Paper 11

A SAS/AF® Application for Parallel Extraction, Transformation, and Scoring of a Very Large Database

Daniel W. Kohn, Ph.D., Torrent Systems Inc., Cambridge, MA

David L. Kuhn, Ph.D., Innovative Idea Design, Cave Creek, AZ

ABSTRACT

This paper describes a SAS/AF application to extract large volumes of data in parallel from a multiple-terabyte RDBMS and directly populate a parallel SAS® data mart. During population, the application allows the user to perform CPU-intensive data transformation/normalization operations in parallel. This application also allows models generated by Enterprise Miner™ software to be deployed in parallel to score the entire data mart, or subsets of it.

Parallel execution is achieved using Torrent Systems' Orchestrate™ application development and runtime environment, which allows the application to

- extract data in parallel from a parallel RDBMS
- load the results of SAS programs back into the database in parallel
- process parallel data streams with parallel instances of a SAS DATA or PROC step for much higher throughput
- store large data sets in parallel, providing faster access and eliminating storage restrictions
- stream data between SAS steps without having to write intermediate results to disk.

The performance benefits of executing SAS extracts and other processes in parallel are well documented. In both production and test environments, parallel processing has allowed SAS applications to process larger workloads more quickly—typically improving performance by a factor equal to the number of processors used. These applications typically show near linear scalability. (An example of linear scalability is where a12-processor system provides 12 times the performance of a single processor.) These results are documented in the IBM Whitepaper, “Achieving Scalable Performance for Large SAS Applications and Database Extracts.”[i]

INTRODUCTION

In today's Fortune 1000 IT shops, already enormous data volumes are growing at a staggering pace. IT shops spanning industries from banking and telecommunications to retail and airlines rely on parallel relational database management tools such as DB2 UDB®, Informix®, Teradata®, and Oracle® to manage the ever-increasing volumes of atomic-level transaction data.

In order to segment these enormous warehouses of information into smaller sets, many IT shops are building and regularly updating data marts. Front-end users query these data marts using a variety of tools. One of the most common front-end users is the SAS analyst who is building analytic models to describe customer behavior and perform segmentation. The SAS analyst may be running standard SQL queries, more complex statistics like logistic regressions, or Enterprise Miner-generated algorithms against these data marts.

To support this analytical expertise, many IT shops are designing and building SAS data marts for SAS analysts. These data marts have been constrained by their enormous volumes of data, the short windows typically available for populating these marts, and the time required to model and score all of the data. Because of these constraints, IT shops are looking to increase application performance scalable frameworks that leverage the parallel processing capabilities of multiprocessor computers (SMPs, MPPs, and clusters).

The SAS/AF application described in this paper was designed to address the constraints described above and to perform all phases of data movement and manipulation ***in parallel***, allowing significant performance gains compared to the same application run sequentially. In particular, the application allowed the user to

- query a parallel database (like DB2, Oracle, or Informix) and stream data in parallel out of the RDBMS
- pipe each stream through a transformation stage
- directly populate a parallel SAS data mart.

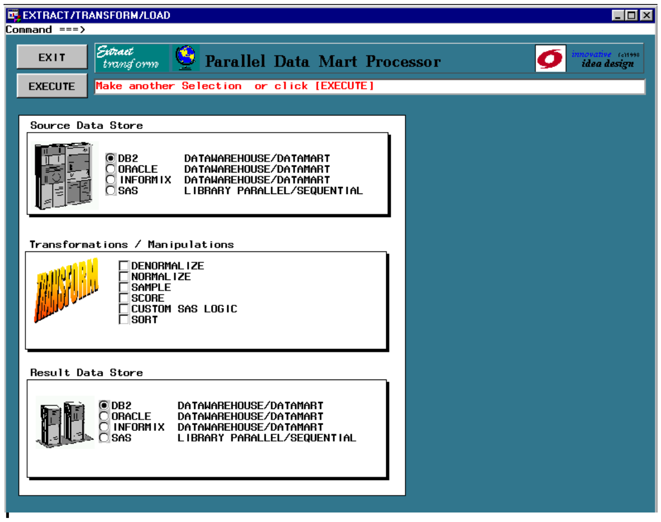
Furthermore, the SAS/AF application described in this paper allows the analyst to build and execute models against the full volume of mart data ***in parallel.***

