Teradata Aster Analytics – Apache Spark Connector

Introduction

Apache Spark™ is becoming a popular open source big data computing and processing framework among the advanced analytics community today. Spark was built to ensure the delivery of speedy and sophisticated analytics with a limited set of algorithms for in memory computing. But, powerful as it is, Spark is still confined to those with advanced data science skills. Teradata's Aster Analytics™ is now delivering a connector that combines Spark algorithms with Aster algorithms, R packages, and other languages (e.g., Python, Java) to deliver multi-genre analytics at scale to a wide variety of advanced analytics professionals. The impact of the Teradata Aster Analytics— Apache Spark connector (henceforth Aster—Spark connector) will be seen in a more rapid adoption of a data driven culture to decision making that leverages Aster as the analytical backbone to support multiple personas, languages, and apps that ensures analytic models are rapidly made production ready.

Importance of Spark

The core Spark engine functions partly as an application programming interface (API) layer and underpins a set of related tools for managing and analyzing data, including a SQL query engine, a library of machine learning algorithms, a graph processing system and streaming data processing software. Spark's developers say it can run jobs 100 times faster than MapReduce when processed in memory and 10 times faster on disk. Today, algorithms that are part of Spark are implemented to address critical use cases on the Internet of Things (IoT), Network Security, Fraud Prevention, and more. All that said, Spark is significantly limited to its adoption primarily because it is not easy to use as well as having a limited suite of Machine Learning only algorithms. Analysts with only a high degree of proficiency in advanced programming languages such as Scala and Python use Spark. Furthermore, Spark limitations including time consuming error troubleshooting processes and the need to allocate memory prior to the implementation of an analytic step further exacerbates the difficulty in implementing Spark technology. Enterprise security is not natively available

Key Benefits

- Multi-genre analytics, which includes text analytics, machine learning, path analytics, graph analytics and more, can be executed in a single analytic solution

- Spark analytic functions (e.g., deep learning) can be called along with other Aster analytic functions from the Aster environment

- Bridges the skills gap by making analytics available for implementation, operationalization, and production ready by a wide range of personas—data scientists, business analysts, business scientists, programmers, lines of business managers, and C-suite executives

- Bridges the resources gap that companies face while hiring hard to find and expensive analytics resources

- Maximizes the return on investment through a broader consumption of advanced analytics that results in timely market interventions that positively affect a business' key performance indicators

- For Aster users the analytics framework is extended to include powerful Spark analytics capabilities without any of the additional effort that is normally expended in learning Scala.

- For Spark users the ability to truly mix and match analytic techniques in intelligent combinations in a highly scalable solution built for high performance is delivered

and only provided by partner solutions. What is required for analysts to be successful in delivering timely resolutions of critical business problems is the idyllic merger of a powerful computing framework like Spark with a multi-genre analytics solution, with enterprise features, that results in analytic speed, scale, and agility. This is delivered by the Aster-Spark Connector.
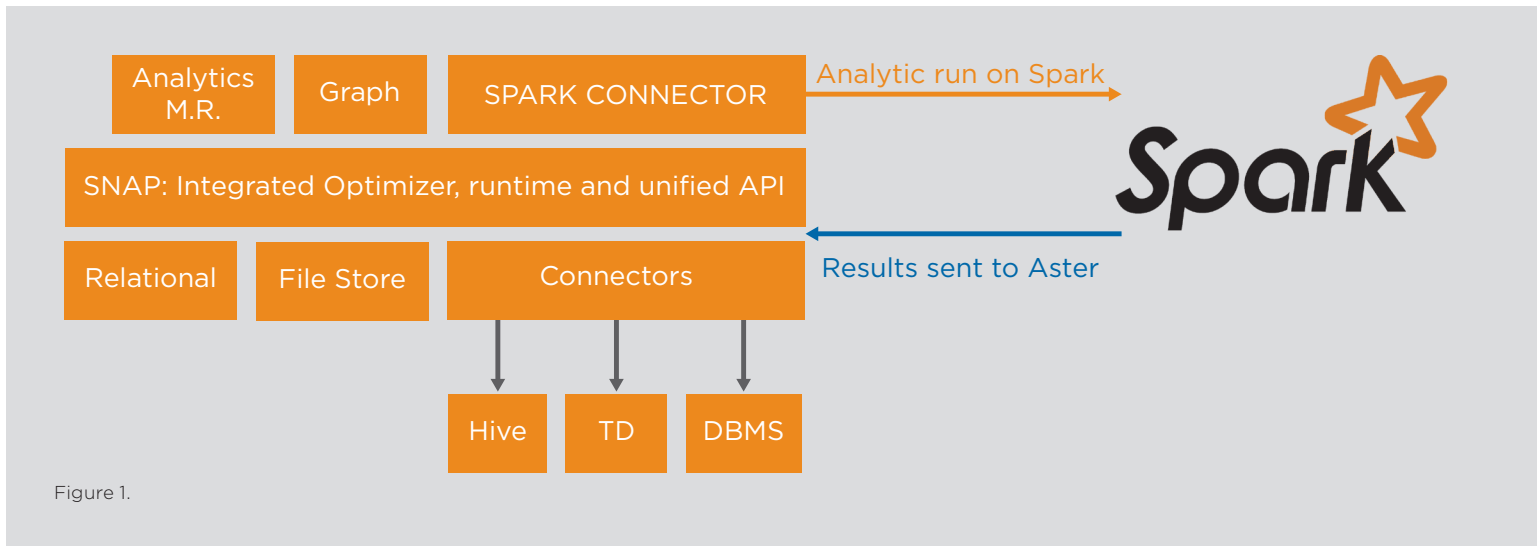
TERADATA

