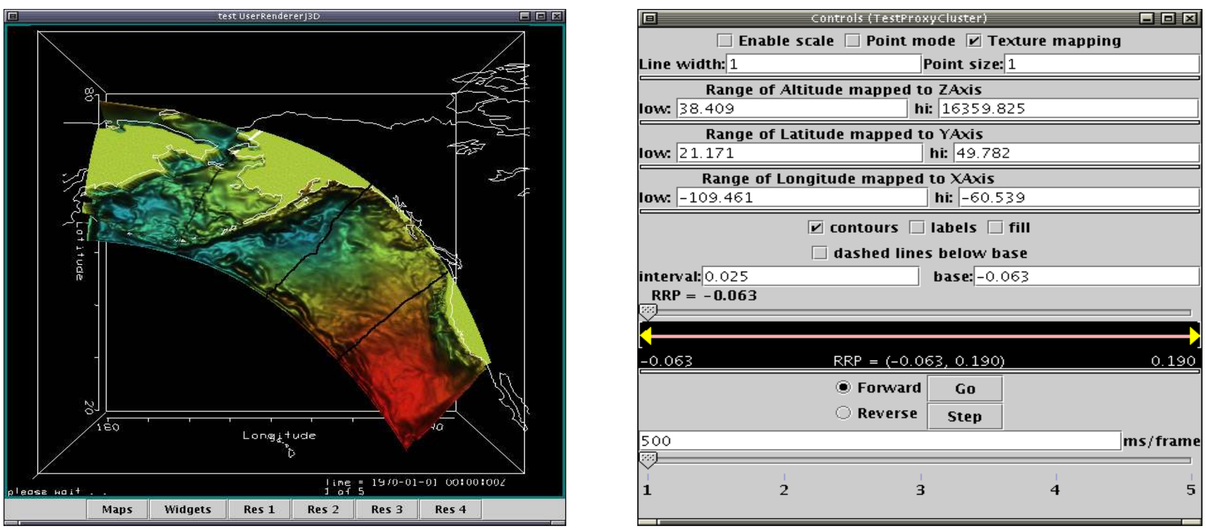

Figure 1: Test composite rendering of an isosurface of model output from an ocean model. The geometry was rendered on four processes and streamed to this client window and stitched together using a sort-last rendering order. While this test used Java RMI for the transfer of geometry, we propose the use of GridFTP to speed transfer rates, and to offer the rendering as a OGSI service. The user can interact with the render window, or request a new isosurface, new variable, or animate through a time series.

B. ParaView: Graphics Framework

For the rendering engine, we propose to use a parallel visualization application called ParaView (www.paraview.org). Created by Kitware in conjunction with Los Alamos National Laboratory (LANL), Sandia National Laboratories, and the Army Research Laboratory, ParaView is a set of parallel processing tools with an emphasis on distributed memory implementations, including parallel algorithms, infrastructure, I/O, support, and display devices. ParaView is open-sourced, and based on Kitware's Visualization Toolkit (VTK) , which is used as the basis in many high-end visualization applications in the geo- and health-sciences such as vGeo (by VRCO), and SurgeryAssist. VTK uses OpenGL hardware-accelerated rendering and compositing techniques, and a modular interface that allows users to write custom data file readers, renderers, and change the GUI interface.

While ParaView can render the typical 3D graphics primitives such as contours and isosurfaces, vector fields, streamlines, and volumes, but can also filter datasets to create subsets by clipping volumes or specifying threshold values, perform calculations on data arrays, or probing data at a point or along a line or plane, adding analytical capability to an already powerful visualization system. Since ParaView runs on distributed-memory systems using MPI, it can function as the compositing component using MPICH-G2 in client/server mode. It supports both *distributed rendering* (where the results are rendered on each node and composited later using the depth buffer), and *local rendering* (where the resulting polygons are collected on one node and rendered locally) depending on whether the requested grid service comes from a local or remote client. It can can also be used on existing immersive displays such as the ImmersaDesk and GeoWall, or tile wall displays.

