维普资讯 http://www.cqvip.com

实时 数据库 数据管理 (10)

计算机科学 1994 Vol. 21 No. 3

42-46

实时数据库系统*)

TP311.13

刘云生

(华中理工大学计算机系 武汉 430074)

摘 要 A simple integration of a traditional database system with a real-time system is not feasible for a real-time database system (RTDBS). The essential characteristics of a RTDBS are temporal consistency of data and timing constraints of transactions. The key issue is to manage transactions time-cognizantly to implement internal, external and mutual consistencies of data, and time constraints of transactions. This depends on transaction model properties, and protocols and algorithms of transaction processing. In this paper, we generally give an overall discussion about the aspects mentioned above.

一、引言

一般实时应用有两个突出的特点，一是其中的活动时间性强，要求在一定的时刻和/或一定的时间内自外部环境收集信息、按彼此联系存取已获得的信息和处理收集的信息、再及时作出响应；二是它们要处理"短暂"数据，这种数据只在一定的时间范围内有效，过时则无意义了。雷达跟踪、指挥控制、化学或核反应过程控制、电话交换、证券交易、CIMS 等是这种应用的典型例子。支持这种应用的数据管理则要求实时数据库系统。

实时数据库系统(RTDBS)就是其事务和数据都可以有定时特性或显式的定时限制的数据库系统。系统的正确性不仅依赖于逻辑结果，而且还依赖于逻辑结果产生的时间。近年来，RTDBS 受到了数据库和实时系统两个领域里的研究者的极大关注，数据库研究工作者旨在利用数据库技术来解决实时系统中的数据管理问题，实时系统研究工作者则致力于为 RTDBS 提供时间驱动调度和资源分配算法。然而，RTDBS 并非是两者在概念、结构、方法上的简单集成，需要对一系列问题进行全面的研究与决策：数据和数据库的结构与组织；事务时限的软硬性、事务模型及其结构特征；事务的调度策略，包括优先级分派、并发控制的协议与算法；I/O 调度、恢复、通信的协议与算法；查询处理算法；数据和事务特性的语义及其与一致性、正确性的关系等，这些问题彼此高度相关且与应用的性质紧密联系。本文就这些问题从概念、解决的方法与技术方面作概括性的讨论。

二、数据库、实时系统和 RTDB

这里主要分析传统数据库、实时系统(RTS)和 RTDB 彼此间的联系与差别以说明什么是 RTDB、为什么要研究 RTDB。

2.1 数据库与实时系统

数据库系统组合了多种功能：(1)数据描述(模型、模式)；(2)数据正确性维护(完整性、一致性检验)；(3)有效的数据存取(数据库组织、操作与存取方法)；(4)查询与事务的正确执行(事务管理、调度与并发控制)；(5)数据的安全与可靠性保护(安全性检验、恢复)。随着当今实时系统处理的信息量越来越大，对这些功能的要求也越来越迫切。

但在传统的数据库系统中，其设计与开发主要强调维护数据的正确性、保持系统代价低、提供友好用户接口。这种数据库系统对传统的商务和事务型应用是有效的、成功的，但它不适合实时应用，关键在于它不考虑与数据及其处理相联系的定时限制(timing-constraint)。系统的性能目标是吞吐量和平均响应时间，而不是各个事务的限制，故系统作调度决策时根本不管各种实时特性。

相反，RTS 所支持的应用具有很强的时间性要求，其处理活动与数据都是定时性的。这种数据的定

*)国防预研和国家自然科学基金资助研究。刘云生 副教授，主要从事现代数据库与信息系统、实时软件与操作系统、软件开发方法学的研究。

维普资讯 http://www.cqvip.com

时性和外部环境所施加的响应时间要求导致了RTS中任务的定时限制，如期限或截止时间。因而RTS强调的是定时性，系统目标是满足各种时间限制。然而传统的实时系统典型地处理具有简单和可预报数据（或资源）要求的简单任务，不涉及维护共享数据的一致性问题。

