

实时事务二次调度

Duplicate Scheduling of Real-Time Transactions for RTDBS

夏家莉 刘云生

(华中科技大学计算机学院 武汉430074)

Abstract There are a great of research results about RTDBS but no effective methods for increasing success ratio of real-time database systems (RTDBS). In this paper, we analyze the function alternative characteristic of real-time transactions, propose that transaction scheduling is divided to two levels--interior schedule and exterior schedule. It can increase systems success ratio. We also study pre-analysis strategy and scheduability about real-time transactions, lastly we compare interior schedule with exterior schedule.

Keywords Real-time database systems, Transaction schedule, Function alternative set

实时事务调度的主要目的是调度实时事务使其满足事务的时间约束,它包括针对多个事务的CPU调度以及与之紧密相关的在多个并发事务之间的数据、I/O及内存等资源的调度^[1,2]。关于调度有大量的研究成果,包括:Earliest Deadline First (Liu 和 Layland, 1973; Abbott 和 Garcia-Molina, 1988 等), Least Slack First (Abbott Garcia, 1992), Weighted Priority (Huang et al., 1989), 以及控制优先级倒置的可抢占算法等。无论哪种调度方法,在动态实时环境下满足所有事务的截止期都是很困难的,而提高系统成功率是实时事务调度的主要目标,为此,本文从一个新的角度来研究这个问题,探讨实时事务的功能替代性对调度的影响。尽管早已有人提出功能替代性,但至今未见深入研究的报道,我们认为由于实时事务的功能替代性,实时事务调度不再是传统的单一的事务调度,而是内部调度和外部调度的统一。

本文试图从以下方面阐述这个问题:①实时事务的功能替代性;②功能替代性导致了实时事务的二次调度;③调度的过程和策略。本文最主要的贡献在于提出了实时事务二次调度,并给出了相应的策略和方法。

1. 具有功能替代性的实时事务

“功能替代”的提出源于客观应用和实时事务调度的需求,主要体现在两个方面:1)满足提高实时事务对实时环境应变能力的要求。一个典型的例子就是现代国防防御系统,执行防御的事务应具有多种执行途径,系统首先选择一个最佳途径,当失败或因故不能执行时,按另一个途径执行,以此提高事务的成功率。类似的情况很普遍,如航空领域中无人驾驶飞行器的实时反应,航海、智能机器人以及各种民用领域中的实时运用。2)从调度的角度来说,当事务重启时,如果依然按原路径执行,导致其重启的原因可能并没有排除,势必造成再次无效重启,然而,如果该事务在重启时能选择替代集,就有可能避开障碍获得成功,提高该事务的成功率。

实时事务由一组功能等价的功能替代集组成,只要其中一个功能替代集能成功执行,该实时事务就能成功提交。

定义1 实时事务的形式定义:

$T ::= (FT, \langle \tau, RR, CC \rangle)$

·FT 是一组功能等价的功能替代集: $FT ::= \{FX_i | 1 \leq i \leq p\}$; $FX_i ::= TRANS$; $TRANS ::= TRANS \mid OP$; OP 是基本的数据库操作和对象操作。

·CC 是该事务的一组约束,包括时间约束、一致性约束等,它由定义在各功能替代集上的一组约束组成: $CC ::=$

$\{CC_i | 1 \leq i \leq D(GT)\}$

·RR 为执行该实时事务所需要的系统资源。它由定义在各功能替代集上的一组系统资源信息组成: $RR ::= \{RR_i | 1 \leq i \leq D(GT)\}$

· $\langle \tau \rangle$ 是定义在各功能替代集上的时序。

在单处理机系统的调度过程中,同一实时事务的功能替代集是互斥的,在每一时刻,某实时事务最多有一个功能替代集参与CPU竞争,因此,系统必须选取一个成功率最高的功能替代集参与资源竞争,以便提高系统的成功率。

2. 实时事务二次调度

从功能上看,实时事务的每个功能替代集都能实现该事务的功能,但由于它们在执行路径上的区别导致性能不同,特别是执行时间和所需资源可能不同。在理想状态下,我们总是希望让性能最好、最适合当前系统状态的功能替代集投入运行,而不是随机地选取一个对象,这就存在调度问题,这个调度在传统的调度之前进行,我们称之为内部调度(IS),传统的针对于多个实时事务的调度称为外部调度(ES),图1表示了它们的对应关系。在图1中,实时事务 T_1 由功能替代集 g_{11} , g_{12} 和 g_{13} 组成,实时事务 T_2 由功能替代集 g_{21} 和 g_{22} 组成,内部调度 SN_1 针对于实时事务 T_1 ,从中选取一个合适的对象 g_{1i} ($i \in [1, 3]$),内部调度 SN_2 针对于实时事务 T_2 ,从中选取一个合适的对象 g_{2j} ($j \in [1, 2]$),而外部调度 SW_1 针对于 g_{1i} 和 g_{2j} ,实际上是对内部调度的结果进行二次调度,经过外部调度的实时事务将正式运行。

