Paper SAS461-2017

Befriend SAS® In-Database Technologies to Accelerate SAS® Integration with Your Data Platform

Tatyana Petrova, SAS Institute Inc., Cary, NC

ABSTRACT

You have SAS® software. You have databases or data platforms like Hadoop, possibly with some large distributed data.

If you already know how to make SAS talk to your data platforms, you have already taken a solid step toward a successful integration.

But you might also want to know how to take this communication to a different level.

If your data platform is the one that is built for massively parallel processing, chances are that SAS has already created the SAS® Embedded Process framework that allows SAS tasks to be embedded next to your data sources for execution. SAS® In-Database Technologies is a family of products that use this framework and provide an accelerated level of integration. This paper explains the core principles behind these technologies and presents application scenarios for each of these products.

INTRODUCTION

In SAS, data access to external data sources is based on two core principles:

- connect SAS with lots of data in a simple and transparent fashion.
- push processing to data platforms as much as possible to minimize data movement and increase efficiency.

Over the years, development in this area has been driven by these principles. The result is an extensive portfolio of data access products that deliver intelligence and flexibility to SAS integration with third-party data providers.

Essentially, SAS provides two levels of integration with external databases and data platforms:

The 1st level of integration establishes the fundamental access to data. It collaborates with the platform via the client libraries of the data vendor. This level of integration is designed to accommodate for specificities of each individual data platform, while effectively hiding the platform complexity from the data requestor. **SAS/ACCESS® software** lie at the heart of this integration.

The SAS/ACCESS® interface for each corresponding data platform enables the connection and intelligent SQL communication between SAS and the SQL engine of the database or Hadoop. The SAS/ACCESS interface is capable of translating SAS code into a single SQL query or a series of SQL queries that are optimized for the native capabilities of the particular data platform. Examples of SAS code that might be translated into native SQL calls for the database are PROC SQL, PROC SUMMARY, PROC FREQ, PROC SORT, and several other procedures. While SAS/ACCESS does not guarantee that all processing would take place in a database, it makes a smart attempt to push down as much processing as possible depending on the SQL code structures provided, the capabilities of the vendor client libraries, and the database capabilities. When portions of processing cannot be completed in database, the necessary amount of data is passed over to SAS to finish the processing.

SAS/ACCESS brings other important benefits such as data types mapping, bulk loading integration, internationalization, and so on. SAS currently supports more than two dozen different SAS/ACCESS interfaces, and this list continues to grow.

The SAS/ACCESS interfaces are an integral part of bridging SAS with external data sources. They are widely adopted in the market, and for many customers, this level of integration is powerful enough to effectively support their data management and data mining use cases.

The 2nd level of integration extends the communication from level 1 by enabling an even deeper integration with databases and data platforms. **SAS In-Database Technologies** is a suite of products that supports this level of integration with the help of the SAS Embedded Process framework. The role of SAS In-Database Technologies is to embed SAS code, including highly complex programs, into a database or a data platform and to execute this code in close collaboration with the database processing engine in a parallel fashion across the distributed data.

A number of resources are available for those interested in learning more about SAS/ACCESS integration.

This paper focuses on the second level of Integration: SAS In-Database Technologies. It explains the fundamental concepts and the architecture of SAS In-Database Technologies. This paper also reviews the portfolio of SAS In-Database Technologies products and explores the purpose and capabilities of each. For a closer look into what happens behind the scenes, see the “Beyond the Fundamentals” section.

SAS In-Database Technologies are available for a number of databases and a data platform (Hadoop). For simplicity, the term ‘databases’ is used throughout the paper to encompass both databases and data platforms like Hadoop.

When discussing the availability of products and components, this paper refers to the products available in the fourth maintenance release of SAS 9.4.

SAS IN-DATABASE TECHNOLOGIES FUNDAMENTALS

DS2

To understand the fundamentals of SAS In-Database Technologies, it is helpful to be familiar with the concepts of SAS® DS2 language. However, knowledge of DS2 is not essential to be able to leverage these technologies. Depending on the use case, data manipulations can be done with other tools including a series of SAS point-and-click applications that generate the code or leverage DS2 and SAS In-Database Technologies behind the scenes.

It is also not essential to be running in database every program that is written in DS2. DS2 can use various execution environments: Base SAS, SAS In-Memory Engine, such as SAS Cloud Analytics Services, or with the help of SAS In-Database Technologies, databases or Hadoop.

DS2 is a procedural programming language that supports modern programming constructs and a rich set of data types. DS2 encourages code reuse and is friendly to object-oriented programming techniques. In addition, it is built with parallelization in mind, providing multi-threaded programming mechanisms. This makes it a great candidate for integration with massively parallel processing (MPP) environments.

To program with DS2, you can use the DS2 procedure. This procedure allows you to write DS2 methods and statements and submit them as a stand-alone program. PROC DS2 is included with Base SAS.

The References section of this paper lists a number of resources for anyone wanting a quick overview or to learn more about this powerful language in detail. “Parallel Data Preparation with the DS2 Programming Language” by J. Secosky and G.Otto is a good place to start because this paper introduces DS2 from the angle most relevant to the current topic.

SAS EMBEDDED PROCESS

The SAS Embedded Process is designed to be deployed on a third-party vendor data platform to support the execution of multithreaded SAS DS2 programs.

The mechanics of how the SAS Embedded Process interacts with the particular data platform are different depending on the data platform. This is where a close collaboration between SAS and database vendors comes to play. Together we co-develop an effective integration of technologies that leverage specific to the platform advances.

The SAS Embedded Process framework truly benefits MPP database environments that consist of a collection of machines (cluster). However, the SAS Embedded Process can be deployed and successfully used on single-node data platforms. In fact, the SAS Embedded Process framework is available for some databases that are not MPP in nature, for example Oracle. Yet, the ability to run a DS2 program in multiple threads in a large Oracle Exadata environment brings its performance benefits.

The SAS Embedded Process is not a product in itself. It is a framework that SAS In-Database Technologies are built upon. In "Beyond the Fundamentals" section of this paper, you can find additional details about the role of SAS Embedded Process in support of particular SAS In-Database Technologies products.

SAS IN-DATABASE TECHNOLOGIES ARCHITECTURE

Previously, this paper noted that the purpose of SAS In-Database Technologies is to allow SAS programs to execute inside the database or in case of Hadoop, data platform.

We also noted that SAS In-Database Technologies are using DS2 as a language that SAS Embedded Process is capable to run in database. So SAS programs that you might want to run in database should either be written in DS2 or be converted to DS2 by SAS In-Database Technologies.

Figure 1 depicts the conceptual SAS In-Database Technologies architecture.

