kyrsideris / **experiment_save_mode.md**

Last active 7 months ago • Report abuse

Apache Spark SQL's `SaveMode`s when writing to Apache Cassandra

`experiment_save_mode.md`

Experimentation on Spark's SaveMode

Experiment on the effect of different `SaveMode` and Cassandra starting from a populated table

Summary

If the cassandra table that spark targets exists then

- `SaveMode.Append` will update it
- `SaveMode.Overwrite` will truncate and insert (but it requires option `"confirm.truncate" -> "true"`)
- `SaveMode.Ignore` will not perform any action on existing table
- `SaveMode.ErrorIfExists` (default) will throw the following exception: https://github.com/datastax/spark-cassandra-connector/blob/v2.0.6/spark-cassandra-connector/src/main/scala/org/apache/spark/sql/cassandra/DefaultSource.scala#L93-L96

Versions: Spark 2.2.0, Cassandra 3.10, spark-cassandra-connector 2.0.6

Step 1: Setup table and values

```
$ cqlsh localhost -u cassandra -p cassandra -e "
DROP KEYSPACE IF EXISTS test_savemodes ;
CREATE KEYSPACE test_savemodes WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
USE test_savemodes;
CREATE TABLE people ( name text, surname text, children int, PRIMARY KEY (name, surname) );
INSERT INTO test_savemodes.people (name, surname, children) VALUES ( 'John', 'Patel', 2 );
INSERT INTO test_savemodes.people (name, surname, children) VALUES ( 'Galina', 'Xin', 1 );
INSERT INTO test_savemodes.people (name, surname) VALUES ( 'Eleni', 'Garcia' );
INSERT INTO test_savemodes.people (name, surname) VALUES ( 'Ode', 'Weber' );
SELECT * FROM test_savemodes.people;"

 name   | surname | children
--------+---------+----------
 Galina |     Xin |        1
  Eleni |  Garcia |     null
   John |   Patel |        2
    Ode |   Weber |     null

(4 rows)
```

Step 2: Use SaveModes.Append

```
$ $SPARK_HOME/bin/spark-shell --packages com.datastax.spark:spark-cassandra-connector_2.11:2.0.6 \
--conf "spark.cassandra.connection.host=127.0.0.1" \
--conf "spark.cassandra.auth.username=cassandra" \
--conf "spark.cassandra.auth.password=cassandra" << EOF

case class Person(name: String, surname: String, children: Int)
```

```
val newNames = spark.sparkContext.parallelize(Seq(Person("Eleni", "Garcia", 1), Person("Galina", "Xin", 2), Person("C
newNames.write.format("org.apache.spark.sql.cassandra").options(Map("table" -> "people", "keyspace" -> "test_savemode
mode(org.apache.spark.sql.SaveMode.Append).save
EOF

cqlsh localhost -u cassandra -p cassandra -e "SELECT * FROM test_savemodes.people;"

 name    | surname | children
---------+---------+----------
 Galina |     Xin |        2
  Eleni |  Garcia |        1
   John |   Patel |        2
  Carlo |    Tran |        1
    Ode |   Weber |     null

(5 rows)
```

