SemiJoin功能设计文档

修订历史

版本	修订日期	修订描述	作者	备注
0.1	2015-12-24	初稿	樊秋实	无

1 系统设计

1.1 综述

SemiJoin功能是对OceanBase数据库新增的一种两表连接操作，属于Inner Join的一种。与OceanBase本身的连接操作不同，它在处理一张小表和一张超大表的连接时能够显著的提升连接的效率。

1.2 名词解释

SemiJoin（半连接)： 一种特殊的数据库表连接操作。
Inner Join： 内连接，传统数据库连接操作之一。
超大表： 记录数在千万甚至亿级别以上的表。

1.3 功能

SemiJoin，又叫半连接，是一种对两张表做连接的优化方法，主要针对的是分布式关系数据库中一张大表和一张小表的Inner Join。原理是通过小表在连接列上的数据，来对大表进行过滤。目的是减少大表在网络上的传输量，从而达到减少连接执行时间的效果。
用户可以在sql语句中使用hint来指定某两张表做SemiJoin。
例： select /*+SEMI_JOIN(A,B,A.c2,B.c3)*/ A.c1, B.c1 from A inner join B on A.c2=B.c3 ;（其中/* */里面的内容为hint）

1.4 性能指标

在相同的环境下，使用SemiJoin的连接操作所用的时间远小于使用OceanBase本身的连接操作所用时间。

2 模块设计

SemiJoin的设计框架分为三个部分，第一部分为hint的解析；第二部分为对小表的处理；第三部分为对大表的过滤。代码实现的流程则按照传统数据库对sql语句的处理流程：首先把sql语句解析成语法树，然后根据语法树生成逻辑计划和物理计划，最后执行该物理计划树的open函数，通过不停的调用物理计划树的get_next_row函数将两张表的连接结果返回给客户端。

2.1 hint解析子模块设计

2.1.1 结构

hint的结构设计：/*+SEMI_JOIN(parameter1，parameter2，parameter3，parameter4)*/

其中parameter1为小表的表名，parameter2为大表的表名，parameter3为小表的连接列的列名，parameter4为大表的连接列的列名。注意：该四个参数的顺序不要写错，并且该四个参数的值要是有意义的，不能是表（列）的别名或者是不存在的表（列）名。如果不符合规范，hint是不起作用的。

2.1.2 关键算法

新增数据结构ObSemiTableList用来存储hint里面的所有信息。在生成物理计划之前通过逻辑计划获得ObSemiTableList信息，根据该信息判断用户是否使用了正确的hint。如果是，则生成新增的SemiJoin的物理计划，否则，生成OceanBase原有的内连接的物理计划。

2.1.3 流程

新增词法节点——>新增语法节点——>根据语法树将hint信息存到逻辑计划结构里——>编写接口get_query_hint用来从逻辑计划中获得hint信息。

2.2 小表处理子模块设计

2.2.1 结构

新增物理操作符ObSemiLeftJoin负责所有对小表的操作。

2.2.2 关键算法

do_sort函数：对小表的数据按照连接列进行排序
do_distinct函数：对小表的已排序的连接列的数据去重，并将去重后的结果存到数组里

2.2.3 流程

构造物理操作符ObSemiLeftJoin——>执行该操作符的open函数——>将小表的所有数据缓存到操作符里面——>对小表的所有数据按照连接列排序——>对小表的连接列数据去重，缓存去重的结果。

2.3 大表过滤子模块设计

2.3.1 结构

通过修改物理操作符ObMergeJoin和操作符ObTableRpcScan完成对大表的过滤操作。

2.3.2 关键算法

在函数change_right_semi_join_op里获取小表在连接列上的所有不同值，并且根据这些不同值修改对大表的scan操作。

2.3.3 流程

执行物理操作符ObMergeJoin的open函数——>执行物理操作符ObSemiLeftJoin的open函数——>获得小表在连接列上的所有不同值——>根据这些不同值修改对大表scan的操作符ObTableRpcScan——>执行操作符ObTableRpcScan的open函数。

3 模块接口

3.1对外接口

ObTransformer::gen_phy_semi_join（ ）：生成SemiJoin物理计划的函数。

3.2内部接口

ObSemiLeftJoin::do_sort（ ）：对小表的所有数据按照连接列排序。
ObSemiLeftJoin::do_distinct（ ）：对小表的连接列数据去重，缓存去重的结果。
ObMergeJoin::set_is_semi_join（ ）：设置当前的连接操作是否为SemiJoin。
ObMergeJoin::do_semi_open（ ）：SemiJoin操作符的open函数。
ObMergeJoin::change_right_semi_join_op（ ）：根据小表在连接列上的不同值修改对大表scan的操作符。

4 使用限制条件和注意事项

1. 多表连接时，只能指定前两张表做semi-join。
2. 只有两张表原来是inner join 的时候，才能指定这两张表做semi-join。

3. 只有当大表的数据远远超过小表，并且两张表在连接列上的交集很小的时候，semi-join优化的效果最好，其他情况下的优化效果可能不是很好。
4. 插入数据后，最好在每日合并后，再做查询操作，这样的优化效果会好一点。

5 性能测试结果分析

todo