**变更记录**

| **变更日期** | **变更内容** | **变更原因** | **变更人** |
| --- | --- | --- | --- |
| 2016-4-21 |  | 新建文档 | 张若弛 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 交叉表实现总结

## 基本目标

重构交叉表，提交运行效率（基本达成）

## 交叉表介绍

原始数据（SQL结果）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **部门** | **人员** | **区域** | **省份** | **指标1** |
| 销售部 | 张三 | 华东区 | 山东 | 100 |
| 销售部 | 张三 | 华东区 | 上海 | 90 |
| 销售部 | 李四 | 华东区 | 上海 | 98 |
| 销售部 | 张三 | 华北区 | 北京 | 99 |
| 销售部 | 李四 | 华北区 | 天津 | 88 |
|  |  |  |  |  |

交叉表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | 华东区 | | 华北区 | |
| 山东 | 上海 | 北京 | 天津 |
| 销售部 | 张三 | 100 | 90 | 99 |  |
| 李四 |  | 98 |  | 88 |

## 原有算法中存在的一些问题

* 原有交叉表对内存的使用控制的不够精细，32位操作系统上10w明细数据会堆溢出
* 原有交叉表整体结构耦合性太强，内存中维护了一个GRID表格结构，在增加维度的过程中整体结构会不断的调整。
* 原有组合表算法比较先进，表数据与表结构解耦，但依赖数据库表的做法增加了算法的复杂度，提高了后期维护成本，并且为将来分布式部署数据库、扩展新的数据库类型等带来潜在的问题来源。
* 原有组合表中存在大量字符串操作，如连接，拆分，对比等对性能造成一定影响。

## 需求对新算法的要求

* 禁止使用数据库表参与运算
* 32位操作系统上能上10w明细不崩溃

## 新算法实现思路

* 交叉表的结构与数据分别存储， 表结构由左侧与上方两个树形结构确定， 指标数据单独存储，在对交叉表进行排序、固定成员、TopN等动作时，仅调整树形结构即可。



* 对DataSet仅遍历一次，直接生成交叉表的主体结构，对于左侧树结构，直接利用相邻两行的数据值对比生成，速度更快。
* 交叉表结构中左侧与上方的树节点都赋予int型唯一标记，即上图树结构维度名称后边括号中的值。 含义如下：

|  |  |  |
| --- | --- | --- |
| **标记** | **含义** |  |
| 0 | 无意义维度，如固定成员中的占位空行 |  |
| 1 | 根节点，根节点的合计定位值 |  |
| 其他 | 维度节点，唯一标识，用户定位指标数据与小计数据 |  |

* 每个指标的数据都存储在HashMap中，无论DataSet明细结果，还是内存中计算出来的小计值。具体的定位方式是将左侧维度的标记值与上方维度的标记值按规则位运算后确定HashMap的Key（Integer类型）。这样做可提高运行效率且减少内存使用，缺点是行列维数个数受限。

|  |  |  |
| --- | --- | --- |
| **左侧维度标记** | **上方维度标记** | **组合后数据含义** |
| 叶子节点 | 叶子节点 | 明细数据 |
| 非叶子节点 | 叶子节点 | 行小计数据 |
| 叶子节点 | 非叶子节点 | 列小计数据 |

32位操作系统下使用情况示意图。



## 交叉表部分类图

主要表的类图



## 交叉表关键时序图



## 更大数据量的优化思路

在数据量进一步加大的情况下的解决方案（控制内存占用， 降低对执行时间的要求）

1. 建议与订阅功能配合，实时操作（可沿用现有算法）需增加硬件要求
2. 行维，列维单独处理，通过sql独立取出
3. 加载排序，TopN, 固定成员等操作相关的指标/小计数据（数据库执行），重新计算维度位置
4. 分区域运算，填充
5. 分区域运算中需要频繁发生数据库操作，读取明细及小计数据
6. 需要将结果转存入数据库

## 前期思路记录

第一版优化思路优势：

* 省去DataSet存储部分，减少内存占用
* 由数据库结果直接生成交叉模型，运行效率提高很多

第一版优化思路劣势：

* 筛选操作下CrossModel运算逻辑会比DataSet上筛选复杂度高很多，不宜于后期维护
* 筛选操作下需要生成CrossModel维度部分子集，也会占用内存，在筛选条件不明确情况下内存使用甚至会超过使用DataSet的占用情况