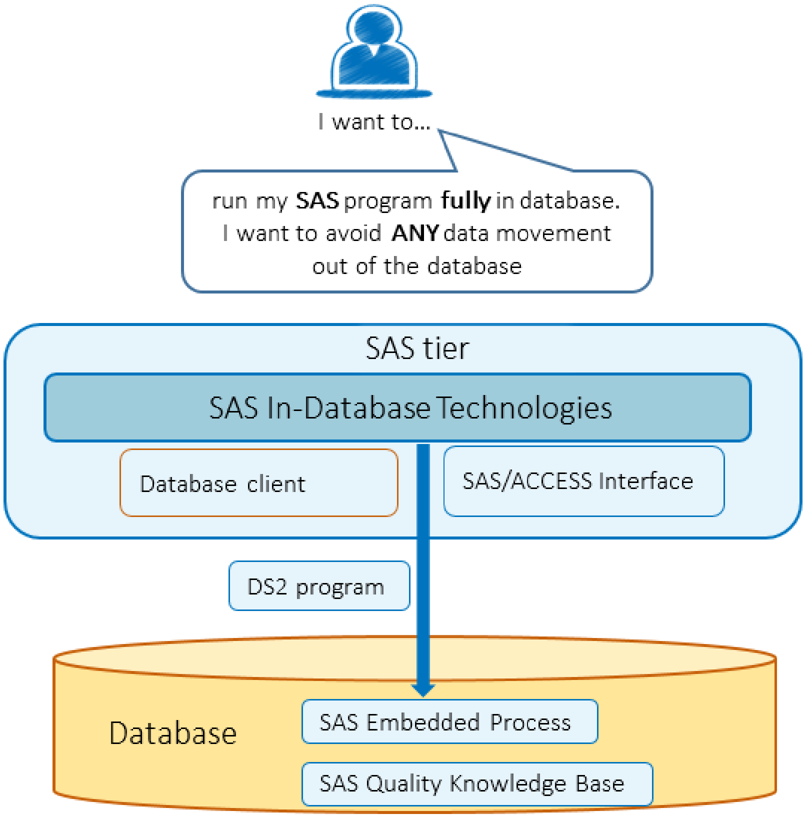

I want to...
run my SAS program fully in database. I want to avoid ANY data movement out of the database
SAS tier
SAS In-Database Technologies
Database client
SAS/ACCESS Interface
DS2 program
Database
SAS Embedded Process
SAS Quality Knowledge Base

Figure 1. SAS In-Database Technologies Architecture

The functionality in SAS In-Database Technologies is relying on SAS/ACCESS to handle the connectivity and communication between SAS and the database. It is essential to have SAS/ACCESS deployed on the same machine that issues the in-database tasks, and a SAS/ACCESS LIBNAME should be created before pointing in-database tasks to the corresponding data source. Recall that SAS/ACCESS is using database client libraries. Therefore, a database client is expected on the SAS tier.

Figure 1 indicates the presence of a SAS® Quality Knowledge Base (QKB) component on the database tier. SAS Quality Knowledge Base plays a role in the support of SAS data quality operations managed by SAS® Data Quality Accelerator. It is not required for other products in the SAS In-Database Technologies family.

In case of a multi-node database environment, the SAS Embedded Process must be deployed on every node of the cluster. If the SAS Quality Knowledge Base is used, it also needs to be available on every node.

The SAS Embedded Process framework serves as a foundation for the SAS In-Database Technologies products. These products augment the SAS Embedded Process framework with additional components, resulting in a specific range of capabilities.

It is time to look at the purpose and capabilities of each of these products.

SAS IN-DATABASE TECHNOLOGIES FAMILY OF PRODUCTS

The SAS In-Database Technologies family of products consists of the following members:

SAS® IN-DATABASE CODE ACCELERATOR

Purpose: enable custom PROC DS2 programs to run in database.

Why It Is Important: DS2 is a powerful language that can assist with the most intricate tasks for data manipulations. With a rich feature set to support multi-threaded programming, PROC DS2 programs can be built for parallelization. SAS In-Database Code Accelerator brings the robustness of PROC DS2 to your database environment by combining the power of your MPP engines with SAS data preparation and advanced analytics computations.

Common Tasks:

- integrate data (join data from multiple sources on various conditions)
- adjust formats
- construct new attributes
- reduce data (de-duplicate, reduce the number of variables)
- transpose data
- custom scoring algorithms
- analytical computations

Graphical User Interface: To integrate with Hadoop, SAS® Data Loader for Hadoop provides a point-and-click interface for profiling, managing, cleansing, and copying data to and from Hadoop. SAS In-Database Code Accelerator for Hadoop is used by a number of SAS Data Loader tasks to complete data transformations in Hadoop.

SAS® SCORING ACCELERATOR

Purpose: operationalize your scoring models; enable publishing and execution of SAS scoring methods on production data in database.

Why It Is Important: After the scoring model is developed with the use of SAS® Enterprise Miner™, SAS® Factory Miner, SAS/STAT®, or other SAS analytics applications, the models need to be put in production to score data. Scoring data usually involves highly intensive computations, and your production data might be quite large. It is also common that data needs to be scored as it comes into the system. Both aspects require scoring to happen next to the data source. SAS Scoring Accelerators enable this critical step in analytics lifecycle: the ability to operationalize your SAS scoring models within your databases environments.

Common Tasks:

- publish a model that was developed using a SAS analytics tool
- execute the model scoring methods in database on new data

Graphical User Interface: SAS® Model Manager (a product available within the SAS® Decision Manager offering) provides a convenient web-based application to manage and operationalize your repository of models. It supports publishing to databases when SAS Scoring Accelerator for the particular database is available.

SAS® DATA QUALITY ACCELERATOR

Purpose: enable SAS data quality operations to execute inside the database

Why It is Important: Data cleansing is an essential task in data preparation. Data standardization, matching, and other data quality operations frequently impact a handful of columns across a wide set of tables. In many cases, completing these operations within the database environment is more efficient than moving all the impacted tables over the network into SAS. Keeping data quality operations next to data sources enables you to improve the quality of more of your data in less time.

Common Tasks: Clean data. Common data quality operations include:

- casing
- attribute extraction
- gender analysis
- identification analysis
- matchcode generation
- parsing
- pattern analysis
- standardization

Graphical User Interface: To integrate with Hadoop, SAS Data Loader for Hadoop provides a point-and-click interface for profiling, managing, cleansing, and copying data to and from Hadoop. SAS® Data Quality Accelerator for Hadoop is used by the data cleansing directives in SAS Data Loader to complete these data quality operations in Hadoop.

Important disclaimer for SAS Data Quality Accelerator for Hadoop: As of the fourth maintenance release for SAS 9.4, only data cleansing code generated by SAS Data Loader for Hadoop is supported for execution in Hadoop. While custom PROC DS2 code with data cleansing operations can be programmed or extracted from SAS Data Loader for Hadoop for further modifications, this capability is considered pre-production. In fact, data quality functions invocation syntax for custom programs might change going forward. Support for custom programs is planned for future releases of SAS Data Quality Accelerator for Hadoop.

PROC TRANSPOSE PUSH DOWN CAPABILITY

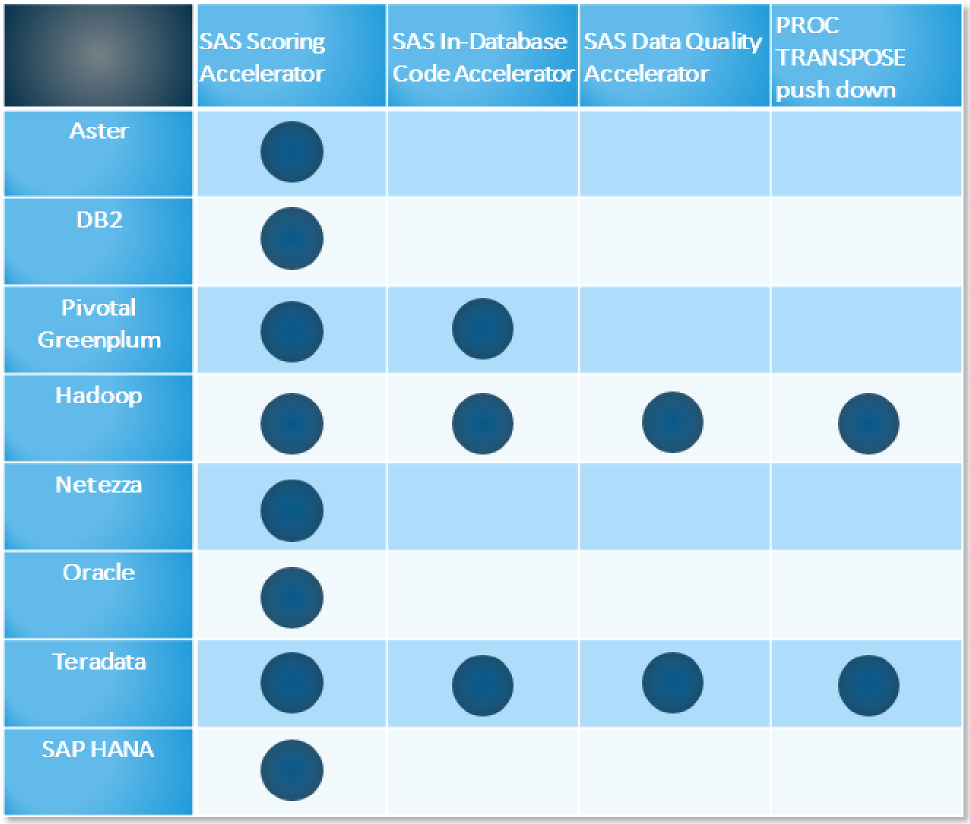
Purpose: enable data transposition to take place in database

