

分片段的移动实时数据库事务模型

党德鹏¹ 周立柱¹ 刘云生²

(清华大学计算机科学与技术系 北京 100084)¹ (华中科技大学计算机学院 武汉 430074)²

摘要 由于移动计算环境的固有特性、数据访问者的移动性、事务的截止期限限制等,传统平坦事务模型无法满足移动实时数据库的需求。本文针对移动实时数据库事务的特点,提出了一种分片段移动实时数据库事务模型。

关键词 移动实时数据库,移动实时事务,可串行化

A Model of Mobile Real-time Transactions

DANG De-Peng¹ ZHOU Li-Zhu¹ LIU Yun-Sheng²

(Department of Computer Science & Technology, Tsinghua University, Beijing 100084)¹

(College of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan 430074)²

Abstract Considering the inherent characteristics of mobile computing environments, moving of database access, and the deadlines of mobile real-time transactions, in this paper, a better transaction model is given.

Keywords Mobile real-time database, Mobile real-time transactions, Serializability

由于移动计算环境的固有特性、数据访问者的移动性、事务的截止期限限制等,传统平坦事务模型无法满足移动实时数据库的需求^[1]。本文针对移动实时数据库事务的特点,提出了一种分片段移动实时数据库事务模型。

1 移动实时事务特征

使用移动计算机访问无线网络中的实时数据需要事务的支持。然而,这种环境中的事务与传统事务显著不同。

首先是事务执行的环境不同。与基于固定网络的传统分布式计算环境相比,移动计算环境具有以下一些主要特点^[3~5]:

1)移动性。在移动计算环境中,一台移动计算机不仅可以在不同的地方连通网络,而且在移动的同时也可以保持网络连接。这种计算平台的移动性可能导致系统访问布局的变化和资源的移动性。而且,个人的移动性(即在不同地方使用当地的计算设备)也随着个人通信网 PCN 与网络计算机 NC 的提出而日益突出。

2)频繁断接性。移动计算机在移动过程中,由于受使用方式、电源、无线通信费用、网络条件等因素的限制,一般不采用持续性联网的工作方式,而是主动或被动地间歇性入网、断接(disconnection)。

3)网络条件多样性。移动计算机的移动性使得不同时间可用的网络条件(如网络带宽、通信代价、网络延迟以及服务质量)是变化多端的,甚至相差悬殊:移动计算机既可以连入高带宽的固定网络中(如 LAN),也可以工作在低带宽的无线广域网中(CDPD),甚至处于断接状态。

4)网络通信的不对称性。由于物理通信媒介的限制,一般的无线网络通信都是不对称的,表现在固定服务器节点可以拥有强大的发送设备,而移动计算机的发送能力非常有限,于是下行链路(服务器到移动计算机)的通信带宽与代价和上行链路(移动计算机到服务器)相差很大。此外,许多客户/服务器应用中的信息流动范型也会引起通信的不对称性。

5)移动计算机的电源能力。移动计算机主要依靠蓄电池供电,而现在电池的容量相当有限,例如一台典型的笔记本电脑在正常的连续使用情况下一般只能维持 2~5 个小时。不幸的是,电池容量在未来 5 年内只能提高 30%,这远远低于同期 CPU 速度和存储容量的发展速度。因此,尽管已有不少节能技术应用于移动计算机,但电池容量问题仍将在长时间内存在。

6)可靠性。无线网络与固定网络相比,可靠性较低,更容易受到干扰而出现网络故障;此外,移动计算机由于其便携性和工作环境,也带来潜在的不安全因素,如碰撞、磁场干扰、遗失、失窃等。

7)规模。许多移动环境,如公共交通信息系统,都要求系统同时支持大量的移动用户并发访问,这就要求移动计算系统必须具有比传统客户/服务器以及分布式系统高得多的可扩展性。

其次是有大量来源于移动客户机的移动事务^[5~7]:

