**Cluster Mode of Spark**

**—— 概览**

本文提供一个简短的概览，关于Spark如何在集群上运行，并更好的理解其中的组件。

**Components**

Spark应用通过多组独立的进程在集群上运行，通过主程序中的SparkContext对象协作（Driver程序）。

在集群上运行时，SparkContext可以连接多种集群管理器类型（Spark单例模式的Mesos或YARN），管理器会为应用分配资源。当连接完成后，Spark会通知集群结点上的执行器，执行器是用于执行计算和存储数据的进程。接着，会发送应用代码到执行器，最后，SparkContext发送任务到执行器执行。

结构上有几点注意：

1. 每个应用获取自己的执行器进程，该进程会持续整个应用的运行时间并在多线程上运行。该特点有利于在执行端和计划端隔离应用。但是，这也意味着数据是无法在应用间共享的，除非使用外部存储系统。
2. Spark对于底层集群管理器是不可知的，只要它可以获取执行程序进程，并且这些进程相互通信，即使在也支持其他应用程序的集群管理器上运行它也相对容易。
3. Driver必须监听并接收从执行器来的连接，因此，driver对于worker来说必须在网络上可寻址。
4. 因为Driver在集群上计划任务，应该尽可能的离worker近，最好是在同一片局域网中。若需要远程调用，最好是为driver开启RPC并就近提交操作。

**Cluster Manager Type**

Spark目前支持以下几种集群管理器：

* Standalone，Spark自带的简单管理器，可以很简单的搭建一个集群。
* Apache Mesos，常用的集群管理器，可以允许Hadoop MapReduce和服务应用。
* Hadoop YARN，资源管理器。
* Kubernetes，自动部署，可扩展，容器化应用管理。

**Submitting Applications**

应用可以通过spark-submit脚本提交到集群中。

**Monitoring**

每个Driver都有个web UI，默认端口4040，会显示运行中的任务，执行器和可用存储。

**Job Scheduling**

Spark可以控制跨应用程序（在集群管理器级别）和应用程序内的资源分配（如果在同一个SparkContext上发生多个计算）。

**Glossary**

Application，Spark构建的用户程序，由driver和executor组成。

Application Jar，包含Spark应用的jar包，某些场景下用户想创建一个现有的jar包含其应用和依赖。其中不应该包含任何Hadoop和Spark的包，会在执行时添加。

Driver Program，执行main函数的应用进程，会创建SparkContext。

Cluster Manager，用于请求资源的外部服务。

Deploy mode，区别driver运行位置，在cluster模式，框架在集群内加载driver，在client模式，在集群外通过submitter加载driver。

Worker node，能运行应用代码的节点。

Executor，工作节点上的进程，执行任务并保存数据在内存或磁盘，每个应用有各自的执行器。

Task，工作单元，会发送给一个执行器。

Job，由多个任务组成的并行计算，这些任务是为响应Spark操作而生成的