

银行业数据仓库的性能优化方法

韩守忠, 郑 诚

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039)

摘 要:首先说明了数据仓库的特点和结构及性能优化技术;然后结合银行业数据仓库的特点,提出了在银行业数据仓库系统开发中提高系统性能的几种设计方法。实践证明,在系统开发过程中,合理运用这些方法,可大大提高数据仓库系统的性能。

关键词:数据仓库;性能优化;宽表;并行;聚集

中图分类号:TP311.138

文献标识码:A

文章编号:1005-3751(2006)04-0196-03

Performance Optimizing Methods of Data Warehouse about Banking

HAN Shou-zhong, ZHENG Cheng

(Ministry of Edu. Key Lab. of Intelligent Computing & Signal Processing, Anhui University, Hefei 230039, China)

Abstract: At first introduces the characteristic, structure of data warehouse and some general knowledge of performance optimizing, then combining the characteristic of the data warehouse of the banking, proposes some design methods to improve systematic function in banking's data warehouse system. Experimental results show that the performance of data warehouse can be improved by using properly these methods.

Key words: data warehouse; performance optimizing; wide table; parallel; mass

0 前 言

数据仓库是用来为决策服务的,具有面向主题、集成、相对稳定、随时间不断变化等特性,它一般涉及海量数据的查询^[1]。数据的大量写入读出,对数据库系统的要求很高,因此在系统设计、实施和维护的过程中,系统的性能是一个不可忽视的问题。为了提高性能,在运行期间,要密切关注应用对系统资源的消耗情况,针对应用的特点及时对系统进行调整,包括调整数据库参数、数据分片放置、创建特殊索引乃至提高系统配置等常规优化方法的研究在改善数据仓库性能方面目前已经取得了一定的效果。但是在设计和规划阶段就设计一个合适的模型和整体框架是一个极其关键的做法,也是从根本上提高数据仓库性能的决定因素^[2]。而银行业有其自身独特的特点,如规模庞大、分支机构众多、内部管理复杂、业务繁杂,所以对于银行业数据仓库的建设及优化提出了更高的要求。

数据仓库应用系统的结构可分为源数据、数据模型、数据展现3部分,如图1所示。

从图1可知,在数据仓库系统中,有3个关键环节:数据处理、数据模型设计、数据展现。数据处理是从数据源

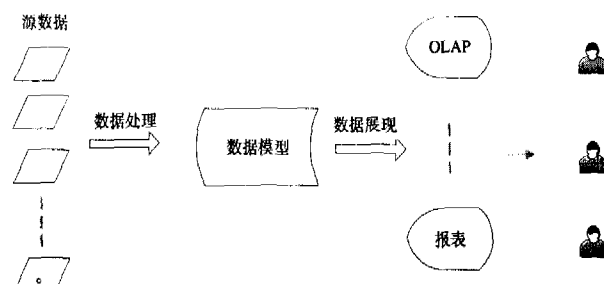


图1 数据仓库应用系统结构

中抽取数据,对其进行检验和整理并根据系统的设计要求,对数据进行组织加工,装载到数据仓库的目标数据库中,并周期性刷新以反映数据源的变化。数据仓库是一个多维数据库,其目标数据库的数据模型是整个数据仓库环境的核心,其内部存放着数据建模的数据和元数据。数据展现端提供访问数据仓库中数据的功能,所以要提高数据仓库系统的性能,主要从3个方面入手:数据处理、数据模型、数据展现。文中根据此3方面并结合笔者参与开发的一个银行数据仓库系统:综合统计系统4.0,提出了数据仓库开发过程中的3种优化方法:并行数据处理、宽表数据存储、数据聚集。

1 并行数据处理

数据仓库系统的数据源大都以几个交易系统为基础,结合了多个应用系统和外部的数据源,如在银行业中,就可能涉及到电话银行、手机银行、自动取款、银行卡等多种

收稿日期:2005-07-27

作者简介:韩守忠(1981-),男,安徽阜阳人,硕士研究生,研究方向为数据库及数据仓库、数据挖掘;郑 诚,副教授,博士,研究方向为网络下的数据库及数据挖掘、机器学习、人工智能。

业务,包括了总帐、分户帐、帐户明细等各个层面的数据^[3],数据量非常大。为了提高数据处理效率,充分利用系统资源,在数据处理的设计方案中可广泛采用并行数据处理。系统数据处理流程中采用的并行处理方式主要包括以下两种:

