Paper 355-2012

SAS® In-Database Capability – Smart Architecture

Gaurav K Agrawal, US Major Financial Company, USA

1. ABSTRACT

SAS capabilities are well recognized throughout various industries and businesses are getting valuable information from the data using SAS capabilities. However day-by-day data is increasing, which is posing urgent need of not only for a perfect architecture but also smart thinking to work effectively and efficiently.

SAS also has recognizing this fact long back and working dedicatedly towards bringing innovative solutions to customers. SAS In-Database processing also one of the smart concepts to do analytics work and provide extremely valuable benefits to business in time. Using this methodology, not only Server resource utilizations can be decreased but this methodology also enables businesses to reduce time to market. This feature is a critical factor in current competitive environment.

Companies already have their data in a data warehouse environment. This is becoming more prominent with the advent of specialized appliances in this technology stack. Traditionally, users have resorted to creating SAS copy of the data in order to execute SAS analytics. This resulted multitude of challenges like performance bottlenecks, requirement of additional infrastructure for storage and processing, need for additional security maintenance and most of all extended proliferation of data causing classical difficult in maintaining one version of truth.

Note: This paper is designed to share knowledge around all options for SAS in-database processing to SAS user community. Hence this paper is not written considering specific database and should be applicable to all databases. Because of the mentioned reason this paper do not talk specific name of macro or database specfic code. This paper will give you a good direction to proceed towards in-database processing specific to your organization and database need.

2. INTRODUCTION

As name signifies “SAS In-Database“ processing allows processing to happen inside Database to utilize resources much more efficiently and effectively. SAS Access Engine does have limited in-built functionality to convert user query to get processed at database level. However users can explicitly also mention in code to pass query to database. SAS Analytics is also one of the extremely useful areas for business. Business/IT users, develops various models to address business problems and to do some prediction using these models. SAS In-Database functionality also provides capabilities to get such analytics done completely in database, which becomes extremely beneficial for industries. This paper will focus on all these capabilities under below topics

- ✓ SQL/Proc Pass through Capabilities
  - Implicit SQL Pass Through
  - Explicit SQL Pass Through
- ✓ In-Database Analytics - SAS Scoring

3. WHY IN-DATABASE? CAPABILITIES/ADVANTAGE

SAS In-Database capability brings lots of benefits for Infrastructure and also from business perspective. In industries data is growing and it’s must to have required storage, Processing power and network for meeting the data processing requirement. SAS In-Database enables environment to leverage Massive Parallel Processing architecture of database and allow processing to happen local at database level and only final result travels to network that too if required. Hence utilizing in-database capability is not just “good to have” concept now rather its “must have” concept to provide competitive edge to business.

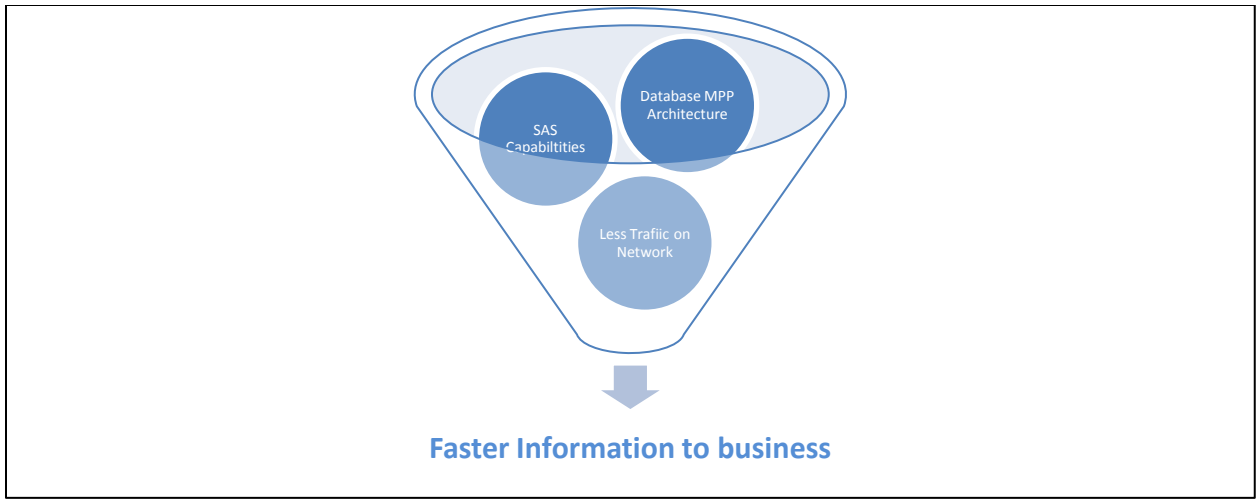

Figure 1: In-Database Capability

SAS In-Database also enables IT organization to avoid data duplicity as we can do processing local to data. Various organizations do maintain centralize repositories of data (Data Warehouse) and further on that some Data Marts are created. SAS In-Database allows organization to maintain this architecture rather bringing all data to SAS environment to create duplicity.

