yarn

W.J.Z

2019.4.18

1 yarn 运行机制

YARN 是 Hadoop 的集群资源管理系统，通过两类长期运行的守护进程提供自己的核心服务：resource manager 和 node manager。node manager 运行在集群所有节点，负责启动和监控容器，容器是用于执行特定应用程序的进程。

1.1 资源请求

YARN 有一个灵活的资源请求模型，当请求多个容器时，可以指定每个容器需要的计算机资源数量（内存和 CPU 数量），还可以指定对容器的本地限制要求。本地限制可用于申请指定节点或机架、或集群中任何位置的容器。YARN 应用可以在运行中的任意时刻提出资源申请。

通常情况下，启动一个容器处理 HDFS 数据块时，应用将会向这样的节点提出申请：储存该数据块三个副本的节点，或是储存这些副本的机架中的一个节点。如果都申请失败，则申请集群中任意一个节点。

1.2 应用生命周期

YARN 应用的生命周期差异性很大，可分为下面三个模型：(1) 一个用户作业对应一个应用，MapReduce 采取这种方式。(2) 作业的每个工作流和每个用户对话对应一个应用，容器可以在作业间重用且可能缓存中间数据，Spark 采用该模式。(3) 多个用户共享一个长期运行的应用，这种应用通常作为一种协调者的角色在运行。

2 Yarn 调度

理想情况下，YARN 应用发出的资源请求应立刻给与满足，但现实情况下资源是有限的，在一个繁忙的集群上，一个应用经常需要等待才能得到所需的资源，YARN 提供多种调度器和配置策略龚我们选择的原因。

2.1 调度选项

YARN 有三种调度器可用：FIFO 调度器、容量调度器、公平调度器。

1. FIFO Scheduler: 将应用放在一个队列中，按照提交的顺序运行应用。该调度器简单易懂，不需要任何配置，但不适用于共享集群。

2. Capacity Scheduler: 预留一部分资源并设立一个独立的专门队列负责小作业，由于牺牲了整个集群的利用率，大作业执行的时间要更长。

3. Fair Scheduler: 调度器会在所有运行的作业之间动态平衡资源。

2.2 容量调度器配置

容量调度器允许多个组织共享一个 Hadoop 集群，每个组织可以分配到全部集群资源的一部分。每个组织被分配一个专门的队列，每个队列被配置为可以使用一定的集群资源。队列可以进一步按层次划分，这样每个组织内的不同用户能共享该组织队列所分配的资源。在一个队列内，使用 FIFO 调度策略对应用进行调度。

容量调度器配置文件:capacity-cheduler.xml,对特定队列进行配置时:yarn-scheduler.capacity.<queue-path>.<sub-property> 进行设置，<queue-path> 表示队列的层次路径 (用圆点隔开)。root 队列下 prod 和 dev 两个队列，dev 下有 eng 和 science 两个队列，配置条件如下：

```
<?xml version="1.0">
<configuration>
    <property>
        <name>yarn.scheduler.capacity.root.queues</name>
        <value>prod,dev</value>
    </property>
    <property>
        <name>yarn.scheduler.capacity.root.dev.queues</name>
        <value>eng,science</value>
    </property>
    <property>
        <name>yarn.shceduler.capacity.root.prod.capacity</name>
        <value>40</value>
    </property>
    <property>
        <name>yarn.schedul;er.capacity.root.dev.capacity</name>
        <value>60</value>
    </property>
    <property>
        <name>yarn.shceduler.capacity.root.dev.maxinum-capacity</name>
        <value>75</value>
    </property>
    <property>
        <name>yarn.scheduler.capoacity.root.dev.eng.capacity</name>
        <value>50</value>
    </property>
    <property>
        <name>yarn.shceduler/capacity.root.dev.science.capacity</name>
        <value>50</value>
    </propety>
</configuration>
```

将应用放置那个队列取决于应用本身:在 MapReduce 中,可以通过设置属性 Mapreduce.job.queuename 来指定要使用的队列。如果队列不存在，在提交时会发送错误。如果未指定队列，name 应用将放在一个名为 default 的默认队列中。

2.3 公平调度器配置

2.3.1 启用公平调度器

hadoop 默认使用容量调度器,启用公平调度器需要将 yarn-site.xml 文件中的 yarn-resourcemanager.scheduler.class 设置为公平调度器的完全限定名:org.aoachehadoop.yarn,server.resourcemanager.shceduler.fair.FairSched

2.3.2 队列配置

公平调度器通过对 fair-scheduler.xml 文件进行配置:

```
<?xml version="1.0">
<allocations>
<defaultQueueSchedulingPolicy>fair</defaultQueueSchedulingPolicy>
<queue name="prod">
    <weight>40</weight>
    <schedulingPolicy>fifo</schedulingPolicy>
</queue>

<queue name="dev">
    <weight>60</weight>
    <queue name="eng" />
    <queue name="science" />
</queue>

<queuePlacementPolicy>
    <rule name="specified" create="false" />
    <rule name="primaryGroup" create="false" />
    <rule name="default" queue="dev.eng" />
</queruePlacementPolicy>
```

每个队列可以有不同的调度策略，队列的默认调度策略可以通过顶层元素 defaultQueueScheduling-Policy 进行设置，如果省略默认使用公平调度。队列的调度策略可以被该队列的 schedulingPolicy 元素指定的策略覆盖。

2.3.3 队列配置

queuePlacementPolicy 元素包含了一个规则列表，每条规则会被依次尝试指导匹配成功。如果没用明确定义队列，则按照用户名为队列名进行创建。

```
<queuePlacementPolicy>
    <rule name="specified" />
    <rule name="user" />
</queuePlacemenPolicy>
```

2.3.4 延迟调度

所有的 YARN 调度器都已本地请求为重，为了增加集群的工作效率，可以等待一小段时间进行分配容器。

YARN 中的每个节点管理器周期性的向资源管理器发送心跳请求，心跳中携带了节点管理器中正运行的容器，新容器可用资源等信息。对于需要运行一个容器的应用来说，每个心跳就是一个潜在的调度机会。

使用延迟调度器时，调度器不会简单使用它收到的用第一个调度机会，而是等待设定的最大数目的调度机会发生，然后才放松本地性限制并接受下一个调度机会。

对于容量调度器：通过设置 yarn.scheduler.capacity.node-locality-delay 来配置延迟调度。配置为正整数，表示调度器在放松节点限制、改为匹配同一机架上的其他节点之前，准备错过的调度机会的数量。

对于公平调度器：yarn.scheduler.fair-locality.threshold.node 设置为 0.5, 表示调度器在接受统一机架中的其他节点之间,将一直等待直到集群中的一半节点都已经给过调度任务。yarn.scheduler.fair.locality.threshold.track 表示在接受另一个机架替代所申请的机架之前需要等待的时长阈值。

2.3.5 主导资源公平性

对于多种资源调度时，如何衡量调度公平性？Yarn 采用 DRF：观察每个用户的主导资源，并将其作为对集群资源使用的一个度量 (对比使用占比最大的资源)。默认情况下，YARN 只考虑内存，不考虑 CPU。启用 DRF 对容量调度器：将 capacity-scheduler.xml 文件中的 org.apache.hadoop.yarn.util.resource.DominantResourceCalculator 设置为 yarn.scheduler.capacity.resource-calculator 即可。对公平调度器设置分配文件中的顶层元素;defaultQueueSchedulingPolicy 为 drf 即可。