



THE
POWER
TO KNOW.

Technical Paper

SAS® In-Database Processing with Teradata:
An Overview of Foundation Technology

Table of Contents

Abstract	1
Introduction	1
Enhancing Base SAS® Procedures	2
Running SAS® Functions using "Shared Nothing" Teradata Processing	4
Understanding SAS® Formats?	5
Deploying SAS® Formats inside the Teradata System	6
Exporting Custom Format Definitions.....	6
Translating PUT() Function Calls into SAS_PUT() Embedded Function References.....	7
Extending Embedded Functions to Advanced Analytics	8
Models and Transformations as Embedded Functions	9
Managed SAS® Servers: Providing a Unified Administrative Experience	12
SAS® Stored Processes as Teradata External Stored Procedures	13
Conclusion	15
Resources	16

Abstract

Customer requirements to understand and act on critical business issues such as fraud detection, credit risk, price optimization, warranty analysis, and customer retention demand efficient handling and utilization of business intelligence (BI), data integration, and analytic components. The scope of analytic computations and the volume of data and data sources are growing at an unprecedented pace. Enterprises need flexibility to manage the analytical life cycle, from discovery to the execution of large numbers of new and existing analytic models that address functional and industry-specific business issues in a secure, scalable manner.

In October 2007, SAS and Teradata Corporation announced a strategic partnership that will enable customers to exploit the best of both companies' core strengths. This will be accomplished through deeper technical integration of their respective products. As part of the SAS® In-Database initiative, the foundation of the SAS and Teradata partnership is to enable businesses to run and optimize key aspects of SAS solutions and analytic processes *within* the Teradata Database. Customers will be able to apply SAS capabilities and analytic functions to data stored in Teradata while exploiting the core parallel processing inherent in Teradata's architecture.

Introduction

Significant trends in analytics and information management, along with regulatory and IT governance pressures, demand new technologies that can turn increasing volumes of data into insight that can be exploited very rapidly. In addition, customer dynamics are changing more frequently, which accelerates the need to develop and refresh analytic models that can be managed securely.

The partnership between SAS and Teradata addresses these challenges by moving analytic tasks closer to the data and improving the integration between SAS®9 and the enterprise data warehouse (EDW) from Teradata. Performing these tasks inside Teradata will significantly reduce bottlenecks that result from moving data over a network. Customers can also tap the respective strengths of SAS and Teradata technologies. In addition to reduced data movement, which results in improved performance run times and reduction in redundant data, analytic models can be developed and deployed faster, turning the data into usable insights.

One SAS and Teradata customer in the retail sector estimates elimination of 6 terabytes of redundant data. A major financial services organization plans to leverage the efficiencies of SAS In-Database processing with their Teradata system to increase SAS analytic processing speed ten-fold. This organization also estimates a reduction in model deployment time (from development to delivery) in terms of hours versus weeks. The reduced total cost of ownership (TCO) these efficiencies generate results in higher return on investment (ROI). Clearly this initiative will enable joint SAS and Teradata customers to focus more on deriving analytic insights from their Teradata system than managing their detail data stores.

The technical objectives of this partnership are to:

- Modify specific SAS components to delegate more processing to Teradata.
- Create a framework for advanced SAS® Analytics, SAS® Business Intelligence, and SAS® Data Integration software to take full advantage of the Teradata database engine's scalable processing power.
- Deliver a unified administration experience between the SAS®9 Platform and Teradata.

Today, product deliverables and continued product enhancements are underway that will be incrementally released and built on top of SAS®9. The initial release, described in this paper, focuses on the enhancement of key SAS® Foundation components that can add immediate value for the customer. These enhancements, in turn, provide a framework for future releases to include SAS®9 Business Intelligence, SAS Data Integration, and SAS Analytic Solution software bundles. The initial release focuses on five SAS®9 architectural elements, which echo the technical objectives of the partnership:

- delegation—SAS procedures are modified to push enhanced SQL to the Teradata system
- “shared nothing” scalability—SAS features are deployed as Teradata functions to take advantage of Teradata's processing power
- custom functions—user-written scoring functions that are deployed as embedded functions in the Teradata database.
- managed server model—the SAS®9 Platform is installed inside the Teradata system to provide a unified administration experience
- integration—SAS® Stored Processes are deployed as Teradata External Stored Procedures

This paper discusses these five elements using architectural models, functional use cases, and examples based on initial implementations.

The next section discusses how enhancements to Base SAS® procedures result in more work being done inside of Teradata.

Enhancing Base SAS® Procedures

The SAS System is comprised of hundreds of modules of pre-packaged functionality. These modules are referred to as *procedures*. There are procedures for basic data manipulation, reporting, forecasting, and statistical analysis among other things. SAS procedures communicate with Teradata via a SAS product called the SAS/ACCESS® Interface to Teradata, which is basically a data adapter that can interface directly with a Teradata system as shown in Figure 1.

