数据仓库中新型动态实视图选择调整算法

葛学彬,周丽娟,王林爽,石 倩
GE Xue-bin,ZHOU Li-juan,WANG Lin-shuang,SHI Qian

首都师范大学 信息工程学院,北京 100037
College of Information Engineering,Capital Normal University,Beijing 100037,China
E-mail:icon666@126.com

GE Xue-bin,ZHOU Li-Juan,WANG Lin-Shuang,et al.New dynamic materialized view selection adjustment algorithm in data warehouse.Computer Engineering and Applications,2010,46(8):120-122.

Abstract: Because static materialized views selection algorithm has many shortcomings,such as larger search space,higher time consumption and excluding query probability and distribution,and the changes in data sources can't be reflected in data warehouse immediately.In view of these,this paper implements dynamic adjustment for static materialized views selection algorithm according to CVLC and IGA,that is,CNUMV algorithm.The algorithm has been proved in reducing search space and time consumption by the experiment.Most of all,because the algorithm considers materialized views mutual relations in influencing view benefit.Consequently,the algorithm can be dynamicly adjusted online and obtains anticipative purpose.

Key words: materialized view;data warehouse;query probability

摘 要:现有的静态实视图选择算法存在搜索空间太大、时间复杂度高以及未考虑查询的概率和分布等诸多缺点,并且当源数据发生变化时,这种变化不能立刻反映到数据仓库,不适合在线运行。针对上述问题在候选视图生成算法和 IGA 算法的基础上,对算法进行了动态调整,从而得出了新型物化视图动态调整算法 CNUMV。经实验证明该算法降低了视图的搜索空间和时间复杂度,更重要的是该算法考虑到了各视图之间相互依赖关系对视图收益的影响,从而使算法能够动态地在线调整,并且用实验证明了 CNUMV 算法的优越性,达到了预期的目的。

关键词:实视图;数据仓库;查询概率
DOI:10.3778/j.issn.1002-8331.2010.08.034 **文章编号:**1002-8331(2010)08-0120-03 **文献标识码:**A **中图分类号:**TP311.13

1 引言

在数据仓库中,实视图是指将部分查询视图预先计算并加以物理存储,这样可以有效地加快数据仓库对查询的响应速度,它已成为提高系统多维分析性能的重要手段。

实视图选择属于 NP-hard 问题[1],最典型的实视图选择算法是 Greedy 算法,后来出现了 PBS 和 BPUS 算法,这些算法在实际效果比 Greedy 算法好一些,但其总体性能却同 Greedy 算法一致,均为不少于最优解的。虽然 BPUS 同 Greedy 算法相比,其实际效果可能会好一些,但其复杂度却没有得到任何改善,而且,由于效益新增了除法运算,其运行速度较 Greedy 算法慢,因而仍然缺乏实用性。在文献[2]中,作者提出了候选视图格生成算法,经过对多维数据格图进行剪枝,将那些对物化视图总体代价的减少没有影响的视图筛选掉,从而大幅度降低了算法的搜索空间,提高了算法的搜索效率和速度。文献[3]即是借助 CVLC 算法的策略对 BPUS 算法进行了改进从而得到 IGA 算法,IGA 算法对收益的定义综合考虑了视图查询和维护等多方面因素,但是 IGA 算法有一个最主要缺陷:它每一步选取的视图都作为将来要物化的视图,没有考虑每选择一个新的视图后,已选视图的效益值会出现衰减,并且数据仓库具有时变特性,随着用户的增加,系统中查询的分布情况亦可能发生变化,使得原有的物化视图集不再适应新的查询分布情况,从而导致查询响应性能的下降。所以需要根据查询概率对 IGA 算法得到的实视图集进行调整,用查询概率高的视图替代已选出的实视图集 V 中收益大大降低的视图,并且根据查询的分布情况,对不再适应新查询的物化视图集进行实时在线调整。现存的各种静态算法由于时间复杂度过高,不适合在线调整。从动态的角度提出了一些复杂度较低适合在线的算法[4]。

