What is the difference between Apache Spark SQLContext vs HiveContext?

What are the differences between Apache Spark SQLContext and HiveContext ?

Some sources say that since the HiveContext is a superset of SQLContext developers should always use HiveContext which has more features than SQLContext. But the current APIs of each contexts are mostly same.

- What are the scenarios which SQLContext/HiveContext is more useful ?.
- Is HiveContext more useful only when working with Hive ?.
- Or does the SQLContext is all that needs in implementing a Big Data app using Apache Spark ?

apache-spark hive apache-spark-sql

edited Jul 28 '16 at 9:27

asked Nov 12 '15 at 7:49

tlarevo
340 1 3 11

3 Answers

Spark 2.0+

Spark 2.0 provides native window functions (SPARK-8641) and features some additional improvements in parsing and much better SQL 2003 compliance so it is significantly less dependent on Hive to achieve core funcionality and because of that `HiveContext` (`SparkSession` with Hive support) seems to be slightly less important.

Spark < 2.0

Obviously if you want to work with Hive you have to use `HiveContext`. Beyond that the biggest difference as for now (Spark 1.5) is a support for window functions and ability to access Hive UDFs.

Generally speaking window functions are a pretty cool feature and can be used to solve quite complex problems in a concise way without going back and forth between RDDs and DataFrames. Performance is still far from optimal especially without `PARTITION BY` clause but it is really nothing Spark specific.

Regarding Hive UDFs it is not a serious issue now, but before Spark 1.5 many SQL functions have been expressed using Hive UDFs and required `HiveContext` to work.

`HiveContext` also provides more robust SQL parser. See for example: py4j.protocol.Py4JJavaError when selecting nested column in dataframe using select statetment

Finally `HiveContext` is required to start Thrift server.

The biggest problem with `HiveContext` is that it comes with large dependencies.

edited May 23 '17 at 12:10

Community ♦
1 1

answered Nov 12 '15 at 9:22

zero323
135k 33 356 469

From your comment, it seems HiveContext's only downside is it's large dependencies. Other than that, is it a safe bet to always use HiveContext instead of SqlContext. I am running SparkR 1.6 environment. – prog_guy Feb 23 '16 at 9:39

From jaceklaskowski.gitbooks.io/mastering-apache-spark/content/… "SparkSession has merged SQLContext and HiveContext in one object in Spark 2.0." – Mr. Tea Dec 18 '17 at 20:16

When programming against Spark SQL we have two entry points depending on whether we need Hive support. The recommended entry point is the HiveContext to provide access to HiveQL and other Hive-dependent functionality. The more basic SQLContext provides a subset of the Spark SQL support that does not depend on Hive.

-The separation exists for users who might have conflicts with including all of the Hive dependencies.

queries using the more complete HiveQL parser, access to Hive UDFs, and the ability to read data from Hive tables.

-Using a HiveContext does not require an existing Hive setup.

answered Nov 27 '16 at 21:07

BigDataScholar
396 2 16

for further reading refer spark.apache.org/docs/1.6.1/sql-programming-guide.html – BigDataScholar Nov 27 '16 at 21:08

even sql context does not require hive setup – anubhav Nov 2 '17 at 11:03

HiveContext is still the superset of sqlcontext,it contains certain extra properties such as it can read the configuration from hive-site.xml,in case you have hive use otherwise simply use sqlcontext

edited Nov 2 '17 at 11:01

answered Oct 30 '17 at 8:24

anubhav
173 1 6