

数据倾斜问题解决

原因

数据倾斜就是我们在计算数据的时候，数据的分散度不够，导致大量的数据集中到了一台或者几台机器上计算，这些数据的计算速度远远低于平均计算速度，导致整个计算过程过慢。

- shuffle的时候按key将数据分散到reduce端，不同reduce获得的key数量差异太大，导致某些reduce结束，某些还差很多
- 数据方面：数据表设计冲突，数据清洗不够完善

我们举一个例子，就说数据默认值的设计吧，假设我们有两张表：

- user（用户信息表）：userid, register_ip
- ip（IP表）：ip, register_user_cnt

这可能是两个不同的人开发的数据表，如果我们的数据规范不太完善的话，会出现一种情况，user表中的register_ip字段，如果获取不到这个信息，我们默认为null，但是在ip表中，我们在统计这个值的时候，为了方便，我们把获取不到ip的用户，统一认为他们的ip为0。

两边其实都没有错的，但是一旦我们做关联了会出现什么情况，这个任务会在做关联的阶段，也就是sql的on的阶段卡死。

- 业务角度

数据往往和业务是强相关的，业务的场景直接影响到了数据的分布。

再举一个例子，比如就说订单场景吧，我们在某一天在北京和上海两个城市多了强力的推广，结果可能是这两个城市的订单量增长了10000%，其余城市的数据量不变。

然后我们要统计不同城市的订单情况，这样，一做group操作，可能直接就数据倾斜了。

数据倾斜发生情况

- Hadoop
 - 有一个多几个reduce卡住
 - 各种container报错OOM
 - 读写的数据量极大，至少远远超过其它正常的reduce
- Spark
 - Executor lost, OOM, Shuffle过程出错
 - Driver OOM
 - 单个Executor执行时间特别久，整体卡在某个阶段不能结束
 - 正常运行的任务突然失败

问题解决

业务解决

- 数据预处理，删除异常数据，比如上面例子中IP为0的数据
- 分布不均的数据处理，先对key做hash处理，让数据均匀分配到reduce1，将reduce1结果按key分配到reduce2

Hadoop优化

- map join
- count distinct的操作，先转成group，再count
- hive.groupby.skewindata=true
- 设置map端输出、中间结果压缩。（不完全是解决数据倾斜的问题，但是减少了IO读写和网络传输，能提高很多效率）

Spark优化

- RDD压缩
- 设置合理的driver内存
- 参考Hive