

Managing Data Quality in a Terabyte-scale Sensor Archive

Bryce Cutt
University of British Columbia Okanagan
brycec@interchange.ubc.ca

Ramon Lawrence
University of British Columbia Okanagan
ramon.lawrence@ubc.ca

ABSTRACT

Sensor networks collect vast amounts of real-time information about the environment, business processes, and systems. Archived sensor data is valuable for long-term analysis and decision making, which requires it be suitably archived, indexed, and validated. In this paper, we describe a general approach to managing and improving data quality by the generation and validation of metadata and the logging of workflow events. The approach has been implemented within a system archiving terabytes of U.S. weather radar data. The data quality system has resulted in the detection of data errors while simplifying the administration of the complex archive system.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; J.2 [Physical Sciences and Engineering]: Earth and atmospheric sciences; H.2.8 [Database Applications]: Scientific Databases; H.2.7 [Database Administration]: Data warehouse and repository

Keywords

data quality, sensor network, real-time warehouse, archive, scientific data, hydrology

1. INTRODUCTION

Real-time sensors are collecting a vast amount of information on our environment, our businesses practices, and our networks and technology. Most data collected is used immediately to detect changes in trend or events of significance requiring response. However, there are also an enormous number of uses for historical, archived sensor data for the analysis of long-term trends, for policy decision making, and for scientific discovery. There has been limited work on how best to archive and manage streaming, real-time data produced by a sensor network.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil
Copyright 2008 ACM 978-1-59593-753-7/08/0003 ...\$5.00.

In this paper, we present an approach for archiving data provenance and quality information produced by the workflow populating the data archive. Provenance and data quality issues have been previously studied in the context of data warehouses and grid systems. Our contribution is the use of domain-specific, generated metadata that allows for identification of data quality issues beyond operational metadata produced by the workflow processes themselves. The generated metadata serves the dual purpose of allowing efficient retrieval of data of interest and validating the quality of the data itself.

We have implemented the approach in a system that manages the collection, archival, and historical and real-time querying of weather (precipitation) data collected by 150 Next Generation Radars (NEXRAD) in the United States. Due to the size of the data set and the complexity of the radar sensors, managing this data requires a real-time data and metadata processing system [6]. The data generated by the radars is subject to instrumental and operator error and is too large to be manually curated and validated. The data quality monitoring subsystem automatically monitors the data load pipeline and flags data that is discovered to be in error. Administrators can then, if possible, correct the flawed data or annotate it as potentially compromised. Wherever possible, the system performs automated data correction and annotates data files suspected to contain errors.

The contributions of this paper are as follows:

- A design for a real-time data quality monitoring system for archiving sensor data and its implementation for the NEXRAD archive system.
- Operational results that demonstrate the effectiveness of the system for detecting and correcting sensor data errors and assisting the administrator in managing the system.

The organization of this paper is as follows. In Section 2, we overview the NEXRAD archive system and the importance of the data it archives. The overall system architecture is provided in Section 3. The data quality monitoring approach and its implementation are covered in Section 4. Operational data in Section 5 shows even for sophisticated sensors the number of data errors is not insignificant and such errors may propagate into the archive unnoticed without a monitoring system. The monitoring system reduced the administration cost and improved data quality after its installation. Finally, the paper closes with related work, conclusions, and future work.

2. NEXRAD BACKGROUND

There are over 150 NEXt generation RADars (NEXRAD) that collect real-time precipitation data across the United States. The radars generate terabytes of data per year. The real-time data produced is primarily used for severe weather forecasting. Historical, archived data has been used for flood prediction, bird and insect migration, and rainfall estimation. The National Research Council (NRC) has labeled the data as a “critical resource”.

The current archive system managed by the National Climatic Data Center (NCDC) maintains a tape archive of the raw data but does not provide a comprehensive and convenient online system to query and process data for research. Currently researchers must manually browse a catalog of the data and request copies of relevant data be pulled from tape by NCDC. This is unsatisfactory. The University Corporation for Atmospheric Research (UCAR) distributes the real-time data over the Internet, but users must archive, extract, filter, and generate tools for processing themselves.

A radar generates data by scanning its coverage area at various inclination angles to produce a three dimensional volume of data. A volume takes about four to six minutes to produce and occupies between 500 KB to 2 MB (compressed) and 5 MB to 20 MB (uncompressed). Thus, each radar produces approximately 300 MB/day for a total of about 45 GB/day for all radars. The data is sent over the Internet via the Internet Data Distribution (IDD) system managed by UCAR where it is accessible to researchers and forecasters. A primary backup is performed on-site on tape and sent to the central archive at NCDC.