4. SQL PASS THROUGH CAPABILITIES

SAS SQL pass through capability provides two kinds of functionalities, which are called as Implicit Pass Through and Explicit Pass Through. Many folks in industry get confused with “pass through” and says all such queries gets passed to database for processing. However that is not true and it works little differently. Please see below architecture diagram for pass through scenario.

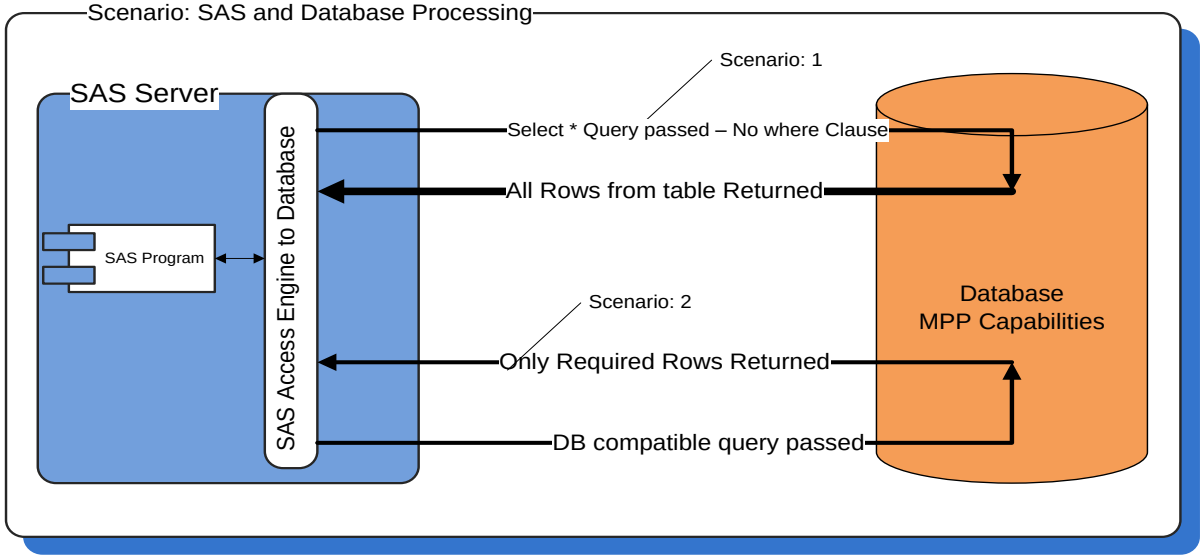

Figure 2: SAS and Database Processing

In first scenario of above depicted figure, it can be noticed that if query is not compatible with the database then all rows will be fetched from database tables to SAS environment and then further processing will be happening in SAS environment only. This scenario will actually put un-necessary load on Database, SAS Server and on Network.

On the other hand in second scenario query is completely passed to Database environment with “where” or “group” clause hence only required rows returns to SAS environment. This will get processing done at database level and will reduce load on Network and on SAS Server. Such scenarios are extremely efficient where huge data is in process.

4.1 SQL PASS THROUGH CAPABILITIES - IMPLICIT SQL PASS THROUGH

Implicit Pass Through is designed to make SAS SQL queries independent of the Database. Hence user writes same SQL in order to interact with Teradata database, which he/she must have used to interact with Oracle database. This enables SAS architecture to provide Integration capabilities without being dependent on underlying database. End user even can write queries to merge tables from different databases and internally SAS Access Engine will take care of the query generation for respective database. SAS PROCs also leverage this functionality of SAS environment in order to interact with Database.

This capability some time poses a limitation also to SAS environment where SAS Access Engine is not able to convert query compatible to database and then complete Database table is fetched to SAS work area and then further logic is applied internally by SAS environment. Such kind of scenarios are noticed in very complex queries OR if SAS Formats/Functions are used in queries/Proc.

