位图索引技术及其研究综述

沈阳理工大学　程　鹏

[摘　要] 位图索引是一种新兴的索引技术，特别适合于只读性海量数据的索引。本文对现有各种位图索引进行了分类，介绍了BBC、WAH、范围编码、区间编码、分箱和基于 Bloom Filter 编码的位图索引压缩和查询技术。比较了各种位图索引的空间和时间复杂度，讨论了如何根据数据的特性选择合适的位图索引，并指出位图索引的未来研究问题和方向。

[关键词] 位图索引　索引　压缩　数据仓库

1.前言

位图索引由 P'ONeil 在 1987 年提出，并在一个商用数据库系统 Model 204 上首次应用 [1]。最简单的位图索引称为基本位图索引(Basic Bitmap Index)，它利用一个位向量(Bit Vector)来表示被索引属性的某个取值是否在被索引数据中存在。基本位图索引的存储结构简单，并可以利用高效的位逻辑运算(AND/OR/NOT/XOR 等)来回答复杂查询[2,5]。

本文综述了位图索引的基本原理以及位图索引的数据压缩和查询技术，比较了常用压缩位图索引的查询性能，并提出了位图索引进一步需要研究的问题，希望对位图索引的应用者和研究者有所裨益。

2.基本位图索引

不失一般性，假定被索引数据以关系表的形式存在，其包含的总记录数为 N，A 为某属性，其基数为 C，可以将 A 的所有取值按顺序映射为 0 到 C- 1 之间的整数。基本位图索引为属性 A 的每个取值 $m_i(0\leq i\leq C-1)$分别建立一个位图(Bitmap)B_i，其中 B_i 的第 j 位$(0\leq j\leq N-1)$记为 b_{ij}，A_j 表示第 j 个记录在属性 A 上的取值，则有

$$\begin{cases} b_{ij}=1, & \text{当 } A_j=m_i \\ b_{ij}=0, & \text{当 } A_j\neq m_i \end{cases}$$

这种编码方案对于等值查询极为有效，所以也称为等值编码方案。例如，表 1 为关系表的某属性(C=9，N=6)对应的基本位图索引。其中，$B_0\sim B_8$ 分别为 9 个取值对应的位图。

表 1　关系表中属性 A 的取值及其基本位图索引

RID	A	B_8	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
0	3	0	0	0	0	0	1	0	0	0
1	2	0	0	0	0	0	0	1	0	0
2	1	0	0	0	0	0	0	0	1	0
3	2	0	0	0	0	0	0	1	0	0
4	8	1	0	0	0	0	0	0	0	0
5	2	0	0	0	0	0	0	1	0	0

3.压缩位图索引

3.1 基于 RLE 压缩的方法

RLE 是数据压缩的一种常用方法，其基本思想为：数据中连续重复出现的值(1 或 0)称为一个"节"(Run)，每个节用其值加上该节的长度表示。采用 RLE 压缩位图索引的典型代表有 BBC 和 WAH。

1) BBC(Byte- aligned Bitmap Code)

BBC 将被压缩的位序列按字节分组为一系列节，压缩后的数据仍然以字节为单位。每节压缩后包括一个 fill 部分和一个 tail 部分。BBC 中的字节包括两类：fill 字节和 literal 字节。fill 字节必须为全 1 或全 0，分别称为 1- fill 或 0- fill，literal 字节则按原文(不压缩)存放各位。单边(One- sided)BBC 只对 0- fill 压缩，适合于稀疏位图索引，双边(Two- sided)BBC 则对 0- fill 和 1- fill 均压缩。一个头字节(Header byte)用于表明节的种类。图 1 为一个位序列对应的 BBC 压缩结果(所有字节均用十六进制表示)。

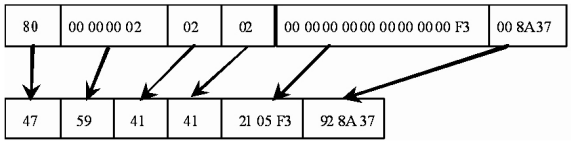

图 1　BBC 位图压缩示例

2) WAH(Word- Alignment Hybrid Code)

