第3章

YARN 资源管理

YARN 是 Hadoop 的资源管理器，负责为应用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在操作系统之上的应用程序。本章主要介绍 YARN 基础架构和 YARN 常用命令。


YARN 概述

3.1 YARN 概述

YARN 是一种通用的 Hadoop 资源管理器和调度平台，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在操作系统之上的应用程序。YARN 是 Hadoop 2.x 版本中的一个新特性。它的出现其实是为了解决 MapReduce 1.x 版本编程框架的不足，提高集群环境下的资源利用率，这些资源包括内存、磁盘、网络和 I/O 等。

YARN 的另一个目标就是拓展 Hadoop，使得它不仅可以支持 MapReduce 计算，还能很方便地支持如 Hive、HBase、Pig、Spark 等应用。因而，在 YARN 中，作业(job)概念被换成了 Application，因为在新的 Hadoop 2.x 中，运行的应用不只是 MapReduce 应用。YARN 这种新的架构设计能够使得各种类型的应用运行在 Hadoop 系统中，并通过 YARN 从系统层面进行统一的管理，各种应用就可以互不干扰地运行在同一个 Hadoop 系统中，共享整个集群资源。

3.2 YARN 基础架构

和 Hadoop 1.x 系列版本相比，在 Hadoop 2.x 系列版本中，加入了 YARN，MapReduce 1.0 升级为 MapReduce 2.0，MapReduce 2.0 架构(又称 YARN 架构)如图 3-1 所示。

YARN 主要由资源管理器(ResourceManager，RM)、节点管理器(NodeManager，NM)、应用管理(ApplicationMaster，App Mstr)和容器(Container)等组件构成。

3.2.1 Container

Container 是 YARN 对计算机计算资源的抽象，是一个逻辑资源单位，它封装了某个节点上的多维度资源，如内存、CPU、磁盘、网络等，当 App Mstr 向资源管理器申请资源时，资源管理器为 App Mstr 返回的资源便是用 Container 表示的。

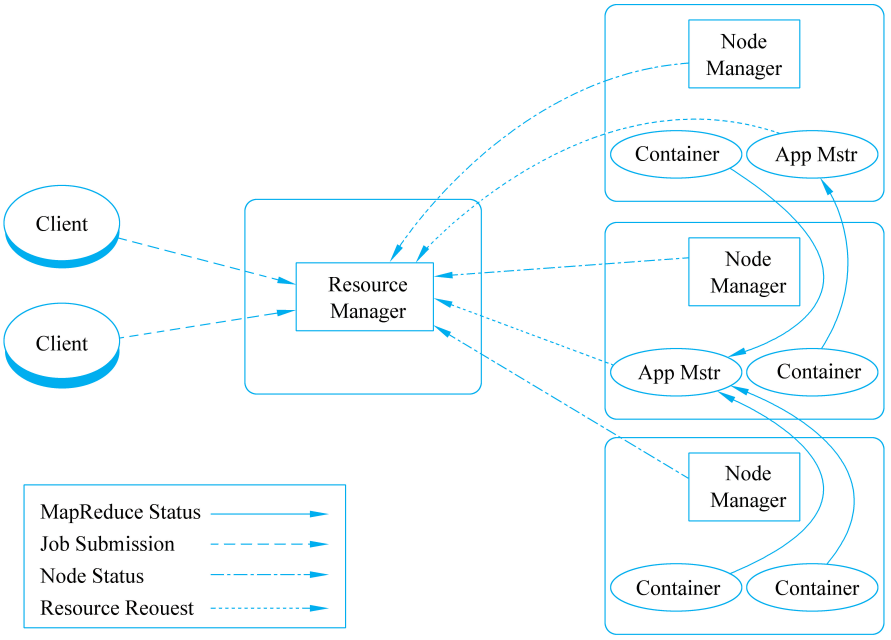

图 3-1　YARN 基础架构

YARN 会为每个任务分配一个 Container，且该任务只能使用该 Container 中约定的资源。

任何一个用户应用必须运行在一个或多个 Container 中，ResourceManager 只负责告诉 App Mstr 哪些 Container 可以用，App Mstr 还需要联系 NodeManager 请求分配具体的 Container。需要注意的是，Container 是一个动态资源划分单位，是根据应用程序的需求动态生成的。

3.2.2　ResourceManager

ResourceManager 负责对各个 NodeManger 上的资源进行统一管理和调度。当用户提交一个应用程序时，需要提供一个用以跟踪和管理这个程序的 App Mstr，它负责向 ResourceManager 申请资源，并要求 NodeManger 启动可以占用一定资源的任务。

ResourceManager 会为每一个 Application 启动一个 App Mstr，App Mstr 分散在各个 NodeManager 节点上。

ResourceManager 主要由两个组件构成：资源调度器（Scheduler）和应用程序管理器（ApplicationManager，ASM）。Scheduler 主要负责协调集群中各个应用的资源分配，负责分配节点上的 Container 资源。

