



## Hadoop2.x 让你真正明白 yarn

日期：20170218

## 问题导读

**1.hadoop1.x 中 mapreduce 框架与 yarn 有什么共同点？**

**2.它们有什么不同点？**

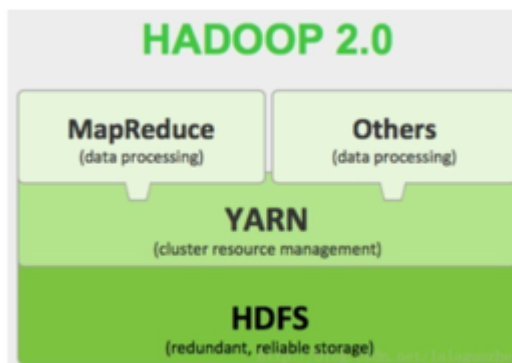
**3.yarn 中有哪些改变？**

**4.yarn 中有哪些术语？**



原文：[about 云日志分析项目准备 6-3-1: Hadoop2.x 让你真正明白 yarn](http://www.aboutyun.com/forum.php?mod=viewthread&tid=20888)

<http://www.aboutyun.com/forum.php?mod=viewthread&tid=20888>



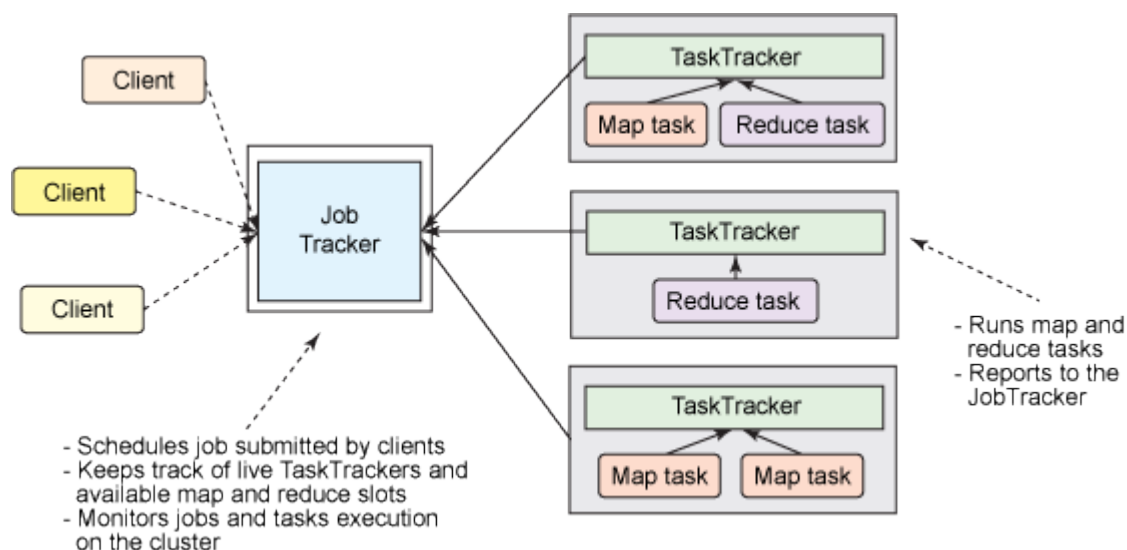
yarn 是比较难懂的一个地方，也是很重要的一个组件，不止 hadoop 使用 yarn，spark，storm 也可以使用 yarn。因此 yarn 的理解是非常重要的。如果刚开始学习，其实还是挺难懂的。因为很多的概念比较抽象。

如果一时理解不了，也是正常的，这时候就需要我们不断的接触和思考，不断的找资料，强化，通过时间，慢慢就能熟记并且理解。下面是个人总结，希望对大家有所帮助。

## 相同点

hadoop2.x 的发展是由于 hadoop1.x 的问题造成的。

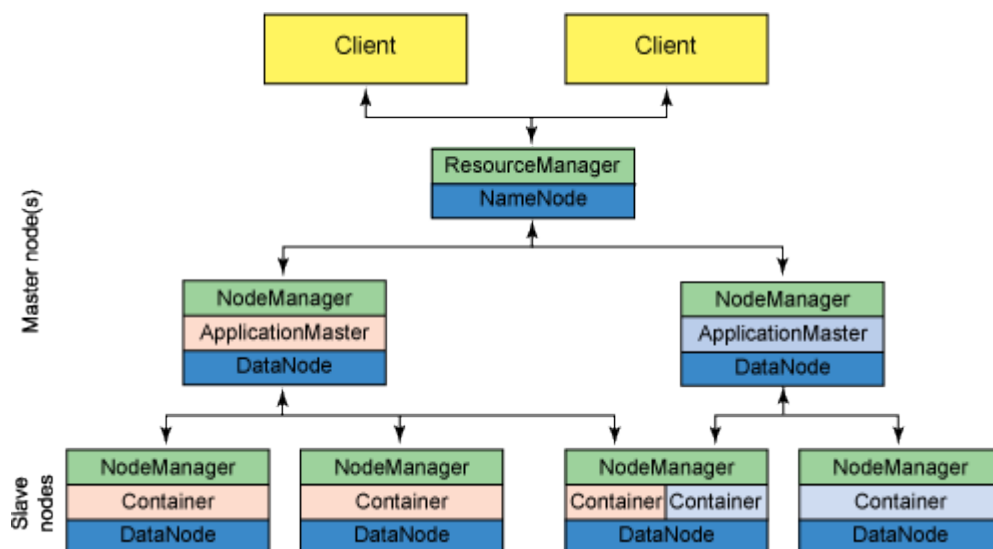
那么是什么问题造成的。比较流行的说法是 jobtracker 的问题，比如单点故障，任务过重。我们知道了除了 Jobtracker，同时还有一个 TaskTracker。我们看下图：



