

vincent_hv

Talk is cheap, show the code!

博客园 闪存 首页 新随笔 联系 管理 订阅 XML

随笔- 86 文章- 0 评论- 3

【译】Spark官方文档——Spark Configuration (Spark配置)

注重版权，尊重他人劳动

转帖注明原文地址：<http://www.cnblogs.com/vincent-hv/p/3316502.html>

Spark主要提供三种位置配置系统：

- 环境变量：用来启动Spark workers，可以设置在你的驱动程序或者conf/spark-env.sh 脚本中；
- java系统性能：可以控制内部的配置参数，两种设置方法：
 - 编程的方式（ 程序中在创建SparkContext之前，使用System.setProperty ("xx", "xxx") 语句设置相应系统属性值 ）；
 - 在conf/spark-env.sh中配置环境变量SPARK_JAVA_OPTS。
- 日志配置：通过log4j.properties实习

一、环境变量

spark安装目录下的conf/spark-env.sh脚本决定了如何初始化worker nodes的JVM，甚至决定了你在本地如何运行spark-shell。在Git库中这个脚本默认是不存在的，但是你可以自己创建它并通过复制con/spark-env.sh.template中的内容来配置，最后要确保你创建的文件可执行。

在spark-env.sh中你至少有两个变量要设置：

- SCALA_HOME，指向你的scala安装路径；或者是SCALA_LIBRARY_PATH指向scala library JARS所在的目录（ 如果你是通过DEB或者RPM安装的scala，他们是没有SCALA_HOME的，但是他们的libraries是分离的，默认在/usr/share/java中查找scala-library.jar ）
- MESOS_NATIVE_LIBRARY，如果你要在Mesos上运行集群的话

另外，还有其他四个变量来控制执行。应该将他们设置在启动驱动程序的环境中来取代设置在spark-env.sh，因为这样这些设置可以自动传递给workers。将他们设置在每个作业中而不是spark-env.sh中，这样确保了每个作业有他们自己的配置。

- SPARK_JAVA_OPTS，添加JVM选项。你可以通过-D来获取任何系统属性；
- SPARK_CLASS_PATH，添加元素到Spark的classpth中；
- SPARK_LIBARAT_OATH，添加本地libraries的查找目录；
- SPARK_MEM，设置每个节点所能使用的内存总量。他们应该和JVM's -Xmx选项的格式保持一致（ e.g.300m或1g ）。注意：这个选项将很快被弃用支持系统属性spark.executor.memory，所以我们推荐将它使用在新代码中。

注意：如果你将他们设置在spark-env.sh中，他们将覆盖用户程序设定的值，这是不可取的。如果你喜欢，你可以选择在spark-env.sh设置他们仅当用户程序没有做任何设置时，例如：

```
if [ -z "$SPARK_JAVA_OPTS" ]; then
SPARK_JAVA_OPTS="-verbose:gc"
fi
```

二、系统属性

通过设置系统属性来配置Spark，你必须通过以下两种方式中的任意一个来达到目的：

- 在JVM中通过-D标志（ 例如：java -Dspark.cores.max=5 MyProgram ）
- 在你的程序中创建SparkContext之前调用System.setProperty，如下：

```
System.setProperty("spark.cores.max", "5")
val sc = new SparkContext(...)
```

更多可配置的控制内部设置的系统属性已经有了合理的默认属性值。然而，有五个属性通常是你想要去控制的：

属性名称	默认值	含义
------	-----	----

昵称：[vincent_hv](#)

园龄：10个月

粉丝：7

关注：1

+加关注

< 2013年10月 >						
日	一	二	三	四	五	六
29	30	<u>1</u>	<u>2</u>	3	4	5
6	7	<u>8</u>	9	10	11	12
13	14	15	16	17	18	19
20	<u>21</u>	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找看

谷歌搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)
[更多链接](#)

