文章编号:1005-8451(2005)03-0016-03

实时主动数据库框架的核心技术与实现

刘小平,张遂征
(铁道科学研究院 电子计算技术研究所,北京 100081)

摘 要:比较于传统的数据库,主动数据库有自己的一套体系架构和实现。首先对主动数据库的一般理论进行论述,给出主动数据库的框架;详细阐述由此框架延伸出的若干核心技术及其实现方法。

关键词:主动事务;触发事务;时间约束;事务期限
中图分类号:TP392 **文献标识码**:A

Principal issue and implementation for active real-time database

LIU Xiao-ping, ZHANG Sui-zheng
(Institute of Computing Technology, China Academy of Railway Sciences, Beijing 100081,China)

Abstract: Compared with the traditional DBMS, Active Real-Time Database System(ARTDBS) owned its architecture and implementation. It was firstly considered the general theory of ARTDBS followed by the architecture , then, expounded the implementation of the kernel matter derived from that architecture.

Key words: triggering transaction; triggered transaction; time constraint; deadline

随着信息技术在各行业的广泛和深入应用,数据库产品的要求也越来越高。主动数据库正是应各种需求发展起来的,目前主要应用于工业控制,民航,军事和股票等各种对时间要求比较严格的实时系统。比较于传统的数据库,主动数据库的应用处于初级阶段,各个系统的需求和实现方式也不尽相同,但是主动数据库有自己的一套体系架构和实现方法。

1 主动数据库框架和核心技术

1.1 框架

主动数据库的基本实现方式是:ECA(Event – Condition Action)当某个事件到达时,触发事务;判断条件是否被满足;若是则触发主动事务。这同传统的数据库没有多大差别,触发器就是这样的实现方式),它们的差别主要体现在时间约束(Time Constraint: Deadline)也就是说,对于主动数据库首先要考虑的问题是时间约束,在期限到达之前尽可能地完成更多的任务(Triggered Transaction)该要求的确定是主动数据库的所有研究方向的出发点和归宿点。

1.2 核心技术

(1)事务的调度策略:在逻辑上下文中,触发事务和主动事务的关系应该怎样?

(2)事务的调度优先权问题:触发事务和主动

收稿日期:2004-10-12
作者简介:刘小平,在读硕士研究生;张遂征,副研究员。

理对象,向服务器端查询对象的属性信息,并把属性信息显示在InfoFrame上。

4 结束语

国际标准化组织多年来致力于开放的地理信息规范的研究,并且制定了一套空间数据表达的规范化模型,而GML技术的出现也在许多GIS领域显示了它的应用潜力,在实现平台方面,.Net技术方兴未艾。应用这些规范与技术,我们提出的RGIS中间件为解决异构数据之间的数据融合作出了尝试。实践证明,这种技术的研究思想是行得通的。

参考文献:

[1] 贾利民,王英杰,秦 勇. 铁路地理信息系统(RGIS)的框架体系[J]. 中国铁道科学,2003,24(1).
[2] Kurt Cagle. XSL高级编程[M]. 韩 平,程永敬,董启雄. 北京:机械工业出版社,2002,4.

事务之间、主动事务之间、触发和主动事务同其它事务之间，谁应该最具有优先权？

（3）对于一个事务，由于不可能常驻内存（Main Memory Resident），所以交换是必然的，而与交换关系紧密的缓存和预取成了主动数据库一个重要的研究方向；

（4）主动事务的发现问题：前面提到触发事务在条件满足时要触发主动事务，主动事务通常并不是一个简单的事务，可能是若干个事务的混合事务，而他们的组合形成和发现成了主动数据库的一个重要的问题（Heart of Matter）。

2　主动数据库基本框架的核心技术与实现

2.1　事务的调度策略

基于触发事务和主动事务的调度策略一般有 3 种，分别是立即调度、延迟调度和分发调度。这里考虑两种事务的调度模型：立即调度（如图 1）和延迟调度（图 2）。

图1说明：即使条件满足在触发事务完成之前，主动事务也不发生，直到触发事务完成，这是延迟触发的方式。

图 2 说明：条件满足，主动事务（Triggered Transaction）马上发生，这是立即触发。

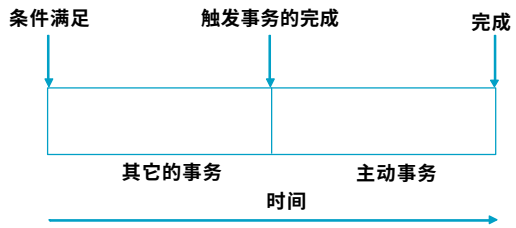

图 1　延迟触发方式

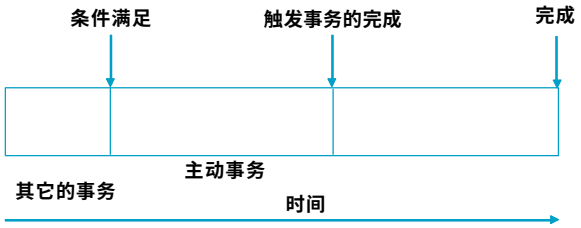

图 2　立即触发方式

2.2　事务的调度分析和优先权的确定

立即触发和延迟触发都把触发事务和主动事务看作一个整体，它们的实现是同步的，为两个平等的事务，调度的依据，即优先权的计算如下。

（1）主动事务生存期的确定

Deadline tt = Min Deadline t, Temporal D）其中 Deadline tt 是主动事务的生存期 Deadline t 是触发事务的生存期，Temporal D 是主动事务的临时数据的生存期。

（2）主动事务的重要性

Criticalitytt=aCriticalityt+b（DataStateD 其中a，b表示权重。

（3）数据状态依赖（Data State Dependent）体现了对触发条件数据的依赖性。

（4）事务优先权的确立

事务优先权的确立是事务的生存期和重要性的函数。一般而言，触发事务和主动事务的优先权要比其他的事务要高，这是可以接受的，如图 3 所示。

图 3　事务的优先权方向

2.3　主动实时数据库的缓存管理

缓存管理指的是规定在内存中应该放置什么数据和放置多久的策略集。

S={strategy | strategy (data, timespan) data∈Memory, timespan∈TimeSpan}

假设：由于在各种触发策略中，分发触发方式具有并行的特点，同比其他的触发方式（立即，延迟）具有较大的优势。所以采用分发的模型。

缓存的模型 对于每个事务（Active TRAC，AT）都有Private Buffer PB）所有的这些事务共享一个Global Buffer GB），访问数据库数据。

算法如下：

（1）虚拟期限，基于触发事务的触发时刻、触发期限和页预取优先权的不确定

事务 Tj的虚拟期限：((Dti + Ati) / 2 + Cj) / pj 其中，Dti 是触发事务T的进入时刻；Ati 是事务T的事务期限；Cj 是时间约束；Pj 是 触发事务 T 触发主动事务 Tj 的概率；

（2）全局缓存的构造 reuse pages，anticipated pages，free pages）分成3条链；

（3）缓存的分配和替换，有4个阶段：

a. System Entry：每个事务Ti有由（Pj，Nij）组成的集合Uti：

Uti={(Pj,Nij)|Pj is a page Nij: the times of the Pj touched}

b. Page Touch Pj is touched one time， Nij - = 1；

if Nij＝0 then (Pj,Nij) is removed from Uti

if Pj∈Tk then reposition in the reuse pages pool according to the virtual deadline of Tk.

Elseif Pj is a candidate for prefetching, then move to the anticipated pool

Else go to the free pool.

c. Transaction Termination：Clear

d. Page Replacement：Clean free pool→dirty free pool→anticipated pool with a comparison of virtual deadline, the highest one first, the lowest last→reuse pool

（4）预取的实现

预取队列的设计：对应于每一个磁盘，都有一个预取队列。其原因有2：a． 控制存取的无限增长（各种不同级别的预取在一个事务的到来后比立即的命令存取要多）；b． 消除冗余。

定义：所有的命令存取总是比预取有更高的优先权。

实现：每一个预取请求都含有这样一个二元组结构：（Pr，VDr），其中，Pr 表示请求的页；VDr 表示该页的虚拟期限。顺序是最近的期限最先服务，在一个非空的队列中，若已经存在一个和当前要申请入队列的请求有相同目标页的请求，当前的请求失败。

2.4 主动实时数据库的并行事件发现

很多事件发现都是采用顺序的，同步的方式，这同事实有时候并不能完全等同，许多同时发生的事情，并非同步的。这里介绍并行的事件发现方式。

（1）基本概念

每个事件是某类事件的一个实例，表示为：<Typename>.<timestamp>

混合事件是由基本事件经过构造成的。

事件类型假设（3大类）：

Constructor：从不同的源产生新的：BEFORE，AND

Collector：从不同的源集合但不产生新的：OR

Selector：从源中来但只有一部分属于该源的子集：FIRST，LAST

这里用到的为：

BEFORE(A,B)：(a,b,t)∈A×B×TIME，t：b的时间戳

AND(A,B)：(a,b,t)∈A×B×TIME，t：最近的时间戳

OR(A,B)：c∈A∪B，与时间顺序无关

FIRST(A)：the oldest instance of A：a，as to '\a'with timestamp≥timestamp Of a，a'∈A；

LAST(A)：the most recent instance of A：a，as to '\a'with timestamp≤timestamp Of a，a'∈A

语义构造：

ALL：ALL(AND(A,B)) => |S| = |A| * |B| && S=<A×B×TIME

（2）混合事件的发现和处理策略

--PPS pipelining parallelism strategy）每个发现节点是归属于某一个特定的处理流程，如图4。

Tnks	A	B	C	D	AND	BEFORE	OR
1	A1	B4					
2			C5	D5			A1
3	A9		C8	D7	(C5,D5,5)		
4					(C5,D7,7)	(B4,(C5,D5,5),5)	A9
5					(C8,D5,8)	(B4,(C5,D7,7),7)	(B4,(C5,D5,5),5)
6					(C8,D7,8)	(B4,(C8,D5,8),8)	(B4,(C5,D7,7),7)
7						(B4,(C8,D7,8),8)	(B4,(C8,D5,8),8)
8							(B4,(C8,D7,8),8)

图4 处理流程举例

3 结束语

作为主动数据库的核心技术，并行事件发现的实现探讨意义勿容质疑。这里作了尝试，虽然在假设上只采用了Collector（OR），Constructor（AND，BEFORE）和SELECTOR(FIRST,LAST)，但他们的可扩展性可以应用到主动数据库的其他语义上。

参考文献：

[1] U.jaeger, J.K. obermaier. Paallel Event Detection in ActiveDatabase Systems: The Heart of the Matter[J]. ARTDB-97 Como, Italy, September 8—9, 1997.