Paper 077-2013

Best Practices in SAS® Data Management for Big Data

Nancy Rausch, SAS Corporation; Malcolm Alexander, SAS Corporation

ABSTRACT

This paper discusses capabilities and techniques for optimizing SAS® data management products for Big Data. It demonstrates how SAS supports Apache Hadoop technologies, including details on the various languages available in the platform and how to use them with SAS. An overview of SAS Data Management integration with the SAS® LASR™ platform is also provided. Practical tips and techniques are presented, including case studies that illustrate how to best use the SAS data management feature set with big data.

INTRODUCTION

Big data trends and related technologies are becoming important to organizations of all types and sizes. This paper introduces the most important technologies in this area, and discusses the ways in which SAS Data Management products support customers in their big data initiatives. We will start with an overview of big data as understood by the industry, discuss common technologies, and finally discuss how SAS helps customers leverage big data for data analysis and decision making.

THE BIG DATA CHALLENGE – BIG DATA IS RELATIVE

Many companies think of data as an organizational asset. As organizations mature in their decision-making processes, the collection, storage, and governance of data becomes more critical. In addition, technology advancements such as the proliferation of mobile devices and applications, along with social changes such as the growth in social media use, are creating new data sources that organizations may want to leverage.

Big data is a concept at the intersection of these trends. While the term is most often thought about in the context of the ever-increasing data volumes organizations deal with, there is more to the story. SAS defines big data as the point at which the volume, velocity, and variety of data exceeds an organization's storage or computation capacity for accurate and timely decision-making. Let's look at each important concept in turn.

VOLUME, VARIETY, AND VELOCITY OF DATA

Many factors contribute to the recent industry increase in data volume; examples include newer enterprise scenarios such as transaction-based data that accumulates with time, text data streaming in from social media, increasing amounts of sensor data being collected, and others. Excessive data volumes have created a storage issue because storage technologies cannot keep pace with data volumes. Storage costs are decreasing with time, but other issues are emerging, such as how to determine relevance amidst the large volumes of data and how to create value from data that is relevant.

Data today comes in many forms and with significant variety. Examples range from traditional databases to hierarchical data stores created by end-users and OLAP systems, to text documents, email, meter-collected data, video, audio, stock ticker data, financial transactions, and others. Recent industry estimates state that over 80 percent of an organization's data is not numeric and therefore is not easy to process with traditional technology. Yet, organizations still need and want to incorporate all of this data in analyses and decision- making.

According to the analyst firm Gartner®, velocity is an indicator of both how fast data is being produced and how fast the data must be processed to meet demand. Industries that are driving increasing data velocity such as Radio-Frequency identification (RFID) tags and smart metering are also driving an increasing need to deal with large volumes of data in near-real time. Reacting quickly enough to deal with velocity is one of the key data challenges when working with big data.

BIG DATA HAPPENS WHEN STORAGE AND COMPUTE DEMAND INCREASE

In traditional data storage environments, servers and computation resources are in place to process the data. However, even using today's traditional data storage mechanisms, there are data challenges that can stretch the capacity of storage systems. Tasks such as simulations and risk calculations, which work on relatively small amounts of data, can still generate computations that can take days to complete, which place them outside the expected decision-making window needed. Other business processes may require long-running ETL-style processes or significant data manipulation. When traditional data storage and computation technologies struggle to provide either the storage or computation power required to work with their data, an organization is said to have a big data issue.

ACCURATE AND TIMELY DECISION-MAKING

Ultimately the goal of most data processing tasks is to come to a business decision. An organization is deemed to have big data when any of the above factors, individually or in combination, make it difficult for an organization to make the business decisions needed to be competitive. While a large organization may have different big data issues compared to the big data concerns of a small firm, ultimately the problem comes down to the same set of challenges to solve.

We will now discuss the technologies that are being used to address big data challenges and how you can bring the power of SAS to help solve the challenges.

SAS AND BIG DATA TECHNOLOGIES

SAS is pursuing a number of complementary strategies for big data, enabling you to decide which approach is right for your enterprise. These strategies are:

- Leveraging emerging big data platforms (Apache Hadoop)
- Creating new technology for problems not well addressed by current big data platforms (SAS LASR and SAS® High Performance Analytics)
- Moving more computation to traditional databases (SAS® In-Database)
- Data virtualization (SAS® Federation Server)

Let's look at each of these in turn, and discuss how SAS Data Management makes dealing with big data easier.

APACHE HADOOP

The most significant new technology that has emerged for working with big data is Apache Hadoop. Hadoop is an open source set of technologies that provide a simple, distributed storage system paired with a parallel processing approach well suited to commodity hardware. Based on original Google and Yahoo innovations it has been verified to scale up to handle big data. Many large organizations have already incorporated Hadoop into their enterprise to process and analyze large volumes of data with commodity hardware using Hadoop. In addition, because it is an open and extensible framework, a large array of supporting tools are available that integrate with the Hadoop framework.

Hadoop Technology Overview

Table 1 describes some of the available Hadoop technologies and their purpose in the Hadoop infrastructure.

