

Phoenix对cdh支持有限，在早期，官方没有提供cdh适配的版本，当前官方也只提供了cdh-5.11.2的适配版本，经测试，这个版本在cdh5.6和cdh5.7均不能运行（其他版本没有测试），如果需要适配其他cdh版本，需要修改Phoenix的源码并编译打包。

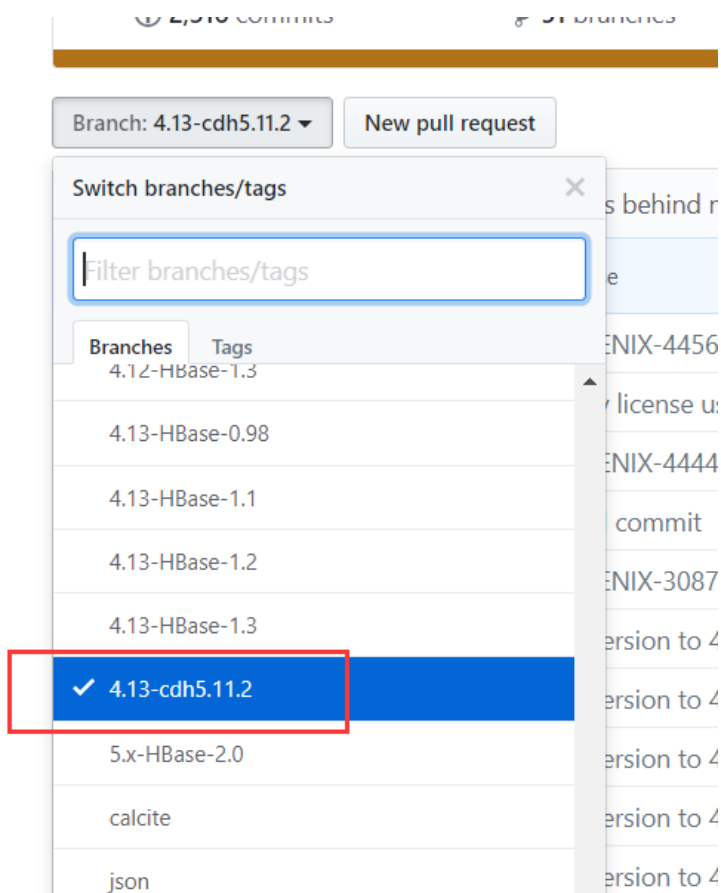
一、修改源码

使用官方4.13.2-cdh5.11.2分支，修改为适配cdh5.7.0的版本

1. 下载源码

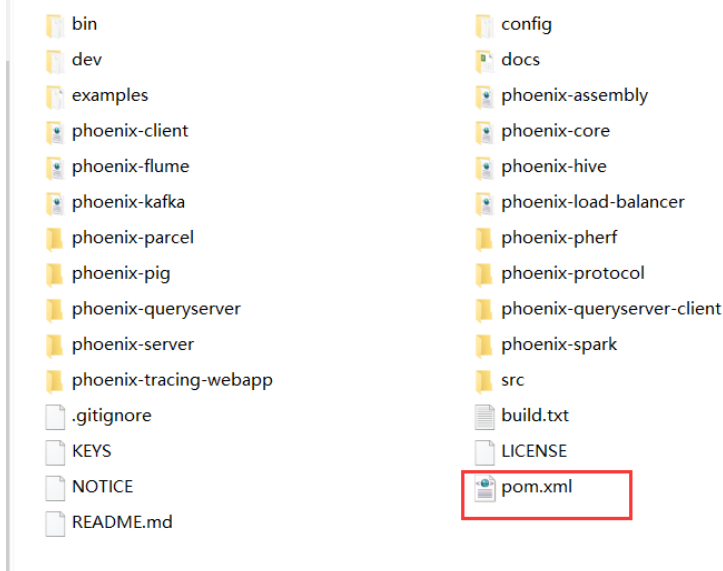
源码地址：<https://github.com/apache/phoenix>

选择版本：选择分支4.13.2-cdh5.11.2



2. 修改pom文件

2.1 修改root路径下的pom文件



修改两个地方：

修改1：将phoenix项目的版本统一为：4.13.2-cdh5.7.0

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.apache.phoenix</groupId>
  <artifactId>phoenix</artifactId>
  <version>4.13.2-cdh5.11.2</version>
  <packaging>pom</packaging>
  <name>Apache Phoenix</name>
  <description>A SQL layer over HBase</description>

  <licenses>
    <license>
      <name>The Apache Software License, Version 2.0</name>
      <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
      <distribution>repo</distribution>
      <comments/>
    </license>
  </licenses>
</project>
```

修改为： 4.13.2-cdh5.7.0

修改2：

