An Architecture for Real-Time Warehousing of Scientific Data

Ramon Lawrence
Department of Computer Science
University of Iowa
Iowa City, IA, USA
ramon-lawrence@uiowa.edu

Anton Kruger
IIHR
University of Iowa
Iowa City, IA, USA
anton-kruger@uiowa.edu

Abstract

The management and processing of terabyte-scale data sets is time-consuming, costly, and an impediment to research. Researchers require rapid and transparent access to the data without being encumbered with the technical challenges of data management. The contribution of this work is an architecture for archiving and analyzing real-time scientific data that isolates researchers from the complexities of data storage and retrieval. Data access transparency is achieved by using a database to store metadata on the raw data, and retrieving data subsets of interest using SQL queries on metadata. The second component is a distributed web platform that transparently distributes data across web servers. We have successfully implemented the architecture on a data set produced by the United States National Weather Service's network of WSR-88D weather radars.

Keywords: real-time archiving, atmospheric sciences/hydrology, weather radars, metadata, NEXRAD

1. Introduction

The amount of scientific data generated is growing extremely rapidly and is challenging our ability to store and query it effectively. This situation will only get worse as large-scale sensor networks are deployed. The challenge is to build a data storage architecture capable of managing the large volumes of data that is easily accessible by the researchers who require the data. It is natural to apply data warehousing techniques [4] to this problem. However, there are several challenges that must be overcome. First, archiving the data without generating any descriptive characteristics is undesirable as it makes it difficult to retrieve the data for later analysis. Metadata must be generated on the data in order to categorize it and allow for retrieval based on its properties. Another problem is that the metadata changes much more frequently than in a typical order processing data warehouse, as scientists change the type and form of metadata as new discoveries are made. Finally, the system architecture must require minimal administration and cost-effectively scale with the data set size.

In this work, we present a general architecture for archiving and retrieving real-time, scientific data. The basis of the architecture is a data warehouse that stores metadata on the raw data to allow for its efficient retrieval. The warehouse allows the researcher to configure the type of metadata used, which may evolve over time. The contributions of this work are:

- A *metadata generation and retrieval system* that allows researcher-defined metadata statistics to be calculated, stored, and used to find data of interest.
- A *transparent data distribution system* that uses the data warehouse to dynamically distribute the data across multiple machines.

- An *architecture implementation archiving weather radar data* that is more usable than the system deployed by the National Climatic Data Center (NCDC), which is the United States government organization mandated to archive the weather radar data set.

The organization of this paper is as follows. In Section 2, we provide background on the challenges of archiving scientific data and illustrate with the example of archiving weather radar data. The system architecture is presented in Section 3 with details on the metadata and data distribution systems. An implementation of the architecture for the weather radar data is discussed in Section 4 along with the benefits realized. Related work is in Section 5, and the paper closes with future work and conclusions.

2. Background

Researchers do not want to invest precious time and resources in data management issues but are becoming increasingly reliant on database systems to manage their large data sets. Large data sets provide two distinct challenges for researchers. First, there is the administration challenge of how to handle the issues of data management, distribution, retrieval, and processing. Although the rapidly decreasing cost of storage has made compiling terabyte-scale data sets more practical, a far more challenging problem is extracting meaningful subsets of the data that are applicable to given research problems. We call the task of finding data relevant to a research question the *data finding* problem.

A particular data warehousing problem that we examine is the collection and storage of real-time data generated by the over 150 Doppler Weather Surveillance Radars operated by the National Weather Service in the United States as part of their NEXRAD system. Each radar generates about 1 TB of data per year. The NEXRAD system was designed and built primarily for severe weather surveillance, and initially little emphasis was placed on archiving data. It was soon realized that the data has tremendous scientific and commercial value [8] beyond surveillance and forecasting, and an archival program was put in place where the raw, unprocessed radar data is sent to the U.S. National Climatic Data Center (NCDC).

Researchers that require the data for their experiments are faced with complicated data management challenges. First, the researcher must *acquire* the data from NCDC that involves manually searching a catalog and requesting NCDC retrieve the data from their tape archive. Not only is there a significant delay in acquiring the data once the correct data set is determined, there is a major challenge in *finding* the data that is applicable to the given research problem. The NCDC maintains basic catalogs, which categorizes data only by date and radar location. Researchers often require more specific properties of their data sets, and browsing the data to find particular data points of interest is unacceptable. Although some real-time radar data is now distributed over the Internet by the University Corporation for Atmospheric Research (UCAR) using their Unidata Internet Data Distribution (IDD) system[1], researchers must build sophisticated systems for processing and filtering the data stream. Once researchers acquire the data, the next task is to *archive* the data. Given that the size of the data is in terabytes, this requires a major investment in hardware and administration. Finally, researchers must also have the ability to *query* and *process* the data. The size of the data makes this is a major issue, and in its raw form, the data is uncompressed and difficult to manipulate without libraries for reading and analysis. Without metadata, researchers must browse the entire data set to find subsets of the data with their required features. Thus, researchers would greatly benefit from a system for archiving radar data in real-time that allows for data retrieval using metadata properties.