Figure 1.

Aster-Spark Connector

The Aster-Spark Connector is a SQL-MapReduce™ statement called from Aster that executes Spark functions from the Spark Library. Analytics professionals can now run Spark functions and scripts as an Aster query operator. The growing set of Spark advanced analytics functions (e.g., Deep Learning algorithms) can be executed on the Spark cluster and the results of that execution can be sent back as inputs to Aster for further analysis.

Users can write new functions in Spark utilizing its popular API and all these can also be included as part of the implementation from Aster and can be combined with other analytic models built in R. All multi-genre analytics would run in Aster and are operationalized at scale to become production ready.
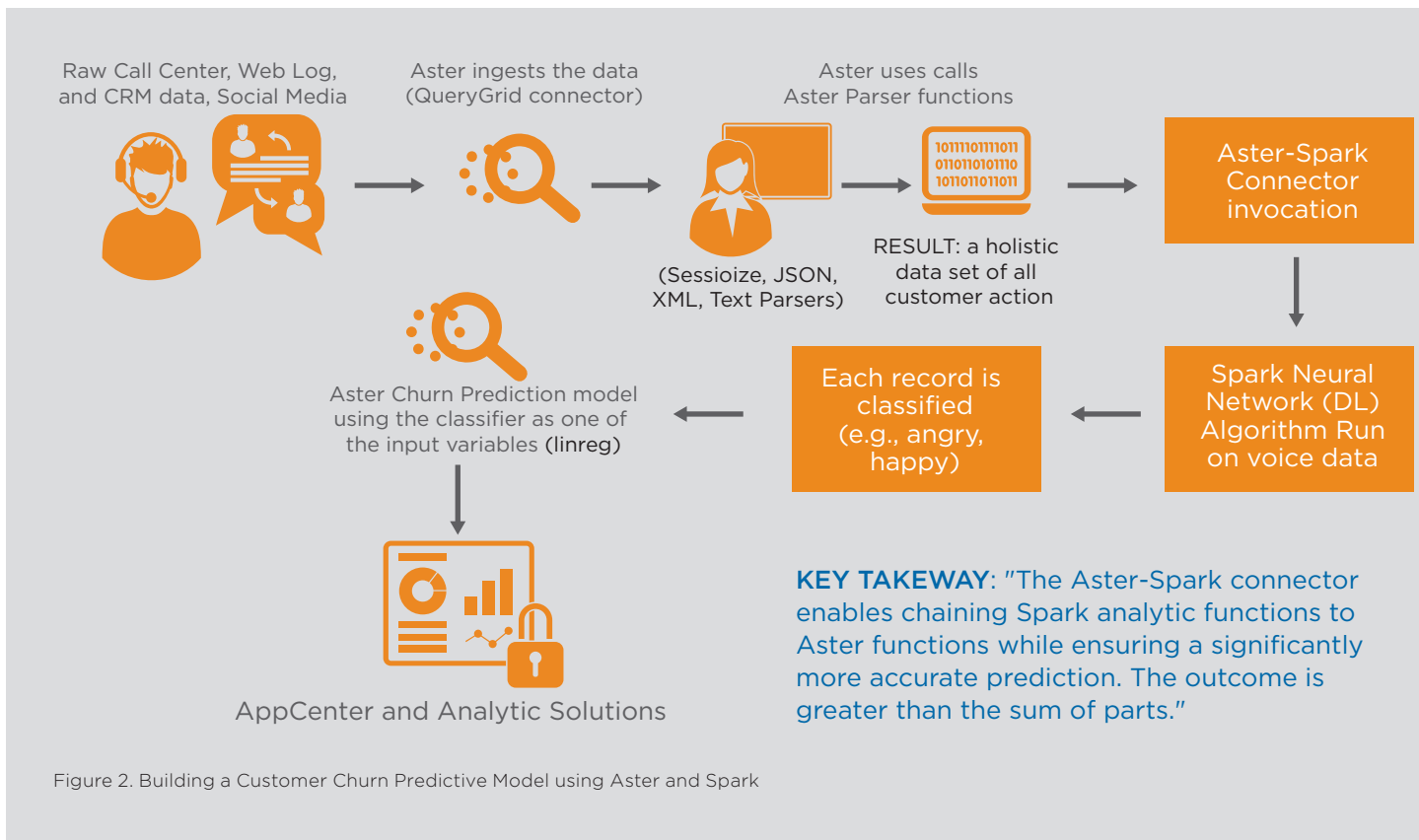
Multi-Genre Analytics with Aster and Spark