```
<parent>
  <groupId>com.cloudera.cdh</groupId>
  <artifactId>cdh-root</artifactId>
  <version>5.11.2</version>
</parent>
```

修改为： 5.7.0

2.2 修改其他子项目的pom文件

将phoenix项目的版本统一为：4.13.2-cdh5.7.0

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:x
  xsi:schemaLocation="http://maven.apache.org/POM/4.
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.apache.phoenix</groupId>
    <artifactId>phoenix</artifactId>
    <version>4.13.2-cdh5.11.2</version>
  </parent>
  <artifactId>phoenix-flume</artifactId>
  <name>Phoenix - Flume</name>

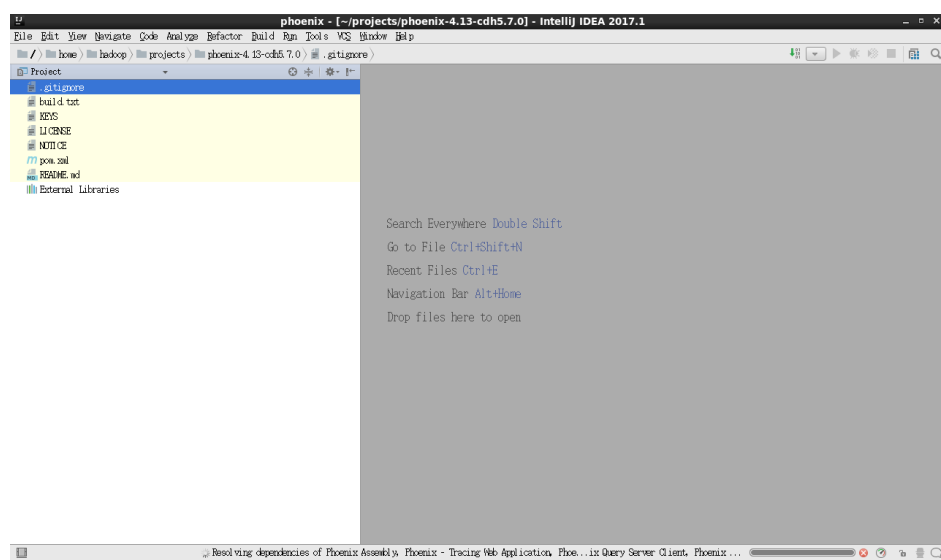
```