[1] http://www.unidata.ucar.edu/projects/idd/overview/idd.html

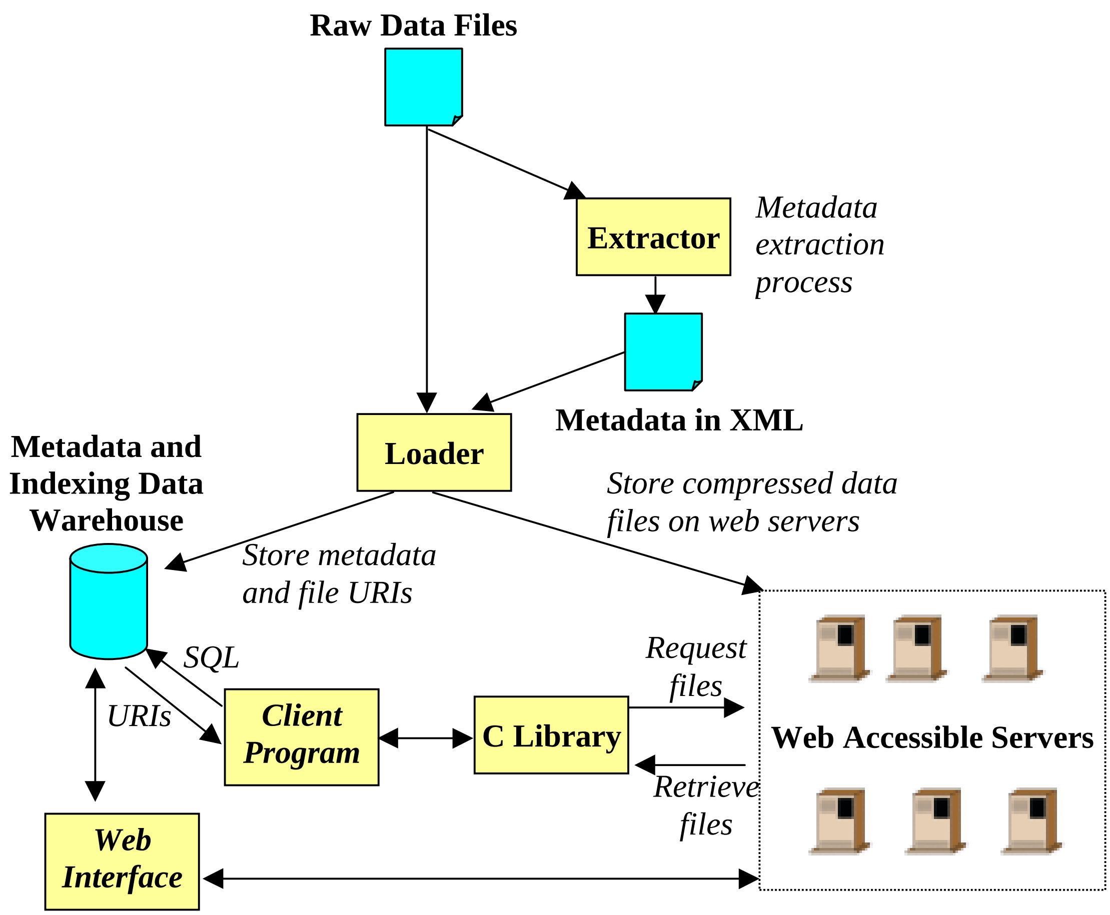

Figure 1. Data Archival Architecture

3. Architecture Overview

The foundation of the architecture (Figure 1) is a data warehouse that archives generated metadata and the physical locations of data files. The data warehouse supports evolving metadata by using a general schema capable of storing a varying number of metadata statistics per data file. The load process processes raw data files using a data set specific `Extractor` module to generate the metadata statistics that will be used for indexing the data file, and outputs them in an XML file. The `Loader` module parses the XML file and loads the metadata statistics into the data warehouse. The load process is general, as it must accept varying types of metadata. The `Loader` also is responsible for compressing the data file and placing it on a retrieval server. The load process occurs in real-time. A *retrieval server* is a web server that archives a subset of the data files. Each file is accessible using its Uniform Resource Identifier (URI), which is stored in the data warehouse. A user retrieves files of interest by querying the data warehouse on the metadata, retrieving the list of URIs of matching data files, and then downloading the actual data files from the retrieval servers. User queries can be performed using a web interface or using a C interface library.