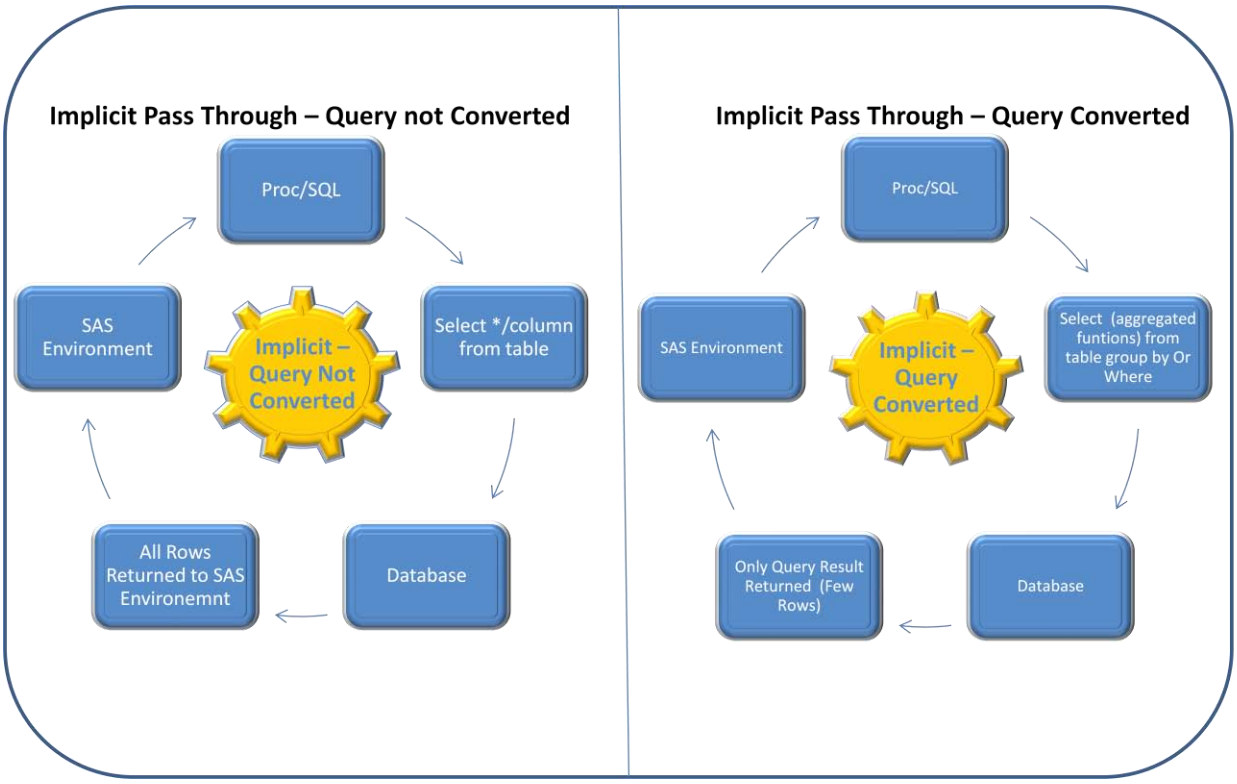

Figure 3: Implicit Pass Through

So as per Implicit SQL Pass Through, user written SQL/Proc is passed to SAS Access Engine to make Database compatible queries. SAS Access Engine passes converted queries to Database. Hence Implicit Pass Through does have its advantages and disadvantages. Every organization has to decide around usage of these features as per their need.

CHALLENGE AND SOLUTION

It has to be noticed that if user are making connection to database through SAS Metadata library then it does not support in-database functionality especially in SAS 9.2. In fact all SAS Proc will not be converted to Database compatible SQL in SAS 9.2.

S. No.	Challenges	Solution

1.	If users are using Metadata Library to make connection to database then in that case SAS PROCedures will not be passed for in-database processing. Hence all data of table will be returned from database to SAS environment.
	
	Note: this limitation is limited to SAS 9.2 environment. However same has been enhanced in SAS 9.3
		In order to overcome this problem user has to make slight adjustment in code and things will work as required and Procs will be passed for in database processing
		
		In order to get this working first assign Database connection using below syntax
		
		`Libname libref <DB> Server=<server> Database=<dbName> authdomain=authdomain;`
		`/* Authdomain name can be obtained from SAS Admin Or can access through XML. Same Information can be obtain from properties of libraries in certain scenarios*/`
		
		`/*Then use SAS Proc against this Libname */`
		`Proc Freq data=libref.table;`
		`Run;`
		
		1. In above example user made the database connection using the Metdata credential information (fetched by authdomain option). Hence Metadata library limitation does not apply here.
		
		2. In such connection Metadata Library security authorization will not be applied and as per database security for connected id (as per “authdomain”) tables will be visible to user
		
		If because of some reason (may be environmental configuration) query is still not being converted then use option SQLGENERATIN=DBMS.
		Note: Implicit Pass through Capability varies for SAS PROC for different database. List can be obtained from SAS Institute depending on database in action.

2.	It has been noticed that in some cases if users uses SAS Formats or SAS functions in SQL that also pose limitation to pass complete query to database.	In order to avoid this situation try to avoid using SAS formats and SAS function in SQL query. If required apply the formats and function on the returned data from SAS Query. Few databases also support mechanism to deploy format in database itself. If you are able to deploy formats in database then using sas_put function even SAS formats can be passed to database. Currently I do see such capabilities are more advanced with Teradata database. However other databases are also catching up on these capabilities.

