# 数据库加速引擎技术研究*)

王元珍[1] 龚卫华[1] 周英飙[1]
(华中科技大学计算机科学与技术学院 武汉 430074)

**摘 要** 数据库加速引擎是位于企业级数据库应用和数据库服务器之间,使用标准接口与底层数据库服务器通信,采用集群方式实现并行处理而不依赖于集成硬件,具有通用性的软件加速系统,能适应对性能要求较高的OLTP应用,该系统的性价比高,可扩展性好,市场前景非常广阔。
**关键词** 数据库集群,联机事务处理,数据分片,线索机制

## Research on the Accelerating Engine Technique of Database

WANG Yuan-Zhen[1] GONG Wei-Hua[1] ZHOU Yin-Biao[1]
(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract** The accelerating engine of database is a general accelerating system through the software technique, which lies between the enterprise applications of database and database server, uses standard interface to communicate with bottom database servers, adopts the method of cluster to realize the parallel processing and is not dependent on integrating hardware, can be appropriate for the high performance requirement of OLTP. This system has the property of high cost performance and fine scalablity. The prospect of market will be very wide.
**Keywords** Database cluster, OLTP, Data partition, Thread mechanism

## 1 引言

针对企业级数据库管理系统随着数据量和并发用户量的大量增加,联机事务(OLTP)处理效率大幅度下降,难以满足企业发展需要尤其是数据集中化的需要,研究高效的数据库加速引擎成为解决该问题的瓶颈。目前,国际上已有第三方的加速产品可供用户选择,但它们都只针对目前数据库产品存在的某一方面的性能问题而提出自己的加速方案:MTI公司推出了V-Cache数据库加速产品,将用户经常访问的表从数据库系统中导出到自带的大容量DRAM存储器之中,由V-Cache完成对于这些表的请求;SeekSystems公司的Fas-File RAID数据库加速产品,使用Adaptive Cache技术,避免了传统FIFO或LRU算法的CACHE的颠簸问题。可是二者都是一种集成软硬件的加速产品,需要大容量DRAM或专用RAID系统支持,价格昂贵,并且不具备通用性。而采用软件加速的产品有DISC公司的OMNIDEX查询加速器只适合于大部分只读的联机分析处理和HyperRoll公司提出了仅适合于分析和报表任务的加速策略。国内只有浪潮软件的海量实时数据库LCMD对企业级数据库加速有一定作用,但海量实时数据库对硬件依赖及要求都比较高。

因此我们提出一种更通用的企业级数据库加速引擎方案,这种加速引擎位于企业级数据库应用和数据库服务器之间,使用标准接口与底层数据库服务器通信,采用集群技术实现并行处理而不依赖于集成的硬件,能适应对性能要求较高的OLTP应用,在大型数据库应用中大显身手,具有非常广阔的市场前景。

## 2 数据库集群技术

在企业级应用中,用户对于数据库的性能尤其是联机事务处理(OLTP)的性能要求很高,需要数据库实时处理数据请求。而OLTP事务的特点是:包含大量的简单的小事务,频繁地更新事务,要求高度的并发性,对响应时间和吞吐量有很高的要求。对于OLTP应用,当前的数据库管理系统在企业级的应用中已经不堪重负,虽然DBMS自身提供了性能优化功能,用户可以通过调整数据库系统的参数如CPU参数、内存参数和I/O参数等来提高性能,但是参数调整只能有限地提高数据库系统的性能。因此,采用并行数据库[1]技术是提高应用性能的有效途径,通过并行地处理大量地的小事务,可以缩短其响应时间,提高吞吐量。

并行处理可以在对称多处理机系统(SMP)和多机系统上实现,现代通用数据库系统已经能够有效地利用对称多处理器资源,但是单机系统的系统负载能力总是有限度的,当系统负载达到极限时,数据库系统整体的运作效率就会严重下降。解决这个问题的根本方法,就是采用多机分布式计算方案,而数据库集群[2]就是其中一种有效的解决方案,数据库集群是一组完整的自治的计算处理单元(节点),每台均运行有数据库系统,通过高速专用网络或者商业通用网络互连,彼此协同计算,作为统一的数据库系统提供服务。对称多处理虽然易于管理和配置,但是在可伸缩性和性能上要逊于集群,规模也受到限制,数据库集群在处理OLTP应用时,具有如下优点:

