一种改进的基于数据库的树存储策略*

汪 建[1],方洪鹰[2],陈昌川[3]
(1. 重庆邮电大学 计算机科学与技术学院,重庆 400065;2. 重庆交通大学 理学院,重庆 400075;
3. 重庆邮电大学 外国语学院,重庆 400065)

摘 要:关系数据库管理系统的优势在于存储以二维数据表为模型的数据结构,而在科学研究领域中,一般树作为重要的数据模型广泛存在。本文讨论的中心问题是如何在数据库管理系统中存放压缩的一般树,并在维护海量数据的同时,降低数据冗余,最后讨论数据一致性的保证和对比分析存储、检索算法的时空复杂度。本文通过对树的压缩存储技术的研究,所产生的结论和方法可以延伸到解决众多非线性数据结构在数据库管理系统中的存储问题。

关键词:数据压缩;一般树;存储;检索;前缀码
中图分类号:TP311.13 文献标识码:A 文章编号:1672-6693(2007)04-0050-04

在计算机科学领域中,数据结构分为线性结构和非线性结构两类。树形数据结构[1](又称一般树)是一种常见且非常重要的非线性结构,其应用非常广泛,是层次模型的典型代表。

目前,线性结构的存储和处理技术比较成熟,而非线性结构因其复杂性和多样性而处理困难。针对一般树(General Tree)的运算没有通用的算法,通常是将其转换为二叉树进行处理,因此二叉树的存储问题一直是学者们研究的重点。

关系数据库是目前海量数据组织处理中最有效的方法,并且它提供了高效的查询服务。但是在关系数据库应用开发中,大部分是处理以二维表为基础的线性结构数据。对于非线性的树形结构,绝大多数关系数据库没作介绍,特别是对树高和度未知的一般树更是没有统一的解决方案和算法。下面将分析树结构在关系数据库存储的一般算法,然后提出一种以前缀码(Prefix Code)为基础的新型树结构压缩存储算法,最后比较两类算法的时间复杂度和空间复杂度。

1 传统的树结构存储算法

1.1 路径表示法[2]

一般树的存储通常是采用记录路径的方式存放层次结构。以文件分配表(FAT)为例,各级目录名和文件名构成了 FAT 结构,如图 1 所示,它是一个典型的树形数据结构。

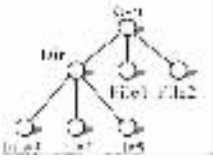

图 1 FAT 结构

将其转化为关系数据库结构,如表 1 所示。其中#NO 字段是数据库自动编号,且是关键字;PathName 字段记录了每一个目录和文件的完整路径。

1.2 双亲表示法[3]

顾名思义,以回溯的方式组织和记录树结构。该方法以二维表作为存储基础,适合关系数据库的组织模式。根据上例可构成双亲表示法的关系表,如表 2 所示。

表中#NO 字段是关键字,代表每一个数据项的编号;NodeName 字段记录了每一个目录或文件的名称;FatherNode 字段标明该节点的父节点在数据库中的编号。

表 1 路径表示法关系表

#NO	PathName
1	Root:\
2	Root:\Dir1\
3	Root:\File1
4	Root:\File2
5	Root:\Dir1\File3
6	Root:\Dir1\File4
7	Root:\Dir1\File5

表 2 双亲表示法关系表

#NO	NodeName	FatherNode
1	Root	-1
2	Dir1	1
3	File1	1
4	File2	1
5	File3	2
6	File4	2
7	File5	2

* 收稿日期:2007-07-16
资助项目:重庆市教育委员会科学技术研究项目(No. 050305)
作者简介:汪建(1978-),男,重庆人,讲师,硕士,研究方向为智能信息处理、数据挖掘。

2 树结构压缩存储法[4-6]

针对同一棵树,“路径表示法”最为简单,处理中间节点和叶子节点的存储问题,方法均相同。但是按照这种处理方式,存储空间浪费严重,且节点间的联系不明确,计算机处理起来较为困难。“双亲表示法”巧妙地删除了“路径表示法”中的冗余的“路径”数据,节省了存储空间。但是由于每个节点与其在关系表中的位置紧密相关,无论是在表中插入、删除数据或对数据进行排序都会改变数据项的位置,从而导致为了维护树结构的正确性付出高昂的开销去调整 FatherNode 字段的内容。

因此怎样解决双亲表示法弊端的问题就变成了怎样使 NodeName 字段与物理位置无关的问题。以下是使用前缀码[7]替代简单的父节点位置的方法。

首先,为了使用前缀编码,必须将一般树结构转化为等价的二叉树[8],具体方法是:建立同层兄弟节点间的联系;保留父节点与第一子节点的联系,删除父节点与其它子节点间的联系,转换算法如图2所示。