Why It Is Important: PROC TRANSPOSE is known for its high CPU intensity. While a resource-costly operation, transpositions are a common task in a data preparation cycle, which qualifies it as a solid candidate to benefit from joining the in-database collection of tools.

Common tasks: data transpositions

Graphical User Interface: The Transpose Data directive in SAS Data Loader for Hadoop executes the transposition in Hadoop. The Transpose Transformation in SAS® Data Integration Studio supports running in Hadoop and Teradata if the corresponding SAS In-Database Technologies software is licensed.

Table 1 maps the availability of the SAS In-Database Technologies components per database in the fourth maintenance release for SAS 9.4.



	SAS Scoring Accelerator	SAS In-Database Code Accelerator	SAS Data Quality Accelerator	PROC TRANSPOSE push down
Aster	●			
DB2	●			
Pivotal Greenplum	●	●		
Hadoop	●	●	●	●
Netezza	●			
Oracle	●			
Teradata	●	●	●	●
SAP HANA	●			

Table 1. Availability of SAS In-Database Technologies Components per Database in SAS 9.4 M4

You might have also heard of SAS® Analytics Accelerator for Teradata. SAS Analytics Accelerator for Teradata was created before the invention of the SAS Embedded Process framework, and it uses a different integration mechanism based on Teradata user-defined functions. The functionality in SAS Analytics Accelerator for Teradata is beyond the scope of this paper.

SAS IN-DATABASE TECHNOLOGIES: THE SUM IS BETTER THAN THE PARTS

Each member of the SAS In-Database Technologies family plays its specific role well. But the most value that this family can produce is when all the members act together in a team effort towards the data processing efficiency.

The previous section highlighted some of the common tasks that these products allow to optimize by processing delegation. These lists of tasks are not exhaustive, especially for SAS In-Database Code Accelerator which supports custom PROC DS2 programming. These lists also somewhat overlap if you compare common tasks per SAS In-Database Technologies product. This overlap is intentional and shows the complementary nature of this product family. This idea might be best illustrated with examples of the data manipulation flows. (See Figures 2 and 3.)

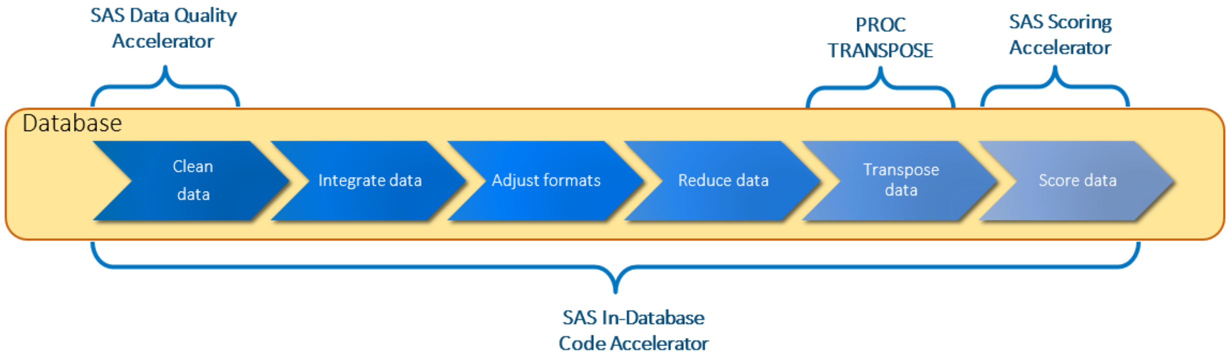


Figure 2. Example of a Flow to Prepare Data for Scoring Followed by the Scoring Step

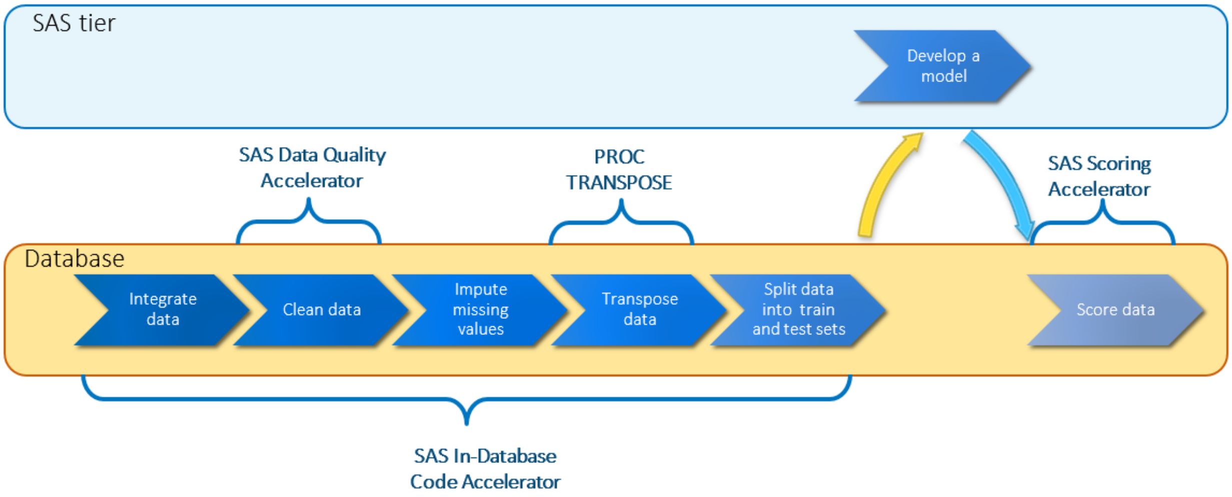


Figure 3. Example of a Flow to Prepare Data for Model Development Followed by the Scoring Step

From these flows, you can see that SAS In-Database Code Accelerator can be used in various capacities:

- basic data pre-processing.
- advanced data preparation.
- customization for any modifications needed to generated DS2 code. These modifications could be data quality operations generated by SAS Data Loader for Hadoop or transpositions code generated by PROC TRANSPOSE. Sophisticated data scientists can also create and push down to databases their custom data scoring algorithms.

BEYOND THE FUNDAMENTALS: HOW DOES IT WORK?

The ultimate goal of SAS In-Database Technologies is to enable SAS programs to execute in a distributed environment native to data sources. A curious reader might wonder what the processing flow is for each SAS In-Database Technologies component. Ultimately, you might want to know what parts of processing take place where, and if data stays in its original location or gets copied over your network.

First, for comparison and deeper understanding, review the process flow for the SAS/ACCESS interfaces. Figure 4 shows a typical task for SAS/ACCESS: request for data based on a certain criteria.

