

# 两种客户/服务器 DBMS 实现及其效率研究\*

冯玉才 金树东 王元珍

(华中理工大学计算机科学与工程系, 武汉 430074)

**摘要** 本文介绍在一个多用户 DBMS 基础上, 实现标准客户/服务器 DBMS 和改进的客户/服务器 DBMS 的基本结构, 其特点是将数据库系统的各层次软件合理地划分到客户机和服务器两边. 实验研究表明了这两种客户/服务器结构以及原有的多用户 DBMS 对于不同类型应用的性能变化.

**关键词** 数据库管理系统, 客户/服务器结构, 功能分布, 吞吐量.

适合客户/服务器(Client-Server)硬件体系结构的数据库管理系统(DBMS)的研究和开发是当前数据库技术发展的重要方面. 已有许多著作研究了多种客户/服务器 DBMS 结构<sup>[1,2]</sup>, 改进和提高这类 DBMS 结构的性能<sup>[3,4]</sup>.

标准 CS 结构和改进的 CS 结构是两种不同的客户/服务器 DBMS 结构<sup>[1]</sup>. 本文介绍在我们研制的多用户 DBMS 基础上实现这两种结构的基本框架, 其特点是将数据库系统的不同软件划分到客户机和服务器两端, 并建立网络上的客户/服务器进程通信<sup>[5]</sup>, 使两端软件合作完成数据库应用. 在标准 CS 结构中, 应用程序在客户机上运行, DBMS 核心软件则在服务器机器上运行. 在改进的 CS 结构中, DBMS 核心的上层软件也在客户机上运行, 客户机上复制数据库的部分数据, 并维护这些数据备份的一致性<sup>[4]</sup>.

我们分别对多用户 DBMS、标准 CS 结构和改进的 CS 结构 DBMS 的性能进行实验研究. 结果表明, 当应用程序分布到网络中时, 客户/服务器 DBMS 结构的性能好于多用户 DBMS. 标准 CS 结构对于更新应用的吞吐量高于改进的 CS 结构, 而后者对于查询应用的性能明显好于前者. 实验结果还表明, 改进的 CS 结构 DBMS 的吞吐量与客户机的数据备份比率有关.

## 1 标准 CS 结构

实现客户/服务器 DBMS 结构的基础是我们设计的基于 UNIX 的多用户 DBMS (见图 1).

在该系统中, 可以有多个应用程序通过共享内存与系统核心进程通信, 由核心进程为应

\* 本文 1993-11-25 收到, 1994-01-24 定稿

作者冯玉才, 1945 年生, 教授, 主要研究领域为数据库, 多媒体技术, 人工智能, 图象识别. 金树东, 1969 年生, 博士生, 主要研究领域为数据库技术. 王元珍, 女, 1945 年生, 副教授, 主要研究领域为数据库管理系统与多媒体技术.

本文通讯联系人: 冯玉才, 武汉 430074, 华中理工大学计算机科学与工程系

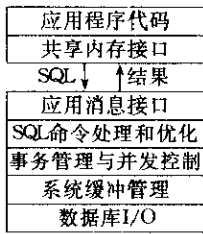


图1 多用户DBMS基本结构

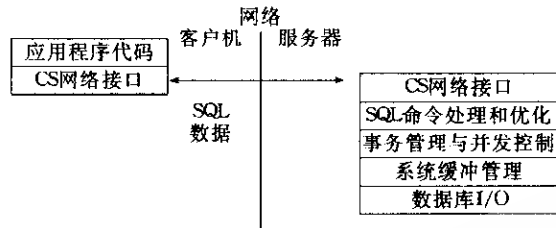


图2 标准CS结构

用程序提供数据或操作. 核心系统有清晰的层次结构,其中消息接口层接收应用发出的操作请求,命令处理和优化层识别应用程序发来的 SQL 命令并处理,可能还要进行一些优化工作;事务管理和并发控制层负责管理用户事务,分配资源,协调多个用户之间的数据共享和互斥存取;系统缓冲区的设置有助于减少硬盘 I/O 次数,提高系统效率,因此要采用有效的方法管理系统缓冲区;I/O 层软件完成数据库的存取操作. 在该系统中采用信号灯 P、V 操作,使核心系统成为消息驱动方式. 一旦应用发出消息,则激发核心系统去获取消息并进行处理,同时还保证了多进程对共享内存的互斥访问.

