代码思路说明

本文件包含两个思路。两个思路适用于不同的环境设置（内存容量等）。两个思路在考题提供的数据上运行时间相近，最优的跑分是思路一的。

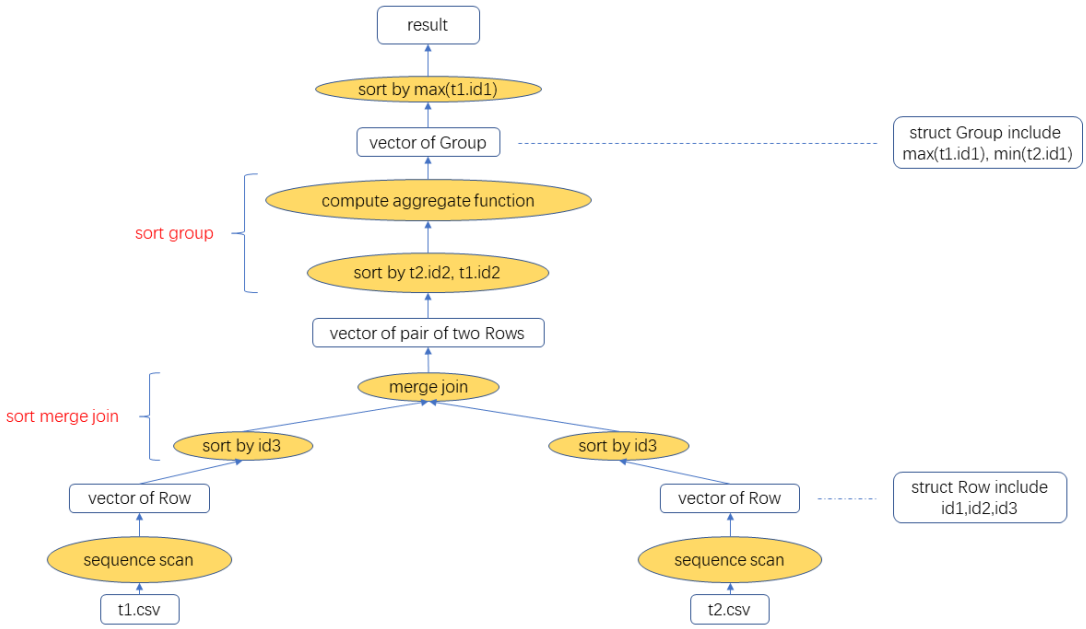
思路一

将两个表从csv文件读取全部记录到内存。

实现查询语句主要的模块采用的方法：

- join： 采用 sort-merge join。t1和t2两个表分别按照各自的id3从小到大排序。然后根据这两个有序的序列进行归并操作。当两行记录的id3字段相等时（满足join谓语条件），就把这两行作为一个pair放入join的结果中。
- group by：采用 sort group。将join结果按照 t2.id2, t1.id2 排序，则属于同一个group的行聚集在一起。遍历每一行，若当前行属于同一个group，则更新聚集函数（max(t1.id1), min(t2.id1)）；若不属于同一个group，则添加新的group，继续向后遍历。这样得到的group是按照 t2.id2, t1.id2 有序的。
- order by：采用 stable sort。group by 部分已经得到了最终选择的 max(t1.id1) 和 min(t2.id1)，并且已经在 t2.id2, t1.id2 有序，所以只需要在 max(t1.id1) 之上稳定排序即可完成order by。

查询执行计划图示：



代码文件：sort_merge_join_sort_group_stable_sort.cpp

c++11

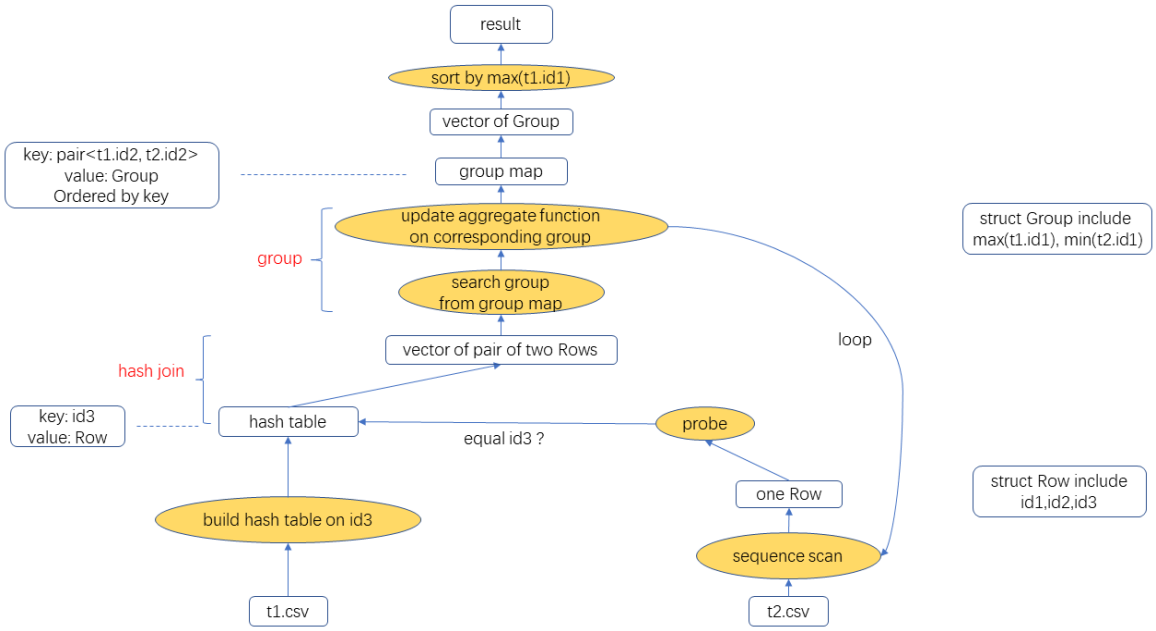
思路二

将t1表读入内存，并建立hash索引。按顺序扫描t2表，每次读取一行，计算join和group。

实现查询语句主要的模块采用的方法：

- join： 采用 hash join。将t1表在id3字段上建立hash索引。针对t2表的每一行记录，用id3字段探测t1表的hash索引，若存在相等的id3，则将相应行放入join的结果中。
- group by：采用 map 数据结构。由于group by 的字段为 t1.id2, t2.id2，所以建立group map，key 为 pair<t1.id2, t2.id2> ，value为 struct Group，value是按照key有序存放的。Group里面包含了最终选择的 max(t1.id1)，min(t2.id1)。每次从join结果中通过key查找相应的Group，更新聚集函数值。
- order by：采用 stable sort。group by 部分已经得到了最终选择的 max(t1.id1) 和 min(t2.id1)，Group也是在 t2.id2, t1.id2 上有序的，所以只需要在 max(t1.id1) 之上稳定排序即可完成order by。

查询执行计划图示：



代码文件：hash_join_group_map_stable_sort.cpp