

<		201
日	一	二
27	28	29
3	4	5
10	11	12
17	18	19
24	25	26
1	2	3

导航
博客园
首页
新随笔
联系
订阅 XML
管理

统计
随笔 - 87
文章 - 870
评论 - 7
引用 - 0

公告
昵称：113013624
园龄：2年4个月
粉丝：28
关注：55
+加关注

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

我的标签
反射(6)
java(4)
内部类(3)
支持向量机(3)
序列化(2)
java json(2)
Git(2)
hadoop(2)
hbase(2)
Kubernetes(2)
更多

随笔分类
ElasticSearch
Hadoop(41)
Hbase(14)
Lucene(29)
文本资料(6)

随笔档案
2016年3月 (1)
2016年2月 (11)
2016年1月 (9)
2015年12月 (30)
2015年11月 (35)
2015年10月 (1)

文章分类
Angular2(7)
BI&ETL(9)
Docker(13)
ElasticSearch
GIT(6)
Hadoop(61)
Hadoop 异常(14)
Hbase(14)
Hive(48)
Java NIO(20)
Java Web(18)
Java 并发编程(5)
Java 范型(8)
Java 基础(65)

Hive over HBase和Hive over HDFS性能比较分析

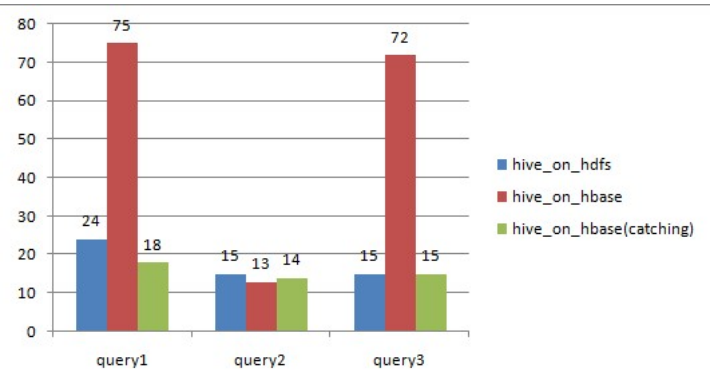
环境配置：

hadoop-2.0.0-cdh4.3.0 (4 nodes, 24G mem/node)
hbase-0.94.6-cdh4.3.0 (4 nodes,maxHeapMB=9973/node)
hive-0.10.0-cdh4.3.0

一、查询性能比较：

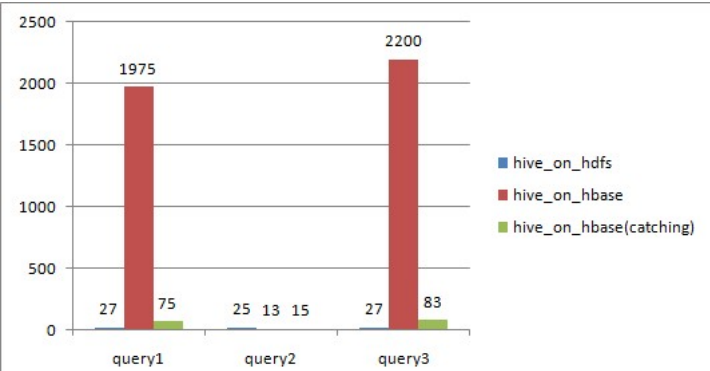
query1:
select count(1) from on_hdfs;
select count(1) from on_hbase;
query2(根据key过滤)
select * from on_hdfs
where key = '13400000064_1388056783_460095106148962';
select * from on_hbase
where key = '13400000064_1388056783_460095106148962';
query3(根据value过滤)
select * from on_hdfs where value = 'XXX';
select * from on_hbase where value = 'XXX';

on_hdfs (20万记录，150M，TextFile on HDFS)
on_hbase(20万记录，160M，HFile on HDFS)



Hive over HBase

on_hdfs (2500万记录，2.7G，TextFile on HDFS)
on_hbase(2500万记录，3G，HFile on HDFS)



Hive over HBase

从上图可以看出，
对于全表扫描，hive_on_hbase查询时候如果不设置catching，性能远远不及hive_on_hdfs；

根据rowkey过滤，hive_on_hbase性能上略好于hive_on_hdfs，特别是数据量大的时候；
设置了caching之后，尽管比不设caching好很多，但还是略逊于hive_on_hdfs；