4.13.2-cdh5.7.0

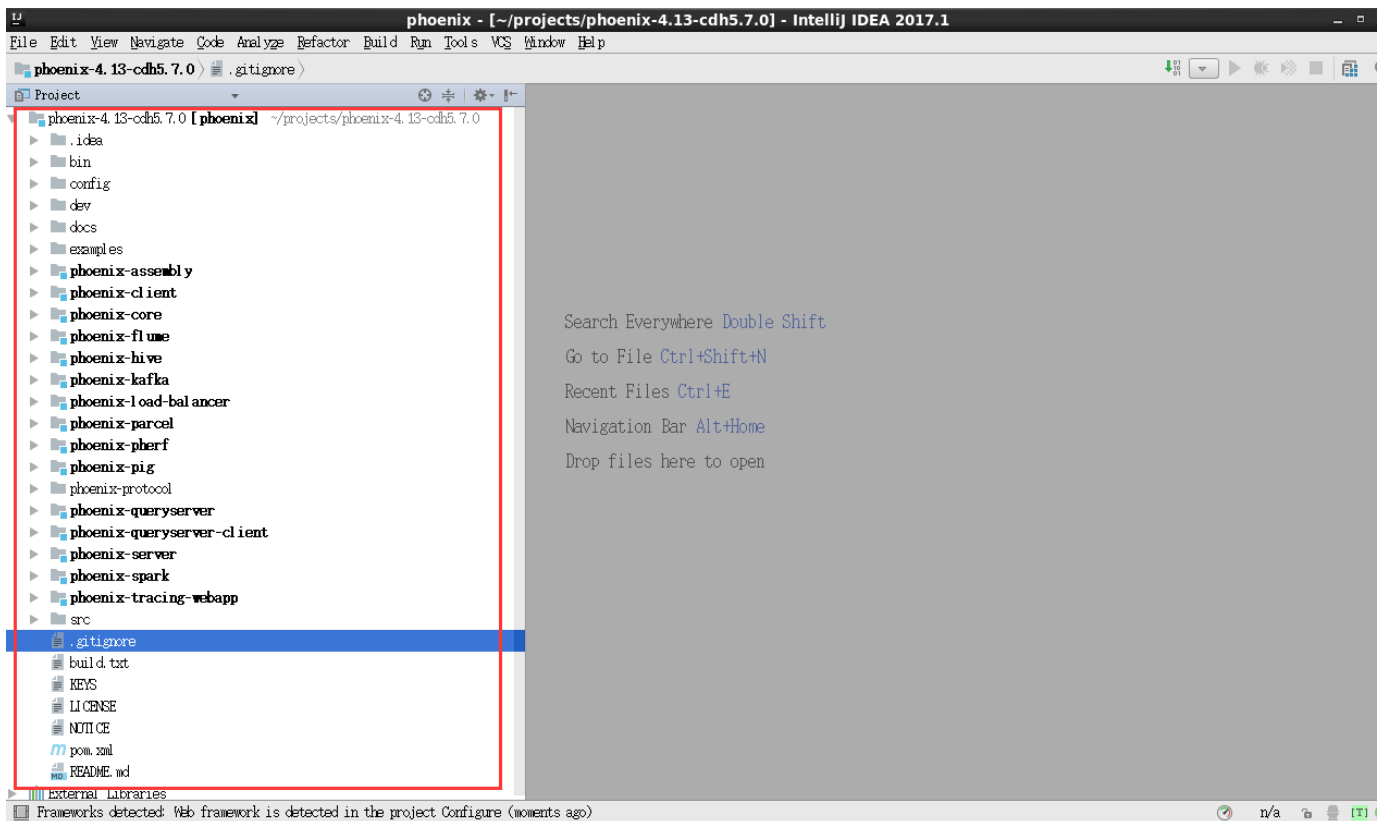
3. 修改源码

3.1 将使用idea打开上面修改过的项目

修改项目目录名称为： phoenix-4.13-cdh5.7.0 ， 然后使用idea导入项目：



导入后，开始下载pom相关的依赖，等待很长时间后，可以看到项目的目录：



3.2 打包

进入到项目的目录，使用mvn package -DskipTests打包项目

```
[hadoop@spark1234 projects]$ cd phoenix-4.13-cdh5.7.0
[hadoop@spark1234 phoenix-4.13-cdh5.7.0]$
[hadoop@spark1234 phoenix-4.13-cdh5.7.0]$ mvn package -DskipTests
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for org.apache.phoenix:phoenix-core:jar:4.13.2-cdh5.7.0
[WARNING] Reporting configuration should be done in <reporting> section, not in maven-site-plugin <configuration> as reportPlugins parameter. @ org.apache.phoenix:phoenix-core:[unknown-version], /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/pom.xml, line 65, column 23
[WARNING]
```

3.3 第一次打包碰到的错误

3.3.1 第一次打包碰到4个错误

执行失败，失败的截图如下：

(1)、第一个错误类：MetricsIndexerSourceImpl

源码所在的路径：

/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/hbase/index/metrics/MetricsIndexerSourceImpl.java

```

[INFO] Finished at: 2018-02-20T07:10:36+08:00
[INFO] Final Memory: 65M/519M
[INFO] -----
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.0:compile (default-compile) on project phoenix-core: Compilation failure:
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/hbase/index/metrics/MetricsIndexerSourceImpl.java:[21,38] 找不到符号
[ERROR] 符号: 类 MutableFastCounter
[ERROR] 位置: 程序包 org.apache.hadoop.metrics2.lib
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/hbase/index/metrics/MetricsIndexerSourceImpl.java:[29,19] 找不到符号
[ERROR] 符号: 类 MutableFastCounter
[ERROR] 位置: 类 org.apache.phoenix.hbase.index.metrics.MetricsIndexerSourceImpl
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/hbase/index/metrics/MetricsIndexerSourceImpl.java:[31,19] 找不到符号
[ERROR] 符号: 类 MutableFastCounter
[ERROR] 位置: 类 org.apache.phoenix.hbase.index.metrics.MetricsIndexerSourceImpl
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/hbase/index/metrics/MetricsIndexerSourceImpl.java:[33,19] 找不到符号
[ERROR] 符号: 类 MutableFastCounter
[ERROR] 位置: 类 org.apache.phoenix.hbase.index.metrics.MetricsIndexerSourceImpl

```

(2)、第二个错误类：DelegateHTable

```

[ERROR] 符号: 类 MutableFastCounter
[ERROR] 位置: 类 org.apache.phoenix.hbase.index.metrics.MetricsIndexerSourceImpl
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/execute/DelegateHTable.java:[282,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/execute/DelegateHTable.java:[284,17] 找不到符号
[ERROR] 符号: 方法 setRpcTimeout(int)
[ERROR] 位置: 类型为org.apache.hadoop.hbase.client.HTableInterface的变量 delegate
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/execute/DelegateHTable.java:[287,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/execute/DelegateHTable.java:[289,24] 找不到符号
[ERROR] 符号: 方法 getRpcTimeout()
[ERROR] 位置: 类型为org.apache.hadoop.hbase.client.HTableInterface的变量 delegate
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/execute/DelegateHTable.java:[292,5] 方法不会覆盖或实现超类型的方法

```

源码所在路径：

/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/execute/DelegateHTable.java

(3)、第三个错误类：TephraTransactionTable.java

```

[ERROR] 符号: 方法 getOperationTimeout()
[ERROR] 位置: 类型为org.apache.hadoop.hbase.client.HTableInterface的变量 delegate
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/TephraTransactionTable.java:[331,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/TephraTransactionTable.java:[336,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/TephraTransactionTable.java:[341,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/TephraTransactionTable.java:[346,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/OmidTransactionTable.java:[346,5] 方法不会覆盖或实现超类型的方法

```

源码所在路径：/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/TephraTransactionTable.java

(4)、第四个错误类：OmidTransactionTable

```

[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/TephraTr
ansactionTable.java:[346,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/OmidTran
sactionTable.java:[340,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/OmidTran
sactionTable.java:[346,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/OmidTran
sactionTable.java:[352,5] 方法不会覆盖或实现超类型的方法
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/OmidTran
sactionTable.java:[358,5] 方法不会覆盖或实现超类型的方法
[ERROR] -> [Help 1]
[ERROR]

