**第二十七课：Spark on yarn 彻底解密**

**本期内容：**

    1、Hadoop yarn解密

    2、Spark on yarn 解密

**一、Hadoop Yarn 解析**

1、Yarn是Hadoop推出整个分布式（大数据）集群的资源管理器，负责资源的管理和分配，基于Yarn我们可以在同一个大数据集群上同时运行多个计算框架，例如Spark，MapReduce、Storm等;

2、Spark On Yarn运行工作流程图

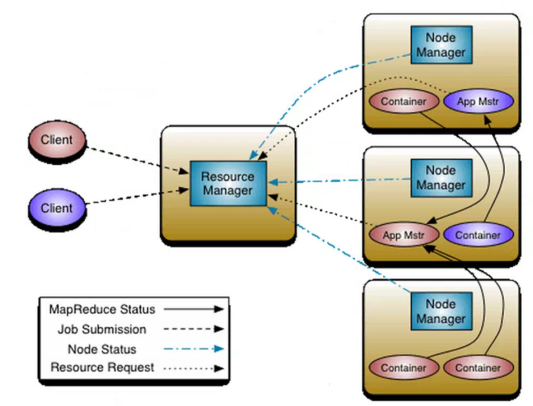


图1 Spark on Yarn 工作流程

    3、Client端向ResourceManager提交Application，ResourceManager接收应用并根据集群资源状况决定在具体某个Node上来启动当前提交的应用程序的任务调度器Driver（ApplicationMaster），然后决定命令具体的某个Node上的资源管理器NodeManager来启动一个新的JVM基础运行程序的Driver部分，当ApplicationMaster启动的时候会下载当前Application相关的Jar等各种资源并基于此决定向ResourceManager申请资源的具体内容（例如需要多少个Container，和Container的配置），ResourceManager接受到ApplicationMaster的资源分配的请求后会最大化的满足资源分配的请求，并把资源的元数据信息发送给ApplicationMaster，ApplicationMaster收到资源的元数据信息后，会根据元数据的信息发送指令给具体机器上的NodeManager，让NodeManager来启动具体的Container，Container在启动后必须向ApplicationMaster注册，当ApplicationMaster获得了用于计算的Container后，开始并行任务的调度和计算，直到作业运行完成。

    需要说明的是：如果ResourceManager第一次没有能够完全完成资源分配的请求，后续ResourceManager发现集群中有新的可用资源时，会主动向ApplicationMaster发送新的可用资源的元数据信息，以提供跟多的资源用于当前程序的运行。

补充说明：

    1）如果是Hadoop的MapReduce计算的话Container不可以复用，如果是Spark on Yarn的话Container可以复用；

    2）Container具体的销毁是由ApplicationMaster来决定的；

    3）ApplicationMaster 发指令给NodeManager让NodeManager销毁Container；

1. **Spark on Yarn的两种运行模式实战**

此时不需要启动Spark集群，只需要启动Yarn即可，Yarn的ResourceManager就相对于Spark Standalone模式下的Master！（我们启动spark集群是要用到standalone，现在有yarn了，就不用spark集群了）

    1、Spark on Yarn的两种运行模式：唯一的决定因素是当前Application从任务调度器Driver运行在什么地方！

        a) Cluster：（如果Spark运行在on Yarn上，根本就没必要启动Spark集群，Master是ResourceManager）

        b) Client：Driver运行在当前提交程序的客户机器上；

        需要说明的是：无论什么模式，只要当前机器运行了Spark，首先需要安装Spark、Scala、Java，不用启动Spark集群，但是Spark On Yarn模式下，应用程序提交后资源和集群的管理模式不同，但作业实际是运行在Spark集群架构中，Standalone模式下有Master，Worker，Spark On Yarn模式下Master相当于ResourceManager。

**实战运行Spark on yarn**

启动Hadoop、HDFS

./start-dfs.

启动Yarn

./start-yarn.sh

启动jobHistoryServer

root@Master:/usr/local/spark-1.6.0-bin-hadoop2.6/# jps

23432 jps

6543 NameNode

7456 HistoryServer

9734 ResourceManager

8398 SecondaryNameNode

提交应用程序：

./spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode client ../lib/spark-examples-1.6.0-hadoop2.6.0.jar 50000

    以上命令提交作业，其中：

        a）--class org.apache.spark.examples.SparkPi：运行SparkPi；

        b）--master yarn：指定作业运行在yarn上；

        c）--deploy-mode client：使用yarn的Client模式运行作业；

        d）../lib/spark-examples-1.6.0-hadoop2.6.0.jar：SparkPi所在的jar的位置；

        e）50000：指定并行度50000（即50000JVM虚拟机）

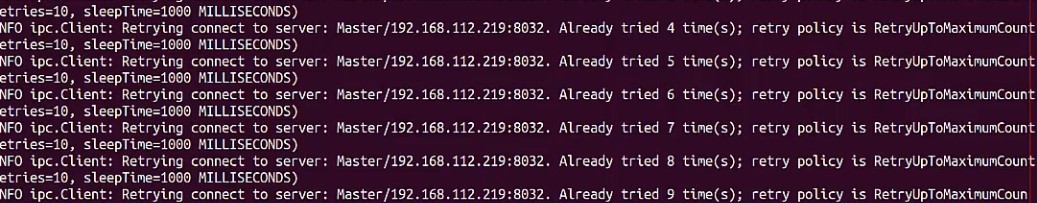
将并行任务变成50000个（50000台虚拟机）

      2、Spark on Yarn的运行实战：

        a) Client模式：方便在命令终端

**天机解密：**Standalone模式下启动Spark集群（也就是启动Master和Worker）其实启动的是资源管理器，真正作业计算的时候和集群资源管理器没有任何关系，所以Spark的Job真正执行作业的时候不是运行在我们启动的Spark集群中的，而是运行在一个个JVM中的，只要在JVM所在的集群上安装配置了Spark即可！

