

文章编号:1001-9081(2010)07-1956-03

基于语义分解的联机分析处理查询并行优化方案

魏莉¹, 杨科华²

(1. 湘南学院 计算机科学系, 湖南 郴州 423000; 2. 湖南大学 计算机与通信学院, 长沙 410082)

(zejzcj002@126.com)

摘要:利用联机分析处理(OLAP)查询中存在的语义关联,对聚集关系与语义分解关系进行了形式化描述,并基于这些关系定义了查询与查询集之间的补集关系,在执行 OLAP 查询集时,可以利用这些关系尽可能地识别查询集中查询的公共部分,并且可以在查询时从多个角度来采取并行优化措施。实验验证表明采用并行优化方案后,系统的整体效率得到了提高。

关键词:联机分析处理;语义关联;语义分解;查询集;并行查询优化

中图分类号: TP311 **文献标志码:** A

Parallel OLAP query optimization method based on semantic decomposition

WEI Li¹, YANG Ke-hua²

(1. Department of Computer Science, Xiangnan University, Chenzhou Hunan 423000, China;

2. School of Computer and Communication, Hunan University, Changsha Hunan 410082, China)

Abstract: This article first gave out formalization definitions of aggregation relation and semantic decomposition relation between the Online Analytical Processing (OLAP) queries. Then the definition of supplementary set between the query and query set was provided. The common query in OLAP set could be found by using these relations and the OLAP queries could be optimized by parallel method from many aspects. Test results show that the system's overall efficiency has been improved by adopting parallel optimization method.

Key words: Online Analytical Processing (OLAP); semantic relation; semantic decomposition; query set; parallel query optimization

0 引言

联机分析处理(Online Analytical Processing, OLAP)技术作为数据仓库中的一种必不可少的技术,是“最终使用者对大规模企业数据直接获取,能动性进行信息分析的过程”,其目的是使使用者对企业全方位状况理解和支持企业的决策。OLAP相关定义中共享多维信息的快速分析环境(Fast Analysis of Shared Multidimensional Information, FASMI)最为人们熟知,在这个定义中,Fast作为一个最重要的因素排在第一位。因此在实际应用中,如果尽快地完成用户所递交的OLAP查询,将有助于提高系统的整体效率,使得使用者更加快速有效地在数据仓库/多维数据集中进行分析。

一种OLAP查询优化方法是对查询的结果进行缓存^[1-2],但这需要额外的空间。近年来,随着并行计算算法的完善和廉价而功能强大的多处理机系统的成熟,使得采用多处理机系统来并行处理OLAP查询成为当前有效提高OLAP查询处理性能的首选技术。文献[3]给出了一种并行处理多维连接和聚集操作的有效方法,但并没有涉及到OLAP的查询方面。文献[4]详细论述了OLAP查询的形式化定义与基于语义的多OLAP查询集优化方案,不过并没有讨论并行相关的查询优化技术。基于上面的分析,本文在对OLAP查询之间关系进行严格的形式化定义的基础上,提出了一种在多处理机上的OLAP查询并行优化方案。

1 OLAP查询相关概念的形式化描述

定义1 OLAP查询。对于一个具有 d 个维,第 i 个维上有 h_i 个层次的多维数据集,其上的OLAP查询可形式化描述为 $Q = [(v_1^1, v_2^1, \dots, v_{h_1}^1), (v_1^2, v_2^2, \dots, v_{h_2}^2), \dots, (v_1^d, v_2^d, \dots, v_{h_d}^d)]$,其中 v_j^i 表示第 i 个维上第 j 个层次上的值。并且 $\forall v_j^i, v_{j'}^i$,如果 $j' > j$,则有 $v_j^i = ' * ' \Rightarrow v_{j'}^i = ' * '$,其中 $*$ 表示“all”。

由定义1中的限制条件可知,我们仅处理具有严格层次性的OLAP查询,而用户所提交的非严格层次性的OLAP查询,均可以转换为一系列的严格层次性查询。例如,对于一个具有两个维:时间(年、月),地点的多维数据集,OLAP查询: $[(2007, *), *]$,即2007年所有月份所有地区的产品销售额是一个具有严格层次性的OLAP查询,而 $[(*, 3), *]$,即每年3月份所有地区的产品销售额则不是,但这个查询可以分解转换为 $[(2006, 3), *], [(2007, 3), *], [(2008, 3), *]$ 等多个具有严格层次性的OLAP查询。

