Corpus

# Apache Spark

## Title:

Title: [SPARK-13740] add null check for \_verify\_type in types.py

Title: [SPARK-13739] Predicate Push Down Through Window Operator

Title: [SPARK-13738] Clean up ResolveDataSource

Title: [SPARK-13737] Add getOrCreate method for HiveContext

Title: [SPARK-13736] Big-Endian plataform issues

Title: [SPARK-13735] Log for parquet relation reading files is too verbose

Title: [SPARK-13734] SparkR histogram

Title: [SPARK-13733] Support initial weight distribution in personalized PageRank

Title: [SPARK-13732] Remove projectList from Windows

Title: [SPARK-13731] expression evaluation for NaN in select statement

Title: [SPARK-13730] Nulls in dataframes getting converted to 0 with spark 2.0 SNAPSHOT

Title: [SPARK-13729] Reimplement the planning tests on SimpleTextRelation

Title: [SPARK-13728] Fix ORC PPD

Title: [SPARK-13727] SparkConf.contains does not consider deprecated keys

Title: [SPARK-13726] Spark 1.6.0 stopping working for HiveThriftServer2 and registerTempTable

Title: [SPARK-13725] Spark 1.6.0 stopping working for HiveThriftServer2 and registerTempTable

Title: [SPARK-13724] Parameter maxMemoryInMB has gone missing in MlLib 1.6.0 DecisionTree.trainClassifier()

Title: [SPARK-13723] YARN - Change behavior of --num-executors when spark.dynamicAllocation.enabled true

Title: [SPARK-13722] No Push Down for Non-deterministic Predicates through Generate

Title: [SPARK-13721] Add support for LATERAL VIEW OUTER explode()

Title: [SPARK-13720] SQLBuilder.toSQL is broken when there is a sort by after having

Title: [SPARK-13719] Bad JSON record raises ﻿java.lang.ClassCastException

Title: [SPARK-13718] Scheduler "creating" straggler node

Title: [SPARK-13717] Let RandomSampler can sample with Java iterator

Title: [SPARK-13716] mvn deploy tried to upload test jar of spark-network-common\_2.10 twice

Title: [SPARK-13715] Remove last usages of jblas in tests

Title: [SPARK-13714] Another ConnectedComponents based on Max-Degree Propagation

Title: [SPARK-13713] Replace ANTLR3 SQL parser by a ANTLR4 SQL parser

Title: [SPARK-13712] Add OneVsOne to ML

Title: [SPARK-13711] Apache Spark driver stopping JVM when master not available

Title: [SPARK-13710] Spark shell shows ERROR when launching on Windows

Title: [SPARK-13709] Spark unable to decode Avro when partitioned

Title: [SPARK-13708] Null pointer Exception while starting spark shell

Title: [SPARK-13707] Streaming UI tab misleading for window operations

Title: [SPARK-13706] Python Example for Train Validation Split Missing

Title: [SPARK-13705] UpdateStateByKey Operation documentation incorrectly refers to StatefulNetworkWordCount

Title: [SPARK-13704] TaskSchedulerImpl.createTaskSetManager can be expensive, and result in lost executors due to blocked heartbeats

Title: [SPARK-13703] Remove obsolete scala-2.10 source files

Title: [SPARK-13702] Use diamond operator for generic instance creation in Java code

Title: [SPARK-13701] MLlib ALS fails on arm64 (java.lang.UnsatisfiedLinkError: org.jblas.NativeBlas.dgemm))

Title: [SPARK-13700] Rdd.mapAsync(): Easily mix Spark and asynchroneous transformation

Title: [SPARK-13699] Spark SQL drops the table in "overwrite" mode while writing into table

Title: [SPARK-13698] Fix Analysis Exceptions when Using Backticks in Generate

Title: [SPARK-13697] TransformFunctionSerializer.loads doesn't restore the function's module name if it's '\_\_main\_\_'

Title: [SPARK-13696] Remove BlockStore interface to more cleanly reflect different memory and disk store responsibilities

Title: [SPARK-13695] Don't cache MEMORY\_AND\_DISK blocks as bytes in memory store when reading spills

Title: [SPARK-13694] QueryPlan.expressions should always include all expressions

Title: [SPARK-13693] Flaky test: o.a.s.streaming.MapWithStateSuite

Title: [SPARK-13692] Fix trivial Coverity/Checkstyle defects

Title: [SPARK-13691] Scala and Python generate inconsistent results

Title: [SPARK-13690] UnsafeShuffleWriterSuite fails on arm64 (SnappyError, no native library is found)

Title: [SPARK-13689] Move some methods in CatalystQl to a util object

Title: [SPARK-13688] Add option to use dynamic allocation even if spark.executor.instances is set.

Title: [SPARK-13687] Cleanup pyspark temporary files

Title: [SPARK-13686] Add a constructor parameter `reqParam` to (Streaming)LinearRegressionWithSGD

Title: [SPARK-13685] Rename catalog.Catalog to ExternalCatalog

Title: [SPARK-13684] Possible unsafe bytesRead increment in StreamInterceptor

Title: [SPARK-13683] Finalize the public interface for OutputWriter[Factory]

Title: [SPARK-13682] Finalize the public API for FileFormat

Title: [SPARK-13681] Reimplement CommitFailureTestRelationSuite

Title: [SPARK-13680] Java UDAF with more than one intermediate argument returns wrong results

Title: [SPARK-13679] Pyspark job fails with Oozie

Title: [SPARK-13678] transformExpressions should only apply on QueryPlan.expressions

Title: [SPARK-13677] Support Tree-Based Feature Transformation for mllib

Title: [SPARK-13676] Fix mismatched default values for regParam in LogisticRegression

Title: [SPARK-13675] The url link in historypage is not correct for application running in yarn cluster mode

Title: [SPARK-13674] Add wholestage codegen support to Sample

Title: [SPARK-13673] script bin\beeline.cmd pollutes environment variables in Windows.

Title: [SPARK-13672] Add python examples of BisectingKMeans in ML and MLLIB

Title: [SPARK-13671] Use different physical plan for existing RDD and data sources

Title: [SPARK-13670] spark-class doesn't bubble up error from launcher command

Title: [SPARK-13669] Job will always fail in the external shuffle service unavailable situation

Title: [SPARK-13668] Reorder filter/join predicates to short-circuit isNotNull checks

Title: [SPARK-13667] Support for specifying custom date format for date and timestamp types

Title: [SPARK-13666] Annoying warnings from SQLConf in log output

Title: [SPARK-13665] Initial separation of concerns

Title: [SPARK-13664] Simplify and Speedup HadoopFSRelation

Title: [SPARK-13663] Upgrade Snappy Java to 1.1.2.1

Title: [SPARK-13662] [SQL][Hive] Have SHOW TABLES return additional fields from Hive MetaStore

Title: [SPARK-13661] Avoid the copy of UnsafeRow in HashedRelation

Title: [SPARK-13660] CommitFailureTestRelationSuite floods the logs with garbage

Title: [SPARK-13659] Remove returnValues from BlockStore APIs

Title: [SPARK-13658] BooleanSimplification rule is slow with large boolean expressions

Title: [SPARK-13657] Support parsing very long AND/OR expression

Title: [SPARK-13656] Delete spark.sql.parquet.cacheMetadata

Title: [SPARK-13655] Fix WithAggregationKinesisBackedBlockRDDSuite

Title: [SPARK-13654] get\_json\_object fails with java.io.CharConversionException

Title: [SPARK-13653] Split DiskBlockObjectWriter to separate object- and byte-based write interfaces

Title: [SPARK-13652] TransportClient.sendRpcSync returns wrong results

Title: [SPARK-13651] Generator outputs are not resolved correctly resulting in runtime error

Title: [SPARK-13650] Usage of the window() function on DStream

Title: [SPARK-13649] Move CalendarInterval out of unsafe package

Title: [SPARK-13648] org.apache.spark.sql.hive.client.VersionsSuite fails NoClassDefFoundError on IBM JDK

Title: [SPARK-13647] also check if numeric value is within allowed range in \_verify\_type

Title: [SPARK-13646] QuantileDiscretizer counts dataset twice in getSampledInput

Title: [SPARK-13645] DAG Diagram not shown properly in Chrome

Title: [SPARK-13644] Add the source file name and line into Logger when an exception occurs in the generated code

Title: [SPARK-13643] Create SparkSession interface

Title: [SPARK-13642] Properly handle signal kill of ApplicationMaster

Title: [SPARK-13641] getModelFeatures of ml.api.r.SparkRWrapper cannot (always) reveal the original column names

## Description:

Description:

Description: <p>Push down the predicate through the Window operator.</p>

Description:

Description: <p>There is a "getOrCreate" method in SQLContext, which is useful to recoverable streaming application with SQL operation. <br/>

<a href="https://spark.apache.org/docs/latest/streaming-programming-guide.html#dataframe-and-sql-operations" class="external-link" rel="nofollow">https://spark.apache.org/docs/latest/streaming-programming-guide.html#dataframe-and-sql-operations</a></p>

<p>But the corresponding method is missing in HiveContext. </p>

Description: <p>We are starting to see few issues when building/testing on Big-Endian platform. This serves as an umbrella jira to group all platform specific issues.</p>

Description: <p>The INFO level logging contains all files read by Parquet Relation, which is way too verbose if the input contains lots of files</p>

Description:

Description: <p>It would be nice to support personalized PageRank with an initial weight distribution besides a single vertex. It should be easy to modify the current implementation to add this support.</p>

Description: <p>projectList is useless. Remove it from the class Window. It simplifies the codes in Analyzer and Optimizer. </p>

Description: <p>We are expecting that arithmetic expression a/b should be:<br/>

1. returning NaN if a=0 and b=0<br/>

2. returning Infinity if a=1 and b=0</p>

<p>Is the expectation reasonable? <br/>

The following is a simple test case snippet that reads from storage and evaluates arithmetic expressions in select.<br/>

It is assuming org.apache.spark.sql.hive.execution.SQLQuerySuite: </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

test(<span class="code-quote">"Expression should be evaluated to Nan/Infinity in Select"</span>) {

withTable(<span class="code-quote">"testNan"</span>) {

withTempTable(<span class="code-quote">"src"</span>) {

Seq((1d, 0d), (0d, 0d)).toDF().registerTempTable(<span class="code-quote">"src"</span>)

sql(<span class="code-quote">"CREATE TABLE testNan(a <span class="code-object">double</span>, b <span class="code-object">double</span>) STORED AS PARQUET AS SELECT \* FROM src"</span>)

}

checkAnswer(sql(

"""

|SELECT a/b FROM testNan

""".stripMargin),

Seq(

Row(<span class="code-object">Double</span>.PositiveInfinity),

Row(<span class="code-object">Double</span>.NaN)

)

)

}

}

== Physical Plan ==

Project [(a#28 / b#29) AS \_c0#30]

+- Scan ParquetRelation: <span class="code-keyword">default</span>.testnan[a#28,b#29] InputPaths: file:/<span class="code-keyword">private</span>/<span class="code-keyword">var</span>/folders/dy/19y6pfm92pj9s40mbs8xd9hm0000gp/T/warehouse--5b617080-e909-4812-90e8-63d2dd0aef5a/testnan

== Results ==

!== Correct Answer - 2 == == Spark Answer - 2 ==

![Infinity] [<span class="code-keyword">null</span>]

![NaN] [<span class="code-keyword">null</span>]

</pre>

</div></div>

Description: <p>Basically I'm putting nulls into a non-nullable LongType column and doing a transformation operation on that column, the result is a column with nulls converted to 0. </p>

<p>I haven't tested this on 1.6.1 or in Scala.</p>

<p>Heres an example </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

from pyspark.sql <span class="code-keyword">import</span> types

from pyspark.sql <span class="code-keyword">import</span> DataFrame, types, functions as F

sql\_schema = types.StructType([

types.StructField(<span class="code-quote">"a"</span>, types.LongType(), True),

types.StructField(<span class="code-quote">"b"</span>, types.StringType(), True),

])

df = sqlCtx.createDataFrame([

(1, <span class="code-quote">"one"</span>),

(None, <span class="code-quote">"two"</span>),

], sql\_schema)

# Everything is fine here

df.collect() # [Row(a=1, b=u'one'), Row(a=None, b=u'two')]

def assert\_not\_null(val):

<span class="code-keyword">return</span> val

udf = F.udf(assert\_not\_null, types.LongType())

df = df.withColumnRenamed('a', <span class="code-quote">"\_tmp\_col"</span>)

df = df.withColumn('a', udf(df.\_tmp\_col))

df = df.drop(<span class="code-quote">"\_tmp\_col"</span>)

# None gets converted to 0

df.collect() # [Row(b=u'one', a=1), Row(b=u'two', a=0)]

</pre>

</div></div>

Description:

Description: <p>Fix the ignored test "Enable ORC PPD" in OrcQuerySuite.</p>

Description: <p>This makes it kinda inconsistent with other SparkConf APIs. For example:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

scala&gt; <span class="code-keyword">import</span> org.apache.spark.SparkConf

<span class="code-keyword">import</span> org.apache.spark.SparkConf

scala&gt; val conf = <span class="code-keyword">new</span> SparkConf().set(<span class="code-quote">"spark.io.compression.lz4.block.size"</span>, <span class="code-quote">"12345"</span>)

16/03/07 10:55:17 WARN spark.SparkConf: The configuration key 'spark.io.compression.lz4.block.size' has been deprecated as of Spark 1.4 and and may be removed in the <span class="code-keyword">future</span>. Please use the <span class="code-keyword">new</span> key 'spark.io.compression.lz4.blockSize' instead.

conf: org.apache.spark.SparkConf = org.apache.spark.SparkConf@221e8982

scala&gt; conf.get(<span class="code-quote">"spark.io.compression.lz4.blockSize"</span>)

res0: <span class="code-object">String</span> = 12345

scala&gt; conf.contains(<span class="code-quote">"spark.io.compression.lz4.blockSize"</span>)

res1: <span class="code-object">Boolean</span> = <span class="code-keyword">false</span>

</pre>

</div></div>

Description: <p>In Spark 1.5.2, DataFrame.registerTempTable works and hiveContext.table(registerTableName) and HiveThriftServer2 see those tables.</p>

<p>In Spark 1.6.0, hiveContext.table(registerTableName) and HiveThriftServer2 do not see those tables, even though DataFrame.registerTempTable does not return an error.</p>

<p>Since this feature used to work in Spark 1.5.2, there is existing code that breaks after upgrading to Spark 1.6.0. so this issue is a blocker and urgent. Therefore, please have it fixed asap.</p>

Description: <p>In Spark 1.5.2, DataFrame.registerTempTable API works correctly and HiveThriftServer2 sees and returns temp tables that are registered via that API.</p>

<p>In Spark 1.6.0, that stopped working. registerTempTable API does not return an error so it is a false positive, and HiveThriftServer2 does not see such tables. And hiveContext.table(registerTableName) indicates it does not see those tables either.</p>

<p>Is there a temporary work-around solution in Spark 1.6.0 ? When would it be fixed ?</p>

<p>Thanks.</p>

Description: <p>DecisionTree.trainClassifier() reports that maxMemoryInMB is too small during training and stops. But when I try to set it, I found that in MLlib of spark 1.6.0 pyspark.mllib.tree.DecisionTree doesn't have this parameter in the named parameter list anymore. </p>

<p>(Also not sure if this is the place for this issue, kindly educate!) </p>

Description: <p>I think we should change the behavior when --num-executors is specified when dynamic allocation is enabled. Currently if --num-executors is specified dynamic allocation is disabled and it just uses a static number of executors.</p>

<p>I would rather see the default behavior changed in the 2.x line. If dynamic allocation config is on then num-executors goes to max and initial # of executors. I think this would allow users to easily cap their usage and would still allow it to free up executors. It would also allow users doing ML start out with a # of executors and if they are actually caching the data the executors wouldn't be freed up. So you would get very similar behavior to if dynamic allocation was off.</p>

<p>Part of the reason for this is when using a static number if generally wastes resources, especially with people doing adhoc things with spark-shell. It also has a big affect when people are doing MapReduce/ETL type work loads. The problem is that people are used to specifying num-executors so if we turn it on by default in a cluster config its just overridden.</p>

<p>We should also update the spark-submit --help description for --num-executors</p>

Description: <p>Non-deterministic predicates should not be pushed down through Generate.</p>

Description: <p>Hive supports the <a href="https://cwiki.apache.org/confluence/display/Hive/LanguageManual+LateralView#LanguageManualLateralView-OuterLateralViews" class="external-link" rel="nofollow">LATERAL VIEW OUTER</a> syntax to make sure that when an array is empty, the content from the outer table is still returned. </p>

<p>Within Spark, this is currently only possible within the HiveContext and executing HiveQL statements. It would be nice if the standard explode() DataFrame method allows the same. A possible signature would be: </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<div class="error"><span class="error">Unable to find source-code formatter for language: scala.</span> Available languages are: actionscript, html, java, javascript, none, sql, xhtml, xml</div><pre>

explode[A, B](inputColumn: <span class="code-object">String</span>, outputColumn: <span class="code-object">String</span>, <span class="code-keyword">outer</span>: <span class="code-object">Boolean</span> = <span class="code-keyword">false</span>)

</pre>

</div></div>

Description: <p>parse this SQL "SELECT COUNT(value) FROM parquet\_t1 GROUP BY key HAVING MAX(key) &gt; 0 SORT BY key" to logical plan and generate SQL string again, the generated SQL string can't be parsed.</p>

<p>This is because, we will add extra `SubqueryAlias` while converting logical plan to SQL string(the `RecoverScopingInfo` rule in SQLBuilder). However, we only update qualifiers in the `Project` operator that just above the added `SubqueryAlias`, logically we also need to update qualifiers in all ancestor plans. For example, if there is a Project under a Sort, if we add `SubqueryAlias` under Project, we need update qualifiers for Sort too.</p>

Description: <p>I have defined a JSON schema, using org.apache.spark.sql.types.StructType, that expects this kind of record :</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>{

"request": {

"user": {

"id": 123

}

}

}

</pre>

</div></div>

<p>There's a bad record in my dataset, that defines field "user" as an array, instead of a JSON object :</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>{

"request": {

"user": []

}

}

</pre>

</div></div>

<p>The following exception is raised because of that bad record :</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>Exception in thread "main" org.apache.spark.SparkException: Job aborted due to stage failure: Task 7 in stage 0.0 failed 4 times, most recent failure: Lost task 7.3 in stage 0.0 (TID 10, 192.168.1.170): java.lang.ClassCastException: org.apache.spark.sql.types.GenericArrayData cannot be cast to org.apache.spark.sql.catalyst.InternalRow

at org.apache.spark.sql.catalyst.expressions.BaseGenericInternalRow$class.getStruct(rows.scala:50)

at org.apache.spark.sql.catalyst.expressions.GenericMutableRow.getStruct(rows.scala:247)

at org.apache.spark.sql.catalyst.expressions.GeneratedClass$SpecificPredicate.eval(Unknown Source)

at org.apache.spark.sql.catalyst.expressions.codegen.GeneratePredicate$$anonfun$create$2.apply(GeneratePredicate.scala:67)

at org.apache.spark.sql.catalyst.expressions.codegen.GeneratePredicate$$anonfun$create$2.apply(GeneratePredicate.scala:67)

at org.apache.spark.sql.execution.Filter$$anonfun$4$$anonfun$apply$4.apply(basicOperators.scala:117)

at org.apache.spark.sql.execution.Filter$$anonfun$4$$anonfun$apply$4.apply(basicOperators.scala:115)

at scala.collection.Iterator$$anon$14.hasNext(Iterator.scala:390)

at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:327)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1.org$apache$spark$sql$execution$aggregate$TungstenAggregate$$anonfun$$executePartition$1(TungstenAggregate.scala:97)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1$$anonfun$2.apply(TungstenAggregate.scala:119)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1$$anonfun$2.apply(TungstenAggregate.scala:119)

at org.apache.spark.rdd.MapPartitionsWithPreparationRDD.compute(MapPartitionsWithPreparationRDD.scala:64)

at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:300)

at org.apache.spark.rdd.RDD.iterator(RDD.scala:264)

at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:38)

at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:300)

at org.apache.spark.rdd.RDD.iterator(RDD.scala:264)

at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:73)

at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:41)

at org.apache.spark.scheduler.Task.run(Task.scala:88)

at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:214)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)

at java.lang.Thread.run(Thread.java:745)

</pre>

</div></div>

<p>Here's a code snippet that reproduces the exception :</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>import org.apache.spark.SparkContext

import org.apache.spark.rdd.RDD

import org.apache.spark.sql.{SQLContext, DataFrame}

import org.apache.spark.sql.hive.HiveContext

import org.apache.spark.sql.types.{StringType, StructField, StructType}

object Snippet {

def main(args : Array[String]): Unit = {

val sc = new SparkContext()

implicit val sqlContext = new HiveContext(sc)

val rdd: RDD[String] = sc.parallelize(Seq(badRecord))

val df: DataFrame = sqlContext.read.schema(schema).json(rdd)

import sqlContext.implicits.\_

df.select("request.user.id")

.filter($"id".isNotNull)

.count()

}

val badRecord =

s"""{

| "request": {

| "user": []

| }

|}""".stripMargin.replaceAll("\n", " ") // Convert the multiline string to a signe line string

val schema =

StructType(

StructField("request", StructType(

StructField("user", StructType(

StructField("id", StringType) :: Nil

)) :: Nil

)) :: Nil)

}

</pre>

</div></div>

Description: <p><b>Data:</b></p>

<ul>

<li>Assume an even distribution of data across the cluster with a replication factor of 3.</li>

<li>In-memory data are partitioned in 128 chunks (384 cores in total, i.e. 3 requests can be executed concurrently(-ish) )</li>

</ul>

<p><b>Action:</b></p>

<ul>

<li>Action is a simple sequence of map/filter/reduce.</li>

<li>The action operates upon and returns a small subset of data (following the full map over the data).</li>

<li>Data are 1 x cached serialized in memory (Kryo), so calling the action should not hit the disk under normal conditions.</li>

<li>Action network usage is low as it returns a small number of aggregated results and does not require excessive shuffling</li>

<li>Under low or moderate load, each action is expected to complete in less than 2 seconds</li>

</ul>

<p><b>H/W Outlook</b><br/>

When the action is called in high numbers, initially the cluster CPU gets close to 100% (which is expected &amp; intended). <br/>

After a while, the cluster utilization reduces significantly with only one (struggler) node having 100% CPU and fully utilized network.</p>

<p><b>Diagnosis:</b><br/>

1. Attached a profiler to the driver and executors to monitor GC or I/O issues and everything is normal under low or heavy usage. <br/>

2. Cluster network usage is very low<br/>

3. No issues on Spark UI except that tasks begin to move from LOCAL to ANY</p>

<p><b>Cause (Corrected as found details in code):</b> <br/>

1. Node 'H' is doing marginally more work than the rest (being a little slower and at almost 100% CPU)<br/>

2. Scheduler hits the default 3000 millis spark.locality.wait and assigns the task to other nodes <br/>

3. One of the nodes 'X' that accepted the task will try to access the data from node 'H' HDD. This adds Network I/O to node and also some extra CPU for I/O.<br/>

4. 'X' time to complete increases ~5x as it goes over Network <br/>

5. Eventually, every node will have a task that is waiting to fetch that specific partition from node 'H' so cluster is basically blocked on a single node</p>

<p>What I managed to figure out from the code is that this is because if an RDD is cached, it will make use of BlockManager.getRemote() and will not recompute the DAG part that resulted in this RDD and hence always hit the node that has cached the RDD.</p>

<ul>

<li>Proposed Fix \*<br/>

I have not worked with Scala &amp; Spark source code enough to propose a code fix, but on a high level, when a task hits the 'spark.locality.wait' timeout, it could make use of a new configuration e.g. recomputeRddAfterLocalityTimeout instead of always trying to get the cached RDD. This would be very useful if it could also be manually set on the RDD.</li>

</ul>

<p><b>Workaround</b><br/>

Playing with 'spark.locality.wait' values, there is a deterministic value depending on partitions and config where the problem ceases to exist.</p>

<p><b>PS1</b> : Don't have enough Scala skils to follow the issue or propose a fix, but I hope that this has enough information to make sense.<br/>

<b>PS2</b> : Debugging this issue made me realize that there can be a lot of use-cases that trigger this behaviour</p>

Description: <p>Currently RandomSampler.sample only accepts Scala iterator. We should also let it accept Java iterator for better compatibility.</p>

Description: <p>spark/pom.xml:<br/>

&lt;plugin&gt;<br/>

&lt;groupId&gt;org.apache.maven.plugins&lt;/groupId&gt;<br/>

&lt;artifactId&gt;maven-jar-plugin&lt;/artifactId&gt;<br/>

&lt;executions&gt;<br/>

&lt;execution&gt;<br/>

&lt;id&gt;prepare-test-jar&lt;/id&gt;<br/>

&lt;phase&gt;prepare-package&lt;/phase&gt;<br/>

&lt;goals&gt;<br/>

&lt;goal&gt;test-jar&lt;/goal&gt;<br/>

&lt;/goals&gt;<br/>

&lt;configuration&gt;<br/>

&lt;excludes&gt;<br/>

&lt;exclude&gt;log4j.properties&lt;/exclude&gt;<br/>

&lt;/excludes&gt;<br/>

&lt;/configuration&gt;<br/>

&lt;/execution&gt;<br/>

&lt;/executions&gt;<br/>

&lt;/plugin&gt;<br/>

-----------------------------------------------------------------------<br/>

spark/network/common/pom.xml:<br/>

&lt;plugin&gt;<br/>

&lt;groupId&gt;org.apache.maven.plugins&lt;/groupId&gt;<br/>

&lt;artifactId&gt;maven-jar-plugin&lt;/artifactId&gt;<br/>

&lt;executions&gt;<br/>

&lt;execution&gt;<br/>

&lt;id&gt;test-jar-on-test-compile&lt;/id&gt;<br/>

&lt;phase&gt;test-compile&lt;/phase&gt;<br/>

&lt;goals&gt;<br/>

&lt;goal&gt;test-jar&lt;/goal&gt;<br/>

&lt;/goals&gt;<br/>

&lt;/execution&gt;<br/>

&lt;/executions&gt;<br/>

&lt;/plugin&gt;</p>

<p>maven-jar-plugin:test-jar goal is bound to both phase "test-compile" and phase "test-jar". As a result, when I tried to deploy spark to nexus repository, test.jar of spark-network-common\_2.10 was tried to be uploaded twice and failed.</p>

Description: <p>Only Spark test code uses jblas at this point. For completeness, these relatively few simple calls should be replaced with equivalent calls to breeze classes, or in some cases maybe even just double arrays.</p>

Description: <p>Current ConnectedComponents algorithm was based on Min-VertexId Propagation, which is sensitive to the place of Min-VertexId.<br/>

While this implementation is based on Max-Degree Propagation.<br/>

First, the degree graph is computed. And in the pregel progress, the vertex with the max degree in a CC is the start point of propagation.<br/>

This new method has advantages over the old one:<br/>

1, The convergence is only determined by the structs of CC, and is robust to the place of vertex with Min-ID.<br/>

2, For spherical CCs in which there may be a concept like 'center', it can accelerate the convergence. For example, GraphGenerators.gridGraph(sc, 3, 3), the old CC need 4 supersteps, while the new one only need 2 supersteps.<br/>

3, If we limit the number of iteration, the new method tend to generate more acceptable results.<br/>

4, The output for each CC is the vertex with max degree in it, which may be more meaningful. And because the vertex-ID is nominal in most cases, the vertex with min-ID in a CC is somewhat meanless.</p>

<p>But there are still two disadvantages:<br/>

1,The message body grows, from (VID) to (VID, Degree). that is (Long) -&gt; (Long, Int)<br/>

2,For graph with simple CCs, it may be slower than old one. Because it need a extra degree computation.</p>

<p>The api is the same like ConnectedComponents:</p>

<p>val graph = ...<br/>

val cc = graph.ConnectedComponentsWithDegree(100)<br/>

or<br/>

val cc = ConnectedComponentsWithDegree.run(graph, 100)</p>

Description: <p>Replace the current SQL Parser by a version written in ANTLR4. This has various advantages:</p>

<ul>

<li>Much simpler structure</li>

<li>No code blowup</li>

<li>Reduction in lines of code</li>

</ul>

Description: <p>Another Meta method for multi-class classification.</p>

<p>Most classification algorithms were designed for balanced data.<br/>

The OneVsRest method will generate K models on imbalanced data.<br/>

The OneVsOne will train K\*(K-1)/2 models on balanced data.</p>

<p>OneVsOne is less sensitive to the problems of imbalanced datasets, and can usually result in higher precision.<br/>

But it is much more computationally expensive, although each model are trained on a much smaller dataset. (2/K of total)</p>

<p>The OneVsOne is implemented in the way OneVsRest did:</p>

<p>val classifier = new LogisticRegression()<br/>

val ovo = new OneVsOne()<br/>

ovo.setClassifier(classifier)<br/>

val ovoModel = ovo.fit(data)<br/>

val predictions = ovoModel.transform(data)</p>

Description: <p>In my application Java spark context is created with an unavailable master URL (you may assume master is down for a maintenance). When creating Java spark context it leads to stopping JVM that runs spark driver with JVM exit code 50.</p>

<p>When I checked the logs I found SparkUncaughtExceptionHandler calling the System.exit. My program should run forever. </p>

<p>package test.mains;</p>

<p>import org.apache.spark.SparkConf;<br/>

import org.apache.spark.api.java.JavaSparkContext;</p>

<p>public class CheckJavaSparkContext {</p>

<p> public static void main(String[] args) {</p>

<p> SparkConf conf = new SparkConf();<br/>

conf.setAppName("test");<br/>

conf.setMaster("spark://sunshinee:7077");</p>

<p> try </p>

{

new JavaSparkContext(conf);

}

<p> catch (Throwable e) </p>

{

System.out.println("Caught an exception : " + e.getMessage());

}

<p> System.out.println("Waiting to complete...");<br/>

while (true) {<br/>

}<br/>

}</p>

<p>}</p>

<p>Output log</p>

<p>Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties<br/>

SLF4J: Class path contains multiple SLF4J bindings.<br/>

SLF4J: Found binding in <span class="error">&#91;jar:file:/data/downloads/spark-1.6.0-bin-hadoop2.6/lib/spark-assembly-1.6.0-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBinder.class&#93;</span><br/>

SLF4J: Found binding in <span class="error">&#91;jar:file:/data/downloads/spark-1.6.0-bin-hadoop2.6/lib/spark-examples-1.6.0-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBinder.class&#93;</span><br/>

SLF4J: See <a href="http://www.slf4j.org/codes.html#multiple\_bindings" class="external-link" rel="nofollow">http://www.slf4j.org/codes.html#multiple\_bindings</a> for an explanation.<br/>

SLF4J: Actual binding is of type <span class="error">&#91;org.slf4j.impl.Log4jLoggerFactory&#93;</span><br/>

16/03/04 18:01:15 INFO SparkContext: Running Spark version 1.6.0<br/>

16/03/04 18:01:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable<br/>

16/03/04 18:01:17 WARN Utils: Your hostname, pesamara-mobl-vm1 resolves to a loopback address: 127.0.0.1; using 10.30.9.107 instead (on interface eth0)<br/>

16/03/04 18:01:17 WARN Utils: Set SPARK\_LOCAL\_IP if you need to bind to another address<br/>

16/03/04 18:01:18 INFO SecurityManager: Changing view acls to: ps40233<br/>

16/03/04 18:01:18 INFO SecurityManager: Changing modify acls to: ps40233<br/>

16/03/04 18:01:18 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(ps40233); users with modify permissions: Set(ps40233)<br/>

16/03/04 18:01:19 INFO Utils: Successfully started service 'sparkDriver' on port 55309.<br/>

16/03/04 18:01:21 INFO Slf4jLogger: Slf4jLogger started<br/>

16/03/04 18:01:21 INFO Remoting: Starting remoting<br/>

16/03/04 18:01:22 INFO Remoting: Remoting started; listening on addresses :<span class="error">&#91;akka.tcp://sparkDriverActorSystem@10.30.9.107:52128&#93;</span><br/>

16/03/04 18:01:22 INFO Utils: Successfully started service 'sparkDriverActorSystem' on port 52128.<br/>

16/03/04 18:01:22 INFO SparkEnv: Registering MapOutputTracker<br/>

16/03/04 18:01:22 INFO SparkEnv: Registering BlockManagerMaster<br/>

16/03/04 18:01:22 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-87c20178-357d-4252-a46a-62a755568a98<br/>

16/03/04 18:01:22 INFO MemoryStore: MemoryStore started with capacity 457.7 MB<br/>

16/03/04 18:01:22 INFO SparkEnv: Registering OutputCommitCoordinator<br/>

16/03/04 18:01:23 INFO Utils: Successfully started service 'SparkUI' on port 4040.<br/>

16/03/04 18:01:23 INFO SparkUI: Started SparkUI at <a href="http://10.30.9.107:4040" class="external-link" rel="nofollow">http://10.30.9.107:4040</a><br/>

16/03/04 18:01:24 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:01:24 WARN AppClient$ClientEndpoint: Failed to connect to master sunshinee:7077<br/>

java.io.IOException: Failed to connect to sunshinee:7077<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216) at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:167)<br/>

at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:200)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:187)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:183)<br/>

at java.util.concurrent.FutureTask.run(FutureTask.java:266)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