当没有启动yarn和spark-all的时候运行提交上述作业，会提示找不到Server，此时集群会一直尝试retry连接，如下如所示：



当在retry的过程中启动了yarn后，集群自动连接，此时上传jar文件到HDFS文件系统，如下代码所示：

Uploading resource file:/usr/local/spark/spark-1.6.0-bin-hadoop2.6/lib/spark-assembly-1.6.0-hadoop2.6.0.jar -> hdfs://Master:9000/user/root/.sparkStaging/application\_1454420297706\_0001/spark-assembly-1.6.0-hadoop2.6.0.jar

注：通过<http://Master:8088>查看作业运行情况

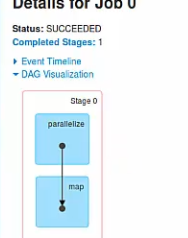
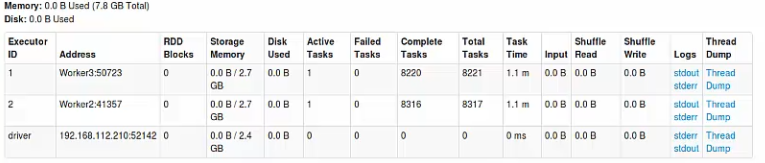


图3 运行SparkPi产生的DAG

因为SparkPi中只有map和reduce操作，所以只有一个Stage。



        cluster的模式下driver在App Master中，driver收集了作业运行的主要信息，而yarn资源管理器管理了作业运行信息的端口，因此在<http://Master:8088>中查看作业运行的详细信息。

      b) Client模式：

./spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode cluster ../lib/spark-examples-1.6.0-hadoop2.6.0.jar 50000

注：可以通过<http://Master:4040>查看日志信息

      3、Spark on Yarn模式下Driver与ApplicationMaster的关系：

        a) Cluster：Driver位于ApplicationMaster进程中，我们需要通过Hadoop默认指定的8088端口来通过Web控制台查看当前的Spark程序运行的信息，例如进度、资源的使用（Cluster的模式中Driver在App Master中）；

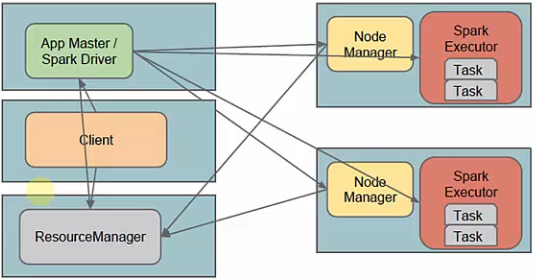


图4 Cluster模式

        b)Driver为提交代码的机器上，此时ApplicationMaster依旧位于集群中且只负责资源的申请和launchExecutor，此时启动后的Eexcutor并不会向ApplicationMaster进程注册，而是向Driver注册！！！

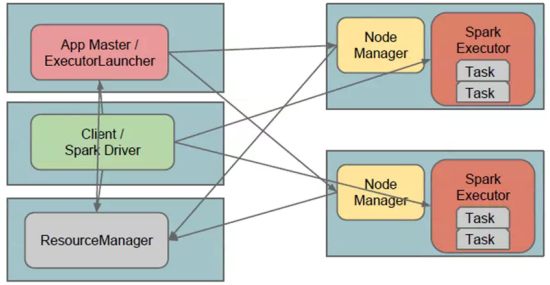


图5 Client模式

        可以看到Driver在客户机上，App Master中主要是用来向ResourceManager申请资源和ExecutorLauncher启动Executor，Executor启动后是向Driver注册的。

三：最佳实践

    1、在Spark on Yarn的模式下HadoopYarn的配置yarn.nodemanager.local-dirs会覆盖Spark的Spark.local.dir；

    2、在 实际生产环境下一班都是采用Cluster，我们会通过HistoryServer来获取最终全部的集群运行的信息；

    3、如果想直接看运行的日志信息，可以使用以下命令：

        yarn logs -ApplicationId <app ID>

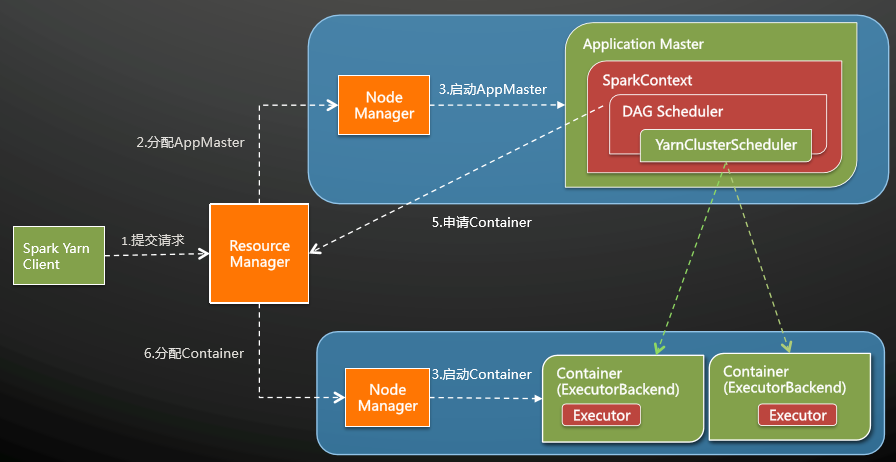


图6 Spark On Yarn运行机制图