在将所有的OLAP查询均转换为具有严格层次性的OLAP查询后,则可以定义OLAP查询之间的关系。

定义2 OLAP查询之间的 $>$ 关系。对于两个OLAP查询 $Q_1 = [(v_1^1, v_2^1, \dots, v_{h_1}^1), \dots, (v_1^d, v_2^d, \dots, v_{h_d}^d)], Q_2 = [(V_1^1, V_2^1, \dots, V_{h_1}^1), \dots, (V_1^d, V_2^d, \dots, V_{h_d}^d)]$,如果 $\forall v_j^i, V_j^i$,均有 $v_j^i = ' * ' \cup v_j^i = V_j^i$ 成立,则称 Q_2 的查询结果可以作为 Q_1 查询结果的一部分,记为 $Q_2 > Q_1$ 。

收稿日期:2009-12-07;修回日期:2010-03-04。

基金项目:湘南学院科研基金资助项目(09Y028);广东省产学研项目(2007A090302079)。

作者简介:魏莉(1977-),女,河南南阳人,副教授,硕士,主要研究方向:数据库与数据仓库、嵌入式软件;杨科华(1979-),男,湖南新化人,副教授,博士,主要研究方向:数据库与数据仓库、嵌入式软件。

> 关系所定义的是具有严格层次性的 OLAP 查询之间是否存在聚集关系,即在进行某一个 OLAP 查询计算时是否需要涉及到另一个 OLAP 查询的计算。

定义3 OLAP 查询之间的语义分解关系 \Rightarrow 。对于一个 OLAP 查询 Q 与一个 OLAP 查询集 $QSet = \{q_1, q_2, \dots, q_n\}$, 如果 OLAP 查询集中的所有查询能经过聚集计算生成查询 Q 的查询结果,并且 $\forall q_i, q_j \in QSet, q_i < q_j$ 都不成立,则称 OLAP 查询集为此 OLAP 查询的一个分解,记为 $Q \Rightarrow QSet$ 。

OLAP 查询之间的 \Rightarrow 所定义的是 OLAP 查询与 OLAP 查询集之间的查询计算关系,在执行 OLAP 查询 Q 时,可以直接对这个查询进行执行,也可以将其分解后,执行与其具有分解关系的查询集中的所有查询,再通过聚集计算得到结果。因此如果有 $Q \Rightarrow QSet$, 则在执行查询时,执行查询 Q 与执行 $QSet$ 得到的查询结果是等价的。

定理1 分解关系 \Rightarrow 具有传递性。

证明 设对于 OLAP 查询 Q 与 OLAP 查询集 $QSet$ 存在 \Rightarrow 关系,即 $Q \Rightarrow QSet$, 对于 $QSet = \{q_1, q_2, \dots, q_n\}$ 中的每个查询,有 $q_i \Rightarrow \{q_{i1}, q_{i2}, \dots, q_{in}\}$, 记 $\{q_{i1}, q_{i2}, \dots, q_{in}\}$ 为 QS_i , 记 $QSet' = \bigcup_{i=1}^n QS_i$, 则根据 \Rightarrow 的定义,有 $QSet \Rightarrow QSet'$, 因此 $\forall q_i, q_j \in QSet', q_i < q_j$ 都不成立,同时由于 $QSet'$ 中的查询都可以经过聚集计算得到 $QSet$ 的查询结果,进而再经过一步聚集计算得到 Q 的查询结果,从而得到 $Q \Rightarrow QSet'$ 。命题得证。

定理2 如果有 $Q \Rightarrow QSet$, 则 $\forall q_i (q_i \in QSet \wedge q_i < Q)$ 成立。

证明 由 \Rightarrow 的定义可知,如果有 $q_i \in QSet$, 则表明 q_i 的查询结果可以作为查询 Q 结果的一部分,并且 q_i 可以向上进行聚集,得到查询 Q , 从而 $q_i < Q$ 成立。命题得证。

