

一个高效的面向数据采集的数据库管理系统*

姚卿达 肖永桥 陈晓蕾

(中山大学软件研究所 广州 510275)

摘要 本文讨论一个高效的面向数据采集的数据库管理系统(LNDBMS)的设计与实现. 它采用 CLIENT/SERVER 体系结构, 并支持 MULTI-SERVER. LNDBMS 是开放系统, 用户能通过它透明地访问其它多个异构数据库, 并已在具有多个服务器的大型智力竞赛系统中得到应用, 性能良好.

关键词 数据库管理系统, CLIENT/SERVER, 开放系统, 异构数据库, 分布式数据库系统, 互操作性, 联邦数据库系统.

文献[1]分析了2种体系结构的数据库管理系统. 其中, 以 FOXBASE 为代表的单一式体系结构的 DBMS 结构简单、使用方便、具有较高的效率, 但其安全性较差, 也不具有并发控制、完整性等功能. 另一种以 ORACLE 为代表的 CLIENT/SERVER 数据库管理系统功能强大, 安全性好, 但对资源的要求较高, 以至于其效率在一般微机资源条件下偏低(注: 我们主要测试了 FOXBASE 和 ORACLE 的插入效率, 而没有全面测试). 基于此, 我们开发了一个面向数据采集的数据库管理系统——LNDBMS. LNDBMS 采用 CLIENT/SERVER 体系结构, 并实现 DCL, 这样提高了安全性. LNDBMS 侧重于提高 INSERT 等操作的效率, 测试表明其效率比 ORACLE, FOXBASE 都要快(参见附录), 这样可以满足实时数据处理的要求, 故称之为面向数据采集的 DBMS.

本文第1部分阐述 LNDBMS 的体系结构及环境; 第2部分阐明 LNDBMS 的模块结构; 第3部分介绍了 LNDBMS 实现的主要技术; 而 LNDBMS 作为一个开放系统与其它数据库系统的互操作在第4部分阐述.

1 LNDBMS 的体系结构及环境

1.1 体系结构

LNDBMS 采用 CLIENT/SERVER 体系结构, 并支持 MULTI-SERVER. 其结构如图1所示. 图中, S_1, S_2, \dots, S_m 表示 SERVER, C_1, C_2, \dots, C_n 表示 CLIENT. SERVER 端运行 DBMS 内核, 它接收 CLIENT 发来的请求, 执行此请求, 并把结果返回给 CLIENT.

* 作者姚卿达, 1937年生, 教授, 主要研究领域为数据库, 知识库, DSS, 网络. 肖永桥, 1969年生, 硕士, 主要研究领域为数据库与知识库. 陈晓蕾, 女, 1972年生, 硕士, 主要研究领域为数据库与知识库.

本文通讯联系人: 姚卿达, 广州 510275, 中山大学软件研究所

本文 1995-07-03 收到修改稿

CLIENT 端提供面向程序员的数据库调用接口 (DCI),并在此基础上提供交互式接口 LND * PLUS 及其它实用工具,如 LND * LOADER, LND * UNLOADER, LND * REPORT, LND * MENU 和 LND * FORMS 等.通过

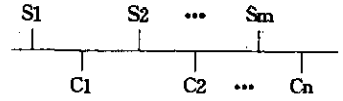


图1

LND * PLUS 可以按 SQL 语法进行建表、插入、更新、删除、检索和列印等数据库操作。

LNDBMS 支持 MULTI-SERVER,即每个 CLIENT 可同时与多个 SERVER 建立连接,但这多个 SERVER 之间是独立的.这样就给用户提供了一个不透明的分布式数据库环境.

1.2 运行环境

LNDBMS 原型系统是在 NOVELL 下采用 NETBIOS 协议开发的,后来转到 DECNET PATHWORKS 的网络环境下开发现在的版本.其 SERVER 运行在 VAX/VMS 环境下,CLIENT 运行在 MS-DOS 下,目前支持 TCP/IP,DECNET,NETBIOS 等网络协议.

