一种有效的关系数据库压缩方法*

骆吉洲[1+], 李建中[1,2]

[1](哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

[2](黑龙江大学 计算机科学与技术学院,黑龙江 哈尔滨 150080)

An Efficient Compression Method of Relational Database

LUO Ji-Zhou[1+], LI Jian-Zhong[1,2]

[1](School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[2](School of Computer Science and Technology, Heilongjiang University, Harbin 150080, China)

+ Corresponding author: Phn: +86-451-86415872, Fax: +86-451-86415827, E-mail: luojizhou@hit.edu.cn

Received 2003-11-14; Accepted 2004-06-10

Luo JZ, Li JZ. An efficient compression method of relational database. *Journal of Software*, 2005,16(2): 205−214. http://www.jos.org.cn/1000-9825/16/205.htm

Abstract: There usually are many attributes, called small-range attributes, with small number of different values in massive relations. The number of combination values of these attributes is also very few in massive relations so that there are a lot of repeated combination values of these attributes in massive relations. It is important to remove the repeated combination values to improve the efficiency of storing and querying massive relations. A compression method for removing the repeated combination values is proposed in this paper. To compress a massive relation, the method partitions the relation into two small relations: one consists of the small-range attributes and the other consists of the rest attributes. The key problem is to identify the small-range attributes. The NP-hardness of this problem is proved, and two approximate algorithms are proposed to solve this problem. The compression algorithms and the query processing based on the compressed method are also discussed. Experimental results show that the compression method has high compression ratio and enhances the query processing performance.

Key words: massive relation; compressed database; small-range attributes' combination; NP-complete problem

摘 要: 海量关系中经常存在小值域属性,关系不仅在这些属性上的互不相同的值的数量很小,而且在这些属性的组合上的值域也很小.因此,海量关系在这些属性上有很多重复的组合值.一种提高数据库的存储和查询效率的重要方法就是消除这些重复取值.为此,提出了拆分压缩技术,它将海量关系拆分成两种较小的关系,其中一种关系的属性由小值域属性组组成,而另一种关系的属性是海量关系的其他属性.该方法的关键是小值域属性

* Supported by the National Natural Science Foundation of China under Grant No.69873014 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2001AA444110 (国家高技术研究发展计划(863)); the National Grand Fundamental Research 973 Program of China under Grant No.G1999032704 (国家重点基础研究发展规划(973)); the Natural Science Foundation of Heilongjiang Province of China under Grant Nos.F0208, zjg03-05 (黑龙江省自然科学基金)

作者简介: 骆吉洲(1975−),男,重庆人,博士生,主要研究领域为压缩数据库技术;李建中(1951−),男,博士,教授,博士生导师,主要研究领域为数据库技术,数据仓库,半结构化数据,Web 信息集成,数据流,传感器网络,数据挖掘,压缩数据库技术.

组的识别问题.在证明了这个问题的 NP-完全性后,给出了两种在海量关系中识别小值域属性组合的算法,并在此基础上提出了海量关系拆分压缩技术,讨论了压缩关系的查询处理方法.实验结果表明,拆分压缩技术可以取得较好的压缩效果,并可以提高数据库查询处理的整体性能.

关键词: 海量关系;压缩数据库;小值域属性组;NP-完全问题

中图法分类号: TP311 **文献标识码**: A

1 问题提出

信息技术的发展使得各领域的信息量都爆炸性地增长,高于 10^{12} 字节的海量数据层出不穷.为了有效地管理海量数据,人们提出了压缩数据库技术[1−15].压缩数据库技术可以提高海量数据的存储效率,也是提高数据库性能的重要途径[3−5].目前,压缩数据库技术的研究主要包括适应于数据库随机存取特征的数据压缩方法、压缩数据上的操作算法、压缩数据库的查询优化技术等.本文研究数据库压缩算法.

经过大量调查研究,我们发现海量关系中常存在一些属性,关系在这些属性上的投影结果非常小.我们称这样的属性组为小值域属性组(small-range attributes' combination,简称 SRAC).例如,在码头的货物运输管理数据库中有一个记录接/发货物情况的关系,它记录运输合同号、货物名称编号、货物供应商编号、货物运送任务编号、货物数量、附加费、折扣、返回标识、货运状况、发货日期、到达日期、签收日期、押运负责人、备注等诸多属性的值.在这个关系中,由货物运送任务编号、发货日期、到达日期、押运负责人、运货方式、返回标识等属性构成了一个 SRAC.我们考察了 3 000 万条记录在这组属性上的投影,其结果低于 1 000 万条记录.再如,在移动通信公司的数据库中,通话详单这个关系包含了约 40 多个属性,其中业务服务类型、业务服务号码、用户类型、费用类型、归属地区号、到访地区号等等 10 多个属性也构成一个 SRAC.在大型商场的交易详单中,商品名称、生产厂家、零售价、销售员等属性也构成一个 SRAC.事实上,海量关系常包含几十个甚至上百个属性,其中往往有多个属性的值域较小并导致小值域属性组的出现.

显然,SRAC 的存在使得海量关系中产生大量数据冗余.若能将 SRAC 从海量关系中分离出来,并将原关系拆分成多个小关系,就可以消除由 SRAC 引起的数据冗余.同时,由于拆分后元组长度减小,每个物理存储块中可以容纳更多元组,则可以减小查询操作时的 I/O 次数,进而改善数据库的性能.如何消除由 SRAC 引起的数据冗余,实现对海量关系的压缩,是一个新的数据库压缩问题.本文旨在研究针对海量关系常存在 SRAC 这个特点的数据压缩方法.

本文首先研究了 SRAC 的识别问题,它是拆分压缩的关键,并证明了这个问题是 NP-完全的,然后给出了近似求解这个问题的贪心算法和遗传算法.进而,提出了海量关系拆分压缩方法.最后,讨论了基于海量关系拆分压缩方法的查询处理技术.