(1) 分析数据处理流程,没有依赖关系的数据处理可以并行执行。

数据仓库系统的数据模型是一个有机的整体,但在数据处理过程中并不是完全互相依赖、密不可分的,对于没有前驱后继关系的数据处理模块可以并行执行。通过分析与设计,理清数据处理流程中的数据处理模块及它们之间的依赖关系,发现数据处理结构的特点,然后不同分支上的数据处理模块可并行执行。

(2) 拆分源文件,使同一数据源中不同特点的数据可以并行处理。

在同一数据源文件中,经常存在着不同特点的数据,需要按照不同的规则进行匹配和处理。当面对大量数据时,这种处理往往非常耗时,甚至不能在用户接受的时间内完成。在此情况下,使用外部程序对文件拆分,将不同特点的数据拆分到不同的文件中,这样后续的数据处理面对的是多个没有依赖关系的数据源,这些数据处理也是可以并行执行的。如银行分户帐余额积数文件,可将其拆分为贷款户、定期户、往来户、表外户、财务户和内部户,每种帐户的数据处理规则单一,且面对的源数据量大大减小,从而缩短了整个分户帐数据处理的时间。数据并行处理结构如图2所示。

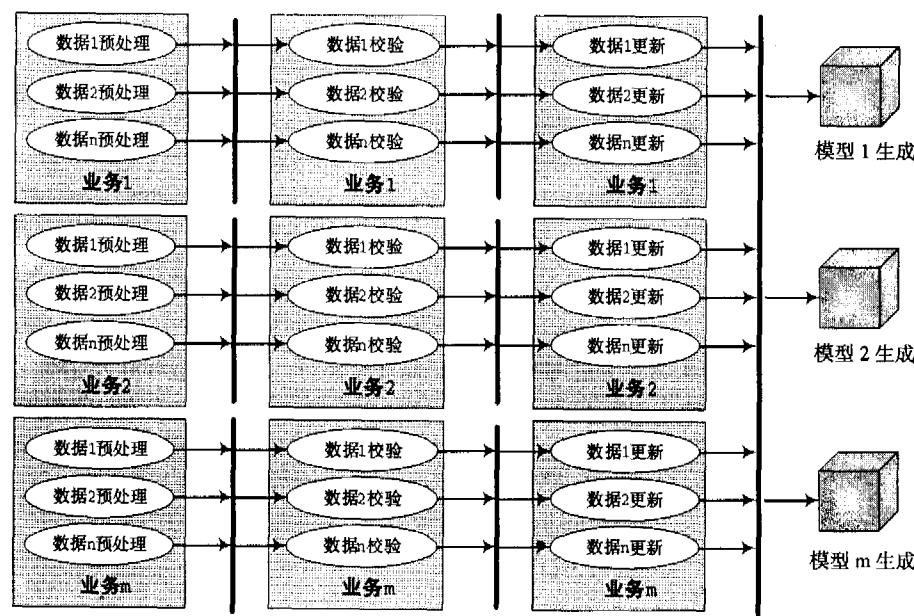


图2 数据并行处理结构图

同时,在实现中系统是否能达到并行优化的目的,以下要求极为关键:高物理内存的 SMP(对称多处理系统),MPP(大规模并行处理器)和簇(Cluster)计算机;有高 I/O 的主机;有足够的 CPU 和内存^[4]。

以上条件,在银行计算机系统配置中都能得到充分的

满足。通过对综合统计系统 4.0 和以前的 3.0 版本的批量数据处理测试发现,同样配置条件下、同样数据量,采用并行处理的 4.0 版本比 3.0 整体速度提高了 60%,达到了以下设计目标:处理速度得到提升;依赖关系减少;处理顺序也可得到控制。

2 宽表数据存储

在总结银行分户帐业务数据特点的基础上,同时根据前台数据分析查看的需要,采用了宽表(列比较多,结构上较宽的表)这种数据存储方式。实现中,模型采用宽表形式的事实表组织业务指标数据,来达到存储空间小、访问速度快的目的。其实质是根据银行业数据的特点,在数据仓库基本模型(星型模型)设计的基础上做出的一种改进。

