

支持断接的移动实时事务调度^{*}

刘云生 王丽娜 廖国琼

(华中科技大学计算机学院 武汉430074)

摘要 在移动计算环境中,事务移动性和频繁断接使得传统实时事务调度机制不足以支持移动实时事务的执行。该文给出了一个支持无线网络频繁断接的软实时事务调度协议 SSRTD。该协议是基于事务具有扩展截止期和结果相似性进行设计的,并可根据移动主机的连接状况决定事务执行结果的发送时间。通过性能测试,表明所提出的调度策略能提高移动实时事务的成功率。

关键词 移动数据库,实时数据库,事务调度,扩展截止期

Scheduling Strategies of Real-Time Transactions with Disconnections in Mobile Computing Environment

LIU Yun-Sheng WANG Li-Na LIAO Guo-Qiong

(Computer Institute of Huazhong University of Science and Technology, Wuhan 430074)

Abstract In mobile computing environment, for transaction's mobility and frequent disconnection, the traditional real-time scheduling mechanisms are not suitable for the execution of mobile real-time transactions. In this paper a scheduling protocol SSRTD for soft real-time transactions is proposed, which is designed based on the extended deadline and result similarity. The transmitting time of transaction's results from base servers to mobile hosts is determined by the connection status of mobile hosts. According to the performance tests, the suggested scheduling strategy can improve the success ratio of real-time transactions.

Keywords Mobile database, Real-time database, Transaction scheduling, Extended deadline

1 引言

移动实时数据库系统 MRTDBS (Mobile Real-Time Database System)是指运行在移动计算环境中的实时数据库系统。但由于无线网络的频繁断接、低速、不可靠和不可预测等特点,使得 MRTDBS 中的事务特性不同于传统分布式数据库和一般的实时数据库系统 RTDBS (Real-Time Database System)中的事务,如事务随移动主机的移动而移动;事务多为长事务;无线网络的通信延迟使得事务截止期更难满足等。因此,移动计算环境中的实时事务大多为软截止期事务,即允许事务有限的超截止期,而不会对系统造成危害。

事务调度是实时数据库事务管理的重要方面。对于移动实时事务而言,除需考虑一般实时事务调度需要考虑的因素(如截止期、价值、关键性等)外,还需考虑移动环境的特殊性。本文着重考虑支持无线网络频繁断接的实时事务调度方法,其目标是保证移动实时事务执行的定时性和连续性。

目前,已有一些针对移动分布式实时数据库的研究。文[1]给出了两种移动实时事务执行模型 ESFH(事务在固定主机上执行)和 ESMH(事务在移动主机上执行)。文[2]提出了一种支持移动分布式实时事务的并发控制机制 DHP-2PL (Distributed High Priority Two-Phase Locking),但它是基于平坦事务模型。文[3]研究了一种移动实时嵌套事务模型下、

结合优先级夭折与优先级继承的并发控制策略 PAI-2PL。文[4]基于功能替代的思想给出了一个支持移动分布式实时嵌套事务的三阶段提交协议 3LRTC,详细地讨论了该协议的对场地故障及断接故障的恢复处理协议。然而,有关移动实时数据库中事务调度的研究很少。文[5]提出了一种根据事务的“扩展截止期”(extended deadline)进行移动实时事务调度的方法 DT-protocol。其“扩展截止期”是由事务所处无线网络的通信质量而定:通信质量高,则事务“扩展截止期”相对较短;通信质量低,则事务“扩展截止期”相对较长。但该方法只是单纯考虑无线网络频繁断接而增加事务的执行时间,这与实时事务原有的定时特性和截止期概念并不相符,并且无线网络通信质量本身是动态变化的,可能时好时坏(由事务当前所在无线网络决定),因此可能出现事务两次调度执行而“扩展截止期”不一致的现象。

本文的 SSRTD(Scheduling Strategy of Real-time Transactions with Disconnections)协议也采用“扩展截止期”方法进行移动实时事务调度。但本文“扩展截止期”是根据软实时事务本身的截止期而定,是固定不变的,同时该方法它也能够有效支持无线网络的频繁断接。

