A DATA WAREHOUSE APPLICATION UNDER UNIX

Cynthia A. Stetz
Merrill Lynch

Abstract

Any data warehousing application requires managing the flow of data into and out of the data warehouse. This paper will discuss how we manage the flow of data from multiple sources to a SUN workstation running UNIX, and what we do with it once we get it there.

Also discussed will be the methods used to increase the value of the data by creating summarized and denormalized views, the distribution of specialized data subsets for decision support, and an archival system designed to maintain the vitality of the warehouse by offloading aged data.

Finally, future plans for enhanced exploitation and expansion of the data warehouse will be presented.

History

The Quality Assurance Reporting & Systems Group supports the data requirements of business analysts both within the QA Group and outside the group in other Merrill Lynch departments. The thrust in data support is towards allowing direct data access by the analysts, without the requirement of programmer intervention for every data request. But, fulfilling these requests involves accessing seventeen data sources under ten Data Base Management Systems (DBMS), which are spread across seven hardware platforms. This data environment makes direct access by anyone other than a programmer impossible. Therefore, the decision was made to capture and consolidate the data in one location.

It was determined that to keep one year's worth of data in a DB2 database on an MVS mainframe would cost three quarter's of a million dollars. This expense did not fit into our budget. An alternative was to bring the data down to a server attached to our NOVELL network. It was decided that the group would acquire a SUN™ SPARCstation 10 and use it as the data repository. It was further decided to utilize the SAS system for data storage and retrieval, report production, and interactive end-user support, giving us a fully integrated environment in which to work.

Once we had this project well under way, we realized that what we were building was a data warehouse! At that time, a data warehouse management plan was developed.

Data Flow

To manage the data warehouse, I have adopted the five flow model as presented by Dr. Richard Hackathorn in his article "Data Warehousing Energizes Your Enterprise".

The five flows are defined as follows:

1. Inflow - Feeding the legacy data into the warehouse
2. Upflow - Aggregating and summarizing the highly detailed data
3. Outflow - Packaged and adhoc queries against both the highly summarized data and the detail data
4. Downflow - Sending older data off to an archive
5. Metaflow - Data about the data; what & where it is, and how to access it

The Quality Assurance Data Warehouse encompasses data used to support a wide variety of projects. For purposes of this paper, I will focus on the Tracking Operational Performance System, or TOPS. The following paragraphs describe how this portion of the warehouse is populated and managed. Other portions may have different data sources and output products, but are managed in the same manner.

Inflow:

The primary data source of the TOPS data is a DB2 database that resides on a
mainframe. The operating system is MVS/TSO/ISPF. The data is outloaded on a weekly basis using an in-house supplied access program. This program allows us to limit the outload to only the most current weekly data. We can not use SAS to perform the outload because it is not available on that CPU.

There are five weekly files, three of which contain detail data and two which are lightly summarized, for a total of approximately 100,000,000 bytes of data. There are also seven control files that are updated and outloaded on a monthly basis. These files control how the data is cleaned, screened and manipulated on input to the DB2 database, as well as how it is sliced and diced and presented on output.

All the files are downloaded using The EXTRA! for Windows File Transfer facility, with the binary format option to maintain the integrity of the many packed fields. The destination for the flat files is a Novell network directory that is accessible from the UNIX side of the system.

This data is then loaded into permanent SAS datasets in a directory on the UNIX SUN™ SPARCstation 10, using SAS informats to convert the data from EBCDIC to ASCII. Data verification and exception reporting takes place during this step, prior to the weekly data being released for general use.

Currently, this data access is a manual process, consisting of submitting a JCL job, checking that it ran successfully, starting the download, checking it periodically to see if it is complete, and then submitting the batch jobs on the UNIX side to update the database and produce weekly reports. Work is close to completion on a system that will fully automate this process.

Outflow:

This is defined as the process that adds value to the informational data contained in the warehouse, to make it more useful for decision support. It includes summarizing, denormalizing, and distributing the warehouse data.

The denormalizing takes the form of expanding the summarized data to include multiple control variables, which allows it to be viewed from different angles, and at various levels, without requiring any further table joins. An example of this is the volume summary table. It is created on a fiscal month basis, at a moderately summarized level. From here, subset, sort and sum routines built into a SAS/AF FRAME application allow analysts to view all, or any portion of, the data at any higher level of summarization.

For other customers, we supply several comma delimited files. These files can easily be imported into a variety of applications, and have eliminated the necessity of re-keying data. This has in turn reduced the error rate in the data and increased confidence in the results derived from the data.

Another data delivery method we use is the SAS permanent dataset. A specific subset of monthly detail data is extracted from the data warehouse and written to a transport file using the CPORT procedure. Our network and SUN workstation are connected in such a way that this file can be written directly onto a network location. The PC/SAS user uses the CIMPORT procedure to load the file into his permanent SAS database. He can then use all the functionality of SAS to manipulate the data, independantly from the data warehouse.

One more way we satisfy our customers' requirements is via cc:Mail. We produce exhibits from the data in the warehouse using the REPORT procedure, writing the output directly to a file. This file is then brought into Microsoft Word, and the formatting is modified by applying a WORD macro. The resulting WORD document can be cc:Mailed to anyone who has the WORD filter, and therefore can be printed remotely.

