



原创

# Parquet性能测试调优及其优化建议



Rawirm

2018-02-17 22:01:46 10821人阅读 2人评论

## Parquet性能测试调优及其优化建议

### 一、我们为什么选择parquet

#### 1、选择parquet的外部因素

- (1) 我们已经在使用spark集群，spark原本就支持parquet，并推荐其存储格式（默认存储为parquet）；
- (2) hive支持parquet格式存储，使用HiveSql查询也是完全兼容的。



#### 2、选择parquet的本身原因

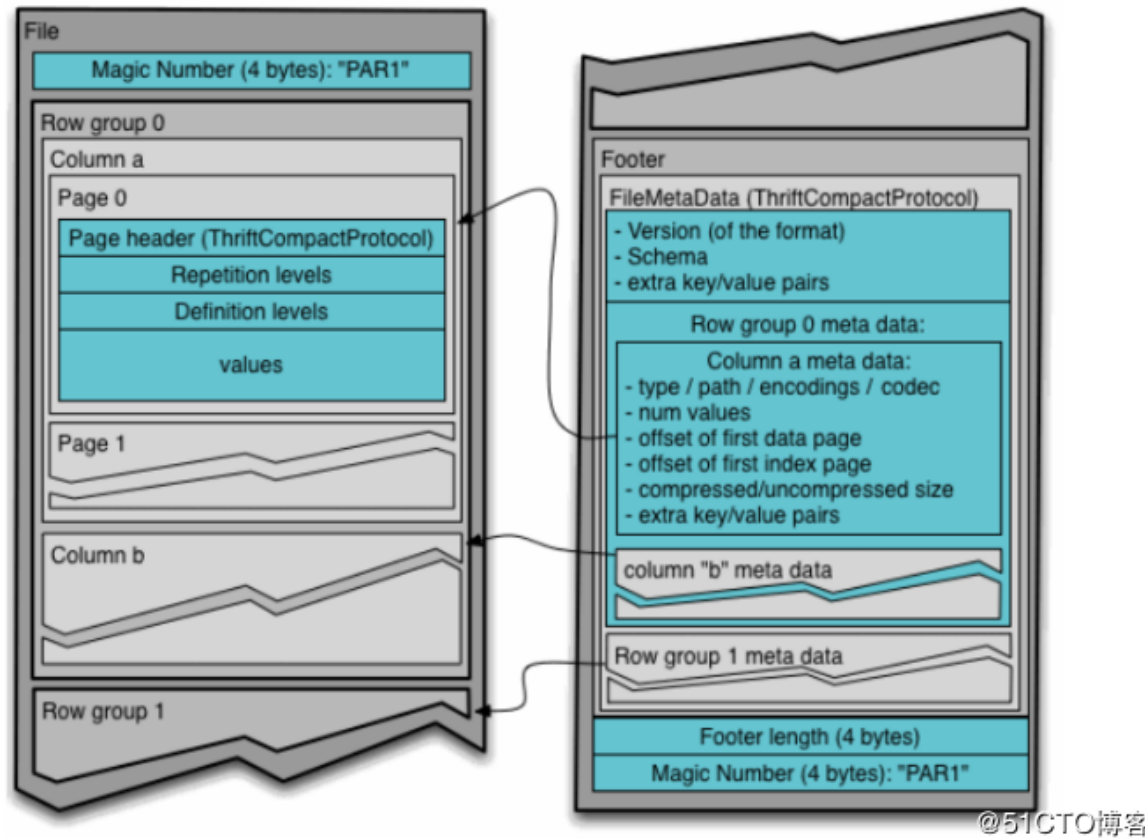
(1) parquet由于每一列的成员都是同构的，可以针对不同的数据类型使用更高效的数据压缩算法，进一步减小I/O。CSV格式一般不进行压缩，通过parquet存储数据有效的节约了空间，不考虑备份情况下，压缩比将近27倍（parquet有四种压缩方式lzo、gzip、snappy、uncompressed，其中默认gzip的压缩方式，其压缩率最高，压缩解压的速率最快）；

