



x2



x1

络等信息，MapReduce 程序的所有 task 都是在一个容器里执行完成的，容器的大小是可以动态调整的

1.2.5、ASM(ApplicationMaster)

这里写错了

应用程序管理器 ASM 负责管理整个系统中所有应用程序，包括应用程序提交、与调度器协商资源以启动 MRAppMaster、监控 MRAppMaster 运行状态并在失败时重新启动它等

目录

1、YARN.....	1
1.1、YARN 概述.....	1
1.2、YARN 的重要概念.....	2
1.2.1、ResourceManager	2
1.2.2、NodeManager	2
1.2.3、MRAppMaster	2
1.2.4、Container.....	2
1.2.5、ASM	2
1.2.6、Scheduler	2
1.3、YARN 架构.....	3
1.4、YARN 作业执行流程.....	3

1、YARN

1.1、YARN 概述

YARN（Yet Another Resource Negotiator）

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运行于操作系统之上的应用程序

- 1、yarn 并不清楚用户提交的程序的运行机制
- 2、yarn 只提供运算资源的调度（用户程序向 yarn 申请资源，yarn 就负责分配资源）
- 3、yarn 中的主管角色叫 ResourceManager
- 4、yarn 中具体提供运算资源的角色叫 NodeManager
- 5、这样一来，yarn 其实就与运行的用户程序完全解耦，就意味着 yarn 上可以运行各种类型的分布式运算程序（mapreduce 只是其中的一种），比如 mapreduce、storm 程序，spark 程序，tez
- 6、所以，spark、storm 等运算框架都可以整合在 yarn 上运行，只要他们各自的框架中有符合 yarn 规范的资源请求机制即可
- 7、yarn 就成为一个通用的资源调度平台，从此，企业中以前存在的各种运算集群都可以整合在一个物理集群上，提高资源利用率，方便数据共享

1.2、YARN 的重要概念

1.2.1、ResourceManager

ResourceManager 是基于应用程序对集群资源的需求进行调度的 Yarn 集群主控节点，负责协调和管理整个集群（所有 NodeManager）的资源，响应用户提交的不同类型应用程序的解析，调度，监控等工作。ResourceManager 会为每一个 Application 启动一个 MRAppMaster，并且 MRAppMaster 分散在各个 NodeManager 节点

它主要由两个组件构成：调度器 (Scheduler) 和应用程序管理器 (ApplicationsManager, ASM)

1.2.2、NodeManager

NodeManager 是 YARN 集群当中真正资源的提供者，是真正执行应用程序的容器的提供者，监控应用程序的资源使用情况（CPU，内存，硬盘，网络），并通过心跳向集群资源调度器 ResourceManager 进行汇报。

1.2.3、MRAppMaster

ApplicationMaster 对应一个应用程序，职责是：向资源调度器申请执行任务的资源容器，运行任务，监控整个任务的执行，跟踪整个任务的状态，处理任务失败以异常情况

1.2.4、Container

Container 是一个抽象出来的逻辑资源单位。它封装了一个节点上的 CPU，内存，磁盘，网络等信息，MapReduce 程序的所有 task 都是在一个容器里执行完成的，容器的大小是可以动态调整的

1.2.5、ASM(ApplicationMaster)