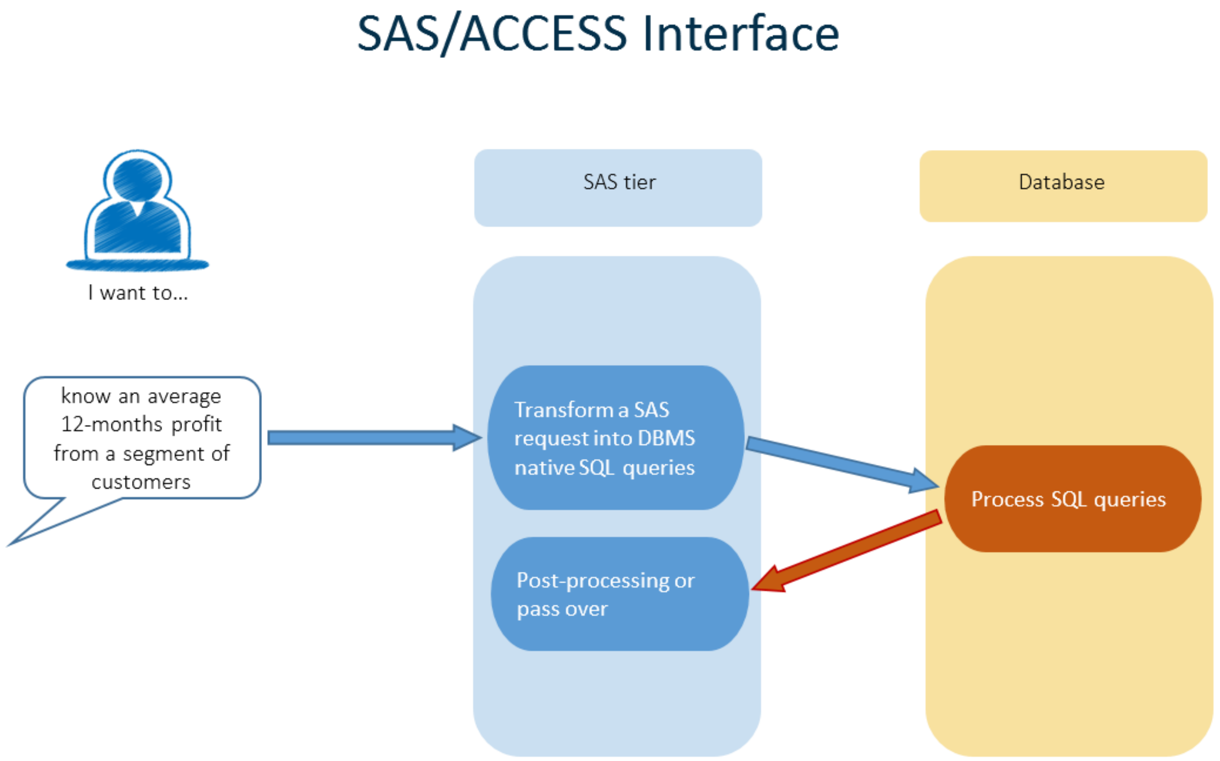

SAS/ACCESS Interface
I want to...
know an average 12-months profit from a segment of customers
SAS tier
Database
Transform a SAS request into DBMS native SQL queries
Process SQL queries
Post-processing or pass over

Figure 4. Processing Flow for SAS/ACCESS Interface

The SAS/ACCESS interfaces do a great job translating the data inquiry from SAS syntax into one or several queries that conform to the DBMS native SQL syntax. It does this intelligently in order to put DBMS-specific optimization techniques to good use. It also knows how to translate many SAS functions into database functions. After the transformation of the request, the database satisfies the queries and passes the results back to SAS. If any parts of the request cannot be pushed down for execution in the database, the necessary data would be transferred from the database to SAS for post-processing.

While the goal for both SAS/ACCESS and SAS In-Database Technologies is to delegate processing, and SAS In-Database Technologies relies on SAS/ACCESS for parts of its functionality, here are some conceptual differences:

- SAS/ACCESS uses SQL as the language of communication with the data platform. SAS In-Database Technologies uses SAS DS2.
- Communication between SAS/ACCESS and the database relies heavily on DBMS clients. SAS In-Database Technologies are dependent on DBMS execution infrastructure and the native database functions developed to collaborate with the SAS Embedded Process.
- The SAS/ACCESS output table can point to a variety of locations. The output from SAS In-Database Technologies should be stored in the database. It can be extracted afterward using SAS/ACCESS and be pointed once again into a variety of locations. However, the immediate output should go into the same database where the input data resides.
- SAS In-Database Technologies take the full block of DS2 code and execute it in database. There is no doubt what portions of a particular thread program are going to be processed in DBMS: the entire program will run. This is an important factor for several reasons. First, you can ensure that no data leaves the data platform, which might be critical for any governing policies in place. Second, you can avoid unexpected spikes in network traffic between SAS and database environments. Third, depending on the SAS In-Database Technologies product and the data

platform, you can operationalize the execution of SAS programs solely by the use of database management tools. (Note: With SAS In-Database Code Accelerator for Greenplum, only thread program runs in Pivotal Greenplum. The data block executes on the SAS tier.)

With that, we can review the specifics of processing flows of SAS In-Database Technologies products. You will see that despite the varying nature of tasks and some differences in invocation mechanisms and syntax, the processing boils down to the consistent integration principles and behavior on the database side.

The processing flow examples in this paper assume the requirements for using SAS In-Database Technologies are met so that all the described components can operate successfully.

SAS IN-DATABASE CODE ACCELERATOR

PROC DS2 programs can consist of two types of code blocks: a thread program and a data program. The thread program is used to declare the sections of the code that can execute in parallel. The data program defines what blocks of code to run and in which order as well as hosts the rest of the statements not designed for parallelism.

Figure 5 highlights the processing flow behind the request issued to SAS In-Database Code Accelerator for Hadoop and SAS In-Database Code Accelerator for Teradata.

