

# 一种新的基于数据流的数据模型

司开君<sup>1,2</sup>, 毛宇光<sup>1,2</sup>

- (1. 南京航空航天大学 信息科学与技术学院, 江苏 南京 210016;  
2. 南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210093)

**摘要:**随着网络的发展和通讯设备的普及,一种新的数据密集型应用逐渐浮出水面,这主要包括:网络监控、电信数据管理、传感器数据监控等。在这些应用中数据采取的是多维的、连续的、快速的、随时间变化的流式数据的形式。同时,这些应用对数据的访问也是多次和连续的,并要求即时的响应,而传统的数据库技术对数据的假设和相应的查询处理技术已经无法适应这种新的应用的要求。因此,文中根据这种流式数据的特征设计了一种新的基于数据流的数据模型,并就今后如何进行数据流管理系统的研究提出一些新的看法。

**关键词:**数据流;数据模型;连续查询;滑动窗口

**中图分类号:**TP311.13

**文献标识码:**A

**文章编号:**1673-629X(2007)01-0001-03

## A New Data Model Based on Data Stream

SI Kai-jun<sup>1,2</sup>, MAO Yu-guang<sup>1,2</sup>

- (1. Coll. of Information Sci. and Techn., Nanjing Univ. of Aeronautics and Astronautics, Nanjing 210016, China;  
2. State Key Lab. for Novel Software Techn. at Nanjing Univ., Nanjing 210093, China)

**Abstract:** The development of the Internet and communication equipments have produced many new applications that include: network monitoring, telecommunication data management, sensor networks and others. In these applications, the data which called data stream is multidimensional, continuous, rapid, and changed with time. The access to data stream is also continuous and many-times, and requires on-time response. In the aspect of the assumptions of the data and the query process technology, the traditional database management systems cannot satisfy these applications. As a result, the purpose of this paper is to design a new data model for the data stream. And at last explore topics about data streams system in the future works.

**Key words:** data stream; data model; continuous query; sliding window

## 0 引言

近年来在数据库领域中,数据挖掘、联机分析处理、内存数据库等技术都得到了迅速的发展,但这些技术大都依然以传统数据库为研究基础。传统数据库存储的是静态的关系型数据记录的集合,它们具有有限的大小、可控制的操作、详细定义的结构,同时这些数据具有持久性。传统数据库中的计算具有时间复杂度和空间复杂度,其查询处理为单次查询,查询计划为静态的,且最终生成确定的查询结果。另外,传统数据库中的数据除非明确地加入了时间戳的属性,否则没有时间的概念。虽然这种模型也能充分地满足某些商业或个人的信息存储的要求,但许多当前的以及今后的应用要求能对不断快速变化的数据流提

供在线分析支持。在当今的网络监控、电信数据管理、传感器数据监控等应用出现之后,数据大都是连续的数据流,而不是有限存储的数据集合,用户需要长期的连续的查询。这些应用的出现表明了传统的数据模式并不能够完全满足应用的需要,对于数据库系统以及数据的处理算法等提出了新的挑战。目前很多大学和机构正在致力于开发一种面向新的应用的数据流管理系统,比较成型的原型系统有斯坦福大学的 STREAM 系统<sup>[1]</sup>,布朗大学、麻省理工大学等联合开发的 Aurora<sup>[2]</sup>,美国加州大学伯克利分校的 TelegraphCQ<sup>[3]</sup>等。有关流式数据库的研究也引起了国内数据库工作者的高度重视,但国内的研究还处于初期阶段,具有一定深度的研究成果还不多见。文中主要目的在于根据流式数据的特征设计一种新的基于数据流的数据模型,并就今后如何进行数据流管理系统的研究提出一些新的看法。

收稿日期:2006-04-11

基金项目:计算机软件新技术国家重点实验室(南京大学)基金课题(A2004009)

作者简介:司开君(1983-),男,安徽全椒人,硕士研究生,研究方向为数据库系统及理论;毛宇光,副教授,博士后,研究方向为数据库理论、特种数据库与多值逻辑。

## 1 数据流数据模型

### 1.1 数据流模式

数据流<sup>[4]</sup>是连续的、无限的、快速的、随时间变化的数

据元素的流。与传统的数据相比,流式数据具有许多自己的特点:它是大量的、连续的、无限的数据,流式数据变化很快,并且要求快速的、即时的响应。

每一个流和关系都有一个固定的模式,该模式由一系列属性组成。针对流元素的到达和关系的更新,预先给定一个离散的、顺序的时间域  $\Gamma$ ,任意一个时刻是时间域  $\Gamma$  中的一个值。为简单起见,可以假定  $\Gamma$  为一组非负整数  $\{0, 1, 2, \dots\}$ , 0 表示最早时刻。时间域  $\Gamma$  给出了应用的时间模型,它不必是系统时间或者时钟时间,只需将它定义为一个离散的、有顺序关系的值域即可。时间戳不是流的模式的一部分,在一个流中,一个时间戳可以标示一个或者多个元素,也可以不标示任何元素。这里只要求一个时间戳可用于标示有限但无界多个元素。

**定义 1** 流  $S$  是一个无界的  $\langle s, \tau \rangle$  的数据包(多重集),其中  $s$  是元组,  $\tau \in \Gamma$  是时间戳,它表明流  $S$  中元组  $s$  到达的逻辑时间。

**定义 2** 关系  $R$  是一种缓慢变化的流。

在此定义的关系与传统的关系不同,传统的关系没有加入时间的概念,而流式数据库系统中的关系的每个元组都加了一个时间戳,因为不管在何种应用中,总有那么一些数据是不变的或者至少在相当长的时间内是不会发生变化的,只有流数据会使系统难以处理这些数据。因为关系的许多操作,特别是更新或删除操作,在数据流的情况下是根本无法实现的。

为了解决上述问题,可以将关系看成一种特殊的流。如此定义统一了关系和流,关系变成了一种特殊的流,无需关系和流之间的转换算子,简化了查询处理。不过为了更加符合人们的使用习惯,可以在定义关系的时候不必考虑其时间戳,由系统完成这部分工作,即当有元组插入时由系统来安排时间戳。这样做既可保证传统关系操作的实现,又能保证查询语言的简化。

## 1.2 数据流连续查询语义

由于将关系和流统一,将关系看成了特殊的流,所以系统查询一般表述为:对于任一查询  $Q$ ,其输入为  $S_1, \dots, S_n$  ( $S_i$  为流),则输出为流  $A$ 。当然  $A$  可能是间歇性的流,因为系统中可以指定输出的时间间隔,之所以存在间歇性,也是由查询需求决定的,可能用户并不需要时时刻刻关心查询结果,而只需返回一定时间间隔内的综合结果,这一般和聚集函数联系在一起。若  $S_i$  都为关系或特别指明查询为一次查询,则  $A$  也为关系。

现在可从另一个角度来分析一下连续查询<sup>[5]</sup>。设  $S_1, \dots, S_n$  为输入,  $Q$  为查询,  $A$  为输出,  $S_i(\tau)$  为  $\tau$  时刻流  $S_i$  的输入,  $A(\tau)$  为  $\tau$  时刻的输出。当  $Q$  为单调时,  $Q$  只对新增的元组进行操作,然后得出结果,则  $A(\tau) = Q[S_1(\tau), S_2(\tau), \dots, S_n(\tau)]$ , 表示输出只同当前时刻的输入有关。当  $Q$  为非单调的,  $Q$  不仅对当前新增的元组,还需对过去一段时间内的元组进行操作,才能得出当前时刻的输出,则  $A(\tau) = Q[S_1(x_1), S_2(x_2), \dots, S_n(x_n)]$ , 其中  $S_i(x_i)$

表示查询  $Q$  得出  $\tau$  时刻结论需要  $S_i$  在之前  $x_i$  长的时间内的所有元组。

## 1.3 数据流操作

传统的数据库系统定义的关系代数<sup>[6]</sup>有五种基本的操作:选择、投影、笛卡儿积、差和并。除了传统意义下的这些操作对实时数据仍然需要外,还要有专门针对数据流的时间维的关系代数操作,以适应数据流数据操作的要求,下面将分别讨论这五个操作。

### 1.3.1 数据流选择操作 $\delta_C^S$

其中  $C$  为选择条件表达式,它支持传统的基于值的条件表达式,同时也支持关于时间的条件表达式。传统关系代数中的选择运算,是为选取满足针对属性指定的选择条件的数据对象,而这些被选取的数据对象的生存期不变。而数据流选择操作  $\delta_C^S$  所选取的数据对象的生存期则是有限制的,即选择同时满足生存期和数值的数据对象。

### 1.3.2 数据流投影操作 $\pi_d^S$

数据流投影操作  $\pi_d^S$  类似于传统的投影操作,即选取由  $d$  指定的属性值组成一个新的流,不过要求数据流是在其生存期上的投影。

### 1.3.3 数据流笛卡儿积操作 $\times^S$

一般笛卡儿积在这里与传统意义是一样的,不管数据对象的生存期。而数据流笛卡儿积  $\times^S$  则定义为两个数据对象在其公共生存期间的传统笛卡儿积。仅当两个数据对象的生存期的交不为空时才执行其数据流笛卡儿积。该运算可以由一个传统的笛卡儿积,再紧跟着关于其公共生存期的时间选择来实现。