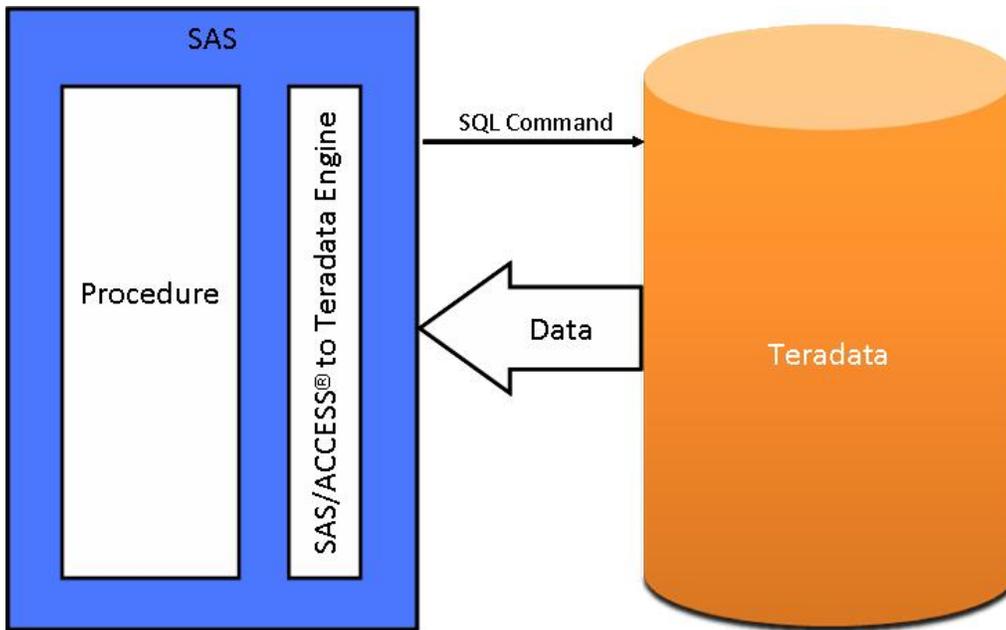


Figure 1. Conventional Processing with SAS and Teradata

Using conventional processing, the SAS procedure asks the SAS/ACCESS engine for all of the rows of the table being processed; the SAS/ACCESS engine generates an SQL SELECT * statement that is passed to the Teradata system. That SELECT statement fetches all of the rows in the table and the SAS/ACCESS engine returns them to the procedure. As the number of rows in the table grows over time, network latency grows because the amount of data that is fetched from the Teradata system to the SAS process increases. With SAS In-Database processing, we can reduce the amount of data extracted from the database management system (DBMS) simply by modifying certain SAS procedures to generate more sophisticated SQL queries. As an example, look at the SAS FREQ procedure.

PROC FREQ computes descriptive statistics based on unique values in the data set and produces n-way tabular reports. Here is a simple example of SAS code that calls PROC FREQ to calculate the frequencies and percentages of customer credit scores by the customers' home state:

```
proc freq data=customer.credit_data;
  table state * credit_score;
```

The DATA= specification on the PROC statement tells PROC FREQ to use the data found in a table called customer.credit_data for its calculation. **Customer** is a logical reference similar to a database schema designation that points to the physical location of the data, which in this example is a Teradata Database. **Credit_data** is the name of the Teradata Database table to use as input. The TABLE statement instructs PROC FREQ to gather statistics on the unique values of the **credit_score** column and group the results by state.

To produce this report, the unique values of the **credit_score** variable found in the **credit_data** table have to be counted by state. Some minimum and maximum column values must also be collected. Conventionally, these counts are done by PROC FREQ, but most database programmers will quickly realize that counting and finding minimum and maximum column values by group is a standard SQL operation. By modifying PROC FREQ to dynamically generate a SAS SQL view, these counts and statistics can be performed in the Teradata system.

Here is the SQL query that gathers the basic descriptive statistics that PROC FREQ needs for our sample case:

```
select count(*) as f1, state, min(state) as f2, credit_score,
       min(credit_score) as f3, max(credit_score) as f4
  from customer.credit_score
 group by state, credit_score
```

The query is given to the SAS/ACCESS engine where it can be passed to the Teradata Database engine. A result set of distinct credit scores by state is returned to the SAS process—not the entire table. This smaller result set is used by PROC FREQ in a SAS process to do the remaining analysis required to produce the final output. If the number of rows in **customer.credit_score** grows over time, the amount of data pulled from the EDW continues to be limited to the number of unique credit score values across all states.

Enabling procedures to delegate more of the processing to the Teradata Database by generating more complex SQL and passing it to Teradata optimizes the PROC FREQ processing. More work is done in the database and less data movement occurs. SAS is researching similar enhancements to other SAS analytic procedures including PROC RANK and PROC SUMMARY.

Each of these SAS procedures can be modified to dynamically generate SQL queries that reference Teradata SQL functions. While some work continues to be done by the procedure itself, this approach provides performance improvements in a number of cases.

While enhancing SAS procedures to generate SQL that is delegated to the Teradata Database offers large performance gains, some SAS analytic functionality cannot be easily expressed as an SQL query. The next interesting challenge that SAS In-Database addresses is the ability to package algorithms that are unique to SAS in such a way that they can be deployed and run inside the Teradata database itself to leverage Teradata's "shared nothing" architecture. The next section discusses a general mechanism for accomplishing this challenge.

Selective Base SAS and statistical procedures are planned to be available in the first Maintenance Release of SAS 9.2 in early 2009. Additional procedures for Base SAS, SAS/STAT® software, and SAS® Enterprise Miner are planned for incremental delivery in subsequent maintenance releases.

Running SAS® Functions using "Shared Nothing" Teradata Processing

Teradata is a *massive parallel processing* (MPP) architecture. MPP systems are also referred to as *shared nothing* environments. This means that each unit of parallelism is self contained and owns a portion of the database, allowing each unit to independently operate on its data, in parallel, completing the task in a fraction of the time. Figure 2 depicts the internal organization of a Teradata system.

