BloomFilter Join 功能设计文档

修订历史

版本	修订日期	修订描述	作者	备注
Cedar 0.2	2016-07-01	BloomFilter Join 功能设计文档	茅潇潇	无

1 需求分析

Cedar是由华东师范大学数据科学与工程研究院基于OceanBase 0.4.2 研发的可扩展的关系数据库 ，实现了巨大数据量上的跨行跨表事务。业务中存在大量的多表连接查询，在处理多表连接时，目前Cedar 0.1进行Merge Join运算时，由Merge Server（MS）分发请求给相应的 Chunk Server（CS），CS合并UPS上的增量数据后把所有数据返回给MS，MS进行排序后做 Merge 操作。若对大表进行查询，CS将传输大量不会产生连接关系的无效数据到MS，浪费了大量的网络传输时间和无效数据的排序时间等,导致查询缓慢。

针对这一问题，Cedar 0.2版本实现了一种基于布隆过滤器的连接优化算法（Bloomfilter Join）。该算法并不将右表数据全部数据发送到计算节点，而是使用布隆过滤器对右表数据进行过滤，再对过滤后的数据进行排序归并连接，这样网络通讯代价将会大大减少，在过滤后的数据集上进行排序操作所消耗的内存资源也会下降。

2 适用场景

当过滤掉的右表数据大于500万行时，BloomFilter Join能发挥较好的性能优势。

3 使用方法

SELECT /*+ JOIN(bloomfilter_join,merge_join,…)*/ <查询内容> from <表名> left (inner, right, full outer) join <表名> on <join条件> ;

> /*+ join (bloomfilter_join, merge_join, bloomfilter_join, merge_join) */括号内需要符合“join _ type (, join_ type)”规则，按照join顺序，表明两两表之间的join使用的算法规则。若hint中的join类型个数多于后面数据表的两两连接个数，则忽略hint中多余的join类型；若hint中的join类型个数少于后面数据表的两两连接个数，则后面未指定的连接都默认执行merge join。

4 原理介绍

基于布隆过滤器的连接算法将关系S在公共属性B上的投影∏B(S)表示在一个布隆过滤器中。假设，节点1上的关系R和节点2上的关系S，在属性R.A=S.B上做连接操作，它的工作流程如下：首先，节点2将关系S的每个元组在属性B上的值插入到一个布隆过滤器BFs里；然后，将BFs发送到节点1，节点1根据BFs过滤掉关系R中不符合连接条件的记录，把R中符合连接条件的记录传送给节点2；最后，在节点2上过滤掉布隆过滤器误判的记录，并进行连接操作。

5 功能简述

BloomFilter Join充分利用了布隆过滤器低空间代价和快速响应的特点，通过对右表数据使用布隆过滤器进行过滤，减少了分布式环境下不必要数据的网络传输代价，降低了数据操作带来的内存资源的消耗，在连接列的选择率较低的情况下显著提高了连接操作的处理性能。

6 设计思路

6.1 子功能模块划分

> BloomFilter Join分为词法语法、逻辑计划、物理计划和物理操作符四个子功能模块。
>
> 1. 词法语法解析层
>   增加Select中的Hint的解析，不涉及其他部分的词法语法解析，影响范围是Select Hint的语法解析部分。
> 2. 逻辑计划层
>   修改Select中的Hint的逻辑计划生成部分的代码，未动其他部分的逻辑计划，所以影响范围限于Select Hint的逻辑计划生成部分。
> 3. 物理计划层
>   修改BloomFilter Join的物理计划生成部分的代码，没有动其他部分的物理计划生成，影响范围限于BloomFilter Join的物理计划生成部分。
> 4. 物理操作符
>   增加ObBloomfilterJoin操作符的代码，并未动其他物理操作符，影响范围限于ObBloomfilterJoin操作符。

6.2 总体设计思路

BloomFilter Join实际分为四个阶段：

1. 将左表的所有数据发送到一台MergerServer上，其中既包括ChunkServer存储的基线数据又包括UpdateServer存储的增量数据；
2. 在MergerServer上根据左表的数据在连接属性上生成布隆过滤器，并将该布隆过滤器作为SQL Expression的一个参数，序列化后传入右表所在的ChunkServer；
3. 右表所在的ChunkServer通过合并UpdateServer上的增量数据获得右表的全部数据后，使用布隆过滤器对数据进行过滤，并将过滤后的数据发送到MergerServer；
4. MergerServer对两张表的数据根据连接类型进行等值连接操作，并把最终结果返回给客户端。

7 参考文献

[1] Bloom B H. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the Acm, 1970, 13(7):422-426.

[2] Chen M S, Hsiao H I, Yu P S. On applying hash filters to improving the execution of multi-join queries[J]. Vldb Journal, 2000, 6(2):121-131.

[3] Mackert L F, Lohman G M. R* Optimizer Validation and Performance Evaluation for Distributed Queries.[C]// Vldb’86 Twelfth International Conference on Very Large Data Bases, August 25-28, 1986, Kyoto, Japan, Proceedings. 1986:84-95.