Session: 3973 – Integrating SAS, Apache Hadoop, and an Enterprise Data Warehouse in a Single Solution

Bob Matsey – Teradata Senior Advanced Analytic Consultant
Agenda

• SAS & Teradata Partnership
• Benefits of In Database
  – Coding Example
• Customer Improvement Examples
• VIYA Integration with Teradata
• Teradata’s UDA
• Agile Analytics with Data Labs
• In-DB Decision management with Decision Manager
• IoT Example – Wearables
• Questions?
The SAS & Teradata Partnership Overview

- Teradata is an Authorized Global Reseller of SAS Solutions
- Partnership began in 2007 to improving analytic performance
- Focus on joint product collaboration and customer success
- More than 450 sales to over 240 customers already
- Teradata has dedicated R&D teams onsite at SAS
- Regular collaboration on Joint Product Roadmap to ensure seamless product integration
Example of In Database with Proc FREQ

Traditional Technique

• Request all rows
• Select state, credit from credit data;
• Calculate frequency count

SQL Pushdown

• Select count(*), state, credit from . . . group by state, credit;
• Return only count

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>SQL Pushdown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows Returned</td>
<td>9,000,000</td>
<td>51</td>
</tr>
<tr>
<td>Time to Process</td>
<td>55 seconds</td>
<td>2 seconds</td>
</tr>
</tbody>
</table>

SAS® Session

Proc Freq;

table state*credit;

SAS/Access to Teradata

SQL

Select count(*), state, credit from . . . group by state, credit;

Return only count
In Database Coding Example

Testing In-database Functionality

Not Running In Database Example: (SQLGENERATION=NONE;) will tell the code to NOT run in database.

Example 1 – Shows running a simple Proc Freq in a SAS program against a larger dataset (at least 1-2 million rows) without in-database capabilities turned on & with SAS log turned on. Then review the SAS log for duration and database performance.

Code Example:

```sas
libname tdXXXX teradata server="XXXserver" database=XXXXP user=&user password=&password;
options sastrace=(,,ds) sastraceloc=saslog nostsuffix;
OPTIONS SQLGENERATION=NONE;
PROC FREQ DATA=tdxxxx.xxxxx;
TABLES XXXX_XXXX;
RUN;
```

Running In Database Example: (SQLGENERATION=DBMS;) Will tell the code to run in database

2nd Example is: Running the same Proc Freq code in a SAS program with the following options: options SQLGENERATION=DBMS. This option says to run the code in database whenever it can, so I highly recommend putting this on ALL your SAS code.

```sas
libname tdXXXX teradata server="XXXserver" database=XXXXP user=&user password=&password;
options sastrace=(,,ds) sastraceloc=saslog nostsuffix;
OPTIONS SQLGENERATION=DBMS DBIDIRECTEXEC set=truncate_bigint 'yes' MSGLEVEL=1;
PROC FREQ DATA=tdxxxx.xxxxx;
TABLES XXXX_XXXX;
RUN;
```

Running these two tests will show,
Example 1 – this will NOT run in database.
Example 2 – will run IN database.
## In-Database Functionality

### SAS/Access to Teradata

**Base Procedures:**
- PROC APPEND
- PROC CONTENTS
- PROC COPY
- PROC DATASETS
- PROC DELETE
- PROC FORMAT
- PROC FREQ
- PROC MEANS
- PROC PRINT
- PROC RANK
- PROC REPORT
- PROC SORT
- PROC SQL
- PROC SUMMARY
- PROC TABULATE

### SAS Code Accelerator for Teradata
- PROC DS2

### SAS Scoring Accelerator for Teradata
- EM/STAT* Models

### SAS Analytics Accelerator for Teradata

**Statistical Analysis Procedures:**
- PROC CANCORR
- PROC CORR
- PROC FACTOR
- PROC PRINCOMP
- PROC REG
- PROC SCORE
- PROC TIMESERIES
- PROC VARCLUS

### SAS Enterprise Miner
- PROC DMDB
- PROC DMINE
- PROC DMREG (Logistic Regression)
- Also nodes for Input, Sample, Partition, Filter, Merge, Expand