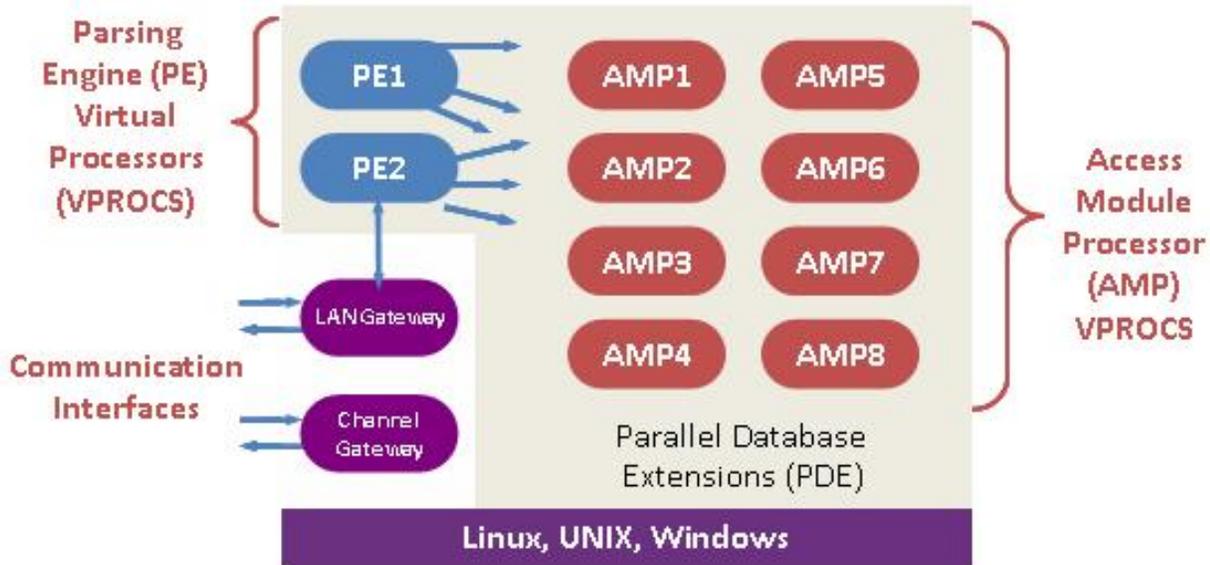


Figure 2. Teradata's Units of Parallelism

Teradata enables many individual servers to be interconnected and to behave as one large single-image system. The Virtual Processor (or VProc) is a key concept in Teradata's unconditional and universal parallelism. *VProcs* are independent software processes that execute tasks. There are two types: Access Module Processors (AMPs) that interface to the data and disk arrays and do the data crunching, and Parsing Engines (PEs), which interface with the applications and users and throttle the workload.

Let's look at a basic example of how SAS In-Database processing can deploy SAS functions across the Teradata Database MPP architecture. The objective is to package SAS code in such a way that it can be executed on Teradata AMPs. If we adhere to "shared nothing" coding principles when doing this, the SAS functions deployed inside the database will seamlessly exploit Teradata's universal parallelism. To demonstrate this, we will use components unique to SAS: SAS formats and SQL Pass-Through Facility.

Understanding SAS® Formats?

SAS formats are basically mapping functions. They change an element of data from one format to another. For example, there are SAS formats to change numeric values to various currency formats and date/time formats. SAS supplies many formats and, in addition, SAS programmers can define custom formats using the SAS FORMAT procedure. For example, the following PROC FORMAT code creates a custom format called \$REGION that maps zip codes into geographic regions:

```
proc format;
  value $region '02129', '03755', '10005' = 'Northeast'
               '27513', '27511', '27705' = 'Southeast'
               '92173', '97214', '94105' = 'Pacific'
               /* etc */;
run;
```

Formats, both those supplied by SAS and those written by users, are used quite frequently in SAS programs. They are referenced in numerous ways, but for SAS In-Database processing, using the PUT() function in SAS SQL is of particular interest. The SAS PUT() function takes a format reference and a data item as input and returns a formatted value. Here is a SAS SQL query that uses the PUT() function to select customers living in the Southeast from a table of all customers:

```
select * from sales.customers where put(zipcode, $region.) = 'Southeast'
```

The SAS SQL processor knows how to process the PUT() function. Today, this example SELECT statement is processed via the SAS/ACCESS Interface to Teradata. The PUT() function returns all the rows in the sales.customers table in the Teradata Database to the SAS System for processing. With SAS In-Database technology, the PUT() function and the \$REGION format definition can be deployed to the Teradata system so that the entire WHERE clause in the example SELECT statement can be processed by Teradata. The next section describes how this is accomplished.

Deploying SAS® Formats inside the Teradata System

Teradata supports embedded functions, which encapsulate a “shared nothing” unit of work and surface it to the Teradata AMPs. Like any other programming language function, an embedded function can take one or more input parameters and return an output value. SAS In-Database technology takes advantage of the function construct to package and deploy SAS functionality into Teradata.

We have ported the SAS PUT() function implementation to Teradata as an embedded function to support the use of SAS formats shown in our example. This function can then be specified in SQL queries submitted to Teradata by the SAS System. The signature of the embedded Teradata function, shown in the following example code as SAS_PUT(), parallels that of the SAS PUT() function:

```
SAS_PUT( source, format .)
```

Once the SAS_PUT() function is defined as an embedded function inside Teradata, the previous example SQL statement that uses the PUT function can be expressed as follows:

```
select * from sales.customers where sas_put(zipcode, $region.) = 'Southeast'
```

The SQL statement can also execute inside the Teradata system. In addition to allowing the filtering operation specified by the WHERE clause to execute in the database, Teradata’s universal parallelism will further enhance performance by distributing the SAS_PUT() function across all the available AMPs in Teradata’s “shared nothing” architecture.

SAS In-Database technology delivers the formats supplied by SAS in libraries that can be installed inside the Teradata system, where they can be invoked by the SAS_PUT() embedded function as needed. But it is still necessary to provide a way to export any custom format definitions created in SAS to Teradata and then enable the SAS SQL processor to translate PUT() function calls into SAS_PUT() embedded function references understood by Teradata.

Exporting Custom Format Definitions

The process for exporting custom format definitions is called *SAS Format Publishing*. The SAS FORMAT procedure enables you to create custom formats that replace raw data values with formatted character values. PROC FORMAT accomplishes this by creating a simple lookup table. SAS In-Database technology contains a tool for exporting these

lookup tables from the SAS System to the Teradata system. This tool is delivered as part of the SAS/ACCESS Interface to Teradata and is known as the SAS Format Publishing Agent for Teradata.

The PUT function, which returns a value using a specific SAS format, is used extensively in SAS processing. In ordinary SAS processing, using the PUT function and SAS formats in an SQL query would result in the query not being pushed to the database for processing. However, SAS In-Database processing, SAS Format Publishing, and recent enhancements to the SAS SQL Pass-Through Facility now make this possible.

The SAS SQL Pass-Through Facility is an implementation of the SAS SQL processor that works in conjunction with the SAS/ACCESS engines. The SQL Pass-Through Facility enables many complex SQL statements to be delegated directly to the underlying database for execution. Improvements to the SQL Pass-Through Facility enable PROC SQL to optimize performance by interpreting SAS SQL statements and, whenever possible, rewriting the SAS SQL into the database SQL and then passing it down to the database for processing. This attempt to optimize performance is made without you having to request it.

For more technical information about these features consult the *Base SAS 9.2 Procedures Guide* (SAS Institute Inc. 2008).

