

# 多数据库事务处理模型中局部代理的设计与实现

卢正鼎 李瑞轩 肖卫军

(华中理工大学计算机科学与技术学院 武汉 430074)

## 【摘要】

在多数据库事务处理模型中，局部代理提供了全局事务管理层与局部数据库系统的接口，它使多数据库系统能够对分布在不同站点的局部数据库进行透明访问。

本文提出了基于两阶段代理（2PCA）的局部代理模型，它集成了支持两阶段提交或单阶段提交的数据系统、以及并不支持事务提交协议的文件系统。本文还介绍了采用专用接口和 ODBC 两种实现局部代理的方法。文中着重介绍了使用 ODBC 方法的具体实现，并对两种方法实现的局部代理作了性能上的比较。

## 【关键词】

多数据库系统 局部代理 两阶段代理 事务处理 开放数据库互连

## THE DESIGN AND IMPLEMENTATION OF THE LOCAL AGENT IN MULTIDATABASE TRANSACTION MODEL

Lu Zhengding Li Ruixuan Xiao Weijun

(College of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan 430074, China)

## ABSTRACT

In multidatabase transaction model, local agent is the interface between the global transaction manager and the local database systems. It provides transparently accessing local database distributed at different sites for multidatabase systems.

This paper presents the local agent model based on two\_phase commit agent(2PCA). The model integrated database systems that two\_phase commit protocol or one\_phase commit protocol is supported and file system without any transaction processing protocol. Two methods, using special interface and using ODBC, which implement the local agent are also introduced. The concrete implementation that using ODBC is specially addressed in this thesis. The performance comparison between the two methods is also discussed.

## KEYWORD

Multidatabase Systems (MDBS) Local Agent Two\_Phase Commit Agent (2PCA) Transaction Processing Open Database Connectivity (ODBC)

---

收稿日期：1997-06-27；修回日期：1998-01-；本课题得到国防预研基金项目资助。卢正鼎，教授，博士生导师，主要研究方向为计算机辅助技术和系统集成技术。李瑞轩，硕士，主要从事多数据库系统研究工作。肖卫军，硕士，主要从事多数据库系统研究工作。

## 1. 引言

数据库技术的发展导致了众多数据库系统的出现。这些数据库可能在物理上和逻辑上都存在着差异，如数据模型的不同。因此，研制能集成现在已有的数据库系统以及文件系统的多数据库系统成为数据库发展的必然趋势。而网络技术的发展为用户访问和处理多数据库中的数据提供了可能，它使得多个异构的数据库之间可以共享它们相互依赖的数据，并具有相互操作的能力。多数据库系统屏蔽了现在已有的各数据库系统不同的访问方法和用户界面，给用户呈现一个访问多种数据库的公共接口，从而减少各数据库之间的差异。

事务处理是多数据库系统中需要解决的关键问题之一。在多数据库事务处理模型中，我们既要保证多数据库事务的全局原子性和数据一致性，而且还要最大限度地保持各个局部数据库系统的自治性。基于局部自治性问题的考虑，一般采用弱化传统事务的 ACID（原子性、一致性、隔离性和持久性）特性的方法来实现多数据库事务处理模型。

局部代理是多数据库事务处理模型中全局事务管理层（也称全局事务处理核心）与局部数据库系统的接口。它主要包括局部事务管理，全局子事务执行状态管理，局部日志的管理及子事务结果处理。本文针对局部代理提出了局部代理模型，并介绍了两阶段提交代理（2PCA，简称两阶段代理）的方法，以解决多数据库系统中各局部数据库系统在数据模型、事务提交协议和并发控制方面的差异。同时，本文还着重介绍了使用 ODBC 实现局部代理的方法，并和采用专用接口的方法作了性能方面的比较。

## 2. 局部代理模型

### 2.1 局部代理在多数据库系统中的位置

多数据库系统（MDBS）是多个现存的、自治的、异构的数据库系统的联合。它在所有局部数据库系统（LDBS）之上构成全局系统管理层，提供外部用户接口，使用户能实现对异种数据库的透明访问。MDBS 屏蔽了不同数据库在物理上和逻辑上的差异，各 LDBS 有充分的自治性。多数据库事务处理模型的体系结构如图（1）所示。