```

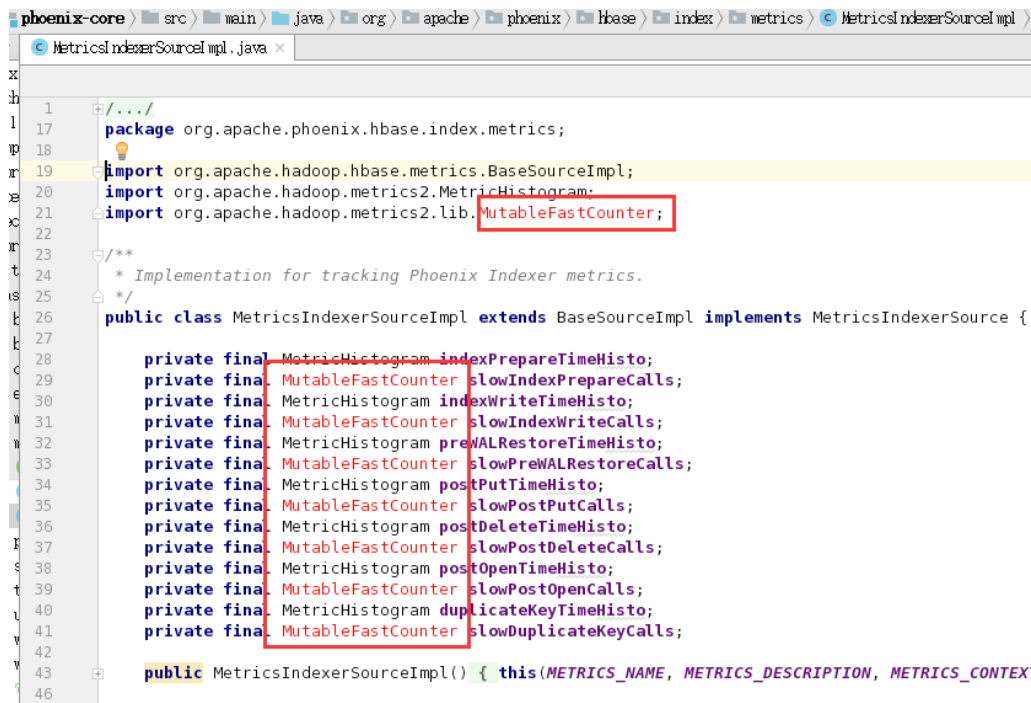
源码所在路径：/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/OmidTransactionTable.java

3.3.2 修改源码

(1)、第一个错误

使用idea打开源码所在的位置：/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/hbase/index/metrics/MetricsIndexerSourceImpl.java

a



```

package org.apache.phoenix.hbase.index.metrics;

import org.apache.hadoop.hbase.metrics.BaseSourceImpl;
import org.apache.hadoop.metrics2.MetricHistogram;
import org.apache.hadoop.metrics2.lib.MutableFastCounter;

/**
 * Implementation for tracking Phoenix Indexer metrics.
 */
public class MetricsIndexerSourceImpl extends BaseSourceImpl implements MetricsIndexerSource {

    private final MetricHistogram indexPrepareTimeHisto;
    private final MutableFastCounter slowIndexPrepareCalls;
    private final MetricHistogram indexWriteTimeHisto;
    private final MutableFastCounter slowIndexWriteCalls;
    private final MetricHistogram preWALRestoreTimeHisto;
    private final MutableFastCounter slowPreWALRestoreCalls;
    private final MetricHistogram postPutTimeHisto;
    private final MutableFastCounter slowPostPutCalls;
    private final MetricHistogram postDeleteTimeHisto;
    private final MutableFastCounter slowPostDeleteCalls;
    private final MetricHistogram postOpenTimeHisto;
    private final MutableFastCounter slowPostOpenCalls;
    private final MetricHistogram duplicateKeyTimeHisto;
    private final MutableFastCounter slowDuplicateKeyCalls;

    public MetricsIndexerSourceImpl() { this(METRICS_NAME, METRICS_DESCRIPTION, METRICS_CONTEXT); }
}

```

将代码中所有的MutableFastCounter修改为MutableFastCounter。

修改后，不再报错：

```
phoenix-core > src > main > java > org > apache > phoenix > hbase > index > metrics > MetricsIndexerSourceImpl >
MetricsIndexerSourceImpl.java x
MetricsIndexerSourceImpl
19 import org.apache.hadoop.hbase.metrics.BaseSourceImpl;
20 import org.apache.hadoop.metrics2.MetricHistogram;
21 import org.apache.hadoop.metrics2.lib.MutableCounter;
22
23 /**
24  * Implementation for tracking Phoenix Indexer metrics.
25  */
26 public class MetricsIndexerSourceImpl extends BaseSourceImpl implements MetricsIndexerSource {
27
28     private final MetricHistogram indexPrepareTimeHisto;
29     private final MutableCounter slowIndexPrepareCalls;
30     private final MetricHistogram indexWriteTimeHisto;
31     private final MutableCounter slowIndexWriteCalls;
32     private final MetricHistogram preWALRestoreTimeHisto;
33     private final MutableCounter slowPreWALRestoreCalls;
34     private final MetricHistogram postPutTimeHisto;
35     private final MutableCounter slowPostPutCalls;
36     private final MetricHistogram postDeleteTimeHisto;
37     private final MutableCounter slowPostDeleteCalls;
38     private final MetricHistogram postOpenTimeHisto;
39     private final MutableCounter slowPostOpenCalls;
40     private final MetricHistogram duplicateKeyTimeHisto;
41     private final MutableCounter slowDuplicateKeyCalls;
42
43     public MetricsIndexerSourceImpl() { this(METRICS_NAME, METRICS_DESCRIPTION, METRICS_CONTEXT, METRICS_J
44
45     public MetricsIndexerSourceImpl(String metricsName, String metricsDescription,
46         String metricsContext, String metricsJmxContext) {
47         super(metricsName, metricsDescription, metricsContext, metricsJmxContext);
48
49         indexPrepareTimeHisto = getMetricsRegistry().newHistogram(INDEX_PREPARE_TIME, INDEX_PREPARE_TIME_D
50         slowIndexPrepareCalls = getMetricsRegistry().newCounter(SLOW_INDEX_PREPARE, SLOW_INDEX_PREPARE DES
```

