Web 环境中的数据仓库维护①

朱红梅, 张 晶, 李 一民

(昆明理工大学云南省计算机技术应用重点实验室, 云南昆明 650051)

摘要 数据仓库作为一种对分布式信息源集成的关键技术而出现. WEB 环境中信息源动态性的出现, 对数据仓库方法提出了挑战. 本文给出信息源的动态类型及其特点(数据更新、模式变化和约束调整)以及它们产生的原因. 探讨了当考虑信息源变化的可能性时数据仓库的维护方法, 包括从包装件、视图定义和数据仓库内容三个方面来适应信息源的变化. 最后, 探讨了 EVE 系统.

关键词: 数据仓库; Web 环境; 动态性; 包装件; EVE

中图分类号: TP393. 4; TP392　　文献标识码: A　　文章编号: 1007- 855X(2001)03- 122- 04

0 前 言

近年来, 分布式信息系统的应用日益广泛. 这些系统的信息源一般通过网络进行异地互连, 因此将不同信息源的数据进行集成就成为一个很重要的研究课题. 那些得益于数字信息的应用迫切地需要一种适当的集成工具, 从而能够充分利用那些分布式的异构数据源.

在比较当前不同的信息集成方法之后, 我们发现: 裁减式信息库构造方法, 也就是我们常说的数据仓库(Data Warehouse)方法, 能够满足我们的要求. 这是因为, 数据仓库(Data Warehouse)方法具有以下特性:

(1) 在安装阶段, 相关的信息从网络上不同的信息源提取, 经过必要的剪裁和转换, 再与来自其他来源的信息组合, 然后载入中心数据库——数据仓库;

(2) 在查询处理阶段, 针对系统的查询需求将直接从数据仓库中得到结果, 而与原始信息源没有太多的交互;

(3) 在运行阶段, 对数据源的修改将过滤出相关信息, 然后按照某些方式传送给数据仓库进行升级;

数据仓库所具有的这些特性, 使得数据仓库实体化视图能够从一个位置查询到所需的全部信息, 因此具有较高的实用性和较好的查询执行效率, 因此, 当进行高速查询处理和数据分析时, 数据仓库技术是一个明智的选择. 同样地, 这种方法在信息源访问代价过于高昂, 网络经常发生延迟以及转换和集成这些中间任务过于复杂, 从而要求人工输入时也是可取的.

对现在许多应用来说, 数据仓库技术是相当卓有成效的. 这些应用的范围很广, 包括商业销售分析、旅游相关信息、医疗科学综合诊断信息、数字图书馆以及远程教育等等.

1 数据仓库体系结构[1]

信息集成技术通常具有多层结构的特征. 在基础环境中, 通常有许多不同的各种各样的信息源, 每一种根据不同的数据模型

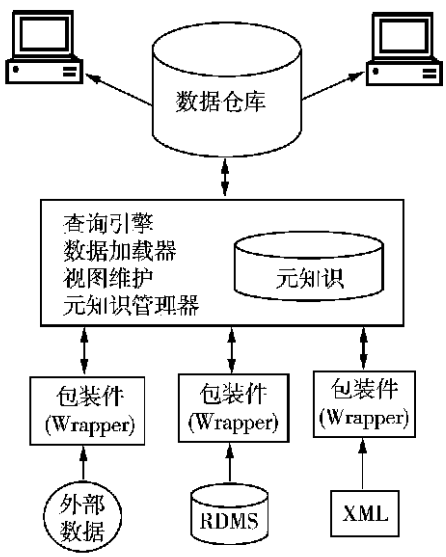

图 1 数据仓库体系结构

① 收稿日期: 2000- 11- 27;
第一作者简介: 朱红梅(1969~), 女, 硕士, 研究方向为数据库原理与数据仓库.

建模, 且各自支持不同的查询界面. 它们可能是传统的关系数据库系统, 面向对象的数据库系统, 也可能是 HTML、SGML 或 XML 文档等新技术. 数据仓库体系结构如上页图 1 所示.

数据仓库存储是一种中央集中式的知识库(至少在逻辑上如此), 它必须支持高速执行的复杂决策支持查询. 由于关系数据库的技术成熟, 因而数据仓库存储也利用了这个技术. 因此, 可定义与具体信息源相关的包装件(Wrapper)来连接源和系统. 包装件负责把源模型转换成数据仓库系统的公用模型, 包括关系模式和约束的剥离, 从指定源到数据仓库系统的大块数据传输, 查询请求的映射, 以及信息源变化的信息.

