第 33 卷 第 7 期 计 算 机 工 程 2007 年 4 月
Vol.33 No.7 Computer Engineering April 2007
·软件技术与数据库· 文章编号：1000—3428(2007)07—0047—03 文献标识码：A 中图分类号：TP311

基于协同缓存的分布式数据库更新机制研究

符青云，刘心松
(电子科技大学 8010 研究室，成都 610054)

摘 要：减小数据库事务中的写操作开销对于分布式数据库系统的性能而言很关键。该文提出了一种基于协同缓存技术的分布式数据库更新机制，通过在分布式数据库服务器节点物理内存之上构建全局协同缓冲池，并利用其缓存写入记录，减小了数据库事务中的磁盘访问开销，研究了基于协同缓存的分布式数据库更新机制与其在该机制下事务性能改进。

关键词：分布式数据库事务；基于协同缓存的分布式数据库更新机制；协同缓冲池

Distributed Database Update Mechanism Based on Cooperate Cache

FU Qingyun, LIU Xinsong
(8010 Group, University of Electronic Science and Technology of China, Chengdu 610054)

【Abstract】 Cost reduction of write operation in database transaction is important in distributed database system design. The paper presents a new update mechanism, named cooperate cache-based distributed database update mechanism (CCDDU). It reduces disk access cost in database transaction by means of cooperative caching pool constructed with physical memory among the servers. This paper describes cooperate cache-based distributed database update mechanism, focusing on how the mechanism improves transaction performance and its availability.

【Key words】 Distributed database transaction; Cooperate cache-based distributed database update mechanism; Cooperate caching pool

1 概述

分布式数据库系统由于具有高吞吐量、高可用性和高可扩展性等优点，而在超大容量数据存储与处理领域得到了广泛应用。由于服务节点间的同步开销与服务器节点内的磁盘访问开销，使得分布式数据库系统的事务处理时间很长。如何减小更新事务的响应时间，从而提高事务性能是分布式数据库研究中的重要问题。

为了减少事务中写日志的开销，提出了很多改进的提交协议，比如：预提交协议(Presumed Commit Protocols, PrC)，预定丢弃协议(Presumed Abort Protocols, PrA)，早期准备协议(Early Perpard Protocols, EP)，协同日志协议(Coordinator Log Protocol, CL)，显式提交协议(Implicit Yes-Vote Protocol, IYV)。

上面的协议关注于减少日志的写入次数，但是对于如何提高日志的写入速度没有相应的研究。

邱等人提出了一种基于全局协同缓冲池的数据库日志更新机制[1]，由于不必在事务提交前后立即将日志写入磁盘，而仅仅将其缓存在协同缓冲池中，仅当协同缓冲池满或者节点轻载时，才将缓冲池中的数据写入磁盘中，改进了提交协议的性能。

但是由于数据库更新事务中记录写入操作中的磁盘访问次数并没有减小，因此在数据库更新很频繁时，连续不断的磁盘访问会使得磁盘过载，并使得磁盘最终成为系统瓶颈。

本文提出了一种基于协同缓存的分布式数据库更新机制，其目的是减小数据库事务中的磁盘操作。通过服务节点间协同缓存更新的记录，不必在事务提交时马上更新数据库文件，只有协同缓存已满，或者节点负载很轻时，才将 Cache 中的内容刷新到磁盘上。该机制可以显著改进分布式数据库系统的性能。

2 基于协同缓存的分布式数据库更新机制

2.1 相关技术

一种减少数据库事务响应时间的方法是主存数据库，由于主存数据库在内存中维护数据，因此在数据库更新时不存在磁盘操作，只有数据备份与日志存储涉及磁盘读写，事务性能得到了很大的提高。但是主存数据库技术存在两个严重问题：

(1)通常的数据库文件很大，难以完全放入物理内存；

(2)主存数据库存在丢失数据的危险，如果物理内存失效，则存储在数据库中的所有数据将会丢失。

尽管主存数据库存在上面的缺点，但其利用物理内存获取高性能的方式值得借鉴。随着网络技术的发展，高速LAN环境的传输速率已经超过了 1 000Mbps，LAN环境下典型的通信延迟不超过 15μs[2]。但是磁盘的性能改进较小，典型的寻道延迟在 8ms左右，物理内存的访问速率远远高于高速网络与磁盘系统，与网络传输延迟和磁盘寻道和读写延迟相比，内存的访问延迟可以忽略。

