

ASSIGNMENT

Task1: Spark-Hive Integration

Sol: Step1: copy the hive-site.xml file from \$HIVE-HOME conf folder to \$SPARK-HOME conf folder

➔ `cp /home/acadgild/install/hive/apache-hive-2.3.2-bin/conf/hive-site.xml /home/acadgild/install/spark/spark-2.2.1-bin-hadoop2.7/conf`

Step2: edit the hive-site.xml file

➔ `vi hive-site.xml`

➔ enter the following property tags in configuration


➔ `<property>`

`<name>hive.metastore.uris</name>`

`<value>thrift://localhost:9083</value>`

`<description>password for connecting to mysql server</description>`

`</property>`

 acadgild@localhost:~/install/spark/spark-2.2.1-bin-hadoop2.7/conf

```
</property>

<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>Root@123</value>
</property>

<property>
  <name>datanucleus.autoCreateSchema</name>
  <value>true</value>
</property>

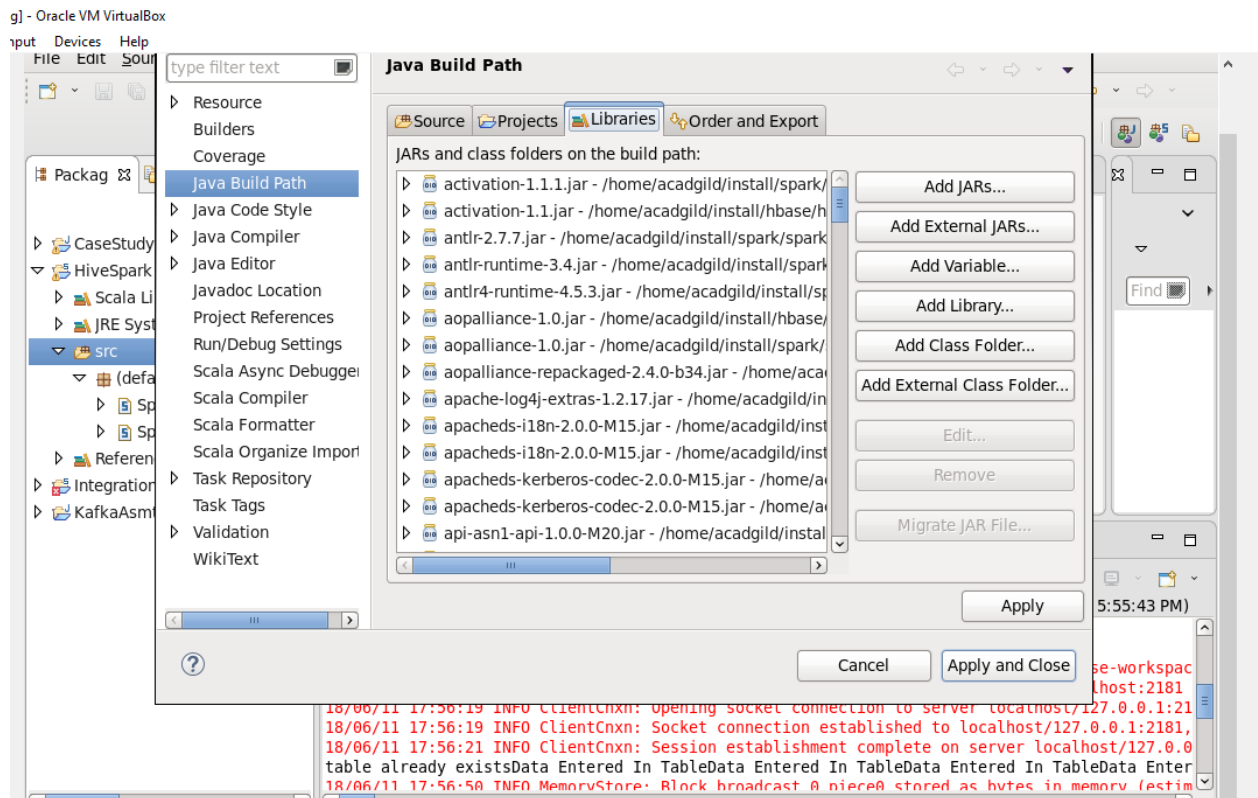
<property>
  <name>datanucleus.fixedDatastore</name>
  <value>true</value>
</property>

<property>
  <name>datanucleus.autoCreateTables</name>
  <value>True</value>
</property>

<property>
  <name>hive.metastore.uris</name>
  <value>thrift://localhost:9083</value>
  <description>password for connecting to mysql server</description>
</property>
```

