Paper SAS390-2014

Washing the Elephant: Cleansing Big data Without Getting Trampled

Mike Frost, SAS Institute;

ABSTRACT

Data quality is at the very heart of accurate, relevant, and trusted information, but traditional techniques that require the data to be moved, cleansed, and repopulated simply can't scale up to cover the ultra-jumbo nature of big data environments. This paper describes how SAS® Data Quality Accelerators for databases like Teradata and Hadoop deliver data quality for big data by operating in situ and in parallel on each of the nodes of these clustered environments. The paper shows how data quality operations can be easily modified to leverage these technologies. It examines the results of performance benchmarks that show how in-database operations can scale to meet the demands of any use case, no matter how big a big data mammoth you have.

THE EVOLUTION OF DATA QUALITY

As with many technology trends, data quality started out as an unsolved problem that needed a solution before it became a fundamental element of what is now modern data management practices. As the need for a way to combine data from various data sources around an organization's IT infrastructure emerged, the data warehouse came into being and began to be employed. *"Aggregate all of data into one place,"* came the advice, usually coming from data warehouse vendors themselves. *"Once it's there, you can store it, report on it, or analyze it and everything will be great."* What organizations quickly began to realize, however, is that while resolving data formatting issues within a data warehouse could be solved by data integration technologies such as Extract, Transform, and Load (ETL) tools, issues like nonstandard data formats, parsed vs. unparsed information, and resolving duplicate information were persisting, which required a new technology.

EARLY DATA QUALITY TOOLS TAKE A SIMPLE APPROACH

What emerged was the first set of data quality tool, components which were designed to address data quality issues within the client-server based architectures that was the predominant processing environment at the time. In these environments, server platform resources that managed data warehouses were at a premium because they required server hardware that was significantly more specialized and therefore expensive to acquire, support, and maintain. As a result, data quality tools emerged that were deployed and ran on client platforms or more modest server platforms that were located across the network from the data warehouse where the source of the data needing the cleansing was located.

To clean the data, the data quality tool would establish a connection with the data warehouse and execute a query that would extract the data set to be cleansed and send it across the network to the data quality tool where it would perform whatever transformations were necessary to clean the data. The new, cleansed results would then be typically written back to the data warehouse for use. This process of extracting, cleansing, and publication of cleansed results was so common that it has become entrenched as the defacto standard for how virtually all data quality processing is handled today. While this approach is adequate for many use cases, it does not hold up in all circumstances. In particular, as big data architectures began to emerge, the limitations of the traditional approach to data quality processing began to become obvious.

THE RISE OF BIG DATA REQUIRES MORE COMPLEX TOOLS

Big data is a solution to the problem of how to store, manage, and efficiently process massive amounts of data. Architecture details among different big data implementations vary, but most leverage a distributed, parallel processing architecture that can easily be expanded to scale up and meet increased capacity demands. By combining improvements and advancements in hardware and software with a better design for handling very large amounts of data, many of the promises made about big data are now starting to sound a lot like what was said about data warehouses: *"Get everything, any kind of data, even data that you don't yet think you will need, into a big data environment. Once it's there, you can store it, report on it, or analyze it and everything will be great."* However, just like with data warehouses, the sheer size and scope of data managed within typical big data environments are creating a data quality problem that requires new technology to solve.

THE BIG DATA STAMPEDE

The difference in orders of magnitude between the size and scale of a big data environment as compared with a data warehouse can be stark. One large financial services company has reported having a single table that consisted of over 60 billion records that they would like to perform data quality processing against multiple times a day. While large tables of this

size aren't rare, frequently running a data quality process against tables of this size is.

