组件

- ResourceManager
  - 总管理者
  - 以后台进程运行，8042访问
  - 根据程序优先级、队列容量、数据位置等标准分配资源、保证集群安全
- NodeManager
  - 节点管理者
  - 启动时向RM注册，并定时发送心跳
  - 维护Container生命周期，监控其资源使用
  - 管理任务依赖，根据AM需要在Con启动前将程序依赖拷贝到本地
- ApplicationMaster
  - 应用管理
  - 向RM申请资源
  - 动态分配来自RM的资源
  - 跟踪任务状态，报告资源使用情况
  - 任务容错
- Container
  - 资源封装运输，类似容器
  - 它封装了某个节点上的多维度资源，如内存、CPU、磁盘、网络等

工作流程

- Client 提交作业到 YARN 上
- Resource Manager 选择一个 Node Manager，启动一个 Container 并运行 Application Master 实例；
- Application Master 根据实际需要向 Resource Manager 请求更多的 Container 资源（如果作业很小, 应用管理器会选择在其自己的 JVM 中运行任务）；
- Application Master 通过获取到的 Container 资源执行分布式计算。

工作流程详解

作业提交

1，客户端提交作业信息到MapReduce 2，MapRecude向RM申请作业ID 3，MapReduce将作业copy到HDFS，包括包括 Jar 包，配置文件, split 信息 （split是有客户端完成的） 4，向RM提交作业

作业初始化

5，RM开启一个Con传到NM，给RM分配资源 6，NM开启进程监控资源使用、任务完成情况 7，通过客户端计算的split结果创建map任务，根据map结果创建reduce对象

任务分配

7，如果任务小，会在JVM中运行，任务大会在分布式系统中运行：通过心跳，向RM请求Con运行map和reduce任务

任务运行

8，当任务分配了Con之后，AM启动Con 9，一个YarnChlid的Java应用初始化任务的资源（配置、JAR包等）

作业完成

10，客户端每5分钟调用waitForCompletion()检查作业是否完成，时间间隔可通过配置文件修改

提交作业命令

```
# 提交格式: hadoop jar jar包路径 主类名称 主类参数
# hadoop jar hadoop-mapreduce-examples-2.6.0-cdh5.15.2.jar pi 3 3
```

调度选项

- FIFO：先进先出
- Fair：公平调度
  - 默认策略
  - 集群只有一个任务时分配所有资源，任务多了均分
- Capacity：容量调度
  - 每个任务分配资源队列
  - 队列资源富裕了可以给别人
  - 没有资源不能跟别人要
  - 大任务很难受

抢占（公平调度专有）

允许调度器终止那些占用资源超过分配给它的容器（Con）

被终止的Con需要重新执行（类似rollback）

延迟调度

繁忙的集群上，应用请求节点，这个节点可能正在运行其他Con，多等几秒可以增加请求节点上分配Con的机会，提高集群效率

容量和公平调度都支持延迟调度

资源公平（内存&CPU）

两个不同类型的应用（moreCPU or more 内存）分到的容器数是不同的

根据两任务CPU和内存需求占集群资源总量的比例不同，判断两任务的主类资源是什么

得到Con数量的占比等于主类资源的占比