The architecture can be applied to any scientific domain and only requires the researcher write the `Extractor` module to parse the raw data and generate the desired metadata statistics. Our architecture defines a standard XML schema that the extraction code must generate to represent the metadata statistics in a form suitable for the `Loader` module to process. An example XML output for the weather data set is given in Figure 2. Properties must be present for all files, whereas files may have different statistics. The load process is largely independent of scientific domain but may have to be modified if specific domain data must be loaded into the warehouse and if a specific raw data compression algorithm is used. The load process must be configured with information on how to place the files on the web servers.

```xml
<job>
   <inFile>KDVN_2004_08_01_000157.raw</inFile>
   <URI>http://nexrad.xxx.edu/KDVN/2004/aug/KDVN_2004_08_01_000157.rle</URI>
   <property name="scanDate">2004-08-01 00:01:57</property>
   <property name="radarId">KDVN</property>
   <statistic name="MaximumRainRate">10.5</statistic>
   <statistic name="ArealCoverage" size="2">
      <param>0</param><value>25</value>
      <param>10</param><value>15</value>
  </statistic>
</job>
```

Figure 2. Sample XML File Generated by Extractor

This architecture achieves the four design goals of scalability, extensibility, cost-effectiveness, and usability. The architecture is *scalable* as new servers can be added to store data files as required, and the archive can efficiently process multiple terabytes of data. These servers require only a web server to provide access to the files. Data files can be transparently replicated, distributed, and re-organized as the data warehouse maintains an index of their locations. This allows for quick re-organization of files, addition and removal of servers, file sharing across organizations, and for data to be quickly added to the archive. The servers can run on different operating systems and hardware since all file communication is done using HTTP. The architecture is *extensible* as the metadata extraction process can be configured to calculate different types of metadata, which allows the architecture to be applied to different scientific domains. The system is also *inexpensive* as off-the-shelf hardware and open-source software can be used for storage including standard PCs. No additional software is required except for the web server, such as Apache (open source). Finally, the system is highly *usable* as the researcher can query on metadata and transparently retrieve data files from multiple servers. The researcher can query the database directly or use the C library or web interface. Applications can be built on top of the C library to perform browsing and advanced search functions to hide even the SQL access and C library calls. The system as a whole provides a single point of contact for clients. It vastly simplifies authentication of clients and reduces security concerns of administrators. The system is easily usable by researchers without extensive computer knowledge and allows them to perform research that may not otherwise be possible.

4. Architecture Implementation for Archiving Weather Radar Data

We have implemented the general architecture for the specific task of archiving weather radar data collected by the NEXRAD system. As discussed, the current state of weather radar data archival is insufficient for researchers who want real-time access to the data without having to manually browse a large, offline repository at NCDC. In our implementation, we process the real-time IDD radar feeds distributed over the Internet by UCAR, which provide the data in raw, uncompressed form. A radar generates a raw data file about every 5 minutes, which is extracted in real-time from the IDD feed. The `Extractor` module is C code that converts the raw data into a compressed form using run-length-encoding [9] and generates metadata statistics. Some of the metadata statistics for this domain include *areal coverage* (the percentage of pixels above multiple reflectivity thresholds), *mean areal precipitation*, and *maximal* and *minimal rain rates*. The extraction code generates the metadata statistics and creates an XML file conforming to the XML schema for the `Loader` to use. The `Loader` is a Java program that parses the XML document using SAX and stores the metadata in the warehouse. Note that the `Loader` stores both the name of the metadata statistic and its computed value. This program also places the data files on the web server and stores the URIs. The file placement strategy for this domain is hierarchical by radar, year, and month. An example URI is:

`http://nexrad.cs.uiowa.edu/data/KDVN/2004/jul/KDVN_2004_07_16_151029.rle`

This URI indicates the server where the file resides "`nexrad.cs.uiowa.edu`" and the file path "`data/KDVN/2004/jul/KDVN_2004_07_16_151029.rle`". The directory structure is organized by radar site ("`KDVN`"), by year ("`2004`"), by month ("`jul`"), and by file name.