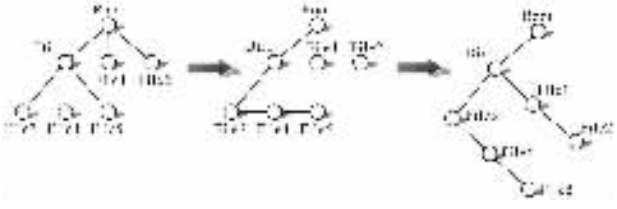

图2 一般树转化为二叉树

然后,采用前缀码编码[9]方式对转换得到的二叉树进行编码:每个节点的左分支编码为0,右分支编码为1,如图3所示。编码后得到树结构压缩存储法关系表,如表3所示。

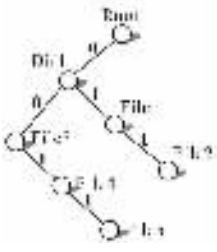

图3 前缀码编码

由于每一个节点的 PrefixCode 字段只跟父节点的 PrefixCode 字段相关,与记录项的物理存储位置无关,不会因为数据的插入、删除和排序操作引发树结构层次混乱,有效地保证数据的一致性。

表3 树结构压缩存储法关系表

#NO	NodeName	PrefixCode
1	Dir1	0
2	File1	01
3	File2	011
4	File3	00
5	File4	001
6	File5	0011

查询任何节点及其子树时,只需要在数据库中按照前缀码左匹配的方式进行搜索即可,例如查询节点‘File3’及其子节点的操作可以使用 SQL 语句:

```
SELECT * FROM general_tree
    WHERE PrefixCode LIKE '00%';
```

以此类推,删除任何节点及其子树时,只需要在数据库中按照前缀码左匹配的方式进行条件删除即可,例如删除节点‘File3’及其子节点的操作可以使用 SQL 语句:

```
DELETE FROM general_tree
    WHERE PrefixCode LIKE '00%';
```

删除子树后得到树结构压缩存储法关系表,如表4所示,树结构如图4所示。

表4 删除子树后的树结构压缩存储法关系表

#NO	NodeName	PrefixCode
1	Dir1	0
2	File1	01
3	File2	011

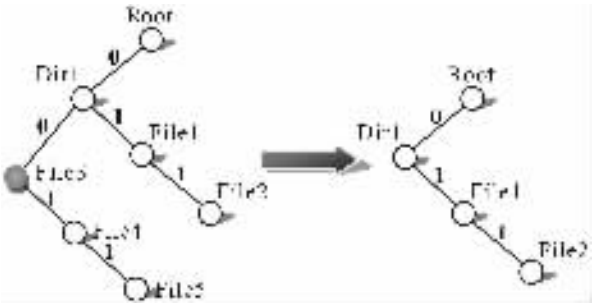

图4 删除子树后的树结构

3 各种存储算法的比较

3.1 空间复杂度

3.1.1 路径表示法 假设系统中目录结构是一棵满 n 叉树、树深为 k 且每个节点宽度为 m,则按照常规的矩阵存储方式:第一层只有1个根节点,占用的绝对存储空间是 m,第二层含有 n 个节点,占用的绝对存储空间是 mn,以此类推第 k 层含有 n^{k-1} 个节点,占用的绝对存储空间是 $n^{k-1}m$。所以整棵树理论上所占用的绝对存储空间是 $\sum_{l=1}^{k} n^{l-1}m$。

事实上以关系模型作为数据组织模式并使用二维表作为存储载体的数据库中要实现变长字段是不可能的,众多的商业数据库系统也没有提供相应的解决方案。因此为了能进行数据存放,必须考虑最“恶劣”的情况。所以路径表示法模型所使用的存储

空间是 $km \cdot \sum_{l=1}^{k} n^{l-1}$。

3.1.2 **树结构压缩存储法** 针对同一种情况,如果使用改进后的树结构压缩存储法模型,存放 NodeName 字段数据消耗的存储空间为 $m \cdot (\sum_{l=1}^{k} n^{l-1} - 1)$,存储前缀编码消耗的空间为 $(k + m - 2) \cdot (\sum_{l=1}^{k} n^{l-1} - 1)$,所以实际消耗的存储空间是 $(k + 2m - 2) \cdot (\sum_{l=1}^{k} n^{l-1} - 1)$。