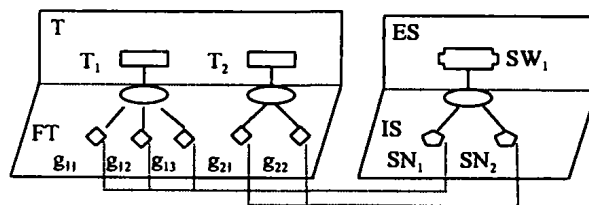


图1 实时事务及其调度关系图

内部调度针对于同一实时事务的多个功能替代集,目的是选取一个最佳的功能替代集;外部调度针对于多个实时事务,目的是在多个实时事务中分配系统资源(包括CPU、数据对象等)。并不是每个功能替代集都具备可调度性,在某个具体的硬实时环境下,那些执行时间超过截止期或因得不到所需资源而长久等待最终必定超过截止期的功能替代集是不可

调度的,如果它们参与竞争系统资源(即:外部调度)只会浪费系统时间,因此,系统应避免让它们进入外部调度。通过预先分析,可以对各功能替代集提取有关结构、行为、时限、资源等方面的信息^[3,4],从而估算功能替代集成功执行的可能性。

定义2 实时事务 T 的功能替代集 FT_i 是可调度的,记作 $SC(FT_i)$ 当且仅当下列条件成立: $(et_i + t_0 \leq d_i) \cap (R_i \subseteq R)$ 。

其中, et_i 是 FT_i 的执行时间, t_0 是系统的当前时间, d_i 是实时事务的截止期, R_i 是 FT_i 所需资源, R 是系统当时可以提供的资源。

定义3 实时事务 T 是可调度的,记作 $SC(T)$, 当且仅当下列条件成立: $\exists FT_i \in FTSC(FT_i)$

定义4 对于一个实时事务 T , 设其功能替代集的集合为 FT , 可调度集 FT_s 为 FT 的子集, 其中的成员满足条件: $\forall i, FT_i \in FT_s, (FT_s \subseteq FT) \cap SC(FT_s)$

实时事务调度过程如图2所示, 假设通过预先分析, 已经取得了实时事务中一个功能替代集的各种运行特性信息, 据此可以判定它们的可调度性, 系统首先进行内部调度, 选取一个功能替代集参与外部调度, 如果该功能替代集能成功执行, 则表示该实时事务能成功执行, 可提交; 如果该功能替代集夭折而实时事务的截止期未到, 则重新进入内部调度, 选取另一个合适的对象。对某个实时事务而言, 当下列条件之一成立时, 停止对该实时事务的此次调度: ①事务的截止期到; ②显式地定义, 当某种情况出现时强迫停止调度; ③该实时事务的所有功能替代集都夭折; ④有一个功能替代集成功执行并提交。

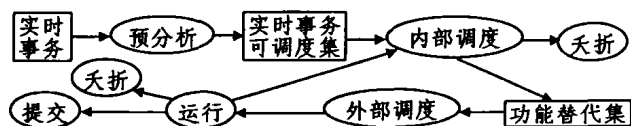


图2 实时事务调度图

3. 内部调度策略

实时事务内部调度策略主要有以下几种:

(1) 执行时间最短优先 该策略将最高优先级分派给具有最短执行时间的功能替代集, 此策略的优点是: ①执行时间最短, 空余时间就最长, 成功的机会更多; ②即使运行失败, 运行对象也尽可能少地占有 CPU, 减轻 CPU 负担, 同时将发现错误的时间尽可能提早, 将有较多的时间重新调度其它对象。

(2) 资源需求种类最少 该策略将最高优先级分派给具有最少资源需求种类的功能替代集, 因为较多的资源需求蕴涵着较多的资源竞争和受阻, 从而导致因资源等待而超过截止期的机会增大, 该策略的优点是降低资源等待风险。

(3) 嵌套层次最少优先 事务的嵌套层次过多增加了调度的难度和资源负担, 考虑到这方面的因素, 该策略将最高优先级分派给具有最少嵌套层次的功能替代集, 此策略一般作为辅助策略使用。