Caused by: java.nio.channels.UnresolvedAddressException<br/>

at sun.nio.ch.Net.checkAddress(Net.java:123)<br/>

at sun.nio.ch.SocketChannelImpl.connect(SocketChannelImpl.java:621)<br/>

at io.netty.channel.socket.nio.NioSocketChannel.doConnect(NioSocketChannel.java:209)<br/>

at io.netty.channel.nio.AbstractNioChannel$AbstractNioUnsafe.connect(AbstractNioChannel.java:207)<br/>

at io.netty.channel.DefaultChannelPipeline$HeadContext.connect(DefaultChannelPipeline.java:1097)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelOutboundHandlerAdapter.connect(ChannelOutboundHandlerAdapter.java:47)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelDuplexHandler.connect(ChannelDuplexHandler.java:50)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:438)<br/>

at io.netty.channel.DefaultChannelPipeline.connect(DefaultChannelPipeline.java:908)<br/>

at io.netty.channel.AbstractChannel.connect(AbstractChannel.java:203)<br/>

at io.netty.bootstrap.Bootstrap$2.run(Bootstrap.java:166)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecutor.java:357)<br/>

at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:357)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:111)<br/>

... 1 more<br/>

16/03/04 18:01:44 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:01:44 WARN AppClient$ClientEndpoint: Failed to connect to master sunshinee:7077<br/>

java.io.IOException: Failed to connect to sunshinee:7077<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:167)<br/>

at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:200)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:187)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:183)<br/>

at java.util.concurrent.FutureTask.run(FutureTask.java:266)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

Caused by: java.nio.channels.UnresolvedAddressException<br/>

at sun.nio.ch.Net.checkAddress(Net.java:123)<br/>

at sun.nio.ch.SocketChannelImpl.connect(SocketChannelImpl.java:621)<br/>

at io.netty.channel.socket.nio.NioSocketChannel.doConnect(NioSocketChannel.java:209)<br/>

at io.netty.channel.nio.AbstractNioChannel$AbstractNioUnsafe.connect(AbstractNioChannel.java:207)<br/>

at io.netty.channel.DefaultChannelPipeline$HeadContext.connect(DefaultChannelPipeline.java:1097)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelOutboundHandlerAdapter.connect(ChannelOutboundHandlerAdapter.java:47)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelDuplexHandler.connect(ChannelDuplexHandler.java:50)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:438)<br/>

at io.netty.channel.DefaultChannelPipeline.connect(DefaultChannelPipeline.java:908)<br/>

at io.netty.channel.AbstractChannel.connect(AbstractChannel.java:203)<br/>

at io.netty.bootstrap.Bootstrap$2.run(Bootstrap.java:166)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecutor.java:357)<br/>

at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:357)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:111)<br/>

... 1 more<br/>

16/03/04 18:02:04 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:02:04 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:02:04 WARN AppClient$ClientEndpoint: Failed to connect to master sunshinee:7077<br/>

java.io.IOException: Failed to connect to sunshinee:7077<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:167)<br/>

at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:200)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:187)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:183)<br/>

at java.util.concurrent.FutureTask.run(FutureTask.java:266)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

Caused by: java.nio.channels.UnresolvedAddressException<br/>

at sun.nio.ch.Net.checkAddress(Net.java:123)<br/>

at sun.nio.ch.SocketChannelImpl.connect(SocketChannelImpl.java:621)<br/>

at io.netty.channel.socket.nio.NioSocketChannel.doConnect(NioSocketChannel.java:209)<br/>

at io.netty.channel.nio.AbstractNioChannel$AbstractNioUnsafe.connect(AbstractNioChannel.java:207)<br/>

at io.netty.channel.DefaultChannelPipeline$HeadContext.connect(DefaultChannelPipeline.java:1097)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelOutboundHandlerAdapter.connect(ChannelOutboundHandlerAdapter.java:47)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelDuplexHandler.connect(ChannelDuplexHandler.java:50)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:438)<br/>

at io.netty.channel.DefaultChannelPipeline.connect(DefaultChannelPipeline.java:908)<br/>

at io.netty.channel.AbstractChannel.connect(AbstractChannel.java:203)<br/>

at io.netty.bootstrap.Bootstrap$2.run(Bootstrap.java:166)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecutor.java:357)<br/>

at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:357)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:111)<br/>

... 1 more<br/>

16/03/04 18:02:24 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:02:24 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:02:24 WARN SparkDeploySchedulerBackend: Application ID is not initialized yet.<br/>

16/03/04 18:02:24 ERROR SparkDeploySchedulerBackend: Application has been killed. Reason: All masters are unresponsive! Giving up.<br/>

16/03/04 18:02:24 WARN AppClient$ClientEndpoint: Failed to connect to master sunshinee:7077<br/>

java.io.IOException: Failed to connect to sunshinee:7077<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:167)<br/>

at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:200)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:187)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:183)<br/>

at java.util.concurrent.FutureTask.run(FutureTask.java:266)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

Caused by: java.nio.channels.UnresolvedAddressException<br/>

at sun.nio.ch.Net.checkAddress(Net.java:123)<br/>

at sun.nio.ch.SocketChannelImpl.connect(SocketChannelImpl.java:621)<br/>

at io.netty.channel.socket.nio.NioSocketChannel.doConnect(NioSocketChannel.java:209)<br/>

at io.netty.channel.nio.AbstractNioChannel$AbstractNioUnsafe.connect(AbstractNioChannel.java:207)<br/>

at io.netty.channel.DefaultChannelPipeline$HeadContext.connect(DefaultChannelPipeline.java:1097)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelOutboundHandlerAdapter.connect(ChannelOutboundHandlerAdapter.java:47)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelDuplexHandler.connect(ChannelDuplexHandler.java:50)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:438)<br/>

at io.netty.channel.DefaultChannelPipeline.connect(DefaultChannelPipeline.java:908)<br/>

at io.netty.channel.AbstractChannel.connect(AbstractChannel.java:203)<br/>

at io.netty.bootstrap.Bootstrap$2.run(Bootstrap.java:166)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecutor.java:357)<br/>

at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:357)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:111)<br/>

... 1 more<br/>

16/03/04 18:02:24 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 44298.<br/>

16/03/04 18:02:24 INFO NettyBlockTransferService: Server created on 44298<br/>

16/03/04 18:02:24 INFO BlockManagerMaster: Trying to register BlockManager<br/>

16/03/04 18:02:24 INFO BlockManagerMasterEndpoint: Registering block manager 10.30.9.107:44298 with 457.7 MB RAM, BlockManagerId(driver, 10.30.9.107, 44298)<br/>

16/03/04 18:02:24 INFO BlockManagerMaster: Registered BlockManager<br/>

16/03/04 18:02:24 INFO SparkUI: Stopped Spark web UI at <a href="http://10.30.9.107:4040" class="external-link" rel="nofollow">http://10.30.9.107:4040</a><br/>

16/03/04 18:02:24 INFO SparkDeploySchedulerBackend: Shutting down all executors<br/>

16/03/04 18:02:24 INFO SparkDeploySchedulerBackend: Asking each executor to shut down<br/>

16/03/04 18:02:24 WARN AppClient$ClientEndpoint: Drop UnregisterApplication(null) because has not yet connected to master<br/>

16/03/04 18:02:24 ERROR SparkUncaughtExceptionHandler: Uncaught exception in thread Thread<span class="error">&#91;appclient-registration-retry-thread,5,main&#93;</span><br/>

java.lang.InterruptedException<br/>

at java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireSharedNanos(AbstractQueuedSynchronizer.java:1039)<br/>

at java.util.concurrent.locks.AbstractQueuedSynchronizer.tryAcquireSharedNanos(AbstractQueuedSynchronizer.java:1328)<br/>

at scala.concurrent.impl.Promise$DefaultPromise.tryAwait(Promise.scala:208)<br/>

at scala.concurrent.impl.Promise$DefaultPromise.ready(Promise.scala:218)<br/>

at scala.concurrent.impl.Promise$DefaultPromise.result(Promise.scala:223)<br/>

at scala.concurrent.Await$$anonfun$result$1.apply(package.scala:107)<br/>

at scala.concurrent.BlockContext$DefaultBlockContext$.blockOn(BlockContext.scala:53)<br/>

at scala.concurrent.Await$.result(package.scala:107)<br/>

at org.apache.spark.rpc.RpcTimeout.awaitResult(RpcTimeout.scala:75)<br/>

at org.apache.spark.deploy.client.AppClient.stop(AppClient.scala:290)<br/>

at org.apache.spark.scheduler.cluster.SparkDeploySchedulerBackend.org$apache$spark$scheduler$cluster$SparkDeploySchedulerBackend$$stop(SparkDeploySchedulerBackend.scala:198)<br/>

at org.apache.spark.scheduler.cluster.SparkDeploySchedulerBackend.stop(SparkDeploySchedulerBackend.scala:101)<br/>

at org.apache.spark.scheduler.TaskSchedulerImpl.stop(TaskSchedulerImpl.scala:446)<br/>

at org.apache.spark.scheduler.DAGScheduler.stop(DAGScheduler.scala:1582)<br/>

at org.apache.spark.SparkContext$$anonfun$stop$7.apply$mcV$sp(SparkContext.scala:1731)<br/>

at org.apache.spark.util.Utils$.tryLogNonFatalError(Utils.scala:1229)<br/>

at org.apache.spark.SparkContext.stop(SparkContext.scala:1730)<br/>

at org.apache.spark.scheduler.cluster.SparkDeploySchedulerBackend.dead(SparkDeploySchedulerBackend.scala:127)<br/>

at org.apache.spark.deploy.client.AppClient$ClientEndpoint.markDead(AppClient.scala:264)<br/>

at org.apache.spark.deploy.client.AppClient$ClientEndpoint$$anon$2$$anonfun$run$1.apply$mcV$sp(AppClient.scala:134)<br/>

at org.apache.spark.util.Utils$.tryOrExit(Utils.scala:1163)<br/>

at org.apache.spark.deploy.client.AppClient$ClientEndpoint$$anon$2.run(AppClient.scala:129)<br/>

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)<br/>

at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308)<br/>

at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$301(ScheduledThreadPoolExecutor.java:180)<br/>

at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:294)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

16/03/04 18:02:24 INFO DiskBlockManager: Shutdown hook called<br/>

16/03/04 18:02:24 INFO ShutdownHookManager: Shutdown hook called<br/>

16/03/04 18:02:24 INFO ShutdownHookManager: Deleting directory /tmp/spark-ea68a0fa-4f0d-4dbb-8407-cce90ef78a52<br/>

16/03/04 18:02:24 INFO ShutdownHookManager: Deleting directory /tmp/spark-ea68a0fa-4f0d-4dbb-8407-cce90ef78a52/userFiles-db548748-a55c-4406-adcb-c09e63b118bd<br/>

Java Result: 50 </p>

Description: <p>On Windows, when we launch <tt>bin\spark-shell.cmd</tt>, it shows ERROR message and stacktrace.</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>C:\Users\tsudukim\Documents\workspace\spark-dev3&gt;bin\spark-shell

[ERROR] Terminal initialization failed; falling back to unsupported

java.lang.NoClassDefFoundError: Could not initialize class scala.tools.fusesource\_embedded.jansi.internal.Kernel32

at scala.tools.fusesource\_embedded.jansi.internal.WindowsSupport.getConsoleMode(WindowsSupport.java:50)

at scala.tools.jline\_embedded.WindowsTerminal.getConsoleMode(WindowsTerminal.java:204)

at scala.tools.jline\_embedded.WindowsTerminal.init(WindowsTerminal.java:82)

at scala.tools.jline\_embedded.TerminalFactory.create(TerminalFactory.java:101)

at scala.tools.jline\_embedded.TerminalFactory.get(TerminalFactory.java:158)

at scala.tools.jline\_embedded.console.ConsoleReader.&lt;init&gt;(ConsoleReader.java:229)

at scala.tools.jline\_embedded.console.ConsoleReader.&lt;init&gt;(ConsoleReader.java:221)

at scala.tools.jline\_embedded.console.ConsoleReader.&lt;init&gt;(ConsoleReader.java:209)

at scala.tools.nsc.interpreter.jline\_embedded.JLineConsoleReader.&lt;init&gt;(JLineReader.scala:61)

at scala.tools.nsc.interpreter.jline\_embedded.InteractiveReader.&lt;init&gt;(JLineReader.scala:33)

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)

at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)

at java.lang.reflect.Constructor.newInstance(Constructor.java:422)

at scala.tools.nsc.interpreter.ILoop$$anonfun$scala$tools$nsc$interpreter$ILoop$$instantiate$1$1.apply(ILoop.scala:865)

at scala.tools.nsc.interpreter.ILoop$$anonfun$scala$tools$nsc$interpreter$ILoop$$instantiate$1$1.apply(ILoop.scala:862)

at scala.tools.nsc.interpreter.ILoop.scala$tools$nsc$interpreter$ILoop$$mkReader$1(ILoop.scala:871)

at scala.tools.nsc.interpreter.ILoop$$anonfun$15$$anonfun$apply$8.apply(ILoop.scala:875)

at scala.tools.nsc.interpreter.ILoop$$anonfun$15$$anonfun$apply$8.apply(ILoop.scala:875)

at scala.util.Try$.apply(Try.scala:192)

at scala.tools.nsc.interpreter.ILoop$$anonfun$15.apply(ILoop.scala:875)

at scala.tools.nsc.interpreter.ILoop$$anonfun$15.apply(ILoop.scala:875)

at scala.collection.immutable.Stream$$anonfun$map$1.apply(Stream.scala:418)

at scala.collection.immutable.Stream$$anonfun$map$1.apply(Stream.scala:418)

at scala.collection.immutable.Stream$Cons.tail(Stream.scala:1233)

at scala.collection.immutable.Stream$Cons.tail(Stream.scala:1223)

at scala.collection.immutable.Stream.collect(Stream.scala:435)

at scala.tools.nsc.interpreter.ILoop.chooseReader(ILoop.scala:877)

at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1$$anonfun$apply$mcZ$sp$2.apply(ILoop.scala:916)

at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply$mcZ$sp(ILoop.scala:916)

at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply(ILoop.scala:911)

at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply(ILoop.scala:911)

at scala.reflect.internal.util.ScalaClassLoader$.savingContextLoader(ScalaClassLoader.scala:97)

at scala.tools.nsc.interpreter.ILoop.process(ILoop.scala:911)

at org.apache.spark.repl.Main$.doMain(Main.scala:64)

at org.apache.spark.repl.Main$.main(Main.scala:47)

at org.apache.spark.repl.Main.main(Main.scala)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:497)

at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:737)

at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:183)

at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:208)

at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:122)

at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel).

16/03/07 13:05:32 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Spark context available as sc (master = local[\*], app id = local-1457323533704).

SQL context available as sqlContext.

Welcome to

\_\_\_\_ \_\_

/ \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_

\_\ \/ \_ \/ \_ `/ \_\_/ '\_/

/\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 2.0.0-SNAPSHOT

/\_/

Using Scala version 2.11.7 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0\_40)

Type in expressions to have them evaluated.

Type :help for more information.

scala&gt; sc.textFile("README.md")

res0: org.apache.spark.rdd.RDD[String] = README.md MapPartitionsRDD[1] at textFile at &lt;console&gt;:25

scala&gt; sc.textFile("README.md").count()

res1: Long = 97

</pre>

</div></div>

<p>Spark-shell itself seems to work file during my simple operation check.</p>

Description: <p>There is a problem decoding Avro data with SparkSQL when partitioned. The schema and encoded data are valid &#8211; I'm able to decode the data with the avro-tools CLI utility. I'm also able to decode the data with non-partitioned SparkSQL tables, Hive, other tools as well... except partitioned SparkSQL schemas.</p>

<p>For a simple example, I took the example schema and data found in the Oracle documentation here:</p>

<p><b>Schema</b></p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-javascript">

{

"type": "record",

"name": "MemberInfo",

"namespace": "avro",

"fields": [

{"name": "name", "type": {

"type": "record",

"name": "FullName",

"fields": [

{"name": "first", "type": "string"},

{"name": "last", "type": "string"}

]

}},

{"name": "age", "type": "<span class="code-keyword"><span class="code-object">int</span></span>"},

{"name": "address", "type": {

"type": "record",

"name": "Address",

"fields": [

{"name": "street", "type": "string"},

{"name": "city", "type": "string"},

{"name": "state", "type": "string"},

{"name": "zip", "type": "<span class="code-keyword"><span class="code-object">int</span></span>"}

]

}}

]

}

</pre>

</div></div>

<p><b>Data</b></p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-javascript">

{

"name": {

"first": "Percival",

"last": "Lowell"

},

"age": 156,

"address": {

"street": "Mars Hill Rd",

"city": "Flagstaff",

"state": "AZ",

"zip": 86001

}

}

</pre>

</div></div>

<p><b>Create</b> (no partitions - works)<br/>

If I create with no partitions, I'm able to query the data just fine.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-sql">

CREATE EXTERNAL TABLE IF NOT EXISTS foo

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'

STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'

OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'

LOCATION '/path/to/data/dir'

TBLPROPERTIES ('avro.schema.url'='/path/to/schema.avsc');

</pre>

</div></div>

<p><b>Create</b> (partitions &#8211; does NOT work)<br/>

If I create with no partitions, and then manually add a partition, all of my queries return an error. (I need to manually add partitions because I cannot control the structure of the data directories, so dynamic partitioning is not an option.)</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-sql">

CREATE EXTERNAL TABLE IF NOT EXISTS foo

PARTITIONED <span class="code-keyword">BY</span> (ds STRING)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'

STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'

OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'

TBLPROPERTIES ('avro.schema.url'='/path/to/schema.avsc');

ALTER TABLE foo ADD PARTITION (ds='1') LOCATION '/path/to/data/dir';

</pre>

</div></div>

<p>The error:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

spark-sql&gt; SELECT \* FROM foo WHERE ds = '1';

Driver stacktrace:

at org.apache.spark.scheduler.DAGScheduler.org$apache$spark$scheduler$DAGScheduler$$failJobAndIndependentStages(DAGScheduler.scala:1431)

at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.apply(DAGScheduler.scala:1419)

at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.apply(DAGScheduler.scala:1418)

at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)

at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:47)

at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGScheduler.scala:1418)

at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSetFailed$1.apply(DAGScheduler.scala:799)

at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSetFailed$1.apply(DAGScheduler.scala:799)

at scala.Option.foreach(Option.scala:236)

at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed(DAGScheduler.scala:799)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOnReceive(DAGScheduler.scala:1640)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1599)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1588)

at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:48)

at org.apache.spark.scheduler.DAGScheduler.runJob(DAGScheduler.scala:620)

at org.apache.spark.SparkContext.runJob(SparkContext.scala:1832)

at org.apache.spark.SparkContext.runJob(SparkContext.scala:1845)

at org.apache.spark.SparkContext.runJob(SparkContext.scala:1858)

at org.apache.spark.SparkContext.runJob(SparkContext.scala:1929)

at org.apache.spark.rdd.RDD$$anonfun$collect$1.apply(RDD.scala:927)

at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:150)

at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:111)

at org.apache.spark.rdd.RDD.withScope(RDD.scala:316)

at org.apache.spark.rdd.RDD.collect(RDD.scala:926)

at org.apache.spark.sql.execution.SparkPlan.executeCollect(SparkPlan.scala:166)

at org.apache.spark.sql.execution.SparkPlan.executeCollectPublic(SparkPlan.scala:174)

at org.apache.spark.sql.hive.HiveContext$QueryExecution.stringResult(HiveContext.scala:635)

at org.apache.spark.sql.hive.thriftserver.SparkSQLDriver.run(SparkSQLDriver.scala:64)

at org.apache.spark.sql.hive.thriftserver.SparkSQLCLIDriver.processCmd(SparkSQLCLIDriver.scala:308)

at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:376)

at org.apache.spark.sql.hive.thriftserver.SparkSQLCLIDriver$.main(SparkSQLCLIDriver.scala:226)

at org.apache.spark.sql.hive.thriftserver.SparkSQLCLIDriver.main(SparkSQLCLIDriver.scala)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:483)

at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:731)

at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:181)

at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:206)

at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:121)

at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

Caused by: org.apache.avro.AvroTypeException: Found avro.FullName, expecting union

at org.apache.avro.io.ResolvingDecoder.doAction(ResolvingDecoder.java:292)

at org.apache.avro.io.parsing.Parser.advance(Parser.java:88)

at org.apache.avro.io.ResolvingDecoder.readIndex(ResolvingDecoder.java:267)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.read(GenericDatumReader.java:155)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.readField(GenericDatumReader.java:193)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.readRecord(GenericDatumReader.java:183)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.read(GenericDatumReader.java:151)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.read(GenericDatumReader.java:142)

at org.apache.hadoop.hive.serde2.avro.AvroDeserializer$SchemaReEncoder.reencode(AvroDeserializer.java:111)

at org.apache.hadoop.hive.serde2.avro.AvroDeserializer.deserialize(AvroDeserializer.java:175)

at org.apache.hadoop.hive.serde2.avro.AvroSerDe.deserialize(AvroSerDe.java:201)

at org.apache.spark.sql.hive.HadoopTableReader$$anonfun$fillObject$2.apply(TableReader.scala:409)

at org.apache.spark.sql.hive.HadoopTableReader$$anonfun$fillObject$2.apply(TableReader.scala:408)

at scala.collection.Iterator$$anon$11.next(Iterator.scala:328)

at scala.collection.Iterator$$anon$11.next(Iterator.scala:328)

at scala.collection.Iterator$class.foreach(Iterator.scala:727)

at scala.collection.AbstractIterator.foreach(Iterator.scala:1157)

at scala.collection.<span class="code-keyword">generic</span>.Growable$class.$plus$plus$eq(Growable.scala:48)

at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:103)

at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:47)

at scala.collection.TraversableOnce$class.to(TraversableOnce.scala:273)

at scala.collection.AbstractIterator.to(Iterator.scala:1157)

at scala.collection.TraversableOnce$class.toBuffer(TraversableOnce.scala:265)

at scala.collection.AbstractIterator.toBuffer(Iterator.scala:1157)

at scala.collection.TraversableOnce$class.toArray(TraversableOnce.scala:252)

at scala.collection.AbstractIterator.toArray(Iterator.scala:1157)

at org.apache.spark.rdd.RDD$$anonfun$collect$1$$anonfun$12.apply(RDD.scala:927)

at org.apache.spark.rdd.RDD$$anonfun$collect$1$$anonfun$12.apply(RDD.scala:927)

at org.apache.spark.SparkContext$$anonfun$runJob$5.apply(SparkContext.scala:1858)

at org.apache.spark.SparkContext$$anonfun$runJob$5.apply(SparkContext.scala:1858)

at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:66)

at org.apache.spark.scheduler.Task.run(Task.scala:89)

at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:213)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)

at java.lang.<span class="code-object">Thread</span>.run(<span class="code-object">Thread</span>.java:745)

</pre>

</div></div>

<p><b>Additional Info</b><br/>

For what it's worth, I found an issue (<a href="https://issues.apache.org/jira/browse/DRILL-957" title="Avro partitioned table fails with AvroTypeException" class="issue-link" data-issue-key="DRILL-957"><del>DRILL-957</del></a>) reported in Apache Drill and related fix that look very simliar to this. I'll look that to this issue.</p>

<p>Originally <a href="http://stackoverflow.com/questions/35826850/spark-unable-to-decode-avro-when-partitioned" class="external-link" rel="nofollow">posted here</a> on StackOverflow as a question, but I felt strongly that this is indeed a bug so I created this issue.</p>

Description: <p>16/03/06 11:49:18 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)<br/>

16/03/06 11:49:18 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)<br/>

16/03/06 11:49:24 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 1.2.0<br/>

16/03/06 11:49:24 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException<br/>

16/03/06 11:49:25 WARN : Your hostname, LAPTOP-B9GV6F82 resolves to a loopback/non-reachable address: fe80:0:0:0:1cbd:55b:b3e4:8e04%12, but we couldn't find any external IP address!<br/>

java.lang.RuntimeException: java.lang.NullPointerException<br/>

at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:522)<br/>

at org.apache.spark.sql.hive.client.ClientWrapper.&lt;init&gt;(ClientWrapper.scala:171)<br/>

at org.apache.spark.sql.hive.HiveContext.executionHive$lzycompute(HiveContext.scala:162)<br/>

at org.apache.spark.sql.hive.HiveContext.executionHive(HiveContext.scala:160)<br/>

at org.apache.spark.sql.hive.HiveContext.&lt;init&gt;(HiveContext.scala:167)<br/>

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)<br/>

at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)<br/>

at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source)<br/>

at java.lang.reflect.Constructor.newInstance(Unknown Source)<br/>

at org.apache.spark.repl.SparkILoop.createSQLContext(SparkILoop.scala:1028)<br/>

at $iwC$$iwC.&lt;init&gt;(&lt;console&gt;:9)<br/>

at $iwC.&lt;init&gt;(&lt;console&gt;:18)<br/>

at &lt;init&gt;(&lt;console&gt;:20)<br/>

at .&lt;init&gt;(&lt;console&gt;:24)<br/>

at .&lt;clinit&gt;(&lt;console&gt;)<br/>

at .&lt;init&gt;(&lt;console&gt;:7)<br/>

at .&lt;clinit&gt;(&lt;console&gt;)<br/>

at $print(&lt;console&gt;)<br/>

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)<br/>

at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)<br/>

at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)<br/>

at java.lang.reflect.Method.invoke(Unknown Source)<br/>

at org.apache.spark.repl.SparkIMain$ReadEvalPrint.call(SparkIMain.scala:1065)<br/>

at org.apache.spark.repl.SparkIMain$Request.loadAndRun(SparkIMain.scala:1340)<br/>

at org.apache.spark.repl.SparkIMain.loadAndRunReq$1(SparkIMain.scala:840)<br/>

at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:871)<br/>

at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:819)<br/>

at org.apache.spark.repl.SparkILoop.reallyInterpret$1(SparkILoop.scala:857)<br/>

at org.apache.spark.repl.SparkILoop.interpretStartingWith(SparkILoop.scala:902)<br/>

at org.apache.spark.repl.SparkILoop.command(SparkILoop.scala:814)<br/>

at org.apache.spark.repl.SparkILoopInit$$anonfun$initializeSpark$1.apply(SparkILoopInit.scala:132)<br/>

at org.apache.spark.repl.SparkILoopInit$$anonfun$initializeSpark$1.apply(SparkILoopInit.scala:124)<br/>

at org.apache.spark.repl.SparkIMain.beQuietDuring(SparkIMain.scala:324)<br/>

at org.apache.spark.repl.SparkILoopInit$class.initializeSpark(SparkILoopInit.scala:124)<br/>

at org.apache.spark.repl.SparkILoop.initializeSpark(SparkILoop.scala:64)<br/>

at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$process$1$$anonfun$apply$mcZ$sp$5.apply$mcV$sp(SparkILoop.scala:974)<br/>

at org.apache.spark.repl.SparkILoopInit$class.runThunks(SparkILoopInit.scala:159)<br/>

at org.apache.spark.repl.SparkILoop.runThunks(SparkILoop.scala:64)<br/>

at org.apache.spark.repl.SparkILoopInit$class.postInitialization(SparkILoopInit.scala:108)<br/>

at org.apache.spark.repl.SparkILoop.postInitialization(SparkILoop.scala:64)<br/>

at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$process$1.apply$mcZ$sp(SparkILoop.scala:991)<br/>

at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$process$1.apply(SparkILoop.scala:945)<br/>

at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$process$1.apply(SparkILoop.scala:945)<br/>

at scala.tools.nsc.util.ScalaClassLoader$.savingContextLoader(ScalaClassLoader.scala:135)<br/>

at org.apache.spark.repl.SparkILoop.org$apache$spark$repl$SparkILoop$$process(SparkILoop.scala:945)<br/>

at org.apache.spark.repl.SparkILoop.process(SparkILoop.scala:1059)<br/>

at org.apache.spark.repl.Main$.main(Main.scala:31)<br/>

at org.apache.spark.repl.Main.main(Main.scala)<br/>

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)<br/>

at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)<br/>

at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)<br/>

at java.lang.reflect.Method.invoke(Unknown Source)<br/>

at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:674)<br/>

at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:180)<br/>

at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:205)<br/>

at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:120)<br/>

at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)<br/>

Caused by: java.lang.NullPointerException<br/>

at java.lang.ProcessBuilder.start(Unknown Source)<br/>

at org.apache.hadoop.util.Shell.runCommand(Shell.java:482)<br/>

at org.apache.hadoop.util.Shell.run(Shell.java:455)<br/>

at org.apache.hadoop.util.Shell$ShellCommandExecutor.execute(Shell.java:715)<br/>

at org.apache.hadoop.util.Shell.execCommand(Shell.java:808)<br/>

at org.apache.hadoop.util.Shell.execCommand(Shell.java:791)<br/>

at org.apache.hadoop.fs.FileUtil.execCommand(FileUtil.java:1097)<br/>

at org.apache.hadoop.fs.RawLocalFileSystem$DeprecatedRawLocalFileStatus.loadPermissionInfo(RawLocalFileSystem.java:582)<br/>

at org.apache.hadoop.fs.RawLocalFileSystem$DeprecatedRawLocalFileStatus.getPermission(RawLocalFileSystem.java:557)<br/>

at org.apache.hadoop.hive.ql.session.SessionState.createRootHDFSDir(SessionState.java:599)<br/>

at org.apache.hadoop.hive.ql.session.SessionState.createSessionDirs(SessionState.java:554)<br/>

at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:508)<br/>

... 56 more</p>

Description: <p>'Streaming' tab on spark UI is misleading when the job has a window operation which changes the batch duration from original streaming context batch duration.</p>

<p>For instance consider:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

val streamingContext = <span class="code-keyword">new</span> StreamingContext(sparkConfig, Seconds(2))

val totalVideoImps = streamingContext.sparkContext.accumulator(0, <span class="code-quote">"TotalVideoImpressions"</span>)

val totalImps = streamingContext.sparkContext.accumulator(0, <span class="code-quote">"TotalImpressions"</span>)

val stream = KafkaReader.KafkaDirectStream(streamingContext)

stream.map(KafkaAdLogParser.parseAdLogRecord)

.filter(record =&gt; {

totalImps += 1

KafkaAdLogParser.isVideoRecord(record)

})

.map(record =&gt; {

totalVideoImps += 1

record.url

})

.window(Seconds(120))

.count().foreachRDD((rdd, time) =&gt; {

println(<span class="code-quote">"Timestamp: "</span> + ImpressionAggregator.millsToDate(time.milliseconds))

println(<span class="code-quote">"Count: "</span> + rdd.collect()(0))

println(<span class="code-quote">"Total Impressions: "</span> + totalImps.value)

totalImps.setValue(0)

println(<span class="code-quote">"Total Video Impressions: "</span> + totalVideoImps.value)

totalVideoImps.setValue(0)

})

streamingContext.start()

streamingContext.awaitTermination()

</pre>

</div></div>

<p>Batch Size before window operation is 2 sec and then after window batches are of 120 seconds each.<br/>

&#8211;</p>

<p>Above code printed following for my application whereas the UI showed different numbers.</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>Timestamp: 2016-03-06 12:02:56,000

Count: 362195

Total Impressions: 16882431

Total Video Impressions: 362195

Timestamp: 2016-03-06 12:04:56,000

Count: 367168

Total Impressions: 19480293

Total Video Impressions: 367168

Timestamp: 2016-03-06 12:06:56,000

Count: 177711

Total Impressions: 10196677

Total Video Impressions: 177711

</pre>

</div></div>

<p>whereas the spark UI shows different numbers as attached in the image. Also if we check the start and end index of kafka partition offsets reported by subsequent batch entries on UI, they do not result in all overall continuous range. All numbers are fine if we remove the window operation though.</p>

Description: <p>An example of how to use TrainValidationSplit in pyspark needs to be added. Should be consistent with the current examples. I'll submit a PR.</p>

Description: <p>Doc Issue : Apache Spark Streaming guide elaborates UpdateStateByKey Operation and goes onto give an example. However for complete code sample it referes to <tt>site.SPARK\_GITHUB\_URL</tt>/blob/master/examples/src/main/scala/org/apache<br/>

/spark/examples/streaming/StatefulNetworkWordCount.scala). StatefulNetworkWordCount.scala has changed to demonstrate more recent API mapWithState.<br/>

This creates confusion in the document. </p>

<p>Till the time more detailed explanation of mapWIthState is added the reference to StatefulNetworkWordCount.scala should be removed.</p>

Description: <p>In some cases, TaskSchedulerImpl.createTaskSetManager can be expensive. For example, in a Yarn cluster, it may call the topology script for rack awareness. When submit a very large job in a very large Yarn cluster, the topology script may take signifiant time to run. And this blocks receiving executors' heartbeats, which may result in lost executors</p>

<p>Stacktraces we observed which is related to this issue:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-quote">"dag-scheduler-event-loop"</span> daemon prio=10 tid=0x00007f8392875800 nid=0x26e8 runnable [0x00007f83576f4000]

java.lang.<span class="code-object">Thread</span>.State: RUNNABLE

at java.io.FileInputStream.readBytes(Native Method)

at java.io.FileInputStream.read(FileInputStream.java:272)

at java.io.BufferedInputStream.read1(BufferedInputStream.java:273)

at java.io.BufferedInputStream.read(BufferedInputStream.java:334)

- locked &lt;0x00000000f551f460&gt; (a java.lang.UNIXProcess$ProcessPipeInputStream)

at sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:283)

at sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:325)

at sun.nio.cs.StreamDecoder.read(StreamDecoder.java:177)

- locked &lt;0x00000000f5529740&gt; (a java.io.InputStreamReader)

at java.io.InputStreamReader.read(InputStreamReader.java:184)

at java.io.BufferedReader.fill(BufferedReader.java:154)

at java.io.BufferedReader.read1(BufferedReader.java:205)

at java.io.BufferedReader.read(BufferedReader.java:279)

