计算机科学1998 Vol. 25 No. 4

数据仓库技术的研究现状及未来方向*)

Current Situation and the Future Direction of Data Warehousing Research

李子木 莫 倩 周兴铭

(国防科技大学计算机系 长沙 410073)

摘 要 Data Warehouse as well as On-line Analytical Processing(OLAP) is an emerging research field in Information Technology. International researchers studied broadly and deeply in Data Warehouse architecture, data organization, view maintenance, multidimensional database (MDDB) modeling, data cube computation and other issues. This paper describes the latest developments of the research in Stanford University, IBM Almaden Research Cenrer, Wisconsin University, Microsoft and AT&T, through which depicts the current state of the art and points out the furure research directions

关键词 Data Warehouse, Materialized view, Multidimensional data, OLAP, Data cube

数据仓库和联机分析处理是决策支持系统的重要组成部分，与传统的联机事务处理不同，是对现有数据进行归纳、分析和推理，从而为决策提供支持。

数据仓库是“面向主题的、集成的、稳定的和随时间变化的数据集合，主要用于决策制定”[1]。数据仓库的这些特点决定了它与传统的面向事务处理的数据库有着本质不同。作为一个新兴的研究领域，数据仓库发展得很快，许多大学和公司都正在这个领域内进行着广泛深入的研究，其中尤以斯坦福大学、IBM Almaden 研究中心、威斯康辛大学、微软和 AT&T 的研究最具代表性。

斯坦福大学正在进行一个名为“WHPS(Warehousing Information Project at Stanford)”的科研项目，他们的研究目标是要生成一些高效的、自动集成异构数据源的算法和工具。这个课题组已经提出了一个基本的数据仓库模型(图1)和一些相应的算法。IBM Almaden 研究中心和微软正在进行一个称为“Quest”的项目。他们的研究重点是多维数据库的建模与组织。威斯康辛大学和 AT&T 的研究侧重于实视图、OLAP 数据组织、数据立方体计算等方面。本文以它们的研究内容为主线，分析总结了数据仓库技术的研究现状，并在文章的最后指出了数据仓库的未来研究方向。

数据仓库中存储的大量数据，通常来源于一个或几个独立的数据源。数据仓库的主要功能是为 OLAP 提供支持。OLAP 查询分析通常需要涉及大量的数据，而且一般要对数据进行投影、连接、分组等复杂处理，这是一个非常耗时的过程。OLAP 却要求它的查询能够被快速响应。于是数据仓库便针对 OLAP 可能的查询对原始数据进行投影、连接、分组等预处理，建立许多“实视图(materialized view)”。它与数据库的“视图”概念不同之处在于：它不是“虚拟”的，而是已经过计算，含有大量数据，并存储在数据仓库中的一张实实在在的表。

通过这些预计算，OLAP 基本上不再需要对原始数据进行复杂处理，而只需在实视图的基础上进

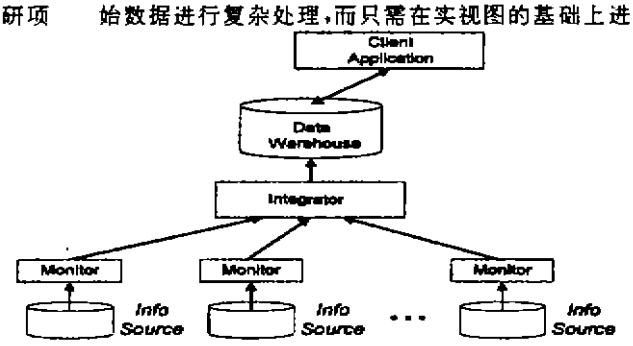

图1 数据仓库基本体系结构

1 视图维护

*)本文研究得到九五预研项目基金资助。李子木 博士生，研究领域为分布式并行数据库、数据仓库；莫 倩 博士生，研究领域为分布式数据库、数据库协同工作；周兴铭 博士生导师，中科院院士，主要研究领域为分布式数据库、分布并行计算、高性能计算机体系结构。

行一些简单的计算便可以完成复杂的查询，从而大大地提高了系统性能，缩短了数据仓库的响应时间。

实视图提高了系统的响应时间，但同时也带来了许多新的问题。数据仓库中的数据来源于其他独立的传统数据库，当这些数据库的原始数据发生变化（插入、删除、更新）时，如何使得数据仓库中的实视图与原始数据的变化保持同步，便成为目前数据仓库技术研究的一个重要领域；另一方面，由于数据的属性连接组合非常多，鉴于空间限制和将来的维护的原因，不可能将它们全部实体化（materialize），只能选择其中一部分进行实体化。那么选择哪些视图进行实体化，以及如何选择也成为目前数据仓库研究的一个热点。下面分别对这两个问题进行讨论。