Step 3: Use SaveModes.Overwrite

Repeat Step 1 again and then:

```
$ $SPARK_HOME/bin/spark-shell --packages com.datastax.spark:spark-cassandra-connector_2.11:2.0.6 \
--conf "spark.cassandra.connection.host=127.0.0.1" \
--conf "spark.cassandra.auth.username=cassandra" \
--conf "spark.cassandra.auth.password=cassandra" << EOF

case class Person(name: String, surname: String, children: Int)
val newNames = spark.sparkContext.parallelize(Seq(Person("Eleni", "Garcia", 1), Person("Galina", "Xin", 2),
Person("Carlo", "Tran", 1))).toDS
newNames.write.format("org.apache.spark.sql.cassandra").options(Map("table" -> "people", "keyspace" ->
"test_savemodes")).
mode(org.apache.spark.sql.SaveMode.Overwrite).save
EOF

cqlsh localhost -u cassandra -p cassandra -e "SELECT * FROM test_savemodes.people;"

java.lang.UnsupportedOperationException: You are attempting to use overwrite mode which will truncate
this table prior to inserting data. If you would merely like
to change data already in the table use the "Append" mode.
To actually truncate please pass in true value to the option
"confirm.truncate" when saving.
  at org.apache.spark.sql.cassandra.CassandraSourceRelation.insert(CassandraSourceRelation.scala:64)
  at org.apache.spark.sql.cassandra.DefaultSource.createRelation(DefaultSource.scala:87)
  at org.apache.spark.sql.execution.datasources.DataSource.write(DataSource.scala:472)
  at org.apache.spark.sql.execution.datasources.SaveIntoDataSourceCommand.run(SaveIntoDataSourceCommand.scala:48)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult$lzycompute(commands.scala:58)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult(commands.scala:56)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.doExecute(commands.scala:74)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:117)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:117)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$executeQuery$1.apply(SparkPlan.scala:138)
  at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
  at org.apache.spark.sql.execution.SparkPlan.executeQuery(SparkPlan.scala:135)
  at org.apache.spark.sql.execution.SparkPlan.execute(SparkPlan.scala:116)
  at org.apache.spark.sql.execution.QueryExecution.toRdd$lzycompute(QueryExecution.scala:92)
  at org.apache.spark.sql.execution.QueryExecution.toRdd(QueryExecution.scala:92)
  at org.apache.spark.sql.DataFrameWriter.runCommand(DataFrameWriter.scala:610)
  at org.apache.spark.sql.DataFrameWriter.save(DataFrameWriter.scala:233)
  ... 49 elided
```

Retry using `"confirm.truncate" -> "true"` in options:

```
ARK_HOME/bin/spark-shell --packages com.datastax.spark:spark-cassandra-connector_2.11:2.0.6 \
f "spark.cassandra.connection.host=127.0.0.1" \
f "spark.cassandra.auth.username=cassandra" \
```

```
f "spark.cassandra.auth.password=cassandra" << EOF

class Person(name: String, surname: String, children: Int)
ewNames = spark.sparkContext.parallelize(Seq(Person("Eleni", "Garcia", 1), Person("Galina", "Xin", 2), Person("Carlo",
mes.write.format("org.apache.spark.sql.cassandra").options(Map("table" -> "people", "keyspace" -> "test_savemodes", "cc
org.apache.spark.sql.SaveMode.Overwrite).save

 localhost -u cassandra -p cassandra -e "SELECT * FROM test_savemodes.people;"


   | surname | children
---+---------+----------
na |     Xin |        2
ni |  Garcia |        1
lo |    Tran |        1

ws)
```

Step 4: Use SaveModes.Ignore

Repeat Step 1 again and then:

```
$ $SPARK_HOME/bin/spark-shell --packages com.datastax.spark:spark-cassandra-connector_2.11:2.0.6 \
--conf "spark.cassandra.connection.host=127.0.0.1" \
--conf "spark.cassandra.auth.username=cassandra" \
--conf "spark.cassandra.auth.password=cassandra" << EOF

case class Person(name: String, surname: String, children: Int)
val newNames = spark.sparkContext.parallelize(Seq(Person("Eleni", "Garcia", 1), Person("Galina", "Xin", 2), Person("C
newNames.write.format("org.apache.spark.sql.cassandra").options(Map("table" -> "people", "keyspace" -> "test_savemode
mode(org.apache.spark.sql.SaveMode.Ignore).save
EOF
cqlsh localhost -u cassandra -p cassandra -e "SELECT * FROM test_savemodes.people;"

 name   | surname | children
--------+---------+----------
 Galina |     Xin |        1
  Eleni |  Garcia |     null
   John |   Patel |        2
    Ode |   Weber |     null

(4 rows)
```