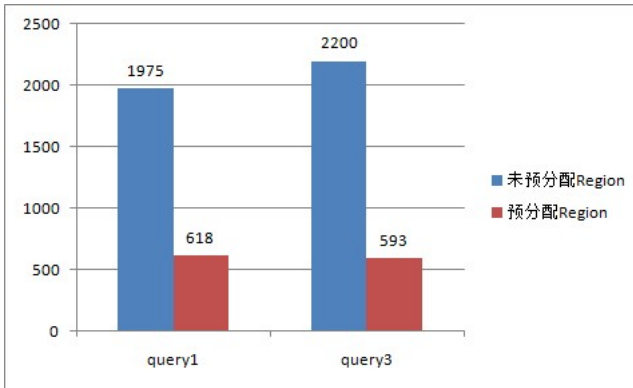
二、Hive over HBase原理

Hive与HBase利用两者本身对外的API来实现整合，主要是靠HBaseStorageHandler进行通信，利用 HBaseStorageHandler，Hive可以获取到Hive表对应的HBase表名，列簇以及列，InputFormat和 OutputFormat类，创建和删除HBase表等。
Hive访问HBase中表数据，实质上是通过MapReduce读取HBase表数据，其实现是在MR中，使用HiveHBaseTableInputFormat完成对HBase表的切分，获取RecordReader对象来读取数据。
对HBase表的切分原则是一个Region切分成一个Split，即表中有多少个Regions，MR中就有多个Map；
读取HBase表数据都是通过构建Scanner，对表进行全表扫描，如果有过滤条件，则转化为Filter。当过滤条件为rowkey时，则转化为对rowkey的过滤；
Scanner通过RPC调用RegionServer的next()来获取数据；

三、性能瓶颈分析

1. Map Task

Hive读取HBase表，通过MR，最终使用HiveHBaseTableInputFormat来读取数据，在getSplit()方法中对 HBase表进行切分，切分原则是根据该表对应的HRegion，将每一个Region作为一个InputSplit，即，该表有多少个Region，就有多少个Map Task；
每个Region的大小由参数hbase.hregion.max.filesize控制，默认10G，这样会使得每个map task处理的数据文件太大，map task性能自然很差；
为HBase表预分配Region，使得每个Region的大小在合理的范围；
下图是给该表预分配了15个Region，并且控制key均匀分布在每个Region上之后，查询的耗时对比，其本质上是Map数增加。



Hive over HBase

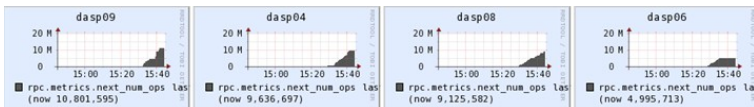
2. Scan RPC 调用：

- 在Scan中的每一次next()方法都会为每一行数据生成一个单独的RPC请求， query1和query3中，全表有2500万行记录，因此要2500万次RPC请求；
- 扫描器缓存（Scanner Caching）：HBase为扫描器提供了缓存的功能，可以通过参数hbase.client.scanner.caching来设置；默认是1；缓存的原理是通过设置一个缓存的行数，当客户端通过RPC请求RegionServer获取数据时，RegionServer先将数据缓存到内存，当缓存的数据行数达到参数设置的数目时，再一起返回给客户端。这样，通过设置扫描器缓存，就可以大幅度减少客户端RPC调用RegionServer的次数；但并不 是缓存设置的越大越好，如果设置的太大，每一次RPC调用将会占用更长的时间，因为要获取更多的数据并传输到客户端，如果返回给客户端的数据超出了其堆的 大小，程序就会终止并跑出OOM异常；