数据库与实时系统的结合导致了RTDBS的产生，它集成两者的概念与要求以同时处理定时性和一致性，这使得调度问题要复杂和困难得多。

2.2 传统数据库与RTDB

对于传统数据库（TDB），其事务具有ACID（A omicity，Consistency，Isolation，Durability）特征，即：

（1）强调一致性、可恢复性和永久性；

（2）事务之间无合作（交互作用、通信）；

（3）进行不可预报的数据存取，故其执行时间不可预测；

（4）无"时间维"，更不显式地考虑时限，

所以，事务的原子性和可串行化是普遍接受的正确性、一致性标准。

在RTDBS中，则有根本性的不同，其事务由下列特性（概念）确定：

（1）可见性，事务执行时可查看另一事务的结果的能力；

（2）正确性，事务本身的正确性及提交事务所产生的数据库状态的一致性；

（3）可恢复性。发生故障时使数据库成为某种被认为是正确状态的能力；

（4）永久性。事务记录其结果到数据库及识别其中数据的有效期的能力；

（5）可预报性。事先预测一个事务是否会满足其时限的能力。

在实时应用环境下这是由于：

（1）传统的平坦原子事务模型已不完全适用，有的要求"复杂事务"（见4.1节）；

（2）事务要合作、协调而并发地执行，故要求有较强的可见性；

（3）及时性比正确性更重要，有时宁愿要求及时而部分不正确的信息，而不愿得到正确但是过时（无效）的信息；

（4）要求"识时"协议和"时间正确"的调度，至于对TDB最重要的可串行化调度与开发控制倒不一定必需；

（5）并非所有数据都是永久的，许多是短暂的，甚至不一定要求所有结果都记录到数据库；

（6）恢复不一定就是数据库状态的完全复原，

所以，RTDBS与TDBS在概念、原理、结构、算法等方面都存在着很大的差别，最根本的区别在于数据与事务的定时限制。这里要指出的是，"实时"并非简单地意味着快，快故然需要，但对RTDBS而言，实时指的是能施加和处理"显式"的定时限制，即使用"识时协议"（time-cognizant protocol）来处理有关的截止时间或定期限制。

三、RTDBS的特征

从数据库的角度看，一个RTS典型地由三个紧密结合的子系统组成：被控系统、执行控制系统、数据系统。被控系统就是所考虑的应用过程，称为外部环境或物理世界；执行控制系统监视被控系统的状态，协调和控制它的活动，称为逻辑世界；数据系统有效地存贮、操纵与管理实时（准确和及时）信息，称为内部世界，执行控制系统和数据系统统称控制系统。内部世界的状态是物理世界状态在控制系统中的映像，执行控制系统通过内部世界状态而感知物理世界状态，且基于此而与被控系统交互作用，所有这些都与时间紧密相联。因而RTDBS的特征主要表现在数据和事务的时间相关上。

3.1 RTDB的数据特征

在RTDBS中，数据只在一定时间内是"流行"的，其值随外部环境状态变化而频繁地改变，所以人们不能只考虑数据库内部状态的一致性，还必须考虑外部状态与内部状态之间的一致性；也不能认为使用数据时，简单地提供其最新值就是最合适的，还必须考虑它与其他被使用数据间的"时间一致性"。这些就是RTDB的数据特征，下面给予较详细的讨论。

定义1 一个RTDB的数据对象为一个三元组 $d:\langle v, evi, tp\rangle$。其中分量 d_v、d_{tp}、d_{evi} 分别为d的当前状态或值；观测时标（observation timestamp），即记录 d_v 到数据库的时间；外部（绝对）有效期（external validity interval），即自 d_{tp} 算起 d_v 具有外部或绝对有效性的时间长度。

1. 内部一致性。就是传统意义下的数据库内部的一致性，即：

定义2 数据d是内部（或逻辑）一致的，当且仅当 d_v 满足所有预先定义的完整性限制。

数据库的变更都是以事务的形式进行的，因而事务的可串行化能提供内部一致性保证。

维普资讯 http://www.cqvip.com

2. 外部一致性。在RTDBS中，数据库依靠逻辑世界与物理世界的频繁交互作用来获取准确、及时的数据，故一个正确的数据库状态必须与物理世界当时的状态一致，即事务使用的数据库中的值 d_v 是在其有效范围 d_{evt} 内。