Translating PUT() Function Calls into SAS_PUT() Embedded Function References

Many SAS business intelligence (BI), data integration, and solution components rely on the SAS SQL processor to turn business rules into SAS SQL statements. SAS In-Database processing transparently maps SAS SQL functions like PUT() into corresponding Teradata embedded functions supplied by SAS. This is handled by the SAS/ACCESS Interface to Teradata. Figure 3 shows the runtime relationship among these components.

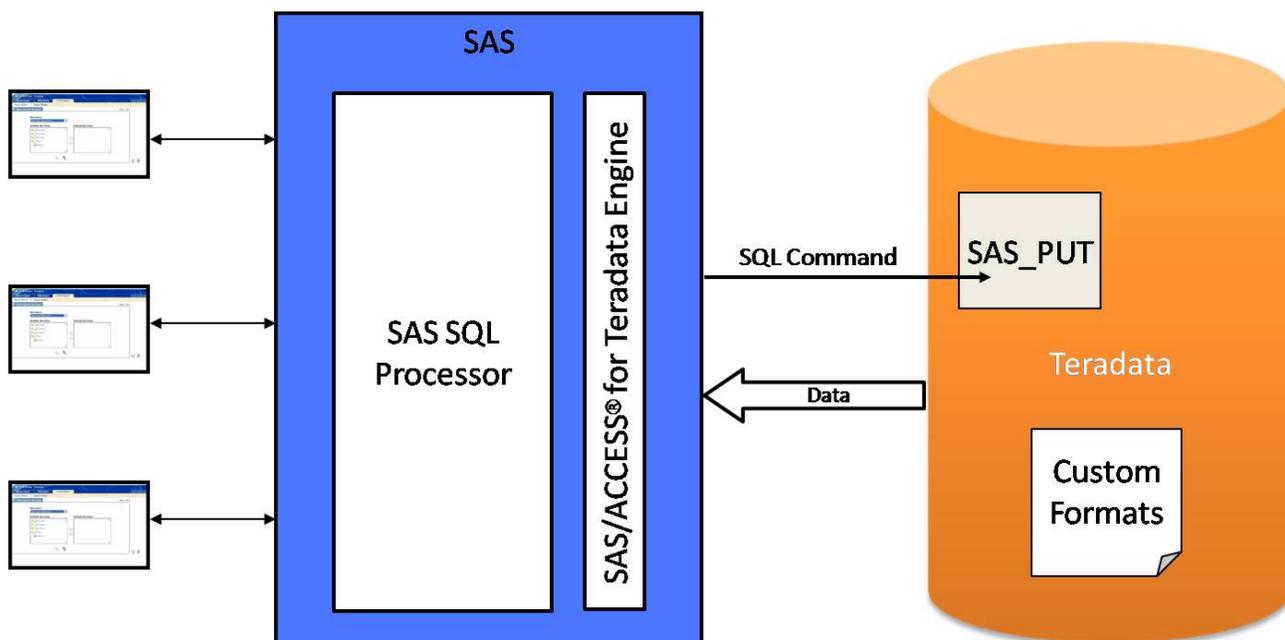


Figure 3. Embedded SAS Format Processing

Today, SAS applications generate SAS SQL code and push it to the SAS®9 Platform, which evaluates the request and routes it to the SAS SQL processor. As shown in Figure 3, when SAS In-Database processing is in effect and the SQL query contains PUT() function references, the SAS SQL processor and the SAS/ACCESS Interface to Teradata collaborate to translate PUT() references into SAS_PUT() embedded function references. The translated query is then passed to the Teradata system where the SAS_PUT() function is executed. As discussed earlier, benefits include improved runtime performance, less data movement, and “shared nothing” scalability.

The modifications to the SAS SQL processor and the SAS/ACCESS Interface to Teradata are initial deliverables that can be exploited by existing SAS applications without any modification to the SAS applications.

Although SAS published formats can be used today in both SAS 9.1.3 and SAS 9.2 clients, the ability to publish formats, both those supplied by SAS and custom formats, is available only with SAS 9.2 using the new SAS Format Publishing Agent in SAS/ACCESS to Teradata software.

Extending Embedded Functions to Advanced Analytics

The embedded function mechanism supported by Teradata can be extended with SAS In-Database processing to enable more sophisticated SAS analytic procedures to leverage the universal scalability of the Teradata system. The SAS System supports hundreds of procedures that perform advanced statistical, forecasting, and optimization functions. Specifically, SAS® Enterprise Miner™ and SAS/STAT® software include procedures to analyze detail data and create highly accurate predictive, optimization, and time-series forecasting models. These procedures often make multiple passes over input data executing statistical algorithms with each iteration. The procedure might apply different inputs to the algorithm being used or apply a different algorithm with each pass. The procedure contains the control logic that determines the number of iterations made over the input data or determines which statistical algorithm is applied with each iteration.

The statistical algorithms tend to operate on individual rows of data independently, which makes them well-suited for “shared nothing” execution in Teradata. By packaging these algorithms so that they can be deployed as Teradata functions, SAS In-Database processing can take advantage of the inherent scalability of Teradata processing while the controlling procedure continues to manage the top-level logic inherent in the analytic process being used.

Using Teradata functions to deploy analytic algorithms inside Teradata is basically the same process as using Teradata function to deploy the SAS PUT() described earlier in this paper. Similarly, enabling advanced SAS analytic procedures to push enhanced SQL directly to the DBMS is basically the same process described earlier in the PROC FREQ scenario. The result of applying these techniques to deploy the unique and powerful SAS analytic algorithms to execute inside a Teradata Database will also be the same: enhanced performance, reduced data movement, and exploitation of Teradata’s parallel processing.

But in addition, applying these techniques will optimize the processes of data discovery and analytic model training. Figure 4 depicts the transfer of work and related reduction in data movement between the current processing environment, which relies on the SAS/ACCESS Interface to Teradata and the environment that is created by SAS In-Database processing.