### DQ Accelerator for Teradata
- Match code
- Parsing/Casing
- Gender/Pattern/Identification analysis
- Standardization

**PROC SCORE works with coefficients from:**
- PROC ACECLUS
- PROC CALIS
- PROC CANDISC
- PROC DISCRIM
- PROC FACTOR
- PROC PRINCOMP
- PROC TCALIS
- PROC VARCLUS
- PROC ORTHOREG
- PROC QUANTREG
- PROC REG
- PROC ROBUSTREG
<table>
<thead>
<tr>
<th>#</th>
<th>Process Name</th>
<th>SAS + Oracle</th>
<th>SAS + 2 Node Teradata</th>
<th>X Faster</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Horizontalization</td>
<td>18 hrs 7 mins</td>
<td>32 mins</td>
<td>34 X</td>
</tr>
<tr>
<td>2</td>
<td>Horizontalization</td>
<td>15 hrs 3 mins</td>
<td>33 mins</td>
<td>27 X</td>
</tr>
<tr>
<td>3</td>
<td>Variable Calculation</td>
<td>6 hrs 57 mins</td>
<td>4 mins</td>
<td>104 X</td>
</tr>
<tr>
<td>4</td>
<td>Scoring</td>
<td>10 hrs 56 mins</td>
<td>11 mins</td>
<td>60 X</td>
</tr>
<tr>
<td>5</td>
<td>Data Mart Generation</td>
<td>27 hrs 50 mins</td>
<td>1 hour 28 mins</td>
<td>19 X</td>
</tr>
</tbody>
</table>
SAS Programs Results

- **Highlights**
  - **GE** – long running queries with sort
    - Execution in Teradata only took 3.75 minutes – 1600X – Old way 103 hours!
  - **OSCAR** – running against Commercial Market Scan data
    - Execution in Teradata was 1 hour 50 minutes against 3 times larger data set – Old way 231 hours

<table>
<thead>
<tr>
<th>#</th>
<th>Business Line</th>
<th>SAS Log Name</th>
<th># of Steps</th>
<th>Days</th>
<th>Hours</th>
<th>Minutes</th>
<th>Days</th>
<th>Hours</th>
<th>Minutes</th>
<th>% of SAS Only</th>
<th>X Times Faster</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>oscar</td>
<td>oscar_mdcd_v3.log</td>
<td>945</td>
<td>9.6</td>
<td>231.6</td>
<td>13,894.1</td>
<td>1.83</td>
<td>110.0</td>
<td>1%</td>
<td>126.3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GE</td>
<td>mk_text_observation_f_sort.log</td>
<td>3</td>
<td>4.3</td>
<td>103.0</td>
<td>6,178.0</td>
<td>3.8</td>
<td>0%</td>
<td>1,625.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ingenix</td>
<td>dcf ~ i3 qc.log</td>
<td>3,401</td>
<td>15.1</td>
<td>908.2</td>
<td></td>
<td>45.8</td>
<td>5%</td>
<td>19.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>humana</td>
<td>humana_dups.log</td>
<td>28</td>
<td>5.6</td>
<td>333.3</td>
<td></td>
<td>18.8</td>
<td>6%</td>
<td>17.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ingenix</td>
<td>analysis ~ 100_indentifying_initial_patients.log</td>
<td>12</td>
<td>1.7</td>
<td>99.4</td>
<td></td>
<td>1.5</td>
<td>2%</td>
<td>66.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ingenix</td>
<td>analysis ~ 200_extracting_mx_claims.log</td>
<td>11</td>
<td>1.1</td>
<td>68.1</td>
<td></td>
<td>1.0</td>
<td>1%</td>
<td>68.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ingenix</td>
<td>analysis ~ 210_extracting_rx_claims.log</td>
<td>12</td>
<td>1.1</td>
<td>68.1</td>
<td></td>
<td>1.0</td>
<td>1%</td>
<td>68.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ingenix</td>
<td>dcf ~ mk_s2009_r12q2.log</td>
<td>20</td>
<td>1.6</td>
<td>98.2</td>
<td></td>
<td>3.8</td>
<td>4%</td>
<td>25.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ingenix</td>
<td>dcf ~ mk_s2010_r12q2.log</td>
<td>20</td>
<td>1.5</td>
<td>87.8</td>
<td></td>
<td>3.6</td>
<td>4%</td>
<td>24.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ingenix</td>
<td>dcf ~ mk_s2011_r12q2.log</td>
<td>20</td>
<td>1.0</td>
<td>61.8</td>
<td></td>
<td>3.4</td>
<td>6%</td>
<td>18.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ingenix</td>
<td>dcf ~ mk_m2011_r12q2.log</td>
<td>20</td>
<td>1.5</td>
<td>87.8</td>
<td></td>
<td>3.6</td>
<td>4%</td>
<td>24.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ingenix</td>
<td>dcf ~ mk_r2011_r12q2.log</td>
<td>20</td>
<td>1.0</td>
<td>61.8</td>
<td></td>
<td>3.4</td>
<td>6%</td>
<td>18.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>pharmetrics</td>
<td>130_af_all_claims.log</td>
<td>12</td>
<td>1.7</td>
<td>101.2</td>
<td></td>
<td>4.7</td>
<td>5%</td>
<td>21.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>pharmetrics</td>
<td>110_af_claims.log</td>
<td>6</td>
<td>52.0</td>
<td>2.7</td>
<td>19.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>pharmetrics</td>
<td>183_table8d.log</td>
<td>43</td>
<td>30.8</td>
<td>3.4</td>
<td>11%</td>
<td></td>
<td></td>
<td></td>
<td>9.1</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>pharmetrics</td>
<td>183_table8b.log</td>
<td>39</td>
<td>30.4</td>
<td>1.5</td>
<td>5%</td>
<td></td>
<td></td>
<td></td>
<td>20.3</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>pharmetrics</td>
<td>182_table2b.log</td>
<td>30</td>
<td>20.6</td>
<td>2.6</td>
<td>13%</td>
<td></td>
<td></td>
<td></td>
<td>7.4</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>pharmetrics</td>
<td>182_table8d.log</td>
<td>43</td>
<td>23.8</td>
<td>1.8</td>
<td>8%</td>
<td></td>
<td></td>
<td></td>
<td>13.2</td>
<td></td>
</tr>
</tbody>
</table>
Agile Analytics – Integrating Data into a Single Solution
Dealing with All Types of Data