Hadoop Technology	Purpose
HDFS	Hadoop Distributed File System (HDFS) is a distributed, scalable, and portable file system written in Java for the Hadoop framework. Users load files to the file system using simple commands, and HDFS takes care of making multiple copies of data blocks and distributing those blocks over multiple nodes in the Hadoop system to enable parallel operation, redundancy, and failover.
MapReduce	The key programming and processing algorithm in Hadoop. The algorithm divides work into two key phases: Map and Reduce. Not all computation and analysis can be written effectively in the MapReduce approach, but for analysis that can be converted, highly parallel computation is possible. MapReduce programs are written in Java. All of the other languages available in Hadoop ultimately compile down to MapReduce programs.

Pig	PigLatin is a procedural programming language available for Hadoop. It provides a way to do ETL and basic analysis without having to write MapReduce programs. It is ideal for processes in which successive steps operate on data.
	
	Here is a PigLatin program example:
	
	`A = load 'passwd' using PigStorage(':');`
	`B = foreach A generate $0 as id;`
	`dump B;`
	`store B into 'id.out';`

Hive	Hive is another alternative language for Hadoop. Hive is a declarative language very similar to SQL. Hive incorporates HiveQL (Hive Query Language) for declaring source tables, target tables, joins, and other functions similar to SQL that are applied to a file or

	set of files available in HDFS. Most importantly, Hive allows structured files such as comma-delimited files, to be defined as tables that HiveQL can query. Hive programming is similar to database programming.
	
	Here is a Hive program example:
	
	`INSERT OVERWRITE TABLE pages`
	`SELECT redirect_table.page_id,`
	`redirect_table.redirect_title,`
	`redirect_table.true_title,`
	`redirect_table.page_latest,`
	`raw_daily_stats_table.total_pageviews,`
	`raw_daily_stats_table.monthly_trend`
	`FROM redirect_table`
	`JOIN raw_daily_stats_table ON`
	`(redirect_table.redirect_title =`
	`raw_daily_stats_table.redirect_title);`

Table 1. Hadoop Technologies

SAS and Hadoop Integration

Figure 1 indicates the integration of various components of SAS and Hadoop. The Hadoop technologies are indicated in grey, traditional SAS technologies indicated in light blue, and newer SAS technologies indicated in dark blue.

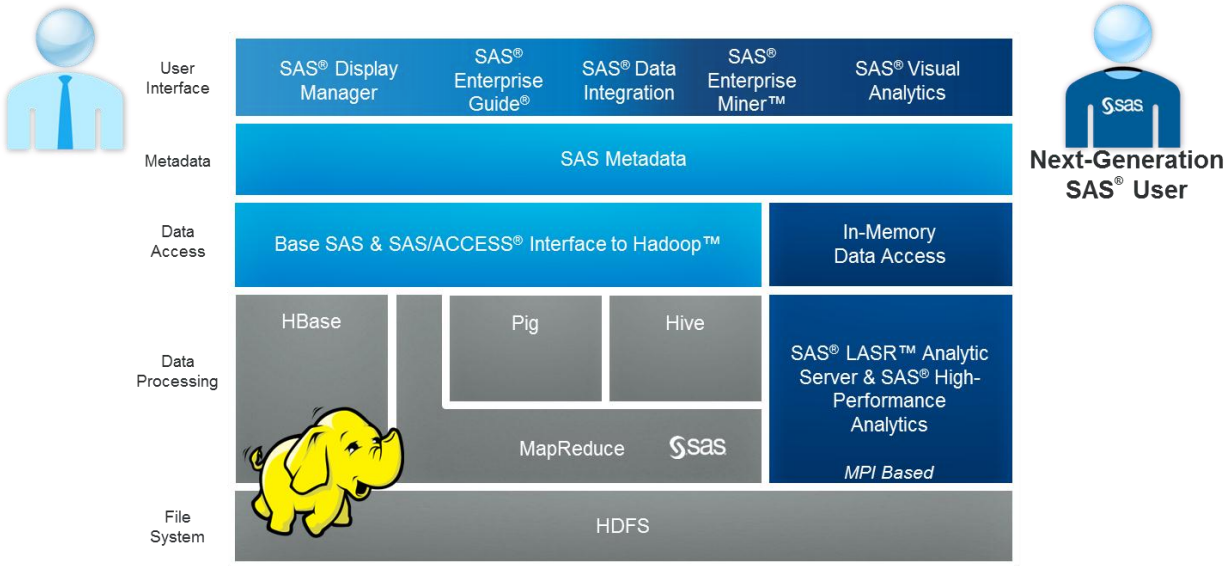

Figure 1. SAS and Hadoop Integration

SAS® Data Integration Studio enables organizations to leverage Hadoop in the following ways:

1. As a data environment, by using a SAS/ACCESS engine
2. As a file-based storage environment, by using SAS File I/O capabilities
3. As a computation environment, by using Hadoop transformations in SAS Data Integration Studio for Pig, HIVE, and MapReduce programming.
4. As a data preparation environment for SAS LASR server with conversion capabilities to LASR Hadoop storage

Accessing Data in HDFS Using SAS File I/O

SAS can access data stored in the HDFS in several ways. The first technique uses file input and output. The SAS file input/output capabilities have been enhanced to read and write directly to the HDFS. Using the SAS infile statement, the File Reader and File Writer transformations in Data Integration Studio can directly read and write HDFS files. Using SAS in combination with Hadoop also adds several unique capabilities that are not part of the Hadoop language itself. This helps you bring the power of SAS to your Hadoop programs. These capabilities are:

- The HDFS is a distributed file system, so components of any particular file may be separated into many pieces in the HDFS. Using SAS, you do not need to know details about the HDFS files or how they are distributed. SAS is able to interact with the distributed components of any file on the HDFS as if it were one consolidated file. You can work with HDFS distributed files like you would work with any other file coming from any other system.
- HDFS does not provide metadata about the structure of the data stored in the HDFS. Using SAS, you can apply SAS formats and automatically discover the structure of any data contained in the HDFS.