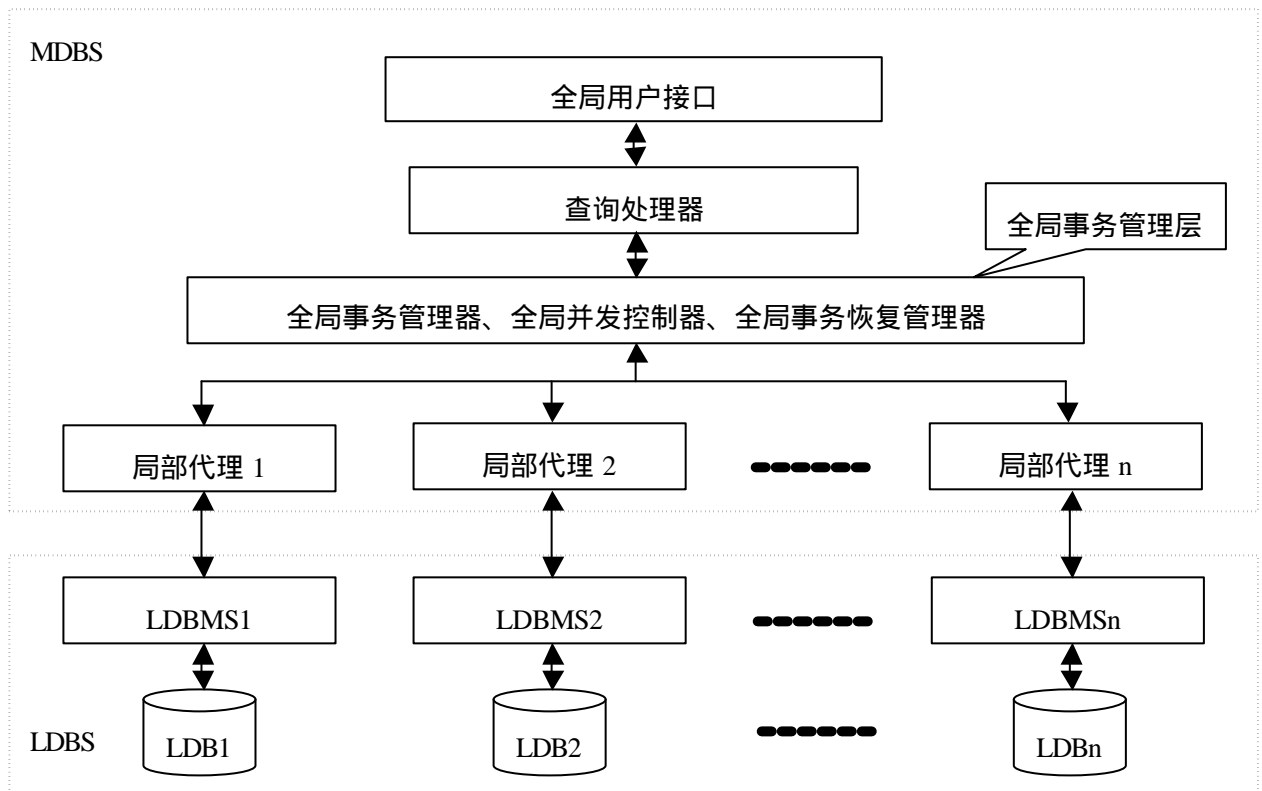
局部代理作为全局事务管理层与 LDBS 的接口，在多数据库事务处理模型中扮演着非常重要的角色。在 MDBS 中，所有对数据库的实际操作都是由 LDBS 完成的。从 LDBS 的角度看，局部代理仅仅是 LDBS 的一个应用。局部代理从全局事务管理层那里接收子事务操作，然后向 LDBS 提交，再将子事务处理的结果返回给全局事务管理层。全局事务管理层根据各个局部代理的执行情况来决定全局事务的提交与否，然后将全局决定通知各个局部代理。在事务提交协议中，每个局部代理作为一个重要的参与者，它必须对全局事务管理层负责。