在一个大型智力竞赛系统应用中,有中心 SERVER 和前端 SERVER 两级,在中心级 SERVER 上运行 ORACLE 7.0,前端 SERVER 由 2 套 ALPHA 2100 和 2 套 VAX 4300 组成,每个前端 SERVER 可同时为 300 个以上客户服务,整个系统客户数已达 10^3 个,并不断扩展.LNDBMS 在前端 SERVER 和 CLIENT 环境下运作.

2 LNDBMS 的模块结构

LNDBMS 分别运行在 CLIENT 和 SERVER 端,按其功能可分为 3 大块:DCI,COMM 和 KERNAL.其中 DCI 是在 CLIENT 端提供的数据库调用接口,COMM 是连接 CLIENT 和 SERVER 的通信模块,KERNAL 即 DBMS,用于数据库管理.如图 2 所示.



图2

2.1 DCI

DCI 模块主要给用户提供一个数据库调用接口,其函数采用比较直观的方式,并遵循 SQL 语法.目前,应

用程序以 C 语言作主语言,通过 DCI 调用实现对数据库访问,LND * PLUS 及其它工具均建立在此 DCI 上.

2.2 COMM

COMM 主要由 2 部分组成:COMM-C 和 COMM-S,它们分别运行在 CLIENT 端和 SERVER 端.COMM-C 又有 2 个小模块:COMMAND_SEND 和 RESULT_RECEIVE.其中 COMMAND_SEND 用于向 SERVER 传送数据库调用请求,RESULT_RECEIVE 用于接收 SERVER 返回的结果.COMM-S 也有 2 个模块:COMMAND_RECEIVE 和 RESULT_SEND.其中 COMMAND_RECEIVE 用于接收 CLIENT 端发来的请求,RESULT_SEND 用于发送执行后的结果给 CLIENT.

COMM 支持多种网络协议.

2.3 KERNAL

2.3.1 数据库结构

FOXBASE 的一个数据库对应于一个独立的文件,而没有集中的管理.ORACLE 的数

数据库完全集中在 SERVER 端的一个或几个物理文件里. LNDBMS 则采用一种折衷的形式, 它的一个表或索引也对应于一个文件, 但这些文件由 KERNAL 统一管理. 采用这种结构, 不但简化了管理, 而且可以提高效率.

LNDBMS 提供各种系统表格来统一记录和管理用户的表、索引和运行信息. 系统表格主要包括表信息表、域表、索引表、视图表、用户信息表、日志文件等, 表信息表存储用户定义的表的状态信息, 用户的基本数据(即元组)则存储在数据库文件中; 域表用于域(Fields)的管理; 索引表存储各种索引信息; 视图表用于数据子模式和视图管理; 用户信息表用于用户 LOGIN 和 LOGOUT 管理; 日志文件用来记录用户对数据库操作的情况.

2.3.2 进程结构

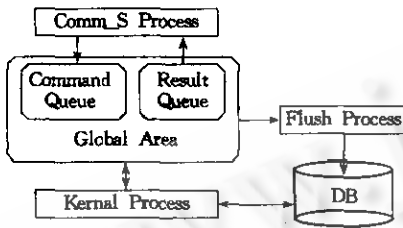


图3

LNDBMS 服务器端有 3 个进程: Comm_S Process, Kernal Process, Flush Process. 其中 Comm_S Process 主要完成与 CLIENT 的通讯, 它把 CLIENT 发来的请求放在 Command Queue, 并从 Result Queue 中取出执行结果传回给 CLIENT. Kernal Process 主要是执行从 CLIENT 发来的请求. 它遵循一定的调度算法, 如 FIFO(事件触发调度), 从 Command Queue

中取出请求, 执行此请求, 然后把结果送到 Result Queue. Flush Process 用于把 Global Area 中的一些更新内容写入外存, 它每隔一定的时间执行一次, 也可以由用户或系统进行调用. 如用户在 LND * PLUS 下可直接发 FLUSH 命令; 用户退出时, 系统会自动地调用一次; 当 Global Area 中的缓冲区满时, 系统也会调用. 见图 3.

