## 5.5 资源调度器

目前，Hadoop作业调度器主要有三种：FIFO Scheduler、Capacity Scheduler和Fair Scheduler。Hadoop2.7.2默认的资源调度器是Capacity Scheduler。

具体设置详见：yarn-default.xml文件

<property>

<description>The class to use as the resource scheduler.</description>

<name>yarn.resourcemanager.scheduler.class</name>

<value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler</value>

</property>

1．先进先出调度器（FIFO），如图4-27所示

 图4-27 FIFO调度器

Hadoop最初设计目的是支持大数据批处理作业，如日志挖掘、Web索引等作业，

为此，Hadoop仅提供了一个非常简单的调度机制：FIFO，即先来先服务，在该调度机制下，所有作业被统一提交到一个队列中，Hadoop按照提交顺序依次运行这些作业。

但随着Hadoop的普及，单个Hadoop集群的用户量越来越大，不同用户提交的应用程序往往具有不同的服务质量要求，典型的应用有以下几种：

批处理作业：这种作业往往耗时较长，对时间完成一般没有严格要求，如数据挖掘、机器学习等方面的应用程序。

交互式作业：这种作业期望能及时返回结果，如SQL查询（Hive）等。

生产性作业：这种作业要求有一定量的资源保证，如统计值计算、垃圾数据分析等。

此外，这些应用程序对硬件资源需求量也是不同的，如过滤、统计类作业一般为CPU密集型作业，而数据挖掘、机器学习作业一般为I/O密集型作业。因此，简单的FIFO调度策略不仅不能满足多样化需求，也不能充分利用硬件资源。

2．容量调度器（Capacity Scheduler），如图4-28所示

 图4-28容量调度器

Capacity Scheduler Capacity Scheduler 是Yahoo开发的多用户调度器，它以队列为单位划分资源，每个队列可设定一定比例的资源最低保证和使用上限，同时，每个用户也可设定一定的资源使用上限以防止资源滥用。而当一个队列的资源有剩余时，可暂时将剩余资源共享给其他队列。

总之，Capacity Scheduler 主要有以下几个特点：

①容量保证。管理员可为每个队列设置资源最低保证和资源使用上限，而所有提交到该队列的应用程序共享这些资源。

②灵活性，如果一个队列中的资源有剩余，可以暂时共享给那些需要资源的队列，而一旦该队列有新的应用程序提交，则其他队列释放的资源会归还给该队列。这种资源灵活分配的方式可明显提高资源利用率。

③多重租赁。支持多用户共享集群和多应用程序同时运行。为防止单个应用程序、用户或者队列独占集群中的资源，管理员可为之增加多重约束（比如单个应用程序同时运行的任务数等）。

④安全保证。每个队列有严格的ACL列表规定它的访问用户，每个用户可指定哪些用户允许查看自己应用程序的运行状态或者控制应用程序（比如杀死应用程序）。此外，管理员可指定队列管理员和集群系统管理员。

⑤动态更新配置文件。管理员可根据需要动态修改各种配置参数，以实现在线集群管理。

3．公平调度器（Fair Scheduler），如图4-29所示



图4-29公平调度器

Fair Scheduler Fair Schedulere是Facebook开发的多用户调度器。

公平调度器的目的是让所有的作业随着时间的推移，都能平均地获取等同的共享资源！当一个作业在运行时，它会使用整个集群但是如果有其他作业提交上来，系统会将空闲的资源分配给新的作业！每个任务大致上会获取平等数量的资源！和传统的调度策略不同的是

它会让小的任务在合理的时间完成，同时不会让需要长时间运行的耗费大量资源的应用挨饿！

同Capacity Scheduler类似，它以队列为单位划分资源，每个队列可设定一定比例的资源最低保证和使用上限，同时，每个用户也可设定一定的资源使用上限以防止资源滥用；当一个队列的资源有剩余时，可暂时将剩余资源共享给其他队列。

当然，Fair Scheduler也存在很多与Capacity Scheduler不同之处，这主要体现在以下几个方面：

①资源公平共享。在每个队列中，Fair Scheduler 可选择按照FIFO、Fair或DRF策略为应用程序分配资源。其中，Fair 策略(默认)是一种基于最大最小公平算法实现的资源多路复用方式，默认情况下，每个队列内部采用该方式分配资源。这意味着，如果一个队列中有两个应用程序同时运行，则每个应用程序可得到1/2的资源；如果三个应用程序同时运行，则每个应用程序可得到1/3的资源。

②支持资源抢占。当某个队列中有剩余资源时，调度器会将这些资源共享给其他队列，而当该队列中有新的应用程序提交时，调度器要为它回收资源。为了尽可能降低不必要的计算浪费，调度器采用了先等待再强制回收的策略，即如果等待一段时间后尚有未归还的资源，则会进行资源抢占：从那些超额使用资源的队列中杀死一部分任务，进而释放资源。

③负载均衡。Fair Scheduler提供了一个基于任务数目的负载均衡机制，该机制尽可能将系统中的任务均匀分配到各个节点上。此外，用户也可以根据自己的需要设计负载均衡机制。

④调度策略配置灵活。Fair Scheduler允许管理员为每个队列单独设置调度策略（当前支持FIFO、Fair或DRF三种）。

⑤提高小应用程序响应时间。由于采用了最大最小公平算法，小作业可以快速获取资源并运行完成