(1)完全的可伸缩性:一个集群可以具有数十台以至于数百台机器,在应用负载增加时,可以通过增建软硬件资源的方法来保证应用性能。

(2)高性能:数据库集群中通过将负载均衡到各个节点,对事务进行并行处理,提高性能。

(3)高性能价格比:数据库集群能够以较低的价格获得与一台大型主机运行的数据库系统一样的事务处理能力。

(4)资源共享:集群系统能有效地支持不同位置的用户对

*)本课题获得国家信息产业部电子发展基金资助项目[2004]42号。王元珍 教授,博士生导师,主要研究方向为分布式多媒体数据库、并行数据库、数据库安全。龚卫华 博士研究生。

信息和资源(硬件和软件)的共享。

因此本文的企业级数据库加速引擎方案就是采用无共享的数据库集群技术并行处理联机事务(OLTP)实现高性价比,如图1所示。

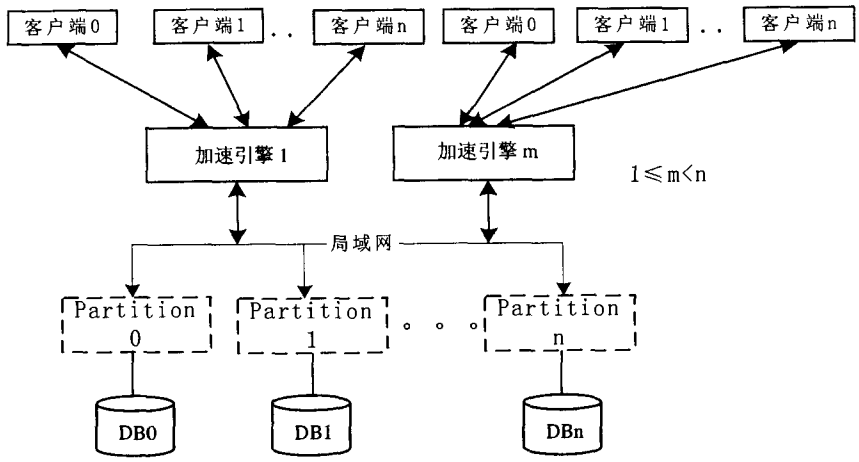


图1 基于加速引擎的数据库集群

从图中可以看出,该体系结构为了避免单个加速引擎因众多客户端的应用形成瓶颈而采用多个加速引擎并行处理以提高系统效率。

## 3 加速引擎结构设计

数据库集群中的各局部数据库系统的自治性使其只能保证局部系统的孤立性,无法与其相邻的站点通信和协作。而处于上层的加速引擎从外部应用程序的视角来看是一个单独的虚拟DBMS,具有模式集成及全局事务[3]协作分工的能力,即引擎根据自己的元数据字典来管理模式集成和将客户端的全局事务划分成对应集群站点上执行的全局子事务,每一个全局子事务就是在对应局部数据库上执行的局部事务[3],可见全局事务由一个或多个局部事务组成,此外引擎中的事务管理器采用2LSR方法[4]使各站点上协作的局部事务能保证全局事务的一致性。

从体系结构上看,加速引擎位于客户端应用和数据库服务器之间负责协调客户端事务在各底层数据库系统的并行处理,其内部结构包括:上层通信管理,语法分析,事务划分,事务管理,恢复管理,下层连接管理等。如图2所示。

从客户端接收全局事务的操作流程为:

1. 上层通信管理接受应用程序通过ODBC或JDBC接口发送的SQL语句或其他命令,通信模块启动相应的工作线程对其作以下处理。

2. 首先语法分析模块对语句进行词法分析,语法分析,生成中间结构。

3. 然后事务划分模块根据元数据字典中表划分或表复制的信息以及查询谓词,对生成的语句中间结构进行进一步的分解或转换。

4. 事务管理器负责事务的并发调度[4,5]和决定提交或回滚事务以确保跨站点事务的可串行化,接受上层模块传递已分解或变换的SQL语句结构,登记其访问控制信息,记录全局日志,并将子语句通过统一数据访问接口分发到局部站点上执行。

