P2.25

RECENT ADVANCES IN THE FSL CENTRAL FACILITY DATA SYSTEMS

Robert C. Lipschutz* and Christopher H. MacDermaid*

NOAA Research − Forecast Systems Laboratory
Boulder, Colorado

*In collaboration with the Cooperative Institute for Research in the Atmosphere (CIRA), Colorado State University, Fort Collins, Colorado

1. INTRODUCTION

The NOAA Forecast Systems Laboratory (FSL) Central Facility ingests, processes, stores, and distributes large volumes of conventional and experimental meteorological data in support of FSL's research, development, and technology transfer mission. Over the years, as data requirements have changed and hardware and software technologies advanced, the real-time systems responsible for these data have evolved, as well. Recent advances to promote reliability and maintainability include the use of such Open Source packages as High-Availability (HA) Linux, SystemImager, and Concurrent Versions System (CVS). Unidata's Local Data Manager (LDM) plays a prominent role throughout the system to provide reliable and easily configurable data transport and event triggering. Finally, the development of the Object Data System(ODS) (Hamer, 2005), which is replacing the earlier NIMBUS system (Wahl et al., 1997), has streamlined software development and configuration management for the many point, gridded, radar, and satellite data sets processed within the Central Facility.

This paper provides some insight into the scope of FSL's real-time data system, describing the current state of the system and discussing the range of services provided. We also highlight several solutions that have been implemented recently, and offer several ideas for the future.

**Corresponding author address:* Bob Lipschutz, NOAA/FSL, R/FST, 325 Broadway, Boulder, CO 80305. Robert.C.Lipschutz@noaa.gov

2. SERVICES

The present system comprises some 40 computers that provide a wide variety of ingest, computing, storage, and distribution services, as depicted in Fig. 1. Ingest subsystems acquire data from satellite down-links, dedicated land lines, and the Internet, amounting to about 150 GBytes of data per day. Data are available to FSL users on a Network File System (NFS) server and, in some cases, via LDM. A wide community of collaborators outside FSL can also acquire data from FSL through a variety of distribution methods, including LDM, FTP, and the web-based NOMADS. Most data sets are also saved onto a Mass Storage System (MSS) for later study by FSL researchers.

2.1 On-line Storage

The central hub for data within FSL is a Network Appliance FAS940 NFS server, generally known by its main data tree, /public. The /public file system is mounted by dozens of desktops and compute servers throughout the lab to provide real-time data to application and model developers. Depending on the data set, typically 2-5 days' data are stored, amounting to nearly 300 GBytes. Additionally, a smaller second file system, /integration, provides a location for testing data from new or updated applications. Total available storage space on this server will be expanding early in 2005, from about 500 Gbytes to over 1 Tbyte.