BIG DATA MEANS BIGGER DATA QUALITY PROBLEMS

Simply put, traditional data quality processing is too slow to be practical to apply to big data environments. In some cases, organizations find that overnight batch data quality processes that took hours to complete for their data warehouse might take multiple weeks to complete when run against big data environments if they complete at all. Organizations take different approaches to working around this limitation including:

- Performing data quality on a fraction of the overall data set
- Scheduling data quality processing as infrequently as possible – instead of running overnight, scheduling processing on a monthly, quarterly, or yearly basis.
- Not performing data quality processing at all

The last choice is perhaps the most problematic for organizations, because it creates a problem that is difficult to overcome once it has become established: a lack of trust in the data. No matter what size of data is used, data quality problems eventually lead to an lack of faith in the decisions being made that are based on analysis of the data. For this reason, organizations are coming to grips with the fundamental question of how to make sure the data their big data environments is accurate and can be trusted without negatively impacting to fundamental business processes or compromising on the scale of the data managed by their environment.

USING BIG DATA TO CLEAN ITSELF

The solution to the problem lies in taking a new approach to data quality processing rather than one that depends upon the traditional approach of extracting, cleansing, and republishing the results. For big data, the most logical approach to take is one that leverages one of the key strengths of the big data environment itself – the distributed processing architecture that can execute work in parallel and that can scale directly with the size of the data.

One way organizations try accomplish this is to implement data quality algorithms that uses the native processing syntax or logic of the big data environment. Organizations that investigate this approach quickly discover that this approach is not practical because big data programmers lack the kind of knowledge needed to write good, flexible data quality algorithms. In addition, if a change is needed to an algorithm, code must be changed, or multiple versions of the algorithm must be coded and maintained to account for variations in how data quality must be applied for different use cases. These limitations quickly spiral out of control, creating a problem that is almost as bad as the problems that this approach was designed to solve.

HOW SAS TAMES BIG DATA QUALITY

SAS has led the way in providing the industry's best data quality solution for many years through products such as Data Quality Server, dfPower, and Data Management Platform, developed and sold through its former DataFlux subsidiary. With the absorption of DataFlux products and expertise into the SAS Data Management Division, these industry-leading data quality capabilities can more easily leverage the power of current and emerging SAS technologies. One such example of SAS technology that can augment SAS data quality capabilities is called the SAS Embedded Process.

TRAINING BIG DATA TO RUN SAS DATA QUALITY

The SAS Embedded Process deploys within a big data platform and runs on all of the nodes of the architecture. It acts as a kind of shell environment for running SAS code, specifically, DS2. When DS2 code is run within the Embedded Process, the Embedded Process manages and distributes the workload across all of the nodes of the big data platform, allowing the code to be executed in a parallel fashion. In this way, SAS offers an ability for other SAS technologies to leverage the power and scalability of a big data platform.

Delivering scalable SAS data quality capabilities to big data environments means developing DS2-based implementations of SAS data quality functions that can run within the Embedded Process, which is exactly what the SAS® Data Quality Accelerator is. This new product consists of an implementation of SAS Data Quality functions in DS2 code that deploys within the Embedded Process. When licensed and configured, these functions can be invoked via interfaces such as user-defined functions or stored procedures. Calling these functions causes the DS2 code that corresponds to SAS Data Quality to load and use the SAS Quality Knowledge Base across all of the nodes of a big data environment to perform actions such as standardization of names, parsing of addresses, or entity extraction from text fields.

Here is a step-by-step breakdown of how the functions work within the Teradata platform:

1. A user connects to Teradata and issues call to special stored procedure defined on Teradata that corresponds to a SAS data quality function.
2. Teradata interprets the stored procedure call and passes the function call and related parameters to the SAS Embedded Process.
3. The Embedded Process loads the appropriate locale of the SAS® Quality Knowledge Base and invokes the data quality DS2 code that corresponds to the data quality function called across all nodes using the associated parameters provided as part of the stored procedure call.
4. Teradata flows data to the Embedded Process for processing. The Embedded Process processes data on each node in Teradata that has rows.
5. Results are passed by the Embedded Process to Teradata for persisting to a table appropriate to the parameters passed in by the user.

In the case of the Teradata platform, invoking the SAS® Data Quality Accelerator is possible through any method by which a user or application can calls a stored procedure. In some cases, users may wish to call the procedures using the native tools and utilities of the Teradata platform itself, either directly via command-line interface or via a script, but SAS customers will likely wish to call them via data quality or data integration jobs.