该文克服了静态算法时间复杂度过高以及视图缩减不明显的缺点,在借鉴文献[2]中候选视图格生成算法和文献[3]中 IGA 算法思想基础上,对算法进行了动态的调整,从而得出了新型物化视图动态调整算法 CNUMV。经实验证明是行之有效的,达到了预期目的。

2 相关代价模型定义

定义 1 最小代价视图。一个查询 q_i 的查询结果通常可以从多维数据格图的多个视图结点计算出来,其中计算代价最小

基金项目:北京市教委科技发展计划项目(the Technology and Development Project of Beijing Education Committee under Grant No.KM200810028016)。
作者简介:葛学彬(1983-),男,硕士研究生,主要研究方向:数据仓库、数据挖掘;周丽娟,博士,教授;王林爽,石倩,硕士研究生。
收稿日期:2008-09-25 **修回日期:**2008-11-24

的视图称为查询 q_i 的最小代价视图 v_i。

定义 2 子视图。对于视图 v,若 u 与 v 至少满足下列条件之一,则称 u 为 v 的子视图:

(1)$u \in v$,即 u 可由 v 进行投影或选择操作直接得到,则称 u 为 v 的子视图,v 的子视图的集合称为 v 的子视图集, 记为 $Child(v)$。(2)$u \leqslant v$,即 u 与 v 满足偏序关系且$|u|<|v|$。

定义 3 候选视图 CV(Candidate View)。如果一个视图 v_i 满足以下则称为候选视图 CV:如果 $p(v_i)=0$,至少存在两个不同的候选视图 u_1,u_2,使得 $u_1 \in Child(v_i)$,$u_2 \in Child(v_i)$,而且 $u_1 \notin Child(u_1)$,$u_2 \notin Child(u_1)$。

定义 4 视图查询试验空间。在当前的统计周期 $T(n)$内,发生的 n 个查询事件集合称为有效样本空间。$Q_{set}(n)$用来描述当前的查询趋势,预示未来可能发生的查询趋势,以更好地适应算法。

定义 5 视图查询试验集合。在 $Q_{set}(n)$内,查询事件相对应的最小代价视图集合为 $V_{set}(n)=\{v_1,v_2,\cdots,v_t\}$,称为有效样本集合。

定义 6 查询代价目标值 $e(V)$。$e_n(V)=\sum c*p_i(n)*|v_i|$,这里 $p_i(n)$是指在第 n 个统计周期 $T(n)$里视图 v_i 发生查询的频率(查询次数/统计周期),$|v_i|$是视图 v_i 的大小。通过数学期望的变化,对查询视图类型分布变化进行定量的估计,参数 c 根据实际情况指定。

定义 7 查询代价目标偏离值 $d(V)$。$d_n(V)=\sum c*(|v_i|-e_n(V))^2*p_i(n)$,这里 $p_i(n)$是指在第 n 个统计周期 $T(n)$里视图 v_i 发生查询的频率(查询次数/统计周期),参数 c 根据实际情况指定,$|v_i|$是视图 v_i 的大小。

因为仅仅利用 $e(V)$并不能完全判断视图的分布情况,还必须通过计算方差来判断查询代价同 $e(V)$的偏离程度。

定义 8 视图查询极限条件 $CNU(V)$。$CNU(V)=\sum P_i*(e(V)-d(V))*|v_i|$。考虑到,可能会出现这样的情况,即在两个相邻的样本集合内两视图$|v_i|=|v_j|$,在 $T(n-1)$周期中 $e_i(n-1)=e_1$。而其他视图的查询概率在两个周期中都不改变,但是其视图集合的查询分布概率发生了本质变化,这样就必须做出调整。

定义 9 视图物化的相对单位空间收益。设已选择的物化视图集为 V,v 不属于 V, 则相对于 V 物化 v 所带来的相对收益:$B(v,V)=((|Q^{-1}(v)|-|v|)*p(v)+\sum(|Q^{-1}(u)|-|v|)*p(u)-c*|v|)/|v|$,公式中 c 为权重,由设计者根据视图维护开销相对于查询计算开销的重要程度来指定。

