ORAAH 2.7.1 Change List Summary

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

ORAAH 2.7.1 release marks another major step to bring Spark-based high performance analytics to the Oracle R Advanced Analytics platform. ORAAH continues to support legacy Hadoop mapReduce based analytics as well in the 2.x product releases, but we encourage our customers to move their code base to use the Spark versions (when available) of same functionality.

1 ORAAH 2.7.1 Changes

1.1 ORCHcore

ORCHcore includes a number of bug fixes and code improvement as lists below.

1.1.1 Non backward-compatible API changes:

- None.

1.1.2 Backward-compatible API changes:

- spark.connect()

`spark.connect()` has a new logical parameter *disconnect*. If set to `TRUE`, it automatically disconnects from the existing spark session, before initiating the new Spark session. If there is an active Spark connection and *spark.connect()* is invoked with `disconnect=FALSE` (the default), an error will be reported and the active connection will not be disconnected.

- orch.connect()

ORAAH can now connect to Oracle databases running in Multitenant Container Database (CDB) mode. A new parameter *pdb* can be used to specify the service name of the Pluggable database (PDB) to which the connection has to be established. Note, parameters *sid* and *pdb* are mutually exclusive and an appropriate error will be reported if both are specified.

- hdfs.fromHive()

`hdfs.fromHive()` can now materilize Hive *ore.frame* objects created out of queries, using `ore.sync(query=c("<ore.frame name>"="<HiveSQL query>")))`. A warning that reports the table name of the materilized query is reported.

- hdfs.toHive()

When `hdfs.toHive()` changes column names to make them Hive compatible, these changes will be reflected in metadata within HDFS too. This resolves an issue that caused analytics to fail due to mismatch in column names present in the formula.

- hdfs.meta()

`hdfs.meta()` now does not allow column name irregularities caused by non-unique names for the HDFS data. If non-unique names are specified, an error is reported.

- orch.sample()

A parameter logical *overwrite* has been added to the `orch.sample()` function. If TRUE, the output HDFS directory or Hive table will be removed and overwritten by the new orch.sample output. It also reports an error if *size* is specified when sampling a Hive table. Currently, only *percent* sampling is supported for Hive tables.

1.1.3 New API functions:

- None.

1.1.4 Other changes:

- ORCH_CLASSPATH environment configuration variable for ORAAH.

ORAAH now supports a new environment configuration variable ORCH_CLASSPATH. You can set the CLASSPATH used by ORAAH using this variable. ORCH_CLASSPATH can be set for the shell or in *Renviron.site* file to make it available for R and ORAAH.

- Wildcard support in CLASSPATH and ORCH_CLASSPATH.

rJava does not support wildcard characters in CLASSPATH environment variable. This lead to users adding exact jar files in the CLASSPATH before they could use Apache Spark analytics. Now ORAAH resolves this issue and will support the wildcards in path. For example, having a path */usr/lib/hadoop/lib/*.jar* in CLASSPATH adds all jars from */usr/lib/hadoop/lib* to rJava CLASSPATH. This makes the process of getting started with ORAAH's Spark features easier and faster. You can use wildcards with ORCH_CLASSPATH too, if you are going to use ORCH_CLASSPATH instead of CLASSPATH for ORAAH configuration.

- Compatible with New R.

ORAAH 2.7.1 packages have been built for R version 3.3.0.

- MIT License version of RJDBC.

ORAAH ships with the new MIT licensed version of the RJDBC package.

- Improved demos.

All demos in ORCH which can be listed by `demo(package="ORCH")` have been improved and made such that they use random generated table names. They also do a fine cleanup after the demo finishes.

- Improved install and uninstall scripts for client.

The install and uninstall scripts for ORAAH client have been made smarter than before. The client installer now checks if all the prerequisites are met before beginning to install ORAAH packages and libraries. If also runs small tests to verify that ORAAH library can be loaded in R and if HDFS is accessible. The user can choose to skip these checks by specify the "-f" option to the *install-client.sh*. The *uninstall-client.sh* now checks for existence of *ORE* package on the cluster node. If found, the user is presented with the option to keep ORE dependencies (OREbase, OREstats, OREserver, OREcommon and OREembed). So if an ORAAH client is also being used as an ORE client, ORE will remain usable after uninstalling ORAAH from that node. Also, in force mode (option "-f") the packages will be removed from all library paths present on the client node. This results in a clean system without any ORAAH presence.

- Improved uninstallation from cluster.

