

支持工作流的 DBMS 模型

孙明 张世栋 洪晓光

(山东大学计算机科学与技术学院 济南250100)

摘要 本文提出支持工作流的数据库管理系统模型,目的在于借助 DBMS 对数据管理的成熟技术和对 DBMS 的扩展,使数据库能够为 workflow 管理系统的任务恢复提供相关数据一致性服务,从而为 workflow 一致性管理做出贡献。

关键词 工作流, DBMS, 一致性

A DBMS Model Supporting Workflow

SUN Ming ZHANG Shi-Dong HONG Xiao-Guang

(School of Computer Science and Technology, Shandong University, Jinan 250100)

Abstract In this paper, we propose a DBMS model supporting workflow, to make databases provide related data consistency service for task recovery of WFMS by utilizing the mature data management technology of DBMS and extending the functionality of DBMS. The main contributions of this paper are in the field of workflow consistency management.

Keywords Workflow, DBMS, Consistency

数据库管理系统(DBMS)是面向数据管理的系统, workflow 管理系统(WFMS)是面向过程管理的系统。二者管理对象范围不同,适用领域不一致,但二者之间依然存在着多种形式的内在联系。如过程管理包含着数据管理的成份,任务执行失败的恢复,必然包含相关数据的恢复等等。WFMS 与 DBMS 之间彼此独立,其优点在于任务分工明确,各自可以集中精力针对自己领域的问题进行管理研究。但二者之间完全独立,不利于其间的协调和优势互补。本文提出支持工作流的数据库管理系统模型,旨在加强 DBMS 同 WFMS 之间的内在联系,使二者的优势可以互补,同时保持 WFMS 同 DBMS 之间的相对独立性。

1 WFS-DBMS 的体系结构

定义1 支持工作流的数据库管理系统(WorkFlow Supported DBMS, WFS-DBMS): 是能够支持将数据管理同 workflow 之间建立联系,可以协助 workflow 管理系统进行必要数据处理的数据库管理系统。

WFS-DBMS 不同于传统 DBMS, 它可为 workflow 管理提供

必要的数据库管理支持,从而使 WFMS 可以借助 WFS-DBMS 管理数据的优势,完成自身的一些数据处理要求。在保证 workflow 一致性方面, WFS-DBMS 可以提供有效的支持。例如,任务 T 执行失败需要恢复, T 所做的数据处理也应被撤消,如果任务 T 的数据处理已经提交,传统的 DBMS 不允许通过撤消相应数据处理事务完成数据恢复,只能依靠 T 的补偿任务进行必要的处理;而 WFS-DBMS 借助自己的数据处理功能,在允许的前提下,通过撤消相关数据处理事务,完成数据恢复,其优势是显而易见的。当然,这要求 WFS-DBMS 具备撤消某些已提交事务的功能,这些功能,可通过对 DBMS 进行扩展完成。必须指明, workflow 任务的恢复,不仅仅是数据的恢复,还包括其它相关工作的恢复和补偿,有些工作甚至不是计算机系统自身可以完成的,但不管怎样,数据恢复永远是 workflow 任务恢复中一项非常重要且相当困难的工作, WFS-DBMS 提供的相关支持功能,可望从支撑系统级别上为任务恢复中数据恢复功能提供支持和帮助,从而减缓 WFMS 的一致性管理难度。