5. 通过底层连接管理模块等待所有子语句执行完毕后将全部结果交回事务管理模块处理,再经过事务管理模块返回结果给应用程序。

6. 恢复管理器模块根据全局事务日志在异常情况下进行事务的恢复操作。

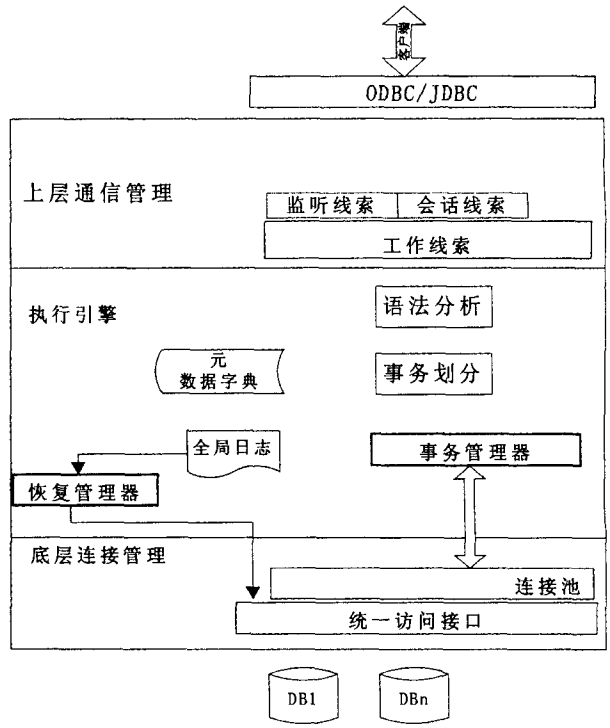


图2 加速引擎内部体系结构图

在该体系结构中加速引擎的核心思想是通过数据库集群并行处理联机事务的软件技术实现系统整体性能的提升。本项目研究了经典的和最新的数据加速技术,将其有机地融合在一起,限于文章篇幅本文将着重介绍加速引擎系统结构中所主要采用的技术:线索机制、数据分片技术和异构数据库集成。

### 3.1 加速引擎中的线索机制

运行于数据库服务器上的加速引擎核心系统充分利用了现代操作系统的多线索特性,建立了多种线索,主要有工作线索、会话线索和连接监听线索三种,各线索之间相互配合、协同工作,共同完成客户的请求。如图2所示,其中工作线索的个数可以在配置文件中设置。

当加速引擎服务器启动时,将根据配置文件设置的工作线索数创建工作线索、会话监听线索等,这些线索在系统运行

过程中处于活动状态。当客户端第一次连接加速引擎系统时，客户端发送一个连接消息，当会话监听线索接收到这个连接后，产生建立一个会话线索，该会话线索保持与客户端的连接，接收来自客户的消息，保存到系统任务队列的客户端请求队列中。系统任务队列保持两个队列：客户端的请求队列、数据库运行对象队列。空闲的工作线索不断检查这两个队列（检查客户端请求队列优先），从这两个队列中取出任务，当工作线索获得一个活动的连接套接字后，工作线索的流程为：

1. 接收客户的 PL/DMSQL 命令。

2. 调用语法分析器（Parser）构造语法树，Parser 在需要时调用词法分析器（Lexer）。

3. 查询元数据字典，把语法树中的标识符转换为内部 ID。

4. 进行授权检查。

5. 根据数据划分和数据复制的信息，将其划分为若干子语句。

6. 记录全局日志。

7. 记录子事务的存取信息。

8. 利用统一数据访问接口将语句发送给下层数据库，等待返回结果。

9. 合并结果。

10. 利用套接字还回结果信息。

因此，系统的处理主要由工作线索完成，工作线索负责处理客户请求，并返回客户结果信息。

线索机制的优点是：允许多个客户端程序同时运行，每个客户端拥有独立的线索空间，各类线索分工明确相互配合、协同工作，线索间独立并发运行，共享部分信息资源，增加了加速引擎的并行处理能力，减少了系统资源切换开销，较大地提高了处理并发事务的效率。