1.1 视图的一致性维护

实视图存储在数据仓库中，而原始数据存在于传统的数据库中，每次当原始数据发生改变时，如果在数据仓库端对实视图频繁地重新计算，则代价将是非常高昂的。因此，目前的研究主要集中在"增量视图维护"方向。在这个方向上，斯坦福大学的研究小组走在最前面。

1.1.1 增量视图维护。斯坦福大学的 Y. Zhuge 最先提出一种基于一个数据源的视图维护方法：ECA 方法（Eager Compensating Algorithm）。这种方法基于 FIFO 模型，针对原始数据的变化构造"补偿请求"查询原始数据，并将查询结果反映在实视图中[2]。随着，Y. Zhuge 又针对"Single update transaction"、"Source-local transaction"和" Global transaction"三种数据源情况，提出了一套 Strobe 算法，来维护多数据源情况下实视图的一致性[3]。

1.1.2 视图的自维护。在通常的增量维护方法中（如 ECA 方法），当视图更新时需通过网络访问数据库中的原始数据。当更新非常频繁或网络带宽有限时，网络将成为瓶颈。为此，将一些原始数据的备份放在数据仓库端，就能极大地提高系统的查询响应时间。视图的自维护便是基于这样一种思想，将维护视图时需要的一些原始数据在数据仓库端做一拷贝，这样，当需要查询有关的原始数据时，便可以直接在数据仓库端找到，而无需再通过网络向数据库请求。

这样带来的一个问题是，数据仓库不可能将所有涉及到的原始数据都在数据仓库端做一备份，而只能选取其中的一部分，选取哪些？如何选取？便是视图自维护所要解决的首要问题。Huyn 和 D. Quass 等人分别在[4]和[5]中讨论了如何通过关键字、子视图，以及完整性约束等方法对视图进行自维护。

1.2 实视图的选择

为了快速响应 OLAP 查询，数据仓库中存储着许多实视图。这些实视图占据着大量的存储空间，而且它们的一致性维护也需要占用大量的 CPU 时间。如何在存储空间有限，用于维护视图的 CPU 时间有限，同时又要最大限度缩短 OLAP 查询时间的情况下，选择视图进行实体化已成为目前数据仓库研究的重要内容。

H. Gupta 对视图结构中的两种基本情况进行了讨论：当母视图必须由与它所关联的所有子视图都参与计算才能得出时，叫作"与"视图结构；当母视图可以由与它关联的任一子视图单独计算得出时，叫做"或"视图。H. Gputa 对这两种基本视图结构提出了一个通用解决框架，并讨论了由它们构成的简单混合情况。V. Harinarayan 利用图论中的"网格"模型对这一问题进行了讨论；通过"网格"可以揭示视图间的依赖关系，利用这些关系可以对选择进行优化。J. Yang 则另辟蹊径，选择大部分视图可以"共享"的公共子视图进行实体化。

2 数据立方体

数据立方体是多维数据库的一种形象表示方法。多维数据库不同于传统的关系数据库，它以多维方式组织数据，并以多维方式显示数据。作为一种新的数据库结构，目前的研究主要集中在两个方面：建模和计算。

2.1 多维数据库建模

这个领域的研究要首推 Jim Gray 等人引入的"CUBE"操作符[6]。"CUBE"将数据属性分为"维"（dimension）和"度量"（measure），将数据由原来的二维关系表组织成多维立方体，在其上进行旋转（pivoting）、切片和切块（slicing&dicing）、下钻（drilling-down）和上翻（rolling-up）等操作。

"CUBE"的引入，极大地提高了传统 SQL 对多维数据的支持，但它的不足之外是没有将"维"和"度量"统一起来，是不对称的。当数据仓库对 OLAP 查询不可预测时，数据仓库无法区分哪些属性是"维"，哪些属性是"度量"，从而无法对它们进行预计算。于是，IBM 提出了一个完全独立的、对称的多维数据库模型[7]。它统一对待"维"和"度量"，并在多维空间定义了 Push、Pull、Destroy Dimension、Restriction、Merge 等基本操作运算，构成了多维数据库的原始模型。

2.2 立方体计算

多维数据库一般建立在数据库之上，所以多维数据库的旋转、下钻、上翻等操作从本质上讲是一组 SQL 语句或一组视图（虚拟或实体）。因此，如何将多维数据库与已有的较成熟的技术（如视图、索引等）统一起来，对立方体进行多维计算及优化，也是目前研究的一个热点。

I. Mumick 描述了一种"summary-delta 表"，利

用它，数据仓库无需对所有数据进行重新计算，只需计算变化的部分，从而减少了维护数据立方体的代价。

立方体中的"CUBE"操作是将立方体的所有"维"进行连接分组，S. Agarwal 等人在[8]中描述了如何用一组层次化的 Group-by 操作来实现"CUBE"这一操作运算。

