

一种基于分布式技术的工程数据存储管理策略*

A Memory Management Strategy of Engineering Data Based on Distributed Technology

李陶深 苏一丹 李劲 宋玲 杨柳
Li Taoshen Su Yidan Li Jing Song Ling Yang Liu

(广西大学计算机与信息工程学院 南宁市西乡塘路10号 530004)

(College of Comp. & Info. Eng., Guangxi Univ., 10 Xixiangtanglu, Nanning, Guangxi, 530004, China)

摘要 基于分布式技术,提出一种实现分布式工程数据存储管理功能的策略,该策略旨在增加全局存储器中可用数据库数量,降低磁盘 I/O 的费用,改进系统存储管理的效率。最后,论述了相应存储管理算法的设计。

关键词 分布式技术 工程数据管理 Client/Network 存储管理策略与算法 全局存储器

中图法分类号 TP 317

Abstract Based on distributed technology, a memory management strategy for implementing distributed engineering data management is proposed. This strategy aims to reduce disk I/O by increasing the portion of the database available in global memory, and improves the system's efficiency. Finally, the design of related memory management algorithm is described.

Key words distributed technology, engineering data management, client/network, memory management strategy and algorithm, global memory

实现分布式工程数据管理有多种不同的途径。目前,Client/Network体系结构(简称C/N结构)以其简单灵活、适应性强、能较好地支持机器的物理分布和协同计算等特点,正成为实现分布式工程数据库系统的最佳选择方案。但是,实现一个基于C/N结构的分布式工程数据库管理系统还有许多问题尚待解决,其中一个问题是如何有效地管理由多个服务器和客户机的存储器组成的全局存储器。本文介绍一种基于C/N结构的分布式工程数据存储管理策略,提出一个能减少磁盘I/O的费用,改进系统存储管理效率的存储管理算法。

1 基于C/N结构分布式工程数据管理模型

用C/N结构作为支撑平台,我们构造了一个基于C/N结构的分布式工程数据管理模型(图1)。

在这一结构模式下,工程数据按照产品数据模型,被合理地分布在服务器和客户机上。我们把公用库和项目库等共享信息存放在全局存储器上,把个人设计库的私人信息存放在客户的局部存储器上。当客户机发出数据存取请求时,将通过WWW服务器进

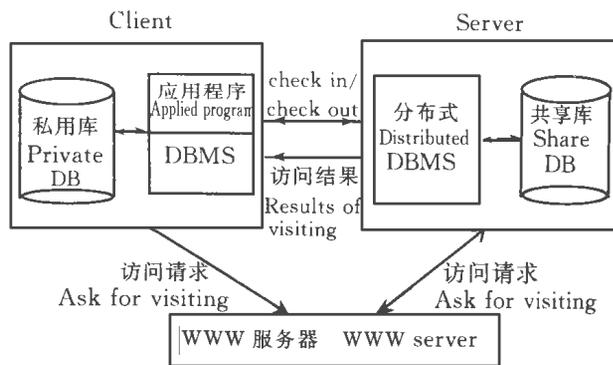


图1 一个基于C/N结构的分布式工程数据管理模型

Fig. 1 A distributed engineering data management model based on C/N architecture

行转发。服务器接到客户发出的请求后,通过检入/检出(Check in/Check out)方式决定是否允许用户将需要经常访问或长期使用的数据库从全局存储器中检出到自己的私人库中。客户机上的用户通过本地管理系统实现对数据的访问与操作。对于被修改的记录数据,检入机构负责把这些数据检入服务器,经服务器进行权限验证之后方可进行数据更新操作。

系统中的数据库服务器实现对工程数据资源的最优化管理,其中包括数据的存储组织、存取方式、数据流向等等,服务器将按照某种策略响应并处理多个客户访问、操作工程数据的并行性请求。WWW服

1998-03-27收稿, 1999-04-27修回。

* 广西自然科学基金项目(桂科自9712008)资助。

务器按照 TCP/IP协议和 Internet工作方式完成客户与服务器之间数据和信息的传送工作,它不仅完成域名解释、路径选择等常规工作,而且接收客户发来的数据请求,在目录中找到所要访问的每个数据页面的驻留位置,并把相关的地址转发给客户。由于多个客户发出的数据存取请求由 WWW 服务器统一处理,使得数据查询、事务操作和数据通信等操作对用户来说都是透明的。