Step3: copy the code to eclipse IDE (SparkHiveTest.scala)

Step4: Add the jar files from spark and hbase folders



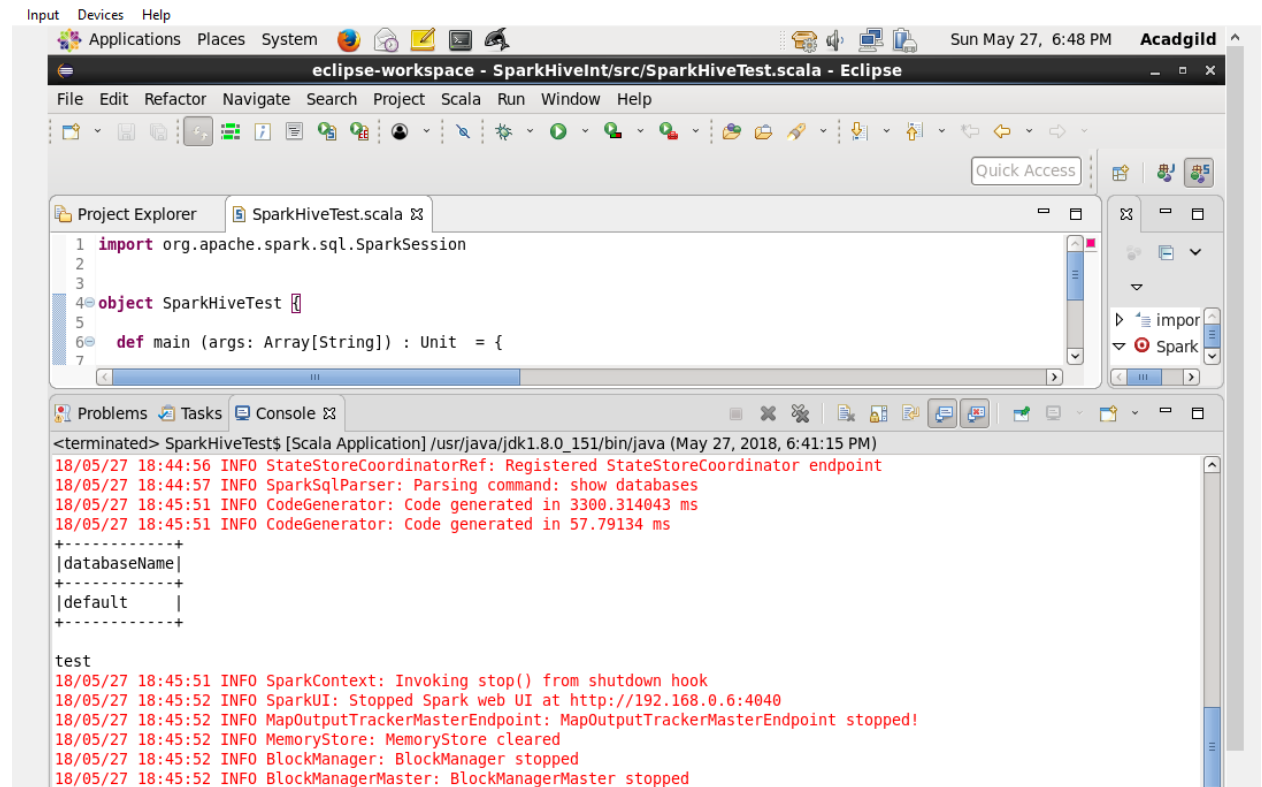
Step5: start hive server2

➔ Hive --service metastore

```
acadgild@localhost:~$ hive --service metastore
2018-06-11 17:49:45: Starting Hive Metastore Server
/home/acadgild/install/hive/apache-hive-2.3.2-bin/bin/ext/metastore.sh: line 29:
export: `Dproc_metastore` -Dlog4j.configurationFile=hive-log4j2.properties -Djava.util.logging.config.file=/home/acadgild/install/hive/apache-hive-2.3.2-bin/conf/parquet-logging.properties ': not a valid identifier
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.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.apache.logging.slf4j.Log4jLoggerFactory]
2018-06-11T17:49:54,399 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Found configuration file file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/conf/hive-site.xml
2018-06-11T17:50:01,387 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - STARTUP_MSG:
/*****
STARTUP_MSG: Starting HiveMetaStore
STARTUP_MSG: host = localhost/127.0.0.1
STARTUP_MSG: args = []
STARTUP_MSG: version = 2.3.2
STARTUP_MSG: classpath = /home/acadgild/install/hive/apache-hive-2.3.2-bin/conf:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/accumulo-core-1.6.0.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/accumulo-fate-1.6.0.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/accumulo-start-1.6.0.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/accumulo-trace-1.6.0.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/activation-1.1.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/aether-api-0.9.0.M2.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/aether-connector-file-0.9.0.M2.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/aether-connector-okhttp-0.9.0.M2.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/aether-impl-0.9.0.M2.jar:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/aether-spi-0.9.0.M2.jar
