

# 带快照的混合数据库系统设计与应用<sup>\*</sup>

周 鹏<sup>1,2</sup> 杨 丹<sup>1</sup> 鱼详训<sup>3</sup>

(重庆大学计算机学院 重庆 400044)<sup>1</sup> (重庆邮电大学计算机科学与技术学院 重庆 400065)<sup>2</sup>

(韩国仁荷大学计算机科学与信息工程系 仁川韩国)<sup>3</sup>

**摘 要** 随着对海量数据处理的广泛需求,对数据库系统的性能也提出了更高的要求。本文介绍了一种带快照的混合数据库系统的设计。它在传统数据库系统基于磁盘存储的基础上,选择部分需被频繁或快速访问的数据以快照的形式组织在主存中,从而兼顾了处理海量数据和快速响应用户访问要求两方面的需求。最后,简要给出了一个在地理信息系统中的应用实例。

**关键词** 混合,数据库,快照

## Design and Application of a Hybrid Database System with Snapshots

ZHOU Peng<sup>1,2</sup> YANG Dan<sup>1</sup> EO Sang-Hun<sup>3</sup>

(College of Computer, Chongqing University, Chongqing 400044)<sup>1</sup>

(College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065)<sup>2</sup>

(College of Computer Science & Information Engineering, Inha University, Incheon, Korea)<sup>3</sup>

**Abstract** The performances of database systems are required more and more efficient with the demands of processing massive data. In this paper, design of a hybrid database system with snapshots is proposed. In proposed system, part data which need to be accessed fast or frequently by users are organized in main memory as snapshots besides the whole data stored in disk like disk-oriented systems, thus the proposed system can fulfill the demands which are the both fast response time and handling massive data. In the last, an application instance based on geographic information system is briefly given out.

**Keywords** Hybrid, Database, Snapshot

## 1 引言

随着信息化社会的发展,作为信息系统核心和基础的数据库技术得到了越来越广泛的应用。但是,随着计算机硬件、软件技术的飞速发展和数据库应用领域的进一步拓展,这些都对数据库系统提出了许多新的要求和挑战。而现有的传统数据库系统的局限集中体现在以下两个方面。首先,存储在传统数据库中的数据类型相对简单,也易于管理和处理。但是,随着应用领域的扩展,现在要求能在数据库中处理新数据类型的需求不断增加,如空间数据、时态数据和多媒体数据等等。这些数据往往都是海量数据,数据量相当巨大,对其进行有效管理的难度大大增加。这就对仅能管理传统简单数据类型的数据系统提出了新的挑战,形势越来越要求其能够有效管理复杂数据类型的海量数据。其次,随着高性能硬件的不断投入使用,越来越多的应用要求数据库系统提供更快响应时间,这就对数据库系统处理数据的速度和有效性提出了更高的要求。传统的基于磁盘的数据库管理系统,由于磁盘 I/O 仍然是个难以突破的瓶颈,因此其越来越不能适应新应用对其处理速度的要求。为了能更有效地处理数据,许多学者提出了很多有用的方法,如索引技术<sup>[1,2]</sup>,增大数据库缓存区<sup>[3]</sup>的方法等等。而随着主存技术的进步,使得将全部数据都放在主存中处理,从而加快数据库系统的处理速度也成为可能。但是,基于主存的数据库系统<sup>[4]</sup>虽然能够提供更快的处理速率,但是对于数据量庞大的复杂数据类型,将全部数

据都放在主存中将付出昂贵的代价,并不是一个理想的选择。

为了解决克服上述局限,提高数据库系统的性能以适应时代的发展。本文介绍了一种带快照的混合数据库系统。它采用分存储层次混合存储数据的方式,除了和基于磁盘的数据库系统一样将全部数据存储在磁盘上以外,又将部分被访问频率较高或者需要较快响应的数据以快照的形式组织在主存中。通过这种方式,一方面能够通过放在主存中的数据,加快响应查询的速度,显著改善数据库系统的性能;另一方面,因为只有部分数据被放在主存中,所以大大减轻了对主存容量的要求,使得我们能够以相对较小的代价显著改善性能。