上图中，有一个 JobTracker，多个 TaskTracker。

## Yarn 比较

我们在来看 yarn



我们看到有一个 ResourceManager，多个 NodeManager。

也就是说 hadoop1.x mapreduce 框架与 hadoop2.x yarn，他们的框架相同之处，都是分布式的。

再次总结相同处：

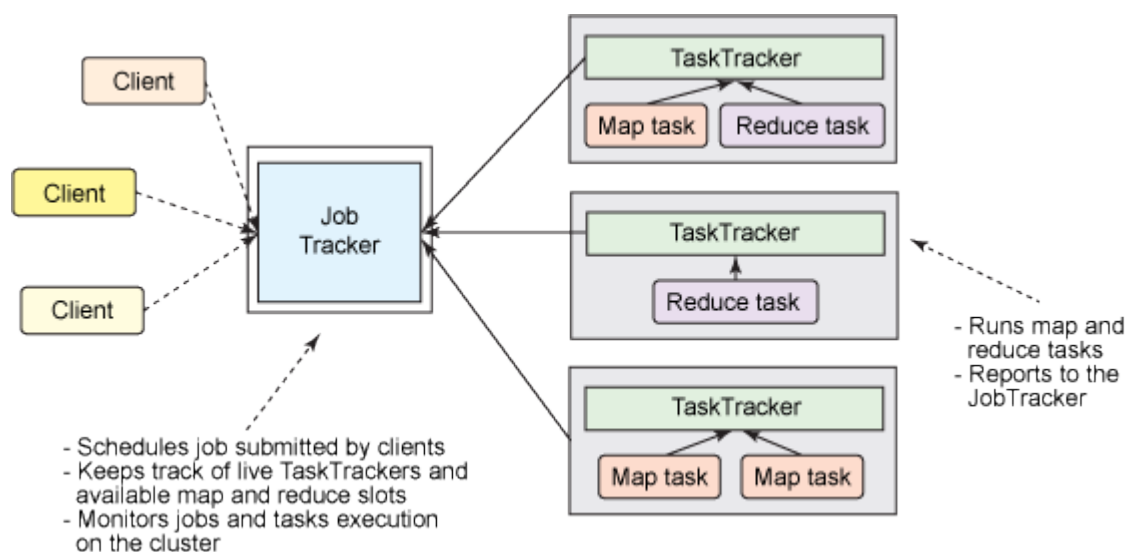
JobTracker 一个，TaskTracker 多个

resourceManager 一个,NodeManager 多个

## 不同点

既然他们框架结构是相同的，那么到底是什么原因，淘汰 JobTracker 机制。

这时候我们就需要看看 JobTracker 到底干了哪些事情。



### 再看上图：JobTracker 概述

JobTracker 其承担的任务有：接受任务、计算资源、分配资源、与 DataNode 进行交流。

在 hadoop 中每个应用程序被表示成一个作业，每个作业又被分成多个任务，JobTracker 的作业控制模块则负责作业的分解和状态监控。

\*最重要的是状态监控：主要包括 TaskTracker 状态监控、作业状态监控和任务状态监控。主要作用：容错和为任务调度提供决策依据。

### TaskTracker 概述

TaskTracker 是 JobTracker 和 Task 之间的桥梁：一方面，从 JobTracker 接收并执行各种命令：运行任务、提交任务、杀死任务等；另一方面，将本地节点上各个任务的状态通过心跳周期性汇报给 JobTracker。

TaskTracker 与 JobTracker 和 Task 之间采用了 RPC 协议进行通信

TaskTracker 的功能：

1.汇报心跳：Tracker 周期性将所有节点上各种信息通过心跳机制汇报给 JobTracker。这些信息包括两部分：

\*机器级别信息：节点健康情况、资源使用情况等。

\*任务级别信息：任务执行进度、任务运行状态等。

2.执行命令：JobTracker 会给 TaskTracker 下达各种命令，主要包括：启动任务(LaunchTaskAction)、提交任务(CommitTaskAction)、杀死任务(KillTaskAction)、杀死作业(KillJobAction)和重新初始化(TaskTrackerReinitAction)。

### 资源 slot 概述

slot 不是 CPU 的 Core，也不是 memory chip，它是一个逻辑概念，一个节点的 slot 的数量用来表示某个节点的资源的容量或者说是能力的大小，因而 slot 是 Hadoop 的资源单位。

hadoop 中什么是 slots

<http://www.aboutyun.com/forum.php?mod=viewthread&tid=7562>

## yarn 详解

Yarn 的基本思想是拆分资源管理的功能，作业调度/监控到单独的守护进程

这里面出现了很多名词：

ResourceManager, NodeManager, ApplicationMaster, Container, 同样下面亦是 yarn 结构图。

**ResourceManager** 是全局的，负责对于系统中的所有资源有最高的支配权。

**ApplicationMaster** 每一个 job 有一个 ApplicationMaster。

**NodeManager**, NodeManager 是基本的计算框架。

NodeManager 是客户端框架负责 containers, 监控他们的资源使用 (cpu, 内存, 磁盘, 网络) 和上报给 ResourceManager/Scheduler.