基于上面的认识，邱等人提出了基于协同内存缓存机制的数据库日志写入策略[1]，有效地改进了数据库系统的性能。本文使用与上面类似的思想构建了一个巨大的分布式协同缓冲池，用于写入记录的缓存，磁盘访问操作的次数可以大大减小，从而提高了数据库事务的性能。

这里，对CCDDU与其它方法在记录缓存方面的不同给出

作者简介：符青云(1975－)，男，博士生，主研方向：分布式并行系统；刘心松，教授、博导

收稿日期：2006-04-10 **E-mail：**qingyunfu@sohu.com

简要描述[3]。

(1)在 ADO，ADO.NET 和 DataSnap 技术中，记录或者缓存于单个的客户端内存中，或者缓存于单个的应用服务器内存中，可以利用的缓存空间仅仅是单个节点分配用于数据缓存的空间；而在 CCDDU 中，通过协同缓存机制，得到的最大缓存空间等于服务节点可用缓存空间之和。由于上述优点，使得 CCDDU 机制非常具有吸引力。

(2)CCDDU 机制构建于分布式数据库系统中，而其它的方法，比如 MicroSoft 的 ADO、ADO.NET，Borland 的 DataSnap 技术中的数据缓存机制构建于客户端或者应用服务器中。在不同的项目中，后者为了利用数据缓存技术，需要在客户端或者应用服务器中添加该特性，增加了开发人员的工作量，而对 CCDDU 机制而言，由于数据缓存技术在数据库系统中实现，对软件开发人员完全透明，没有额外的工作。

2.2 数据库更新过程

2.2.1 参与分布式数据库事务的两类主机

定义 1 事务发起者 R

事务发起者 R 指分布式数据库服务器系统中发起事务的节点。

定义 2 事务参与者 A

事务参与者 A 指分布式数据库服务器系统中按照事务发起者的请求，参与数据缓存与数据库更新的节点。

数据库事务中允许多个参与者，参与者的数目依赖于可用性要求，所要求的可用性越高，则参与者数目越多。参与者可以通过数据库系统管理员预先定义，也可以在运行时根据节点资源的可用性动态计算而确定。每个参与者不必缓存所有记录，只需缓存某些数据，以便减小 RAM 开销并最大化 RAM 的利用率。本文假定系统内的所有节点参与协同缓存，协同缓存池中没有重复记录。

2.2.2 基于协同缓存的分布式数据库更新过程

当用户触发数据库事务时，数据库服务器节点按照下列过程为用户协同提供服务：

(1)发起者 R 向外发送请求消息与要更新或者插入的数据记录，并为新的记录在内存中创建缓存表；

(2)参与者 A 接收消息与待更新的数据记录，并在其主存中为接收的记录创建缓存表；

(3)R 与 A 决定将数据记录插入缓存表，还是用新值更新缓存表中的记录；

(4)当满足回写条件时，节点将缓存的记录更新到数据库中。

2.2.3 数据一致性问题

数据一致性问题在数据库系统设计中很重要，对本系统而言，数据一致性问题存在于两个方面：(1)服务节点间的数据一致性；(2)协同缓冲池中的记录与数据库中存储记录的数据一致性。服务节点间的数据一致性已经有了很多有用的研究成果，因而这里只讨论协同缓冲池与数据库之间的数据一致性问题。

数据库系统处理数据一致性问题的方式如下：

(1)如果是记录更新请求，待更新的记录将会缓存在协同缓冲池中。

(2)如果是数据库读请求，则与请求表相关的缓存记录首先被刷新到数据库中，并释放在协同缓冲池中占据的空间。然后，数据库系统按照请求返回记录。

(3)如果协同缓冲池的剩余空间低于某个门限，则对当前未加锁的表执行回写操作，操作结束后，释放占据的缓存空间。

2.3 基于 CCDDU 机制的分布式数据库系统的可用性分析

对基于协同缓存的分布式数据库系统而言，记录不必在事务提交后就马上写入磁盘，而是缓存在某些节点的主存中。因而仅当所有节点出现故障，才会导致数据完全丢失。

由于记录只在主存中缓存很小一段时间，在这个时间段内，所有节点失效的情况很少见。假定每个节点的失效概率为 λ，并且节点的失效概率相互无关。则对一个 n 节点的缓存系统而言，整个系统完全不能对外提供服务的概率 p_f 为

