

移动环境下实时事务处理*)

李国徽 杨进才 王洪亚

(华中科技大学计算机学院 武汉430074)

Real-time Transaction Processing in a Mobile Environment

LI Guo-Hui YANG Jin-Cai WANG Hong-Ya

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

EMAIL: guohuili@public.wh.hb.cn

Abstract With the advances in mobile computing technology, there is an increasing demand for processing real-time transactions in a mobile environment. This paper proposes a multi-version data model. Based on this data model, we present a relative timing correctness criterion for real-time transactions in a mobile environment. As long as the data staleness is in a controlled extent, it is unnecessary for all the accessed data are valid when a real-time transaction commits. Also this paper gives a transaction execution model based on the image transaction concept. All these strategies not only decrease the cost for the transaction concurrency control, but also increase the possibility to meet mobile real-time transactions' deadlines by sacrificing the relatively transaction timing correctness. This is in accordance with the real-time transaction property that a relatively correct and in-time result is more preferable than an absolutely correct but obsolete result.

Keywords Mobile real-time database, Mobile real-time transaction, Data validity interval, Absolute timing correctness, Relative timing correctness

1 引言

实时数据库就是其中的事务和数据都可能有时限限制的数据库。近些年来,越来越多的研究人员对于移动实时数据库系统(Mobile Real-Time Database System; MRTDBS)中高效的事务处理方法进行了研究,一般认为移动实时数据库就是移动环境(如 GSM 网络和无线局域网)所支持的实时数据库系统^[1,2]。对于实时事务来说,其完成时间一般有相应的时间约束^[3],满足事务截止时间约束往往比事务逻辑完全正确更重要,例如对于股票查询的延迟响应会导致失去很好的交易机会。为了满足事务的截止时间约束,人们提出了不同的事务调度技术和并发控制协议。

移动实时数据库系统由移动客户机和固定主机组成,移动客户机通过无线网络和固定主机通信而固定主机间通过高速有线网络连接。移动客户机提交移动实时事务访问存储在固定主机和移动主机上的数据。

无线网络给 MRTDBS 带来了许多新的问题。与有线网络相比,无线网络传输速度更慢并且相对不可靠;而且由于无线通信容易受到外界环境的影响,还会经常出现移动客户机的断接现象。

这些固有的限制使得 MRTDBS 所使用的技术和分布式实时数据库系统有很大的不同。在移动(无线)环境下,满足事务截止期限变得更加困难,不同优先级事务数据冲突的解决代价更高,例如采用高优先级两段锁协议这样的重启事务方案时^[4],由于事务访问的数据可能分布在移动客户机和固定主机上,所以事务重启会花费很多的时间,当数据的值频繁更新并且这种更新发源于移动客户机时,保证事务的绝对时间

正确性变得更加困难。总之,移动环境的固有特点使得移动环境下的实时事务处理变得更加困难。本文提出了一种多版本的实时数据模型,基于此定义了实时事务的相对时间正确标准,并给出了移动实时事务基于“影子”事务的执行模型。

2 系统模型

一个典型的 MRTDBS 包括移动客户机(MHs)、固定主机(FHs)、移动支持基站(MSSs)、固定网络和移动网络,如图1所示。FHs 通过固定网络相连,整个系统地理上被划分为一些互连的子单元(Cell),每个子单元内有一个 MSS 负责与该单元内的 MHs 通信。一个 MH 通过本单元的 MSS 与固定主机进行通信。

假定系统中有两类数据库:本地数据库(LDB)和备份数据库(BDB)。本地数据库维护系统中数据项的主拷贝,而其中每个数据项又在备份数据库中有相应的备份。本地数据库分布在 MHs 和 FHs 中,而 BDB 仅仅分布在固定主机上。备份数据库中数据项应与本地数据库中数据同步,而这种同步会受到无线网络断接的影响。

使用备份数据库有两个目的:一是当移动主机上的本地数据库由于网络断接访问不到时,只要该版本在用户时限要求之内,事务仍可使用备份数据库中该数据历史版本;二是可以提高系统的可靠性。

数据库中数据分两类:实时数据和非实时数据。实时数据的有效性随着时间的改变而改变,并且其值的变化非常快,典型的实时数据是移动目标的位置信息。非实时数据不经常变化,通常记录一些较少变化的信息,如客户的地址。