### 3.2 数据分片技术

数据库集群加速的有效性要求事务尽可能平均分配到结点上执行，为了避免数据倾斜和执行倾斜，导致某些结点成为热点，设计优良的划分策略能够达到均匀的数据分布，有利于数据服务器之间的负载平衡。许多研究表明，数据分片对于并行系统的性能具有很大的影响，划分策略包括 Round-Robin、Range、Hash、Hybrid-Range 等一维划分方法，以及 BERD、CMD、MAGIC 等多维数据划分方法。

加速引擎采用常用的一维 Hash 数据分片方法对数据库集群上的站点进行并行数据操纵保持各站点间数据的一致性，事务划分模块根据事务类型对 SQL 语句进行以下处理：

插入（insert）：根据元组值和划分/复制信息发送到单个或多个站点。

删除（delete）：如果存在划分列上的谓词，根据划分/复制信息将删除语句发送到单个或多个站点，否则发往全部站点。

更新（update）：如果更新划分列，则会引起数据迁移，需要将更新转换为查询-删除-插入序列。然后根据谓词和划分/复制决定将语句发送到单个，多个或者全部站点。如果更新非划分列，则与删除类似，根据谓词和划分/复制决定将删除语句发送到单个，多个或者全部站点。

查询（select）：集群对于查询的加速主要是查询内并行，如果存在划分列上的谓词，根据划分信息将删除语句发送到单个或多个站点。否则发送到所有站点，如果查询涉及到多个站点，事务管理器还需要合并查询结果，对于普通查询，集函数，分组，排序等需要相关处理。

因此，数据分片技术按最常用的谓词将关系表划分为小的片断，并把这些小片均匀地分布到系统中的所有数据服务器上，使得加速引擎能够并行地读写关系表，相当于增加了 I/O 带宽，在很大程度上消除了 I/O 性能瓶颈。数据分片降低了数据的聚集度，对于表的 I/O 操作能够由多个数据服务器共同完成，因此能够减少查询的响应时间并提高整个系统的吞吐量。

此外，加速引擎中还使用并行 SQL 处理技术实现了查询内的并行性和操作内的并行性，从而极大地提高了系统的并发性能。

### 3.3 异构数据库集成

加速引擎的上层通信及底层连接管理模块都采用通用的访问接口如 ODBC，其优点是不需修改集群站点上数据库的 API 接口，而且客户端应用程序也不必设计专门的通讯接口，从而实现了异构、自治数据库的集成，具有广泛的通用性和移植性。在这种应用环境中，引擎可以集成不同硬件和操作系统平台以及不同数据模型的数据源，在统一的集成数据库上开发新的应用，同时也保证各个站点数据库的自治性还支持旧有应用，具有兼容性。因此加速引擎不仅适合于典型的多数据库集群系统应用，也适合于异构数据的集成应用。

## 4 性能试验

TPC-C 作为联机事务处理（OLTP）的性能基准测试标准已经取得了广泛的认可，它模拟了非常接近现实环境的商业应用软件应用环境[6]。这个商业环境是一个批发商的货物管理流，该批发公司有 N 个仓库，每个仓库共九张关系表维持大约 60 万条记录量。在运行时，TPC-C 测试程序模拟终端用户按照规定的混合比执行五种事务，其中包括三个更新事务：新订单、付款、发货；两个查询事务：订单查询，库存查询，并且事务的比率和负载的混合都不会随着系统的负载和响应时间的增大而改变。

测试方法是：在五种事务并发执行的情况下，测试一定时间间隔内所执行的新订单事务的数量，系统性能由 TPC-C 吞吐率衡量，单位为 tpmC，是指每分钟内系统处理的新订单事务数量，其性价比定义为总价格/性能[7]，单位是美元 $/tpmC。

加速引擎测试方案如图 1 所示，加速引擎服务器使用 2 台机器并行，运行 3 个 TPC-C 测试程序作为客户端，模拟 1680 个用户的负载，在 8 个站点数据库服务器上的仓库总量达到 21×8 个（每个站点数据库是采用国产自主知识版权 DM4 装载 21 个仓库），每个仓库中有 8 张表按照仓库号在站点间进行 Hash 划分，而 Item 表在站点间进行复制，建立符合 OLTP 商业应用的环境。新订单事务统计图为如图 3。