2 相关工作

压缩数据库技术可以提高存储效率并改善数据库的整体性能,故该技术已受到工业界和学术界的重视.事实上,Oracle 的开发者已将成熟的字典压缩技术整合到数据库物理层中,改善了数据库的存储效率和数据库的整体性能[6].另一些数据库压缩技术已广泛地被商用数据库采用,其中包括缩写、空值悬挂、游程编码、差值压缩等.迄今为止,已有的压缩数据库实验系统包括 IMS[7],AODB [8]和 NDB[9],这些系统均是将成熟的压缩技术应用到数据库系统中构建的原型系统.

目前,数据库压缩技术的研究主要集中在以下 4 个方面:(1) 数据库压缩方法的研究,包括使用已有的数据压缩算法压缩数据库和研究新的适应于数据库存取模式的数据压缩方法.被应用于数据库压缩的传统数据压缩方法包括字典压缩技术、游程压缩技术、算术压缩和 LZ 压缩技术[3],提出的适应于数据库存取模式的数据压缩方法包括索引压缩方法[10]、保序压缩技术[11]、基于语义的压缩技术[12]、面向块的增量压缩方法[13]、映射完全的数据压缩算法[9].(2) 压缩数据上的操作算法的研究,主要是在压缩数据上无须解压缩或解压代价极小

的数据操作算法[2,9,14].(3) 压缩数据库上的查询优化处理技术的研究.目前的研究结果很少,只有文献[4,15]研究了数据压缩技术对查询优化技术的影响,提出了适合于压缩数据库的查询优化策略以提高数据库的性能.(4) 压缩数据库的自动设计技术的研究.文献[15]提出了一种压缩数据库的自动设计方法.这种方法根据给定的数据库、数据库上常执行的查询及其被执行的概率,从可用的压缩方法集合中挑选一组方法来压缩给定的数据库,使得这些查询效率最高.

在这方面的所有研究中,压缩方法的研究是最活跃的,特别是针对海量数据及其操作的具体特点来研究数据库压缩算法.文献[12]根据海量关系中一些元组在某些属性上取值的相似性,提出了一种有损的语义压缩方法.作者将海量关系的元组分类并为每类选取一个代表元组.对每类的任意元组,均用代表元组表示,除非它与代表元组的误差超出了指定范围.文献[9]为了在压缩数据库中实现无须解压缩的连接操作,将关系拆分成二元关系,并用 MASK 方法实现了数据库的压缩存储.文献[10]根据高维数据的特点,提出了一种有效的索引压缩方法.文献[13]中针对统计数据库中元组的聚类性质提出了面向块的增量压缩方法.

本文根据海量关系中经常存在小值域属性组这个特征,提出了海量关系的拆分压缩方法.实验结果表明,拆分压缩技术可以取得较好的压缩效果,并可以提高数据库查询处理的整体性能.

3 小值域属性组识别问题的 NP-完全性

3.1 问题的定义

设 $R\langle X_1,X_2,\ldots,X_n\rangle$是一个 n 元关系,其中 X_i 标识 R 的第 i 个属性.如果$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}\subset\{X_1,X_2,\ldots,X_n\}$,则我们将 R 在$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$上的投影结果关系记为 $R\langle X_{i_1},X_{i_2},\ldots,X_{i_k}\rangle$.在下面的讨论中,我们用 m 和 m'分别表示 R 的元组数和 $R\langle X_{i_1},X_{i_2},\ldots,X_{i_k}\rangle$中的元组数.

定义 1. 设$\alpha\in(0,1)$,$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}\subset\{X_1,X_2,\ldots,X_n\}$,如果 $m'/m\le\alpha$,则称$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$是关系 R 的一个 α-小值域属性组.$1-\alpha$称为小值域属性组的冗余度.

设$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}\subset\{X_1,X_2,\ldots,X_n\}$,$\{Y_1,Y_2,\ldots,Y_{n-i_k}\}=\{X_1,X_2,\ldots,X_n\}-\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$,我们可以考虑将关系拆分成两个关系 $R\langle X_{i_1},X_{i_2},\ldots,X_{i_k}\rangle$和 $R\langle Y_1,Y_2,\ldots,Y_{n-i_k}\rangle$.这样,我们就消除了关系 R 中由小值域属性组$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$引起的数据冗余.读者可能会问,如何从 $R\langle X_{i_1},X_{i_2},\ldots,X_{i_k}\rangle$和 $R\langle Y_1,Y_2,\ldots,Y_{n-i_k}\rangle$恢复关系 R?我们将在第 5 节给出解决这个问题的方法.

为了便于讨论,假设 R 的所有属性都是定长的,属性 X_i 的宽度为 $w_i,i=1,2,\ldots,n$.这样,存储整个关系所需要的空间为 $m\sum_{i=1}^{n}w_i$,而存储 $R\langle X_{i_1},X_{i_2},\ldots,X_{i_k}\rangle$和 $R\langle Y_1,Y_2,\ldots,Y_{n-i_k}\rangle$的空间分别为 $m'\sum_{i=1}^{n}w_i$ 和 $m\sum_{i\in\{1,\ldots,n\}-\{i_1,\ldots,i_k\}}w_i$.这样,如果对 R 的存储改为对 $R\langle X_{i_1},X_{i_2},\ldots,X_{i_k}\rangle$和 $R\langle Y_1,Y_2,\ldots,Y_{n-i_k}\rangle$的存储,则节省的空间可以表示为

$$m\sum_{i=1}^{n}w_i-\left[m'\sum_{i=1}^{i_k}w_i+m\sum_{i\in\{1,\ldots,n\}-\{i_1,\ldots,i_k\}}w_i\right]=(m-m')\sum_{i=1}^{i_k}w_i \tag{1}$$

如果式(1)大于 0,则这种策略可以达到压缩效果.我们的问题是如何在关系 R 上找到某个小值域属性组,使得式(1)达到最大.于是,我们得到下述的小值域属性组识别问题:

输入:关系 $R\langle X_1,X_2,\ldots,X_n\rangle$,属性 X_i 的存储空间 $w_i,i=1,2,\ldots,n$;

输出:$\{X_1,X_2,\ldots,X_n\}$的子集 A,使得 $(m-m')\sum_{i=1}^{i_k}w_i$ 达到最大.