(2) 查询的时候不需要扫描全部的数据，而只需要读取每次查询涉及的列，这样可以将I/O消耗降低N倍，另外可以保存每一列的统计信息(min、max、sum等)；

(3) 分区过滤与列修剪中，parquet结合spark可以实现分区过滤（spark sql, rdd的filter和where关键

[在线客服](#)

## 2. parquet的列式存储格式的解析（仅了解）



Parquet文件在磁盘上的分布情况如上图，所有的数据被水平切分成Row group，一个Row group包含这个Row group对应的区间内的所有列的column chunk。一个column chunk负责存储某一列的数据，这些数据是这一列的Repetition level，Definition level和Values。一个column chunk是由Page组成的，Page是压缩和编码的单元，对数据模型来说是透明的。一个Parquet文件最后是Footer，存储了文件的元数据信息和统计信息。Row group是数据读写时候的缓存单元，所以推荐设置较大的Row group从而带来较大的并行度，当然也需要较大的内存空间作为代价。一般情况下推荐配置一个Row group大小1G，一个HDFS块大小1G，一个HDFS文件只含有一个块。

## 3. Parquet性能测试



②测试目的：验证spark在读取普通文件和parquet文件性能时，在读取相同的列的速率上面，比普通的文件效率更高，随着列的增加读取的效率会降低。

### ③测试原理：

由于以下特性，使得列式存储对于一些运算速率相对行式存储运行速率更快：

(1) 由于每一列的成员都是同构的，可以针对不同的数据类型使用更高效的数据压缩算法，进一步减小I/O。

(2) 由于每一列的成員的同构性，可以使用更加适合CPU pipeline的编码方式，减小CPU的缓存失效。

### ④测试步骤

(1) 使用C\_PORT表建立hive表，同样建立一个C\_PORT\_PARQUET，使用stored as parquet将表存储为parquet格式；

(2) 编写spark读取语句，对列数进行查询读取操作；

(3) 增加读取列数，在机器上spark提交任务运行记录运行时间；

(4) 对比运行时间，得出最终结论。

### ⑤测试结果

约27005w数据      普通hive表      request表      测试结果：

查询列数	普通hive表耗时	Parquet表耗时
1列	2分53秒	2分42秒
5列	3分53秒	1分27秒
20列	5分58秒	3分56秒
35列	9分16秒	9分36秒
50列	13分19秒	8分11秒



### ⑥总结结论

通过以上五组数据列的读取得知，随着列数的增加，读取的时间增加，相对于parquet和普通hive的读取速率相近，由此在列数较多时，读取非全部列数据，建议使用parquet存储可以增加读取效率。

在线  
客服



(2) 测试parquet列式存储在对多列数据进行列式计算的效率

①测试环境：58.56机器、spark1.6、sts、hive等

②测试目的：验证spark在读取普通文件和parquet文件性能时，针对某些列式运算列式存储的性能更佳，即读取计算速率更快。

③测试原理：

由于以下特性，使得列式存储对于一些运算速率相对行式存储运行速率更快：

(1) 查询的时候不需要扫描全部的数据，而只需要读取每次查询涉及的列，这样可以将I/O消耗降低N倍，另外可以保存每一列的统计信息(min、max、sum等)，实现部分的谓词下推。

(2) 由于每一列的成员都是同构的，可以针对不同的数据类型使用更高效的数据压缩算法，进一步减小I/O。

(3) 由于每一列的成员的成员的同构性，可以使用更加适合CPU pipeline的编码方式，减小CPU的缓存失效。

④测试步骤

(1) 使用C\_PORT表建立hive表，同样建立一个C\_PORT\_PARQUET，使用stored as parquet将表存储为parquet格式；

(2) 编写spark读取语句，包含列式计算的sum，avg以及max，min语句；

(3) 在机器上spark提交任务运行记录运行时间；

(4) 对比运行时间，得出最终结论。

⑤测试结果

第一组：



在线  
客服



测试结果:

时间	普通hive表	Parquet表
耗时	2分14秒	1分37秒
耗时	2分24秒	1分08秒
耗时	2分27秒	1分36秒
平均耗时	2分33秒	1分27秒

第二组:

约27005w数据      普通hive表      request表      (按照每天小时分组, 2个求和, 3个求平均运算, 2个求最大值, 2个求最小值)

测试结果:

时间	普通hive表	Parquet表
耗时	2分22秒	1分38秒
耗时	2分58秒	1分51秒
耗时	2分31秒	1分38秒
平均耗时	2分37秒	1分42秒



在线  
客服

第三组:

约27005w数据      普通hive表      request表      (按照每天小时分组, 4个求和, 4个求平均运算, 4个求最大值, 4个求最小值)



时间	普通hive表	Parquet表
耗时	3分03秒	1分58秒
耗时	2分45秒	2分03秒
耗时	2分48秒	2分06秒
平均耗时	2分52秒	2分02秒

## ⑥总结结论

通过三组数值的比对计算，列式存储格式parquet针对列式计算效率比普通的行式存储有明显的优势，运算的效率提升在30%-40%左右，效率更高，执行效率更快。

### 1. 测试普通文件和parquet文件的压缩效率对比

①测试环境：58.56机器、spark1.6、sts、hive等

②测试目的：验证测试普通文件和parquet文件的压缩效率对比，在压缩存储相同数据时，存储为parquet文件压缩效率更高，占用的空间更小。

③测试原理：

(1) 由于每一列的成员都是同构的，可以针对不同的数据类型使用更高效的数据压缩算法，进一步减小I/O。

(2) 由于每一列的成员的同构性，可以使用更加适合CPU pipeline的编码方式，减小CPU的缓存失效。

④测试步骤

(1) 同样的SparkSql运行，存储方式不同。生成相同数据量的parquet文件和普通文件存储；

(2) 分别查看生成的Parquet文件和普通文件的大小，对比结果。

⑤测试结果

结果如下图：



在线  
客服



```
[us1_odso@et15856 ~]$ cd zhanghaiyong/
[us1_odso@et15856 zhanghaiyong]$ hadoop fs -ls zhang/
Found 9 items
drwxr-xr-x - us1_odso us1_odso 0 2017-08-18 17:02 zhang/20170713
drwxr-xr-x - us1_odso us1_odso 0 2017-08-23 16:07 zhang/20170815
drwxr-xr-x - us1_odso us1_odso 0 2017-08-23 12:38 zhang/topo_add_stb
drwxr-xr-x - us1_odso us1_odso 0 2017-08-23 12:39 zhang/topo_city_bars_olt
drwxr-xr-x - us1_odso us1_odso 0 2017-08-20 10:31 zhang/zyone
drwxr-xr-x - us1_odso us1_odso 0 2017-09-01 11:19 zhang/zyproinfoBefore 存储为普通文件
drwxr-xr-x - us1_odso us1_odso 0 2017-09-01 10:30 zhang/zyproinfoNow
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:34 zhang/zyproinfoTwo
drwxr-xr-x - us1_odso us1_odso 0 2017-09-01 17:47 zhang/zyproinfoTest
[us1_odso@et15856 zhanghaiyong]$ hadoop fs -ls zhang/zyproinfoTwo
Found 15 items
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:14 zhang/zyproinfoTwo/zy_11_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:15 zhang/zyproinfoTwo/zy_12_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:16 zhang/zyproinfoTwo/zy_13_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:16 zhang/zyproinfoTwo/zy_14_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:19 zhang/zyproinfoTwo/zy_15_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:16 zhang/zyproinfoTwo/zy_15_1_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:21 zhang/zyproinfoTwo/zy_16_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:23 zhang/zyproinfoTwo/zy_17_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:27 zhang/zyproinfoTwo/zy_18_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:30 zhang/zyproinfoTwo/zy_19_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:30 zhang/zyproinfoTwo/zy_20_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:33 zhang/zyproinfoTwo/zy_21_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:34 zhang/zyproinfoTwo/zy_22_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:34 zhang/zyproinfoTwo/zy_23_01.parquet
drwxr-xr-x - us1_odso us1_odso 0 2017-08-31 15:37 zhang/zyproinfoTwo/zy_project_infoa.parquet 存储为parquet文件
[us1_odso@et15856 zhanghaiyong]$ hadoop fs -du -h -s zhang/zyproinfoTwo/zy_project_infoa.parquet
1.2 G 3.6 G zhang/zyproinfoTwo/zy_project_infoa.parquet
[us1_odso@et15856 zhanghaiyong]$ hadoop fs -du -h -s zhang/zyproinfoBefore
4.2 G 12.6 G zhang/zyproinfoBefore
[us1_odso@et15856 zhanghaiyong]$
```