Access Data in HDFS Using SAS/ACCESS for HADOOP

The second technique available to interact with files stored in HDFS uses the new SAS/ACCESS for Hive engine. The new engine provides libname access to any data stored in Hadoop. It utilizes the SAS metadata server to provide additional control and manageability of resources in Hadoop.

The engine is designed to leverage the Hadoop Hive language. Using Hive, SAS can treat comma-separated or other structured files as tables which can be queried using SAS Data Integration Studio, and ETL can be built using these tables as any other data source. Figure 2 shows a Data Integration Studio Job performing a join using Hadoop Hive. Notice the indicator on the top right of the tables that show the tables to be Hadoop data.

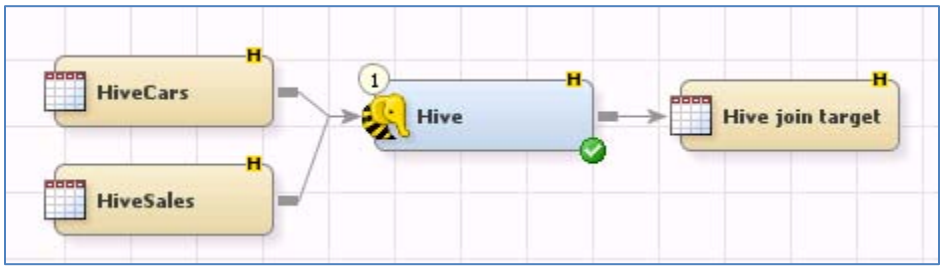

Figure 2. Example Data Integration Studio Job With Hive Transform

COMPUTATION IN HADOOP

SAS Data Integration Studio provides a series of transformations shown in Figure 3 which are useful for working with data in Hadoop.

Figure 3. Hadoop Transforms Available in Data Integration Studio

More details of these transforms are shown in Table 2.

Transform	Function
Hadoop Container	Convenience transformation allowing multiple Hadoop programs to be bundled into one transformation.
Hadoop File Writer	Move a structured file in the local system to a file in HDFS.
Hadoop File Reader	Move a file in HDFS to a structured file in the local system.
Pig	Choose from a set of available program templates in the Pig language that help you get started writing ETL programs in Pig, and/or write your own PigLatin program to manipulate and process data in Hadoop using the Pig language.
Hive	Choose from a set of available program templates in the Hive language that help you get started writing ETL programs in Hive, and/or write your own Hive code to query, subset, filter, or otherwise process Hadoop data using the Hive language.
Map Reduce	Choose a Java jar file containing MapReduce programs to be submitted to the Hadoop system.
Transfer From Hadoop	Transfer one or more files in the HDFS to the local system.
Transfer To Hadoop	Transfer one or more files on the local system to the Hadoop HDFS.

Table 2. Transform Details

The example job in Figure 4 shows a number of these transformations used in a Hadoop analysis:

1. Access HDFS data in Hadoop and run a Pig Job against that file.
2. Create a HIVE result table.
3. Filter the table with HIVE using custom code.
4. Transfer the results back to the local file system.

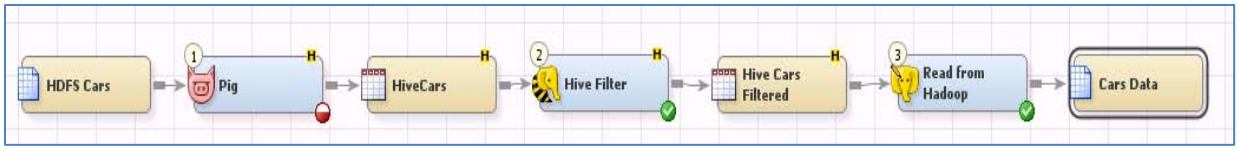

Figure 4. Example Job Showing SAS Data Integration Studio Hadoop Transforms

SAS DATA MANAGEMENT AND LASR

The SAS LASR server provides an in-memory, distributed computational system similar to Hadoop. SAS LASR is ideal for analytic algorithms for which the MapReduce paradigm is not well suited. As an in-memory server, you still need to feed data to LASR, and SAS Data Integration Studio simplifies this process. You register tables using the new SAS/ACCESS to LASR engine and then SAS Data Integration Studio can be used to perform a diverse set of tasks on LASR data, just like it would for other data sources. Figure 5 shows a LASR table being loaded with SAS Data Integration.

Figure 5. Example Job Showing SAS Data Integration Studio Integration With SAS LASR Tables

LASR does not support joins or pushdown optimization as other databases do therefore, if you have data that needs to be joined or modified, you need to perform the data manipulation prior to loading the data into LASR. You can do this in SAS using standard PROC SQL, or, if your data is already in Hadoop, you might want to directly perform the joins in Hadoop. You can create joins in Hadoop using one of the Data Integration Studio Hadoop transforms. There are examples and templates available to help you build your code, as shown in Figure 6. Once you have completed the data preparation stage in Hadoop, you can convert the Hadoop files or tables to LASR format using the Convert to LASR template available in the Pig transform. After the data has been converted to LASR format, it is ready to fast load directly into LASR.