REQUIREMENTS

The SAS/AF application executes from a PC control station running Windows95 or Windows NT. SAS/CONNECT® is used to communicate with the UNIX host. The RDBMS systems are located on the UNIX host, as are the SAS data marts. All data manipulations are performed on the UNIX host. SAS/Access® to the parallel database can be used to read the database catalog, allowing the user to interactively choose the table and field names needed to construct queries. SAS/AF also controls the Torrent® Orchestrate application environment—also installed on the UNIX host—which enables the parallel extraction, manipulation, and modeling of data.

EXTRACTING LARGE RESULT SETS FROM AN RDBMS USING PARALLEL STREAMS

Figure 1 shows the main screen to the interface, on which the user specifies the source database.



Command ===>

Make another Selection or click [EXECUTE]

Source Data Store

● DB2 DATAWAREHOUSE/DATAMART
○ ORACLE DATAWAREHOUSE/DATAMART
○ INFORMIX DATAWAREHOUSE/DATAMART
○ SAS LIBRARY PARALLEL/SEQUENTIAL

Transformations / Manipulations

□ DENORMALIZE
□ NORMALIZE
□ SAMPLE
□ SCORE
□ CUSTOM SAS LOGIC
□ SORT

Result Data Store

● DB2 DATAWAREHOUSE/DATAMART
○ ORACLE DATAWAREHOUSE/DATAMART
○ INFORMIX DATAWAREHOUSE/DATAMART
○ SAS LIBRARY PARALLEL/SEQUENTIAL

Figure 1. The SAS/AF interface main screen

The application uses SAS/Access to retrieve all table and field metadata information directly from the database catalogs. The Query

Definition Screen (Figure 2) displays the results to the user as a hierarchy of databases, tables, and columns.

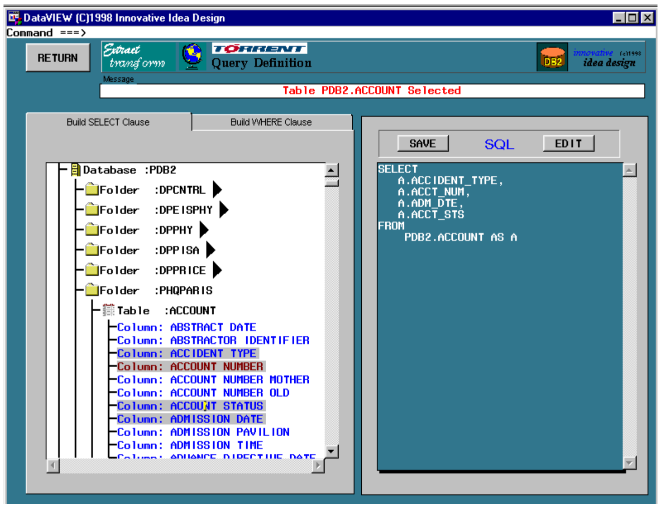


Table PDB2.ACCOUNT Selected

```
Database :PDB2
  Folder   :DPCNTRL
  Folder   :DPEISPHY
  Folder   :DPPHY
  Folder   :DPPISA
  Folder   :DPPRICE
  Folder   :PHQPARIS
    Table   :ACCOUNT
      Column: ABSTRACT DATE
      Column: ABSTRACTOR IDENTIFIER
      Column: ACCIDENT TYPE
      Column: ACCOUNT NUMBER
      Column: ACCOUNT NUMBER MOTHER
      Column: ACCOUNT NUMBER OLD
      Column: ACCOUNT STATUS
      Column: ADMISSION DATE
      Column: ADMISSION PAVILION
      Column: ADMISSION TIME
      Column: ADVANCE DIRECTIVE DATE
```

```
SELECT
   A.ACCIDENT_TYPE,
   A.ACCT_NUM,
   A.ADM_DTE,
   A.ACCT_STS
FROM
    PDB2.ACCOUNT AS A
```