ApplicationManager 主要管理整个系统中所有应用程序，接收应用程序的提交请求，为应用分配第一个 Container 来运行 App Mstr，包括应用程序提交、与 Scheduler 协商资源以启动 App Mstr、监控 App Mstr 运行状态并在失败时重新启动它等。

3.2.3　NodeManager

YARN 集群每个节点都运行一个 NodeManager 进程，NodeManager 的职责如下。

（1）管理节点上的资源和任务。

(2) 接收 ResourceManager 的请求,分配 Container 给应用的某个任务。

(3) 接收并处理来自 App Mstr 的 Container 启动、停止等各种请求,管理每个 Container 的生命周期,监控每个 Container 的资源使用(如内存、CPU 等)情况,并通过心跳消息向 ResourceManager 汇报本节点资源(如 CPU、内存等)的使用情况和 Container 的运行状态。

(4) 执行 YARN 上面应用的一些额外的服务,如 MapReduce 的 shuffle 过程。

当一个计算节点启动时,NodeManager 会向 ResourceManager 进行注册并告知 ResourceManager 自己有多少资源可用。在运行期,通过 NodeManager 和 ResourceManager 协同工作,这些信息会不断被更新并保障整个集群发挥出最佳状态。

3.2.4 Application Master

系统中运行的每个应用,都会对应一个 App Mstr 实例,它的主要功能是向 ResourceManager 申请资源并进一步分配给内部任务,和 NodeManager 协同工作来运行应用的各个任务并跟踪它们的状态及监控各个任务的执行,遇到失败的任务还负责重启它。

3.2.5 Client

Client(客户端)就是提交程序向 YARN 申请资源的地方,可以是 MapReduce、Spark 等。

3.3 YARN 常用命令

Hadoop 的 YARN 命令具有广泛的使用范围,它可以管理大量的 Hadoop 任务,例如,获取和杀死正在运行的应用程序、获取作业和守护程序日志,甚至管理 ResourceManager 的上下线。

3.3.1 YARN 启动与停止

在启动 YARN 之前,先启动 Hadoop 集群。

执行下面命令启动 Hadoop:

```
$cd /usr/local/hadoop
$./sbin/start-dfs.sh
```

执行下面的命令启动 YARN:

```
$./sbin/start-yarn.sh
$jps
2737 SecondaryNameNode
3031 NodeManager
2535 DataNode
2903 ResourceManager
2363 NameNode
```

启动后执行 jps 命令查看运行的进程,发现 ResourceManager 和 NodeManager 两个进程就表明 YARN 正常启动了。

3.3.2　用户命令

用户命令主要包括 application、applicationattempt、classpath、container、jar、logs、node、queue 和 version。下面仅给出几个最常用的命令的用法。

1. application

application 命令的语法格式如下：

```
yarn application [options]
```

常用的命令选项如下。

-appStates ＜States＞：与-list 一起使用，可根据输入的用逗号分隔的应用程序状态列表来过滤应用程序。应用程序状态包括：ALL、NEW、NEW_SAVING、SUBMITTED、ACCEPTED、RUNNING、FINISHED、FAILED 和 KILLED。

-appTypes ＜Types＞：与-list 一起使用，可以根据输入的用逗号分隔的应用程序类型列表来过滤应用程序。

-help：展示所有命令选项的帮助信息。

-list：列出 RM 中的应用程序。

-kill ＜ApplicationId＞：终止应用程序。

-status ＜ApplicationId＞：打印应用程序的状态。

```
$yarn application -list -appStates ALL  #查看状态为 ALL 的 Application 列表
Total number of applications (application-types: [] and states: [NEW, NEW_
SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED]):2
                Application-Id        Application-Name        Application-Type
          User         Queue                 State          Final-State
Progress                           Tracking-URL
application_1663853681771_0002              word count              MAPREDUCE
      hadoop       default              FINISHED            SUCCEEDED
   100%    http://Master:19888/jobhistory/job/job_1663853681771_0002
application_1663853681771_0001              word count              MAPREDUCE
      hadoop       default              FINISHED            SUCCEEDED
   100%    http://Master:19888/jobhistory/job/job_1663853681771_0001
$yarn application -status application_1663853681771_0001 #查看应用的统计报告
Application Report :
    Application-Id : application_1663853681771_0001
    Application-Name : word count
    Application-Type : MAPREDUCE
    User : hadoop
    Queue : default
    Start-Time : 1663854043426
    Finish-Time : 1663854076698
    Progress : 100%
    State : FINISHED
    Final-State : SUCCEEDED
    Tracking-URL : http://Master:19888/jobhistory/job/job_1663853681771_0001
    RPC Port : 32881
    AM Host : Master
    Aggregate Resource Allocation : 110663 MB-seconds, 67 vcore-seconds
    Diagnostics :
```

2. jar

运行 JAR 文件,用户可以将写好的 MapReduce 应用代码打包成 JAR 文件,用这个命令去运行它。执行 jar 命令运行 x.jar 的语法格式如下:

```
yarn jar x.jar [mainClass] 参数...
$cd /usr/local/hadoop
$start-dfs.sh                          #启动 Hadoop
$start-yarn.sh                         #启动 yarn
$hdfs dfs -rm -r output                #如果 output 文件夹存在,先删除
```