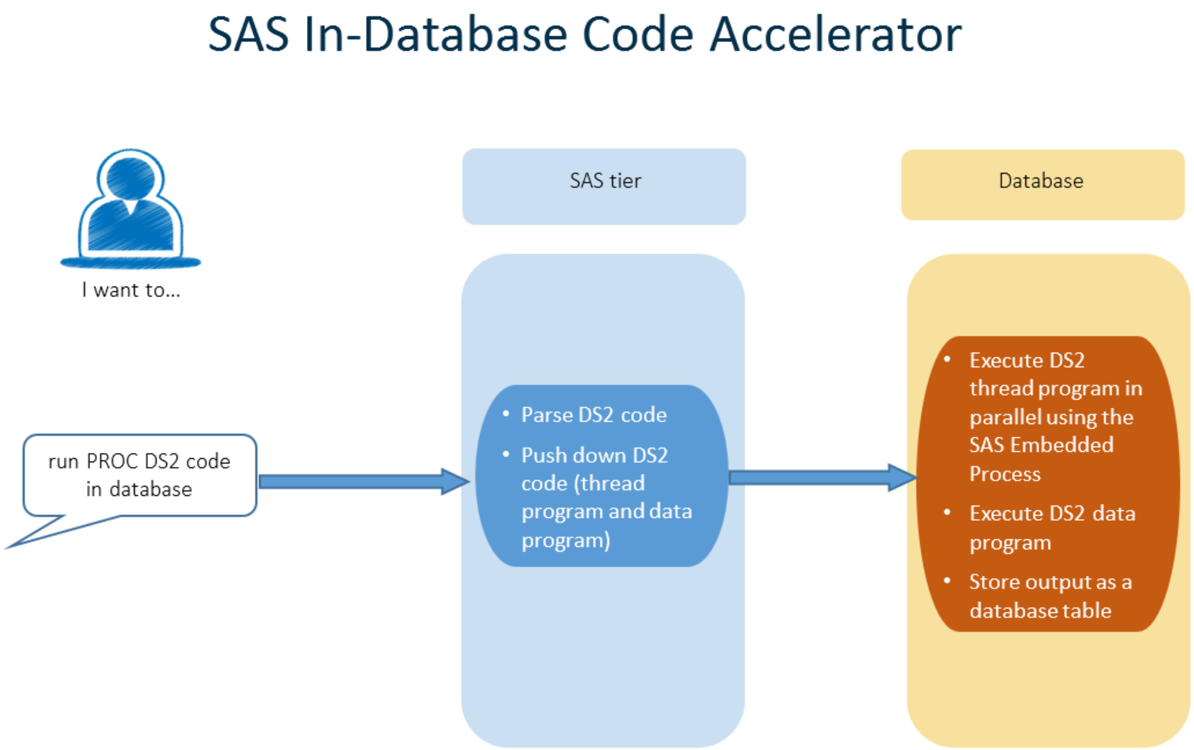

SAS In-Database Code Accelerator
I want to...
run PROC DS2 code in database
SAS tier
Database
• Parse DS2 code
• Push down DS2 code (thread program and data program)
• Execute DS2 thread program in parallel using the SAS Embedded Process
• Execute DS2 data program
• Store output as a database table

Figure 5. Processing Flow for SAS In-Database Code Accelerator for Hadoop and SAS In-Database Code Accelerator for Teradata

The SAS In-Database Code Accelerator processing flow includes these steps:

1. On the SAS tier, PROC DS2 program gets parsed.

   The DS2ACCEL=yes option in the PROC DS2 statement indicates to SAS In-Database Code Accelerator that this program is aimed to run in the corresponding database.

2. After this option is encountered, the whole DS2 program (both thread and data blocks) gets transferred to the database.
3. On the database side, the DS2 thread program gets executed in parallel across the database cluster with the help of the SAS Embedded Process.
4. For Hadoop and Teradata, the DS2 data program gets executed on the database side by a single thread taking temporary output from the execution of the thread program and generating a database output table. With SAS In-Database Code Accelerator for Greenplum, DS2 data program block executes on the SAS tier.

The specifics of how the SAS Embedded Process communicates with the database engine and orchestrates the parallel execution can differ depending on the database. As we stressed earlier, this is a highly integrated communication and is heavily dependent on the database internal methods of operation.

In the case of Teradata, for example, there typically are several Teradata AMPs (Access Module Processors) for each Teradata node. The SAS Embedded Process on each node establishes a communication channel between each DS2 thread and a corresponding Teradata AMP. This high-performance communication channel handles getting data to and from the processing thread. Each thread executes on its own subset of rows. See Figure 6 for the mechanics of the communication between the SAS Embedded Process and Teradata AMPs on each node of a Teradata cluster.

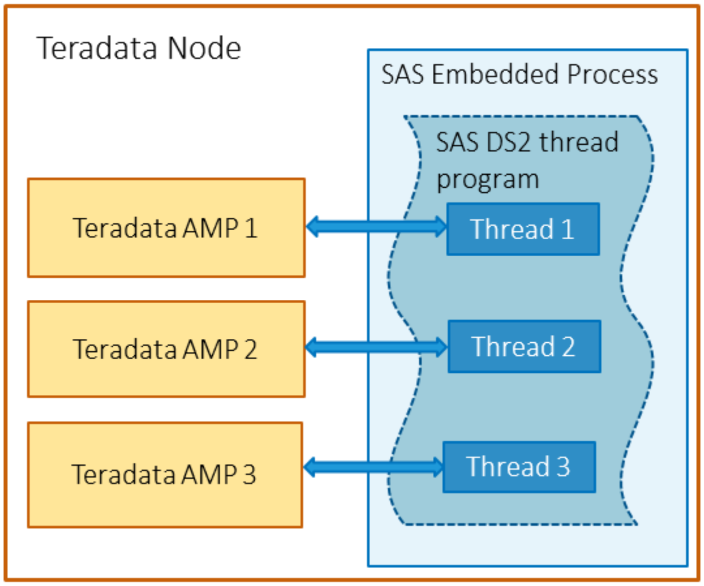


Figure 6. Mechanics of SAS DS2 Thread Program Execution by SAS Embedded Process in Teradata

A number of technical resources are available on the specifics of the SAS Embedded Process integration with each supported database. For the purposes of this paper, it is sufficient to realize that only the thread block of PROC DS2 truly benefits from the distributed execution. The data program runs in a single-threaded fashion. The best practices of PROC DS2 advise to design for parallelism as much as possible. But there is still a benefit of executing the non-parallel data block in database to avoid any data movement out of the database.

SAS SCORING ACCELERATOR

Several SAS analytics applications that focus on model development allow the export of the resulting SAS scoring model code.

SAS Scoring Accelerators allow deploying these models in production to your database environments.

Figure 7 highlights the processing flow behind the request issued to SAS Scoring Accelerator.

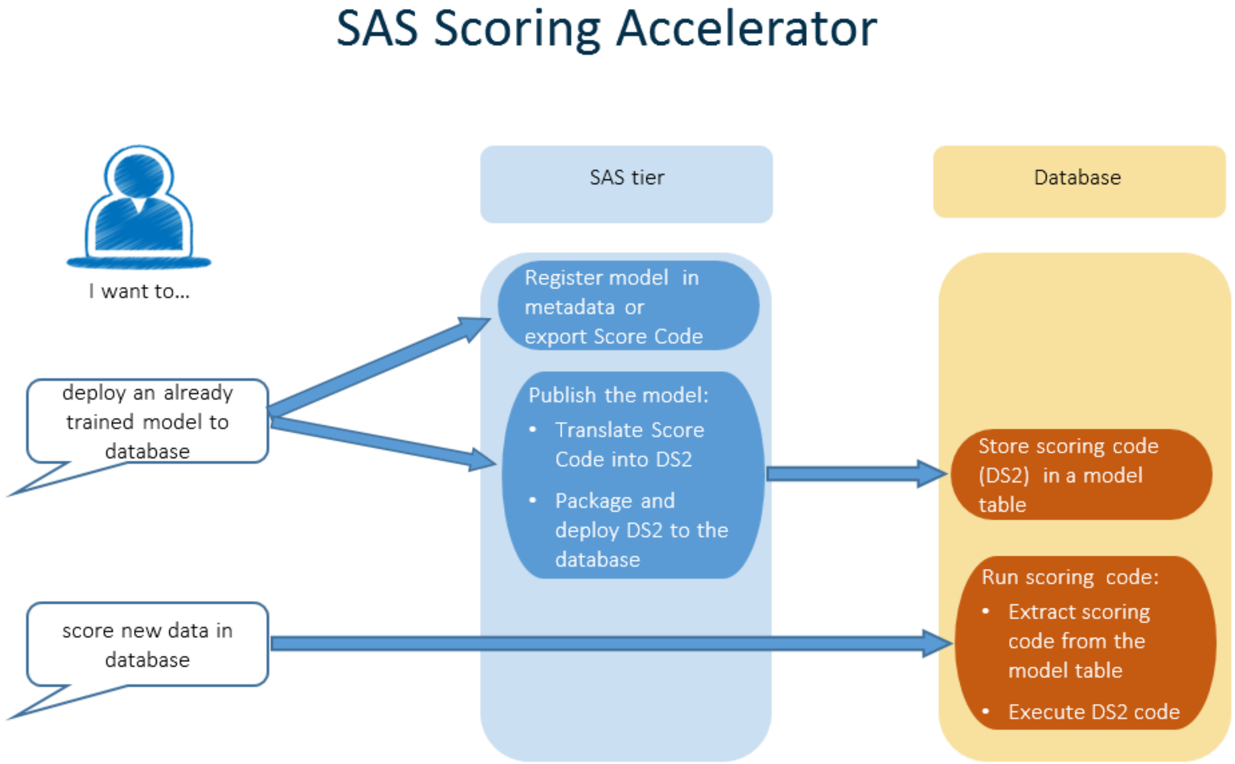

SAS Scoring Accelerator
I want to...
deploy an already trained model to database
score new data in database
SAS tier
Register model in metadata or export Score Code
Publish the model:
• Translate Score Code into DS2
• Package and deploy DS2 to the database
Database
Store scoring code (DS2) in a model table
Run scoring code:
• Extract scoring code from the model table
• Execute DS2 code