Note: As mentioned some of the database supports to publish formats inside database and Teradata is ahead in terms of integration with SAS. There are format-publishing macros available with SAS Access Engine license. However steps to publish formats to database are kind of same as later described to publish models. Only main difference will be that separate macro has to be used to publish formats in database rather same macro used to publish models. At database side also certain installation will be required.

4.2 SQL PASS THROUGH CAPABILITIES - EXPLICIT SQL PASS THROUGH

Explicit Pass through is designed to ensure that user written queries are passed to database. User writes explicit connection statement to database with SQL compatible to database. This SQL is internally passed to SAS Access Engine and then SAS Access Engine takes the responsibility to pass this SQL to Database without any change. If SQL is not compatible with database then Database engine will throw error to SAS Access Engine, which will pass to user SAS log.

Hence in order to ensure that query gets passed to database, users have to ensure that they have required SQL writing knowledge to database and only compatible SQL has to be written. This arrangement will ensure that SQL processing is completely happening inside database and only required result is returned to SAS environment. This architecture will provide faster response for the user query and will reduce load on network also.

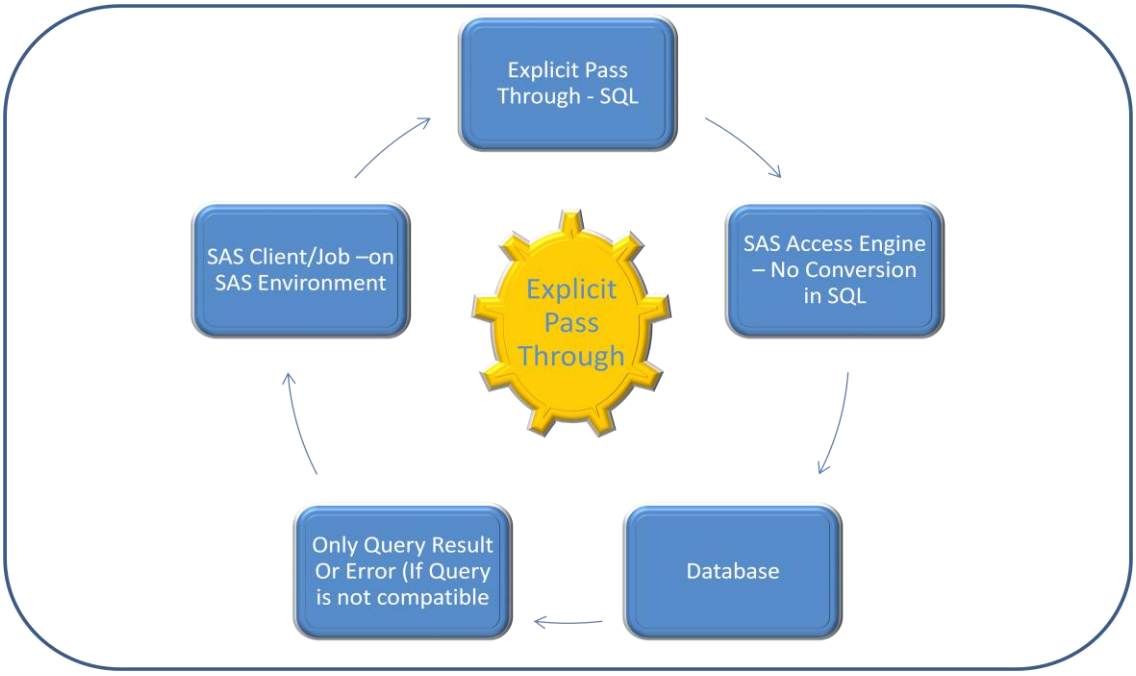

Figure 4: Explicit Pass Through

CHALLENGE AND SOLUTION

S. No.	Challenges	Solution

1.	If users are using Metadata Library to make connection to database then in that case user will face problem in using explicit pass through. Especially in case when faceless/group/proxy id is in action. User will not have password of such IDs.
	
	Also User would like to write code in such a manner they are not required to put password in database connection string of SAS code.
		In order to overcome this problem user has to make slight adjustment in code and things will work as required.
		