本地文件/home/hadoop/Education.txt 中的内容是 Education is not the filling of a pail but the lighting of a fire,将该文件上传到 HDFS 的 input 目录下,命令如下:

```
$hdfs dfs -put /home/hadoop/Education.txt input
```

执行 yarn jar 命令运行系统自带的词频统计应用 wordcount,命令如下:

```
$cd /usr/local/hadoop
$ yarn jar ./share/hadoop/mapreduce/hadoop - mapreduce - examples - 2. 7. 7. jar
wordcount input output
```

查看 HDFS 中 output 目录中的文件中的词频统计结果:

```
$hdfs dfs -cat output/*
Education    1
a    2
but    1
filling    1
fire    1
is    1
lighting    1
not    1
of    2
pail    1
the    2
```

3. applicationattempt

applicationattempt 用来打印应用程序尝试的报告,该命令的语法格式如下:

```
yarn applicationattempt [options]
```

常用的命令选项如下。

help:查看帮助。

-list <Application ID>:获取应用程序尝试的列表。

-status <Application Attempt ID>:打印应用程序尝试的状态。

```
$yarn applicationattempt -help   #查看帮助
usage: applicationattempt
-help
-list <Application ID>
-status <Application Attempt ID>
```

查看应用 application_1663853681771_0001 所有的 attempt 的命令如下:

```
$yarn applicationattempt -list application_1663853681771_0001
22/09/22 22:59:06 INFO client.RMProxy: Connecting to ResourceManager at Master/
192.168.1.13:8032
```

```
Total number of application attempts :1
        ApplicationAttempt-Id                 State
AM-Container-Id                          Tracking-URL
appattempt_1663853681771_0001_000001              FINISHED
container_1663853681771_0001_01_000001
http://Master:8088/proxy/application_1663853681771_0001/
```

查看具体某一个 application attemp 的报告的命令如下：

```
$yarn applicationattempt -status appattempt_1663853681771_0001_000001
Application Attempt Report :
    ApplicationAttempt-Id : appattempt_1663853681771_0001_000001
    State : FINISHED
    AMContainer : container_1663853681771_0001_01_000001
    Tracking-URL : http://Master:8088/proxy/application_1663853681771_0001/
    RPC Port : 32881
    AM Host : Master
    Diagnostics :
```

4. container

container 命令用来打印应用所使用的 Container 的统计信息，语法格式如下：

```
yarn container [options]
```

常用的命令选项如下。

-help：获取使用帮助。

-list ＜Application AttemptID＞：应用程序尝试的 Containers 列表。

-status ＜ContainerID＞：打印 Container 的状态。

```
$yarn container -help      #查看使用帮助
usage: container
-help
-list <Application Attempt ID>
-status <Container ID>
```

查看某一个 application attemp 下所有的 Container 的命令如下：

```
$yarn container -list appattempt_1663853681771_0001_000001
22/09/22 23:09:48 INFO client.RMProxy: Connecting to ResourceManager at Master/
192.168.1.13:8032
Total number of containers :0
Container-Id          Start Time      Finish Time      State      Host        Node
Http Address
```

3.3.3　管理命令

下列这些管理命令对 Hadoop 集群的管理员是非常有用的。

1. daemonlog

daemonlog 命令针对指定的守护进程，获取/设置日志级别，语法格式如下：

```
#打印运行在<host:port>的守护进程的日志级别
yarn daemonlog -getlevel <host:httpport><classname>
#设置运行在<host:port>的守护进程的日志级别
yarn daemonlog -setlevel <host:httpport><classname><level>
```

2. nodemanager

nodemanager 命令启动 NodeManager。

```
$yarn nodemanager
23/01/29 14:53:24 INFO nodemanager.NodeManager: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NodeManager
STARTUP_MSG:   host =Master/127.0.1.1
STARTUP_MSG:   args =[]
STARTUP_MSG:   version =2.7.7
```

3. resourcemanager

resourcemanager 命令用于启动 ResourceManager,语法格式如下:

```
yarn resourcemanager [-format-state-store]
```

参数-format-state-store 的具体形式如下。

-refreshQueues:重载队列的 ACL,状态和调度器特定的属性,ResourceManager 将重载 mapred-queues 配置文件。

-refreshUserToGroupsMappings:刷新用户到组的映射。

-refreshSuperUserGroupsConfiguration:刷新用户组的配置。

-refreshAdminAcls:刷新 ResourceManager 的 ACL 管理。

-refreshServiceAclResourceManager:重载服务级别的授权文件。

-getGroups [username]:获取指定用户所属的组。

```
$yarn resourcemanager -getGroups hadoop
23/01/29 15:06:48 INFO resourcemanager.ResourceManager: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting ResourceManager
STARTUP_MSG:   host =Master/127.0.1.1
STARTUP_MSG:   args =[-getGroups, hadoop]
```

3.4 习 题

1. YARN 是什么?
2. 什么是 YARN 的 ResourceManager?
3. 什么是 YARN 的 NodeManager?
4. YARN 运行应用程序的命令是什么?