3 主要技术

3.1 一致性

LNDBMS 的数据集中存放在 SERVER 端, 而这些数据由多个 Kernal Process 共享存取, 这样若对并发操作不加控制就会存取和存储不正确的数据, 破坏数据库的一致性. 目前通行的并发控制都采用了封锁机制.^[2] LNDBMS 也采用了可靠的封锁机制, 用以防止 LNDBMS 进程之间发生破坏性的不可重复读、读脏数据、丢失修改等问题. LNDBMS 的封锁管理主要是根据 SQL 语句类型、事务的状态、加锁的选择项完成必要的加锁和锁的释放, 所有这些功能都由系统自动完成, 无需用户干预.

LNDBMS 的锁主要有共享锁、独占锁 2 种. 锁的申请和释放信息放在 Global Area 中的封锁队列里. 封锁队列由 3 个队列组成: 活动锁队列、等待锁队列和空闲队列. 其中活动锁队列存放已被批准的锁的信息; 等待锁队列存放等待批准的锁的信息. 封锁管理主要通过 LOCK 和 UNLOCK 2 个模块完成. LNDBMS 也提供了有效的死锁检测机制.

LNDBMS 采用日志文件来记录事务的状态. 事务开始、结束及对数据库的修改操作都登记在日志文件中.^[2,3] 若事务成功地终止, 则该事务对数据库进行的操作保证一致性. 若运行期间发生故障, 如掉电, 导致系统非正常终止, LNDBMS 在重新启动时会检查日志文件, 找出登记在上面的所有非正常终止事务, 并将其强行滚回, 撤消其对数据库的修改. 当发生严重故障如介质故障时, LNDBMS 提供数据库转储的方法, 以把数据库从不一致状态恢复

到某种一致状态. LNDBMS 提供了 UNLOADER 和 LOADER 机制来进行数据库转储. 用户可定期用 UNLOADER 将数据库文件用某种格式(如文本文件)转储在磁盘中, 出现介质故障时, 则可用 LOADER 将存储在磁盘中的数据库后备副本加以恢复.

3.2 安全性

为了保证数据库的安全, 防止不合法的使用所造成的数据泄露、更改或破坏, 我们提供设置用户权限及密码的机制. 任何一个用户要登录 LNDBMS, 查询或修改数据, 都必须在系统的数据字典中标识相应的用户名、密码、权限. 系统管理员 DBA 可对用户授予权限与取消权限. 客户对数据库访问必须通过 LNDBMS 才能进行, 而不能像 FOXPRO 那样应用程序直接打开 DBF 文件并加锁操作, 这不但减少了共享锁开销, 也提高了安全性. LNDBMS 还设计了表属性级的安全性措施, 并在日志中保存了用户操作数据的记录.

3.3 FLUSH 技术

内核在运行中对数据库及系统表格的修改, 有些是在内存中进行的, 这样做的目的是为了高效, 但若出现系统故障, 则会丢失数据. 为了能够把内存的内容保存到磁盘中, 我们实现了一个 FLUSH 机制. 它根据系统内部时钟, 定期地将内存中更新的系统表格、数据库、索引文件保存到磁盘中去; 当每一个与 SERVER 连接的 CLIENT 停止工作并与 SERVER 断开连接(Logout)时, 系统会将内存中的内容通过 FLUSH 保存到磁盘中去; 当 Global Area 的缓冲区满时, 系统也会自动调用 FLUSH 存储内存的内容; 另外用户还可通过 LND * PLUS 自行调用 FLUSH 来保存内存数据, 以减少故障出现时的数据丢失.