Step 5: Use SaveModes.ErrorIfExists

Repeat Step 1 again and then:

```
$SPARK_HOME/bin/spark-shell --packages com.datastax.spark:spark-cassandra-connector_2.11:2.0.6 \
--conf "spark.cassandra.connection.host=127.0.0.1" \
--conf "spark.cassandra.auth.username=cassandra" \
--conf "spark.cassandra.auth.password=cassandra" << EOF

case class Person(name: String, surname: String, children: Int)
val newNames = spark.sparkContext.parallelize(Seq(Person("Eleni", "Garcia", 1), Person("Galina", "Xin", 2), Person("C
newNames.write.format("org.apache.spark.sql.cassandra").options(Map("table" -> "people", "keyspace" -> "test_savemode
mode(org.apache.spark.sql.SaveMode.ErrorIfExists).save
EOF
cqlsh localhost -u cassandra -p cassandra -e "SELECT * FROM test_savemodes.people;"

java.lang.UnsupportedOperationException: 'SaveMode is set to ErrorIfExists and Table
test_savemodes.people already exists and contains data.
Perhaps you meant to set the DataFrame write mode to Append?
```

```
Example: df.write.format.options.mode(SaveMode.Append).save()" '
  at org.apache.spark.sql.cassandra.DefaultSource.createRelation(DefaultSource.scala:92)
  at org.apache.spark.sql.execution.datasources.DataSource.write(DataSource.scala:472)
  at org.apache.spark.sql.execution.datasources.SaveIntoDataSourceCommand.run(SaveIntoDataSourceCommand.scala:48)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult$lzycompute(commands.scala:58)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult(commands.scala:56)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.doExecute(commands.scala:74)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:117)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:117)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$executeQuery$1.apply(SparkPlan.scala:138)
  at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
  at org.apache.spark.sql.execution.SparkPlan.executeQuery(SparkPlan.scala:135)
  at org.apache.spark.sql.execution.SparkPlan.execute(SparkPlan.scala:116)
  at org.apache.spark.sql.execution.QueryExecution.toRdd$lzycompute(QueryExecution.scala:92)
  at org.apache.spark.sql.execution.QueryExecution.toRdd(QueryExecution.scala:92)
  at org.apache.spark.sql.DataFrameWriter.runCommand(DataFrameWriter.scala:610)
  at org.apache.spark.sql.DataFrameWriter.save(DataFrameWriter.scala:233)
  ... 49 elided
```

Try without any `SaveMode` option:

```
$ $SPARK_HOME/bin/spark-shell --packages com.datastax.spark:spark-cassandra-connector_2.11:2.0.6 --conf "spark.cassar

case class Person(name: String, surname: String, children: Int)
val newNames = spark.sparkContext.parallelize(Seq(Person("Eleni", "Garcia", 1), Person("Galina", "Xin", 2), Person("C
newNames.write.format("org.apache.spark.sql.cassandra").options(Map("table" -> "people", "keyspace" -> "test_savemode
EOF
cqlsh localhost -u cassandra -p cassandra -e "SELECT * FROM test_savemodes.people;"

java.lang.UnsupportedOperationException: 'SaveMode is set to ErrorIfExists and Table
test_savemodes.people already exists and contains data.
Perhaps you meant to set the DataFrame write mode to Append?
Example: df.write.format.options.mode(SaveMode.Append).save()" '
  at org.apache.spark.sql.cassandra.DefaultSource.createRelation(DefaultSource.scala:92)
  at org.apache.spark.sql.execution.datasources.DataSource.write(DataSource.scala:472)
  at org.apache.spark.sql.execution.datasources.SaveIntoDataSourceCommand.run(SaveIntoDataSourceCommand.scala:48)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult$lzycompute(commands.scala:58)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult(commands.scala:56)
  at org.apache.spark.sql.execution.command.ExecutedCommandExec.doExecute(commands.scala:74)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:117)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply(SparkPlan.scala:117)
  at org.apache.spark.sql.execution.SparkPlan$$anonfun$executeQuery$1.apply(SparkPlan.scala:138)
  at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
  at org.apache.spark.sql.execution.SparkPlan.executeQuery(SparkPlan.scala:135)
  at org.apache.spark.sql.execution.SparkPlan.execute(SparkPlan.scala:116)
  at org.apache.spark.sql.execution.QueryExecution.toRdd$lzycompute(QueryExecution.scala:92)
  at org.apache.spark.sql.execution.QueryExecution.toRdd(QueryExecution.scala:92)
  at org.apache.spark.sql.DataFrameWriter.runCommand(DataFrameWriter.scala:610)
  at org.apache.spark.sql.DataFrameWriter.save(DataFrameWriter.scala:233)
  ... 48 elided
```