Figure 7. Processing Flow for SAS Scoring Accelerator

After the model scoring code is created, the processing flow by SAS Scoring Accelerator includes these steps:

1. Invoked via the SAS Scoring Accelerator publishing macro, SAS Scoring Accelerator inserts the model code into a pre-created database table for hosting SAS models. (In the case of Hadoop, models are stored as individual files.) There are two formats in which the model code can be supplied to SAS Scoring Accelerator. These formats are a SAS DATA Step or an Analytic Store (a newer packaging format to deliver larger DS2 objects). SAS Scoring Accelerator translates a SAS DATA Step into SAS DS2. In the case of an Analytic Store format, SAS Scoring Accelerator passes the object as is.

2. Depending on the platform, the method for invoking the scoring execution might be different. For the majority of databases, SAS Scoring Accelerator provides the running model macro, while for some, like Teradata, an SQL invocation is the way to pass the call to the corresponding stored procedure on the Teradata side. Regardless of the invocation method, the corresponding model code gets extracted from the model table and is executed fully in database with the help of the SAS Embedded Process. The execution of the DS2 code at this point is very similar to the corresponding step in the processing flow for SAS In-Database Code Accelerator. The scored output is stored in the database.

Note that the same processing flow is used for the programmatic models deployment as with SAS Model Manager, which provides a point-and-click toolset for models management and deployment.

SAS DATA QUALITY ACCELERATOR

As of the fourth maintenance release for SAS 9.4, SAS Data Quality Accelerators are available for two data platforms: Teradata and Hadoop. While they both use the SAS Embedded Process framework, the interaction vehicle is different.

For Teradata, a series of Teradata stored procedures are created to support SAS data quality purposes. These stored procedures can be called with PROC SQL in an explicit SQL pass-through mode of SAS/ACCESS® Interface to Teradata or by any product that supports the Teradata SQL dialect.

For Hadoop, SAS® Data Quality Accelerator is building on top of the SAS In-Database Code Accelerator approach with the use of PROC DS2 and specific libraries for data quality. Starting with the fourth maintenance release of SAS 9.4, these data quality libraries are included in the deployment package for the SAS Embedded Process.

For Hadoop and Teradata, both methods leverage SAS Quality Knowledge Base (QKB) packages that must be installed on a database in addition to the SAS Embedded Process. SAS QKBs provide a collection of data quality rules for data quality operations and are specific to the operation and locale.

Figure 8 represents the processing flow behind a typical data cleansing request issued to SAS Data Quality Accelerator for Teradata.

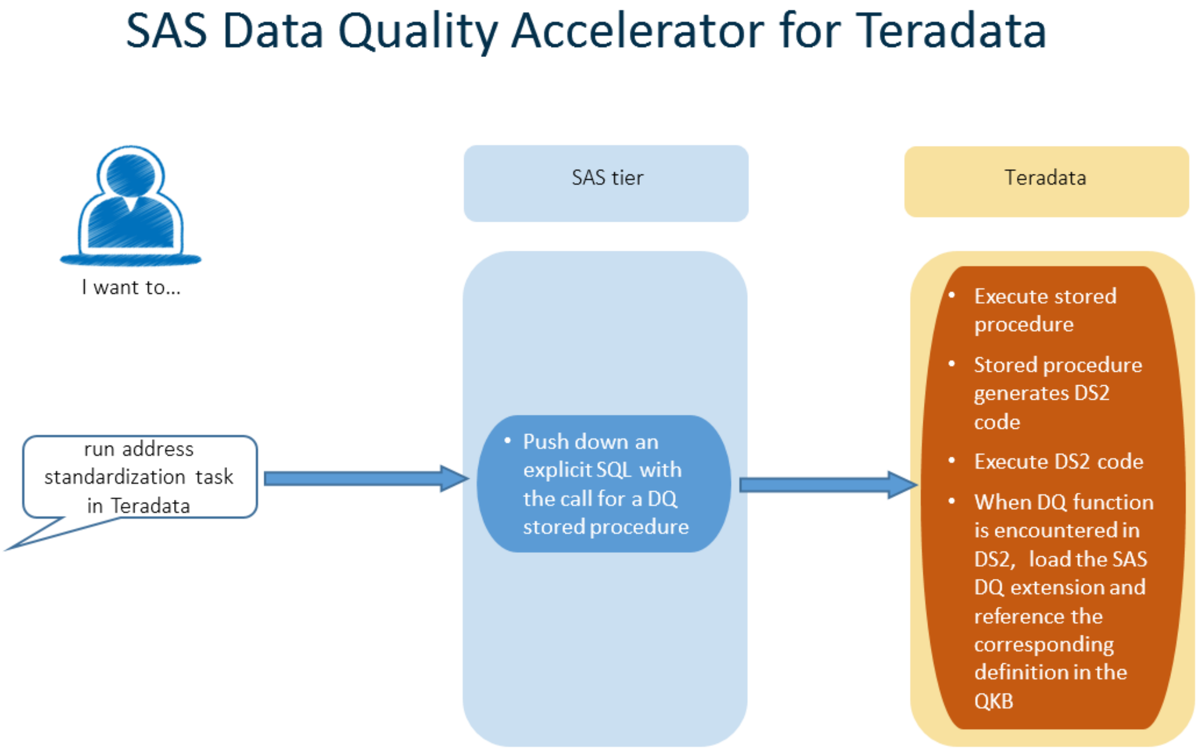

SAS Data Quality Accelerator for Teradata
I want to...
run address standardization task in Teradata
SAS tier
Push down an explicit SQL with the call for a DQ stored procedure
Teradata
Execute stored procedure
Stored procedure generates DS2 code
Execute DS2 code
When DQ function is encountered in DS2, load the SAS DQ extension and reference the corresponding definition in the QKB

Figure 8. Processing Flow for SAS Data Quality Accelerator for Teradata

The processing flow for SAS Data Quality Accelerator for Teradata shows these steps:

1. A call to the Teradata stored procedure is passed down to Teradata.

   Recall that a call to a stored procedure can be done outside of SAS session. Therefore, this step is optional.

2. On Teradata, the stored procedure is executed. The stored procedure generates a SAS DS2 program.

3. On Teradata, the DS2 thread program is executed in parallel across the database cluster with the help of the SAS Embedded Process. All data quality methods are implemented within the thread block of DS2 program.

4. When the SAS Embedded Process encounters DQ functions in a thread block, a call is made to the DQ library which, in turn, references the corresponding definition in SAS Quality Knowledge Base.

   Note: To supply the requested definitions, SAS Quality Knowledge Base must be available in memory. The first time a reference is made to the particular SAS Quality Knowledge Base, it gets loaded into memory. On Teradata, the SAS Quality Knowledge Base stays in memory for the duration of the SAS Embedded Process session. As a result, every new reference to the same SAS Quality Knowledge Base does not cause an overhead of QKB reload.

