实时数据仓库的一种实现方法①

龙晓泉　戴牡红　李 鹏　李河清 (湖南大学 软件学院 湖南 长沙 410082)

摘　要： 为了改善传统的数据仓库只能分析历史数据，数据抽取周期过长以及无法满足实时分析的要求等缺点，提出了一种实时数据仓库的装载方法，该方法将采取复制表结构、设置查询限制等条件实现在数据仓库中数据持续高效的装载，并且阐述了如何使用数据仓库架构以及客户终端 OLTP 查询来实现实时数据的整合。

关键词： 实时；数据仓库；实现方法；OLTP；ETL

Implementaction of Real-Time Data Warehouse

LONG Xiao-Quan, DAI Mu-Hong, LI Peng, LI He-Qing

(School of Software, Hunan University, Changsha 410082, China)

Abstract: Traditional warehouse systems cannot do real-time analysis due to its relatively long data extraction period. In order to improve it, this paper puts forward a real-time data warehouse implementation method. This method adopts copying table structure and setting limitation on querying to loading data in data warehouse consistently and effectively. It also demonstrates how to integrate the real-time data by using the data warehouse architecture and OLTP queries in client.

Keywords: real-time; data warehouse; implement method; OLTP; ETL

1　引言

在当今信息密集的环境下，对于数据仓库的需求日益增长。传统数据仓库的数据更新一般是每天，每周或是每个月更新一次[1]，这就意味着它的数据不是最新的，从 OLTP 系统更新的记录保存是不包括数据区的，这说明最近的业务记录没有纳入数据区，然而，对于电子商业，股票经济，在线通讯以及决策系统等信息，需要及时发送给依赖它们的知识工作者或决策者，他们依据最新的这些数据信息去占领市场，由此，提出了实时数据仓库的要求。

2　实时数据仓库概述

2.1 实时数据仓库的概念

实时数据仓库(Real-time Data Warehouse, RTDW)[2]是两种事物的组合：实时行为和数据仓库。实时行为是一种即时发生的行为。行为可以是任何事情，如超市中小商品的销售行为。一旦行为完成，就有关于它的数据。数据仓库捕获有关商业行为的数据，而实时数据仓库在商业行为发生时就捕获数据。当商业行为完成时，相关数据就已经进入到数据仓库并且能立即使用。换句话说，实时数据仓库是这样一个系统，只要行为发生、数据变得可用时，用户就可以实时获取信息，从而做出战术型决策。

2.2 实时数据仓库研究现状

实时数据仓库面临的第一个挑战就是数据抽取、转换、清洗、加载进数据仓库的过程[3]。几乎所有的 ETL 工具和系统，不管是由厂商提供的还是用户单独编程实现的，都是基于批处理的工作模式.源数据通常按每天、每周或每月这种固定的周期加载进数据仓库。而且在数据加载的过程中，数据仓库处于停工的状态，用户不允许访问数据仓库。一般这种 ETL 过程是在夜晚进行的，所以对传统数据仓库的用户没有什么影响，但是实时数据仓库就不允许数据仓库处于这种停工的状态[4]。

① 收稿时间:2009-09-16;收到修改稿时间:2009-10-21

基于 ETL 实时数据仓库数据加载方式是批处理的过程，是通过不断缩短批处理的周期，尽量接近实时。这是一种准实时数据仓库的实现模式。若用户对实时性的要求并不高，例如可以接受按一天或几小时的实时性，这是一种很好解决方案，因为这种方案基于传统数据仓库，是对传统数据仓库的改进，不会改变原有投资;但如果用户实时性要求比较高，采用这种方式，就要不断增加硬件投入，通过提高系统的执行性能解决实时性的问题，又会增加用户的投资，因此这是一种准实时的方式，而不是真正意义上的实时数据仓库。