数据仓库的中间件是一些工具集, 负责对独立信息源提供的服务进行剥离, 对数据仓库存储进行适当的管理. 例如, 包括对各个信息源的信息进行过滤和合并的工具, 管理信息空间的元描述工具, 以及在任何信息源变化情况下维护数据仓库存储的工具. 最近几年, 大量的软件商已开始为数据集成开发"中间件"软件, 它们主要应用在基于 Web 的电子商务.

2　信息源的动态类型

独立信息源通常是自治的, 且除了支持数据仓库本身之外, 它有自己的存在意义和一定的目的. 事实上, 独立信息源可能不会意识到访问它们的客户之一也许是与集成系统相连的包装件. 信息源自治的一个重要结果是在较高层的数据综合层的控制之外, 这些源会发生自我改变. 许多信息源, 特别是基于 Web 的数据库, 也许不仅改变它们的数据, 而且在不与用户配合的情况下, 改变它们的性能.

在中间层, 可见的变化的类型可被分为下面几类:

(1) 数据更新, 如值的改变和添加元组;

(2) 关系模式的变化, 如加列, 删表;

(3) 约束调整, 如去除主健;

(4) 统计和元数据调整.

从大量的观察来看, 这些变化都是可能的, 因此, 需要研究它们对数据仓库的影响. 大多数商业数据仓库系统只处理数据更新时的消息传递, 但常以批模式的方法处理(这是指它们先收集所有更新, 在一个工作期后, 立即把它们加载到数据仓库仓储). 最近的研究已有了进展, 可以在数据实时更新时进行数据仓库的维护和数据传递.

3　数据仓库的维护[2]

从动态信息源上集成数据的数据仓库, 其维护是十分复杂的. 现在从三个方面探讨数据仓库系统的维护方法:

3.1　在包装件层次上的维护

当前, 包装件的许多解决办法是建立在信息源合作的基础上的, 在这种意义上, 信息源向数据仓库的更高层次报告它们的数据更新. 对于独立的信息源, 正在研究发现变化的各种方法和策略. 如果不是数据更新, 而是其它变化, 则需要开发算法来查找各种信息源中的数据和模式更新, 以便为一个信息集成系统中可能的数据提供者提供较高层次的类.

假设信息源可以用一个定制的包装件来集成到数据仓库中, 那么在模式变化后, 这种包装件的演化(理想状态时, 不需要人工输入)变得非常重要, 但实现起来十分困难. 如果发布无地址性, 对集成系统来说动态信息源会因连接超时而不可访问. 带自身演化机制的包装件生成器技术的利用, 是解决这个问题的关键一步. 最好的解决方法应该包括通过包装件转换所有更新类型的能力.

就独立的信息源而言, 各种数据仓库或其它集成服务可能需要在不同的时段查询信息源. 而且, 当包装件碰巧要发布更新通知时, 数据仓库管理员不可能准备好接受更新通知. 许多计算上复杂的变化将使得算法的产生相当昂贵, 也就是不可行的. 因此, 动态数据仓库体系结构的一个重要组成部分就是记录所有变化情况的历史服务器. 它收集来自信息源的所有变化, 提供基于中间层的信息源变化的查询服务. 这样的服务器能够支持对来自中间层工具的固定查询请求的注册登记.

3.2 提高数据仓库定义的适应性

如果数据仓库关系模式是根据指定基础信息源输出模式的结果来定义, 信息源的动态性可能使这种传统特性遇到难题. 因此, 需要在技术上要么正确传递变化给数据仓库定义本身, 要么把数据仓库与它们适当地隔离开.

许多数据仓库系统, 特别是建立在中型、大型信息环境之上的数据仓库, 使用某种形式的元知识来适应特定信息源导入数据仓库的发现和集成. 当没有必要包括一个静态全局模式时(这个模式是为构造一个协作式数据库系统来提取所有信息源的关系模式的共同点而建立) , 这样的元知识基础(MKBs) 典型地包含关于所有被注册的信息源的关系模式和性能的信息. 如果信息源是动态的, 必须开发算法解决元知识本身以便继续正确反映基础信息空间.

3.3 调整数据的内容

一个数据仓库可能有特定目的数据装入, 这种装载向仓库提供抽取的数据集. 一旦一个数据仓库的定义和模式结构已经演化, 有必要相应地调整数据仓库存储内容. 由于对系统性能上的要求以及数据仓库存储尺寸上的要求, 这种数据仓库内容调整理想上是渐进地完成, 而不是完全的数据重载[1]. 因此, 数据仓库内容的可调整增量技术成了一个关键.

在信息源数据更新的情况下, 数据仓库的维护可能会构建得相当好, 而在数据和模式都有更新的混合模式下, 维护将变得相当困难. 而且由于模式信息方面的动态性, 增量数据仓库维护算法就需要处理信息源更新和维护查询的并发性. 而且, 由于信息源模式的变化, 提交给信息源的维护查询遇到的是一个变化的模式, 它与提交的查询格式不再匹配, 因而查询的结果也是失败的[3].