图 2 给出在多用户 DBMS 基础上实现标准 CS 结构 DBMS 的基本结构,其实现关键是在应用程序和核心系统间建立网络上的客户/服务器通信接口,取代原有的共享内存接口,而应用程序代码和 DBMS 核心系统的其它各层次模块不必改动. 应用程序编译生成时,与网络接口库链接在一起. 这样生成的应用程序运行时,可以调用这些接口库例程,向服务器机器上的数据库服务进程登录,并在网络上传输 SQL 命令和数据.

在这一结构中,客户机上的应用程序将 SQL 命令组装成消息发送给服务器,驱动 DBMS 核心进程的操作,这类似于远程过程调用(RPC)方式. 一旦服务器进程接收到客户消息,便调用相应的系统定义过程处理该 SQL 命令,并将结果反馈给客户应用程序. 在具体实现中采用 TCP/IP 协议的套接字接口.

服务器进程采用网络 I/O 多路复用技术,接收从多个相连客户发来的信息. 任何一个网络 I/O 事件发生时,驱动服务器进程获得命令. 若系统中有多个客户请求服务,则服务器进程可采用模拟多线索的执行方式. 当用户线索需要数据库存取时,将请求交给 I/O 软件,而服务器切换到其它用户线索运行. 而当数据库 I/O 完成时,再唤醒等待数据的用户线索.

## 2 改进的 CS 结构

### 2.1 基本结构

实现改进的 CS 结构的基本方法是将原有多用户 DBMS 核心系统的各层次软件拆成两部分,并加入合适的协调机制和网络接口,构造成客户机 DBMS 和服务器 DBMS 软件,使之形成一个完整的 DBMS. 图 3 给出了改进的 CS 结构的基本结构.

在这一结构中,应用程序、应用程序接口与多用户系统相同,它们在客户机上运行. 客户机 DBMS 包含多用户 DBMS 的上层软件,而且它可以通过远程资源请求接口与服务器 DBMS 实现信息交换. 服务器 DBMS 包含原 DBMS 的低层软件,如并发控制、数据库存取模块,它通过网络接口获得从客户机发来的消息并为之服务. 客户机 DBMS 和服务器

DBMS 之间传输数据、锁信息以及一些控制消息。

改进的 CS 结构的特点是在可以在客户机上存储数据库的部分数据备份。采用这一策略的好处是可充分利用客户机资源,达到客户机和服务器的磁盘 I/O 并行。同时,还要求采用客户数据缓冲技术有效管理这些数据备份,并维护其一致性。

## 2.2 功能划分

采用改进的 CS 结构 DBMS 的动机在于充分利用客户机资源。在标准的 CS 结构中,当系统中应用增加时,服务器负担过重,往往客户机的性能也相当可观,通过将 DBMS 上层软件划分到客户机端有助于使系统负载得到更合理的分配。

这里要考虑到客户/服务器功能划分界线的选择,该问题与系统资源状况和应用特性有关。一般地,DBMS 的上层软件被划分到客户机端,应用程序接口、用户命令的预处理和一些优化工作应当在客户机上完成。若客户机上有相当的内存资源,则提供客户端数据缓冲能减少对服务器数据的请求。更进一步地,如果客户机的硬盘资源可用,并且维护这些数据的代价不大,则设置客户数据备份能够得益不少。

在我们实现的改进的 CS 结构中,客户/服务器功能划分偏重于在客户机上完成相当多的工作。客户机上不仅运行应用程序,还承担了原有 DBMS 核心的上层软件功能。由于在客户机的磁盘上保存了部分数据,因而这些数据的存取也成为客户机的负担。客户机端的各部分软件设计特点如下:(1) 应用程序、应用通信接口和客户机 DBMS 的命令处理层与多用户 DBMS 的相应部分相同;(2) 引入客户机 DBMS 的缓冲管理层,当上层模块需要存取数据时,调用本层软件中的过程;(3) 客户机 DBMS 的数据备份 I/O 软件以页为单位存储数据,采用索引技术提高页的查找速度;(4) 远程资源请求接口是客户机 DBMS 向服务器 DBMS 请求资源、交换信息的必经之路,数据传输单位为页,这与缓冲管理和页级锁统一起来,简化了设计,同时锁信息与数据请求结合在一起,能减少网络费用。

