J9.17 IMPLEMENTATION OF AN ONLINE DATABASE AT THE SURFACE REFERENCE DATA CENTER

Michael D. Klatt *, Mark. L. Morrissey, and J. Scott Greene
University of Oklahoma, Norman, Oklahoma

1. INTRODUCTION

The Surface Reference Data Center (SRDC) at the Environmental Verification and Analysis Center is tasked with producing gridded rain gauge estimates for verification purposes in support of the Global Precipitation Climatology Project. Integral to the SRDC mission is the distribution of data and derived products to GPCP participants and the research community in general. Easy access to all available data is also essential to internal users. The existing database is no longer adequate and is being replaced with a completely new system.

Experience with the current data handling scheme at SRDC provides much of the inspiration for the new system. The main design goals for the new database are accessibility and maintainability. Regardless of how valuable data in a database is, if it is difficult to access it will not be utilized. It is vital for the user to be able to get the data he wants in the format he wants it in. Equally important is that the database is easy to maintain. Data ingest and update procedures should be streamlined and robust. The system should be designed with flexibility in mind so that it is straightforward to integrate new data sets. Maximum effort should be expended during design and development to create a system that takes minimal effort to keep going. The person who developed the database should not be the only one who can maintain it.

2. DATABASE COMPONENTS

2.1 Server

A dedicated server will be used to run the new database and host the SRDC web site. The server is a Gateway 6400 with a 933 MHz Pentium III processor, 256 MB of SDRAM, and an 18 GB hard drive. All of these are expandable as needed. The operating system chosen for this server is Red Hat Linux. Linux provides a superior combination of performance, flexibility, and security when compared to other operating systems. Although cost was not a primary consideration, the open-source nature of Linux went a long way towards its selection.

* *Corresponding author address:* Michael D. Klatt, EVAC, University of Oklahoma, 710 Asp Avenue, Suite 8, Norman, OK 73069; e-mail: mdklatt@ou.edu

2.2 Database Management System

PostgreSQL has been chosen as the database management system (DBMS) software. PostgreSQL is an object-relational DBMS based on SQL. It has many advanced database features, such as concurrency control and triggers. PostgreSQL supports API's for many different programming languages (such as C++, Perl, and Java) and protocols (ODBC, JDBC). This is crucial to developing stand-alone applications which use the database. A particularly attractive feature is the capability for user-defined types and functions. The ability to integrate complex data handling into the database itself (such as validation or transformations) greatly simplifies the development of client applications. Object-oriented support includes inheritance and some polymorphism. This is very useful in conjunction with the C++ or Java API to create a cohesive object-oriented system. Like Linux, PostgreSQL is open-source. In fact, it is part of the latest Red Hat distribution, which simplifies installation and configuration.

2.3 User Interface

The most important component of the new database from the user's point of view is the interface. Making data available via the World Wide Web is an obvious choice, but there is an ever increasing number of ways to do this. A combination of CGI and HTML forms has been chosen over other options, such as Java applets or servlets. Unlike newer technologies, CGI/HTML is extremely portable and well-supported by nearly every browser and web server. A user-friendly and robust interface can be built using only HTML forms. CGI applications can be developed using a wide variety of programming languages. In an environment where both C++ and Fortran are used extensively this is important. The learning curve for CGI applications is not as steep as other technologies, an important factor in deploying the new database in a timely fashion. One drawback of a CGI application is that it cannot interact with the user during execution. This makes features like user input validation more difficult. Another potential disadvantage is that CGI applications are server-side rather than client-side. This is good for the user, but may hinder performance if there are a large number of simultaneous users. However, based on the current usage statistics this should not be a problem in the near future.

3. IMPLEMENTATION

3.1 Languages

The new database is still in its infancy so many implementation details have not been determined yet. One thing that has been decided is that the main development language will be C++, which is a useful general-purpose language that supports object-oriented programming. C++ classes to support CGI functionality and the PostgreSQL API have already been completed.

Fortran 95 will be used for numerically intensive applications, for which it is superior to C++. For example, applications which perform objective analysis on the data will be written in Fortran. However, any such calculations which are performed frequently across different applications should ideally be integrated into the DBMS as user-defined functions. One drawback to using Fortran is that there is no corresponding PostgreSQL API. It will be necessary create a Fortran interface to database access functions compiled in another language. The concept of a Fortran 95 module is ideal for this purpose.

3.2 Data Ingest

The diverse array of data sets used by SRDC makes data ingest a challenge. A C++ class hierarchy is being developed to represent each data source as a separate object. Incorporating a new data set is a straightforward matter of deriving a new class. The ingest application will read the original files and send the data to the database as text. Data validation must be performed during ingest, but there are several options for this. Validation can be performed by the ingest application. PostgreSQL also supports validation through constraints and triggers. Experimentation will be necessary to determine the ideal approach.

3.3 Data Analysis

The generation of SRDC products and analysis will be automated wherever possible. The applications that do this will connect directly to the database. Some advanced analysis may be necessary as part of the user interface, for example allowing the user to download gridded data. This analysis could be integrated into the CGI application that retrieves the data or into the database itself. Again, experimentation is necessary to determine the best way to do this.

3.4 Data Access

The primary method of retrieving data will be via a series of web pages which use CGI to call server-side applications to retrieve the data. The user will be able to select a custom data set by submitting selection criteria using HTML forms. Form options will be generated in real time to reflect what is actually in the database. For example, the form will have default dates that corresponds to the available date range for the parameter or product selected.

Access to certain data will need to be restricted. Registered users will be maintained in a separate database which keeps track of any special access privileges. In the future it will be possible to store a user profile with various preferences and defaults that can be used to simplify a data request. User identification will be maintained throughout a session using HTTP cookies.

4. SUMMARY

SRDC has outgrown the capabilities of its existing database. A new database system is being developed with the two primary goals of accessibility and maintainability in mind. The database is being built around the PostgreSQL DBMS and a Web-based CGI/HTML interface. While much of the preliminary design work has been completed, many of the implementation details have to be worked out. The C++ and Fortran 95 languages will be used for application development. Work is proceeding on the various applications that are needed for data ingest, analysis, and access. For more information, visit the SRDC web site at **http://www.evac.ou.edu/srdc**.