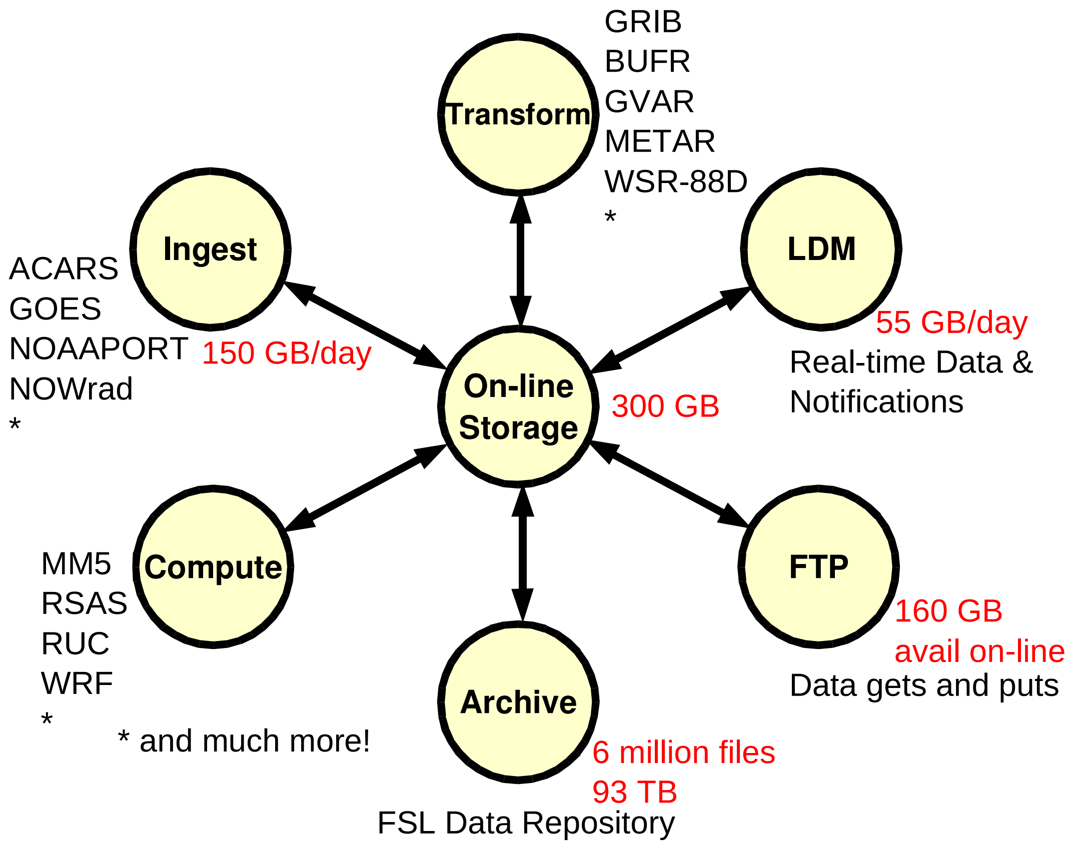

Fig. 1. Central Facility data services.

2.2 Ingest

A variety of data ingest subsystems acquire the raw data and products that are subsequently decoded or transformed for storage on /public. A list of data sources for many of the key datasets is shown in Table 1.

FSL ingest systems include a GOES direct read-out satellite ground station that is jointly operated with NOAA Space Environment Center for receiving GOES Variable (GVAR) data from both GOES-10 and GOES-12. Two other satellite receiver systems also operate within the Central Facility, acquiring the NWS's operational NOAAPORT data stream and the WSI Corp.'s NOWrad national mosaic radar products. A dedicated TCP/IP link to ARINC, Inc. provides the important ACARS aircraft observation data stream. Numerous other datasets are acquired using Unidata's LDM protocol and FTP.

2.3 Transform

Data acquired from the various sources typically must be decoded or otherwise transformed to facilitate their use by users' applications. In many cases, the processing converts data from a receipt format, such as ASCII text or GVAR, to Unidata's NetCDF format, which is more easily accommodated by many applications. For model grids in GRIB format, selected fields may be extracted and put into NetCDF to reduce the size of the resulting file. Also to reduce file size, GOES GVAR data are optionally subsampled and/or sectorized to a specified window during the conversion to NetCDF.

Decoders for many data types have been developed, operating on such data formats as ACARS, METAR, SAO, RAOB, PIREP, Airmet, Sigmet, TAF, Maritime, VAD, BUFR, GRIB, GVAR, and WSR-88D Level II. While a number of decoders still run in the legacy NIMBUS