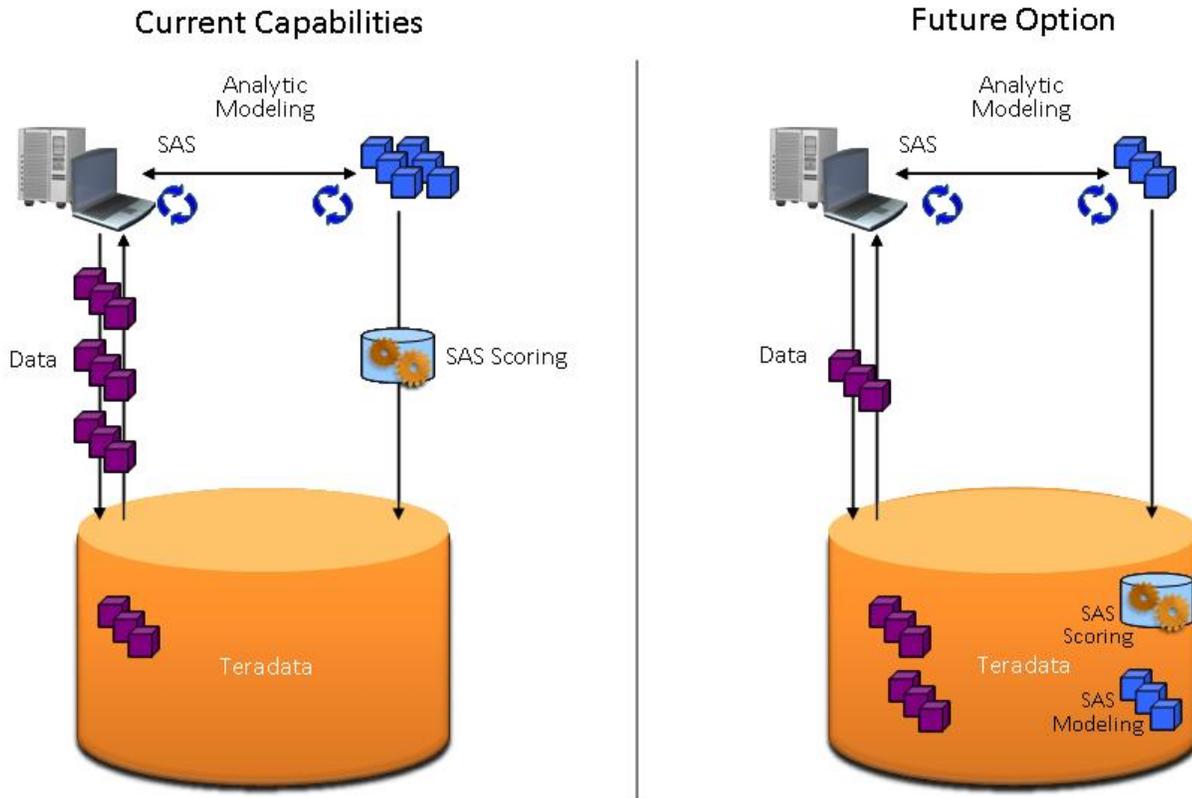


Figure 4. Conventional Processing Compared to Embedded Processing

This application of SAS In-Database technology is being pursued for a future release. Early delivery of such functions as the SAS_PUT() function and PROC FREQ enhancements lay a framework that makes such functionality possible.

Models and Transformations as Embedded Functions

Today, scoring functions and custom data transformations are frequently developed in the SAS System using the SAS DATA step programming language. The DATA step is a powerful 4GL programming language that enables tables to be manipulated and analyzed. The DATA step uses input from data tables to compute values, select specific input records for processing, and use conditional logic. DATA step execution is driven by an “implicit loop.” This loop processes individual rows of the input table and subsequently writes the results of processing the row to an output table. Thus, the logic encapsulated within a DATA step’s implicit loop naturally lends itself to “shared nothing” execution—each input row can be processed independently of any other row.

A further aim of SAS In-Database processing is to encapsulate the logic inside the DATA step implicit loop in a context that can be executed inside Teradata. SAS Enterprise Miner generates DATA step code that can be translated into an embedded function in Teradata. Due to architectural issues related to the implementation of the SAS DATA step engine, the process of translating a DATA step into a Teradata function is best handled by the SAS® Table Server Programming Language (TSPL). Relative to the DATA step, TSPL is a lightweight implementation that can be executed outside a SAS process. The SAS 9.2 releases of SAS In-Database technology include the tools needed to translate a user’s existing

SAS Enterprise Miner-generated DATA step program into TSPL-based functions. Figure 5 shows the process flow that translates a DATA step into such an embedded function using a product called the SAS Scoring Accelerator for Teradata.