3.2 极大向量问题的NP-完全性

为证明海量关系中小值域属性组识别问题是 NP-完全问题,我们定义极大向量问题,并证明其 NP-完全性.

定义 2. 设$\langle v_1,v_2,\ldots,v_n\rangle\in\{0,1\}^n$,$\langle t_1,t_2,\ldots,t_n\rangle\in\{0,1\}^n$.若 $v_i\le t_i,i=1,2,\ldots,n$,则称向量$\langle v_1,v_2,\ldots,v_n\rangle$和$\langle t_1,t_2,\ldots,t_n\rangle$满足关系$\le$.

定义 3. 设 $W=\langle w_1,w_2,\ldots,w_n\rangle$是任意向量,其中 $w_i\in N^+$.$\forall V=\langle v_1,v_2,\ldots,v_n\rangle\in\{0,1\}^n$,定义 V 与 W 的$\cdot$运算为

$V\cdot W=\sum_{i=1}^{n}v_iw_i$.

注意到,关系≤是$\{0,1\}^n$上的偏序关系.为了简单计,我们将$\langle 1,1,\ldots,1\rangle\in\{0,1\}^n$记为 $\boldsymbol{e}$.

定义 4. 如下问题称为极大向量问题:给定 n 维向量空间中的权值向量 $W=\langle w_1,w_2,\ldots,w_n\rangle$,投影函数 $f(V)$: $\{0,1\}^n\to N^+$,且 f 是增函数(即若 $V_1\le V_2$,则 $f(V_1)\le f(V_2)$),找出 $V_0\in\{0,1\}^n$,使得$[f(\boldsymbol{e})-f(V_0)](V_0\cdot W)=\max_{V\in\{0,1\}^n}[f(\boldsymbol{e})-f(V)](V\cdot W)$.

引理 1. 设 $a>0,b>0$ 且 $a+b=c$(c 是常数),则 $ab\le c^2/4$ 且等号成立,当且仅当 $a=b=c/2$.

定理 1. 极大向量问题是一个 NP-完全问题.

证明:极大向量问题是一个 NP 问题,因为我们可以使用非确定的图灵机在多项式时间内枚举$\{0,1\}^n$的每个向量 V,并通过计算$[f(\boldsymbol{e})-f(V_0)](V_0\cdot W)$来判断 V 是否为问题的解.

下面将集合划分问题归约到极大向量问题.设(S,g)是集合划分问题的任意实例.取 $n=|S|$并建立 S 到集合$\{1,2,\ldots,n\}$的一一映射 map,称这个映射为 S 到集合$\{1,2,\ldots,n\}$上的索引函数,并将其逆函数记为 map^{-1}.

令权值向量$W=\langle g(map^{-1}(1)),g(map^{-1}(2)),\ldots,g(map^{-1}(n))\rangle$,投影函数$f(V)=V\cdot W$.这样,得到极大向量问题的一个实例$(n,W,f)$.显然,上述过程可在多项式时间内完成.注意,极大向量问题实例(n,W,f)中,$[f(\boldsymbol{e})-f(V)]+(V\cdot W)=f(\boldsymbol{e})=\sum_{x\in S}g(x)$(常数).

下面证明(S,g)有解当且仅当(S,w,f)有解.

(1) 设 $A\subseteq S$ 是(S,g)的一个解.于是 $\sum_{x\in A}g(x)=\sum_{x\in S-A}g(x)$.根据索引函数得到$\{1,2,\ldots,n\}$的一个子集$\{map(x)|x\in A\}$.进而得到向量 $V=\langle v_1,v_2,\ldots,v_n\rangle\in\{0,1\}^n$,其中 $v_i=1$,若 $i\in\{map(x)|x\in A\}$,否则 $v_i=0$.于是,

$$V\cdot W=f(V)=\sum_{i=1}^{n}v_iw_i=\sum_{x\in A}v_{map(x)}w_{map(x)}=\sum_{x\in A}w_{map(x)}=\sum_{x\in A}g(map^{-1}(map(x)))=\sum_{x\in A}g(x),$$

$$f(\boldsymbol{e})-f(V)=\sum_{i=1}^{n}w_i-\sum_{i=1}^{n}v_iw_i=\sum_{i=1}^{n}(e-v_i)w_i=\sum_{x\in S-A}w_{map(x)}=\sum_{x\in S-A}g(map^{-1}(map(x)))=\sum_{x\in S-A}g(x).$$

从而,$f(\boldsymbol{e})-f(V)=V\cdot W$.根据转换过程的说明和引理 1 可知,$V$ 就是极大向量问题实例(n,W,f)的一个解.

(2) 设极大向量问题实例(S,w,f)的解为 $V=\langle v_1,v_2,\ldots,v_n\rangle$.由 V 得到 S 的子集 $A=\{x\in S|v_{map(x)}=1\}$.类似于式(1)中的计算可以得到 $f(\boldsymbol{e})-f(V)=\sum_{x\in S-A}g(x)$ 和 $V\cdot W=\sum_{x\in A}g(x)$,判断 $f(\boldsymbol{e})-f(V)=V\cdot W$ 是否成立.我们断言:当该等式成立时,A是(S,g)的解;否则,(S,g)无解.前半部分结论显然,现用反证法证明后半部分结论.设(S,g)的一个解为 B,由式(1)的过程得到(n,w,f)的另一解 V_B,它使$[f(\boldsymbol{e})-f(V_B)](V_B\cdot W)$最大且 $f(\boldsymbol{e})-f(V_B)=V_B\cdot W$.由 $f(\boldsymbol{e})-f(V)\ne V\cdot W$ 和引理 1 可知,$[f(\boldsymbol{e})-f(V)](V\cdot W)<[f(\boldsymbol{e})-f(V_B)](V_B\cdot W)$,这与 V 是(S,w,f)的解矛盾.

