

用扩展 RDBMS 实现多媒体文件存储和查询

王 磊, 宋庆平, 刘惠珍, 曹海萍
(北京工业大学 计算机学院, 北京 100022)

摘 要: 阐述了用常用的关系型数据库管理系统来存储图像文件的技术原理, 分析并比较了链表和B树两种数据结构在作为图像数据存储的数据结构时, 对查询响应时间大小的影响, 得出B树的查找快于链表的结论. 结合具体科研任务, 论述了在城市风貌影像数据库的开发中数据表的设计与图像数据解释方面的实现技术, 证实了扩展关系型数据库管理系统在管理图像数据方面是有效的.

关键词: 多媒体数据库; 关系数据库管理系统; B树

中图分类号: TP 312

文献标识码: A

文章编号: 0254-0037(2002)03-0345-04

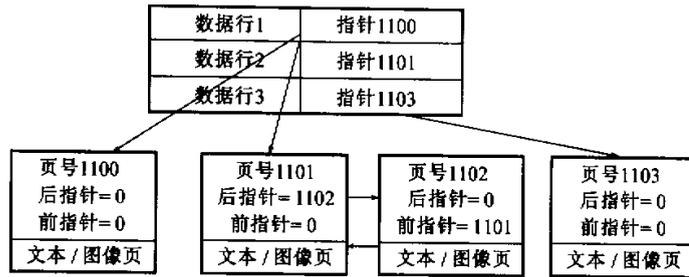
随着计算机多媒体技术的飞速发展以及人们对多媒体文档管理要求的不断提高, 将多媒体文档(如图片、声音、影像文件等)放在数据库中统一管理已势在必行. 多媒体数据有数据量大、数据类型多样化等特点, 多媒体数据库在处理数据对象、数据类型、数据结构等方面都与传统关系型数据库有很大差别^[1], 多媒体文件的特点更适合于面向对象数据库^[2]. 现在面向对象数据库技术的应用还不够成熟, 而且用户往往更喜欢在不改变原有数据库系统的基础上增加对多媒体文档管理的要求, 故用扩展现存的关系数据库管理系统(RDBMS)来支持类似于二进制对象的多媒体文件是一个可行的办法.

1 RDBMS 的扩展

大多数主导的关系数据库为二进制和自由格式的文本采用了通常称为二进制大型对象 BLOB(binary large objects)的数据类型, 作为对关系型数据库不能存储多媒体数据这个问题的临时解决办法^[1]. BLOB是用于图像或其他二进制数据类型的对象. 关系型数据库表包括了 BLOB 的位置信息, 这些信息实际存储在分离的文本/图像页上^[3]. 下面以 MS SQL Server 7.0 为例, 分析它是如何实现 BLOB 存储的.

在 MS SQL Server 7.0 中, 若数据表的某一列的数据类型为 image 或 text, 则该列是一个存放 BLOB 的列. 而实际上, 真正的 BLOB 并不能以二进制的格式存放在该列中, 因为 MS SQL Server 7.0 的最大列宽为 8 060 b^[4], 一个分辨率为 800*600 的 jpeg 格式的图像至少要有 50 kb, 远远超过了列宽的最大限度. 数据类型为 image 或 text 的列中存放的是一个 16 b 的数据, 被定义为 varbinary(16); 它包含了一个指针指向文本/图像页链的首页. 如果需要, 文本/图像数据就会被串起来存储数据, 最大可达 2 Gb^[2]. 每个文本/图像页都包括了一个嵌入式指针, 指向用于存储该列的下一个链接文本/图像页, 如图 1 所示.

在文本/图像页的存储中, 有两种数据结构: 一种是链表结构, 即把所有的图像页用双向链表串起来, 在 MS SQL Server 6.5 中使用; 一种是 B 树结构, 即图像页在逻辑上用 B 树存储和管理, 在 MS SQL Server 7.0 中使用^[4]. 由此可见, 若想查询一条包含多媒体数据的记录, 需两次 I/O 操作: 一次用于查找该记录在数据页中的准确位置, 即查找指针的操作; 一次是根据指针查找图像页的操作. 假设 t 为查询一条包含多媒体数据记录的时间, t_1 为查找该记录在数据页中准确位置的时间, t_2 为查找图像页的时间, 则有 $t = t_1 + t_2$. 对于 t_1 , 与一般的数据记录查找时间没什么区别; 对于 t_2 , 它的大小值得研究.