$$p_f(n) = \lambda^n \tag{1}$$

假定每个节点的平均无故障工作时间为 100h，为确保数据安全而引入的强制刷新周期为 5min，则对于一个 5 节点组成的缓冲池而言，失效概率为

$$p_f = \left(\frac{5}{100 * 60}\right)^5 = 5.78 \times 10^{-10}$$

这表明基于协同缓存的分布式数据库更新机制不会产生明显的可用性下降。事实上，由于节点的平均无故障工作时间远远大于 100h，因此实际系统的失效概率会非常低，远远小于上面得到的结果。

3 实验

3.1 实验环境

实验采用图 1 的环境验证本文思想。需要指出的是，实验并没有修改数据库系统的源码，而是在每个节点加入一个应用服务器进程，由应用服务器进程利用操作系统提供的 ADO 数据访问对象中的记录缓存功能实现写入记录的缓存，以便简化实现，同时又反映本文更新策略的行为特征。

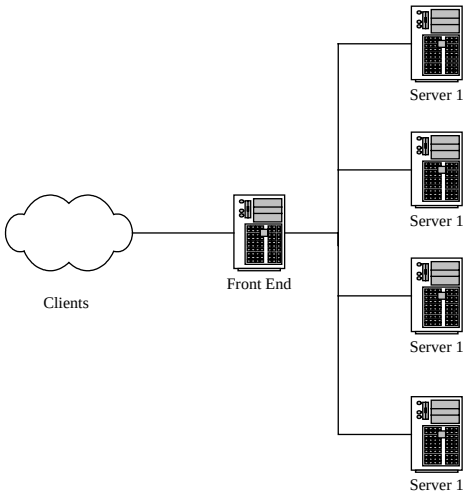

图 1 实验环境

图 1 中所有服务节点都是 1.7GHz 的 P4，256MB RAM 的 PC，硬盘为 80GB 7200RPM，采用 Windows XP SP 操作系统与 MySQL 4.01 for Windows。实验客户端由 5 个 1.0GHz 的 Althon，256MB RAM 的 PC 组成。客户端与服务器连接在同一个 100Mbps 的快速以太网交换机上。每个服务器上用于数据缓存的 DataSet 组件由 ADO 对象提供，设置其容量为 32MB。所有数据库中建立 5 张表，每个客户端只对其中的一张表执行插入操作，服务节点的数据库全冗余，客户端生成记录，并将记录与插入请求发送到前端节点，由前端节点将数据转发给应用服务器。

采用协同缓存机制的系统和没有采用协同缓存机制的系统之间的区别在于收到记录插入请求时前端服务节点的行

为。对于采用协同缓存技术的系统而言，前端按照可用性要求从后端选择某些节点并将请求多播到选择的节点。对于没有采用协同缓存技术的系统而言，前端将请求广播到后端所有节点。

3.2 实验结果

每个客户端生成 2.5MB~40MB 的记录，每条记录长度为 1KB，字段值由客户端随机生成。当所有客户端发送实验结束的消息到前端时，前端向所有应用服务器广播消息，应用服务器提交所有缓存在本节点 DataSet 中的记录，这将使得所有在协同缓存池中缓存的数据被提交到 MySQL 数据库中。最后应用服务器进程通过 MySQL 提供的数据库复制工具执行节点间数据库同步操作。