5. On the database side, the DS2 data program gets executed by a single thread taking output from the thread program execution and generating a database table.

Figure 9 represents a processing flow behind a typical data cleaning request issued to SAS Data Quality Accelerator for Hadoop.

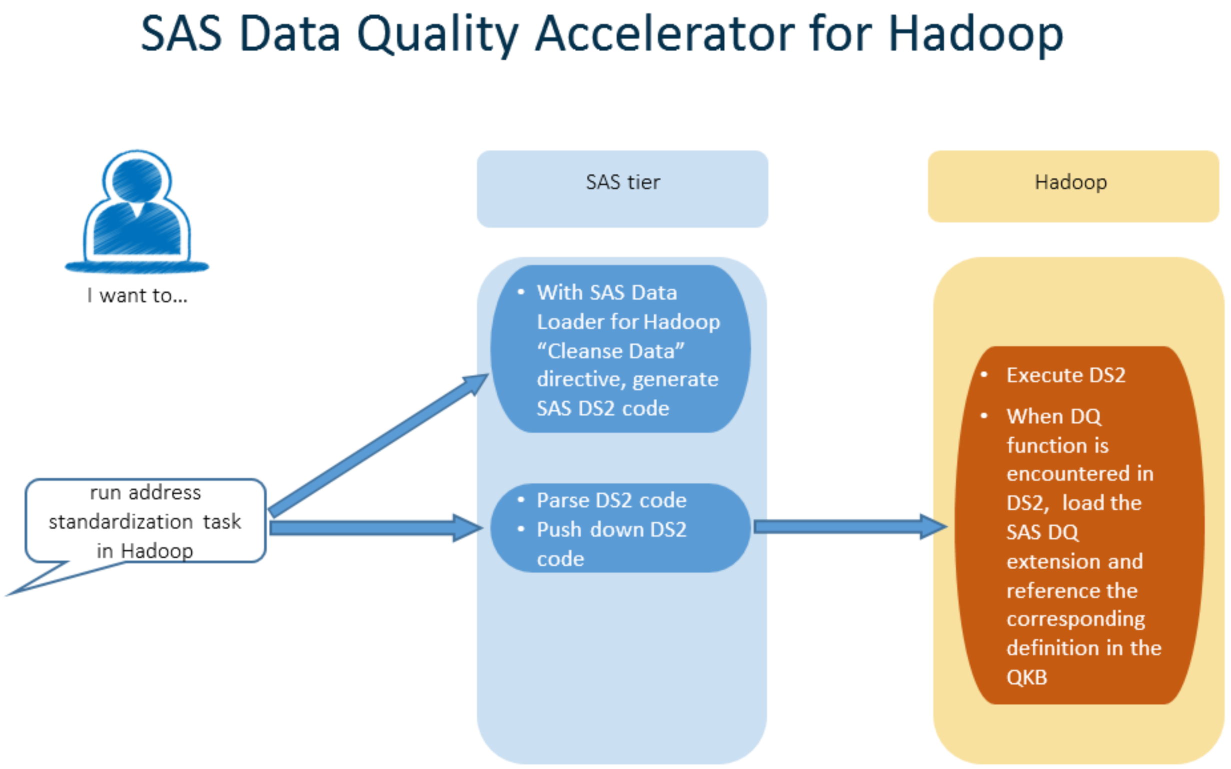

SAS Data Quality Accelerator for Hadoop
I want to…
run address standardization task in Hadoop
SAS tier
With SAS Data Loader for Hadoop "Cleanse Data" directive, generate SAS DS2 code
Parse DS2 code
Push down DS2 code
Hadoop
Execute DS2
When DQ function is encountered in DS2, load the SAS DQ extension and reference the corresponding definition in the QKB

Figure 9. Processing Flow for SAS Data Quality Accelerator for Hadoop

The processing flow for SAS Data Quality Accelerator for Hadoop includes these steps:

Initial step: Because SAS Data Quality Accelerator for Hadoop is currently driven by SAS Data Loader for Hadoop, the initial step is to engage with the Cleanse Data directive in SAS Data Loader for Hadoop. The result is a PROC DS2 program. The generated program is targeted for execution in Hadoop and already includes the necessary DS2ACCEL=yes option.

1. On the SAS tier side, the PROC DS2 program is parsed.
2. Once the DS2ACCEL=yes option is encountered, the whole DS2 program (both thread and data blocks) gets transferred to Hadoop.
3. On the Hadoop side, the DS2 thread program is executed in parallel across the Hadoop cluster with the help of the SAS Embedded Process. SAS Embedded Process in Hadoop runs as a MapReduce Application. All DQ methods are implemented within the thread block of the DS2 program and take advantage of MapReduce scalability.

4. When the SAS Embedded Process encounters DQ functions in a thread block, a call is made to the DQ library which, in turn, references the corresponding definition in the SAS Quality Knowledge Base.

   Note: The SAS Quality Knowledge Base must be in memory in order to provide the requested definitions. The first time a reference is made to the particular SAS Quality Knowledge Base, it gets loaded in memory and stays for the duration of the DS2 program execution. As a result, every new reference to the same SAS Quality Knowledge Base does not cause an overhead of its reload.

5. On the database side, the DS2 data program gets executed by a single thread taking output from the thread program execution and generating a database table.

SAS PROC TRANSPOSE PUSH DOWN

This procedure's push down is operated by an implicit translation of the PROC TRANSPOSE syntax into DS2 code. The code is then deployed and executed with the help of the SAS Embedded Process.

You might be familiar with a similar capability enabled by SAS/ACCESS interfaces to certain databases. A subset of SAS procedures (FREQ, MEANS, RANK, REPORT, SORT, SUMMARY, and TABULATE) can be pushed down to databases using the implicit translation of the procedure code into SQL.

The push down of PROC TRANSPOSE is based on a different paradigm. Utilization of SAS DS2 and the SAS Embedded Process framework opened the door for innovations in support for in-database execution of additional SAS procedures. PROC TRANSPOSE is the first one enabled using this method.

We will now review what processing steps allow transposition to take place in database. (See Figure 10.)

