Paper 399-2013

Leveraging Big Data Using SAS® High-Performance Analytics Server

Priyadarshini Sharma, SAS Institute Inc., Schaumburg, IL

ABSTRACT

With an increasing amount of information being collected by organizations, SAS® High-Performance Analytics Server enables decision-makers to create greater business value by leveraging big data. This paper describes some best practices and techniques to use when dealing with big data analytics. The following topics are discussed: (1) different methods of loading data for Teradata and Greenplum appliances (2) checking the distribution of loaded data on the appliance, and (3) using the new HPDS2 procedure in SAS® 9.3. This paper also gives a brief overview of how SAS High-Performance Analytics Server enables you to use 100% of your data to get more precise insights and to build complex models at breakthrough speed, using results from an example model.

INTRODUCTION

Organizations are collecting enormous amounts of data every day, and the ability to handle, store, and analyze it is becoming increasingly complex. Solving this “big data” problem is crucial to staying ahead of the competition and to being innovative. SAS High-Performance Analytics Server is a leading-edge technology that leverages big data to make better-informed decisions at breakthrough speeds.

SAS High-Performance Analytics Server is an in-memory analytics offering, which enables you to build complex models using structured, unstructured, or streaming data within minutes, or even seconds, to get more accurate and timely insights. Its core offering for advanced analytics capability (data mining and predictive analysis, text mining, optimization, and econometrics) handles all aspects of the analytics lifecycle, spanning data exploration, model development, and model deployment. It empowers you to analyze 100% of the data, quickly add new or more variables, run more iterations for optimum modeling, build more models, and still use the same easy-to-use interfaces SAS offers to create high-performance models.

SAS® High-Performance Analytics procedures can execute in symmetric multiprocessing (SMP) or massively parallel processing (MPP) mode. SMP mode is multithreading on the client machine, MPP mode is a computing model in which calculations occur on several nodes in a distributed computing environment. MPP mode has several variations:

- Client-data (or local-data) mode
- Alongside-the-database mode
- Alongside HDFS mode
- Alongside LASR mode

In terms of infrastructure, SAS High-Performance Analytics Server is designed to run on purpose-built high-performance appliances from EMC Greenplum and Teradata, or on Hadoop Distributed File System (HDFS) running on commodity hardware to solve complex problems in a highly scalable, distributed environment using in-memory analytics processing. This paper provides techniques for loading data and checking the distribution of the loaded data on Teradata and EMC Greenplum distributed databases. Also learn more about the new HPDS2 procedure and see results from a high-performance model that predicts churn using telecom data.

LOADING DATA IN A DISTRIBUTED ENVIRONMENT

In order to run the SAS High-Performance Analytics procedures in alongside-the-database mode, data needs to be distributed to the appliance. Here are a few methods that can be used to load the data:

- HPDS2 procedure
- DATA step
- PROC APPEND
- SQL pass-through

HPDS2 PROCEDURE

The HPDS2 procedure executes DS2 language statements in a SAS High-Performance Analytics environment for parallel execution. It is an efficient method for moving big data to the appliance. (See section HPDS2 Procedure.)

```
proc hpds2 data=sampsio.hmeq
           out =gplib.hmeq(distributed_by='distributed randomly');
           performance host="&GRIDHOST" install="&GRIDINSTALLLOC";
```

```
            data DS2GTF.out;
              method run();
              set DS2GTF.in;
              end;
            enddata;
run;
```

DATA STEP

Data can be loaded in bulk directly from the DATA step into the appliance. In order to load data to the appliance, use the BULKLOAD option and specify the distribution key for the table by using the dbcreate_table_opts (for Teradata) or distributed_by (for Greenplum) data set option. By default, Teradata takes the first column as the Primary Index (PI). Be sure that you specify 'no primary index' if there is no PI when loading data on Teradata. In the case of Greenplum, if no distribution key is mentioned, data is distributed randomly across the nodes by default.

```
data tdlib.simulate(bulkload=yes dbcreate_table_opts='unique primary index(PKEY)');
set work.simulate;
run ;
```

PROC APPEND

Similar to the DATA step, the APPEND procedure can also be used to load data in bulk to the appliance by using BULKLOAD and the dbcreate_table_opts (for Teradata) or distributed_by (for Greenplum) data set option.

```
proc append base=gplib.simulate (bulkload=yes distributed_by='distributed randomly')
            data=work.simulate;
run;
```

The DATA step and PROC APPEND use the SAS/ACCESS® engine, whereas the HPDS2 procedure runs within the framework of SAS Embedded Process.