图1 文本/图像页组织结构^[2]

2 B树与链表结构查找时间的比较

首先, t_2 的大小与管理图像页使用的数据类型有关. 若为链表结构, 链表的查找在最坏的情况下要访问链表中所有的结点, 最好的情况只访问一个结点. 若设其关键字和给定值进行过比较的记录个数的平均值, 即平均查找长度^[5]为 L , 结点数为 n , 则有

$$L = \sum_{i=1}^n P_i C_i = np_1 + (n-1)p_2 + \dots + 2p_{n-1} + p_n = (1/n) \sum_{i=1}^n (n-i+1) = (n+1)/2 \quad (1)$$

其中: P_i 为查找表中第 i 个记录的概率, 在此为 $1/n$; C_i 为找到表中其关键字与给定值相等的第 i 个记录时, 和给定值已进行过比较的关键字数.

B树, 又称为平衡树 (balanced-tree), 是一种平衡的多路查找树, 在文件系统中很有用^[5]. 一个 B树包括: 一个顶层, 称为根; 一个底层, 称为叶子 (通常为第 0 层); 以及 0 层到顶层的多个中间层. 所有的中间层都包含顺序排列的关键字 K_1, K_2, \dots, K_n , 以及指向下一层结点的指针 A_1, A_2, \dots, A_n . A_{i-1} 指针所指的子树的所有关键字都小于 K_i . 一棵 m 阶 B树除根之外每个非终端结点至少有 $m/2$ 棵子树. B树的搜索就是顺指针查找结点和在结点的关键字中进行查找交叉进行的过程. B树最大的特点是平衡性, 即从任一叶结点到根的路径一样长, 这样就避免了搜索效率因记录而异所引起的差异.

根据 B树的定义, 一个 m 阶 B树第 1 层至少有 1 个结点; 第 2 层至少有 2 个结点; 由于除根之外每个非终端结点至少有 $m/2$ 棵子树, 则第 m 层至少有 $2(m/2)^{m-2}$ 个结点. 设 N 为关键字个数, 从根结点到关键字所在结点的路径上涉及的结点数为 l , ($l = m - 1$) 则有

$$N \geq 2 * (m/2)^{m-1} \quad (2)$$

即
$$L \leq \log_{m/2} (N/2) + 1 \quad (3)$$

对于一个数据量为 100 kb 的多媒体数据来说, MS SQL Server 7.0 会为其分配 13 页. 若用链表结构, 根据式 (1), 查找长度 $L = (13 + 1) / 2 = 7$; 若用 B树, 根据式 (2)、(3), 至少要建立一棵四阶的 B树, 则查找长度 $L = \log_{4/2} (13/2) + 1 = 2.7 + 1 = 3.7$. 可见, 采用 B树结构查找图像页时, 查找长度小, 可节省不少时间.

由以上分析可知, 单个多媒体数据的查找时间与记录个数无关, 而与数据本身的大小以及所使用的存储结构有关. 链表的平均查找时间呈线性增长; B树的查找时间呈对数增长. 数据量大时, 对数增长明显缓慢于线性增长, 故采用 B树可减少查找时间. 在使用 B树作为数据存储的数据结构的前提下, 数据量越大, DBMS 为其分配的页越多, 需要建立的 B树的层次越多, 查找时间越长. 数据量超过 100 kb 的数据需要 B树的中间结点, 这些中间结点会占用一些磁盘空间; 在记录数很大的情况下, 这些磁盘空间会给本身已经庞大的多媒体数据页造成负担. 所以, 在记录数适中, 并且多媒体数据在 1 Mb 以内的情况下, 用 RDBMS 管理多媒体数据是可行而且有效的.