Users may retrieve data files by querying on metadata. An example query that researchers can now execute that they could not before is:

```
"Find all the volumes (files) for the KDVN radar in the last 5 years
with a mean areal precipitation > XX and maximum rain rate < YY mm/h."
```

Researchers and students at our institution are currently using the warehouse. The system is deployed on a dual PowerG5 with an attached 1.5 TB RAID. The system is currently archiving 30 radars which each generate a data file every 5 minutes (~105,000 files/day). The average daily archive rate is 3 GB/day (compressed), which requires processing approximately a 15 GB/day uncompressed data stream. The system is also capable of adding historical data from NCDC into the archive. Even with a large data set, the overhead to query the warehouse and retrieve the data files is minimal. Our organization has already seen the benefits of reduced administration costs and researcher productivity. We are currently working with NCDC on an NSF funded project to deploy the architecture at NCDC, so that the benefits of the data archive architecture may be available throughout the United States.

5. Related Work

Related work includes the work on metadata catalogs in Data Grid systems [7]. A Data Grid is intended for the distribution and analysis of large-scale scientific data sets. The metadata catalog is used to retrieve files on the grid with certain properties. Prototype implementations of metadata catalogs include the metadata catalog service [11], Chimera [6], DataCutter [3], and MCAT Metadata Catalog [2]. The general idea of these services is that logical data files have associated metadata attributes that can be used to determine their content and relevance. Users query the metadata catalog service to find the desired files, and the physical files may be retrieved from one or more servers on the grid. The Chimera system is specifically focused on storing metadata on generated or derived data products and their provenance.

A metadata catalog system has been implemented for the GriPhyn [1] and Earth System Grid-II [5]. Thus, our work has similarities to these proposed approaches to scientific data management. We have demonstrated an effective approach based on data warehousing and applied it to weather radar data. Our work is unique as we also focus on how the metadata can be efficiently *generated* as well as being stored and queried. An advantage of our approach is that it does not require integration with complicated grid technology.

Another related area of research is maintaining data warehouses that evolve [10]. Our data warehouse supports evolution of metadata and varying types of metadata per data file. This is slightly different than evolution of fixed schemas with the same attributes for each object.

6. Future Work and Conclusions

We have presented a scaleable architecture for managing large, real-time, scientific data sets. The foundation of the architecture is a real-time data warehouse that archives metadata for querying and provides data distribution transparency. The metadata allows efficient retrieval of relevant data by properties. By distributing files over web servers and retrieving them using HTTP URIs, no proprietary network protocols are used and issues with server heterogeneity are removed. Further, the system is extensible as the files can be distributed across multiple servers (which may not even all belong to the same organization). By isolating researchers from the burdens of data handling, researchers can spend more time on developing experiments and using the large data set to its full potential. We demonstrated the feasibility of the architecture by implementing it to archive weather radar data generated by the NEXRAD system. The valuable data is archived and used by researchers and students in ways not possible under the current archival system at NCDC. The automatic system for archiving and analyzing NEXRAD data isolates geoscientistics from the complexities of data storage and retrieval.

Future work involves collaborating with NCDC to deploy the architecture at multiple institutions and at NCDC. We are developing a method to allow research groups to share their data archives, so that organizations can cooperate and share the burden of archiving the data.

7. References

[1] P. Avery and I. Foster. The GriPhyN Project: Towards petascale Virtual Data Grids. Technical Report GriPhyN-2001-15, 2001, www.griphyn.org.

[2] C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC Storage Resource Broker. In *Proceedings of CASCON'98*, 1998.

[3] M. Beynon, T. Kurç, Ü. Çatalyürek, C. Chang, A. Sussman, and J. Saltz. Distributed processing of very large datasets with DataCutter. *Parallel Computing* 27(11): 1457-1478, 2001.

[4] S.Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1):65-74, 1997

[5] A. Chervenak, E. Deelman, C. Kesselman, B. Allcock, I. Foster, V. Nefedova, J. Lee, A. Sim, A. Shoshani, B. Drach, D. Williams, and D. Middleton. High-performance Remote Access to Climate Simulation Data: A Challenge Problem for Data Grid Technologies. *Parallel Computing* 29(10): 1335-1356, 2003.

[6] I. Foster, J. Vöckler, M. Wilde, and Y. Zhao. Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation. *SSDBM* 2002: 37-46.

[7] I. Foster, J. Vöckler, M. Wilde, and Y. Zhao. The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration. *CIDR 2003*.

[8] NRC 1999, Enhancing Access to NEXRAD Data—A Critical National Resource. *National Academy Press*, Washington D.C. ISBN 0-309-06636-0, 1999.

[9] A. Kruger and W.F. Krajewski. Efficient Storage of Weather Radar Data. *Software Practice and Experience*, **27**(6), 623–635, 1997.

[10] E. Rundensteiner, A. Koeller, and X. Zhang. Maintaining Data Warehouses over Changing Information Sources. *Communications of the ACM* 43(6): 57-62, 2000.

[11] G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman. A Metadata Catalog Service for Data Intensive Applications. *SC2003*.