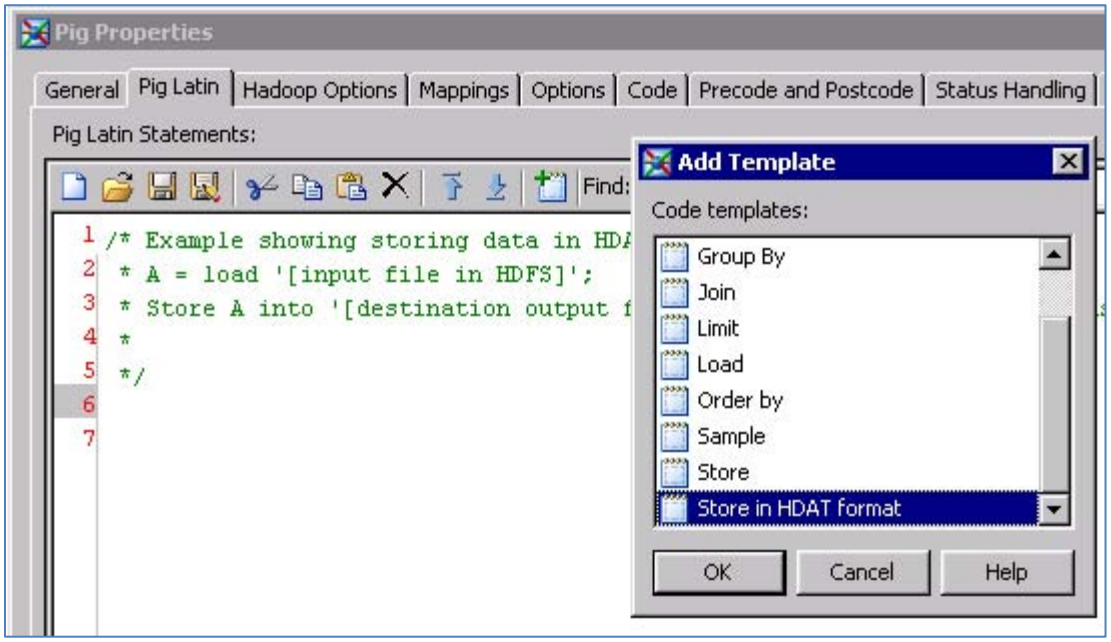

Figure 6. Example Code Templates in the Pig Transform

SAS IN-DATABASE DATA QUALITY

SAS, by means of the SAS/ACCESS technologies and accelerator products, has been optimized to push down computation to the data. By reducing data movement, processing times decrease and users are able to more

efficiently use compute resources and database systems. SAS/ACCESS engines already do implicit pass-through to push joins, where clauses, and even Base SAS® procedures such as SORT, TABULATE, and other operations down to databases. SAS® Scoring Accelerator and SAS® Analytics Accelerator provide additional capabilities by providing a SAS® Embedded Process that actually runs SAS code in a target database system, enabling orders of magnitude performance improvements in predictive model scoring and in the execution of some algorithms.

SAS has added the capability to push data quality capabilities into the database. The SAS® Data Quality Accelerator for Teradata enables the following data quality operations to be generated without moving data by using simple function calls:

- Parsing
- Extraction
- Pattern Analysis
- Identification Analysis
- Gender Analysis
- Standardization
- Casing
- Matching

Example performance improvements are indicated in the graph in Figure 7. Both scales are logarithmic, so we can see, for example, that data quality functions were performed on 50 million records in just over 2 minutes on a 16 node Teradata cluster, while a PC was only able to process approximately 200,000 records in the same amount of time. The graph shows that performance improvements scale linearly, which means that as you add more nodes and processing power, the performance of your in-database data quality programs continue to improve.

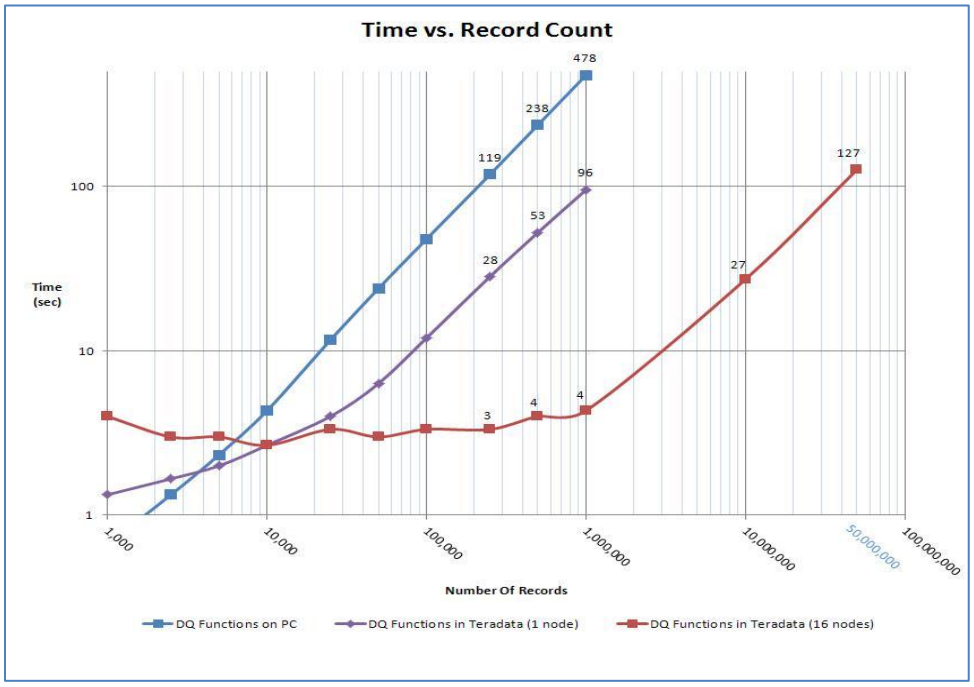

Figure 7. Performance Details Showing Performance Improvement Using In-database Data Quality

SAS will be adding these in-database data quality capabilities to additional database platforms over time.

