

分布式工程数据库系统中事务管理机制相关技术的研究*

Research on Technology of Transaction Management Mechanism in Distributed Engineering Database System

李陶深

廖国琼

陈国宁

Li Taoshen

Liao Guoqiong

Chen Guoning

(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. and Info. Engi., Guangxi Univ., 10 Xixiangtanglu, Nanning, Guangxi, 530004, China)

摘要 从分布式工程设计事务环境和事务管理特点出发,给出一种分布式工程数据库系统(DEDBS)事务管理子系统的体系结构,提出一种适用于DEDBS中的事务两阶段提交协议,介绍两种能支持工程设计事务的并发控制方法:基于2PL的扩充分层封锁模型的并发控制方法和改进的乐观并行控制方法。

关键词 分布式工程数据库系统 事务管理 并发控制 分层封锁 两阶段提交协议

中图法分类号 TP 311.133.1

Abstract Considering the environment of distributed engineering design transaction and the properties of transaction management, an architecture of distributed engineering database system (DEDBS) is given. A transaction two-phase commit protocol which is suitable for DEDBS is proposed. Two concurrency control methods which can support engineering design transaction are also presented. They are the concurrency control method of extended declamation locking model based on 2PL and the improving optimistic concurrency control method.

Key words distributed engineering database system, transaction management, concurrency control, delamination lock, two-phase commit protocol

分布式工程数据库系统(Distributed Engineering Data Base System,以下简称DEDBS)是一种具有分布数据管理能力、面向工程设计应用的数据库系统。在DEDBS中,分布式事务管理一般处于核心位置,它的主要任务是调度、管理分布式数据库上运行的事务,负责它们的初启与结束,监督它们的执行^[1]。在完成这些任务的过程中,事务管理子系统要通过一系列的方法和技术,维护分布事务的性质和数据库的一致性和完整性,并采用适当的策略,从事务管理的角度来解决系统的可靠性、事务的并发控制以及系统资源利用等问题。由于工程设计事务具有长期性、合作性、试探性和分布性等特点,传统事务管理机制中的并发控制机制和常规分布式事务的提交机制不能很好地适用于DEDBS中的事务管理。因此,有必要对DEDBS中的事务管理机制进行研究。

分布式事务的整个执行过程是以协议的方式完成的。针对分布式事务处理的不同方面与不同问题,可以有不同相应的协议。目前,在分布式事务处理协

议的研究和应用方面涉及比较多的是一些以两阶段提交(Two Phase Commit,以下简称2PC)协议^[1]为基本模式的协议,如乐观的两阶段提交(O2PC)协议^[2]、谨慎的两阶段提交(P2PC)协议^[2]、可靠的两阶段提交(R2PC)协议^[3]等。虽然这些协议在可靠性、效率和可用性方面都有了很大的改进,但是它们没有考虑到分布式工程设计的特殊要求。

在DEDBS中的并发控制机制中,常用的事务调度模型有以锁方式为基础的形式模型、以时间印方式为基础的时间模型,以及乐观的并发控制方法。由于基于两阶段封锁(Two Phase Lock,以下简称2PL)协议的封锁技术比较成熟,许多研究人员都是从不同角度对其进行改进和扩充,使之满足工程设计的需要。例如,谓词二阶段封锁^[4]、利它锁^[5]就是从支持工程长设计事务的角度出发,要求长设计事务在遵守2PL规则的同时,尽早释放不再使用的数据项上的锁,以减少其它事务的等待时间,提高并发度;嵌套两阶段有序相容性封锁^[6]则是一种从支持合作设计事务的角度出发,以嵌套事务模型为基础,以有序相容性封锁为手段的两阶段封锁方法。

针对分布式工程设计事务环境和事务管理的特

2001-03-29收稿,2001-06-04修回。

* 广西自然科学基金项目(9712008)和国家自然科学基金项目(59868001)资助。

点,本文给出了一种 DEDBS系统事务管理子系统的体系结构,提出了一种适用于 DEDBS系统事务调度管理的事务两阶段提交协议,以及两种能支持工程设计事务的并发控制方法:即基于 2PL的扩充分层封锁模型的并发控制方法和改进的乐观并行控制方法