这里写错了

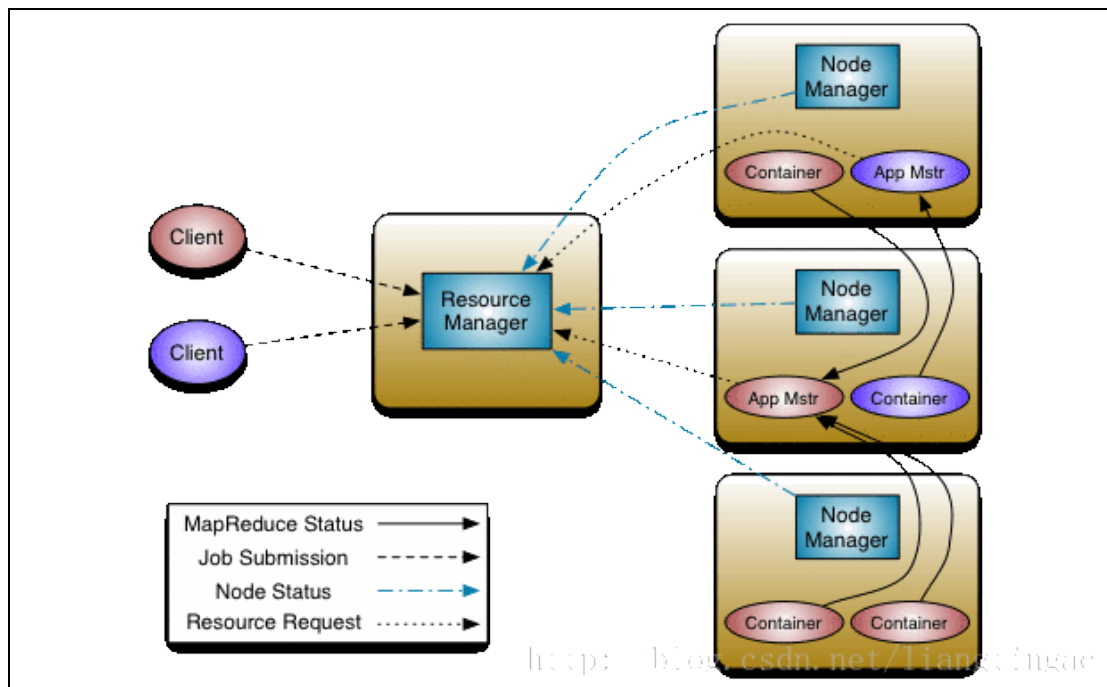
应用程序管理器 ASM 负责管理整个系统中所有应用程序，包括应用程序提交、与调度器协商资源以启动 MRAppMaster、监控 MRAppMaster 运行状态并在失败时重新启动它等

1.2.6、Scheduler

调度器根据应用程序的资源需求进行资源分配，不参与应用程序具体的执行和监控等工作。资源分配的单位就是 Container，调度器是一个可插拔的组件，用户可以根据自己的需求实现自己的调度器。YARN 本身为我们提供了多种直接可用的调度器，比如 FIFO, Fair Scheduler

和 Capacity Scheduler 等

1.3、YARN 架构



1.4、YARN 作业执行流程

YARN 作业执行流程：

- 1、用户向 YARN 中提交应用程序，其中包括 ApplicationMaster 程序，启动 ApplicationMaster 的命令，用户程序等。
- 2、ResourceManager 为该程序分配第一个 Container，并与对应的 NodeManager 通讯，要求它在这个 Container 中启动应用程序 MRAppMaster。
- 3、MRAppMaster 首先向 ResourceManager 注册，这样用户可以直接通过 ResourceManager 查看应用程序的运行状态，然后将为各个任务申请资源，并监控它的运行状态，直到运行结束，重复 4 到 7 的步骤。
- 4、MRAppMaster 采用轮询的方式通过 RPC 协议向 ResourceManager 申请和领取资源。5、一旦 MRAppMaster 申请到资源后，便与对应的 NodeManager 通讯，要求它启动任务。
- 6、NodeManager 为任务设置好运行环境（包括环境变量、JAR 包、二进制程序等）后，将任务启动命令写到一个脚本中，并通过运行该脚本启动任务。
- 7、各个任务通过某个 RPC 协议向 MRAppMaster 汇报自己的状态和进度，以让 MRAppMaster 随时掌握各个任务的运行状态，从而可以在任务败的时候重新启动任务
- 8、应用程序运行完成后，AM(MRAppMaster)向RM(ResourceManager) 注销并关闭自己。。