Spark on YARN集群模式作业运行全过程分析

《Spark on YARN集群模式作业运行全过程分析》
《Spark on YARN客户端模式作业运行全过程分析》
《Spark:Yarn-cluster和Yarn-client区别与联系》
《Spark和Hadoop作业之间的区别》
《Spark Standalone模式作业运行全过程分析》（未发布）

下面是分析Spark on YARN的Cluster模式，从用户提交作业到作业运行结束整个运行期间的过程分析。

客户端进行操作

1、根据yarnConf来初始化yarnClient，并启动yarnClient
2、创建客户端Application，并获取Application的ID，进一步判断集群中的资源是否满足executor和ApplicationMaster申请的资源，如果不满足则抛出IllegalArgumentException；
3、设置资源、环境变量：其中包括了设置Application的Staging目录、准备本地资源（jar文件、log4j.properties）、设置Application其中的环境变量、创建Container启动的Context等；
4、设置Application提交的Context，包括设置应用的名字、队列、AM的申请的Container、标记该作业的类型为Spark；
5、申请Memory，并最终通过yarnClient.submitApplication向ResourceManager提交该Application。
当作业提交到YARN上之后，客户端就没事了，甚至在终端关掉那个进程也没事，因为整个作业运行在YARN集群上进行，运行的结果将会保存到HDFS或者日志中。

提交到YARN集群，YARN操作

1、运行ApplicationMaster的run方法；
2、设置好相关的环境变量。
3、创建amClient，并启动；
4、在Spark UI启动之前设置Spark UI的AmIpFilter；
5、在startUserClass函数专门启动了一个线程（名称为Driver的线程）来启动用户提交的Application，也就是启动了Driver。在Driver中将会初始化SparkContext；
6、等待SparkContext初始化完成，最多等待spark.yarn.applicationMaster.waitTries次数（默认为10），如果等待了的次数超过了配置的，程序将会退出；否则用SparkContext初始化yarnAllocator；

怎么知道SparkContext初始化完成？
其实在5步骤中启动Application的过程中会初始化SparkContext，在初始化SparkContext的时候将会创建YarnClusterScheduler，在SparkContext初始化完成的时候，会调用YarnClusterScheduler类中的postStartHook方法，而该方法会通知ApplicationMaster已经初始化好了SparkContext

7、当SparkContext、Driver初始化完成的时候，通过amClient向ResourceManager注册ApplicationMaster

8、分配并启动Executeors。在启动Executeors之前，先要通过yarnAllocator获取到numExecutors个Container，然后在Container中启动Executeors。如果在启动Executeors的过程中失败的次数达到了maxNumExecutorFailures的次数，maxNumExecutorFailures的计算规则如下：

```
// Default to numExecutors * 2, with minimum of 3
private val maxNumExecutorFailures = sparkConf.getInt("spark.yarn.max.executor.failures",
    sparkConf.getInt("spark.yarn.max.worker.failures", math.max(args.numExecutors * 2, 3)))
```

那么这个Application将失败，将Application Status标明为FAILED，并将关闭SparkContext。其实，启动Executeors是通过ExecutorRunnable实现的，而ExecutorRunnable内部是启动CoarseGrainedExecutorBackend的。

9、最后，Task将在CoarseGrainedExecutorBackend里面运行，然后运行状况会通过Akka通知CoarseGrainedScheduler，直到作业运行完成。