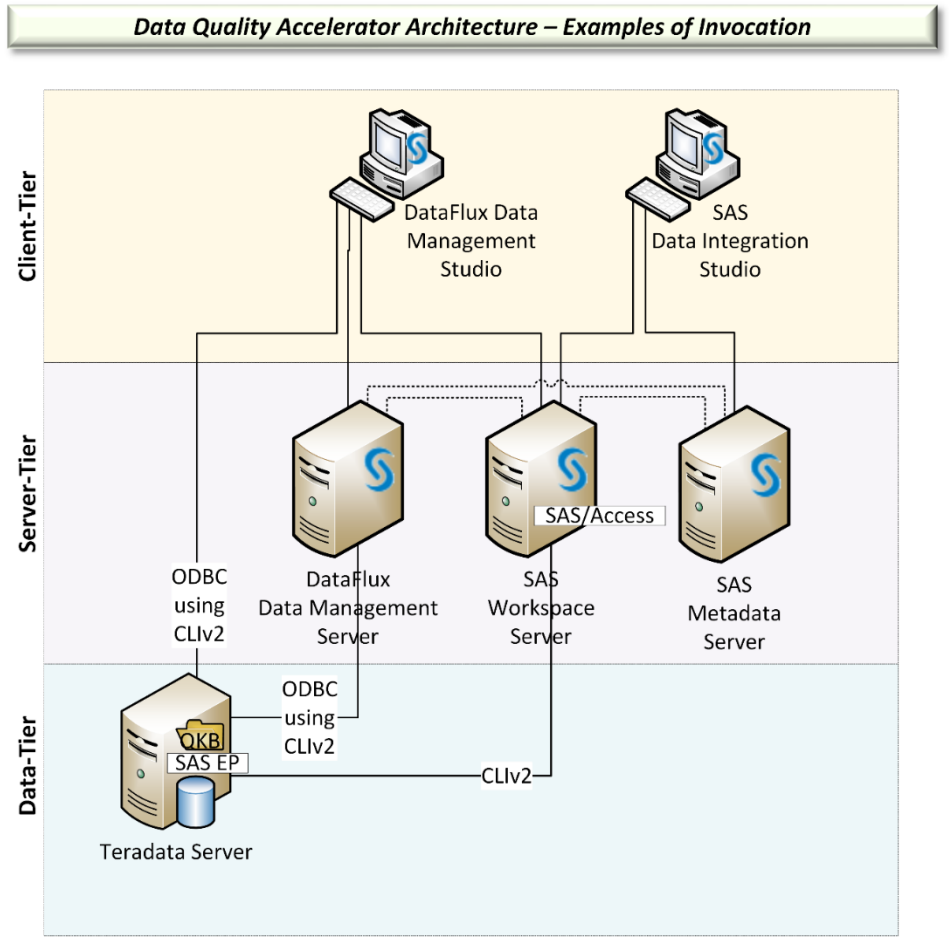

Figure 1. Invoking the Data Quality Accelerator for Teradata from SAS ® products

The above figure shows the mechanisms used by the DataFlux Data Management Studio / Server data job and code generated by SAS® Data Integration Studio to invoke the Accelerator through a call to a stored procedure. Although there is no transform in Data Integration Studio that generates code for calling the specific stored procedures used by the SAS® Data Quality Accelerator, user-generated code, such as proc sql with explicit passthrough, can be used and defined as a user-written transform. In this scenario, the libname definition and the user-written code would be stored in the SAS Metadata Server,

which is also illustrated. In the case of Data Management Studio and Server, the SQL node can be used to pass through the syntax for calling the stored procedures directly as part of a job flow.

DATA QUALITY FUNCTION INPUT AND OUTPUT EXAMPLES

So what kinds of SAS data quality methodologies can be applied to big data architectures? The following is a list of data quality functions that are available today with the SAS® Data Quality Accelerator for Teradata and will be available for additional big data architectures, such as Hadoop, in the future.