SQL PASS THROUGH

Data can also be loaded using the PROC SQL implicit or explicit pass-through method, but these methods are not as efficient as the others discussed in this paper.

Implicit pass-through example:

```
proc sql;
create table tdlib.test_train (dbcreate_table_opts='unique primary index(pkey)')as
select * from tdlib.test1
where partition=0;
select count(*) from tdlib.test_train as cnt_train; quit;
```

Explicit pass-through example:

```
proc sql;
connect to teradata( server=&gridserver user=&user password=&uid_pwd database=&db);
create table tdlib.test_valid as select * from connection to Teradata
(select * from test1
where partition=1);
quit;
```

In addition, techniques using utility tools such as Teradata Parallel Transporter (TPT) and Gpfdist from Teradata and Greenplum respectively can also be used to load very large data.

CHECKING DATA DISTRIBUTION

Evenly distributed data is essential for efficient parallel processing. Uneven data distribution can have a negative impact on performance. A unique primary index is recommended when loading big data to ensure even distribution. The following methods can be used to check the data distribution on Greenplum and Teradata.

GREENPLUM

Distribution of a table in terms of rows stored on each segment can be checked by using the following SQL code:

```
select gp_segment_id, count(*) from public.simulate group by 1 order by 1 desc;
```

Data Output | Explain | Messa

	gp_segment_id integer	count bigint
1	191	2593
2	190	2597
3	189	2561
4	188	2578
5	187	2602
6	186	2595
7	185	2639
8	184	2613
9	183	2574
10	182	2603

Display 1: Table Distribution Per Segment

Greenplum provides an administrative schema called gp_toolkit that can be used to check data distribution. The gp_toolkit schema contains a number of views that you can access using SQL commands. The gp_toolkit schema is accessible to all database users, although some objects might require superuser permissions.

The gp_skew_coefficients view

This view shows the distribution based on the coefficient of variation for the data stored on each segment. A low value is an indication of good distribution. High values indicate data skew.

```
select * from gp_toolkit.gp_skew_coefficients;
```

Output pane

Data Output | Explain | Messages | History

	skcoid oid	skcnamespace name	skcrelname name	skccoeff numeric
292	331283	public	sgf binary	0.00114
293	52797075	public	simulate	0.80816
294	52797585	public	hmeq ds2	16.2729
295	52797679	public	simulate ds2	0.83073
296	52797774	public	simulate new	0.32451
297	52797869	public	simulate n2	0.32451

Display 2: Skew Coefficients

The gp_skew_idle_fractions view

This view shows the percentage of the system that is idle during a table scan, which is an indicator of uneven data distribution. For example, a value of 0.1 indicates 10% skew, a value of 0.5 indicates 50% skew, and so on. Tables that have more than 10% skew should have their distribution policies evaluated.

```
select * from gp_toolkit.gp_skew_idle_fractions;
```

Output pane

Data Output | Explain | Messages | History

	sifoid oid	sifnamespace name	sifrelname name	siffraction numeric
292	331283	public	sgf binary	0.00003220
293	52797075	public	simulate	0.01840683
294	52797585	public	hmeq ds2	0.32518115
295	52797679	public	simulate ds2	0.02355955
296	52797774	public	simulate new	0.00831429
297	52797869	public	simulate n2	0.00831429

Display 3: Idle Fraction

TERADATA

Hash-related functions are used in the following example to check data distribution across all Access Module Processors (AMPs).

```
select hashamp (hashbucket (hashrow ("pkey"))) as amp#, count(*)
from hps.simulate
group by 1
order by 1;
```

Answerset 1

	AMP#	Count(*)
1	0	880
2	1	829
3	2	880
4	3	880
5	4	889
6	5	859
7	6	874
8	7	864
9	8	855
10	9	881
11	10	847
12	11	863