1 分布式工程数据库系统中事务管理子系统的体系结构

在 DEDBS中,一个事务是指一组在数据库的数据项上进行的操作,它将对数据库的状态产生影响。如果出现某些事件影响到事务的正常完成,那么事务操作将被夭折,这时要把数据库的状态恢复到事务操作开始前状态。DEDBS的事务管理机制主要涉及工程设计事务的提交管理机制、并发控制技术等问题。

一个大的工程设计事务往往分成许多项目事务,而每个项目又可分为许多子项目事务,一直分解下去,直到每个事务成为单独的设计者事务,这样就形成了一个嵌套结构的事务模型^[6]。为简化对问题的描述,我们将一个工程设计事务分解成两层:项目事务(Project Transaction, 以下简称 PT) 和设计者事务(Designer Transaction, 以下简称 DT) 图 1给出了一个基于嵌套结构的 DEDBS事务管理子系统的体系结构

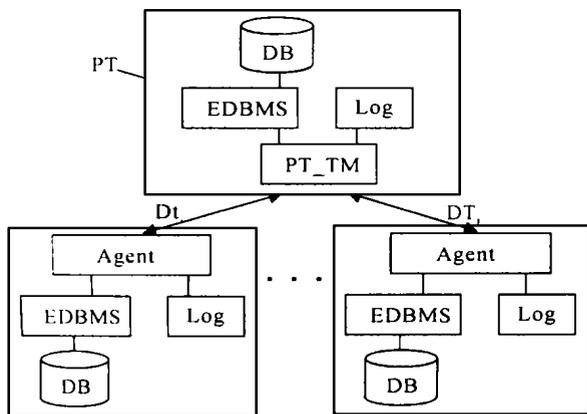


图 1 分布式工程数据库系统中事务管理子系统的体系结构
Fig. 1 Architecture of the transaction management subsystem of distributed engineering database system

在图所示的体系结构中,与事务的提交机制和并发控制有关的模块的功能如下:

(1) PT_TM (项目事务管理器) 负责将一个项目事务分解为多个设计者事务,并根据事先确定的事务调度次序进行事务调度,通过设计事务代理 (Agent),将各个子事务分发到各个不同场地,并接受 Agent 提交的设计结果。PT_TM 相当于协调者 (coordinator),负责协调各代理之间的操作,发出事务开始、提交和夭折等原语。它还管理项目事务的日

志,负责项目事务的恢复工作等。

(2) Agent (设计事务代理) 其作用是接收来自 PT_TM 的操作要求,并发送到各个设计者的私有工程数据库管理系统;设计者事务完成后将设计结果通过 PT_TM 提交到项目库中去。各个 Agent 相当于一个大的工程设计事务的参与者 (participant)

(3) Log (日志) 日志是发生故障时进行数据库恢复的主要依据,它记录了事务的有关信息和状态,反映了协议的进展情况。每个场地都有日志记录,它存放在稳定存储器中。

2 DEDBS中的两阶段提交协议

在分布式数据库系统 (DDBS) 中,分布事务的执行被分解为若干个场地为基础的子事务的执行。为了确保分布事务的原子性和一致性,要求 DDBS 必须保证参与分布事务的所有子事务要么成功提交,要么都失败回退。这种保证是通过分布式事务提交机制来实现的。

2.1 2PC协议

为了实现事务的正确提交,许多 DEDBS 都是采用 2PC 协议。2PC 协议的基本思想是坚持只有在分布事务所涉及的各子事务都同意提交该事务之后才能让它的所有操作对数据库产生影响。2PC 协议是建立在两个阶段通信规则的基础上的。第一阶段是协调者征询所有参加者对事务提交问题的意见,各参加者如果已经进入准备提交 (READY) 的状态就发出同意提交的消息,否则给出夭折意见。在第二阶段中协调者根据各参加者发回的意见,做出提交或夭折决定,并发向所有的参加者,后者则据此结束自己的子事务。2PC 协议的执行过程如下:

阶段 1: 协调者在其日志中建立 Begin commit 纪录,并向每个参与者发出 Prepare 消息,要求参与者在给定的时间内做出是否同意提交的决定,随后协调者进入等待 (Wait) 状态;每个参与者根据自己的子事务执行情况做出相应的决定,如果子事务全部执行完成,则向协调者发“Commit” (提交) 消息,否则发“Abort” (夭折) 消息。

阶段 2 如果协调者收集到的意见全部是 Commit,说明全体参与者一致同意提交,那么协调者就把一个全局的 Commit 消息发给每个参与者,表示可以把本事务提交。如果有一个参与者表示不能提交,则协调者做出全局夭折本事务的决定,即向每个参与者发送 Abort 消息。每个参加者根据协调者的命令完成实际的提交或夭折,并向协调者发出应答消息。协调者在收到所有应答消息后,在日志上写下

End of transaction 纪录, 表示该事务已经结束

2PC协议是以一种相当简洁的方式来保证分布事务的原子提交性质,但是没有考虑到工程设计事务的特点,不能很好地适用于分布式工程数据库系统的事务管理。例如,在工程设计过程中,一个设计项目往往分成几个设计方案由许多设计者共同完成,然后从已完成的不同方案中选择一个最佳方案。因此,个别子设计事务的成败不一定影响到整个项目事务的执行,整个项目事务也不一定因为个别子设计事务的失败而夭折;同时,整个项目事务的完成也不必等待全部子事务都完成。

2.2 适用于 DEDBS 的事务两阶段提交协议

我们在对基本 2PC 和三阶段提交协议 (3PC) 的分析和比较的基础上,设计了一种能用于 DEDBS 的事务 2PC 协议^[7]。这一协议设置了一个事务协调者负责分布事务的分解、子事务的调度、协调子事务的执行和决定分布事务的提交和夭折等。该协议也将事务的执行分为两个阶段:即事务的调度阶段和事务的结束阶段。在事务的调度阶段,PT_TM 按照事先确定的子事务执行顺序和优先级向各个 Agent 调度子事务执行并等待应答,所有条件满足的子事务可以同时执行。每当有子事务提交或夭折时要检查项目事务的结束条件是否满足。如果满足则进入事务结束阶段,如不满足则继续等待。在事务结束阶段,如果成功结束,则向公有库提交设计结果,并向仍处于执行状态的 Agent 发出事务结束命令;如果失败结束,则只向仍处于执行状态的 Agent 发出事务结束命令。当 PT_TM 收到所有执行状态的 Agent 应答报文后,清除项目库,结束整个项目事务。图 2 是事务执行过程中 PT_TM 和 Agent 之间的报文交换。



图 2 分布式工程数据库两阶段提交协议的消息交换过程
Fig. 2 Message exchange procedure in the two-phase commit protocol of distributed engineering database

在该协议中,PT_TM 是协调者,负责调度和协调它的各个子事务的执行,决定事务本身及其子事务

是否提交或夭折。各个 Agent 是参与者,它不能自行决定是否提交和夭折,只有收到父事务的同意报文后,才能提交或夭折。因为工程设计事务各个子事务是分工协作的,它的提交和夭折可能会影响到其它子事务的执行,甚至个别子事务的夭折会导致整个事务的失败。因此,由项目事务决定子事务是否提交和夭折,是符合工程设计原则的。

3 支持工程设计事务的并发控制方法

并发控制技术是 DEDBS 中事务管理的基本任务之一,其目的是保证 DEDBS 中多个事务的高效正确执行。目前,分布式并发控制技术是以集中式数据库的并发控制技术为基础并针对分布式系统的特点扩展而来。集中式数据库的并发控制技术以串行化理论为基础,并以它为模型来检验方法的正确性。在 DEDBS 中它也起到同样的作用。按照串行化理论,一个在数据库上运行的事务的所有操作,按其性质可被分为两类,即读和写,它们所操作的数据对象的集合被分别称为读集与写集^[8]。

