数据仓库的索引技术

【摘要】

数据仓库系统中性能问题是非常重要的问题，索引技术是提高性能的有效手段之一。本文分析与评述了数据仓库中所使用的索引技术，这些技术包括：位图索引、B 树索引、连接索引、投影索引等。针对各自的特点，对这几种技术进行了比较，对数据仓库的研究与开发有较大的指导意义。

【关键词】

数据仓库 位图索引 B 树索引 连接索引 投影索引

【ABSTRACT】

The efficiency of data warehouse is an important issue. Index technologies can improve the efficiency of data warehouse. The paper analyzes and compare with index technologies in data warehouse, including B-Tree index, bitmap index, join index and projection index. We compare with their characteristics, which can give an effective guide on development and study on data warehouse.

【KEYWORDS】

Data Warehouse；Bitmap Index；B-Tree Index；Join Index；Projection Index

1 前言

数据仓库(Datawarehouse)是一个面向主题的、集成的、稳定的、随时间而变化的包含大量历史数据的数据集合，它用于支持经营管理中的决策制定过程。

在数据仓库中使用高效的索引技术不仅是必要的，而且是可行的。数据仓库面向分析型应用，其数据是相对稳定的，对数据仓库的操作主要是读取查询数据，很少进行更新。这些少量的更新操作一般都是在非工作时间进行的，而且采用批处理方式，具有周期性。基于数据仓库的上述特点，在进行数据的更新和索引的重新组织时，数据和索引都处于未被使用的状态，因此可以采用一些复杂的索引来提高数据仓库的查询性能。目前，传统的数据库索引技术仍然是数据仓库中建立索引的重要方法。同时，新的数据仓库索引技术也在不断发展。

2 数据仓库常用的索引技术

数据仓库管理系统中的索引能够提供一个相对快捷的方式定位数据。数据仓库中常用的索引技术有位图索引(bitmap index)、B 树索引(B-Tree index)、连接索引(join index)、投影索引(projection index)等。

2.1 位图索引

位图索引是数据仓库系统最常用的索引技术，目前有两种位图索引，一种是简单的位图索引，另一种是对简单位图索引的改进，称为编码位图索引。位图索引基本设计思想是用一个 0、1 位串来表示一个元组某一属性的取值，位串中的位的位置表示了关系表中元组的位置。

2.1.1 简单位图索引技术(Simple Bitmap Index)

对于属性域中的每个值 v，设计一个不同的位向量 Bv。如果给定的属性域包含 n 个不同的取值，则位图索引中的每项要用 n 位向量来表示。如果数据表中某一元组的属性值为 v，则在位图索引的对应行表示该值的位为 1，该行的其它位为 0。

假设在一个与销售事实相对应的数据立方中，有一个顾客的性别属性 Gender，一个是产品的种类属性 Item。其中 Gender 属性有两个不同的值：“M”和“F”。产品的种类属性 Item 有四个不同的取值：“a”，“b”，“c”和“d”。设数据立方中共有 8 个元组。如果在 Gender 属性上建立位图索引则需要 2 个位向量，每个向量共 8 位。在 Item 上建立位图索引需要四个位向量，每个向量共 8 位。这两个索引如图 1 所示。

基本表

RID	Item	Gender
R1	a	F
R2	a	M
R3	b	F
R4	b	F
R5	b	M
R6	c	F
R7	c	M
R8	d	M

Item 位图索引表

RID	a	b	c	d
R1	1	0	0	0
R2	1	0	0	0
R3	0	1	0	0
R4	0	1	0	0
R5	0	1	0	0
R6	0	0	1	0
R7	0	0	1	0
R8	0	0	0	1

Gender 位图索引表

RID	M	F
R1	0	1
R2	1	0
R3	0	1
R4	0	1
R5	1	0
R6	0	1
R7	1	0
R8	1	0

图 1 简单位图索引示意图

2.1.2 编码位图索引(Encoded Bitmap Index)