BBC 以字节为单位进行位运算，然而计算机的 CPU 以字为单位进行位运算，所以 K. Wu 等人提出了以字为单位的位图索引编码：WAH。为了加快直接在压缩位图上的位运算速度，WAH 采用了更为简单的编码方式。WAH 中没有头字或头字节，这样就消除了压缩数据的前后依赖性，便于 CPU 并行处理。

WAH 中只有两类字：literal 字和 fill 字，用最高位以示区别(0 为 literal，1 为 fill)。令计算机的字长为 w，则 literal 字可保存 w- 1 个位。fill 字的次高位表示重复位是 0 还是 1，余下的 w- 2 位则用于表示一个节的长度(以字为单位)。图 2 表示一个位序列对应的 WAH 压缩结果，其中最后两行分别为未压缩的位序列和 WAH 压缩后的位序列(16 进制)。表中最后一个节只剩下 4 位，称为 Active 字，在存储和查询时需要做特殊处理。直接在 WAH 上的位图索引查询算法[8]。

128 位	1×1，20×0，3×1，79×0，25×1			
31 位分节	1，20×0，3×1，7×0	62×0	10×0，21×1	4×1
压缩前	40000380	00000000 00000000	001FFFFF	0000000F
WAH	40000380	80000002	001FFFFF	0000000F

图 2　WAH 压缩位图示例(m×b 表示重复出现 m 个 b)

3.2 基于编码的方法

基于编码的方法是通过减少位图的个数实现位图压缩，这种方法对于位图索引的稀疏度一般不太敏感，但是在查询时可能需要访问大量的位图。P.O'Neil 首先提出了 Bit- sliced 位图索引编码方案[1,2,6]，可将基数为 C 的属性对应的位图个数压缩至$[\log_2 C]$。C.Y. Chan 提出了更具一般性的、基于属性值分解和编码的位图压缩方案框架，使基本位图索引和 Bit- sliced 位图索引成为该框架的特例。该框架将属性值分解为多个组部(Component)，每个组部可分别按等值、范围、区间编码方案进行编码。

1) 属性值的分解

表 2 基 <3,3> 等值编码方案

RID	A	分 解	B_{12}	B_{11}	B_{10}	B_{02}	B_{01}	B_{00}
0	3	1×3+0	0	1	0	0	0	1
1	2	0×3+2	0	0	1	1	0	0
2	1	0×3+1	0	0	1	0	1	0
3	2	0×3+2	0	0	1	1	0	0
4	8	2×3+2	1	0	0	1	0	0
5	2	0×3+2	0	0	1	1	0	0
6	2	0×3+2	0	0	1	1	0	0
7	0	0×3+0	0	0	1	0	0	1
8	7	2×3+1	1	0	0	0	1	0
9	5	1×3+2	0	1	0	1	0	0
10	6	2×3+0	1	0	0	0	0	1
11	4	1×3+1	0	1	0	0	1	0

对于任一属性值 v(0≤ v≤ C- 1),给定一组整数序列 $<b_{n-1}, b_{n-2},\cdots,b_1>$,并定义 $b_n=\lceil C/\prod b_i \rceil$,1≤ i≤ n- 1,则 v 可以表示为一系列数字位 $<v_n,v_{n-1},\cdots,v_1>$ 的形式。

$$v=v_n(\prod_{j=1}^{n-1} b_j)+\cdots+v_i(\prod_{j=1}^{i-1} b_j)+\cdots+v_2 b_1+v_1,\quad 0\le v_i\le b_i$$

其中 $<b_n, b_{n-1},\cdots,b_1>$ 称为基,若 $b_n=b_{n-1}=\cdots=b_1=b$,则称为“基为 b 的统一基(Uniform)”。基将一个属性值分解为 n 个组部,每个组部可以分别建立对应的位图索引,从而使位图索引的个数与 n 以对数级减少,而查询时间与 n 呈线性增长。例如对于表 1 的基本位图索引,按基 <3,3> 以及等值编码方案(若 $v_i=j$,则编码为 1),其对应的压缩位图索引如表 2 所示。