### 2.2 局部代理模型

通常，多个不同的数据库组成一个异构或异质 MDBS。从广义上来说，MDBS 的单元子系统可以是关系数据库系统、网状数据库系统，也可以是层次数据库系统，甚至还可以是文件系统。不同的数据库系统支持不同的事务提交协议，有的并不支持事务处理。例如 Oracle、Sybase、SQLServer、Informix 等支持两阶段提交（2PC）协议，dBase、Foxbase、Foxpro、Access 等支持单阶段提交协议，而文件系统则不支持事务

处理。这使多数据库事务处理变得异常复杂。

这里提出一种适合多数据库环境的局部代理模型。局部代理模型是 MDBS 中全局事务管理层与多个异构或异质的 LDBS 之间接口的实现模式的总和。在局部代理模型中，一个局部代理关联一个 LDBS，它负责与该 LDBS 的接口。一个局部代理由一个 2PCA 方法来实现。而每个 LDBS 由于其数据模型及其它一些内部结构和实现机制的不同，它们有各自不同的事务提交协议。2PCA 方法针对不同的事务提交协议有不同的实现途径。这就构成了局部代理的结构框架。



图(1) 多数据库事务处理模型的体系结构

在 MDBS 中，事务作为它的基本执行单位，一般须支持两阶段提交协议。局部代理模型就是要对那些并不支持两阶段提交协议的 LDBS 模拟准备状态，甚至是准备和提交两个状态，使 MDBS 中的全局事务管理层所处理的事务均能以两阶段提交协议来完成。这里我们采用 2PCA 方法实现局部代理模型，它主要解决 MDBS 中事务提交协议不一致的问题。下面分别介绍单阶段提交数据库系统和文件系统上的 2PCA 方法。

#### (1) 单阶段提交数据库系统

在支持单阶段提交协议的数据库系统中，事务一旦执行成功就立即提交；当执行失败时，就立即回退掉。它不具备两阶段提交协议中的准备提交状态。2PCA 所要实现的就是要模拟一个准备状态，使 MDBS 全局事务管理层所看到的事务在各 LDBS 上的处理是无差异的。

在单阶段提交数据库系统的 2PCA 方法中，局部代理设置一个局部日志，它记录一个事务（需要指出，局部代理中所执行的事务一般是指全局子事务）所执行的各个步骤。如果事务执行失败，情况比较简单，只需向全局层汇报事务执行失败即可。如果事务执行成功，根据单阶段提交协议，它会立即局部提交。此时，局部代理向全局事务管理层发送该 LDBS 上的子事务执行成功的信息。最后，若全局管理层发布全局提交命令，则局部代理不用采取任何进一步的动作；若为全局回退命令，那么局部代理应根据日志中记录的信息

产生一个补偿事务让局部数据库去执行，以恢复数据库原来的状态。这里存在提交过的子事务的结果会显露给其它事务的问题。若最后的决定为全局提交，则不存在任何问题；但是一旦决定全局回退，则会发生意想不到的后果。在这里我们假设一个事务在全局提交或全局回退之前，不存在其它事务读取该事务所处理过的数据的情况。

## (2) 文件系统

由于文件系统不支持事务处理，所以局部代理模型中需要为其模拟准备和提交两个状态。事务一般具有 ACID 特性，下面主要从原子性和隔离性两方面来介绍文件系统的两阶段代理。

事务的原子性是指事务的操作系列要么全部正确地完成，要么全不被执行。文件系统并不具有这样的特性，可能它的一部分读写文件操作成功，而另一部分读写文件操作失败。成功的操作会直接写到物理存储上去，而不管有没有失败出现。为了保证模拟的文件系统上的事务支持原子性，在此采用备份文件的方法。如果所有对该文件的操作成功，则释放备份文件空间；一旦有文件操作失败，则使用备份文件来覆盖原有文件，从而保证其原子性。相应地，文件系统上的模拟事务的一致性也得到了保证。

隔离性是指某个未完成的事务在提交之前不能对其它事务显露其中间结果。在文件系统中，由于模拟事务的一个操作完成之后，其它事务即可看到修改后的该文件，所以文件系统并不具有隔离性。在这里，主要采用锁的机制来实现其隔离性。当一个事务对某个文件开始操作之前，即给该文件上锁，不允许其它事务再读写该文件；当该事务提交或回退之后，即释放该文件上的锁，以让其它事务对该文件进行操作。文件系统上的模拟事务的持久性是显然的。

