[6:21 AM] Kevin Ou

有时间就看看 <https://www.youtube.com/watch?v=_0Wpwj_gvzg>

​

[6:22 AM] Kevin Ou

想你做一个关于rowgroupsize on parquet 的测试。

[6:23 AM] Kevin Ou

把3 gb gzip 的fwcdr 转成parquet 在 rowgroupsize 256mb 和 512mb两种情况测试select 的速度。

​

[6:24 AM] Kevin Ou

你要先看明白parquet ->rowgroupsize->hdfs chunk size  再试

[6:25 AM] Kevin Ou

bigdatamr10 ba 和ka 的folder

​

[6:25 AM] Kevin Ou

 做job 就功能性能都要涉及

​

[6:26 AM] Kevin Ou

例如今天send 给barry 个job

​

[6:26 AM] Kevin Ou

一个简单的mapping

https://stackoverflow.com/questions/41700231/spark-parquet-statisticsmin-max-integration

<https://github.com/apache/parquet-mr/tree/master/parquet-tools>

hadoop jar /home/mapr/pq\_tools/parquet-tools-1.9.0.jar cat /HDS\_VOL\_TMP/test\_par/part-00000-90f85898-6484-43bc-8a33-6846eb1356f3-c000.snappy.parquet | head -500

hadoop jar /home/mapr/pq\_tools/parquet-tools-1.9.0.jar meta /HDS\_VOL\_TMP/test\_par/part-00000-90f85898-6484-43bc-8a33-6846eb1356f3-c000.snappy.parquet | head -500

hadoop jar /home/mapr/pq\_tools/parquet-tools-1.9.0.jar dump -n /HDS\_VOL\_TMP/test\_par/part-00000-90f85898-6484-43bc-8a33-6846eb1356f3-c000.snappy.parquet | head -500

hadoop jar /home/mapr/pq\_tools/parquet-tools-1.9.0.jar dump -n /HDS\_VOL\_TMP/pqtest/part-01468-f3915d1a-83b5-4166-bde3-1c44c09d9260-c000.gz.parquet | head -500

<https://parquet.apache.org/documentation/latest/>

<https://parquet.apache.org/>

http://blogs.quovantis.com/how-to-convert-csv-to-parquet-files/

no:

job sthehpn

<https://www.youtube.com/watch?v=_0Wpwj_gvzg>

[6:22 AM] Kevin Ou

想你做一个关于rowgroupsize on parquet 的测试。

[6:23 AM] Kevin Ou

把3 gb gzip 的fwcdr 转成parquet 在 rowgroupsize 256mb 和 512mb两种情况测试select 的速度。

​

[6:24 AM] Kevin Ou

你要先看明白parquet ->rowgroupsize->hdfs chunk size  再试

java jar ./parquet-tools-<VERSION>.jar –help

<https://translate.google.com/translate?hl=zh-CN&sl=en&u=http://spark.apache.org/docs/latest/rdd-programming-guide.html&prev=search&pto=aue>

bigdatamr10 ba 和ka 的folder

Kevin.txt

parquet-reader --only-metadata --json --columns=0,1,2 widey\_event\_visit\_start\_datetime\_sorted.pq

sc.setLogLevel("INFO")

parquet.strings.signed-min-max.enabled

row\_group\_offsets

Parquet parameters are part of the hadoop options and can be set before the parquet write command like this:

val sc : SparkContext // An existing SparkContext.

sc.hadoopConfiguration.setInt("parquet.block.size", 1024 \* 1024 \* 1024)

Property Name Default Meaning

|  |
| --- |
|  |
| spark.sql.parquet.binaryAsString | false | 一些其它的Parquet-producing系统，特别是Impala和其它版本的Spark SQL，当写出Parquet模式的时候，二进制数据和字符串之间无法区分。这个标记告诉Spark SQL将二进制数据解释为字符串来提供这些系统的兼容性。 |
| spark.sql.parquet.cacheMetadata | true | 打开parquet元数据的缓存，可以提高静态数据的查询速度 |
| spark.sql.parquet.compression.codec | gzip | 设置写parquet文件时的压缩算法，可以接受的值包括：uncompressed, snappy, gzip, lzo |
| spark.sql.parquet.filterPushdown | false | 打开Parquet过滤器的pushdown优化。因为已知的Paruet错误，这个特征默认是关闭的。如果你的表不包含任何空的字符串或者二进制列，打开这个特征仍是安全的 |
| spark.sql.hive.convertMetastoreParquet | true | 当设置为false时，Spark SQL将使用Hive SerDe代替内置的支持 |

hdfs fsck /tmp/test.txt -files -blocks

hadoop fs -D dfs.blocksize=268435456 -copyFromLocal /tmp/test/payroll-april10.csv blksize/payroll-april10.csv

Hi.