最新随笔

1. [linux解压zip乱码解决方案](#)
2. [全能系统监控工具dstat](#)
3. 【转】[linux sar命令详解](#)
4. 【原】[gnome3增加自定义程序快捷方式](#)
5. 【原】[Ubuntu13.04安装、卸载Gnom e3.8](#)
6. 【原】[安装、卸载、查看软件时常用的命令](#)
7. 【原】[中文Ubuntu主目录下的文档文件夹改回英文](#)
8. 【原】[Ubuntu ATI/Intel双显卡 驱动安装](#)
9. 【原】[Ubuntu 12.04 ATI显卡设置双屏显示](#)
10. 【转】[Hadoop vs Spark性能对比](#)

随笔分类

[Android\(8\)](#)
[Hadoop\(2\)](#)
[Java\(20\)](#)
[JVM\(3\)](#)
[Linux\(23\)](#)
[others\(1\)](#)
[Scala\(5\)](#)
[Spark\(20\)](#)
[数据结构与算法\(2\)](#)

除了这些，在某些情况下以下属性可能也是需要设置的：

属性名	默认值	含义
spark.mesos.coarse	false	如果设置为了"true", 将以 粗粒度共享模式 运行在Mesos集群上， 这时候Spark会在每台机器上面获得一个长期运行的Mesos任务，而不是对每个Spark任务都要产生一个Mesos任务。对于很多短查询，这个可能会有些许的延迟，但是会大大提高Spark工作时的资源利用率。
spark.default.parallelism	8	在用户没有指定时，用于分布式随机操作(groupByKey,reduceByKey等等)的默认的任务数。
spark.storage.memoryFraction	0.66	Spark用于缓存的内存大小所占用的Java堆的比率。这个不应该大于JVM中老年代所分配的内存大小，默认情况下老年代大小是堆大小的2/3，但是你可以通过配置你的老年代的大小，然后再去增加这个比率。
spark.ui.port	(random)	你的应用程序控制面板端口号，控制面板中可以显示每个RDD的内存使用情况。
spark.shuffle.compress	true	是否压缩映射输出文件，通常设置为true是个不错的选择。
spark.broadcast.compress	true	广播变量在发送之前是否先要被压缩，通常设置为true是个不错的选择。
spark.rdd.compress	false	是否要压缩序列化的RDD分区（比如，StorageLevel.MEMORY_ONLY_SER）。在消耗一点额外的CPU时间的代价下，可以极大的提高减少空间的使用。
spark.reducer.maxMbInFlight	48	同时获取每一个分解任务的时候，映射输出文件的最大的尺寸（以兆为单位）。由于对每个输出都需要我们去创建一个缓冲区去接受它，这个属性值代表了对每个分解任务所使用的内存的一个上限值，因此除非你机器内存很大，最好还是配置一下这个值。
spark.closure.serializer	spark.JavaSerializer	用于闭包的序列化类。通常Java是可以胜任的，除非在你的驱动程序中分布式函数(比如map函数)引用了大量的对象。
spark.kryo.serializer.buffer.mb	32	Kryo中运行的对象的最大尺寸（Kryo库需要创建一个不小于最大的单个序列化对象的缓存区）。如果在Kryo中出现"buffer limit exceeded"异常，你就需要去增加这个值了。注意，对每个worker而言，一个核心就会有一个缓冲。
spark.broadcast.factory	spark.broadcast.HttpBroadcastFactory	使用哪一个广播实现
spark.locality.wait	3000	在发布一个本地数据任务时候，放弃并发布到一个非本地数据的地方前，需要等待的时间。如果你的很多任务都是长时间运行的任务，并且看到了很多的脏数据的话，你就该增加这个值了。但是一般情况下缺省值就可以很好的工作了。
spark.worker.timeout	60	如果超过这个时间，独立部署master还没有收到worker的心跳回复，那么就认为这个worker已经丢失了。
spark.akkas.frameSize	10	在控制面板通信（序列化任务和任务结果）的时候消息尺寸的最大值，单位是MB。如果你需要给驱动器发回大尺寸的结果（比如使用在一个大的数据集上面使用collect()方法），那么你就该增加这个值了。
spark.akkas.threads	4	用于通信的actor线程数量。如果驱动器有很多CPU核心，那么在大集群上可以增大这个值。
spark.akkas.timeout	20	Spark节点之间通信的超时时间，以秒为单位
spark.driver.host	(local hostname)	驱动器监听主机名或者IP地址。
spark.driver.port	(random)	驱动器监听端口号
spark.cleaner.ttl	(disable)	Spark记忆任何元数据(stages生成，任务生成等等)的时间(秒)。周期性清除保证在这个时间之前的元数据会被遗忘。当长时间几小时，几天的运行Spark的时候设置这个是很有用的。注意：任何内存中的RDD只要过了这个时间就会被清除掉。
spark.streaming.blockInterval	200	从网络中批量接受对象时的持续时间。
spark.task.maxFailures	4	task失败重试次数