DATA FEDERATION AND BIG DATA

Data federation is a data integration capability that allows a collection of data tables to be manipulated as if they were a single table while retaining their existing autonomy and integrity. It differs from traditional ETL/ELT methods because it pulls only the data needed out of the source system. Figure 8 is an illustration of the differences between traditional ETL/ELT and data federation.

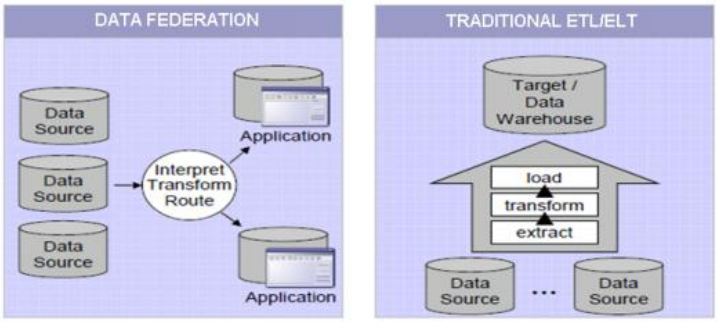

Figure 8. The Differences Between ETL and Data Federation

Data federation is ideally suited when working with big data because the data federation technique allows you to work with data stored directly in the source systems. Using data federation you only pull the subset of data that you need when you need it.

The SAS® Federation Server is the heart of the SAS data federation capabilities. Using the SAS Federation Server you can combine data from multiple sources, manage sensitive data through its many security features, and improve data manipulation performance through in-database optimizations and data caching. The server has fully threaded I/O, push-down optimization support, in-database caching, many security features including row-level security, an integrated scheduler for managing cache refresh, a number of native data access engines for database access, full support for SAS data sets, auditing and monitoring capabilities, and a number of other key features. Using the SAS Federation Server, you can gain centralized control of all your underlying data from multiple sources. Figure 9 is a high-level overview of the SAS Federation Server.

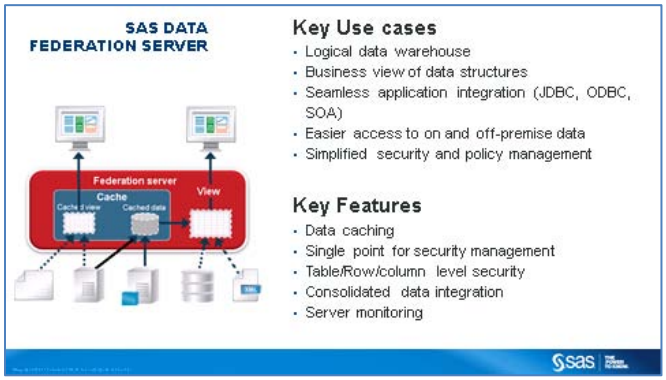

Figure 9. Overview of the SAS Data Federation Server Architecture

There are a number of big data scenarios which the SAS Federation Server is ideally suited. The following use cases illustrate some of these scenarios.

SAS Federation Server Use Case 1: Data is Too Sensitive

In this scenario, illustrated in Figure 10, data is owned by organizations that do not want to grant direct access to their tables. The data may be owned by organizations that charge for each access, or is deemed mission critical. Users are not allowed to go directly against the actual tables.

SAS Federation Server provides an ideal answer to this problem, because it funnels the data access through the federation server itself, so that multiple users do not have or need access to the base tables. A data cache can be optionally inserted into the result stream, so that even if the underlying tables are not accessible, for example if the source system is down, data can still be supplied to end-users. Also, the federation server provides a single point for managing all security so that users do not have to have to be granted direct access to the underlying tables.

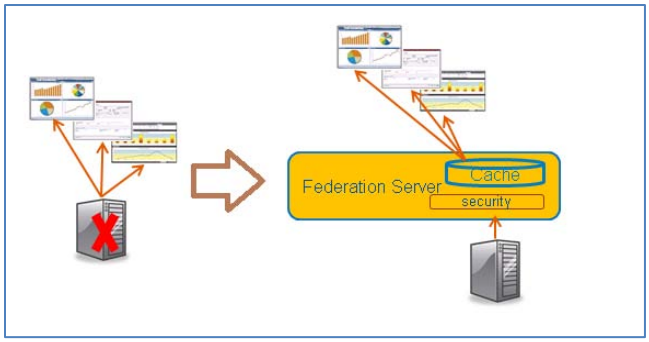

Figure 10. Example Federation Scenario When Data is Too Sensitive

Data Federation Use Case 2: Data is Too Diverse

In this use case, illustrated in Figure 11, the data is stored in multiple source systems which all have different security models, duplicate users, and different permissions. This makes it hard to control permissions in a consistent way and requires every application to be customized to handle every data source. If there are changes needed, for example if a user has to be removed or added to a new system, each application must be updated to handle the change. In a large system with a lot of data, this can become increasingly difficult to manage, especially over time.