2) 基于范围的编码方案

范围编码在等值编码的基础上，将每个组部中值为 1 的左侧所有位全部置 1,其他位不变。比如,对于表 2 对应的等值编码,其对应的范围编码(若 $v_i\le j$,则编码为 1)如表 3 所示。由于每个组部的最高位恒为 1,故其位图不需要实际存储,位图个数进一步压缩为 4 个。当采用基为 b 的统一基以及范围编码时,索引退化为 Bit- sliced 位图索引。

表 3 基 <3,3> 范围编码方案

RID	A	B_{11}	B_{10}	B_{01}	B_{00}
0	3	1	0	1	1
1	2	1	1	0	0
2	1	1	1	1	0
3	2	1	1	0	0
4	8	0	0	0	0
5	2	1	1	0	0
6	2	1	1	0	0
7	0	1	1	1	1
8	7	0	0	1	0
9	5	1	0	0	0
10	6	0	0	1	1
11	4	1	0	1	0

表 4 基 <9> 区间编码方案

RID	A	B_{04}	B_{03}	B_{02}	B_{01}	B_{00}
0	3	0	1	1	1	1
1	2	0	0	1	1	1
2	1	0	0	0	1	1
3	2	0	0	1	1	1
4	8	0	0	0	0	0
5	2	0	0	1	1	1
6	2	0	0	1	1	1
7	0	0	0	0	0	1
8	7	1	0	0	0	0
9	5	1	1	1	0	0
10	6	1	1	0	0	0
11	4	1	1	1	1	0

3) 基于区间的编码方案

针对形如“$v_1\le A\le v_2$”的区间型查询,Chan 提出了更为有效的区间编码方案[4]。该方法所需空间约为等值编码或范围编码所需空间的一半，却有更好的区间查询性能，对于等值查询或范围查询也可较好支持。对于第 i 个组部,区间编码需要$\lceil b_i/2 \rceil$个位图,对于每个位图 B_{ij},若 $j\le v_i\le j+m$,($m=\lfloor b_i/2 \rfloor - 1$),则编码为 1。对于表 1 所示的位图索引,若基选为 <9>，则 m=3，采用区间编码的位图索引如表 4 所示。其中位图 $B_{00}\sim B_{04}$ 对应的编码区间分别为[0,3]、[1,4]、[2,5]、[3,6]和[4,7]。与范围编码类似,区间编码对于等值查询、范围查询和区间查询,每个组部最多只需 2 次位图扫描即可完成[4]。

3.3 基于分箱的方法

该方法特别适用于有序属性(如整数、实数等)。首先,将被索引属性的取值分解为若干连续的区间,称为“箱”,然后位图索引针对每一个箱进行索引,从而显著减少了位图的数量。但基于分箱的位图索引对查询具有不稳定性,如果查询条件不是正好处在箱的边界,那么必须再次到边界箱对应的原始数据中检查数据是否符合查询条件,这一操作为称为“候选项检查(Candidate Check)”[15],其消耗的时间可能数倍于检索位图索引所需时间。对于表 5 所示的分箱位图索引，若查询条件为“37≤ A<63”,需涉及箱 1、2、3,而箱 1 和 3 为边界箱,此时需要对箱 1 和箱 3 对应的位图中位为 1 的元组进行候选项检查，结果只有 61.7 符合查询条件。

表 5 基于分箱的位图索引

属性 A	B_4 [0,20)	B_3 [20,40)	B_2 [40,60)	B_1 [60,80)	B_0 [80,100)
34.7	0	1	0	0	0
94	0	0	0	0	1
24.9	0	1	0	0	0
15.5	1	0	0	0	0
61.7	0	0	0	1	0
67.2	0	0	0	1	0
58.6	0	0	1	0	0