2.3 实时数据仓库的基本要求

数据仓库为分析过程，制作决策和数据挖掘工具提供信息，数据仓库收集的数据来自多样化的操作源系统 OLTP(On-Line Transaction Processing)以及中间数据存储组合的综合商业数据，通过使用 OLAP (On-Line Analytical Processing)了解不同用户的需求[5]。企业通常在数据仓库的数据区中存储完整的历史记录，而获取决策信息的一般过程是使用 OLAP 工具，这些工具有自己的数据资源,通过 ETL(Extraction, Transformation and Loading)工具对这些记录更新，从 OLAP 系统中负责确定和提取相关数据，然后定制数据，并把它们集成到一个相同的格式中，清洗数据，并使格式保持一致，以便更新数据仓库中的数据区，最后装载所有格式化了的数据进数据库中。

目前，高尔夫运动在我国的越来越广泛的普及，作为一家为高尔夫运动者提供打球预定服务的公司，其所面对的数量急剧的增加，为给领导决策和业务人员技术分析提供多角度、极具参考价值的第一手辅助信息，湖南鹰皇公司设计开发了一个高尔夫决策支持系统，从异构数据源中抽取，转换和装载(ETL)数据的操作，预处理加工后构建一个涵盖预订、产品、话务、结算等日常业务的数据仓库，然后基于相关业务分析指标和多维度视角对数据仓库进行面向主题的数据挖掘和深度分析。

现在，湖南鹰皇高尔夫决策支持系统每天以达到兆或千兆字节的速度产生大量数据，数据区通常要接受大量的记录以及完成加入，排序，分组和计算功能等措施，为了优化这些过程，数据仓库使用了规模较大以及复杂程度较高的内部数据结构(如索引，分区等)[6]。鉴于更新的频率和容量，实时交易数据的扩展最可能使服务器超负荷。以及伴随数据仓库的数据区域复杂的操作，大大降低了 OLAP 系统的性能。

总之，实时数据仓库的目的在于减少做决策的时间，并努力实现零延时，弥补智能反应系统和系统过程中的差距。我们的目标是把使用批量装载的数据仓库系统(禁止分析访问)改为零延迟能够提供最新数据的分析环境[7]，从而使得新信息在整个机构中进行实时传送。要想超越传统数据仓库的标准，关键问题在于如何确保持续数据的集成，并降低系统的负效应，如 OLTP 系统和 OLAP 系统的可用性和响应时间。

采用实时数据仓库，必须解决以下两种激进的数据状态变化。(1)必须保持连续的数据更新，由于不断的数据集成，将影响到行插入。(2)数据更新必须与 OLAP 的运行并行[8]。因此，本文的主要特征表现在三个方面：

(1) 快速有效的把 OLTP 中的数据转入数据仓库中，从而保持最新的数据。

(2) 在进行持续数据集成过程中，使 OLAP 响应时间降到最低。

(3) 当用户和 OLAP 应用处于脱机状态时，通过减少更新时间窗口，使数据仓库的可用性达到最大。

2.4 实时数据仓库所面临的问题以及解决方案

对于一个实时数据仓库(RTDW,所谓的“零延迟数据仓库环境”[9]的一部分)来说，数据的获取和集成是至关重要的，在 OLTP 和 OLAP 系统之间，如何确保 OLTP 中的数据持续进入 OLAP 系统中，为了实时可行，需要考虑以下问题：

(1) 运行的 OLTP 系统必须满足较短时间响应需求，为了实现系统最优化的可行性，RTDW 需满足并解决这一问题。

(2) 与交易记录直接相关的表存在数据仓库中，对于数目比较庞大的表，新数据的增加，以及由此产生的操作，如索引更新，参照完整性约束等，都会影响 OLAP 系统的运行和数据的可用性。

因此，工作的重心在数据仓库方面，为持续数据集成，完成 ETL 装载过程提供一个有效的方法。

本文将展现一种解决方法，这种方法能进行高效持续的数据集成，与此同时允许 OLAP 运行，且把性能消耗降到最低，此外，要把交易信息的记录以及其决策支持数据库的更新做到延迟最小化。这就涉及到系统的数据仓库终端，即如何高效执行 ETL 装载过程

