

## 6.4 Spark运行基本流程

接下来，我们来一起了解一下spark运行的基本流程：前面我们给大家讲过，整个应用程序提交以后，它有一个指挥所就是Driver，它相当于一个管家，就是应用程序的指挥所，而指挥所我们知道需要一个指挥官，后面我们写代码的时候会经常生成一个叫做Sparkcontext的对象，而这个对象就是指挥官。我们给大家讲过，整个应用程序提交以后，它有一个指挥所就是Driver，它相当于一个管家，就是应用程序的指挥所，而指挥所我们知道需要一个指挥官，后面我们写代码的时候会经常生成一个叫做Sparkcontext的对象，而这个对象就是指挥官。由它负责对整个作业的调度，作业怎么分解，分解之后交给不同的节点去执行，执行完之后怎么把不同节点上执行的结果汇总提交给用户，这些都由指挥官说了算，这个指挥官就是SparkContext，这个指挥官就呆在指挥所里，指挥所就是Driver。如图6.4.1，当应用程序提交了以后，究竟发生了什么呢？

首先需要构建集群，这是最基本的运行环境，整个环境要搭建起来，怎么构建集群呢？集群有了，应用程序提交了，所以应用提交的时候，要先给它运行应用的主节点，就是Driver节点，一旦确定好，指挥所建好以后，就给他配备了指挥官，就是SparkContext对象，如图6.4.2

由这个对象向资源管理器去申请资源，然后负责把整个作业分成不同的阶段，并且把每个阶段的任务调度到不同的工作节点去完成，这都是它的工作。执行过程中，它还要负责监控，一旦失败了，它要负责恢复过来，如图6.4.3

第二步就是资源管理器，如图6.4.4上节我们讲过，资源管理器有几种，可能是YARN，可能是Mesos，资源管理器收到来自SparkContext资源请求之后，资源管理器会为Executor的进程去分配资源去执行，分配什么资源呢，即CPU和内存资源，分配好资源后就可以去启动Executor的进程了，就是由进程再派生出线程，Executor进程是驻留在不同的worknode节点上的，可能有几百甚至上千台机器。这几百上千台机器Worknode上面都是各个Executor进程，如图6.4.5

然后第三步，这个Sparkcontext对象要根据RDD的依赖关系，构建一张DAG图，怎么知道RDD的依赖关系呢，写的代码就是针对RDD的一次又一次操作，每一行语句写下去，就是对他的一次操作，这个操作就会被转换成一个有向无环图就是DAG图，如图6.4.6

生成这个图做什么呢？这个图会被提交到DAG Scheduler的模块，去进行解析，解析它做什么呢，解析它是要把DAG图分解成不同的Stage，如图6.4.7

每个Stage都包含若干个Task，所以Stage是一个Task集合，得到一个又一个任务阶段之后呢，就提交给下一个Taskscheduler，如图6.4.8

这个Taskscheduler就是具体负责把任务到底分发到哪个节点上面去，，由它来负责，它怎么分发呢，我们整个分发过程了解一下，Taskscheduler不会拿到任务就往外扔，而是各个WorkNode上面的那些Executor会主动向TaskTracker去申请运行任务，然后TaskTracker就会把我们的任务根据它申请的情况，扔到相对应的Worknode上，让Executor进程去派送线程去执行，任务怎么分配呢，这里有一个基本原则，什么原则呢，就是这么多的数据要计算到底应该把它扔到哪个节点上面呢，

有一个原则就是要保证计算向数据靠拢。数据在机器A，就优先把数据扔到所在节点，完成本地化处理，而不是把数据远程传输到计算所在的机器B上。TaskScheduler就本着计算向数据靠拢的原则，如果数据在机器A上，就把程序给机器A，而不是把它扔给机器B，这是它的基本原则，TaskScheduler就会向Task运行的地方发送相关的代码，代码就是执行计算的，把它给到机器A。（需要做一个小动画）

第四步，这些任务在Excutor上面运行以后呢，它会得到结果，这个结果又会反馈给TaskSchedule，再由TaskSchedule往上传递给DAGScheduler。然后再由Sparkcontext对象做一个最后的处理，或者返回给用户看，或者写入到HDFS，或者写入到其他的文件当中去。这就是Spark执行的整个流程。

然后就知道SparkContext对象就是整个运行过程的指挥官，而它也代表了应用程序连接集群的通道，应用程序怎么和整个集群里几千台机器发生关联呢，是由这个SparkContext建立的一个通道，（制作一个小动画）

它把应用程序拿过来进行分发，扔到不同的机器上去，所以Sparkcointext就相当于在底层集群之间建立起了这样的一个连接，你是通过这个对象才和底层的几千台机器发生交互的。这正好体现了我们古人“大事化小，小事化了”

处事思想。