3.4 基于 Bloom Filter 的方法

Bloom Filter 采用哈希方法用位串表示数据集合，可以有效支持数据元素的哈希查找操作，但由于哈希函数的冲突性会使数据元素的查找造成一定的误差。

基于 Bloom Filter 的位图索引将位图索引视为布尔矩阵,令 M[i,j]为布尔矩阵的第 i 行第 j 列元素,$H_k(i,j)$为对行 i、列 j 进行变换的第 k 个哈希函数(1≤ k≤ p),其返回值 b_k 为一个 m 位的二进制数,AB 为一个长度为 2^m 的二进制位串,称为近似位图(Approximate Bitmap)。建立位图索引时，将 M 中所有值为 1 的元素依次运用 p 个哈希函数 H_k 计算出 b_k,并令 $AB[b_k]=1$。查询时,首先根据查询条件将查询变换为布尔矩阵的一系列行列对 <i,j>,再利用每个哈希函数 H_k 计算出 b_k,若至少存在一个 b_k 使 $AB[b_k]=0$,则可以断定 M[i,j]=0,反之,M[i,j]以极高的概率等于 1。若要获得精确结果,则需要到原始数据中进行“候选检查”。利用近似位图查找时,若 M[i,j]=1,则查找结果必然正确,但 M[i,j]=0 时,查找结果可能认为 M[i,j]=1,这种情况称为假阳性(False Positive)。假阳率的理论值近似为 $(1-e^{-ps/n})^p$，其中 p 为哈希函数的个数,$n=2^m$ 为近似位图的位数,s 为布尔矩阵中包含 1 的个数。对于只读性数据,s 不变,n 可以预设,因而假阳率是可控的。n 越大,假阳率越低,给定 n 的值,p 的最佳值可以通过对理论值公式求导得出。

对于给定被索引数据集可以有三种建立近似位图的方法。1)整个数据集对应一个近似位图,适合于高维度、高基数的数据集 ;2)每个属性对应一个近似位图,适合于数据分布偏斜的数据集 ;3)每个取值对应一个近似位图,适合于均匀分布的数据集。

4.各种压缩位图索引的性能比较

文献[2,5,7,11]先后比较了各种位图索引的空间和时间性能,结果并没有发现任何一种位图索引在任何情况下均最优，而是取决于不同的数据类型、分布、查询类型等因素。下面对各种位图索引的特点和性能作一比较。

BBC 是一种以字节为单位的 RLE 类位图压缩方案,其压缩率与 LZ 接近，但不需要解压缩即可直接支持位运算。在位图稀疏的情况下,BBC 的查询速度可以快于基本位图索引。WAH 简化了 BBC 的编码方案,消除了字之间的依赖性,虽然其存储空间较 (下转第 533 页)

大纲明确了课程内容的重点、难点及解决办法。在教学过程中及时把教研、科研成果或学科最新发展成果引入课堂，扩展学生视野。教学内容的设计紧紧围绕以下两方面：一方面为学生解决科学与工程中的实际问题奠定基础，另一方面为学生后继课程的学习提供必要的知识准备。

⑵ 在实验课教学方面，实验课是数值分析课程中的重要的实践教学环节，目的是使学生得到选择算法、编写程序、分析数值结果、编写数值试验报告、课堂讨论等环节的综合训练，巩固理论课教学内容，培养学生使用计算机进行科学计算和解决实际问题的能力，为以后从事现代数学科研工作以及科学与工程计算打下良好的基础。围绕这一目的，提出了实验性教学的设计思想，主要包括强化基础实验部分；补充综合型实验和创新型实验部分；重视实验结果的总结和实验报告的书写。

三、教学方法与手段的改进与创新

1.教学方法

在教学过程中采用“问题教学”的授课方式。其基本思路是：采用数学建模的思想和方法，从所要解决的实际问题出发运用所学知识通过归纳、分析、提炼等手段建立其相应的数学模型，从而提出相应的数学问题；然后从理论上研究、讨论解决这个数学问题的基本思想、方法，分析该方法的优缺点及所能解决问题的类型，进而给出解决实际问题的数学思想、方法；最后让学生亲自动手编程做实验，实践用所学的知识来解决一些简单的实际问题，通过实践掌握计算方法解决实际问题的基本过程、思考方式和规律，做到活学活用，学以致用。这样不仅可以激发学生学习数学、特别是计算数学的兴趣和欲望，还将教师扎实的理论知识与丰富的实践能力、解决实际问题的心得体会通过教师授课与学生实验这两个环节传授给学生。另外，在上机时布置一些有启发性的题目，并自行设计算法解决，从而调动了学生的积极性，有效地培养学生的创新精神和创新能力，促进了学生的个性发展。必须充分发挥学生的积极性，除了增强自习和习题在教学中的作用外，努力结合计算机程序设计进行数值试验显得异常重要。为了使学生切实掌握教材的内容，并把学生听讲、自习和上机试验的情况反馈给教师，编写切合课程内容的、题量适宜的习题集以及上机实习指导书。注重教学方法的多样化：采用“提问式”教学，增加“演示型”教学，重视“思想方法”的讲解，倡导“参与型”教学，采用灵活多样的方式组织课堂教学，提高学生的学习兴趣。