在 DEDBS 中,每个结点上的事务是可以并发的,而不同结点上的事务则完全是可并行的。以集中式数据库的并发控制技术为基础的分布式并发控制的重点就是要构造一个可保证全局顺序的可串行化的调度,即以局部各结点的可串行化为基础,再针对各结点的执行情况来保证整个系统的全局顺序。我们在传统的并发控制技术基础上,设计了两种支持工程设计事务的并发控制方法,即基于 2PL 协议的扩充分层锁模型的并发控制方法和改进的乐观并发控制方法。

3.1 基于 2PL 的扩充分层封锁机制

以锁方式模型为基础的并发控制算法在分布式数据库系统中得到了较为广泛的应用。该模型的基本思想是:事务对任何数据项进行操作前必须先申请对该数据项的封锁,只有申请到锁以后,即加锁成功以后,才能对数据项进行操作。操作完成后,要释放已经申请的锁。通过锁的共享及排斥的特性,可以实现事务的可串行化调度。

目前在分布式系统中使用锁方式模型实现并发的最著名的算法是 2PL 协议。采用 2PL 协议的分布式系统可以保证事务的调度是可串行化的。该协议遵循以下的原则^[8]:

- (1) 事务是具有良好行为的;
- (2) 封锁中遵守锁方式中的共享与互斥语义的规则;
- (3) 一旦事务开始释放封锁便不会再有新的封锁要求;
- (4) 对事务的封锁分为两个阶段:即上升阶段和释放阶段。

分布式 2PL 方法基本上是上述模式的直接扩展。但是这种简单的读写锁模型只区分两种锁：共享锁 (S) 和排它锁 (X)，它不能很好地适应工程设计事务中一些特殊性的需要。为此，我们设计了一种基于 2PL 的扩充分层锁模型^[9]。在这一模型中，我们对简单的锁模型进行了扩充，在原有的共享锁 (S) 和排它锁 (X) 的基础上增加 4 种锁：更新锁 (U)、浏览锁 (D)、意向共享锁 (IS)、意向排它锁 (IX) 和意向更新锁 (IU)，各种锁及工作方式定义如下：

- X——排它锁：用于独占事务访问数据对象；
 - U——更新锁：用于更新事务访问数据对象；
 - S——共享锁：用于读事务访问数据对象；
 - D——浏览锁：用于浏览事务访问数据对象；
 - IX——意向排它锁：用于对已上 X 锁的数据对象的所有上层结点进行封锁；
 - IU——意向更新锁：用于对已上 U 锁的数据对象的所有上层结点进行封锁；
 - IS——意向读锁：用于对已上 S 锁的数据对象的所有上层结点进行封锁。
- 为此，我们建立如图 3 所示的相应的相容性矩阵。

	IX	IU	IS	X	U	S	D
IX	+	+	+	-	-	-	-
IU	+	+	+	-	-	⊕	+
IS	+	+	+	-	-	+	+
X	-	-	-	-	-	-	-
U	-	-	-	-	-	⊕	+
S	-	-	+	-	-	+	+
D	-	+	+	-	+	+	+

图 3 扩充分层锁模型的相容性矩阵

Fig. 3 Extended delamination lock model compatibility matrix

+ —— 相容 Compatible; - —— 不相容 Incompatible; ⊕ —— 有序相容 Ordered compatible.