Table 1. FSL Data Sources

Provider	***Protocol***	***Data type***
AirDat LLC.	LDM	TAMDAR aircraft weather obs
ARINC Inc.	TCP/IP	ACARS aircraft weather obs
AWC	LDM	Satellite, aviation products
CONDUIT	LDM	Model grids (Eta, GFS, RUC)
FNMOC	LDM	Model grids (NOGAPS)
GOES-10/12	Downlink	GVAR satellite imager/sounder
NCAR	LDM,FTP	Aviation weather products
NCEP	FTP	Model grids
NESDIS	FTP	Satellite products
NOAAPORT	Downlink	NWS/AWIPS data stream
NSSL	FTP	3-D radar tiles
NWSTG	FTP	Model grids
SSEC (UofWisc)	FTP	Satellite products
U. of Maryland	LDM	CRAFT Level-II WSR-88D radar
WSI Corp.	Downlink	NOWrad radar

environment, we continue to migrate data handling to the ODS approach, discussed below. Notably, the GRIB, GVAR, WSR-88D Level II and BUFR data handling have all benefited from the more consistent and easily configured ODS.

2.4 Local Data Manager (LDM)

LDM, a package integral to FSL processing, is used for both data ingest and distribution. Input and output via LDM presently accounts for some 55 Gbytes of data traffic per day. As noted above, a number of data sets are acquired via LDM from such providers as NCAR, CONDUIT and FNMOC. In addition, several FSL hosts provide outbound LDM connections to external clients. Recipients of LDAD and MADIS datasets include NCAR, many NOAA offices, and a large number of universities and private organizations (Miller et al., 2005). GOES, profiler, raw ACARS, and various model data are also distributed to a smaller set of recipients within NOAA and NCAR.

In addition to sending and receiving data, LDM runs on many platforms throughout the Central Facility, providing data transport among local hosts, and for process control using LDM's 'pqact' utility. LDM ingest and distribution hosts communicate with LDM processes on several compute platforms for moving data into and out of the system. Further, data event notification messages that are generated by the ingest and transform applications are available throughout the system to provide a distributed triggering mechanism.

Beyond simply moving data and triggering subsequent processing, many ODS client applications map directly onto the LDM queue to extend the basic LDM capabilities for a variety of data handling needs, as described by Hamer, (2005). For example, the program LdmGrib2Flat efficiently extracts and stores NOAAPORT GRIB data from the LDM queue, creating a dynamically generated directory tree based on the GRIB center/subcenter/model/grid identifiers found within the data (GRIB2 is similarly handled, with the addition of a level for "Discipline"). Another client, LdmFdr2Tar, streams data (e.g., NOAAPORT grid messages) to tar files that are ultimately compressed and moved to FSL's Mass Store System as part of the FSL Data Repository (FDR). Other daemon applications continually inject such data as ACARS reports into LDM for transport to decoding processes.

The notification messages generated within the system are used to trigger, in turn, a series of actions as each step completes. For example, completion of a GOES GVAR image frame triggers the generation of a GOES raw NetCDF file that then activates a remapping application. A key optimization within the system leverages these notification messages to copy product files generated on a local disk to the /public NFS server; this is particularly beneficial for large gridded and satellite data sets. Further, the script responsible for copying the product files also moves existing instances on /public out of the way to avoid clobbering a file that may be currently open by an application.

Data and notifications are also distributed to general FSL users for triggering their own event-driven processing applications based upon data arrival.

2.5 File Transfer Protocol (FTP)

Numerous datasets are acquired from remote FTP servers at the NWS Telecommunications Gateway, NCEP, NESDIS, SSEC, NCAR, and other sites. The development and configuration management of scripts for acquiring new data has been facilitated by a set of FTP tools that can be applied for many different datasets. For example, all GRIB files are fetched and processed with a single script, ftpFlatGrib, that uses a standard configuration file method for specifying server information, remote and local path, filename pattern, and notification key string. These tools provide mechanisms for keeping track of which files have already been acquired, as well as for killing earlier FTP jobs that appear to be hung.

The Central Facility also supports FTP for distribution of many datasets to collaborators. Datasets available via FTP include LDAD and MADIS (MacDermaid et al., 2005), and grids from several models run at FSL.