1)移动事务可能需要将它们的计算分解为操作集,其中一些在移动主机上运行而另一些在固定主机上运行。

2)移动事务需要基站的支持来进行通讯和计算。

3)当移动主机从一个单元移动到另一个单元,事务的状态及其进展也必须透明地移动。

4)许多移动事务是长寿事务。

再者,移动实时数据库系统事务具有定时性^[8,10~13]。定时性则包含两方面的含义:

定时限制。事务的执行具有显式的定时限制,典型为截止期。事务(或系统)的正确性不仅依赖于其逻辑结果的正确性,还依赖于逻辑结果产生的时间。这是由于系统需要随时跟踪外部环境而引起的。

定时正确性。事务能按合适的时间要求正确执行。这是由于要求数据对于系统的各种决策活动随时有效所引起的,它要求权衡定时限制与数据一致性等多方面因素,提供合适的调度算法。

按截止期的性质,即超截止期对系统带来的影响,事务可分为如下三类:

- 硬(截止期/实时)事务——超截止期会导致恶果(价值函数取大且可能不断增加的负值)。它对应于安全紧急性活动。

- 固(截止期/实时)事务——一旦到达截止时间,其价值立即降为零,此后固定为零(也不会为负)。

- 软(截止期/实时)事务——超截止期仍有一定的价值,且价值不断下降,直到某一时刻(称为最终有效时间)降到零,此后保持为零(不会为负)。

图 2.1 中(a)~(c)分别给出描绘硬、固、软实时事务的典型例子。其中 v, t 两坐标轴分别为价值函数与时间, d 为截止期(deadline), e 为“最终有效时间”。 r 为放行(release)时间,当 $t < r$ 时, $v(t) = 0$ 意思是在事务未准备好以前启动是无价值的。这种分类有利于设计事务优先级分派及调度策略。显然,固实时事务是软实时事务在最终有效时间与截止时间重合情况的特例,而移动实时数据库中大多数事务属于软实时事务和固实时事务。

最后,事务的一致性需求多样化^[1,2]。

数据库并非总是处于内部且外部一致性状态。数据库中的对象可处于两种一致性组合的四种状态之一。一个对象可能是内部一致的但非外部一致,这意味着它的值是过期的。一对象可外部一致但内部不一致,即其值是新的,但某些内部约束不满足。一个对象可同时内外一致或内外不一致。

每一事务均要求数据库是内部和/或外部一致的。换句话说,仅那些满足预定义的一致性要求的数据对象可被事务使用。比如,打印月财务报表的事务要求一个内部一致性视图;而在紧急状态下作出决策关闭核反应堆的事务更强调外部一致视图。许多应用中,放松可串行化而强调外部一致性会更有效且更易被接受。基于不同一致性需求的事务分类如下:

- 强内部/无外部一致性:只须控制事务并发执行时其动作的交替以保证数据库内部一致性。移动实时数据库事务具有截止期约束,若允许事务完成时间小于事务执行所需时间,系统有两种选择:abort 该事务且 undo 它对数据库的全部影响;或允许该事务产生一个不精确结果。为了提高性能可以采取后者。可是,不管哪一种,任何事务都不允许访问内部不一致数据。换言之,任何时候都必须保证数据库的一致性,事务调度遵循可串行化原则。总之,强调内部一致性而对外部一致性不予考虑。

- 折衷内部/外部一致性:许多应用允许暂时的内部不一致性,这可为维护外部一致性提供便利,从而取得内部、外部一致性的折衷。

- 无内部/强外部一致性:有些移动实时数据库事务必须使用绝对及时数据,这时候维护外部一致性是最重要的。可以不顾内部一致性的理由在于:应用语义上其不重要;有时可用补偿事务恢复数据库的内部一致性;实时数据应反映现实世界,使用了外部一致性数据的事务可视为现实世界中当前状态的全部数据进入数据库而数据库达到一致状态后是可串行化的。