基于这一相容性矩阵，扩充分层锁模型可以提供以下几种访问 CAD 数据的方式：

- (1) 独占数据访问 采用 X 锁。因为 X 锁同其它任何锁都不相容，所以当希望本事务访问某数据其间拒绝其它事务对其进行任何访问时，选用 X 锁；
- (2) 更新数据 采用 U 锁。因为 U 锁同 D 锁相容、同 S 锁有序相容，因此当对某数据进行更新且允许其它事务读（或浏览）此数据时，选用 U 锁；
- (3) 读取数据 采用 S 锁。采用 S 锁读取某数据时，拒绝 X 和 U 锁访问，即不允许其它事务修改该数据，但允许浏览（上 D 锁）数据；
- (4) 浏览数据 采用 D 锁可对数据进行浏览，它

只与 IX、X 不相容，允许其它访问同时进行。

该模型为工程设计事务管理提供了较大的灵活性，因为工程设计者可以根据自己的设计要求选用 S、X、U 和 D 锁，各种意向锁则是在加 X、U、R 锁时自动加上的。当希望对某数据长期独占访问时，可选择 X 锁；当对修改的数据允许被授权事务读取时，可选择 U 锁。对读锁的选择可根据准确参考和一般参考的原则来决定：准确参考是指要求读取的数据是准确的数据，也就是说读取数据的准确性直接影响到设计结果的准确性；而一般参考则是指访问者对读取的数据只作大致了解，不要求十分准确，如管理者只希望了解设计的进度等。因此，当读取的数据直接影响到设计结果时就选择 R 锁，而仅仅是一般了解某数据时可选用 D 锁。增加 D 锁主要是从工程设计过程中有时允许存在不一致的情形下考虑的。定义的各种意向锁是在加 X、U、R 锁时自动对该数据对象的上层结点加上的，从而防止了对结点的外层结点加上不相容的锁。由于该模型是根据设计的实际情况选择最佳封锁粒度进行封锁的，比较有利于提高工程设计事务的并发度。

3.2 改进的乐观并发控制方法

基于 2PL 的扩充分层封锁机制是在成熟的 2PL 协议的基础上改进而来的，在一定程度上满足了工程设计事务的要求。但是由于这一机制仍然存在封锁自身的缺陷，如等待和死锁等，从而影响了事务执行的并行度。乐观的并发控制方法则可以克服等待和死锁的现象，它的一个基本前提是：事务间极少发生冲突，大多数事务可以无冲突正确执行完毕。传统的乐观并发控制方法是在事务终止时才进行事务的同步。该方法将事务的执行过程分为以下 3 个阶段：

- (1) 读段。读出所需的数据，进行相应的数据处理，决定回写结果的值，并把该值保留在一个暂存空间，而不是写入数据库；
- (2) 检验段。检测对数据库的这一更新操作是否会引起数据库的不一致性；
- (3) 写段。通过检测阶段后，将 (1) 中的结果的值写入数据库。

传统的乐观并发控制方法是利用回退重启机制来解决冲突的。这种将回退重启作为数据一致性手段的并发控制方法在工程设计事务中存在许多缺陷：

- (1) 事务的回退重启将浪费大量的工作；
- (2) 占用大量的存储空间，因为需要大量的存储空间来存储那些需要写入数据库的数据；
- (3) 事务并发度低，因为检测事务有无冲突需要花费较多的时间，这将降低事务的并发度；

(4)重启大量的合作设计事务将可能导致长事务“饥饿”的现象

为了解决传统的乐观并发控制方法的这些问题,我们设计了一种改进的乐观并发控制方法,它能较好地满足工程设计事务并发处理的要求,提高事务的并发度^[10]。

与传统的乐观并发控制方法一样,改进的乐观并发控制方法也是将事务的执行分为读段、检验段和写段,其中读段的功能是从父事务工作区重读所需的访问对象副本检出(check out)到本事务工作区中,记录各副本的原始版本号后执行设计事务;检验段检验设计数据是否正确,以确定事务是直接进入写段还是经过修改后才能进入写段;写段是将子事务的设计对象检入(check in)回父事务的工作区。

改进的乐观并发控制方法的读段和写段与一般工程数据库的 check out/check in阶段相对应,唯一的区别是 check out时要记录读集的原始版本,check in前要经过检验。检验段的详细工作步骤如下: 步骤 1 将本事务读集原始版本同父事务工作区相同对象版本号比较; 步骤 2 如果两个版本号完全相同,表示检测正确, goto 步骤 6; 步骤 3 如果两个版本号不相同,则父事务向子事务通知已经修改的对象; 步骤 4 子事务将已修改对象的更新版本 check out到本事务工作区; 步骤 5 将本工作区版本同 check out来的更新版本合并,产生一个新版本; 步骤 6 进入写段。