2 移动实时数据库系统模型

图1是移动实时数据库系统的体系结构。移动计算环境中

^{*}项目资助:国家自然科学基金(编号60073045),国防预研基金(编号00J15.3.3.JW0529);中国博士后科学基金项目(2003034482),华中科技大学博士后科学基金项目(2003021)。刘云生 教授,博士生导师,研究方向为现代数据库理论与技术及其集成实现、数据库与信息系统开发、实时数据库工程软件方法与软件开发。王丽娜 硕士研究生,研究方向为现代数据库技术、移动、实时数据库系统等。廖国琼 博士,主要研究方向为现代(实时、主动、内存、移动、工程等非传统)数据库理论与技术及其集成实现、数据库与信息系统开发。

有两类主机:基站服务器 BS(Base Server)和移动主机 MH(Mobile Host)。BS 之间通过固定网络(Fixed Network)相互通信。BS 通过无线界面与位于它所负责的无线通信单元(Wireless Cell)内的 MH 通信。该系统中每个基站服务器和移动主机都与一个实时数据库管理系统 RTDBMS(Real-Time Database Management System)和一个实时数据库 RT-DB(Real-Time Database)相连。整个移动实时数据库由分布在移动环境下的所有 RTDB 组成。

假设位于 BS 中数据库为主数据库,且每个 MH 通过无线网络通信查询或者更新主数据库。进一步假定,无线网络断接发生频率高但持续时间短,且断接时间小于事务扩展截止期。

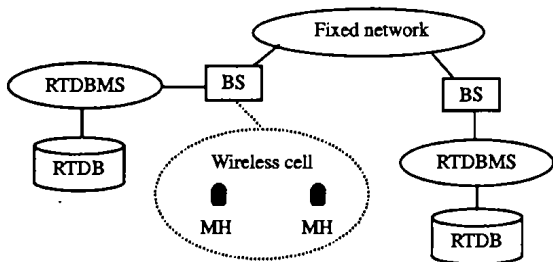


图1 移动实时数据库系统结构

3 扩展截止期和数据相似度

在实时数据库中,事务严格的 ACID 特性已得到放松,可采用放松的可串行化标准,允许暂时不一致^[6]。这样可减少事务之间互相阻塞,以提高事务成功率。但这些在移动实时数据库中还不够。由于无线网络固有的缺陷,如低速、有限带宽、频繁断接和高度不可靠等,事务很难在原始截止期前得到精确结果。为此我们给出“扩展截止期”和“结果相似性”的概念。

图2 为一个软实时事务示意图,即事务超截止期后仍有一定价值,但不断下降,直到某一时刻(最终有效期),其价值降为零,此后保持为零(不会为负)^[6]。

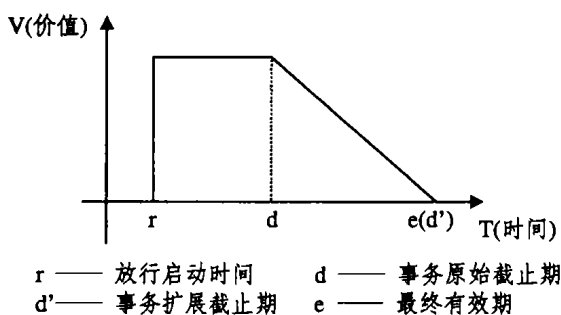


图2 软实时事务

扩展截止期是在事务原始截止期的基础上根据事务的价值计算的,其时间长度为从原始截止期到事务价值变为零,所以它与“最终有效期”一致。若按文[5]扩展截止期概念,当网络通信质量低(断接频率高),计算出的事务扩展截止期可能大于本文中计算出的扩展截止期,但此时事务可能没有任何价值;而当网络通信质量高(断接频率低)的情况下,计算出的事务扩展截止期可能小于本文中计算出的扩展截止期,导致不必要的事务夭折。

对于实时数据库而言,及时性有时比正确性更加重要,不太精确的近似值在一定范围内也是可以接受的。因此,我们

有:

定义1(数据相似性) 对一个数据对象 D , d_1 和 d_2 是 D 的两个值。若 d_1, d_2 之间的距离 $DIS(d_1, d_2) \leq \sigma$, 则 d_1 和 d_2 是相似的,记为 $d_1 \approx d_2$ 。其中 σ 是具体应用语义所要求的阈值。 $DIS(d_1, d_2) = |f(d_1) - f(d_2)|$, f 是将 D 的域值映射到数值空间的函数。