Figure 2. The Extract/Transform Query Definition Screen

The user selects columns to include in the extract by clicking on a table column to add it to the SQL Select statement. The user may select columns from a single database table or may select columns from multiple tables and build a complex SQL query whose result set is returned from the database. The query is manually editable and changeable.

The generated SQL is loaded into a Torrent Orchestrate operator, which manages the parallel connections between the RDBMS and the downstream processes. This method allows complex joins to be performed within the database, taking full advantage of its internal parallelization, while Orchestrate manages the external parallelization. This results in a parallel flow of data within the RDBMS, out of the RDBMS, and into the downstream SAS processing.

The application executes as follows:

- The generated query—including UNIX host name, database name, and full query logic—is written to the Torrent Orchestrate Server Repository using SAS/Access to ODBC. All information entered to the server repository is done using standard SQL commands.
- The Torrent Orchestrate server then executes the query against the database using a parallel client method that extracts *n* streams of data—one for each of *n* partitions of the table(s) containing the data.
- The Torrent Orchestrate framework manages the complexities of passing the parallel streams of data from the RDBMS directly to any downstream processes.

Despite the parallel query mechanism built into most modern databases, result sets are still returned through a single coordinator node of the database***, producing at this node a sequential bottleneck to data throughput.* By generating multiple output streams from the database, this application removes the single most common rate-limiting step found in data management systems**. Only the number of partitions used to store data in the RDBMS now limits the rate of the extract application's performance.

Once the extraction logic has been specified, the user chooses a destination for the data. The extracted data may be used to

- ♦ feed a transformation step only
- ♦ populate a mart directly
- ♦ feed a transformation step and then populate a data mart.

PERFORMING TRANSFORMATION AND MANIPULATION OPERATIONS IN PARALLEL

The user may transform the extracted data by selecting from five categories of manipulations (all of which can execute in parallel):

- Denormalization/Normalization Transposes
- Sorting
- Scoring
- Sampling
- Custom SAS programming logic

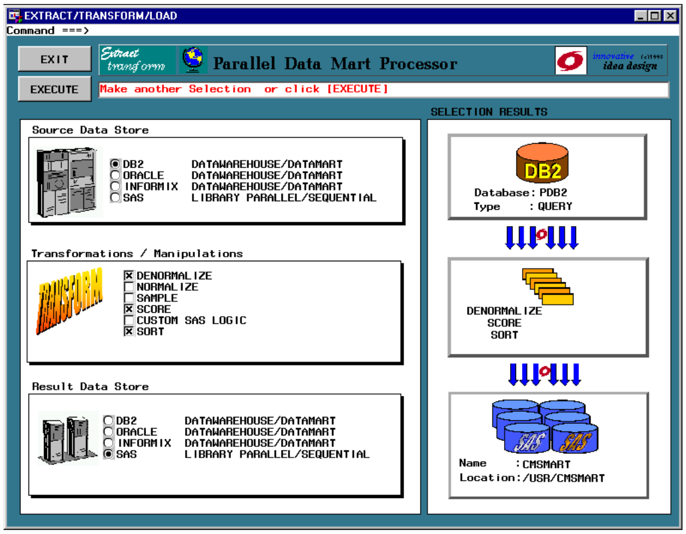
Depending on the options selected, the interface opens additional windows, in which the user specifies required information. In some cases, the user may also include existing SAS code. For example, if the user chooses a scoring manipulation, a dialog box prompts for the name and location of the SAS code file that contains the scoring logic.