		Traditional Approach
		`PROC SQL;`
		`CONNECT TO TERADATA(USER= PASSWORD= SERVER=`
		`DATABASE= <other connection options>);`
		`CREATE table _name_ AS /* Optional */`
		`SELECT * FROM CONNECTION TO TERADATA`
		`( Teradata SQL Statement);`
		`DISCONNECT FROM TERADATA;`
		`QUIT;`
		
		Recommended Approach
		`PROC SQL;`
		`CONNECT TO TERADATA(Authdomain= SERVER=`
		`DATABASE= <other connection options>);`
		`CREATE table _name_ AS /* Optional */`
		`SELECT * FROM CONNECTION TO TERADATA`
		`( Teradata SQL Statement);`
		`DISCONNECT FROM TERADATA;`
		`QUIT;`
		
		/* Authdomain name can be obtained from SAS Admin Or can access through XML. Same Information can be obtain from properties of libraries in certain scenarios*/

		This syntax will even avoid the requirement to explicitly write password inside SAS code. In above code Authdomain will fetch credential from Metadata to help make connection with database. There are few point to be noticed here 1) Only those who have security permissions to fetch it can access "Authdomain". 2) In such connection Metadata Library security authorization will not be applied and as per database security for connected id (as per "authdomain") tables will be visible to user

5. IN-DATABASE ANALYTICS – SAS MODEL SCORING

SAS Enterprise Miner is one of the most used modeling tools in industry. SAS recognized the capabilities of database MPP architecture for processing and considering same SAS has launched product called SAS Scoring Accelerator. This product enables SAS Environment to push SAS models inside database and eliminates the need to move data between SAS environment and the database. This in turn reduces the resource utilization and elapse time for model processing. Reduced elapse time enables business to take completive edge in their business. Below architecture depicts that how models can be published inside the database.

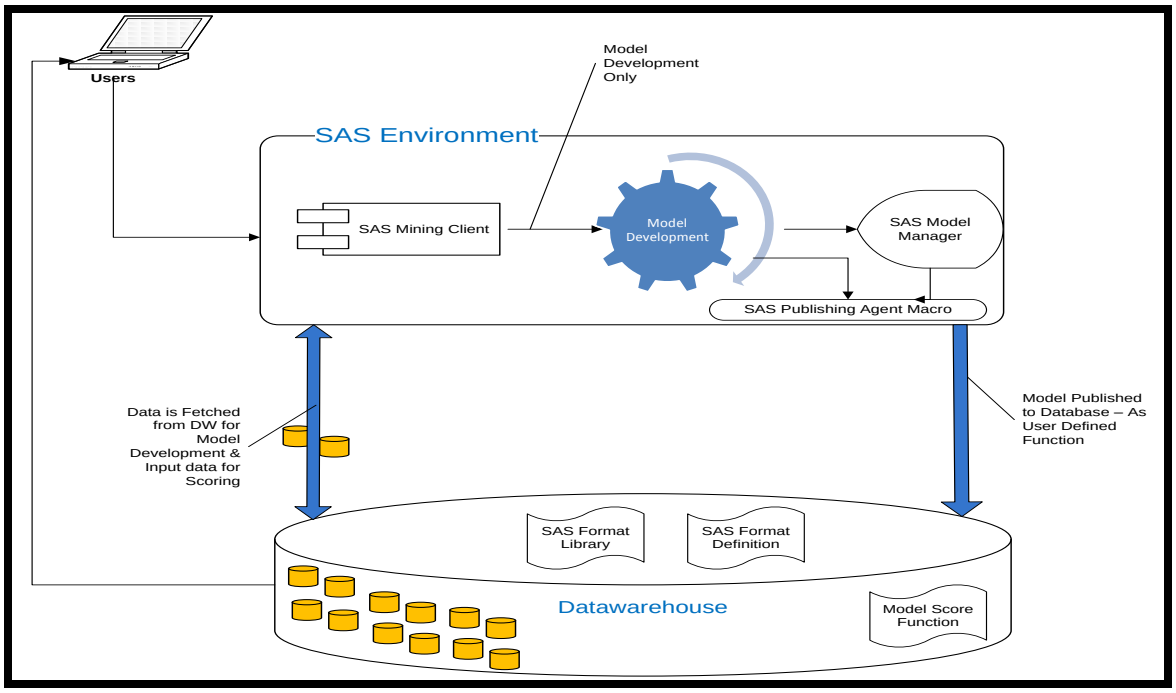

Figure 5: SAS Model Publishing Architecture

SAS Model can be published to database by two methods and it has been described here. Following steps will be followed to publish model through **SAS Publishing Agent Macro**

1) User will develop SAS Model using SAS Enterprise Miner tool
2) User/Admin will export the model using SAS Export node, which is enabled because of SAS Scoring Accelerator component
3) Various Files will be created as per this export steps
4) SAS Publishing Agent (SAS Macros) provided under the license of SAS Access engine will be used to publish exported contents to respective database to define as UDF (User Defined Function)
5) SAS Model will be deployed in Database as user defined function
6) User can use these function in their SQL to do scoring

Following steps will be followed to publish model through **SAS Model Manager**

1) User will develop SAS Model using SAS Enterprise Miner tool
2) User/Admin will register Model Package inside Model Manager
3) User/Admin will Publish Model Directly in Database using Model Manager tool
4) SAS Model will be deployed in Database as user defined function
5) Users can use these function in their SQL to do scoring

5.1 SAS SCORING – IN-DATABASE ARCHITECTURAL IMPLEMENTATION DETAILS

Below Table Describe the details of various components required for SAS Model Scoring to work in database.