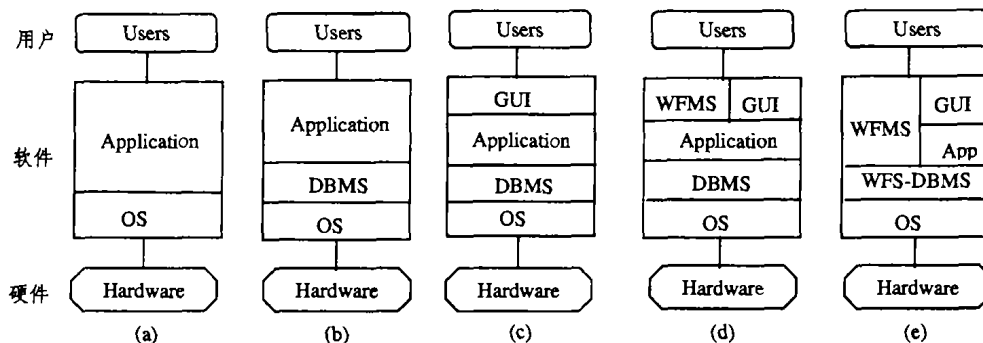


图1 软件体系结构示意图

软件体系结构的发展历程如图1所示。首先是操作系统从具体的应用软件中分离来,操作系统对管理的主要硬件设备负责,并为应用软件提供服务,如图1(a)所示,操作系统的形

成和发展,对软件技术的发展贡献巨大。计算机技术的进一步发展和对数据管理需求的增长,使对数据的管理也从具体应用软件中分离来,形成数据库管理系统 DBMS,基于 DBMS

的软件体系结构如图1(b)所示。进一步,对界面管理技术的成熟,又把用户界面管理部分从应用程序中分离出来,形成用户界面管理系统。这样,应用程序从繁琐复杂的用户界面管理中解放出来,且保证了用户程序界面风格的相似性,便于用户使用应用程序。用户界面管理功能从应用程序中分离后,新的软件体系结构如图1(c)所示。随着 workflow 概念的提出和 workflow 技术的进步,workflow 管理系统 WFMS 作为一个通用系统协助对过程的管理,使对过程的管理和控制也分离成为一个独立的层次,基于 WFMS 的软件体系结构如图1(d)所示。

由图1(d)可看出,作为管理和控制过程运行的 WFMS 同管理数据的 DBMS 之间并没有建立直接联系,这一点是指 WFMS 完成对过程的管理和运行控制这一核心职能时,并没有也无法借助 DBMS 提供的服务。在 DBMS 增强对 workflow 管理的支持功能,形成 WFS-DBMS 之后,WFS-DBMS 可以直接为 WFMS 的过程管理控制提供服务,也就是 WFMS 和 WFS-DBMS 之间有直接的联系和服务关系,软件体系结构演化为如图1(e)所示的体系结构。

WFS-DBMS 对外提供的服务可分为三类,即对 WFMS 提供的工作流一致性相关服务、对受 WFMS 管理控制的过程任务提供的数据库服务,以及向与 workflow 无关的应用提供的普通数据库服务。同 DBMS 一样,WFS-DBMS 有集中式和分布式之分。本文为简单起见,以集中式 WFS-DBMS 为研究对象,目的在于将研究的焦点集中于 DBMS 如何满足向 WFMS 提供服务的需求。WFS-DBMS 的分布需求,可在集中式 WFS-DBMS 的基础上,参照 DBMS 的分布技术予以实现。

2 WFS-DBMS 对 DBMS 的扩展策略

WFS-DBMS 的目标是在 DBMS 的基础上,扩展功能以能够对 WFMS 提供支持服务。WFS-DBMS 扩展功能主要是向 workflow 任务一致性管理提供数据一致性服务。

WFS-DBMS 主要通过建立数据库事务与 workflow 任务之间的联系、扩展传统 DBMS 日志、允许撤消已提交的事务,为 workflow 任务补偿提供数据恢复服务。数据库根据日志恢复数据,是对数据的物理补偿,即补偿后数据将恢复到任务执行前的状况,是理想的补偿方式,且 WFS-DBMS 根据数据库日志进行数据恢复,有正确性保证,并可减少书写补偿程序的工作量。