The transformation step of the application can run in parallel, with the data feed from the extraction step performed directly, *without* first landing it to disk. This process, which saves execution time and disk space, is referred to as *pipeline parallelism.* When extracting large volumes of data from a source database, pipeline parallelism leads to performance gains by reducing I/O and allowing simultaneous execution of extraction and transform operations. Furthermore, pipeline parallelism eliminates sequential bottlenecks by allowing both the extraction and transformation to be performed.

If the transformations are the same for each record, it is not necessary to partition the parallel data streams from the data extract. If BY group operations are specified, then partitioning might be required. The application will automatically perform the partitioning, without user specification.

The user may also choose to populate the data mart directly without transforming the extracted data. In this case, the application still uses pipeline parallelism.

Figure 3 shows the options selected for the execution of the application. The options selected here will run a query against DB2, perform denormalize, score, and sort manipulations, and then land the data to a parallel SAS data mart.



Command ===>

Make another Selection or click [EXECUTE]

Source Data Store

● DB2 DATAWAREHOUSE/DATAMART
○ ORACLE DATAWAREHOUSE/DATAMART
○ INFORMIX DATAWAREHOUSE/DATAMART
○ SAS LIBRARY PARALLEL/SEQUENTIAL

Transformations / Manipulations

☒ DENORMALIZE
□ NORMALIZE
□ SAMPLE
☒ SCORE
□ CUSTOM SAS LOGIC
☒ SORT

Result Data Store

○ DB2 DATAWAREHOUSE/DATAMART
○ ORACLE DATAWAREHOUSE/DATAMART
○ INFORMIX DATAWAREHOUSE/DATAMART
● SAS LIBRARY PARALLEL/SEQUENTIAL

SELECTION RESULTS

Database: PDB2
Type : QUERY

DENORMALIZE
SCORE
SORT

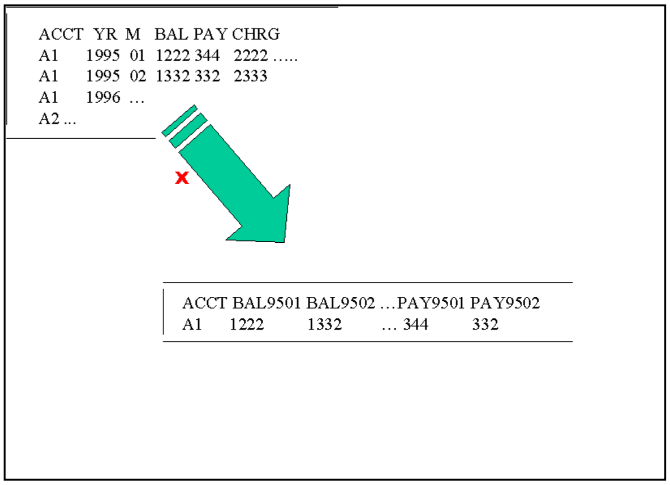
Name : CMSMART
Location: /USR/CMSMART

Figure 3.

The section below details the five categories of transformation/manipulation operations.

Denormalize/Normalize

Denormalization and normalization transformations are essentially transpose operations that stretch or collapse a data set. This facilitates the execution of some types of analyses. In the SAS language, DATA steps are typically used to perform this action.



ACCT	YR	M	BAL	PAY	CHRG
A1	1995	01	1222	344	2222 …..
A1	1995	02	1332	332	2333
A1	1996	…			
A2 ...					

ACCT	BAL9501	BAL9502	…PAY9501	PAY9502
A1	1222	1332	… 344	332

Figure 4 depicts a denormalization of a file by the keys ACCT, YR, and M.

Since BY group processing is required to perform this transformation, partitioning may also be required. If so, the application will uses Torrent Orchestrate partitioners to perform the task automatically. The user does not have to specify any additional information and the data is not landed to disk.

Sort

If the user chooses this manipulation, a window opens in which the user specifies the key variables for the operation. As in denormalization, if partitioning is required, the application will use Torrent Orchestrate partitioners to perform the task automatically.