@51CTO博客

经过最终执行结果，存储为普通文件的总大小为12.6G,存储为parquet文件的大小为3.6G，存储所占空间减少了近70%，因此存储为parquet文件占用的空间更小。

#### 四、Parquet在实际项目中的应用建议

- (1) 当读取的列数并非全部列数，建议使用parquet格式存储（建表时使用stored by parquet）；
- (2) 在进行列式计算或者向量计算时，建议也使用parquet格式存储，可以提高运算效率；
- (3) 如果有文件需要备份存储，可以使用parquet文件进行压缩，可以有效的节约空间，提高压缩效率和速率。



在线  
客服



0

收藏
 分享

上一篇：不使用Sqoop流程，利用Cac...



Rawirm
 11篇文章，9W+人气，0粉丝



提问和评论都可以，用心的回复会被更多人看到和认可



Ctrl+Enter 发布
 取消
 发布

2条评论

按时间正序 | 按时间倒序



sudao3
 1楼 2018-02-28 15:12:41

1

压缩率的对比，不是应该是普通文本文件和parquet文件都使用相同的压缩格式（如Gzip）压缩，然后对比压缩率吗，普通文本压缩一下也会变小的呀？？？和原始对比并没有体现parquet格式文件本身的压缩率的高啊？？

作者
 Rawirm:@sudao3





## 推荐专栏

更多



## 带你玩转高可用

前百度高级工程师的架构高可用实战

共15章 | 曹林华

¥ 51.00

495人订阅

订 阅

## 猜你喜欢

[不使用Sqoop流程，利用CacheManager直接完成Spark...](#)[磁盘性能指标--IOPS、吞吐量及测试](#)[Jmeter压力测试（简单的http请求）-实例](#)[亿级 Elasticsearch 性能优化](#)[Linux添加新硬盘、分区、格式化、自动挂载](#)[Jmeter之仿真高并发测试（集合点）](#)[快速部署DBus体验实时数据流计算](#)[生产SparkStreaming数据零丢失最佳实践\(含代码\)](#)[Spark RDD转换成DataFrame的两种方式](#)[大数据分布式平台Hadoop2.7.7 + Spark2.2.2搭建](#)[Java代码实现发送邮件](#)[如何在linux下解压缩rar格式的文件压缩包](#)[用Windows Storage Server 2008做iSCSI存储服务器](#)[MBR与GPT分区格式\(实例-创建大于2TB的分区\)](#)[体验vSphere 6之5-添加虚拟交换机及iSCSI存储](#)[为VMware ESXi主机添加本地存储的过程及注意事项 - ...](#)[Spark Thrift JDBCServer应用场景解析与实战案例](#)[Spark SQL常见4种数据源\(详细\)](#)[SparkStreaming的实战案例](#)[Spark on Yarn with Hive实战案例与常见问题解决](#)在线  
客服