系统还为客户提供一组功能函数,供各客户操作数据使用,包括数据的建立、删除、访问、修改等。客户机上的应用程序可利用这组功能函数,实现对全局存储器上的工程数据进行存取操作。

2 全局存储器的管理策略

在一个基于 C/N结构的分布式 EDBMS中,整个数据库存放在一个由服务器的存储器和客户的存储器组成的全局存储器上。磁盘 I/O是确定系统效率的最主要因素,因此要实现全局存储器的有效管理,必须将访问磁盘的 I/O费用减至最少。这也是系统设计中提高存储管理效率的主要目标。

管理一个全局存储器的最基本管理策略可描述如下:当一个客户需要存取数据库的某个数据页面时,它首先检查自己的存储器是否存有该页面。如果没有找到,则客户向服务器发出数据页面访问请求。服务器接到客户请求后,立即检查该页面是否在自己的存储器中,如果页面在存储器中,则服务器将该页面直接发送给客户;如果不在,服务器首先把客户请求的页面从磁盘读到自己的存储器中,然后再把页面送给客户。这一基本策略虽然可行,但是没有考虑页面的存储管理问题,也没有考虑如何优化存储器,以及解决页面访问的死锁与冲突问题。为此,参照虚拟存储管理的思想,我们把客户和服务器的存储器组成一个统一的、容量更大的全局存储器,并在全局存储器上为当前数据库保持一个较大的存储区,使大多数对数据库的访问都能在全局存储器上完成,以此减少磁盘 I/O的次数。在管理策略上,我们对基本管理策略作了以下修改:

(1) 客户的数据存取请求通过 WWW 服务器进行转发。我们设置了一个全局数据目录,使 WWW 服务器可以快速找出每个数据页面在全局存储器上的驻留位置,或在磁盘上的存储位置。

(2) WWW 服务器接到客户发出的存取数据请求后,立即在全局目录中寻找所访问的数据页面,如果找到,则把页面在全局存储器上的驻留位置转发给客户,并把该页面标志置为“准淘汰”状态,以便在

需要淘汰一个页面时作为优先淘汰的对象。如果该页面不驻留在全局存储器上,则从磁盘上把该页面读到数据库服务器的存储器中,然后把它送给请求该页面的客户。

(3) 如果全局存储器无空间存放所请求的新页面,则必须淘汰一个存储页面。WWW 服务器首先检查被淘汰的页面是否为全局存储器中唯一的副本,如果是,则把该页面存放在数据库服务器的存储器中;如果数据库服务器的存储器没有空间,或者没有标志为“准淘汰”的页面,则 WWW 服务器将寻找能存放该页面的其它服务器或客户。如果找不到任何一个可存放被淘汰页面的客户或服务器,则该页面被送到负责存放该页面最原始副本的服务器中,由该服务器按照最近最少使用(LRU)算法将该页面放在适当的存储位置上。这样,就保证了被淘汰页面至少有一个副本驻留在全局存储器上。

(4) 为了解决页面访问的冲突问题,我们采用这样的策略:各个客户对同一页面的读请求可以同时进行,但是当某个客户正在读取该页面时,任何一个客户就不能对该页面进行写入;当某个客户正在修改该页面时,其它对该页面的写请求或读请求都被拒绝。

以上改进对提高系统性能起到了积极的作用,尤其在数据库的数据页面比较小的情况下效果更为明显。另外,在存储管理中设置了“准淘汰”标志,使全局存储器中的页面替换变得容易实现。

3 全局存储器管理的算法设计

3.1 全局存储器的管理算法

为了实现上述的全局存储器管理策略,我们在设计存储管理算法时,对客户存储管理和服务器的存储管理算法分别予以考虑。

客户存储器的存储管理算法步骤设计如下:

(1) 当一个执行的客户进程需要访问数据库中的一个页面时,首先要根据访问的类型获得访问该页面的一个读锁或写锁。

(2) 如果执行的客户进程已拥有所申请的锁,则转(3);否则,客户向服务器请求获得该锁。

(3) 获得读锁或写锁后,如果要访问的页面就存放在本客户的局部存储器中,则客户进程可直接访问该页面;否则转(4)。

(4) 客户向 WWW 服务器发出访问页面的请求,在发出页面请求之前,客户首先检查其存储器,看看是否有空闲的页面接收一个新的页面。如果没有空闲页面,则按 LRU 算法淘汰一个现有页面。但是在淘汰一个现有页面之前,必须把淘汰该页面的消息发送