传统的 DBMS 事务具备 ACID 性质,即原子性、一致性、隔离性和永久性。其中永久性是指事务一旦提交,不允许再被撤消。扩展的 DBMS 事务模型,如 Sagas 模型等,放宽了子事务在隔离性等方面的限制,以提高长事务的并行性,但对子事务的补偿,依然需要执行补偿程序,不能由 DBMS 自动完成,严格地说,事务提交后具备的永久性没有被突破。WFS-DBMS 为支持任务恢复,将研究在具备一定条件下,撤消已提交事务,也就是在可能的条件下,突破事务的永久性性质:

传统 DBMS 的日志 Log,是对数据库操作轨迹的记录,每个操作记录可用一个五元组表示:

$$LogRecord = \langle Tid, TimeStamp, Cid, OldValue, NewValue \rangle$$

其中, Tid 为事务标识,在整个数据库系统内唯一标识一个事务; $TimeStamp$ 是 DBMS 提供的时间戳,整个系统内唯一并按操作先后严格增序排列; Cid 是当前 $LogRecord$ 所修改的分量标识, $OldValue$ 是分量修改前的值,当操作为插入操作时该值为空; $NewValue$ 是分量修改后的值,对删除操作该值为空。日志对数据库操作轨迹的记录使 DBMS 可以根据日志进行事务的重演或者撤销,以保证事务的一致性。数据库系

统的日志有未提交和已提交两种状态(本文不考虑分布事务的“准备提交”等状态,这些状态不影响 WFS-DBMS 的有关讨论),未提交事务可由 DBMS 自身进行恢复,本文不予讨论;已提交事务具有永久性,不允许恢复。本文讨论对已提交事务可以进行的恢复。

传统 DBMS 的事务进行必要的数据封锁管理,以实现事务的并行性,事务完成后释放事务拥有的所有封锁,事务封锁信息不做永久记录。

2.1 WFS-DBMS 对事务存贮结构的扩展

WFS-DBMS 为能够在条件具备的前提下,撤消已提交的事务,必须对事务存贮结构进行扩展。扩展后的事务结构中,将记录事务同 workflow 任务之间的联系,并永久记录事务对数据的封锁情况。

定义2 WFS-DBMS 扩展的事务,是一个四元组 $\langle Tid, TaskId, TLock, Log \rangle$ 。

其中, Tid 是 DBMS 中的事务唯一标识,同传统事务的 Tid ; $TaskId$ 是 workflow 系统中任务的标识,在 workflow 系统中唯一标识一个任务,对不是 workflow 任务发起的事务, $TaskId$ 为空; $TLock$ 是对事务封锁信息的记录; Log 是数据库操作日志,同传统 DBMS 的 Log 。

约定1 在 WFS-DBMS 中,对由 workflow 任务发起的数据库事务,一个事务必须属于一个任务,不允许一个事务涉及多个任务。显然,这种约束是可以接受的。它符合任务及事务的自然语义。

约定2 在 WFS-DBMS 中,不允许一个任务发起两个或两个以上的数据库事务。在这里,一个任务发起的事务,不包含该任务的子任务发起的事务。虽然从自然语义上讲,一个 workflow 任务可发起多个事务,但如果一个任务可发起多个事务,按其语义,它一定可以划分为多个子任务,最终可使每一个任务最多只发起一个事务。出于实现方便上的考虑,WFS-DBMS 对一个任务允许发起的事务数量进行限制,要求一个任务最多只能发起一个事务,显然,这个约束是可以接受的。

$Tlock$ 记录了事务对数据的封锁情况。事务对每一个数据库单元的封锁,使用一条封锁记录 $LockRecord$ 来保存。这样, $Tlock$ 是 $LockRecord$ 的集合。 $LockRecord$ 自身也可表示为四元组:

$$LockRecord = \langle Tid, TimeStamp, Cid, LockType \rangle$$

其中, Tid 是事务标识,含义同上; $TimeStamp$ 是事务封锁数据的时间戳,时间戳性质同上; Cid 是封锁数据的分量标识,这里假设封锁粒度为分量,有关封锁粒度的讨论同传统 DBMS 中相应的讨论; $LockType$ 是封锁的类型,包括共享封锁(S锁)和独占封锁(X锁)两种锁类型,定义同传统 DBMS 中相应的定义。