- locked &lt;0x00000000f5529740&gt; (a java.io.InputStreamReader)

at org.apache.hadoop.util.Shell$ShellCommandExecutor.parseExecResult(Shell.java:728)

at org.apache.hadoop.util.Shell.runCommand(Shell.java:524)

at org.apache.hadoop.util.Shell.run(Shell.java:455)

at org.apache.hadoop.util.Shell$ShellCommandExecutor.execute(Shell.java:715)

at org.apache.hadoop.net.ScriptBasedMapping$RawScriptBasedMapping.runResolveCommand(ScriptBasedMapping.java:251)

at org.apache.hadoop.net.ScriptBasedMapping$RawScriptBasedMapping.resolve(ScriptBasedMapping.java:188)

at org.apache.hadoop.net.CachedDNSToSwitchMapping.resolve(CachedDNSToSwitchMapping.java:119)

at org.apache.hadoop.yarn.util.RackResolver.coreResolve(RackResolver.java:101)

at org.apache.hadoop.yarn.util.RackResolver.resolve(RackResolver.java:81)

at org.apache.spark.scheduler.cluster.YarnScheduler.getRackForHost(YarnScheduler.scala:38)

at org.apache.spark.scheduler.TaskSetManager$$anonfun$org$apache$spark$scheduler$TaskSetManager$$addPendingTask$1.apply(TaskSetManager.scala:210)

at org.apache.spark.scheduler.TaskSetManager$$anonfun$org$apache$spark$scheduler$TaskSetManager$$addPendingTask$1.apply(TaskSetManager.scala:189)

at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)

at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:47)

at org.apache.spark.scheduler.TaskSetManager.org$apache$spark$scheduler$TaskSetManager$$addPendingTask(TaskSetManager.scala:189)

at org.apache.spark.scheduler.TaskSetManager$$anonfun$1.apply$mcVI$sp(TaskSetManager.scala:158)

at scala.collection.immutable.Range.foreach$mVc$sp(Range.scala:141)

at org.apache.spark.scheduler.TaskSetManager.&lt;init&gt;(TaskSetManager.scala:157)

at org.apache.spark.scheduler.TaskSchedulerImpl.createTaskSetManager(TaskSchedulerImpl.scala:187)

at org.apache.spark.scheduler.TaskSchedulerImpl.submitTasks(TaskSchedulerImpl.scala:161)

- locked &lt;0x00000000ea3b8a88&gt; (a org.apache.spark.scheduler.cluster.YarnScheduler)

at org.apache.spark.scheduler.DAGScheduler.org$apache$spark$scheduler$DAGScheduler$$submitMissingTasks(DAGScheduler.scala:872)

at org.apache.spark.scheduler.DAGScheduler.org$apache$spark$scheduler$DAGScheduler$$submitStage(DAGScheduler.scala:778)

at org.apache.spark.scheduler.DAGScheduler.handleJobSubmitted(DAGScheduler.scala:762)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1362)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1354)

at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:48)

<span class="code-quote">"sparkDriver-akka.actor.<span class="code-keyword">default</span>-dispatcher-15"</span> daemon prio=10 tid=0x00007f829c020000 nid=0x2737 waiting <span class="code-keyword">for</span> monitor entry [0x00007f8355ebd000]

java.lang.<span class="code-object">Thread</span>.State: BLOCKED (on object monitor)

at org.apache.spark.scheduler.TaskSchedulerImpl.executorHeartbeatReceived(TaskSchedulerImpl.scala:362)

- waiting to lock &lt;0x00000000ea3b8a88&gt; (a org.apache.spark.scheduler.cluster.YarnScheduler)

at org.apache.spark.HeartbeatReceiver$$anonfun$receiveWithLogging$1.applyOrElse(HeartbeatReceiver.scala:46)

at scala.runtime.AbstractPartialFunction$mcVL$sp.apply$mcVL$sp(AbstractPartialFunction.scala:33)

at scala.runtime.AbstractPartialFunction$mcVL$sp.apply(AbstractPartialFunction.scala:33)

at scala.runtime.AbstractPartialFunction$mcVL$sp.apply(AbstractPartialFunction.scala:25)

at org.apache.spark.util.ActorLogReceive$$anon$1.apply(ActorLogReceive.scala:53)

at org.apache.spark.util.ActorLogReceive$$anon$1.apply(ActorLogReceive.scala:42)

at scala.PartialFunction$class.applyOrElse(PartialFunction.scala:118)

at org.apache.spark.util.ActorLogReceive$$anon$1.applyOrElse(ActorLogReceive.scala:42)

at akka.actor.ActorCell.receiveMessage(ActorCell.scala:498)

at akka.actor.ActorCell.invoke(ActorCell.scala:456)

at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:237)

at akka.dispatch.Mailbox.run(Mailbox.scala:219)

at akka.dispatch.ForkJoinExecutorConfigurator$AkkaForkJoinTask.exec(AbstractDispatcher.scala:386)

at scala.concurrent.forkjoin.ForkJoinTask.doExec(ForkJoinTask.java:260)

at scala.concurrent.forkjoin.ForkJoinPool$WorkQueue.runTask(ForkJoinPool.java:1339)

at scala.concurrent.forkjoin.ForkJoinPool.runWorker(ForkJoinPool.java:1979)

at scala.concurrent.forkjoin.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:107)

</pre>

</div></div>

Description:

Description: <p>Java 7 or higher supports `diamond` operator which replaces the type arguments required to invoke the constructor of a generic class with an empty set of type parameters (&lt;&gt;). Currently, Spark Java code use mixed usage of this. This issue replaces existing codes to use `diamond` operator and add Checkstyle rule.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

-List&lt;JavaPairDStream&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;&gt; kafkaStreams = <span class="code-keyword">new</span> ArrayList&lt;JavaPairDStream&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;&gt;(numStreams);

+List&lt;JavaPairDStream&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;&gt; kafkaStreams = <span class="code-keyword">new</span> ArrayList&lt;&gt;(numStreams);

</pre>

</div></div>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

-Set&lt;Tuple2&lt;<span class="code-object">Integer</span>, <span class="code-object">Integer</span>&gt;&gt; edges = <span class="code-keyword">new</span> HashSet&lt;Tuple2&lt;<span class="code-object">Integer</span>, <span class="code-object">Integer</span>&gt;&gt;(numEdges);

+Set&lt;Tuple2&lt;<span class="code-object">Integer</span>, <span class="code-object">Integer</span>&gt;&gt; edges = <span class="code-keyword">new</span> HashSet&lt;&gt;(numEdges);

</pre>

</div></div>

<p><b>Reference</b><br/>

<a href="https://docs.oracle.com/javase/8/docs/technotes/guides/language/type-inference-generic-instance-creation.html" class="external-link" rel="nofollow">https://docs.oracle.com/javase/8/docs/technotes/guides/language/type-inference-generic-instance-creation.html</a></p>

Description: <p>jblas fails on arm64.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

ALSSuite:

Exception encountered when attempting to run a suite with class name: org.apache.spark.mllib.recommendation.ALSSuite \*\*\* ABORTED \*\*\* (112 milliseconds)

java.lang.UnsatisfiedLinkError: org.jblas.NativeBlas.dgemm(CCIIID[DII[DIID[DII)V

at org.jblas.NativeBlas.dgemm(Native Method)

at org.jblas.SimpleBlas.gemm(SimpleBlas.java:247)

at org.jblas.DoubleMatrix.mmuli(DoubleMatrix.java:1781)

at org.jblas.DoubleMatrix.mmul(DoubleMatrix.java:3138)

at org.apache.spark.mllib.recommendation.ALSSuite$.generateRatings(ALSSuite.scala:74)

</pre>

</div></div>

Description: <p>Spark is great for synchronous operations.</p>

<p>But sometimes I need to call a database/web server/etc from my transform, and the Spark pipeline stalls waiting for it.</p>

<p>Avoiding that would be great!</p>

<p>I suggest we add a new method RDD.mapAsync(), which can execute these operations concurrently, avoiding the bottleneck.</p>

<p>I've written a quick'n'dirty implementation of what I have in mind: <br/>

<a href="https://gist.github.com/paulo-raca/d121cf27905cfb1fafc3" class="external-link" rel="nofollow">https://gist.github.com/paulo-raca/d121cf27905cfb1fafc3</a></p>

<p>What do you think?</p>

<p>If you agree with this feature, I can work on a pull request.</p>

Description: <p>Hi,</p>

<p>While writing the dataframe to HIVE table with "SaveMode.Overwrite" option.</p>

<p>E.g.<br/>

tgtFinal.write.mode(SaveMode.Overwrite).saveAsTable("tgt\_table")</p>

<p>sqlContext drop the table instead of truncating.</p>

<p>This is causing error while overwriting.</p>

<p>Adding stacktrace &amp; commands to reproduce the issue,</p>

<p>Thanks &amp; Regards,<br/>

Dhaval</p>

Description: <p>Analysis exception occurs while running the following query.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

SELECT ints FROM nestedArray LATERAL VIEW explode(a.b) `a` AS `ints`

</pre>

</div></div>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

Failed to analyze query: org.apache.spark.sql.AnalysisException: cannot resolve '`ints`' given input columns: [a, `ints`]; line 1 pos 7

'Project ['ints]

+- Generate explode(a#0.b), <span class="code-keyword">true</span>, <span class="code-keyword">false</span>, Some(a), [`ints`#8]

+- SubqueryAlias nestedarray

+- LocalRelation [a#0], [[[[1,2,3]]]]

</pre>

</div></div>

Description: <p>Here is a reproducer</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

&gt;&gt;&gt; from pyspark.streaming <span class="code-keyword">import</span> StreamingContext

&gt;&gt;&gt; from pyspark.streaming.util <span class="code-keyword">import</span> TransformFunction

&gt;&gt;&gt; ssc = StreamingContext(sc, 1)

&gt;&gt;&gt; func = TransformFunction(sc, lambda x: x, sc.serializer)

&gt;&gt;&gt; func.rdd\_wrapper(lambda x: x)

TransformFunction(&lt;function &lt;lambda&gt; at 0x106ac8b18&gt;)

&gt;&gt;&gt; bytes = bytearray(ssc.\_transformerSerializer.serializer.dumps((func.func, func.rdd\_wrap\_func, func.deserializers)))

&gt;&gt;&gt; func2 = ssc.\_transformerSerializer.loads(bytes)

&gt;&gt;&gt; print(func2.func.\_\_module\_\_)

None

&gt;&gt;&gt; print(func2.rdd\_wrap\_func.\_\_module\_\_)

None

&gt;&gt;&gt;

</pre>

</div></div>

Description: <p>Today, both the MemoryStore and DiskStore implement a common BlockStore API, but I feel that this API is inappropriate because it abstracts away important distinctions between the behavior of these two stores.</p>

<p>For instance, the disk store doesn't have a notion of storing deserialized objects, so it's confusing for it to expose object-based APIs like putIterator() and getValues() instead of only exposing binary APIs and pushing the responsibilities of serialization and deserialization to the client.</p>

<p>As part of a larger BlockManager interface cleanup, I'd like to remove the BlockStore API and refine the MemoryStore and DiskStore interfaces to reflect more narrow sets of responsibilities for those components.</p>

Description: <p>When a cached block is spilled to disk and read back in serialized form (i.e. as bytes), the current BlockManager implementation will attempt to re-insert the serialized block into the MemoryStore even if the block's storage level requests deserialized caching.</p>

<p>This behavior adds some complexity to the MemoryStore but I don't think it offers many performance benefits and I'd like to remove it in order to simplify a larger refactoring patch. Therefore, I propose to change the behavior such that disk store reads will only cache bytes in the memory store for blocks with serialized storage levels.</p>

<p>There are two places where we request serialized bytes from the BlockStore:</p>

<p>1. getLocalBytes(), which is only called when reading local copies of TorrentBroadcast pieces. Broadcast pieces are always cached using a serialized storage level, so this won't lead to a mismatch in serialization forms if spilled bytes read from disk are cached as bytes in the memory store.</p>

<p>2. the non-shuffle-block branch in getBlockData(), which is only called by the NettyBlockRpcServer when responding to requests to read remote blocks. Caching the serialized bytes in memory will only benefit us if those cached bytes are read before they're evicted and the likelihood of that happening seems low since the frequency of remote reads of non-broadcast cached blocks seems very low. Caching these bytes when they have a low probability of being read is bad if it risks the eviction of blocks which are cached in their expected serialized/deserialized forms, since those blocks seem more likely to be read in local computation.</p>

<p>Therefore, I think this is a safe change.</p>

Description:

Description: <p>Fixed the following flaky test:</p>

<p><a href="https://amplab.cs.berkeley.edu/jenkins/job/spark-master-test-sbt-hadoop-2.7/256/testReport/junit/org.apache.spark.streaming/MapWithStateSuite/\_It\_is\_not\_a\_test\_/" class="external-link" rel="nofollow">https://amplab.cs.berkeley.edu/jenkins/job/spark-master-test-sbt-hadoop-2.7/256/testReport/junit/org.apache.spark.streaming/MapWithStateSuite/\_It\_is\_not\_a\_test\_/</a></p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

sbt.ForkMain$ForkError: java.io.IOException: Failed to delete: /home/jenkins/workspace/spark-master-test-sbt-hadoop-2.7/streaming/checkpoint/spark-e97794a8-b940-4b21-8685-bf1221f9444d

at org.apache.spark.util.Utils$.deleteRecursively(Utils.scala:934)

at org.apache.spark.streaming.MapWithStateSuite$$anonfun$2.apply$mcV$sp(MapWithStateSuite.scala:47)

at org.apache.spark.streaming.MapWithStateSuite$$anonfun$2.apply(MapWithStateSuite.scala:45)

at org.apache.spark.streaming.MapWithStateSuite$$anonfun$2.apply(MapWithStateSuite.scala:45)

</pre>

</div></div>

Description: <p>This issue fixes the following potential bugs and Java coding style detected by Coverity and Checkstyle.</p>

<ul>

<li>Implement both null and type checking in equals functions.</li>

<li>Fix wrong type casting logic in SimpleJavaBean2.equals.</li>

<li>Add `implement Cloneable` to `UTF8String` and `SortedIterator`.</li>

<li>Remove dereferencing before null check in `AbstractBytesToBytesMapSuite`.</li>

<li>Fix coding style: Add '{}' to single `for` statement in mllib examples.</li>

<li>Remove unused imports in `ColumnarBatch` and `JavaKinesisStreamSuite.java`.</li>

<li>Remove unused fields in `ChunkFetchIntegrationSuite`.</li>

<li>Add `stop()` to prevent resource leak.</li>

</ul>

<p>Please note that the last two checkstyle errors exist on newly added commits after <a href="https://issues.apache.org/jira/browse/SPARK-13583" title="Remove unused imports and add checkstyle rule" class="issue-link" data-issue-key="SPARK-13583"><del>SPARK-13583</del></a>.</p>

Description: <p>Here is an example that Scala and Python generate different results</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

Scala:

scala&gt; <span class="code-keyword">var</span> i = 0

i: Int = 0

scala&gt; val rdd = sc.parallelize(1 to 10).map(\_ + i)

scala&gt; rdd.collect()

res0: Array[Int] = Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

scala&gt; i += 1

scala&gt; rdd.collect()

res2: Array[Int] = Array(2, 3, 4, 5, 6, 7, 8, 9, 10, 11)

Python:

&gt;&gt;&gt; i = 0

&gt;&gt;&gt; rdd = sc.parallelize(range(1, 10)).map(lambda x: x + i)

&gt;&gt;&gt; rdd.collect()

[1, 2, 3, 4, 5, 6, 7, 8, 9]

&gt;&gt;&gt; i += 1

&gt;&gt;&gt; rdd.collect()

[1, 2, 3, 4, 5, 6, 7, 8, 9]

</pre>

</div></div>

<p>The difference is Scala will capture all variables' values when running a job every time, but Python just captures variables' values once and always uses them for all jobs.</p>

Description: <p>UnsafeShuffleWriterSuite fails because of missing Snappy native library on arm64.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

Tests run: 19, Failures: 0, Errors: 2, Skipped: 0, Time elapsed: 6.437 sec &lt;&lt;&lt; FAILURE! - in org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite

mergeSpillsWithFileStreamAndSnappy(org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite) Time elapsed: 0.072 sec &lt;&lt;&lt; ERROR!

java.lang.reflect.InvocationTargetException

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithFileStreamAndSnappy(UnsafeShuffleWriterSuite.java:389)

Caused by: java.lang.IllegalArgumentException: org.xerial.snappy.SnappyError: [FAILED\_TO\_LOAD\_NATIVE\_LIBRARY] no <span class="code-keyword">native</span> library is found <span class="code-keyword">for</span> os.name=Linux and os.arch=aarch64

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithFileStreamAndSnappy(UnsafeShuffleWriterSuite.java:389)

Caused by: org.xerial.snappy.SnappyError: [FAILED\_TO\_LOAD\_NATIVE\_LIBRARY] no <span class="code-keyword">native</span> library is found <span class="code-keyword">for</span> os.name=Linux and os.arch=aarch64

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithFileStreamAndSnappy(UnsafeShuffleWriterSuite.java:389)

mergeSpillsWithTransferToAndSnappy(org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite) Time elapsed: 0.041 sec &lt;&lt;&lt; ERROR!

java.lang.reflect.InvocationTargetException

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithTransferToAndSnappy(UnsafeShuffleWriterSuite.java:384)

Caused by: java.lang.IllegalArgumentException: java.lang.NoClassDefFoundError: Could not initialize class org.xerial.snappy.Snappy

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithTransferToAndSnappy(UnsafeShuffleWriterSuite.java:384)

Caused by: java.lang.NoClassDefFoundError: Could not initialize class org.xerial.snappy.Snappy

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithTransferToAndSnappy(UnsafeShuffleWriterSuite.java:384)

Running org.apache.spark.JavaAPISuite

Tests run: 90, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 52.526 sec - in org.apache.spark.JavaAPISuite

Running org.apache.spark.unsafe.map.BytesToBytesMapOnHeapSuite

Tests run: 12, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 9.761 sec - in org.apache.spark.unsafe.map.BytesToBytesMapOnHeapSuite

Running org.apache.spark.unsafe.map.BytesToBytesMapOffHeapSuite

Tests run: 12, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 8.967 sec - in org.apache.spark.unsafe.map.BytesToBytesMapOffHeapSuite

Running org.apache.spark.api.java.OptionalSuite

Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec - in org.apache.spark.api.java.OptionalSuite

Results :

Tests in error:

UnsafeShuffleWriterSuite.mergeSpillsWithFileStreamAndSnappy:389-&gt;testMergingSpills:337 » InvocationTarget

UnsafeShuffleWriterSuite.mergeSpillsWithTransferToAndSnappy:384-&gt;testMergingSpills:337 » InvocationTarget

</pre>

</div></div>

Description: <p>When we add more DDL parsing logic in the future, SparkQl will become very big. To keep it smaller, we'll introduce helper "parser objects", e.g. one to parse alter table commands. However, these parser objects will need to access some helper methods that exist in CatalystQl. The proposal is to move those methods to an isolated ParserUtils object.</p>

Description: <p>When both spark.dynamicAllocation.enabled and spark.executor.instances are set, dynamic resource allocation is disabled (see <a href="https://issues.apache.org/jira/browse/SPARK-9092" title="Make --num-executors compatible with dynamic allocation" class="issue-link" data-issue-key="SPARK-9092"><del>SPARK-9092</del></a>). This is a reasonable default, but I think there should be a configuration property to override it because it isn't obvious to users that dynamic allocation and number of executors are mutually exclusive. We see users setting --num-executors because that looks like what they want: a way to get more executors.</p>

<p>I propose adding a new boolean property, spark.dynamicAllocation.overrideNumExecutors, that makes dynamic allocation the default when both are set and uses --num-executors as the minimum number of executors.</p>

Description: <p>Every time parallelize is called it creates temporary file for rdd in spark.local.dir/spark-uuid/pyspark-uuid/ directory. This directory deletes when context is closed, but for long running applications with permanently opened context this directory growth infinitely.</p>

Description: <p>`LinearRegressionWithSGD` and `StreamingLinearRegressionWithSGD` does not have `regParam` as their constructor arguments. They just depends on GradientDescent's default reqParam values. </p>

<p>To be consistent with other algorithms, we had better add them.</p>

Description:

Description: <p>We unsafely increment a volatile (bytesRead) in a call back, if two call backs are triggered we may under count bytesRead. This issue was found using coverity.</p>

Description: <p>We need to at least remove bucketing. I would also like to remove <tt>Job</tt> and the configuration stuff as well if possible.</p>

Description: <p>The current file format interface needs to be cleaned up before its acceptable for public consumption:</p>

<ul class="alternate" type="square">

<li>Have a version that takes Row and does a conversion, hide the internal API.</li>

<li>Remove bucketing</li>

<li>Remove RDD and the broadcastedConf</li>

<li>Remove SQLContext (maybe include SparkSession?)</li>

<li>Pass a better conf object</li>

</ul>

Description: <p>This test case got broken by <a href="https://github.com/apache/spark/pull/11509" class="external-link" rel="nofollow">#11509</a>. We should reimplement it as a format.</p>

Description: <p>I am trying to incorporate the Java UDAF from <a href="https://github.com/apache/spark/blob/master/sql/hive/src/test/java/org/apache/spark/sql/hive/aggregate/MyDoubleAvg.java" class="external-link" rel="nofollow">https://github.com/apache/spark/blob/master/sql/hive/src/test/java/org/apache/spark/sql/hive/aggregate/MyDoubleAvg.java</a> into an SQL query. <br/>

I registered the UDAF like this:<br/>

sqlContext.udf().register("myavg", new MyDoubleAvg());</p>

<p>My SQL query is:<br/>

SELECT AVG(seqi) AS `avg\_seqi`, AVG(seqd) AS `avg\_seqd`, AVG(ci) AS `avg\_ci`, AVG(cd) AS `avg\_cd`, AVG(stdevd) AS `avg\_stdevd`, AVG(stdevi) AS `avg\_stdevi`, MAX(seqi) AS `max\_seqi`, MAX(seqd) AS `max\_seqd`, MAX(ci) AS `max\_ci`, MAX(cd) AS `max\_cd`, MAX(stdevd) AS `max\_stdevd`, MAX(stdevi) AS `max\_stdevi`, MIN(seqi) AS `min\_seqi`, MIN(seqd) AS `min\_seqd`, MIN(ci) AS `min\_ci`, MIN(cd) AS `min\_cd`, MIN(stdevd) AS `min\_stdevd`, MIN(stdevi) AS `min\_stdevi`,SUM(seqi) AS `sum\_seqi`, SUM(seqd) AS `sum\_seqd`, SUM(ci) AS `sum\_ci`, SUM(cd) AS `sum\_cd`, SUM(stdevd) AS `sum\_stdevd`, SUM(stdevi) AS `sum\_stdevi`, myavg(seqd) as `myavg\_seqd`, AVG(zero) AS `avg\_zero`, AVG(nulli) AS `avg\_nulli`,AVG(nulld) AS `avg\_nulld`, SUM(zero) AS `sum\_zero`, SUM(nulli) AS `sum\_nulli`,SUM(nulld) AS `sum\_nulld`,MAX(zero) AS `max\_zero`, MAX(nulli) AS `max\_nulli`,MAX(nulld) AS `max\_nulld`,count( \* ) AS `count\_all`, count(nulli) AS `count\_nulli` FROM mytable</p>

<p>As soon as I add the UDAF myavg to the SQL, all the results become incorrect. When I remove the call to the UDAF, the results are correct.<br/>

I was able to go around the issue by modifying bufferSchema of the UDAF to use an array and the corresponding update and merge methods. </p>

Description: <p>Hello,</p>

<p>I'm trying to run pi.py example in a pyspark job with Oozie. Every try I made failed for the same reason: key not found: SPARK\_HOME. <br/>

Note: A scala job works well in the environment with Oozie. </p>

<p>The logs on the executors are:</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/mnt/hd4/hadoop/yarn/local/filecache/145/slf4j-log4j12-1.6.6.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/mnt/hd2/hadoop/yarn/local/filecache/155/spark-assembly-1.6.0-hadoop2.7.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/opt/application/Hadoop/hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.

SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

log4j:ERROR setFile(null,true) call failed.

java.io.FileNotFoundException: /mnt/hd7/hadoop/yarn/log/application\_1454673025841\_13136/container\_1454673025841\_13136\_01\_000001 (Is a directory)

at java.io.FileOutputStream.open(Native Method)

at java.io.FileOutputStream.&lt;init&gt;(FileOutputStream.java:221)

at java.io.FileOutputStream.&lt;init&gt;(FileOutputStream.java:142)

at org.apache.log4j.FileAppender.setFile(FileAppender.java:294)

at org.apache.log4j.FileAppender.activateOptions(FileAppender.java:165)

at org.apache.hadoop.yarn.ContainerLogAppender.activateOptions(ContainerLogAppender.java:55)

at org.apache.log4j.config.PropertySetter.activate(PropertySetter.java:307)

at org.apache.log4j.config.PropertySetter.setProperties(PropertySetter.java:172)

at org.apache.log4j.config.PropertySetter.setProperties(PropertySetter.java:104)

at org.apache.log4j.PropertyConfigurator.parseAppender(PropertyConfigurator.java:809)

at org.apache.log4j.PropertyConfigurator.parseCategory(PropertyConfigurator.java:735)

at org.apache.log4j.PropertyConfigurator.configureRootCategory(PropertyConfigurator.java:615)

at org.apache.log4j.PropertyConfigurator.doConfigure(PropertyConfigurator.java:502)

at org.apache.log4j.PropertyConfigurator.doConfigure(PropertyConfigurator.java:547)

at org.apache.log4j.helpers.OptionConverter.selectAndConfigure(OptionConverter.java:483)

at org.apache.log4j.LogManager.&lt;clinit&gt;(LogManager.java:127)

at org.slf4j.impl.Log4jLoggerFactory.getLogger(Log4jLoggerFactory.java:64)

at org.slf4j.LoggerFactory.getLogger(LoggerFactory.java:285)

at org.apache.commons.logging.impl.SLF4JLogFactory.getInstance(SLF4JLogFactory.java:155)

at org.apache.commons.logging.impl.SLF4JLogFactory.getInstance(SLF4JLogFactory.java:132)

at org.apache.commons.logging.LogFactory.getLog(LogFactory.java:275)

at org.apache.hadoop.service.AbstractService.&lt;clinit&gt;(AbstractService.java:43)

Using properties file: null

Parsed arguments:

master yarn-master

deployMode cluster

executorMemory null

executorCores null

totalExecutorCores null

propertiesFile null

driverMemory null

driverCores null

driverExtraClassPath null

driverExtraLibraryPath null

driverExtraJavaOptions null

supervise false

queue null

numExecutors null

files null

pyFiles null

archives null

mainClass null

primaryResource hdfs://hadoopsandbox/User/toto/WORK/Oozie/pyspark/lib/pi.py

name Pysparkpi example

childArgs [100]

jars null

packages null

packagesExclusions null

repositories null

verbose true

Spark properties used, including those specified through

--conf and those from the properties file null:

spark.executorEnv.SPARK\_HOME -&gt; /opt/application/Spark/current

spark.executorEnv.PYTHONPATH -&gt; /opt/application/Spark/current/python

spark.yarn.appMasterEnv.SPARK\_HOME -&gt; /opt/application/Spark/current

Main class:

org.apache.spark.deploy.yarn.Client

Arguments:

--name

Pysparkpi example

--primary-py-file

hdfs://hadoopsandbox/User/toto/WORK/Oozie/pyspark/lib/pi.py

--class

org.apache.spark.deploy.PythonRunner

--arg

100

System properties:

spark.executorEnv.SPARK\_HOME -&gt; /opt/application/Spark/current

spark.executorEnv.PYTHONPATH -&gt; /opt/application/Spark/current/python

SPARK\_SUBMIT -&gt; true

spark.app.name -&gt; Pysparkpi example

spark.submit.deployMode -&gt; cluster

spark.yarn.appMasterEnv.SPARK\_HOME -&gt; /opt/application/Spark/current

spark.yarn.isPython -&gt; true

spark.master -&gt; yarn-cluster

Classpath elements:

Failing Oozie Launcher, Main class [org.apache.oozie.action.hadoop.SparkMain], main() threw exception, key not found: SPARK\_HOME

java.util.NoSuchElementException: key not found: SPARK\_HOME

at scala.collection.MapLike$class.default(MapLike.scala:228)

at scala.collection.AbstractMap.default(Map.scala:58)

at scala.collection.MapLike$class.apply(MapLike.scala:141)

at scala.collection.AbstractMap.apply(Map.scala:58)

at org.apache.spark.deploy.yarn.Client$$anonfun$findPySparkArchives$2.apply(Client.scala:1045)

at org.apache.spark.deploy.yarn.Client$$anonfun$findPySparkArchives$2.apply(Client.scala:1044)

at scala.Option.getOrElse(Option.scala:120)

at org.apache.spark.deploy.yarn.Client.findPySparkArchives(Client.scala:1044)

at org.apache.spark.deploy.yarn.Client.createContainerLaunchContext(Client.scala:717)

at org.apache.spark.deploy.yarn.Client.submitApplication(Client.scala:142)

at org.apache.spark.deploy.yarn.Client.run(Client.scala:1016)

at org.apache.spark.deploy.yarn.Client$.main(Client.scala:1076)

at org.apache.spark.deploy.yarn.Client.main(Client.scala)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:731)

at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:181)

at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:206)

at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:121)

at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

at org.apache.oozie.action.hadoop.SparkMain.runSpark(SparkMain.java:104)

at org.apache.oozie.action.hadoop.SparkMain.run(SparkMain.java:95)

at org.apache.oozie.action.hadoop.LauncherMain.run(LauncherMain.java:47)

at org.apache.oozie.action.hadoop.SparkMain.main(SparkMain.java:38)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.apache.oozie.action.hadoop.LauncherMapper.map(LauncherMapper.java:236)

at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:54)

at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:453)

at org.apache.hadoop.mapred.MapTask.run(MapTask.java:343)

at org.apache.hadoop.mapred.LocalContainerLauncher$EventHandler.runSubtask(LocalContainerLauncher.java:380)

at org.apache.hadoop.mapred.LocalContainerLauncher$EventHandler.runTask(LocalContainerLauncher.java:301)

at org.apache.hadoop.mapred.LocalContainerLauncher$EventHandler.access$200(LocalContainerLauncher.java:187)

at org.apache.hadoop.mapred.LocalContainerLauncher$EventHandler$1.run(LocalContainerLauncher.java:230)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:471)

at java.util.concurrent.FutureTask.run(FutureTask.java:262)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)

at java.lang.Thread.run(Thread.java:745)

log4j:WARN No appenders could be found for logger (org.apache.hadoop.mapreduce.v2.app.MRAppMaster).

log4j:WARN Please initialize the log4j system properly.

log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

</pre>

</div></div>

<p>The workflow used for Oozie is the following:</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>&lt;workflow-app xmlns='uri:oozie:workflow:0.5' name='PysparkPi-test'&gt;

&lt;start to='spark-node' /&gt;

&lt;action name='spark-node'&gt;

&lt;spark xmlns="uri:oozie:spark-action:0.1"&gt;

&lt;job-tracker&gt;${jobTracker}&lt;/job-tracker&gt;

&lt;name-node&gt;${nameNode}&lt;/name-node&gt;

&lt;master&gt;${master}&lt;/master&gt;

&lt;mode&gt;${mode}&lt;/mode&gt;

&lt;name&gt;Pysparkpi example&lt;/name&gt;

&lt;class&gt;&lt;/class&gt;

&lt;jar&gt;${nameNode}/User/toto/WORK/Oozie/pyspark/lib/pi.py&lt;/jar&gt;

&lt;spark-opts&gt;--conf spark.yarn.appMasterEnv.SPARK\_HOME=/opt/application/Spark/current --conf spark.executorEnv.SPARK\_HOME=/opt/application/Spark/current --conf spark.executorEnv.PYTHONPATH=/opt/application/Spark/current/python&lt;/spark-opts&gt;

&lt;arg&gt;100&lt;/arg&gt;

&lt;/spark&gt;

&lt;ok to="end" /&gt;

&lt;error to="fail" /&gt;

&lt;/action&gt;

&lt;kill name="fail"&gt;

&lt;message&gt;Workflow failed, error message[${wf:errorMessage(wf:lastErrorNode())}]&lt;/message&gt;

&lt;/kill&gt;

&lt;end name='end' /&gt;

&lt;/workflow-app&gt;

</pre>

</div></div>

Description:

Description: <p>It would be nice to be able to use RF and GBT for feature transformation:<br/>

First fit an ensemble of trees (like RF, GBT or other TreeEnsambleModels) on the training set. Then each leaf of each tree in the ensemble is assigned a fixed arbitrary feature index in a new feature space. These leaf indices are then encoded in a one-hot fashion.</p>

<p>This method was first introduced by facebook(<a href="http://www.herbrich.me/papers/adclicksfacebook.pdf" class="external-link" rel="nofollow">http://www.herbrich.me/papers/adclicksfacebook.pdf</a>), and is implemented in two famous library:<br/>

sklearn (<a href="http://scikit-learn.org/stable/auto\_examples/ensemble/plot\_feature\_transformation.html#example-ensemble-plot-feature-transformation-py" class="external-link" rel="nofollow">http://scikit-learn.org/stable/auto\_examples/ensemble/plot\_feature\_transformation.html#example-ensemble-plot-feature-transformation-py</a>)<br/>

xgboost (<a href="https://github.com/dmlc/xgboost/blob/master/demo/guide-python/predict\_leaf\_indices.py" class="external-link" rel="nofollow">https://github.com/dmlc/xgboost/blob/master/demo/guide-python/predict\_leaf\_indices.py</a>)</p>

<p>I have implement it in mllib:</p>

<p>val features : RDD<span class="error">&#91;Vector&#93;</span> = ...<br/>

val model1 : RandomForestModel = ...<br/>

val transformed1 : RDD<span class="error">&#91;Vector&#93;</span> = model1.leaf(features)</p>

<p>val model2 : GradientBoostedTreesModel = ...<br/>

val transformed2 : RDD<span class="error">&#91;Vector&#93;</span> = model2.leaf(features)</p>

Description: <p>The default value of regularization parameter for `LogisticRegression` algorithm is different in Scala and Python. We should provide the same value.</p>

<div class="code panel" style="border-style: solid;border-width: 1px;"><div class="codeHeader panelHeader" style="border-bottom-width: 1px;border-bottom-style: solid;"><b>Scala</b></div><div class="codeContent panelContent">

