维普资讯 http://www.cqvip.com

考虑更新频率等因素的物化视图有效更新算法

王新军　洪晓光　王海洋
（山东大学计算机科学与技术学院　济南　250100）

An Efficient Refreshment Algorithm for Materialized View Considering the Update Frequencies

WANG Xin-Jun　HONG Xiao-Guang　WANG Hai-Yang
(School of Computer Science and Technology, Shandong University, Jinan 250100)

Abstract　For the materialized views in the fast LAN or computing grid environment, it is a very important problem that how to refresh them efficiently when data sources have changed. In this paper, we take the update frequencies and the size of source relations into account and present a partition strategy and an efficient algorithm by creating auxiliary views. Our algorithm may decrease the cost of join operation and communication on network as low as possible.

Keywords　Update frequency, Materialized view, Refreshment algorithm, Data warehouse

1. 引言

物化视图的维护是数据仓库研究的一个重要问题，特别是对定义在多个远端数据源上的物化视图，如何保证当数据源发生变化时物化视图的内容与数据源的内容达到某种程度的一致。有关这方面的研究很多，有的是将所有的源数据都复制到数据仓库中进行本地维护，但显然这样做太费存储空间。通常是根据数据源的变化对存储在仓库中的物化视图进行增量维护，这时就需要保证与数据源达到某种程度的一致。大多数商用数据仓库系统都是将对数据源的更新作为更新事务按队列方式组织并周期性地传到仓库进行批处理。因为用户对数据仓库的查询都是发生在白天，所以为了保证数据一致性商用系统大都在夜间对仓库进行更新，但现在刷新周期越来越短，因此如何有效地实现物化视图的增量计算成为重点要解决的问题。

文[6]提出了一种对没有物化、完全物化和部分物化的视图的更新时间的分析模型和初步的实验结果；文[8]讨论了借助辅助空间提高更新效率的方法，它对视图中的每个元组保存两个副本、每个更新都带有时间戳；文[7]也提出了一个基于时间戳的算法，但只能进行周期性的更新而不是实时增量更新。文[4]提出了一种利用数据仓库中的额外空间存储辅助视图从而节省更新时间的算法。

我们在文[4]提出的算法基础上作了进一步改进，提出了一种考虑各个数据源更新频率的、更加高效的物化视图更新算法。本文描述了相关的基本概念和前人所做的工作；介绍了我们改进后的、更加高效的物化视图更新算法；最后是总结。

2. 基本概念和相关工作

2.1　数据仓库模型

文[9]中给出的数据仓库的架构如图1所示，该系统有 n 个数据源端(site_1，…，site_n)，另一端则存储和保持数据仓库的物化视图。假定每个数据源与数据仓库之间的通信是可靠的并且是FIFO，即它们之间的数据不会丢失并且是按发送的次序进行传递。理论上说每个数据源都可以存储任意数量的基本关系，但为简单起见，假设 site_i 仅有单一关系 R_i($1\leqslant i\leqslant n$)。各数据源都是关系数据模型且相互独立、完全自治，各源关系的更新自动执行并且结果作为独立数据单位被传送到数据仓库。

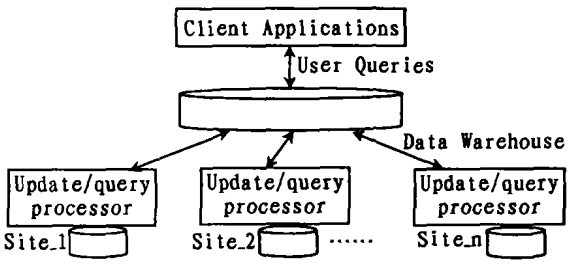

图1　一个数据仓库的架构

2.2　视图定义模型

在数据仓库系统中，为提高 OLAP 查询速度，通常在基本关系之上存储若干物化视图。本文中假定视图是由 SPJ 表达式定义的，即由来自 n 个数据源的关系(也可以是物化了的视图)$\{R_1,\cdots,R_n\}$定义的视图 V 为：

$$\Pi_X\sigma_P(R_1\infty\cdots\infty R_i\infty\cdots\infty R_n)$$

其中 X 为进行投影运算的属性列表，P 为进行选择运算的条件。又因为在由 SPJ 表达式定义的视图维护中，连接运算的费用要远远大于投影和选择运算，所以我们不妨假定视图仅由 n 个关系的连接构成，即 $V=R_1\infty\cdots\infty R_i\infty\cdots\infty R_n$。