3 用 RDBMS 管理某城市风貌影像的多媒体数据

某城市风貌影像查询系统中的风貌影像均来自某城市各个街道、胡同、建筑的原型. 根据需求, 采用

MS SQL Server 7.0 作为存放多媒体数据的数据库。虽然扩展 RDBMS 可以解决多媒体数据的存储和查询问题,但是对于包含文本和图像列的数据表要格外小心,因为当更新包括文本和图像列的数据行时,日志记录信息的数量可能会造成严重的性能问题。任何一次读取数据都至少要有两次 I/O 操作,一次用于读取指针,一次用于读取数据,这势必会影响最后的查询响应时间。为了减缓这些问题,采用了下面的办法:①多媒体数据与元数据分离。元数据是描述多媒体数据的有关信息结构化数据(如某张照片的拍摄时间、拍摄地点等等),需要经常的查询、修改,更新频率比较高;多媒体数据属非结构化数据,不能随时修改、删除,也应避免反复查询,更新频率比较低,故将元数据与多媒体数据分离,可以避免修改元数据时所带来的日志信息的数据增多,同时分离的结构清晰,易于维护。多媒体数据区与元数据区之间用指针相联系,指针为自定义的带有数据表名信息的编号。②查询多媒体数据与查询元数据分离。查询多媒体数据与查询元数据分离是指查询时先查元数据,元数据确定后再查多媒体数据,这样可避免查询多媒体数据与查询元数据所耗费的时间累积,造成查询响应时间过长,同时也可减少客户端接受数据的负担。

在实现 jpeg 照片的显示时,需要把 jpeg 数据从数据库中调出来并显示。这里采用了内存流赋给 jpeg 类的方式,其算法如下:

```
var Ms: TStream;  
    fjpg: TJPEGImage;  
begin  
MS:=phototable.CreateBlobStream(phototable.FieldByName('zhaopian'),bmRead);  
    try  
        fjpg:=TJPEGImage.Create;  
        fjpg.LoadFromStream(MS);  
        Image1.Picture.Assign(fjpg);  
    finally  
        MS.Free;
```

根据以上分析,把多媒体数据存放到 RDBMS 的数据库中,在记录数适中,并且多媒体文件大小适中的情况下,用 RDBMS 管理多媒体数据是可行而且有效的。目前,某城市风貌影像查询系统测试记录达到 5 000 条,均为 100 kb 左右的 jpeg 照片,运行情况良好。

4 结束语

多媒体数据库是数据库发展的新方向,但目前尚没有成熟的面向对象数据库技术,故用 RDBMS 扩展来解决多媒体文件的存储和查询是一个可行的办法。扩展 RDBMS 中对多媒体数据的查询响应时间与多媒体文件本身的大小和所采用的数据结构有关,故用扩展 RDBMS 来解决中小型多媒体数据库问题是有效的。

参考文献:

- [1] PRABHAT K A, THAKRAR K. Multimedia Systems Design[M]. 北京: 电子工业出版社, 1998.
- [2] SUBRAHMANIAN V S, SUSHIL, JAJODIA. Multimedia Database Systems Issues and Research Directions[M]. USA: Springer-Verlag. 1998.
- [3] SOLOMON D, RANKINS R. SQL Server 6.5 开发指南[M]. 熊桂喜, 等译. 北京: 清华大学出版社, 1999.
- [4] SPENIK M, SLEDGE O, FERNANDEZ K, et al. SQL Server 7.0 系统管理指南[M]. 北京: 清华大学出版社, 1999.
- [5] 严蔚敏, 吴伟民. 数据结构[M]. 北京: 清华大学出版社, 1992.

Storage and Retrieval of Multimedia Data by Extended RDBMS

WANG Lei, SONG Qing-ping, LIU Hui-zhen, CAO Hai-ping

(College of Computer Science, Beijing Polytechnic University, Beijing 100022, China)

Abstract: In order to solve the difficult problem of storage and retrieval resulted from complex image with large data, the extended relational database is used to storage and manage the image. Here the technical principle of storing and retrieving multimedia data using extended RDBMS is illustrated. The retrieval time between two data structures, B tree and list as the structure to store multimedia data, is analyzed and compared. The result is that B tree is better than list in retrieval time. Then, the implement technique in the database table design and image data explanation of a city landscape retrieval system is illustrated, and the authors also verified that using extended relational database management system to store and manage image data is effective and practicable.

Key words: multimedia database; relational database management system(RDBMS); B tree

上接第303页

Study on the Planning of On-side Parking

PENG Li-ren, REN Fu-tian

(College of Architecture and Civil Engineering, Beijing Polytechnic University, Beijing 100022, China)

Abstract: The thesis discusses some aspects of on-side parking plan, which include the status and role of on-side parking, the impact factors of on-side parking plan and the principles of on-side parking plan. The on-side parking is an important part of urban parking system and can supplement the off-side parking; planning on-side parking system should take into account the following factors: road features, traffic features, off-side parking facilities and traffic control level and should be done according to the principles of being systematic, coordinative, dynamic and maneuverable.

Key words: on-side parking; off-side parking; traffic control