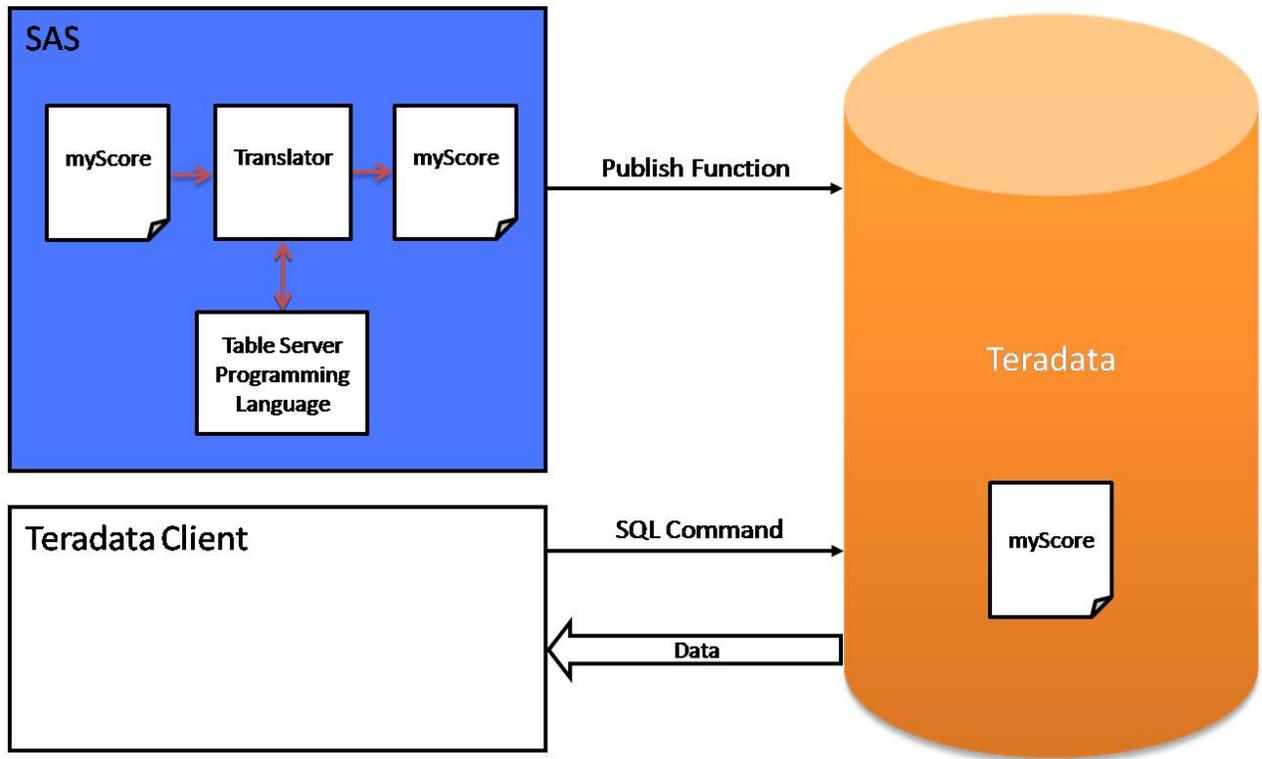


Figure 5. Publishing a Scoring Function

In this scenario, an analyst uses SAS Enterprise Miner to create a DATA step program: a model score called “myscore”. Using a new tool, PROC TSPLTRAN, the DATA step code is translated into a Teradata function definition using the TSPL engine. The function definition is published to the Teradata system using standard Teradata SQL commands. The scoring model is then available to a Teradata client application as an embedded function named “myscore” that can be invoked via any SQL statement.

Traditionally, we think of scoring as a back-end operation after data is loaded into the DBMS. For example, “myscore” could be executed by referencing it in the SELECT clause of an SQL statement. In performance testing experiments with this methodology, we have shown the inherent performance benefit of running scoring models as an embedded function. Figure 6 shows the results of one such performance test.

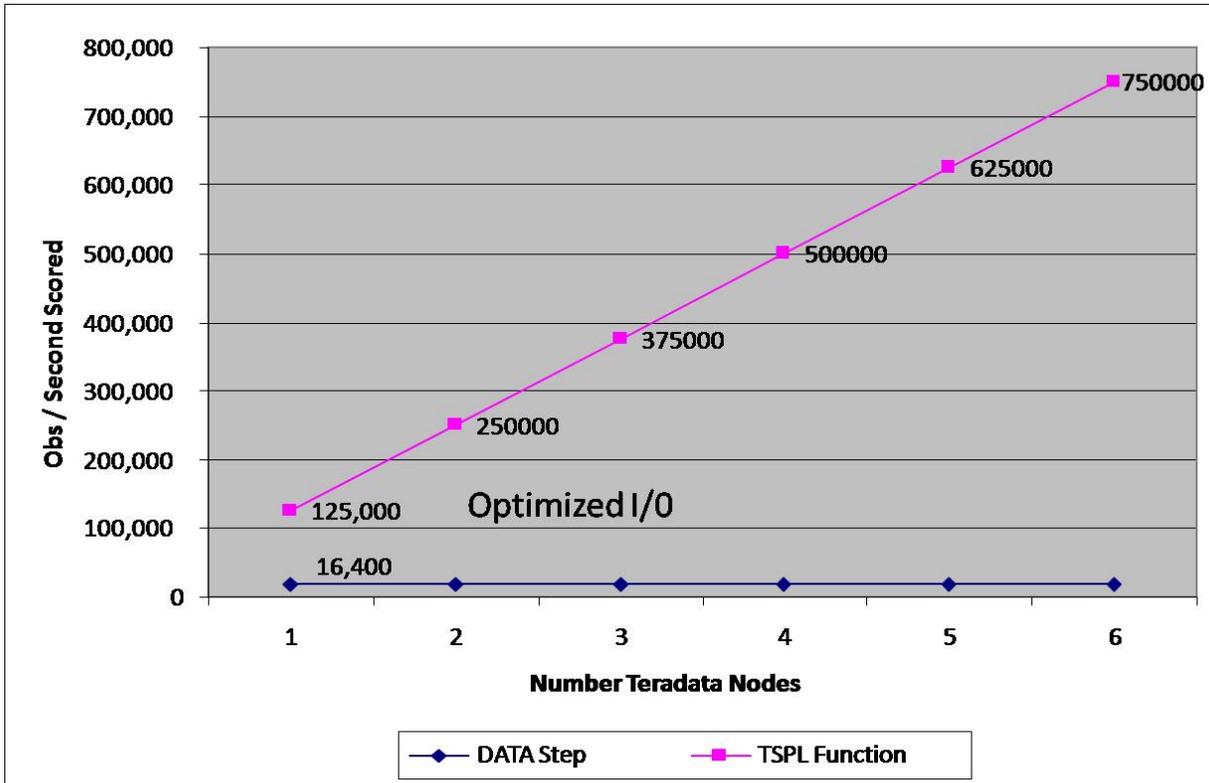


Figure 6. Performance Test Results

Because the exact number of rows scored per second depends on the complexity of the DATA step program, the throughput achieved with any given scoring model or transformation will vary. The DATA step-based approach requires the input data to be extracted from the Teradata system via the SAS/ACCESS Interface to Teradata. The flatness of the associated performance curve shown in Figure 6 reflects the bottleneck caused by extracting all the rows of the input table from Teradata to SAS. The performance curve given by the SAS In-Database approach reflects the result of executing the TSPL function in Teradata's scalable architecture: as AMPs are added to the Teradata system, the TSPL function scales linearly.

Additional applications of TSPL-based functions add further value to SAS In-Database processing. TSPL functions can be used to insert scoring models into the data load process. During the loading process or at "point of transaction," they can be used to execute complex business rules. For example, a model score that predicts a customer's propensity to respond favorably to a cross-sell or up-sell opportunity can be associated with an SQL trigger applied to table operations. When a customer interaction in a call center results in a read or update of the customer's database information, the TSPL-based function can be automatically executed and generates a real-time notification to the call center operator of the new sales opportunity. In this way, SAS scoring models deployed as Teradata functions provide additional business value—advanced analytic processing applied to the load process or at the point of transaction.