Until the construction of the NEXRAD archive system [6], there was no online resource for researchers to manipulate the NEXRAD data. Researchers either had to build their own system for processing the raw data stream distributed by the IDD system or request data from the NCDC tape archive using a limited query facility. The NEXRAD archive system allows real-time and historical access to data generated by the radars and contains tools for the scientific analysis of the data set (such as for rainfall estimation). Thus, the data is available to many more researchers at lower cost. The archive is funded by NSF as a collaboration between the University of Iowa, Princeton University, UCAR, and NCDC.

A unique feature of the system is that the raw data is analyzed while converted and compressed for storage. This analysis produces metadata describing the data. The metadata includes the radar name and scan time and computed metadata statistics such as *areal coverage* (measure of reflectivity in a scan). Metadata statistics help researchers find relevant data. For instance researchers can request data that shows significant meteorological activity (which is detected by the areal coverage metadata) instead of browsing by radar and date. Since this is a research system the data load components and metadata generator are highly customizable to support the generation and storage of new metadata statistics as they are developed.

3. ARCHIVE ARCHITECTURE

The NEXRAD archive system receives real-time radar scan data from each radar, generates metadata that describes the incoming radar data, and stores the metadata and radar data in separate archives. The new monitoring

system logs events and exceptions in a separate monitor database for review by administrators.

The architecture of the NEXRAD archive system is separated into various modules (see Figure 1). The IDD reader module decodes the real-time radar data stream into separate raw data files for further processing. Each data file contains one volume scan. The RAW-to-RLE converter converts the raw, radar format into a more efficient compressed (R-run L-length E-encoding) format. The RAW and RLE files proceed through the pipeline to the metadata extractor that analyzes each volume scan and generates descriptive metadata used for indexing and retrieval. The loader module copies (and potentially replicates) the RAW and RLE data files to Internet-accessible storage servers. The loader also updates a database storing the metadata information. The metadata server provides a query facility allowing researchers to find the data of interest and download from the data storage servers. Users of the system (or user applications) submit a query to the metadata archive module by specifying certain metadata properties that the resulting data must conform to. The metadata archive returns a list of files and their locations in the data archives. The user (or user app) then retrieves the files from the data archive as required.

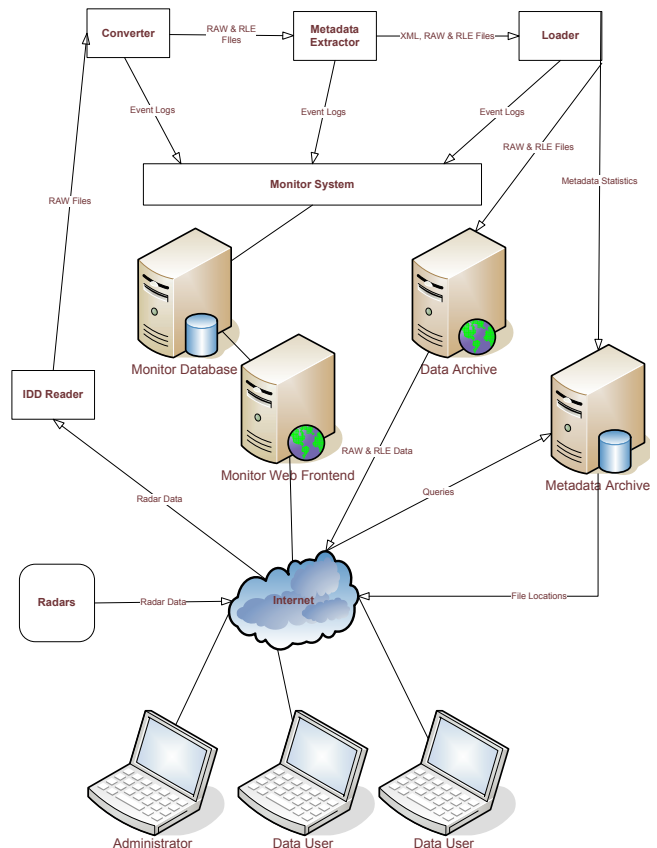


Figure 1: NEXRAD Archive System Architecture

The archive system is a pipelined Extraction, Transform, Load (ETL) process. Each module functions independently on input files and communicates with each other using XML. This allows for high-performance and fault tolerance as the processes can be duplicated and distributed across machines.

Operational data identified the requirement for a more advanced monitoring subsystem to improve data quality and aid administration.