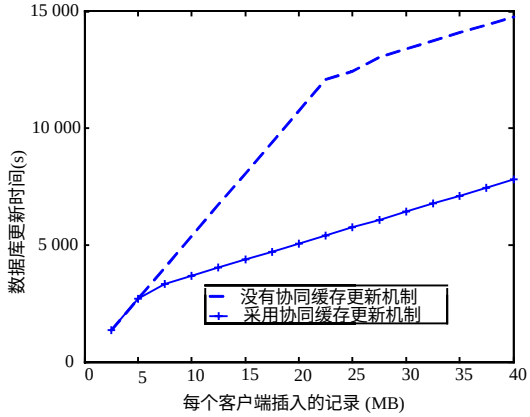

图 2 实验结果对比

从图 2 可以看出，当总的记录规模小于可用缓存规模时，采用协同缓存机制的系统和没有采用协同缓存机制的系统性能差别很小，因为记录可以完全缓存在数据集中，并一次性提交到数据库中。当插入的记录大于数据集规模时，采用协同缓存机制的系统性能明显优于没有采用协同缓存机制的系统，插入记录的突发性越强，两种更新机制之间的性能差距越大，因为对基于协同缓存机制的系统而言，更多的插入记录可以在协同缓冲池中缓存，减小了由于协同缓冲池容量有限而导致的磁盘强制写入操作。当写入记录容量大于协同缓存容量时，二者的性能差距减小，因为协同缓冲池容量不足导致协同缓存机制仍然存在很多记录的强制写入操作；但是由于协同缓存池可以缓存更多的写入记录，因此仍然具有较少的写入开销。

4 结论

本文提出了一种基于协同缓冲池的分布式数据库更新机制，同时与未采用该技术的常规更新机制进行了对比，表明该机制具有很好的性能。随后的研究将着眼于系统性能的定量研究，并针对主流数据库与异构环境开发能够提供协同缓存更新的服务器组件。

参考文献

1 邱元杰，刘心松，杨　峰. 一种高效的分布式数据库日志机制[J]. 计算机研究与发展, 2004, 41(11).

2 Ciaccio G, Thlert M, Schnor B. Exploiting Gigabit Ethernet Capacity for Cluster Applications[C]//Proc. of the 27th Annual IEEE Conf. on Local Computer Networks. 2002.

3 李　维. Delphi 7 高效数据库程序设计[M]. 北京: 机械工业出版社, 2003.

（上接第 37 页）

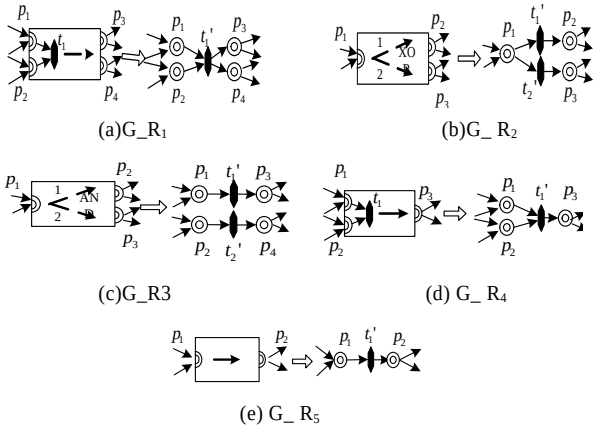

图 4 柔性工作流模型组件级化简规则

3 结论

从目前的趋势来看，工作流系统正向大型、灵活、动态、自适应和互操作等方向发展，过程验证扮演着越来越重要的角色。但是，柔性工作流验证领域的研究还比较欠缺。本文在这方面做了一些探索，把过程合理化验证和规约验证技术应用于基于交互学习的柔性工作流建模中的形式化验证，取得了较好效果。基于 Petri 网的合理性验证的优点在于：合理性定义清晰；验证方法具有良好的形式化基础；可验证的问题包括初始化、结束、迹等价、死锁、活锁、安全性和有界性等。规约技术的优点在于：能够在特性保持的前提下，将模型缩减到适当规模；将化简技术总结为规则集的形式，易于理解和操作。

同时，提出了没有解决的问题和将来的工作：(1)可达性、覆盖性等性质的验证需要构造可达图或可达树，在对大型模型进行操作时可能造成状态空间爆炸；(2)有些推导关系是有条件的，如要求过程模型是自由选择 Petri 网等；(3)应给出一定限制条件下的规则集完备性；(4)还需对过程数据流和资源流进行验证；(5)对更大规模的、复杂的过程化简研究还不足。

致谢：在写作的过程中，时力科技提供了良好的科研环境，开发中心的同事们也给予了很大帮助，在此表示感谢。

参考文献

1 范玉顺，王　刚，高　展. 企业建模理论与方法学导论[M]. 北京: 清华大学出版社, 2001.

2 张绍华，李赛寒，张世超. 基于交互学习的柔性工作流[J]. 小型微型计算机系统, 2005, 26(7): 1270-1274.

3 VanderAalst W M P. The Application of Petri Nets to Workflow Management[J]. Journal of Circuits, System and Computers, 1998, 8(1): 21-66.

4 周建涛，史美林，叶新铭. 工作流过程建模中的形式化验证技术[J]. 计算机研究与发展, 2005, 42(1): 1-9.