

张伟的专栏

目录视图 摘要视图 RSS 订阅

个人资料



javastart

访问: 121514次  
积分: 1890  
等级:   
排名: 第13271名  
  
原创: 26篇 转载: 263篇  
译文: 0篇 评论: 10条

文章搜索

文章分类

- java (28)
- javascript (1)
- jsp (2)
- jtee (1)
- mssql (1)
- tomcat (6)
- 数据库开发 (2)
- 架构 (16)
- 系统架构 (11)
- 大数据 (131)
- oracle (1)
- mysql (23)
- nosql (13)
- liunix (6)
- hadoop (13)
- spark (42)
- hbase (0)
- clouderamanger (1)
- sqoop (0)
- 算法 (10)
- kafka (6)
- scala (6)
- flume (1)
- hive (5)

文章存档

2016撸课第二期之你听课我买单，快来撸你想听的课！ 2016大数据应用调查 AngularJS知识库发布

Spark动态资源分配-Dynamic Resource Allocation

2016-02-16 12:23 138人阅读 评论(0) 收藏 举报

分类: spark (42)

目录(?) [+]

Spark动态资源分配-Dynamic Resource Allocation

Spark lxw1234@qq.com 2个月前 (12-31) 1429°C 2评论

关键字: spark、资源分配、dynamic resource allocation

Spark中，所谓资源单位一般指的是executors，和Yarn中的Containers一样，在Spark On Yarn模式下，通常使用–num-executors来指定Application使用的executors数量，而–executor-memory和–executor-cores分别用来指定每个executor所使用的内存和虚拟CPU核数。相信很多朋友至今在提交Spark应用程序时候都使用该方式来指定资源。

假设有这样的场景，如果使用Hive，多个用户同时使用hive-cli做数据开发和分析，只有当用户提交执行了Hive SQL时候，才会向YARN申请资源，执行任务，如果不提交执行，无非就是停留在Hive-cli命令行，也就是个JVM而已，并不会浪费YARN的资源。现在想用Spark-SQL代替Hive来做数据开发和分析，也是多用户同时使用，如果按照之前的方式，以yarn-client模式运行spark-sql命令行（<http://lxw1234.com/archives/2015/08/448.htm>），在启动时候指定–num-executors 10，那么每个用户启动时候都使用了10个YARN的资源（Container），这10个资源就会一直被占用着，只有当用户退出spark-sql命令行时才会释放。

spark-sql On Yarn，能不能像Hive一样，执行SQL的时候才去申请资源，不执行的时候就释放掉资源呢，其实从Spark1.2之后，对于On Yarn模式，已经支持动态资源分配（Dynamic Resource Allocation），这样，就可以根据Application的负载（Task情况），动态的增加和减少executors，这种策略非常适合在YARN上使用spark-sql做数据开发和分析，以及将spark-sql作为长服务来使用的场景。

本文以Spark1.5.0和hadoop-2.3.0-cdh5.0.0，介绍在spark-sql On Yarn模式下，如何使用动态资源分配策略。

YARN的配置

首先需要对YARN的NodeManager进行配置，使其支持Spark的Shuffle Service。

修改每台NodeManager上的yarn-site.xml:

```
##修改
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle,spark_shuffle</value>
</property>
##增加
<property>
<name>yarn.nodemanager.aux-services.spark_shuffle.class</name>
<value>org.apache.spark.network.yarn.YarnShuffleService</value>
```

2016年03月 (30)

2016年02月 (42)

2016年01月 (16)

2015年12月 (12)

2015年11月 (23)

展开

阅读排行

数据库–Cobar分布式数

(11255)

Windows下IntelliJ IDEA

(7769)

spark 案例集群测试整理

(5313)

spark1.3.0\_\_for\_hadoo

(5032)

Spark与Mysql(JdbcRDC

(4460)

构建高并发高可用的电

(4359)

Java环境配置

(3517)

Spark MLlib Deep Lear

(2941)

金蝶 eas webservice 接

(2920)

Spark History Server配

(2749)

评论排行

Windows下IntelliJ IDEA

(4)

[原]Spark MLlib Deep Le

(2)