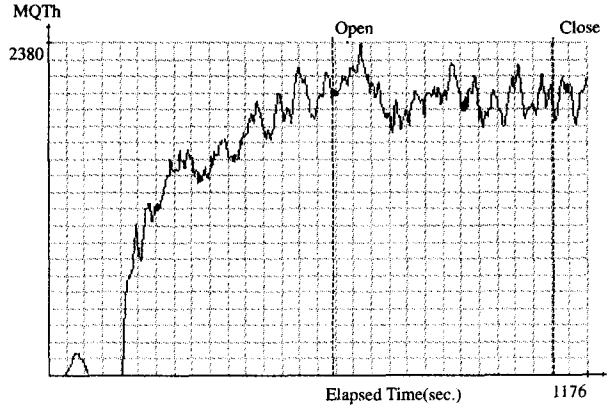


图 3 新订单事务统计图

最终测试结果如表1所示。

表1 加速引擎TPC-C测试结果

| 事务类型 | 90%响应时间(s) | 平均响应时间(s) | 最大响应时间(s) | 事务混合比 |
|---|---|---|---|---|
| 新订单 | 3.65 | 1.63 | 9.73 | 44.13% |
| 付款 | 2.81 | 0.99 | 10.24 | 43.63% |
| 发货 | 4.50 | 2.31 | 11.71 | 4.07% |
| 订单查询 | 3.58 | 1.37 | 8.21 | 4.12% |
| 库存查询 | 3.86 | 1.54 | 7.91 | 4.05% |
| tmpC = 2023.9 | | | | |

测试结果符合TPCC标准中事务90%响应时间(都小于5秒)和事务混合比的要求[7]。

相对于同等硬件条件下的单站点数据库服务器的TPC-C直接测试,单站点最多在27个仓库下吞吐量的tmpC值为295.6,加速引擎系统的加速比为2023.9/295.6 = 6.85,引擎中的集群站点效率达到6.85/8×100%= 86%。其性价比已超过国内部分名牌高档服务器的性能,整个测试环境的估计费用为(以下价格均折算为美元):

表2 整个测试环境的估计费用

| | 硬件 | 操作系统 | 数据库 |
|---|---|---|---|
| 服务器端 | 8*400$(400$/每站点) | RedHat Linux Fedora Core 2免费 | 达梦数据库个人版 For Linux 50$ |
| 客户端 | 500$ | Windows XP中文专业版 200$ | |
| 网络 ECOM 16口 10M/100M交换机 50$ | | | |
| 合计: 4000$ | | | |

加速引擎的性价比为:Price/tpmC= 4000 $/2023.9= 1.98$,而目前(截止2004年12月)国际上TPCC测试性价比居于前列的系统[8]为:

表3 TPC-C测试性价比排名

| 排名 | 硬件 | 操作系统 | 数据库 | tpmC | Price/tpmC |
|---|---|---|---|---|---|
| 1 | DELL PowerEdge 2650 | Windows 2003 Server | SQL Server 2000 Standard Ed | 22052 | 1.50$ |
| 2 | DELL PowerEdge 2850 | Windows 2003 Server | SQL Server 2000 Standard Ed | 26410 | 1.53$ |
| … | … | … | … | … | … |
| 10 | HP Proliant ML350T03 | Windows 2003 Standard Ed | SQL Server 2000 Standard Ed | 17192 | 1.96$ |
| 11 | DELL PowerEdge 2650 | Windows 2003 Server | SQL Server 2000 Standard Ed | 20108 | 2.06$ |

**结论及市场前景** 测试结果表明,数据库加速引擎已经达到较高的性价比指标,而且系统效率达到86%,但绝对tpmC值尚有差距。主要原因在限于硬件条件规模没有作更多节点的测试,但是按照上面测试结果以及TPC-C基准测试的特点,加速引擎中数据库集群划分的特征[9],加速引擎站点能够在保证性价比基本不变的前提下进行扩展,从而达到更高的tpmC值。另一方面,按照目前市场硬件的价格,在费用不变的情况下,节点硬件可以采用带有SATA磁盘阵列的更好的系统,使得系统性价比进一步提高。