以银行余额指标为例。一般地,银行分户帐数据的存储方式,是每日每帐户记录余额,即使帐户某日的余额没有发生变动,也需要复制前日余额作为当日余额。但事实是,帐户余额每日变动的频率非常小,大部分帐户的变动集中在几次或一次。此变化频率可以使用宽表来记录帐户某个指标的变化。将每日余额值记录到列上,即一天的余额值用一列记录,如下面的贷款帐户指标表(ACCT-TARGET-DK)所示,它把时间维拆分为年和月、日,分别放在行和列上,而不是传统地把时间(年,月,日)作为一个维(数据要求到日粒度的),利用此方法可达到性能优化的目的。

ACCT-TARGET-DK (ACCT-SEQ, YEAR, TARGET-TYPE, COL0101, COL0102, COL0103, ..., COL1230, COL1231)

其中 ACCT-SEQ——帐户序列号,外键,贷款帐户维表的主键

YEAR——年份

TARGET-TYPE——指标类型

COLXXYY——XX 月 YY 号某帐户某个类型的指标值

传统形式的表到宽表的结构变化如图3所示。

下面是一组通过对贷款帐户指标数据(1680000 条记录)测试的具体结论比较值:

* 传统结构:需要空间大小

500M/日 × 365 日 =

182500M/年

* 宽表方式结构:需要空间大小

3372M/年

比较:后者的存储方式大小是前者的 1.8%。

通过大量数据的测试发现,通过宽表储存数据在执行

过程中具有以下优点:

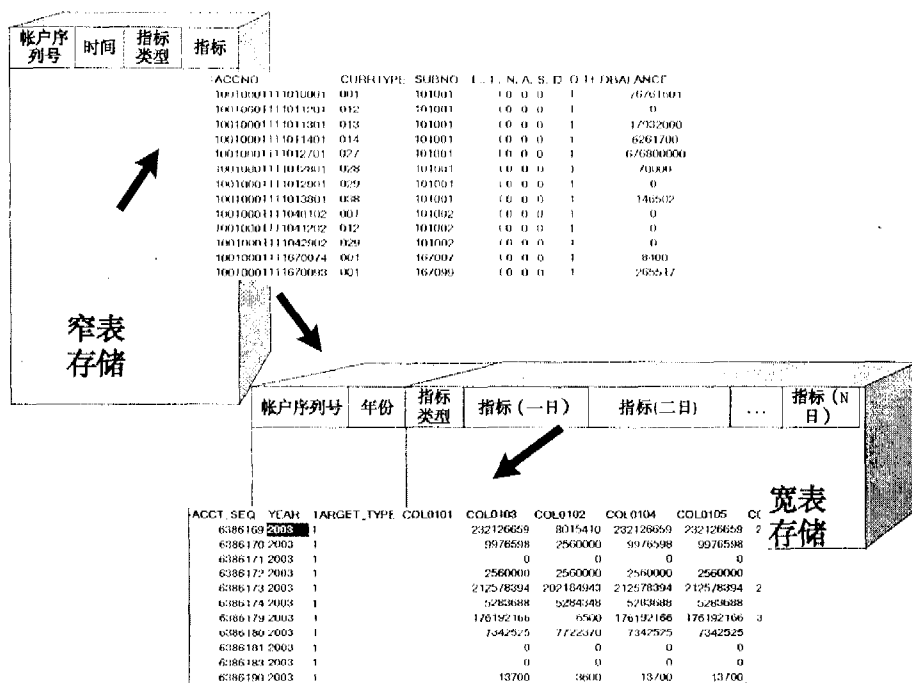


图 3 表结构变化图

(1)最小的数据存储。

采用宽表存储,避免了帐户指标每日一条余额记录,从而使数据记录数大大减少。在综合统计 4.0 开发的测试环境中,综合统计 3.0 中占用 186G 空间的数据,在 4.0 版本中采用了宽表存储数据,只占用了 35G 的空间,整个数据表的存储空间比以前的空间减少了 80% 以上。

(2)最快的查询性能。

数据量大大减少后,定位某帐户指标值的时间显然加快,同时,对该帐户的余额进行列运算的速度相对其它存储方式而言,也能达到较快寻址。要查询满足一定条件的数据,在一个记录数较少的表中查找和在记录数很大的表中查找相比,所需时间显然不同。

