一:Spark Executor工作原理

1. 需要特别注意的是在CoarseGrainedExecutorBackend启动时想Driver注册Executor，其实是注册ExecutorBackend实例，和Executor实例之间没有直接的关系！
2. CoarseGrainedExecutorBackend是Executor运行所在的进程名称，Executor才是真正处理Task的对象，Executor内部是通过线程池的方式来完成Task的计算的；
3. CoarseGrainedExecutorBackend和Executor一一对应
4. CoarseGrainedExecutorBackend是一个消息通信提，可以发送信息给Dirver并可以接受Driver中发送过来的指令，例如启动Task等。
5. 在Driver进程中有两个至关重要的Endpoint；
   1. ClientDriver：主要负责向Master注册当前的程序，是AppClient的内部成员；
   2. DriverEndpoint：这是整个程序运行时候的驱动器，是CoarseGrainedSchedulerBackend的内部成员；
6. 在Driver中通过ExecutorData封装并注册ExecutorBackend的信息到Dirver的内存数据结构executorDataMap中
7. 实际执行的时候DriverEndpoint会把信息写入CoarseGrainedSchedulerBackend的内存数据结构ExecutorDataMap中，所以说最终是注册给了CoarseGrainedSchedulerBackend，也就是说CoarseGrainedSchedulerBackend掌握了当前程序分配的所有的ExecutorBackend进程，而在每一个ExecutorBackend进程实例中会通过Executor对象来负责具体Task执行，在运行的时候使用synchronized关键字来保证ExecutorDataMap安全的并发写操作
8. CoarseGrainedExecutorBackend收到DriverEndpoint发送过来的RegisteredExecutor消息后启动Executor实例对象，而Executor实例对象是事实上负责真正的Task计算的。