构建高并发高可用的电

(1)

Spark MLlib Deep Lear

(1)

Spark RDD写入RMDB(

(1)

Spark MLlib Deep Lear

(1)

使用Spark MLlib来训练

(0)

Tomcat配置技巧Top10

(0)

不负众望 Java API文档

(0)

Java编程:配置tomcat5数

(0)

推荐文章

\*正方形数量问题终极解答

\*几种软负载均衡策略分析

\*浅谈Storm流式处理框架

\*阿里2016实习offer五面经验与总结

\*公司app升级UI总结

\*三十分钟理解博弈论“纳什均衡”-- Nash Equilibrium

最新评论

Spark MLlib Deep Learning Net  
bima2015: 博主您好, 您的代码结构显示不出来呢, 另外求博主完整源码

Spark RDD写入RMDB(Mysql)方  
小母虫: 您好 我们现在用的是sqlserver数据库 现在向数据库插入数据 insertintoJD...

Windows下IntelliJ IDEA中调试S  
flyingsky2008: @yy2314703:你知道怎么用idea远程提交了么?

Spark MLlib Deep Learning Net  
hxwd110: 您好 我想咨询下/\* \* sigmoid激活函数 \* X = 1/(1+exp(-P));...

构建高并发高可用的电商平台架  
guoxiaodiu\_: 很赞!

Windows下IntelliJ IDEA中调试S  
yy2314703: 代码一样, 集群没

</property>

<property>

<name>spark.shuffle.service.port</name>

<value>7337</value>

</property>

将\$SPARK\_HOME/lib/spark-1.5.0-yarn-shuffle.jar拷贝到每台NodeManager的

\$ {HADOOP\_HOME}/share/hadoop/yarn/lib/下。

重启所有NodeManager。

Spark的配置

配置\$SPARK\_HOME/conf/spark-defaults.conf, 增加以下参数:

1.

spark.shuffle.service.enabled true //启用External shuffle Service服务

2.

spark.shuffle.service.port 7337 //Shuffle Service服务端口, 必须和yarn-site中的一致

3.

spark.dynamicAllocation.enabled true //开启动态资源分配

4.

spark.dynamicAllocation.minExecutors 1 //每个Application最小分配的executor数

5.

spark.dynamicAllocation.maxExecutors 30 //每个Application最大并发分配的executor数

6.

spark.dynamicAllocation.schedulerBacklogTimeout 1s

7.

spark.dynamicAllocation.sustainedSchedulerBacklogTimeout 5s

动态资源分配策略:

开启动态分配策略后, application会在task因没有足够资源被挂起的时候去动态申请资源, 这种情况意味着该application现有的executor无法满足所有task并行运行。spark一轮一轮的申请资源, 当有task挂起或等待spark.dynamicAllocation.schedulerBacklogTimeout(默认1s)时间的时候, 会开始动态资源分配; 之后会每隔spark.dynamicAllocation.sustainedSchedulerBacklogTimeout(默认1s)时间申请一次, 直到申请到足够的资源。每次申请的资源量是指数增长的, 即1,2,4,8等。

之所以采用指数增长, 出于两方面考虑: 其一, 开始申请的少是考虑到可能application会马上得到满足; 其次要成倍增加, 是为了防止application需要很多资源, 而该方式可以在很少次数的申请之后得到满足。

资源回收策略

当application的executor空闲时间超过spark.dynamicAllocation.executorIdleTimeout (默认60s) 后, 就会被回收。

使用spark-sql On Yarn执行SQL, 动态分配资源

1.

./spark-sql --master yarn-client \

2.

--executor-memory 1G \

3.

-e "SELECT COUNT(1) FROM ut.t\_ut\_site\_log where pt >= '2015-12-09' and pt <= '2015-12-10'"

[liuxiaowen@dev bin]\$ ./spark-sql --master yarn-client \

> --executor-memory 1G \

> -e "SELECT COUNT(1) FROM ut.t\_ut\_site\_log where pt >= '2015-12-09' and pt <= '2015-12-10'"

SET spark.sql.autoBroadcastJoinThreshold=314572800

SET hive.support.sql11.reserved.keywords=false

SET spark.sql.hive.version=1.2.1

SET spark.sql.hive.version=1.2.1

[Stage 0:> (0 + 12) / 123]

该查询需要123个Task。

问题。第一个pi的例子我这一跑就提示集群没有资源 master设为local就没问...