3 索引优化

不论是数据库还是数据仓库，索引始终是一个重要的研究内容。数据仓库与数据库结构的不同，使得它的索引结构也应作相应的变化。由于数据仓库是为 OLAP 提供支持的，因此它执行的大部分操作是只读操作，而且数据仓库中的数据是相对稳定的，这样，就可以为数据仓库中的数据建立多种索引以提高查询性能，文[9]对这一问题进行了深入讨论。

数据仓库中的索引一般是在视图实体化之后另外单独建立的。由于索引同实视图一样也要占用大量的存储空间，所以应在选择实视图的同时将索引的建立也考虑进去，这样才能从整体上使系统得到最优。文[10]对这一问题进行了分析。

4 数据仓库的未来研究方向

前面我们对数据仓库技术的研究进行了综述，其中有些领域已经进行了较为深入的工作，有些领域也是最近才涉及的。在数据仓库技术中，除了以上所述的研究领域外，仍有许多其它领域等待着我们去研究。下面列举其中的一部分。

数据仓库多采用"雪片"(snow-flake)模式，其优点是可以方便地组织层次结构数据。在层次结构下，当基表发生变化时，如何传播变化，如何高效维护各级实视图的一致性，是实现"下钻"和"上翻"的关键。层次问题是数据仓库必须解决好的一个基本问题。

数据仓库存储的主要是历史数据，时间在数据仓库中是个很重要的概念。目前大多数商用产品都没有为用户提供表示历史数据的专门机制，只是提出了各种各样合并历史数据到数据仓库的方法。怎样表示时间？怎样处理其他"维"和"时间维"的关系？当时间变化时，怎样调整数据仓库模式？这些都是下一步要研究的课题。

数据仓库的物理设计是另一个需要认真研究的课题。目前的大多数数据仓库产品是在原来传统数据库基础上做一些索引、并行扫描等工作，以适应 DSS 要求。从本质上说，这些产品的核心仍是面向 OLTP 的。OLAP 和 OLTP 有着本质上的不同，若要使数据仓库对 DSS 支持得更好，必须从数据仓库的核心设计开始就面向 OLAP，这包括内存设计、Cache 算法、表格分割、存储控制、数据压缩与索引等。

数据仓库是一个庞大复杂的系统，如何对它进行有效而正确的管理已经成为数据仓库研究的重要课题。它包括许多内容：资源管理、查询调度、数据组织、视图管理、前后端功能分割、元数据组织等，尤其是当数据仓库正在调入或刷新大量实视图和索引时发生了故障，系统应如何建立检查点，如何恢复。这些领域都需要做进一步的深入研究。

系统的可用性是衡量系统好坏的重要标准。如果一个系统不能随时为用户提供服务，我们说它的可用性是较差的。事实上，目前的大多数数据仓库产品正是这种情况，它们都是利用夜晚时间对数据仓库中的数据进行脱机维护。随着应用的不断深入，当数据量非常大时，这种维护方式所用的时间会越来越长；另外，随着企业业务的全球化，专门用来进行更新维护的"夜晚时间"会越来越难以确定；特别是一些对实时要求比较高的关键任务，如战场决策，这一问题将变得更加严重。如何在联机状态下为用户提供服务的同时对系统进行热维护，是数据库未来研究的重要方向。

参 考 文 献

[1] W. H. Inmon, Building the Data Warehouse. QED Technical Publishing Group, 1992

[2] Y. Zhuge et al., View Maintenance in a Warehousing Environments, Proc. of the ACM SIGMOD Conf. 1995

[3] Y. Zhuge et al., The Strobe Algorithms for Multi-Source Warehouse Consistency, Proc. of the Conf. on Parallel and Distributed Information Systems, Miami Beach, Florida, Dec. 1996

[4] N. Huyn, Exploiting Dependencies to Enhence View Self-Maintainability, Technical Report, Stanford University, 1997

[5] D. Quass et al., Making Views Self-Maintainable for Data Warehousing, Same to[3]

[6] J. Gray et al., Data Cube: A Relational Aggregation Operation Generalizing Group-by, Cross-tab, and Sub-totals, Proc. of the ICDE, 1996

[7] R. Agawal et al., Modeling Multidimensional Databases, Proc. of the ICDE, UK, 1997

[8] S. Agarwal et al., On the Computation of Multidimensional Aggregates, In the Proc. of VLDB Conf., 1996

[9] P. O'Neil, D. Quass, Improve Query Performance with Variant Indexes, Proc. of the ACM SIGMOD Conf., Tucson, Arizona, USA, May 1997

[10] H. Gupta et al., Index Selection for OLAP, Same to [7]