所以，需要为少量的RPC请求次数和客户端以及服务端的内存消耗找到平衡点。

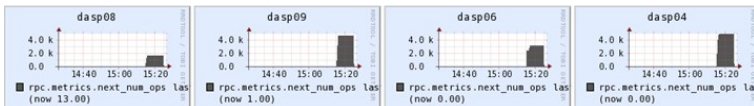
rpc.metrics.next_num_ops

未设置caching,每个RegionServer上通过next()方法调用RPC的次数峰值达到1000万：



Hive over HBase

设置了caching=2000，每个RegionServer上通过next()方法调用RPC的次数峰值只有4000：



Hive over HBase

Java 集合类框架(14)
Java 进阶(173)
Java 枚举(4)
Java 面向对象(25)
Java 内存分析(11)
Java 容器(19)
Java 尚学堂(79)
Java 数据结构和算法
Java 数据库操作(17)
Java 网络编程(11)
Java 文件操作(31)
Java 线程(26)
Java 异常(11)
Java 张孝祥系列(13)
Java 正则表达式(9)
Jquery EasyUI(2)
JUnit(5)
Linux 学习(45)
Lucene
MATLAB(6)
maven(21)
MongoDB(4)
Mysql(73)
NoSql(1)
Nutch(3)
python(4)
Solr
Spark(44)
Storm(5)
XML(1)
Zookeeper(18)
分词工具(3)
深入理解Java虚拟机
数据库(1)
数据挖掘(6)
文本(11)
新浪微博接口(3)
字符串相似度 编辑距

最新评论

- Re:hadoop入门-去看看赵强老师的公有一节专门讲Hadoo完之后就on知道怎么学377317956，里面详
- Re:hadoop入门-关于Mapreduce的讲习视频，讲的很详细
- Re:Java的基本数据类型是32位、表示的整数；占3字节吗?long数据类型是64位补码表示的整数；占一般都是八个字节吧。
- Re:Java的基本数据类型您好，请问如何检测类型，并将其输出
- Re:[一起学习Hive]感谢分享，正好用得

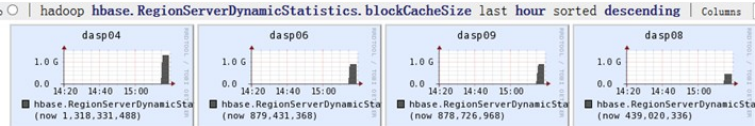
阅读排行榜

- 浅谈hbase表中数(份) (4334)
- Hadoop学习笔记常见算法(1144)
- Lucene4.7 分页(!)
- Lucene add、up更新与search查询(转)
- 为什么Hbase能实

推荐排行榜

- hadoop MapRed (task)、管理角色

设置了caching之后，几个RegionServer上的内存消耗明显增加：



Hive over HBase

- 扫描器批量 (Scanner Batch)：缓存是面向行一级的操作，而批量则是面向列一级的操作。批量可以控制每一次next()操作要取回多少列。比如，在扫描器中设置setBatch(5),则一次next()返回的Result实例会包括5列。
- RPC请求次数的计算公式如下：
RPC请求次数 = (表行数 * 每行的列数) / Min(每行的列数, 批量大小) / 扫描器缓存

因此，在使用Hive over HBase，对HBase中的表做统计分析时候，需要特别注意以下几个方面：

- 对HBase表进行预分配Region，根据表的数据量估算出一个合理的Region数；
- rowkey设计上需要注意，尽量使rowkey均匀分布在预分配的N个Region上；
- 通过set hbase.client.scanner.caching设置合理的扫描器缓存；
- 关闭mapreduce的推测执行：

```
set mapred.map.tasks.speculative.execution = false;
set mapred.reduce.tasks.speculative.execution = false;
```

转载请注明：lxw的大数据田地 » Hive over HBase和Hive over HDFS性能比较分析

分类: Hbase,Hive

好文要顶

关注我

收藏该文

1130136248
关注 - 55
粉丝 - 28

+加关注

0

0

« 上一篇：hadoop 2.2.0 关于map和reduce的个数的设置

posted on 2016-05-22 20:51 1130136248 阅读(118) 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](#)。

- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】腾讯云域名+云解析 限时折扣抓紧抢购
- 【推荐】可嵌入您系统的“在线Excel”！SpreadJS 纯前端表格控件
- 【推荐】阿里云“全民云计算”优惠升级



最新IT新闻:

- 微软变形传感器专利曝光，或用于HoloLens及穿戴设备
 - Lily无人机东山再起，但混然于众矣
 - 本土创业“出海”难？这家公司拿下了北美70%的市场份额
 - 《愤怒的小鸟》开发商Rovio将宣布上市计划：估值可达20亿美元
 - 《王者荣耀》上线“隐身”功能：再不怕游戏中遇到老板
- » 更多新闻...

开发用极光 省心功能强

推送IM短信统计分享

最新知识库文章:

- 做到这一点，你也可以成为优秀的程序员
 - 写给立志做码农的大学生
 - 架构腐化之谜
 - 学会思考，而不只是编程
 - 编写Shell脚本的最佳实践
- » 更多知识库文章...