I've a file with 128Mb block size

I'd like to change an existing file's blocksize using:

hdfs dfs -mv /user/myfile.txt /tmp

hdfs dfs -D dfs.blocksize=268435456 -cp /tmp/myfile.txt /user

It works

When I try to use a distcp, with -p to preserve original file's attributes, target file's blocksize doesn't change

hadoop distcp -p -D dfs.block.size=268435456 /tmp/myfile.txt /user/myfile.txt

Can't understand where am I wrong

我很难找到如何在Spark API中指定镶木地板文件编写器的行组大小。

我找到了一种方法，即使用具有此选项的快速镶木地板python模块：

来自fastparquet import write

write有参数：  row\_group\_offsets

此外，row\_group大小的最佳数量是多少？

由于快速镶木地板，我做了一些实验。例如，选择100万的row\_groupsize比10 000快10倍。但如果我选择超过100万，它开始减慢我的简单查询。

提前感谢您的帮助

Parquet参数是hadoop选项的一部分，可以在镶木地板写入命令之前设置，如下所示：

val sc : SparkContext // An existing SparkContext.

sc.hadoopConfiguration.setInt("parquet.block.size", 1024 \* 1024 \* 1024)

<https://stackoverflow.com/questions/32356143/what-does-setmaster-local-mean-in-spark>

<https://parquet.apache.org/> video and youtobe video

<https://github.com/search?q=row_group_offsets&type=Code>

parquet:

<https://www.google.com/search?rlz=1C1GCEU_zh-CNHK857HK857&ei=-hwVX_fOLtaQr7wPzqSM0As&q=parquet+file+jianshu&oq=parquet+file+jianshu&gs_lcp=CgZwc3ktYWIQAzoCCAA6BQgAEIsDOgUIIRCgAToHCCEQChCgAVConAZY1asGYKOzBmgAcAB4AYAB1QGIAe0MkgEFMi45LjGYAQCgAQGqAQdnd3Mtd2l6uAECwAEB&sclient=psy-ab&ved=0ahUKEwj3s8XL_trqAhVWyIsBHU4SA7oQ4dUDCAw&uact=5>

<https://www.google.com/search?rlz=1C1GCEU_zh-CNHK857HK857&ei=KHcVX8eQDOaGr7wPxsCuoAo&q=parquet&oq=parquet&gs_lcp=CgZwc3ktYWIQAzICCAAyAggAMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIADICCABQwRRYwRRgmBZoAHAAeACAAaABiAGCApIBAzEuMZgBAKABAaoBB2d3cy13aXrAAQE&sclient=psy-ab&ved=0ahUKEwjHxMTL1NvqAhVmw4sBHUagC6QQ4dUDCAw&uact=5>

<https://stackoverflow.com/questions/39187622/how-do-you-control-the-size-of-the-output-file/39282893>

hadoop fs -stat

val blockSize= 1024\*1024\*100

sc.hadoopConfiguration.setInt("dfs.blocksize", blockSize)

sc.hadoopConfiguration.setInt("parquet.block.size",blockSize)

df1.write.partitionBy("DATE").parquet("output\_file\_path")

val blocksize:Int = 1024\*1024\*1024

sc.hadoopConfiguration.setInt("dfs.blocksize", blocksize) //also tried dfs.block.size

sc.hadoopConfiguration.setInt("parquet.block.size", blocksize)

val df = spark.read.csv("/path/to/testfile3.txt")

df.write.parquet("/path/to/output/")

sc.\_jsc.hadoopConfiguration().setInt("dfs.blocksize", some\_value)

sc.\_jsc.hadoopConfiguration().setInt("parquet.block.size", some\_value)

<https://webcache.googleusercontent.com/search?q=cache:1Zovw1ZRRQMJ:https://blog.csdn.net/qq_27639777/article/details/81069893+&cd=3&hl=zh-CN&ct=clnk&gl=hk>