对于连接运算可用参与运算的关系的元组数表示运算的费用，如上述计算 V 的表达式的费用可表示为：$cost(V)=|R_1|+|R_2|+\cdots+|R_n|$，其中$|R_i|$为关系 R_i 的元组数。

如果数据仓库中的物化视图是定义在一个广域网的多个远端数据源之上的，则对于数据源的更新要以消息的方式传送给数据仓库(因而通信的开销应包含其中)。文[1]中说明了因某个源关系发生变化而引起视图更新的情况。如对于定义在三个不同数据源 site_1、site_2、site_3上的关系 R_1、R_2和 R_3

王新军　副教授，博士生，研究方向为数据库、工作流管理系统，软件工程。

维普资讯 http://www.cqvip.com

上的视图 $V=R_1\infty R_2\infty R_3$，若发生了一个对 R_1 的更新 ΔR_1，则对 V 相应更新应为：

$$(R_1\cup\Delta R_1)\infty R_2\infty R_3=(R_1\infty R_2\infty R_3)\cup(\Delta R_1\infty R_2\infty R_3))=(R_1\infty R_2\infty R_3)\cup((\Delta R_1\infty R_2)\infty(\Delta R_1\infty R_3))$$

为此数据仓库应向 site_2和 site_3分别发送请求 $Q_1=\Delta R_1\infty R_2$ 和 $Q_2=\Delta R_1\infty R_3$，数据源端的机器经过计算后将结果 A1、A2返回数据仓库，仓库计算 $\Delta V=A1\infty A2$ 后对 V 进行更新。后面将会指出，为保证数据一致性，当进行上述计算时应对源关系 R_2 和 R_3 加锁。

2.3 一致性概念

我们用数据仓库状态 ws 表示仓库中当前的内容，ws_0，ws_1，…，ws_f 表示了因视图一系列的变更而引起的数据仓库状态改变的一个序列。对形如 $\Pi_X\sigma_P(R_1\infty\cdots\infty R_i\infty\cdots\infty R_n)$ 的表达式定义的视图 V，令 $V(ws_j)$ 表示 V 在状态 ws_j 的内容。令 n 元矢量 ss_j 表示某一时刻各个数据源的状态，其第 i 个分量 $ss_j[i]$ 为第 i 个数据源的内容，对数据元的更新也以串行方式执行，最终状态为 ss_q。$V(ss_j)$ 为基于数据源状态 ss_j 计算出来的 V 的内容。

假定初始有 $V(ws_0)=V(ss_0)$，文[2]定义了数据仓库与数据源之间的四级一致性如下：

(1) 收敛性：经过有限次更新操作，最终 $V(ws_f)=V(ss_q)$。

(2) 弱一致性：满足收敛性，且对所有的 ws_j 均存在数据源的某个状态 ss_i 满足 $V(ws_i)=V(ss_i)$，即视图的每个状态都对应数据源的某个状态。

(3) 强一致性：满足收敛性，且对任意 $ws_i<ws_j$，有 $ss_k\leqslant ss_l$ 满足 $V(ws_i)=V(ss_k)$ 和 $V(ws_j)=V(ss_l)$，即视图的每个状态都按某种次序对应数据源的某个状态。

(4) 完全一致性：满足强一致性，且对每个数据源状态 ss_i 都存在一个 ws_j 满足 $V(ws_j)=V(ss_i)$，即在视图的状态和数据源的状态之间存在一个完全的映射序列。

不难看出，完全一致性是一个很好的性质，它可以使视图与数据源在每一个状态都保持一致。但在实际中这种要求过于苛刻，有时收敛性就足够了，即便是它有某些中间状态是不合法的。在多数情况下能够保证强一致性的批处理更新方式是比较可行的。对于定义在多个远端数据源上的物化视图，如果各数据源的更新是相互独立并可能是同时进行的，就易出现视图与数据源不一致的情况。如在上例中，若数据仓库向 site_2发送了请求 Q_1 并期望得到结果 $A1=\Delta R_1\infty R_2$，但 site_2同时也接到一个更新操作 ΔR_2，则 Q_1 的计算结果就变成了 $A1'=\Delta R_1\infty(R_2\cup\Delta R_2)=(\Delta R_1\infty R_2)\cup(\Delta R_1\infty\Delta R_2)$，即 $V(ws_1)=A1'$，显然与更新 ΔR_1 预期的结果相比多出了 $\Delta R_1\infty\Delta R_2$（称为脏数据），从而破坏了一致性。