SAS Federation Server solves this problem. All of the data access security can be managed singly in the SAS Federation Server, so multiple users do not go through to the base tables and security and permissions are managed in a single place for all target applications. In addition, by leveraging the optional data cache, you can provide access to data for multiple endusers without having to recalculate the result sets every time for each individual user. This adds to system efficiency.

Figure 11. Example Federation Scenario When Data is Too Diverse

Data Federation Use Case 3: Data is Too Ad Hoc

When data is changing frequently, constant updates are needed to maintain integration logic. It becomes difficult to make a repeatable integration process, especially if there are many data integration applications that need access to the same data. The application logic for accessing the data must be distributed into each application, and any changes in the data require corresponding changes in every application. As data grows in volume and complexity, and the number of applications that need to have access to the data grows, it becomes increasingly difficult to manage all of the distributed data access logic in all of the various applications, especially if the data is frequently changing.

SAS Federation Server solves this use case well. Using SAS Federation Server, it is easy to insert new views or modify existing views to accommodate different applications and users without requiring changes to the underlying data sources or to applications. SAS Federation Server provides a single point of control for all integration logic. Plus, since the data access is managed through the SAS Federation Server, data integration as well as security and permissions are managed in a single place for all target applications. Figure 12 is an illustration of this use case.

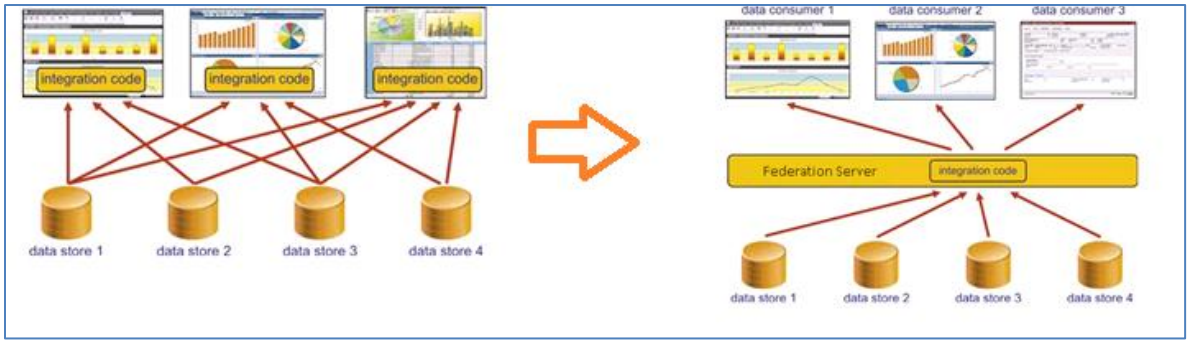

Figure 12. Example Federation Scenario When Data is Too Ad Hoc

SAS Federation Server supports access by using ODBC, JDBC, or through the SAS/ACCESS to Federation Server libname engine. A server management web client is available for administering and monitoring the server. From this interface you can monitor multiple serves from the web client interface. Figure 13 is an example of the manager web client.

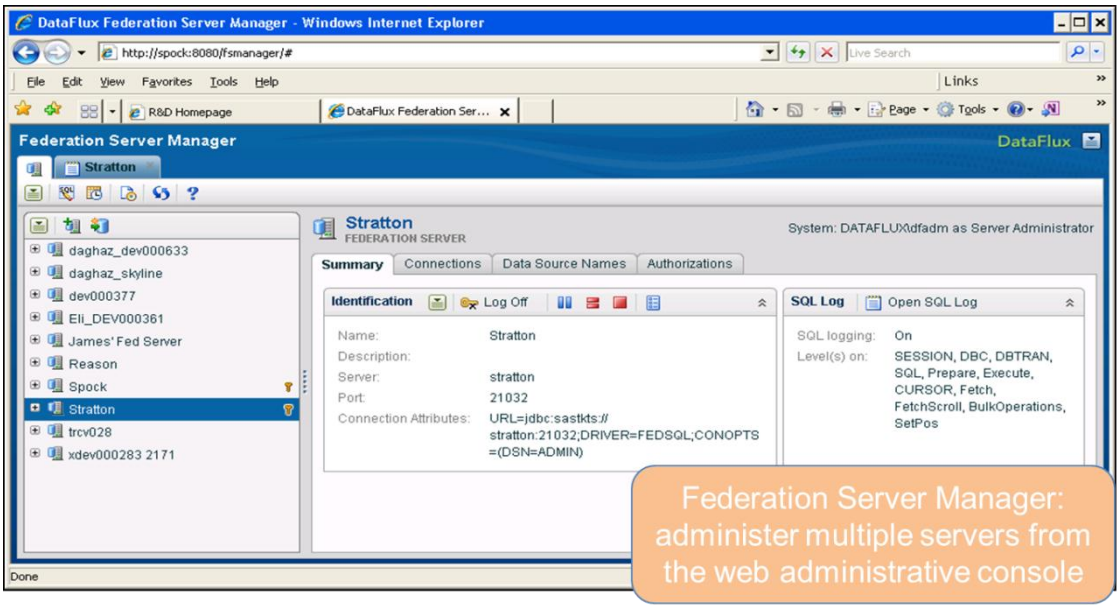

Figure 13. Federation Server Manager Overview

Cache management is available for the SAS Federation Server by using the Federation Server Manager web client. You can create cached queries, and you can target them to a database target that you choose or to in-memory storage. This helps with data performance because data joins and the results cache can reside completely in the database or directly in memory. SAS is unique in its ability to store its results cache directly in any database and in its ability to maximize performance by using pushdown optimizations.