这些特征使得传统事务模型无法适用于移动实时数据库,需要研究新的事务模型^[9]。

2 移动实时事务模型

本文提出一种通用的移动实时事务模型。为了适应移动

实时数据库的特点,它归纳了现有移动事务模型和实时事务模型的基本思想^[6~8,14],把事务分解为更小的片段。

定义 1 一个移动实时数据库事务(TXN)就是一个三元组: $TXN = \langle s, \prec, c \rangle$, 其中, S 为构成该事务的一组片段。各片段可在任意单元中运行,多个片段可在同一单元中运行,但单独一个片段只能在一个单元中运行即片段不能跨单元运行。每个片段中的操作仅涉及维护内部一致性或外部一致性,不会同时涉及内部和外部一致性。

$$\langle s \rangle ::= \langle \langle \text{SEGMENT} \rangle, \{ \langle \text{SEGMENT} \rangle \} \rangle$$

$$\langle \text{SEGMENT} \rangle ::= \langle \langle \text{SEGMENT} \rangle, \{ \langle \text{SEGMENT} \rangle \} \rangle$$

$$\langle \text{SEGMENT} \rangle = \langle P, R \rangle$$

$$\langle P \rangle ::= \langle \langle OP \rangle, \{ \langle OP \rangle \} \rangle$$

$$\langle OP \rangle ::= \langle OB-OP \rangle | \langle TM-OP \rangle$$

其中 $OB-OP$ 和 $TM-OP$ 分别是数据库对象操作(如读、写等)和事务管理操作(如事务的开始、提交等)。 R 为执行事务片段所需系统资源,包括数据资源集、CPU、缓冲区、磁盘、支持基站等。

- $\prec$ 为 S 上的一个偏序(时序)。

- C 为该事务所维持的限制。

定义 2 移动实时数据库中的每个原子操作可记做: $\sigma_i = A_i: [x]$ 。其中, i 表示事务标识; s 表示事务片段标识; j 表示节点标识; k 表示单元标识; A 表示操作类型:读或写; x 表示操作的数据对象。

定义 3 移动实时数据库事务集 T 的一个执行经历 h 是一个四元组: $h = (D, T, \Sigma, \pi)$ 。其中, D 表示该移动实时数据库; T 表示该事务集; Σ 表示原子操作集; π 表示 Σ 上的一个偏序。

定义 4 移动实时数据库事务集 T 的执行经历 h 在节点 j 的投影定义为: $h^{j,k} = (D, T, \Sigma^{j,k}, \pi^{j,k})$, 其中, $\Sigma^{j,k} = \{ \sigma_{i,k} | \forall k \in K(\sigma_{i,k} \in \Sigma, \sigma_{i,k} \text{ 在节点 } j \text{ 处理}) \}$ 。

定义 5 移动实时数据库事务集 T 的执行经历 h 在单元 k 的投影定义为: $h^{j,k} = (D, T, \Sigma^{j,k}, \pi^{j,k})$, 其中, $\Sigma^{j,k} = \{ \sigma_{i,k} | \forall j \in J(\sigma_{i,k} \in \Sigma, \sigma_{i,k} \text{ 在单元 } k \text{ 中处理}) \}$ 。

定义 6 移动实时数据库事务集 T 的执行经历 h 在单元 k 的投影定义为: $h^{j,k} = (D, T, \Sigma^{j,k}, \pi^{j,k})$, 其中, $\Sigma^{j,k} = \{ \sigma_{i,k} | \sigma_{i,k} \in \Sigma, \sigma_{i,k} \text{ 在节点 } j \text{ 单元 } k \text{ 中处理} \}$ 。

定义 7 移动实时数据库事务集 T 的执行经历 h 对事务 i 的投影定义为: $h^{j,k} = (D, T, \Sigma^{j,k}, \pi^{j,k})$, 其中, $\Sigma^{j,k} = \{ \sigma_{i,k} | \forall s(\sigma_{i,k} \in \Sigma \wedge \sigma_{i,k} \in T_i) \}$ 。

