8.4 AN ADVANCED LINUX PROTOTYPE OF AWIPS

Herb Grote, Darien Davis, Carl Bullock
NOAA/OAR/Forecast Systems Laboratory,
Boulder, Colorado

Jason Tuell
NOAA/NWS/Office of Science and Technology
Silver Spring, Maryland

1. INTRODUCTION

In early 1998, FSL recognized the potential of the Linux Operating System to help improve AWIPS performance. Initial tests were performed at FSL to identify the risks and magnitude of the effort to port the AWIPS D2D software from the Hewlett Packard platform to Linux (Grote, 1999). Although significant code changes were identified, the benefits were seen to outweigh the cost of the software development effort. FSL successfully ported the AWIPS workstation code in 2000, and worked with the National Weather Service (NWS) to test a Linux workstation in several operational offices (Davis et al., 2001). Since then FSL has been anxious to develop a complete AWIPS prototype system on a Linux platform. The prototype would include hardware and software architectural changes to improve system performance, maintainability, and expandability.

2. DEVELOPMENT AREAS

2.1 *Linux Architecture*

In 2004 FSL, with the support of the NWS, started a two year project known as ALPS (Advanced Linux Prototype) to complete the data acquisition and processing portion of theLinux port and to accelerate the deployment of an all-Linux system to NWS field offices. The basic architecture was coordinated with NGIT (Northrop Grumman Information Technology) technical staff and then presented to the NWS for review. The NWS has endorsed the proposed architecture, and is proceeding with the deployment of the hardware and phased implementation of the system. The hardware architecture used for ALPS development is shown in Fig. 1. It closely resembles that proposed for the NWS field deployment. The heart of the system is the NAS (Network Attached Storage) device that stores all meteorological and other essential datasets. A high speed ethernet connects the NAS to the other processing elements in the system. The Linux servers perform the heavy data processing such as data decoding, grid preparation for GFE (Graphical Forecast Editor), and data transformations. The performance of the workstation is improved by faster access to the data on the NAS, with less contention with other data processing, and improved processor performance on the workstation. Preliminary tests indicate a greater than two-to-one performance improvement over the current AWIPS HP and Linux architecture. In order to retain the high system availability, Open Source failover software (heartbeat) allows the automatic failover of the servers.

Corresponding author address: U. H. Grote, Forecast Systems Laboratory, NOAA/R/FS4 325 Broadway, Boulder, CO 80305-3328

The AWIPS relational database (Informix) is being replaced with PostgreSQL, which will run on the servers. Unlike other data, the data stored in PostgreSQL will reside on disks directly attached to the servers. The PostgreSQL software will perform the data replication to allow system failover.

2.2 *Distributed Data*

In addition to the redesign of the hardware architecture, ALPS also implements some changes to the basic software infrastructure. One of these changes is driven by the requirement for AWIPS to be able to acquire the latest models and observed datasets. These datasets tend to get larger as the resolution of the models and sensors increases. The current approach of sending nearly all of the datasets over the SBN (Satellite Broadcast Network) is not the most efficient solution. ALPS is implementing an approach that allows selective data to be "pulled" from remote data servers.

The new data management paradigm allows certain time critical datasets to continue to reside on the local storage device, while less time critical data can be retrieved from servers at remote locations. The core of this "pull" technology is the OPeNDAP software maintained by OPenDAP, Inc. with technical support from UNIDATA. The software makes it possible for applications to use NetCDF calls to retrieve a specific subset of data from remote OPeNDAP servers. Catalogs in XML format contain the necessary information to route the data requests to a server that has the desired data. In ALPS, the data address can be a local disk or a URL at a remote server. The OPeNDAP software on the remote server checks its catalog, and retrieves only the requested data from its database (not the entire file). At first, the retrieved data will be stored in NetCDF format on the remote servers, but preprocessing at the server will eventually make it possible to retrieve data in other format, such as GRIB.