[原]Spark MLlib Deep Learning suifenglight: 1065075152@qq.com

[原]Spark MLlib Deep Learning suifenglight: 大神，为什么有很多内容都是空白呢？可以发我QQ邮箱吗？麻烦了！！

Windows下IntelliJ IDEA中调试S guotong1988: 学习

Windows下IntelliJ IDEA中调试S cughuhao: 学习了，谢谢~

Executors (31)

Memory: 0.0 B Used (16.1 GB Total)  
Disk: 0.0 B Used

Executor ID	Address	RDD Blocks	Storage Memory	Disk Used	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time	Input	Shuffle Read	Shuffle Write	Logs	Thread Dump
1	slave015-2972	0	0.0 B / 530.3 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
10	slave015-5282	0	0.0 B / 530.3 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
11	slave015-1280	0	0.0 B / 530.3 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
12	slave015-4962	0	0.0 B / 530.3 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
13	slave005-43026	0	0.0 B / 530.3 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
14	slave005-37110	0	0.0 B / 530.3 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump

从AppMaster的WEB界面可以看到，总共有31个Executors，其中一个Driver，既有30个Executors并发执行，而30，正是在spark.dynamicAllocation.maxExecutors参数中配置的最大并发数。如果一个查询只有10个Task，那么只会向Yarn申请10个executors的资源。

需要注意：

如果使用

./spark-sql --master yarn-client --executor-memory 1G

进入spark-sql命令行，在命令行中执行任何SQL查询，都不会执行，原因是spark-sql在提交到Yarn时候，已经被当成一个Application，而这种，除了Driver，是不会被分配到任何executors资源的，所有，你提交的查询因为没有executor而不能被执行。

而这个问题，我使用Spark的ThriftServer（HiveServer2）得以解决。

使用Thrift JDBC方式执行SQL，动态分配资源

首选以yarn-client模式，启动Spark的ThriftServer服务，也就是HiveServer2。

配置ThriftServer监听的端口号和地址

```
1. vi $SPARK_HOME/conf/spark-env.sh
2. export HIVE_SERVER2_THRIFT_PORT=10000
3. export HIVE_SERVER2_THRIFT_BIND_HOST=0.0.0.0
```

以yarn-client模式启动ThriftServer

```
1. cd $SPARK_HOME/sbin/
2. ./start-thriftserver.sh \
3. --master yarn-client \
4. --conf spark.driver.memory=3G \
5. --conf spark.shuffle.service.enabled=true \
6. --conf spark.dynamicAllocation.enabled=true \
7. --conf spark.dynamicAllocation.minExecutors=1 \
8. --conf spark.dynamicAllocation.maxExecutors=30 \
9. --conf spark.dynamicAllocation.sustainedSchedulerBacklogTimeout=5s
```

启动后，ThriftServer会在Yarn上作为一个长服务来运行：

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress
application_1451452849859_0048	liuxiaowen	org.apache.spark.sql.hive.thriftserver.HiveThriftServer2	SPARK	root.default	Wed, 30 Dec 2015 08:21:14 GMT	N/A	RUNNING	UNDEFINED	

使用beeline通过JDBC连接spark-sql

```
1. cd $SPARK_HOME/bin
2. ./beeline -u jdbc:hive2://localhost:10000 -n lxw1234
```

```
[liuxiaowen@dev bin]$ ./beeline -u jdbc:hive2://localhost:10000 -n lxw1234
Connecting to jdbc:hive2://localhost:10000
Connected to: Spark SQL (version 1.5.0)
Driver: Spark Project Core (version 1.5.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 1.5.0 by Apache Hive
0: jdbc:hive2://localhost:10000> use liuxiaowen;
+-----+
| result |                               lxw的大数据田地 -- lxw1234.com
+-----+
+-----+
No rows selected (0.151 seconds)
0: jdbc:hive2://localhost:10000> |
```

