



链滴

# Spark2.x 集群搭建与参数详解

作者: [rxz](#)

原文链接: <https://ld246.com/article/1500286920223>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在前面的[Spark发展历程和基本概念](#)中介绍了Spark的一些基本概念，熟悉了这些基本概念对于集群搭建是很有必要的。我们可以了解到每个参数配置的作用是什么。这里将详细介绍Spark集群搭建以及ml参数配置。Spark的集群搭建分为分布式与伪分布式，分布式主要是与hadoop Yarn集群配合使用伪分布式主要是单独使用作为测试。

## Spark完全分布式搭建

由于Hadoop和Spark集群占用的内存较高，这种完全分布式集群的搭建对于跑应用来说太吃力，如有服务器可以尝试，这里采用虚拟机方式实验，详情请看我的博客园：[Hadoop2.8与spark2.1集群搭建](#)

这里是一个早前的实验，但是确实没有怎么使用，因为当时对参数没有做优化，也没有理解其中个参的含义。所以跑应用很吃力，而且会非常卡。下面直接介绍其中的各个参数。

## Spark配置参数详解

上一步配置Spark集群的时候我们使用了三个节点：一个Master，两个Worker，我们启动脚本的时候就会启动三个守护进程，分别名为Master, worker, worker。

**conf/slaves.xml**配置:在这里我们指定哪些机器作为Worker节点。伪分布式的话不要配置，因为Master, worker会在一个节点上启动。

**conf/spark-env.sh**配置：核心配置文件，配置的是spark应用的运行环境，详细配置了各个组件的细节。下面是他的一些参数：

```
<table class="table table-bordered table-striped table-condensed">
  <tr>
    <th width="30%">参数</th><th>简介</th>
  </tr>
  <tr>
    <td>SPARK_MASTER_IP</td><td>指定master进程所在的机器的ip地址</td>
  </tr>
  <tr>
    <td>SPARK_MASTER_PORT</td><td>指定master监听的端口号（默认是7077）</td>
  </tr>
  <tr>
    <td>SPARK_MASTER_WEBUIPORT</td><td>指定master web ui的端口号（默认是8080）</td>
  </tr>
  <tr>
    <td>SPARK_LOCAL_DIRS</td><td>指spark的工作目录，包括了shuffle map输出文件，以及久化到磁盘的RDD等</td>
  </tr>
  <tr>
    <td>SPARK_WORKER_PORT</td><td>worker节点的端口号，默认是随机的</td>
  </tr>
  <tr>
    <td>SPARK_WORKER_CORES</td><td>worker节点上，允许spark作业使用的最大cpu数量默认是机器上所有的cpu core</td>
  </tr>
  <tr>
    <td>SPARK_WORKER_MEMORY</td><td>worker节点上，允许spark作业使用的最大内存量格式为1000m, 2g等，默认最小是1g内存</td>
  </tr>
  <tr>
```

SPARK_WORKER_INSTANCES	当前机器上的worker进程数量，默认是1，可以置成多个，但是这时一定要设置SPARK_WORKER_CORES，限制每个worker的cpu数量
SPARK_WORKER_DIR	spark作业的工作目录，包括了作业的日志等，默认是spark_home/work
SPARK_DAEMON_MEMORY	分配给master和worker进程自己本身的内存，默认是1g
SPARK_PUBLIC_DNS	master和worker的公共dns域名，默认是空

其中一些参数跟我们单独启动master和worker进程的命令行参数后面添加参数是一样的，如：`sbin/start-master.sh --port 7078(或--memory 500m)`，类似这种方式，可以指定一样的配置属性。我们可以在spark-env.sh中就去配置好这些参数，但是有时候需要根据情况临时需改配置，我们就可以使用种在启动Master/Worker脚本时，添加参数的方式来进行配置。通常还是在脚本spark-env.sh中配置，脚本命令行参数通常用于临时的情况。

也就是说，有些master和worker的配置，可以在spark-env.sh中部署时即配置，但是也可以在start-slave.sh脚本启动进程时命令行参数设置`但是命令行参数的优先级比较高，会覆盖掉spark-env.sh中的配置`。

参数	简介
SPARK_MASTER_OPTS	设置master的额外参数，使用"-Dx=y"设置各个参数

比如说`export SPARK\_MASTER\_OPTS="-Dspark.deploy.defaultCores=1"`

参数(x)	默认值(y)	简介
spark.deploy.defaultCores	无限大	每个spark作业最多在standalone集群中使用多少个cpu core，默认是无限大，有多少用多少
spark.deploy.retainedApplications	200	在spark web ui上最多显示多少个application的信息
spark.deploy.retainedDrivers	200	在spark web ui上最多显示多少个driver的信息

spark.deploy.spreadOut	true	资源调度策略，spreadOut会尽量将application的executor进程分布在更多worker上，适合基于hdfs文件计算的情况，提升数据本地化概率；非spreadOut会尽量将executor分配到一个worker上，适合计算密集型的作业
spark.deploy.timeout	60	单位秒，一个worker多少时间没有响应之，master认为worker挂掉了

参数	简介
SPARK_WORKER_OPTS	worker的额外参数，使用"-Dx=y"设置各个参数

SPARK\_WORKER\_OPTS的一些参数配置 (x, y) 如下:

参数(x)	默认值(y)	简介
spark.worker.cleanup.enabled	false	是否启动自动清理worker工作目录，默认是false
spark.worker.cleanup.interval	1800	单位秒，自动清理的时间间隔默认是30分钟
spark.worker.cleanup.appDataTtl	7 * 24 * 3600	默认将一个spark业的文件在worker工作目录保留多少时间，默认是7天

参数	简介
SPARK_DAEMON_JAVA_OPTS	设置master和worker自己的jvm参数，使用"-Dx y"设置各个参数

这里一些内存相关的参数大家看过就明白了，为什么之前用分布式的集群，每个worker节点才1个g内存，根本是没有办法使用standalone模式和yarn模式运行作业的，仅仅是启动进程耗费的内存就已经严重了，其中一些内存分配不合理就会导致启动的时候资源分配失败。如果还要在Yarn模式上运行话，Hadoop集群进程还会占用一部分内存。所以说Spark使用是非常消耗资源的。有时候资源也会为性能的一个瓶颈。