(3)较好的处理速度。

虽然将每日余额值更新到列上的时间是较长的,但由于记录数显然减少的原因,其在数据源生成宽表数据模型的处理时间上也占有一定优势,同时在前台数据展现客户端能得到很好的体现。

宽表存取数据的性能测试结果总结如表 1 所示。

表 1 宽表存取数据的性能测试结果

存储结构	占用空间	处理效率	查询性能
时间片方式	较小	最慢	较快
时间宽表方式	最小	较快	最快
账户索引方式	最大	最快	最慢

3 数据物化

数据仓库包含了海量数据,而进行 OLAP 或数据挖掘等应用查询,要求系统的数据展现客户端在较短时间内(如几秒内)给予响应,同时基于数据仓库的决策支持系统应用一个显著特点是:用户一般关注的是综合统计数据,

所以,在系统开发过程中对一些数据预先进行物化,避免在查询时实时计算,可大大提高系统的响应性能。物化通过某个(些)维度对某个(些)指标进行汇总,使数据达到一个用户感兴趣的高粒度层次^[5]。

从用户和应用的角度来看,数据仓库中的数据是否需要物化有二个决定因素:一是看数据的使用率,各种操作(包括查询、上卷、下钻)经常使用到的数据,需要将其物化,这样可以显著提高整个数据仓库的响应速度和用户满意度;二是看数据的重要性和用户的需要,如果可以为一些重要的用户或关键的应用提供较好的响应速度,即使该物化方案的使用率较低,也需要将该数据物化^[6]。

经分析,在银行业数据仓库系统中,适于进行物化的主题有:帐户模型、客户模型、帐户机构模型、总帐模型、个人储蓄模型、客户人行模型、客户信贷模型。通过在项目测试中比较,发现经过物化后取数据,反应速度大大提高。如要查贷款帐户(1680000 条记录,日粒度数据)中帐户 0890965423546700008765 在 2005 年的贷款额,系统反应时间为 30 秒,聚集后从物化表(780000 条记录,年粒度数据)中取数只需要 4 秒。

4 结束语

并行处理是在数据仓库构建过程中,为了提高数据处理效率依据各个主题数据处理关系确定的一种数据处理方式,但在系统资源相对有限的情况下,并行执行的程序过多,并不能达到这一目标,甚至可能适得其反。因此,对并行度的把握极其关键,要考虑现有系统的配置,在实践中反复测试并根据测试结果进行调整。宽表存储是在一定条件下采用的一种数据存储方法,否则并不能达到优化的目的。对于聚集物化方案,需要分析所有的用户和应用的需求,研究实际使用中需要哪些维度粒度层次的汇总信息,从而确定所有可能涉及的聚集和估算使用的频度。

虽然这些优化方案是针对某些情况下的一种设计,但在其他领域或系统中如果具有相近的处理机制和数据变化情况,都可以考虑采用这几种方式对他们进行处理,从而达到优化系统性能的目的。

参考文献:

- [1] Immon W H. 数据仓库[M]. 王志海等译. 北京:机械工业出版社. (下转第 202 页)

须和远程授权引擎共享这个密钥。USM 允许各个用户只记住自己的密码,根据用户密码设置用户密钥。各个授权引擎再将用户密钥和本引擎的 ID 结合起来形成用户本地密钥,以共享用户密钥并实现用户密钥本地化。采取这种方式,可以减缓字典式密钥攻击的速度;可以使用户密钥独立于网管系统而只和用户密码有关。同时,由于不同用户的密钥各不相同,并且同一个用户在不同代理上的密钥也不相同,因此一个用户在某一个代理上的密钥受到损害不会影响别的代理。使用的密钥可以通过 SNMP 进行远程设置。管理台和代理之间就通过此密钥对报文的 PDU 进行加解密,使通信可以安全可靠地进行。

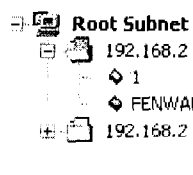
4 VACM(基于视图的访问控制模型)

VACM 解决的主要问题是合法实体是否有权限去操作它在 PDU 中所要求的 MIB 对象,并将用户和特定的 MIB 视图关联起来。此外,它还可以为特定的安全模型和安全级别定义不同的 MIB 视图。在具体实现权限管理时,引入了组(group)的概念,通过设置它的属性来设置它所规定的权限^[7]。一个用户若属于一个组,那么他就拥有了这个组所规定的权限。组中应包括以下属性:安全模型、安全级别、上下文名(可选)以及读/写/通知视图名。利用安全模型和安全名作为索引找到一个记录,形成一个组名。