本文随后部分组织如下,第 2 节介绍了带快照的混合数据库系统的基本结构及其主要设计。第 3 节简要介绍了一个基于地理信息系统的实际应用。最后是小结。

## 2 带快照的混合数据库系统

在本节,首先介绍了混合数据库系统的基本结构,然后简述了其设计的关键点即如何有效刷新快照。

### 2.1 系统基本结构

众所周知,视图是关系数据库管理系统提供给用户以多种角度观察数据库中数据的一种重要机制。视图实际上是从一个或几个基本表(或视图)导出的表,它与基本表的不同之处在于,基本表是实际存储在数据库系统的数据所组成的,而视图是一个虚表,数据库中只存放视图的定义,而不存放视图对应的数据,这些数据仍然存放在原来的基本表中。这样,一

<sup>\*</sup>基金项目:重庆市自然科学基金项目(CSTC,2006BB2234)。周 鹏 博士研究生,讲师,研究方向为数据库技术,图像处理。

且基本表中的数据发生变化,视图对应的数据也会随之改变。因此,从这个意义上讲,视图就像一个窗口,透过它可以看到数据库中不同用户分别感兴趣的数据及其变化。视图一经定义之后,用户就可以向对基本表一样对视图进行查询。但是,由视图的特点可知,视图最终是定义在基本表之上的,因此对视图的一切操作最终也必须要转换为对基本表的操作,这一转换的过程被称为视图消解。正因为在用视图进行查询时需要视图消解,所以视图在给用户提供众多方便的同时,用户并不能够通过使用视图而加快查询的速度。

快照是从视图的概念衍生而来的。与视图一样,快照也是从一个或几个基本表(或视图)导出的表,但与数据库通常只存储视图所对应的查询语句不同,一个快照可以理解为一个其内容已计算并存储了的视图,也就是说快照不是虚拟的,它不是通过在其他表上的定义来表示自身,而是通过它们各自的物化的数据备份<sup>[5]</sup>。定义一个快照就像是执行一个查询,但不同的是,查询的结果是以一个特定的名字保存在数据库中,它是一个只读的关系。另外,快照被定期更新,也就是说,它必须能够在原始数据变化时保持其内容同步更新,以免引起不一致。

从上述快照的特点可知,虽然快照包含冗余数据,因为其内容可以通过视图的定义和数据库中其他数据得到。然而,在许多情况下,直接读一个快照中的内容比通过执行相应视图的查询而得到该视图的内容要简单得多。这就使得数据库系统中的快照,在很多应用中对于提高系统性能显得非常有意义。这类应用的典型要求是数据被冻结在某一个适当的时刻,或是只需要与某个确切时间相近的数据就可以了。快照不但可以使这些数据被冻结而又不影响其他事务在这些数据上的更新操作,更重要的是如果被查询的数据已经包含在快照中,则系统可以通过直接使用快照完成查询,避免了视图消解的过程,这就显著地提高了响应查询的速度。快照存在的上述优点,正是设计带快照的混合数据库系统的出发点。

传统的数据库系统将全部数据都存放在外存中,当需要时再选中其中的一部分读到主存中来处理。这种传统组织数据的方式虽然能够存储和管理大量的数据,但已经越来越不能适应一些要求快速、实时处理数据的应用对其提出的要求。而如果把全部数据都存储在主存中,虽然能够显著地加快处理数据的速度,满足快速、实时的要求。但是,对于一些数据量要求特别大,甚至是要求处理海量数据的应用来说,又将付出巨大的代价。正因为如此,对于一些既要求存储海量数据,又只有一部分数据需被快速、实时处理的应用来说,作为折中的混合存储系统就显得非常有意义。

混合存储系统采用多种存储设备,综合考虑对数据的处理需求,以分层次的方式将全部数据存储在不同的设备中,以此来管理和组织数据<sup>[6]</sup>。在数据库系统中,我们也可以根据对数据处理的需要把数据组织成各种不同的结构放在不同层次的存储系统中。

