Yarn

Apache Hadoop YARN （Yet Another Resource Negotiator，另一种资源协调者）是一种新的 Hadoop 资源管理器，它是一个通用资源管理系统，可为上层应用提供统一的资源管理和调度，它的引入为集群在利用率、资源统一管理和数据共享等方面带来了巨大好处。

YARN的核心思想

将JobTracker和TaskTacker进行分离，它由下面几大构成组件：

a. 一个全局的资源管理器 ResourceManager

b.ResourceManager的每个节点代理 NodeManager

c. 表示每个应用的 ApplicationMaster

d. 每一个ApplicationMaster拥有多个Container在NodeManager上运行

Yarn的优点

大大减小了 JobTracker（也就是现在的 ResourceManager）的资源消耗，并且让监测每一个 Job 子任务 (tasks) 状态的程序分布式化了，更安全、更优美。

在新的 Yarn 中，ApplicationMaster 是一个可变更的部分，用户可以对不同的编程模型写自己的 AppMst，让更多类型的编程模型能够跑在 Hadoop 集群中，可以参考 hadoop Yarn 官方配置模板中的 mapred-site.xml 配置。

对于资源的表示以内存为单位 ( 在目前版本的 Yarn 中，没有考虑 cpu 的占用 )，比之前以剩余 slot 数目更合理。

老的框架中，JobTracker 一个很大的负担就是监控 job 下的 tasks 的运行状况，现在，这个部分就扔给 ApplicationMaster 做了，而 ResourceManager 中有一个模块叫做 ApplicationsMasters( 注意不是 ApplicationMaster)，它是监测 ApplicationMaster 的运行状况，如果出问题，会将其在其他机器上重启。

Container 是 Yarn 为了将来作资源隔离而提出的一个框架。这一点应该借鉴了 Mesos 的工作，目前是一个框架，仅仅提供 java 虚拟机内存的隔离,hadoop 团队的设计思路应该后续能支持更多的资源调度和控制 , 既然资源表示成内存量，那就没有了之前的 map slot/reduce slot 分开造成集群资源闲置的尴尬情况。

## YARN的主要架构

### ResourceManager（RM）

### ApplicationMaster（AM）

### NodeManager（NM）

### Container

集中式架构

集中式调度器(Monolithic Scheduler)的特点是，资源的调度和应用程序的管理功能全部放到一个进程中完成，开源界典型的代表是MRv1 JobTracker的实现。这样设计的缺点很明显，扩展性差：首先，集群规模受限；其次，新的调度策略难以融入到现有代码中，比如之前仅支持MapReduce作业，现在要支持流式作业，而将流式作业的调度策略嵌入到中央调度其中是一项很难的工作。

双层调度架构

为了克服集中式调度器的不足，双层调度器是一种很容易被想到的解决之道，它可看作是一种分而治之的机制或者是策略下放机制：双层调度器仍保留一个经简化的集中式资源调度器，但具体任务相关的调度策略则下放到各个应用程序调度器完成。这种调度器的典型代表是Mesos。Mesos调度器由两部分组成，分别是资源调度器和框架(应用程序)调度器,其中，资源调度器负责将集群中的资源分配给各个框架(应用程序),而框架(应用程序)调度器负责将资源进一步分配给内部的各个任务，用户很容易将一种框架或者系统接入Mesos.

双层调度器的特点是：各个框架调度器并不知道整个集群资源使用情况，只是被动地接受资源；资源调度器仅将可用的资源推送给各个框架，而由框架自己选择是使用还是拒绝这些资源；一旦框架接受到新资源，再进一步将资源分配给其内部的任务，进而实现双层调度。然而这种调度器也是有缺点，主要表现在以下两个方面:1.各个框架无法知道整个集群的实时资源使用情况；采用悲观锁，并发粒度小。