### 1.3.4 数据流差操作 $-^S$

一般的差操作是针对两个数据对象具有完全同样的值和生存期进行的,而数据流差操作  $-^S$  则是针对数据对象具有同样的值,但是有不同的生存期来进行的。设有两个数据流  $R$  和  $S$ ,其数据流差  $P = R -^S S$  定义为:

$$\forall x_i \in R, x_j \in S, x_i = x_j, \text{当且仅当 } T(x_i) \cap T(x_j) \neq T(x_i) \text{ 时,此时 } T(x_i) = T(x_i) - T(x_j) \text{。而仅当 } x_i = x_j \text{ 时,且 } T(x_i) \subseteq T(x_j) \text{ 时, } x_i \notin T \text{。其中 } T(x_i) \text{ 表示数据流中数据的生存期。}$$

### 1.3.5 数据流并操作 $\cup^S$

有时需要维护有不同生存期而有同样值的数据流的完整性,或者可能要求将两个这种对象作为一个而让其生存期为两者的并。为此,引入了一种时间归并操作,表示为  $\alpha$ 。其定义为:设有数据对象集  $A$ ,  $\alpha$  施加于  $A$ ,则将  $A$  中所有同值,而不同生存期的数据对象分别并为一个,其生存期为各生存期的并。即:

$$\forall x_i \in A, \text{令 } A_i = \{x_i\} \cup \{x_j \mid x_j \in A \wedge x_j = x_i \wedge T(x_i) \neq T(x_j)\} (i = 1, 2, \dots, n), A = \bigcup_{i=1}^n A_i$$

$$\text{则 } \alpha A_i = (x_i, T(x_i) = \bigcup_{x_j \in A} T(x_j)), (i = 1, 2, \dots, n), \alpha A = \{\alpha A_i \mid i = 1, 2, \dots, n\} \text{。}$$

在我们的数据流模型中,把关系模式与流模式相统

一,把关系看成了一种特殊的流,把传统的几个关系操作移植到了数据流上。此外数据流上还有一些特有的操作,这些操作基本上与关系数据库中的操作类似,只是从关系元组扩展到了流元组,在此就不赘述了。

#### 1.4 滑动窗口模型

在大部分应用中,用户往往关心的只是最近的数据,因此需要把各种操作的范围限制在易管理的窗口内处理。通常在数据流上进行一些简单的查询操作,比如选择、投影等,显然是很简单的,但是一旦查询涉及到多个流或者聚集函数时,则在流上无法计算。相比这两种情况可以发现其要求的数据量不同。第一种情况只需获得一个元组即可运算并给出答案,也就是 1.2 节所说的单调查询,但对于第二种情况来说则是非单调的,它所要求的是数据流的全部数据,考虑到数据流的无限性,显然是不可运算的。解决的方法是用近似的一部分数据,特别是当前的一部分数据来代替全部数据用于运算,这种技术便是滑动窗口技术<sup>[7]</sup>,下面简要讨论一下滑动窗口的类型。

在数据流的持续查询系统中,滑动窗口可以看作是数据流有限部分的一个历史性快照。基于这种定义可以将滑动窗口划分为两种类型:基于时间的滑动窗口、基于流元组的滑动窗口。

##### 1.4.1 基于时间的滑动窗口

作用在流  $S$  上的一个基于时间的滑动窗口的形式为:  $S[w, v]$ ,  $w$  表示到窗口的时间周期大小,  $v$  表示窗口滑动大小,  $v$  可省略,如省略  $v$  则窗口按照系统自定义的单位时间滑动。基于时间的滑动窗口捕获了流上一段时间内的流元组作为输出,该窗口的形式表达式为:

$$S[w, v] = \{ \langle s, \tau' \rangle \mid \langle s, \tau' \rangle \in S \ \& \ \tau' \geq \tau - w \ \& \ \tau' \leq \tau \}, \text{其中 } \tau \text{ 为当前时刻。}$$

例 1:在一个股票交易所中,存在一个具备如下模式的数据流:Stock (ID, Price, Ticker, Time)。其中属性 ID 表示股票的 ID 号,Price 表示股票价格,Ticker 为股票交易者,时间域为 DATETIME 类型,该应用流元素的时间戳表示交易时的实际物理时间。