S. No.	Component	SAS Environment	Database Environment
1.	SAS License	✓ SAS Access Engine to Database ✓ SAS Scoring Accelerator to Database ✓ SAS Model Manager (Optional) – this will need SAS Content Server also	✓ SAS Scoring Accelerator to Database – Installation required at Database level with some write privileges to certain Ids
2.	Installation	✓ Base SAS ✓ SAS Access Engine to Database ✓ SAS Scoring Accelerator ✓ SAS Model Manager (Optional) ✓ Database Client in SAS Environment	✓ Database Obviously ✓ SAS Scoring Accelerator Components ✓ On some database C compiler (some appliance do have this by default) ✓ Publish mandatory functions in database as part of installation (These functions will be used to publish models)

5.2 IN-DATABASE ANALYTICS – SCENARIO ANALYSIS BEFORE AND AFTER IN-DATABASE FOR MODELS

SAS In-Database provides capability to process model completely inside the database. Model score process utilizes the MPP architecture of database to process model much faster and result will be stored in Data warehouse itself, which in turn will reduce load on Network.

MODLE SCORING – BEFORE IN-DATABASE

In the scenario where in-database processing capabilities are not utilized for model processing generally below steps will be followed.

1) Modeling data will be created in Data warehouse
2) This modeling data will be extract to SAS environment
3) Using SAS Enterprise Miner Model will be developed
4) Scoring input data will be created in Data warehouse with required logic
5) Scoring input Data will be extracted to SAS environment
6) SAS Enterprise Miner Model code will be executed against the data extracted from data warehouse
7) Final scored data will be loaded in data warehouse to be accessed by various application

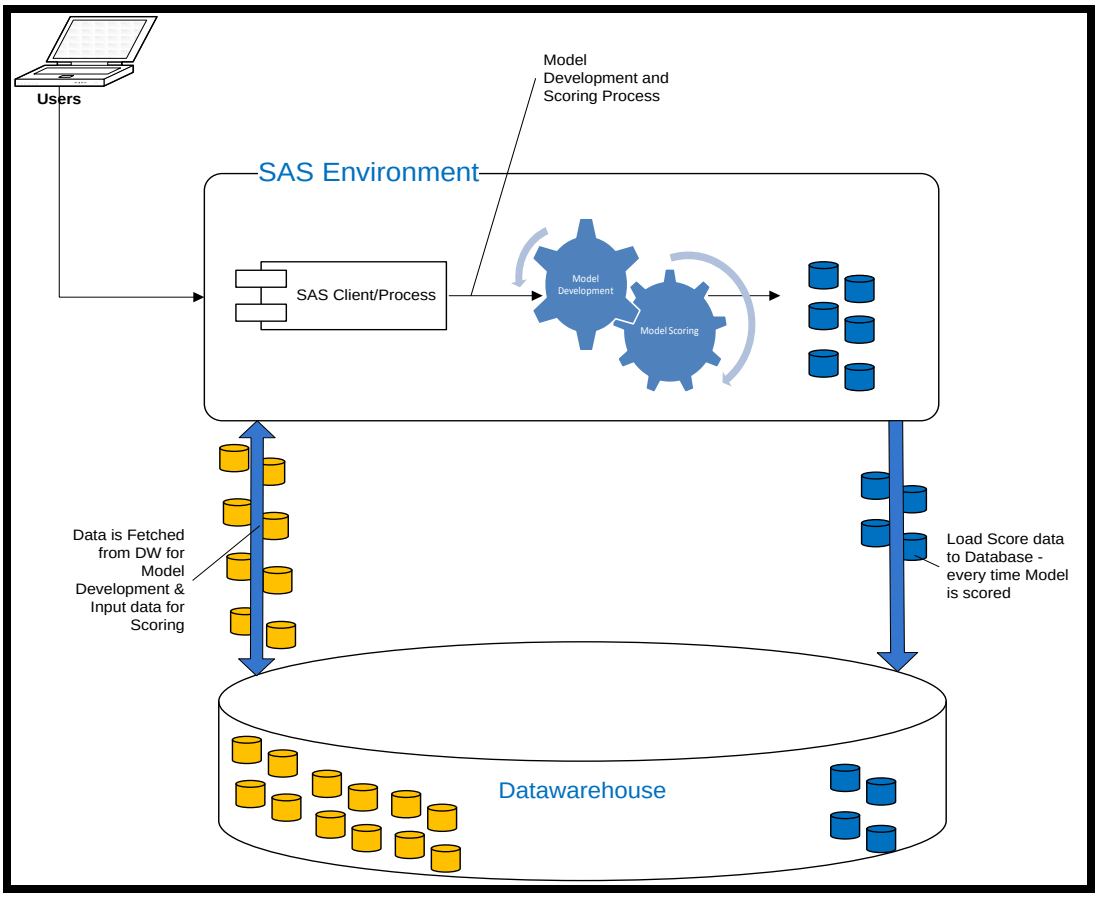

Figure 6: SAS Model Scoring in SAS Environment

It must have been noticed that above steps lead to lot of data being transferred between Data warehouse and SAS environment. Also scoring of data is completely happening in SAS environment. In addition to that after processing, data will be loaded in Data warehouse. This is leading to delayed information to business.