显然,可在多项式时间内把极大向量问题实例(S,w,f)的解转换为集合划分问题实例(S,g)的解. □

3.3 小值域属性组识别问题的NP-完全性

给关系 R 的所有属性规定一个顺序(如 R 的第 i 个属性就是 X_i),则属性集合$\{X_1,X_2,\ldots,X_n\}$的任意子集 A 可表示为$\{0,1\}$上的一个 n 维向量$\langle x_1,x_2,\ldots,x_n\rangle\in\{0,1\}^n$,其中 $x_i=1$ 表示 X_i 在 A 中,$x_i=0$ 表示 X_i 不在 A 中.进而,R 在 A 上的投影结果 $R\langle A\rangle$可表示为 $R\langle x_1,x_2,\ldots,x_n\rangle$.在寻找关系 R 的小值域属性组时,确定子关系 $R\langle x_1,x_2,\ldots,x_n\rangle$的元组数必须扫描关系 R.下面将证明:即使关系扫描可以"有效完成",小值域属性组识别问题也是NP-完全的.所谓"有效完成"是指存在一个函数$f:\{0,1\}^n\to N^+$,使子关系 $R\langle x_1,x_2,\ldots,x_n\rangle$的元组数可以通过计算 f 在$(\langle x_1,x_2,\ldots,x_n\rangle)$的函数值来获得.若 A,B 是关系 R 的属性集的两个子集且 $A\subseteq B$(这意味着 A 的向量表示$\langle x_1,x_2,\ldots,x_n\rangle$和 B 的向量表示$\langle y_1,y_2,\ldots,y_n\rangle$满足$\{0,1\}^n$ 上的偏序关系≤),则子关系 $R\langle A\rangle$的元组数不超过子关系 $R\langle B\rangle$的元组数.于是,如果上面的函数 f 存在,则$f(\langle x_1,x_2,\ldots,x_n\rangle)\le f(\langle y_1,y_2,\ldots,y_n\rangle)$,即 f 是一个增函数.

基于上面的讨论,小值域属性组识别问题可以重新表述为:

输入:关系 R 的属性个数 n,各个属性的存储空间 w_i 构成的向量 $W=\langle w_1,w_2,\ldots,w_n\rangle$,计算子关系 $R\langle x_1,x_2,\ldots,x_n\rangle$

的元组数的增函数f:$\{0,1\}^n \to N^+$;

输出:$V_0 \in \{0,1\}^n$ 使得$[f(\boldsymbol{e})-f(V_0)](V_0 \cdot W)=\max_{V\in\{0,1\}^n}[f(\boldsymbol{e})-f(V)](V \cdot W)$.

重新表述的问题是极大向量问题,我们已经证明它是 NP-完全的.这说明求解小值域属性组识别问题时需要的扫描数据库的遍数不可能表达成属性个数 n 的多项式,除非 $NP=P$.于是,我们得到如下结论.

定理 2. 小值域属性组识别问题是一个 NP-完全问题.

4 小值域属性组的识别算法

从第 3 节我们看到,即使假定扫描数据中海量关系的过程可以"有效完成",小值域属性组的识别问题也是 NP 完全的.现在我们给出两种近似算法来求解这个问题.为了有效控制算法扫描海量关系的遍数,算法要求用户指定小值域属性组的冗余度下限 $1-\alpha$.事实上,如果用于拆分压缩的小值域属性组的冗余度过低,拆分压缩的效果将很差.因此,用于拆分压缩的小值域属性组的冗余度应该高于某个下界.值得注意的是,输出集合中属性的个数与α直接相关,α越大,输出的属性集合中属性的个数越多,但是压缩的效果不一定好.因此,怎样设置α的一个恰当的值,是一个值得关注的问题.下面我们分别给出这两种算法.

4.1 基于贪心策略的小值域属性组识别算法

算法的输入是数据库、各个属性的存储空间及冗余度的下限 $1-\alpha$(即α的上限),其输出为小值域属性组 A.算法开始时置 A 为空集,然后每次选择当前 A 的最优扩充属性 X 加入 A,直到 A 的冗余度超过设定的上限α.这里需要指出两点:(1) 作为α的上限是通过经验得出的;(2) 最优扩充是指把 X 加入 A 后所节省的空间最大的扩充.下面是这种算法的描述:

算法. Recon-greedy

输入:数据库 $R\langle X_1,X_2,\ldots,X_n\rangle$,各个属性的宽度 $W[i]$,上限α.

输出:一个小值域属性组 A,使得 $(m-m')\sum_{i=1}^{i_k} w_i$ 最大,其冗余度以 $1-\alpha$为下界.

1. $A\leftarrow\varnothing$,Freq$\leftarrow$0,$m\leftarrow R\langle X_1,X_2,\ldots,X_n\rangle$中的元组数;
2. WHILE (Freq<α) DO
3. $Temp\leftarrow$\$; $W[Temp]\leftarrow\infty$, $m_{Temp}\leftarrow m$;
4. FOR $\{X_1,X_2,\ldots,X_n\}\backslash A$ 中的每个属性 X DO /*确定最优扩充属性*/
5. 扫描数据库 R 确定子关系 $R\langle A\cup\{X\}\rangle$的元组数 m';
6. IF $m'/m<\alpha$ 且$(m-m')(w+W[X])<(m-m_{Temp})(w+W[Temp])$ THEN
7. $Temp\leftarrow X$, $m_{Temp}\leftarrow m'$;
8. $Freq\leftarrow m_{Temp}/m$; /*根据最终找到的属性做扩充*/
9. IF $(Freq<\alpha)$ THEN
10. $A\leftarrow A\cup\{Temp\}$, $w\leftarrow w+W[Temp]$;
11. 返回 A.