给 WWW 服务器, 由 WWW 服务器把该页面的副本保存起来。在得到服务器的确认回答后, 才能实施页面的淘汰工作。

服务器的存储管理基本算法步骤设计如下:

(1) 接收到某客户发出的一个页面请求消息后, 搜索服务器的存储器, 以寻找该页面。

(2) 如果某客户所请求的页面在数据库服务器的存储器中, 则 WWW 服务器直接把页面地址送给客户, 并把该页面标志置为 " 准淘汰 " 状态; 转 (5)。

(3) 如果该页面的副本不在数据库服务器的存储器中, 那么 WWW 服务器搜索全局存储器, 查看该页面是否驻留在其它的服务器或某一客户的局部存储器中。如果找到了驻留该页面的服务器或某客户, 则服务器把页面地址送给请求该页面客户。

(4) 如果请求的页面没有存放在全局存储器上, 那么 WWW 服务器从磁盘上把该页面读到数据库服务器的存储器中, 并把它送给请求该页面的客户。

(5) 如果为了接受新页面而必须淘汰一个现有页面, 则 WWW 服务器检查所淘汰的页面是否是全局存储器中唯一的副本。如果是, 则 WWW 服务器首先检查数据库服务器的存储器是否有空闲的页面空间, 如果有, 则把所要淘汰的页面存入数据库服务器的存储器中; 若无空间, 则 WWW 服务器将寻找一个能提供存放该页面的远程客户, 找到后把要淘汰的页面送到该远程客户中。如果找不到合适的客户, 则用 LRU 算法确定淘汰全局存储器中的某页面, 以腾出位置存放被淘汰的页面。

3.2 页面访问的冲突问题

在存储管理算法中, 为了解决页面访问冲突问题, 算法中考虑了以下内容:

(1) 当一个客户要求修改一个页面时, 它首先向服务器申请一个访问该页面的写锁。如果申请失败, 说明这一修改操作与其它客户的操作发生冲突, 服务器封锁请求修改的客户, 并将等待该页面的消息送到哪些正在对该页面进行读或写操作的客户。当所有对该页面的读锁和写锁都被成功地释放收回时, 写锁才被授予发出修改请求的客户。

(2) 当一个客户请求对某页面进行读访问时, 首先向服务器请求一个读锁。如果此时系统中有若干客

户对该页面进行读, 则服务器允许把读锁授予发出请求的客户。如果有一个客户拥有对该页面的写锁, 则服务器将封锁请求读的客户, 直到写锁被释放后客户才获得该页面的读锁。

3.3 死锁问题的对策

为了检测可能发生的死锁, 我们采用资源等待图来描述各客户进程使用全局存储器上的页面情况。检测死锁时, 首先判别资源等待图是否出现环路, 因为资源等待图的环路是产生死锁的必要条件; 然后通过化简资源等待图的方法来检测此环路是否形成死锁 (有关死锁的判定和检测方法请参考文献 [5])。

出现死锁时, 可按照某种顺序逐个撤消导致死锁的众客户进程, 直至死锁状态消除。被撤消的进程可在以后由原所属的客户重新启动, 并且它仍然可以访问以前要访问的相同数据页。

4 结语

以上讨论的分布式工程数据存储管理策略和相应算法在工程数据库容量较大的情况下特别有效。如果工程数据库规模较小, 或者服务器的存储器容量较大, 使得数据库的大多数页面都可存放在服务器的存储器中, 那么采用基本管理策略就可以了, 没有必要应用特别的存储器管理技术。

参考文献

- 1 李陶深, 张龙祥. 工程数据库中复杂对象的存储管理. 长沙铁道学院学报, 1991, 9 (1): 21~ 28.
- 2 Franklin M J, Carey M J, Livny M. Global memory management in client-server DBMS architectures. Proc of the International Conference on Very Large Data Base Systems, Canada, 1992.
- 3 Turkan Y, Ulusoy O. An efficient memory management algorithm for client-server database management systems. The Computer Journal, 1996, 39 (5): 115~ 128.
- 4 杨 军, 李思昆, 郭 阳. 基于分布式对象技术的 CAD 集成框架的模型与结构. 国防科技大学学报, 1998, 20 (2): 35~ 38.
- 5 王素华. 操作系统教程. 北京: 人民邮电出版社, 1995.
- 6 张龙祥, 李陶深, 张学丽. 微机 CAD 系统实用开发技术. 武汉: 湖北科学技术出版社, 1994.

(责任编辑: 黎贞崇)