通过上面的方法，我们已在文件系统中模拟了支持单阶段提交协议的事务，利用本文 2.2(1) 中所介绍的方法可以进一步实现文件系统上的两阶段代理。

## 3. 局部代理的设计与实现

### 3.1 采用专用接口实现局部代理

目前我们所面对的几乎都是商品化的 DBMS，一般而言，这些 DBMS 的内部结构并不可知，且不允许改动，要为每个 LDBS 设计一个专用接口，我们只能根据这些 DBMS 的用户资料来构造局部代理。

在多数据库事务处理模型中，局部代理要求以各 LDBS 支持的数据库语言的子事务语句序列为输入，以子事务各语句的处理结果为输出，对于查询语句，需要返回关于该查询结果关系的模式描述以及全部元组数据。根据我们对 Oracle、Sybase 和 Informix 进行的分析和我们自行研制的实验系统 MDBT\_1 中的验证实践，利用它们的宿主语言如 Oracle 的 Pro\*c、Informix 的 ESQL/C 等的动态机制和带参机制，可以实现满足上述要求的 Oracle 局部代理、Sybase 局部代理和 Informix 局部代理。此外，文件系统、dBase、Foxbase 及 Foxpro 等的数据库文件格式对用户是开放的，可以直接对其操作，从而与数据库接口，它们的局部代理的实现则相对较为容易。在局部代理与 LDBS 的通讯方面，目前的网络技术已提供了足够的支持，如具有网络双向通讯的 socket 为局部代理的实现提供了技术支持。

### 3.2 使用 ODBC 实现局部代理

ODBC 定义了访问多个后端数据源（它们是 LDBS 的实例）的统一界面标准，它独立于 LDBS，为开发者提供了一套命令集用于与远程（或本地）数据库建立连接，并向其发送 SQL 语句。ODBC 的目的是要

使开发者可不依赖于 DBMS 厂商的应用工具，按标准的方法开发自己的访问后端数据库的应用程序，从而代替现有的一些数据库前端应用软件。

ODBC 的体系结构如图 (2) 所示。应用程序调用库函数来实现对 SQL 语句的语法检查以及数据类型的检查，将检查过的结果移交驱动程序管理器；驱动程序管理器确定相应目标数据库的驱动器，并执行初始化；然后由驱动器把目标数据库的逻辑名映射成数据库的实际名，完成客户 API 到服务器 API 的转换；并发送 SQL 请求，提供事务处理服务以及与网络通信协议的接口。当 LDBS 响应后，再把结果返回给应用程序。我们开发出来的局部代理正处于这里的应用程序的位置上。

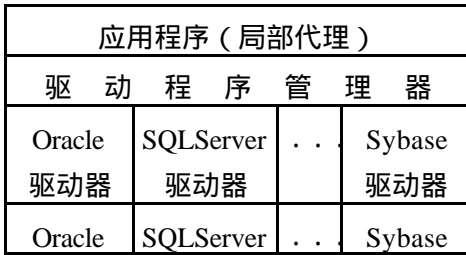


图 (2) ODBC 的体系结构



图 (3) 使用 ODBC 实现局部代理的程序结构

既然局部代理所要完成的中心任务是对全局管理层发布下来的子事务进行处理，那么使用 ODBC 设计的局部代理也必须为完成局部站点上全局子事务的处理提供最优的支持。在我们所开发的多数据库原型系统 MDBT\_1 中，我们使用 ODBC 设计了一个通用的局部代理，其程序结构如图 (3) 所示，步骤大致如下：

(1) 局部代理启动并建立 ODBC 环境，它为 ODBC 函数调用建立工作空间。局部代理或自动建立与 ODBC 连接，或让用户登录。由局部代理把用户提供的信息传给 ODBC，它根据这些信息建立与数据源的通信。局部代理可以在任何时候通过 ODBC 与数据源建立连接。按照一般多数据库理论，一个数据源应配置一个局部代理。在 MDBT\_1 中，这一点是通过单进程多线索技术实现的。