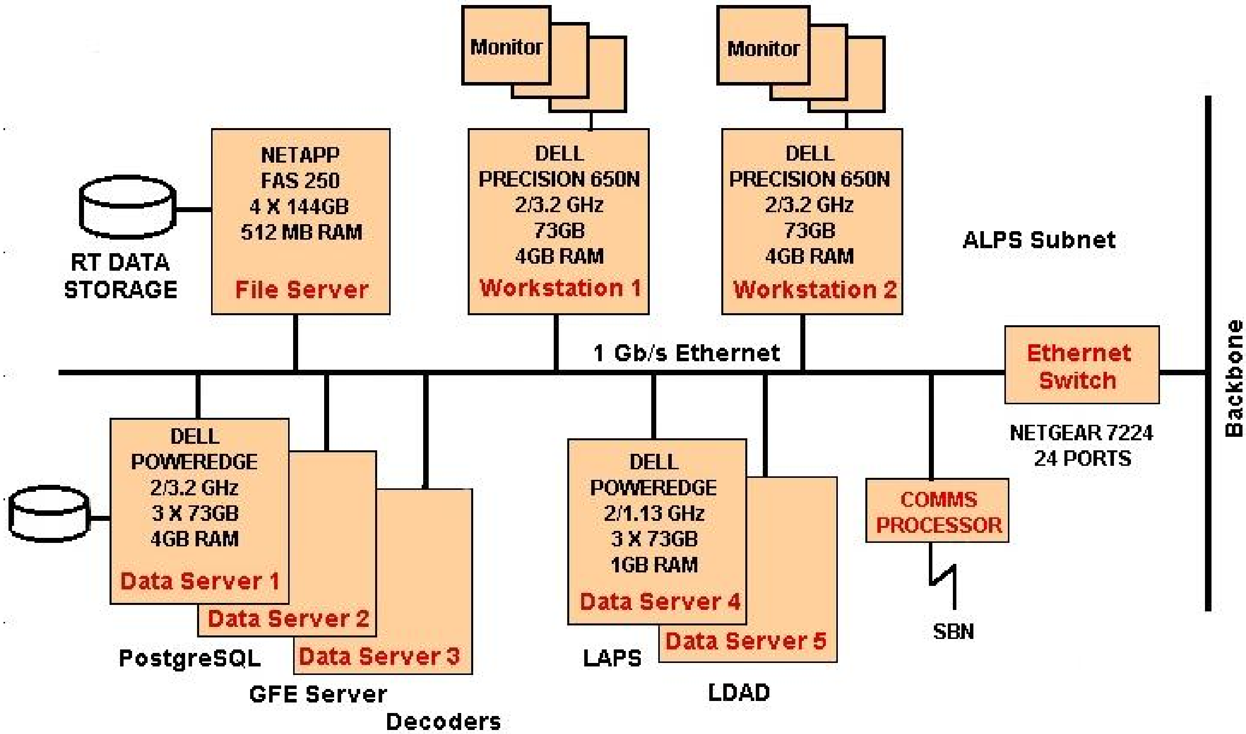

Fig. 1. The ALPS development system depicting the Network Attached Storage device and 1 Gigabit Ethernet connecting the Linux processors.

In order for the user to know what data are available at the remote servers at any time, a new inventory mechanism is being explored. The two approaches under consideration are periodic queries (searches) by the WFOs for new data, and a scheduled broadcast via the SBN (possibly, at one-minute intervals) of all new data that are available at a server. A prime candidate for the initial server is the NOMADS server at the National Centers for Environmental Prediction (Davis et al., 2005) since it maintains a good set of current model data and is currently supporting external users.

The proposed "pull" technology can be effective in a real-time environment only if the time it takes to retrieve a particular dataset is relatively fast. The current AWIPS WAN technology will cause this approach to be of limited use because of its slow network response. However, it is expected that this network will be upgraded with newer technology that offers a significantly higher bandwidth. ALPS is modifying the AWIPS code in anticipation of the new NWS network implementation.

2.3 *Applications Programmer Interface*

ALPS also features a new API (Application Programmers Interface). This interface is designed to significantly simplify the integration of new applications. It also makes it easier to add new functionality, such as graphic annotation, and makes it possible to incorporate external functionality into AWIPS. This new API should make it possible to converge NMAP (a component of N-AWIPS) and AWIPS capabilities.