Upflow:

In addition to supplying many customers with data in one form or another, we produce a variety of printed reports from the data in the warehouse. These reports are done on a weekly, monthly, and/or quarterly basis, and present both detail and summarized data. Since these are regularly scheduled production reports, they are run as background jobs on the UNIX SUN workstation. UNIX scripts, or executables, are used to invoke SAS source files. The SAS code uses PROC PRINTTO to
control the destination of the output. Thus, one SAS invocation can produce many separate reports. The reports are then automatically printed by the UNIX scripts, as many copies as are required.

In addition to the production reports, we have built a Graphical User Interface (GUI), using the SAS/AF Software and FRAME entry, which sits on top of our SAS data warehouse. This application, the Quality Assurance Departmental OnDemand Reporting System (QA-DORS) allows the analysts to produce the most frequently requested adhoc reports on their own. Multiple report formats and content options make it robust enough to handle the vast majority of requests for data. Removing these adhoc requests from our programmers’ workload has allowed them to be more productive.

**Downflow:**

The archival procedure, which keeps the warehouse from growing stale, is executed after the fiscal month has ended and been processed. At that time, we want to offload the previous year’s month to our optical jukebox archive location. Using a combination of UNIX shell commands and SAS code to determine just which datasets need to be moved, we have the system write the necessary SAS code.

The archive is run overnight, to minimize impact on the system. The log is checked in the morning for any errors that might have occurred. If the copy went smoothly, then a second overnight job is run, which deletes the data from the warehouse.

**Metaflow:**

The best data warehouse is only as good as the information that is available about it. In order to have a data warehouse that is fully utilized, you must supply the intended audience with the tools to understand and exploit its capabilities. One of these tools is the data dictionary, which describes the contents of the datasets in the warehouse, and how they are related.

Originally, the data dictionary for our warehouse was created as a Microsoft WORD document. While flexible, allowing any information about the data to be stored freeform, maintaining the document has proven to be a headache, and is therefore not done in as timely a manner as is desirable. In looking for an alternative, we turned to the SAS system.

Using a combination of the PROC DATASETS CONTENTS option on the tables in the warehouse, and PROC CONTENTS on the SASHELP library, we can create fairly extensive metadata about our warehouse. The PROC DATASETS with the CONTENTS option gives a list of all the datasets in the directory, with the number of entries and variables each one contains. Using the SASHELP.VCOLUMN dataset, we are able to produce a cross reference table that lists all the variables in the warehouse, and every dataset that contains them. These procedures have been incorporated into the system so that they are executed automatically when the data is updated. In this way, we always have up-to-date information about our data warehouse, without requiring manual intervention by the programmers that support it.

**Future Plans**

As mentioned previously, we are completing a system that will automate the DB2 access and download of the data. Once this is in place, it will open the door to our capturing some large data tables that it is impossible to download now. We plan to integrate this data into the warehouse, identifying opportunities to combine the new with the existing information to further enhance its business value.

We also want to explore the concept of data 'mining', where the data indicates when something interesting is happening, rather than relying exclusively on external queries.

There are plans to add additional features to QA-DORS, in the form of additional report and content options. Drill-down reports are being developed, which will allow an analyst to start with a highly summarized view of the data, and subsequently investigate various levels of detail.

Applications using frontend products such as Visual Basic and Power Builder have
been developed and deployed. These applications utilize the ODBC capability of the SAS system to access the SAS datasets residing on the UNIX workstation across platforms from the NOVELL network, making the physical location of the data transparent to the user.

Alternative data delivery methods are regularly investigated and evaluated. We will also consider requests for additional distributed data extracts, for customers who want to work in environments other than SAS.

And, in the area of increasing the value of the warehouse by making more information about it available, we are looking forward to will incorporating future developments in SAS Data Dictionary products into our metadata procedures.

Conclusions:

We have realized many advantages by moving from a mainframe based environment to a data warehouse under UNIX. First and foremost is the greatly increased and improved access to the data, for both the programmers and the business analysts. Data requests can be satisfied sooner, and that means reports are produced in a more timely manner.

And, by delivering the data in the form that the customers need it allows them to spend more time analyzing rather than rekeying and verifying.

By creating the QA-DORS system, which was only possible once we built the data warehouse, the analysts can do their own adhoc reporting. This has allowed programmer time to be spent more productively, enhancing the system, and getting done some of those projects there never seems to be enough time for.

Finally, by utilizing the SAS system for building and manipulating our data warehouse, we have a fully integrated system with a robustness that should serve us admirably, well into the future.

References


Trademarks

SAS, SAS/FSP, SAS/AF and Frame Entry are registered trademarks of the SAS Institute, Inc.

Sun is a trademark of Sun Microsystems, Inc.

UNIX is a registered trademark of UNIX System Laboratories, Inc.

SPARCstation is a trademark of SPARC International, Inc., licensed exclusively to SUN Microsystems, Inc.

Other brand and product names are registered trademarks of their respective companies.

Author Contact Information

Cynthia A. Stetz
Senior Programmer Analyst
Quality Assurance Group/QA Reporting & Systems
Merrill Lynch
265 Davidson Ave. 4th floor
Somerset, NJ. 08873
(908) 764-1024

Internet Address:
cynthia_stetz@ml.com