<pre class="code-java">

scala&gt; <span class="code-keyword">new</span> org.apache.spark.ml.classification.LogisticRegression().getRegParam

res0: <span class="code-object">Double</span> = 0.0

</pre>

</div></div>

<div class="code panel" style="border-style: solid;border-width: 1px;"><div class="codeHeader panelHeader" style="border-bottom-width: 1px;border-bottom-style: solid;"><b>Python</b></div><div class="codeContent panelContent">

<pre class="code-java">

&gt;&gt;&gt; from pyspark.ml.classification <span class="code-keyword">import</span> LogisticRegression

&gt;&gt;&gt; LogisticRegression().getRegParam()

0.1

</pre>

</div></div>

Description: <p>Current URL for each application to access history UI is like: <br/>

<a href="http://localhost:18080/history/application\_1457058760338\_0016/1/jobs/" class="external-link" rel="nofollow">http://localhost:18080/history/application\_1457058760338\_0016/1/jobs/</a> or <a href="http://localhost:18080/history/application\_1457058760338\_0016/2/jobs/" class="external-link" rel="nofollow">http://localhost:18080/history/application\_1457058760338\_0016/2/jobs/</a></p>

<p>Here <b>1</b> or <b>2</b> represents the number of attempts in <tt>historypage.js</tt>, but it will parse to attempt id in <tt>HistoryServer</tt>, while the correct attempt id should be like "appattempt\_1457058760338\_0016\_000002", so it will fail to parse to a correct attempt id in <tt>HistoryServer</tt>.</p>

<p>This is OK in yarn client mode, since we don't need this attempt id to fetch out the app cache, but it is failed in yarn cluster mode, where attempt id "1" or "2" is actually wrong.</p>

<p>So here we should fix this url to parse the correct application id and attempt id.</p>

<p>This bug is newly introduced in <a href="https://issues.apache.org/jira/browse/SPARK-10873" title="Change history to use datatables to support sorting columns and searching" class="issue-link" data-issue-key="SPARK-10873"><del>SPARK-10873</del></a>, there's no issue in branch 1.6.</p>

<p>Here is the screenshot:</p>

<p><img src="https://issues.apache.org/jira/secure/attachment/12791437/Screen%20Shot%202016-02-29%20at%203.57.32%20PM.png" align="absmiddle" border="0" /></p>

Description: <p>Sample operator doesn't support wholestage codegen now. This issue is opened to add support for it.</p>

Description: <p><tt>bin\beeline.cmd</tt> pollutes environment variables in Windows.<br/>

The similar problem is reported and fixed in <a href="https://issues.apache.org/jira/browse/SPARK-3943" title="Some scripts bin\\*.cmd pollutes environment variables in Windows." class="issue-link" data-issue-key="SPARK-3943"><del>SPARK-3943</del></a>, but <tt>bin\beeline.cmd</tt> is added later.</p>

Description: <p>add the missing python examples of BisectingKMeans for ml and mllib</p>

Description: <p>Right now, we use PhysicalRDD for both existing RDD and data sources, they are becoming much different, we should use different physical plans for them.</p>

Description: <p>There's a particular snippet in spark-class <a href="https://github.com/apache/spark/blob/master/bin/spark-class#L86" class="external-link" rel="nofollow">here</a> that runs the spark-launcher code in a subshell.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

# The launcher library will print arguments separated by a NULL character, to allow arguments with

# characters that would be otherwise interpreted by the shell. Read that in a <span class="code-keyword">while</span> loop, populating

# an array that will be used to exec the <span class="code-keyword">final</span> command.

CMD=()

<span class="code-keyword">while</span> IFS= read -d '' -r ARG; <span class="code-keyword">do</span>

CMD+=(<span class="code-quote">"$ARG"</span>)

done &lt; &lt;(<span class="code-quote">"$RUNNER"</span> -cp <span class="code-quote">"$LAUNCH\_CLASSPATH"</span> org.apache.spark.launcher.Main <span class="code-quote">"$@"</span>)

</pre>

</div></div>

<p>The problem is that the if the launcher Main fails, this code still still returns success and continues, even though the top level script is marked <tt>set -e</tt>. This is because the launcher.Main is run within a subshell.</p>

Description: <p>Currently we are running into an issue with Yarn work preserving enabled + external shuffle service. </p>

<p>In the work preserving enabled scenario, the failure of NM will not lead to the exit of executors, so executors can still accept and run the tasks. The problem here is when NM is failed, external shuffle service is actually inaccessible, so reduce tasks will always complain about the “Fetch failure”, and the failure of reduce stage will make the parent stage (map stage) rerun. The tricky thing here is Spark scheduler is not aware of the unavailability of external shuffle service, and will reschedule the map tasks on the executor where NM is failed, and again reduce stage will be failed with “Fetch failure”, and after 4 retries, the job is failed.</p>

<p>So here the actual problem is Spark’s scheduler is not aware of the unavailability of external shuffle service, and will still assign the tasks on to that nodes. The fix is to avoid assigning tasks on to that nodes.</p>

<p>Currently in the Spark, one related configuration is “spark.scheduler.executorTaskBlacklistTime”, but I don’t think it will be worked in this scenario. This configuration is used to avoid same reattempt task to run on the same executor. Also ways like MapReduce’s blacklist mechanism may not handle this scenario, since all the reduce tasks will be failed, so counting the failure tasks will equally mark all the executors as “bad” one.</p>

Description: <p>If a filter predicate or a join condition consists of `IsNotNull` checks, we should reorder these checks such that these non-nullability checks are evaluated before the rest of the predicates.</p>

<p>For e.g., if a filter predicate is of the form `a &gt; 5 &amp;&amp; isNotNull(b)`, we should rewrite this as `isNotNull(b) &amp;&amp; a &gt; 5` during physical plan generation.</p>

<p>cc <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=nongli" class="user-hover" rel="nongli">Nong Li</a> <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=yhuai" class="user-hover" rel="yhuai">Yin Huai</a></p>

Description: <p>Currently, CSV data source does not support to parse date and timestamp types in custom format and infer the type of timestamp type in custom format.</p>

<p>It looks quite many of users want this feature. It would be great to set custom date format.</p>

<p>This was reported in spark-csv.<br/>

<a href="https://github.com/databricks/spark-csv/issues/279" class="external-link" rel="nofollow">https://github.com/databricks/spark-csv/issues/279</a><br/>

<a href="https://github.com/databricks/spark-csv/issues/262" class="external-link" rel="nofollow">https://github.com/databricks/spark-csv/issues/262</a><br/>

<a href="https://github.com/databricks/spark-csv/issues/266" class="external-link" rel="nofollow">https://github.com/databricks/spark-csv/issues/266</a></p>

<p>Currently I submitted a PR for this in spark-csv<br/>

<a href="https://github.com/databricks/spark-csv/pull/280" class="external-link" rel="nofollow">https://github.com/databricks/spark-csv/pull/280</a></p>

Description: <p>Whenever I run spark-shell I get a bunch of warnings about SQL configuration:</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>16/03/03 19:00:25 WARN hive.HiveSessionState$$anon$1: Attempt to set non-Spark SQL config in SQLConf: key = spark.yarn.driver.memoryOverhead, value = 26

16/03/03 19:00:25 WARN hive.HiveSessionState$$anon$1: Attempt to set non-Spark SQL config in SQLConf: key = spark.yarn.executor.memoryOverhead, value = 26

16/03/03 19:00:25 WARN hive.HiveSessionState$$anon$1: Attempt to set non-Spark SQL config in SQLConf: key = spark.executor.cores, value = 1

16/03/03 19:00:25 WARN hive.HiveSessionState$$anon$1: Attempt to set non-Spark SQL config in SQLConf: key = spark.executor.memory, value = 268435456

</pre>

</div></div>

<p>That should't happen, since I'm not setting those values explicitly. They're either set internally by Spark or come from spark-defaults.conf.</p>

Description: <p>The goal here is to break apart: File Management, code to deal with specific formats and query planning.</p>

Description: <p>A majority of Spark SQL queries likely run though <tt>HadoopFSRelation</tt>, however there are currently several complexity and performance problems with this code path:</p>

<ul class="alternate" type="square">

<li>The class mixes the concerns of file management, schema reconciliation, scan building, bucketing, partitioning, and writing data.</li>

<li>For very large tables, we are broadcasting the entire list of files to every executor. <a href="https://issues.apache.org/jira/browse/SPARK-11441" title="HadoopFsRelation is not scalable in number of files read/written" class="issue-link" data-issue-key="SPARK-11441">SPARK-11441</a></li>

<li>For partitioned tables, we always do an extra projection. This results not only in a copy, but undoes much of the performance gains that we are going to get from vectorized reads.</li>

</ul>

<p>This is an umbrella ticket to track a set of improvements to this codepath.</p>

Description: <p>The JVM memory leaky problem reported in <a href="https://github.com/xerial/snappy-java/issues/131" class="external-link" rel="nofollow">https://github.com/xerial/snappy-java/issues/131</a> has been resolved.</p>

<p>1.1.2.1 was released on Jan 22nd.</p>

<p>We should upgrade to this release.</p>

Description: <p>Currently, the SHOW TABLES command in Spark's Hive ThriftServer, or equivalently the HiveContext.tables method, returns a DataFrame with only two columns: the name of the table and whether it is temporary. It would be really nice to add support to return some extra information, such as:</p>

<ul class="alternate" type="square">

<li>Whether this table is Spark-only or a native Hive table</li>

<li>If spark-only, the name of the data source</li>

<li>potentially other properties</li>

</ul>

<p>The first two is really useful for BI environments connecting to multiple data sources and that work with both Hive and Spark.</p>

<p>Some thoughts:</p>

<ul class="alternate" type="square">

<li>The SQL/HiveContext Catalog API might need to be expanded to return something like a TableEntry, rather than just a tuple of (name, temporary).</li>

<li>I believe there is a Hive Catalog/client API to get information about each table. I suppose one concern would be the speed of using this API. Perhaps there are other APis that can get this info faster.</li>

</ul>

Description: <p>We usually build the HashedRelation on top of array of UnsafeRow, the copy could be avoided.</p>

<p>The caller of HashedRelation need to do the copy if it's needed.</p>

<p>Another approach could be making the copy() of UnsafeRow smart so that it know when should copy the bytes or not, this could be also useful for other components. </p>

Description: <p><a href="https://github.com/apache/spark/pull/11439" class="external-link" rel="nofollow">https://github.com/apache/spark/pull/11439</a> added a utility method "testQuietly". We can use it for CommitFailureTestRelationSuite.</p>

Description: <p>In preparation for larger refactorings, I think that we should remove the confusing returnValues() option from the BlockStore put() APIs: returning the value is only useful in one place (caching) and in other situations, such as block replication, it's simpler to put() and then get().</p>

Description: <p>When run TPCDS Q3 <span class="error">&#91;1&#93;</span> with lots predicates to filter out the partitions, the optimizer rule BooleanSimplification take about 2 seconds (it use lots of sematicsEqual, which require copy the whole tree).</p>

<p>It will great if we could speedup it.</p>

<p><span class="error">&#91;1&#93;</span> <a href="https://github.com/cloudera/impala-tpcds-kit/blob/master/queries/q3.sql" class="external-link" rel="nofollow">https://github.com/cloudera/impala-tpcds-kit/blob/master/queries/q3.sql</a></p>

<p>cc <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=marmbrus" class="user-hover" rel="marmbrus">Michael Armbrust</a></p>

Description: <p>CatalystQl can't parse an expression with hundreds of AND/OR <span class="error">&#91;1&#93;</span>, it will fail as StackOverflow. </p>

<p><span class="error">&#91;1&#93;</span> <a href="https://github.com/cloudera/impala-tpcds-kit/blob/master/queries/q3.sql" class="external-link" rel="nofollow">https://github.com/cloudera/impala-tpcds-kit/blob/master/queries/q3.sql</a></p>

Description: <p>Looks like spark.sql.parquet.cacheMetadata is not used anymore. Let's delete it to avoid any potential confusion.</p>

Description: <p>`WithAggregationKinesisBackedBlockRDDSuite` test consistently hangs and fails due to timeout. Currently, this is observed in <a href="https://github.com/apache/spark/pull/11481" class="external-link" rel="nofollow">PR-11481</a> and <a href="https://github.com/apache/spark/pull/11438" class="external-link" rel="nofollow">PR-11438</a>. Inevitably, those PR removes their changes on Kinesis module to skip Kinesis test.</p>

<div class="code panel" style="border-style: solid;border-width: 1px;"><div class="codeHeader panelHeader" style="border-bottom-width: 1px;border-bottom-style: solid;"><b>SparkPullRequestBuilder-52389</b></div><div class="codeContent panelContent">

<pre class="code-java">

[info] WithAggregationKinesisBackedBlockRDDSuite:

Using endpoint URL https:<span class="code-comment">//kinesis.us-west-2.amazonaws.com <span class="code-keyword">for</span> creating Kinesis streams <span class="code-keyword">for</span> tests.

</span>[2016-03-03 02:11:56.990918] [0x00007fabfb82f700] [info] [kinesis\_producer.cc:79] Created pipeline <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-7296082419112679238"</span>

[2016-03-03 02:11:56.991038] [0x00007fabfb82f700] [info] [shard\_map.cc:83] Updating shard map <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-7296082419112679238"</span>

[2016-03-03 02:11:57.230083] [0x00007fabfa42d700] [info] [shard\_map.cc:163] Successfully updated shard map <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-7296082419112679238"</span> found 2 shards

[info] - Basic reading from Kinesis (2 seconds, 116 milliseconds)

[info] - Read data available in both block manager and Kinesis (65 milliseconds)

Attempting to post to Github...

&gt; Post successful.

Build step 'Execute shell' marked build as failure

Archiving artifacts

Recording test results

Test FAILed.

Refer to <span class="code-keyword">this</span> link <span class="code-keyword">for</span> build results (access rights to CI server needed):

https:<span class="code-comment">//amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/52389/

</span>Test FAILed.

Finished: FAILURE

</pre>

</div></div>

<div class="code panel" style="border-style: solid;border-width: 1px;"><div class="codeHeader panelHeader" style="border-bottom-width: 1px;border-bottom-style: solid;"><b>NewSparkPullRequestBuilder-2608</b></div><div class="codeContent panelContent">

<pre class="code-java">

[info] WithAggregationKinesisBackedBlockRDDSuite:

[2016-03-03 10:28:05.512018] [0x00007f0fe5705700] [info] [kinesis\_producer.cc:79] Created pipeline <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-1569968002408612511"</span>

[2016-03-03 10:28:05.512313] [0x00007f0fe5705700] [info] [shard\_map.cc:83] Updating shard map <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-1569968002408612511"</span>

[2016-03-03 10:28:05.739267] [0x00007f0fe4303700] [info] [shard\_map.cc:163] Successfully updated shard map <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-1569968002408612511"</span> found 2 shards

[info] - Basic reading from Kinesis (1 second, 504 milliseconds)

[info] - Read data available in both block manager and Kinesis (37 milliseconds)

Build timed out (after 180 minutes). Marking the build as failed.

Build was aborted

Archiving artifacts

Recording test results

Finished: FAILURE

</pre>

</div></div>

Description: <p>I execute next query on my data:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

select count(distinct get\_json\_object(regexp\_extract(line, <span class="code-quote">"^\\p{ASCII}\*$"</span>, 0), '$.event')) from

(select line from logs.raw\_client\_log where year=2016 and month=2 and day&gt;28 and line rlike <span class="code-quote">"^\\p{ASCII}\*$"</span> and line is not <span class="code-keyword">null</span>) a

</pre>

</div></div>

<p>And it fails with </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

Error: org.apache.spark.SparkException: Job aborted due to stage failure: Task 420 in stage 168.0 failed 4 times, most recent failure: Lost task 420.3 in stage 168.0 (TID 13064, nod5-2-hadoop.anchorfree.net): java.io.CharConversionException: Invalid UTF-32 character 0x6576656e(above 10ffff) at <span class="code-object">char</span> #47, <span class="code-object">byte</span> #191)

at com.fasterxml.jackson.core.io.UTF32Reader.reportInvalid(UTF32Reader.java:189)

at com.fasterxml.jackson.core.io.UTF32Reader.read(UTF32Reader.java:150)

at com.fasterxml.jackson.core.json.ReaderBasedJsonParser.loadMore(ReaderBasedJsonParser.java:153)

at com.fasterxml.jackson.core.json.ReaderBasedJsonParser.\_skipWSOrEnd(ReaderBasedJsonParser.java:1855)

at com.fasterxml.jackson.core.json.ReaderBasedJsonParser.nextToken(ReaderBasedJsonParser.java:571)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject$$anonfun$eval$2$$anonfun$4.apply(jsonExpressions.scala:142)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject$$anonfun$eval$2$$anonfun$4.apply(jsonExpressions.scala:141)

at org.apache.spark.util.Utils$.tryWithResource(Utils.scala:2202)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject$$anonfun$eval$2.apply(jsonExpressions.scala:141)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject$$anonfun$eval$2.apply(jsonExpressions.scala:138)

at org.apache.spark.util.Utils$.tryWithResource(Utils.scala:2202)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject.eval(jsonExpressions.scala:138)

at org.apache.spark.sql.catalyst.expressions.GeneratedClass$SpecificMutableProjection.apply(Unknown Source)

at org.apache.spark.sql.execution.Expand$$anonfun$doExecute$1$$anonfun$3$$anon$1.next(Expand.scala:76)

at org.apache.spark.sql.execution.Expand$$anonfun$doExecute$1$$anonfun$3$$anon$1.next(Expand.scala:62)

at org.apache.spark.sql.execution.aggregate.TungstenAggregationIterator.processInputs(TungstenAggregationIterator.scala:512)

at org.apache.spark.sql.execution.aggregate.TungstenAggregationIterator.&lt;init&gt;(TungstenAggregationIterator.scala:686)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1$$anonfun$2.apply(TungstenAggregate.scala:95)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1$$anonfun$2.apply(TungstenAggregate.scala:86)

at org.apache.spark.rdd.RDD$$anonfun$mapPartitions$1$$anonfun$apply$20.apply(RDD.scala:710)

at org.apache.spark.rdd.RDD$$anonfun$mapPartitions$1$$anonfun$apply$20.apply(RDD.scala:710)

at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:38)

at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:306)

at org.apache.spark.rdd.RDD.iterator(RDD.scala:270)

at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:38)

at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:306)

at org.apache.spark.rdd.RDD.iterator(RDD.scala:270)

at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:73)

at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:41)

at org.apache.spark.scheduler.Task.run(Task.scala:89)

at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:214)

</pre>

</div></div>

<p>Basically Spark sells me the idea, that I have character 敮 in my data. But query </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

select line from logs.raw\_client\_log where year=2016 and month=2 and day&gt;27 and line rlike <span class="code-quote">"敮"</span>

</pre>

</div></div>

<p>returns nothing.</p>

Description: <p>DiskBlockObjectWriter exposes methods for writing both byte streams and objects. Mixing both of these interfaces into a single class is somewhat messy and requires hacks in certain code to support it (see DummySerializerInstance). I think that we should cleanly separate these interfaces into separate classes.</p>

Description: <p>TransportClient is not thread safe and if it is called from multiple threads, the messages can't be encoded and decoded correctly. Below is my code,and it will print wrong message.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-keyword">public</span> <span class="code-keyword">static</span> void main(<span class="code-object">String</span>[] args) <span class="code-keyword">throws</span> IOException, InterruptedException {

TransportServer server = <span class="code-keyword">new</span> TransportContext(<span class="code-keyword">new</span> TransportConf(<span class="code-quote">"test"</span>,

<span class="code-keyword">new</span> MapConfigProvider(<span class="code-keyword">new</span> HashMap&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;())), <span class="code-keyword">new</span> RankHandler()).

createServer(8081, <span class="code-keyword">new</span> LinkedList&lt;TransportServerBootstrap&gt;());

TransportContext context = <span class="code-keyword">new</span> TransportContext(<span class="code-keyword">new</span> TransportConf(<span class="code-quote">"test"</span>,

<span class="code-keyword">new</span> MapConfigProvider(<span class="code-keyword">new</span> HashMap&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;())), <span class="code-keyword">new</span> NoOpRpcHandler(), <span class="code-keyword">true</span>);

<span class="code-keyword">final</span> TransportClientFactory clientFactory = context.createClientFactory();

List&lt;<span class="code-object">Thread</span>&gt; ts = <span class="code-keyword">new</span> ArrayList&lt;&gt;();

<span class="code-keyword">for</span> (<span class="code-object">int</span> i = 0; i &lt; 10; i++) {

ts.add(<span class="code-keyword">new</span> <span class="code-object">Thread</span>(<span class="code-keyword">new</span> <span class="code-object">Runnable</span>() {

@Override

<span class="code-keyword">public</span> void run() {

<span class="code-keyword">for</span> (<span class="code-object">int</span> j = 0; j &lt; 1000; j++) {

<span class="code-keyword">try</span> {

ByteBuf buf = Unpooled.buffer(8);

buf.writeLong((<span class="code-object">long</span>) j);

ByteBuffer byteBuffer = clientFactory.createClient(<span class="code-quote">"localhost"</span>, 8081).

sendRpcSync(buf.nioBuffer(), <span class="code-object">Long</span>.MAX\_VALUE);

<span class="code-object">long</span> response = byteBuffer.getLong();

<span class="code-keyword">if</span> (response != j) {

<span class="code-object">System</span>.err.println(<span class="code-quote">"send:"</span> + j + <span class="code-quote">",response:"</span> + response);

}

} <span class="code-keyword">catch</span> (IOException e) {

e.printStackTrace();

}

}

}

}));

ts.get(i).start();

}

<span class="code-keyword">for</span> (<span class="code-object">Thread</span> t : ts) {

t.join();

}

server.close();

}

<span class="code-keyword">public</span> class RankHandler <span class="code-keyword">extends</span> RpcHandler {

<span class="code-keyword">private</span> <span class="code-keyword">final</span> Logger logger = LoggerFactory.getLogger(RankHandler.class);

<span class="code-keyword">private</span> <span class="code-keyword">final</span> StreamManager streamManager;

<span class="code-keyword">public</span> RankHandler() {

<span class="code-keyword">this</span>.streamManager = <span class="code-keyword">new</span> OneForOneStreamManager();

}

@Override

<span class="code-keyword">public</span> void receive(TransportClient client, ByteBuffer msg, RpcResponseCallback callback) {

callback.onSuccess(msg);

}

@Override

<span class="code-keyword">public</span> StreamManager getStreamManager() {

<span class="code-keyword">return</span> streamManager;

}

}

</pre>

</div></div>

<p>it will print as below<br/>

send:221,response:222<br/>

send:233,response:234<br/>

send:312,response:313<br/>

send:358,response:359<br/>

...</p>

Description: <p>Seq(("id1", "value1")).toDF("key", "value").registerTempTable("src")<br/>

sqlContext.sql("SELECT t1.\* FROM src LATERAL VIEW explode(map('key1', 100, 'key2', 200)) t1 AS key, value")</p>

<p>Running above repro results in :</p>

<p>java.lang.ClassCastException: java.lang.Integer cannot be cast to org.apache.spark.unsafe.types.UTF8String<br/>

at org.apache.spark.sql.catalyst.expressions.BaseGenericInternalRow$class.getUTF8String(rows.scala:46)<br/>

at org.apache.spark.sql.catalyst.expressions.GenericInternalRow.getUTF8String(rows.scala:221)<br/>

at org.apache.spark.sql.catalyst.expressions.GeneratedClass$SpecificUnsafeProjection.apply(generated.java:42)<br/>

at org.apache.spark.sql.execution.Generate$$anonfun$doExecute$1$$anonfun$apply$9.apply(Generate.scala:98)<br/>

at org.apache.spark.sql.execution.Generate$$anonfun$doExecute$1$$anonfun$apply$9.apply(Generate.scala:96)<br/>

at scala.collection.Iterator$$anon$11.next(Iterator.scala:370)<br/>

at scala.collection.Iterator$$anon$11.next(Iterator.scala:370)<br/>

at scala.collection.Iterator$class.foreach(Iterator.scala:742)<br/>

at scala.collection.AbstractIterator.foreach(Iterator.scala:1194)<br/>

at scala.collection.generic.Growable$class.$plus$plus$eq(Growable.scala:59)<br/>

at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:104)<br/>

at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:48)<br/>

at scala.collection.TraversableOnce$class.to(TraversableOnce.scala:308)<br/>

at scala.collection.AbstractIterator.to(Iterator.scala:1194)<br/>

at scala.collection.TraversableOnce$class.toBuffer(TraversableOnce.scala:300)<br/>

at scala.collection.AbstractIterator.toBuffer(Iterator.scala:1194)<br/>

at scala.collection.TraversableOnce$class.toArray(TraversableOnce.scala:287)<br/>

at scala.collection.AbstractIterator.toArray(Iterator.scala:1194)<br/>

at org.apache.spark.rdd.RDD$$anonfun$collect$1$$anonfun$13.apply(RDD.scala:876)<br/>

at org.apache.spark.rdd.RDD$$anonfun$collect$1$$anonfun$13.apply(RDD.scala:876)<br/>

at org.apache.spark.SparkContext$$anonfun$runJob$5.apply(SparkContext.scala:1794)<br/>

at org.apache.spark.SparkContext$$anonfun$runJob$5.apply(SparkContext.scala:1794)<br/>

at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:69)<br/>

at org.apache.spark.scheduler.Task.run(Task.scala:82)<br/>

at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:227)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)<br/>

at java.lang.Thread.run(Thread.java:745)</p>

Description: <p>Is there some guidance of the usage of the Window() function on DStream. Here is my academic use-case for which it fails.</p>

<p>Standard word count</p>

<p> val ssc = new StreamingContext(sparkConf, Seconds(6))<br/>

val messages = KafkaUtils.createDirectStream(...)<br/>

val words = messages.map(\_.<em>2).flatMap(</em>.split(" "))<br/>

val window = words.window(Seconds(12), Seconds(6)) <br/>

window.count().print()</p>

<p>For the first batch interval it gives the count and then it hangs (inside the unionRDD)</p>

<p>I say the above use-case is academic since one can achieve similar fuctionality by using instead the more compact API<br/>

words.countByWindow(Seconds(12), Seconds(6))<br/>

which works fine. </p>

<p>Is the first approach above not the intended way of using the .window() API</p>

Description: <p><tt>CalendarInterval</tt> is part of user facing interface as it can be returned DataFrame contents, but historically it was put in <tt>unsafe.types</tt> package for no good reason. We should probably move it out.</p>

Description: <p>When running the standard Spark unit tests on the IBM Java SDK the hive VersionsSuite fail with the following error.</p>

<p>java.lang.NoClassDefFoundError: org.apache.hadoop.hive.cli.CliSessionState when creating Hive client using classpath: ..............</p>

Description:

Description: <p>getSampledInput counts the dataset twice as you see here : <a href="https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/ml/feature/QuantileDiscretizer.scala#L116" class="external-link" rel="nofollow">https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/ml/feature/QuantileDiscretizer.scala#L116</a></p>

Description: <p>In my Chrome 49, the execution DAG diagram can't be shown properly. Only a few grey dots lays there. Thought this is what I'm supposed to see at first. </p>

<p>It works fine in Firefox, though.</p>

<p>See the attachment below.</p>

Description: <p>This is to show a message points out the origin of a generated method when an exception occurs in the generated method at runtime.</p>

<p>An example of a message (the first line is newly added)</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

07:49:29.525 ERROR org.apache.spark.sql.catalyst.expressions.GeneratedClass$GeneratedIterator: The method GeneratedIterator.processNext() is generated <span class="code-keyword">for</span> filter at Test.scala:23

07:49:29.526 ERROR org.apache.spark.executor.Executor: Exception in task 1.0 in stage 2.0 (TID 4)

java.lang.NullPointerException:

at ...

</pre>

</div></div>

Description:

Description: <p>Currently when running Spark on Yarn with yarn cluster mode, the default application final state is "SUCCEED", if any exception is occurred, this final state will be changed to "FAILED" and trigger the reattempt if possible. </p>

<p>This is OK in normal case, but if there's a race condition when AM received a signal (SIGTERM) and no any exception is occurred. In this situation, shutdown hook will be invoked and marked this application as finished with success, and there's no another attempt.</p>

<p>In such situation, actually from Spark's aspect this application is failed and need another attempt, but from Yarn's aspect the application is finished with success.</p>

<p>This could happened in NM failure situation, the failure of NM will send SIGTERM to AM, AM should mark this attempt as failure and rerun again, not invoke unregister.</p>

<p>So to increase the chance of this race condition, here is the reproduced code:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

val sc = ...

<span class="code-object">Thread</span>.sleep(30000L)

sc.parallelize(1 to 100).collect()

</pre>

</div></div>

<p>If the AM is failed in sleeping, there's no exception been thrown, so from Yarn's point this application is finished successfully, but from Spark's point, this application should be reattempted.</p>

<p>The log normally like this:</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>16/03/03 12:44:19 INFO ContainerManagementProtocolProxy: Opening proxy : 192.168.0.105:45454

16/03/03 12:44:21 INFO YarnClusterSchedulerBackend: Registered executor NettyRpcEndpointRef(null) (192.168.0.105:57461) with ID 2

16/03/03 12:44:21 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.0.105:57462 with 511.1 MB RAM, BlockManagerId(2, 192.168.0.105, 57462)

16/03/03 12:44:23 INFO YarnClusterSchedulerBackend: Registered executor NettyRpcEndpointRef(null) (192.168.0.105:57467) with ID 1

16/03/03 12:44:23 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.0.105:57468 with 511.1 MB RAM, BlockManagerId(1, 192.168.0.105, 57468)

16/03/03 12:44:23 INFO YarnClusterSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.8

16/03/03 12:44:23 INFO YarnClusterScheduler: YarnClusterScheduler.postStartHook done

16/03/03 12:44:39 ERROR ApplicationMaster: RECEIVED SIGNAL 15: SIGTERM

16/03/03 12:44:39 INFO SparkContext: Invoking stop() from shutdown hook

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/metrics/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/stage/kill,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/api,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/static,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/executors/threadDump/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/executors/threadDump,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/executors/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/executors,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/environment/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/environment,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/storage/rdd/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/storage/rdd,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/storage/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/storage,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/pool/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/pool,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/stage/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/stage,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/jobs/job/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/jobs/job,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/jobs/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/jobs,null}

16/03/03 12:44:39 INFO SparkUI: Stopped Spark web UI at http://192.168.0.105:57452

16/03/03 12:44:39 INFO YarnAllocator: Driver requested a total number of 0 executor(s).

16/03/03 12:44:39 INFO YarnAllocator: Canceling requests for 0 executor containers

16/03/03 12:44:39 INFO YarnClusterSchedulerBackend: Shutting down all executors

16/03/03 12:44:39 WARN YarnAllocator: Expected to find pending requests, but found none.

16/03/03 12:44:39 INFO YarnClusterSchedulerBackend: Asking each executor to shut down

16/03/03 12:44:39 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices

(serviceOption=None,

services=List(),

started=false)

16/03/03 12:44:39 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!

16/03/03 12:44:39 INFO MemoryStore: MemoryStore cleared

16/03/03 12:44:39 INFO BlockManager: BlockManager stopped

16/03/03 12:44:39 INFO BlockManagerMaster: BlockManagerMaster stopped

16/03/03 12:44:39 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!

16/03/03 12:44:39 INFO SparkContext: Successfully stopped SparkContext

16/03/03 12:44:39 INFO ApplicationMaster: Final app status: SUCCEEDED, exitCode: 0, (reason: Shutdown hook called before final status was reported.)

16/03/03 12:44:39 INFO ApplicationMaster: Unregistering ApplicationMaster with SUCCEEDED (diag message: Shutdown hook called before final status was reported.)

16/03/03 12:44:39 INFO AMRMClientImpl: Waiting for application to be successfully unregistered.