Enabling Self Service Data Loading & Analytics with a Teradata’s Data Labs
Business Need for Agile Analytics

Flexibility vs. IT Process

• Analyze quickly
  – Test New Theories
  – New Data

• Does the new data provide additional insight?

• Does the new insight cause a change in thinking or direction?

• Test Fast
  – Was the theory right? (Success or Failure)

• Productionize what works; discard what doesn’t!
  – Add the new application
  – Add the new data
  – Or delete and move on!
Don’t Just Use Production Data – Evolve It

3rd Party Data
- Often rented, supplier data and/or format needs to change, value needs validation, only applies to some projects

Temporary & Research Data
- Exploratory metrics and aggregates, requirements not fully defined, short lived, early stage work

Pre-Production Data & Prototypes
- Excel Spreadsheets
- Oracle, SQL Server, SAS datasets, Access DB, others can be loaded
- Comma delimited, space delimited, other data types
Teradata Data Labs Architecture

Analytic Sandboxes with Governance

- Data Lab(s) inside the EDW or DW Appliance to easily join to production data via Views
- Load experimental, untested data from external sources
- Rapid prototyping, exploratory and experimentation analysis
- Beyond a Sandbox
  - An architecture that enables governance
    ✓ Works within your current data warehouse environment
  - Data lab portlets for IT and Business analyst
    ✓ Self-provisioning system that simplifies implementation, management and use

R, Python, SPSS, SAS, SQL

- SAS data
- csv data
- Hadoop data

Active Workload Management

Data Labs

Teradata Database

External Data
Teradata Data Lab Hierarchy

Data Lab Objects

Data Lab hierarchy to manage user groups, space, and workload

**Database**
- Database where the lab group resides
  - Normal Teradata user database

**Lab Group**
- Workspace allocated for a group of users to create their own data labs.
  - Groups can be arranged by department or project
  - Groups can be made private
  - Lab Group is a fixed size that's shared by users.

**Data Labs**
- Workspaces allocated for analysis
  - Can be for a single user or X number of users
  - Data Labs expire
  - Data Labs are allocated with a fix size, but are elastic

**Table**
- Database table to store the data
  - User can create table and load data
Example: Lab Group Hierarchy

Viewpoint

- **Marketing Lab Group**
  - Campaign Lab
  - Promotion Lab

- **Sales Lab Group**
  - Demand Curve Lab
  - Sales Forecast Lab
  - Cust Retention Lab

- **Data Scientist Lab Group**
  - Risk Analytics Lab
  - Customer Segmentation Lab
  - Helen’s Personal Lab

*Teradata Database*
SAS is Built into the Teradata Analytics Platform

Teradata’s strategy is to allow the customer to choose the tools they want.
QUESTIONS ???