执行查询：

select count(1) from ut.t\_ut\_site\_log where pt = '2015-12-10';

该任务有64个Task:

```
os.version=2.6.32-431.el6.x86_64
derby.system.home=null
Database Class Loader started - derby.database.classpath=''
SET spark.sql.hive.version=1.2.1
OK
[Stage 0:> (0 + 20) / 64]
```

而监控页面上的并发数仍然是30:

Executors (31)

Memory: 0.0 B Used (14.3 GB Total)  
Disk: 0.0 B Used

lxw的大数据田地 -- lxw1234.com

Executor ID	Address	RDD Blocks	Storage Memory	Disk Used	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time	Input	Shuffle Read	Shuffle Write	Logs	Thread Dump
1	slave001	43913	0	0.0 B / 441.9 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr Thread Dump
10	slave003	37828	0	0.0 B / 441.9 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr Thread Dump
11	slave014	41573	0	0.0 B / 441.9 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr Thread Dump
12	slave006	33894	0	0.0 B / 441.9 MB	0.0 B	1	0	0	1	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr Thread Dump

执行完后，executors数只剩下1个，应该是缓存数据，其余的全部被回收：

Executors (2)

Memory: 0.0 B Used (1824.3 MB Total)

Disk: 0.0 B Used

lxw的大数据田地 -- lxw1234.com

Executor ID	Address	RDD Blocks	Storage Memory	Disk Used	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time	Input	Shuffle Read	Shuffle Write	Logs	Thread Dump
21	slave012	44125	0	0.0 B / 441.9 MB	0.0 B	0	3	3	30.0 s	256.0 MB	3.1 KB	104.0 B	stdout stderr	Thread Dump
driver	172.16.212.17:37559	0	0.0 B / 1382.4 MB	0.0 B	0	0	0	0	0 ms	0.0 B	0.0 B	0.0 B		Thread Dump

这样，多个用户可以通过beeline，JDBC连接到Thrift Server，执行SQL查询，而资源也是动态分配的。

需要注意的是，在启动ThriftServer时候指定的spark.dynamicAllocation.maxExecutors=30，是整个ThriftServer同时并发的最大资源数，如果多个用户同时连接，则会被多个用户共享竞争，总共30个。

这样，也算是解决了多用户同时使用spark-sql，并且动态分配资源的需求了。

Spark动态资源分配官方文档：<http://spark.apache.org/docs/1.5.0/job-scheduling.html#dynamic-resource-allocation>

您可以关注 [lxw的大数据田地](#)，或者 [加入邮件列表](#)，随时接收博客更新的通知邮件。

转载请注明：[lxw的大数据田地](#) » [Spark动态资源分配-Dynamic Resource Allocation](#)

顶

0

踩

0

上一篇

Spark UI (基于Yarn) 分析与定制

下一篇

【论文笔记】SparkNET: 用Spark训练深度神经网络

我的同类文章

spark (42)

• 使用Spark MLib来训练并服...

2016-03-31

阅读 28

• spark 开发环境读取mysql

2016-03-31

阅读 10

• Apache Spark Jobs 性能调...

2016-03-28

阅读 15

• 揭开Spark Streaming神秘...

2016-03-27

阅读 12

• 深入分析Parquet列式存储格式

2016-03-23

阅读 7

• Apache Spark as a Service

2016-03-31

阅读 3

• 在spark中操作mysql数据 ---...

2016-03-31

阅读 8

• 操作技巧：将 Spark 中的文...

2016-03-28

阅读 13

• Supported syntax of Spark ...

2016-03-24

阅读 12

• 揭开Spark Streaming神秘...

2016-03-23

阅读 22

更多文章

猜你在找

- Spark 1. x大数据平台
- 【大数据技术公开课】YARN & Docker在Hulu的实战
- Spark零基础入门（6）：Scala 模式匹配
- Spark零基础入门（3）：Scala函数
- Spark零基础入门（2）：解析Scala集合操作

英语雅思培训

呼叫中心系统

海报设计

日本纳豆激酶

互动投影

螨虫吸尘器

海报设计

查看评论

暂无评论

您还没有登录,请[登录](#)或[注册](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap