

建立 Web 信息集成系统^{*}

Building Web Information Integration Systems

蔡俊杰 孙建伶 董金祥

(浙江大学1564信箱 杭州310027)

Abstract The Web is based on a browsing paradigm that makes it difficult to retrieve composite information from multiple sites. Web information integration systems, as a sophisticated solution to this problem, enable easy retrieval for comprehensive information from the Web. Based on a prototype we have implemented a Web information integration system, which is called Aoede. This paper introduces the design and implementation of the systems by analyzing many of its main issues, including architecture, information resources, wrapper, mediator, query optimization and execution. Difficult problems are highlighted and respective possible solutions to them are also given.

Keywords Web information integration, Mediator, Wrapper, Query executor

Web 是一个巨大的信息库,它由成百万个异构的 Web 信息源组成,宏观上如同一个庞大的联邦数据库^[1]。随着 Web 规模的日益庞大,要在 Web 中查找感兴趣的信息变得越来越困难。通常的做法是先用搜索引擎进行关键字匹配查找,然后从搜索引擎返回的网站、网页列表中挑选几个进行访问以获得具体的信息。但是搜索引擎只起到提示引导作用,它所提供的查询功能非常有限。Web 上相关信息的分布特性使得用户经常必须访问多个站点才能获得比较全面的信息,不仅费时费力,而且用户往往迷失于 Web 网页之间错综复杂的连接之中而无法有效地获得准确的信息。

Web 信息集成系统通过把相关的 Web 站点的信息综合起来并提供统一的查询界面,使用户可以直接获得完整的信息却不用关心信息的具体来源和获得方式,使用户获得如同数据库查询一样的服务。

Web 信息集成系统具有诱人的应用前景。例如,可以通过集成多家网上商店的商品信息进行网上导购;可以把跟旅游相关的所有站点集成起来实现旅游综合服务,等等。

Web 信息集成近几年开始受到数据库、Web 和人工智能研究者的关注^[2,3],开始了一些研究项目,开发了几个原型系统,如:Southern California 的 Ariadne^[4]、AT&T 的 Whirl^[5]、Washington 大学的 Information Manifold^[3],以及 Georgia 大学的 INFORHARNESS^[6]等。

Aoede 是我们开发的一个 Web 信息集成系统^[7],它实现了对中文歌坛信息的全面集成,能根据用户的查询要求自动地从多个相关的站点中搜索歌坛信息,并把得到的信息综合起来返回给用户。Aoede 针对 Web 的最新发展趋势,大量采用隐藏页作为信息源;开发了专门的并行查询技术,使得系统具有较高的查询处理能力。

本文结合 Aoede,介绍建立 Web 信息集成系统所面临的主要问题,包括体系结构、信息源、中介器(mediator)、包装器(wrapper)、查询优化与执行等,指出 Web 信息集成系统的关键问题及其可能的解决方法。

1 体系结构

建立 Web 信息集成系统可以采用两种不同的途径:数据仓库和虚拟数据库^[2]。数据仓库把需要的数据从多个 Web 信息源下载到本地从而建立一个数据库,用户的查询都在本地数据库上进行。其优点是查询性能可以得到保证,缺点是所有数据都需要冗余,而且实现数据同步非常困难。

虚拟数据库则是将用户的查询分解成对各个信息源的子查询后再分别执行,然后把它们返回的数据综合起来得到最后的查询结果。其优点是数据不需冗余,并保证总是最新的。但由于 Web 信息源的独立性,若在保证查询性能就必须设计合理高效的查询执行计划并进行有效的查询优化。与数据仓库相比,在信息源数

^{*} 本文研究得到航天工业总公司国防预研基金资助。蔡俊杰 硕士生,主要研究领域为数据库、Web 信息集成。孙建伶 副教授,主要研究领域为对象数据库、Web 数据管理等。董金祥 教授,博士生导师,主要研究领域为工程数据库、CAD、CIMS 等。

量很多而且数据更新频繁的 Web 环境中,虚拟数据库更适合作为实现 Web 信息集成系统。

Aoede 把两种途径结合起来,探索了一种基于缓冲的虚拟数据库途径,即把信息源中相对稳定的数据存储在本地,而其他更新比较频繁的数据则在查询执行过程中动态地从信息源中获取,从而既满足了数据的实时性要求,又保证了较高的查询效率。

Aoede 的体系结构如图1所示。

Web 信息集成系统与传统数据库应用相比有两个截然不同的特点。

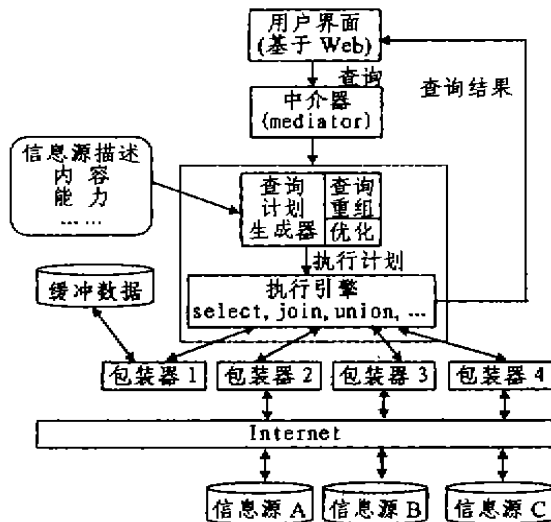


图1 Aoede 的体系结构

首先,数据组织方式有差异。在传统数据库应用中,开发者首先根据应用的需求设计一个数据库模式,然后通过插入记录等步骤实现这个应用。而 Web 信息集成系统是在已有的 Web 信息源的基础上建立的,开发者需要设计的是一个中介器。中介器是一系列虚拟关系,是对多个物理信息源的逻辑综合,类似于分布式数据库的全局模式。用户的查询请求是针对中介器的。中介器的关系通常并不存放在本地,而是在查询过程中从其对应的信息源提取数据然后组装起来。在处理用户输入的查询时,系统必须首先将其分解成对信息源的模式的查询,这个过程称为查询重组。为了进行查询重组和优化,系统需要一系列关于信息源的描述信息。

其次,为了获取数据,Web 信息集成系统的查询执行引擎并不访问本地的存储管理器,而是调用一系列称为包装器的程序。包装器实现对每个信息的抽象,向上层屏蔽信息源的物理细节,并在查询过程中完成与信息源的交互,把需要的数据从 Web 信息源返回的 HTML 代码中提取出来,然后转化成可以被系统进一

步处理的形式。

图1中,包装器2、3和4是典型的包装器,而包装器1则用来读取保存在本地的缓冲数据,目的是统一执行引擎与数据之间的界面,使执行引擎不必关心数据的具体来源。

建立一个 Web 信息集成系统的一般过程为:首先根据应用需求寻找和选择合适的 Web 站点作为信息源,然后对信息源进行抽象并开发相应的包装器;在此基础上设计中介器,使其具有足够的能力,最后实现针对中介器的用户查询界面。

2 信息源

信息源的选择直接关系到中介器、包装器的设计,影响到系统性能和能力,它是整个 Web 信息集成系统的基础。

2.1 隐藏页

信息源一般由一系列页面组成,这些页面按查询能力可分为基于表单(Form)的页面和基于层状索引结构的页面,前者又被称为隐藏页(hidden page)。

隐藏页通常具有一定的查询能力。它一般是服务器端的脚本或者 CGI 程序,以 HTML 表单作为其用户查询界面,根据输入的查询条件访问其网站的后台数据库,然后把查询结果包装成 HTML 代码返回给用户的浏览器。隐藏页使大部分查询在信息源端完成而只返回查询结果,大大减轻网络的传输负担,而网络传输是 Web 信息集成系统最大的瓶颈。

另外,据统计目前隐藏页在整个 Web 中所占的比例已经超过80%^[1],也就是说 Web 上的绝大多数信息是以隐藏页的形式提供的,因此 Web 信息集成系统必须尽量选择隐藏页作为信息源。

为了充分利用隐藏页的查询能力从而提高系统性能,必须为其设计专门的接口并作特定的优化。Aoede 开发了专门的并行查询技术,针对隐藏页分批返回数据的现象,Aoede 使用多线程并行读取各批数据,使访问效率大大提高。

2.2 信息源的描述

信息源的描述^[9]是系统进行查询优化和生成查询计划的基础。这些描述包括:信息源提供的属性集合、信息源的响应速度、查询处理能力以及数据的完备性等;其中最关键的是对属性集合的描述,即信息源的内容与中介器的关系间的语义映射,这种映射可以用两种方法表达:GAV(Global as view)和 LAV(Local as view)。

GAV 把中介器中的每一个关系 R 表示为一个对信息源页面的关系 S 的查询,从而指定了如何从 S 中获取 R 的记录。LAV 则是把 S 表示为对 R 的查询。

