how to export a table dataframe in pyspark to csv?

I am using spark-1.3.1 (pyspark) and I have generated a table using a SQL query. I now have an object that is a DataFrame. I want to export this DataFrame object (I have called it "table") to a csv file so I can manipulate it and plot the columns. How do I export the DataFrame "table" to a csv file?

Thanks!

python sql apache-spark dataframe export-to-csv

asked Jul 13 '15 at 13:56

Killian Tattan

39 1 6

3 Answers

If data frame fits in a driver memory you can convert Spark DataFrame to local Pandas DataFrame using `toPandas` method and then simply use `save` :

```
df.toPandas().to_csv('mycsv.csv')
```

Otherwise you can use spark-csv:

- Spark 1.3

  ```
  df.save('mycsv.csv', 'com.databricks.spark.csv')
  ```

- Spark 1.4+

  ```
  df.write.format('com.databricks.spark.csv').save('mycsv.csv')
  ```

In Spark 2.0+ you can use `csv` data source directly:

```
df.write.csv('mycsv.csv')
```

edited Jul 5 at 18:04

answered Jul 13 '15 at 14:36

zero323
69.1k 16 82 148

If you cannot use spark-csv, you can do the following:

```
df.rdd.map(lambda x: ",".join(map(str, x))).coalesce(1).saveAsTextFile("file.csv")
```

If you need to handle strings with linebreaks or comma that will not work. Use this:

```
import csv
import cStringIO
```

```
def row2csv(row):
    buffer = cStringIO.StringIO()
    writer = csv.writer(buffer)
    writer.writerow([str(s).encode("utf-8") for s in row])
    buffer.seek(0)
    return buffer.read().strip()

df.rdd.map(row2csv).coalesce(1).saveAsTextFile("file.csv")
```

answered Mar 22 at 20:30

jbochi
15.3k 9 51 78

How about this (in you don't want an one liner) ?

```
for row in df.collect():
    d = row.asDict()
    s = "%d\t%s\t%s\n" % (d["int_column"], d["string_column"], d["string_column"])
    f.write(s)
```

f is a opened file descriptor. Also the separator is a TAB char, but it's easy to change to whatever you want.

answered Jun 22 at 14:06

Matei Florescu
520 3 13
