Druid建议

段卫东

# 1.数据建模

问：相同的统计metric，但是有二三十个维度，我是需要按照查询需求把这些维度拆分开成多张datasource呢，还是直接弄成一个datasource，因为druid不支持表连接，如果拆分开datasource的话，在查询的时候遇到跨datasource的查询，就增加了查询的难度，我这里选择的是一张大的datasource，请问是否合理呢？

答：合理，druid不支持join

但是druid不提倡大的datasource，如果有些维度字段是不进行关联查询的，就尽可能的把字段拆开进行处理。

离线索引的时候，segment的生成数量要分片，因为分片的数量是跟hadoop任务的reduce数量相对应的。如果不分片，生成一个segment，早造成hadoop任务执行非常慢。

# 2.数据结构版本迭代问题

问:

数据结构上，涉及到版本迭代的问题，每次版本迭代，都需要新增一些字段，而新增的字段，在老数据里面是没有新增的字段的。例如：dim1这个字段是新增的，新数据里面的value取1,2，而同时假设老数据里面dim1的值默认是1，这样要求，dim1=1做筛选的时候，筛选出新旧数据，dim1=2,只筛出新数据。这种情况，有没有固定的解决方案呢？

问:

我们打算新增一个version字段，来负责维护每次版本迭代新增的字段，同时，在业务查询的SQL中，向用户屏蔽这种版本迭代出现的差异

答：应该可以。

druid本身是列式存储的，所以不对不存在的维度进行null筛选，这样的话，druid本身并不支持上述问题的解决方案。并且也不建议每次版本迭代都重新压索引的数据。

druid本身支持从自身流出并且流回索引的操作，可以减少字段，变更时间粒度，但是不支持增加字段的操作。

# 3.慢查询

关注慢查询，然后通过metric去优化，通过queryId去从头到尾关联查询，查出慢的原因所在。可以通过采集日志，druid也提供http接口，来获取基本的日志信息。为了方便查询问题，可以把日志级别修改成info级别，利于定位问题和优化，否则druid就是一个黑盒子，细节的东西我们都不知道。

# 4.数据优化