算法中第 2 步的 WHILE 循环至多为 n 次,且每次 WHILE 循环中第 4 步的 FOR 循环至多执行 n 次,每次 FOR 循环中都包含了一次数据库扫描.因此扫描数据库的次数最坏情况是 $O(n^2)$.虽然如此,对于海量关系,扫描$O(n^2)$次关系的时间仍是惊人的.对此,有如下两方面的考虑:(1) 压缩数据库通常是一次压缩多次使用,压缩数据库的时间复杂度常放到次要的地位来考虑.(2) 对海量数据关系,可先通过随机抽样获得适量的元组,再在样本上运行上面的算法,以节省扫描数据库需要的时间;然后还可以对在样本上得到的小值域属性组作进一步修改,得到更准确的小值域属性组.

4.2 基于遗传算法的小值域属性组识别算法

为了进一步减少识别小值域属性组的时间复杂性,本节给出一种遗传算法.小值域属性组识别问题是一个

组合优化问题,而且第 3 节中描述的属性集合的向量表示自然地实现了这个问题的解空间的编码.将编码后的解空间$\{0,1\}^n$视为种群空间,将任意属性集合的编码向量$\langle x_1,x_2,\ldots,x_n\rangle$视为种群空间中的个体染色体.于是,可以用遗传算法来求解.这里采用杰出遗传算法.初始时,把冗余度未超过冗余度下限的单属性的向量表示选入初始种群.种群中个体$\langle x_1,x_2,\ldots,x_n\rangle$的适应度就是利用该属性组进行拆分压缩所能够节省的空间的大小.然后,在种群中按照适应度大小随机选择母体进行杂交、变异产生新的种群,并在新种群中保留上一代种群中适应度最大的个体,重新计算各个个体的适应度.重复上述过程,直到下述条件之一成立:(1) 连续 N 代种群中适应度最大的个体不发生变化;(2) 种群中个体数低于某下限;(3) 总循环次数超过指定上界.最后所得适应度最好的属性组作为小值域属性组输出.

算法. Recon-Gen.

输入:数据库 $R\langle X_1,X_2,\ldots,X_n\rangle$,各个属性宽度 $W[i],\alpha,N$.

输出:一个小值域属性组 A,使得 $(m-m')\sum_{i=1}^{i_k} w_i$ 最大,其冗余度以 $1-\alpha$为下界.

1. 将冗余度不超过α的单属性的向量表示加入初始种群 $Q[]$,并计算初始种群中各个个体的适应度 $F[]$;
2. $fit\_count\leftarrow 0, loop\_count\leftarrow 0$;
3. WHILE (($loop_count\le$循环上界)且($fit\_count\le N$)且($Q[]$中的个体数≥下限)) DO
4. $k\leftarrow Q[]$中个体的总数,$f\leftarrow\sum_{i=1}^{k} F[i]$;
5. FOR i=1 TO k–1 DO
6. 随机从 $Q[]$中选取个体$\langle x_1,x_2,\ldots,x_n\rangle$和$\langle y_1,y_2,\ldots,y_n\rangle$; /* $Q[t]$被选中的概率为 $F[t]/f$ */
7. 将选中的两个个体在随机选定的位置杂交得到个体$\langle z_1,z_2,\ldots,z_n\rangle$;
8. 将$\langle z_1,z_2,\ldots,z_n\rangle$的每个基因位等概率(取 0.1)地变异;将新得到的个体插入下一代种群 $Q'[]$中,并计算其适应度;
9. 将 $Q[]$中适应度最大的个体插入 $Q'[]$中,并得到相应的适应度;
10. 删除 $Q'[]$中冗余度超过α的个体;
11. IF ($Q[]$中适应度最大的个体与 $Q'[]$中适应度最大的个体相同) THEN fit_count++;
ELSE $fit\_count\leftarrow 0$;
12. 删除 $Q[]$中所有个体并调换 $Q[]$和 $Q'[]$的角色;
13. 输出 $Q[]$中适应度最大的个体.

算法 Recon-gen 的时间复杂度取决于算法的 3 个终止条件中使用的参数.我们的实验结果表明,上述算法的运行时间低于基于贪心策略的算法,并能够获得更好的结果.

5 海量关系的拆分压缩方法

给定海量关系 R 和冗余度下限 $1-\alpha$,利用第 4 节中描述的算法,可以得到关系 R 的α小值域属性组 $A=\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$($R\langle X_{i_1},X_{i_2},\ldots,X_{i_k}\rangle$的元组数为 m').令$\{Y_1,Y_2,\ldots,Y_{n-i_k}\}=\{X_1,X_2,\ldots,X_n\}-\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$.关系 R 的拆分压缩十分简单,只需将 R 拆分成两个关系 R_1 和 R_2,使得 R_1 和 R_2 的属性集合分别是包含$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$和$\{Y_1,Y_2,\ldots,Y_{n-i_k}\}$的最小属性集合.在拆分过程中,我们需要考虑下面的两个问题:(1) 如何保证关系 R=Join(R_1,R_2)?Join 表示两个关系的连接操作;(2) 易知,关系 R 的 t(≥1)个元组对应关系 R_1 中的一个元组和关系 R_2 的 t 个元组.拆分压缩时如何反映这种对应关系?

为解决上述两个问题,引入单射$\phi:R\langle X_{i_1},X_{i_2},\ldots,X_{i_k}\rangle\rightarrow N^+$,并用它作为 R_1 和 R_2 之间的连接属性(如,可以具体定义为$\phi(x_{i_1},x_{i_2},\ldots,x_{i_k})=\sum_{j=1}^{k}\left(\prod_{n=0}^{j-1}|X_{i_n}|\right)x_{i_j}$,其中$|X_{i_j}|$表示属性 X_{i_j} 的值域大小,x_{i_j} 是数值化后的属性值.)于是,可以把 R 拆分为 R_1 和 R_2,其模式为 $R_1=R(X_{i_1},X_{i_2},\ldots,X_{i_k},\phi)$和 $R_2=R(\phi,Y_1,Y_2,\ldots,Y_{n-i_k})$,并将属性$\phi$定义为关系 R_1 的主键.