定义4 OLAP 查询之间的补集关系。对于一个 OLAP 查询 Q 与一个 OLAP 查询集 $QSet = \{q_1, q_2, \dots, q_n\}$, 则 OLAP 查询集 Q 基于 $QSet$ 的补集为一个 OLAP 查询的集合,记为 $\overline{Q_{QSet}}$, 定义为:存在一个 $QSet$ 的子集 $QSet'$, 使得 $Q \Rightarrow QSet' \cup \overline{Q_{QSet}}$ 成立,并且 $\overline{Q_{QSet}} \cap Q_{set} = \emptyset$ 。如果不存在其他任何补集 $\overline{Q_{QSet}'}$, 使得 $\overline{Q_{QSet}'} \Rightarrow \overline{Q_{QSet}}$, 则称 $\overline{Q_{QSet}'}$ 为最小补集。

在上述定义中,如果 $QSet$ 为已经获得查询结果的查询集合, Q 为正在进行的查询,则在执行调度时,可以比较执行查询 Q 与执行查询 $\overline{Q_{QSet}}$ 的预估代价,选择代价较小的执行,可以提高系统的整体效率, $\overline{Q_{QSet}} \cap Q_{set} = \emptyset$ 确保所得到的补集中不包含已完成的查询,即不重复执行已经完成的 OLAP 查询。值得注意的是,OLAP 查询集 Q 基于 $QSet$ 的最小补集可能有多个。例如,假设 Q 为查询:2007 年全年的销售额,而 $QSet$ 为 {2007 年 1 季度的销售额,2007 年 2 季度的销售额,2007 年 4 季度的销售额,2007 年全年华中地区的销售额,2007 年全年华东地区的销售额,2007 年全年其他地区的销售额}, 则 {2007 年 3 季度的销售额}、{2007 年华北地区的销售额} 均为查询 Q 基于 $QSet$ 的最小补集(假设地区维只包含华中,华东,华北,其他四个值)。

定理3 对于一个 OLAP 查询 Q 与一个 OLAP 查询集 $QSet = \{q_1, q_2, \dots, q_n\}$, 如果 $\overline{Q_{QSet}}$ 与 $\overline{Q_{QSet}}$ 均为查询 Q 基于 $QSet$ 的最小补集,则有 $\overline{Q_{QSet}} \cap \overline{Q_{set}} = \emptyset$ 。

证明 由最小补集的定义可知,如果 $\overline{Q_{QSet}}$ 为查询 Q 基于查询集 $QSet$ 的最小补集,则 $\overline{Q_{QSet}}$ 为 $QSet$ 中某些查询在一个维度上的分解,并且在一个维度上只有一个这样的分解。由于各

个维度各不相同,因此分解的结果也不同,从而命题得证。

2 基于语义分解的并行 OLAP 查询优化算法

由上面的定义可知,OLAP 查询之间的 > 关系及语义分解关系对于 OLAP 的并行查询有着很重要的作用,根据 > 关系可以确定 OLAP 查询之间的语义聚集关系,而语义分解关系与补集则用于分析未完成查询及已完成查询的关系,尽可能地考虑重用已完成的 OLAP 查询的可能性,因此将有助于系统的整体优化。同时由定理3可知,一个 OLAP 查询集 Q 基于 $QSet$ 的最小补集可能有多个,并且最小补集之间不存在交集,因此可以特别适用于并行的情况。

并行 OLAP 查询优化算法的主要思路是:首先执行那些最“基本”的 OLAP 查询(根据 > 关系来计算),然后对于未完成的 OLAP 查询,则判断是否可以利用已完成的查询来减少执行代价(根据语义分解关系 \Rightarrow 进行计算与判断),如果某个 OLAP 查询有多个基于已完成 OLAP 查询的补集,则可以进行并行计算,然后执行代价最小的那个补集。算法的具体过程如下(其中带有 \rightarrow 标志的代码表示可以并行执行的程序段):