2.4 相关的工作

为解决一致性问题，文[2]提出了 ECA 算法，但该算法只适用于单一远端数据源的情况。对于多数据源的情况[3]，对视图和源关系加以限制，提出了一种 strobe 算法。这两种算法均是采用补偿的方法。为了能够保证完全一致性，文[3]又进而给出了一个 C-Strobe 算法，在处理下一个更新之前对前一个更新进行完整处理。文[1]给出了一种能够保证完全一致性的 SWEEP 算法，它不再向数据源发送补偿查询而是在本地进行处理。与 C-Strobe 算法相比，SWEEP 算法对每一个更新的响应时间为 O(n)，向数据源发送的消息数也为 O(n)，而且不需要在 V 中保存所有源关系的关键字。文[1]中指出 SWEEP 算法所需的更新时间为 $T_{refresh}=2(n-1)*t$，其中 n 是数据源的个数，t 是数据源与数据仓库之间的平均通信时间，与广域网上的通信时间相比在数据仓库或数据源上进行计算的时间可以忽略不计，即可以认为由于某个源关系的更新而引发的对视图的更新时间就是通过广域网向其余 n－1 个远端数据源发送请求和接收结果的通信时间和。

文[4]在数据仓库中开辟额外空间建立辅助视图，同时考虑不同数据源关系的更新频率提出了一种提高更新效率的算法，该算法先将 n 个源关系按各自的更新频率作为权进行排序，然后等分成 $K=[n/p]$ 个互不相交的组，每组有 p 个源关系，在分组时尽量使得分在各组中的数据源关系的更新频率值的和相等。然后为每一组定义一个辅助视图 $V_k=\Pi_{X(k)}\sigma_{P(k)}(R_{P(k-1)+1}\infty R_{P(k-1)+2}\infty\cdots\infty R_{Pk})$，其中 $1\leqslant k\leqslant K$，X(k) 和 P(k) 分别由出现在 X 和 P 中的与 $R_{P(k-1)+1}$，…，R_{Pk} 有关的部分构成，则显然有 $V=\Pi_X\sigma_P(V_1\infty\cdots\infty V_K)$。假定所有这些辅助视图都被数据仓库物化存储且数据仓库能够判定某个更新或是查询结果属于哪个辅助视图，若有对源关系 R_i 的更新 ΔR_i 引起了其所属辅助视图 V_k 的变化，利用 SWEEP 算法计算得到 ΔV_k。下一步对 V 的更新 $\Delta V=\Pi_X\sigma_P(V_1\infty V_2\infty\cdots\infty V_{k-1}\infty\Delta V_k\infty\cdots\infty V_K)$ 可在数据仓库中进行，因为 $V_j(1\leqslant j\leqslant K)$ 均在本地保存。与文[1]的 SWEEP 算法相比，该算法所需的更新时间至少要减少[n/p]倍，因为在划分时分在各组中的源关系的更新频率之和相等（或近似相等），这就意味着更新频率高的源关系所在的组中的关系个数就少，该组对应的辅助视图更新所需的网上传送时间就少，所以加权后对整个视图 V 的更新费用更低。

3. 改进后的更新算法和分析模型

在文[4]提出的划分思想基础上，我们可以对算法和分析模型做出如下的改进：

·无论是文[1]提出的 SWEEP 算法还是文[4]提出的按更新频率等值划分算法，在实现时均是将发生变化的某个源关系，如 R_1 的更新 ΔR_{j_1} 所产生的请求 Q_{j_2}，…，Q_{j_p} 分别传给所有其他的或是同组内其他的远端数据源，与关系 R_{j_2}，…，R_{j_p}，各自进行连接运算 $\Delta R_{j_1}\infty R_{j_i}$ 后再将结果 A_{j_2}，…，A_{j_p} 回传到数据仓库最后连接得到 ΔV。由于 ΔR_{j_1} 在 A_{j_2}，…，A_{j_p} 中可能存在大量冗余属性，我们将各数据源端原有的计算除保留一个外其余都改成半连接（semi-join），即 $R_{j_i}\infty\Delta R_{j_1}$，从而可减小 A_{j_i} 的规模、节省网上传输的费用，特别是当关系 R_{j_1} 具有较多的属性时。