按照数据值改变的方式,我们把实时数据又分为基数据

*)论文受国家青年自然科学基金(6020301)项目资助。李国徽 博士,副教授。杨进才 博士研究生,副教授。王洪亚 博士研究生。

(base items)和派生数据(derived items)两大类。基数据用于记录外部环境中对象的实时状态,例如股票的最新交易价格等。为保证基数据的值与对应的外部对象状态的一致性,必须对基数据进行不断更新来捕捉外部对象的变化。根据基数据值如何改变我们进一步将其分为两类:一类是数据值以固定周期更新,我们称之为周期性(periodic)基数据,例如雷达对记录飞机位置的数据项每秒更新五十次等。如果更新间隔是不规则的,我们称之为零星(sporadic)基数据,例如股票交易系统最后一次交易的价格等。

派生数据来源于基数据,如股票综合指数。本文中我们假设派生数据来源于固定的基数据集(这与现实是符合的)且由移动实时事务(Mobile Real-Time Transaction, MRTT)生成,其中 MRTT 指的是移动客户机提交的实时事务。

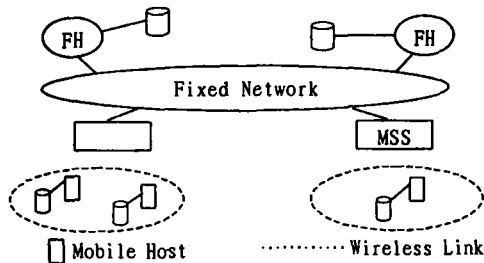


图1 移动实时数据库系统结构

3 数据模型及事务时间正确性

为了实现 MRTT 的有效处理,我们提出了一个多版本实时数据模型。在该模型中,每个数据项有多个版本存在于系统中,这样做的目的有两个:一是当执行事务无法访问到最新版的数据时,可以利用数据旧版本来实现事务的相对时间正确性。移动环境中网络断接是经常出现的,如果执行事务所需的数据在一个处于断接状态的移动客户机上, MRTT 就会被挂起而超过了截止期限,为使该事务及时完成,可利用老版本数据进行替代。二是在版本多本实时数据模型中可使并发控制代价极大地减少,很多实时事务的并发控制协议采用事务重启来解决不同优先级事务间数据冲突的问题,这在移动环境中会造成非常大的代价,而多版本实时数据模型可以极大地降低事务处理并发控制的代价。

3.1 多版本实时数据模型

多版本实时数据模型中,对应每个数据对象有一个版本序列,例如 x_i 就是数据对象 x 的第 i 个版本。一个数据项的新版本在移动客户机上生成后先被存储在 LDB 上,然后会被复制到固定主机上,复制成功后,该数据对象的旧版本将会从移动主机的 LDB 中移出以减少内存和占用(这考虑到移动客户机存储空间的有限性)。对一个数据对象 x 来说, LDB 上的版本和 BDB 中的版本共同组成了到当前时间为止所有数据版本的集合。

每个数据版本与一有效期(Validity Interval, VI)相联,例如 x_i 的有效期可表示为 $VI(x_i) = [LTB(x_i), UTB(x_i)]$,其中 LTB 和 UTB 分别表示 x_i 的时间下界(low time bound)和上界(upper time bound),有效期表示了该数据版本在该段时间内是有效的。

对于基数据而言,为了捕捉外部对象的实时状态而对数据进行的更新产生了新版本。更新发生时,数据项被赋予一个相应的时间戳,更新完成后时间戳被设置为新数据版本的

LTB , 而 UTB 暂时设置为无穷大,直到下一次更新发生。同时上一个数据版本的 UTB 将被设置为该次更新时的时间戳,因此任意两个基数据项的有效期不相互重叠,且后继版本的 VI 紧接着上一个版本的 VI 。

派生数据的新版本是由 MRTT 产生的, MRTT 读取基数据项并由这些基数据项生成派生数据项的一个新值。因此为了保持时间一致性,派生数据的有效期应由相关基数据项的 VI 决定。

假定一个移动实时事务先读取一些实时数据,然后基于这些读取的实时数据经过某种计算(推导)而得出一个(或一些)派生(实时)数据对象。

定义1 移动实时事务 T 生成的派生数据项 dx_i 的有效期定义为 T 读取的所有数据项的有效期交集。

$$VI(dx_i) = \bigcap \{VI(x_j) \mid x_j \in DS(T)\}$$

其中 $DS(T)$ 是应用事务 T 所读取的数据集。

在图2中有三个数据对象, O_1 是一个基数据对象且以固定的周期更新而产生新的版本, O_2 是一个零星基数据对象, O_3 是一派生数据对象,可以看出 O_3 版本之间存在间隔。

