COL/NE/03

Automated Network Synchronization of netCDF datasets

FY 2004 Proposal to the NOAA HPCC Program

August 11, 2003

| Title Page | Proposed Project | Budget Page |

Principal Investigator: **L. Charles Sun**

Line Organization: NESDIS
Routing Code: E/OC1
Address:

NODC
Bldg: SSMC3 Rm: 4635
1315 East West Hwy
Silver Spring, MD 20910-33282

Phone: (301) 713-3272 x111
Fax: (301) 713-3302
E-mail Address: Charles.Sun@noaa.gov

Joe Sirott
Joe.Sirott@noaa.gov

Donald W. Denbo
Donald.W.Denbo@noaa.gov

Proposal Theme: **Technologies for Collaboration, Visualization, or Analysis**

L. Charles Sun
Oceanographer
NOAA/NESDIS/NODC

H. Lee Dantzler
Director
NOAA/NESDIS/NODC

Automated Network Synchronization of netCDF datasets

Proposal for FY 2004 HPCC Funding

Prepared by: Charles Sun and Joe Sirott

Executive Summary:

Argo is a major component of the global ocean observing system which will ultimately consist of an array of 3,000 free-drifting profiling floats. Floats are deployed by Argo principal investigators who transmit data from their floats every 10 days to a national Argo Data Assembly Center (DAC), which aggregates the data and sends it forward to the two Global Data Assembly Centers (GDACs) (in Brest, France and Monterey, California) from which NODC acquires the data for archiving by version. Argo file transfers are large and are accomplished by FTP in the official Argo netCDF format. These FTP transfers are very slow, subject to interruption, and there is no synchronization of data between source and receiver to assure that the transmission was complete or correct.

We propose to implement a simple algorithm for network synchronization of netCDF data files that will decrease transmission time and ensure accurate and complete transmission by transmitting only the parts of the file which have been updated. The algorithm takes advantage of the block based structure of netCDF and cryptographic hash signatures to transfer only those netCDF blocks that have changed since the last synchronization.

Although our effort is directed towards Argo data transfers, there are other applications for our algorithm, including mobile users, such as disaster response teams, who frequently have intermittent and/or low bandwidth connections to the Internet and need to be able to quickly update local datasets for later offline use and researchers who want local access to large model datasets and need to update their local data with remote data.

Problem Statement:

As access to distributed data grows, it is increasingly important that users have access to tools that allow data to be synchronized. Data centers are increasingly required to archive data from distributed sites; mobile users frequently have intermittent and/or low bandwidth connections to the Internet and need to be able to quickly update local datasets for later offline use; researchers want local access to large model datasets and need to update their local data with remote data as new model timepoints are generated without downloading large amounts of data. This problem is particularly evident in near-realtime netCDF-formatted file transfers of Argo profiling float data from the Argo principal investigators to the national data assembly centers to the global data assembly centers to the archive center.

One difficulty with large Argo FTP file transfers is guaranteeing that data at the receiving center accurately mirrors the data from the sender. Argo datasets contain a large number of small netCDF files (ranging from a half MB to one MB when fully operational. Currently, NODC must check (and download) a total of 1.5 GB for about 1200 floats every day. It takes about 3.5 hours (clock time) to finish the job. Once the Argo array is completed in FY 2006, there will be a total of 3000 floats, and total file sizes will be more than 3 GB daily.

Current technology relies on FTP transfers and file timestamps – the last modified date of files at the GDAC is checked and if a file timestamp has changed the entire file is downloaded to the archive center. While this has the benefit of simplicity, it can also be time consuming; for example, a single FTP transfer can take three hours, and not infrequently times out and must be completely reinitiated. These transfers can also be unstable; for example, if an administrator at a GDAC copies the files to a new disk and forgets to reset the timestamps, software at the archive center would attempt to download the entire profile database. For larger files, a small change in an attribute might result in the unnecessary download of an entire file.

Our proposal would mitigate the costs and difficulties of this data mirroring, and would also have the side benefit of validating the integrity of files after a data transfer occurs.

Relationship to NOAA HPCC objectives: This proposal addresses the primary HPCC goal of "*… providing greater access to its vast holdings of real-time and historical information to users in a more complete, more usable form, and much more timely manner through increased use of advanced technologies associated with the Internet and follow-on networks"* directly by providing more reliable and thorough file transfers. As part of the "**Technologies for Collaboration, Visualization, or Analysis**" theme, the proposal would "*require the cooperation of multiple sites*", "*demonstrate new techniques for working with NOAA data and information,*" and "*develop tools that are extensible, scalable, and available for easy deployment throughout NOAA*." The proposal would also enhance the "**Disaster Planning, Mitigation, Response and Recovery**" portion of NOAA's mission, as it would provide a mechanism to allow mobile users improved access to updated real time, rapidly changing datasets.

Proposed Solution:

We will develop a Web application and client program that will automatically synchronize netCDF datasets between the client and the web application. An administrator will place netCDF files under the control of the web application; the web application will then calculate a cryptographic hash (probably MD5) of each file. If the file is larger than a certain size, then the block based structure of netCDF will be utilized and hashes will also be calculated on the header section of the netCDF file as well as all of the data blocks that contain the netCDF data hyperslabs. If the file is modified, then the hashes will be recalculated.

The client will then use these hashes for synchronization. A researcher or program will be able to request the synchronization of one or more files from the Web application. The client will determine if the user has already downloaded the file; if the file doesn't exist on the client side, a simple HTTP transfer will be used to download the file and the operation will be completed. If the file does exist, the client will calculate cryptographic hashes on the local file and compare them with the hashes on the server. Any differences will cause the client to download the netCDF hyperslabs that have changed. No additional data need be downloaded. The local netCDF file will be updated with any changed hyperslabs and, optionally, the integrity of the newly created local file will be checked by recalculating the hashes on the local file and comparing them with those on the server.

Both the client and Web application will be written in Java. The Web application will utilize Java servlet technology and will have configuration information stored in a XML configuration file. The client will have a Python language scripting interface and, resources permitting, a simple Swing user interface will be developed.

Development will proceed as follows:

1. Experiment with various hash algorithm tradeoffs.
2. Experiment with different file/hash database formats (XML vs. relational database).
3. Prototype of Web application
4. Prototype of client
5. Add scripting interface to client
6. Test on transfers between Argo Data Assembly Center at PMEL and the Argo Data Archive Center at NODC
7. Add Swing user interface (resources permitting).

Analysis:

Rationale: Existing mirroring strategies rely on FTP and unreliable file timestamps, with are slow, error prone, and do not provide synchronization to assure that data transfer is complete or accurate. A more reliable, lower maintenance strategy is needed.

Scope: The Argo program is global and international in scope. However, this technology has the potential to benefit many users outside the Argo program, especially mobile users. For example, workers in a disaster relief situation would be able to connect to the Internet via a low bandwidth phone line, quickly update their local real time datasets with updated datasets on remote servers, and return to work in the field.

Performance Measures:

This project will be successful if it demonstrates a prototype of a system for secure, accurate, timely synchronized network transfers of Argo netCDF files from national and international data assembly centers to the archive center at NODC.

Milestones/Deliverables

Month 6 – Working, proof of concept prototype
Month 12 – Final product