3.4 INSERT 技术

LNDBMS 作为一种面向数据采集的 DBMS, INSERT 技术无疑必须高效实现. 为了达到这个目的, 我们采用了缓冲与延迟写技术. 所采集到的数据先放入数据库缓冲区内, 若缓冲区已满, 系统通过自动进行的 FLUSH 技术保存入磁盘. 系统表格的信息会因数据的采集而相应发生改变, 若插入了一条记录, 就将改动的系统表格写入文件中, 这种频繁的磁盘读写会降低效率. LNDBMS 采用延迟写技术, 即当采集的数据量到了一定程度, 才将系统表格写入文件中, 这样就减少了文件读写的次数. 采用这种延迟写技术, 在系统出现故障时可能会导致数据库状态不一致, 这种不一致在系统重新启动时会自动判断并处理.

3.5 INDEX 技术

为了快速地检索数据库, 我们可对之进行索引, 其中包括 B+ 树索引、散列索引、顺序文件索引.

B+ 树是一个平衡的多分树, 检索效率较高.

散列索引技术可对记录进行快速随机检索.

散列索引(HASH)文件存储的是散列表, 其结构为

```
struct {  
    unsigned int offset; 该记录在数据库中的位置  
    unsigned int next; 与该结点发生碰撞的另一结点存储在散列表中的位置  
}
```

进行散列索引时, 数据库必须具有主码, 将该主码通过散列函数求出其在 HASH 文件中的地址. 若该地址已有其它记录的结点存放, 意味着发生了碰撞, 则将该记录结点存放在

HASH 文件末尾,然后通过拉链法,将发生碰撞的结点以链表形式连接起来。

进行检索时,只要将主码通过散列函数求出其在散列表中的位置,如不存在碰撞链表,即可得到该记录在数据库中的位置.如有碰撞链表,则必须将碰撞链表中的每个结点对应的记录进行检索.因此建立散列索引,不宜发生碰撞的次数过多,碰撞链表不宜过长,否则检索的效率将大大降低。

顺序文件索引的原理是将记录顺序存储到数据库中,并将相应的记录号作为数据库主码存储起来,并返回给客户作为字段使用和处理.这样其主码值在数据库中从小到大顺序排列,类似于 FOXPRO 的数据库实现.顺序文件索引的实现很简单,且检索时高效,根据记录号可直接求出记录在数据库中的存储地址。

3.6 查询优化

LNDBMS 把查询分成2类:简单查询和复杂查询.对不同的查询采用下面不同的优化方法:

(1)简单查询.简单查询是指只涉及一个表的查询,即没有连接运算和子查询.对于这类简单查询,LNDBMS 直接在基表上进行,而不需构造查询树,也不需产生临时表.这样系统的开销大大减少,简单查询的效率得到提高.事实上,用户发出的查询有相当一部分是简单查询,这种查询相当于 FOXPRO 的 LOCATE 或 SEEK 命令。

(2)复杂查询.复杂查询是指涉及不少于2个表的查询或嵌套查询.对于复杂查询, LNDBMS 先构造查询树,接着将查询条件(选择运算)和投影运算尽可能下移(相当于求出作用于每个基表的^{最大}查询条件),然后以二目运算为准按后序遍历顺序依次求解各子查询(求解过程中会产生一些临时表,这些临时表与基表一样参加下一步的求解,其上的查询条件由有关的基表得到),最后得到结果临时表,以一次一个记录的方式返回给用户。

4 LNDBMS 与其它数据库系统的互操作性

随着数据库产品日益增多,用户越来越多地面临异构的多数据库环境,这样对异构数据库之间的互操作的要求也就越来越强.文献^[4~8]分别介绍了网关方法(Gateway)、多数据库方法(Multi Database)、分布式数据库方法(Distributed Database)、联邦数据库方法(Federated Database).在文献[9]中,我们分析比较了上述4种方法,并提出了联邦数据库系统 LNFDBS 的一些基本思想. LNDBMS 从设计之初就考虑到了这种互操作要求,这一点恰好弥补了 LNDBMS 本身某些功能的不足。

