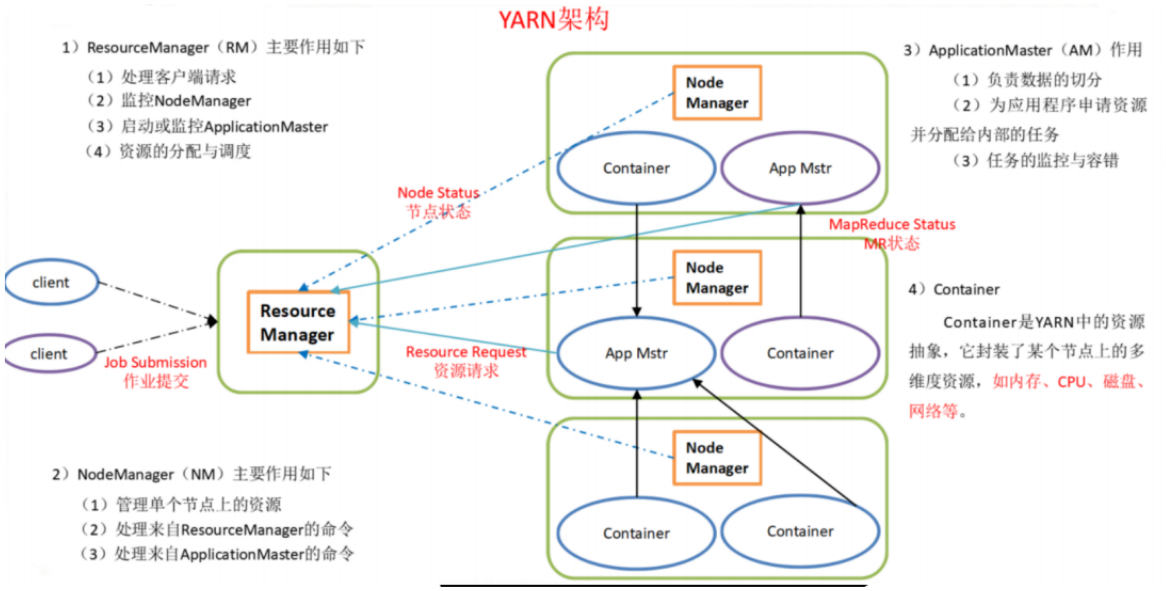
Yarn5 工作机制和作业提交

Yarn资源调度器

> Yarn是一个资源调度平台，负责为运算程序提供服务器运算资源，相当于一个分布式的操作系统平台，而MapReduce等运算程序则相当于操作系统之上的应用程序

`Yarn`主要由`ResourceManager`、`NodeManager`、`ApplicationMaster`和`Container`等组件构成如所示



ResourceManager(RM)

1. 处理客户端请求
2. 监控NodeManager
3. 启动或监控ApplicationMaster
4. 资源的分配和调度

NodeManager(NM)

1. 管理单个节点上的资源
2. 处理来自ResourceManager的命令
3. 处理来自ApplicationMaster的命令

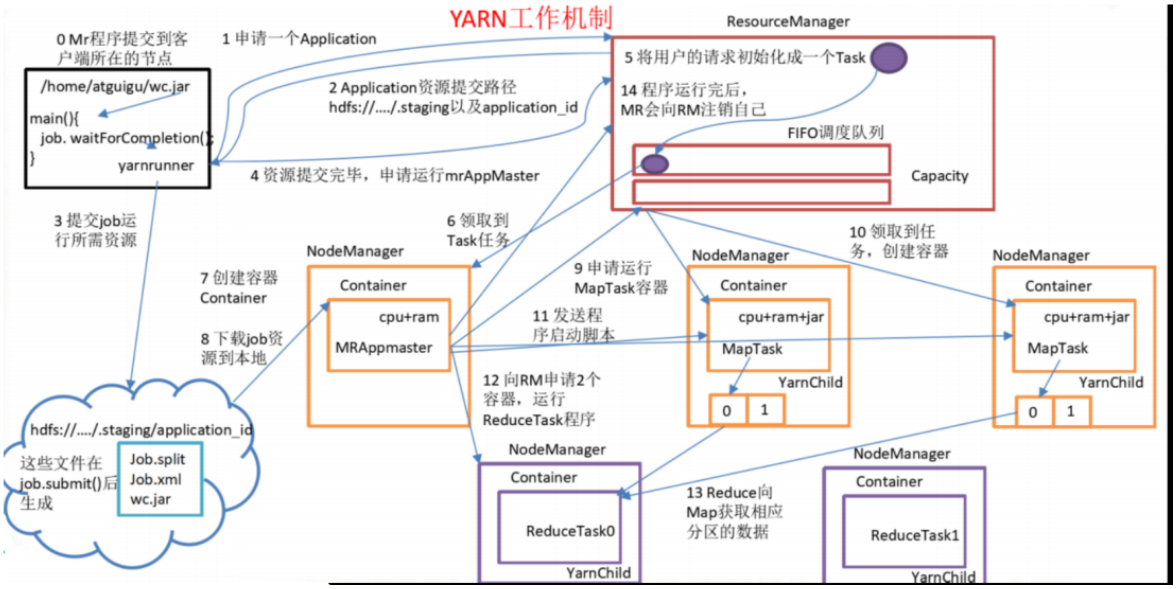
ApplicationMaster(AM)