Casing

Casing applies context-sensitive case rules to text. It operates on character content, such as names, organizations, and addresses.

INPUT	FUNCTION	OUTPUT
SAS INSTITUTE	Lowercase	sas institute
	Uppercase	SAS INSTITUTE
	Propercase	SAS Institute

Table 1: Casing examples

Extraction

Extraction returns one or more extracted text values, or tokens, as output.

INPUT	OUTPUT	
Blue men's long-sleeved button-down collar denim shirt	Color:	Blue
	Material:	Denim
	Item:	Shirt

Table 2: Extraction examples

Gender Analysis

Gender analysis evaluates the name or other information about an individual to determine the gender of that individual. If the evaluation finds substantial clues that indicate gender, the function returns a value that indicates that the gender is female or male. If the evaluation is inconclusive, the stored procedure returns a value that indicates that the gender is unknown.

INPUT	OUTPUT
Jane Smith	F
Sam Adams	M
P. Jones	U

Table 3: Gender Analysis examples

Identification Analysis

Identification analysis returns a value that indicates the category of the content in an input character string. The available categories and return values depend on your choice of identification definition and locale.

INPUT	OUTPUT
John Smith	NAME
SAS Institute	ORGANIZATION

Table 4: Identification Analysis examples

Matching

Matching analyzes the input data and generates a matchcode for the data, which represents a condensed version of the character value. Similar strings get identical matchcodes.

A sensitivity value can be specified that indicates the degree of similarity that should be applied to consider something a match. For higher sensitivities, two values must be very similar to produce the same matchcode. At lower sensitivities, two values might produce the same matchcode despite considerable dissimilarities. From there, records can be clustered by sorting by matchcodes and fuzzy lookups can be performed via matchcode searches.

INPUT	OUTPUT
Gidley, Scott A	XYZ$$$
Scotty Gidleigh	XYZ$$$
Mr Scott Gidlee Jr.	XYZ$$$
Mr Robert J Brauer	ABC$$$
Bob Brauer	ABC$$$

Table 5: Matching examples

Parsing

Parsing segments a string into semantically atomic tokens.

INPUT	OUTPUT	
Mr. Roy G. Biv Jr	Prefix:	Mr.
	Given Name:	Roy
	Middle Name:	G.

	Family Name:	Biv
	Suffix:	Jr

Table 6: Parsing examples

Pattern Analysis

Pattern analysis returns a simple representation of a text string's character pattern, which can be used for pattern frequency analysis in profiling jobs. Pattern analysis identifies words or characters in the input data column as numeric, alphabetic, non-alphanumeric, or mixed. The choice of pattern analysis definition determines the nature of the analysis.

INPUT	OUTPUT
919-677-8000	999-999-999
NC	AA

Table 7: Pattern Analysis examples

Standardization

Standardization generates a preferred standard representation of data values. Standardization definitions are provided for character content such as dates, names, and postal codes. The available standardization definitions vary from one locale to the next. The return values are provided in the appropriate case, and insignificant blank spaces and punctuation are removed. The order of the elements in the return values might differ from the order of the elements in the input character values.

INPUT	OUTPUT
N car	NC
919.6778000	(919) 677–8000
Smith, Mister James	Mr. James Smith

Table 8: Standardization examples

These examples provide an illustration of how SAS data quality functions can be applied to clean data. The results for these functions are determined by the specified SAS® Quality Knowledge Base definition and locale used, so country-specific variations to these results can be expected. In addition, because the Quality Knowledge Base can be customized, users of these functions can make changes to out-of-the-box functionality.

TIPPING THE SCALES FOR BETTER PERFORMANCE

Performing data quality processing within a big data environment is useful, but only if it performs in a manner that makes it useful to handle the sorts of demands imposed by the size of the data managed by the architecture. Most importantly, that performance needs to scale up by delivering more work capacity as more resources become available to a big data environment.