和数据仓库的数据区中连续数据的集成。基于数据仓库的现有模式，具体方案是为每个表创立一个复制表，这个表中没有任何内容，索引，主键以及其它任何约束条件，在这个新模式中，复制的表将不断接收和记录来自中转区中的数据，并被不断装载。而且在装载过程中所消耗的时间和资源降到最低，这些资源是数据集成过程中必不可缺的一部分，当数据集成过程中，由于复制表中缺乏通常可优化的数据结构(如索引),导致含有最新数据的数据库性能降低。当用户或是管理员对数据仓库的性能不被接受时，在复制表中现有数据可以用来更新原有模式，更新完后，复制的表将再次创建为空内容的表，重新获得高性能。下面将展示如何在新模式中适应 OLAP 查询，充分利用实时集成进来的最新数据。

3　实时数据仓库的实现方法

在数据仓库方面，更新庞大的表和相关结构(如索引，物化视图，以及其它综合组件)，以及持续数据的集成使执行 OLAP 查询工作变成一项非常艰巨的任务。持续数据仓库装载方法总体结构如图 1 所示：

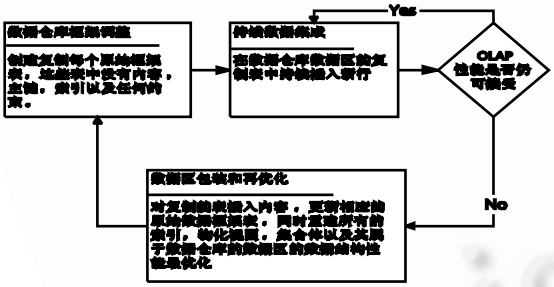

图 1　持续数据仓库装载方法总体架构图

下面的方案将会使更新过程中所需的处理时间和工作量最小化，这有助于数据仓库的脱机更新。因为，数据已在数据区和持续的数据集成中，所有的 OLTP 数据抽取和转换都在执行。此外，复制表的数据结构与原始的数据仓库框架中的一样，由于这种更新可以一步到位，通过从临时表中的剪贴行为来恢复原表，这就减少了数据仓库中数据区的装入时间，使用的方法主要集中以下几个个领域：(1)数据仓库架构的调整；(2)ETL 装载过程；(3)OLAP 查询的调整。这些主要基于一个非常简单的原则：在很少或没有内容的表中进行新行插入远比在内容多的表中进行新行插入的速度快得多。这正是 OLTP 数据源保持尽可能少的必要记录的主要原因。

3.1 数据仓库架构的调整

在湖南鹰皇高尔夫决策支持系统中会员预订的数据仓库架构，含有二维表(会员表 Member 和预订表 Book，代表业务描述实体)及一个实体表(会员卡表 MemberCard，储存所有业务交易的集合)，为了简化图表，数据没有显示，数据仓库可以存储每个会员，每张会员卡，每天预订的情况。具体关系如图 2 所示：

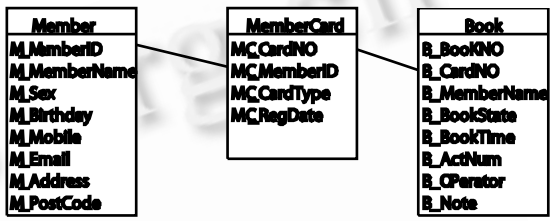

图 2　会员预订数据仓库架构图

对于数据仓库框架的调整，将采用下列方法：

创建一个数据仓库中所有表的结构性副本，该副本能够接收新数据。这些临时表在创建时不赋予任何内容，没有定义任何索引、主键、以及任何类型的约束。对于每个临时表而言，必须创建额外的属性用来存储新插入行数据的唯一序列标识。

