深入浅出 Hadoop YARN

caolei

Exported on 01/10/2020

Table of Contents

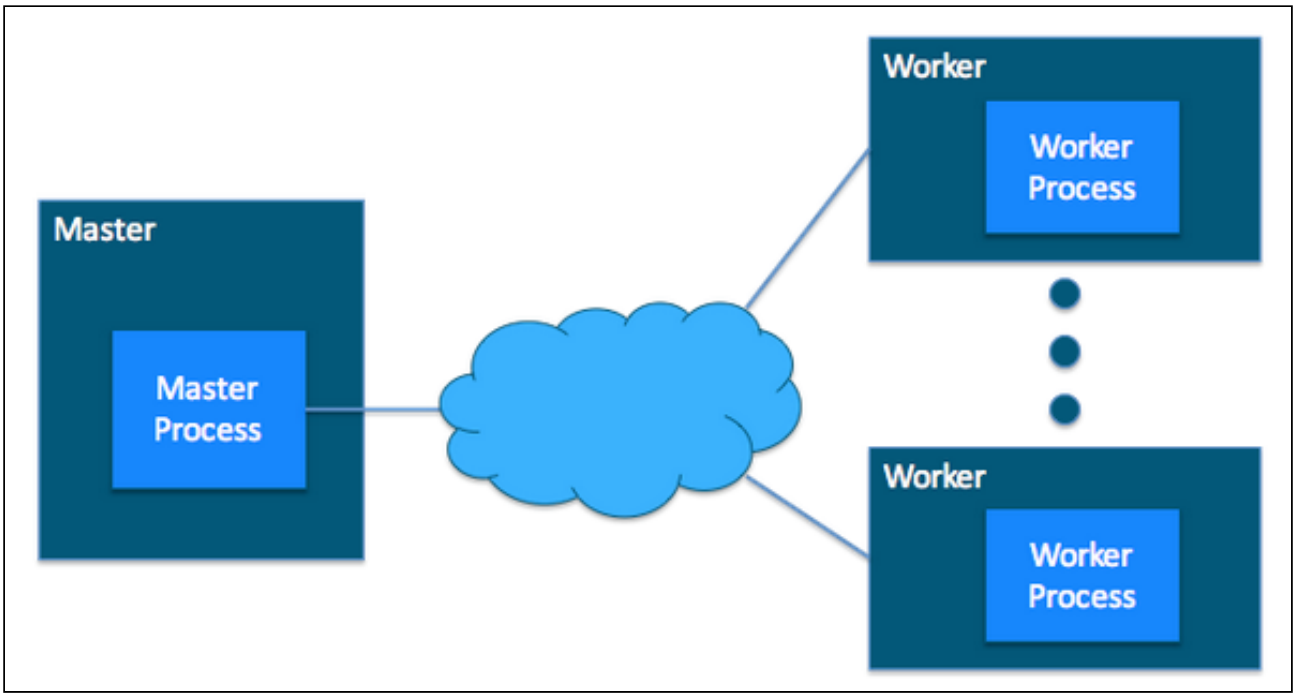
YARN (Yet Another Resource Negotiator)在Apache Hadoop生态系统中处于资源管理层，虽然已经已经发布了多个版本，但是很多用户仍对它有些疑问，比如产生缘由、用途和工作原理等。这篇文章想主要从以下三个方面对YARN进行一个深入浅出的介绍。

- YARN组件的基本介绍
- 阐述MapReduce作业如何集成YARN计算模型(注意:虽然Apache Spark也集成了YARN，但是本文将特别关注MapReduce。有关Spark on YARN的信息，请参阅本文[1])
- 描述YARN调度器的工作原理、调度器配置示例

1 http://blog.cloudera.com/blog/2014/05/apache-spark-resource-management-and-yarn-app-models/

1 Cluster Basics (Master/Worker)

首选先阐明下集群的定义：由高速本地网络连接的两台或两台以上的主机(YARN术语中也称为节点)称为集群。从Hadoop的角度来看，集群中可以有上千台主机。在Hadoop集群中，主要存在两种类型的主机：