服务器 DBMS 主要完成各用户事务间的并发控制、服务器缓冲管理和数据库存取等基本功能。与标准 CS 结构相比,服务器 DBMS 的负担要轻得多,因而它能为更多客户机提供服务。服务器 DBMS 的各模块设计特点如下:(1) 远程请求响应模块根据收到消息的不同类型,分别调用相应处理过程分派执行,并将结果返回给客户机 DBMS,在自定义的消息结构中包含远程请求者标识、消息类型、请求的数据页编码和锁类型、传输的数据等信息;(2) 事务管理和并发控制层为各客户机上的应用事务提供统一管理,分配资源,检测并消除死锁;(3) 服务器缓冲管理和数据库存取软件与多用户 DBMS 的相应部分相同。

## 2.3 数据一致性

在客户机上存储数据备份有助于提高系统效率,然而当客户应用程序更新数据并提交事务时,会使得其它客户机上的数据备份“过期”。因此在实现的改进型 CS 结构中,要在客户机 DBMS 和服务器 DBMS 之间建立某种协议,保证客户机上数据备份与服务器数据库的

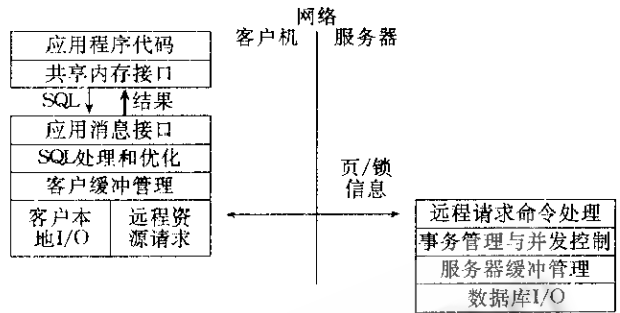


图3 改进的CS结构

一致性.

解决这个问题的方法是将客户数据备份的一致性维护与远程资源请求结合在一起. 服务器 DBMS 为每个客户机 DBMS 保存一个数据备份信息表, 它知道客户机存储哪些页, 以及这些页是否为当前版本. 当某客户机上的应用程序更新数据库的页并提交事务后, 服务器 DBMS 将其它客户机数据备份中的相应页标记为“过期”. 当客户机 DBMS 请求资源时, 服务器 DBMS 将该客户机上的数据过期信息返回.

### 3 效率研究

#### 3.1 实验系统参数

实验系统中物理资源和实验数据库的主要参数如下: 服务器 CPU 速度—80486/33MHz, 服务器硬盘寻道时间—12ms, 所有客户机的平均 CPU 速度—80386/33MHz, 所有客户机的平均硬盘寻道时间—12ms, 网络带宽—10Mbps, 数据库的大小—4MB, 页大小—1KB, 数据库系统缓冲区大小—400 页, 更新一个元组所花费的平均 CPU 时间—2ms, 查询一个元组所花费的平均 CPU 时间—1ms.

#### 3.2 衡量标准

我们采用整个数据库系统的吞吐量作为比较不同结构 DBMS 性能的标准. 为此设计了一组应用程序, 每个应用程序是 100 个事务的流. 有两种类型的应用程序: 纯查询应用和混合型应用, 在纯查询事务中, 100 个事务全是查询事务, 而混合型应用有 5% 的更新事务. 每个查询事务完成一次小的查询, 只涉及到 2—3 个页. 更新事务则提交一个小的更新, 要查找到元组, 作修改后再写回硬盘, 一般要更新 4—6 页. 查询与更新事务费用的比在 1:5 到 1:10 之间.

