

99 家居数据统计问题总结

Case Study

作者 冯海雄

程序中心研发部大数据解决方案组

2015 年 4 月 14 日

- flume-ng 接收日志卡死问题
- flume 接收日志时发现时间区间不同的日志信息, 经仔细排查是由于接收数据的时间窗口移动的问题导致的
- flume 部署的时候遇到资源加载冲突的问题, 剔除无关的 jar 包之后正常
- flume 打包默认打包了许多依赖项, 在 pom 中的依赖项添加 `<scope>provided</scope>` 属性重新打包即可

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

- 备份表当前表
- 清空表当前表
- 从备份表中查出符合条件的数据插入当前表中

现在的操作方式是

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

- 以 **overwrite** 的方式写表保证幂等性
- **overwrite** 不是一个原子操作具体测试流程如下
- 启动一个任务反复执行 **overwrite** 写表操作
- 启动一个任务循查询表中的记录数量发现记录数量保持不变
- 再启动一个任务执行 **overwrite** 写表操作观察表中的数据发现记录数量翻倍。

数据清洗的思路

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统

计部分

保证幂等性的操作方
式

数据清洗

数据处理

数据统计部分

按照 wordcount 的思路进行 mapreduce 任务在 reduce 阶段输出每个 key 的一个 value 数据清洗过程中遇到的问题:

- reduce 阶段输出丢弃了重复的所有记录导致数据大量丢失,原因是 reduce 阶段的逻辑判断错误,将有重复的数据都去掉了。
- mr 提交任务太慢目前数据清洗已经用 spark 重构,结果和 mr 程序清洗的结果一致,效率比 mr 程序清洗快一个数量级。

数据处理主要功能

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统
计部分

保证幂等性的操作方
式

数据清洗

数据处理

数据统计部分

对数据进行预处理,执行的时候会更新新增表并输出

- startuplogimeiyymmdd
- startuploguidyymmdd
- validlogimeiyymmdd
- validloguidyymmdd

出现的问题以及运行逻辑

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

数据处理阶段数据丢失,主要是新增表的丢失。原因是新增表更新时的逻辑如下:

- 删除旧的新增备份表。
- 对新增表进行备份
- 清空新增表
- 从备份表中取出今日新增用户,今日登录的历史用户和今日未登录的历史用户,插入到新增表。
- 删除备份表

问题的原因

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

在上述过程中如果失败退出,会使得新增表数据不完整的,而重试的时候删除了上一次生成的备份表导致新增表数据彻底丢失。实际上,这一整个过程需要当做一个原子操作,如果在处理过程中出错,必须将数据回溯到处理前的状态。

更新数据处理的逻辑如下

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

- 检查是否存在旧的备份表,如果存在,则这一份数据是完整的新增表的数据,略过第二步。
- 对新增表进行备份,生成备份表
- 清空新增表
- 从备份表中取出今日新增用户,今日登录的历史用户和今日未登录的历史用户,插入到新增表。
- 删除备份表

数据处理结束后的检查

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

- 为保证数据处理结果中没有 NULL 我们检查了计算结果中是否有 NULL 如果有 NULL 则重新计算
- 为保证新增表数据不减少我们做了新增表的检查,如果发现新增表有减少的清空则重新计算
- 在删除备份表阶段如果运行失败且重新运行后新增表是递增状态则计算是否会丢数据是不确定的

spark-sql 问题描述

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

spark-sql 执行任务后产生的结果文件数量过多,造成用户的文件数量达到上限,导致文件不能创建,任务无法提交

spark-sql 文件太多的解决办法

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

- 对有 `reduce` 输出的操作可以设置 `spark-sql` 的输出数量
- `set mapred.reduce.tasks = 15;`
- `set spark.sql.shuffle.partitions=15;`
- 对于解析之后没有 `reduce` 输出的最终结果的文件数量依赖于源表的文件数量
- 文件名出现 `copy` 的原因, 同一张表多次插入时会产生这样的文件名, 输出文件的数量由上面的设置来决定
- 对于一些复杂关联表的操作输出文件数量具有不确定性。

分时统计主要功能

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

统计某时段内的新增用户数、新增有效用户数和截至当前时段的活跃用户数以及本时段内活跃用户数, 和本时段内应用被开启的次数

分时统计出现的问题

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

在计算分时统计时由于表关联不当导致结果表中日活跃用户数据错误计算分时统计表的统计逻辑如下：

- 计算当前时段的新增用户所有启动次数得到表 a
- 计算当前时段的新增有效用户数得到表 b
- 计算截至当前时段的活跃用户数得到表 c
- 计算当前时段的新增用户数得到表 d

分时统计错误的原因

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统
计部分

保证幂等性的操作方
式

数据清洗

数据处理

数据统计部分

数据丢失主要原因是 **a** 中只包含了但前时段的用户信息, **c** 表中包含了截至但前时段的多有用户信息, 当 **a** 表为左表的时候对导致其他时间段的数据不能出现在最终结果中

修正后的分时计算表的统计逻辑

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

- 极端截至当前时段的活跃用户数得到表 c
- 计算当前时段的新增用户所有启动次数得到表 a
- 计算当前时段的新增有效用户数得到表 b
- 计算当前时段的新增用户数得到表 d

用户概况出现的问题

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统计部分

保证幂等性的操作方式

数据清洗

数据处理

数据统计部分

同一天不同时段内如果用户的版本发生变化则新增表中以最新时间戳的版本信息为准

主要逻辑如下：

- 计算本日累计启动用户数
- 计算本日累计活跃用户数

按天统计主要逻辑如下：

- 计算本日累计活跃用户数
- 计算本日累计启动用户数

区域统计区域统计逻辑如下

99 家居数据统

计问题总结

冯海雄

数据接收部分

数据处理与统
计部分

保证幂等性的操作方
式

数据清洗

数据处理

数据统计部分

- 计算 IMEI 的区域统计信息覆盖写入结果表
- 计算 UID 的区域统计信息覆盖写入结果表

修正后区域统计逻辑

- 计算 IMEI 的区域统计信息覆盖写入结果表
- 计算 UID 的区域统计信息增量写入结果表