Package ‘RImpala’

October 13, 2014

Version 0.1.5

Date 2014-10-12

Title R and Impala

Author Vijay Raajaa, Austin Chungath Vincent, Sachin Sudarshana, Vikas Raguttahalli

Maintainer Vijay Raajaa <vijay.raajaa@mu-sigma.com>

Contact Austin Chungath Vincent <austincv@gmail.com>,Vikas Raguttahalli <vikas.r@mu-sigma.com>, Sachin Sudarshana <sachin.sudarshana@gmail.com>

Description RImpala facilitates the connection and execution of distributed queries using Cloudera Impala, which is a massively parallel processing (MPP) SQL query engine that runs natively in Apache Hadoop. Impala supports jdbc integration which RImpala utilizes to establish the connection between R and Impala. Thanks to Mu Sigma for their continued support throughout the development of the package.

Depends R (>= 2.7.0), rJava (>= 0.5-0)

SystemRequirements Java (>= 1.5)

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-13 08:05:07

R topics documented:

`RImpala-package` *A Package to interface R and Impala*

Description

RImpala-package contains the R functions required to connect, execute queries and retrieve back results from Impala. It uses the rJava package to create a JDBC connection to any of the impala servers running on a Hadoop Cluster.

Details

Package:	RImpala
Type:	Package
Version:	1.0.0
Date:	2013-09-06
License:	file LICENSE

Installation

RImpala uses the JDBC drivers provided by Cloudera Impala. We need to install them before we can use the RImpala package. Cloudera provides the JBDC jars on their website that can be downloaded directly.

There are two ways to do this:

A. If you have Cloudera Impala installed on the machine running R then you will have the necessary JDBC jars already in place (probably in "/usr/lib/impala/lib") and you can use them to initiate the connection to Impala.
B. If the machine running R is a different server than the Impala server then you need to download the JDBC jars from **https://downloads.cloudera.com/impala-jdbc/impala-jdbc-0.5-2.zip** or from the server running Impala and extract it to a location that can be accessed by the R user.

After you have installed the JDBC drivers you can start using the RImpala package.
Have a look at `rimpala.init` and `rimpala.connect` to establish connection to Impala.

Author(s)

Vijay Raajaa <vijay.raajaa@mu-sigma.com>,
Austin Chungath Vincent <austin.cv@mu-sigma.com>,
Vikas Raguttahalli <vikas.r@mu-sigma.com>,
Sachin Sudarshana <sachin.sudarshana@mu-sigma.com>

References

http://www.cloudera.com/content/cloudera/en/products/cdh/impala.html - Cloudera's page on Impala

rimpala.close *Function to close the JDBC connection to Impala*

Description

This function closes a sucessful connection to Impala-server

Usage

```
rimpala.close()
```

Value

"Connection Closed" is displayed on the console when the JDBC connection is successfully closed

Author(s)

Vijay Raajaa <vijay.raajaa@mu-sigma.com>,
Austin Chungath Vincent <austin.cv@mu-sigma.com>,
Vikas Raguttahalli <vikas.r@mu-sigma.com>,
Sachin Sudarshana <sachin.sudarshana@mu-sigma.com>

Examples

```
## Not run:
library(RImpala)
rimpala.init()
rimpala.connect(IP="127.0.0.1",port="21050")
rimpala.close()

## End(Not run)
```

rimpala.connect *Establishes a JDBC connection to a machine running Impala*

Description

This function creates a connection to the impalad daemon running on a machine in a Hadoop Cluster. The IP of the machine and the port on which the impalad daemon is running is passed as an argument.

Usage

```
rimpala.connect(IP="localhost",port="21050",principal="noSasl")
```

Arguments

`IP`	The IP of the machine to which the connection needs to be established. Default value is localhost
`port`	The port on the machine where the Impala daemon is running. Default value is 21050
`principal`	The principal to use if you require Kerberos authentication.The principal must be the same user principal you used when starting Impala. For example: "impala/myhost.example.com@H2.EXAMPLE.COM". Default value is "noSasl"

Value

"Connection Established" is displayed on the console upon successful connection.