2.6 NOMADS

FSL is a collaborator in the NOAA Operational Model Archive and Distribution System (NOMADS) concept of distributing gridded datasets using web-based methods (Rutledge et al., 2003). NOMADS serves data using OPen source project for Network Data Access Protocol (OpeNDAP). Datasets available from FSL's NOMADS servers include MADIS and RUC (see http://nomads.fsl.noaa.gov).

2.7 Data Saving

While FSL is not an official NOAA data archive site, the lab has long maintained systems to save real-time data for extended periods, mainly to facilitate case review within FSL, and to support Unidata's COMET forecaster training program. The FSL Data Repository (FDR) system, which has recently been streamlined using ODS methods, saves to FSL's MSS all raw NOAAPORT data in hourly compressed tar files that can be restored and played through an AWIPS case generation system. All FTP'd GRIB files, as well as selected GRIB data received via LDM from CONDUIT, are also saved. Finally, many /public data sets are tar'd and compressed at the request of FSL users.

3. OPEN SOURCE SOLUTIONS

FSL's data systems continue to advance in reliability and maintainability, largely by leveraging Open Source solutions.

3.1 High-Availability (HA) Linux

While computer system component reliability has significantly improved in recent years, we continue to search for solutions that reduce "single points of failure." Toward that end, a number of data system servers have recently been paired using the Open Source package called High-Availability Linux (http://www.linux-ha.org).

In an HA configuration, one host is the designated primary, while the other is the failover, or backup, host. Both hosts run the HA heartbeat software, sending messages to each other over a serial link. If the heartbeat stops on the primary host, then its mate will detect the fault and automatically kill the power on the primary and assume the primary host's role, starting all the necessary applications. Importantly, the failover host takes over the pair's virtual IP address, thus enabling downstream clients to connect to the alternate host using the same address. The strategy of killing the power on the original primary ensures that it doesn't try to reassert itself before its problem can be assessed, or, worse, try to partially function as primary while in a bad state. Once heartbeat is restarted on the normal primary system, *e.g.*, after hardware has been repaired, the roles switch back to the original arrangement. Email notices are sent to system administrators and other support staff for all heartbeat events.

While the HA backup host would normally be idle except for the heartbeat processes, we instead manually start the real-time software on that system so that the LDM queue and other local data are fully populated. As a result, failovers result in minimal data loss. In situations where only the primary system should run, for instance in moving files to /public, software on the backup host avoids running by checking for the existence of a PRIMARY flag file.

In addition to increasing system availability, the HA arrangement also facilitates software integration by providing an identically configured alternate platform for short-term tests.

3.2 Concurrent Versions System (CVS)

CVS is a popular Open Source package for software version control. (http://www.cvshome.org). While CVS is typically used for maintaining source code in a team environment, we have found it particularly valuable in configuring and installing run-time files on the many Central Facility real-time systems. The goal of our CVS strategy is to allow any platform to assume a real-time processing role simply by acquiring the appropriate files from the repository, and installing them using a custom shell script, updatesys, that knows where the various types of files belong. We also further manage files by using special tags to indicate which versions are ready for integration or production.

The repository is organized by host role for various types of files, including LDM and application configuration files, crontab files, and lists of directory paths and symbolic links that need to be created. In addition, the CVS 'modules' file specifies the set of binary files and scripts needed for each role. To install new or updated files on a target host, updatesys obtains the role-specific files from the CVS repository, and copies them to the appropriate run-time locations. The installer will also create any specified directories and symbolic links, if necessary. For the HA pairs, the same set of files is installed on both nodes.

3.3 SystemImager

In conjunction with specifying run-time files in CVS, we also employ SystemImager (http://www.systemimager.com), an Open Source package for automating Linux Operating System installation and production deployment. A new platform can be quickly brought up with a nominal operating system image using SystemImager, and then customized into its real-time role with CVS and updatesys. For systems that have not been migrated to HA, we reserve several spare platforms that are available to be re-imaged and configured, as needed. To failover to one of these nodes, we simply load the basic system with the correct hostname and IP address, run updatesys to install the appropriate run-time environment, and reboot the system. While more convenient than loading an operating system from scratch, this procedure is still manual, and so is not a long-term strategy for us.