定义3 数据d是外部(或绝对)一致的，当且仅当 $(t_c-d_{tp})\leqslant d_{evt}$ (t_c 为当前或检测时间)。

数据的外部有效期 d_{evt} 可以通过足够充分的物理世界对应参数的取样来获得。外部一致性与时间限制相联，它已不能用可串行化来实现。因为现有的确保可串行化的并发控制协议都是基于封锁或回退/重起动(rollback/restart)的，封锁会导致优先级颠倒(低优先级事务阻塞高优先级事务)，而回退/重起动除浪费大量系统资源从而引起事务超时限外，实时系统中有的事务根本不能回退/重起动，如记录飞行目标位置和速度等状态的事务、执行火箭"点火"操作的事务。对于这种反应物理世界不可逆事件的事务，不能封锁或回退，必要时只有先牺牲内部一致性而确保外部一致性，然后再求得内部一致性。

3. 相互一致性。一组相关数据被使用时，存在着它们之间在时间上的相互(或相对)一致性问题。

定义4 用来作决策或导出新数据的一组数据称为一个相互(相对)一致集，每一这样的数据集R都有一与之相联的相互有效期(mutual validity interval)，记为 R_{mvi}。

定义5 设R是一个相互一致集，$d\in R$。d是相互(或相对)一致的，当且仅当 $\forall d'\in R\ |d_{tp}-d'_{tp}|\leqslant R_{mvi}$。

相互一致性保证了用来作决策或导出新数据的一组数据是在公共的有效时间范围内彼此接近地产生的。R_{mvi} 的获得则不像 d_{evt} 那么简单，且它们之间存在着相关性[1]，R中各数据的 d_{evt} 的最小者对 R_{mvi} 起着决定作用。

4. 数据库状态正确性。它意味着内部、外部和相互一致性的全部。

定义6 若一个数据既是外部一致又是相互一致的，则说它是时间一致的。

外部一致性和相互一致性都与时间紧密相关，故统称为时间一致性。要实现时间一致性，系统必须使用识时协议来处理事务的定时性。

当且仅当一个数据同时是内部一致和时间一致的，它才有正确状态。而数据库有正确状态则意味着其中每一数据都有正确状态。

3.2 RTDB的事务特征

定时性是RTDB事务的根本特点，定时可以是绝对、相对或周期时间。它一方面是由数据的时间一致性所引起的，这种定时性往往取定期或周期性限制的形式，如"每5秒取样温度"，"每天7：00起动机器人"等。引起事务定时性的另一根源是物理世界施加于控制系统的反应时间要求，它典型地取施加于非定期事务的截止时间限制形式，如"若温度达到1000℃，则在5秒内加冷却剂到反应推"。

定时性包括了两方面的含义：

- 定时限制——事务的执行具有显式的时限，如期限、截止时间等。这是由于控制系统要不断跟踪被控系统而引起的，它要求RTDBS必须要有时间处理机构。时限还可有软硬之分(见4.3节)。
- 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控制系统的各种决策活动随时有效而引起的，它要求权衡定时限制及数据一致性等多方面因素，提供合适的调度算法。

完全实现事务的定时性和数据的一致性是一个复杂而困难的任务，它还依赖于后面各节所述的事务模型与处理。

四、RTDB事务模型

4.1 复杂事务

实时应用经常需要复杂事务模型。不同于传统事务只有平坦的单层内部结构，复杂事务可以有复杂的内部构造，形式描述如下：

Complex-Tran ::= (Longlife-Tran) | (Multilevel-Tran) | (Nested-Tran) | (Split-Tran) | (Cooperative-Tran)

- Longlife-Tran：循环(endless)或无终止(open-end)事务。
- Multilevel-Tran：由操作和子操作以树结构组成的事务。
- Nested-Tran：由子事务以树结构组成的事务。

多层事务类似传统事务由一组操作组成。所不同的是它有内部层次结构，这又类似于嵌套事务，但嵌套事务的部件是事务，不必具有原子性，其层次结构是"显式"的，是一种用户可使用的技术；多层事务的部件是操作，有原子性，其内部结构是"隐式"的，是一种系统设施。嵌套事务是为了使失败局部化和利用并行机制，而多层事务是为了实现如在复杂对象上的操作。

- Split-Tran：一个事务分枝出另一事务。分枝事

维普资讯 http://www.cqvip.com

务与原事务可以是独立的或序列的。

·Cooperative-Tran::=〈FLat-Coop〉|〈Hierarchical-Coop〉

·Flat-Coop：“发送者”、“接收者”间的平坦通信，“接收者”对“发送者”有夭折依赖。

·Hierarchical-Coop：层次间通信合作的事务。其层次结构类似于嵌套事务，但没有像嵌套事务中那样的操作对象继承性，通信的“对象流”也只存在于相邻两层之间。