16/03/03 12:44:39 INFO ApplicationMaster: Deleting staging directory .sparkStaging/application\_1456975940304\_0004

16/03/03 12:44:40 INFO ShutdownHookManager: Shutdown hook called

</pre>

</div></div>

<p>So basically, I think only after the finish of user class, we could mark this application as "SUCCESS", otherwise, especially in the signal stopped scenario, it would be better to mark as failed and try again (except explicitly KILL command by yarn).</p>

Description: <p>getModelFeatures of ml.api.r.SparkRWrapper cannot (always) reveal the original column names. Let's take the HouseVotes84 data set as an example:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-keyword">case</span> m: XXXModel =&gt;

val attrs = AttributeGroup.fromStructField(

m.summary.predictions.schema(m.summary.featuresCol))

attrs.attributes.get.map(\_.name.get)

</pre>

</div></div>

<p>The code above gets features' names from the features column. Usually, the features column is generated by RFormula. The latter has a VectorAssembler in it, which leads the output attributes not equal with the original ones.</p>

<p>E.g., we want to learn the HouseVotes84's features' name "V1, V2, ..., V16". But with RFormula, we can only get "V1\_n, V2\_y, ..., V16\_y" because <a href="https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/ml/feature/VectorAssembler.scala#L75" class="external-link" rel="nofollow">the transform function of VectorAssembler</a> adds salts of the column names.</p>

## Title and Description

Title: [SPARK-13740] add null check for \_verify\_type in types.py

Description:

Title: [SPARK-13739] Predicate Push Down Through Window Operator

Description: <p>Push down the predicate through the Window operator.</p>

Title: [SPARK-13738] Clean up ResolveDataSource

Description:

Title: [SPARK-13737] Add getOrCreate method for HiveContext

Description: <p>There is a "getOrCreate" method in SQLContext, which is useful to recoverable streaming application with SQL operation. <br/>

<a href="https://spark.apache.org/docs/latest/streaming-programming-guide.html#dataframe-and-sql-operations" class="external-link" rel="nofollow">https://spark.apache.org/docs/latest/streaming-programming-guide.html#dataframe-and-sql-operations</a></p>

<p>But the corresponding method is missing in HiveContext. </p>

Title: [SPARK-13736] Big-Endian plataform issues

Description: <p>We are starting to see few issues when building/testing on Big-Endian platform. This serves as an umbrella jira to group all platform specific issues.</p>

Title: [SPARK-13735] Log for parquet relation reading files is too verbose

Description: <p>The INFO level logging contains all files read by Parquet Relation, which is way too verbose if the input contains lots of files</p>

Title: [SPARK-13734] SparkR histogram

Description:

Title: [SPARK-13733] Support initial weight distribution in personalized PageRank

Description: <p>It would be nice to support personalized PageRank with an initial weight distribution besides a single vertex. It should be easy to modify the current implementation to add this support.</p>

Title: [SPARK-13732] Remove projectList from Windows

Description: <p>projectList is useless. Remove it from the class Window. It simplifies the codes in Analyzer and Optimizer. </p>

Title: [SPARK-13731] expression evaluation for NaN in select statement

Description: <p>We are expecting that arithmetic expression a/b should be:<br/>

1. returning NaN if a=0 and b=0<br/>

2. returning Infinity if a=1 and b=0</p>

<p>Is the expectation reasonable? <br/>

The following is a simple test case snippet that reads from storage and evaluates arithmetic expressions in select.<br/>

It is assuming org.apache.spark.sql.hive.execution.SQLQuerySuite: </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

test(<span class="code-quote">"Expression should be evaluated to Nan/Infinity in Select"</span>) {

withTable(<span class="code-quote">"testNan"</span>) {

withTempTable(<span class="code-quote">"src"</span>) {

Seq((1d, 0d), (0d, 0d)).toDF().registerTempTable(<span class="code-quote">"src"</span>)

sql(<span class="code-quote">"CREATE TABLE testNan(a <span class="code-object">double</span>, b <span class="code-object">double</span>) STORED AS PARQUET AS SELECT \* FROM src"</span>)

}

checkAnswer(sql(

"""

|SELECT a/b FROM testNan

""".stripMargin),

Seq(

Row(<span class="code-object">Double</span>.PositiveInfinity),

Row(<span class="code-object">Double</span>.NaN)

)

)

}

}

== Physical Plan ==

Project [(a#28 / b#29) AS \_c0#30]

+- Scan ParquetRelation: <span class="code-keyword">default</span>.testnan[a#28,b#29] InputPaths: file:/<span class="code-keyword">private</span>/<span class="code-keyword">var</span>/folders/dy/19y6pfm92pj9s40mbs8xd9hm0000gp/T/warehouse--5b617080-e909-4812-90e8-63d2dd0aef5a/testnan

== Results ==

!== Correct Answer - 2 == == Spark Answer - 2 ==

![Infinity] [<span class="code-keyword">null</span>]

![NaN] [<span class="code-keyword">null</span>]

</pre>

</div></div>

Title: [SPARK-13730] Nulls in dataframes getting converted to 0 with spark 2.0 SNAPSHOT

Description: <p>Basically I'm putting nulls into a non-nullable LongType column and doing a transformation operation on that column, the result is a column with nulls converted to 0. </p>

<p>I haven't tested this on 1.6.1 or in Scala.</p>

<p>Heres an example </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

from pyspark.sql <span class="code-keyword">import</span> types

from pyspark.sql <span class="code-keyword">import</span> DataFrame, types, functions as F

sql\_schema = types.StructType([

types.StructField(<span class="code-quote">"a"</span>, types.LongType(), True),

types.StructField(<span class="code-quote">"b"</span>, types.StringType(), True),

])

df = sqlCtx.createDataFrame([

(1, <span class="code-quote">"one"</span>),

(None, <span class="code-quote">"two"</span>),

], sql\_schema)

# Everything is fine here

df.collect() # [Row(a=1, b=u'one'), Row(a=None, b=u'two')]

def assert\_not\_null(val):

<span class="code-keyword">return</span> val

udf = F.udf(assert\_not\_null, types.LongType())

df = df.withColumnRenamed('a', <span class="code-quote">"\_tmp\_col"</span>)

df = df.withColumn('a', udf(df.\_tmp\_col))

df = df.drop(<span class="code-quote">"\_tmp\_col"</span>)

# None gets converted to 0

df.collect() # [Row(b=u'one', a=1), Row(b=u'two', a=0)]

</pre>

</div></div>

Title: [SPARK-13729] Reimplement the planning tests on SimpleTextRelation

Description:

Title: [SPARK-13728] Fix ORC PPD

Description: <p>Fix the ignored test "Enable ORC PPD" in OrcQuerySuite.</p>

Title: [SPARK-13727] SparkConf.contains does not consider deprecated keys

Description: <p>This makes it kinda inconsistent with other SparkConf APIs. For example:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

scala&gt; <span class="code-keyword">import</span> org.apache.spark.SparkConf

<span class="code-keyword">import</span> org.apache.spark.SparkConf

scala&gt; val conf = <span class="code-keyword">new</span> SparkConf().set(<span class="code-quote">"spark.io.compression.lz4.block.size"</span>, <span class="code-quote">"12345"</span>)

16/03/07 10:55:17 WARN spark.SparkConf: The configuration key 'spark.io.compression.lz4.block.size' has been deprecated as of Spark 1.4 and and may be removed in the <span class="code-keyword">future</span>. Please use the <span class="code-keyword">new</span> key 'spark.io.compression.lz4.blockSize' instead.

conf: org.apache.spark.SparkConf = org.apache.spark.SparkConf@221e8982

scala&gt; conf.get(<span class="code-quote">"spark.io.compression.lz4.blockSize"</span>)

res0: <span class="code-object">String</span> = 12345

scala&gt; conf.contains(<span class="code-quote">"spark.io.compression.lz4.blockSize"</span>)

res1: <span class="code-object">Boolean</span> = <span class="code-keyword">false</span>

</pre>

</div></div>

Title: [SPARK-13726] Spark 1.6.0 stopping working for HiveThriftServer2 and registerTempTable

Description: <p>In Spark 1.5.2, DataFrame.registerTempTable works and hiveContext.table(registerTableName) and HiveThriftServer2 see those tables.</p>

<p>In Spark 1.6.0, hiveContext.table(registerTableName) and HiveThriftServer2 do not see those tables, even though DataFrame.registerTempTable does not return an error.</p>

<p>Since this feature used to work in Spark 1.5.2, there is existing code that breaks after upgrading to Spark 1.6.0. so this issue is a blocker and urgent. Therefore, please have it fixed asap.</p>

Title: [SPARK-13725] Spark 1.6.0 stopping working for HiveThriftServer2 and registerTempTable

Description: <p>In Spark 1.5.2, DataFrame.registerTempTable API works correctly and HiveThriftServer2 sees and returns temp tables that are registered via that API.</p>

<p>In Spark 1.6.0, that stopped working. registerTempTable API does not return an error so it is a false positive, and HiveThriftServer2 does not see such tables. And hiveContext.table(registerTableName) indicates it does not see those tables either.</p>

<p>Is there a temporary work-around solution in Spark 1.6.0 ? When would it be fixed ?</p>

<p>Thanks.</p>

Title: [SPARK-13724] Parameter maxMemoryInMB has gone missing in MlLib 1.6.0 DecisionTree.trainClassifier()

Description: <p>DecisionTree.trainClassifier() reports that maxMemoryInMB is too small during training and stops. But when I try to set it, I found that in MLlib of spark 1.6.0 pyspark.mllib.tree.DecisionTree doesn't have this parameter in the named parameter list anymore. </p>

<p>(Also not sure if this is the place for this issue, kindly educate!) </p>

Title: [SPARK-13723] YARN - Change behavior of --num-executors when spark.dynamicAllocation.enabled true

Description: <p>I think we should change the behavior when --num-executors is specified when dynamic allocation is enabled. Currently if --num-executors is specified dynamic allocation is disabled and it just uses a static number of executors.</p>

<p>I would rather see the default behavior changed in the 2.x line. If dynamic allocation config is on then num-executors goes to max and initial # of executors. I think this would allow users to easily cap their usage and would still allow it to free up executors. It would also allow users doing ML start out with a # of executors and if they are actually caching the data the executors wouldn't be freed up. So you would get very similar behavior to if dynamic allocation was off.</p>

<p>Part of the reason for this is when using a static number if generally wastes resources, especially with people doing adhoc things with spark-shell. It also has a big affect when people are doing MapReduce/ETL type work loads. The problem is that people are used to specifying num-executors so if we turn it on by default in a cluster config its just overridden.</p>

<p>We should also update the spark-submit --help description for --num-executors</p>

Title: [SPARK-13722] No Push Down for Non-deterministic Predicates through Generate

Description: <p>Non-deterministic predicates should not be pushed down through Generate.</p>

Title: [SPARK-13721] Add support for LATERAL VIEW OUTER explode()

Description: <p>Hive supports the <a href="https://cwiki.apache.org/confluence/display/Hive/LanguageManual+LateralView#LanguageManualLateralView-OuterLateralViews" class="external-link" rel="nofollow">LATERAL VIEW OUTER</a> syntax to make sure that when an array is empty, the content from the outer table is still returned. </p>

<p>Within Spark, this is currently only possible within the HiveContext and executing HiveQL statements. It would be nice if the standard explode() DataFrame method allows the same. A possible signature would be: </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<div class="error"><span class="error">Unable to find source-code formatter for language: scala.</span> Available languages are: actionscript, html, java, javascript, none, sql, xhtml, xml</div><pre>

explode[A, B](inputColumn: <span class="code-object">String</span>, outputColumn: <span class="code-object">String</span>, <span class="code-keyword">outer</span>: <span class="code-object">Boolean</span> = <span class="code-keyword">false</span>)

</pre>

</div></div>

Title: [SPARK-13720] SQLBuilder.toSQL is broken when there is a sort by after having

Description: <p>parse this SQL "SELECT COUNT(value) FROM parquet\_t1 GROUP BY key HAVING MAX(key) &gt; 0 SORT BY key" to logical plan and generate SQL string again, the generated SQL string can't be parsed.</p>

<p>This is because, we will add extra `SubqueryAlias` while converting logical plan to SQL string(the `RecoverScopingInfo` rule in SQLBuilder). However, we only update qualifiers in the `Project` operator that just above the added `SubqueryAlias`, logically we also need to update qualifiers in all ancestor plans. For example, if there is a Project under a Sort, if we add `SubqueryAlias` under Project, we need update qualifiers for Sort too.</p>

Title: [SPARK-13719] Bad JSON record raises ﻿java.lang.ClassCastException

Description: <p>I have defined a JSON schema, using org.apache.spark.sql.types.StructType, that expects this kind of record :</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>{

"request": {

"user": {

"id": 123

}

}

}

</pre>

</div></div>

<p>There's a bad record in my dataset, that defines field "user" as an array, instead of a JSON object :</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>{

"request": {

"user": []

}

}

</pre>

</div></div>

<p>The following exception is raised because of that bad record :</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>Exception in thread "main" org.apache.spark.SparkException: Job aborted due to stage failure: Task 7 in stage 0.0 failed 4 times, most recent failure: Lost task 7.3 in stage 0.0 (TID 10, 192.168.1.170): java.lang.ClassCastException: org.apache.spark.sql.types.GenericArrayData cannot be cast to org.apache.spark.sql.catalyst.InternalRow

at org.apache.spark.sql.catalyst.expressions.BaseGenericInternalRow$class.getStruct(rows.scala:50)

at org.apache.spark.sql.catalyst.expressions.GenericMutableRow.getStruct(rows.scala:247)

at org.apache.spark.sql.catalyst.expressions.GeneratedClass$SpecificPredicate.eval(Unknown Source)

at org.apache.spark.sql.catalyst.expressions.codegen.GeneratePredicate$$anonfun$create$2.apply(GeneratePredicate.scala:67)

at org.apache.spark.sql.catalyst.expressions.codegen.GeneratePredicate$$anonfun$create$2.apply(GeneratePredicate.scala:67)

at org.apache.spark.sql.execution.Filter$$anonfun$4$$anonfun$apply$4.apply(basicOperators.scala:117)

at org.apache.spark.sql.execution.Filter$$anonfun$4$$anonfun$apply$4.apply(basicOperators.scala:115)

at scala.collection.Iterator$$anon$14.hasNext(Iterator.scala:390)

at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:327)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1.org$apache$spark$sql$execution$aggregate$TungstenAggregate$$anonfun$$executePartition$1(TungstenAggregate.scala:97)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1$$anonfun$2.apply(TungstenAggregate.scala:119)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1$$anonfun$2.apply(TungstenAggregate.scala:119)

at org.apache.spark.rdd.MapPartitionsWithPreparationRDD.compute(MapPartitionsWithPreparationRDD.scala:64)

at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:300)

at org.apache.spark.rdd.RDD.iterator(RDD.scala:264)

at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:38)

at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:300)

at org.apache.spark.rdd.RDD.iterator(RDD.scala:264)

at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:73)

at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:41)

at org.apache.spark.scheduler.Task.run(Task.scala:88)

at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:214)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)

at java.lang.Thread.run(Thread.java:745)

</pre>

</div></div>

<p>Here's a code snippet that reproduces the exception :</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>import org.apache.spark.SparkContext

import org.apache.spark.rdd.RDD

import org.apache.spark.sql.{SQLContext, DataFrame}

import org.apache.spark.sql.hive.HiveContext

import org.apache.spark.sql.types.{StringType, StructField, StructType}

object Snippet {

def main(args : Array[String]): Unit = {

val sc = new SparkContext()

implicit val sqlContext = new HiveContext(sc)

val rdd: RDD[String] = sc.parallelize(Seq(badRecord))

val df: DataFrame = sqlContext.read.schema(schema).json(rdd)

import sqlContext.implicits.\_

df.select("request.user.id")

.filter($"id".isNotNull)

.count()

}

val badRecord =

s"""{

| "request": {

| "user": []

| }

|}""".stripMargin.replaceAll("\n", " ") // Convert the multiline string to a signe line string

val schema =

StructType(

StructField("request", StructType(

StructField("user", StructType(

StructField("id", StringType) :: Nil

)) :: Nil

)) :: Nil)

}

</pre>

</div></div>

Title: [SPARK-13718] Scheduler "creating" straggler node

Description: <p><b>Data:</b></p>

<ul>

<li>Assume an even distribution of data across the cluster with a replication factor of 3.</li>

<li>In-memory data are partitioned in 128 chunks (384 cores in total, i.e. 3 requests can be executed concurrently(-ish) )</li>

</ul>

<p><b>Action:</b></p>

<ul>

<li>Action is a simple sequence of map/filter/reduce.</li>

<li>The action operates upon and returns a small subset of data (following the full map over the data).</li>

<li>Data are 1 x cached serialized in memory (Kryo), so calling the action should not hit the disk under normal conditions.</li>

<li>Action network usage is low as it returns a small number of aggregated results and does not require excessive shuffling</li>

<li>Under low or moderate load, each action is expected to complete in less than 2 seconds</li>

</ul>

<p><b>H/W Outlook</b><br/>

When the action is called in high numbers, initially the cluster CPU gets close to 100% (which is expected &amp; intended). <br/>

After a while, the cluster utilization reduces significantly with only one (struggler) node having 100% CPU and fully utilized network.</p>

<p><b>Diagnosis:</b><br/>

1. Attached a profiler to the driver and executors to monitor GC or I/O issues and everything is normal under low or heavy usage. <br/>

2. Cluster network usage is very low<br/>

3. No issues on Spark UI except that tasks begin to move from LOCAL to ANY</p>

<p><b>Cause (Corrected as found details in code):</b> <br/>

1. Node 'H' is doing marginally more work than the rest (being a little slower and at almost 100% CPU)<br/>

2. Scheduler hits the default 3000 millis spark.locality.wait and assigns the task to other nodes <br/>

3. One of the nodes 'X' that accepted the task will try to access the data from node 'H' HDD. This adds Network I/O to node and also some extra CPU for I/O.<br/>

4. 'X' time to complete increases ~5x as it goes over Network <br/>

5. Eventually, every node will have a task that is waiting to fetch that specific partition from node 'H' so cluster is basically blocked on a single node</p>

<p>What I managed to figure out from the code is that this is because if an RDD is cached, it will make use of BlockManager.getRemote() and will not recompute the DAG part that resulted in this RDD and hence always hit the node that has cached the RDD.</p>

<ul>

<li>Proposed Fix \*<br/>

I have not worked with Scala &amp; Spark source code enough to propose a code fix, but on a high level, when a task hits the 'spark.locality.wait' timeout, it could make use of a new configuration e.g. recomputeRddAfterLocalityTimeout instead of always trying to get the cached RDD. This would be very useful if it could also be manually set on the RDD.</li>

</ul>

<p><b>Workaround</b><br/>

Playing with 'spark.locality.wait' values, there is a deterministic value depending on partitions and config where the problem ceases to exist.</p>

<p><b>PS1</b> : Don't have enough Scala skils to follow the issue or propose a fix, but I hope that this has enough information to make sense.<br/>

<b>PS2</b> : Debugging this issue made me realize that there can be a lot of use-cases that trigger this behaviour</p>

Title: [SPARK-13717] Let RandomSampler can sample with Java iterator

Description: <p>Currently RandomSampler.sample only accepts Scala iterator. We should also let it accept Java iterator for better compatibility.</p>

Title: [SPARK-13716] mvn deploy tried to upload test jar of spark-network-common\_2.10 twice

Description: <p>spark/pom.xml:<br/>

&lt;plugin&gt;<br/>

&lt;groupId&gt;org.apache.maven.plugins&lt;/groupId&gt;<br/>

&lt;artifactId&gt;maven-jar-plugin&lt;/artifactId&gt;<br/>

&lt;executions&gt;<br/>

&lt;execution&gt;<br/>

&lt;id&gt;prepare-test-jar&lt;/id&gt;<br/>

&lt;phase&gt;prepare-package&lt;/phase&gt;<br/>

&lt;goals&gt;<br/>

&lt;goal&gt;test-jar&lt;/goal&gt;<br/>

&lt;/goals&gt;<br/>

&lt;configuration&gt;<br/>

&lt;excludes&gt;<br/>

&lt;exclude&gt;log4j.properties&lt;/exclude&gt;<br/>

&lt;/excludes&gt;<br/>

&lt;/configuration&gt;<br/>

&lt;/execution&gt;<br/>

&lt;/executions&gt;<br/>

&lt;/plugin&gt;<br/>

-----------------------------------------------------------------------<br/>

spark/network/common/pom.xml:<br/>

&lt;plugin&gt;<br/>

&lt;groupId&gt;org.apache.maven.plugins&lt;/groupId&gt;<br/>

&lt;artifactId&gt;maven-jar-plugin&lt;/artifactId&gt;<br/>

&lt;executions&gt;<br/>

&lt;execution&gt;<br/>

&lt;id&gt;test-jar-on-test-compile&lt;/id&gt;<br/>

&lt;phase&gt;test-compile&lt;/phase&gt;<br/>

&lt;goals&gt;<br/>

&lt;goal&gt;test-jar&lt;/goal&gt;<br/>

&lt;/goals&gt;<br/>

&lt;/execution&gt;<br/>

&lt;/executions&gt;<br/>

&lt;/plugin&gt;</p>

<p>maven-jar-plugin:test-jar goal is bound to both phase "test-compile" and phase "test-jar". As a result, when I tried to deploy spark to nexus repository, test.jar of spark-network-common\_2.10 was tried to be uploaded twice and failed.</p>

Title: [SPARK-13715] Remove last usages of jblas in tests

Description: <p>Only Spark test code uses jblas at this point. For completeness, these relatively few simple calls should be replaced with equivalent calls to breeze classes, or in some cases maybe even just double arrays.</p>

Title: [SPARK-13714] Another ConnectedComponents based on Max-Degree Propagation

Description: <p>Current ConnectedComponents algorithm was based on Min-VertexId Propagation, which is sensitive to the place of Min-VertexId.<br/>

While this implementation is based on Max-Degree Propagation.<br/>

First, the degree graph is computed. And in the pregel progress, the vertex with the max degree in a CC is the start point of propagation.<br/>

This new method has advantages over the old one:<br/>

1, The convergence is only determined by the structs of CC, and is robust to the place of vertex with Min-ID.<br/>

2, For spherical CCs in which there may be a concept like 'center', it can accelerate the convergence. For example, GraphGenerators.gridGraph(sc, 3, 3), the old CC need 4 supersteps, while the new one only need 2 supersteps.<br/>

3, If we limit the number of iteration, the new method tend to generate more acceptable results.<br/>

4, The output for each CC is the vertex with max degree in it, which may be more meaningful. And because the vertex-ID is nominal in most cases, the vertex with min-ID in a CC is somewhat meanless.</p>

<p>But there are still two disadvantages:<br/>

1,The message body grows, from (VID) to (VID, Degree). that is (Long) -&gt; (Long, Int)<br/>

2,For graph with simple CCs, it may be slower than old one. Because it need a extra degree computation.</p>

<p>The api is the same like ConnectedComponents:</p>

<p>val graph = ...<br/>

val cc = graph.ConnectedComponentsWithDegree(100)<br/>

or<br/>

val cc = ConnectedComponentsWithDegree.run(graph, 100)</p>

Title: [SPARK-13713] Replace ANTLR3 SQL parser by a ANTLR4 SQL parser

Description: <p>Replace the current SQL Parser by a version written in ANTLR4. This has various advantages:</p>

<ul>

<li>Much simpler structure</li>

<li>No code blowup</li>

<li>Reduction in lines of code</li>

</ul>

Title: [SPARK-13712] Add OneVsOne to ML

Description: <p>Another Meta method for multi-class classification.</p>

<p>Most classification algorithms were designed for balanced data.<br/>

The OneVsRest method will generate K models on imbalanced data.<br/>

The OneVsOne will train K\*(K-1)/2 models on balanced data.</p>

<p>OneVsOne is less sensitive to the problems of imbalanced datasets, and can usually result in higher precision.<br/>

But it is much more computationally expensive, although each model are trained on a much smaller dataset. (2/K of total)</p>

<p>The OneVsOne is implemented in the way OneVsRest did:</p>

<p>val classifier = new LogisticRegression()<br/>

val ovo = new OneVsOne()<br/>

ovo.setClassifier(classifier)<br/>

val ovoModel = ovo.fit(data)<br/>

val predictions = ovoModel.transform(data)</p>

Title: [SPARK-13711] Apache Spark driver stopping JVM when master not available

Description: <p>In my application Java spark context is created with an unavailable master URL (you may assume master is down for a maintenance). When creating Java spark context it leads to stopping JVM that runs spark driver with JVM exit code 50.</p>

<p>When I checked the logs I found SparkUncaughtExceptionHandler calling the System.exit. My program should run forever. </p>

<p>package test.mains;</p>

<p>import org.apache.spark.SparkConf;<br/>

import org.apache.spark.api.java.JavaSparkContext;</p>

<p>public class CheckJavaSparkContext {</p>

<p> public static void main(String[] args) {</p>

<p> SparkConf conf = new SparkConf();<br/>

conf.setAppName("test");<br/>

conf.setMaster("spark://sunshinee:7077");</p>

<p> try </p>

{

new JavaSparkContext(conf);

}

<p> catch (Throwable e) </p>

{

System.out.println("Caught an exception : " + e.getMessage());

}

<p> System.out.println("Waiting to complete...");<br/>

while (true) {<br/>

}<br/>

}</p>

<p>}</p>

<p>Output log</p>

<p>Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties<br/>

SLF4J: Class path contains multiple SLF4J bindings.<br/>

SLF4J: Found binding in <span class="error">&#91;jar:file:/data/downloads/spark-1.6.0-bin-hadoop2.6/lib/spark-assembly-1.6.0-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBinder.class&#93;</span><br/>

SLF4J: Found binding in <span class="error">&#91;jar:file:/data/downloads/spark-1.6.0-bin-hadoop2.6/lib/spark-examples-1.6.0-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBinder.class&#93;</span><br/>

SLF4J: See <a href="http://www.slf4j.org/codes.html#multiple\_bindings" class="external-link" rel="nofollow">http://www.slf4j.org/codes.html#multiple\_bindings</a> for an explanation.<br/>

SLF4J: Actual binding is of type <span class="error">&#91;org.slf4j.impl.Log4jLoggerFactory&#93;</span><br/>

16/03/04 18:01:15 INFO SparkContext: Running Spark version 1.6.0<br/>

16/03/04 18:01:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable<br/>

16/03/04 18:01:17 WARN Utils: Your hostname, pesamara-mobl-vm1 resolves to a loopback address: 127.0.0.1; using 10.30.9.107 instead (on interface eth0)<br/>

16/03/04 18:01:17 WARN Utils: Set SPARK\_LOCAL\_IP if you need to bind to another address<br/>

16/03/04 18:01:18 INFO SecurityManager: Changing view acls to: ps40233<br/>

16/03/04 18:01:18 INFO SecurityManager: Changing modify acls to: ps40233<br/>

16/03/04 18:01:18 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(ps40233); users with modify permissions: Set(ps40233)<br/>

16/03/04 18:01:19 INFO Utils: Successfully started service 'sparkDriver' on port 55309.<br/>

16/03/04 18:01:21 INFO Slf4jLogger: Slf4jLogger started<br/>

16/03/04 18:01:21 INFO Remoting: Starting remoting<br/>

16/03/04 18:01:22 INFO Remoting: Remoting started; listening on addresses :<span class="error">&#91;akka.tcp://sparkDriverActorSystem@10.30.9.107:52128&#93;</span><br/>

16/03/04 18:01:22 INFO Utils: Successfully started service 'sparkDriverActorSystem' on port 52128.<br/>

16/03/04 18:01:22 INFO SparkEnv: Registering MapOutputTracker<br/>

16/03/04 18:01:22 INFO SparkEnv: Registering BlockManagerMaster<br/>

16/03/04 18:01:22 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-87c20178-357d-4252-a46a-62a755568a98<br/>

16/03/04 18:01:22 INFO MemoryStore: MemoryStore started with capacity 457.7 MB<br/>

16/03/04 18:01:22 INFO SparkEnv: Registering OutputCommitCoordinator<br/>

16/03/04 18:01:23 INFO Utils: Successfully started service 'SparkUI' on port 4040.<br/>

16/03/04 18:01:23 INFO SparkUI: Started SparkUI at <a href="http://10.30.9.107:4040" class="external-link" rel="nofollow">http://10.30.9.107:4040</a><br/>

16/03/04 18:01:24 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:01:24 WARN AppClient$ClientEndpoint: Failed to connect to master sunshinee:7077<br/>

java.io.IOException: Failed to connect to sunshinee:7077<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216) at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:167)<br/>

at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:200)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:187)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:183)<br/>

at java.util.concurrent.FutureTask.run(FutureTask.java:266)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

Caused by: java.nio.channels.UnresolvedAddressException<br/>

at sun.nio.ch.Net.checkAddress(Net.java:123)<br/>

at sun.nio.ch.SocketChannelImpl.connect(SocketChannelImpl.java:621)<br/>

at io.netty.channel.socket.nio.NioSocketChannel.doConnect(NioSocketChannel.java:209)<br/>

at io.netty.channel.nio.AbstractNioChannel$AbstractNioUnsafe.connect(AbstractNioChannel.java:207)<br/>

at io.netty.channel.DefaultChannelPipeline$HeadContext.connect(DefaultChannelPipeline.java:1097)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelOutboundHandlerAdapter.connect(ChannelOutboundHandlerAdapter.java:47)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelDuplexHandler.connect(ChannelDuplexHandler.java:50)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:438)<br/>

at io.netty.channel.DefaultChannelPipeline.connect(DefaultChannelPipeline.java:908)<br/>

at io.netty.channel.AbstractChannel.connect(AbstractChannel.java:203)<br/>

at io.netty.bootstrap.Bootstrap$2.run(Bootstrap.java:166)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecutor.java:357)<br/>

at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:357)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:111)<br/>

... 1 more<br/>

16/03/04 18:01:44 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:01:44 WARN AppClient$ClientEndpoint: Failed to connect to master sunshinee:7077<br/>

java.io.IOException: Failed to connect to sunshinee:7077<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:167)<br/>

at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:200)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:187)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:183)<br/>

at java.util.concurrent.FutureTask.run(FutureTask.java:266)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

Caused by: java.nio.channels.UnresolvedAddressException<br/>

at sun.nio.ch.Net.checkAddress(Net.java:123)<br/>

at sun.nio.ch.SocketChannelImpl.connect(SocketChannelImpl.java:621)<br/>

at io.netty.channel.socket.nio.NioSocketChannel.doConnect(NioSocketChannel.java:209)<br/>

at io.netty.channel.nio.AbstractNioChannel$AbstractNioUnsafe.connect(AbstractNioChannel.java:207)<br/>

at io.netty.channel.DefaultChannelPipeline$HeadContext.connect(DefaultChannelPipeline.java:1097)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelOutboundHandlerAdapter.connect(ChannelOutboundHandlerAdapter.java:47)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelDuplexHandler.connect(ChannelDuplexHandler.java:50)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:438)<br/>

at io.netty.channel.DefaultChannelPipeline.connect(DefaultChannelPipeline.java:908)<br/>

at io.netty.channel.AbstractChannel.connect(AbstractChannel.java:203)<br/>

at io.netty.bootstrap.Bootstrap$2.run(Bootstrap.java:166)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecutor.java:357)<br/>

at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:357)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:111)<br/>

... 1 more<br/>

16/03/04 18:02:04 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:02:04 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:02:04 WARN AppClient$ClientEndpoint: Failed to connect to master sunshinee:7077<br/>

java.io.IOException: Failed to connect to sunshinee:7077<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:167)<br/>

at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:200)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:187)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:183)<br/>

at java.util.concurrent.FutureTask.run(FutureTask.java:266)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

Caused by: java.nio.channels.UnresolvedAddressException<br/>

at sun.nio.ch.Net.checkAddress(Net.java:123)<br/>

at sun.nio.ch.SocketChannelImpl.connect(SocketChannelImpl.java:621)<br/>

at io.netty.channel.socket.nio.NioSocketChannel.doConnect(NioSocketChannel.java:209)<br/>

at io.netty.channel.nio.AbstractNioChannel$AbstractNioUnsafe.connect(AbstractNioChannel.java:207)<br/>

at io.netty.channel.DefaultChannelPipeline$HeadContext.connect(DefaultChannelPipeline.java:1097)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelOutboundHandlerAdapter.connect(ChannelOutboundHandlerAdapter.java:47)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelDuplexHandler.connect(ChannelDuplexHandler.java:50)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:438)<br/>

at io.netty.channel.DefaultChannelPipeline.connect(DefaultChannelPipeline.java:908)<br/>

at io.netty.channel.AbstractChannel.connect(AbstractChannel.java:203)<br/>

at io.netty.bootstrap.Bootstrap$2.run(Bootstrap.java:166)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecutor.java:357)<br/>

at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:357)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:111)<br/>

... 1 more<br/>

16/03/04 18:02:24 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:02:24 INFO AppClient$ClientEndpoint: Connecting to master spark://sunshinee:7077...<br/>

16/03/04 18:02:24 WARN SparkDeploySchedulerBackend: Application ID is not initialized yet.<br/>

16/03/04 18:02:24 ERROR SparkDeploySchedulerBackend: Application has been killed. Reason: All masters are unresponsive! Giving up.<br/>

16/03/04 18:02:24 WARN AppClient$ClientEndpoint: Failed to connect to master sunshinee:7077<br/>

java.io.IOException: Failed to connect to sunshinee:7077<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:216)<br/>

at org.apache.spark.network.client.TransportClientFactory.createClient(TransportClientFactory.java:167)<br/>

at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:200)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:187)<br/>

at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:183)<br/>

at java.util.concurrent.FutureTask.run(FutureTask.java:266)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

Caused by: java.nio.channels.UnresolvedAddressException<br/>

at sun.nio.ch.Net.checkAddress(Net.java:123)<br/>

at sun.nio.ch.SocketChannelImpl.connect(SocketChannelImpl.java:621)<br/>

at io.netty.channel.socket.nio.NioSocketChannel.doConnect(NioSocketChannel.java:209)<br/>

at io.netty.channel.nio.AbstractNioChannel$AbstractNioUnsafe.connect(AbstractNioChannel.java:207)<br/>

at io.netty.channel.DefaultChannelPipeline$HeadContext.connect(DefaultChannelPipeline.java:1097)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelOutboundHandlerAdapter.connect(ChannelOutboundHandlerAdapter.java:47)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.ChannelDuplexHandler.connect(ChannelDuplexHandler.java:50)<br/>

at io.netty.channel.AbstractChannelHandlerContext.invokeConnect(AbstractChannelHandlerContext.java:471)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:456)<br/>

at io.netty.channel.AbstractChannelHandlerContext.connect(AbstractChannelHandlerContext.java:438)<br/>

at io.netty.channel.DefaultChannelPipeline.connect(DefaultChannelPipeline.java:908)<br/>

at io.netty.channel.AbstractChannel.connect(AbstractChannel.java:203)<br/>

at io.netty.bootstrap.Bootstrap$2.run(Bootstrap.java:166)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor.runAllTasks(SingleThreadEventExecutor.java:357)<br/>

at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:357)<br/>

at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:111)<br/>

... 1 more<br/>

16/03/04 18:02:24 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 44298.<br/>

16/03/04 18:02:24 INFO NettyBlockTransferService: Server created on 44298<br/>

16/03/04 18:02:24 INFO BlockManagerMaster: Trying to register BlockManager<br/>

16/03/04 18:02:24 INFO BlockManagerMasterEndpoint: Registering block manager 10.30.9.107:44298 with 457.7 MB RAM, BlockManagerId(driver, 10.30.9.107, 44298)<br/>

16/03/04 18:02:24 INFO BlockManagerMaster: Registered BlockManager<br/>

16/03/04 18:02:24 INFO SparkUI: Stopped Spark web UI at <a href="http://10.30.9.107:4040" class="external-link" rel="nofollow">http://10.30.9.107:4040</a><br/>

