COL/CE/10

Real-Time Data Accessibility via RMI Client/Server Technology

FY 2004 Proposal to the NOAA HPCC Program

August 11, 2003

| Title Page | Proposed Project | Budget Page |

Principal Investigator: **Janet Gibson** (must be a NOAA employee)

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

Environmental Technology Laboratory
325 Broadway
R/E
Boulder, CO 80305

Phone: (303) 497-6148
Fax: (303) 497-6978

E-mail Address: Janet.S.Gibson@noaa.gov

Other Investigator 1 RichBeeler@noaa.gov

Other Investigator 2

Other Investigator 3

Proposal Theme: Technologies for Collaboration, Visualization, or Analysis

Signature 1 (required)	*Signature 2 (required)*	*Signature 3 (optional)*
Janet Gibson	Richard H. Beeler	William Neff
IT Specialist	IT Specialist	Director
Environmental Technology Laboratory	Environmental Technology Laboratory	Environmental Technology Laboratory

Real-Time Data Accessibility via RMI Client/Server Technology

Proposal for FY 2004 HPCC Funding

Prepared by: Janet Gibson

Executive Summary:

NOAA's Environmental Technology Laboratory (ETL) produces, through research field projects and operational test bed platforms, valuable real-time data from a wide spectrum of instrumentation that are currently of limited availability to collaborators and colleagues. These data are commonly used to support emergency and operational decisions. In order to increase accessibility of these data and therefore reach a larger user community, as well as to provide data products in a timely manner, a multi-tier client/server architecture built upon distributed objects needs to be developed and demonstrated.

Distributed objects, by capturing data abstractions early, allow for operating system dependencies to vanish, network security to be assured, temporal transmission latency to be minimized, and scientific instrumentation to move toward standardization. The code is reusable, maintainable and easily extensible while it also provides a migration path from existing data acquisition and distribution systems. With this architecture in place, remote control of the instruments becomes reachable.

A standardized system built upon distributed objects, would allow all data to be accessible from centralized servers, thus facilitating application sharing. This infrastructure provides an opportunity to integrate data from multiple and diverse sensors in a real-time genre implemented on a web-based platform. Distributed objects will provide data in an XML format such as SensorML, thus allowing for integration with COTS packages. This framework will allow for future development of data archival and historical access to data.

Problem Statement:

ETL has various remote sensing instruments that operate in test bed platforms and in research projects. These instruments gather real-time meteorological data that have important applications. Commonly these critical data are used in decision support and emergency planning. Although an in-depth discussion of the many projects that use ETL real-time data is beyond the scope of this proposal, a brief introduction to two such applications is included to emphasize the importance of these data. For the past several years ETL has participated in the CalJet/PacJet experiments which strive to better detect flooding condition precursors during west-coast winter storms. Project collaborators include the National Weather Service (NWS), CA Dept of Water Resources, CA Governor's Office of Emergency Services, Forecast System Labs (FSL), Dept. of Navy, NASA and others. A second important application is the Ground-

based Remote Icing Detection System (GRIDS) which when operational will issue icing hazard warnings to the FAA, the NWS and others. Reference the following web sites for more information about the CalJet/PacJet experiment and GRIDS project:

http://www.etl.noaa.gov/programs/1998/caljet/CALJET_overview.html,
http://www.noaanews.noaa.gov/magazine/stories/mag28.html,
http://www.etl.noaa.gov/technology/grids.

The public's need for "Timely Access to Weather and Water Information" is a critical theme in NOAA's Strategic Plan. This proposal will improve access and timeliness to real-time data acquired by remote sensing instruments to support these goals.

By introducing advanced software technology at the source of data acquisition, ETL will develop portable software modules that can be used on various data acquisition systems. Improvements in discovery, transfer, processing and display of these data will improve NOAA's ability to more quickly and accurately meet scientific objectives and deliver forecasts.

Application to HPCC Program Objectives

This proposal is an important step in providing better access to critical real-time instrument data generated by NOAA. Mission effectiveness is improved by having more complete real-time operational information and displays available when and where they are needed. This proposal directly addresses the HPCC Program Guidelines and the HPCC program primary goal of greater access to real-time information using advanced technologies. It also offers an opportunity for improving technology for access to critical data by building an infrastructure of quickly accessible real-time data distribution for use in decision support and emergency planning, thus fulfilling another of the HPCC goals.