4.2 事务的相关性

在RTDBS中，事务之间存在着多种相关性：

1. 结构相关。它是由事务模型的结构特征而引起的。我们以依赖概念来描述这种相关性，设 t_a 和 t_b 为两个事务，“⇒”、“→”分别表示“蕴含”和“先于”，$BEGIN_t$、$COMMIT_t$、$ABORT_t$ 分别表示事务 t 的“开始”、“提交”、“夭折”操作，最通常的依赖有：

(1)提交依赖(CD)：t_a CD t_b iff $COMMIT_{ta} \Rightarrow (COMMIT_{tb} \Rightarrow (COMMIT_{tb} \rightarrow COMMIT_{ta}))$

(2)开始依赖(BD)：t_a BD t_b iff $BEGIN_{ta} \Rightarrow (BEGIN_{tb} \rightarrow BEGIN_{ta})$

(3)夭折依赖(AD)：t_a AD t_d iff $ABORT_{tb} \Rightarrow ABORT_{ta}$

(4)顺序依赖(SD)：t_a SD t_d iff $BEGIN_{ta} \Rightarrow (e \rightarrow BEGIN_{ta})$，$e \in \{COMMIT_{tb}, ABORT_{tb}\}$

还可以有许多其他依赖，这取决于事务模型结构，上述是各种依赖集的交集。

2. 行为相关。不像结构相关是直接的，它是由不同事务在共享数据对象上的操作之间的同步而引起的一种间接相关，通常用事务的“冲突关系”[2]来表示：t_a CR t_b：$\exists d \exists p,q(CT(p_{ta}(d), q_{tb}(d)) \wedge (p_{ta}(d) \rightarrow q_{tb}(d)))$