16/03/04 18:02:24 INFO SparkDeploySchedulerBackend: Shutting down all executors<br/>

16/03/04 18:02:24 INFO SparkDeploySchedulerBackend: Asking each executor to shut down<br/>

16/03/04 18:02:24 WARN AppClient$ClientEndpoint: Drop UnregisterApplication(null) because has not yet connected to master<br/>

16/03/04 18:02:24 ERROR SparkUncaughtExceptionHandler: Uncaught exception in thread Thread<span class="error">&#91;appclient-registration-retry-thread,5,main&#93;</span><br/>

java.lang.InterruptedException<br/>

at java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireSharedNanos(AbstractQueuedSynchronizer.java:1039)<br/>

at java.util.concurrent.locks.AbstractQueuedSynchronizer.tryAcquireSharedNanos(AbstractQueuedSynchronizer.java:1328)<br/>

at scala.concurrent.impl.Promise$DefaultPromise.tryAwait(Promise.scala:208)<br/>

at scala.concurrent.impl.Promise$DefaultPromise.ready(Promise.scala:218)<br/>

at scala.concurrent.impl.Promise$DefaultPromise.result(Promise.scala:223)<br/>

at scala.concurrent.Await$$anonfun$result$1.apply(package.scala:107)<br/>

at scala.concurrent.BlockContext$DefaultBlockContext$.blockOn(BlockContext.scala:53)<br/>

at scala.concurrent.Await$.result(package.scala:107)<br/>

at org.apache.spark.rpc.RpcTimeout.awaitResult(RpcTimeout.scala:75)<br/>

at org.apache.spark.deploy.client.AppClient.stop(AppClient.scala:290)<br/>

at org.apache.spark.scheduler.cluster.SparkDeploySchedulerBackend.org$apache$spark$scheduler$cluster$SparkDeploySchedulerBackend$$stop(SparkDeploySchedulerBackend.scala:198)<br/>

at org.apache.spark.scheduler.cluster.SparkDeploySchedulerBackend.stop(SparkDeploySchedulerBackend.scala:101)<br/>

at org.apache.spark.scheduler.TaskSchedulerImpl.stop(TaskSchedulerImpl.scala:446)<br/>

at org.apache.spark.scheduler.DAGScheduler.stop(DAGScheduler.scala:1582)<br/>

at org.apache.spark.SparkContext$$anonfun$stop$7.apply$mcV$sp(SparkContext.scala:1731)<br/>

at org.apache.spark.util.Utils$.tryLogNonFatalError(Utils.scala:1229)<br/>

at org.apache.spark.SparkContext.stop(SparkContext.scala:1730)<br/>

at org.apache.spark.scheduler.cluster.SparkDeploySchedulerBackend.dead(SparkDeploySchedulerBackend.scala:127)<br/>

at org.apache.spark.deploy.client.AppClient$ClientEndpoint.markDead(AppClient.scala:264)<br/>

at org.apache.spark.deploy.client.AppClient$ClientEndpoint$$anon$2$$anonfun$run$1.apply$mcV$sp(AppClient.scala:134)<br/>

at org.apache.spark.util.Utils$.tryOrExit(Utils.scala:1163)<br/>

at org.apache.spark.deploy.client.AppClient$ClientEndpoint$$anon$2.run(AppClient.scala:129)<br/>

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)<br/>

at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308)<br/>

at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$301(ScheduledThreadPoolExecutor.java:180)<br/>

at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:294)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)<br/>

at java.lang.Thread.run(Thread.java:745)<br/>

16/03/04 18:02:24 INFO DiskBlockManager: Shutdown hook called<br/>

16/03/04 18:02:24 INFO ShutdownHookManager: Shutdown hook called<br/>

16/03/04 18:02:24 INFO ShutdownHookManager: Deleting directory /tmp/spark-ea68a0fa-4f0d-4dbb-8407-cce90ef78a52<br/>

16/03/04 18:02:24 INFO ShutdownHookManager: Deleting directory /tmp/spark-ea68a0fa-4f0d-4dbb-8407-cce90ef78a52/userFiles-db548748-a55c-4406-adcb-c09e63b118bd<br/>

Java Result: 50 </p>

Title: [SPARK-13710] Spark shell shows ERROR when launching on Windows

Description: <p>On Windows, when we launch <tt>bin\spark-shell.cmd</tt>, it shows ERROR message and stacktrace.</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>C:\Users\tsudukim\Documents\workspace\spark-dev3&gt;bin\spark-shell

[ERROR] Terminal initialization failed; falling back to unsupported

java.lang.NoClassDefFoundError: Could not initialize class scala.tools.fusesource\_embedded.jansi.internal.Kernel32

at scala.tools.fusesource\_embedded.jansi.internal.WindowsSupport.getConsoleMode(WindowsSupport.java:50)

at scala.tools.jline\_embedded.WindowsTerminal.getConsoleMode(WindowsTerminal.java:204)

at scala.tools.jline\_embedded.WindowsTerminal.init(WindowsTerminal.java:82)

at scala.tools.jline\_embedded.TerminalFactory.create(TerminalFactory.java:101)

at scala.tools.jline\_embedded.TerminalFactory.get(TerminalFactory.java:158)

at scala.tools.jline\_embedded.console.ConsoleReader.&lt;init&gt;(ConsoleReader.java:229)

at scala.tools.jline\_embedded.console.ConsoleReader.&lt;init&gt;(ConsoleReader.java:221)

at scala.tools.jline\_embedded.console.ConsoleReader.&lt;init&gt;(ConsoleReader.java:209)

at scala.tools.nsc.interpreter.jline\_embedded.JLineConsoleReader.&lt;init&gt;(JLineReader.scala:61)

at scala.tools.nsc.interpreter.jline\_embedded.InteractiveReader.&lt;init&gt;(JLineReader.scala:33)

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)

at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)

at java.lang.reflect.Constructor.newInstance(Constructor.java:422)

at scala.tools.nsc.interpreter.ILoop$$anonfun$scala$tools$nsc$interpreter$ILoop$$instantiate$1$1.apply(ILoop.scala:865)

at scala.tools.nsc.interpreter.ILoop$$anonfun$scala$tools$nsc$interpreter$ILoop$$instantiate$1$1.apply(ILoop.scala:862)

at scala.tools.nsc.interpreter.ILoop.scala$tools$nsc$interpreter$ILoop$$mkReader$1(ILoop.scala:871)

at scala.tools.nsc.interpreter.ILoop$$anonfun$15$$anonfun$apply$8.apply(ILoop.scala:875)

at scala.tools.nsc.interpreter.ILoop$$anonfun$15$$anonfun$apply$8.apply(ILoop.scala:875)

at scala.util.Try$.apply(Try.scala:192)

at scala.tools.nsc.interpreter.ILoop$$anonfun$15.apply(ILoop.scala:875)

at scala.tools.nsc.interpreter.ILoop$$anonfun$15.apply(ILoop.scala:875)

at scala.collection.immutable.Stream$$anonfun$map$1.apply(Stream.scala:418)

at scala.collection.immutable.Stream$$anonfun$map$1.apply(Stream.scala:418)

at scala.collection.immutable.Stream$Cons.tail(Stream.scala:1233)

at scala.collection.immutable.Stream$Cons.tail(Stream.scala:1223)

at scala.collection.immutable.Stream.collect(Stream.scala:435)

at scala.tools.nsc.interpreter.ILoop.chooseReader(ILoop.scala:877)

at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1$$anonfun$apply$mcZ$sp$2.apply(ILoop.scala:916)

at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply$mcZ$sp(ILoop.scala:916)

at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply(ILoop.scala:911)

at scala.tools.nsc.interpreter.ILoop$$anonfun$process$1.apply(ILoop.scala:911)

at scala.reflect.internal.util.ScalaClassLoader$.savingContextLoader(ScalaClassLoader.scala:97)

at scala.tools.nsc.interpreter.ILoop.process(ILoop.scala:911)

at org.apache.spark.repl.Main$.doMain(Main.scala:64)

at org.apache.spark.repl.Main$.main(Main.scala:47)

at org.apache.spark.repl.Main.main(Main.scala)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:497)

at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:737)

at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:183)

at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:208)

at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:122)

at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel).

16/03/07 13:05:32 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Spark context available as sc (master = local[\*], app id = local-1457323533704).

SQL context available as sqlContext.

Welcome to

\_\_\_\_ \_\_

/ \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_

\_\ \/ \_ \/ \_ `/ \_\_/ '\_/

/\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 2.0.0-SNAPSHOT

/\_/

Using Scala version 2.11.7 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0\_40)

Type in expressions to have them evaluated.

Type :help for more information.

scala&gt; sc.textFile("README.md")

res0: org.apache.spark.rdd.RDD[String] = README.md MapPartitionsRDD[1] at textFile at &lt;console&gt;:25

scala&gt; sc.textFile("README.md").count()

res1: Long = 97

</pre>

</div></div>

<p>Spark-shell itself seems to work file during my simple operation check.</p>

Title: [SPARK-13709] Spark unable to decode Avro when partitioned

Description: <p>There is a problem decoding Avro data with SparkSQL when partitioned. The schema and encoded data are valid &#8211; I'm able to decode the data with the avro-tools CLI utility. I'm also able to decode the data with non-partitioned SparkSQL tables, Hive, other tools as well... except partitioned SparkSQL schemas.</p>

<p>For a simple example, I took the example schema and data found in the Oracle documentation here:</p>

<p><b>Schema</b></p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-javascript">

{

"type": "record",

"name": "MemberInfo",

"namespace": "avro",

"fields": [

{"name": "name", "type": {

"type": "record",

"name": "FullName",

"fields": [

{"name": "first", "type": "string"},

{"name": "last", "type": "string"}

]

}},

{"name": "age", "type": "<span class="code-keyword"><span class="code-object">int</span></span>"},

{"name": "address", "type": {

"type": "record",

"name": "Address",

"fields": [

{"name": "street", "type": "string"},

{"name": "city", "type": "string"},

{"name": "state", "type": "string"},

{"name": "zip", "type": "<span class="code-keyword"><span class="code-object">int</span></span>"}

]

}}

]

}

</pre>

</div></div>

<p><b>Data</b></p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-javascript">

{

"name": {

"first": "Percival",

"last": "Lowell"

},

"age": 156,

"address": {

"street": "Mars Hill Rd",

"city": "Flagstaff",

"state": "AZ",

"zip": 86001

}

}

</pre>

</div></div>

<p><b>Create</b> (no partitions - works)<br/>

If I create with no partitions, I'm able to query the data just fine.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-sql">

CREATE EXTERNAL TABLE IF NOT EXISTS foo

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'

STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'

OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'

LOCATION '/path/to/data/dir'

TBLPROPERTIES ('avro.schema.url'='/path/to/schema.avsc');

</pre>

</div></div>

<p><b>Create</b> (partitions &#8211; does NOT work)<br/>

If I create with no partitions, and then manually add a partition, all of my queries return an error. (I need to manually add partitions because I cannot control the structure of the data directories, so dynamic partitioning is not an option.)</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-sql">

CREATE EXTERNAL TABLE IF NOT EXISTS foo

PARTITIONED <span class="code-keyword">BY</span> (ds STRING)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'

STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'

OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'

TBLPROPERTIES ('avro.schema.url'='/path/to/schema.avsc');

ALTER TABLE foo ADD PARTITION (ds='1') LOCATION '/path/to/data/dir';

</pre>

</div></div>

<p>The error:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

spark-sql&gt; SELECT \* FROM foo WHERE ds = '1';

Driver stacktrace:

at org.apache.spark.scheduler.DAGScheduler.org$apache$spark$scheduler$DAGScheduler$$failJobAndIndependentStages(DAGScheduler.scala:1431)

at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.apply(DAGScheduler.scala:1419)

at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.apply(DAGScheduler.scala:1418)

at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)

at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:47)

at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGScheduler.scala:1418)

at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSetFailed$1.apply(DAGScheduler.scala:799)

at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSetFailed$1.apply(DAGScheduler.scala:799)

at scala.Option.foreach(Option.scala:236)

at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed(DAGScheduler.scala:799)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOnReceive(DAGScheduler.scala:1640)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1599)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1588)

at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:48)

at org.apache.spark.scheduler.DAGScheduler.runJob(DAGScheduler.scala:620)

at org.apache.spark.SparkContext.runJob(SparkContext.scala:1832)

at org.apache.spark.SparkContext.runJob(SparkContext.scala:1845)

at org.apache.spark.SparkContext.runJob(SparkContext.scala:1858)

at org.apache.spark.SparkContext.runJob(SparkContext.scala:1929)

at org.apache.spark.rdd.RDD$$anonfun$collect$1.apply(RDD.scala:927)

at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:150)

at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:111)

at org.apache.spark.rdd.RDD.withScope(RDD.scala:316)

at org.apache.spark.rdd.RDD.collect(RDD.scala:926)

at org.apache.spark.sql.execution.SparkPlan.executeCollect(SparkPlan.scala:166)

at org.apache.spark.sql.execution.SparkPlan.executeCollectPublic(SparkPlan.scala:174)

at org.apache.spark.sql.hive.HiveContext$QueryExecution.stringResult(HiveContext.scala:635)

at org.apache.spark.sql.hive.thriftserver.SparkSQLDriver.run(SparkSQLDriver.scala:64)

at org.apache.spark.sql.hive.thriftserver.SparkSQLCLIDriver.processCmd(SparkSQLCLIDriver.scala:308)

at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:376)

at org.apache.spark.sql.hive.thriftserver.SparkSQLCLIDriver$.main(SparkSQLCLIDriver.scala:226)

at org.apache.spark.sql.hive.thriftserver.SparkSQLCLIDriver.main(SparkSQLCLIDriver.scala)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:483)

at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:731)

at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:181)

at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:206)

at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:121)

at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

Caused by: org.apache.avro.AvroTypeException: Found avro.FullName, expecting union

at org.apache.avro.io.ResolvingDecoder.doAction(ResolvingDecoder.java:292)

at org.apache.avro.io.parsing.Parser.advance(Parser.java:88)

at org.apache.avro.io.ResolvingDecoder.readIndex(ResolvingDecoder.java:267)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.read(GenericDatumReader.java:155)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.readField(GenericDatumReader.java:193)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.readRecord(GenericDatumReader.java:183)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.read(GenericDatumReader.java:151)

at org.apache.avro.<span class="code-keyword">generic</span>.GenericDatumReader.read(GenericDatumReader.java:142)

at org.apache.hadoop.hive.serde2.avro.AvroDeserializer$SchemaReEncoder.reencode(AvroDeserializer.java:111)

at org.apache.hadoop.hive.serde2.avro.AvroDeserializer.deserialize(AvroDeserializer.java:175)

at org.apache.hadoop.hive.serde2.avro.AvroSerDe.deserialize(AvroSerDe.java:201)

at org.apache.spark.sql.hive.HadoopTableReader$$anonfun$fillObject$2.apply(TableReader.scala:409)

at org.apache.spark.sql.hive.HadoopTableReader$$anonfun$fillObject$2.apply(TableReader.scala:408)

at scala.collection.Iterator$$anon$11.next(Iterator.scala:328)

at scala.collection.Iterator$$anon$11.next(Iterator.scala:328)

at scala.collection.Iterator$class.foreach(Iterator.scala:727)

at scala.collection.AbstractIterator.foreach(Iterator.scala:1157)

at scala.collection.<span class="code-keyword">generic</span>.Growable$class.$plus$plus$eq(Growable.scala:48)

at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:103)

at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:47)

at scala.collection.TraversableOnce$class.to(TraversableOnce.scala:273)

at scala.collection.AbstractIterator.to(Iterator.scala:1157)

at scala.collection.TraversableOnce$class.toBuffer(TraversableOnce.scala:265)

at scala.collection.AbstractIterator.toBuffer(Iterator.scala:1157)

at scala.collection.TraversableOnce$class.toArray(TraversableOnce.scala:252)

at scala.collection.AbstractIterator.toArray(Iterator.scala:1157)

at org.apache.spark.rdd.RDD$$anonfun$collect$1$$anonfun$12.apply(RDD.scala:927)

at org.apache.spark.rdd.RDD$$anonfun$collect$1$$anonfun$12.apply(RDD.scala:927)

at org.apache.spark.SparkContext$$anonfun$runJob$5.apply(SparkContext.scala:1858)

at org.apache.spark.SparkContext$$anonfun$runJob$5.apply(SparkContext.scala:1858)

at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:66)

at org.apache.spark.scheduler.Task.run(Task.scala:89)

at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:213)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)

at java.lang.<span class="code-object">Thread</span>.run(<span class="code-object">Thread</span>.java:745)

</pre>

</div></div>

<p><b>Additional Info</b><br/>

For what it's worth, I found an issue (<a href="https://issues.apache.org/jira/browse/DRILL-957" title="Avro partitioned table fails with AvroTypeException" class="issue-link" data-issue-key="DRILL-957"><del>DRILL-957</del></a>) reported in Apache Drill and related fix that look very simliar to this. I'll look that to this issue.</p>

<p>Originally <a href="http://stackoverflow.com/questions/35826850/spark-unable-to-decode-avro-when-partitioned" class="external-link" rel="nofollow">posted here</a> on StackOverflow as a question, but I felt strongly that this is indeed a bug so I created this issue.</p>

Title: [SPARK-13708] Null pointer Exception while starting spark shell

Description: <p>16/03/06 11:49:18 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)<br/>

16/03/06 11:49:18 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)<br/>

16/03/06 11:49:24 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 1.2.0<br/>

16/03/06 11:49:24 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException<br/>

16/03/06 11:49:25 WARN : Your hostname, LAPTOP-B9GV6F82 resolves to a loopback/non-reachable address: fe80:0:0:0:1cbd:55b:b3e4:8e04%12, but we couldn't find any external IP address!<br/>

java.lang.RuntimeException: java.lang.NullPointerException<br/>

at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:522)<br/>

at org.apache.spark.sql.hive.client.ClientWrapper.&lt;init&gt;(ClientWrapper.scala:171)<br/>

at org.apache.spark.sql.hive.HiveContext.executionHive$lzycompute(HiveContext.scala:162)<br/>

at org.apache.spark.sql.hive.HiveContext.executionHive(HiveContext.scala:160)<br/>

at org.apache.spark.sql.hive.HiveContext.&lt;init&gt;(HiveContext.scala:167)<br/>

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)<br/>

at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)<br/>

at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source)<br/>

at java.lang.reflect.Constructor.newInstance(Unknown Source)<br/>

at org.apache.spark.repl.SparkILoop.createSQLContext(SparkILoop.scala:1028)<br/>

at $iwC$$iwC.&lt;init&gt;(&lt;console&gt;:9)<br/>

at $iwC.&lt;init&gt;(&lt;console&gt;:18)<br/>

at &lt;init&gt;(&lt;console&gt;:20)<br/>

at .&lt;init&gt;(&lt;console&gt;:24)<br/>

at .&lt;clinit&gt;(&lt;console&gt;)<br/>

at .&lt;init&gt;(&lt;console&gt;:7)<br/>

at .&lt;clinit&gt;(&lt;console&gt;)<br/>

at $print(&lt;console&gt;)<br/>

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)<br/>

at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)<br/>

at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)<br/>

at java.lang.reflect.Method.invoke(Unknown Source)<br/>

at org.apache.spark.repl.SparkIMain$ReadEvalPrint.call(SparkIMain.scala:1065)<br/>

at org.apache.spark.repl.SparkIMain$Request.loadAndRun(SparkIMain.scala:1340)<br/>

at org.apache.spark.repl.SparkIMain.loadAndRunReq$1(SparkIMain.scala:840)<br/>

at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:871)<br/>

at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:819)<br/>

at org.apache.spark.repl.SparkILoop.reallyInterpret$1(SparkILoop.scala:857)<br/>

at org.apache.spark.repl.SparkILoop.interpretStartingWith(SparkILoop.scala:902)<br/>

at org.apache.spark.repl.SparkILoop.command(SparkILoop.scala:814)<br/>

at org.apache.spark.repl.SparkILoopInit$$anonfun$initializeSpark$1.apply(SparkILoopInit.scala:132)<br/>

at org.apache.spark.repl.SparkILoopInit$$anonfun$initializeSpark$1.apply(SparkILoopInit.scala:124)<br/>

at org.apache.spark.repl.SparkIMain.beQuietDuring(SparkIMain.scala:324)<br/>

at org.apache.spark.repl.SparkILoopInit$class.initializeSpark(SparkILoopInit.scala:124)<br/>

at org.apache.spark.repl.SparkILoop.initializeSpark(SparkILoop.scala:64)<br/>

at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$process$1$$anonfun$apply$mcZ$sp$5.apply$mcV$sp(SparkILoop.scala:974)<br/>

at org.apache.spark.repl.SparkILoopInit$class.runThunks(SparkILoopInit.scala:159)<br/>

at org.apache.spark.repl.SparkILoop.runThunks(SparkILoop.scala:64)<br/>

at org.apache.spark.repl.SparkILoopInit$class.postInitialization(SparkILoopInit.scala:108)<br/>

at org.apache.spark.repl.SparkILoop.postInitialization(SparkILoop.scala:64)<br/>

at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$process$1.apply$mcZ$sp(SparkILoop.scala:991)<br/>

at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$process$1.apply(SparkILoop.scala:945)<br/>

at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$process$1.apply(SparkILoop.scala:945)<br/>

at scala.tools.nsc.util.ScalaClassLoader$.savingContextLoader(ScalaClassLoader.scala:135)<br/>

at org.apache.spark.repl.SparkILoop.org$apache$spark$repl$SparkILoop$$process(SparkILoop.scala:945)<br/>

at org.apache.spark.repl.SparkILoop.process(SparkILoop.scala:1059)<br/>

at org.apache.spark.repl.Main$.main(Main.scala:31)<br/>

at org.apache.spark.repl.Main.main(Main.scala)<br/>

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)<br/>

at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)<br/>

at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)<br/>

at java.lang.reflect.Method.invoke(Unknown Source)<br/>

at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:674)<br/>

at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:180)<br/>

at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:205)<br/>

at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:120)<br/>

at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)<br/>

Caused by: java.lang.NullPointerException<br/>

at java.lang.ProcessBuilder.start(Unknown Source)<br/>

at org.apache.hadoop.util.Shell.runCommand(Shell.java:482)<br/>

at org.apache.hadoop.util.Shell.run(Shell.java:455)<br/>

at org.apache.hadoop.util.Shell$ShellCommandExecutor.execute(Shell.java:715)<br/>

at org.apache.hadoop.util.Shell.execCommand(Shell.java:808)<br/>

at org.apache.hadoop.util.Shell.execCommand(Shell.java:791)<br/>

at org.apache.hadoop.fs.FileUtil.execCommand(FileUtil.java:1097)<br/>

at org.apache.hadoop.fs.RawLocalFileSystem$DeprecatedRawLocalFileStatus.loadPermissionInfo(RawLocalFileSystem.java:582)<br/>

at org.apache.hadoop.fs.RawLocalFileSystem$DeprecatedRawLocalFileStatus.getPermission(RawLocalFileSystem.java:557)<br/>

at org.apache.hadoop.hive.ql.session.SessionState.createRootHDFSDir(SessionState.java:599)<br/>

at org.apache.hadoop.hive.ql.session.SessionState.createSessionDirs(SessionState.java:554)<br/>

at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:508)<br/>

... 56 more</p>

Title: [SPARK-13707] Streaming UI tab misleading for window operations

Description: <p>'Streaming' tab on spark UI is misleading when the job has a window operation which changes the batch duration from original streaming context batch duration.</p>

<p>For instance consider:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

val streamingContext = <span class="code-keyword">new</span> StreamingContext(sparkConfig, Seconds(2))

val totalVideoImps = streamingContext.sparkContext.accumulator(0, <span class="code-quote">"TotalVideoImpressions"</span>)

val totalImps = streamingContext.sparkContext.accumulator(0, <span class="code-quote">"TotalImpressions"</span>)

val stream = KafkaReader.KafkaDirectStream(streamingContext)

stream.map(KafkaAdLogParser.parseAdLogRecord)

.filter(record =&gt; {

totalImps += 1

KafkaAdLogParser.isVideoRecord(record)

})

.map(record =&gt; {

totalVideoImps += 1

record.url

})

.window(Seconds(120))

.count().foreachRDD((rdd, time) =&gt; {

println(<span class="code-quote">"Timestamp: "</span> + ImpressionAggregator.millsToDate(time.milliseconds))

println(<span class="code-quote">"Count: "</span> + rdd.collect()(0))

println(<span class="code-quote">"Total Impressions: "</span> + totalImps.value)

totalImps.setValue(0)

println(<span class="code-quote">"Total Video Impressions: "</span> + totalVideoImps.value)

totalVideoImps.setValue(0)

})

streamingContext.start()

streamingContext.awaitTermination()

</pre>

</div></div>

<p>Batch Size before window operation is 2 sec and then after window batches are of 120 seconds each.<br/>

&#8211;</p>

<p>Above code printed following for my application whereas the UI showed different numbers.</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>Timestamp: 2016-03-06 12:02:56,000

Count: 362195

Total Impressions: 16882431

Total Video Impressions: 362195

Timestamp: 2016-03-06 12:04:56,000

Count: 367168

Total Impressions: 19480293

Total Video Impressions: 367168

Timestamp: 2016-03-06 12:06:56,000

Count: 177711

Total Impressions: 10196677

Total Video Impressions: 177711

</pre>

</div></div>

<p>whereas the spark UI shows different numbers as attached in the image. Also if we check the start and end index of kafka partition offsets reported by subsequent batch entries on UI, they do not result in all overall continuous range. All numbers are fine if we remove the window operation though.</p>

Title: [SPARK-13706] Python Example for Train Validation Split Missing

Description: <p>An example of how to use TrainValidationSplit in pyspark needs to be added. Should be consistent with the current examples. I'll submit a PR.</p>

Title: [SPARK-13705] UpdateStateByKey Operation documentation incorrectly refers to StatefulNetworkWordCount

Description: <p>Doc Issue : Apache Spark Streaming guide elaborates UpdateStateByKey Operation and goes onto give an example. However for complete code sample it referes to <tt>site.SPARK\_GITHUB\_URL</tt>/blob/master/examples/src/main/scala/org/apache<br/>

/spark/examples/streaming/StatefulNetworkWordCount.scala). StatefulNetworkWordCount.scala has changed to demonstrate more recent API mapWithState.<br/>

This creates confusion in the document. </p>

<p>Till the time more detailed explanation of mapWIthState is added the reference to StatefulNetworkWordCount.scala should be removed.</p>

Title: [SPARK-13704] TaskSchedulerImpl.createTaskSetManager can be expensive, and result in lost executors due to blocked heartbeats

Description: <p>In some cases, TaskSchedulerImpl.createTaskSetManager can be expensive. For example, in a Yarn cluster, it may call the topology script for rack awareness. When submit a very large job in a very large Yarn cluster, the topology script may take signifiant time to run. And this blocks receiving executors' heartbeats, which may result in lost executors</p>

<p>Stacktraces we observed which is related to this issue:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-quote">"dag-scheduler-event-loop"</span> daemon prio=10 tid=0x00007f8392875800 nid=0x26e8 runnable [0x00007f83576f4000]

java.lang.<span class="code-object">Thread</span>.State: RUNNABLE

at java.io.FileInputStream.readBytes(Native Method)

at java.io.FileInputStream.read(FileInputStream.java:272)

at java.io.BufferedInputStream.read1(BufferedInputStream.java:273)

at java.io.BufferedInputStream.read(BufferedInputStream.java:334)

- locked &lt;0x00000000f551f460&gt; (a java.lang.UNIXProcess$ProcessPipeInputStream)

at sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:283)

at sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:325)

at sun.nio.cs.StreamDecoder.read(StreamDecoder.java:177)

- locked &lt;0x00000000f5529740&gt; (a java.io.InputStreamReader)

at java.io.InputStreamReader.read(InputStreamReader.java:184)

at java.io.BufferedReader.fill(BufferedReader.java:154)

at java.io.BufferedReader.read1(BufferedReader.java:205)

at java.io.BufferedReader.read(BufferedReader.java:279)

- locked &lt;0x00000000f5529740&gt; (a java.io.InputStreamReader)

at org.apache.hadoop.util.Shell$ShellCommandExecutor.parseExecResult(Shell.java:728)

at org.apache.hadoop.util.Shell.runCommand(Shell.java:524)

at org.apache.hadoop.util.Shell.run(Shell.java:455)

at org.apache.hadoop.util.Shell$ShellCommandExecutor.execute(Shell.java:715)

at org.apache.hadoop.net.ScriptBasedMapping$RawScriptBasedMapping.runResolveCommand(ScriptBasedMapping.java:251)

at org.apache.hadoop.net.ScriptBasedMapping$RawScriptBasedMapping.resolve(ScriptBasedMapping.java:188)

at org.apache.hadoop.net.CachedDNSToSwitchMapping.resolve(CachedDNSToSwitchMapping.java:119)

at org.apache.hadoop.yarn.util.RackResolver.coreResolve(RackResolver.java:101)

at org.apache.hadoop.yarn.util.RackResolver.resolve(RackResolver.java:81)

at org.apache.spark.scheduler.cluster.YarnScheduler.getRackForHost(YarnScheduler.scala:38)

at org.apache.spark.scheduler.TaskSetManager$$anonfun$org$apache$spark$scheduler$TaskSetManager$$addPendingTask$1.apply(TaskSetManager.scala:210)

at org.apache.spark.scheduler.TaskSetManager$$anonfun$org$apache$spark$scheduler$TaskSetManager$$addPendingTask$1.apply(TaskSetManager.scala:189)

at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)

at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:47)

at org.apache.spark.scheduler.TaskSetManager.org$apache$spark$scheduler$TaskSetManager$$addPendingTask(TaskSetManager.scala:189)

at org.apache.spark.scheduler.TaskSetManager$$anonfun$1.apply$mcVI$sp(TaskSetManager.scala:158)

at scala.collection.immutable.Range.foreach$mVc$sp(Range.scala:141)

at org.apache.spark.scheduler.TaskSetManager.&lt;init&gt;(TaskSetManager.scala:157)

at org.apache.spark.scheduler.TaskSchedulerImpl.createTaskSetManager(TaskSchedulerImpl.scala:187)

at org.apache.spark.scheduler.TaskSchedulerImpl.submitTasks(TaskSchedulerImpl.scala:161)

- locked &lt;0x00000000ea3b8a88&gt; (a org.apache.spark.scheduler.cluster.YarnScheduler)

at org.apache.spark.scheduler.DAGScheduler.org$apache$spark$scheduler$DAGScheduler$$submitMissingTasks(DAGScheduler.scala:872)

at org.apache.spark.scheduler.DAGScheduler.org$apache$spark$scheduler$DAGScheduler$$submitStage(DAGScheduler.scala:778)

at org.apache.spark.scheduler.DAGScheduler.handleJobSubmitted(DAGScheduler.scala:762)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1362)

at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1354)

at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:48)

<span class="code-quote">"sparkDriver-akka.actor.<span class="code-keyword">default</span>-dispatcher-15"</span> daemon prio=10 tid=0x00007f829c020000 nid=0x2737 waiting <span class="code-keyword">for</span> monitor entry [0x00007f8355ebd000]

java.lang.<span class="code-object">Thread</span>.State: BLOCKED (on object monitor)

at org.apache.spark.scheduler.TaskSchedulerImpl.executorHeartbeatReceived(TaskSchedulerImpl.scala:362)

- waiting to lock &lt;0x00000000ea3b8a88&gt; (a org.apache.spark.scheduler.cluster.YarnScheduler)

at org.apache.spark.HeartbeatReceiver$$anonfun$receiveWithLogging$1.applyOrElse(HeartbeatReceiver.scala:46)

at scala.runtime.AbstractPartialFunction$mcVL$sp.apply$mcVL$sp(AbstractPartialFunction.scala:33)

at scala.runtime.AbstractPartialFunction$mcVL$sp.apply(AbstractPartialFunction.scala:33)

at scala.runtime.AbstractPartialFunction$mcVL$sp.apply(AbstractPartialFunction.scala:25)

at org.apache.spark.util.ActorLogReceive$$anon$1.apply(ActorLogReceive.scala:53)

at org.apache.spark.util.ActorLogReceive$$anon$1.apply(ActorLogReceive.scala:42)

at scala.PartialFunction$class.applyOrElse(PartialFunction.scala:118)

at org.apache.spark.util.ActorLogReceive$$anon$1.applyOrElse(ActorLogReceive.scala:42)

at akka.actor.ActorCell.receiveMessage(ActorCell.scala:498)

at akka.actor.ActorCell.invoke(ActorCell.scala:456)

at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:237)

at akka.dispatch.Mailbox.run(Mailbox.scala:219)

at akka.dispatch.ForkJoinExecutorConfigurator$AkkaForkJoinTask.exec(AbstractDispatcher.scala:386)

at scala.concurrent.forkjoin.ForkJoinTask.doExec(ForkJoinTask.java:260)

at scala.concurrent.forkjoin.ForkJoinPool$WorkQueue.runTask(ForkJoinPool.java:1339)

at scala.concurrent.forkjoin.ForkJoinPool.runWorker(ForkJoinPool.java:1979)

at scala.concurrent.forkjoin.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:107)

</pre>

</div></div>

Title: [SPARK-13703] Remove obsolete scala-2.10 source files

Description:

Title: [SPARK-13702] Use diamond operator for generic instance creation in Java code

Description: <p>Java 7 or higher supports `diamond` operator which replaces the type arguments required to invoke the constructor of a generic class with an empty set of type parameters (&lt;&gt;). Currently, Spark Java code use mixed usage of this. This issue replaces existing codes to use `diamond` operator and add Checkstyle rule.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

-List&lt;JavaPairDStream&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;&gt; kafkaStreams = <span class="code-keyword">new</span> ArrayList&lt;JavaPairDStream&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;&gt;(numStreams);