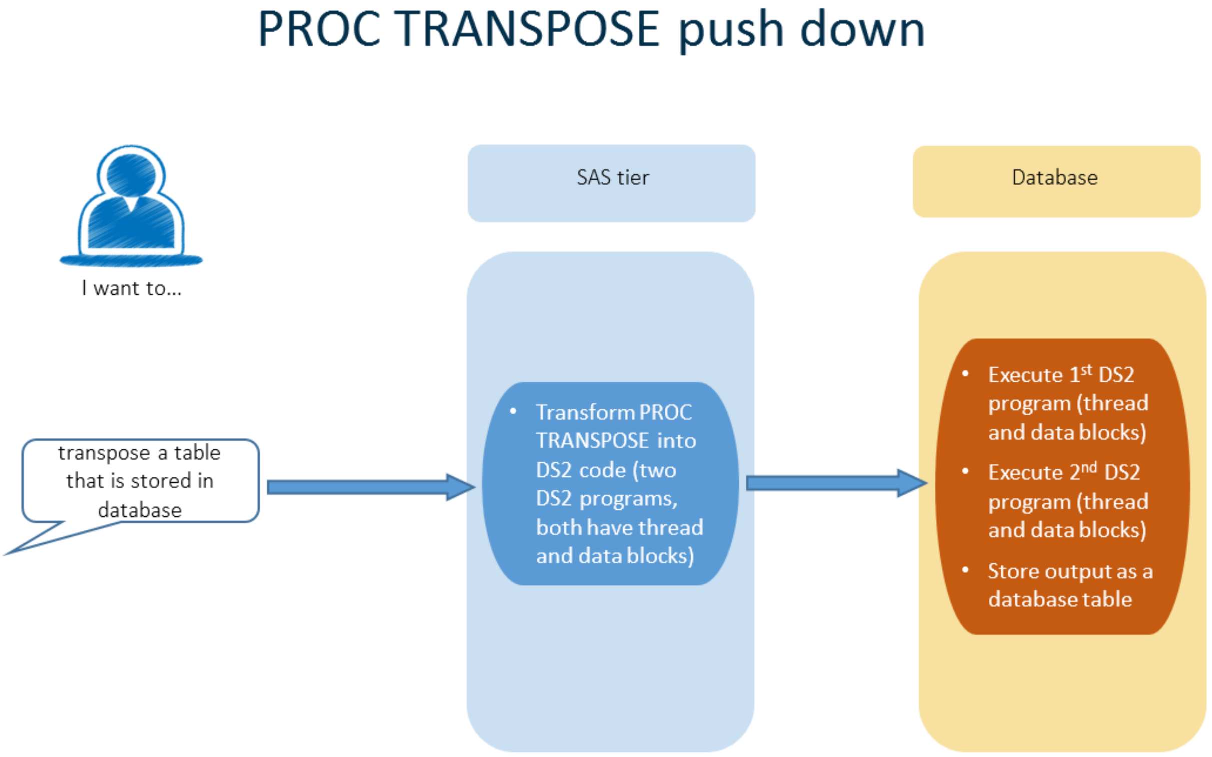

PROC TRANSPOSE push down
I want to...
transpose a table that is stored in database
SAS tier
• Transform PROC TRANSPOSE into DS2 code (two DS2 programs, both have thread and data blocks)
Database
• Execute 1st DS2 program (thread and data blocks)
• Execute 2nd DS2 program (thread and data blocks)
• Store output as a database table

Figure 10. Processing Flow for PROC TRANSPOSE Push Down

The processing flow for PROC TRANSPOSE push down includes these steps:

1. The PROC TRANSPOSE syntax is translated into two DS2 programs. The purpose of the first DS2 program is to define the metadata for the transposition output table. The purpose of

the second DS2 program is to perform the actual transposition. Both DS2 programs include the corresponding thread and data code blocks.

2. On the database side, the first DS2 program is executed The SAS Embedded Process helps run the thread block in parallel, followed by the single-threaded run of the data block. The generated metadata is written to the HASH table.
3. On the database side, the second DS2 program gets executed leveraging the HASH table produced by the first DS2 program. Note that the thread block for the second DS2 program holds all the data manipulation. The task of the data block is to invoke the thread. Therefore, the actual transposition is performed in parallel.

For more information about the mechanics of the push down of PROC TRANSPOSE, see “The Future of PROC TRANSPOSE: How SAS Is Rebuilding Its Foundation by Making What Is Old New Again” by S. Mebust.

CONCLUSION

SAS/ACCESS interfaces are essential in establishing an intelligent communication between SAS and your data sources. SAS In-Database Technologies are the next step in an efficient data integration story. If you made an investment in a powerful database or a sizeable data platform cluster, SAS In-Database Technologies can help you bring SAS into that powerful environment.

As SAS In-Database Technologies are integrated closely with the native framework of the database in a very specific and meaningful way to that database, inevitably there are differences in how these technologies operate. Luckily for SAS programmers, DS2 language is used as a consistent communication venue, and tackling the knowledge of DS2 significantly increases the application scenarios for what you can accomplish with these technologies. Alternatively, a number of point-and-click interfaces are provided by SAS products that are integrated with SAS In-Database Technologies. For example, SAS Model Manager allows scoring models operationalization in database without the need to interact with the code, and SAS Data Loader for Hadoop supplies a variety of directives for data cleansing and manipulation that generate and submit all the supporting DS2 code for you.

We encourage you to befriend SAS In-Database Technologies for your data preparation and analytics cycles, as together they provide an efficient and comprehensive coverage of functionality for bringing SAS to your data.

REFERENCES AND RECOMMENDED READING

Mebust, Scott. 2017. “The Future of PROC TRANSPOSE: How SAS Is Rebuilding Its Foundation by Making What Is Old New Again.” *Proceedings of the SAS Global 2017 Conference*.

Secosky, Jason, Ray, R, and Otto, G. 2014. “Parallel Data Preparation with the DS2 Programming Language.” *Proceedings of the SAS Global 2014 Conference*. Available at http://support.sas.com/resources/papers/proceedings14/SAS329-2014.pdf .

Sober, Steven. 2015. “Now That You Have Your Data in Hadoop, How Are You Staging Your Analytical Base Tables?.” *Proceedings of the SAS Global 2015 Conference*. Available at http://support.sas.com/resources/papers/proceedings15/SAS1866-2015.pdf

Ghazaleh, David. 2016. “Exploring SAS Embedded Process Technologies on Hadoop.” *Proceedings of the SAS Global 2016 Conference*. Available at http://support.sas.com/resources/papers/proceedings16/SAS5060-2016.pdf

Ray, Robert and Eason, W. 2016. “Data Analysis with User-Written DS2 Packages.” *Proceedings of the SAS Global 2016 Conference*. Available at http://support.sas.com/resources/papers/proceedings16/SAS6462-2016.pdf

Frost, Mike. 2014. “Washing the Elephant: Cleansing Big Data Without Getting Trampled.” *Proceedings of the SAS Global 2014 Conference*. Available at http://support.sas.com/resources/papers/proceedings14/SAS390-2014.pdf

Rineer, Brian. 2015. “Garbage In, Gourmet Out: How to Leverage the Power of the SAS Quality Knowledge Base” *Proceedings of the SAS Global 2015 Conference*. Available at http://support.sas.com/resources/papers/proceedings15/SAS1852-2015.pdf

Craver, Mark. 2015. “SAS Data Management: Technology Options for Ensuring a Quality Journey Through the Data Management Process.” *Proceedings of the SAS Global 2015 Conference*. Available at http://support.sas.com/resources/papers/proceedings15/SAS1907-2015.pdf

Petrova, Tatyana. 2015. “SAS and SAP Business Warehouse on SAP HANA – What’s in the Handshake?” *Proceedings of the SAS Global 2015 Conference*. Available at http://support.sas.com/resources/papers/proceedings15/SAS1856-2015.pdf

SAS® 9.4 In-Database Products: User’s Guide, Seventh Edition. Available at http://support.sas.com/documentation/cdl/en/indbug/69750/PDF/default/indbug.pdf

Jordan, Mark. 2016. *Mastering the SAS® DS2 Procedure: Advanced Data Wrangling Techniques*. Cary, NC: SAS Institute Inc.

Eberhardt, Peter. 2016. *The DS2 Procedure: SAS® Programming Methods at Work*. Cary, NC: SAS Institute Inc.

Svolba, Gerhard. 2006. *Data Preparation for Analytics Using SAS®*. Cary, NC: SAS Institute Inc.

Svolba, Gerhard. 2012. *Data Quality for Analytics Using SAS®*. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

The author extends sincere gratitude to everyone contributing to development, support, and enablement of SAS In-Database Technologies product line. In addition, the author would like to thank the following individuals who generously provided their time and consultation for this paper:

David Ghazaleh, SAS Institute Inc.

Brian Kinnebrew, SAS Institute Inc.

Scott Mebust, SAS Institute Inc.

Greg Otto, Teradata

Robert Ray, SAS Institute Inc.

Brian Rineer, SAS Institute Inc.

Paul Segal, Teradata

Steven Sober, SAS Institute Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Tatyana Petrova
100 SAS Campus Drive
Cary, NC 27539
SAS Institute Inc.
919-677-8000
Tatyana.Petrova@sas.com