MODEL SCORING – AFTER IN-DATABASE

In the scenario where in-database processing capabilities are utilized, model development cycle remain same. However model-processing architecture gets changed completely. Below mentioned steps will be followed.

1) Modeling data will be created in Data warehouse
2) This modeling data will be extract to SAS environment
3) Using SAS Enterprise Miner Model will be developed
4) Developed Model will be published in Data warehouse using SAS Scoring Accelerator
5) Scoring data will be created in Data warehouse with required logic
6) Published model code will be executed inside database only against the data created in data warehouse
7) Final scored data will become available directly in Data warehouse and there will be no need to move scored data

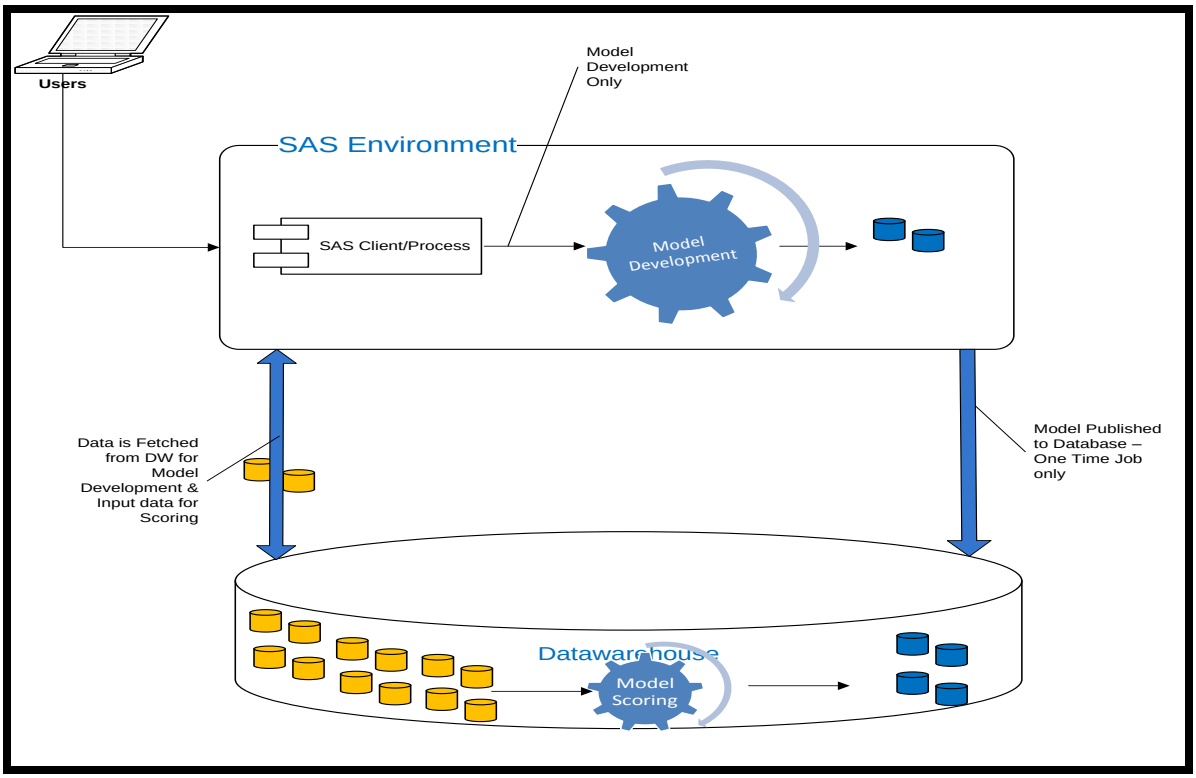

Figure 7: SAS Model Scoring in Database

If we notice then there is data movement only for the process of Model development (one time). After that all processing is happening inside data warehouse only, which utilizes MPP architecture for faster processing. By this approach network load is very much reduced and business is getting required information very fast. Point to be noted that in order to utilize such capabiltiy Data warehouse Infrastructure need to be design considering the model-scoring load.