因此加速引擎提供类似数据库SQL语言的接口,使用在数据库和应用系统之间,采用软件加速方法能够达到与特殊集成的昂贵硬件加速技术相当的功能,实现了同等硬件条件下联机事务处理性价比成倍提高,从而以较低的成本满足企业级应用需求,具有良好的通用性,移植性和可扩展性。项目产业化后,可以满足电信、金融等领域大规模OLTP处理的需要,市场前景非常广阔。

## 参考文献


1 蒋蜀,陈佩佩,谢立. 并行数据库的研究. 计算机研究与发展,1994,31(1)

2 刘晖,彭勤科,沈钧毅. 基于结点代理的数据库集群服务器. 小型微型计算机系统,2003,24(2):225~229

3 肖卫军,卢正鼎,李兵,李瑞轩. 一种多数据库事务模型. 小型微型计算机系统,2003,24(12)

4 吴志晴,卢正鼎,郭宜斌. MDBS中2LSR调度正确性的方法研究. 华中理工大学学报,2000,28(9)

5 庄成三. 基于事务语义的多数据库系统并发存取控制方法. 计算机学报,1996,19(5)

6 Leutenegger S T, Dias D M. A modeling study of the TPC-C benchmark. Computers and Structures, 1993,11(8): 22~31

7 Transaction Processing Performance Council. TPC BENCHMARK C Standard Specification Revision 5.3. April 2004. http://www.tpc.org/tpcc/spec/tpcc_current.pdf

8 Transaction Processing Performance Council. Top Ten TPC-C by Price/Performance Version 5 Results As of 16-Dec-2004. http://www.tpc.org/tpcc/results/tpcc_price_perf_results.asp

9 Piantedosi J A, Sathaye A S, Shakshober D J. Performance Measurement of TruCluster Systems under the TPC-C Benchmark. Digital Technical Journal, 1996,8(3):46~57

10 Oracle. An Oracle White Paper. Database Architecture: Federated vs. Clustered. February 2002. www.oracle.com/technology/tech/windows/rdbms/ClusterComp.pdf


(上接第74页)

物理环境和计算环境融合为一个智能环境,该智能环境支持人与信息的双向主动交互机制。基于该模型,开发了一个原型系统UbiPresn和面向老人的应用,验证了系统的有效性。基于该模型可以构建较为复杂的智能环境相关应用。

## 参考文献


1 Phan T, Zorpas G, Bagrodia R. An extensible and scalable content adaptation pipeline architecture to support heterogeneous clients. In: Proc. of the 22nd Intl. Conf. on Distributed Computing Systems (ICDCS 2002), July 2002. 507~516

2 Nitto E D, Sassarolil G, Zuccalà M. Adaptation of we contents and services to terminals capabilities: the @Terminals approach. In: Proc. of the 1st IEEE Intl. Conf. on Pervasive Computing and Communication (PerCom 2003), Fort Worth, USA, March 2003. 433~440

3 de Lara E, Wallach D S, Zwaenepoel W. Puppeteer: Component-based adaptation for mobile computing. In: Proc. of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS), San Francisco, California, March 2001

4 Device mosaic web browser. http://www.opentv.com/dm

5 Espial escape web browser. http://www.espial.com

6 Wang H, Zhou X, Zhang T. Information stream oriented content adaptation for pervasive computing. In: Proc. of the 2005 IEEE Intl. Conf. on E-Technology, E-Commerce, and E-Service (EEE'05), Hong Kong: IEEE Press, March-April 2005

7 Klyne G, Reynolds F, Woodrow C, Ohto H, Hjelm J, Butler M H, Tran L. Composite capability/preference profiles (CC/PP): Structure and vocabularies 1.0, W3C Recommendation, Jan. 2004

8 Wang H, Zhou X, Zhang T. An Active Information Space Infrastructure for Smart Homes. In: Proc. of the 3rd Intl. Conf. on Smart homes and health Telematic (ICOST2005) Canada, July 4, 2005