算法1 基于语义分解的并行 OLAP 查询优化算法。

输入:OLAP 查询集 $QSET$

- 1) 定义已完成查询队列 $done_queue$, 初始化为空;
- 2) 定义待处理查询队列 do_queue , 初始化为 $QSET$;
- 3) 定义正处理查询队列 $doing_queue$, 初始化为空;
- 4) 对于 do_queue 中的查询
- 5) 计算查询之间的 > 关系,并以列表形式保存;
- 6) 如果查询 q 不出现在任一 > 关系的右边,则将其移入至 $doing_queue$ 中;
/* 由 > 关系的定义可知,对于任意一个非空的 OLAP 查询集,至少存在一个这样的查询 $q *$ /
- 7) While (do_queue 非空)
- 8) \rightarrow {从 $doing_queue$ 中选取若干个查询,分配到空闲处理机上执行,执行完后的查询,加入到 do_queue 队列中;}
- 9) \rightarrow {
- 10) 从 do_queue 选取查询,如果其基于 $done_queue$ 的最小补集为空,则移至完成队列;
- 11) 如果此查询有多个基于 $done_queue$ 的最小补集,则计算每个最小补集中查询预估代价的总和,根据目前空闲处理机的个数及预先定义的阈值,确定所能处理的最小补集个数 n , 并按总和从小到大,选择不多于 n 个最小补集加入到 $doing_queue$ 中;
- 12) }
- 13) \rightarrow {对于 $doing_queue$ 中的查询,计算其基于 $done_queue$ 的补集,如果补集为空,则移至完成队列}
- 14) End while

由算法可以看出,算法的优化过程体现在:

①在算法的第4),5)行,已经根据>关系对 OLAP 查询集中的查询进行预处理,从而避免重复查询;

②在算法的第8)行,对于相互之间没有>关系的查询,即这些查询之间没有聚集关系,也没有分解关系,因而可以分配到空闲的处理机上独立执行;

③在算法的第9)~11)行,对于未完成的 OLAP 查询进行调度时,不仅是考虑已完成 OLAP 查询对于未完成 OLAP 查询的直接导出关系,而且也根据语义分解关系 \Rightarrow 来考虑已完成 OLAP 查询对于未完成 OLAP 查询的间接导出关系,进一步地利用了已完成的查询;

④在算法的13)行,尽可能地利用空闲处理机来处理最

小补集,从而提高了系统的总体效率。

可见,算法1利用OLAP查询之间的 $>$ 关系与 \Rightarrow 关系,充分利用已完成的查询来得到整个查询集的结果,并且中间计算最小补集与查询都可以并行在不同的处理机上运行,达到并行的效果。

3 实验结果与性能分析

为了测试优化算法的有效性,对算法进行了测试。测试所采用的数据集为TPC-R^[4],实验所采用的是若干台Intel Pentium IV 2.6 GHz, 512 MB内存,运行Windows 2003 Server的PC机所搭建的并行处理网络。

实验1 随机选取了若干个OLAP查询,并比较随机调度与优化算法之间的总执行时间,其结果如图1所示。由图可以看出,当OLAP查询数量较少时,查询之间的 $>$ 关系与语义分解关系并不多,因而查询的执行顺序对于总的执行时间影响不大,优化算法由于要进行OLAP查询的预处理(计算OLAP查询之间的 $>$ 关系与语义分解关系)需要一定的CPU时间,因此可能优化后的总执行时间还比未优化时的查询执行时间要长;但在OLAP查询数量增加时,优化算法能识别出各个查询之间的公共查询并先执行得到结果,后续查询可以利用已有的结果得到,因此优化后的执行时间就少于未优化时的查询执行时间,并且这种效果随着OLAP查询数量的增多而愈加显著。

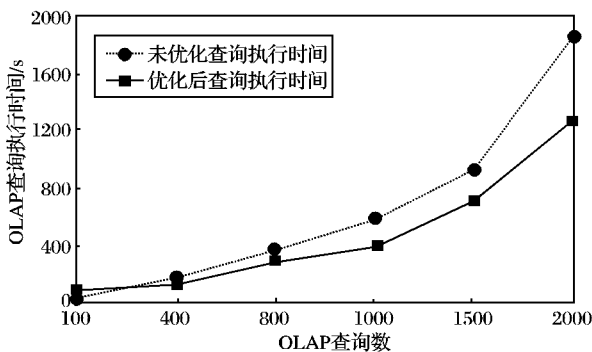


图1 基于语义分解的并行调度与随机调度执行时间比较

实验2 考查对于特定数量的OLAP查询,随机选取了1000个OLAP查询,考查处理机数目及补集阈值对于优化效果的影响,同时与未经优化的查询时间相比较,其结果如图2所示。由图可以看出,由于优化算法采用并行的方式来执行,因此处理机数目越多,优化效果越明显。同时,补集阈值的选取对于优化效果也有所影响,补集阈值越大,则能够选取更多的最小补集来确定语义分解关系与聚集关系,但同时最小补