+List&lt;JavaPairDStream&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;&gt; kafkaStreams = <span class="code-keyword">new</span> ArrayList&lt;&gt;(numStreams);

</pre>

</div></div>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

-Set&lt;Tuple2&lt;<span class="code-object">Integer</span>, <span class="code-object">Integer</span>&gt;&gt; edges = <span class="code-keyword">new</span> HashSet&lt;Tuple2&lt;<span class="code-object">Integer</span>, <span class="code-object">Integer</span>&gt;&gt;(numEdges);

+Set&lt;Tuple2&lt;<span class="code-object">Integer</span>, <span class="code-object">Integer</span>&gt;&gt; edges = <span class="code-keyword">new</span> HashSet&lt;&gt;(numEdges);

</pre>

</div></div>

<p><b>Reference</b><br/>

<a href="https://docs.oracle.com/javase/8/docs/technotes/guides/language/type-inference-generic-instance-creation.html" class="external-link" rel="nofollow">https://docs.oracle.com/javase/8/docs/technotes/guides/language/type-inference-generic-instance-creation.html</a></p>

Title: [SPARK-13701] MLlib ALS fails on arm64 (java.lang.UnsatisfiedLinkError: org.jblas.NativeBlas.dgemm))

Description: <p>jblas fails on arm64.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

ALSSuite:

Exception encountered when attempting to run a suite with class name: org.apache.spark.mllib.recommendation.ALSSuite \*\*\* ABORTED \*\*\* (112 milliseconds)

java.lang.UnsatisfiedLinkError: org.jblas.NativeBlas.dgemm(CCIIID[DII[DIID[DII)V

at org.jblas.NativeBlas.dgemm(Native Method)

at org.jblas.SimpleBlas.gemm(SimpleBlas.java:247)

at org.jblas.DoubleMatrix.mmuli(DoubleMatrix.java:1781)

at org.jblas.DoubleMatrix.mmul(DoubleMatrix.java:3138)

at org.apache.spark.mllib.recommendation.ALSSuite$.generateRatings(ALSSuite.scala:74)

</pre>

</div></div>

Title: [SPARK-13700] Rdd.mapAsync(): Easily mix Spark and asynchroneous transformation

Description: <p>Spark is great for synchronous operations.</p>

<p>But sometimes I need to call a database/web server/etc from my transform, and the Spark pipeline stalls waiting for it.</p>

<p>Avoiding that would be great!</p>

<p>I suggest we add a new method RDD.mapAsync(), which can execute these operations concurrently, avoiding the bottleneck.</p>

<p>I've written a quick'n'dirty implementation of what I have in mind: <br/>

<a href="https://gist.github.com/paulo-raca/d121cf27905cfb1fafc3" class="external-link" rel="nofollow">https://gist.github.com/paulo-raca/d121cf27905cfb1fafc3</a></p>

<p>What do you think?</p>

<p>If you agree with this feature, I can work on a pull request.</p>

Title: [SPARK-13699] Spark SQL drops the table in "overwrite" mode while writing into table

Description: <p>Hi,</p>

<p>While writing the dataframe to HIVE table with "SaveMode.Overwrite" option.</p>

<p>E.g.<br/>

tgtFinal.write.mode(SaveMode.Overwrite).saveAsTable("tgt\_table")</p>

<p>sqlContext drop the table instead of truncating.</p>

<p>This is causing error while overwriting.</p>

<p>Adding stacktrace &amp; commands to reproduce the issue,</p>

<p>Thanks &amp; Regards,<br/>

Dhaval</p>

Title: [SPARK-13698] Fix Analysis Exceptions when Using Backticks in Generate

Description: <p>Analysis exception occurs while running the following query.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

SELECT ints FROM nestedArray LATERAL VIEW explode(a.b) `a` AS `ints`

</pre>

</div></div>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

Failed to analyze query: org.apache.spark.sql.AnalysisException: cannot resolve '`ints`' given input columns: [a, `ints`]; line 1 pos 7

'Project ['ints]

+- Generate explode(a#0.b), <span class="code-keyword">true</span>, <span class="code-keyword">false</span>, Some(a), [`ints`#8]

+- SubqueryAlias nestedarray

+- LocalRelation [a#0], [[[[1,2,3]]]]

</pre>

</div></div>

Title: [SPARK-13697] TransformFunctionSerializer.loads doesn't restore the function's module name if it's '\_\_main\_\_'

Description: <p>Here is a reproducer</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

&gt;&gt;&gt; from pyspark.streaming <span class="code-keyword">import</span> StreamingContext

&gt;&gt;&gt; from pyspark.streaming.util <span class="code-keyword">import</span> TransformFunction

&gt;&gt;&gt; ssc = StreamingContext(sc, 1)

&gt;&gt;&gt; func = TransformFunction(sc, lambda x: x, sc.serializer)

&gt;&gt;&gt; func.rdd\_wrapper(lambda x: x)

TransformFunction(&lt;function &lt;lambda&gt; at 0x106ac8b18&gt;)

&gt;&gt;&gt; bytes = bytearray(ssc.\_transformerSerializer.serializer.dumps((func.func, func.rdd\_wrap\_func, func.deserializers)))

&gt;&gt;&gt; func2 = ssc.\_transformerSerializer.loads(bytes)

&gt;&gt;&gt; print(func2.func.\_\_module\_\_)

None

&gt;&gt;&gt; print(func2.rdd\_wrap\_func.\_\_module\_\_)

None

&gt;&gt;&gt;

</pre>

</div></div>

Title: [SPARK-13696] Remove BlockStore interface to more cleanly reflect different memory and disk store responsibilities

Description: <p>Today, both the MemoryStore and DiskStore implement a common BlockStore API, but I feel that this API is inappropriate because it abstracts away important distinctions between the behavior of these two stores.</p>

<p>For instance, the disk store doesn't have a notion of storing deserialized objects, so it's confusing for it to expose object-based APIs like putIterator() and getValues() instead of only exposing binary APIs and pushing the responsibilities of serialization and deserialization to the client.</p>

<p>As part of a larger BlockManager interface cleanup, I'd like to remove the BlockStore API and refine the MemoryStore and DiskStore interfaces to reflect more narrow sets of responsibilities for those components.</p>

Title: [SPARK-13695] Don't cache MEMORY\_AND\_DISK blocks as bytes in memory store when reading spills

Description: <p>When a cached block is spilled to disk and read back in serialized form (i.e. as bytes), the current BlockManager implementation will attempt to re-insert the serialized block into the MemoryStore even if the block's storage level requests deserialized caching.</p>

<p>This behavior adds some complexity to the MemoryStore but I don't think it offers many performance benefits and I'd like to remove it in order to simplify a larger refactoring patch. Therefore, I propose to change the behavior such that disk store reads will only cache bytes in the memory store for blocks with serialized storage levels.</p>

<p>There are two places where we request serialized bytes from the BlockStore:</p>

<p>1. getLocalBytes(), which is only called when reading local copies of TorrentBroadcast pieces. Broadcast pieces are always cached using a serialized storage level, so this won't lead to a mismatch in serialization forms if spilled bytes read from disk are cached as bytes in the memory store.</p>

<p>2. the non-shuffle-block branch in getBlockData(), which is only called by the NettyBlockRpcServer when responding to requests to read remote blocks. Caching the serialized bytes in memory will only benefit us if those cached bytes are read before they're evicted and the likelihood of that happening seems low since the frequency of remote reads of non-broadcast cached blocks seems very low. Caching these bytes when they have a low probability of being read is bad if it risks the eviction of blocks which are cached in their expected serialized/deserialized forms, since those blocks seem more likely to be read in local computation.</p>

<p>Therefore, I think this is a safe change.</p>

Title: [SPARK-13694] QueryPlan.expressions should always include all expressions

Description:

Title: [SPARK-13693] Flaky test: o.a.s.streaming.MapWithStateSuite

Description: <p>Fixed the following flaky test:</p>

<p><a href="https://amplab.cs.berkeley.edu/jenkins/job/spark-master-test-sbt-hadoop-2.7/256/testReport/junit/org.apache.spark.streaming/MapWithStateSuite/\_It\_is\_not\_a\_test\_/" class="external-link" rel="nofollow">https://amplab.cs.berkeley.edu/jenkins/job/spark-master-test-sbt-hadoop-2.7/256/testReport/junit/org.apache.spark.streaming/MapWithStateSuite/\_It\_is\_not\_a\_test\_/</a></p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

sbt.ForkMain$ForkError: java.io.IOException: Failed to delete: /home/jenkins/workspace/spark-master-test-sbt-hadoop-2.7/streaming/checkpoint/spark-e97794a8-b940-4b21-8685-bf1221f9444d

at org.apache.spark.util.Utils$.deleteRecursively(Utils.scala:934)

at org.apache.spark.streaming.MapWithStateSuite$$anonfun$2.apply$mcV$sp(MapWithStateSuite.scala:47)

at org.apache.spark.streaming.MapWithStateSuite$$anonfun$2.apply(MapWithStateSuite.scala:45)

at org.apache.spark.streaming.MapWithStateSuite$$anonfun$2.apply(MapWithStateSuite.scala:45)

</pre>

</div></div>

Title: [SPARK-13692] Fix trivial Coverity/Checkstyle defects

Description: <p>This issue fixes the following potential bugs and Java coding style detected by Coverity and Checkstyle.</p>

<ul>

<li>Implement both null and type checking in equals functions.</li>

<li>Fix wrong type casting logic in SimpleJavaBean2.equals.</li>

<li>Add `implement Cloneable` to `UTF8String` and `SortedIterator`.</li>

<li>Remove dereferencing before null check in `AbstractBytesToBytesMapSuite`.</li>

<li>Fix coding style: Add '{}' to single `for` statement in mllib examples.</li>

<li>Remove unused imports in `ColumnarBatch` and `JavaKinesisStreamSuite.java`.</li>

<li>Remove unused fields in `ChunkFetchIntegrationSuite`.</li>

<li>Add `stop()` to prevent resource leak.</li>

</ul>

<p>Please note that the last two checkstyle errors exist on newly added commits after <a href="https://issues.apache.org/jira/browse/SPARK-13583" title="Remove unused imports and add checkstyle rule" class="issue-link" data-issue-key="SPARK-13583"><del>SPARK-13583</del></a>.</p>

Title: [SPARK-13691] Scala and Python generate inconsistent results

Description: <p>Here is an example that Scala and Python generate different results</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

Scala:

scala&gt; <span class="code-keyword">var</span> i = 0

i: Int = 0

scala&gt; val rdd = sc.parallelize(1 to 10).map(\_ + i)

scala&gt; rdd.collect()

res0: Array[Int] = Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

scala&gt; i += 1

scala&gt; rdd.collect()

res2: Array[Int] = Array(2, 3, 4, 5, 6, 7, 8, 9, 10, 11)

Python:

&gt;&gt;&gt; i = 0

&gt;&gt;&gt; rdd = sc.parallelize(range(1, 10)).map(lambda x: x + i)

&gt;&gt;&gt; rdd.collect()

[1, 2, 3, 4, 5, 6, 7, 8, 9]

&gt;&gt;&gt; i += 1

&gt;&gt;&gt; rdd.collect()

[1, 2, 3, 4, 5, 6, 7, 8, 9]

</pre>

</div></div>

<p>The difference is Scala will capture all variables' values when running a job every time, but Python just captures variables' values once and always uses them for all jobs.</p>

Title: [SPARK-13690] UnsafeShuffleWriterSuite fails on arm64 (SnappyError, no native library is found)

Description: <p>UnsafeShuffleWriterSuite fails because of missing Snappy native library on arm64.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

Tests run: 19, Failures: 0, Errors: 2, Skipped: 0, Time elapsed: 6.437 sec &lt;&lt;&lt; FAILURE! - in org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite

mergeSpillsWithFileStreamAndSnappy(org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite) Time elapsed: 0.072 sec &lt;&lt;&lt; ERROR!

java.lang.reflect.InvocationTargetException

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithFileStreamAndSnappy(UnsafeShuffleWriterSuite.java:389)

Caused by: java.lang.IllegalArgumentException: org.xerial.snappy.SnappyError: [FAILED\_TO\_LOAD\_NATIVE\_LIBRARY] no <span class="code-keyword">native</span> library is found <span class="code-keyword">for</span> os.name=Linux and os.arch=aarch64

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithFileStreamAndSnappy(UnsafeShuffleWriterSuite.java:389)

Caused by: org.xerial.snappy.SnappyError: [FAILED\_TO\_LOAD\_NATIVE\_LIBRARY] no <span class="code-keyword">native</span> library is found <span class="code-keyword">for</span> os.name=Linux and os.arch=aarch64

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithFileStreamAndSnappy(UnsafeShuffleWriterSuite.java:389)

mergeSpillsWithTransferToAndSnappy(org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite) Time elapsed: 0.041 sec &lt;&lt;&lt; ERROR!

java.lang.reflect.InvocationTargetException

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithTransferToAndSnappy(UnsafeShuffleWriterSuite.java:384)

Caused by: java.lang.IllegalArgumentException: java.lang.NoClassDefFoundError: Could not initialize class org.xerial.snappy.Snappy

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithTransferToAndSnappy(UnsafeShuffleWriterSuite.java:384)

Caused by: java.lang.NoClassDefFoundError: Could not initialize class org.xerial.snappy.Snappy

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.testMergingSpills(UnsafeShuffleWriterSuite.java:337)

at org.apache.spark.shuffle.sort.UnsafeShuffleWriterSuite.mergeSpillsWithTransferToAndSnappy(UnsafeShuffleWriterSuite.java:384)

Running org.apache.spark.JavaAPISuite

Tests run: 90, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 52.526 sec - in org.apache.spark.JavaAPISuite

Running org.apache.spark.unsafe.map.BytesToBytesMapOnHeapSuite

Tests run: 12, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 9.761 sec - in org.apache.spark.unsafe.map.BytesToBytesMapOnHeapSuite

Running org.apache.spark.unsafe.map.BytesToBytesMapOffHeapSuite

Tests run: 12, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 8.967 sec - in org.apache.spark.unsafe.map.BytesToBytesMapOffHeapSuite

Running org.apache.spark.api.java.OptionalSuite

Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec - in org.apache.spark.api.java.OptionalSuite

Results :

Tests in error:

UnsafeShuffleWriterSuite.mergeSpillsWithFileStreamAndSnappy:389-&gt;testMergingSpills:337 » InvocationTarget

UnsafeShuffleWriterSuite.mergeSpillsWithTransferToAndSnappy:384-&gt;testMergingSpills:337 » InvocationTarget

</pre>

</div></div>

Title: [SPARK-13689] Move some methods in CatalystQl to a util object

Description: <p>When we add more DDL parsing logic in the future, SparkQl will become very big. To keep it smaller, we'll introduce helper "parser objects", e.g. one to parse alter table commands. However, these parser objects will need to access some helper methods that exist in CatalystQl. The proposal is to move those methods to an isolated ParserUtils object.</p>

Title: [SPARK-13688] Add option to use dynamic allocation even if spark.executor.instances is set.

Description: <p>When both spark.dynamicAllocation.enabled and spark.executor.instances are set, dynamic resource allocation is disabled (see <a href="https://issues.apache.org/jira/browse/SPARK-9092" title="Make --num-executors compatible with dynamic allocation" class="issue-link" data-issue-key="SPARK-9092"><del>SPARK-9092</del></a>). This is a reasonable default, but I think there should be a configuration property to override it because it isn't obvious to users that dynamic allocation and number of executors are mutually exclusive. We see users setting --num-executors because that looks like what they want: a way to get more executors.</p>

<p>I propose adding a new boolean property, spark.dynamicAllocation.overrideNumExecutors, that makes dynamic allocation the default when both are set and uses --num-executors as the minimum number of executors.</p>

Title: [SPARK-13687] Cleanup pyspark temporary files

Description: <p>Every time parallelize is called it creates temporary file for rdd in spark.local.dir/spark-uuid/pyspark-uuid/ directory. This directory deletes when context is closed, but for long running applications with permanently opened context this directory growth infinitely.</p>

Title: [SPARK-13686] Add a constructor parameter `reqParam` to (Streaming)LinearRegressionWithSGD

Description: <p>`LinearRegressionWithSGD` and `StreamingLinearRegressionWithSGD` does not have `regParam` as their constructor arguments. They just depends on GradientDescent's default reqParam values. </p>

<p>To be consistent with other algorithms, we had better add them.</p>

Title: [SPARK-13685] Rename catalog.Catalog to ExternalCatalog

Description:

Title: [SPARK-13684] Possible unsafe bytesRead increment in StreamInterceptor

Description: <p>We unsafely increment a volatile (bytesRead) in a call back, if two call backs are triggered we may under count bytesRead. This issue was found using coverity.</p>

Title: [SPARK-13683] Finalize the public interface for OutputWriter[Factory]

Description: <p>We need to at least remove bucketing. I would also like to remove <tt>Job</tt> and the configuration stuff as well if possible.</p>

Title: [SPARK-13682] Finalize the public API for FileFormat

Description: <p>The current file format interface needs to be cleaned up before its acceptable for public consumption:</p>

<ul class="alternate" type="square">

<li>Have a version that takes Row and does a conversion, hide the internal API.</li>

<li>Remove bucketing</li>

<li>Remove RDD and the broadcastedConf</li>

<li>Remove SQLContext (maybe include SparkSession?)</li>

<li>Pass a better conf object</li>

</ul>

Title: [SPARK-13681] Reimplement CommitFailureTestRelationSuite

Description: <p>This test case got broken by <a href="https://github.com/apache/spark/pull/11509" class="external-link" rel="nofollow">#11509</a>. We should reimplement it as a format.</p>

Title: [SPARK-13680] Java UDAF with more than one intermediate argument returns wrong results

Description: <p>I am trying to incorporate the Java UDAF from <a href="https://github.com/apache/spark/blob/master/sql/hive/src/test/java/org/apache/spark/sql/hive/aggregate/MyDoubleAvg.java" class="external-link" rel="nofollow">https://github.com/apache/spark/blob/master/sql/hive/src/test/java/org/apache/spark/sql/hive/aggregate/MyDoubleAvg.java</a> into an SQL query. <br/>

I registered the UDAF like this:<br/>

sqlContext.udf().register("myavg", new MyDoubleAvg());</p>

<p>My SQL query is:<br/>

SELECT AVG(seqi) AS `avg\_seqi`, AVG(seqd) AS `avg\_seqd`, AVG(ci) AS `avg\_ci`, AVG(cd) AS `avg\_cd`, AVG(stdevd) AS `avg\_stdevd`, AVG(stdevi) AS `avg\_stdevi`, MAX(seqi) AS `max\_seqi`, MAX(seqd) AS `max\_seqd`, MAX(ci) AS `max\_ci`, MAX(cd) AS `max\_cd`, MAX(stdevd) AS `max\_stdevd`, MAX(stdevi) AS `max\_stdevi`, MIN(seqi) AS `min\_seqi`, MIN(seqd) AS `min\_seqd`, MIN(ci) AS `min\_ci`, MIN(cd) AS `min\_cd`, MIN(stdevd) AS `min\_stdevd`, MIN(stdevi) AS `min\_stdevi`,SUM(seqi) AS `sum\_seqi`, SUM(seqd) AS `sum\_seqd`, SUM(ci) AS `sum\_ci`, SUM(cd) AS `sum\_cd`, SUM(stdevd) AS `sum\_stdevd`, SUM(stdevi) AS `sum\_stdevi`, myavg(seqd) as `myavg\_seqd`, AVG(zero) AS `avg\_zero`, AVG(nulli) AS `avg\_nulli`,AVG(nulld) AS `avg\_nulld`, SUM(zero) AS `sum\_zero`, SUM(nulli) AS `sum\_nulli`,SUM(nulld) AS `sum\_nulld`,MAX(zero) AS `max\_zero`, MAX(nulli) AS `max\_nulli`,MAX(nulld) AS `max\_nulld`,count( \* ) AS `count\_all`, count(nulli) AS `count\_nulli` FROM mytable</p>

<p>As soon as I add the UDAF myavg to the SQL, all the results become incorrect. When I remove the call to the UDAF, the results are correct.<br/>

I was able to go around the issue by modifying bufferSchema of the UDAF to use an array and the corresponding update and merge methods. </p>

Title: [SPARK-13679] Pyspark job fails with Oozie

Description: <p>Hello,</p>

<p>I'm trying to run pi.py example in a pyspark job with Oozie. Every try I made failed for the same reason: key not found: SPARK\_HOME. <br/>

Note: A scala job works well in the environment with Oozie. </p>

<p>The logs on the executors are:</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/mnt/hd4/hadoop/yarn/local/filecache/145/slf4j-log4j12-1.6.6.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/mnt/hd2/hadoop/yarn/local/filecache/155/spark-assembly-1.6.0-hadoop2.7.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/opt/application/Hadoop/hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.

SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

log4j:ERROR setFile(null,true) call failed.

java.io.FileNotFoundException: /mnt/hd7/hadoop/yarn/log/application\_1454673025841\_13136/container\_1454673025841\_13136\_01\_000001 (Is a directory)

at java.io.FileOutputStream.open(Native Method)

at java.io.FileOutputStream.&lt;init&gt;(FileOutputStream.java:221)

at java.io.FileOutputStream.&lt;init&gt;(FileOutputStream.java:142)

at org.apache.log4j.FileAppender.setFile(FileAppender.java:294)

at org.apache.log4j.FileAppender.activateOptions(FileAppender.java:165)

at org.apache.hadoop.yarn.ContainerLogAppender.activateOptions(ContainerLogAppender.java:55)

at org.apache.log4j.config.PropertySetter.activate(PropertySetter.java:307)

at org.apache.log4j.config.PropertySetter.setProperties(PropertySetter.java:172)

at org.apache.log4j.config.PropertySetter.setProperties(PropertySetter.java:104)

at org.apache.log4j.PropertyConfigurator.parseAppender(PropertyConfigurator.java:809)

at org.apache.log4j.PropertyConfigurator.parseCategory(PropertyConfigurator.java:735)

at org.apache.log4j.PropertyConfigurator.configureRootCategory(PropertyConfigurator.java:615)

at org.apache.log4j.PropertyConfigurator.doConfigure(PropertyConfigurator.java:502)

at org.apache.log4j.PropertyConfigurator.doConfigure(PropertyConfigurator.java:547)

at org.apache.log4j.helpers.OptionConverter.selectAndConfigure(OptionConverter.java:483)

at org.apache.log4j.LogManager.&lt;clinit&gt;(LogManager.java:127)

at org.slf4j.impl.Log4jLoggerFactory.getLogger(Log4jLoggerFactory.java:64)

at org.slf4j.LoggerFactory.getLogger(LoggerFactory.java:285)

at org.apache.commons.logging.impl.SLF4JLogFactory.getInstance(SLF4JLogFactory.java:155)

at org.apache.commons.logging.impl.SLF4JLogFactory.getInstance(SLF4JLogFactory.java:132)

at org.apache.commons.logging.LogFactory.getLog(LogFactory.java:275)

at org.apache.hadoop.service.AbstractService.&lt;clinit&gt;(AbstractService.java:43)

Using properties file: null

Parsed arguments:

master yarn-master

deployMode cluster

executorMemory null

executorCores null

totalExecutorCores null

propertiesFile null

driverMemory null

driverCores null

driverExtraClassPath null

driverExtraLibraryPath null

driverExtraJavaOptions null

supervise false

queue null

numExecutors null

files null

pyFiles null

archives null

mainClass null

primaryResource hdfs://hadoopsandbox/User/toto/WORK/Oozie/pyspark/lib/pi.py

name Pysparkpi example

childArgs [100]

jars null

packages null

packagesExclusions null

repositories null

verbose true

Spark properties used, including those specified through

--conf and those from the properties file null:

spark.executorEnv.SPARK\_HOME -&gt; /opt/application/Spark/current

spark.executorEnv.PYTHONPATH -&gt; /opt/application/Spark/current/python

spark.yarn.appMasterEnv.SPARK\_HOME -&gt; /opt/application/Spark/current

Main class:

org.apache.spark.deploy.yarn.Client

Arguments:

--name

Pysparkpi example

--primary-py-file

hdfs://hadoopsandbox/User/toto/WORK/Oozie/pyspark/lib/pi.py

--class

org.apache.spark.deploy.PythonRunner

--arg

100

System properties:

spark.executorEnv.SPARK\_HOME -&gt; /opt/application/Spark/current

spark.executorEnv.PYTHONPATH -&gt; /opt/application/Spark/current/python

SPARK\_SUBMIT -&gt; true

spark.app.name -&gt; Pysparkpi example

spark.submit.deployMode -&gt; cluster

spark.yarn.appMasterEnv.SPARK\_HOME -&gt; /opt/application/Spark/current

spark.yarn.isPython -&gt; true

spark.master -&gt; yarn-cluster

Classpath elements:

Failing Oozie Launcher, Main class [org.apache.oozie.action.hadoop.SparkMain], main() threw exception, key not found: SPARK\_HOME

java.util.NoSuchElementException: key not found: SPARK\_HOME

at scala.collection.MapLike$class.default(MapLike.scala:228)

at scala.collection.AbstractMap.default(Map.scala:58)

at scala.collection.MapLike$class.apply(MapLike.scala:141)

at scala.collection.AbstractMap.apply(Map.scala:58)

at org.apache.spark.deploy.yarn.Client$$anonfun$findPySparkArchives$2.apply(Client.scala:1045)

at org.apache.spark.deploy.yarn.Client$$anonfun$findPySparkArchives$2.apply(Client.scala:1044)

at scala.Option.getOrElse(Option.scala:120)

at org.apache.spark.deploy.yarn.Client.findPySparkArchives(Client.scala:1044)

at org.apache.spark.deploy.yarn.Client.createContainerLaunchContext(Client.scala:717)

at org.apache.spark.deploy.yarn.Client.submitApplication(Client.scala:142)

at org.apache.spark.deploy.yarn.Client.run(Client.scala:1016)

at org.apache.spark.deploy.yarn.Client$.main(Client.scala:1076)

at org.apache.spark.deploy.yarn.Client.main(Client.scala)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:731)

at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:181)

at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:206)

at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:121)

at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

at org.apache.oozie.action.hadoop.SparkMain.runSpark(SparkMain.java:104)

at org.apache.oozie.action.hadoop.SparkMain.run(SparkMain.java:95)

at org.apache.oozie.action.hadoop.LauncherMain.run(LauncherMain.java:47)

at org.apache.oozie.action.hadoop.SparkMain.main(SparkMain.java:38)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.apache.oozie.action.hadoop.LauncherMapper.map(LauncherMapper.java:236)

at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:54)

at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:453)

at org.apache.hadoop.mapred.MapTask.run(MapTask.java:343)

at org.apache.hadoop.mapred.LocalContainerLauncher$EventHandler.runSubtask(LocalContainerLauncher.java:380)

at org.apache.hadoop.mapred.LocalContainerLauncher$EventHandler.runTask(LocalContainerLauncher.java:301)

at org.apache.hadoop.mapred.LocalContainerLauncher$EventHandler.access$200(LocalContainerLauncher.java:187)

at org.apache.hadoop.mapred.LocalContainerLauncher$EventHandler$1.run(LocalContainerLauncher.java:230)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:471)

at java.util.concurrent.FutureTask.run(FutureTask.java:262)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)

at java.lang.Thread.run(Thread.java:745)

log4j:WARN No appenders could be found for logger (org.apache.hadoop.mapreduce.v2.app.MRAppMaster).

log4j:WARN Please initialize the log4j system properly.

log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

</pre>

</div></div>

<p>The workflow used for Oozie is the following:</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>&lt;workflow-app xmlns='uri:oozie:workflow:0.5' name='PysparkPi-test'&gt;

&lt;start to='spark-node' /&gt;

&lt;action name='spark-node'&gt;

&lt;spark xmlns="uri:oozie:spark-action:0.1"&gt;

&lt;job-tracker&gt;${jobTracker}&lt;/job-tracker&gt;

&lt;name-node&gt;${nameNode}&lt;/name-node&gt;

&lt;master&gt;${master}&lt;/master&gt;

&lt;mode&gt;${mode}&lt;/mode&gt;

&lt;name&gt;Pysparkpi example&lt;/name&gt;

&lt;class&gt;&lt;/class&gt;

&lt;jar&gt;${nameNode}/User/toto/WORK/Oozie/pyspark/lib/pi.py&lt;/jar&gt;

&lt;spark-opts&gt;--conf spark.yarn.appMasterEnv.SPARK\_HOME=/opt/application/Spark/current --conf spark.executorEnv.SPARK\_HOME=/opt/application/Spark/current --conf spark.executorEnv.PYTHONPATH=/opt/application/Spark/current/python&lt;/spark-opts&gt;

&lt;arg&gt;100&lt;/arg&gt;

&lt;/spark&gt;

&lt;ok to="end" /&gt;

&lt;error to="fail" /&gt;

&lt;/action&gt;

&lt;kill name="fail"&gt;

&lt;message&gt;Workflow failed, error message[${wf:errorMessage(wf:lastErrorNode())}]&lt;/message&gt;

&lt;/kill&gt;

&lt;end name='end' /&gt;

&lt;/workflow-app&gt;

</pre>

</div></div>

Title: [SPARK-13678] transformExpressions should only apply on QueryPlan.expressions

Description:

Title: [SPARK-13677] Support Tree-Based Feature Transformation for mllib

Description: <p>It would be nice to be able to use RF and GBT for feature transformation:<br/>

First fit an ensemble of trees (like RF, GBT or other TreeEnsambleModels) on the training set. Then each leaf of each tree in the ensemble is assigned a fixed arbitrary feature index in a new feature space. These leaf indices are then encoded in a one-hot fashion.</p>

<p>This method was first introduced by facebook(<a href="http://www.herbrich.me/papers/adclicksfacebook.pdf" class="external-link" rel="nofollow">http://www.herbrich.me/papers/adclicksfacebook.pdf</a>), and is implemented in two famous library:<br/>

sklearn (<a href="http://scikit-learn.org/stable/auto\_examples/ensemble/plot\_feature\_transformation.html#example-ensemble-plot-feature-transformation-py" class="external-link" rel="nofollow">http://scikit-learn.org/stable/auto\_examples/ensemble/plot\_feature\_transformation.html#example-ensemble-plot-feature-transformation-py</a>)<br/>

xgboost (<a href="https://github.com/dmlc/xgboost/blob/master/demo/guide-python/predict\_leaf\_indices.py" class="external-link" rel="nofollow">https://github.com/dmlc/xgboost/blob/master/demo/guide-python/predict\_leaf\_indices.py</a>)</p>

<p>I have implement it in mllib:</p>

<p>val features : RDD<span class="error">&#91;Vector&#93;</span> = ...<br/>

val model1 : RandomForestModel = ...<br/>

val transformed1 : RDD<span class="error">&#91;Vector&#93;</span> = model1.leaf(features)</p>

<p>val model2 : GradientBoostedTreesModel = ...<br/>

val transformed2 : RDD<span class="error">&#91;Vector&#93;</span> = model2.leaf(features)</p>

Title: [SPARK-13676] Fix mismatched default values for regParam in LogisticRegression

Description: <p>The default value of regularization parameter for `LogisticRegression` algorithm is different in Scala and Python. We should provide the same value.</p>

<div class="code panel" style="border-style: solid;border-width: 1px;"><div class="codeHeader panelHeader" style="border-bottom-width: 1px;border-bottom-style: solid;"><b>Scala</b></div><div class="codeContent panelContent">

<pre class="code-java">

scala&gt; <span class="code-keyword">new</span> org.apache.spark.ml.classification.LogisticRegression().getRegParam

res0: <span class="code-object">Double</span> = 0.0

</pre>

</div></div>

<div class="code panel" style="border-style: solid;border-width: 1px;"><div class="codeHeader panelHeader" style="border-bottom-width: 1px;border-bottom-style: solid;"><b>Python</b></div><div class="codeContent panelContent">

<pre class="code-java">

&gt;&gt;&gt; from pyspark.ml.classification <span class="code-keyword">import</span> LogisticRegression

&gt;&gt;&gt; LogisticRegression().getRegParam()

0.1

</pre>

</div></div>

Title: [SPARK-13675] The url link in historypage is not correct for application running in yarn cluster mode

Description: <p>Current URL for each application to access history UI is like: <br/>

<a href="http://localhost:18080/history/application\_1457058760338\_0016/1/jobs/" class="external-link" rel="nofollow">http://localhost:18080/history/application\_1457058760338\_0016/1/jobs/</a> or <a href="http://localhost:18080/history/application\_1457058760338\_0016/2/jobs/" class="external-link" rel="nofollow">http://localhost:18080/history/application\_1457058760338\_0016/2/jobs/</a></p>

<p>Here <b>1</b> or <b>2</b> represents the number of attempts in <tt>historypage.js</tt>, but it will parse to attempt id in <tt>HistoryServer</tt>, while the correct attempt id should be like "appattempt\_1457058760338\_0016\_000002", so it will fail to parse to a correct attempt id in <tt>HistoryServer</tt>.</p>

<p>This is OK in yarn client mode, since we don't need this attempt id to fetch out the app cache, but it is failed in yarn cluster mode, where attempt id "1" or "2" is actually wrong.</p>

<p>So here we should fix this url to parse the correct application id and attempt id.</p>

<p>This bug is newly introduced in <a href="https://issues.apache.org/jira/browse/SPARK-10873" title="Change history to use datatables to support sorting columns and searching" class="issue-link" data-issue-key="SPARK-10873"><del>SPARK-10873</del></a>, there's no issue in branch 1.6.</p>

<p>Here is the screenshot:</p>

<p><img src="https://issues.apache.org/jira/secure/attachment/12791437/Screen%20Shot%202016-02-29%20at%203.57.32%20PM.png" align="absmiddle" border="0" /></p>

Title: [SPARK-13674] Add wholestage codegen support to Sample

Description: <p>Sample operator doesn't support wholestage codegen now. This issue is opened to add support for it.</p>

Title: [SPARK-13673] script bin\beeline.cmd pollutes environment variables in Windows.