ParaView was written as a rendering *framework*, and, as such, provides several methods for user modification, and for adjusting the application to fit the user's needs. Firstly, ParaView, allows the user to create additional modules by writing an XML description of the interface. ParaView provides this API connection to allow users/developers to add their own VTK filters to ParaView without writing any special code and/or re-compiling . File readers and converters for this project will be quickly created using the XML ParaView interface. The second framework feature is that ParaView is fully scriptable using the Tcl interpreter language. Every command executed during a session can be saved in a trace file which can be re-loaded to reproduce the session. Furthermore, since the session file is simply a Tcl script, it can be edited/modified and loaded to obtain different results.

Analysis:

Our participation in the MEAD expedition, a major project within the National Center for Supercomputing Applications (NCSA), has been key to our success in the NOAAGrid project, and in expanding the high-performance cyberinfrastructure within NOAA. Investigators Moore and Hermann continue to work with NCSA on this and other projects, leveraging innovative ideas from their efforts, and putting a NOAA presence onto the TeraGrid.

Potential users of this service, and the NOAAGrid in general, include, but are not limited to: the Hurricane Research Center at AOML (where visualization of storm-track models are key to their effort), the Global Carbon group at GFDL (whose cluster administrators collaborated with us in the FY04 NOAAGrid effort), and the NOAA Coastal Data Development Center (where significant efforts are being made to add 3D visualizations to GIS systems). The NOAAGrid framework is such that any NOAA lab or group wishing to participate need only install the Globus Toolkit, be added as a resource to the NOAAGrid meta-scheduler, and the computational framework and grid services of the Grid will be available to them.

There are several ways in which this proposal fits within the NOAA HPCC program objectives. One is that the NOAA Grid facilitates exploitation of high-end NGI technologies such as Abilene, National Lambda Rail and the TeraGrid by improving access to these high-speed resources. A second is that the proposal calls for use of significant off-the-shelf products such as the Globus Toolkit, MPICH-G2, and ParaView. Finally, the establishment of these grid services would be a start to a true cyberinfrastructure workflow system for NOAA laboratories.

Investigator Moore leads advanced scientific visualization research at PMEL, and has been developing systems on diverse platforms such as the ImmersaDesk, the GeoWall, the Access Grid (through AGAVE, the Access Grid Advanced Visualization Environment), and has experience in developing parallel applications using MPI, using the Visualization Toolkit (VTK) and Java3D, as well as integrating visualization frameworks into existing software (e.g., integrating advanced 3D rendering algorithms and oceanographic tools into ESRI ArcGIS products for use in ecosystem modeling). Investigator Hermann has expertise both distributed-memory ocean modeling and visualizing the results. While he has been using stove-pipe solutions to satisfy his research requirements, he is willing to participate in this effort to move NOAA researchers onto a common framework for resource management. Investigator Hyder is a network specialist at NOAA-Boulder. He participated in the creation of the NOAAGrid, is implementing the Grid in IPv6, and will assist us in the effort to enable parallel socket data transfer. Investigator Mansell is an experienced storm modeler at the National Severe Storms Lab (NSSL), and can provide the connection to expand the NOAAGrid to Norman, Oklahoma. We are uniquely qualified to provide this project as a next step in NOAA's high performance computing arena.

Performance Measures:

This project will be successful if a Grid service providing an advanced scientific visualization application can be made operational. In particular, the service should run on the NOAAGrid, implementing the X.509 certificate authority for security, and use the Grid meta-scheduler for job submission. The service should read model output files in popular formats (netCDF, HDF, or plain-text) as well as in-situ data files, and it should provide interactive parallel composite visualizations of the type now used in legacy visualization systems.

Milestones/Deliverables

Month 01 - Install GridFTP and insure grid services are enabled
Month 02 - Write XML data file reader interfaces
Month 04 - Composite renderer compiled, first test of parallel rendering
Month 04 - Enable remote composite rendering using parallel sockets
Month 05 - Make renderer application available as Grid service