运行 Spark 应用

运行 Spark 应用

您可以在本地或者群集中运行 Spark 应用程序，既可以通过使用 交互式 Shell 的方式， 又可以通过 提交一个应用程序 的方式。通常在执行数据探索阶段和为了 Ad-Hoc 分析 时运行交互式的 Spark 应用程序。

因为 Scala 编译代码方式的一些局限性，一些运行在交互式 shell 的嵌套定义的应用也许会发生一个 Task not serializable 异常。Cloudera 建议您提交那些应用。

为了在群集中运行应用程序，Spark 需要一个 Cluster Manager（群集管理者）。Cloudera 支持两种 Cluster Manager : YARN 和 Spark Standlone。当运行在 YARN 上时，Spark 应用程序的进程被 YARN ResourceManager 和 NodeManager 角色管理。当运行在 Spark Standlone 上时，Spark 应用程序的进程被 Spark Master 和 Worker 角色管理。

在 CDH5 中，Cloudera 建议在 YARN 群集管理上运行 Spark 应用，而不是在 Spark Standlone 群集管理上运行，YARN 上运行有下列的好处 :

- 您可以动态共享和集中配置所有运行在 YARN 上的框架的相同的群集资源的 pool（池）。
- 您可以使用 所有 YARN 调度器的特性 用于分类，隔离，和优先级的工作。
- 您选择需要使用的 Executor 的数量。与此相反，Spark Standlone 需要每个应用去在群集中的每个主机上运行一个 Executor。
- Spark 可能运行在没启用 Kerberos 的 Hadoop 群集并且在它的进程中使用 安全的身份认证。

更多关于监控 Spark 应用的信息，请看 监控 Spark 应用。

继续阅读 :

提交 Spark 应用

为了提交由一个 Python 文件或者一个编译并且打包的 Java 或者 Spark JAR 的应用，使用 spark-submit 脚本。

spark-submit 语法

```
spark-submit --option value \
  application jar | python file [application arguments]
```

例子 : 在 YARN 上运行 SparkPi 演示了如何去运行一个简单的例子，SparkPi，与 Spark 打包。它计算了 PI 的近似值。

spark-submit 参数

选项	描述
application jar	包含一个 Spark 应用和所有依赖的 JAR 文件的路径。这个路径必须在您的群集内部是可见的，请看 高级依赖管理 。
python file	包含一个 Spark 应用的 Python 文件的路径。这个路径必须在您的群集内部是可见的，请看 高级依赖管理 。
application arguments	传递到您的 main class 中 main 方法的参数。

spark-submit 选项

使用 --option value 而不是 --option=value 的形式指定 spark-submit 的选项。（使用一个空格而不是一个等号）。

of an equals sign.)

选项	描述
class	对于 Java 和 Scala 应用来说。包含应用的 main 方法是完全限定的 classname（类名）。例如 : org.apache.spark.examples.SparkPi。
conf	Spark 配置属性 用 key=value 的格式。多个 values 包含空格，使用引号 "key=value"。
deploy-mode	部署模式 : cluster 和 client。在 cluster 模式中，driver 运行在 worker 主机上。在 client 模式中，driver 作为一个外部的 client 运行在本地。生产环境的 Job 使用 cluster 模式。 client 模式更适合用于交互式和调试案例，这样您可以立刻看到您应用的输出。为了在 YARN 上运行时看到部署模式的效果，请看 部署模式。 默认 : client。
driver-class-path	配置和进入到 dirvier 的 classpath 入口。JARs 使用 --jars 添加的 JARs 被自动带包含到 classpath 中。
driver-cores	cluster 模式中 driver 使用的 core 的数量。 默认 : 1。
driver-memory	分配给 driver 的最大的堆内存大小（表示为一个 JVM 字符串，例如 1024m，2g，等等），您可以使用 spark.driver.memory 属性。
files	逗号分隔的文件列表被放到每个 executor 的工作目录。该路径必须在您的群集内部是全局可见的，请看 高级依赖管理。
jars	添加 JARs 加载到 driver 的 classpath 中，并且 executo 在 cluster 模式中或者在 client 模式的 executor classpath 中。该路径必须在您的群集内部是全局可见的，请看 高级依赖管理。
master	运行应用的位置。
packages	逗号分隔的 Maven JARs 的坐标的列表，包括在 driver 和 executor classpath 上。本地的 Maven，Maven central，和被指定在仓库中的远程仓库用于说说的顺序。坐标的格式是 groupId:artifactId:version。
py-files	逗号分隔的 .zip，.egg，或者 .py 文件用来放到 PYTHONPATH 上，该路径必须在您的群集内部是全局可见的，请看 高级依赖管理。
repositories	逗号分隔的远程仓库用来搜索在 package 中指定的 Maven 坐标。

Master Values

Master	描述
local	用一个 worker 线程本地运行 Spark（也就是说，不是并行的）。
local[K]	用 K 个 worker 线程本地运行 Spark （理想状态，设置这个值为您主机的 Core 数量）
local[*]	用您主机上尽可能多的逻辑 Core 本都运行 Spark。
spark://host:port	使用 Spark Standlone 群集管理器与指定的主机和端口（默认 : 7077）运行。
yarn	使用 YARN cluster manager 运行。群集位置由 HADOOP_CONF_DIR 或者 YARN_CONF_DIR 决定。请看 配置环境。

群集执行概述

Spark 通过 driver 程序协调操作。当 driver 程序运行时，Spark 框架初始化您群集中处理数据的主机上的 executor 进程。当您提交 Spark 应用程序到群集中是发生了如下事情 :

- driver 启动并且调用 Spark 应用程序中的 main 方法。
- driver 从群集管理器中请求资源用来启动 executor。
- 群集管理器为了 driver 程序启动 executor。
- driver 运行应用程序。基于应用程序的 transformation 和 action，driver 发送任务到 executor。
- 任务运行在 executor 上以计算和保存结构。
- 如果启用了 动态分配，在指定的阶段 executor 是空闲时，它们将会释放。
- 当 driver 的 main 方法退出或者调用了 SparkContext.stop，它中断了任何未完毕的 executor 并释放来自群集管理器的资源。

继续阅读

- YARN 上运行 Spark 应用
- 使用 PySpark
  - 使用 IPython 和 Jupyter Notebook 运行 Spark 应用
  - 运行 Spark Python 应用
- 优化 Spark 应用