数据流的核心技术与应用发展研究综述*

杨 颖[1,2]，韩忠明[1]，杨 磊[3]

(1. 东华大学 信息学院，上海 200051；2. 广西大学 计算机与信息工程学院，广西 南宁 530004；3. 广西计算中心，广西 南宁 530022)

摘 要：在数据流基本概念的基础上，综述了数据流领域中主要的流模型、构造概要数据结构的算法、连续查询处理和优化技术等热点问题，以及数据流的应用发展。

关键词：数据流；概要数据结构；连续查询；近似处理

中图法分类号：TP274 文献标识码：A 文章编号：1001-3695(2005)11-0004-04

Survey on Key Technology and Application Development for Data Streams

YANG Ying[1,2], HAN Zhong-ming[1], YANG Lei[3]

(1. *College of Information Technology, Donghua University, Shanghai* 200051, *China*; 2. *College of Computer & Information Engineering, Guangxi University, Nanning Guangxi* 530004, *China*; 3. *Guangxi Computing Center, Nanning Guangxi* 530022, *China*)

Abstract: Based on the concept of data streams, this paper reviews the main data stream models, the algorithms for synopsis data structure, continuous query processing and optimization as well as the application development for data streams.

Key words: Data Streams; Synopsis Data Structure; Continuous Query; Approximate Processing

当前，在传感器网络、网络监控、通信数据管理、股票分析等应用领域中产生一种新型数据——数据流。这些数据流可以是关系元组、网络性能参数、电话记录、传感器读入值等。它们以大量的、连续的、随时间变化的、无法预测的无限制的流的形式到达。如何对这些流数据进行存储、查询、处理成为目前国际数据库研究领域的一个热点。

与传统数据库应用模型相比，流数据模型具有以下特点：①数据连续、实时到达；②数据量大、无限制并且无法预知；③数据一经处理，除非特意保存，否则不能被再次取出处理，即一次性处理(One-pass)，或者再次提取数据的代价昂贵。而传统数据库技术的特点是：①数据静态存储在介质中，可以被多次利用；②用户通过数据操纵语言(Data Manipulation Language, DML)来获取查询结果。如果利用传统技术处理流模型，必须将数据全部存储到介质中，然后通过提交 DML 语句访问存储介质来获取查询结果。但由于流数据量大且到达速度快，传统数据库技术难以实现实时处理。因此，数据流的特性决定了数据流的处理以自适应的、近似查询为其核心技术。

1 相关研究

数据流技术是当前国际数据库领域的一个研究热点，文献[1]着重介绍了如何构建数据流管理系统 DSMS。文献[2]介绍了流数据模型下查询和挖掘的一些算法。而目前已存在的流数据分析和管理的项目和相关研究有：

(1) 斯坦福大学的 STREAM 项目拓展了 SQL 语言在数据流上的处理功能，开发连续查询和关系的查询语言。通过特殊的窗口操作项将流数据转换为关系处理，并将结果转为数据流。其 Web 站点为 http://www.stanford.edu/stream。

(2) 伯克利大学的 TelegraphCQ 是一个连续查询处理系统，其重点在于共享查询估算和自适应查询处理。其 Web 站点为 http://telegraph.cs.Berkeley.edu。

(3) 布朗大学 Aurora 是一个应用于监控的面向工作流的系统。它允许用户通过安排盒子(操作项)和箭头(操作项间的工作流)来创建查询计划。Aurora* 和 Medusa 介绍了两种大规模分布式的流处理系统。Aurora 节点属于一个公共管理域的分布式版本。其 Web 站点为 http://www.cs.brown.edu/research/auraro。

(4) 威斯康星州立大学的 NiagaraCQ 是一个为监控动态 Web 目录而设计的执行多个连续查询系统。它将流数据划分为组，相似查询的组数据共享一个执行计划。其 Web 站点为 http://www.cs.wisc.edu/Niagara。

