1. 集群启动的时候启动了Master和Worker.
2. 用户程序提交：
   1. Client：Dirver运行在Client端，开始初始化SparkContext。SparkContext的初始化的过程中初始化SparkDeploySchedulerBackend，SparkDeploySchedulerBackend启动的时候：
      1. 首先调用它父类的start方法（源码如下两行）：

override def start() {  
 super.start()

* + - 1. 根据程序执行的流程，所以先执行super。Start方法。进入之后他会初始化DriverEndpoint，因为他是RPC。所以调用他的Onstart方法。他的Onstart方法里面怎么又ReviveOffers。这个消息会导致任务的执行。（此时并没有作业啊！仅仅是初始化了SparkContext啊）

Ii. 然后回来继续执行，主要的作用就是想Master注册App。进而导致Executor的分配。（Executor注册给DirverEndpoint的时候也会调用makeOffers，这个也会导致作业的运行啊！但是此时还没有任务触发作业。）

* 1. SparkContext执行完成之后，然后sc。Runjob触发作业的运行，然后调用DAGScheduler，TaskScheduler,最后交给SparkDeploySchedulerBackend来提交作业给ExecutorBackend来运行作业。

1. 在CoarseGrainedSchedulerBackend里面时候ctrl+f 搜索ReviveOffers和makeOffers这两个关键字，可以看到许多地方使用了这个两个，他们的调用时机到底在什么时候，难道我的逻辑有问题吗？
2. <http://www.cnblogs.com/barrenlake/p/4891589.html>
3. <http://blog.csdn.net/joeyangy/article/details/40345725>
4. DT笔记<https://opensourceteam.gitbooks.io/bigdata/content/spark/source/11-taskscheduleryuan_li_he_yuan_ma_jie_mi.html>
5. http://www.jianshu.com/p/9a059ace2f3a