定义 8 移动实时数据库事务集 T 的执行经历 h 对事务片段 s 的投影定义为: $h^{j,k} = (D, T, \Sigma^{j,k}, \pi^{j,k})$, 其中, $\Sigma^{j,k} = \{ \sigma_{i,k} | \forall i(\sigma_{i,k} \in \Sigma \wedge \sigma_{i,k} \in T_{i,s}) \}$ 。

定义 9 如果两个执行经历能够将同一个初始状态转换为相同的结束状态,则说这两个状态等价,记为: $h_1 \equiv h_2$ 。任意一个移动实时数据库的执行经历 h 是可串行化的当且仅当至少存在一个串行经历 h_s 满足: $h^{j,k} \equiv h_s^{j,k}$ 。

小结 传统的具有 ACID 特性的原子事务模型不适合移动实时应用环境。针对移动实时应用的特点与要求,本文提出了一种分片段事务模型,并给出了形式化定义。相对于传统的“原子操作序列”的事务定义,一个移动实时数据库事务包含一组片段。分片段事务模型具有很多优点,很适合移动实时应用环境:

- 1) 可有效支持事务执行的移动性、分布性。

(下转第 117 页)

指针三种方式进行,具有较强的灵活性与方便性。

- 信息结点结构:层次结构中的每个结点都代表一个信息实体,把它称为一个信息结点,每一个结点(包括树根结点)都可以为空或存放信息,也可以存放文本、图片、数据库表等各类信息。

- 索引结构 为加快信息库的访问速度,提高信息的利用效率,可以在信息库中加入索引机制。

- 信息实体运算 通过信息流模型的建立,可以对各种类型和深度的信息实体进行运算。

第三层是信息库和推理机,信息管理的实现,需要建立完善的信息库系统,建立如下的信息分类:

- 业务对象信息 包括业务的内涵、外延,事务的类型、属性、状态,业务的安全密级,业务流程等。

- 角色信息 包括角色的内涵、外延,角色的属性,角色的操作权限,角色间的上下级关系,角色的授权,角色中的信息等。

- 业务对象流信息 角色在处理各业务过程中的记录,各类业务对象的处理程序。

- 业务对象涉及到的相关业务背景信息 与处理的业务对象密切相关的行业及部门的背景信息。

- 基本背景信息 区域背景、资源、环境、法律、法规、地方及国家的社会经济信息。

在信息推理中,选择信息的过程称为控制策略,一个控制策略由两部分组成:一是推理方法、一是搜索方法。其中推理方法包括正向推理、反向推理和混合推理;搜索方法分为非启发式搜索方法和启发式搜索方法。非启发式搜索方法包括广度优先搜索法、深度优先搜索法,该类方法的特点是:通用性强,但搜索效率低;启发式搜索方法包括局部择优搜索法、全局择优搜索法,该类方法的特点是可以加快搜索速度,但要反复尝试选择最合适的评价函数,因此实时性较差。根据办公信息流处理的具体特点,我们的设计拟选择正向推理、非启发式搜索策略。程序中设计的谓词和规则用于搜索信息库,并跟踪记录信息库对象和用户输入的值,以便达到推理的目的。对于不精确的信息推理结果的可靠性附加一个可靠性的估价

值—可信度因子。

第四层主要是关于信息的获取,本系统信息的获取可以从两方面来考虑,一种信息来自于系统。综合运用索引、抽取、匹配、数据挖掘等来获取整个系统中的相关决策支持信息。另一种来自于与系统交互的办公人员。

结束语 通过将信息表示成实体,得到了可处理的信息流模型,从理论分析和初步应用来看,有如下几个特点:

1. 该模型的形成源于一般的哲学思想方法,不依赖于某一领域,具有较强的通用性。

2. 采用元模型来组合成整体模型,使模型可拆分和组合,因此在应用上具有很大的灵活性。

3. 通过将角色引入到模型中,简化了组织中信息流建模的复杂性。

通过以上的模型,我们与其它软件有限公司合作开展“十五”国家科技攻关计划课题—《以信息管理为核心的办公信息流处理软件及工程实现》课题,目前该项目在顺利进展中。

参考文献

- 1 Forza C, Salvador F. Information flows for high-performance manufacturing[J]. International Journal of Production Economics, 2001,70(1): 21~36
- 2 Ahlswede R, Cai N. Network Information Flow[J]. Information Theory, 2000,46 (4): 1204 ~1216
- 3 Klyubin A S, Polani D. Organization of the information flow in the perception-action loop of evolved agents[C]. Evolvable Hardware, In: Proce. 2004 NASA/DoD Conf. on, 2004, 24-26;177 ~180
- 4 Alghathbar K, Wijesekera D. Analyzing information flow control policies in requirements engineering[J]. Policies for Distributed Systems and Networks. In: POLICY 2004. Proc. 2004. 193 ~196
- 5 Samarati P, Bertino E. Information flow control in object-oriented systems[J]. Knowledge and Data Engineering, 1997,9(4): 524~538
- 6 Uddin Ahmed A, Cheng Z, Saito S. Information flow model and estimations for services on the Internet[C]. Advanced Information Networking and Applications. In: AINA 2004. 18th Intl. Conf. 2004,1: 499~505
- 7 于淼,王延章. 公文流转过程中活动的分解和管理[J]. 计算机工程, 2003,29(22):18~20
- 4 何新贵,唐常杰,李霖,刘云生. 特种数据库技术. 北京:科学出版社,2000
- 5 Barbara D. Mobile computing and databases-A survey. IEEE Transactions on Knowledge and Data Engineering, 1999, 11(1): 108~117
- 6 Pitoura E, Bhargava B. Revising transaction concepts for mobile computing. In: Pro. of Workshop on Mobile Computing Systems and Applications, 1995. 164 ~168
- 7 Dunham M H, Helal A, Balakrishnan S. A mobile transaction model that captures both the data and movement behavior. Mobile Networks and Applications, 2003, 2: 149~162
- 8 李国徽. 主动实时嵌套事务及其处理: [博士学位论文]. 华中理工大学,1999. 6
- 9 党德鹏. 移动实时数据库的一致性维护: [博士学位论文]. 华中科技大学,2003. 3
- 10 刘云生. 现代数据库技术. 国防工业出版社,2001
- 11 Ramamitham K. Real-time databases. Int. Journal of Distributed and Parallel Databases, 1993, 1(2):199~226
- 12 Stankovic J A, Son S H, Hansson J. Misconceptions about real-time databases. Computer, 1999, 32(6): 29~37
- 13 Haritsa J R, Ramamitham K. Real-time databases in the new millennium. Real-time systems, 2000, 19(3): 205~289
- 14 Bhargava B. Concurrency Control in Database Systems,1999, 11 (1): 3~17
- 1 Kayan E, Ulusoy O. Real-time transaction management in mobile computing systems. In: Proc. of 6th Intl. Conf. on Database Systems for Advanced Applications, 2002. 127 ~134
- 2 Ulusoy O. Real-time data management for mobile computing. In: International Workshop on Issues and Applications of Database Technology (IADT'98). Berlin, Germany, July 1999. 233~240
- 3 贾焰,李霖,等. 分布式数据库技术. 国防工业出版社, 2000. 7

参考文献

(上接第 108 页)

- 2) 可有效支持长寿事务,因事务被划分成片段而使各自截止期缩短,再经一定的动态优先级分派和调度策略,从而可大大改善长寿事务满足其截止期的机会,并且当发生故障时,可更精细地确定恢复的范围,而不至于浪费大量已完成的工作,这不但使这些长寿事务能按时完成,也有利于其它事务及系统的实时性的实现。

- 3) 可很好地适应频繁断接。

- 4) 允许一个事务内部的并发执行,有利于使最紧急的事务被及时处理。

- 5) 为灵活地维护移动实时数据库的一致性奠定了基础。