Description: <p><tt>bin\beeline.cmd</tt> pollutes environment variables in Windows.<br/>

The similar problem is reported and fixed in <a href="https://issues.apache.org/jira/browse/SPARK-3943" title="Some scripts bin\\*.cmd pollutes environment variables in Windows." class="issue-link" data-issue-key="SPARK-3943"><del>SPARK-3943</del></a>, but <tt>bin\beeline.cmd</tt> is added later.</p>

Title: [SPARK-13672] Add python examples of BisectingKMeans in ML and MLLIB

Description: <p>add the missing python examples of BisectingKMeans for ml and mllib</p>

Title: [SPARK-13671] Use different physical plan for existing RDD and data sources

Description: <p>Right now, we use PhysicalRDD for both existing RDD and data sources, they are becoming much different, we should use different physical plans for them.</p>

Title: [SPARK-13670] spark-class doesn't bubble up error from launcher command

Description: <p>There's a particular snippet in spark-class <a href="https://github.com/apache/spark/blob/master/bin/spark-class#L86" class="external-link" rel="nofollow">here</a> that runs the spark-launcher code in a subshell.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

# The launcher library will print arguments separated by a NULL character, to allow arguments with

# characters that would be otherwise interpreted by the shell. Read that in a <span class="code-keyword">while</span> loop, populating

# an array that will be used to exec the <span class="code-keyword">final</span> command.

CMD=()

<span class="code-keyword">while</span> IFS= read -d '' -r ARG; <span class="code-keyword">do</span>

CMD+=(<span class="code-quote">"$ARG"</span>)

done &lt; &lt;(<span class="code-quote">"$RUNNER"</span> -cp <span class="code-quote">"$LAUNCH\_CLASSPATH"</span> org.apache.spark.launcher.Main <span class="code-quote">"$@"</span>)

</pre>

</div></div>

<p>The problem is that the if the launcher Main fails, this code still still returns success and continues, even though the top level script is marked <tt>set -e</tt>. This is because the launcher.Main is run within a subshell.</p>

Title: [SPARK-13669] Job will always fail in the external shuffle service unavailable situation

Description: <p>Currently we are running into an issue with Yarn work preserving enabled + external shuffle service. </p>

<p>In the work preserving enabled scenario, the failure of NM will not lead to the exit of executors, so executors can still accept and run the tasks. The problem here is when NM is failed, external shuffle service is actually inaccessible, so reduce tasks will always complain about the “Fetch failure”, and the failure of reduce stage will make the parent stage (map stage) rerun. The tricky thing here is Spark scheduler is not aware of the unavailability of external shuffle service, and will reschedule the map tasks on the executor where NM is failed, and again reduce stage will be failed with “Fetch failure”, and after 4 retries, the job is failed.</p>

<p>So here the actual problem is Spark’s scheduler is not aware of the unavailability of external shuffle service, and will still assign the tasks on to that nodes. The fix is to avoid assigning tasks on to that nodes.</p>

<p>Currently in the Spark, one related configuration is “spark.scheduler.executorTaskBlacklistTime”, but I don’t think it will be worked in this scenario. This configuration is used to avoid same reattempt task to run on the same executor. Also ways like MapReduce’s blacklist mechanism may not handle this scenario, since all the reduce tasks will be failed, so counting the failure tasks will equally mark all the executors as “bad” one.</p>

Title: [SPARK-13668] Reorder filter/join predicates to short-circuit isNotNull checks

Description: <p>If a filter predicate or a join condition consists of `IsNotNull` checks, we should reorder these checks such that these non-nullability checks are evaluated before the rest of the predicates.</p>

<p>For e.g., if a filter predicate is of the form `a &gt; 5 &amp;&amp; isNotNull(b)`, we should rewrite this as `isNotNull(b) &amp;&amp; a &gt; 5` during physical plan generation.</p>

<p>cc <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=nongli" class="user-hover" rel="nongli">Nong Li</a> <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=yhuai" class="user-hover" rel="yhuai">Yin Huai</a></p>

Title: [SPARK-13667] Support for specifying custom date format for date and timestamp types

Description: <p>Currently, CSV data source does not support to parse date and timestamp types in custom format and infer the type of timestamp type in custom format.</p>

<p>It looks quite many of users want this feature. It would be great to set custom date format.</p>

<p>This was reported in spark-csv.<br/>

<a href="https://github.com/databricks/spark-csv/issues/279" class="external-link" rel="nofollow">https://github.com/databricks/spark-csv/issues/279</a><br/>

<a href="https://github.com/databricks/spark-csv/issues/262" class="external-link" rel="nofollow">https://github.com/databricks/spark-csv/issues/262</a><br/>

<a href="https://github.com/databricks/spark-csv/issues/266" class="external-link" rel="nofollow">https://github.com/databricks/spark-csv/issues/266</a></p>

<p>Currently I submitted a PR for this in spark-csv<br/>

<a href="https://github.com/databricks/spark-csv/pull/280" class="external-link" rel="nofollow">https://github.com/databricks/spark-csv/pull/280</a></p>

Title: [SPARK-13666] Annoying warnings from SQLConf in log output

Description: <p>Whenever I run spark-shell I get a bunch of warnings about SQL configuration:</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>16/03/03 19:00:25 WARN hive.HiveSessionState$$anon$1: Attempt to set non-Spark SQL config in SQLConf: key = spark.yarn.driver.memoryOverhead, value = 26

16/03/03 19:00:25 WARN hive.HiveSessionState$$anon$1: Attempt to set non-Spark SQL config in SQLConf: key = spark.yarn.executor.memoryOverhead, value = 26

16/03/03 19:00:25 WARN hive.HiveSessionState$$anon$1: Attempt to set non-Spark SQL config in SQLConf: key = spark.executor.cores, value = 1

16/03/03 19:00:25 WARN hive.HiveSessionState$$anon$1: Attempt to set non-Spark SQL config in SQLConf: key = spark.executor.memory, value = 268435456

</pre>

</div></div>

<p>That should't happen, since I'm not setting those values explicitly. They're either set internally by Spark or come from spark-defaults.conf.</p>

Title: [SPARK-13665] Initial separation of concerns

Description: <p>The goal here is to break apart: File Management, code to deal with specific formats and query planning.</p>

Title: [SPARK-13664] Simplify and Speedup HadoopFSRelation

Description: <p>A majority of Spark SQL queries likely run though <tt>HadoopFSRelation</tt>, however there are currently several complexity and performance problems with this code path:</p>

<ul class="alternate" type="square">

<li>The class mixes the concerns of file management, schema reconciliation, scan building, bucketing, partitioning, and writing data.</li>

<li>For very large tables, we are broadcasting the entire list of files to every executor. <a href="https://issues.apache.org/jira/browse/SPARK-11441" title="HadoopFsRelation is not scalable in number of files read/written" class="issue-link" data-issue-key="SPARK-11441">SPARK-11441</a></li>

<li>For partitioned tables, we always do an extra projection. This results not only in a copy, but undoes much of the performance gains that we are going to get from vectorized reads.</li>

</ul>

<p>This is an umbrella ticket to track a set of improvements to this codepath.</p>

Title: [SPARK-13663] Upgrade Snappy Java to 1.1.2.1

Description: <p>The JVM memory leaky problem reported in <a href="https://github.com/xerial/snappy-java/issues/131" class="external-link" rel="nofollow">https://github.com/xerial/snappy-java/issues/131</a> has been resolved.</p>

<p>1.1.2.1 was released on Jan 22nd.</p>

<p>We should upgrade to this release.</p>

Title: [SPARK-13662] [SQL][Hive] Have SHOW TABLES return additional fields from Hive MetaStore

Description: <p>Currently, the SHOW TABLES command in Spark's Hive ThriftServer, or equivalently the HiveContext.tables method, returns a DataFrame with only two columns: the name of the table and whether it is temporary. It would be really nice to add support to return some extra information, such as:</p>

<ul class="alternate" type="square">

<li>Whether this table is Spark-only or a native Hive table</li>

<li>If spark-only, the name of the data source</li>

<li>potentially other properties</li>

</ul>

<p>The first two is really useful for BI environments connecting to multiple data sources and that work with both Hive and Spark.</p>

<p>Some thoughts:</p>

<ul class="alternate" type="square">

<li>The SQL/HiveContext Catalog API might need to be expanded to return something like a TableEntry, rather than just a tuple of (name, temporary).</li>

<li>I believe there is a Hive Catalog/client API to get information about each table. I suppose one concern would be the speed of using this API. Perhaps there are other APis that can get this info faster.</li>

</ul>

Title: [SPARK-13661] Avoid the copy of UnsafeRow in HashedRelation

Description: <p>We usually build the HashedRelation on top of array of UnsafeRow, the copy could be avoided.</p>

<p>The caller of HashedRelation need to do the copy if it's needed.</p>

<p>Another approach could be making the copy() of UnsafeRow smart so that it know when should copy the bytes or not, this could be also useful for other components. </p>

Title: [SPARK-13660] CommitFailureTestRelationSuite floods the logs with garbage

Description: <p><a href="https://github.com/apache/spark/pull/11439" class="external-link" rel="nofollow">https://github.com/apache/spark/pull/11439</a> added a utility method "testQuietly". We can use it for CommitFailureTestRelationSuite.</p>

Title: [SPARK-13659] Remove returnValues from BlockStore APIs

Description: <p>In preparation for larger refactorings, I think that we should remove the confusing returnValues() option from the BlockStore put() APIs: returning the value is only useful in one place (caching) and in other situations, such as block replication, it's simpler to put() and then get().</p>

Title: [SPARK-13658] BooleanSimplification rule is slow with large boolean expressions

Description: <p>When run TPCDS Q3 <span class="error">&#91;1&#93;</span> with lots predicates to filter out the partitions, the optimizer rule BooleanSimplification take about 2 seconds (it use lots of sematicsEqual, which require copy the whole tree).</p>

<p>It will great if we could speedup it.</p>

<p><span class="error">&#91;1&#93;</span> <a href="https://github.com/cloudera/impala-tpcds-kit/blob/master/queries/q3.sql" class="external-link" rel="nofollow">https://github.com/cloudera/impala-tpcds-kit/blob/master/queries/q3.sql</a></p>

<p>cc <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=marmbrus" class="user-hover" rel="marmbrus">Michael Armbrust</a></p>

Title: [SPARK-13657] Support parsing very long AND/OR expression

Description: <p>CatalystQl can't parse an expression with hundreds of AND/OR <span class="error">&#91;1&#93;</span>, it will fail as StackOverflow. </p>

<p><span class="error">&#91;1&#93;</span> <a href="https://github.com/cloudera/impala-tpcds-kit/blob/master/queries/q3.sql" class="external-link" rel="nofollow">https://github.com/cloudera/impala-tpcds-kit/blob/master/queries/q3.sql</a></p>

Title: [SPARK-13656] Delete spark.sql.parquet.cacheMetadata

Description: <p>Looks like spark.sql.parquet.cacheMetadata is not used anymore. Let's delete it to avoid any potential confusion.</p>

Title: [SPARK-13655] Fix WithAggregationKinesisBackedBlockRDDSuite

Description: <p>`WithAggregationKinesisBackedBlockRDDSuite` test consistently hangs and fails due to timeout. Currently, this is observed in <a href="https://github.com/apache/spark/pull/11481" class="external-link" rel="nofollow">PR-11481</a> and <a href="https://github.com/apache/spark/pull/11438" class="external-link" rel="nofollow">PR-11438</a>. Inevitably, those PR removes their changes on Kinesis module to skip Kinesis test.</p>

<div class="code panel" style="border-style: solid;border-width: 1px;"><div class="codeHeader panelHeader" style="border-bottom-width: 1px;border-bottom-style: solid;"><b>SparkPullRequestBuilder-52389</b></div><div class="codeContent panelContent">

<pre class="code-java">

[info] WithAggregationKinesisBackedBlockRDDSuite:

Using endpoint URL https:<span class="code-comment">//kinesis.us-west-2.amazonaws.com <span class="code-keyword">for</span> creating Kinesis streams <span class="code-keyword">for</span> tests.

</span>[2016-03-03 02:11:56.990918] [0x00007fabfb82f700] [info] [kinesis\_producer.cc:79] Created pipeline <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-7296082419112679238"</span>

[2016-03-03 02:11:56.991038] [0x00007fabfb82f700] [info] [shard\_map.cc:83] Updating shard map <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-7296082419112679238"</span>

[2016-03-03 02:11:57.230083] [0x00007fabfa42d700] [info] [shard\_map.cc:163] Successfully updated shard map <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-7296082419112679238"</span> found 2 shards

[info] - Basic reading from Kinesis (2 seconds, 116 milliseconds)

[info] - Read data available in both block manager and Kinesis (65 milliseconds)

Attempting to post to Github...

&gt; Post successful.

Build step 'Execute shell' marked build as failure

Archiving artifacts

Recording test results

Test FAILed.

Refer to <span class="code-keyword">this</span> link <span class="code-keyword">for</span> build results (access rights to CI server needed):

https:<span class="code-comment">//amplab.cs.berkeley.edu/jenkins//job/SparkPullRequestBuilder/52389/

</span>Test FAILed.

Finished: FAILURE

</pre>

</div></div>

<div class="code panel" style="border-style: solid;border-width: 1px;"><div class="codeHeader panelHeader" style="border-bottom-width: 1px;border-bottom-style: solid;"><b>NewSparkPullRequestBuilder-2608</b></div><div class="codeContent panelContent">

<pre class="code-java">

[info] WithAggregationKinesisBackedBlockRDDSuite:

[2016-03-03 10:28:05.512018] [0x00007f0fe5705700] [info] [kinesis\_producer.cc:79] Created pipeline <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-1569968002408612511"</span>

[2016-03-03 10:28:05.512313] [0x00007f0fe5705700] [info] [shard\_map.cc:83] Updating shard map <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-1569968002408612511"</span>

[2016-03-03 10:28:05.739267] [0x00007f0fe4303700] [info] [shard\_map.cc:163] Successfully updated shard map <span class="code-keyword">for</span> stream <span class="code-quote">"KinesisTestUtils-1569968002408612511"</span> found 2 shards

[info] - Basic reading from Kinesis (1 second, 504 milliseconds)

[info] - Read data available in both block manager and Kinesis (37 milliseconds)

Build timed out (after 180 minutes). Marking the build as failed.

Build was aborted

Archiving artifacts

Recording test results

Finished: FAILURE

</pre>

</div></div>

Title: [SPARK-13654] get\_json\_object fails with java.io.CharConversionException

Description: <p>I execute next query on my data:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

select count(distinct get\_json\_object(regexp\_extract(line, <span class="code-quote">"^\\p{ASCII}\*$"</span>, 0), '$.event')) from

(select line from logs.raw\_client\_log where year=2016 and month=2 and day&gt;28 and line rlike <span class="code-quote">"^\\p{ASCII}\*$"</span> and line is not <span class="code-keyword">null</span>) a

</pre>

</div></div>

<p>And it fails with </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

Error: org.apache.spark.SparkException: Job aborted due to stage failure: Task 420 in stage 168.0 failed 4 times, most recent failure: Lost task 420.3 in stage 168.0 (TID 13064, nod5-2-hadoop.anchorfree.net): java.io.CharConversionException: Invalid UTF-32 character 0x6576656e(above 10ffff) at <span class="code-object">char</span> #47, <span class="code-object">byte</span> #191)

at com.fasterxml.jackson.core.io.UTF32Reader.reportInvalid(UTF32Reader.java:189)

at com.fasterxml.jackson.core.io.UTF32Reader.read(UTF32Reader.java:150)

at com.fasterxml.jackson.core.json.ReaderBasedJsonParser.loadMore(ReaderBasedJsonParser.java:153)

at com.fasterxml.jackson.core.json.ReaderBasedJsonParser.\_skipWSOrEnd(ReaderBasedJsonParser.java:1855)

at com.fasterxml.jackson.core.json.ReaderBasedJsonParser.nextToken(ReaderBasedJsonParser.java:571)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject$$anonfun$eval$2$$anonfun$4.apply(jsonExpressions.scala:142)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject$$anonfun$eval$2$$anonfun$4.apply(jsonExpressions.scala:141)

at org.apache.spark.util.Utils$.tryWithResource(Utils.scala:2202)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject$$anonfun$eval$2.apply(jsonExpressions.scala:141)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject$$anonfun$eval$2.apply(jsonExpressions.scala:138)

at org.apache.spark.util.Utils$.tryWithResource(Utils.scala:2202)

at org.apache.spark.sql.catalyst.expressions.GetJsonObject.eval(jsonExpressions.scala:138)

at org.apache.spark.sql.catalyst.expressions.GeneratedClass$SpecificMutableProjection.apply(Unknown Source)

at org.apache.spark.sql.execution.Expand$$anonfun$doExecute$1$$anonfun$3$$anon$1.next(Expand.scala:76)

at org.apache.spark.sql.execution.Expand$$anonfun$doExecute$1$$anonfun$3$$anon$1.next(Expand.scala:62)

at org.apache.spark.sql.execution.aggregate.TungstenAggregationIterator.processInputs(TungstenAggregationIterator.scala:512)

at org.apache.spark.sql.execution.aggregate.TungstenAggregationIterator.&lt;init&gt;(TungstenAggregationIterator.scala:686)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1$$anonfun$2.apply(TungstenAggregate.scala:95)

at org.apache.spark.sql.execution.aggregate.TungstenAggregate$$anonfun$doExecute$1$$anonfun$2.apply(TungstenAggregate.scala:86)

at org.apache.spark.rdd.RDD$$anonfun$mapPartitions$1$$anonfun$apply$20.apply(RDD.scala:710)

at org.apache.spark.rdd.RDD$$anonfun$mapPartitions$1$$anonfun$apply$20.apply(RDD.scala:710)

at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:38)

at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:306)

at org.apache.spark.rdd.RDD.iterator(RDD.scala:270)

at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:38)

at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:306)

at org.apache.spark.rdd.RDD.iterator(RDD.scala:270)

at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:73)

at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:41)

at org.apache.spark.scheduler.Task.run(Task.scala:89)

at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:214)

</pre>

</div></div>

<p>Basically Spark sells me the idea, that I have character 敮 in my data. But query </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

select line from logs.raw\_client\_log where year=2016 and month=2 and day&gt;27 and line rlike <span class="code-quote">"敮"</span>

</pre>

</div></div>

<p>returns nothing.</p>

Title: [SPARK-13653] Split DiskBlockObjectWriter to separate object- and byte-based write interfaces

Description: <p>DiskBlockObjectWriter exposes methods for writing both byte streams and objects. Mixing both of these interfaces into a single class is somewhat messy and requires hacks in certain code to support it (see DummySerializerInstance). I think that we should cleanly separate these interfaces into separate classes.</p>

Title: [SPARK-13652] TransportClient.sendRpcSync returns wrong results

Description: <p>TransportClient is not thread safe and if it is called from multiple threads, the messages can't be encoded and decoded correctly. Below is my code,and it will print wrong message.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-keyword">public</span> <span class="code-keyword">static</span> void main(<span class="code-object">String</span>[] args) <span class="code-keyword">throws</span> IOException, InterruptedException {

TransportServer server = <span class="code-keyword">new</span> TransportContext(<span class="code-keyword">new</span> TransportConf(<span class="code-quote">"test"</span>,

<span class="code-keyword">new</span> MapConfigProvider(<span class="code-keyword">new</span> HashMap&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;())), <span class="code-keyword">new</span> RankHandler()).

createServer(8081, <span class="code-keyword">new</span> LinkedList&lt;TransportServerBootstrap&gt;());

TransportContext context = <span class="code-keyword">new</span> TransportContext(<span class="code-keyword">new</span> TransportConf(<span class="code-quote">"test"</span>,

<span class="code-keyword">new</span> MapConfigProvider(<span class="code-keyword">new</span> HashMap&lt;<span class="code-object">String</span>, <span class="code-object">String</span>&gt;())), <span class="code-keyword">new</span> NoOpRpcHandler(), <span class="code-keyword">true</span>);

<span class="code-keyword">final</span> TransportClientFactory clientFactory = context.createClientFactory();

List&lt;<span class="code-object">Thread</span>&gt; ts = <span class="code-keyword">new</span> ArrayList&lt;&gt;();

<span class="code-keyword">for</span> (<span class="code-object">int</span> i = 0; i &lt; 10; i++) {

ts.add(<span class="code-keyword">new</span> <span class="code-object">Thread</span>(<span class="code-keyword">new</span> <span class="code-object">Runnable</span>() {

@Override

<span class="code-keyword">public</span> void run() {

<span class="code-keyword">for</span> (<span class="code-object">int</span> j = 0; j &lt; 1000; j++) {

<span class="code-keyword">try</span> {

ByteBuf buf = Unpooled.buffer(8);

buf.writeLong((<span class="code-object">long</span>) j);

ByteBuffer byteBuffer = clientFactory.createClient(<span class="code-quote">"localhost"</span>, 8081).

sendRpcSync(buf.nioBuffer(), <span class="code-object">Long</span>.MAX\_VALUE);

<span class="code-object">long</span> response = byteBuffer.getLong();

<span class="code-keyword">if</span> (response != j) {

<span class="code-object">System</span>.err.println(<span class="code-quote">"send:"</span> + j + <span class="code-quote">",response:"</span> + response);

}

} <span class="code-keyword">catch</span> (IOException e) {

e.printStackTrace();

}

}

}

}));

ts.get(i).start();

}

<span class="code-keyword">for</span> (<span class="code-object">Thread</span> t : ts) {

t.join();

}

server.close();

}

<span class="code-keyword">public</span> class RankHandler <span class="code-keyword">extends</span> RpcHandler {

<span class="code-keyword">private</span> <span class="code-keyword">final</span> Logger logger = LoggerFactory.getLogger(RankHandler.class);

<span class="code-keyword">private</span> <span class="code-keyword">final</span> StreamManager streamManager;

<span class="code-keyword">public</span> RankHandler() {

<span class="code-keyword">this</span>.streamManager = <span class="code-keyword">new</span> OneForOneStreamManager();

}

@Override

<span class="code-keyword">public</span> void receive(TransportClient client, ByteBuffer msg, RpcResponseCallback callback) {

callback.onSuccess(msg);

}

@Override

<span class="code-keyword">public</span> StreamManager getStreamManager() {

<span class="code-keyword">return</span> streamManager;

}

}

</pre>

</div></div>

<p>it will print as below<br/>

send:221,response:222<br/>

send:233,response:234<br/>

send:312,response:313<br/>

send:358,response:359<br/>

...</p>

Title: [SPARK-13651] Generator outputs are not resolved correctly resulting in runtime error

Description: <p>Seq(("id1", "value1")).toDF("key", "value").registerTempTable("src")<br/>

sqlContext.sql("SELECT t1.\* FROM src LATERAL VIEW explode(map('key1', 100, 'key2', 200)) t1 AS key, value")</p>

<p>Running above repro results in :</p>

<p>java.lang.ClassCastException: java.lang.Integer cannot be cast to org.apache.spark.unsafe.types.UTF8String<br/>

at org.apache.spark.sql.catalyst.expressions.BaseGenericInternalRow$class.getUTF8String(rows.scala:46)<br/>

at org.apache.spark.sql.catalyst.expressions.GenericInternalRow.getUTF8String(rows.scala:221)<br/>

at org.apache.spark.sql.catalyst.expressions.GeneratedClass$SpecificUnsafeProjection.apply(generated.java:42)<br/>

at org.apache.spark.sql.execution.Generate$$anonfun$doExecute$1$$anonfun$apply$9.apply(Generate.scala:98)<br/>

at org.apache.spark.sql.execution.Generate$$anonfun$doExecute$1$$anonfun$apply$9.apply(Generate.scala:96)<br/>

at scala.collection.Iterator$$anon$11.next(Iterator.scala:370)<br/>

at scala.collection.Iterator$$anon$11.next(Iterator.scala:370)<br/>

at scala.collection.Iterator$class.foreach(Iterator.scala:742)<br/>

at scala.collection.AbstractIterator.foreach(Iterator.scala:1194)<br/>

at scala.collection.generic.Growable$class.$plus$plus$eq(Growable.scala:59)<br/>

at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:104)<br/>

at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:48)<br/>

at scala.collection.TraversableOnce$class.to(TraversableOnce.scala:308)<br/>

at scala.collection.AbstractIterator.to(Iterator.scala:1194)<br/>

at scala.collection.TraversableOnce$class.toBuffer(TraversableOnce.scala:300)<br/>

at scala.collection.AbstractIterator.toBuffer(Iterator.scala:1194)<br/>

at scala.collection.TraversableOnce$class.toArray(TraversableOnce.scala:287)<br/>

at scala.collection.AbstractIterator.toArray(Iterator.scala:1194)<br/>

at org.apache.spark.rdd.RDD$$anonfun$collect$1$$anonfun$13.apply(RDD.scala:876)<br/>

at org.apache.spark.rdd.RDD$$anonfun$collect$1$$anonfun$13.apply(RDD.scala:876)<br/>

at org.apache.spark.SparkContext$$anonfun$runJob$5.apply(SparkContext.scala:1794)<br/>

at org.apache.spark.SparkContext$$anonfun$runJob$5.apply(SparkContext.scala:1794)<br/>

at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:69)<br/>

at org.apache.spark.scheduler.Task.run(Task.scala:82)<br/>

at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:227)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)<br/>

at java.lang.Thread.run(Thread.java:745)</p>

Title: [SPARK-13650] Usage of the window() function on DStream

Description: <p>Is there some guidance of the usage of the Window() function on DStream. Here is my academic use-case for which it fails.</p>

<p>Standard word count</p>

<p> val ssc = new StreamingContext(sparkConf, Seconds(6))<br/>

val messages = KafkaUtils.createDirectStream(...)<br/>

val words = messages.map(\_.<em>2).flatMap(</em>.split(" "))<br/>

val window = words.window(Seconds(12), Seconds(6)) <br/>

window.count().print()</p>

<p>For the first batch interval it gives the count and then it hangs (inside the unionRDD)</p>

<p>I say the above use-case is academic since one can achieve similar fuctionality by using instead the more compact API<br/>

words.countByWindow(Seconds(12), Seconds(6))<br/>

which works fine. </p>

<p>Is the first approach above not the intended way of using the .window() API</p>

Title: [SPARK-13649] Move CalendarInterval out of unsafe package

Description: <p><tt>CalendarInterval</tt> is part of user facing interface as it can be returned DataFrame contents, but historically it was put in <tt>unsafe.types</tt> package for no good reason. We should probably move it out.</p>

Title: [SPARK-13648] org.apache.spark.sql.hive.client.VersionsSuite fails NoClassDefFoundError on IBM JDK

Description: <p>When running the standard Spark unit tests on the IBM Java SDK the hive VersionsSuite fail with the following error.</p>

<p>java.lang.NoClassDefFoundError: org.apache.hadoop.hive.cli.CliSessionState when creating Hive client using classpath: ..............</p>

Title: [SPARK-13647] also check if numeric value is within allowed range in \_verify\_type

Description:

Title: [SPARK-13646] QuantileDiscretizer counts dataset twice in getSampledInput

Description: <p>getSampledInput counts the dataset twice as you see here : <a href="https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/ml/feature/QuantileDiscretizer.scala#L116" class="external-link" rel="nofollow">https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/ml/feature/QuantileDiscretizer.scala#L116</a></p>

Title: [SPARK-13645] DAG Diagram not shown properly in Chrome

Description: <p>In my Chrome 49, the execution DAG diagram can't be shown properly. Only a few grey dots lays there. Thought this is what I'm supposed to see at first. </p>

<p>It works fine in Firefox, though.</p>

<p>See the attachment below.</p>

Title: [SPARK-13644] Add the source file name and line into Logger when an exception occurs in the generated code

Description: <p>This is to show a message points out the origin of a generated method when an exception occurs in the generated method at runtime.</p>

<p>An example of a message (the first line is newly added)</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

07:49:29.525 ERROR org.apache.spark.sql.catalyst.expressions.GeneratedClass$GeneratedIterator: The method GeneratedIterator.processNext() is generated <span class="code-keyword">for</span> filter at Test.scala:23

07:49:29.526 ERROR org.apache.spark.executor.Executor: Exception in task 1.0 in stage 2.0 (TID 4)

java.lang.NullPointerException:

at ...

</pre>

</div></div>

Title: [SPARK-13643] Create SparkSession interface

Description:

Title: [SPARK-13642] Properly handle signal kill of ApplicationMaster

Description: <p>Currently when running Spark on Yarn with yarn cluster mode, the default application final state is "SUCCEED", if any exception is occurred, this final state will be changed to "FAILED" and trigger the reattempt if possible. </p>

<p>This is OK in normal case, but if there's a race condition when AM received a signal (SIGTERM) and no any exception is occurred. In this situation, shutdown hook will be invoked and marked this application as finished with success, and there's no another attempt.</p>

<p>In such situation, actually from Spark's aspect this application is failed and need another attempt, but from Yarn's aspect the application is finished with success.</p>

<p>This could happened in NM failure situation, the failure of NM will send SIGTERM to AM, AM should mark this attempt as failure and rerun again, not invoke unregister.</p>

<p>So to increase the chance of this race condition, here is the reproduced code:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

val sc = ...

<span class="code-object">Thread</span>.sleep(30000L)

sc.parallelize(1 to 100).collect()

</pre>

</div></div>

<p>If the AM is failed in sleeping, there's no exception been thrown, so from Yarn's point this application is finished successfully, but from Spark's point, this application should be reattempted.</p>

<p>The log normally like this:</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>16/03/03 12:44:19 INFO ContainerManagementProtocolProxy: Opening proxy : 192.168.0.105:45454

16/03/03 12:44:21 INFO YarnClusterSchedulerBackend: Registered executor NettyRpcEndpointRef(null) (192.168.0.105:57461) with ID 2

16/03/03 12:44:21 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.0.105:57462 with 511.1 MB RAM, BlockManagerId(2, 192.168.0.105, 57462)

16/03/03 12:44:23 INFO YarnClusterSchedulerBackend: Registered executor NettyRpcEndpointRef(null) (192.168.0.105:57467) with ID 1

16/03/03 12:44:23 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.0.105:57468 with 511.1 MB RAM, BlockManagerId(1, 192.168.0.105, 57468)

16/03/03 12:44:23 INFO YarnClusterSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.8

16/03/03 12:44:23 INFO YarnClusterScheduler: YarnClusterScheduler.postStartHook done

16/03/03 12:44:39 ERROR ApplicationMaster: RECEIVED SIGNAL 15: SIGTERM

16/03/03 12:44:39 INFO SparkContext: Invoking stop() from shutdown hook

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/metrics/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/stage/kill,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/api,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/static,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/executors/threadDump/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/executors/threadDump,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/executors/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/executors,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/environment/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/environment,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/storage/rdd/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/storage/rdd,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/storage/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/storage,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/pool/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/pool,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/stage/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/stage,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/stages,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/jobs/job/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/jobs/job,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/jobs/json,null}

16/03/03 12:44:39 INFO ContextHandler: stopped o.e.j.s.ServletContextHandler{/jobs,null}

16/03/03 12:44:39 INFO SparkUI: Stopped Spark web UI at http://192.168.0.105:57452

16/03/03 12:44:39 INFO YarnAllocator: Driver requested a total number of 0 executor(s).

16/03/03 12:44:39 INFO YarnAllocator: Canceling requests for 0 executor containers

16/03/03 12:44:39 INFO YarnClusterSchedulerBackend: Shutting down all executors

16/03/03 12:44:39 WARN YarnAllocator: Expected to find pending requests, but found none.

16/03/03 12:44:39 INFO YarnClusterSchedulerBackend: Asking each executor to shut down

16/03/03 12:44:39 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices

(serviceOption=None,

services=List(),

started=false)

16/03/03 12:44:39 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!

16/03/03 12:44:39 INFO MemoryStore: MemoryStore cleared

16/03/03 12:44:39 INFO BlockManager: BlockManager stopped

16/03/03 12:44:39 INFO BlockManagerMaster: BlockManagerMaster stopped

16/03/03 12:44:39 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!

16/03/03 12:44:39 INFO SparkContext: Successfully stopped SparkContext

16/03/03 12:44:39 INFO ApplicationMaster: Final app status: SUCCEEDED, exitCode: 0, (reason: Shutdown hook called before final status was reported.)

16/03/03 12:44:39 INFO ApplicationMaster: Unregistering ApplicationMaster with SUCCEEDED (diag message: Shutdown hook called before final status was reported.)

16/03/03 12:44:39 INFO AMRMClientImpl: Waiting for application to be successfully unregistered.

16/03/03 12:44:39 INFO ApplicationMaster: Deleting staging directory .sparkStaging/application\_1456975940304\_0004

16/03/03 12:44:40 INFO ShutdownHookManager: Shutdown hook called

</pre>

</div></div>

<p>So basically, I think only after the finish of user class, we could mark this application as "SUCCESS", otherwise, especially in the signal stopped scenario, it would be better to mark as failed and try again (except explicitly KILL command by yarn).</p>

Title: [SPARK-13641] getModelFeatures of ml.api.r.SparkRWrapper cannot (always) reveal the original column names

Description: <p>getModelFeatures of ml.api.r.SparkRWrapper cannot (always) reveal the original column names. Let's take the HouseVotes84 data set as an example:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-keyword">case</span> m: XXXModel =&gt;

val attrs = AttributeGroup.fromStructField(

m.summary.predictions.schema(m.summary.featuresCol))

attrs.attributes.get.map(\_.name.get)

</pre>

</div></div>

<p>The code above gets features' names from the features column. Usually, the features column is generated by RFormula. The latter has a VectorAssembler in it, which leads the output attributes not equal with the original ones.</p>

<p>E.g., we want to learn the HouseVotes84's features' name "V1, V2, ..., V16". But with RFormula, we can only get "V1\_n, V2\_y, ..., V16\_y" because <a href="https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/ml/feature/VectorAssembler.scala#L75" class="external-link" rel="nofollow">the transform function of VectorAssembler</a> adds salts of the column names.</p>