一:Spark天堂之门

1. Spark程序在运行的时候分为Driver和Executor两部分；
2. Spark的程序编写是基于SparkContext的，具体来说包含两方面：
   1. Spark编程的核心基础-RDD，是由SparkContext来最初创建（第一个RDD一定是由SparkContext来创建的）
   2. Spark程序的调度优化也是基于SparkContext
3. Spark程序的注册是通过SparkContext实例化时候产生的对象来完成的其实是SchedulerBackend来注册程序）。
4. Spark程序运行的时候要通过ClusterManager获得具体的计算资源，计算资源的获取也是通过SparkContext产生的对象来申请的（其实是SchedulerBackend来获取计算资源的）；
5. SparkContext崩溃或者结束的时候整个Spark程序也结束了

总结：

SparkContext开启了天堂之门：Spark程序是通过SparkContext发布到Spark集群的;

SparkContext导演天堂世界：Spark程序的运行都是在SparkContext为核心的调度器的指挥下进行的；

SparkContext关闭天堂之门：SparkContext崩溃或者结束的时候整个Spark程序也结束了

二：SparkContext使用案例鉴赏：

三：SparkContext天堂内幕

1. SparkContext构建的构建的顶级三大核心对象：DAGScheduler，TaskScheduler，SchedulerBackend，其中：
   1. DAGScheduler是面向Job的Stage的高层调度器；
   2. TaskScheduler是一个借口，根据具体的Cluster Manager的不同会有不同的实现，standalone模式下具体的实现是TaskSchedulerImpl；
   3. SchedulerBackend是一个接口，根据具体的Cluster Manager的不同会有不同的实现，standalone模式下具体的实现是SparkDeploySchedulerBackend；
2. 从整个程序运行的角度来讲，SparkContext包含四大核心对象：DAGScheduler，TaskScheduler，SchedulerBackend，MapoutputTrackerMaster
3. SparkDeploySchedulerBackend有三大核心功能：
   1. 负责与Master链接注册当前程序；
   2. 接受集群中为当前应用程序而分配的计算资源Executor的注册并管理Executors
   3. 负责发送Task到具体的Executor执行；

补充说明的是：SparkDeploySchedulerBackend是被TaskSchedulerImpl来管理的。

1. 当通过SparkDeploySchedulerBackend注册程序给Master的时候会把Command提交给Master，Master发指令给Worker去启动Executor所在的进程的时候加载的main方法所在的入口类就是command中的CoarseGrainedExecutorBackend，当然你也可以实现自己的ExecutorBackend，在CoarseGrainedExecutorBackend中启动Executor，Executor通过线程池并发执行Task。（Executor实现注册然后实例化）