Display 4: Distribution across All AMPs

The results can be made more descriptive by showing the percentage of rows stored in each AMP. This example shows the distribution for a table with one billion rows.

```
select hashamp(hashbucket(hashrow("pkey"))) as "amp#",
count(*) as "number of rows per amp",
c.cnt as "total number of rows",
cast((count(*)*100) as decimal(15,2))/cast(c.cnt as decimal(15,2)) as "% of rows per
amp"
from hps.simulate,
(select count(*) as cnt from hps.simulate) c
group by 1
order by 1,2 desc;
```

Answerset 1

	AMP#	Number of rows per AMP	Total number of rows	% of rows per AMP
1	0	880	500,000	0.18
2	1	829	500,000	0.17
3	2	880	500,000	0.18
4	3	880	500,000	0.18
5	4	889	500,000	0.18
6	5	859	500,000	0.17
7	6	874	500,000	0.17
8	7	864	500,000	0.17
9	8	855	500,000	0.17
10	9	881	500,000	0.18
11	10	847	500,000	0.17
12	11	863	500,000	0.17
13	12	860	500,000	0.17
14	13	879	500,000	0.18
15	14	895	500,000	0.18

Display 5: Row Percentage on Each AMP

HPDS2 PROCEDURE

The HPDS2 procedure is a high-performance implementation of the DS2 (DATA step 2) language statement in a distributed computing environment. It integrates the DATA step with object-oriented concepts for parallel execution. PROC HPDS2 is most useful when significant amounts of computationally intensive, row-independent logic must be applied to the data.

FEATURES

The HPDS2 procedure offers the following features:

- Provides the ability to execute DS2 code in parallel
- Enables DS2 code to be executed on a local client machine or on the SAS High-Performance Analytics grid
- Enables control of the level of parallelism per execution node and the number of nodes to engage
- Performs a syntax check of the DS2 code on the local client machine before sending it to the grid for execution
- Manages data migration to the location of execution and movement back to the client machine as needed
- Runs within the framework of SAS Embedded Process

EXAMPLE

```
proc hpds2 data = work.simulate (drop=a b c)
            out = gplib.simulate (distributed_by='distribute by(pkey)');

       performance host="&GRIDHOST" install="&GRIDINSTALLLOC";
       data DS2GTF.out;
       dcl double valid;
              method run();
              set DS2GTF.in;
              if ranuni(1)<0.7 then valid=0;
              else valid=1;
              end;
       enddata;
run;
```

Note: DS2GTF.out and DS2GTF.in refer to the DS2 Grid Table Function driver *and must be included* as part of the DATA statement.

OUTPUT

```
NOTE: The HPDS2 procedure is executing in the distributed computing environment
with 32 worker nodes.
NOTE: The data set GPLIB.SIMULATE has 500000 observations and 8 variables.
NOTE: There were 500000 observations read from the data set WORK.SIMULATE.
NOTE: PROCEDURE HPDS2 used (Total process time):
       real time           12.36 seconds
       cpu time            2.43 seconds
```

RESULT

Display 6 below shows the performance information listing execution mode as Distributed, the number of compute nodes, and the number of threads per node.

The HPDS2 Procedure

Performance Information	
Host Node	green1.unx.sas.com
Data Server	10.16.6.33
Install Location	/opt/d312h3/laxno/121126/TKGrid
Execution Mode	Distributed
Grid Mode	Symmetric
Number of Compute Nodes	32
Number of Threads per Node	24

Display 6: PROC HPDS2 Performance Information

LIMITATIONS

The HPDS2 procedure has the following limitations:

- Limited to single input stream
- Does not support Merge capabilities
- Due to threading and parallel execution, ordering of output (sorting), lagging, and BY-group processing are not possible
- Does not support the overwrite option
- Does not support informats and user-defined formats
- Does not support the REAL, TINYINT, NCHAR, TIMESTAMP, DATE, and TIME data types
- Does not currently support delimited identifiers (for example, dcl double "a%& b")
- Does not currently support use of nested SQL within the SET statement
- The PUT statement does not currently write any data to the client log

SAS® HIGH-PERFORMANCE ANALYTICS SERVER

Using SAS® High-Performance Analytics Server, you can create and run complex models on never-before-possible volumes of data against thousands of attributes. Analytical results that used to take weeks or days to generate are now available in seconds.