·文[4]提出的算法是按更新频率值进行的等值划分，这种划分的出发点是将请求 Q_{j_2}，…，Q_{j_p} 和结果 A_{j_2}，…，A_{j_p} 在广域网上传输的时间之和即 $2*(p-1)*t$ 近似地看作是该辅助物化视图更新的开销，而与此相比在数据仓库或数据源上进行计算的时间可以忽略不计。但是随着源关系规模的不断膨胀，进行连接运算的时间开销也越来越大，另一方面网络技术的发展对带宽的不断提升以及基于高速网络的、面向大区域的高性能、分布式并行计算的网格的出现使得数据仓库和远端数据源之间的通信费用不断降低，因此在数据源结点上进行的连接运算的费用就不能不考虑在总的开销之内。

如果用 f_{j_i} 表示源关系 R_{j_i} 的更新频率（$\sum_{i=1}^{n}f_i=1$），用 R_{j_i} 包含的元组数 $|R_{j_i}|$ 近似地表示 $R_{j_i}\infty\Delta R_{j_1}$ 的计算费用，$|R_{SUM_Pk}|$

维普资讯 http://www.cqvip.com

$=\sum_{i=P(k-1)+1}^{pk}|R_i|$表示一个划分 $R_{P(k-1)+1},\cdots,R_{Pk}$中所有关系元组数之和，则对其相应辅助视图 V_k 进行更新所需的连接运算费用为 $T_{Vk_join}=\sum_{i=P(k-1)+1}^{pk}(f_1*(|Rsum_pk|-|R_i|))$，可见当在总的费用中连接运算所占的比重较大时，应当在划分时在分配有较大 f_i 值的源关系 R_i 的组中尽量放进那些元组数较少的其他源关系，以求得较小的连接费用。改进后算法的形式化描述如下：

```
void Freq_Size_Partition(R[],n,K); //R[]存放源关系，n 为源关系个数
{ k=1; //划分为 K 组，k 为当前组号
  Fk=0; //第 k 组中关系的更新频率之和
  Gk=Φ; //存放第 k 组关系的集合
  M =⌊Σ_{i=1}^{n} fi/K⌋; //每一组中关系的更新频率和的期望值
  Sort_on_freq(R[],f[],n);//n 个源关系按更新频率从大到小排序，f[]记录排序编号
  Sort_on_size(R[],s[],n);//n 个源关系按元组数从大到小排序，s[]记录排序编号
  for (k=1;k<=K;k++)
  { for (i=1;i<=n,f[i]==0;i++) {}; //找到当前未被选中的更新频率值最大的关系
    Gk= Gk∪{R[f[i]]}; Fk:= Fk+f_f[i]; f[i]:=0;
    j=n;
    while (Fk<M && j>=1)
    { if (s[j]!=0) { Gk= Gk∪{R[s[j]]}; Fk:= Fk+f_s[j]; s[i]:=0;}
      j--;
    }
}
```

上述算法在保证划分后的各组具有相同的更新频率和的基础上，将包含元组数较少的关系分到更新频率值较高的关系所在的组中，以获得较低的连接运算费用。如果在更新某一辅助视图时给其他辅助视图加锁，则上述算法可以保证完全一致性。

·如果将源关系的关键属性都加进视图定义中，对视图进行增量更新的时间还会大大减少，其代价就是需要在数据仓库中占用更多的空间。那么如果源关系按照更新频率进行了分组，每一组都对应了一个辅助视图。对于那些包含着高更新频率源关系的“特殊组”所对应的辅助视图，我们在其所基于的各个源关系的关键属性上建立索引，可以提高辅助视图的更新效率。

总结　本文讨论了对于在高速局域网或计算网格环境下定义在多个数据源上的物化视图，利用数据仓库中的额外空间建立辅助视图，提出了根据各个源关系的规模（所含元组数）和不同的更新频率因素进行划分的策略和相应算法，最大可能地减小连接计算费用和网上传输费用的有效更新算法并进行了相关的分析。

参考文献

1 Agrawal D, Abbadi A E, Singh A, Yurek T. Efficient View Maintenance at Data

2 Y. Zhuge et al. View Maintenance in a Warehousing Environment. In: Proc. of ACM SIGMOD Conf. 1995. 316～327

3 Zhuge Y, et al. The Strobe Algorithm for Multi-source Warehouse Consistency. In:Proc. of Intl. Conf. On Parallel and Distributed Information Systems, Miami Beach,FL, USA 1996