GAV 的优点是简化了查询重组的过程,而 LAV 的优点是有益于系统的信息源的增减。

3 包装器

包装器的功能相当于一个子查询的执行引擎,能根据给定的条件筛选数据,包装器完成与信息源的所有交互任务,屏蔽信息源的物理差异。

可以让包装器只提取数据,也可以让它再对提取到的数据进行筛选从而实现信息源未提供的查询能力。

在与基于层状索引结构的信息源页面交互时,包装器直接发出页面请求,然后从返回的 HTML 代码中提取数据。在与隐藏页交互时,包装器必须首先提交一个填写了查询条件的表单给信息源,再从其返回的动态页面中提取数据。

因为 HTML 页面是面向显示而不是面向内容的,所以要从 HTML 文档中提取数据并不容易。一种策略是先根据 HTML 标签分析页面的语法结构,然后在特定的语法层次中读取特定的数据。另一种策略是利用页面中的“标志”直接定位需要提取的数据。Aoede 中的包装器采用了后一种策略,因为其效率较高。

无论采用何种策略,手工开发包装器总是不大方便。同时,信息源页面的结构和内容经常会发生变化。这些都使得包装器的开发和维护变得相当困难,因此一个完善的 Web 信息集成系统应该提供能迅速生成包装器的工具。这种工具可以是基于某种定制的页面描述规则的,也可以采用启发式归纳学习的方法来半自动地生成包装器^[10]。

4 中介器

中介器类似于传统数据库应用中的全局模式,是 Web 信息集成系统的全局数据视图。

4.1 中介器的能力

一个 Web 信息集成系统的中介器必须具有足够的能力才能满足系统的应用需求。中介器的能力包括两个方面:

首先是中介器的数据完备性,即,中介器是否提供了足够的数据以覆盖应用的需求。其次是中介器的查询表达力,即,中介器对于它所收集到的数据提供了什么形式的查询,比如,对于歌手的查询,是只能按名字进行点查询,还是能按年龄进行范围查询?

中介器的能力受到信息源能力的制约。通常需要集成众多的信息源才能使中介器获得足够的能力。因此,需要从现有的信息源计算出中介器的能力^[11];另外,为了防止因某些信息源的暂时失效而使中介器的能力受到损害,还可以有意使信息源有所冗余。

4.2 中介器到信息源的映射

查询执行过程中的另一个重大问题是如何确定两个不同的信息源所指的两个对象实际上是真实世界的同一实体。其原因在于每个信息源都使用其特定的命名机制,例如,浙江大学可能被称为“浙江大学”,也可能被简称为“浙大”。

在对多个信息源进行综合查询时必须识别这些同一实体的不同别名才能进行有效的查询操作比如连接。最简单的方法是建立一张别名映射表,但是代价比较大,而且维护困难,另一种方法是用映射函数代替映射表,但其前提是别名之间必须存在固定的联系规则。

Aoede 采用的是由 Whirl 的开发者提出的近似匹配的方法^[12]。这种方法假设,对真实世界中的同一实体,人们倾向于使用类似的名称来标识它们,而一定程度的近似对查询结果造成的负面影响是可以忍受的。

5 查询处理

5.1 查询缓冲

为了克服网络传输这个瓶颈,Aoede 引入了查询缓冲的概念,即把来自单一或多个信息源的常用数据存储在本地作为新的“信息源”。由于 Web 数据的数量庞大和更新频繁的特点,像数据仓库那样缓冲所有数据并不可行,而应有所选择。选择的标准包括访问频率、更新频率、用户的查询对信息的即时性的要求、信息源的结构和组织方式等。根据这些标准,系统就可自动选择数据进行缓冲,也可以由用户来指定。通常,需要做缓冲的是那些没有查询能力的基于层状索引结构的信息源页面的数据。

5.2 查询计划

查询计划生成器根据信息源的描述信息把用户提交的针对中介器的查询重组为对各个信息源的子查询,这个过程包括确定需要用到信息源页面、针对每个涉及到的信息源页面的子查询(例如需要提取的属性、查询条件等)以及各个子查询的执行顺序等。

由于多个信息源之间很可能会有冗余信息,因此查询计划生成器在选择信息源时余地很大。最简单的选择策略是首先访问数据最完备的信息源,如果找不到符合条件的信息再考虑其他信息源。当然,较好的方法是对它们进行并行访问从而使获取的数据具有最大限度的完备性。

子查询之间的执行顺序决定于它们之间的依赖关系。如果子查询 B 需要子查询 A 的输出作为其输入,那么 A 就依赖于 B,这意味着 A 必须在 B 之后执行即串行执行。没有依赖关系的子查询可以并行执行,这将大幅度提高系统性能。

5.3 查询优化和执行

Web 环境的特点使 Web 信息集成系统在进行查询优化和执行时面临着几个重大的挑战。我们通过对 Aoede 系统的开发总结出以下三点:

•统计信息的缺乏:为了估计某个执行计划的代价,优化器需要大量的关于数据的统计信息,比如关系的大小、属性的值的分布等,而在 Web 应用中这些信息不仅很难得到而且也不可靠。

•数据传输的不可预知性:在对信息源进行访问时经常会出现“初始延迟”或者通讯中断的现象,因此即使是一个很高效的计划,在执行时还是可能因为数据传输的不稳定变得效率低下。

•信息源的不稳定性:某些时候可能无法访问部分信息源,虽然有时候可以由镜像站点或者同类信息源代替,但在找不到“替身”的情况下,就要考虑如何最大限度地利用目前可用的信息源来完成用户的查询^[12]。

针对这些问题,Aoede 采用了动态适应性的查询优化和执行方法^[14],即把优化和执行穿插起来进行从而实现动态的优化,这一点与传统数据库管理系统把查询优化和执行截然分开的做法不同(有些分布式数据库也在执行过程中有动态的优化)。比如,针对某些查询系统并不一定要在执行前就制定完整的计划,如果优化器没有足够的信息来比较几个查询计划之间的优劣,那么可以先把计划的一部分提交给执行引擎,根据其实际执行情况决定计划的其余部分。

结论 Web 正以不可思议的速度渗透到人类生活的各个方面,如何更好地利用 Web 上浩如烟海的信息已成为人们日益关心的问题,毫无疑问,Web 信息集成系统将在其中扮演十分重要角色。

然而,目前设计和实现一个 Web 信息集成系统还不是一项简单的工作,虽然在过去的几年中已经取得了很多成果,但是 Web 信息集成的研究还处于起步阶段,有一些困难的问题需要寻求解决的途径。

比如包装器的自动生成一直是个困难的问题。虽然 XML 的出现可能引导网站的建设者们用易于机器阅读的形式来表达其网站的数据,从而大大简化包装器的开发,但 XML 不能解决所有的问题,因为它只能使网页的结构和内容变得清晰,而让机器理解其内容的语义仍然是个难题。

再比如,目前在 Web 信息集成系统的开发过程中,信息源的搜集、整理及描述都必须通过手工完成,当系统规模很大时,这些工作将变得非常繁琐和枯燥。所以未来的研究方向,除了研究更有效的查询优化和

执行方法外,还应该把重点放在如何自动搜索和集成新的信息源这个难题上。

参考文献

- 1 Bernstein P, et al. The Asilomar Report on Database Research. ACM SIGMOD Record, 1998, 27(4)
- 2 Daniela F, Alon L, Alberto M. Database Techniques for the world-wide web: a survey. SIGMOD Record, 1998, 27(3): 59~74
- 3 Hears M A, et al. Information integration. IEEE Intelligent Systems, 1998, 13(5)
- 4 Luis A J, et al. ARIADNE: A system for constructing mediators for Internet sources (system demonstration). In: Proc. of ACM SIGMOD Conf. on Management of Data, New York: ACM Press, 1998. 561~563
- 5 William C. A Demonstration of WHIRL (demonstration abstract). In: Proc. of SIGIR6, New York: ACM Press, 1999. 327
- 6 Shah K, Sherb A. INFOHARNESS: Managing Distributed, Heterogeneous Information. IEEE Internet Computing, 1999, 3(6)
- 7 <http://aoede.mycool.net>, 2000
- 8 Lawrence S, Giles C L. Searching the world wide web. Science, 1998, 280(4): 98~100
- 9 Aoln L, et al. Querying heterogeneous information sources using source descriptions. In: Proc. of the Int. Conf. on Very Large Data Bases, San Francisco, CA: Morgan Kaufmann Publishers Inc., 1996. 251~262
- 10 Jean-Robert G, Louiqa R, Esther V M, Laura B. Wrapper generation for web accessible data sources. In: Proc. of the CoopIS, Los Alamitos, CA: IEEE Computer Society Press, 1998. 14~23
- 11 Yerneni R, et al. Computing Capabilities of Mediators. In: Proc. of ACM SIGMOD Conf. on Management of Data, Philadelphia PA: ACM Press, 1999. 443~454
- 12 William C. Integration of heterogeneous databases without common domains using queries based on textual similarity. In: Proc. of ACM SIGMOD Conf. on Management of Data, New York: ACM Press, 1998. 201~212
- 13 Philippe B, Anthony T. Partial answers for unavailable data sources. In: Proc. of the 1998 Workshop on Flexible Query Answering Systems (FQAS'98), Berlin: Springer, 1998. 43~54
- 14 Zachary G I, et al. An Adaptive Query Execution System for Data Integration. In: Proc. of SIGMOD Conf. New York: ACM Press, 1999. 299~310