WFS-DBMS 对事务结构的扩展,为 WFS-DBMS 向 workflow 系统提供数据一致性服务奠定了基础。即在可能的前提下,通过撤消 workflow 任务已经提交的事务,为任务一致性关系提供数据恢复服务。

2.2 WFS-DBMS 对 WFMS 提供一致性服务的策略

WFS-DBMS 按照下述策略,为 workflow 一致性提供服务:

策略1 workflow 任务同数据库事务之间建立对应关系。

WFS-DBMS 中建立并保持 workflow 任务同数据库事务之间的对应关系,是 WFS-DBMS 为 WFMS 提供数据一致性服务的前提。WFS-DBMS 扩展的事务结构可以存贮任务同事务之间对应关系;二者之间对应关系的建立则需要在工作流任务发起事务时完成。

传统 DBMS 事务管理命令为:

Begin Transaction

Commit/Rollback

即事务以 Begin Transaction 开始,以 Commit 或 Rollback 结束。WFS-DBMS 将事务管理命令扩展为

Begin Transaction [TaskId]

Commit/Rollback

即以 Begin Transaction [TaskId] 为事务起始命令。其中 TaskId 为可选项,在工作流任务发起事务时,提供该选项;在非工作流任务发起的事务中,不写此选项,此时与传统事务起始命令相同。WFS-DBMS 对事务结束命令没有扩展,且结束命令 Commit 和 Rollback 的语义与传统 DBMS 中相应语义相同。

WFS-DBMS 通过对事务管理命令的扩展,可以感知和记录事务与工作流任务之间的对应关系,该对应关系在 WFS-DBMS 扩展的事务结构中永久存贮。

策略2 WFS-DBMS 事务对数据封锁的记录。

传统 DBMS 事务在执行期间,对事务涉及的数据按一定规则进行封锁,以保证并行事务调度的正确性。在传统 DBMS 中,事务运行结束,事务所获取的封锁全部释放,并且封锁信息不再记录。

在 WFS-DBMS 中,事务对数据的封锁不变,并在事务每申请获得一个锁之后,立即将获得锁的有关信息,以 Lock-Record 的形式,存贮在 WFS-DBMS 扩展事务的 Tlock 中,并永久保存,除非事务被撤消。

传统 DBMS 保证并行事务调度正确性的经典技术,是遵从两阶段锁协议(2PL)或者遵从更严格的协议。2PL 协议要求事务在读、写数据之前,必须获得有关数据的封锁。这意味着,WFS-DBMS 在读写数据之前,一定获取了有关数据的封锁,并将封锁写入了扩展事务的 Tlock 中,这条规则为 WFS-DBMS 撤消已提交事务的条件判定奠定了基础。

策略3 WFS-DBMS 撤消已提交事务的条件判定。

WFS-DBMS 在合适的条件下,允许撤消已提交的事务。WFS-DBMS 作为 DBMS,本质任务是进行数据管理。为工作流系统提供数据恢复服务,前提条件是有关服务不能影响数据库数据的一致性。为此,WFS-DBMS 应 WFMS 要求,试图撤消一个已提交事务之前,将首先判定撤消该事务是否会影响 DBMS 数据的一致性。在撤消事务会影响 DBMS 数据一致性时,撤消工作不予进行。

在满足图2规定的条件时,WFS-DBMS 允许撤消已提交的事务 T ,事务 T 的撤消不会影响 DBMS 数据一致性。

对任意的 LockRecord LR1:

```
(
  (LR1.TID=T.TID ∧ LR1.lockType='x 锁')
  ⇒(
    不存在 LockRecordLR2,满足:
    (LR2.TID≠T.TID ∧ LR2.TimeStamp>LR1.TimeStamp
     ∧ LR2.CID=LR2.CID)
  )
)
```