设新增属性ϕ的宽度为 v.结合第 3 节的讨论,我们知道存储整个关系所需要的空间为 $m\sum_{i=1}^{n}w_i$,而存储 R_1 和 R_2 的空间分别为 $m'\left(\sum_{i=1}^{i_k}w_i+v\right)$ 和 $m\left(\sum_{i\in\{1,\ldots,n\}\setminus\{i_1,\ldots,i_k\}}w_i+v\right)$.故,通过拆分压缩方法压缩掉的空间为

$$m\sum_{i=1}^{n}w_i-\left[m'\left(\sum_{i=1}^{i_k}w_i+v\right)+m\left(\sum_{i\in\{1,\ldots,n\}\setminus\{i_1,\ldots,i_k\}}w_i+v\right)\right]=(m-m')\sum_{i=1}^{i_k}w_i-(m+m')v \tag{2}$$

由于 $m'/m\leq\alpha$,压缩比的下限为 $(1-\alpha)\Sigma_{i=1,\ldots,i_k}w_i-(1+\alpha)v$.拆分压缩前,需要由此估算拆分压缩的压缩比.如果这个压缩比是合理的,则对关系 R 进行拆分压缩;否则,放弃拆分压缩.

拆分压缩时,对于关系 R 的每个元组,我们首先分别计算它在属性集合$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$和$\{Y_1,Y_2,\ldots,Y_{n-i_k}\}$上的投影$(x_{i_1},x_{i_2},\ldots,x_{i_k})$和$(y_{i_1},y_{i_2},\ldots,y_{n-i_k})$;然后计算 $N_{(x_1,x_2,\ldots,x_k)}=\phi(x_{i_1},x_{i_2},\ldots,x_{i_k})$;最后将$(N_{(x_1,x_2,\ldots,x_k)},y_{i_1},y_{i_2},\ldots,y_{n-i_k})$和$(x_{i_1},x_{i_2},\ldots,x_{i_k},N_{(x_1,x_2,\ldots,x_k)})$分别插入关系 R_1 和 R_2.由于ϕ是关系 R_1 的主键,当$(x_{i_1},x_{i_2},\ldots,x_{i_k},N_{(x_1,x_2,\ldots,x_k)})$在关系 R_1 中已经存在时,后一个插入操作自动失败.下面是拆分压缩算法.

算法. Splitting-Compression.

输入:关系 R,R 各属性的宽度,R 的一个α小值域属性组$\{x_{i_1},x_{i_2},\ldots,x_{i_k}\}$,$\varepsilon>0$.

输出:关系 R 的拆分 R_1 和 R_2.

1. 估算压缩比下限 *Ratio*;
2. IF *Ratio*<ε,返回;
3. 建立关系模式 $R_1=R(X_{i_1},X_{i_2},\ldots,X_{i_k},\phi)$和 $R_2=R(\phi,Y_1,Y_2,\ldots,Y_{n-i_k})$;
4. FOR R 的每个元组 T DO
5. 计算 T 在$\{X_{i_1},X_{i_2},\ldots,X_{i_k}\}$和$\{Y_1,Y_2,\ldots,Y_{n-i_k}\}$上的投影$(x_{i_1},x_{i_2},\ldots,x_{i_k})$和$(y_{i_1},y_{i_2},\ldots,y_{n-i_k})$;
6. 计算 $N_{(x_1,x_2,\ldots,x_k)}=\phi(x_{i_1},x_{i_2},\ldots,x_{i_k})$;
7. 将$(N_{(x_1,x_2,\ldots,x_k)},y_{i_1},y_{i_2},\ldots,y_{n-i_k})$插入关系 R_2,将$(x_{i_1},x_{i_2},\ldots,x_{i_k}\ N_{(x_1,x_2,\ldots,x_k)})$插入关系 R_1.

显然,算法 Splitting_Compression 仅需要扫描一遍关系 R.

定理 3. $R=Join(R_1,R_2)$,其中 $R_1=R(X_{i_1}\ X_{i_2},\ldots,X_{i_k},\phi)$,$R_2=R(\phi,Y_1,Y_2,\ldots,Y_{n-i_k})$.

证明:根据小值域属性组的定义和拆分压缩算法,直接验证 $R\subseteq Join(R_1,R_2)$和 $R\supseteq Join(R_1,R_2)$即可. □

6 基于压缩方法的查询处理

拆分压缩后,对原关系 R 的查询变成了对关系 R_1 和 R_2 的查询,这涉及到 Join 操作.完成此查询有两种方法:

方法 1. 直接在上述两个关系上进行连接操作.

方法 2. 按照下面的 3 个步骤进行:(1) 将原查询 q 分解为在关系 R_1 上的查询 q_1 和在关系 R_2 上的查询 q_2;(2) 分别执行查询 q_1 和 q_2;(3) 将由(2)得到的结果执行以ϕ为连接属性的自然连接,得到最终结果.

数据库的查询优化器对这两种方法的代价进行比较,选择代价较小的方法进行操作.另外,还可以在ϕ上建立索引以提高 Join 操作的效率.

我们将在另文中详细讨论拆分压缩后的关系上的查询优化与处理方法.

7 实验结果

我们实现了小值域属性组的提取算法和海量关系的压缩算法,下面是我们的实验结果.实验运行于 Linux6.2 平台上的 Oracle 数据库,服务器是配置为 PIV2.0GHz、内存 256Mb 的 PC 机.实验数据采用 TPC-H 生成的 4G 的数据量.压缩的关系是 TPC-H 中的 LINEITEM 关系,因为该关系的数据量占整个数据库数据量的约 80%,其中包含 30 007 486 条记录.实验主要包含以下 3 方面的内容.