4 EVE: 迈向可伸缩的数据仓库[4]

可演化的视图环境(Evolable View Environment, 简称 EVE) 工程是全面确定动态数据源的第一个尝试. 当一个信息源加入到系统中来, 它把内容、性能以及与其它数据源之间可能的相关性传递给 EVE.

EVE 的一个核心特点是可演化的 SQL(E- SQL) , 它是 SQL 的扩展, 允许视图定义器拥有对视图演化的优先选择. 使用 E- SQL, 定义视图的用户可指定什么信息是不可缺少的, 什么信息可被其它信息源的相似信息代替, 以及是否一个变化的视图范围是可接受的(这个由加在视图查询元素上的优先参数决定) . 关键的一点是将视图定义重写到另外一个视图(该视图仍保存有用户的目的语义) 来演化视图.

一旦系统接收到一个影响视图源模式的变化, 视图同步模块以一种用户能接受的方式决定视图查询重写的替换技术, 这种重写带有适应数据仓库定义的目的. 视图同步算法基于可得到的元知识, 对被影响的视图组件找寻合适的替代物, 对于 E- SQL 标识不出的非主要视图组件将被删除. 相应地, 元知识基础(Metaknowledge Bases, 简称 MKBs) 演化器将演化元知识以便与信息空间的调整状态相匹配. 因为视图同步算法可以生成许多可能的查询重写, 这就需进行新的视图定义. 出于这个目的, 用质量代价模型(Quality- and- Cost) 来估计重写的质量和代价. 每一个可能的查询重写通常将保持不同量和不同类型的信息. 同时, 每个新视图查询可能引起不同的视图维护代价. 用这两个维度, QC 模型可互相比较不同的视图查询.

视图维护者通常应用不同的策略来更新数据仓库的数据内容, 包括用视图同步器调整视图定义以及当源数据更新后调整视图定义. 最后, 并发控制模块使演化视图定义的方法具体化, 也使数据并发和模式更新情况下维护视图内容. 由此, 已有的视图同步和视图维护算法被集成进系统, 在共同的协议下共存和协作.

5 结束语

数据仓库作为一种对分布式信息源集成的关键技术而出现. 然而, 象 Web 这样的网络环境下, 必须考虑自治、异构和动态信息源的信息集成方法. 由于信息源的动态性将被集成, 对数据仓库方法提出了挑战. 本文给出动态类型的特点(数据更新、模式变化和约束调整) 以及它们产生的原因. 探讨了当考虑信息源变化的可能性时数据仓库系统的维护, 给出解决这些问题的一些方法, 特别是探讨了 EVE 系统, 这种可伸缩

的数据仓库技术允许更多的用户充分利用网上的分布式信息, 可以提高用户和系统管理员的效率, 甚至在基础系统变化的情况下自动维护信息的用户界面.

参考文献:

[1] 王珊等. 数据仓库技术与联机分析处理[M]. 科学出版社, 1998. 6.

[2] Agrawal, D., El Abbadi, A., Singh, A., and Yurek, T. Efficient view maintenance at data warehouse[J]. In Proceedings of SIGMOD, 1997, 417~ 427.

[3] Chawathe, S. S. Abiteboul, S., and Widom., J. Representing and querying changes in semistructured data[J]. In Proceedings of the International Conference on Data Engineering. (Feb. 1998), 4~ 13.

[4] Lee, A. J., Koeller, A., Nica, A., and Rundensteiner, E. A. Data warehouse evolution: Trade– offs between quality and cost of query rewritings[J]. In Proceedings of IEEE International Conference on Data Engineering. Special Poster Session, Sydney, Australia (Mar. 1999), 255.

Maintaining Data Warehouses on Web Enviroments

ZHU Hong– mei, ZHANG Jing, LI Yi– min

(Key Lab of Computer Technology Application of Yunnan Province, Kunming University of Science and Technology, Kunming 650051, China)

Abstract Data warehouse have emerged as on key technology for the integration of distributed information sources. Challenging issues for data warehouse solutions arise due to the dynamicity of the information sources on Web environments. This article shows up the types of dynamicity (such as data updates, schema changes, and constraint modification) as well as their generation. This article also identifies maintaining for data warehousing systems that occurs when the possibility of information source changes is taken into consideration. Such maintaining includes adapting wrappers to information source changes, adapting view definition to changes, or adapting the data content of the data warehouse. Lastly, Evolvable View Environment (EVE) system is outlined.

Key words: Data Warehouse; Web environments; Dynamicity; Wrapper; EVE