4. ARCHIVE MONITORING SYSTEM

The archive monitoring system is designed to monitor the data flow of real-time data through the ETL pipeline process, detect and resolve ETL process errors, and detect and potentially correct errors in the underlying data. In this section, we describe the generic approach and its application to the NEXRAD archive system.

4.1 General Architecture

The monitoring system architecture consists of four components:

- Standardized logging in ETL processes
- A database-backed log processor
- Metadata statistics for indexing and data validation
- A web-based administration portal

The first component of the architecture is ETL modules that use standardized logging methods to capture important events in their processing. Most ETL systems perform some form of logging, but the logging is usually to flat text files. With each module having its own logging format and file, it is difficult to analyze log files, especially for trends across modules. Each ETL module has its logging system changed to write out standard XML log records where each log record contains a timestamp, the module identifier, and the error code and message. These log record files are then consumed by a log processor that loads the logs into a database system. In this way, all log data from all modules is in the database and provenance and data quality can be tracked across the entire workflow. The web-based administration portal provides a way for monitoring ETL module status, searching and browsing the logs (from the database), and manually correcting data errors.

Beyond basic file metadata, the system computes derived metadata whose primary use is for indexing in order to help find data files of interest. By supplying expected ranges of the computed metadata, the monitoring architecture can use metadata to validate the quality of the data archived. If computed metadata is outside expected ranges, then the data is annotated as potentially compromised. Some errors may be automatically corrected (such as a bad file timestamp), others can only be flagged in the archive (such as a file that has an areal coverage outside of the acceptable range). Thus, computed metadata provides both an indexing and validation method of the received sensor data.

For example, areal coverage is a percentage between 0 and 100 representing the amount of the volume where reflectivity (most likely precipitation) is detected. If the computation of areal coverage during metadata generation results in a value outside of this range (such as a negative number), then the data likely has some form of corruption and is flagged as compromised in the archive. Date and time fields such as scan times can be auto-corrected based on the current time of the system. Finally, more complicated metadata such as masks to filter out spurious data (but still correctly captured by the sensor) is computed, stored, and validated as well.

For instance, a sensor may pick up reflected information from terrain objects (referred to as ground clutter) that is not valid precipitation data.

4.2 Monitoring System Implementation

The monitoring architecture was added to the NEXRAD archive system by updating the ETL modules to export XML-based log records to a log buffer. The amount of code changes was less than 100 lines per module. The log processor polls the buffer for log record files. Once a file is found, the XML is parsed, and the log record information is stored in a Postgres database. If the log processor crashes the log events that have not yet been loaded into the database are available in the buffer for loading after recovery. The logging system is a push-based approach that does not interfere with the normal ETL workflow. For performance and security reasons, the log database is stored at a separate machine from the data/metadata archive. This allows failures in the logging system itself to not prevent progress in the data archiving. The types of events logged to the monitoring system can be tuned so that when an administrator notices odd events, such as many failures in a short period of time, they can enable a more thorough event logging of the operation of each module.

There is a web-based front-end for administrators (see Figure 2) that shows, at a glance, the status of the core modules as well as a list of recent error and warning events. The administrator can control the modules using this interface. Another part of the front-end allows the administrator to view all recent events or a filtered list of all events. The front-end also displays various statistics and graphs of events logged in the monitor database and provides information regarding data coming from each radar. The system provides an overview of the modules, errors/warnings per module, critical event notifications, and reports on the archival system performance and number of files archived. The provenance of each source file as it passes through the ETL workflow to the archive is easily accessible.

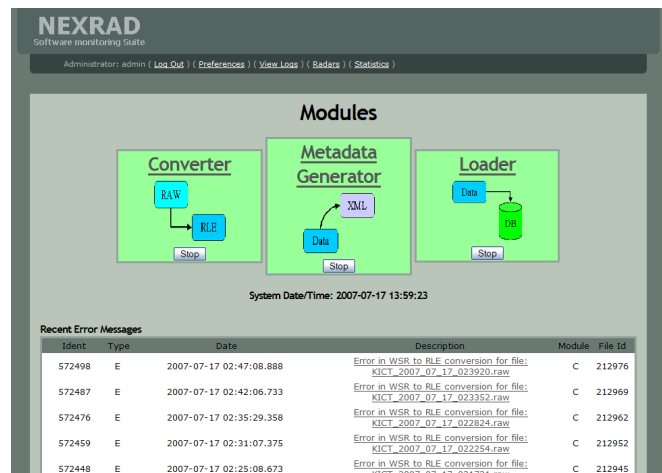


Figure 2: Monitoring System Module Status View

A watchdog process watches the operation of the other modules and helps them recover from common errors and return to operation if they fail. The watchdog operates essentially as an automated administrator and performs restart of

failed processes, handling of corrupt data files, and limited error correction. The watchdog logs its activities.