编码位图索引是对简单位图索引的改进。为了压缩位图索引向量，利用编码的方法把某一属性的不同值进行编码。编码位图索引由一个位图向量集合(不同于简单位图索引的位图向量集)、一个映射表(MappingTable)和一个检索布尔函数(Retrieval Boolean Function)集组成。其中映射表实现对一个属性域值的编码，该映射表定义了一组检索布尔函数 f，实现属性值到编码值的映射。一个检索函数的自变量为属性 I 的域，其值域是具有 k 个比特位的编码数。这里，k=(log2|I|)，|I|表示属性 I 的域中的不同取值个数，即属性 I 的基数。一个元组 j(记为 tj)中的属性 I 的值记为 tj．I，根据映射表 tj．I 的编码值为 f(tj．I)，它的第 i 位记做 f(tj．I)[i](i=0，1，…，k-1)，则属性 I 的索引位图向量集中的第 j 行的 Bi 位等于 f(tj．I)[i] (i=0，1，…，k-1)。

例如，图 2 是对应图 1 的基本表中的 Item 属性的编码位图索引。其中，“a”编码为“00”，“b”编码为“01”，“c”编码为“10”，“d”编码为“11”。对于 Item 取值为“a”的元组，在位图向量 B1 和 B0 上相应的取值都为“0”。类似的 Item 取值为“b”的元组，位图向量 B，和 B0 的取值为“0”和“1”。本例中，属性 Item 有 4 个不同的取值，所以 k=2。

基本表

RID	Item
R1	a
R2	a
R3	b
R4	b
R5	b
R6	c
R7	c
R8	d

位图向量

RID	B2	B0
R1	0	0
R2	0	0
R3	0	1
R4	0	1
R5	0	1
R6	1	0
R7	1	0
R8	1	1

映射表

Item 值	编码值
a	00
b	01
c	10
d	11

图 2 编码位图索引举例

上例中在 Item 属性上有四个不同值，对于简单位图索引要四个位向量，而对于码的位图索引只需二个位向量。如果有 12000 个不同值，对于简单位图索引需要 12000 个位向量，对于编码位图索引只需(log2|12000|)=14 个位向量。显然编码位图索引较简单位图索引有较大的优势。

2.2 B 树索引

B 树是一种动态调节的平衡树，它引入了一种效率很高的外查找机制，比较适合于字段值分散且重复值少的字段。一个 B 树索引包含一个由高层结点和相继低层结点组成的层次结构。 在 B 树索引中有两类结点：

1) 分支结点：简单地指向相应的低层结点。
2) 叶子结点： 存放 B 树方法的实际内容。 即包含指向叶子所对应的行的实际位置。

在 B 树索引中，一个非常重要的变量就是建立在键值基础上的分区索引。分区索引是一种特殊的B树索引。在这种索引中，根据一定范围的键值，表被分解成若干小部分(分区)。利用时间进行分区是常用的方法。

B 树结构的特点是简洁性、易维护性及支持具有高可选择性列值的高速检索。这种方法适合于对索引列值等值查找和范围查找的查询。表的大小，无论它包含数百行还是数百万行记录，对于从其相应表中提取用 B 树索引的数据的速度差别很小，甚至没有影响。