**ApplicationMaster** 首先它是一个框架库，它的功能官网说的不够系统，大意，由于 NodeManager 执行和监控任务需要资源，所以通过 ApplicationMaster 与 ResourceManager 沟通，获取资源。换句话说，ApplicationMaster 起着中间人的作用。

转换为更专业的术语：**AM** 负责向 **ResourceManager** 索要 **NodeManager** 执行任务所需要的资源容器，更具体来讲是 **ApplicationMaster** 负责从 **Scheduler** 申请资源，以及跟踪这些资源的使用情况以及任务进度的监控。

**ResourceManager** 有两个组件：调度器和应用程序管理器。

调度器 (Scheduler) 是可插拔的，比如有 Fair Scheduler、Capacity Scheduler 等，当然调度器也可以自定义。

更多相关内容：

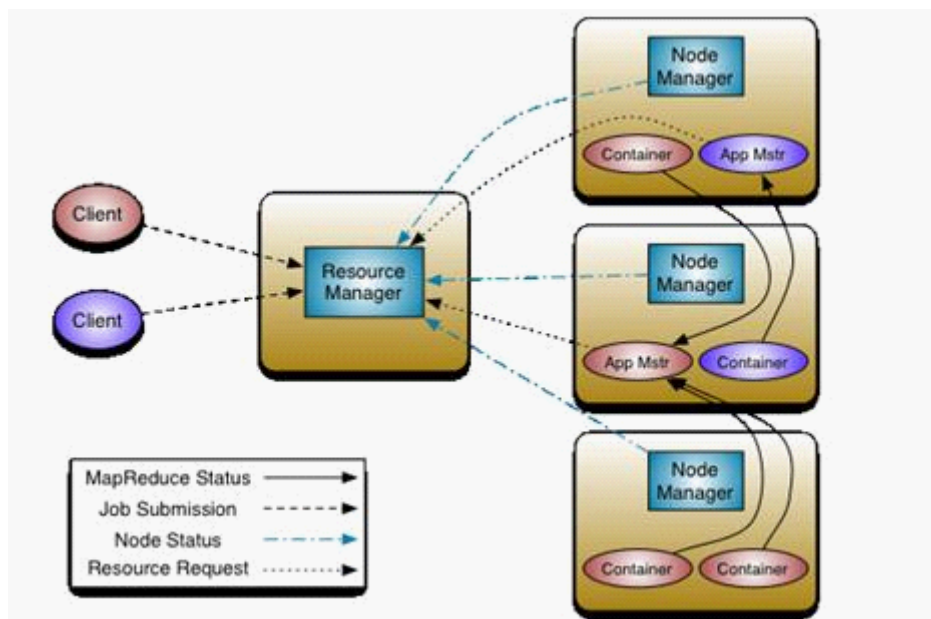
Hadoop YARN 配置参数剖析(4)—Fair Scheduler、Capacity Scheduler 相关参数

<http://www.aboutyun.com/forum.php?mod=viewthread&tid=5864>

### 应用程序管理器

负责接收提交的任务，指定 ApplicationMaster 申请资源 (container) ，协调并提供在 ApplicationMaster 容器失败时的重启功能。

而下图也是官网提供内容，大家可以参考下。



## 总结

为了更好的理解，我们就需要跟 hadoop1.x 比较：

为何要使用 yarn。

我们看到 JobTracker 的功能被分散到各个进程中包括 ResourceManager 和 NodeManager：

比如监控功能，分给了 NodeManager，和 Application Master。

ResourceManager 里面又分为了两个组件：调度器及应用程序管理器。

也就是说 Yarn 重构后，JobTracker 的功能，被分散到了各个进程中。同时由于这些进程可以被单独部署所以这样就大大减轻了单点故障，及压力。

同时我们还看到 Yarn 使用了 Container，而 hadoop1.x 中使用了 slot。slot 存在的缺点比如只能 map 或则 reduce 用。Container 则不存在这个问题。这也是 Yarn 的进步。

参考资源：

hadoop 入门:第六章 YARN 文档概述

<http://www.aboutyun.com/forum.php?mod=viewthread&tid=17338>

<http://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/YARN.html>

<http://www.2cto.com/kf/201311/260826.html>

更多 about 云文档

链接: <https://pan.baidu.com/s/1c2EHU2O> 密码: pr2y

搜索:

www.aboutyun.com



qq7 群: 552029443

捐助

[hadoop 生态系统零基础入门及大数据实战](#)