(2)、第二个错误

使用idea打开源码所在的位置：/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/execute/DelegateHTable.java

```
exception
execute
  AggregatePlan
  BaseQueryPlan
  ClientAggregatePlan
  ClientProcessingPlan
  ClientScanPlan
  CommitException
  CorrelatePlan
  CursorFetchPlan
  DelegateHTable
  DelegateQueryPlan
  DescVarLengthFastByteCompa
  HashJoinPlan
  LiteralResultIterationPlan
  MutationState
  RuntimeContext
  RuntimeContextImpl
  ScanPlan
  SortMergeJoinPlan
  TupleProjectionPlan
  TupleProjector
  UnionPlan
279 return delegate.checkAndMutate(row, family, qualifier, compareOp, val
280 }
281
282 @Override
283 public void setRpcTimeout(int i) {
284     delegate.setRpcTimeout(i);
285 }
286 Cannot resolve method 'setRpcTimeout(int)'
287
288 @Override
289 public int getRpcTimeout() {
290     return delegate.getRpcTimeout();
291 }
292
293 @Override
294 public void setOperationTimeout(int i) {
295     delegate.setOperationTimeout(i);
296 }
297
298 @Override
299 public int getOperationTimeout() {
300     return delegate.getOperationTimeout();
301 }
302
303 @Override
304 public boolean[] existsAll(List<Get> gets) throws IOException {
305     return delegate.existsAll(gets);
306 }
```

其中delegate的变量声明为：

protected final HTableInterface delegate;

由于delegate没有setRpcTimeout、getRpcTimeout、setOperationTimeout和getOperationTimeout四个方法，所以报错，修改这段代码如下：


```

1  public void setRpcTimeout(int i) {
2      delegate.getConfiguration().set(HConstants.HBASE_RPC_TIMEOUT_KEY,
String.valueOf(i));
3  }
4
5  public int getRpcTimeout() {
6      return
Integer.parseInt(delegate.getConfiguration().get(HConstants.HBASE_RPC_TIMEOUT_
KEY));
7  }
8
9  public void setOperationTimeout(int i) {
10
delegate.getConfiguration().set(HConstants.HBASE_CLIENT_OPERATION_TIMEOUT,
String.valueOf(i));
11  }
12
13  public int getOperationTimeout() {
14      return
Integer.parseInt(delegate.getConfiguration().get(HConstants.HBASE_CLIENT_OPERA
TION_TIMEOUT));
15  }
16

```

(3)、第三个错误、第四个错误

/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-
core/src/main/java/org/apache/phoenix/transaction/TephraTransactionTable.java



The screenshot shows a code editor with the following code snippet:

```

326     CompareOp compareOp, byte[] value, RowMutations mutation)
327     throws IOException {
328         return transactionAwareHTable.checkAndMutate(row, family, qualifier, compareOp, value, r
329     }
330
331     @Override
332     public void setOperationTimeout(int i) {
333         // transactionAwareHTable.setOperationTimeout(i);
334     }
335
336     @Override
337     public int getOperationTimeout() { return 0; //transactionAwareHTable.getOperationTimeout();
338
339
340
341     @Override
342     public void setRpcTimeout(int i) {
343         // transactionAwareHTable.setRpcTimeout(i);
344     }
345
346     @Override
347     public int getRpcTimeout() { return 0; //transactionAwareHTable.getRpcTimeout(); }
348
349
350
351

```

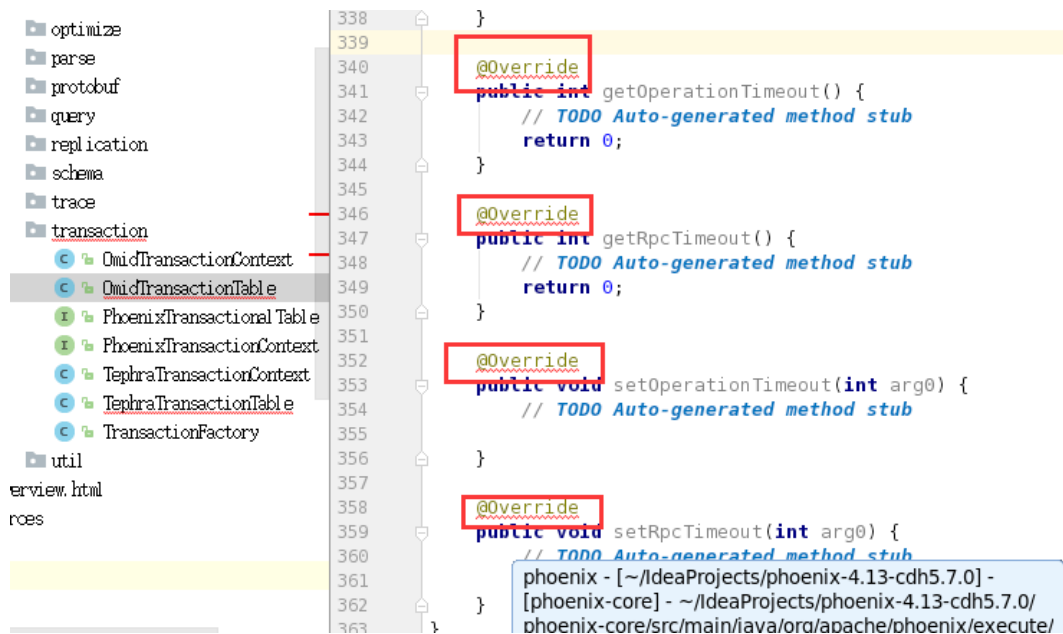
On the left sidebar, the following items are listed:

- transactionContext
- transactionTable
- TransactionalTable
- TransactionContext
- transactionContext
- transactionTable
- transactionFactory

将四个@Override给去掉即可。

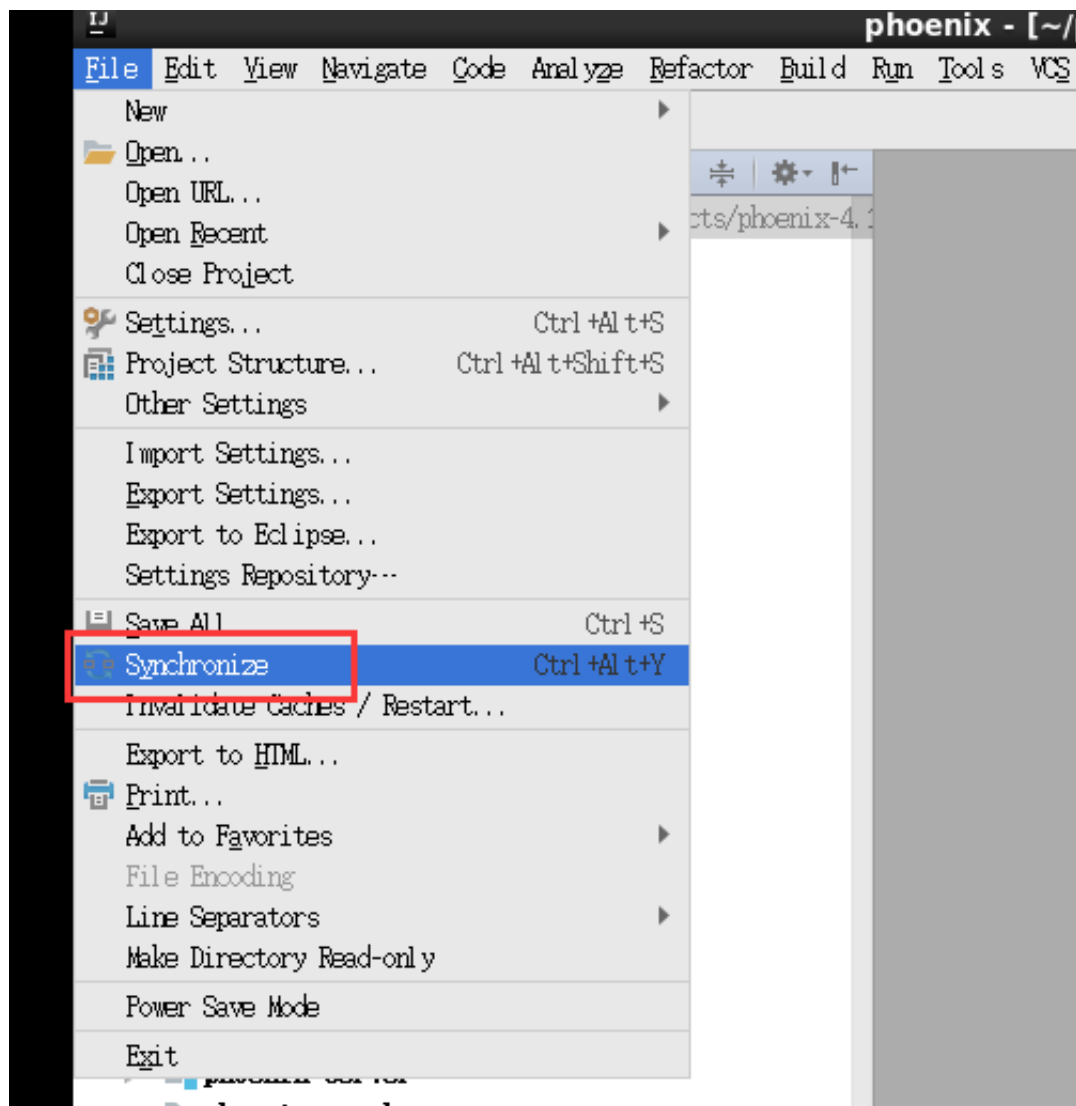
(4)、第四个错误

/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-core/src/main/java/org/apache/phoenix/transaction/OmidTransactionTable.java