Author(s)

Vijay Raajaa <`vijay.raajaa@mu-sigma.com`>,
Austin Chungath Vincent <`austin.cv@mu-sigma.com`>,
Vikas Raguttahalli <`vikas.r@mu-sigma.com`>,
Sachin Sudarshana <`sachin.sudarshana@mu-sigma.com`>

Examples

```
## Not run:
library("RImpala")
rimpala.init()
rimpala.connect(IP="127.0.0.1",port="21050")
rimpala.close()
rimpala.connect(IP="localhost",port="21050",principal="impala/myhost.example.com@H2.EXAMPLE.COM")

## End(Not run)
```

rimpala.describe *Function to describe any table present in Hive's metastore*

Description

This function runs the describe query of Impala against the table passed as an argument to the function

Usage

```
rimpala.describe(table)
```

Arguments

`table`	The name of the table that needs to be described

Value

Returns an dataframe that contains the details of the table as displayed by the describe command

Author(s)

Vijay Raajaa <vijay.raajaa@mu-sigma.com>,
Austin Chungath Vincent <austin.cv@mu-sigma.com>,
Vikas Raguttahalli <vikas.r@mu-sigma.com>,
Sachin Sudarshana <sachin.sudarshana@mu-sigma.com>

Examples

```
## Not run:
library("RImpala")
rimpala.init()
rimpala.connect("127.0.0.1","21050")
des=rimpala.describe(table="sample_table")

## End(Not run)
```

rimpala.init *Adds the folder containing the jars for Impala in the Classpath*

Description

Initializing the package by adding the required jars to the Classpath

Usage

```
rimpala.init(impala_home=NULL,libs="/usr/lib/impala/lib")
```

Arguments

`impala_home` The home folder of Impala. Default is NULL

`libs` The directory in which the jars required for establishing a connection to Impala are required Default path is "/usr/lib/impala/lib"

Details

This should be the first function that should be executed once the RImpala package is installed and loaded

Value

"Classpath added succesfully" is displayed on the addition of a valid path.

Author(s)

Vijay Raajaa <`vijay.raajaa@mu-sigma.com`>,
Austin Chungath Vincent <`austin.cv@mu-sigma.com`>,
Vikas Raguttahalli <`vikas.r@mu-sigma.com`>,
Sachin Sudarshana <`sachin.sudarshana@mu-sigma.com`>

Examples

```
## Not run:
library("RImpala")
rimpala.init(libs="/usr/lib/impala/lib")

## End(Not run)
```

`rimpala.invalidate` *Invalidates the metadata of a one or all tables*

Description

This function invalidates metadata of the table passed as an argument to it. Metadata invalidation is required if a table has been changed in Hive.

Usage

```
rimpala.invalidate(table=" ")
```

Arguments

`table` The name of the table whose metadata needs to be invalidated. Default is NULL

Value

The metadata of the table passed as an argumented is invalidated or marked as stale from the cache. If no argument is passed, all the metadata of all the tables are invalidated.

Author(s)

Vijay Raajaa <vijay.raajaa@mu-sigma.com>,
Austin Chungath Vincent <austin.cv@mu-sigma.com>,
Vikas Raguttahalli <vikas.r@mu-sigma.com>,
Sachin Sudarshana <sachin.sudarshana@mu-sigma.com>

Examples

```
## Not run:
library("RImpala")
rimpala.init()
rimpala.connect(IP="127.0.0.1",port="21050")
rimpala.invalidate(table="sample")

## End(Not run)
```

---

rimpala.query *Function to run a Query in Impala*

Description

This function executes the Query specified as an argument in Impala. If no query is passed, the show tables query is run as default

Usage

```
rimpala.query(Q="show tables")
```

Arguments

Q	The Query to be executed on Impala. The default query is show tables.