其他项目还包括 StatStream(实时流数据统计的监控系统)、Tribeca(Internet 实时流量监控工具)、Gigascope(分布式网络监控体系结构)等。

2 数据流模型和算法

2.1 数据流模型

根据不同的时序范围，数据流模型主要包括滑动窗口模型(Sliding Window Model)、界标模型(Landmark Model)和快照模型(Snapshot Model)三种模型。若设 n 表示当前时间戳，s, e 分别是两个已知的时间戳。滑动窗口模型描述的是数据流中最新的 W(W 也称为滑动窗口大小)个数据，其查询范围是 $\{a_{n-W+1}, \ldots, a_n\}$，随着数据的不断到达，窗口中旧数据从窗口

收稿日期：2004-11-03；修返日期：2005-03-28

基金项目：国家“863”计划资助项目(2002AA4Z3430)；广西大学科研基金资助项目(CC060012)

一端移出,新数据从窗口另一端移入。界标模型的查询范围从某一个已知的初始时间点到当前时间点为止,即$\{a_s, \ldots, a_n\}$。快照模型则将操作限制在两个预定义的时间戳之间,表示为$\{a_s, \ldots, a_e\}$。界标模型和滑动窗口模型由于要不断处理新来的数据,更接近于真实应用,因而得到更加广泛的研究。

2.2　数据流算法

数据流的特性决定了数据流算法的核心是设计单遍扫描算法(One-pass Algorithm),即在一个远小于数据规模的内存空间里不断更新一个代表数据集的结构——概要数据结构,使得在任何时候都能够根据这个结构迅速获得近似查询结果。实时地给出近似查询结果。算法的复杂性测量尺度为时间复杂性和空间复杂性,概要数据结构(Synopsis Data Structure)的规模至多应该是次线性的,即如果流的长度为 N,则概要数据结构大小不超过 O(polylog(N)),并且处理流上每一组数据的时间不超过 O(polylog(N))。生成概要数据结构的常用方法有随机抽样技术、写生技术、滑动窗口技术、直方图技术、小波技术和哈希方法等。

2.2.1　随机抽样(Random Samples)

抽样方法是从数据集中抽取小部分能代表数据集基本特征的样本,并根据该样本集合获得近似查询结果。文献[3]单遍扫描数据集,生成均匀抽样集合。令样本集合的容量为 S,在任一时刻 n,数据流中的元素都以 S/n 的概率被选取到样本集合中去。如果样本集合大小超出 S,则从中随机去除一个样本。该方法的表达效率不高。文献[4]改进了样本集合的表示方法。对于仅出现一次的元素,仍然用元素代码表示;而对于多次出现的元素,则利用结构 <value, count> 表示,Value 表示元素代码,Count 表示样本集合中该元素的数目。如上面样本集合(1,1,1,1,1,1,2,2,2,3...)可表示为(<1,6>,<2,3>,...)。节约更多的空间。在文献[5]中,当样本集合溢出时,首先将概论参数 T 提高到 T'。对于其中的任意一个元素,首先以概率 T/T' 之后以概率 $1/T$ 判断是否减去 1。一旦该计数器值降为 0,或者某一次随机判断之后计数器的值没有减小,则结束对该元素的操作。该方法能有效地获得数据集中的热门元素列表。

2.2.2　写生技术(Sketching Technique)