积分与排名

积分 - 5935
排名 - 17402

最新评论

1. Re:全能系统监控工具dstat

感觉好高级的样子，我也下载来玩完

--花瓣奶牛
2. Re:【原】Ubuntu13.04安装、卸载Gnome3.8

马上应该有13.10了。

--杨琼
3. Re:scala实现kmeans算法

在oschina上一位大牛给我的指点，原文贴上，供跟多的孩纸学习：oldpig 发表于 2013-09-03 10:45 1. Source.getLinesr 返回的Iterator已经够用了，不需要toArray 2. 随机初始化k个质心，可以考虑使用Array.fill 3. 如果你要测算法的计算时间，应将两条println语句放到startTime之前 4. 计算movement可以考虑使用...

--vincent_hv

阅读排行榜

1. Ubuntu 13.04 完全配置(3093)
2. Android控件TextView的实现原理分析(213)
3. 【转】JVM (Java虚拟机) 优化大全和案例实战(174)
4. 【转】Spark：一个高效的分布式计算系统(138)
5. 修改Ubuntu12.04 开机启动菜单，包括系统启动等待时间，系统启动顺序(132)

评论排行榜

1. 【原】Ubuntu13.04安装、卸载Gnome3.8(1)
2. scala实现kmeans算法(1)
3. 全能系统监控工具dstat(1)
4. 【转】linux sar命令详解(0)
5. 【原】gnome3增加自定义程序快捷方式(0)

推荐排行榜

1. 【转】Spark源码分析之-Storage模块(2)
2. 【转】弹性分布式数据集：一种基于内存的集群计算的容错性抽象方法(1)
3. 【转】Spark：一个高效的分布式计算系统(1)
4. linux解压zip乱码解决方案(1)
5. 全能系统监控工具dstat(1)

三、配置日志

Spark使用log4j来记录。你可以在conf目录中添加log4j.properties文件来配置。一种方法是复制本地已存在的log4j.properties.template

分类: [Spark](#)

绿色通道: [好文要顶](#) [关注我](#) [收藏该文](#) [与我联系](#)



[vincent_hv](#)

[关注 - 1](#)

[粉丝 - 7](#)

[+加关注](#)

0

0

(请您对文章做出评价)

« 上一篇: [【转】Spark 体系结构](#)

» 下一篇: [【转】Spark性能测试报告](#)

posted @ 2013-09-12 11:20 [vincent_hv](#) 阅读(122) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

[博客园首页](#) [博问](#) [新闻](#) [闪存](#) [程序员招聘](#) [知识库](#)

最新IT新闻:

- [Google绝密项目：海上数据中心？](#)
 - [WP8版《愤怒的小鸟》和《割绳子》系列全限免](#)
 - [10个疯狂安卓设备：烤箱、冰箱、一个按钮](#)
 - [三星智能手表退货率达30%](#)
 - [Java程序员应该知道的10个面向对象理论](#)
- » [更多新闻...](#)

最新知识库文章:

- [软件开发启示录——迟到的领悟](#)
 - [《黑客帝国》里的锡安是不是虚拟世界](#)
 - [深入理解Linux中内存管理](#)
 - [工程师文化引出的组织行为话题](#)
 - [如何用美剧真正提升你的英语水平](#)
- » [更多知识库文章...](#)

Copyright ©2013 [vincent_hv](#)