Value

The result of the Query is returned into a dataframe if the Query is valid and does not have any errors.

Author(s)

Vijay Raajaa <vijay.raajaa@mu-sigma.com>,
Austin Chungath Vincent <austin.cv@mu-sigma.com>,
Vikas Raguttahalli <vikas.r@mu-sigma.com>,
Sachin Sudarshana <sachin.sudarshana@mu-sigma.com>

Examples

```
## Not run:
library("RImpala")
rimpala.init()
rimpala.connect(IP="127.0.0.1",port="21050")
res = rimpala.query("Select * from sample_table")

## End(Not run)
```

---

rimpala.refresh *Refreshes and loads the new metadata for the given table*

Description

This function refreshes the metadata of the table passed as an argument to it.

Usage

```
rimpala.refresh(table="table_name")
```

Arguments

table	The name of the table whose metadata needs to be refreshed. This is a mandatory argument.

Value

The metadata of the table passed as an argument is refreshed and the new metadata is immediately loaded into the cache.

Author(s)

Vijay Raajaa <vijay.raajaa@mu-sigma.com>,
Austin Chungath Vincent <austin.cv@mu-sigma.com>,
Vikas Raguttahalli <vikas.r@mu-sigma.com>,
Sachin Sudarshana <sachin.sudarshana@mu-sigma.com>

Examples

```
## Not run:
library("RImpala")
rimpala.init()
rimpala.connect(IP="127.0.0.1",port="21050")
rimpala.refresh(table="sample")

## End(Not run)
```

`rimpala.showdatabases` *Function to list all the databases present*

Description

This function returns the list of databases present in Hive's metastore that is leveraged by Impala

Usage

```
rimpala.showdatabases()
```

Value

The list of databases present in Hive's metastore is returned into a dataframe.

Author(s)

Vijay Raajaa <`vijay.raajaa@mu-sigma.com`>,
Austin Chungath Vincent <`austin.cv@mu-sigma.com`>,
Vikas Raguttahalli <`vikas.r@mu-sigma.com`>,
Sachin Sudarshana <`sachin.sudarshana@mu-sigma.com`>

Examples

```
## Not run:
library("RImpala")
rimpala.init()
rimpala.connect("127.0.0.1","21050")
rimpala.showdatabases()

## End(Not run)
```

`rimpala.showtables` *Function to display the list of all the tables present*

Description

This function retrieves the list of tables present in the current working database

Usage

```
rimpala.showtables()
```

Value

List of tables present in the current database is returned into a dataframe

Author(s)

Vijay Raajaa <vijay.raajaa@mu-sigma.com>,
Austin Chungath Vincent <austin.cv@mu-sigma.com>,
Vikas Raguttahalli <vikas.r@mu-sigma.com>,
Sachin Sudarshana <sachin.sudarshana@mu-sigma.com>

Examples

```
## Not run:
library("RImpala")
rimpala.init()
rimpala.connect(IP="127.0.0.1",port="21050")
rimpala.showtables()

## End(Not run)
```

rimpala.usedatabase *Function to change the current working database*

Description

This function changes the current database to the database specified as an argument to the function

Usage

```
rimpala.usedatabase(db)
```

Arguments

db	The name of the database.

Value

Changes the database to the specified database and prints "Database changed to <*Database name*>" on the console

Author(s)

Vijay Raajaa <vijay.raajaa@mu-sigma.com>,
Austin Chungath Vincent <austin.cv@mu-sigma.com>,
Vikas Raguttahalli <vikas.r@mu-sigma.com>,
Sachin Sudarshana <sachin.sudarshana@mu-sigma.com>

Examples

```
## Not run:
library("RImpala")
rimpala.init()
rimpala.connect(127.0.0.1,"21050")
rimpala.usedatabase(db="sample_db")

## End(Not run)
```

Index