异构数据库环境下语义集成过程的并行计算方法研究*)

强保华[1,2] 吴中福[1] 陈 凌[2] 吴开贵[1] 余建桥[2]

(重庆大学计算机科学与工程学院 重庆400044)[1] (西南农业大学信息学院 重庆400716)[2]

摘 要 区分相同属性是异构数据库环境下语义集成中的一个重要环节,主要的方法是用特征描述属性来评估属性之间的相似性。虽然这种方法具有较高自动化和易于实现的特点,但它将花费更多的时间来比较所有的属性且不能在语义集成中实现并行计算。本文提出了一种基于数据类型的方法来实现异构数据环境下相同属性的确定,这种方法具有在描述比较时间的同时实现语义集成的并行计算的特点。实验结果表明我们的方法能提高系统性能并且不降低查准率和查全率。

关键词 语义集成,相同(似)属性,异构库,基于数据类型的方法

The Research on Parallel Computation of Semantic Integration Process in Heterogeneous Databases

QIANG Bao-Hua WU Zhong-Fu Chen Ling WU Kai-Gui YU Jian-Qiao

(College of Computer Science and Engineering, Chongqing University, Chongqing 400044)

Abstract Identifying corresponding attributes is an important issue of semantic integration in heterogeneous databases. The main method uses the characteristics describing attributes to evaluate the similarity of attributes. Although this method has highly automated and easily realized characteristics, it will cost more time to compare all attributes, and cannot realize the parallel computation of semantic integration. Accordingly, in this paper we present a data-type-based approach to identify corresponding attributes in heterogeneous databases. Our approach has the characteristics of decreasing the comparing times as well as realizing the parallel computation of semantic integration. The experimental results show our method can improve the system performance without reducing the precision ratio and recall ratio.

Keywords Semantic integration, Corresponding attributes, Heterogeneous databases, Data-type-based approach

1 引言

在异构数据库环境下能够区分出相同和不相同的属性是实现数据库互操作的一个重要前提条件。许多参考文献讨论了如何区分异构数据库中相同属性的问题[1~8]。但这些方法不能区分出现实世界中存在的不相同(似)的数据类型描述的同一属性(后面给出数据类型之间相同(似)性的定义)。例如,

关系模式:Student (Sno, Sname, Sage, Sdept)

假如描述属性 Sno 的数据类型是整型,则用于区分属性的特征向量类似为:

(data type, length, key or not, value constraints, average, min, max)

特征向量的具体取值为:

(int, 4, key, not null, 95030, 95001, 95059)[1] (1)

假如描述属性 Sno 的数据类型是字符型,则用于区分属性的特征向量类似为:

(data type, length, key or not, value constraints, the ratio of the number of numerical characters to the total number of characters, the ratio of white-space characters to total characters, statistics on length)

特征向量的具体取值为:

(char, 5, key, not null, 1, 0, 5)[2] (2)

显然,由于用不相同(似)数据类型描述同一属性时特征向量的巨大差异性,目前通过比较描述属性的特征向量信息不能够区分出不相同(似)的数据类型描述的同一属性(如(1)和(2))。

既然利用描述属性的特征向量信息不能区分出不相同(似)的数据类型描述的同一属性,我们认为在不相同(似)的数据类型描述的属性中进行相似属性的确定是无实际意义的。所以,本文提出一种基于数据类型的方法来实现异构数据环境下相同属性的确定。该方法要求首先对各个数据库中的数据根据数据类型分类,然后在数据库之间具有相同数据类型描述的属性内部进行属性是否相同的确定。由于属性根据数据类型进行了分类,从而可以实现不同数据类型内确定相同属性过程的并行计算(在第3部分理论上分析了该方法并行计算的可行性),同时,该方法也明显地减少了语义集成过程中属性的比较次数。实验结果显示我们提出的方法能明显提高系统的运行效率,并且不降低语义集成中数据的查准率和查全率。

2 根据数据类型对属性分类

在确定数据库间相同属性的过程中,目前主要采用的方

*)本文研究得到国家自然科学基金项目(No.60073047)的资助。强保华 博士研究生,讲师,主要研究方向为异构数据库集成、信息集成、网络安全。吴中福 教授,博士生导师,主要研究方向为计算机与网络安全。陈 凌 硕士研究生,主要研究方向为分布式数据库、网络安全。吴开贵 博士后,主要研究方向为信息安全。余建桥 博士,教授,主要研究方向为人工智能、分布式数据库。

[1,2]这些数据来源于我院学生信息数据库。

法是通过比较所有属性对之间的特征向量是否相同来确定，但这些方法很难区分出不相同(似)的数据类型描述的同一属性，且浪费较多的系统时间用于在不相同(似)的数据类型描述的属性中进行属性相似性的确定。为此，我们提出一种在异构数据库环境下基于数据类型的语义集成方法，该方法通过比较相同(似)数据类型描述的属性对的相似性来确定数据库中的属性是否相同。该方法有两个显著特点：

1. 由于属性根据数据类型归类后，描述属性的特征向量趋于一致，向量的维度减少，这样属性比较时将花费较少的时间；

2. 由于相同属性的确定是在相同(似)的数据类型描述的属性内进行，因此整个语义集成过程可以在不相同(似)的数据类型之间进行并行计算。

我们把描述属性的数据类型分为三大类：数值型(如：int，float，small int，big int 等)；字符型(如 char，varchar(5)等)；稀有类型(如 data，time，timestamp 等)。

定义1 在上面的三大类划分中每一类内部的数据类型之间，称为相同(似)的数据类型。

根据我们的数据分类原则，可以把以下的模式1和模式2的属性根据数据类型进行分类(参见图1)。

模式1 employee (emp_id int(4)，fname varchar(20)，lname varchar(30)，job_id smallint(2)，hire_date datetime (8))

模式2 discounts (discounttype varchar(40)，stor_id char(4)，discount decimal(4,2))

根据数据类型对属性进行分类后，在异构数据库环境中进行语义集成时属性之间的比较次数明显减少，这样将花费较少的时间进行相同(似)属性的确定，从而能够明显地提高系统的性能。下面先进行理论上的分析。

情形1：比较所有的属性。假设 m 和 n 分别表示数据库1和数据库2中的属性个数。$Total\_times1$ 表示为了在数据库1和数据库2中确定相同属性所进行的比较次数。那么

$Total\_times1 = m * n$

情形2：根据属性的数据类型进行比较。假设在数据库1和数据库2中分别有两种数据类型，比如整型和字符型。其中在数据库1中数据类型为整型的属性的数量为 $m1$，数据类型为字符型的属性的数量为 $m2$；而在数据库2中数据类型为整型的属性的数量为 $n1$，数据类型为字符型的属性的数量为 $n2$。$Total\_times2$ 表示为了在数据库1和数据库2中确定相同的属性所进行的比较次数；$Part1\_times$ 表示在数据库1和数据库2中数据类型为整型的属性之间所进行的比较次数；$Part2\_times$ 表示在数据库1和数据库2中数据类型为字符型的属性之间所进行的比较次数。那么

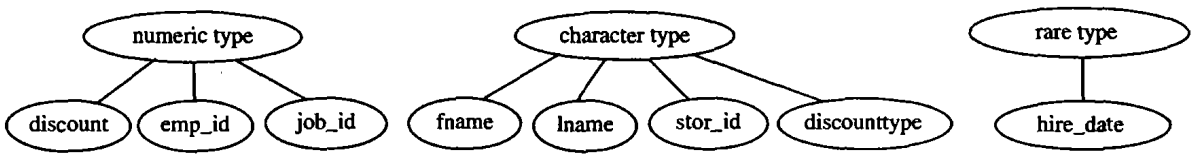

图1 根据数据类型对属性进行分类

$m = m1 + m2$；$n = n1 + n2$；

$Part1\_times = m1 * n1$；

$Part2\_times = m2 * n2$；

$Total\_times2 = Part1\_times + Part2\_times = m1 * n1 + m2 * n2$；

不妨假设 $m2 >= m1$。那么

$m2 = m1 + r$ 当 $r >= 0$

$Total\_times2 = m1 * n1 + (m1 + r) * n2 = m1 * n + r * n2$；

从情形1、2中，我们能够计算出两种情形下比较次数 $Total\_times1$ 和 $Total\_times2$ 的差值，用 $Difference$ 表示。这样

$$Difference = Total\_times1 - Total\_times2$$
$$= m2 * n - r * n2 >= m2 * n - m2 * n2$$
$$= m2 * n1.$$

当 $m2 \gg m1$，$n1 \gg n2$ 时，我们所提出的基于数据类型的方法在异构数据库环境下进行语义集成有着明显的优势。这里我们仅假设在数据库1和数据库2中有两种数据类型且存在相同的数据类型。我们的假设都是比较保守的估计，如果每个数据库中的属性根据数据类型可以分为三类，该方法会有更大的优势。

3 异构数据库环境下相同属性的确定过程及并行计算的可行性

该部分分析了本文提出的方法在理论上并行计算的可行性。

并行性指可同时执行性与自治合作性。并行计算就是利用多个物理处理部件分工合作，共同为完成某一整体任务的同时执行。并行计算具有自治性、同时性、合作性等特征。

定义2 设 $A1, \cdots, An$ 为某一程序中的诸程序段，若 $A1, \cdots, An$ 分别在 n 个不同的处理机上同时异步执行的结果等同于 $A1, \cdots, An$ 按所有可能的任意顺序执行的结果，则称 $A1, \cdots, An$ 是可并行执行的。

并行算法的最重要特征是有多个可并行执行的并行模块，算法设计最主要的工作就是并行划分，并行模块是并行划分的结果。并行划分方法主要是水平划分。

所谓水平划分，是指其划分后得到的计算模块可以同时执行的划分。常见的有以下几种水平划分：

论域分解：将问题定义域的数据结构进行划分，从而将一个问题划分成多个同样的问题。

计算分解：将一计算分解成可同时执行的多个子计算，这些子计算未必相同。

数据分解：将同一操作(程序段)的被操作数据集合进行分解。分解得到的诸并行模块，其操作(程序)相同，只是被操作的数据不同。

本文提出的基于数据类型的异构数据库集成方法，与水平划分方法中的数据分解方法一致。通过将同一数据库中的属性根据数据类型分类，分解后得到的诸并行模块，其操作(程序)相同，只是被操作的数据不同。

前面给出了并行算法的定义以及在进行并行计算前的数据分解，如果仅根据定义来判断是否可以进行并行计算难度较大。那么系统要满足什么条件才能并行执行呢？下面给出一个数据相关的判定方法，该判定方法也是实现并行计算的重

要条件。

设 A 为一程序段，I_A，O_A 分别为 A 的输入使用、输出定义变量集。

定义3(数据相关) 设 A，B 为某一程序中的两个程序段，若 $(I_A \cap O_B) \cup (I_B \cap O_A) \cup (O_A \cap O_B) \neq \phi$，则称 A 与 B 数据相关；否则，称之为数据无关

我们用数据相关性来分析本文提出的基于数据类型的方法进行异构数据库集成并行计算的可行性。

分析：令 I_A：属性的数据类型为整型的输入属性集；O_A：在整型数据类型内进行相同属性确定后的输出数据集；I_B：属性的数据类型为字符型的属性集；O_B：在字符型数据类型内进行相同属性确定后的输出数据集；A：处理整型属性的程序段；B：处理字符型属性的程序段。

根据我们数据的分类方法，相同属性的确定过程分别是在同一数据类型属性内进行的，即分别在整型属性之间和字符型属性之间进行。显然，在 I_A 与 O_B 之间、I_B 与 O_A 之间、O_A 与 O_B 之间，都不存在交叉部分。因此有如下结果：

$$(I_A \cap O_B) \cup (I_B \cap O_A) \cup (O_A \cap O_B) = \phi$$

这表明，进行异构数据集成过程中的两个程序段 A 与 B 是数据无关的。这说明本文提出的方法有较好的可并行性。

4 异构数据库环境下相同属性确定的方法

有很多方法可以用来进行异构数据环境下相同属性的确定，在引言部分我们已经提及了一些。在文[9]中，提出了一种使用属性的元数据信息进行相同属性确定的方法，该方法根据属性的元数据信息排除了大部分不相等的属性对，从而解决了文[10,11]中确定属性关系时比较费时的问题。但是，该方法仅使用属性的元数据信息，并未使用数据内容信息，使用文[9]中的规则，则具有相同元数据信息描述的不同属性不能被区分开。同时，该方法不能实现相同属性确定过程的并行计算。我们认为为了提高数据的查准率和查全率，描述属性的特征向量中应增加描述数据内容方面的信息，通过计算模式信息和数据内容与统计信息的概率值来确定相同属性，具体方法如下：

4.1 在数值型属性间进行相同属性的确定

在第2部分，所有的属性已根据数据类型进行了分类。现在的任务是根据我们前面的属性分类原则进行相同属性的确定。首先，进行数值型属性间相同属性的确定。我们把描述数值型属性特征的信息分为三类：模式信息、数据限制和数据内容。