教学过程，从本质上讲是一种特定学习环境的营造过程。在课堂教学中，教师切忌包揽一切，忽视学生学习的主体作用，而应“与学生一起做”，同时注意将新的知识点导入到学生熟悉的、容易明白的知识和思维情景中，最后达到“使学生能够做”，实现“以人为本，学为核心”的教学目的。

2.教学手段

第一，注意现代多媒体技术的使用，恰当地引入多媒体教学手段，利用数学软件和课件帮助学生理解数值分析教学的难点，或用现有数学软件对一些简单问题求解的过程直接进行演示。教师采用多媒体课件还可省去在课堂上书写、画图的大量时间，有利于教师把精力集中在讲透基本概念、基本原理和算法的构造等方面，从而提高课堂教学效率和教学质量。

第二，注重网络的使用，充分利用网络的方式与学生互动和交流，如有些补充材料和习题解答，在课堂上没有时间讲，就通过网络发送给学生；学生有问题，就发邮件给老师，由老师来解答，同时还开通了网上答疑系统，安排教师定时进行系统维护，及时回复学生的提问，积极主动引导学生的学习兴趣，使学生的学习不再受到时间和空间的限制。

第三，重视科学计算软件的使用。数值分析是面向实际问题求解的，离不开科学计算中的数学软件。数值分析上机实习要求学生能编制结构化的通用程序，输出计算结果，并对结果进行分析。采用学生熟悉的高级语言自编程序是学生学习数值分析不可缺少的一个步骤。另一方面，使用成熟的计算软件做数学实验，可保证学生把更多的时间和精力投入到算法学习中，以提高数学、计算机综合素质。同时，对于比较简单的数学问题直接应用计算软件求出数值解，可使问题简单化，便于学生取得相应的计算经验，印证和补充理论，为进一步发展打下良好的基础。

通过数值实验这门课程的课内外训练，学生巩固了课堂内容，学会了从事科学计算的 C 语言、Matlab 数学软件等必要工具，锻炼了实践能力，培养了创新能力，同时也为继续学习后续专业课程奠定了良好的实践基础。当然，作为数值分析教学改革的工作和实践，还需要进一步的研究和完善。

参考文献

[1] 杨扬.数值分析课程教学初探[J].徐州教育学院学报,2008,23(3):144-145.

[2] 张尧学.认真学习教育部 2001 年 4 号文件.狠抓高等教育教学质量[J].大学数学教育,2002,(1):7-10.

[3] 姜启源等.数学模型[M].北京:高等教育出版社,2003.

[4] 陈晓瑛,龚日朝.国内外数学实验教学的现状分析与展望[J].株洲师范高等专科学校学报,2004,(10):50-52.

[5] 王正林等.精通 MATLAB 科学计算[M].北京:电子工业出版社,2007.

（上接第 531 页） BBC 偏大，但查询性能得到 2～100 倍的提高[8]。BBC 和 WAH 主要目标是压缩每一个位图的空间，故没有解决查询范围较宽的范围查询或区间查询问题。

Chan 研究表明，具有两个组部的属性值分解可以很好地平衡索引空间和查询性能[3]。等值编码对于等值查询速度最快，但对于范围查询速度最慢。范围编码对于范围查询速度最快。区间编码对于等值查询、范围查询和区间查询速度等同于范围编码，但是空间约节省一半，总体上性能优于范围编码。另外，基于编码的位图索引也可利用其他压缩算法(如 LZ、BBC、WAH 等)进行进一步压缩，以进一步减少 I/O 操作时间，从而加快位图索引的查询。