针对这两种不同应用, 实验研究多用户 DBMS、标准 CS 结构和改进的 CS 结构 DBMS 性能并加以比较. 在这些系统中, 同时运行  $N$  个应用程序, 从其重叠的时间段上取样, 统计系统完成的事务个数, 并求得每秒钟完成的事务数 (TPS) 作为系统的吞吐量. 对于  $N$  的不同取值, 分别得到系统吞吐量, 然后考察不同系统吞吐量与同时运行的客户应用间的关系.

#### 3.3 多用户 DBMS 性能

将多用户 DBMS 核心软件 and 应用程序在同一机器 (服务器) 上运行, 由 UNIX 系统对其进行分时调度, 实验结果如图 4 所示.

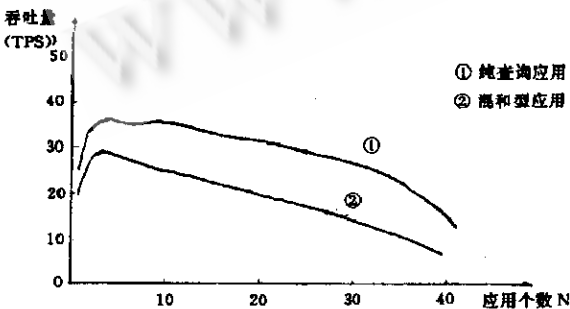


图 4 多用户 DBMS 的性能

对于纯查询应用, 当只有一个应用运行时, 吞吐量为 25.0TPS. 当应用数  $N$  较小时,  $N$  增大使得吞吐量也有增加, 并在  $N=4$  时达到 35.7TPS. 此后的一个区间上, 吞吐量稳定在 35TPS 左右. 随着  $N$  继续增大, 多用户系统性能开始下降, 特别是当  $N > 35$  后, UNIX 超负荷使 DBMS 性能明显下降.

多用户 DBMS 对于混合型应用的

吞吐量变化情况相似.但其值为纯查询应用的 70—80%,这是因为更新的费用高达查询的数倍.

### 3.4 标准 CS 结构性能

在标准 CS 结构的实验中, DBMS 核心软件在服务器上运行, 应用程序分布在许多客户机上. 实验结果如图 5 所示.

对于纯查询应用, 客户应用较少时, 系统性能比多用户 DBMS 差,  $N=1$  时吞吐量值为 15.5TPS. 这是因为网络上的标准 CS 结构比单机上的多用户 DBMS 松散一些. 随着并行的客户应用增加, 吞吐量增加较快, 当  $N=7$  时达到 52.3TPS. 此后系统达到饱和状态, 吞吐量一直保持在 50TPS 左右. 与多用户 DBMS 不同, 在标准 CS 结构中当客户应用较多时, DBMS 核心系统才达到饱和, 并且应用的增加不明显降低系统吞吐量.

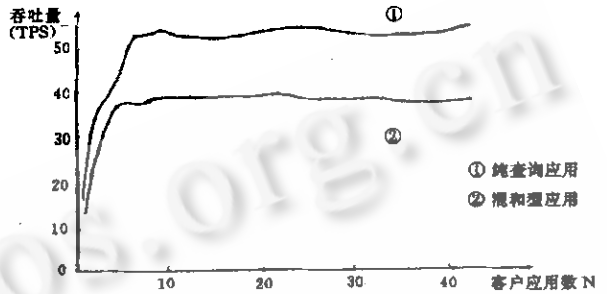


图 5 标准 CS 结构性能

标准 CS 结构对于混和型应用的性能变化曲线与纯查询应用类似. 客户应用极少时, 性能坏于多用户 DBMS, 随着  $N$  值的增大, 其吞吐量超过 30TPS 并一直保持在较高的水平.

### 3.5 改进的 CS 结构性能

我们在服务器上运行服务器 DBMS, 在各客户机上运行客户机 DBMS 和应用程序. 实验考虑了客户机数据备份大小不同的两种情形, 分别为 400 页和 800 页, 即客户机数据备份比率分别为 0.1 和 0.2. 但客户机 DBMS 的缓冲区大小固定为 100 页. 图 6 显示对于纯查询应用和混和型应用, 采用不同的客户数据备份比率时, 改进型 CS 结构的性能变化情况.