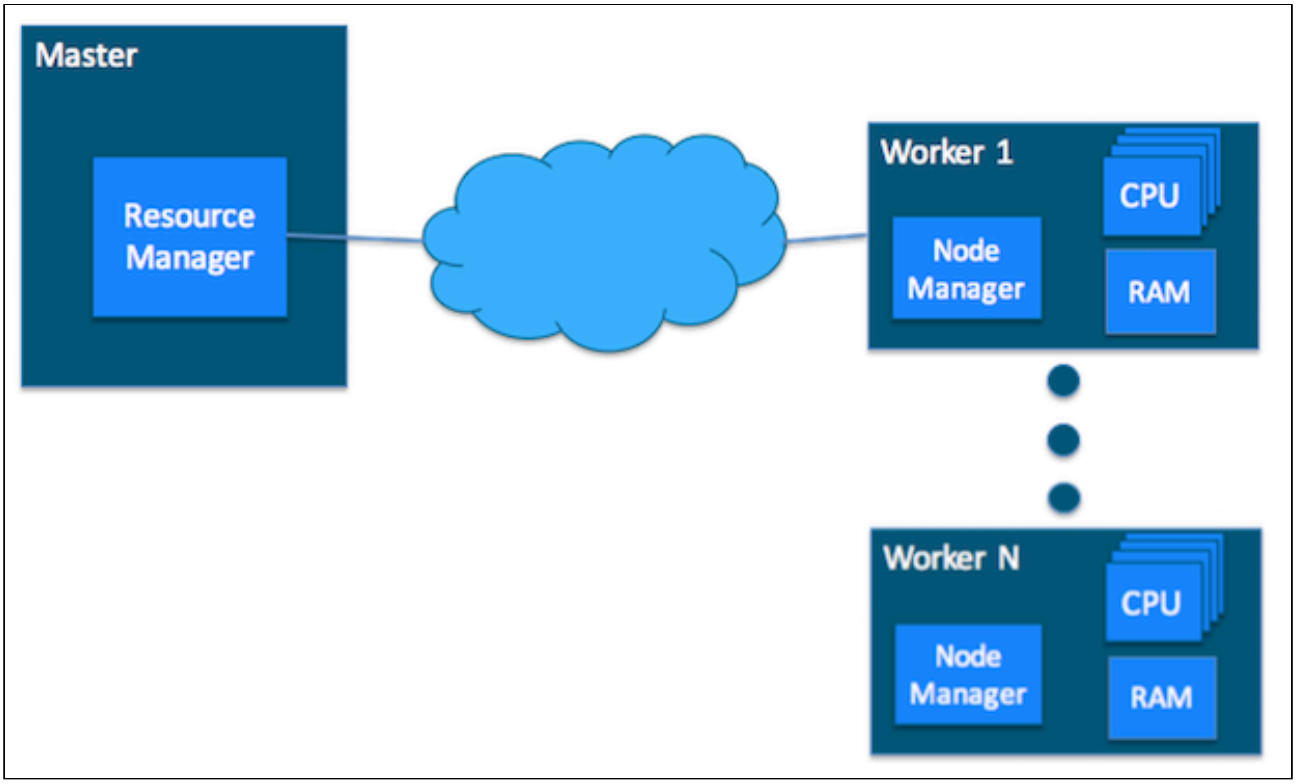


从概念上讲，Master主机是客户端程序的入口点。Master主机将工作任务发送到集群的其余部分即Worker主机上。(在Hadoop中，集群在技术上可以是单个主机。这种设置通常用于调试或简单测试，不推荐用于典型的Hadoop线上环境。)

2 YARN Cluster Basics (Master/ResourceManager, Worker/NodeManager)

在YARN集群中，主要有两种类型的主机：

- ResourceManager是主守护进程，它与客户机通信，跟踪集群上的资源，并通过将任务分配给节点管理器来编排工作。
- NodeManager是一个工作守护进程，它启动并跟踪在工作主机上生成的进程。