LNDBMS 与其它数据库系统的互操作有3种方法:①静态方式,主要是数据格式的转换.这是一种间接的方法,其原理是先把其它数据库格式成批地转换为本数据库的格式,然后用本数据库系统操纵它. LNDBMS 提供了2个相应的工具 LND * LOADER 与 LND * UNLOADER.其中 LND * LOADER 是将一定格式的数据转换并装载到 LNDBMS 内核中去,目前能支持.ORA.,.TXT 和.DBF 等格式.LND * UNLOADER 是将内核中的.LND数据卸出并转换成一定的格式,它是 LOADER 的逆操作,支持的数据库类型同 LOADER.②动态方式,即直接通过一个 DBMS 实时地访问另一个 DBMS 所管理的数据库,当数据格式不一样时,还要实时地进行转换. LNDBMS 目前在内核提供了操作 ORACLE 数据库,主要方法是调用 ORACLE 的动态 SQL 语句. ORACLE * CONNECT 就是一个类似的工具.此

方法一般由 DBMS 厂商提供,因此功能受厂商限制。③开放方式,即各个不同的 DBMS 都遵守某种协议来操作其它的数据库。这种方法依赖于厂商较少,具有广阔的前途。目前有 RDA^[10],ODBC^[11]等协议。这种技术将在 LNDBMS 中实现。

5 结 语

本文介绍了一个高效的面向数据采集的数据库管理系统(LNDBMS)的设计与实现,并简单介绍了 LNDBMS 与其它数据库系统的互操作性,有关进一步的技术将另文发表。

参考文献

- 1 肖永桥,姚卿达.单一式和客户/服务器数据库系统:测试及分析.计算机科学,1994,22(5):61~63.
- 2 萨师焯,王珊.数据库系统原理.北京:高等教育出版社,1990.
- 3 周龙骧.数据库管理系统实现技术.武汉:中国地质大学出版社,1990.
- 4 唐雪飞.数据库的互操作性.计算机科学,1993,20(3):25~29.
- 5 Litwin W, Mark L, Roussopoulos N. Interoperability of multiple autonomous database. ACM Computing Surveys, 1990,22(3):10~18.
- 6 Ceri S, Pelafatti G. Distributed databases—principles and system. New York: McGraw—Hill, 1984.
- 7 Heimbigner D, Mcleod D. A federated architecture for information management. ACM Trans. on Inf. Syst., 1985,3(3):253~278.
- 8 Sheth A P, Larson J A. Federated database systems for managing distributed, heterogenous and autonomous databases. ACM Computing Surveys, 1990,22(3):183~236.
- 9 肖永桥,姚卿达.联邦数据库系统的研究.计算机科学,1994,23(6):55~57.
- 10 ISO. International Standard 7489. Open Systems Interconnection Basic Reference Model, 1984.
- 11 Microsoft Inc. . ODBC: Open Database System Connectivity. 1992.

附录:LNDBMS,ORACLE 和 FOXBASE 测试数据(INSERT)

(1)环境:DECnet PATHWORKS V5.0服务器为 VAX 3100,工作站为386微机。

(2)测试事务:INSERT+SELECT(单表),读写记录的长度为140B,测试时间为 10^3 个事务的时间(局部统计的平均时间)。

(3)测试数据

DBMS	时间	秒平均数	备注
ORACLE	50s	20	1000/50=20
FOXBASE	20s	50	1000/20=50
LNDBMS	10s	100	1000/10=100

A DATA COLLECTION ORIENTED DBMS WITH HIGH PERFORMANCE

Yao Qingda Xiao Yongqiao Chen Xiaoheng

(Institute of Computer Software Zhongshan University Guangzhou 510275)

Abstract LNDBMS is a data collection oriented DBMS with high performance, which uses CLIENT/SERVER architecture and supports MULTI-SERVER. It is also an open system, by which users can manipulate transparently multiple heterogeneous databases. LNDBMS has been well applied to a large project with multiple servers. Its design and implementation are described.

Key words Database management system, CLIENT/SERVER architecture, open system, heterogeneous database, distributed database system, interoperability, federated database system.