基于 RLE 的方法从纵向压缩了每个位图的长度，基于编码或分箱的方法从横向减少了需要存储的位图的数量，而基于 Bloom Filter 的方法抛弃了传统位图索引利用硬件快速实现位运算的优势，代之以只对值为 1 的位进行哈希存储，其查询时间复杂度不依赖于整个数据集的大小，而是通过直接访问的方式，只访问数据集中与查询条件相关的子集即可，其时间复杂度与查询相关子集的大小成正比。实验表明，在使用空间与 WAH 相当，查询精度控制在 90%以上的情况下，近似位图的查找速度比 WAH 快 1～3 个数量级[9]。近似位图是提出较晚的位图索引压缩方法，目前研究表明，近似位图方法尤其适合于查询条件涉及少量数据子集的查询，以及只需近似查询的应用(如数据可视化[9]、数据仓库环境[9]等)。

5.结论及展望

位图索引虽早已被提出，但在商用数据库中远不及 B 树索引使用普遍，其主要原因为：采用位图索引需要修改数据库系统的底层设计，用户对位图索引还不了解，加之位图索引的各种压缩方案的优劣还没有得到公认，位图索引的最终设计模型还有待于进一步研究[11]。如何将各种位图索引有机结合起来，建立自适应、动态位图索引策略，甚至在同一位图索引中，每个位图均可采用不同的压缩方案[5]，联合发挥各种压缩技术的优势，也是位图索引的研究方向之一。

参考文献

[1] P.O'Neil. Model 204 architecture and performance. In Proceedings of the 2nd international Workshop on High Performance Transaction Systems, September 28- 30, Asilomar, CA, USA, 1987,40- 59.

[2] P. O'Neil and D. Quass. Improved query performance with variant indexes. SIGMOD '97. ACM Press, New York, NY, 38- 49.

[3] C. Chan and Y. Ioannidis. Bitmap index design and evaluation. SIGMOD '98. ACM Press, New York, NY, 355- 366.

[4] C. Chan and Y. Ioannidis. An efficient bitmap encoding scheme for selection queries. SIGMOD '99. ACM Press, New York, NY, 215- 226.

[5] T. Johnson. Performance measurements of compressed bitmap indices. VLDB 1999. Morgan Kaufmann Publishers, San Francisco, CA. 278- 289.

[6] D. Rinfret, P. O'Neil, and E. O'Neil. Bit- sliced index arithmetic. SIGMOD Rec. 2001, 30(2), 47- 57.

[7] K. Wu, E. Otoo, and A. Shoshani. On the performance of bitmap indices for high cardinality attributes. VLDB 2004, Toronto, 30, 24 - 35.

[8] K. Wu, E. Otoo, and A. Shoshani. Optimizing bitmap indices with efficient compression. ACM Transactions on Database Systems, 2006, 31(1), 1- 38.

[9] T. Apaydin, G. Canahuate, H. Ferhatosmanoglu, and A. S. Tosun. Approximate encoding for direct access and query processing over compressed bitmaps. VLDB 2006, Seoul, Korea. 846 - 857.

[10] R.R.Sinha, M.Winslett. Multi- resolution bitmap indexes for scientific data. ACM Transactions on Database System,2007,32(3), article # 16.

[11] E. O'Neil, P. O'Neil and K. Wu. Bitmap index design choices and their performance implications. IDEAS 2007, 72- 84.

[12] G. Antoshenkov. Byte- aligned bitmap compression. Technical Report, Oracle corp., 1994.

[13] G.Canahuate, M.Gibas, H.Ferhatosmanoglu. Update conscious bitmap indices. SSBDM 2007.

[14] M.C. Wu, A.P. Buchmann. Encoded bitmap indexing for data warehouses. ICDE 1998, Orlando, Florida, USA. IEEE Computer Society Press.

[15] D.Roteom,K.Stockinger, K.Wu. Efficient binning for bitmap indices on hi- cardinality attributes. Techinical Report, Barckley Lab. 2006.

[16] B.Bloom. Space/ time Tradeoffs in hash coding with allowable errors. Communications of the ACM, 13(7):422- 426, July 1970.