将四个@Override给去掉即可。

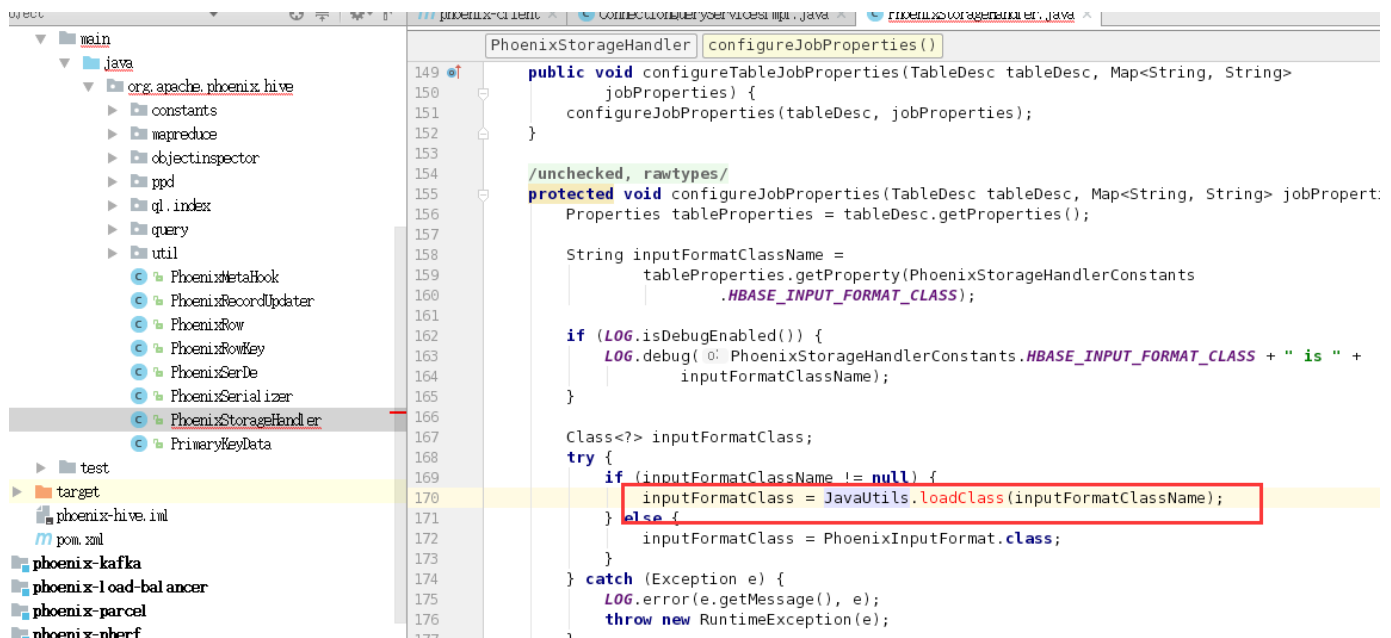
修改完后 同步到磁盘，即可使用mvn继续打包了：



3.4 继续打包，第二次错误

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.0:compile (default-compile) on project phoenix-hive: Compilation failure
[ERROR] /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-hive/src/main/java/org/apache/phoenix/hive/PhoenixStorageHandler.java:[170,45] 找不到符号
[ERROR] 符号: 方法 loadClass(java.lang.String)
[ERROR] 位置: 类 org.apache.hadoop.hive.common.JavaUtils
[ERROR] -> [Help 1]
```

使用idea定位到源码所在位置：/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-hive/src/main/java/org/apache/phoenix/hive/PhoenixStorageHandler.java



将错误行inputFormatClass = JavaUtils.loadClass(inputFormatClassName); 修改为：
inputFormatClass = JavaUtils.getClassLoader().loadClass(inputFormatClassName);
修改源码后，同步到磁盘，继续打包。

后面没有错误了，截一张打包完成后的界面：

```
[INFO] --- maven-assembly-plugin:2.5.2:single (prepare-parcel) @ phoenix-parcel ---
[INFO] Reading assembly descriptor: src/build/parcel.xml
[WARNING] Cannot include project artifact: org.apache.phoenix:phoenix-parcel:pom:4.13.2-cdh5.7.0; it doesn't have an associated file or directory.
[INFO] Building tar: /home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-parcel/target/APACHE_PHOENIX-4.13.2-cdh5.7.0.p0.0.parcel.tar
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] Apache Phoenix ..... SUCCESS [ 11.739 s]
[INFO] Phoenix Core ..... SUCCESS [ 01:06 min]
[INFO] Phoenix - Flume ..... SUCCESS [ 2.866 s]
[INFO] Phoenix - Kafka ..... SUCCESS [ 7.740 s]
[INFO] Phoenix - Pig ..... SUCCESS [ 01:11 min]
[INFO] Phoenix Query Server Client ..... SUCCESS [ 17.712 s]
[INFO] Phoenix Query Server ..... SUCCESS [ 6.413 s]
[INFO] Phoenix - Pherrf ..... SUCCESS [ 5.990 s]
[INFO] Phoenix - Spark ..... SUCCESS [ 11.535 s]
[INFO] Phoenix - Hive ..... SUCCESS [ 38.538 s]
[INFO] Phoenix Client ..... SUCCESS [ 01:04 min]
[INFO] Phoenix Server ..... SUCCESS [ 7.039 s]
[INFO] Phoenix Assembly ..... SUCCESS [ 54.865 s]
[INFO] Phoenix - Tracing Web Application ..... SUCCESS [ 8.845 s]
[INFO] Phoenix Load Balancer ..... SUCCESS [ 1.654 s]
[INFO] Phoenix Parcels for CDH ..... SUCCESS [ 10.782 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 06:30 min
[INFO] Finished at: 2018-02-20T08:04:33+08:00
[INFO] Final Memory: 149M/784M
[INFO] -----
[hadoop@spark1234 phoenix-4.13-cdh5.7.0]$
```

打包完成后，在phoenix-assembly的target目录下面生成需要的压缩包：