观察过去 10 小时内 ID 号为“0404401”的股票的平均交易价格,并且每 1 小时计算一次,可以用 CQL 语言表示为:

```
SELECT Avg (Price)
FROM Stock[10 hours, 1 hour][8]
WHERE ID="0404401"
```

##### 1.4.2 基于流元组的滑动窗口

作用在流  $S$  上的一个基于元组的滑动窗口形式为:  $S[n, v]$ ,  $S$  表示流的名称,  $n$  表示选取最近到达系统的  $n$  个元组,  $v$  表示窗口滑动大小,  $v$  可省略,如省略  $v$ ,则窗口以单位元组个数滑动,即如果窗口已满,则按照先进先出的原则删除最先进入窗口的元组,将后续到达的元组加入窗口。很明显,基于元组的窗口捕获了流上最近的  $n$  个元组作为输出,其形式表达式为:

$$S[n, v] = \{ \langle s, \tau' \rangle \mid \langle s, \tau' \rangle \in S \ \& \ \tau' \leq \tau \ \& \ \text{size}(\langle s, \tau' \rangle) \leq n \}, \text{其中 } \tau \text{ 为当前时刻。}$$

仍用例 1 中的流模式来说明基于流元组的滑动窗口的意义。假设每 1 小时计算过去 1000 个 stock 的元组中的平均价格:

```
SELECT Avg (Price)
FROM Stock [1000, 1 hour][8]
```

在此定义了一种滑动窗口模型,通过该模型可以利用窗口中当前的近似的一部分数据来代替全部数据,来实现在单个流或多个流上的聚集以及对于数据流的查询等一些复杂的操作。

## 2 展 望

随着科技的发展,特别是互联网技术的广泛应用,将会有越来越多的数据流数据出现在更多的应用中,因此对数据流处理的研究和应用将变得越来越迫切。然而目前对数据流数据库管理系统的研究还处于原型试验阶段,还没有形成一整套完整的方法和规范,还需要在如下方面做进一步研究:

### (1) 数据模型、查询语言和原型系统。

对于数据模型的研究,可以在特殊的应用环境下,提出新的数据操作和新的数据模型。目前提出的数据流查询语言主要有三种:以关系为基础的语言、以对象为基础的语言和过程式语言。如何扩展查询语言的表达能力以便支持数据流查询等方面有待进一步的研究。原型系统方面,虽然国外已经做出了几个成功的原型系统,但国内的研究还处于起步阶段,可以在这方面做些研究。

### (2) 查询处理及优化。

查询处理是数据流管理系统的核心,由于国内原型系统研究还处于起步阶段,所以相应的查询处理和优化的研究也不多。一般来说查询处理主要是针对查询处理器和具体的数据结构实现等方面的研究,以及优化近似查询的研究,而查询优化主要从减少内存使用和提高系统处理效率两方面来考虑。

### (3) 系统调度和负载分流。

由于数据流管理系统主要是连续查询而不是一次查询,所以查询间算子是长时间运行的,这才有了系统调度<sup>[9]</sup>的概念,调度可以是针对查询内的算子,也可针对查询间的算子,这同操作系统中进程的调度有些类似,因此,可以引入操作系统中的部分概念和算法来进行数据流管理系统中算子调度的研究。负载分流主要是指系统在超负荷的情况下,使用何种方法来减少系统负荷。

### (4) 其他方面。

对数据流管理系统的研究还包括结合传感器技术的数据流数据库系统的研究,以及数据流管理系统的接口问题、分布式数据流数据库系统、结合实际应用的数据挖掘、查询优化与减少内存使用等方面的深入研究。

(下转第 8 页)

为三类进行测试,即对煤炭供应商分为好、中、差三类,差值较小的三分之一作为第一类,差值较大的三分之一作为第三类,其余的作为第二类。然后将样本分为四类进行测试。

### 2.4 实验结果

从实验结果(见表 1)可以看到,样本分为三类时,交叉覆盖算法的分类准确率为 88.33%,而核覆盖算法分类的准确率达到 91.67%;样本分为四类时,分类准确率有所下降,分别为 71.67%和 73.33%。可以看出,当样本分为三类时,利用覆盖算法对煤炭供应商进行评测,取得了不错的效果;分为四类时,准确率有所下降。笔者认为有以下原因:首先,算法本身还有待进一步的改进;其次,对学习样本的分类是由专家根据经验进行的,当分类增多时,没有一个确定的分类标准,也一定程度上造成了测试准确率的下降。