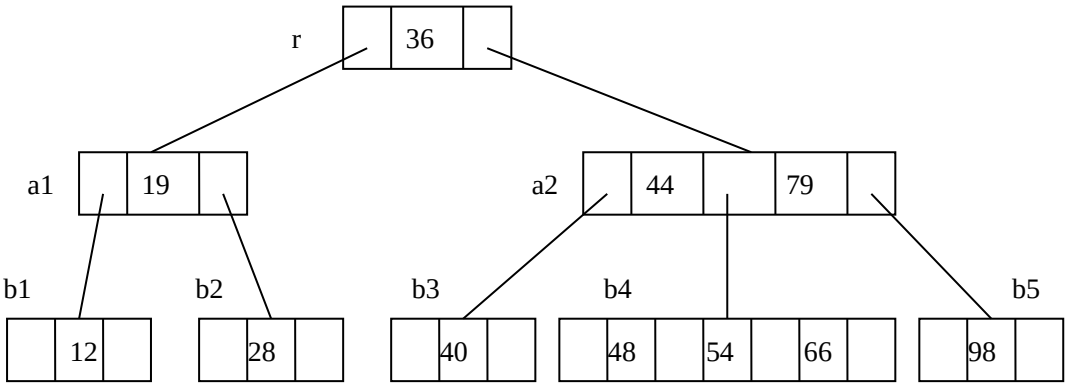
下图是一个 B 树索引的例子，在其中查找行标识 48 的步骤如下：

(1)从根结点 r 开始，因为 45>36，所以根据 36 右侧的指针找到结点 a2；

(2)在结点 a2 中，44>48>79，根据 44 和 79 之间的指针找到结点 b4；

(3)在结点 b4 中存在行标识 48，查找任务结束。

图 3 B-树结构



2.3 连接索引

复杂的查询语句通常需要多表连接，连接索引能够提高多表连接的性能。连接索引包括这样的索引项，它的内容是连接的且满足连接要求的表的元组标志(TID) ，它的每一元组包括所有要连接的表的元组标志(TID)。例如，如果两个关系 R(RID，A)和 S(SID，B) 在属性 A 和 B 上连接，则连接索引记录包含(RID，SID) 对，其中 RID 和 SID 分别来自 R 和 S 的记录标志符。连接索引能够在多表中建立，连接索引能够将所有符合连接条件的连接记录下来。为进一步加快查询处理速度，可以将连接索引和位图索引混合起来使用，即在连接索引中使用 Bitmap 技术，利用 Bitmap 技术的性质，指出实际表中的值。

例如，在一个星型模式中，事实表 Sales 与维表 Customer 和 Item 三者之间的链接关系如图 4 所示。它们的链接索引表如图 5 所示。

对连接索引的进一步改进是位图连接索引。上例中，我们要对顾客的 Gender 为男性的销售建立位图连接索引，这要通过事实表 Sales 与维表 Customer 作连接操作来完成。事实表 Sales 中与 Customer 的 Gender 属性等于 M 的 Customer_id 相对应的元组，位图连接索引的值为 1，否则为 0。如图 6 所示，位向量的长度等于事实表的元组数。

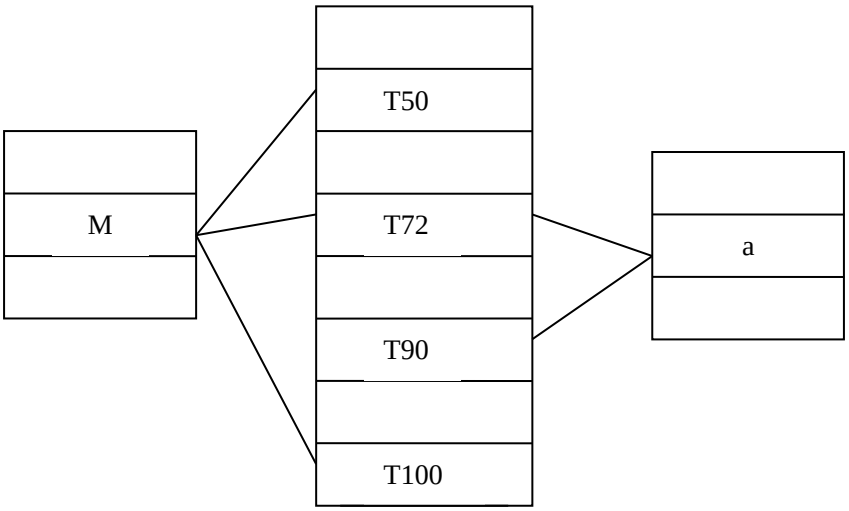


图 4 事实表 Sales 与 Customer 和 Item 表的连接

Customer	Sales_key
...	...
M	T50
M	T72
M	T100
...	...

Customer/Sales
连接索引表

Item	Sales_key
...	...
a	T72
a	T90
...	...

Item/Sales
连接索引表

Customer	Item	Sales_key
...	...	...
M	a	T72
...	...	...