VACM 通过 3 张表:安全组表、视图子树表、访问控制表来完成检测机制^[7]。安全组表将消息中的安全模型和安全名映射为一个组名,而组名作为访问控制表中索引的一部分。访问控制表将一个组名、上下文以及安全信息映射为一个 MIB 视图。视图子树表标识一个 MIB 视图的特定信息^[1]。由于视图子树表功能独立,所以将其独立保留为一张表。安全组表功能较简单,因此将它合并到访问控制表中。

当协议实体接收到各种安全管理请求时,一个命令响应程序将为请求中绑定的每个 MIB 对象调用 VACM 系统来决定是否允许访问。首先在安全组表中,通过安全模型和安全名作为索引找到一个记录,形成一个组名。然后,利用上下文、组名、安全模型、安全等级和视图类型作为索引,找到一个记录,映射成一个 MIB 视图。最后利用视图子树表,判断所要求的 OID 是否属于该 MIB 视图。

如果属于,则授权该用户进行操作。综上所述,由于每个用户拥有的安全模型、安全级别组合相对较少,这样做不会大量增加组的数目,而且简化了 VACM 机制。下图为通过 SNMPc 查看到交换机配置的组。



Mo	Name	vacmGroupNam	ToGroupStorageType	ToGroupS
1	public	public	volatile	active
2	private	private	volatile	active
3	1111	aaaa	nonVolatile	active
3	2222	bbbb	nonVolatile	active

图 4 组配置图

5 结束语

SNMP 作为一种简单高效的网络管理框架,越来越受到用户的青睐。SNMPv3 中的各个模块的功能不断完善,安全管理也有了极大的提高。文中详细介绍了 SNMPv3 的框架结构,说明了其中各个模块的功能和新的消息格式。SNMPv3 虽然解决了 SNMP 中最重要的安全问题,但它也不是尽善尽美的。比如,从理论上来说 SNMPv3 是无法利用 msgID 来保护消息不被重复的。它没有解决 SNMP 的管理信息库问题。SNMP 的管理信息库结构复杂,庞大而且冗余,它包含的标准包括 MIB, MIB2, RMON 和 RMON2 等,还有各个企业私有的 MIB,其中不少信息都是重复或相似的。

参考文献:

- [1] 翟 纲,但海涛,诸昌铃.基于 SNMPv3 的安全网管的研究[J].通信技术,2003(1):106-108.
- [2] 王 华,王宗宁,高传善.SNMPv3:完善 SNMP 的安全机制[J].计算机工程,1997,23:350-351.
- [3] 王荣华,刘世栋,杨 林.SNMPv3 在网络安全管理系统中的应用[J].网络安全技术与应用,2004(4):22-24.
- [4] 乐 毅,肖德宝.基于 SNMPv3 的策略网管的设计与实现[J].通讯和计算机,2005(2):62-65.
- [5] 金 鹏,郝 平.SNMPv3 中的安全机制[J].通信技术,2002(4):77-79.
- [6] 刘 燕.基于 SNMPv3 网络管理系统的研究与设计[D].武汉:武汉大学,2000.
- [7] Zeltserman D. SNMPv3 与网络管理[M]. 潇湘工作室译.北京:人民邮电出版社,2000.
- [8] 何 炜,陈 思.SNMPv3 网络管理中的安全机制[J].现代电信科技,2003(11):28-30.

(上接第 198 页)

出版社,2005.

- [2] Ponniah P. Data Warehousing Fundamentals[M]. 段云峰,等译.北京:电子工业出版社,2003.
- [3] 赵先信.银行内部模型和监管模型[M].上海:上海人民出版社,2003.

- [4] 钱雪忠.典型数据并发访问问题的探讨[J].微机发展,2003,13(6):64-66.
- [5] Burleson D K. Oracle High-Performance SQL Tuning[M]. 刘 砚,等译.北京:机械工业出版社,2002.
- [6] 郑谦益. Sybase Sql Server 性能优化技术及应用研究[J].微机发展,2003,13(1):85-86.