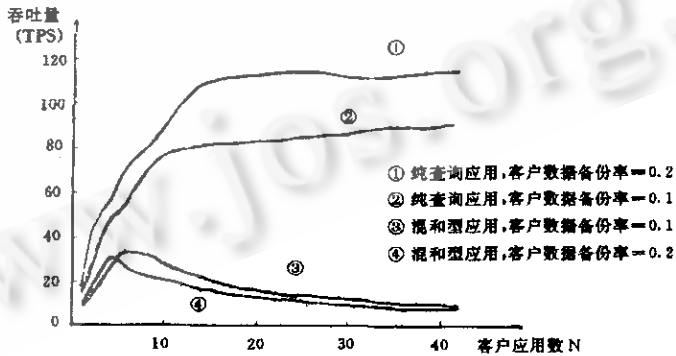


图 6 改进的 CS 结构性能

首先来看系统对于纯查询应用和 0.1 的客户数据备份率这种情形. 系统吞吐量从最初的 12.5TPS 迅速增加到  $N=10$  时的 70.0TPS, 此后服务器接近饱和, 吞吐量缓慢增长, 并在客户应用数达到 40 时, 超过 90TPS. 增大客户机的数据备份比率, 纯查询应用的吞吐量增长更快, 并且当  $N=15$  时吞吐量达到 108TPS, 此后增长才变缓, 到  $N=40$  时其值接近 120TPS.

改进的 CS 结构对混合型应用的吞吐量变化与纯查询应用有较大的不同. 当客户应用个数很小( $N < 5$ )时, 系统吞吐量有较高的增长率, 一度超过 30TPS. 但此后不再增加, 特别是当  $N > 10$  时, 出现吞吐量的下降趋势. 这是因为频繁的更新造成客户数据一致性维护负担加重. 还可以看到, 增大客户机的数据备份率后, 由于数据一致性维护花费加重, 而使得系统性能下降更明显.

## 4 结 论

本文描述了一个多用户 DBMS 基础上实现两种客户/服务器结构框架. 从功能分布方面来看, 标准 CS 结构只将应用程序分布到网络中的客户机上, 而改进的 CS 结构中客户机上可存储部分数据, 承担更大比重的功能. 从系统设计的角度来看, 将多用户 DBMS 转化为标准 CS 结构比较简单, 而改进的 CS 结构设计复杂得多.

实验结果表明, 客户/服务器功能分布使得系统中应用增加时, 性能更好. 结构紧密的多用户 DBMS 适用于应用极少的环境. 对两种 CS 结构的比较表明, 标准 CS 结构对于更新应用效果较好, 而系统中同时运行的查询应用较多时, 改进的 CS 结构的性能明显更优. 实验研究还表明, 在改进的 CS 结构中, 随着客户数据备份比率的增大, 系统对查询应用的吞吐量提高, 而对更新应用正好相反.

## 参 考 文 献

- 1 Roussopoulos N, Delis A. Modern Client—Server DBMS Architectures. ACM SIGMOD Record, September 1991.
- 2 Dewitt D, Maier D, Futersack P *et al.* A study of three alternative workstation—server architectures for object—oriented database systems. Proc. of 16th Int. Conf. on VLDB, 1990.
- 3 Franklin M, Carey M. Global memory management in client—server DBMS architectures. Proc. of 18th Int. Conf. on VLDB, 1992.
- 4 Carey M, Franklin M, Livny M *et al.* Data caching tradeoffs in client—server DBMS architecture. ACM—SIGMOD Conf. on Management of Data, May 1991.
- 5 AT&T. UNIX 网络程序员指南. 北京: 电子工业出版社, 1992.

# THE IMPLEMENTATION AND PERFORMANCE STUDY OF TWO CLIENT—SERVER DBMS ARCHITECTURES

Feng Yucai Jin Shudong Wang Yuanzhen

(Department of Computer Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract** This paper presents the implementation frameworks of two client—server DBMS architectures, which are based on our multiuser DBMS. In these two architectures, the layers of database system software are separated to client side and server side reasonably. Experiment results are shown to indicate the performance aspect of the client—server architectures and original multiuser DBMS to different kinds of application.

**Key words** Database management system, client—server architecture, functional distribution, throughout.