SAS Federation Server supports multiple database languages, including an optional database generic language called TKSQL that you can use. The TKSQL language is based on ANSI standards and supports most SQL constructs of most of the most popular database languages. If you choose to use the TKSQL language, you can write SQL that can be run on any database that the SAS Federation Server supports. Figure 14 shows an example of queries in the SAS Federation Server.

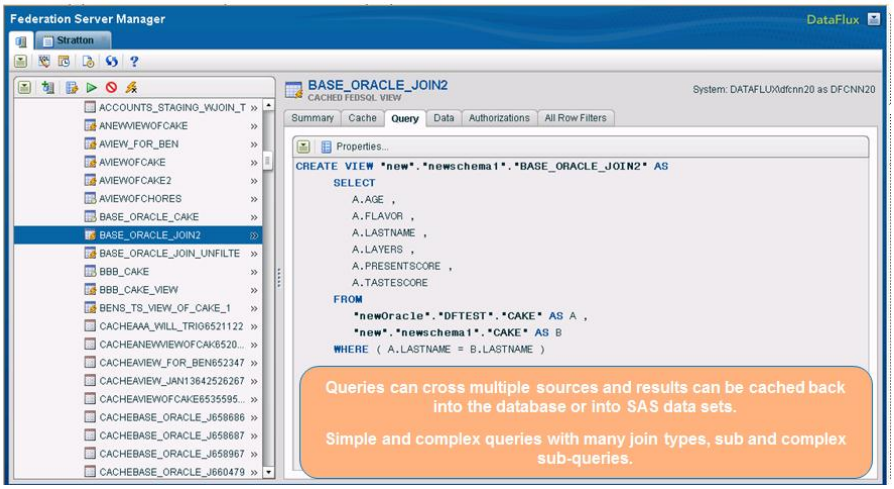

Figure 14. Query Building in SAS Federation Server Manager

The data cache also has an integrated scheduler so that you can schedule cache refreshes. There are many date and time based options, and the integrated scheduler supports custom programming using the CRON language. Figure 15 is an example of the integrated scheduler. You can also choose to schedule cache refreshes using your own scheduler, if you prefer.

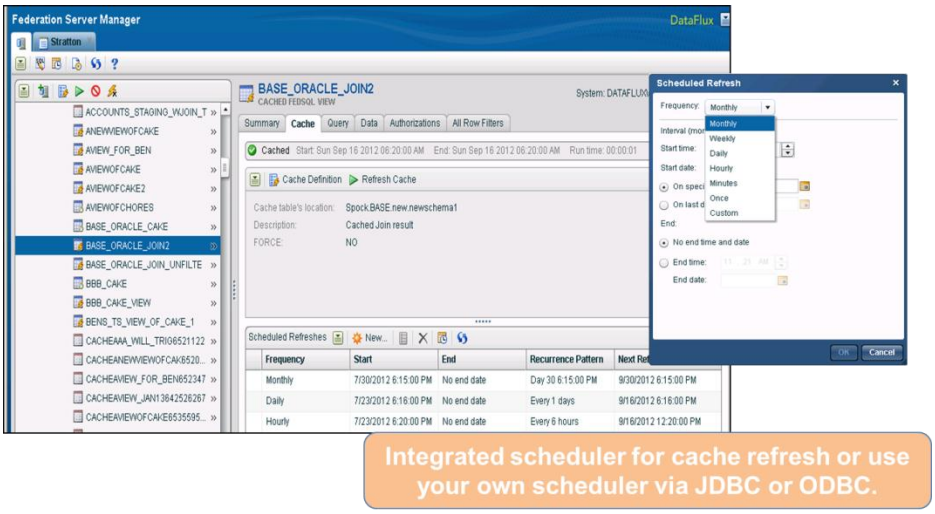

Figure 15. Integrated Scheduler

Many security features are supported in the SAS Federation Server. Permissions can be granted or denied on many levels including Database, Catalog, Schema, Table, and Column. Row level security is also supported for all databases including SAS data sets. Row level permissions can be granted by using where clauses defined in the view, and can include complex SQL constructs such as subqueries and joins. Row level permissions can be granted across multiple data sources that have been federated in the SAS Federation Server. For example, you can create a view that pulls data from three different tables coming from three different databases, and then apply row level permissions to the resulting target. Permissions supported include select, update, reference, insert, create, and others. Figure 16 is an example of some of the security features available in the SAS Federation Server.