```

Step6: Run the program

unning] - Oracle VM VirtualBox



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, and Help. The Project Explorer on the left shows the file SparkHiveTest.scala. The main editor displays the following Scala code:

```
1 import org.apache.spark.sql.SparkSession
2
3
4 object SparkHiveTest {
5
6   def main (args: Array[String]) : Unit = {
7
```

The Console window at the bottom shows the execution output for the SparkHiveTest application. The output includes log messages from the Spark framework and the results of a SQL query:

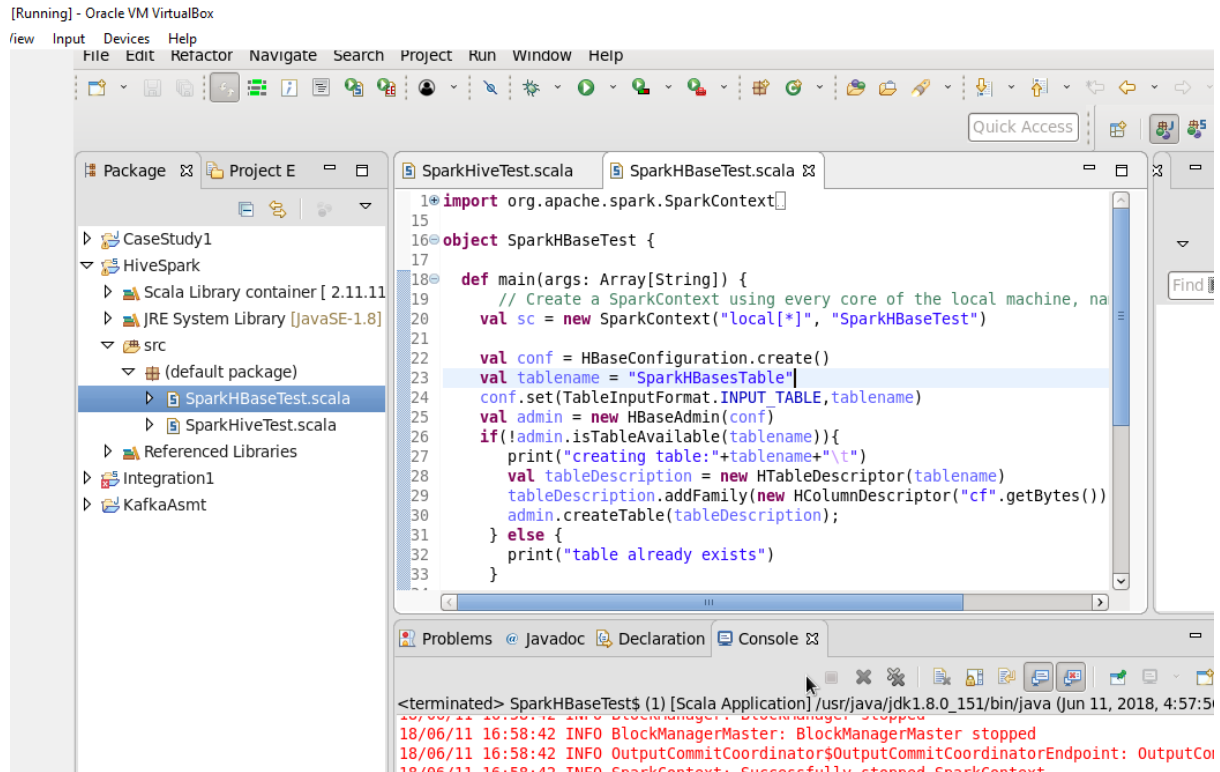
```
<terminated> SparkHiveTest$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (May 27, 2018, 6:41:15 PM)
18/05/27 18:44:56 INFO StateStoreCoordinatorRef: Registered StateStoreCoordinator endpoint
18/05/27 18:44:57 INFO SparkSqlParser: Parsing command: show databases
18/05/27 18:45:51 INFO CodeGenerator: Code generated in 3300.314043 ms
18/05/27 18:45:51 INFO CodeGenerator: Code generated in 57.79134 ms
+-----+
|databaseName|
+-----+
+-----+
|default     |
+-----+

test
18/05/27 18:45:51 INFO SparkContext: Invoking stop() from shutdown hook
18/05/27 18:45:52 INFO SparkUI: Stopped Spark web UI at http://192.168.0.6:4040
18/05/27 18:45:52 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
18/05/27 18:45:52 INFO MemoryStore: MemoryStore cleared
18/05/27 18:45:52 INFO BlockManager: BlockManager stopped
18/05/27 18:45:52 INFO BlockManagerMaster: BlockManagerMaster stopped
```

