SAS® In-Memory Analytics for Hadoop
SAS® In-Memory Analytics for Hadoop
Venkateswarlu Toluchuri, United Health Group (Optum), Hyderabad, India

The Optum solution uses SAS’s Fraud Framework and Optum’s deep health care expertise and extensive health care claims and fraud case datasets to identify and prevent instances of fraud, waste and abuse for payers. The solution delivers broad detection capabilities including rules, flags, predictive modeling, text mining and SAS Visual Analytics to identify possible instances of provider and consumer fraud, including multi-party fraud schemes and organized crime.

Challenges
- Understand LASR Server architecture
- Data flow in to LASR Server
- Different techniques of loading data in to SAS LASR
- Understand the analytics life cycle process in SAS In-Memory
- Different statements in PROC IMSTAT

LASR and Hadoop
The LASR Analytic Server integrates with Hadoop by reading and writing SAS data in SASHDAT format in the Hadoop Distributed File System (HDFS).

Types of SAS® In-Memory Analytics Products
- SAS High Performance Products (Procedure Language Interface)
  - Statistics
  - Econometrics
  - Text Mining
  - Optimization
- SAS LASR Analytic Server Products (Point and Click web applications)
  - SAS Visual Analytics
  - SAS Visual Statistics
  - SAS In-Memory Statistics

Parallel (Asymmetric) Data Load
- The data is not co-located.
- LASR table blocks exist on dedicated hardware while the asymmetric provider table blocks exist on separate hardware.
- The blocks are pushed from the data provider into LASR just like with co-located data except that, they travel across a dedicated network.
- The number of provider nodes does not have to be equal to the number of LASR nodes (thus the term, asymmetric)
- Data does not pass through the LASR Head node for distribution. The blocks are pushed straight from the provider into the LASR worker nodes.
- The mapping algorithm, that maps blocks to worker nodes, is extremely simple and tries to distribute the blocks as evenly as possible.
- The SAS Embedded Process (EP) must be installed on the parallel data provider.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.
Paper 1271 - 2017

SAS® In-Memory Analytics for Hadoop

Venkateswarlu Toluchuri, United Health Group (Optum), Hyderabad, India

Data Load Engine Techniques

**SAS®**
- All data sources that a SAS procedure or a data step can be loaded into a LASR Analyst Server by this technique. This means that any data source is supported by Hadoop, Spark, SQL, etc.
- Data loading via SAS® Table also happens in parallel. It is thus not an efficient way to load big data. However, when data loading via SAS® Table is used, performance can be improved.

**SASHIQ**
- Data stored permanently in HDFS in the distributed data nodes. Data read is in memory in parallel by SAS® High Performance procedures and released from memory when completed execution.
- SAS® Hadoop is a proprietary SAS format that high performance procedures and PROC LASR can read. To understand SAS® Hadoop, let's consider a SAS saved in distributed files and distributed across a multi-machine Hadoop file system. Each block contains a subset of the data in a single host and metadata information that allows the SAS® Analyst Server to read it efficiently.
- The SAS® Hadoop engine deals with SAS® Hadoop and CSV files. The engine is uni-directional for data; it does not load data from a SAS® Hadoop file back to the SAS session. Data moves only from the SAS® Hadoop session into the SAS® Engine.

**HDFS**
- Data stored permanently in HDFS in the distributed data nodes. Read and write HDFS tab files exactly like SAS® datasets.
- Load BASE SAS table in SAS® Server.
- Binary files have a name ending with “.sasx” and load them.

**Advanced LASR Loading Techniques Comparison**

<table>
<thead>
<tr>
<th>COMPARISON</th>
<th>BASE SAS</th>
<th>PROC LASR</th>
<th>PROC NSI52</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Append</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>HDAT Source</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Filter, Transform HDAT</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Table Name and Tag</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Join, Merge</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Distributed Processing</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>HDAT Memory Mapping</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

Data in Memory tables can be partitioned. A partition is a set of data that shares the same key. In the LASR Analyst Server, a partition implies that all observations in the partition reside on the same node.

Partitions are constructed based on the formatted values of the partition variable. If you partition using multiple variables, the server constructs a single character key on the values.

How to Create a Partitioned Table:

- Partitioned tables can be created using PROC PARTITION—data step option.
- By loading a SAS® Hadoop file that has been previously partitioned.

PARTITION statements create temporary partitioned tables which can be promoted to memory.

 joven, merges are supported in the PARTITION options.

Proc IMSTAT Statements

- **BASE SAS**
- **PROC LASR**
- **PROC NSI52**

**Partition**
- **YES**
- **NO**

**Append**
- **YES**
- **NO**

**HDAT Source**
- **NO**
- **YES**
- **NO**

**Filter, Transform HDAT**
- **NO**
- **YES**

**Table Name and Tag**
- **YES**
- **NO**

**Join, Merge**
- **YES**
- **NO**

**Distributed Processing**
- **YES**
- **NO**

**HDAT Memory Mapping**
- **NO**
- **YES**

**Analytics Life Cycle support SAS In-Memory**

**Prepare Data**
- Access structured, unstructured data
- Data filtering

**Explore Data**
- Access structured, unstructured data
- Data filtering

**Develop Models**
- Join tables, promote tables, compute columns
- Group filtering, partitioning, data ordering within partitions

**Scoring**
- Explore multiple modeling approaches using advanced analytical and machine learning algorithms
- Combine structured and unstructured data for predictive analytics
- Classification, predictions

- Integration with SAS® Model Manager for model management and model performance monitoring
- Generation of SAS® data step code to execute model on new data

**Proc IMSTAT Statements**

- **BASE SAS**
- **PROC LASR**
- **PROC NSI52**

**Data Manipulation**

- **Data Exploration/Visualization**

- **Modeling**

- **Model Deployment**

- **Miscellaneous**
  - FREE
  - SAVE
  - STORE

- **Score**
  - CODE