The high-performance procedures offer a core set of analytics functionalities for data exploration, data mining, predictive analysis, text mining, optimization, and econometrics, and forecasting. These algorithms and nodes are threaded for concurrent execution and have been adapted for a distributed computing environment.

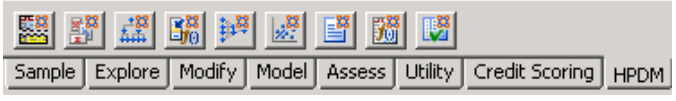
Here are the high-performance procedures available in SAS® High-Performance Analytics Server 12.2.

Data Preparation / Exploration	Predictive Analytics & Data Mining	Text Mining*	Optimization*	Econometrics & Forecasting
▪ HPDS2	▪ HPLOGISTIC	▪ HPTMINE	▪ HPLSO	▪ HPCOUNTREG
▪ HPSAMPLE	▪ HPREG	▪ HPTMSCORE	▪ OPTMODEL	▪ HPSEVERITY
▪ HPSUMMARY	▪ HPLMIXED		DECOMP#	▪ HPQLIM
▪ HPDMDB	▪ HPNEURAL		▪ OPTMODEL	
▪ HPCORR	▪ HPNLIN		Multistart#	
▪ HPIMPUTE	▪ HPDECIDE		▪ OPTMILP	
	▪ HPFOREST		Tuner option#	
	▪ HP4SCORE			
	▪ HPREDUCE			
	▪ HPBIN			

**Currently available only for Teradata and EMC Greenplum*

 # Experimental procedure in HPAS 12.2

The high-performance enabled nodes in SAS® Enterprise Miner available via a new HPDM tab are:
HP Data Partition, HP Explore, HP Forest, HP Impute, HP Neural, HP Regression, HP Text Miner*, HP Transform, HP Variable Selection.



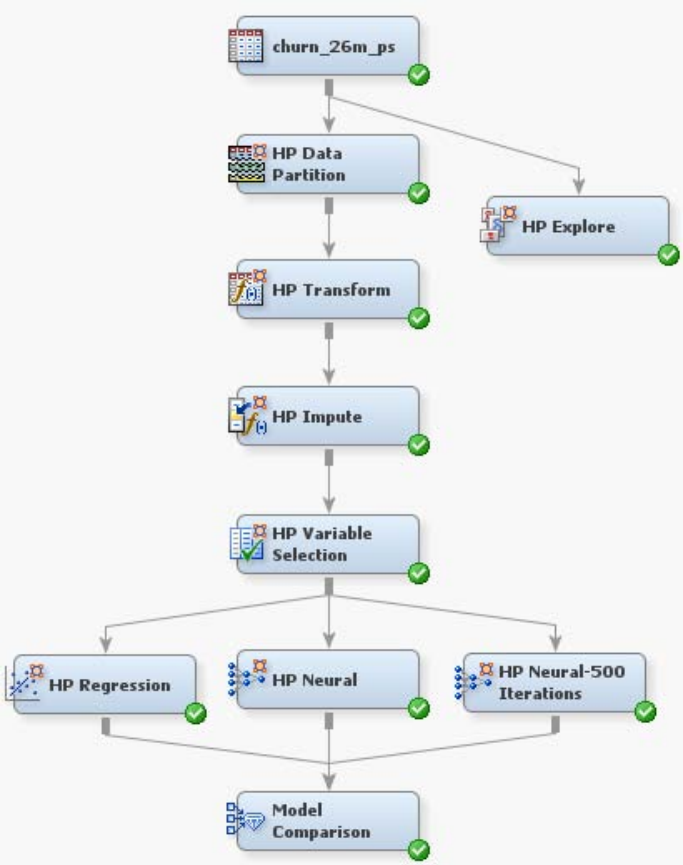
Display 7: HPDM Nodes

DATA

The telecommunications industry sees high volumes of voluntary and involuntary churn annually. It costs significantly more to acquire a new customer than to retain an existing one. A good churn model can provide early warnings for retaining customers and can also provide valuable insights into the drivers of churn. This type of analysis can be used in many industries.

The churn model below is built using telecom data that has 26.4 million rows, 435 variables, and an event rate of 8.5%. This data has attributes such as demographics, customer billing, technical support, service plan, call pattern usage, equipment, delinquency, and so on.

