



链滴

使用 spark 从 kafka 消费数据写入 hive 动态分区表 (二)

作者: [ludengke95](#)

原文链接: <https://ld246.com/article/1558346728369>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



使用spark从kafka消费数据写入hive动态分区表（二）

上次咱们说到数据从kafka到hive，也从hive非分区表到分区表的迁移。经过测试发现曲线救国的方虽然kafka到hive快了，但是hive非分区表到分区表贼慢，再一次难受，不着急咱们慢慢来分析原因。

分析日志

拿到日志文件看看什么日志最多，什么操作最耗时间。日志大体分为一下几类：

- 初始化日志：这个耗时忽略不计（包括分配executor，创建临时文件夹啥的等等）。
- task日志：这个贼重要，可以知道那个task执行失败，task位于那个几点，总共有多少任务，每个任务的执行时间，每个任务的分区（具体是从哪里看，下面两行日志一目了然呀）。

```
Starting task 53.0 in stage 0.0 (TID 47, IP, executor 5, partition 53, PROCESS_LOCAL, 4920 byte )  
Finished task 53.0 in stage 0.0 (TID 40) in 76 ms on IP(executor 5) (24/3135)
```

- Rename日志：就是讲数据写入的临时文件重命名为对应的hive表数据文件。

```
metadata.Hive: Renaming src: hdfs://分区表文件存储路径/.hive-staging_hive_2019-05-20_10-24  
43_342_2230325038880850463-1/-ext-10000/tradedate=2017-12-02/part-00086-e563b05e-  
202-4510-8951-3d05d246c279.c000, dest: hdfs://分区表文件存储路径/tradedate=2017-12-02/  
art-00086-e563b05e-3202-4510-8951-3d05d246c279.c000, Status:true
```

经过对比时间发现：task执行时间也就几十秒，Rename执行时间几分钟，甚至十几分钟。

对症下药

问题找到就是Rename阶段时间长拖慢了整个进度，想办法解决呗。

- 原版sql执行: `insert into partition(分区字段) select * from`,可以理解为把非分区表的文件生成份,然后把备份rename成对应表的分区的数据文件(当然其实内部不是这样的,这样只是方便理解如果是这样的分区字段怎么办呢?有兴趣可以去了解一下,这个sql是怎么执行的)。
- 新版sql执行: `insert into partition(分区字段) select * from table DISTRIBUTE BY 分区字段`。DISTRIBUTE BY 就可以理解为mysql中的group by,就是分组,但是这个分组可不一样,这个会大大减少reduce的数量(相比之前的sql,每个文件中的每个分区字段对应一个reduce结果集)。现在的reduce数量 \approx 之前的reduce数量/文件数。

我为什么这么着重的说reduce,因为reduce的数量——对应hive表数据文件的数量(仅仅针对本次入涉及的分区)。reduce的数量减少,那么临时文件数量减少,那么rename的对象就少,大大减少了rename所消耗的时间。

重点

hive非分区表到分区表的数据迁移sql:

`insert into partition(分区字段) select * from table DISTRIBUTE BY 分区字段`