基于这种方法，用于支持实时数据仓库的修改列表如图 3 所示：

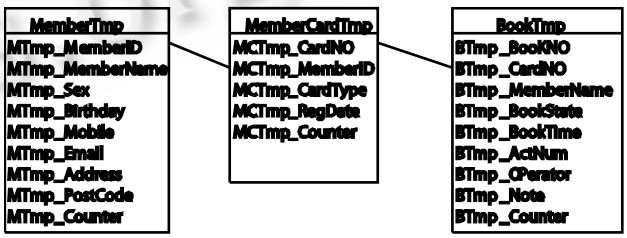

图 3　支持实时数据仓库修改的会员预订架构图

在每个临时表中的唯一序列标识符属性，它记录的是在数据区中每个新增行的序列。这将为每个新增行标记准确的序列。对于在重新恢复过程中存储先前数据以及对于放弃那些已经被更新过的数据是十分有用的。例如,当相同的用户在 OLTP 系统中做了两次更新，导致在临时表 BookTmp 中添加两行新数据。但仅有最当前的这行数据是相关的。这个最当前行即为

对于同一个用户拥有 BTmp_Counter 属性最大值的那行数据。

通过实验认为，记录插入的过程是持续数据集成的一个固有方面，这种数据集成方法使用空的或者比较小的表，中间没有任何限制以及附带与之相关的物理文件，通过记录插入过程，可以确保对实现目标提供了最简单最快的逻辑上和物理上的支持。

实际上，在数据仓库架构中的逻辑和物理结构的显著改变，就如图 3 中显示的那样，这就可以使 ETL 过程的执行在这种方式下进行，为了减轻复杂度，通过简单标准的 SQL 指令或者 DBMS 装载软件进行数据装载，如 SQL*Loader,在数据区中，所需数据可以很容易的获取，并且独立使用 ETL 工具。

3.2 ETL 装载过程

为了刷新数据仓库，当 ETL 应用提取并把 OLTP 数据转换成合适装载入 DW 数据区时，它应该快速的插入一条记录作为临时表的新行，并用自增序列数字填充唯一标识属性。在执行相应的打包和再次优化技术后，这个数字自 1 开始标记，然后为每行新增数据自动加 1。使用 ETL 工具完成持续数据集成的算法类似于：

在 OLTP 系统中为每条新记录创建触发器(在 commit 以后)

(1) 从 OLTP 系统中提取新记录

(2) 清除和转换 OLTP 数据，使之转换成目标表的格式。

(3) 将增加记录插入值统计到计数器中

(4) 在数据仓库临时目标表中创建一条新记录

(5) 插入数据到数据仓库临时目标表的记录中，并且插入记录的值到唯一计数器中。

(6) 结束触发器

下面，就用一个实例来说明数据仓库的更新情况，图 4 中展示了在数据仓库的临时表中插入的数据，其记录反应的是一个会员电话预订打高尔夫的整个状态过程,在临时表的每一个状态变化中,BTmp_Counter 的值会自动加 1,根据预订单号 BTmp_BookNO,以及起计数器作用的 BTmp_Counter，可以知道 BTmp_Counter 值最大的这条记录才是最新的值。其中，在临时表中属性的增加不是随意的，是由数据库管理员完成的。在数据仓库中最有效的数据是数字以及附加物。

BTmp_BookNO	BTmp_CardNO	BTmp_MemberName	BTmp_BookState	...	BTmp_BookTime	BTmp_Counter
2009071021	5000000007	陈小虎	订单受理	...	2009-7-11 23:23:54	1101
2009071021	5000000007	陈小虎	到帐确认	...	2009-7-12 8:23:21	1102
2009071021	5000000007	陈小虎	球场联系	...	2009-7-12 9:19:35	1103
2009071021	5000000007	陈小虎	球场传真	...	2009-7-12 9:20:09	1104
2009071021	5000000007	陈小虎	发短信息	...	2009-7-12 9:20:32	1105
2009071021	5000000007	陈小虎	已到场	...	2009-7-15 9:00:35	1106

图 4　插入数据仓库中临时表的部分数据

数据装载使用了最简单的方法写入数据：添加新的记录，其它的写入方法都需要执行更多的时间和复杂的任务。

3.3 OLTP 查询调整

下面对湖南鹰皇高尔夫决策支持系统使用 OLAP 查询，计算过去一个月会员预订的下场总人数，查询语句如下：

```
SELECT B_MemberName,Sum(B_ActNum)
FROM Book
WHERE BTmp_BookTime >=add_months (SysDate,-1)
AND B_BookState=’已到场’
GROUP BY B_MemberName;
```