1. 负责数据的切分
2. 为应用程序申请资源并且分配给内部的任务
3. 任务的监督和容错

Container

Container是YARN中的资源抽象，它封装了某个节点上的多维度资源，如：内存、CPU、磁盘、网络等

YARN工作机制（重点）



工作机制详解

（1）MR程序提交到客户端所在的节点，然后封装成一个YarnRunner

（2）YarnRunner向ResourceManager申请一个Application。

（3）RM将应用程序的资源路径返回给YarnRunner

（4）该程序将运行所需要的资源提交到HDFS上

（5）程序资源提交完成之后，申请运行MRAppMaster

（6）RM将用户请求初始化为一个Task

（7）其中一个NodeManager领取到Task任务

（8）该NodeManager创建容器Container，并产生MRAppmaster

（9）Container从HDFS上拷贝资源到本地

（10）MRAppmaster向RM申请运行MapTask资源

（11）RM将运行MapTask任务分配给另外两个NodeManager，另外两个NodeManager分别领取任务并且构建容器

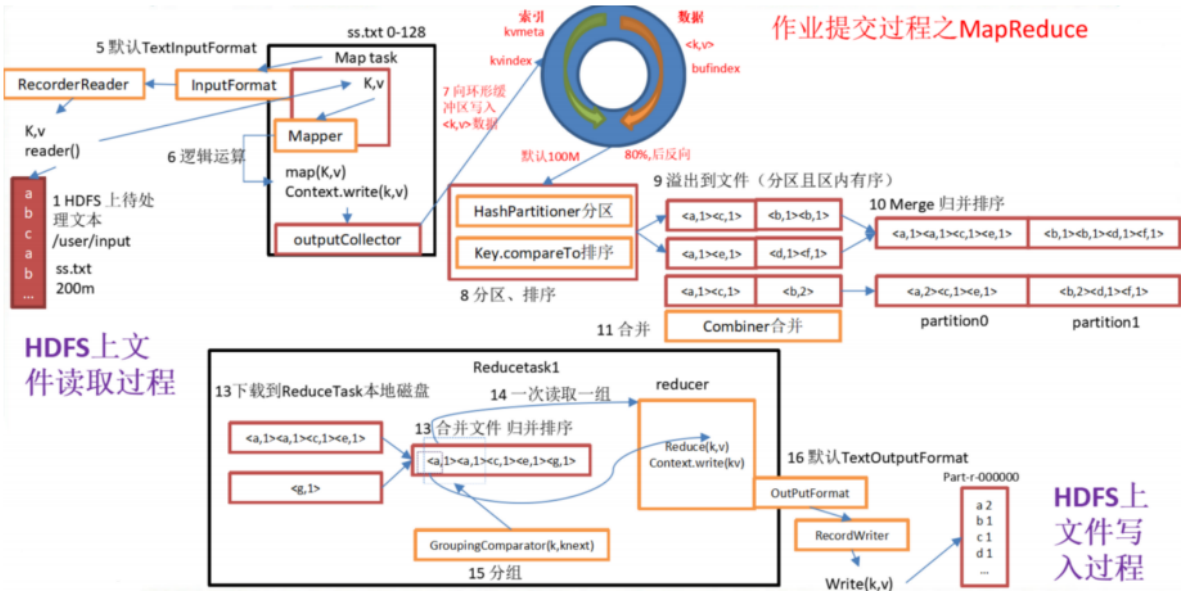
（12）MR向两个收到任务的NodeManager发送程序启动脚本，这两个NodeManager分别启动MapTask，MapTask对数据分区排序

（13）MRAppMaster等待所有MapTask运行完毕之后，向RM申请容器，运行ReduceTask

（14）ReduceTask向MapTask获取相应分区的数据

（15）程序运行完毕后，MR会向RM申请注销自己

作业提交全过程



（1）作业提交

第1步：Client调用job.waitForCompletion方法，向整个集群提交MapReduce作业

第2步：Client向RM申请一个作业id

第3步：RM给Client返回该job资源的提交路径和作业id

第4步：Client提交jar包、切片信息和配置文件到指定的资源提交路径

第5步：Client提交完资源后，向RM申请运行MRAppMaster

（2）作业初始化

第6步：当RM收到Client的请求后，将该job添加到容量调度器中

第7步：某一个空闲的NM领取到该Job

第8步：该NM创建Container，并产生MRAppmaster

第9步：下载Client提交的资源到本地

（3）任务分配

第10步：MRAppMaster向RM申请运行多个MapTask任务资源

第11步：RM将运行MapTask任务分配给另外两个NodeManager，另两个NodeSManager分别领取任务并创建容器。

（4）任务运行

第12步：MR向两个接收到任务的NodeManager发送程序启动脚本，这两个NodeManager分别MapTask、MapTask对数据分区排序

第13步：MRAppMaster等待所有MapTask运行完毕后，向RM申请容器，运行ReduceTask。

第14步：ReduceTask向MapTask获取相应分区的数据

第15步：程序运行完毕后，MR会向RM申请注销自己

（5）进度和状态更新

YARN中的任务将其进度和状态（包括counter）返回给应用管理器，客户端每秒（通用mapreduce.client.progreemonitor.pollinterval设置）向应用管理器请求进度更新，展示给用户

（6）作业完成

除了向应用管理器请求作业进度外，客户端每5分钟都会通过调用waitForCompletion()来检查作业是否完成。时间间隔可以通过mapreduce.client.completion.pollinterval来设置。作业完成之后，应用管理器和Container会清理工作状态。作业的信息会被作业历史服务器存储以备之后用户核查