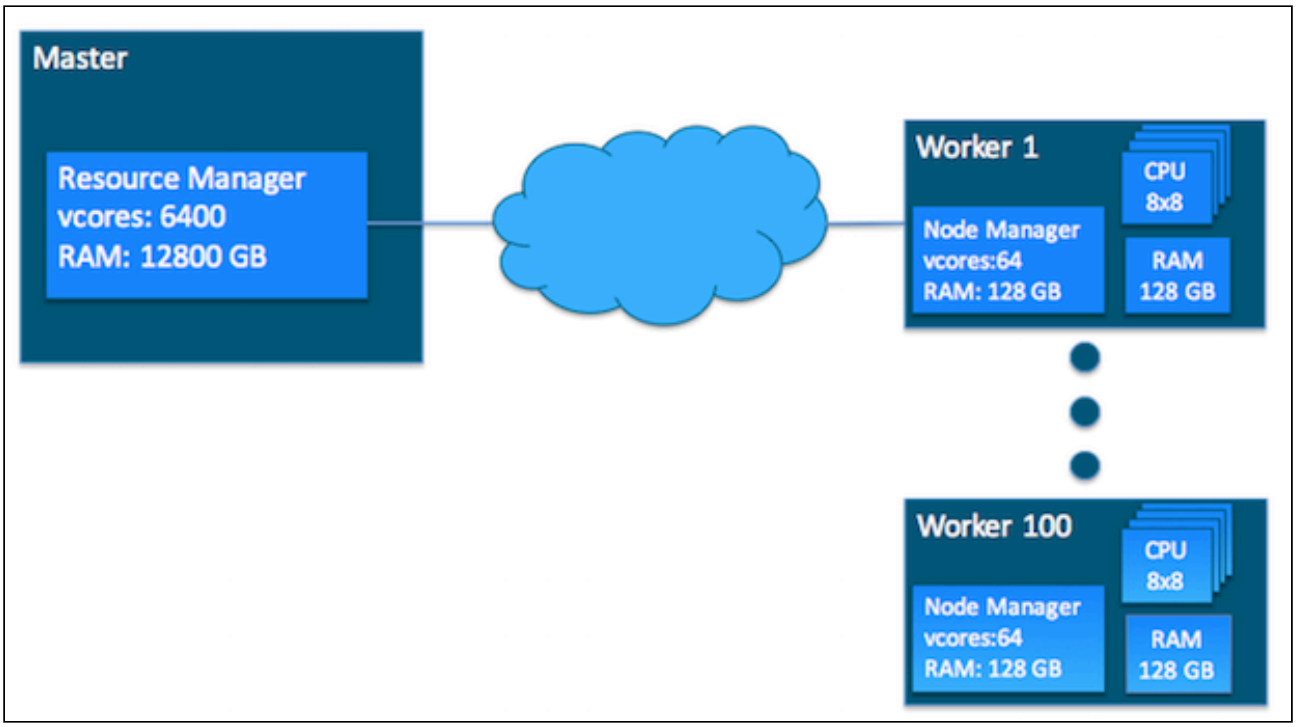


3 YARN Configuration File

YARN 配置文件是一个包含多个可配置属性的XML文件，此文件位于集群中每个节点的固定位置，用于配置ResourceManager和NodeManager。默认情况下，该文件名为yarn-site.xml。关于这个文件中的具体属性配置会在后面的部分中介绍。