5.3 IN-DATABASE ANALYTICS - CHALLENGES

There are certain points, which have to be considered in order to move towards Model Scoring In-Database. Model development exercise has to ensure that model development process is compliant to the requirements of SAS Scoring Accelerator to publish model inside database. There are certain nodes of SAS Enterprise Miner, which are not supported by SAS Scoring Accelerator and those nodes has to be avoided in order develop Models. For example SAS Code Node can not be used inside SAS Modeling exercise if developed Model need to be publish in database. Similarly there are few other nodes, which are not supported for inside database processing. Hence company has to evaluate benefits before proceeding in the direction of model scoring inside database.

Note: These considerations are documented for SAS 9.2 platform. In coming version of SAS these limitation should be reduced and SAS Tech Support can be contacted to take latest list.

6. CONCLUSION

Other than infrastructure benefits, in-database brings lots of business value also. Because of this, business get information very quick with reduce cost, which intern becomes profitable. Faster information will bring a competitive edge to business in market. SAS in-Database concept brings SAS environment and Database MPP processing capability together. This concept helps to process data more local to DW and brings only required data to SAS environment.

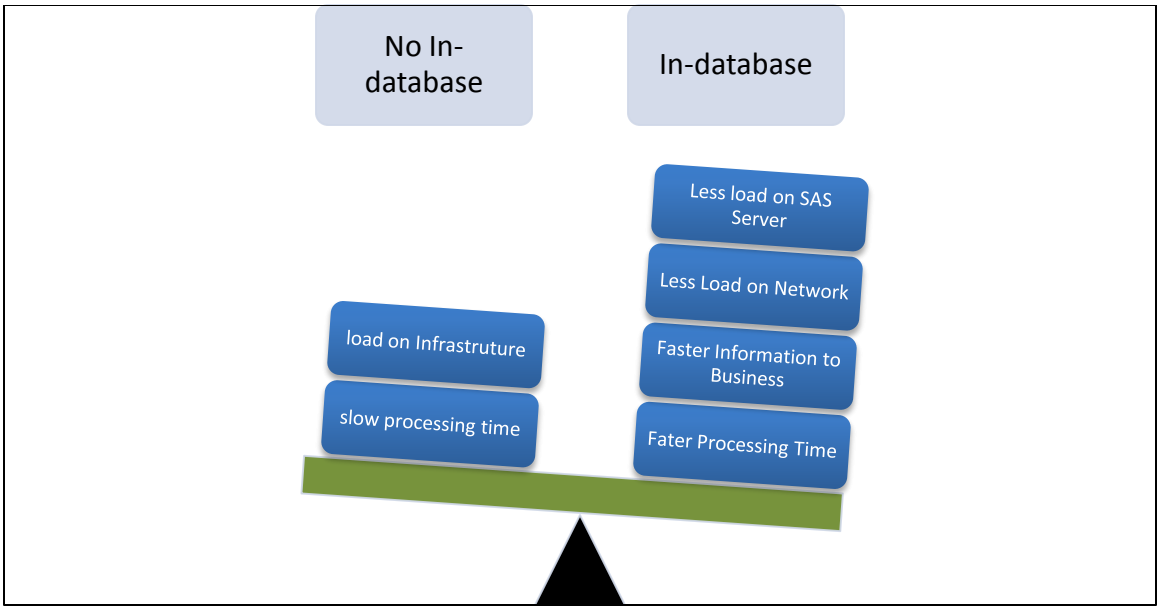

Figure 8: SAS in-Database – Architectural Benefits

It is definitely recommended that companies move in this direction to get faster business value. Deploying models to database will extremely enhance the processing capability. I recommend companies to trigger small evaluation effort before utilizing these capabilities.

7. REFERENCE & RECOMMENDED READING

S. No.	Topic	Link
1.	Teradata Scoring Accelerator Details	http://support.sas.com/documentation/cdl/en/scraccltdug/63138/HTML/default/viewer.htm#n1s9fxl3fr6mirn13srwpttisp18.htm
2.	Greenplum Scoring Accelerator Details	http://support.sas.com/documentation/cdl/en/scracclgpug/62997/HTML/default/viewer.htm#p0nt4kclg2vhkun1jn0llx9s2vsd.htm
3.	Some of the SAS Procedures Details	http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/viewer.htm#procwhatsnew902.htm
4.	Overview of SAS Scoring Accelerator in 9.3	http://support.sas.com/documentation/cdl/en/indbug/64690/HTML/default/viewer.htm#n0bfz6qhhupvsyn1ncvkmnqp84ir.htm

8. ACKNOWLEDGEMENTS

I would like to thank Hitesh Sharma and Subodh Agrawal for reviewing this paper. My special thank to Monika Singhal for giving me idea to write paper on this topic.

9. CONTACT INFORMATION

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

Name: Gaurav K Agrawal
E-mail: gaurav_agrawal@yahoo.com