The uninstall script for ORAAH server has been made smarter than before. If you are using a cluster node both as a Hadoop node and as a client node, then in order to uninstall ORAAH, you must run *uninstall-client.sh* on this node after running *uninstall-server.sh* from any cluster node. Earlier, users had to use the force option “-f” for *uninstall-client.sh* script. Because the client and server side of ORAAH include some of the same files, the *uninstall-client.sh* script and *uninstall-server.sh* would both try to remove this common subset of files along with all of the others. This caused the second uninstall script to fail simply because it could not find files that were already deleted. Using “-f” ensured that the script continued. This problem has been resolved and no errors will be reported in the *uninstall-client.sh* script if it detects the library or package was removed by *uninstall-server.sh* before. If you are using multiple nodes of the cluster as ORAAH client nodes then simply run *uninstall-client.sh* on all such nodes. No false errors will be reported after *uninstall-server.sh* was run once on the cluster.

1.2 ORCHstats

With the ORAAH 2.6.0 release we had introduced support for selected algorithms from Apache MLlib with support from our proprietary optimized data transformation and data storage algorithms. This interface is built on top of Apache Spark and designed to improve performance and interoperability of MLlib and ORAAH machine learning functionality with R. Building on the initial success of the Apache MLlib integration, ORAAH 2.7.0 expanded coverage of exported MLlib algorithms and greatly improved support for the R formula engine.

Machine learning and statistical algorithms require a Distributed Model Matrix (DMM) for their training phase. For supervised learning algorithms, DMM captures a target variable and explanatory terms; for unsupervised learning DMM captures explanatory terms only. Distributed Model Matrices have their own implementation of the R formula, which closely follows the R formula syntax, but is much more efficient from the performance perspective. See the reference manual for detailed information about features supported in the Oracle formula.

ORCHstats includes a number of bug fixes and code improvement as lists below.

1.2.1 Non backward-compatible API changes:

- None.

1.2.2 Backward-compatible API changes:

- example(orch.ml.linear) & example(orch.ml.lasso)

The example scripts for `orch.ml.linear()` and `orch.ml.lasso()` have been updated to examples that converge. Earlier these examples did not converge on Spark MLlib and the coefficients were NaNs.

- summary() for orch.ml.gmm and orch.ml.kmeans models

When `summary()` is invoked for Spark Mllib GMM and K-means models, error is reported instead of useless information, since summary for such models is not supported by MLlib.

- hdfs.write()

`hdfs.write()` now supports writing Spark Dataframes to HDFS.

1.2.3 New API functions:

- None.

1.2.4 Other changes:

- help("orch.neural2")

Additonal documentation for `orch.neural2()` has been added in this release. With orch.neural2, the model building and scoring are completely implemented with Spark. It no longer relies on any Hadoop MapReduce implementation. Since the codebase involves R, C and Java, we redesigned the code to connect R and Java using the rJava package, and connect Java with C using JNI. The core calculation remains in C while the main iteration loop stays in Java. Like orch.lm2 and orch.glm2, the new function uses the RDD-based distributed model matrix and the Oracle R formula parser.

1.3 OREbase

OREbase is now includes better support for Hive, especially on BDA secure cluster with enabled SSL connection and Keberos authentication. OREbase also includes a number of bugfixes and improvements across the codebase. See below for more details.

1.3.1 Non backward-compatible API changes:

- None.

1.3.2 Backward-compatible API changes:

- ore.create()

The performance of `ore.create()` for Hive has been improved significantly in this release. Also, users can create a table in another Hive database instead of the one they are connected to, if needed. For example, if user is connected to *default* database and wants to create a table *sample* in *hivedb1* database, the ore.create *table* parameter should be "hivedb1.sample".

A new parameter *append* for `ore.create` lets you append an ore.frame or data.frame to an existing Hive table specified by the *table* parameter. If the table does not exist, and `append=TRUE`, the warning is reported and a new table is created.

1.3.3 Other changes:

- Automated Hive JDBC driver lookup.

From release 2.7.0, ORAAH started checking for the Hive and Hadoop libraries at certain known locations for different type of Hive installations like RPM, or Cloudera Parcel. These locations are scanned and the relevant java libraries are added to the CLASSPATH for the driver when initiating the RJDBC connection. If you have Hive running from a custom install location you can specify environment variables ORCH_HADOOP_HOME (or default HADOOP_HOME) and ORCH_HIVE_HOME (or default HIVE_HOME) and the required libraries will be picked from there. Now `ore.connect()` extends support to older versions of Hive by looking up additional older dependencies. Also, if any jar file is missing, then an appropriate warning message is reported.

2 Copyright Notice