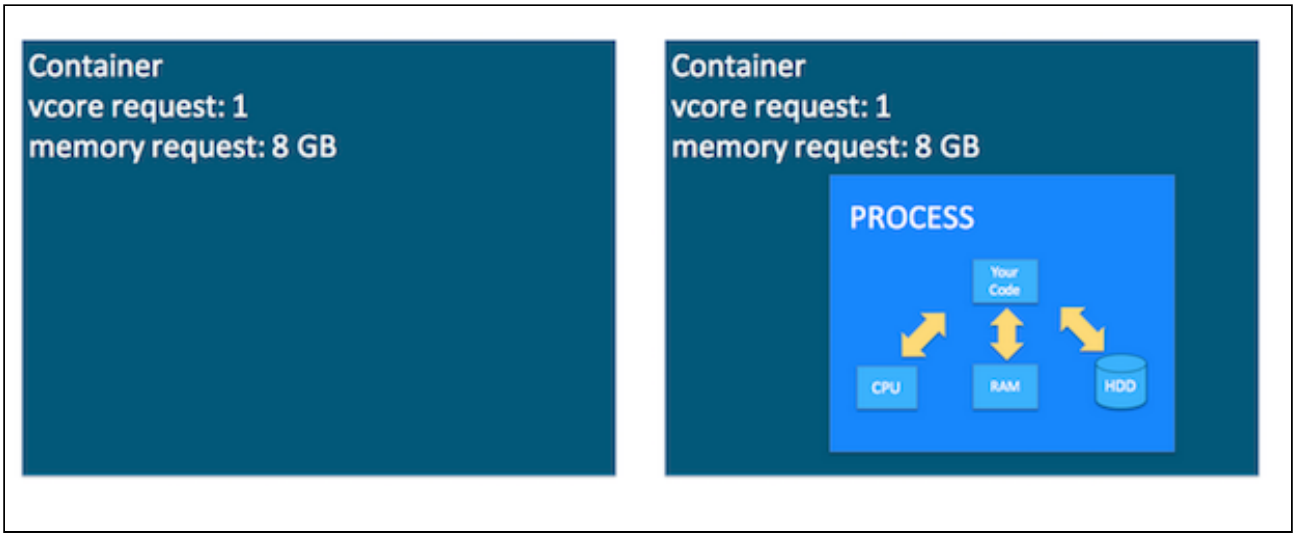
4 YARN Requires a Global View

YARN 目前主要管理两种资源：vcore和memory。每个节点管理器跟踪自己的本地资源，并将其资源配置发送到ResourceManager，后者保存集群可用资源的运行总数。通过跟踪总数，ResourceManager知道如何根据请求分配资源。(Vcore在YARN中有特殊的含义。您可以简单地将其视为“usage share of a CPU core”。如果您的任务是cpu非密集型(有时称为I/ o密集型)，那么可以将vcore与物理内核的比例设置为大于1，以最大限度地使用硬件资源。



5 Containers

容器在YARN中是一个非常重要的概念。容器(Container)这个东西是 Yarn 对资源做的一层抽象。就像我们平时开发过程中，经常需要对底层一些东西进行封装，只提供给上层一个调用接口一样，Yarn 对资源的管理也是用到了这种思想。目前，容器承载的资源包括vcore和内存 (左)。



一旦资源请求被确认授权，NodeManager就会启动一个名为task的进程。上图的右侧显示了作为容器内的流程运行的任务。(后面更详细地介绍YARN如何在特定主机上调度容器)

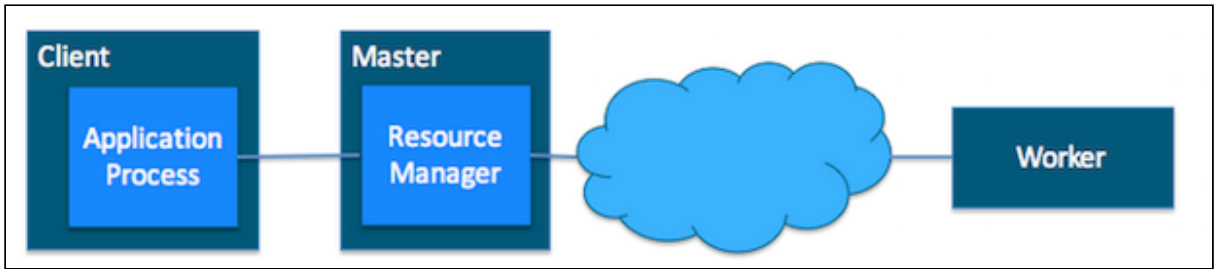
6 YARN Cluster Basics (Running Process/ApplicationMaster)