The figure below illustrates the differences in how a traditional data quality process compares with one that uses the SAS® Data Quality Accelerator in a common scenario, de-duplication of records. In a traditional process, the data is extracted from the source to a remote Data Quality Server, transformed, and then published back to the source. In this example, the source data

impacted must traverse the network twice, first to the Data Quality Server, then the cleansed result over to the final destination.

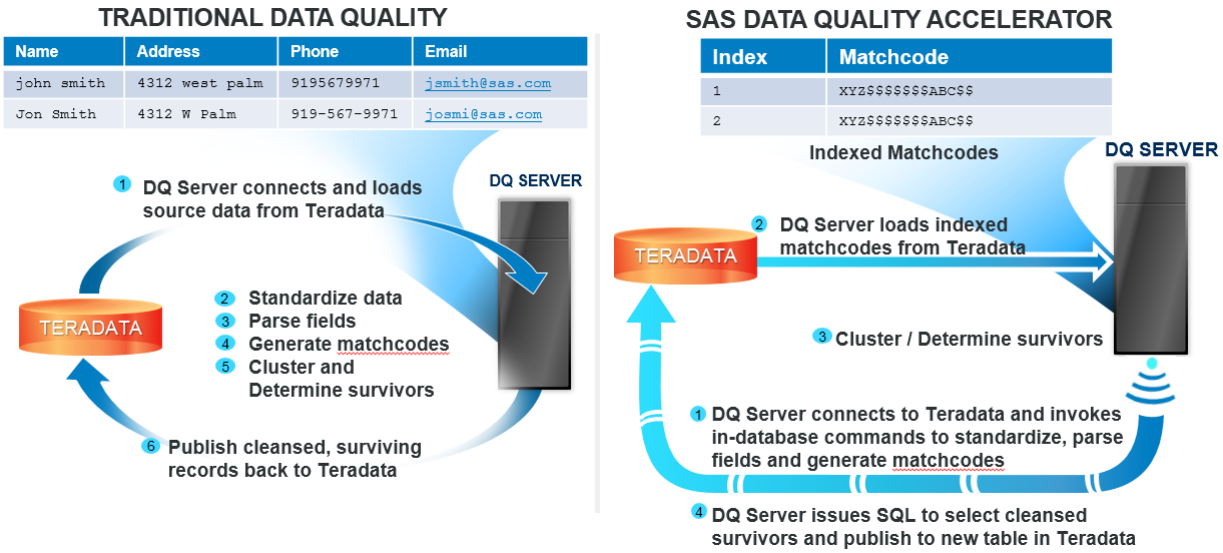

Figure 2. Comparing a traditional data quality process with the SAS® Data Quality Accelerator for Teradata

By comparison, the SAS® Data Quality Accelerator eliminates the need to move the source data to the Data Quality Server, instead performing much of the work within the data source (in this example Teradata) itself. The only information that traverses the network is a set of results based on the in-database data quality processing. These results are then referenced for further processing, but the most resource-intensive steps of the process take place inside Teradata, and the Data Quality Server merely orchestrates the publication of the cleansed results into a destination table.

The kinds of results that one can achieve with this architecture is shown in the table below. In this example, one function is benchmarked comparing the speed in a traditional data quality architecture with that of performing within a big data architecture using the SAS® Data Quality Accelerator:

PROCESS DESCRIPTION	TRADITIONAL OR IN-DATABASE DQ	RECORDS	MINUTES	APPROX. RECS / MINUTE
Gender Analysis	Traditional	500,000	30:59	16,000
Gender Analysis	In-Database	50,000,000	5:19	10,000,000

Table 10. Comparing the performance of a data quality function via a traditional data quality process versus running within a Big Architecture

In the example used in this table, the throughput of the function executed using the SAS® Data Quality Accelerator for Teradata is over 600 times greater than that of a traditional data quality process. The benefit of this approach is illustrated well by these results, but how well do these results scale as more resources are available in the big data environment?