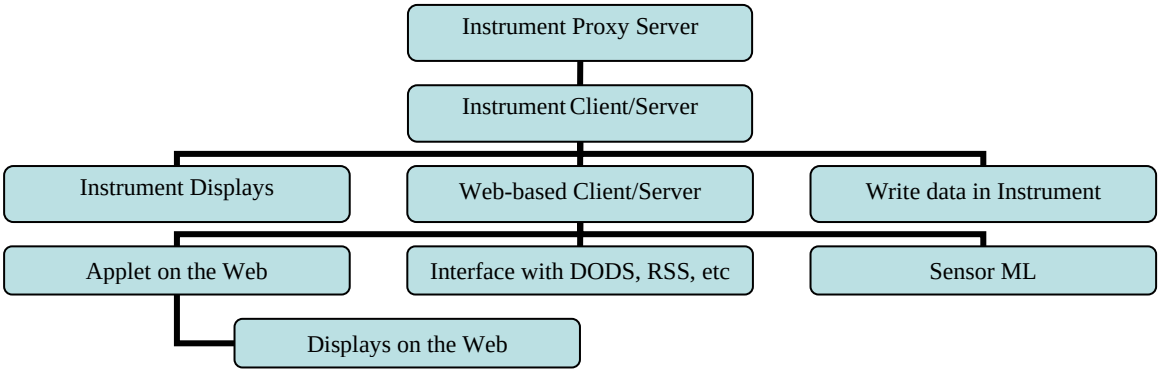
Proposed Solution:

Developing a platform independent multi-tier client/server based architecture provides a standard interface for ETL's instruments. By writing client/server code in Java using remote method invocation (RMI) technology, not only will the real-time data streams be available to any registered RMI process, but similar web-based client/server code running on the ETL web server would be able to ingest objects from multiple instruments and serve the data to graphical display software, cataloging software, RSS (see Ann Keane's funded HPCC proposal 2002), DODS, and other COTS packages. These applications would be shared among distributed systems including systems at the instrument locations.

Once the preliminary design process has been developed, one instrument's proxy server will need to be written to simulate real-time data. The proxy server simulates the actual real-time data stream from the instrument and serves it using either socket or RMI protocols. Therefore, the client/server process that interfaces with the instrument will need to ingest data via a socket or RMI. This local client/server will serve RMI data to the remote web-based computer and to any other RMI processes on the local machine. Once the instrument's client/server software has

been written; the proxy server and the local client/server software will need to be integrated and tested.

Remote web-based client/server software will be developed to further demonstrate the technology, by accepting and serving distributed objects via RMI. A Java applet will be designed and implemented that will allow users to access and display the data. The same application software that displays the data from the applet could be shared on the field system. Once demonstrated on one instrument, a suite of proxy servers will be written in order to extend this architecture to additional instruments. A SensorML application will be written and demonstrated. The diagram below illustrates the proposed system.



Analysis:

When implementing a distributed architecture solution, several options are available. These include: a Berkeley socketing solution, a scp solution, a CORBA solution, an RPC solution, a Java RMI solution. Choosing a Java RMI solution offers several benefits. The code is transportable across most platforms and the byte ordering problems disappear. This solution supports object oriented design, is inherently extensible, offers network security and promotes application sharing. It also offers a framework for interfacing with standard OpenSource and COTS packages.

It appears that RMI technology has not been fully explored as a viable medium for real-time data collection and dissemination. Although this technology should be able to satisfy most of the real-time data sets available, large bandwidth real-time data sets may “push the envelope” of this technology and will be carefully monitored.

A multi-tier client/server architecture would allow users in various locations to get to the data directly. It allows the data to be filtered if necessary and provides a secure encapsulation of the data. This solution also offers an upgrade path from the existing field systems. Since it can be ported easily between various operating systems, it provides a common solution for suites of diverse instruments. For all of the reasons listed, a Java RMI client/server solution is a state-of-the-art solution.

Alternatives

Solutions to this problem that have been investigated include:

A standard Berkeley socketing solution typically sends hardware specific binary data streams. This introduces the byte ordering problem referred to as "big-endian vs. little-endian", which requires additional machine dependent software.

Writing files and sending them via scp introduces an unacceptable time latency in real-time data distribution. It also decouples the system, i.e., the processes are unaware of data availability. Thus a process has to poll the system to detect arrival of data.

A Common Object Request Broker Architecture (CORBA) based system has been considered. It could accomplish the desired goals. Because CORBA offers a very extensive and general solution, it also has a lot more overhead. CORBA's disadvantages include slower system throughput, as well as software expense.

An RPC solution is generally stateless, not object oriented, and is machine dependent, thus introducing the byte ordering problem discussed in the Berkeley socketing alternative above.

Performance Measures:

Successfully demonstrate a single instrument's real-time data stream being served and displayed on the ETL web server via distributed objects.

Demonstrate SensorML as a reliable technology for real-time data acquisition applications.

Milestones

Month .5 – Extend existing client/server design to include aggregation of multiple data types
Month 1 – Develop client/server code for one instrument, proxy servers for several instruments and test
Month 1.5 – Develop web aggregation server client/server code and test
Month 2 – Develop applet and test
Month 2.5 – Develop SensorML application
Month 3 – System integrated, tested and demonstrated

Deliverables

- A web-based applet that uses real-time data to display products
- A demonstration of a client/server distributed object system
- SensorML application
- Project final report