定义2(结果相似性) $RS_1(T)$ 和 $RS_2(T)$ 为事务 T 的两个结果集, $RS_1(T) = \{d_{i1} | 1 \leq i \leq n\}$, $RS_2(T) = \{d_{i2} | 1 \leq i \leq n\}$, $\forall i$, 若 $d_{i1} \approx d_{i2}$, 则 $RS_1(T)$ 和 $RS_2(T)$ 是相似的,记为 $RS_1(T) \approx RS_2(T)$ 。

不同事务可以具有不同的结果相似性需求。“结果相似性”使得事务在扩展截止期即将到达,但还未执行完的情况下可以接收一个不精确的结果,并且事务执行过程中可采用放松的可串行化标准,允许数据库暂时不一致,减少了事务夭折概率。因此,采用“扩展截止期”和“结果相似性”方法来调度移动实时数据库中的软实时事务,能保证更多事务可在其扩展截止期前成功完成。

4 移动软实时事务调度策略

4.1 计算发送执行结果时间

在移动计算环境中,移动事务通常由移动主机发起,交给 BS 执行后等待执行结果。在等待过程中,移动主机可能主动或意外断接,因此 BS 必须根据移动主机的连接情况、事务的“扩展截止期”和“结果相似性”等因素决定发送执行结果(精确或不精确)的合适时间。若 BS 估计断接将在它发送整个结果前发生,并且在断接期间,事务可能会错过扩展截止期,若此时事务结果满足相似性,那么在断接前可发送一个结果给 MH。另一种情况是,若 BS 估计网络稍后会重新连接,并且不会错过事务扩展截止期,则可稍后发送一个更加精确的结果给 MH。

因此,为确定 BS 发送执行结果的时间,每次一个新的通信连接建立时,要求 BS 估计在其扩展截止期前是否会发生断接。令:

T ——MH 发起并在 BS 上执行的移动事务

$D(T)$ —— T 的原始截止期

$D'(T)$ —— T 的扩展截止期

$SR(T)$ ——BS 向 MH 发送 T 执行结果的时间。BS 在此时必须发送事务结果,在这个时间之后,BS 与 MH 之间可能发生网络断接,事务可能会超过扩展截止期

t_c ——BS 与 MH 建立连接的时刻

$C(T)$ ——表示 T 是否已经提交(0表示未提交,1表示已提交)

$CDMin$ ——BS 与 MH 之间最小可能可靠通信持续时间

$DDMax$ ——BS 与 MH 之间最大可能断接持续时间

下面列出了三种情况下 $SR(T)$ 的确定方法:

(1) $CDMin < D'(T) - t_c < CDMin + DDMax$: 若网络断接在事务到达它的扩展截止期前发生,则事务可能难以在扩展截止期前执行完成。因此 $SR(T)$ 的最晚时间为 $t_c + CDMin$ 。在这个时刻,若结果满足相似性,BS 必须发送这个结果给 MH,否则很难保证无线网络在事务扩展截止期前会重新建立连接。

(2) $0 < D'(T) - t_c < CDMin$: 这表示网络断接在时间 D' 前不会发生。因此 $SR(T)$ 的最晚时间为 $D'(T)$, 就是事务的扩展截止期。在这个时刻,若结果满足相似性,BS 也必须发

送这个结果,否则事务就会由于超出扩展截止期而被夭折。

(3) $CDMin + DDMax < D'(T) - t_c$:这里又分两种情况,若网络断接之后通信不再建立,那么 $SR(T)$ 的最晚时间为 $D' - DDMax$;若网络断接之后通信又重新建立,则同(1)和(2)。

因此,在事务执行过程中,若当前时间等于 $SR(T)$,若 $C(T) = 0$ 且结果满足相似性,BS 可发送事务结果给 MH。

4.2 优先级分派

实时数据库中的事务是基于优先级来调度执行的。传统的优先级分派策略有最早放行最优先,截止期最早最优先等。移动实时数据库中要考虑频繁断接问题,因此 SSRTD 协议中事务的优先级是基于 SR 时间来分派,随着 SR 的变化而动态改变。此策略类似于截止期最早最优先,它使具有最早 SR 时间的事务优先级最高。每当 MH 与 BS 建立连接时,对于 MH 在 BS 上的事务,根据新的 SR 时间进行一次优先级调整,同时将 BS 上的接纳,就绪,运行队列按优先级重新排序。令:

$Pri(T)$ —— T 的优先级

优先级分派算法描述如下:

```

算法名称:ComputeTransPri
输入:D'(T)
输出: Pri(T)
算法:
tc = GetConnectionTime(T);
case D'(T) ∈ [tc + CDMin, tc + CDMin + DDMax]:
    SR(T) = tc + CDMin;
case D'(T) ∈ [tc, tc + CDMin]:
    SR(T) = D'(T);
case D'(T) ∈ [tc + CDMin + DDMax, ∞]:
    SR(T) = D'(T) - DDMax;
/* map 为优先级分派函数 */
Pri(T) = map(SR(T));
    
```

4.3 调度策略

在实时数据库系统中,若系统超载,则可能导致较多移动软实时事务错过其扩展截止期,严重降低系统的性能,因此我们对事务的执行增加了接纳控制和放行控制。接纳是根据系统当前状态(是否超载、是否满足事务最小内存需求等)判断是否接纳新事务。若同意接纳,事务则进入接纳(Admission)队列。而放行是判断事务的存取数据集是否已在内存数据库,若不在,则装入数据后再放行,进入事务就绪(Ready)队列。接纳算法描述如下:

```

算法名称:TransAdmControl
输入:D'(T)
输出:无
算法:
/* 接收到 MH 发出的事务请求,BS 进行接纳判断 */
if (the admission conditions are held)
then
    C(T) = 0; /* 置事务未提交标志 */
    /* 计算 T 的优先级 */
    Pri(T) = ComputeTransPri(D'(T));
    /* 根据事务优先级大小插入接纳队列 */
    inserts(T, BSAdmQ);
else
    /* 不满足接纳条件则拒绝事务请求 */
    T is refused
    
```

这样,当调度时机(事务发生状态转换时)到,则调度就绪队列中最高优先级事务执行,进入运行(Running)队列。在执行过程中,事务可能由于 CPU 竞争和数据竞争而挂起,甚至终止。令:

$BSReadyQ$ ——BS 上的就绪队列
 $BSRunQ$ ——BS 上的运行队列
 $MaxPri(ReadyQ)$ ——就绪队列最高优先级
 $MinPri(RunQ)$ ——运行队列最低优先级

$T(MaxPri(ReadyQ))$ ——就绪队列最高优先级事务

$T(MinPri(RunQ))$ ——运行队列最低优先级事务

$T(MaxPri(RunQ))$ ——运行队列最高优先级事务

$MAXRunQNum$ ——运行队列最大事务数

$CurrentRunQNum$ ——当前运行队列已有事务数

于是 BS 上执行事务调度策略可描述如下算法:

```

算法名称:ScheTrans
输入:MaxPri(ReadyQ), MinPri(RunQ)
输出:无
算法:
/* 当前运行队列已满 */
if (CurrentRunQNum == MAXRunQNum)
then
    if (MaxPri(ReadyQ) > MinPri(RunQ))
    then
        /* 挂起运行队列最低优先级事务 */
        suspend(T(MinPri(RunQ)));
        /* 运行队列最低优先级事务按优先级插入就绪队列 */
        inserts(T(MinPri(RunQ)), BSReadyQ);
        /* 就绪队列最高优先级事务按优先级插入运行队列 */
        inserts(T(MaxPri(ReadyQ)), BSRunQ);
    else
        /* 退出此次调度 */
        exit schedule
if (T(MaxPri(RunQ)) 已经运行过)
then
    /* 唤醒运行队列最高优先级事务继续执行 */
    wakeup(T(MaxPri(RunQ)))
else
    /* 调度运行队列最高优先级事务开始执行 */
    exec T(MaxPri(RunQ))
    
```

5 模拟实验结果

我们在国家资助、自行研制的嵌入式主动实时数据库原型系统 ARTs-EDB 上模拟移动计算环境完成了给出的协议 SSRTD 的性能测试。测试的性能指标为事务超截止期比率 MR(Missing Ratio); $MR = NumMiss / NumTotal \times 100\%$ 。其中 NumMiss 表示超截止期事务数量,NumTotal 表示事务总数。