第 1 组实验研究小值域属性组识别算法中不同的冗余度下限对算法输出结果的影响.实验中,多次在关系

LINEITEM 上运行算法 Recon-greedy,每次设置不同的冗余度下限.表 1 给出了实验结果.表中前 3 列分别给出了列名、各列值域的大小和各列的存储空间大小.表的后 3 列中为 1 的单元分别给出了冗余度下限分别为 0.3,0.5 和 0.7 时算法提取的 SRAC 中包含的属性.当 $1-\alpha=0.3$ 时,SRAC 上的不同取值的数量是 7 103 769.当 $1-\alpha=0.5$ 或 0.4 时,SRAC 上的不同取值的数量是 10 032 094.此外,当向这个 0.5SRAC 添加任意其他属性时,冗余度都在 0.2 以下.因此,适于 SRAC 识别算法的冗余度下限的合理的范围是[0.4,0.6].因此,在后面的实验中,始终设置冗余度的下限为 0.5.

Table 1 The identification of SRAC

表 1 小值域属性组的识别

Attribute name	Range size of attribute	Storage space of attribute	Selected attributes by algorithm		
			α=0.5	α=0.3	α=0.7
Orderkey	7 500 000	21	0	0	0
Partkey	5 960 934	21	0	0	0
Suppkey	300 000	21	0	0	0
Linenumber	7	38	1	1	1
Quantity	50	21	1	0	1
Extendedprice	2 093 773	21	0	0	0
Discount	11	21	1	1	1
Tax	9	21	0	0	0
Returnflag	3	1	1	0	1
Linestatus	2	1	1	0	1
Shipdate	2 526	7	1	1	1
Commitdate	2 466	7	0	0	0
Reciptdate	2 555	7	0	0	0
Shipinstruct	4	25	1	1	1
Shipmode	7	10	1	1	1
Commit	11 466 069	44	0	0	0

第 2 组实验是两个 SRAC 识别算法性能的比较.置冗余度下限为 0.5,然后分别执行算法 Recon-greedy 和 Recon-gen.图 1 给出了两种算法识别的 SRAC 时属性个数随扫描数据库的遍数的变化情况.两条曲线基本呈阶梯状变化,这是由于,在遗传算法中要等到每一代种群处理完之后才可能引起频繁属性的个数的变化,而贪心算法中每次循环完成后才引起小值域属性组的一次扩充.其次,在得到 SRAC 的过程中,遗传算法的收敛速度较贪心算法快很多,这主要是因为遗传算法在连续两代种群的变化过程中可能向 SRAC 中引入多个属性,且淘汰机制的引入使得种群规模逐渐减小.最后,两种算法得出的 SRAC 之间没有严格的包含关系.

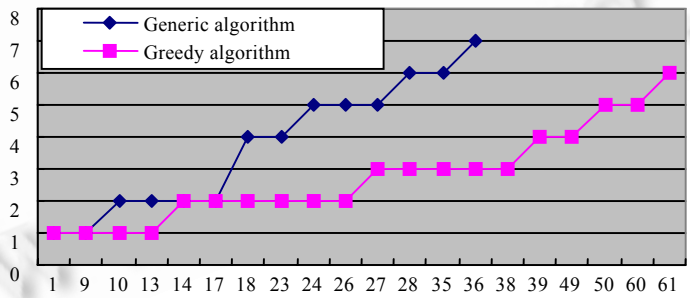

Fig.1 The comparison of SRAC identification algorithms

图 1 小值域属性组识别算法的比较

第 3 组实验研究拆分压缩对数据库性能的影响.设置冗余度上限为 0.5,由算法 Recon-gen 得到一个 0.5 小值域属性组,它包含 Linenumber,quantity,discount,shipdate,shipinstruct,shipmode 等 6 个属性,由此按照第 5 节中的方法将关系拆分成两种关系,压缩掉的空间是原表所用空间的 21%.然后,从 TPC-H 的查询中挑选了 8 个 sql 语句在压缩后的数据库中执行,结果见表 2.第 1 列给出了 sql 语句的编号;第 2 列是相应语句在未压缩的数据库中的执行时间;第 3、第 4 列是相应语句在压缩数据库中的执行时间,两列的区别为是否使用了新增属性上的索引.注意,第 8 个 sql 语句是一个嵌套的 sql 语句,且内层 sql 语句的 select 分句中出现了表达式 *l*_extendedprice*(1–*l*_discount)–ps_supplycost**l*_quantity,这个表达式中的属性被拆分到两种关系中,而查询处

理时未能将表达式分解成两种关系上的子表达式,这导致操作时间的增加.

Table 2 The performance of the compressed database

表 2 压缩数据库的性能

Query	Original relation	Compressed relation without new index	Compressed relation with new index
sql1	155.533	90.457	87.364
sql2	158.157	130.115	132.158
sql3	145.331	180.627	160.151
sql4	281.164	177.572	160.771
sql5	223.532	108.787	106.844
sql6	667.95	878.063	424.218
sql7	1 459.428	1 083.628	1 060.685
sql8	5 427.895	6 194.844	5 947.602

此外,文献[21]中也讨论了 LINEITEM 关系的压缩.在压缩比方面,Oracle 的压缩方法的压缩比为 38%.但是,Oracle 的压缩方法本质上是将字典压缩方法整合到数据库的物理层实现上,故拆分压缩后的海量关系仍然可以用 Oracle 的压缩方法进一步压缩.从数据库整体性能上来看,Oracle 的压缩方法使得数据库整体性能提高了约 10%.如果将拆分压缩后的关系用 Oracle 的压缩方法作进一步压缩,压缩比和数据库整体性能均有可能进一步提高.目前,Oracle 的这种压缩方法还不可得,也很难从底层直接实现,因此,关于这两种压缩方法的结合还无法进行讨论.