·模式信息：包括数据类型、长度、是否为键属性；

·数据限制：包括外键信息、属性取值范围限制、是否允许为空；

·数据内容：包括最大值、最小值、平均值、标准差；

我们部分采用文[9]中的算法来确定属性间的相似度。

步骤1：相似度 $= Sim_{field\text{-}specification\&data\text{-}content} = 0$

步骤2：比较属性对之间特征向量中的每一分量，如果匹配就从表1中分配一个合适的概率值给 $Sim_{field\text{-}specification\&data\text{-}content}X$。

步骤3：$Sim_{field\text{-}specification\&data\text{-}content} = Sim_{field\text{-}specification\&data\text{-}content} + (1 - Sim_{field\text{-}specification\&data\text{-}content}) * Sim_{field\text{-}specification\&data\text{-}content}X$，重复步骤2和步骤3。

步骤4：最后对相似度结果进行规范化。

关于步骤4的规范化，主要考虑到表1中的最大的相似度不一定能达到1，我们可以设定一个阈值，比如最大相似度的80%，只要两个属性的相似度达到该阈值，我们就认为这两个属性是相同(似)的。表1中的概率值是基于我们大量实验中的统计值和对每一项参数重要性进行分析的结果。在第4部分的实验中，我们将验证这些概率值是合理的。

表1 数值型属性间相似度比较的概率值

规则		相似度 (Sim)
模式信息	相同数据类型	0.08
	相同字段长度	0.10
	均为键属性	0.06
	均不是键属性	0.03
数据限制	均参照另外的关系	0.08
	均不参照另外的关系	0.06
	均有范围限制	0.07
	均无范围限制	0.06
	均有非空值限制	0.08
	均无非空值限制	0.05
数据内容	近似的最大值[3]	0.14
	近似的最小值[4]	0.14
	近似的平均值[5]	0.10
	近似的标准差[6]	0.06

4.2 在字符型属性间进行相同属性的确定

在该部分，我们同样把描述字符型属性的信息分为三类：其中模式信息、数据限制如数值型属性，数据内容部分包括：字符型属性的具体取值中数字字符占整个字符的比率，空白字符占整个字符的比率，字符所占空间的统计长度。其中，字符所占空间的统计长度是指实际用来存储字符的长度，而不是事先分配的存储空间的长度。具体的规则及其概率值见表2。

表2 字符型和稀有型属性间相似度比较的概率值

规则		相似度 (Sim)
模式信息	相同数据类型	0.08
	相同字段长度	0.10
	均为键属性	0.06
	均不是键属性	0.03
数据限制	均参照另外的关系	0.08
	均不参照另外的关系	0.06
	均有范围限制	0.07
	均无范围限制	0.06
	均有非空值限制	0.08
	均无非空值限制	0.05
数据内容	近似的数字字符比率[7]	0.10
	近似的空白字符比率[8]	0.10
	近似的字符所占空间的统计长度[9]	0.18

4.3 在稀有型属性间进行相同属性的确定

由于稀有型属性具有类似于字符型属性的特征，我们同样使用表2中的规则来进行稀有型属性间相似度的确定。

在该部分，我们给出了属性相似度确定的规则，由于属性的相似度确定是在我们所划分的三大类数据类型内部分别进行，这从理论上保证了该过程并行计算的可行性。通过语义集

3,4,5,6代表属性对值的比率在1～3之间，7,8,9这些代表被比较属性对值的比率在1～5之间

成过程的并行计算将大大提高系统的性能，特别是在大型异构数据环境下，该方法有较大的优势。

5 实验和结论

在异构数据库环境下进行相同属性的确定主要考虑两大因素：系统性能；结果的准确度。在该部分，我们对根据数据类型来确定属性之间相似度的方法进行验证，主要通过比较基于数据类型的方法和非基于数据类型的方法语义集成过程中系统所花费的时间及数据的查准率、查全率来进行评价。

我们的实验环境为：CPU 为 P_{IV}1.7G，内存128M，操作系统为 Windows 2000 Server，数据库为 SQL Server 2000和 ACCESS 2000。这里有四组实验数据，详细信息参见表3。

表3 样本数据的详细信息

参数 / 样本数据	SQL Server 2000中的表		ACCESS 2000中的表	
	表名	属性个数	表名	属性个数
Sample 1	employees	16	employee	8
Sample 2	categories	8	jobs	4
Sample 3	Suppliers	11	ordirs	14
Sample 4	Products	10	sales	6