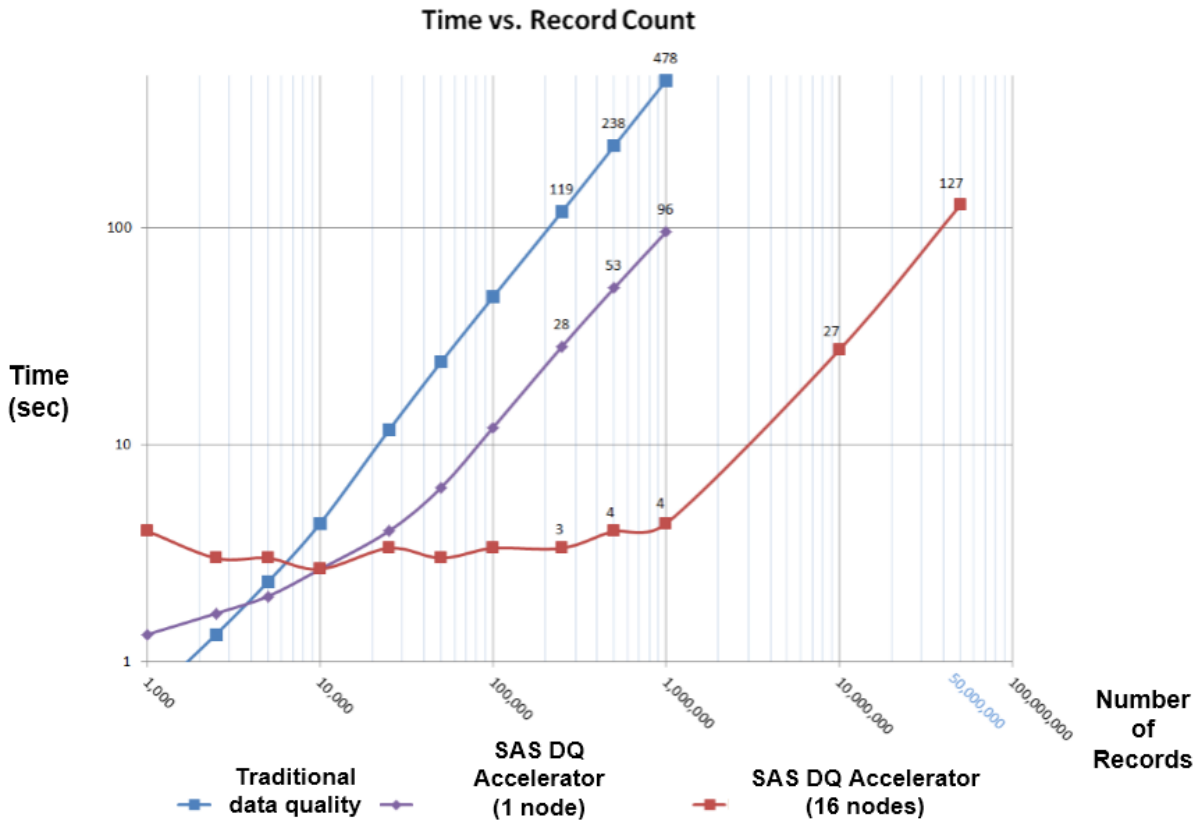

Figure 3. Comparing a traditional data quality process with the SAS® Data Quality Accelerator for Teradata across multiple Teradata architectures

In the above figure, the results of one test show that as records increase by orders of magnitude, the time required to process them is linearly correlated to the available resources of the big data architecture, also Teradata, in this example. This is important because it shows that performance will be maintained by the SAS® Data Quality Accelerator for a big data architecture as the size of the data to be cleansed increases.

MIGRATING HERDS OF BIG DATA CREATES OPPORTUNITY

Like many technologies that once featured prominently in organizations, big data architectures are currently looming large and casting a large shadow over the business by delivering the opportunity for disruptive business change. It does not come without a thoughtful approach to managing the data that these architectures are capable of storing and analyzing, however. By using technologies that take advantage of the processing power and scalability of these behemoths such as SAS® Data Quality Accelerators, organizations can ensure that they can keep up with the pace of change while still getting the best quality results available.