4. OBJECT DATA SYSTEM (ODS)

As a follow-on to FSL's earlier NIMBUS system, the ODS object-oriented methods streamline software development and configuration management, and provide significantly better throughput performance, as detailed by Hamer, (2005).

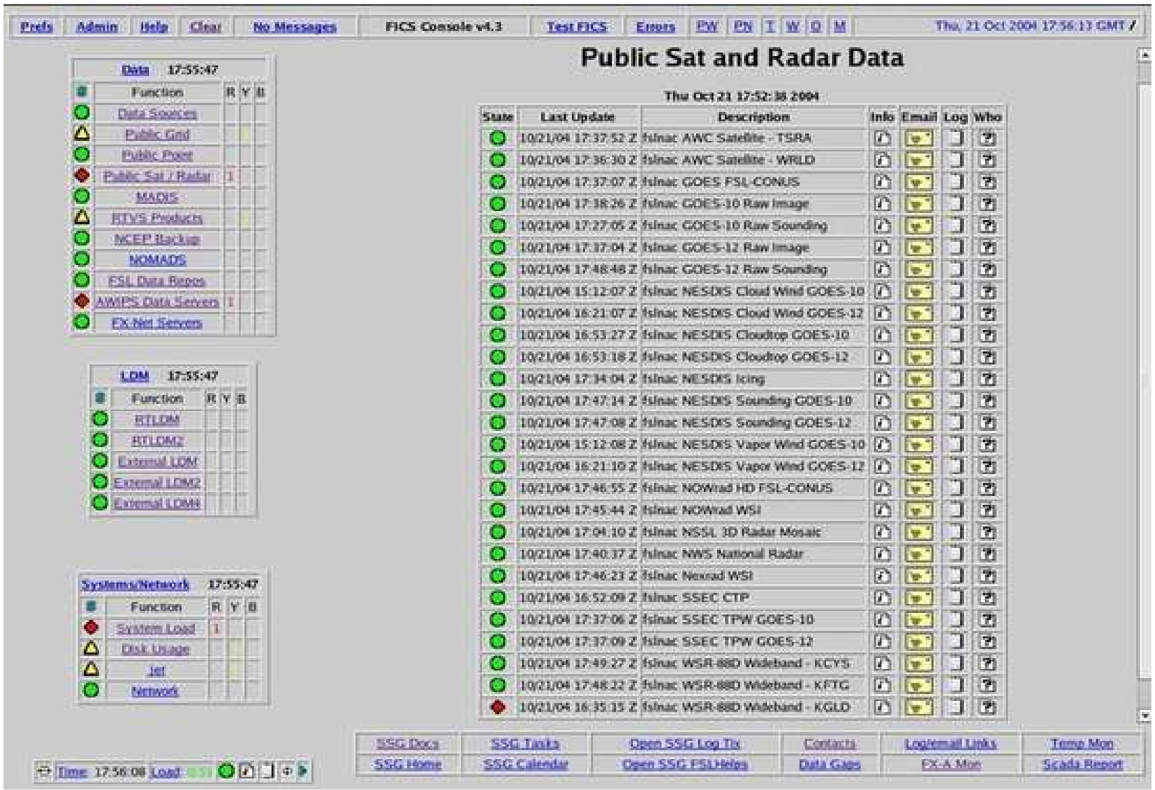

Fig. 2. The FICS monitor display.

As mentioned above, several ODS clients directly monitor the LDM queue to find needed data. Gridded data in GRIB format are dynamically stored to appropriate directories, based upon information within the GRIB headers. Similarly, WSR-88D Level II data are continually directed to locations determined by the radar site identifier.