3. 时限相关。事务的定时限制(尤其是相对时限)带来了事务间的第三种相关性，它体现了事务的紧迫度，如“事务 t_1 必须在 t_2 提交后5秒内完成”。

4.3 RTDB事务的类型

按事务使用数据的方式来分类，犹如传统事务一样可以分为：

·只写事务——获取物理世界状态并写入数据库；

·只读事务——读取数据库中数据并发送给执行控制部件；

·更新事务——导出新数据并存入数据库。

这种分类便于改造并发控制方案。但在RTDBS中，通常按时限的性质分类更有利于设计调度算法，如此有[1]：

·硬截止事务——超截止期则导致恶果(价值为负)。它对应于安全危急性活动。

·软截止事务——超截止期后还有一定价值，典型地到某一时间点，价值降到零，此后保持为零(不会为负)。

·固截止事务——一旦达到截止期，其价值立即固定为零(不会为负)。它实质上是软截止事务的一种特例，故无须单独处理。

图1给出了用价值函数建模这些事务的典型例子，取分段线性函数是为了更典型而不是必需的。要确保所有硬截止时限是极难的。

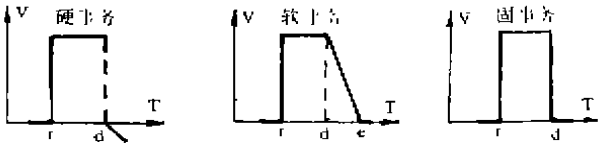

图1

4.4 RTDB事务的正确性

在RTDBS中，事务正确性的内涵与TDBS的有很大差别，它要复杂得多，包括[2]：

1. 结果正确性。它包括两种情形：·绝对正确——事务结果数据有正确状态；·相对正确——事务结果数据虽不具正确状态，但只要在其一定范围内也可认为是正确的。

2. 结构正确性。满足事务间的各种依赖。

3. 行为正确性。满足并发操作的经历约束[7]。

4. 时间正确性。满足各种定时限制。

五、RTDBS的事务处理

在RTDBS中，关于事务的处理要比TDBS的更复杂得多，主要有下列方面的问题要解决。

5.1 调度模型

调度模型由下列部件组成：

1. 事务模型：说明调度所要考虑的事务间的联系。

2. 资源模型：确定各种共享和排他资源。

3. 中断模型：确定是否可优先抢占及优先抢占的方式与机构(若可优先抢占)。

4. 负载模型：说明事务的执行及其时间分布，包括周期、非周期及随机事务。

5. 延迟代价模型：建模硬、软截止事务的价值(或延迟的代价)与时限的函数关系及其表示机制。

维普资讯 http://www.cqvip.com

6. 调度算法：给出基于上述各模型的支持时限和一致性的调度策略。

5.2 优先级分派

RTDBS 必须基于事务的时限和给予系统的价值来分派优先级，可用的策略有[5]：

1. 截止期最早优先(EDF)。即具最早截止期者优先级最高。其主要缺点是已过或几乎要过截止期者将获得最高优先级，这显然不合适。

2. 可达截止期最早优先(EFDF)。所谓一个事务 T 的截止期是当前"可达到"的，乃指 $t+(E-P)\leqslant d$，这里 t 为当前时间，E、P 分别为事务 T 的执行时间估算和已执行时间，d 为其截止期。它是克服了 EDF 的缺点之改进策略。

3. 空余时间最短优先(LSF)。事务 T 的空余时间 $S=d-(t+E-P)$，即推迟 T 的执行而仍能满足其截止期的可推迟时间量估算。要注意，LSF 与 EDF 和 EFDF 两者都不一样，它考虑了当前时间与剩余执行时间估算 $C=E-F$，故随事务的停止与执行其优先级动态升降。但该策略也有类似于 EDF 的缺点，同样可限制 $S\geqslant 0$，即只考虑可达截止期情形。

4. 价值最高优先(HVF)。每一事务有一价值函数，其值最大者最优先。问题是如何合理地构造价值函数，一个例子[4]是：

$$V(T)=c(w_1(t-t_s)-w_2d+w_3P-wS)$$

其中 t、d、P、S 的意义同上，c、t_s 分别为 T 的危急度、开始时间，w_i 为加权因子。

5. 价值比最大优先(GVDF)。价值比(Value Density)函数为：

$$VD=\frac{v(t+C)}{C}=\frac{v(t+E-P)}{E-P}$$

这里也有如何确定价值 v 的问题。

5.3 并发控制

并发控制就是解决事务冲突关系的策略及其实现。在 RTDBS 中，这种冲突关系的解决除要考虑数据一致性，还要考虑事务的优先级、截止期、已执行和距完成的时间长度，即要有"识时"性。

T_H、T_R 分别表示已占有和请求一个数据对象的事务，$P(T_x)$表示事务 T_x 的优先级。常用的并发控制协议有：

1. 优先级继承(PI)。若 $P(T_H)<P(T_R)$，则 T_H "继承"T_R 的优先级以继续执行而尽快完成。这样克服了通常 DBMS 中的标准顺序执行协议(FCFS)所带来的"优先级颠倒"(低优先级事务阻塞高优先级事务)的情形。

2. 优先抢占(PA)。若 $P(T_R)>P(T_H)$，则夭折 T_H 而让 T_R"抢先"执行，否则 T_R 等待。这样克服了 PI 协议最终还是使高优先级事务被阻的情形，对实时应用更合适。

3. 有条件优先抢占(CPA)。若$(P(T_R)>P(T_H))\wedge(S_R<C_H)$，则夭折 T_H，否则 T_H 继续执行。这里 S_R 和 C_H 分别为 T_R 的空余时间和 T_H 的剩余执行时间估算。该协议显然比 PA 更合理，系统性能更高。

以上各种协议都是"锁式"的，属于"悲观"协议范畴。还可以有乐观协议和基于时标的协议，它们都不要求锁机制，此处不再介绍。

六、结束语

我们讨论了什么是 RTDB，它的根本特点：数据的时间一致性和事务的定时限制；实现这些特点的核心问题是有关事务处理的各个方面，包括事务模型的结构与特性，事务的优先级分派、调度与并发控制的识时协议，这些比 TDBS 要复杂、困难得多，但一些在 RTS 和 TDBS 中有效的技术与方法可以进行"识时"性改造而使用。

RTDBS 还是一个新的研究领域，还有许多方面如 I/O 调度与缓冲管理、恢复、性能提高、超载管理等本文没有讨论，主动的 RTDB 和分布式 RTDB 又是两个新的分枝。总之，RTDB 已成为一个新的数据库研究方向，有广阔的天地让人们驰骋。(参考文献共7篇略)