所以,采用树结构压缩存储法存储同样的一棵树比路径表示法节约空间 $(m - 1) \cdot (k - 2) \cdot (\sum_{l=1}^{k} n^{l-1} - 1) + k \cdot m$,即 $(m - 1) \cdot (k - 2) \cdot (\frac{n^2}{1 - n} - 1) + k \cdot m$。

3.2 **时间复杂度**

3.2.1 **路径表示法** 假设字符串的长度为 L,子串的长度为 l,则在字符串中搜索子串的匹配算法的平均时间复杂度为 $\frac{1 + L - l}{2} \cdot l$。根据上例可知路径表示法的记录项长度为 $\frac{1 + k}{2} \cdot m$,(k 为树的深度)。所以在路径表示法中进行字串匹配的平均时间复杂度为 $\frac{2 + (1 + k)m - 2l}{4} \cdot l$。

3.2.2 **树结构压缩存储法** 同样,假设字符串的长度为 L,子串的长度为 l,则在字符串中搜索子串的匹配算法的平均时间内复杂度为 $\frac{1 + L - l}{2} \cdot l$。因为压缩树结构存储法记录项长度为 m。所以在树结构压缩存储法中进行字串匹配的平均时间复杂度为 $\frac{1 + m - l}{2} \cdot l$。

所以,采用树结构压缩存储法存储同样的一棵树比路径表示法节约的时间为 $\frac{(1 - k) \cdot m \cdot l}{4}$。

4 结论

通过上述分析不难看出,改进之后的树结构压缩存储法克服了路径表示法使用冗余存储方式浪费空间过多的缺点,同时克服了双亲表示法中数据操作困难的缺点。无论是时间复杂度还是空间复杂度都具有相当的优势。该算法可以应用到各种需要在关系数据库中存放树形结构数据的场合,并已在重庆邮电大学的“基于网络的自主学习系统”项目中得到充分验证。

参考文献:

[1] CLIFFORD,SHAFFER A. A Practical Introduction to Data Structures and Algorithm Analysis[M]. 2nd ed. Beijing: Publishing House of Electronic Industry,2004.

[2] 严蔚敏,吴伟民. 数据结构[M]. 北京:清华大学出版社,2002.

[3] 李威,万新光. 树形数据顺序存储映象和链式存储映象转换的方法[J]. 哈尔滨电工学院学报,1995,18(1):100-104.

[4] 廖江东. 关于一类树的优美性[J]. 重庆师范大学学报(自然科学版),2007,24(2):15-18.

[5] 刘晓锋,吴亚娟. 哈夫曼编码的一种基于树型模式匹配的改进型算法[J]. 西华师范大学学报(自然科学版),2006,27(1):21-23.

[6] 郑相周. DBMS 中的树形结构关系数据库[J]. 微型电脑应用,1995(2):76-78.

[7] LIDDELL. M,MOFFAT. A. Hybrid Prefix Codes for Practical Use[J]. Data Compression Conference,2003,12:77-79.

[8] BASSINO F,CLEMENT J,SEROUSSI G,et al. Optimal Prefix Codes for Some Families of Two-dimensional Geometric Distributions[J]. Data Compression Conference,2006,3:113-122.

[9] MILIDIU R L,MELLO C G. Crypto-compression Prefix Coding[J]. Data Compression Conference,2006,3:1-5.

An Advanced Storage Strategy of Tree Based on RDMS

WANG Jian[1], *FANG Hong-ying*[2], *CHEN Chang-chuan*[3]

(1. College of Computer Science and Technology, Chongqing University of Posts and Telecoms, Chongqing 400065;
2. College of Science, Chongqing Jiaotong University, Chongqing 400074;
3. College of Foreign Language, Chongqing University of Posts and Telecoms, Chongqing 400065, China)

Abstract: The superiority of relational database management system is to deal with two-dimensional table, it doesn't support the tree fitly. The general tree is a very representative data structure in the research of science, and is applied to many different domains too.

Maintaining a great quantity of data and reducing its redundancy are emphases of research on data structure. A utility method will be given to compress and store general tree with relational database management system in this paper. Prefix code is commonly used to resolve the problem of frequency related data. It is also available in compression of database. The pivotal aim is to establish the relationship between two-dimensional data table and prefix code. By comparing the result with path expression method and parent express method, its consistency, time complexity and space complexity will be discussed later. The conclusions and methods of this paper can be used to resolve the problems of other nonlinear data structure's storage in database management system based on two-dimensional table.

Key words: data compress; general tree; storage; search; Prefix Code

(责任编辑 游中胜)