Customer/Item/Sales
连接索引表

图 5 事实表 Sales 与 Customer 和 Item 表连接的连接索引表

F	1 1 0 0 0 1 0 1 1 0 0 0 0 0 1 1 1 1
M	0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0

图 6 事实表 Sales 关于 Gender 属性对应的位图连接索引

2.4 投影索引

投影索引是利用投影的概念把某一个表 T 上的某一属性 A 的值以同样的顺序都保存起来，投影索引的每一行是属性 A 的一个值。值索引中值 X 的次序与 T 中值 X 的次序一致。数据仓库中的查询一般只在表中的部分属性上进行，在这些建立投影索引会大大

提高查询的性能。假设 C 是表 T 的一个列，那么在该列上建立的投影索引就是 C 中的列值按行号排序的存储序列。该索引的存储序列中各值的存储位里以存储页 P 和在该页中的存储位置 S 来表示。如果列 C 的取值长度为 6 个字节，每一个存储页的大小为 3K 字节，则每一个存储页可以存放 500 个列 C 的值。由于列 C 的索引是按行号排序的，只要给出行号 R，就可以根据公式(3.1) 计算出相应列 C 取值的存储页 P 及在该页中的存储位置 S；如果已知列 C 取值相应的存储位里，根据公式(3.2) 可以确定其行号。

(公式 3.1) P=R/500(取整)；S=R%500(取模)

(公式 3.2)R=500*P+S

虽然投影索引类似于一种垂直分割，但表 T 中的记录不一定垂直存储，垂直存储的只是列 C 的索引。而该索引的各项取值是从表 T 中的列 C 中复制过来的，不影响表 T 的存储格式。

首先采用投影索引的是 Sybase，使用的名称是“快速投影索引”。在不仅要检索到所需的记录，而且要确定索引列值的情况下，投影索引是非常有效的。

3 各种索引技术的特点及比较

索引	描述	优点	缺点	商业实现
B 树索引	把存储块组织成一棵树来减少 I／0 操作	• 需要少的 I／O 操作； • 适合于高基数的列； • 能自动数据文件大小相适应的索引层次； • 索引空间的需求独立于被索引列的基数； • 易于创建；	• 低基数的列效果不好； • 不支持即席查询；对宽范围查询 I／O 代价相对较高； • 在获取数据之间索引不能合并；	大多数据的商用数据(0raleRed Brick 等)
简单位图索引	为索引属性的每个取值建立一个位向量，向量长度等于元组数。对应的元组中相应的索引位置 1，若该值出现，否则为 0	• 适合于低基数的列； • 利用位操作； • 获取数据之前可合并索引； • 便于在并行机上执行； • 可以高效地完成对属性的数量型函数(如 count) 的查询； • 易于创建； • 易于插入新的索引值； • 适合于 OLAP；	• 高基数的列效果不好； • 更新索引列代价较高； • 不能很好处理稀疏数据	OracleAscenialSybaseRedBrick DB2
编码位图索引	对属性的域进行二进制编码	• 有效地利用空间； • 可以实现宽的范围查询；	・对于等值查询效率低； •很难找到好的的编码方案； ・当有新的属性值出现，现有的位数不能满足，需重建；	DB2

位图连接索引	索引的创建通过维表对事实表的约束实现的	• 灵活性较好； • 有效地执行； • 支持星型查询；	• 索引列的次序很重要	OracleAscenialRed Brick
投影索引	通过实际存储被索引表的列值建索引	• 当只检索表中的几个列时，加速比较高；	• 只能用于检索原数据；	Sybase

4 总结

为了提高查询速度，最理想的情况是查询能够尽可能使用已有的索引。因此，应该为多数表建立多个索引，特别是要在维表上建立索引。在星型模型中，一个维表可能包含数量较多的记录，并经常以各种不同的方式被使用，而与维表有关的事实表的记录更是数量繁多。因此，如果在维表上有高效的索引，就能够避免对维表和事实表的整体扫描,提高查询处理速度。本文只讨论了索引建立中的一些常见方法和问题，要想为数据仓库建立一个良好的索引系统，还需结合具体的数据情况进行具体分析。

【参考文献】

[1] 彭木根.数据仓库技术与实现[M]. 北京：电子工业出版社，2002.226-230.
[2] 夏红霞，赵杨．数据仓库中的索引技术[J].微机发展，2000，(6) :50.
[3] [美] Joyce Bischoff, Ted Alexander 著，成栋等译，数据仓库技术，电子工业出版社 1998. 6.
[4] 周丽娟，柳池．关于数据仓库若干关键技术的研究[J]．微机发展 2002. (2):29.
[5] 武森，胡波．数据仓库的索引技术．教学与科研，2001，(12)：54～55