8 结 论

本文根据海量关系中经常出现小值域属性组这一特点提出了拆分压缩技术.本文给出了两种在海量关系中识别小值域属性组合的算法,实验表明,算法中使用的冗余度下限的合理范围应该在 0.4~0.6 之间.实验结果表明,拆分压缩技术可以取得较好的压缩效果,并能够提高数据库查询处理的性能.

References:

[1] Margritis D, Faloutsos C, Thrun S. Netcube: A scalable tool for fast data mining and compression. In: Apers PMG, ed. Proc. of the 27th Int'l Conf. on Very Large Data Bases. Mumbai: Morgan Kaufmann Publishers, Inc, 2001. 311−320.

[2] Gao H, Li JZ. Cube algorithms for very large compressed data warehouse. Journal of Software, 2001,12(6): 830~839 (in Chinese with English abstract).

[3] Westmann T, Kossmann D. The implementation and performance of compressed database. ACM SIGMOD Record, 2000,29(3): 55−67.

[4] Chen ZY, Gehrke J, Korn F. Query optimization in compressed database systems. In: Sellis T, ed. Proc. of the ACM SIGMOD Int'l Conf. on the Management of Data. New York: ACM Press, 2001. 271−282.

[5] Ray G, Harisa JR, Seshadri S. Database compression: A performance enhancement tool. In: Chaudhuri S, Deshpande A, Krishnamurthy R, eds. Proc. of the Conf. on Management of Data. India: Tata McGraw-Hill, 1995. 106−125.

[6] Poess M, Potapov D. Data compression in oracle. In: Freytag JC, Lockemann PC, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases. Mumbai: Morgan Kaufmann Publishers, Inc, 2003. 937−947.

[7] Cormack GC. Data compression on a database system. Communications of the ACM, 1985,28(12):1336−1342.

[8] Westmann T, Kossmann D. The implementation and performance of compressed database. ACM SIGMOD Record, 2000,29(3): 55−67.

[9] O'Connell SJ, Winterbottom N. Performing joins without decompression in a compressed database system. ACM SIGMOD Record, 2003,32(1):55−67.

[10] Berchtold S. Independent quantization: An index compression technique for high dimensional data space. In: Proc. of the 16th Int'l Conf. on Data Engineering. New Orleans: IEEE Computer Science Society Press, 2000. 577−588.

[11] Antoshenkov G, Lomet D, Murry J. Order preserving string compression. In: Su YW, ed. Proc. of the 12th Int'l Conf. on Data Engineering. New Orleans: IEEE Computer Science Society Press, 1996. 655−663.

[12] Jagadish HV, Ng RT, Ooi BC, Tung AKH. ItCompress: An iterative semantic compression algorithm. In: Su YW, ed. Proc. of the 16th Int'l Conf. on Data Engineering. New Orleans: IEEE Computer Science Society Press, 2004. 646−657.

[13] NG WK, Ravishhankar CV. Relational database compression using augmented vector quantization. In: Yu PS, ed. Proc. of the 16th Int'l Conf. on Data Engineering. New Orleans: IEEE Computer Science Society Press, 1995. 540−549.

[14] Li JZ, Rotem D, Srivastava J. Aggregation algorithms for very large compressed data warehouses. In: Atkinson MP, Orlowska ME, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases. Mumbai: Morgan Kaufmann Publishers, Inc, 1999. 651−662.

[15] Chen ZY. Building compressed database system [Ph.D. Thesis]. New York: Connell University, 2002.

附中文参考文献:

[2] 高宏,李建中.超大型压缩数据仓库上的 CUBE 算示.软件学报,2001,12(6):830−839.

2005 年全国软件与应用学术会议

征 文 通 知

由中国计算机学会系统软件专业委员会和软件工程专业委员会联合主办，南京大学计算机软件新技术国家重点实验室及计算机科学与技术系承办的"2005 年全国软件与应用学术会议(NASAC 2005)"将于 2005 年 10 月 18 日~20 日在江苏南京召开。

一、征文范围 (但不限于下列内容)

1.操作系统的理论与实践
2.面向对象与软件 Agent
3.软件中间件与应用集成
4.软件语言与编译技术
5.需求工程的方法与技术
6.分布式与嵌入式软件设计
7.软件工程工具与环境
8.软件体系结构与设计模式
9.构件技术与软件复用
10.软件过程管理与改进
11.软件开发方法及自动化
12.软件理论与形式化方法
13.软件质量、测试与验证
14.软件标准与规范

二、论文要求

1. 论文必须未在杂志和会议上发表和录用过。
2. 论文篇幅限定 6 页(A4 纸)内。
3. 应以 PDF 或 PS 格式提交论文。有关文章的版心等格式及参考文献样式将在会议网址上公布，敬请关注。

三、重要日期

1. 论文投稿截止日期(邮戳为准)：2005 年 5 月 30 日
2. 论文录用通知日期：2005 年 7 月 10 日
3. 软件产品发布和演示报名截止日期：2005 年 9 月 18 日
4. 学术会议及活动日期：2005 年 10 月 18 日~20 日

四、联系方式

1. 会务组地址: 210093 江苏 南京 汉口路 22 号 计算软件新技术国家重点实验室
电话: 025-83593467；025-83593670
传真: 025-83686596；025-83593467
EMAIL: keysoftlab@nju.edu.cn
NASAC 2005 网址: http://keysoftlab.nju.edu.cn/NASAC2005
2. 联系人
(1).论文接收: 陶先平 徐煊
(2).其他会务: 武林红 胡又林 张建莹
3. 投稿人注意

电子投稿是可接受的。投稿时，随稿附上您的详细地址、邮编、电话(办公室、实验室、家中)、手机以及 E-mail 地址。更详细的内容请访问 NASAC2005 网址。