Some organizations have already applied similar methodologies to executing analytic models inside databases. However the technology currently available to them requires reimplementing of the scoring model by hand from DATA step code into SQL- or C-based embedded functions. This increases programming costs and time to deployment. SAS In-Database capabilities significantly reduce these costs by automating the translation of DATA step programs into Teradata functions that can be easily deployed into the enterprise data warehouse. Future releases of SAS In-Database will further optimize

the management and deployment of TSPL-based function scoring functions by integrating tools, such as the translator that converts DATA step programs to TSPL-based functions, with SAS® Model Manager, a component that addresses the needs of customers who deploy analytical models into operational environments.

The ability to publish SAS Enterprise Miner (release 5.3 and above) models as Teradata functions is available today with the SAS Scoring Accelerator for Teradata.

Managed SAS® Servers: Providing a Unified Administrative Experience

Related to SAS In-Database Processing, SAS has researched and tested various hardware appliance distribution models. This entails hosting the SAS® 9 Platform on hardware residing inside the DBMS chassis. The goal is to extend “turnkey” capabilities of appliances by integrating SAS software within the Teradata system. Such an offering can include SAS Business Intelligence, Analytics, and SAS Data Integration software installed inside a Teradata system and preconfigured to use SAS In-Database processing. We are also pursuing ways to deliver SAS Analytic Solutions using this integrated model. SAS Analytic Solutions are being re-engineered to run on the SAS®9 Platform; their implementation leverages SAS BI and data integration components. SAS® Anti-Money Laundering is a specific example. We are studying how this solution’s software, analytic models, and data models can be deployed inside a Teradata system when the system is built and configured before it is delivered to the customer. Figure 7 shows an architecture model for such a configuration.

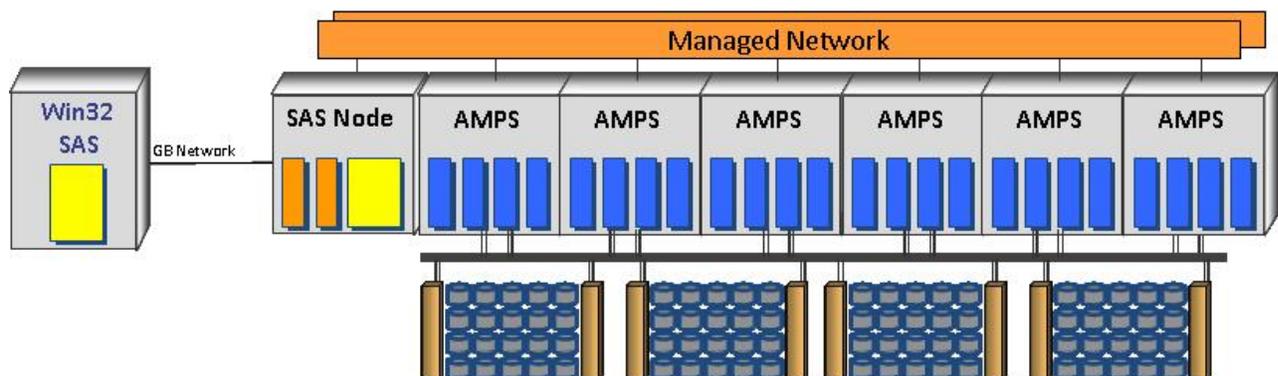


Figure 7. Managed Server Architecture

The SAS node depicted here represents a processing node inside the Teradata chassis that has been configured with SAS®9 Platform and solution software. A SAS client connects to the SAS node inside the Teradata system and, except for improved performance, the client application’s behavior remains unchanged. From the system administrator’s perspective, there is a single piece of hardware to manage that delivers both the Teradata system and SAS®9 analytic solution services.

In addition to adding value through simplified deployment and administration experiences, running SAS server processes inside a Teradata system brings performance gains with a dedicated in-cabinet ethernet. In one experiment conducted early in development, we connected a SAS process directly to a Teradata Managed Ethernet Server (MES) node resulting in significant performance improvement.

The partnership between SAS and Teradata provides significant business value by offering an integrated package that enriches the functionality of the Teradata system with access to SAS processes while the SAS software components

benefit from the highly scalable nature of Teradata's architecture. The joint customer's time to deployment is reduced, and their IT department experiences reduced infrastructure cost via a consolidated administration "footprint" between two enterprise-class computing systems that otherwise are managed as separate and distinct.

This will be accomplished by adhering to specific design principles:

1. Fully exploit Teradata system hardware resources.

The internal architecture in production today in the SAS System to interoperate with the Teradata system results in frequent movement of data between the two environments. This data movement can result in random and inconsistent activity across Teradata AMPs. Enabling SAS procedures to delegate more work to the Teradata system, and deploying select SAS functions that lend themselves to "shared nothing" processing, means AMPs can be fully utilized. This results in higher IT ROI in hardware and software costs.

2. Maximize the manageability of SAS artifacts deployed inside the Teradata system.

Components deployed inside the Teradata system, for example PROC FORMAT definitions, TSPL-based functions, and even External Stored Procedure definitions for SAS Stored Processes execution (described in the next section) are persisted inside the EDW as native database objects. This enables the SAS artifacts to be managed like any other database object. Adhering to this principle ensures the integration goals of the SAS In-Database initiative.

3. Maintain reliability and integrity inside the Teradata system.

Organizations rely on Teradata systems to deliver 24x7 availability and maintain the correctness of the information stored in the database. SAS In-Database components are implemented in such a way that an error during execution does not disrupt services or compromise data integrity.

These rules keep the design and implementation of SAS In-Database technology aligned with the initiative's overall business goals. IT infrastructure costs are reduced and investments are maximized. This extends the business value created by SAS In-Database processing beyond optimized business process implementation and execution.

While the SAS Managed Server initiative is distinct from the SAS In-Database processing initiative in many ways, there is overlap in the areas of SAS procedure enhancements, embedded formats, and DATA step programs deployed as Teradata functions.

SAS Managed Servers have been tested and validated for SAS Foundation, SAS Business Intelligence, and SAS Analytics in both SAS 9.1.3 and SAS 9.2 environments. Bundled and Packaged offerings of available software configurations are planned to be available in 2009.

