

混合实时事务的接纳控制机制 ACMHRTT

白似雪 曹重华 程春雷 肖水晶 邬昌兴
(南昌大学计算机科学与技术系 南昌330029)

摘要 提出了一种含有实时和非实时两部分的混合实时事务接纳控制机制 ACMHRTT (Admission Control Mechanism for Hybrid Real-Time Transactions)。ACMHRTT 将混合事务分为真实时和非实时两部分,综合考虑混合事务的执行需求和价值,目的是为保留系统资源,有效控制负载,减少了被系统接纳执行后而又不能满足截止期的事务,提高实时数据库系统事务处理的成功率和可靠性,使系统收益最大化。着重分析了 ACMHRTT 的模型,对 ACMHRTT 与传统的 ACM 作了分析比较,说明了比传统的 ACM 协议更优,并提出了须进一步研究的问题。

关键词 混合实时事务,接纳控制,实时数据库系统

Admission Control Mechanism for Hybrid Real-Time Transactions: ACMHRTT

BAI Shi-Xue CAO Chong-Hua CHENG Chun-Lei XIAO Shui-Jing WU Chang-Xing
(Department of Computer Science & Technology, Nanchang University, Nanchang 330029)

Abstract A new admission control mechanism, for hybrid real-time transactions consisting of both genuine real-time and non-real-time parts (ACMHRTT), is put forward. ACMHRTT considers both the executing requirement and the value of hybrid real-time transactions. In order to maximize system profit, it reserves system resources and controls effectively system load, by reducing the number of transactions being accepted for execution but not able to come to its deadline. This paper emphasizes on analyzing ACMHRTT model, comparing ACMHRTT with traditional admission control mechanism. In addition, it suggests some research to be done later.

Keywords Hybrid real-time transaction, Admission control, Real-time database system

1 引言

实时数据库就是其数据和事务都有显式定时限制的数据库,系统的正确性不仅依赖于事务的逻辑结果,而且依赖于该逻辑结果所产生的时间。实时数据库在概念、方法和技术上都与传统的数据库有很大的不同。其核心问题是事务的处理既要确保数据的一致性,又要保证事务的正确性,而它们都与定时相关联。

实时数据库一个非常重要的性质是事务有多种定时限制,其中最典型的是事务截止期。按事务截止期分类,实时事务一般可以分为三类:硬实时事务、软实时事务、固实时事务。

实时数据库系统要求有较高的成功率,大量的并发控制协议致力于将超过截止期的事务减至最少,但是这些协议只有当系统超载时才能体现出优势,问题是当系统超载时,超过截止期的事务数量又大幅度上升。如果这些事务一开始就不被接纳,系统就不会在这些事务上浪费资源。Nagy 和 Bestavros 提出的接纳控制协议将事务分为两部分:主任务和补偿任务^[1,2]。它们不考虑事务的最坏执行时间和资源需求,不能预计阻塞。Chakravarthy 等人设置负载因子并与调度相结合,将系统负载控制在一定的范围内,系统资源利用率较低^[3]。文[4]提出了接纳控制机制 IACM,考虑了事务的执行需求和价值。

但在现实世界中存在一类实时事务,它包含了真实时(硬实时^[5]、软实时、固实时)和非实时两部分,如交通指挥的电子警察对过往车辆进行实时扫描,而对违章车辆的各种处理则是非实时的。而上述的各种协议、方法没有对这类混合事务的真实时和非实时部分加以区别,结果造成资源的浪费,甚至可以使接纳的事务没有被接纳,致使系统效益不能最大化。本文

考虑这种情形而提出一种混合实时事务接纳控制机制 ACMHRTT。

2 混合实时事务

实时数据库系统(RTDBS)中的事务通常是“复杂事务”,即嵌套、合并/分裂、通信和合作事务模型等,传统的 ACID(原子、一致、隔离、永久)性的事务模型对实时数据库已不适用;故 RTDBS 必须支持这种复杂事务,处理事务间的结构行为和和时间相关性。

在实际应用中,实时事务除了单纯硬实时、软实时、固实时、非实时之外,还有一类同时包含了硬实时与非实时、固实时与非实时、软实时与非实时的混合事务。把硬实时、软实时、固实时统称为真实时,混合实时事务都可化分为 T_r (真实时)与 T_n (非实时)两个子事务。

定义1 事务 $T = \langle C, D, I, \Delta \rangle$, 其中 C 为 T 的期望执行时间, D 为截止期, I 为初始相位, $\Delta = C_r / C$ (C_r 为 T 的真实时部分的期望执行时间);若 $0 < \Delta < 1$, 则称 T 为一个混合事务。

显然若 $\Delta = 1$ (真实时事务)和 $\Delta = 0$ (非实时事务)为其特例。

利用“分裂模型”将混合事务分裂为真实时和非实时两个子事务,以此来综合考虑事务的执行需求和价值,判断此混合事务能否被接纳,从而使系统有效利用,同时使系统收益最大化。

3 混合实时事务价值模型

在 RTDBS 中如果不能满足实时事务的截止期,则硬实时事务会给系统造成灾难,软实时事务的价值在其截止期后

也会降低,固实时事务的价值截止期后变为零,非实时事务的价值不随时间变化。混合实时事务的实时性和价值可由用户预先设定。设事务 T_i 的截止期为 D_i ,真实时部分价值为 x_{ri} ,非实时部分的价值为 x_{ni} ,若 T_i 为硬实时与非实时的混合事务,则当 T_i 不能顺利完成时将给系统造成灾难设为 y_{ni} 。 $f(t)$ 为软实时事务超过截止期 t 时间对系统造成的损失;可为每个混合事务设置一个函数 $V_i(t)$,则各种混合事务的价值函数为:

若 $T_i \in HHRT$ (含硬实时的混合事务),则

$$V_i(t) = \begin{cases} x_{ri} + x_{ni} & A_i \leq t \leq D_i \\ -y_{ni} + x_{ni} & t > D_i \end{cases}$$

若 $T_i \in HSRT$ (含软实时的混合事务),则

$$V_i(t) = \begin{cases} x_{ri} + x_{ni} & A_i \leq t \leq D_i \\ x_{ri} - f(t - D_i) + x_{ni} & t > D_i \end{cases}$$

若 $T_i \in HFRT$ (含固实时的混合事务),则

$$V_i(t) = \begin{cases} x_{ri} + x_{ni} & A_i \leq t \leq D_i \\ x_{ni} & t > D_i \end{cases}$$

当 T_i 为纯真实时事务时 $x_{ni} = 0$ 。

当 T_i 为非实时事务时 $V_i(t) = x_{ni}$ 。

这样我们就可用一致的观点 $V_i(t)$ 看待所有的混合事务价值。

4 混合实时事务接纳控制策略

ACMHRTT 的接纳控制策略可分为基本接纳策略、抢占接纳策略和空隙接纳策略三大类。基本接纳策略根据新事务的执行需求和系统当前负载决定是否接纳该新事务;抢占接纳策略应用于系统超载时,其总体思想是:即使系统容量饱和,也必须及时接纳某些紧急事务(一般包含硬实时的混合事务),但决定是否接纳该紧急事务的标准不仅仅是优先级,还要衡量系统为此得到的收益是否是大于损失;空隙接纳策略充分体现了实时事务的优越性,只有当负载小于阈值,CPU 有空隙时才接纳非实时部分。

ACMHRTT 涉及的属性有: D_i 为混合实时事务 T_i 的截止时间; t_{ai} 为混合实时事务 T_i 的到达时间; SF 为事务的松弛因子; t_{ri} 为混合实时事务 T_i 的真实时部分估算执行时间; t_{ci} 为混合实时事务 T_i 价值为零的时刻; t_{ri} 为混合实时事务 T_i 的真实时部分的剩余执行时间; t_0 为系统的当前时间; HRT 为混合实时事务集。

4.1 基本接纳策略

(1) 乐观无冲突策略 ONCP(Optimism Non-conflict Protocol) 此策略是一种理想化的策略,不考虑资源竞争。假设事务在系统中与其它事务不发生冲突,独占资源。

系统接纳新到达事务的条件是: $D_i > t_{ai} + SF \times t_{ri}$

(2) 价值有效策略 VEP(Value Efficiency Protocol) 如果混合实时事务 T_i 的 $D_i \leq t_{ai} + SF \times t_{ri}$,但系统接受 T_i 可获得收益。即:

$$T_i \in HHRT \quad V_i(t) = -y_{ni} + x_{ni} > 0$$

$$\text{或 } T_i \in HSRT \quad V_i(t) = x_{ri} - f(t - D_i) + x_{ni} > 0$$

$$\text{或 } T_i \in HFRT \quad V_i(t) = x_{ni} > 0$$

则此事务是可接纳的。

(3) 乐观串行策略 OSP(Optimism Serialization Protocol)

(1)(2)策略只是假设系统立即执行该事务,则该事务可能成功完成而提交,但在正常情况下,大多数 RTDBS 中活跃着多个事务,它们竞争包括 CPU 在内的各种资源,因此必须

考虑 CPU 时间总量是否足够分配给所有实时事务。

OSP 要求被系统接纳的事务 T_i 满足以下条件:

$$\textcircled{1} D_i \geq t_{ai} + SF \times t_{ri}$$

$$\textcircled{2} \sum_{T_j \in HRT} t_{ri} \times SF \leq \max(D_1, D_2, \dots) - t_0$$

(4) 悲观串行策略 PSP(Pessimism Serialization Protocol) 满足 OSP 和 VOSP 的事务可能有一种情况:存在两个事务 $T_i, T_j (i \geq j)$,事务 T_i 的优先级大于 T_j ,被 T_j 抢占,导致 T_i 因超时而夭折。为防止此现象发生,PSP 规定被系统接纳的事务 T_i 满足以下条件:

$$\textcircled{1} D_i \geq t_{ai} + SF \times t_{ri}$$

$$\textcircled{2} \sum_{T_j \in HRT} t_{ri} \times SF \leq \max(D_1, D_2, \dots) - t_0$$

$$\textcircled{3} \forall T_i, T_j \in HRT (i \geq j) \sum_{j=1}^i (t_{ri} \times SF) \leq (D_i - t_0)$$

4.2 抢占接纳策略

(1) 乐观价值优势抢占策略 OVPP(Optimism Priority of Value Preempt Protocol) 如果紧急事务 T_i 到达系统,即使系统满负荷,也应允许系统有条件地接纳高优先级的任务,其条件是存在优先级较低的任务,并且被抢占的任务的总价值小于该事务的价值,即 T_i 满足条件:

$$\textcircled{1} D_i \geq t_{ai} + SF \times t_{ri}$$

$$\textcircled{2} \exists TS \subset HRT \sum_{T_j \in TS} V(j) < V(i)$$

(2) 悲观价值优势抢占策略 PVPP(Pessimism Priority of Value Preempt Protocol) 如事务 T_i 抢占成功被系统接纳后,有可能优先级较其它事务低而不能按时执行,致使它在截止期之前得不到所需的时间,则此抢占是无意义的。为防止此情形,接纳事务 T_i 必须同时满足以下条件:

$$\textcircled{1} D_i \geq t_{ai} + SF \times t_{ri}$$

$$\textcircled{2} \exists TS \subset HRT \sum_{T_j \in TS} V(j) < V(i)$$

$$\textcircled{3} \sum_{T_j \in TS} t_{ri} \geq t_{ri} \times SF \cap (\forall T_j \in TS (D_j \leq D_i))$$

4.3 空隙接纳策略

实时数据库系统除了处理混合实时事务外,还要处理非实时事务,系统优先处理混合实时事务,只有当 CPU 有足够空闲才接纳非实时事务。系统接纳非实时事务 T_i 的条件是:

$$\frac{\sum_{T_i \in HRT} t_{ri}}{\max(t_{c1}, t_{c2}, \dots, t_{cn})} < \sigma \quad \text{其中 } \sigma \text{ 是阈值}$$

5 性能分析

混合实时事务接纳控制机制区分了事务中的真实时子事务和非实时子事务,综合考虑系统负载,减轻事务处理部件的负担,降低资源竞争程度,提高资源利用率,Nagy 和 Bestavros 提出的接纳控制协议,以及夏家莉提出的 IACM 都没有考虑混合实时事务情况,结果造成资源浪费,甚至可接纳的事务没有接纳,致使降低了系统效益。ACMHRTT 在处理混合实时事务时比传统的接纳控制协议更优。

6 需进一步研究的问题

对于一个混合事务,怎样将它有效地划分为真实时子事务和非实时子事务?开发怎样的应用程序确定混合事务的价

(下转第 134 页)

上的 16×16 区域内不断增加兴趣数量来进行多次仿真(每次仿真的节点数目都是4000),仿真结果如图4所示。从图4可以看出,在同一分布和节点数的情况下,兴趣数量的增加会增加路由跳数和消息数。并且在最坏情况下路由跳数以 $O(m)$ 增长,而且消息数的增长会小于兴趣的增长。

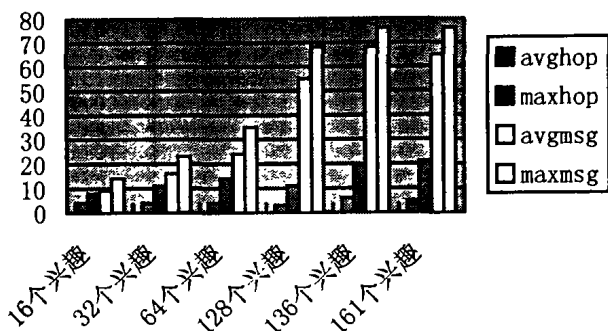


图4 同一兴趣分布下的兴趣增加

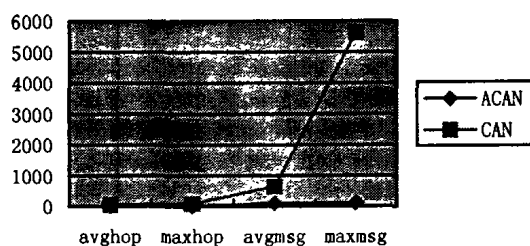


图5 MCAN 和 CAN 路由性能比较

5.3 MCAN 和 CAN 的比较

为了对 CAN 和 MCAN 两种情况下的路由进行比较,仿真将 MCAN 中10000个分布在 16×16 区域内的172个兴趣上的节点组织成一个二维 CAN 系统,对这10000个节点在这两种情况下的路由做分析(如图5)。从图5可以看出,CAN 中的路由路径长度是 $O(n^{1/4})$,消息数是 $O(n)$ 。由此可以得出结论,同样的节点在 MCAN 中可以获得比在 CAN 中更好的路由路径长度和路由消息数,即 MCAN 可以在保障系统可扩展

性条件下改善路由性能。

总结 本文提出 MCAN 上的兴趣聚类方法,利用 DHT 定位目标兴趣聚类,各个节点可以完全分布地通过局部信息成功路由到目标节点上,使得 MCAN 具有很好的扩展性。新节点可以随时地加入和离开系统,并且当节点失败时 MCAN 具有容错性。MCAN 在每一跳上都选择最小的邻居节点来转发路由,因而具有较低的延迟。经仿真表明 MCAN 的路由路径与系统中的节点数量没有关系,只和兴趣的分布和数量有关——路由在最好情况下至多经过 $O(m^{1/4})$ 跳就能到达目标节点,最坏情况下经过 $O(m)$ 跳能到达目标聚类。MCAN 仍存在很多不足,如节点的同时加入和离开的处理,以及如何应对拒绝服务攻击构建安全的 MCAN。在将来的工作中我们将对上面的两个问题进行研究,进一步完善 MCAN。

参考文献

- 1 Gnutella website: www.gnutella.com
- 2 Clarke I, et al. Freenet: A distributed anonymous information storage and retrieval system. Freenet project at freenet.sourceforge.net.
- 3 Rowston A, Druschel P. Pastry: Scalable, decentralized object location and routing for large scale peer-to-peer systems. Microsoft Research Ltd. Cambridge
- 4 Stoica I, et al. Chord: A scalable peer-to-peer lookup service for internet applications. ACM Sigcomm, 2001
- 5 Xu Zhiyong, Hu Yiming. SBARC: A Supernode Based Peer-to-Peer File Sharing System. <http://www.ececs.uc.edu/~oscar/papers>.
- 6 Sripanidkulchai K, Maggs B, Zhang Hui. Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems. Carnegie Mellon University
- 7 Schlosser M, et al. HyperCup-Shaping Up Peer-to-Peer Networks. Stanford University
- 8 Kamvar S D, Schlosser M T, Garcia-Molina H. The EigenTrust Algorithm for Reputation Management in P2P Networks. <http://www-db.stanford.edu/>

(上接第125页)

值,从而提高用户生产率?怎样使 RTDBMS 通用化?这些都是须进一步研究的问题。

总结 为提高实时数据库系统事务处理的成功率和可靠性,考虑事务的执行需求和价值,降低资源竞争程度,提高利用率,使系统收益最大化,针对现实世界中存在的混合事务,提出一种混合实时事务的接纳控制策略。分析了不同混合实时事务的价值模型、各种接纳控制策略及 ACMHRTT 算法,与传统的接纳控制协议作了简单比较分析,提出了须进一步研究的问题。

参考文献

- 1 Nagy S, Bestavros A. Admission control for soft-deadline transactions in ACCORD. In: Proc. of the 3rd IEEE Real-Time Technolo-

gy and Applications Symposium, Montreal, Canada, 1997. 160~165

- 2 Nagy S, Bestavros A. Value-cognizant admission control for RT-DB systems. In: Proc. of the 17th IEEE Real-time Systems Symposium, Washington D. C, USA, 1996. 230
- 3 Chakravarthy S, Hong D, Johnson T. Incorporating load factor into the scheduling of soft real-time transactions: [Technical Report TR94-024]. Department of Computer and Information Science, University of Florida, 1994
- 4 夏家莉. 适用于嵌入式实时数据库系统的接纳控制机制 IACM. 计算机学报, 2004, 27(3): 295~300
- 5 刘云生, 何冰, 冉龙波. 混合实时事务的延期单调速率调度算法及其可调度分析. 计算机学报, 2004, 27(3): 289~294
- 6 萨师煊, 王珊. 数据库系概论(第三版). 高等教育出版社, 2002. 2