带快照的混合数据库系统的设计主要涉及两个存储层次,即磁盘存储器、主存存储器。和传统的数据库系统一样,数据库中的全部数据都被存储在访问速度较慢、价格便宜的

磁盘上。但是本系统的特点在于,除将全部数据都放在磁盘上的同时,为了加快系统对用户查询的响应速度,将一些被访问频率较高或要求较快响应的数据,组织成快照的形式存储在访问速度较快的主存中。这样,当用户要查询的数据包含在主存的快照中时,系统就能通过读取快照迅速获得查询结果,显著地加快系统响应查询的速度。为了支持上述的功能,系统在设计时,在传统的数据库管理系统的基础上扩展了主存存储管理器和混合查询处理器这两个部件来实现对上述功能的支持。本系统中的主存存储管理器除了完成将数据从磁盘上取到主存中,并决定哪些数据应被缓存到主存中的常规功能外,还能支持按照系统的命令,复制被选中的一部分数据,然后将它们组织成一个个的快照存储在主存中,并在每个快照的全生命周期内负责对其进行管理,如定时刷新快照等。而混合查询处理器通过对传统数据库管理系统中的查询处理器的扩展来实现。它在本系统中扩展的主要功能是:分析用户的查询请求,以判断用户查询请求所涉及的数据是否已被包含在主存快照中。如果是,则只需调用主存存储管理器,从快照中读出数据;如果否,则需从磁盘中读出数据。该系统的一个示意图如图1所示。

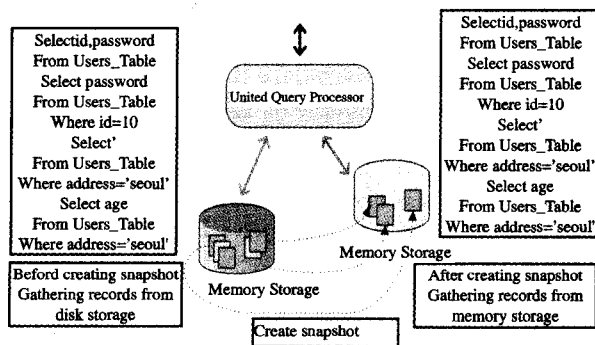


图1 带快照的混合数据库系统示意图

在带快照的混合数据库系统中,如何有效地刷新主存中的快照,从而保证数据的一致性系统设计的关键。在下一部分,将介绍如何在带快照的混合数据库系统中刷新视图。

## 2.2 快照的刷新

在系统中,管理员可以通过设置一个参数,根据同一个查询被执行次数,来决定系统是否把所涉及的数据自动从基表中抽取出来创建快照存储在主存中,在创建快照同时还可以设置其刷新周期。

在系统设计中,即允许用户通过常用的如投影、选择等简单关系操作创建基于单张基表的快照,也允许用户通过连接等复杂关系操作创建基于多张基表的快照。对于上述两种不同类型的快照,系统分别采用了不同的刷新方法。对于主存中基于多张基表的快照,由于其使用了涉及多张基表的复杂关系操作来创建,因此系统通过相对简单、定期重新创建的方式对其进行刷新。

对于只涉及单张基表的快照,系统采用了增量刷新的方法。在增量刷新方法中,主要包括记录日志、精简日志记录和发送刷新信息三个步骤,如图2所示。



图2 快照刷新过程示意图

第一步,系统使用了一种日志记录发生在快照刷新期间对基表的修改。数据库系统中的每张基表都对应了一张日志表,为了便于与用于数据库恢复的日志相区别,它被称为修改日志。修改日志的主要结构如图3所示。

Time Stamp	Modif. Type	Value
------------	-------------	-------

图3 修改日志的结构

修改日志记录主要包括以下三个字段。

(1)时间戳(TimeStamp):用于记录每个基表修改操作发生的时刻。当快照刷新时刻到来时,通过扫描记录的时间戳,就能找出发生在上一次刷新之后的修改操作,根据这些记录就可以刷新快照。

(2)修改类型(Modif. Type):用于记录修改操作的类型,即更新(UPD)、插入(INS)或删除(DEL)操作。通过记录对基表修改操作的类型,就能确定对快照刷新时相应的操作。

(3)元组值(Value):对于插入和删除操作,修改日志只记录对基表修改后的值;对于更新操作,需要同时记录修改前后的值。

随后,根据在创建快照时设置的刷新周期,系统在对应时刻对快照进行刷新。系统首先扫描快照涉及基表的修改日志记录,找出与快照相关的记录。为了减少发送给快照的无用刷新信息,系统设计了一个对修改日志记录进行精简的预处理过程。它将对基表中同一个元组的多次修改进行归并,即只按照元组的最后修改结果作为刷新的依据,这样就能减少对快照内容的无用刷新。