表4 不同方法的 Precision 与 Recall 对比值

参数 / 样本数据	使用模式和数据内容信息				仅使用模式信息	
	比较所有属性		根据数据类型比较		比较所有属性	
	Precision	Recall	Precision	Recall	Precision	Recall
Sample 1	100%	83.33%	100%	83.33%	60%	80%
Sample 2	100%	100%	100%	100%	70%	90%
Sample 3	62.5%	100%	62.5%	100%	25%	80%
Sample 4	66.67%	100%	66.67%	100%	50%	83.5%

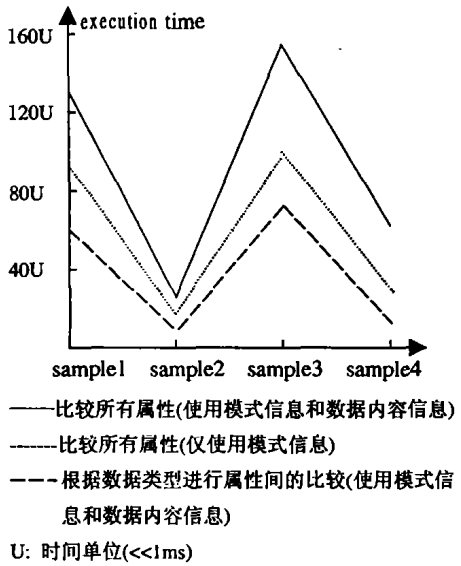

图2 不同方法的执行时间对比

在 Sample 1中，表 employees 中有16个属性，表 employee 中有8个属性。如果不根据数据类型来进行属性相似度的确定，那么比较的次数是128次。如果根据数据类型来进行属性相似度的确定，比较的次数只有66次。在 Sample 2、Sample 3 和 Sample 4中，使用基于数据类型的方法进行属性相似度确定的比较次数及比较时间也明显较少，并且保持了较高的数据查准率和查全率。这可以从图2和表4中反映出来(这里未考虑相同属性确定过程的并行计算，如果考虑并行计算，花费的时间将会更少)。

本文给出了一种在异构数据环境下确定属性间相似度的方法，该方法的一个明显的特点是易于实现语义集成过程的并行计算。实验结果显示该方法能明显地提高系统性能，并且不减小数据的查准率和查全率，同时该方法的查准率和查全率要高于文[9]中方法所得到的查准率和查全率。我们下一步的工作是在大型异构数据库环境下进行实验并且解决确定不相同(似)的数据类型描述的同一属性的问题。

参 考 文 献

1 Hayne S, Ram S. Multi-user view integration system (MUVIS): An exprt system for view integration. In: Proc. in the 6th Intl. Conf. on Data Engineering, IEEE, Feb. 1990. 402～409

2 Salton G, Yang C S, Yu C T. A theory of term importance in automatic text analysis. Journal of the American Society for Information Science, 1975, 26(1): 33～44

3 Benkley S S, Fandozzi J F, Housman E M, Woodhouse G M. Data element tool-based analysis (DELTA): [Technical Report MTR 95B0000147]. The MITRE Corporation, Bedford, MA

4 Li W-S, Clifton C, Liu S-Y. Database integration using neural networks: implementation and experiences. Knowledge and Information Systems, 2000, 2: 73～96

5 Li W-S, Clifton C. Semantic integration in heterogeneous databases using neural networks. In: Proc. of the 20th VLDB Conf. Santiago, Chile, 1994

6 Premerlani W J, Blaha M R. An approach for reverse engineering of relational databases. Communications of the ACM, 1994, 37 (5): 42～49

7 Yu C, Sun W, Dao S, Keirsey D. Determine relationships among attributes for interoperability of multi-database systems. In: Proc. of RIDE-IMS. IEEE, March, 1991

8 Sheth A, Larson J. Federated database systems for managing distributed heterogeneous, and autonomous databases. Computer Surveys, 1990, 22(3): 183～236

9 Li W-S, Clifton C. Using field specifications to determine attribute equivalence in heterogeneous databases. In: Third Intl. Workshop on Research Issues on Data Engineering: INTEROPERABILITY IN MULTIDATABASE SYSTEMS, Vienna, Austria, 1993. 174～177

10 Larson J A, Navathe S B, Elmasri R. A theory of attribute equivalence in database with application to schema integration. Transaction on Software Engineering, 1989, 15(4): 449～463

11 Sheth A, Larson J, Cornelio A, Navathe S B. A tool for integrating conceptual schemas and user views. In: Proc. of the 4th Intl. Conf. on Data Engineering, Los Angeles, CA, IEEE, 1988

12 刘键. 并行程序设计方法学. 华中理工大学出版社, 2000