接下里，我们需要认识YARN的两个新术语：

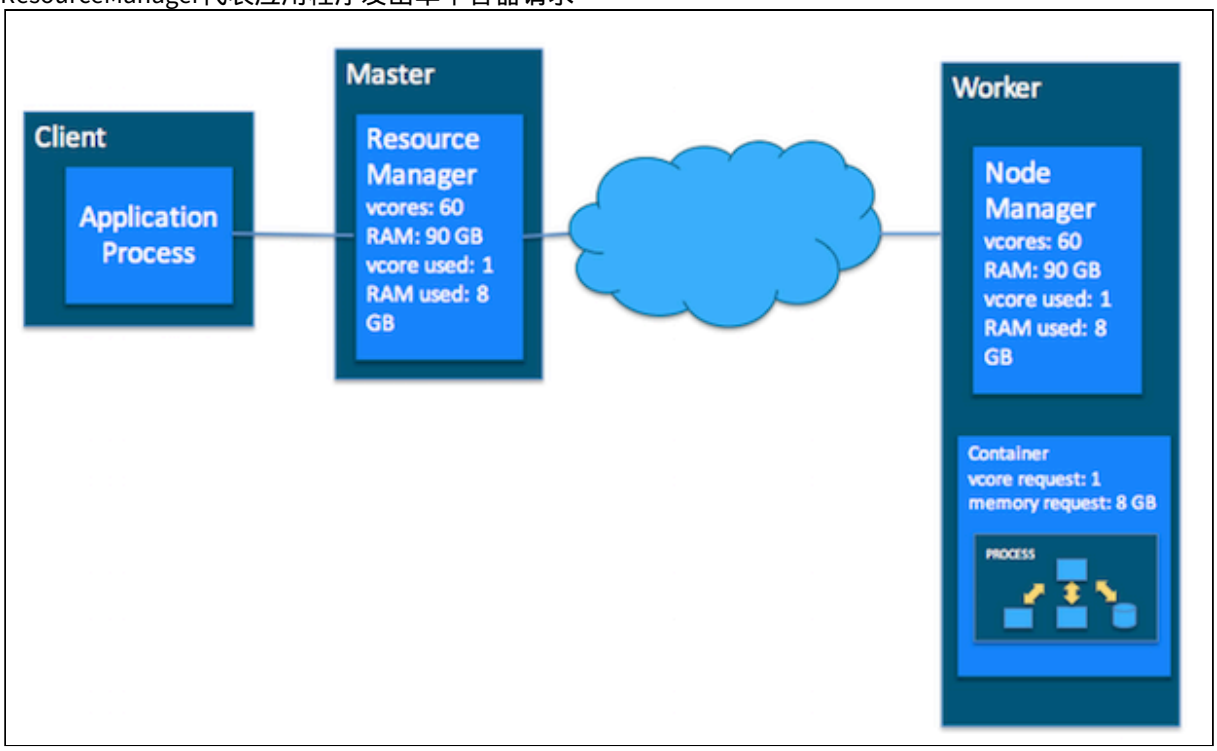
- Application: 由一个或者多个task组成的YARN客户端程序
- ApplicationMaster: 对于每个正在运行的应用程序(Application), 有一段被称为ApplicationMaster的特殊代码可以帮助协调YARN集群上的任务。ApplicationMaster是应用程序启动后运行的第一个进程。

在YARN集群上运行应用程序包括以下步骤:

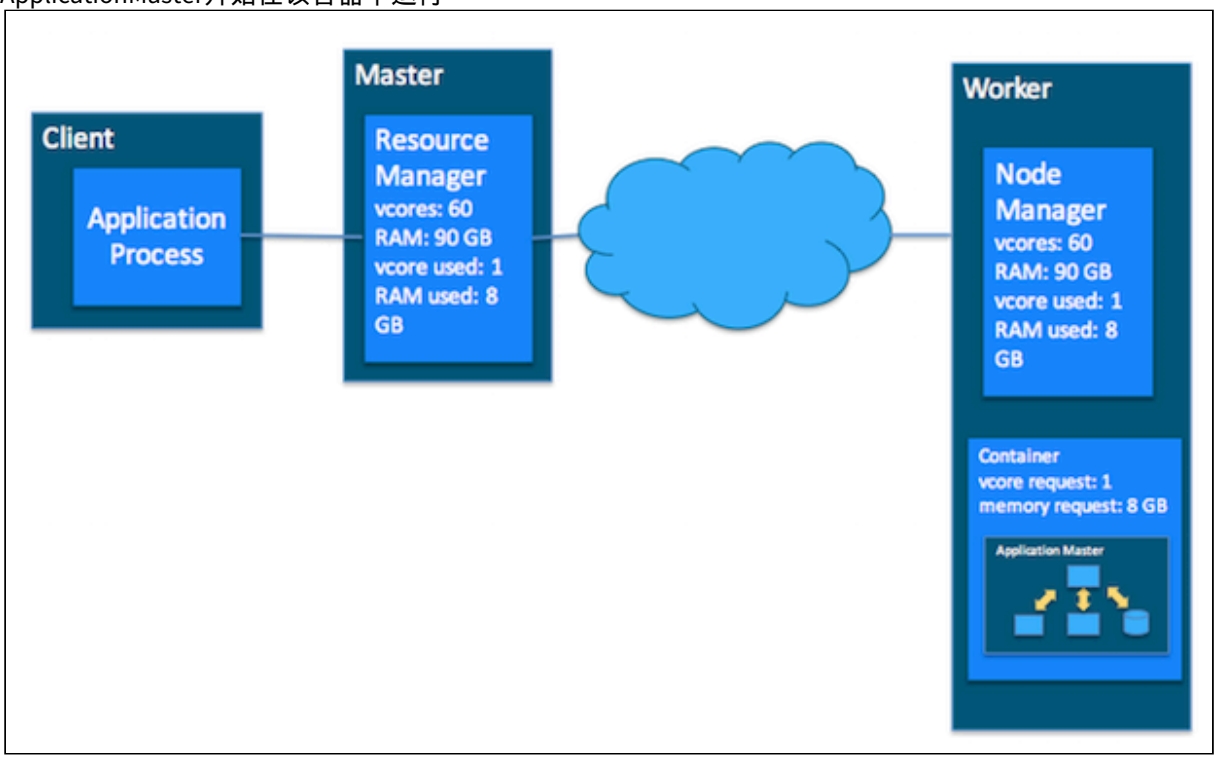
1. 应用程序启动并与集群的ResourceManager对话



2. ResourceManager代表应用程序发出单个容器请求



3. ApplicationMaster开始在该容器中运行

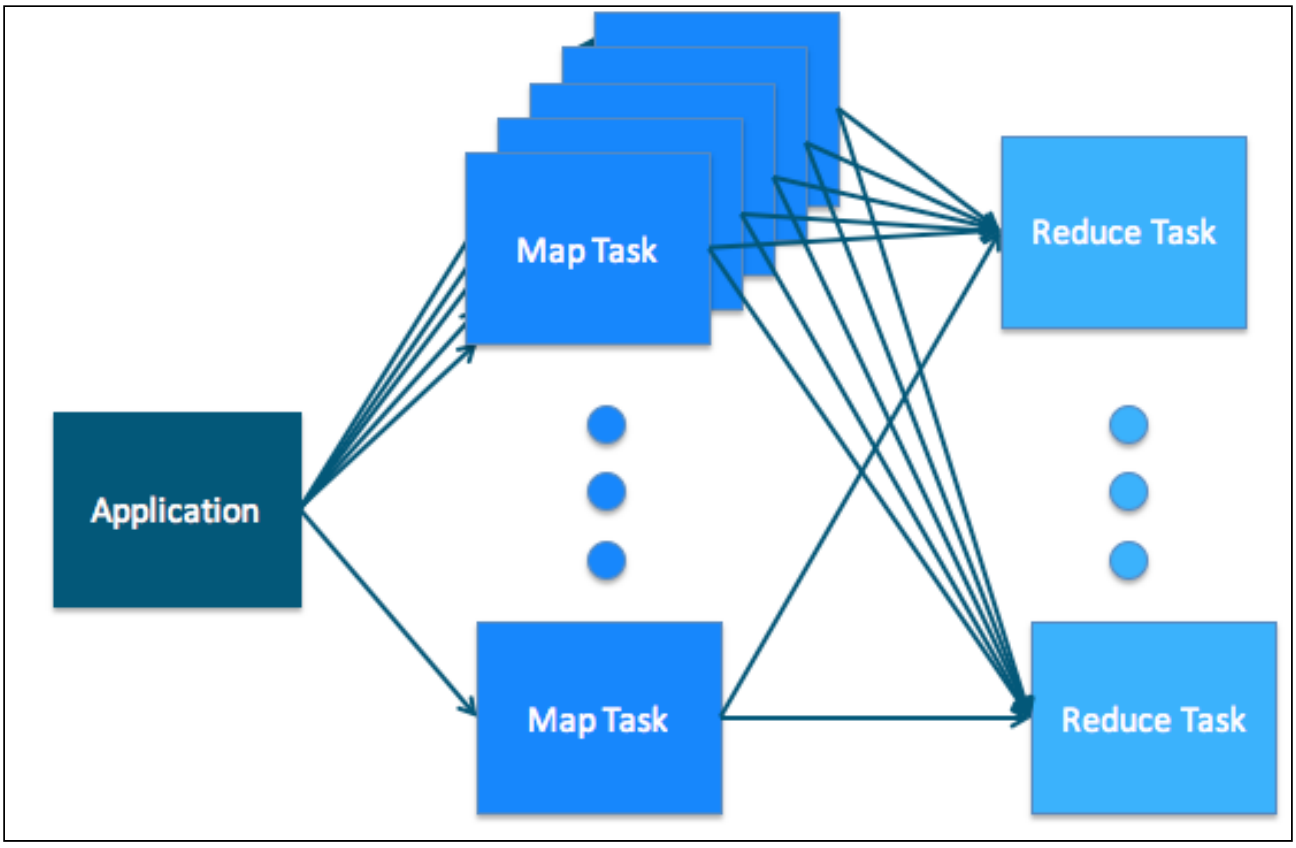


4. ApplicationMaster从ResourceManager请求后续容器，这些容器分配给应用程序运行任务。这些任务与步骤3中分配的ApplicationMaster进行大部分状态通信

5. 一旦所有任务完成，ApplicationMaster就退出。最后一个容器从集群中释放
6. 应用程序客户机退出。(在容器中启动的ApplicationMaster更具体地称为托管AM。不受管理的应用程序管理程序超出YARN的控制)