工程设计事务根据它的性质可归纳为更新事务和查询事务(读事务)两种。通过对上述的改进乐观并发控制方法的应用分析,我们发现在工程事务执行过程中存在着数据暂时不一致的现象,如读-写不一致、写-写不一致。因此,由必要对改进的乐观并发控制方法作进一步完善,具体的改进措施如下:

(1)将工程设计事务分为更新、查询事务两类,对于查询事务不考虑并发控制,不设置检验段,不保留读集,这样就减少了查询事务的并发控制开销。但是这样做不利于更新事务;

(2)当更新事务提交时,将修改对象的更新版本及时通知已读取该对象的活动兄弟读事务,使得各相关读事务能尽快地了解有关对象的最新信息,避免不必要的工作浪费。

4 结语

本文提出的 DEDBS的事务 2PC协议与基本的 2PC协议的根本区别在于全局事务是以事先确定的可提交状态集作为决定整个全局事务及其子事务是否可以提交或夭折的依据,并赋予了全局事务具有单

方面决定自身及其子事务提交或夭折的能力,从而克服了两阶段提交协议中因个别子事务的夭折导致全局事务的夭折和全局事务必须等待所有子事务完成后才能结束的缺陷。基于 2PL的扩充分层封锁机制是以两阶段有序相容性封锁方法为基础,结合工程数据库的层次性,对简单读/写锁模型进行扩充,它能较好地支持复杂合作设计事务和提高事务的并发度。改进的乐观并发控制方法在冲突的检验方式和解决冲突的方式上与传统的乐观并发控制方法有了比较大的改进,具体表现在该方法所检验的是事务读集原始版本是否被更改,如被更改,则报告有冲突发生;当由冲突发生时,它是采用版本合并的方法来解决冲突问题。它解决了传统乐观并发控制方法存在的对工程事务处理支持较弱的问题,避免了长事务的回退重启及“饥饿”现象。

我们利用上述的 DEDBS的事务 2PC协议和改进的乐观并发控制方法实现了一个分布式事务管理子系统的样机系统。实验表明,我们提出的事务提交协议和并发控制方法是合理、可行的,可满足分布式工程数据库系统中事务管理的需要^[9]。

参考文献

- 1 郑振楣,于戈,郭敏. 分布式数据库. 北京: 科学出版社, 1998.
- 2 Bipins C, Desai, Bowton S. Boutros performance of a two phase commit protocol. Information and Software Technology, 1996, 38 (9): 581~ 599.
- 3 Ho Dong Yoo, Myoung Ho Kim. A reliable global atomic commitment protocol for distributed multidatabase system. Information Science. 1995, 82 (1/2): 75~ 102.
- 4 Henry F. Korth on long-durative CAD transaction. Information Science. 1988, 46 (3): 73~ 107.
- 5 Barghouti N S, Kaiser G E. Concurrency control in advanced database application. Acm Computing Surveys, 1991, 23 (3): 269~ 317.
- 6 齐进,张家明,周伯鑫. 工程数据库中一种支持合作设计事务的并发控制机制. 计算机研究和发展, 1998, 35 (11): 987~ 990.
- 7 廖国琼,李陶深. 分布式工程数据库系统中事务提交机制的研究. 计算机辅助设计与图形学学报, 2001, 13(4): 357~ 361.
- 8 周龙骧,邵为民,顾君忠等. 分布式数据库管理系统实现技术. 北京: 科学出版社, 1998.
- 9 廖国琼. 分布式工程数据库系统中事务管理的研究(硕士学位论文). 南宁: 广西大学, 2000.
- 10 廖国琼,李陶深. 一种支持工程设计事务的乐观并发控制方法. 计算机工程, 2000, 26 (7): 24~ 25.

(责任编辑: 黎贞崇)