我们选择文[5]中协议 DT-protocol 同 SSRTD 协议进行性能比较与分析。在 DTP 中,事务扩展截止期根据网络通信质量确定,事务调度中没有接纳控制和放行管理。表1为实验主要模拟参数。

表1 模拟参数表

参数	参数值
移动主机数量(NumMHs)	50
固定主机数量(NumFHs)	20
无线单元数量(NumCells)	6
每个单元信道数(NumCHs)	12
数据库数量(NumFDBs)	7
数据库大小(SizeDB)	650
调度策略	SSRTD 与 DTP
事务类型	软实时事务

性能测试结果如图3与图4所示。其中,事务发出间隔时间 TT(Think Time)表示移动主机执行完一个事务后经过时间 TT 再发出新事务。随着 TT 的增加,系统事务负荷不断减少。断接概率 DP(disconnection probability)表示无线网络的通信质量。随着 DP 增加,通信质量下降。图3中,假定 TT 值是固定的2sec。图4中假定断接概率固定为0.5%。从图3可以看出,采用 SSRTD 协议事务超截止期比率 MR 比采用 DTP 协议时要低。主要原因是 SSRTD 协议中事务扩展截止期与

(下转第162页)

模时得到的一个个模型实例是相符的。

总结 为了解决数据仓库概念模型的交流与共享,本文提出了一种基于 XML 的多维概念模型,它充分利用了 XML 的可扩展性和统一标准性,不仅能很好地解决这一问题,而且也能为元数据的集成与共享打下了基础。

针对多维模型的特点,本文定义了一个特定的 DTD,该 DTD 能够完整地描述多维概念模型的各种语义特征。对于多维概念建模,目前比较新的方法是基于 UML 的多维概念建模方法,它的一个显著优点就是建立在一种广泛接受的面向对象建模语言的基础上,从而使得开发人员在进行具体应用时,不必重新学习一些新的模型及其相关概念,便于广泛应用。而且,这种方法与 OMG 组织制定的 CWM 标准也是相符的,为今后的工作奠定了一个基础^[6]。根据这一实际情况,本文设计了基于 XML 的多维概念模型与基于 UML 类图的多维概念模型之间的映射方法。把 UML 类图映射为相应的 XML 文档,这给模型的交流与共享,以及元数据的集成与共享带来了极大的方便。把 XML 文档逆向映射为 UML 类图,能够对 XML 数据进行快速的图示浏览,从而为多个多维模型的集成提供了一定的指导。

在具体的实现中,我们选择了在 Rational Rose 2002 中嵌入一套自定义的组件,该组件不仅广泛支持基于 UML 的多维概念建模方法^[14],而且也能非常方便地进行 XML 文档与 UML 类图之间的转换。

参考文献

1 Kimball R. The data warehouse toolkit. New York, John Wiley

& Sons, 1996
 2 Trujillo J C, Palomar M, Gomez J, Song I-Y. Designing data warehouses with oo conceptual Models. IEEE Computer, 2001, 34(12): 66~75
 3 Abelló A. YAM²: a multidimensional conceptual model: [PhD Thesis]. Computer Science in the Facultat d'Inforàtica de Barcelona (FIB)2002
 4 Blaschka M. FIESTA: A framework for schema evolution in multidimensional databases: [PhD thesis]. Technische Universitat Munchen, Germany, 2000
 5 Tryfona N, Busborg F, Christiansen J G B. starER: A conceptual model for data warehouse design. In: Proc. of the ACM 2nd Intl. Workshop on Data warehousing and OLAP (DOLAP'99), Kansas City, Missouri, USA, 1999. 3~8
 6 OMG. Common warehouse metamodel, February 2001. Version 1.0.
 7 Moody D L, Kortink M A R. From enterprise models to dimensional models: a methodology for data warehouse and data mart design. In: Manfred A. Jeusfeld, eds. Proc. s of the Second intl. Worskshop on Design and Management of Data Warehouses 2000, Stockholm, Sweden
 8 Pederson T B, Jensen C S. Multidimensional data modeling for complex data. In: Proc. of the 15th Intl. Conf. on Data Engineering (ICDE'99), Sydney Australia: IEEE Computer Society, 1999. 336~345
 9 Jensen M R, Møller T H, Pedersen T B. Converting XML data to UML diagrams for conceptual data integration. Data & Knowledge Engineering, 2003, 44(3): 323~346
 10 李建中, 高宏. 一种数据仓库的多维数据模型. 软件学报, 2000, 11(7): 908~917
 11 陈明, 吴国文, 施伯乐. 数据仓库概念模型的设计. 小型微型计算机系统, 2002, 23(12): 1453~1458
 12 李琪, 白英彩. 数据仓库中维的建模和查询. 计算机研究与发展, 2002, 39(5): 612~618
 13 Wang Xiao-ling, DONG Yi-sheng. XML-based data cube [C]. RIDE02, San Jose, USA