4 Wang Hui, et al. Efficient Refreshment of Materialized Views with Multi Sources. In Proc. of CIKM'99, Kansas City, MO, USA,1999. 375～382

5 Bakgaard L, Roussopoulos N. Efficient Refreshment of Data Warehouse Views:[TR3642]. DCS,Univ. of Maryland at College Park,Dec. 1996

6 Hull R, Zhou G. Towards the Study of Performance Trade-offs Between Materialized and Virtual Integrated Views. In:Proc. of Workshop on Materialized Views: Techniques and Applications Montreal,1996. 91～102

7 Labrinidis A, Roussopoulos N. Reduction of Materialized View staleness Using Online Updates: [TR3878]. DCS, Univ. of Maryland at College Park,Feb. 1998

8 Quass D, Widom J. Online Warehouse View Maintenance. In: Proc. of ACM SIGMOD Conf. Tucson, Arizona,1997. 393～404

9 Hammer J,Garcia-Molina H, Widom J,Labio W, Zhuge Y. The Stanford Data Warehousing Project. IEEE Bulletin of the Technical Committee on Data Engineering,1995, 18(2):41～48

（上接第72页）

集基础上的推荐方法在准确率、覆盖率、综合测度以及所使用的规则数目上进行了比较。测试结果表明该方法比较适用于个性化推荐。

·为了进一步提高个性化推荐的准确率我们给出了几种个性化推荐中的页面加权技术。

在将来的研究中我们将把以上几种页面加权技术应用到基于实际数据的个性化推荐中，根据测试结果以便选取一种更好的方法。

参考文献

1 Cooler R,Mobasher B,Srivastav J. Web Mining: Information and Pattern Discovery on the World Wide Web. (ICTAI1997)

2 Srivastava J,Cooley R,Deshpande M,Tan P. Web Usage Mining: Discovery and Application of Usage Patterns from Web Data. ACM SIGKDD, Jan. 2000

3 Mobasher B,Dai H,Luo T,et al. Discovery of aggregate usage profiles for Web personalization. In:Proc. of the WebKDD 2000 Workshop at the ACM SIGKDD2000, boston,Aug. 2000

4 Mobasher B. Mining Web Usage Data for Automatic Site Personalization. To appear in Gaul, W.,& Ritter,G. (Eds.), Classification , Automation, and New Media, Springer. 2001

5 Mobasher B,Dai H,Luo T, et al. Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data. In:Proc. of The IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization(ITWP01)

6 Gaul W, Schmidt-Thieme L. Recommender Systems Based on Navigation Path Features. In: Proc. of the WEBKDD 2001 Workshop on Mining Web Log Data Across All Customer Touch Points. The Seventh ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining ,San Franscisco,CA,2001

7 Mobasher B. WebPersonalizer: A Server-Side Recommender System Based on Web Usage Mining:[Technical Report TR99-110]. Department of Computer Science, Depaul University

8 Mobasher B,Dai H,Luo T,et al. Effective Personalization Based on Association Rule Discover from Web Usage Data. In:Proc. of The 3rd ACM Workshop on Web Information and Data Management

9 Fu X, Budzik J, Hammond K. Mining navigation history for recommendation. In: Proc. 2000 int. conf. intelligent user interfaces,New Orleans,LA,ACM,Jan. 2000. 106～112

10 Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proc. Int. Conf. Very Large DataBase, 487-499, Santiago,Chile

11 Liu B,Hsu W,Ma Y. Mining Association Rules with Multiple Minimum Supports. KDD-99 San Diego CA USA,1999

12 Yang W,Sergio L,Alvarez A,et al. Efficient Adaptive-Support Association Rule Mining for Recommender Systems. Data Mining and Knowledge Discovery,2002,6:83～105

13 Cooler R,Mobasher B,Srivastava J. Data Preparation for Mining World Web Browsing Patterns. Knowledge and Information System(1999)

14 Han J,Pei J,Yin Y. Mining Frequent Patterns without Candidate Generation. In: Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data(SIGMOD'00)

15 丁增喜，刘洋，江志纲，等. 基于关联规则发现和页面加权的个性化推荐技术的研究. 小型微型计算机系统 sup. 2002,23

16 丁增喜. 支持 E-Services 的关联规则挖掘方法的研究与实现:[东北大学毕业论文]. 2003