SAS® Stored Processes as Teradata External Stored Procedures

Another project being pursued for a 2009 release of SAS In-Database processing relates to the SAS Managed Server initiative: SAS® Stored Process Server integration.

The SAS Stored Process Server is a key component of the SAS®9 Platform. It executes SAS programs (called stored processes in this context) that are stored on a central server. These programs use SAS reporting capabilities to deliver

analysis, business trends, and forecasting insights. Stored processes enable customers to centrally maintain and manage code, providing better control over changes, enhancing security and application integrity, and ensuring that every client executes the latest version of the code.

More specifically, the SAS Stored Process Server implements a set of interfaces that enable a client to invoke SAS programs. We plan to integrate SAS Stored Processes with Teradata External Stored Procedures in an early release of SAS In-Database technology making it possible to execute a SAS program from a Teradata application. An External Stored Procedure is a custom Teradata procedure written in C/C++ or Java Language that can operate on data stored in relational tables, database structures, or even on external data. A Teradata application uses the SQL CALL statement to execute an External Stored Procedure that in turn invokes the SAS Stored Process Server interfaces. Figure 8 shows the relationship between these components.

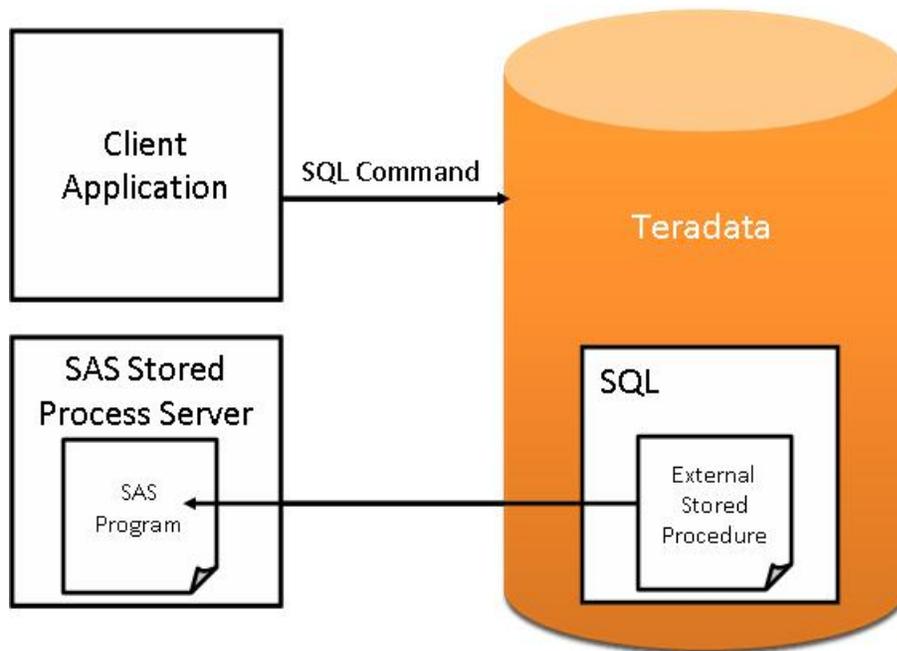


Figure 8. Calling a SAS Stored Process from an External Stored Procedure

The ability to integrate SAS Stored Processes with Teradata External Stored Procedures is a very powerful and flexible tool. SAS analytic solutions use stored processes to implement business logic and reporting; SAS® Data Integration Studio uses them for reusable ETL workflows. Coupling SAS Stored Processes to Teradata External Stored Procedures helps create a single, tightly coupled package for customers who choose to standardize a SAS solution on a Teradata system. Furthermore, this architecture supports the SAS Managed Server initiative by tightly connecting the SAS Stored Process Server to Teradata's functional and administrative systems.

The ability to call a SAS Stored Process from a Teradata External Stored Procedure is planned to be available with the release of SAS 9.2 Phase 2 in 2009.

Conclusion

The technology being jointly pursued by SAS and Teradata adds value to an organization by

- lowering the total cost of ownership
- enhancing the effectiveness of analysts, who can stay focused on higher-value tasks
- scaling SAS processes inside the Teradata system
- increasing performance and reducing data movement
- optimizing resource utilization across the analytic/warehousing environment
- enabling analytic SAS functions to occur at the point of transaction
- streamlining data discovery and model development processes resulting in faster time to analytic answers
- improving data quality and consistency
- supporting corporate governance processes
- increasing process robustness by reducing implementation restrictions

The 2008 releases of SAS In-Database processing with Teradata addresses customer expectations for efficient use of software and hardware resources to accelerate the creation, deployment, and execution of mission-critical business applications. IT costs are reduced via simplified deployment processes and integrated system management.

The initial releases of SAS In-Database Processing with Teradata immediately adds value to customers' business processing environments by enabling Base SAS summary procedures to delegate more work to the Teradata system. This enables SAS formats and scoring models to run inside the EDW, and integrates SAS stored processes with Teradata External Stored Procedures. The initially planned functionality provides an immediate performance benefit to SAS Business Intelligence, Data Integration, and Solution software suites by targeting low-level SAS functions and procedures that are used throughout the SAS®9 Platform. These enhancements are designed to enable their capabilities to be transparently available to the rest of the SAS software portfolio without modification or re-deployment.

In addition, initial technology creates an architectural framework for other SAS components in the areas of Business Intelligence, Data Integration, and Advanced SAS Analytics so that these software components can take advantage of the SAS In-Database technology in the future.

The SAS and Teradata partnership offers our customers innovative solutions and delivers better, smarter answers faster. Together, SAS and Teradata have developed a joint, multi-phase roadmap to deliver these enhancements to our customers starting in 2008.

To learn more about SAS In-Database Processing with Teradata, visit either company's Web site¹ or contact the SAS and Teradata Center of Excellence (CoE).

¹ SAS and Teradata Partnership Web sites: www.sas.com/Teradata and www.teradata.com/SAS

Resources

SAS Institute Inc. 2008. *Base SAS® 9.2 Procedures Guide*. Cary, NC: SAS Institute Inc. Available at support.sas.com/documentation/cdl/en/proc/59565/HTML/default/procwhatsnew902.htm.