为了充分利用架构修改方法和包含在 OLTP 查询响应中最当前的数据，查询被重写应该考虑以下原则：“from”语句应该与被请求的所有原始行数据和临时表中相关数据连接在一起，排除所有来自于“where”语句中的所有限制声明值。对上面语句的调整将在下面显示。从中可以看出所有问题表的相关行都被连接起来用于提供 OLAP 查询问题。在原始架构中，根据约束条件过滤掉数据结构集中所使用的行。调整后的查询语句如下：

```
SELECT B_MemberName,Sum(B_ActNum)
FROM (SELECT B_MemberName,B_ActNum
    FROM Book
      WHERE B_BookTime >=add_months (SysDate,-1)
      AND B_BookState=’已到场’)
      UNION ALL
      (SELECTE BTmp_MemberName, BTmp_ActNum
      FROM BookTmp
```

```
        WHERE BTmp_BookTime >=add_
months(SysDate,-1)
        AND BTmp_BookState='已到场')
    GROUP BY B_MemberName;
```

这样修改的优点是：如果 OLAP 用户想查询最当前的信息时，他们只需要复制临时表。比如，临时表中装载了在其创建之前的会员预订的所有记录，而用户只想知道会员当天的预订记录，可以通过以下 SQL 语句获得：

```
    SELECTE BTmp_MemberName,
Sum(BTmp_ActNum)
    FROM BookTmp
    WHERE BTmp_BookTime=SysDate
    GROUP BY BTmp_MemberName;
```

通过这种方法，有助于得到数据仓库中最新的数据，因为这些数据储存在复制的临时表中，而这些临时表占用的空间极小，使得在最新数据查询处理过程中 CPU,内存以及 I/O 成本消耗降到最低。从理论上来说，这也使得当交易发生时，传递最新的决策信息将成为可能。

3.4 小结

在整个 OLAP 资源系统中，只有插入新的记录才会使数据仓库更新，因为这个过程不需要对任何表中的数据进行锁定，也不需要在写入数据前(如更新或删除操作)对之前已存储的数据进行搜索，所以可以把时间降到最低，由于在复制表中没有建索引、主键。因此也就不需要去锁定记录，此外，这些复制表也没有任何的限制，如参照完整性。那么也就不需要索引更新和参照完整性等进行交叉检查耗时较大的操作。

4 结语

本文提到了关于实时数据仓库的基本要求，并且提供了支持实时数据仓库实施的方案，实现了持续数据集成的同时，尽量减少在用户结束数据仓库执行查询时所受的影响。通过实现数据结构的复制和修改查询语句，充分利用了新实时数据仓库的架构。

在接下来的工作中将使用 ETL 工具，将上述方法同 OLTP 系统中的例程整合起来，这将为本方案查询语句的优化提供了很大的上升空间。

参考文献

1 Ponniah P.段云峰,等译.数据仓库基础.北京:电子工业出版社, 2004.

2 Inmon WH. Building the Data Warehoue. John Wiley &Sons, Inc, 2003.

3 张旭峰,施伯乐.增量 ETL 过程自动化产生方法的研究.计算机研究与发展, 2006, 43(6):1097－1103.

4 张俊,张忠能,实时数据仓库体系架构的研究.计算机工程, 2004,30(z1):180－183.

5 Basu R. Challenges of Real-time Data Warehousing DM Review, 2003.

6 Yao A Z X. Scheduling for data warehouse ETL processing and data mining execution. [2009-6-7] http:// www.freepatentsonline.com/7058615.html

7 尤玉林,张宪民.一种可靠的数据仓库中 ETL 策略与架构设计. 计算机工程与应用, 2005 41(10):173－174.

8 Michael H. Real Time Warehouse Evolution. Data Warehouse: The Next Stage in Data DM Review, 2003.

9 Simitsis, A. Mapping Conceptual to Logical Models for ETL Processes, OI