3 CNUMV 算法的主要思想及算法描述

首先把最近统计周期中的有效样本空间 $Q_{set}(n)$作为输入,利用文献[2]中的候选视图生成算法思想,计算出查询目标值和查询目标偏离值,并且比较两个周期内的目标值以及目标偏离值,如果在这两个周期内两个值发生了较大变化则对算法进行调整,此外还要计算 $CNU(V)$并与给定的条件极限值 $limit$ 相比较,如果不满足,则进行动态调整。

CNUMV 算法分两步进行:第一部分大幅缩减视图的候选空间以得到 CV、查询概率集 $Q_{set}(n)$和多维数据格图 V_{set} 作为输入,计算出 $p(n)$进而计算出查询概率约束 $CNU(V)$看是否满足条件,不满足则进入算法循环,进行算法调整,直至满足条件。第二部分在第一部分得出的候选视图集的基础上,充分利用视图收益、查询条件极限值以及视图的查询概率,用查询概率高的视图替代已选出的物化视图集 V 中收益大大降低的视图。

第一部分

输入:候选视图集合 CV,存储空间 S,最近统计周期中的有效样本空间,视图的查询概率集和视图的查询概率,以及给定极限条件值 $limit$。

输出:新的候选视图集合 V_N。

```
Begin
Initial:V_N=Ø
if((e_n(n)/P_i==e_n-1(n-1))/P_i-1&&(CNU(V)<limit);
else
{
 P_i=P_i/limit;
 B(v,V)=∑(P_i+e_n(v))*d_n(v);
 Flag=True;
 CV=P_i*(CV-{V_b})/CNU(V);
 Do
 {
 v={v|max(B(v,V)),v∈CV};
 if(S/CNU(V)>limit)
        {
          V_N=V_N∪v;
          P_i=(P_i+CNU(V)/limit);
          S=S/limit-|v|;
          limit=limit*max(P_i);
        }
else
       Flag=False;
 }
} While(CV/CNU(V)! =P_i/limit &&Flag);
return V_N;
```

第二部分

输入:V_N,S。

输出:最终应该被物化的视图集合 V。

```
Initial:Flag=True;CV=V_N;V=Ø;
while(CV/CNU(V)! =P_i/limit&&(d_n(n)==d_n-1(n-1))&&Flag)
{
 if((p(v)+CNU(V))/|v|<(p(u)+CNU(u)/|u|)Flag=false;
 else
 {
       if (S/limit>(|v|+CNU(v)*P_i)
       {
       while(P_i*C(V∪{v})<C(V)*P_i-1)
          {
          B(v,CV-C(V∩w))*=(limit+P_i)
          u=u_i/CNU(U_i-1)//(u_i={u_i|min(limit*p(u)/|u|),u_i∈V});
          Flag1=true;
       }
      CV=(CV-{V_b})/CNU(V);
      }
     v=v_i/CNU(v_i-1) //(v_i={v_i|max(limit*p(v)/|v|),v_i∈CV});
     while(V∩w! =NULL)
     {
      limit=limit*max(P_i);
      w=min(V-V_delete)//Get the min view in V-V_delete
```

```
if((v∈Child(w)||w∈Child(v))&&(CNU(V)<limit)
{
    If(limit<=1&&C(V)>C(V-{w}))
        {
        B(v,V)=∑limit*(P_{i-1}+E_n(v))*D_n(v);
        V=V-V∩{w};
        S=S+limit*(|w|-|v|);
        }
    else
        {
        CNU(V)=limit*CNU(V-{w});
        V_delete=V_delete∪(V∩{w});
        if(V==V_delete)Flagl=False;
        B(v,V)=∑(P_{i-1}+e_{n-1}(v))*d_{n-1}(v);
        }
    CV=CV-limit({v}+{w});
else
  {
      V_i=V_i∪{w};
      limit=limit*max(P_i);
      V_delete=Φ;
    }
    Flag1=true;
    V=V∪V_i;
  }
}
}
```