Score

Scoring is the process of applying an equation or model to the records in a file and calculating a result or score. All models, whether created with Enterprise Miner software or other SAS procedures, may be run in parallel against the application data. Because Enterprise Miner-generated models tend to be very CPU-intensive, executing them in parallel significantly improves.

When the user selects the Score option, a window opens in which the user specifies the name of the SAS program to perform the scoring. A parser scans the SAS code to determine the execution mode that optimizes parallel performance.

Sample

It is common for users to build data marts from samples of a warehouse. It is also common for analysts to sub-sample the data mart when building models. The application supports sampling in parallel.

Selecting the Sample option displays a window that allows the user to specify a sample percentage of the extracted data the should be passed.

Custom SAS programming logic

Many SAS statistical procedures may be used to generate models in parallel, depending upon the existence of BY conditionals. For instance, if the data is or can be partitioned using the state field, and the analyst wishes to do a logistic regression of the data BY state, this model can be created in parallel. Where appropriate, the application allows the user to create such models in parallel.

Selecting this option opens a window in which the user specifies the name of the SAS program file to be used. As not all SAS programs can be run in parallel, a parser scans this code to determine the best mode of execution.

POPULATING DATA MARTS

To populate a data mart, the user selects a mart type, which opens additional windows for the user to specify the name and location of the target mart.

If the data is returned to an RDBMS data mart or data warehouse, parallel Torrent Orchestrate database load operators perform the action. This maintains end-to-end parallel processing from extraction to mart population.

Alternatively, the user may want to land the data to either a parallel or sequential SAS library. In this case, a Torrent Orchestrate SAS operator controls the creation of the SAS data sets. It is important to note here that the parallel SAS library is merely a collection of standard sequential SAS data sets, any one of which may be used in any SAS program. If parallel libraries are used, then the application maintains parallel processing from extraction to mart population. If a sequential SAS library is chosen, Torrent Orchestrate collects all the parallel data into one sequential SAS data set.

APPLICATION EXTENSIBILITY

This application is specific to data mart population and analysis. However, this application solves a problem which is ubiquitous in the IT world. The tools used to build the application are extensible and may be used to solve many other specific IT problems.

A custom control interface for any type of application can be built with SAS/AF. SAS/CONNECT and SAS/Access provide all the features needed to connect to a UNIX platform and talk to the RDBMS and the Torrent Orchestrate Server. By coupling the power and function of SAS software with the Torrent Systems' parallel application development and runtime environment many computational tasks can be run in parallel.

CONCLUSIONS

Variations on this application have been performed at a number of Fortune 1000 IT shops. Users have realized two valuable goals.

First is ease of use. The application described here and others deployed in IT shops put an easy-to-use, tailored SAS/AF face on parallel applications that are performing a tremendously complex set of UNIX operations under the hood. This means that shops need not maintain code or employ experts in parallel programming, database, or UNIX to achieve the benefits of high-performance parallel processing. Meanwhile, the IT shops can leverage their existing SAS expertise.

The second advantage is what the user gains by performing these large-volume extracts and SAS model executions in parallel. The user is now

- fully leveraging the processing power of multiprocessor parallel hardware—be it SMP or MPP architecture—to process a larger volume of data
- fitting jobs within existing batch windows
- running the application within a reasonable timeframe—no matter how large the data volume grows—because this parallel application is fully scalable.

CONTACT INFORMATION

Daniel W. Kohn
Torrent Systems, Inc.
5 Cambridge Center
Cambridge, MA 02142
Voice: 617.354.8484
Fax   : 617.354.6767
Email: dkohn@torrent.com

David L. Kuhn
Innovative Idea Design
PO Box 7085
Cave Creek, AZ 85327
Voice: 602.488.8085
Fax   : 602.488.9093
Email: ideas@extremezone.com

REFERENCES

[i] IBM Corporation and Torrent Systems (1997), "Achieving Scalable Performance for Large SAS Applications and Database Extracts," *A Torrent Systems and IBM Corporation Whitepaper.*