集的计算时间也相应增加;补集阈值越小,最小补集的计算时间越少,但也越难于找到最优的最小补集,因此往往采用实验来确定一个经验值。就本实验中选取的1000个OLAP查询而言,如果处理机的数据为 n ,则补集阈值取 $n/4$ 到 $n/3$ 之间的整数,效果较好。

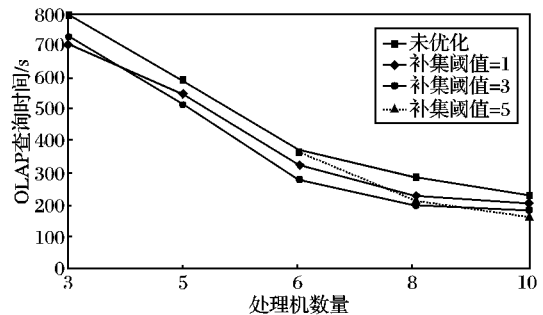


图2 用于并行的处理机数目与补集阈值对于查询优化的影响

4 结语

本文对OLAP查询进行形式化描述,定义了OLAP查询之间的聚集关系与分解关系,并采用并行计算OLAP查询补集的方式对OLAP查询进行整体优化,从而减少查询耗费的代价,改善了系统的性能。以后的研究方向将继续对模糊OLAP查询的匹配度进行研究,并对查询优化算法进行改进,以进一步提高模糊OLAP查询的效率。

参考文献:

- [1] SHIM J, SCHEUERMANN P, VINGRALEK R. Dynamic caching of query results for decision support system[C]// Proceedings of the 11th International Conference on Scientific and Statistical Database Management. Washington, DC: IEEE Computer Society, 1999: 254-263.
- [2] YANG J, KARLAPLEM K, LI Q. Algorithms for materialized view design in data warehousing environment[C]// Proceedings of the 23rd International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers, 1997: 136-145.
- [3] 薛永生, 黄震华, 段江娇, 等. 一种并行处理多维连接和聚集操作的有效方法[J]. 计算机研究与发展, 2004, 41(10): 1661-1669.
- [4] 杨科华. 提高联机分析处理(OLAP)性能若干关键技术的研究[D]. 南京: 东南大学, 2006.
- [5] Transaction Processing Performance Council TPC. TPC benchmarks H and R (decision support) [EB/OL]. [2009-08-06]. <http://www.tpc.org/>.

(上接第1955页)

地解决绝大多数非空间坐标地址的匹配问题。但是由于我国的门牌号分布并非完全有规律,使用地理编码技术定位出的点位的几何精度不高,因此它只适合使用在对空间数据精度要求较低的领域中。

参考文献:

- [1] 张铁燕, 翁敬农, 黄坚. 城市地理编码方法的探索与实践[C]// 中国地理信息系统协会第九届年会论文集. 杭州: 中国地理信息系统协会, 2005: 731-736.
- [2] 李军, 李琦, 毛东军, 等. 北京市地理编码数据库的研究[J]. 计算机工程与应用, 2004, 40(2): 1-3.
- [3] ZHANG XUEHU, MA HAOMING, LI QI. An address geocoding solution for Chinese cities[C]// Proceedings of Geoinformatics. [s.

l.]: International Society for Optical Engineering, 2006: 1-9.

- [4] 陈细谦, 迟忠先, 金妮. 城市地理编码系统应用与研究[J]. 计算机工程, 2004, 30(23): 50-52.
- [5] 朱建伟, 王泽民. 地理编码原理及其本地化解决方案[J]. 北京测绘, 2004(2): 24-27.
- [6] 孙亚夫, 陈文斌. 基于分词的地址匹配技术[C]// 中国地理信息系统协会第四次会员代表大会暨第十一届年会论文汇编. 北京: 中国地理信息系统协会, 2007: 114-125.
- [7] 孙茂松, 左正平, 黄昌宁. 汉语自动分词词典机制的实验研究[J]. 中文信息学报, 2000, 14(1): 1-6.
- [8] 郭会, 宋关福, 马柳青, 等. 地理编码系统设计与实现[J]. 计算机工程, 2009, 35(1): 295-299.