定理1 对每个实时数据对象,其两个不同版本的有效期互不重叠。

证明:1 如前所述,一个基数据对象(不管是周期对象还是零星对象)的不同版本,有效期是不会互相重叠的。

2 对于派生数据对象 x , 我们假定 x 由事务 T 生成,生成时读取了基数据项 u_1, v_1, \dots 。假设 x 的两个不同数据版本 x_{k1}, x_{k2} 的有效期有重叠,即 $VI(x_{k1}) \cap VI(x_{k2}) \neq \emptyset$, 那么至少存在有这样一个基数据对象 u , 它的两个版本 u_{11} 和 u_{12} 被事务 T 用于生成派生数据对象 x 的两个不同版本 x_{k1} 和 x_{k2} , 假设 x_{k1} 派生自 u_{11} , x_{k2} 派生自 u_{12} , 根据派生数据项有效期的定义有:

$$VI(x_{k1}) \subseteq VI(u_{11}), VI(x_{k2}) \subseteq VI(u_{12}), VI(x_{k1}) \cap VI(x_{k2}) \neq \emptyset \Rightarrow VI(u_{11}) \cap VI(u_{12}) \neq \emptyset$$

所得结论与一个基数据项两个版本的有效期不重叠冲突,所以假设错误,因此任意派生数据对象的两个不同版本的有效期不会重叠。

综上所述所有时变数据项的两上不同版本的有效期都不会重叠,定理得证。

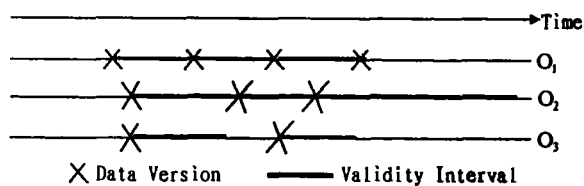


图2 不同数据对象的有效期

3.2 事务相对时间正确性

无线环境的固有特点使得在 MRTDBS 中实现实时事务的绝对时间正确性非常困难,为此我们提出一种相对时间正确性标准。

定义2 设事务 T 提交时刻为 ct , 其读取的数据集为 R , 如果 $\forall O \in R, LTB(O) \leq ct \leq UTB(O)$, 我们就说该事务是绝对时间正确的。

对于满足绝对时间正确性的事务,它所访问的数据都应该是该数据项的最新版本,也就是说 UTB 是该数据对象的所有版本中最大的。

在移动环境中,实现事务绝对时间正确性的代价很大,例如,一个 MRTT 访问了一个实时数据对象的最新版本,然后向一个移动客户机发送请求要访问另一个数据项的最新版本,由于网络断接,后者可能要经过一段延迟才能得到,在这期间,可能又生成了前面访问数据对象的新版本,为保持绝对时间一致性,事务必须重启。考虑到这些情况,我们对绝对时间一致性做了一定程度的弱化,只要求事务提交时所有访问数据的版本“足够新”。

定义3 设事务 T 启动时间为 st ,提交时间为 ct ,用户所允许的“过期度”(staleness)为 Δ ,读取数据集为 R 的事务如果

$$\exists t, (st - \Delta) \leq t \leq ct, \forall O \in R, LTB(O) \leq t \leq UTB(O).$$

则称该事务是相对时间正确的。

对相对时间正确的事务,并不要求事务提交时事务所读取的所有数据项都是有效的,而只要求所有数据项在 $(st - \Delta)$ 之后的某个时间点上有效的。

从上面定义可以看出,相对时间正确的事务不允许读取在 $(st - \Delta)$ 之前某个时间点无效的数据。

4 移动实时事务处理

4.1 移动实时事务

定义4 一个移动实时事务的形式化定义为一个4元组:

$$T = (OP, D, <_r, \Delta)$$

其中 OP 是事务的操作集;D 是事务的截止期; $<_r$ 是事务中操作的一个偏序关系; Δ 是事务能够容忍的最大数据陈旧度(staleness),数据的陈旧度定义为当前时间与该数据的有效期上界的差值。如果该差值小于 Δ ,那么事务就可以使用该数据。

一个 MRTT 的执行可能需要访问固定主机和移动客户机上的数据,下面我们将介绍基于“影子”事务的事务执行模型,使得在移动环境中事务截止期和时间正确性要求能够更好地得到满足。

4.2 基于影子事务的事务执行模型

考虑到在移动网络环境中,移动客户机上数据访问延迟是事务无法在截止期前提交的一个重要原因。影子事务模型的主要目的是减少事务数据访问延迟。

一个移动实时事务 T 在由移动客户机生成之后,提交到该移动客户机当前所在子单元对应的基站(一个固定主机)上,当事务被基站接受后,在基站上生成该事务对应的影子事务(Image Transaction, IT),生成的影子事务由一些子影子事务(Sub Image Transaction, SIT)构成。每一 SIT 负责从各种不同的移动客户机或固定主机上预取 T 执行所需的所有数据。