At every stage of the data load pipeline the potential for errors exist. RAW data entering the system from the radars may be compromised due to equipment and operational failures. This data is provided “as-is” by the IDD system. The level of corruption ranges from small data errors to critical errors rendering the data unusable. The converter is subjected to these errors as the first stage of processing which may result in bad conversion (zero-length or corrupted files) or failure of the conversion software itself (critical error). Bad data passed on to the metadata extractor may cause the metadata generation code to fail or generate incorrect results. The loader generates errors if unable to load to remote data stores and databases. All components may generate errors if data is unable to be passed to other stages of the pipeline. Generated metadata is validated against expected ranges to detect data corruption. In Figure 3 is a table summarizing the errors detected and corrected by the monitoring system.

5. OPERATIONAL RESULTS

The NEXRAD archive system has been running for over two years and has collected approximately 6 TB of data for 30 radars. The data archive system is relatively stable and the administrator has developed automated scripts to restart failed processes (such as the converter) when drastic errors occur in the pipeline. Before the monitoring system was installed, there was no measurement on data quality or a perspective on how much data was lost due to errors in conversion and loading into the archive.

The NEXRAD monitoring system has been processing data for a subset of the radars since May 2007. During this time, over 42,000 volume files were processed. The converter module discovered 1636 (almost 4%) of those files to be in error and corrupted. A further 375 files (less than 1%) generated errors in later modules. These errors were logged to the monitoring system and the files were removed from the pipeline at the point of error and buffered for later analysis by the administrator. Analyzing the error data reveals that error files tend to be clustered in time. This is to be expected as operational or system errors may affect multiple scans during a block of time before the issue is resolved. While an error rate of less than 5% is relatively small, these errors would have gone completely unnoticed with the original system and in some cases would have been archived and made available to researchers as if it was valid data.

The automated watchdog restarted a ETL module 11 times. Failures occur due to code faults, network errors, and file system errors.

Using the web interface, the administrator spends less than 5 minutes per day in monitoring the system compared to approximately 30 minutes per day searching individual logs. This results in a significant 80% reduction in administration time and a greatly increased confidence in the quality and operation of the archive.

The overhead on the ETL pipeline modules to log XML records instead of to flat files is minimal. Storing log records in a database separate from the operational system results in no performance penalty to the ETL workflow and normal archive operations.

Overall, the quality of the archive is improved by the automated flagging of files in error both due to metadata and

conversion problems. These files are not lost from the system as before, but are also not treated as regular data. The monitoring system has minimal overhead on the operational system while providing a cleaner, more efficient interface for the administrator to monitor and resolve issues with the ETL pipeline.

6. PREVIOUS WORK

The NEXRAD archive system is similar to a real-time data warehouse. Data warehouse Extraction, Transform, Load (ETL) tools are available from various vendors. However, the focus of these tools is on extracting and cleaning data from other systems not from scientific sensors. Further, there is an implicit assumption that the data is available for extraction at all times. Processing a real-time stream (such as that produced by radars and sensor networks) means that if the incoming information is not handled and loaded into the data warehouse in a given amount of time, the data will be lost and unrecoverable. If errors occur, the ETL system must continue to archive data and not discard data in error until administrators have intervened. It is especially important to have an active monitoring system in this domain. Research on ETL pipelined processes [1] has not considered the archiving of streaming data.

Jeffery et al. [5] describe a system that processes and cleans real-time sensor data based on temporal and spatial information before passing the data on to applications. There is no mention of archiving the sensor data streams, monitoring the operation of the cleaning system, or logging data error statistics. Since the NEXRAD radars have limited overlap and correspond to areas of varying weather, spatial cleaning can not easily be performed on this data. Our metadata statistics allow for the detection of data errors but not cleaning. Unlike provenance-based curation [2], the data produced by the system is too large to be human-analyzed and manual correction is not always possible.

Monitoring and workflow systems such as Chimera [3] and Earth System Science Workbench [4] are used to track operational performance and provenance in grids. These systems generate operational metadata across workflow processes, but do not use derived metadata to detect data quality issues.