(上接第157页)

网络质量无关,只与事务本身的实时特性相关,所以随着 DP 增加,基本上是线性变化。图4是由于 SSRTD 采用了接纳控制和放行管理,有效地控制了系统的负载,减少了事务运行时内外存的 I/O 交换时间。

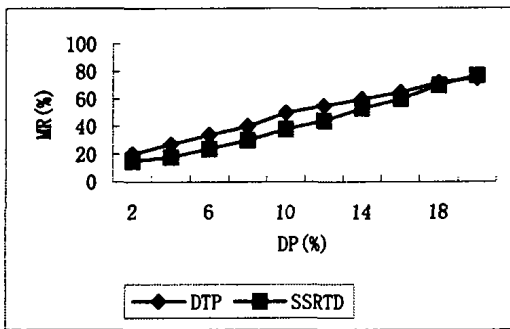


图3 断接概率对 MR 的影响

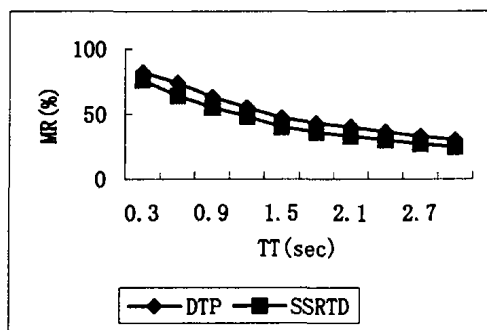


图4 事务负荷对 MR 的影响

移动分布环境中实时事务的执行。因此,必须为移动分布式实时事务建立新的事务模型及其事务管理机制。本文分析了移动计算环境的软实时事务调度管理,研究了一种根据移动主机的连接情况、事务的扩展截止期和结果相似性等因素决定发送执行结果的实时事务调度策略 SSRTD。全文工作概括如下:

- (1)给出了一个移动实时数据库系统模型;
- (2)给出了“扩展截止期”和“结果相似性”的概念;
- (3)给出了根据“扩展截止期”和“结果相似性”计算事务执行结果的发送时间方法;
- (4)给出了 SSRTD 协议的相关算法;
- (5)对 SSRTD 协议进行了性能测试和分析,证明其具有较好的性能。

移动实时事务是一个新研究领域,有许多方面需要研究。我们将进行的下一步工作重点是研究移动实时数据库恢复处理机制。

参考文献

1 Chrysanthi P K. Transaction processing in mobile computing environment. In: Proc. of 6th IEEE Workshop on Advances in Parallel and Distributed Systems, New Jersey, 1993
 2 Lam K Y, Kuo T W, Tsang W H. Concurrency control in mobile distributed real-time database systems. Information systems, 2000, 25(4): 261~286
 3 廖国琼, 刘云生. 移动实时嵌套事务的并发控制. 计算机学报, 2003, 26(10): 1326~1331
 4 刘云生, 廖国琼. 移动实时嵌套事务提交. 软件学报, 2003, 14(1): 139~145
 5 Saad-Bouzeffrane S, Sadeg B, Amanton L. In: Proc. of 4th IEEE Intl. Symposium on Object-Oriented Real-Time Distributed Computing, Magdeburg, Germany, 2001
 6 刘云生. 现代数据库技术(第一版). 北京:国防工业出版社, 2001

结束语 传统分布式实时事务管理机制不能较好地支持