定义5 一个移动实时事务 T 的影子事务 IT(T)定义成一个3元组:

$$IT(T) = (P', D, \Delta)$$

其中 $P' = \{p_i | p_i \text{ 是事务 T 中的数据访问过程 } p_i \text{ 的数据预取}$

过程};D 是映像事务的截止期; Δ 是影子事务所能容忍的数据最大陈旧度。

影子事务 IT(T)的截止期和陈旧度分别是 T 的截止期限和陈旧度, Δ 用来决定影子事务将哪些版本从固定主机或移动客户机预取过来。我们用截止期最早最优先(earliest deadline first, EDF)方法确定 MRTT 的优先级,为照顾数据预取过程,所有影子事务被赋予比 MRTT 更高的优先级,而不同影子事务之间的调度方式是该映像事务的截止期限越小则优先级越高。需要注意的是影子事务的执行是并行的,所以预取过程并不具备原事务中操作的顺序关系。

对于 MRTT 中的每一个操作 p, p 访问的读数据集用 DS(p)表示,为减少 p 操作的数据访问延迟, $\forall p \in MRTT$,在 p 执行以前 DS(p)中的所有数据都必须已预取到该移动客户机对应基站上,一个影子事务对应一个包含操作 p_i 的集合,其中对每个 $p_i \in T, p_i$ 具有下列的属性:

$$(1) DS(p'_i) \supseteq DS(p_i)$$

(2) p_i 是一个只读操作

在影子事务从相应的移动客户机或固定主机上预取到所有的数据之后,事务 T 开始执行,此时 T 所需的所有数据都已经能快速访问到了。

结论 移动环境下的实时事务处理近些年来受到了越来越多的关注,由于无线网络固有的特点,在 MRTDBS 中一个事务很难既满足绝对时间正确性要求又在截止期前正确提交。在本文中,我们考虑到无线网络的特性如经常断接及低带宽等因素,重新定义了相对时间正确性标准,并采用多版本实时数据模型和基于影子事务的执行模型来实现相对时间正确性。

在我们的事务执行模型中,事务从移动客户机生成后,被传输到它对应的基站上,然后被预分析并生成一个影子事务,影子事务又包含了多个子影子事务,每个子事务包含一个或多个读操作集,将事务 T 所需的数据预取到基站上。通过数据预取,数据访问的延迟被大大地降低,从而更好地满足了事务的截止期要求。

参考文献

- 1 Kayan E, Ulusoy O. An Evaluation of Real-Time Transaction Management Issues in Mobile Database Systems. *The Computer Journal*, 1999, 42(6): 501~510
- 2 Ulusoy O. Real-Time Data Management for Mobile Computing. In: Proc. of Intl. Workshop on Issues and Applications of Database Technology (IADT'98), Berlin, Germany, 1998. 233~240
- 3 Abbott R J, Garcia-Molina H. Scheduling Real-Time Transactions: A Performance Evaluation. *ACM Transactions on Database Systems*, 1992, 17(3): 513~560
- 4 Lee V C S, Lam K, Kao B. Priority Scheduling of Transactions in Distributed Real-time Databases. *Journal of Real-time Systems*, 1998, 15(1): 31~36
- 5 Abowd G, Allen R, Garlan D. Using style to understand descriptions of software architecture. *ACM SIGSOFT Software Engineering Notes*, 1993, 18(5): 9~20
- 6 周莹新,艾波. 软件体系结构建模研究. *软件学报*, 1998, 9(11): 866~872
- 7 焦文品,史忠植. 用 XYZ/E 形式化体系结构风格. *软件学报*, 2000, 11(3): 410~415
- 8 骆华俊,唐稚松,郑建丹. 可视化体系结构描述语言 XYZ/ADL. *软件学报*, 2000, 11(8): 1024~1029
- 9 唐稚松,等. 时序逻辑程序设计与软件工程(上册). 北京:科学出版社, 1999
- 10 Spriverly J. *The Notation: A Reference Manual*. Englewood Cliffs: Prentice Hall, 1989
- 11 Allen R, Garlan D. Formalizing architectural connection. In: Proc. of the 16th Intl. Conf. on Software Engineering. May 1994. 71~80
- 12 Magee J, Kramer J. Dynamic structure in software architectures. *ACM SIGSOFT Software Engineering Notes*, 1996, 21(6): 3~14

(上接第3页)