A related project is Linked Environments for Atmospheric Discovery (LEAD) [7] with the grand vision of a complete, grid-based scientific environment for severe weather forecasting. This system also uses provenance collection on scientific workflows [8]. As the primary focus is real-time forecasting, less emphasis has been placed on archiving the data for historical purposes and the use of metadata for data retrieval and for data quality purposes.

7. CONCLUSIONS

Archiving data streams will become increasingly important as historical data collected by sensor networks is used for long-term analysis. There has been limited work on archiving data streams specifically regarding the issue of data quality. We have presented an approach for logging the operation of an ETL pipeline processing a data stream. The logging system is database-based allowing for efficient operation and analysis. A distinctive feature of the work is the use of computed metadata on the raw data stream to validate the raw data quality and to detect (and sometimes

Module	Error Name	Error Description
Converter	RAW file corruption	RAW files corrupted due to incorrect radar operation or errors during transit are flagged so that they do not cause other modules to generate errors. They are discarded or recovered by an administrator.
	Error during conversion	An error occurred while converting/compressing the raw radar data files. This could be due to file corruption not yet detected or due to the incorrect operation of the converter module. These files are handled like corrupt files as above.
	Error sending files to Generator	An error occurred while trying to move the original RAW and generated RLE files to the metadata generator. This could be due to file system or network errors.
Generator	Error sending files to data archive	An error occurred while trying to move the RAW and RLE files to the data archive.
	Error sending XML file to Loader	An error occurred while trying to move the generated XML metadata file to the metadata loader.
	Error parsing RLE file	An error occurred while trying to parse metadata from the RLE file.
Loader	Error parsing XML metadata file	An error occurred while trying to parse the XML metadata file.
	Error connecting to database	An error occurred while trying to connect to the database.
	Error loading into database	An error occurred while trying to load metadata from the XML file into the database. This could be due to invalid metadata that does not conform to database rules.
	Metadata Error	Metadata generated outside of expected range results in data file flagged with a warning.
	Date Error	Auto-correct date based on current date.
	Metadata property not in use	The XML metadata could contain a property that is no longer in use or whose name was entered incorrectly by the metadata generator.

Figure 3: Detected and Corrected Errors for NEXRAD Monitoring System

repair) data corruption and errors before archival. The approach has been implemented on the NEXRAD archive system that processes a large data stream from complicated weather radar sensors. Even with sophisticated sensors, data errors consist of about 5% of the data, and the automated techniques for detecting the errors greatly improve the quality of the archive and the efficiency of its administration.

Future work includes determining if temporal information (multiple scans over time by the same radar) can be used to detect and repair collected data.

8. ACKNOWLEDGMENTS

The NEXRAD project is supported under NSF Medium ITR Grant 0427422-002. We acknowledge our collaborators at the University of Iowa for providing the data sets and system access required to conduct this research. Dr. Lawrence is also supported by an NSERC discovery grant and University of British Columbia Okanagan internal funding.

9. REFERENCES

- [1] R. M. Bruckner, B. List, and J. Schiefer. Striving towards Near Real-Time Data Integration for Data Warehouses. In *4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, pages 317–326, 2002.
- [2] P. Buneman, A. P. Chapman, and J. Cheney. Provenance Management in Curated Databases. In *ACM SIGMOD International Conference on Management of Data*, pages 539–550, 2006.
- [3] I. T. Foster, J.-S. Vöckler, M. Wilde, and Y. Zhao. Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation. In *Proceedings of the 14th International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 37–46, 2002.
- [4] J. Frew and R. Bose. Earth System Science Workbench: A Data Management Infrastructure for Earth Science Products. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 180–189, 2001.
- [5] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom. Declarative Support for Sensor Data Cleaning. In *4th International Conference on Pervasive Computing*, pages 83–100, 2006.
- [6] A. Kruger, R. Lawrence, and E. Dragut. Building a Terabyte NEXRAD Radar Database for Hydrometeorology Research. *Computers and Geosciences*, 32(2):247–258, 2006.
- [7] B. Plale, D. Gannon, J. Brotzge, K. Droegemeier, J. Kurose, D. McLaughlin, R. Wilhelmson, S. Graves, M. Ramamurthy, R. D. Clark, S. Yalda, D. A. Reed, E. Joseph, and V. Chandrasekar. CASA and LEAD: Adaptive Cyberinfrastructure for Real-Time Multiscale Weather Forecasting. *Computer*, 39(11):56–64, 2006.
- [8] Y. L. Simmhan, B. Plale, and D. Gannon. A Framework for Collecting Provenance in Data-Centric Scientific Workflows. In *IEEE International Conference on Web Services (ICWS)*, pages 427–436, 2006.