表 1 实验结果

分类数	算法	学习样本数	测试样本数	正确分类样本数	准确率	覆盖数
三类	交叉覆盖	361	60	53	88.33%	38
	核覆盖	361	60	55	91.67%	35
四类	交叉覆盖	361	60	43	71.67%	50
	核覆盖	361	60	44	73.33%	53

从实验中还可以看出,无论样本是分为三类还是四类时,核覆盖算法的分类准确率都较交叉覆盖算法有所提高,其对算法分类准确率的提升效果还是很明显的。覆盖算法处理海量数据更为有效,当知识库中累计了更多的知

(上接第 3 页)

### 3 结束语

如今许多应用都需要处理连续的数据流,而不仅仅是传统的固定存储的数据集合。在当今的网络监控、电信数据管理、传感器数据监控等应用中,数据采取的是多维的、连续的、快速的、随时间变化的流式数据的形式,对数据的访问也是多次和连续的,并要求即时的响应。笔者根据这种流式数据的特征设计了一种新的基于数据流的数据模型,并就今后如何进行数据流管理系统的研究提出了一些新的看法。

#### 参考文献:

[1] Arasu A, Babcock B, Babu S, et al. STREAM: The Stanford Data Stream Management System[EB/OL]. 2004. <http://dbpubs.stanford.edu/pub/2004-20>.

[2] Abadi D J, Carney D. Aurora: a new model and architecture for data stream management[J]. VLDB Journal, 2003, 12(2):120-139.

[3] Chandrasekharan S, Cooper O. TelegraphCQ: Continuous data-flow processing for an uncertain world[C]// In Proc of the 1st Conf: on Innovative Data Systems Research. Asilomar, USA:

识(即学习样本更多)时,相信预测的准确率将更高。

### 3 结论

利用前向神经网络的覆盖算法及其改进算法核覆盖算法对煤炭供应商的供煤情况进行了评测,在选取数据的时候选取了煤炭供应商评测时最应综合考虑的煤质、煤价等方面数据,实验结果与统计理论中加权平均的方法进行了比较,证明取得了不错的效果。

#### 参考文献:

[1] 张 铃,张 钺. M-P 神经元模型的几何意义及其应用[J]. 软件学报,1998,9(5):334-338.

[2] 张 铃,张 钺,殷海风. 多层前向网络的交叉覆盖设计算法[J]. 软件学报,1999,10(7):737-742.

[3] Vapnik V N. Statistical Learning Theory[M]. New York: John Wiley & Sons,1998.

[4] 邓乃扬,田英杰. 数据挖掘中的新方法——支持向量机[M]. 北京:科学出版社,2004.

[5] 张 铃. 基于核函数的 SVM 机与三层前向神经网络的关系[J]. 计算机学报,2002,25(7):696-700.

[6] 吴 涛,张 铃,张燕平. 机器学习中的核覆盖算法[J]. 计算机学报,2005,28(8):1295-1301.

[7] 赵 姝,张燕平,张 媛,等. 基于交叉覆盖算法的改进算法——核平移覆盖算法[J]. 微机发展,2004,14(11):1-3.

[8] 张燕平,张 铃,段 正. 构造性核覆盖算法在图像识别中的应用[J]. 中国图像图形学报,2004,9(11):1304-1308.

[s. n. ], 2003:269-280.

[4] Babcock B, Babu S, Datar M, et al. Models and Issues in Data Streams Systems[C]//In: Popa L. Proc of the 21st ACM SIGACT - SIGMOD - SIGART Symp, on Principles of Database Systems. Madison: ACM Press, 2002:1-16.

[5] Babu S, Widom J. Continuous Queries over Data Streams[J]. SIGMOD Record, 2001,30(3):109-120.

[6] 萨师焯,王 珊. 数据库系统概论[M]. 第 3 版. 北京:高等教育出版社,2000:13-121.

[7] Datar M, Gionis A, Indyk P, et al. Maintaining Stream Statistics Over Sliding Windows[J]. SIAM Journal on Computing, 2001,31(6):1794-1813.

[8] Arasu A, Babu S, Widom J. The CQL Continuous Query Language: Semantic Foundations and Query Execution[R/OL]// Technical report, Stanford University, 2003. <http://dbpubs.stanford.edu/pub/2003-67>.

[9] Motwani R, Widom J, Arasu A. Query processing, approximation, and resource management in a data stream management system[C/OL]// In: Proc. of the 1st Biennial Conf. on Innovative Data Syst. Res (CIDR), 2003. <http://newdbpubs.stanford.edu/pub/2002-41>.