Task2: Spark-Hbase Integration

Sol: Step1: Add the code to same scala project.

➔ SparkHbaseTest.scala // program file



The screenshot shows an IDE window titled "[Running] - Oracle VM VirtualBox". The main editor displays the file "SparkHBaseTest.scala" with the following code:

```
1 import org.apache.spark.SparkContext
15
16 object SparkHBaseTest {
17
18 def main(args: Array[String]) {
19     // Create a SparkContext using every core of the local machine, na
20     val sc = new SparkContext("local[*]", "SparkHBaseTest")
21
22     val conf = HBaseConfiguration.create()
23     val tablename = "SparkHBasesTable"
24     conf.set(TableInputFormat.INPUT_TABLE, tablename)
25     val admin = new HBaseAdmin(conf)
26     if(!admin.isTableAvailable(tablename)){
27         print("creating table:"+tablename+"\n")
28         val tableDescription = new HTableDescriptor(tablename)
29         tableDescription.addFamily(new HColumnDescriptor("cf".getBytes()))
30         admin.createTable(tableDescription);
31     } else {
32         print("table already exists")
33     }
34 }
```

The bottom panel shows the "Console" tab with the following output:

```
<terminated> SparkHBaseTest$ (1) [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 11, 2018, 4:57:51)
18/06/11 16:58:42 INFO BlockManagerMaster: BlockManagerMaster stopped
18/06/11 16:58:42 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped
18/06/11 16:58:42 INFO SparkContext: Successfully stopped SparkContext
```

Step2: Run the program.

Step3: In Putty -> cd install/hbase/hbase-1.2.6/bin

./start-hbase.sh // start hbase

hbase shell // start hbase shell

scan "SparkHBasesTable"

```
hbase(main):001:0> scan "SparkHBasesTable"
ROW COLUMN+CELL
 row1 column=cf:column1, timestamp=1528716512101, value=value1
 row10 column=cf:column1, timestamp=1528716512269, value=value10
 row2 column=cf:column1, timestamp=1528716512180, value=value2
 row3 column=cf:column1, timestamp=1528716512194, value=value3
 row4 column=cf:column1, timestamp=1528716512204, value=value4
 row5 column=cf:column1, timestamp=1528716512209, value=value5
 row6 column=cf:column1, timestamp=1528716512214, value=value6
 row7 column=cf:column1, timestamp=1528716512227, value=value7
 row8 column=cf:column1, timestamp=1528716512242, value=value8
 row9 column=cf:column1, timestamp=1528716512259, value=value9
10 row(s) in 1.3850 seconds
hbase(main):002:0>
```