ODS data transform applications also have been designed for generality. In particular, the ODS GRIB-to-NetCDF scheme eliminates special coding for any new model data; rather, converting GRIB to NetCDF is entirely configuration-driven by specifying field attributes within the NetCDF 'cdl' description file. For retrospective data processing, ODS also utilizes appropriate GRIB tables for the specific time frame of the data.

5. MONITORING

Real-time system monitoring in the FSL Central Facility continues to evolve, as well, but largely relies on a long-established scheme. The system, known as FICS (Facility Information and Control System), deploys agent scripts on the real-time hosts that regularly report status information to a central FICS server. The reports provide 'Up,' 'Attention,' and 'Down' indications for numerous system components. Scripts on the server then parse these reports to update a set of HTML pages that depict system conditions. The pages are displayed and refreshed as part of a CGI script, routinely viewed by the FSL Operations staff, who are responsible for troubleshooting problem conditions.

Figure 2 illustrates a typical FICS display. Links in the summary tables, organized by category on the left side of the page, provide access to more detailed information on the right. Further information about each dataset, including detailed troubleshooting procedures, can be obtained by clicking on the "Info" link for each dataset. Other links provide operators with preformatted email and trouble ticket templates to facilitate sending outage notices to users, and to open problem tracking tickets.

In addition to the FICS file agent that compares file age against configurable attention (yellow) and down (red) thresholds, other agents report on the state of LDM connections, disk usage, system loading, and network devices. The health of NOAA's high performance computer at FSL is monitored, as well.

6. FUTURE DIRECTIONS

FSL's Central Facility data systems continue to expand with each new dataset acquired or distributed. Using tools already at hand, most of these data sets are easily accommodated, with the caveat that computer resources are finite and data files, especially grids, are continually getting larger. Unfortunately, with the present HA arrangement, expanding our processing capacity necessitates installing two computers at a time into a computer room with finite power and cooling. As a result, we view the HA technology as an interim step toward a truly scalable processing environment.

We are now beginning to investigate the Open Source virtual clustering technology known as Single System Image (SSI) (http://www.openssi.org). The SSI concept links a set of computers so that the cluster appears as one big computer, providing scalability and manageability, along with high availability. The cluster's distributed processing environment offers load leveling, and cluster-wide process control, signaling, and communication. Programs that are not "cluster aware" can run in this environment without modification. The challenge for us will be to discover the best use of SSI within the context of our LDM-dependent ODS environment.

In the coming year, we also intend to finish migrating legacy NIMBUS applications to the ODS environment. In particular, a number of data decoders (e.g., for translating ASCII METAR and RAOB data types) must be ported. The goal will be to wrap the existing core decoder software within ODS C++ object methods.

7. REFERENCES

Hamer, P., 2005: FSL Central Facility data system concepts. *Preprints, 21st International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, San Diego, CA., Amer. Meteor. Soc.

MacDermaid, C. H., R. C. Lipschutz, P. Hildreth, R. A. Ryan, A. B. Stanley, M. F. Barth, and P. A. Miller, 2005: Architecture of MADIS data processing and distribution at FSL. *Preprints, 21st International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, San Diego, CA., Amer. Meteor. Soc.

Miller, P. A., M. F. Barth, and L. A. Benjamin, 2005: An update on MADIS observation ingest, integration, quality control, and distribution capabilities. *Preprints, 21st International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, San Diego, CA., Amer. Meteor. Soc.

Rutledge, G. K., J. Alpert, R. Stouffer, B. Domenico, L. Buja, D. N. Williams, B. Doty, M. Kafatos, and S. Hankin, 2002: The NOAA Operational Model Archive and Distribution System (NOMADS). *Preprints, 18th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, Orlando, FL., Amer. Meteor. Soc.

Wahl, G. M., R. L. Minton, and P. A. Mandics, 1997: NIMBUS: An advanced meteorological data acquisition, processing, and distribution system. *Preprints, 13th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, Long Beach, CA., Amer. Meteor. Soc.