(2) 建立连接后，局部代理需要建立语句句柄。实际上是分配内存存储，以便通过它执行 ODBC 命令。建立语句句柄后，局部代理可能需要获取 LDBS 结构方面的信息以便工作，如了解当前连接对事务支持的程度。这可通过字典和统计函数调用来获取这些信息。

(3) 接下来是局部代理的核心内容，即完成对 LDBS 上全局子事务的处理。在这里，子事务可以由多个标准 SQL 语句或一个 SQL 语句组成。在有多个 SQL 语句的情况下，必须是由全局事务管理层根据各语句之间的语义联系进行串行化处理过的，这样才能有效地保持多数据库中数据的一致性。子事务语句执行完毕后，需要根据语句的执行情况决定子事务的提交与否。如果 LDBS 上子事务的所有语句都执行成功，则提交该子事务，否则回退该子事务，并将子事务提交或回退的信息送给全局事务管理层。

(4) 在局部代理退出工作之前，应首先释放 ODBC 语句句柄，再释放数据库连接，最后释放 ODBC 环境。在驱动程序关闭与数据库服务器的会话之后，才能最后停止局部代理的工作。

#### 4. 两种实现方法的性能比较

采用专用接口实现的局部代理在性能方面的优点是明显的。它不仅支持各数据库公有的数据类型和

公共特性，而且可以支持各 LDBS 特有的数据类型，充分体现各 LDBS 的私有特性，如数据模型、存取方法、事务协议、并发控制、用户接口等方面。这将大大提高整个 MDBS 的效率。并且，对那些并不支持两阶段提交协议的数据库系统及文件系统，采用这种方法能较容易地实现局部代理，较好地体现 MDBS 的异构性。但与此同时，我们要获得其优良的性能，也必须付出昂贵的代价。在开发局部代理之前，必须对各个 LDBS 的数据类型、数据库语言、接口特性、事务模式、封锁协议、权限控制等方面有相当深入的了解，而且需要为每一个 LDBS 开发一个专用接口，而真正做到这一点是比较困难的。在实现局部代理的过程中，也还会由于不了解各 LDBS 的内部结构而遇到许多问题。

使用 ODBC 开发局部代理则把问题简化了许多。采用 ODBC 实现的局部代理实际上是一个公用接口，它所支持的数据类型是各 LDBS 所共有的数据类型，它不需要我们深入地了解各 LDBS 所独有的某些特性。由于 ODBC 正趋于标准化和它本身的规范性，使用 ODBC 设计局部代理更为方便，实现难度大为简化。和大多数情况一样，采用这种方法实现的局部代理在性能上将会有所欠缺。它缺少了许多 LDBS 的私有特性，而且由于它并非 LDBS 的专用接口，使得 MDBS 的效率有所降低。

## 5. 结束语

局部代理作为全局事务管理层与局部数据库系统的接口，为多数据库事务处理提供了有力的支持。由于基于 SAG 标准的 ODBC 接口即将成为事实上的工业标准，因此本文提出了使用 ODBC 实现局部代理的方法。根据上述讨论的局部代理模型，在我们自行研制的多数据库原型系统 MDBT\_1 中，我们使用 ODBC 开发出了一个通用的局部代理 LA\_1。全局事务管理层可以根据需要启动多份 LA\_1，来负责管理不同的数据源。这种方法适合于那种并不复杂的多数据库系统，并且它的子系统最好具有更多相同的数据类型、数据库语言、事务模式、封锁协议、接口特性等。如果希望得到高效的局部代理，那得采用专用接口的方法来设计局部代理。尽管这种方法复杂得多，但它还是值得研究的。

### 【参考文献】

- 【1】 Sheth, A.P. and Larson, J.A., Federated Database System for Managing Distributed, Heterogeneous, and Autonomous Database, ACM Computing Surveys, 1990, 22(3): 183-236.
- 【2】 Wen-Syan Li and Chris Clifton, Dynamic Integration in Multidatabase Systems, Journal of Database Management, 1996, 7(1): 28-40.
- 【3】 Shirley A.Becker, Rick Gibson and Nancy L.Leist, A Study of a Generic Schema for Management of Multidatabase Systems, Journal of Database Management, 1996, 7(4): 14-20.
- 【4】 金远平，王能斌，联邦数据库管理系统 FDBMS 的设计与实现，计算机学报，1993, 16(6): 431-436.