MODEL FLOW



Display 8: Model Flow

The model is built using HP Logistic Regression and HP Neural network methods. The data is split into 70% training and 30% validation. A few variables are transformed for normal distribution and imputed for missing values where applicable. The HP Variable Selection node is run using a sequential variable selection method that performs supervised and unsupervised selection in sequence. HP Logistic Regression and HP Neural network models (50 and 500 iterations) are run in alongside-the-database mode for massively parallel processing (MPP) using a Teradata appliance. In alongside-the-database mode, data required for analytical processing is retrieved from the database and loaded into memory. This mode is suitable when processing large data sets.

RESULTS

Table 1 shows the HP Neural model run with 500 Iterations as the top-performing model. It took only 5 minutes 30 seconds to run the model with a lift of 8.2%. HP Neural with 50 iterations and HP Regression took only 1 minute 30 seconds to fit the model. The model performance improved when the number of iterations of the model that were run increased, with little increase in run time.

Model	Run Time	Cumulative Lift	Misclassification
HP Neural-500 Iterations	5 Min. 28.25 Sec.	8.20	6.5%
HP Neural-50 Iterations	1 Min. 29.34 Sec.	7.94	6.7%
HP Regression	1 Min. 37.11 Sec.	7.75	6.9%

Table 1: Model Comparison

CONCLUSION

SAS High-Performance Analytics Server adds significant value to your organization. The distributed, in-memory environment enables you to analyze all of your data, quickly add new or more variables, and use advanced analytics techniques and more iterations to build accurate models at breakthrough speed. With a shorter time to run models, more what-if scenarios can be evaluated to get more precise insights. Furthermore, it uses the same easy-to-use interfaces to create high-performance models, eliminating the learning curve for existing users.

APPENDIX

```
data work.simulate;
array _a{8} _temporary_ (0,0,0,1,0,1,1,1);
array _b{8} _temporary_ (0,0,1,0,1,0,1,1);
array _c{8} _temporary_ (0,1,0,0,1,1,0,1);
do pkey=1 to 500000;
x = rantbl(1,0.28,0.18,0.14,0.14,0.03,0.09,0.08,0.06);
a = _a{x};
b = _b{x};
c = _c{x};
x1 = int(ranuni(1)*400);
x2 = 52 + ranuni(1)*38;
x3 = ranuni(1)*12;
lp = 6. -0.015*(1-a) + 0.7*(1-b) + 0.6*(1-c) + 0.02*x1 -0.05*x2 - 0.1*x3;
y = ranbin(1,1,(1/(1+exp(lp))));
output;
end;
run;
```

REFERENCES

[1] SAS Institute Inc. 2013. *SAS High-Performance Analytics Server 12.2: User's Guide.* Cary, NC: SAS Institute Inc.

[2] SAS Institute Inc. 2012. *SAS Enterprise Miner High-Performance Data Mining Node Reference for SAS 9.3, Third Edition.* Cary, NC: SAS Institute Inc.

[3] EMC Corporation. 2011. *Greenplum® Database 4.1 Administrator Guide.*

[4] Teradata Corporation. 2009. *SQL Functions, Operators, Expressions, and Predicates.*

[5] SAS Institute Inc. SAS Institute white paper. "SAS® High-Performance Analytics Server: What Could You Do with Faster, Better Answers?" http://www.sas.com/resources/whitepaper/wp_41948.pdf.

[6] Hughes, Arthur Middleton. "Churn reduction in the telecom industry." *Direct Marketing News*, January 24, 2007. Available at http://www.dmnews.com/churn-reduction-in-the-telecom-industry/article/94238/#.

ACKNOWLEDGMENTS

I would like to thank Mauro Cazzari, David Prevette, Leslie Anderson, and Patrick Maher for their support and valuable input on this paper.

RECOMMENDED READING

Information about SAS/ACCESS® Interface to Teradata and SAS/ACCESS® Interface to Greenplum in *SAS/ACCESS 9.3 for Relational Databases: Reference, Second Edition*.

CONTACT INFORMATION

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

Priyadarshini Sharma
Sr. Associate Analytical Consultant
SAS Institute Inc.
Schaumburg, IL 60173
+1 919-531-9488
Priya.Sharma@sas.com