Most big data solutions today require highly specialized and expensive data science skills to design, build, and maintain their analytics solutions. Now add to it the additional complexity of maintaining a Spark cluster and executing advanced analytics algorithms on it. This difficulty is further compounded, given Spark's focus on Machine Learning algorithms, by having to shoehorn analytic results obtained on Spark with results obtained in solutions. However, all these issues are easily addressed with the Aster-Spark connector. The Aster-Spark connector allows the user to call any of the 40+ ML Lib functions from Spark and these could be used alongside any of the 100+ advanced analytics functions found on Aster Analytics in addition to all the R packages. This is truly a multi-genre analytics implementation in that almost all use cases can now be addressed by intelligently mixing functions from two powerful solutions without the hassle of learning a new programming language such as Scala or Python.

Important Sample Use Cases covered by the Aster-Spark Connector

- Smart grid management
- Financial fraud identification
- Compute network security
- Retail product clustering
- National security threat risk assessment
- Smart cities planning

TERADATA

Figure 2. Building a Customer Churn Predictive Model using Aster and Spark

The flow diagram contains the following labeled elements:

- Raw Call Center, Web Log, and CRM data, Social Media
- Aster ingests the data (QueryGrid connector)
- Aster uses calls Aster Parser functions
- (Sessioize, JSON, XML, Text Parsers)
- RESULT: a holistic data set of all customer action
- Aster-Spark Connector invocation
- Spark Neural Network (DL) Algorithm Run on voice data
- Each record is classified (e.g., angry, happy)
- Aster Churn Prediction model using the classifier as one of the input variables (linreg)
- AppCenter and Analytic Solutions
- KEY TAKEWAY: "The Aster-Spark connector enables chaining Spark analytic functions to Aster functions while ensuring a significantly more accurate prediction. The outcome is greater than the sum of parts."

Conclusion

The Aster–Spark connector enables business users with less experience in Spark to tap the memory based spark system for deep and machine learning analytics and then operationalize in Aster. It is an industry first integration of Spark Analytics with Aster Analytics in which pre-built analytics functions from both solutions can be executed from Aster without the need for the burdensome acquisition of new technical capabilities. Furthermore, depending on the use case being addressed, analytical functions from either solution can be intelligently combined in a multi-genre environment to deliver highly impactful insights for the business. Customers benefit from greater analytical precision and the ability to seamlessly move from an analytical environment to an operational one.

For More Information

To find out how the Teradata Aster Analytics solution can help you take advantage of big data volumes in a fast, efficient and cost effective way while you improve your decision-making capabilities and grow a stronger, more productive business please contact your local Teradata representative or visit **Teradata.com**. Join the Aster Community at **Aster-Community.Teradata.com** to learn about Aster Analytics.