图2 WFS-DBMS 允许撤消已提交的事务 T 的判定条件

满足该条件表示:任何事务 T 修改的数据,没有在 T 之后被其他事务封锁过。也就是说,事务 T 修改的数据,没有被其它事务读、写过。至于事务 T 读操作作用到的数据,由于 T 本身并没有修改该数据,在 T 之后有没有其它事务对此数据操作,不在考虑范围之内。显然,该条件是撤消一个已提交事务的充分条件,但不一定是必要条件,因而是该条件可以被采用的,但不一定是最优的判定条件。

WFS-DBMS 中,使用图2给出的判定条件,判定一个已提交事务是否可以撤消。该条件的判定非常方便,事实上,该条件本身就是按元组关系演算的形式书写的,因而对数据系统来讲,条件判定非常方便。

策略4 WFS-DBMS 撤消事务的策略

WFS-DBMS 撤消事务的算法

步骤1:WFS-DBMS 在接到要撤消一个事务 T 的要求时,判定事务 T 是否已经提交,如果该事务尚未提交,转步骤2;否则转步骤3;

步骤2:按照传统 DBMS 的 Rollback 方法,执行 Rollback 操作,撤消事务工作完成。

步骤3:判定事务 T 是否由工作流任务发起,如果不是由工作流任务发起,撤消事务工作失败;如果是由工作流任务发起,转步骤4;

步骤4:WFS-DBMS 判定事务 T 是否具备撤消的条件。如果具备撤消条件,转步骤5;否则转步骤6;

步骤5:根据事务 T 的 Log,按照传统 DBMS 撤消未提交事务的方法,撤消事务 T ,撤消工作完成。

步骤6:WFS-DBMS 根据允许撤消事务判定条件,计算因为哪些事务的存在,致使事务 T 不能被撤消,并将这些事务对应的工作流任务连同事务 T 不能被撤消一并通知 WFMS。事务撤消工作以失败结束。

上述的算法中,步骤6计算并通知 WFMS 哪些工作流任务发起的事务,致使事务 T 不能被撤消。这样做的目的在于,使 WFMS 可以决定,是否要通过撤消有关的任务,使事务 T 可以被撤消;当然,如果有非工作流任务发起的事务,致使事务 T 不能被撤消,WFMS 接到的通知,将是无法通过撤消有关的任务,完成事务 T 的撤消工作。

为保证事务可撤消条件判定的有效性,WFS-DBMS 获取的锁,不仅需要传统 DBMS 需要的封锁,而且须根据日志文件,对已经被删除或修改的数据,判定其 OldValue 是否在事务操作的影响范围之内,如果是,同样为其形成一条封锁记录,记入 Tlock 中。这样,可为其后撤消条件的严格性奠定基础。

结论 WFS-DBMS 的提出,为 DBMS 向 WFMS 提供数据一致性服务探索了一条道路。WFS-DBMS 的实现,可望大大减轻工作流一致性方面的压力。本文的研究工作还仅限于理论探讨。希望这方面的探讨能够为有关产品的开发奠定基础,使 WFS-DBMS 的研发成为可能。

参考文献

- 1 Rusinkiewicz M, Sheth A P. Specification and Execution of Transactional Workflows. Modern Database Systems, ACM Press and Addison-Wesley, 0-201-59098-0, 1995. 592~620
- 2 Grefen P, Vonk J, Apers P. Global Transaction Support for Workflow Management Systems. VLDB Journal, 2001, 10: 316~333
- 3 张世栋,王新军,王海洋,等.基于离线数据载体的分布数据一致性研究.小型微型计算机
- 4 崔立真,张世栋,郑永清,王海洋.一种工作流动态修改方法.系统仿真学报,2003,15(2):264~266
- 5 王涛,黄力芹,吴耿峰.工作流管理的发展历程和趋势.计算机工程与科学,2001,23:97~100