The applications are separate processes that communicate directly with the workstation display process known as the IGC (Interactive Graphic Capability) via socket communications (Fig. 2). The API avoids the D2D inter-process communications in order to limit the number of D2D libraries that an application has to link with to create an executable module. The API provides low level interfaces in C that provide basic drawing primitives, such as lines and polygons, and higher level primitives such as fronts. It also allows the application to determine the status of the IGC in order to create the appropriate display. The C language was chosen since it is relatively easy to create C++, FORTRAN, Java, or other "wrappers" to allow applications to be written in more than one language.

A new ALPS "depictable" is being written in C++ that converts the requests from the application to C++ methods in the IGC. The ALPS depictable will be able to use many of the existing IGC methods (capabilities) to perform the requested actions. It will also use new IGC methods to implement more advanced graphic features. In order to implement true multicolor graphics, and graphic layering (not to be confused with graphic overlays), significant modifications are being made to the IGC. All graphics will be treated as images and merged with other displays using alpha blending (transparency).

Once the new API has been fully implemented, the D2D Extension interface will be deprecated. Existing extensions will be rewritten to use the new API and to eliminate the complex IGC interface that currently exists. The two most complex extensions in D2D, WarnGen and the Interactive Skew-T, will be modified last. This transition can be gradual since the extension interface will coexist with the new API for a period of time.

A very attractive feature of the new API design is that applications will be treated as "plugins." New applications can be placed in a predefined directory and automatically integrated with D2D when the system is initialized. Unless the developer prefers a specific location on the menu for the application, a default menu will automatically be updated with the new application name. The D2D system will be aware that a new process has been started and will manage it appropriately.

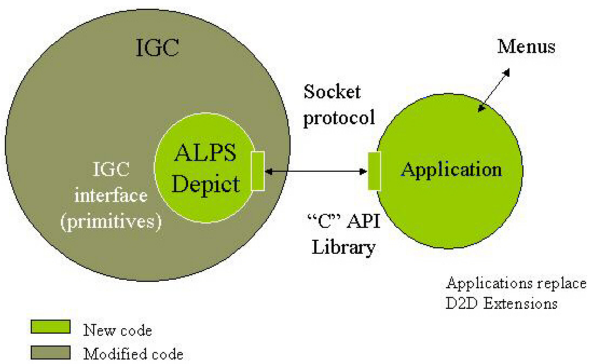

Fig. 2. Simplified diagram of the ALPS application interface

3. SUMMARY

The ALPS prototype is being developed to help the AWIPS project deal with existing or anticipated problems. These range from overloaded processors to inability to acquire and display new model and sensor data. The porting of AWIPS to an all-Linux architecture is also a risk reduction activity that should expedite the NWS transition to Linux in the field. The concept of distributed data and the development of a new API that includes a drawing capability are expected to provide greater flexibility in acquiring and displaying new data. The new API is also expected to reduce software development and maintenance costs in the long term. The ALPS project has focused on changes to the AWIPS infrastructure that are beyond the scope of the scheduled operational builds.

4. REFERENCES

Davis, D., U. H. Grote, M. Biere, 2001: Porting AWIPS to Linux. 17th Inter. Conf. on IIPS for Meteorology, Oceanography, and Hydrology, Albuquerque, NM, Amer. Meteor. Soc., *84-86.*

Davis D. L., T. B. Kent, G. J. Edwards, J. E. Ramer, U. H. Grote, C. S. Bullock, E. J. Mandel, J. P. Tuell, and T. Hopkins, 2005: Investigating using NOMADS in an AWIPS environment. 21st Inter. Conf. on IIPS for Meteorology, Oceanography, and Hydrology, San Diego, CA, Amer. Meteor. Soc., *2.21*

Grote, U. H., 1999: AWIPS/LINUX: An Advanced Weather Interactive Processing System on Linux. Seventh Workshop on Meteorological Operational Systems, European Centre for Medium-Range Weather Forecasts, Reading, England, *177-180*