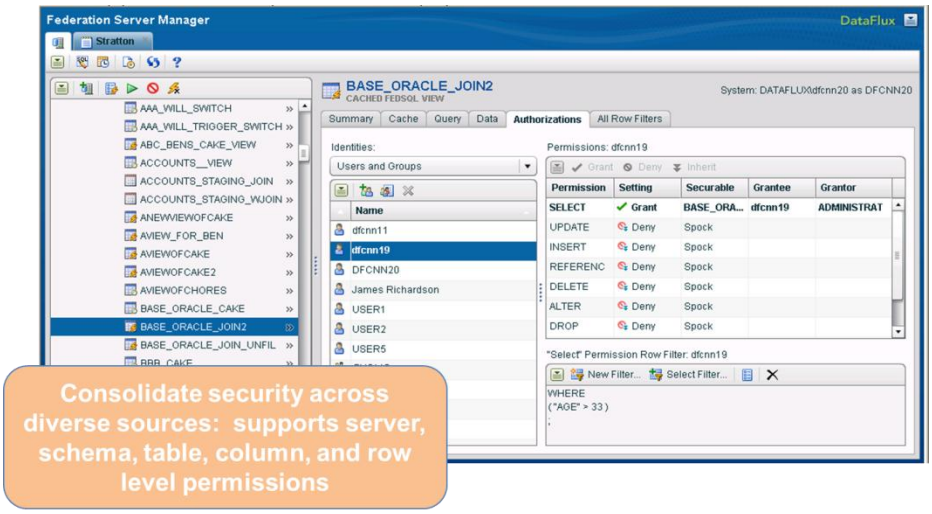

Figure 16. Security Example

There are also many server and query monitoring capabilities that let you track server and data access. Using the monitoring features, you can identify performance bottlenecks and view system usage. You can spot queries that are used often which will help you identify patterns that might be good to persist in the data cache. You can also identify which user is accessing which resources and monitor the SQL coming through the server. There are many logging levels available to help you track detailed data access for the purposes of auditing or for regulation compliance. You can also pinpoint exactly how the SQL is performing in the server in each step as it moves through the server, and track row inserts, updates, and deletions for every table that is accessed through the SAS Federation Server. Figure 17 is an example of some of the monitoring, logging, and auditing capabilities available in the Federation Server.

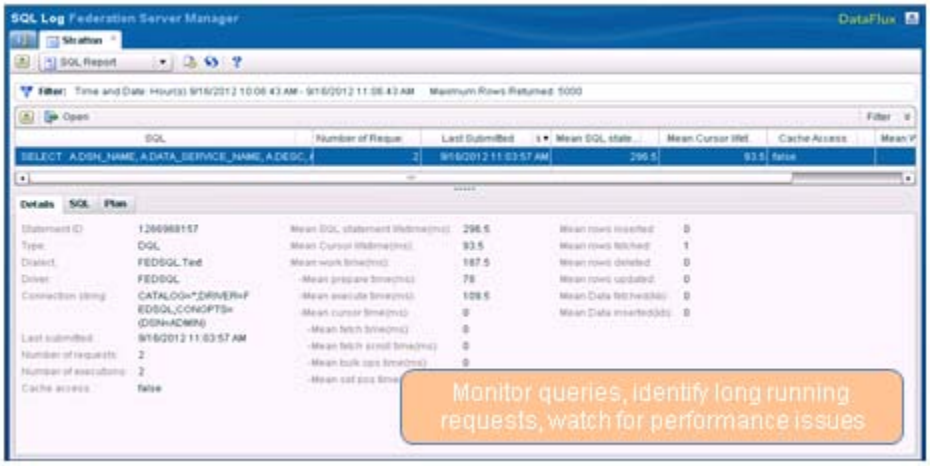

Figure 17. Monitoring Example

CONCLUSION

SAS is following multiple strategies in helping organizations effectively leverage big data, and this paper introduces a number of those strategies. SAS helps users access data in big data systems and use that data just as it would in any other data source. SAS also moves beyond the capabilities found in traditional data storage systems and even today's big data systems by providing enhanced capabilities, such as in-memory processing, in-database data quality, and data federation.

RECOMMENDED READING

- SAS® Enterprise Data Management and Integration Discussion Forum. Available at http://communities.sas.com/community/sas_enterprise_data_management_integration
- Alexander, Malcolm and Nancy Rausch. 2013. "What's New in SAS® Data Management". Proceedings of the SAS Global Forum 2013 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings13/070-2013.pdf.
- Rausch, Nancy, et al, 2012. "What's New in SAS® Data Management". Proceedings of the SAS Global Forum 2012 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings12/110-2012.pdf.
- Rausch, Nancy and Tim Stearn. 2011. "Best Practices in Data Integration: Advanced Data Management". Proceedings of the SAS Global Forum 2011 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/137-2011.pdf.
- Hazejager, Wilbram and Pat Herbert. 2011. "Innovations in Data Management – Introduction to Data Management Platform". Proceedings of the SAS Global Forum 2011 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/137-2011.pdf.
- Hazejager, Wilbram and Pat Herbert. 2011. "Master Data Management, the Third Leg of the Data Management Stool: a.k.a. the DataFlux® qMDM Solution". Proceedings of the SAS Global Forum 2011 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/146-2011.pdf.

CONTACT INFORMATION

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

Nancy Rausch
SAS Institute Inc.
Cary, NC 27513
Work Phone: (919) 677-8000
Fax: (919) 677-4444
E-mail: Nancy.Rausch@sas.com
Web: support.sas.com

Malcolm Alexander
SAS Institute Inc.
Cary, NC 27513
Work Phone: (919) 677-8000
Fax: (919) 677-4444
E-mail: Malcolm.Alexander@sas.com
Web: support.sas.com