最后,根据精简后的日志记录,将相应的刷新信息发送给存储在主存中的快照,快照就能够被系统快速刷新,以保证混合数据库中数据的一致性。

通常,在一个快照的刷新周期,其对应基表中被修改的相应元组与整个基表中的数据比较起来是很少的。因此,使用带时间戳技术的修改日志对快照进行增量刷新与完全重建快照的刷新方法比较起来,虽然增加了系统的复杂性,但是仍然是非常有效的。

### 3 应用实例

实际上,本系统最理想的应用环境是在有海量数据需要管理,但其中有一部分数据需要被冻结在某一个适当的时刻,或是只需要与某个确切时间相近的数据,即无需数据随时保持一致性的场合。下面以地理信息系统(GIS:Geographic Information System)为例进行说明。在地理信息系统中,一个典型应用是能够向用户提供电子地图查询服务。想用户通过这种服务,空间数据库系统必须负责管理大量的包含空间信

息的电子地图。由实际情况可知,电子地图中的数据通常是定期维护,并不需要被随时刷新。因此,如果某个明确的区域被用户频繁地查询,那么系统就可以通过把该区域涉及的数据从电子地图中抽取出来放在主存中创建快照。之后,当用户查询再次涉及该区域时,系统就能迅速地通过快照返回结果,而无需再到数据库中去查找相应的数据。在图4中,给出了从电子地图中抽取数据创建快照的例子。



图4 在电子地图中创建快照

除了在地理信息系统中的应用外,本系统也易于在现有的、广泛使用的基于磁盘的数据库系统的基础上进行扩展,从而使现有的数据库能满足更多实际应用对数据库系统性能提出的要求。

**结论及后继研究** 本文介绍了带快照的混合数据库系统的设计与应用。介绍了它的基本结构,并通过一个基于地理信息系统的实例介绍它的应用。

带快照的混合数据库系统还需要进一步的发展和完善,其后继研究工作将主要集中在以下两个方面:

(1)如何通过有效的策略选择参数,从而在基本表中选择合适的数据组成快照。

(2)如何更好地管理主存中快照,使其能够及时得到更新和删除。

### 参考文献

- 1 Bayer R, McCreight E M. Organization and maintenance of large ordered indices. *Acta Informatica*, 1972, 9(3)
- 2 Knuth D E. *The art of computer programming*. Addison Wesley, Sorting and Searching, 1973, 3
- 3 Chou H T, Dewitt D J. An evaluation of buffer management strategies for relational database systems. In: *Proceedings of 11th conference on Very Large Data Bases*, September 1985
- 4 Garcia-Molina H, Salem K. Main memory database systems: an overview. *IEEE Transactions on Knowledge and Data Engineering*, 1992, 4(6)
- 5 Gupta, Mumick I S. Maintenance of materialized views: problems, techniques and applications. *IEEE Data Engineering, Bulletin*, 1995, 18(2)
- 6 Stonebraker M. Managing persistent objects in a multi-level store. In: *Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data*, 1991, 20(2)

(上接第240页)

### 参考文献

- 1 侯志强,韩崇昭. 视觉跟踪技术综述. *自动化学报*, 2006, 32(4): 603~614
- 2 刘钢,刘明,匡海鹏,修吉宏,翟林培. 多目标跟踪方法综述. *光电与控制*, 2004, 11(3): 26~29
- 3 王亮,胡卫明,谭铁牛. 人运动的视觉分析综述. *计算机学报*, 2002, 25(3): 225~237
- 4 Yilmaz A, Javed O, Shah M. Object Tracking: A Survey. *ACM Computing surveys*, 2006, 38(4): 1~45
- 5 Veenman C, Reinders M, Backer E. Resolving motion correspondence for densely moving points. *IEEE Transaction Pattern Anal-*

ysis and Machine Intelligence, 2001, 23(1): 54~72

- 6 Serby D, Koller Meier S, Gool L V. Probabilistic object tracking using multiple features. In: *IEEE International Conference of Pattern Recognition(ICPR)*, 2004, 184~187
- 7 Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 2003, 25: 564~575
- 8 Yilmaz A, Li X, Shah M. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 2004, 26(11): 1531~1536
- 9 Otsu N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979, 9(1): 62~66