如何在 Hadoop 2.2.0 环境下使用 Spark 2.2.x

Apache Spark 2.2.0 于今年7月份正式发布，

- 这个版本是 Structured Streaming 的一个重要里程碑，因为其可以正式在生产环境中使用，实验标签（experimental tag）已经被移除；
- CBO （Cost-Based Optimizer）有了进一步的优化；
- SQL完全支持 SQL-2003 标准；
- R 中引入了新的分布式机器学习算法；
- MLlib 和 GraphX 中添加了新的算法

更多详情请参见：Apache Spark 2.2.0新特性详细介绍
。这么多的新特性，我们肯定很迫不及待地使用。但是如果你线上使用的 Hadoop 集群版本是 2.5 及其之前版本，那对不起了，这个版本的 Spark 你用不了了！因为从 Spark 2.2.0 开始移除了对 Apache Hadoop 2.5 及其之前版本的支持（详见：Remove support for Hadoop 2.5 and earlier）！
到 GitHub 下载支持 Hadoop 2.2.x 的 Spark 2.2.1 版本
难道我们得为了使用 Apache Spark 2.2.x 而升级咱们的 Hadoop 集群？这代价也太高了吧。。那咋办呢？其实我们可以修改 Apache Spark 2.2.x 关于 Hadoop 及 YARN 的相关代码，让它支持 Apache Hadoop 2.5 及其之前版本就行了。

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

Spark 2.2.x 不支持 Hadoop 2.2.0 的原因

其实 Spark 2.2.0 不支持 Hadoop 2.5 及其之前版本，是因为这个版本的 Spark 移除了那些兼容不同版本的 Hadoop 代码。因为 Hadoop 2.x 后面几个版本添加了许多类方法等，这些方法和类在之前的 Hadoop 版本不存在，特别是 Hadoop 2.2（我公司目前还在使用 Hadoop 2.2.0 ）、Hadoop 2.3 和 Hadoop 2.6 相比少了许多功能，为了支持不同版本的 Hadoop，Spark 代码里面就要编写很多兼容的代码，其中使用大量使用了反射机制。比如在 Client 类中创建 ApplicationSubmissionContext 的时候设置 spark.yarn.tags，Spark 2.1.0 是像下面实现的：

```
try {
  // The setApplicationTags method was only introduced in Hadoop 2.4+, so we need to use
  // reflection to set it, printing a warning if a tag was specified but the YARN version
  // doesn't support it.
  val method = appContext.getClass().getMethod(
    "setApplicationTags", classOf[java.util.Set[String]])
  method.invoke(appContext, new java.util.HashSet[String](tags.asJava))
} catch {
  case e: NoSuchMethodException =>
    logWarning(s"Ignoring ${APPLICATION_TAGS.key} because this version of " +
       "YARN does not support it")
}
```

因为 Spark 2.1.x 支持 Hadoop 2.2.x，而 setApplicationTags 是从 Hadoop 2.4 开始引入的，所以为了兼容 Hadoop 2.2.x，不得不这么写。而 Spark 2.2.x 移除了对 Hadoop 2.5 及其之前版本的支持，所以上面的代码可以直接一行搞定：

```
appContext.setApplicationTags(new java.util.HashSet[String](tags.asJava))
```

而像这样的代码在 Spark 2.1.0 存在很多，估计是这些代码太臃肿而且难以维护，所以社区宣布不再支持 Hadoop 2.5 及其之前版本了！

解决办法

好了，基于上面的分析，如果想在 Hadoop 2.2.x 上使用 Spark 2.2.x，我们完全可以把 Spark 2.2.x 移除的代码给弄回来，不就支持了吗？基于这个思路，我基于 Spark 2.2.1 和 Spark 2.1.0，将支持 Hadoop 2.2.x 的代码全部移回去了！然后可以直接把 Apache Spark 2.2.1 跑在 Hadoop 2.2.0 上，相关代码在：https://github.com/397090770/spark-2.2-for-hadoop-2.2。

上面代码在编译的时候如果不指定 Hadoop 默认使用的是 2.6.5，不过你可以通过 hadoop.version 参数指定你需要的版本，具体的编译如下：

```
# Apache Hadoop 2.2.0
./dev/make-distribution.sh --tgz -Phadoop-2.2 -Pyarn -DskipTests

# Apache Hadoop 2.3.0
./dev/make-distribution.sh --tgz -Phadoop-2.2 -Dhadoop.version=2.3.0 -Pyarn -DskipTests

# Apache Hadoop 2.3.0
./dev/make-distribution.sh --tgz -Phadoop-2.2 -Dhadoop.version=2.5.0 -Pyarn -DskipTests

# Apache Hadoop 2.6.5
./dev/make-distribution.sh --tgz -Phadoop-2.6 -Pyarn -DskipTests

# Apache Hadoop 2.7.X and later
./dev/make-distribution.sh --tgz -Phadoop-2.7 -Pyarn -DskipTests
```

小技巧

在编译的时候想加快速度，可以修改 ./dev/make-distribution.sh 文件的下面语句：

```
BUILD_COMMAND=("$MVN" -T 1C clean package -DskipTests $@)
```

上面的 1C 代表使用1个核进行编译，所以如果你的电脑不止一个核，你可以增加核的个数；另外，编译的时候加上 clean 参数之后，不管你代码是否修改了，都会编译，所以我们可以去掉这个参数，最后修改的结果如下：

```
BUILD_COMMAND=("$MVN" -T 4C package -DskipTests $@)
```