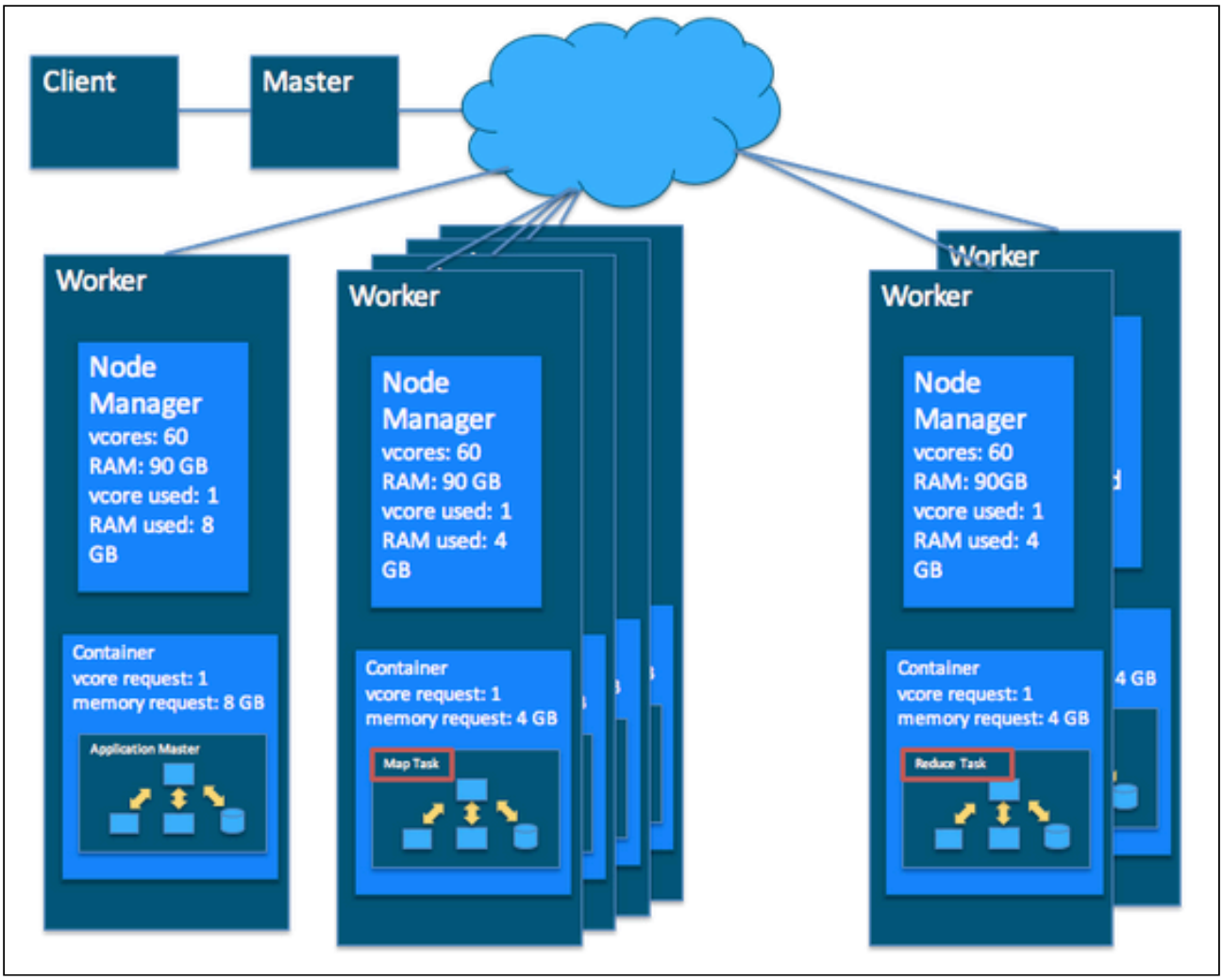
7 MapReduce Basics

在MapReduce中，应用程序由Map任务和Reduce任务组成。Map任务和Reduce任务与YARN任务非常一致。



8 Putting it Together: MapReduce and YARN

下图演示了map任务和reduce任务如何映射到YARN中的任务



在MapReduce应用程序中，有多个map任务，每个任务都运行在集群中某个工作主机上的容器中。类似地，有多个reduce任务，每个任务也运行在工作主机上的容器中。

同时在YARN方面，ResourceManager、NodeManager和ApplicationMaster协同工作，管理集群的资源，确保任务以及相应的应用程序干净地完成。

9 简单总结下

针对上述内容进行简单的总结：

1. 集群由由内部高速网络连接的两个或多个主机组成。Master主机是保留来控制集群其余部分的少量主机。Worker主机是集群中的非Master主机。
2. 在YARN运行的集群中，主进程称为ResourceManager，辅助进程称为NodeManager。
3. YARN的配置文件名为yarn-site.xml。集群中的每个主机上都有一个副本。ResourceManager和NodeManager需要它才能正常运行。YARN跟踪集群上的两个资源:vcore和memory。每个主机上的NodeManager跟踪本地主机的资源，ResourceManager跟踪集群的总资源。
4. YARN中的容器保存集群上的资源。YARN确定集群中主机上的空间，以确定容器的货舱大小。一旦分配了容器，容器就可以使用这些资源。
5. YARN应用包括三个部分:
   应用程序客户端，这是程序在集群上运行的方式。
   一种应用程序控制程序，它使YARN能够代表应用程序执行分配。
   在YARN分配的容器中执行实际工作(在过程中运行)的一个或多个任务。
6. MapReduce应用程序由map任务和reduce任务组成。
7. 在YARN集群中运行的MapReduce应用程序与MapReduce应用程序范例非常相似，但是添加了ApplicationMaster作为YARN需求。