```
[hadoop@spark1234 phoenix-assembly]$ cd target/
[hadoop@spark1234 target]$ ls
archive-tmp maven-shared-archive-resources phoenix-4.13.2-cdh5.7.0-source.tar.gz phoenix-4.13.2-cdh5.7.0.tar.gz
[hadoop@spark1234 target]$
```

在idea控制台loginfo如下信息，其实不是报错，为了防止误解，修改源码：

```
throw (new Builder(SQLExceptionCode.CANNOT_ESTABLISH_CONNECTION)).setRootCause(var2).build().buildException();
}
}
}

Run SparkSavePhoenix500
18/02/20 06:39:09 INFO zookeeper.ZooKeeper: Initiating client connection, connectString=spark1234:12181 sessionTimeout=90000 watcher=hconnection-0x1ccd84fd0x0, quorum=spark1234:12181, baseZNode=/hbase
18/02/20 06:39:09 INFO zookeeper.ClientCnxn: Opening socket connection to server spark1234/192.168.1.22:12181. Will not attempt to authenticate using SASL (unknown error)
18/02/20 06:39:10 INFO zookeeper.ClientCnxn: Socket connection established, initiating session, client: /192.168.1.22:49038, server: spark1234/192.168.1.22:12181
18/02/20 06:39:10 INFO zookeeper.ClientCnxn: Session establishment complete on server spark1234/192.168.1.22:12181, sessionId = 0x161ae2377f00058, negotiated timeout = 40000
18/02/20 06:39:17 INFO query.ConnectionQueryServicesImpl: HConnection established. Stacktrace for informational purposes: hconnection-0x1ccd84fd java.lang.Thread.getStackTrace(Thread.java:1559)
org.apache.phoenix.util.LogUtil.getCallerStackTrace(LogUtil.java:55)
org.apache.phoenix.query.ConnectionQueryServicesImpl.openConnection(ConnectionQueryServicesImpl.java:410)
org.apache.phoenix.query.ConnectionQueryServicesImpl.access$400(ConnectionQueryServicesImpl.java:256)
org.apache.phoenix.query.ConnectionQueryServicesImpl$12.call(ConnectionQueryServicesImpl.java:2409)
org.apache.phoenix.query.ConnectionQueryServicesImpl$12.call(ConnectionQueryServicesImpl.java:2384)
org.apache.phoenix.util.PhoenixContextExecutor.call(PhoenixContextExecutor.java:76)
org.apache.phoenix.query.ConnectionQueryServicesImpl.init(ConnectionQueryServicesImpl.java:2384)
org.apache.phoenix.jdbc.PhoenixDriver.getConnectionQueryServices(PhoenixDriver.java:255)
org.apache.phoenix.jdbc.PhoenixEmbeddedDriver.createConnection(PhoenixEmbeddedDriver.java:150)
org.apache.phoenix.jdbc.PhoenixDriver.connect(PhoenixDriver.java:221)
java.sql.DriverManager.getConnection(DriverManager.java:664)
java.sql.DriverManager.getConnection(DriverManager.java:208)
org.apache.phoenix.mapreduce.util.ConnectionUtil.getConnection(ConnectionUtil.java:98)
org.apache.phoenix.mapreduce.util.ConnectionUtil.getOutputConnection(ConnectionUtil.java:82)
org.apache.phoenix.mapreduce.util.ConnectionUtil.getOutputConnection(ConnectionUtil.java:78)
org.apache.phoenix.mapreduce.util.PhoenixConfigurationUtil.getUpsertColumnMetadataList(PhoenixConfigurationUtil.java:285)
org.apache.phoenix.spark.ProductRDDFunctions$$anonfun$1.apply(ProductRDDFunctions.scala:41)
org.apache.phoenix.spark.ProductRDDFunctions$$anonfun$1.apply(ProductRDDFunctions.scala:37)
org.apache.spark.rdd.RDD$$anonfun$mapPartitions$1$anonfun$5$20.apply(RDD.scala:710)
```

定位到源码：

/home/hadoop/projects/phoenix-4.13-cdh5.7.0/phoenix-
core/src/main/java/org/apache/phoenix/query/ConnectionQueryServicesImpl.java

```
queryServices.transactionsEnabled,
QueryServicesOptions.DEFAULT_TRANSACTIONS_ENABLED);
this.connection = HBaseFactoryProvider.getHConnectionFactory().createConnection(this.config);
GLOBAL_HCONNECTIONS_COUNTER.increment();
logger.info("HConnection established. Stacktrace for informational purposes: " + connection + " " + LogUtil.getCallerStackTrace());
// only initialize the tx service client if needed and if we succeeded in getting a connection
// to HBase
if (transactionsEnabled) {
    initTxServiceClient();
}
```

删除