(4) 加权平衡优先 该策略综合考虑多方面的因素, 通过构造一个优先级函数来分派优先级, 如:

$$P(FT_i) = (a_1 et_i + a_2 R_i / R + a_3 NE) / (a_1 + a_2 + a_3)$$

其中 a_1, a_2, a_3 是权值, NE 是 FT_i 的嵌套层次。

本文举例说明一种基于时间和资源约束的内部调度算法。对于实时事务 T , 按以上分析它有一个调度集, 将此调度集按执行时间最短优先分为 j 类, 并赋予相应的优先级, 然后, 对这 j 个类的成员分别按资源需求种类最少排序, 可以由以下方法得到:

$$j = (d_i - t_0) / 2et_k$$

其中 $et_k = \min(et_i, 1 \leq i \leq n)$ n 为实时事务的可调度几和包含的功能替代集的个数。

调度算法

输入: 实时事务的可调度集 KT

输出: T 的调度结果 (T 或 F)

{ 是系统对 T 进行外部调度, 返回 T 表示能成功执行, 否则重新进入内部调度 }

1. 对 KT 按执行时间由短到长排序 $\Rightarrow AM[1 \dots n]$;
2. 计算 et_k 取 j 值;
3. 对 $AM[(k-1)j, kj]$ $k \in [1, n/j]$ 按资源需求种类由少到多排序;
4. $h = 1$;
5. While $h \leq n$
 { $b_i = \text{schedule}(AM(h))$;
 if $b = \text{true}$ then exit else $h = h + 1$ }
6. if $b < > \text{true}$ then print("the transaction is not schedulable")
7. 结束

外部调度策略已经很成熟, 本文着重讨论内部调度与外部调度的关系, 其关系体现在以下几方面:

①内部调度与外部调度紧密联系, 内部调度是基础, 只有通过内部调度才能选出一个可调度的功能替代集参与外部调度, 外部调度是最终目的, 只有通过外部调度, 实时事务才能真正投入执行。

②两者的调度对象不同, 内部调度针对于某个实时事务的所有功能替代集, 外部调度针对于系统接纳的所有实时事务。

③两者的调度目的不同, 内部调度是为了提高外部调度的成功率, 外部调度是为多个实时事务分派系统资源, 保证实时事务的具体执行。

结束语 实时事务调度因其功能替代性而更加复杂, 内部调度剔除不可调度的功能替代集, 禁止它们进入外部调度, 提高了外部调度的效率和成功率, 本文的主要工作在于: ①提出了基于功能替代性的实时事务模型; ②分析了实时事务的功能替代性对调度的影响; ③提出了实时事务调度分为内部调度和外部调度; ④给出了内部调度策略, 并进一步讨论了内部调度和外部调度的关系。

参考文献

- 1 Kim Young-KuK. Predictability and Consistency in Real-time Transaction Processing. [PhD thesis]. Department of Computer Science, University of Virginia, May 1995
- 2 Bestavros A, Braoudakis S. Value-cognizant speculative concurrency control. In: Proc. of VLDB'95: The Intl. Conf. on Very Large Databases, Zurich, Switzerland, Sep. 1995
- 3 Bestavros A, Braoudakis S. Timeliness via. Speculation for real-time databases. In: Proc. of RTSS'94: The 14th IEEE Real-Time System Symposium, San Juan, Puerto Rico, Dec. 1994
- 4 刘云生. 关于实时数据库事务. 软件学报, 1995, 6(10)
- 5 Chen D, Mok A K, Nixon M. Providing real-time support through component object model. Microprocessors and Microsystems, 1999, 23: 145~154
- 6 Lowe G. Scheduling-Oriented Models for Real-Time. The Computer Journal, 1995, 38(6)
- 7 Vrbsky S V, Tomic S. Satisfying temporal consistency constraints of real-time databases. The Journal of Systems and Software, 1999, 45: 45~60
- 8 何新贵, 刘云生, 等. 特种数据库技术. 科学技术出版社, 1999
- 9 Abbott R, Garcia-Molina H. Scheduling real-time transactions. ACM, SIGMOD Record, 1988, 17(1): 71~81
- 10 Son S H. Issues and Approaches to Supporting timeliness and security in real-time database Systems. Journal of Systems Architecture, 2000, 46: 397~410
- 11 刘云生, 胡国玲. 实时主存数据库事务的预分析处理. 软件学报, 1996, 7
- 12 Konana P, Ram S. Transaction Management Mechanism for Active and Real-time Databases: A Comprehensive Protocol and A Performance Study. The Journal of System and Software, 1998, 42: 205~228