4 实验验证分析

4.1 实验环境与实验设计

实验环境的硬件平台是 Dell 品牌机,CPU2.4 GHz,1.00 GB 内存。操作系统为 Windows2003 Server 数据库平台为 SQL Server 2000。在现有的各种实视图静态算法中,较具代表性的有 BPUS 和 Greedy,虽然这些算法在一定条件下执行效率还可以,但是它们需要满足众多的前提条件,从而使它们在实际数据仓库项目应用中受到很大的限制。从理论上讲,CNUMV 算法能较好地改进以上算法中存在的问题,为了验证 CNUMV 动态调整算法较 Greedy 和 BPUS 算法有较好的效果,进行了如下实验。

4.2 性能分析

在实验中,为了清楚地验证算法性能,设计数据仓库中维度从 3 个逐渐增加至 6 个,并且所有的维表都是依据一个事实表,在用户提交查询后的每个统计周期,分别调用 BPUS 算法、Greedy 算法和 CNUMV 算法进行比较。在维数由 3 逐渐增加到 6 的过程中,CNUMV、BPUS 和 Greedy 算法使用候选视图数量分别对应为 101,241,3 452,4 235,算法第一步处理后候选视图的平均数量为:79,107,173,278。可以看出视图的数量已经大幅减少。虽然视图数量大幅减少,但是如果将如此数量的视图全部物化需要的时间还是比较长。

因此,在第一步的基础上再调用 CNUMV 动态调整算法进行实视图的进一步选择。选择后候选视图集大幅减少,视图的数量减少至 14,30,51,64,此时的候选视图集可以作为最终的视图进行物化。图 1 分别从数据和图像的角度展现了各算法运行时间对比结果。

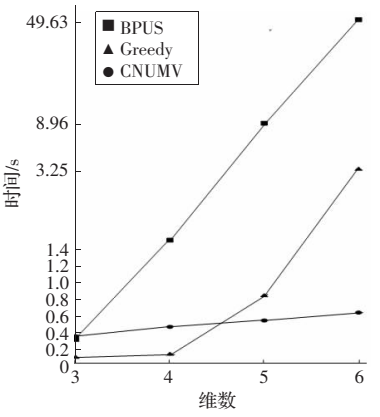

图 1 算法运行时间对比

从图 1 可以看到 CNUMV 算法与 BPUS 和 Greedy 算法相比,CNUMV 算法的运行时间明显比其他两个算法优越,尤其在维度增加的情况下,它的优越性更加明显。

CNUMV 算法包含有 BPUS 算法的思想,但比 BPUS 算法具有更好的效果。首先是两种算法面临的视图空间有很大的差异,其次 BPUS 并没有考虑到查询概率,默认为和查询分布及查询概率无关,但是实际情况并不是这样。CNUMV 经过调整后筛选掉了收益衰减较大的实视图,在此,用实验对三种算法得到的实视图集查询响应性能进行比较,采用平均响应时间来衡量查询的响应能力。响应时间对比如图 2。

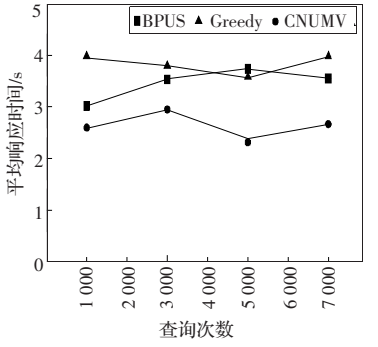

图 2 算法查询响应时间对比

从图 2 的结果可以看出,CNUMV 算法效果明显优于 BPUS 和 Greedy,从实验结果可以得出 CNUMV 算法无论从理论上还是从实际应用中都证明了是一种行之有效的实视图选择算法,达到了预期要求。

5 结论

对数据仓库中的实视图选择算法进行了研究,由于现存的静态选择存在很多缺点,这样就造成了选择出的视图集并不是最优的,针对以上方面提出的 CNUMV 算法能很好地解决以上问题,并且经实验证明达到了预期的效果,选出了最适合的实视图集。

参考文献:

[1] Harinarayan V,Rajaraman A,Ullman J D.Implementing data cubes efficiently[C]//Jagadish H V,Mumick I S.Proc of the 1996 ACM SIGMOD International Conference Management of Data.New York: ACM Press,1996:205-216.

(下转 143 页)