假设 $S=(x_1, \ldots, x_N)$ 是一个元素序列,其中 x_i 属于值域 $D=\{1, \ldots, d\}$。多样性 $m_i=|\{j \mid x_j=I\}|$ 表示序列中值 i 出现的次数。$k>0$,序列 S 的第 k 个频率因子(k 阶矩)(Frequency Moment) F_k 定义为 $F_k=\sum_{i=1}^{d} m_i^k$。频率因子获取的是序列 S 的值分布的统计。例如,F_0 可以表示元素序列中不同值的个数,F_1 可以表示元素序列的长度,F_2 可以表示自连接的大小。在文献[6]中介绍了用 Sketch 技术估算数据项不同值的数量 F_0 的方法。该方法使用线性哈希函数仅需要 O(log$d$) 内存和 O(log$d$ + log$N$) 的空间,并应用于许多数据库文献如连接大小估算,估算变量的 L_1 范式和处理多个流的复杂聚合。F_2 的计算与计算关系的自连接相似,文献[7]进一步扩展[5]估算两个不同关系连接大小的方法到估算多方向连接以应答复杂聚合,同时还提供了优化分割数据域并对每个分割独立估算的技术以便最小化整个内存需求。Indyk[8]提供了一个统一框架以计算 L_p 范式($p \in (0,2)$),充分利用了 p-stable 分布的特性,扩展了 Sketch 方法,可估算出数据集的 p 阶矩大小,其中 $0<p<2$。

2.2.3　直方图(Histogram)

直方图技术是将一个大数据集划分为多个连续的桶(Bucket),即小数据集,且每个桶都有一个特征值。主要的直方图有等宽直方图(Equi-width Histogram)、V-优化直方图(V-optimal Histogram)、端偏倚直方图(End-biased Histogram)等。

等宽直方图将值范围数据分割成近似相等的部分,使各个桶的高度(即桶所含的数据量)比较平均。文献[9]提出的算法运用两个重要的操作(拆分和合并)和两个门槛值(上限门槛和下限门槛)。V-优化直方图[10]是使各桶的方差之和最小。假设数据集中各个元素的值为 $v_1, v_2, \ldots, v_n$,b_i 表示元素 v_i 所在桶的平均值,则 $\sum(v_i - b_i)^2$ 的值最小。端偏倚直方图主要应用于冰山查询,即维护数据项属性的简单聚合(计数)并查询其聚合值是否超过指定的阈值。这种冰山查询出现在许多应用中如数据挖掘、数据仓库、信息获取、销售分析、复制检测和聚类。Motwani 介绍了随机决策算法用于数据流的频率计数和冰山查询。其算法为自适应抽样并且维护一个不同频率项的抽样。当一个数据项存在于抽样中,它的频率是增值的,低频率的数据项周期性地被删除。算法需要 O(1/ε log(εN)) 空间,其中 N 是数据流的长度,1/ε为平均抽样率。

2.2.4　滑动窗口(Sliding Windows)

滑动窗口模型下构造概要数据结构的问题在于,当新数据不断到达,旧数据过期时,如何处理过期数据,使得查询结果保持可靠。基本窗口法是按照时间顺序划分成 k 个等宽的子窗口,每个基本窗口包含 W/k 个元素,且由一个小结构表示基本窗口的特征。如果窗口所包含的元素均已过期,则删除表征这个基本窗口的小结构。用户可以基于这些未过期的小结构得到近似查询结果。文献[12]描述了在滑动窗口上均匀抽样的样本集合。在任何时间点 n,流中的元素以概率 1/min(n, W) 被添加到样本集合中去。当元素被选择到样本集合中去时,需决定一个备选元素,以便于当这个元素过期时代替该元素。由于在数据流中不能够预测将来的数据,因此,仅从 [n+1...n+W] 中随机选取一个数作为备选元素的时间戳 t。当到达时间点 t 时,这个备选元素才最终被确定,即样本集合中的任一元素,均有一个备选元素的“链”,元素一旦过期,马上用“链”上的下一个元素来取代。

2.2.5　小波技术(Wavelets Techniques)

小波分析方法是一种通用的数字信号处理技术。类似于傅里叶变换,小波变换可将输入的模拟量,变换成一系列的小波参数,根据内存大小提取顶端的 n 个高能量参数,近似还原原始信号。小波种类很多,最常见且最简单的是哈尔小波(Haar Wavelet)。文献[12]提出了一种基于哈尔小波技术,在数据流上生成直方图的算法。该算法将整个数据集转换为一系列的小波参数,然后有选择地保留有限个高能量参数,从而近似模拟原始数据集。对于时序数据而言,只需要保存最多 log$N$ 个计数器,就能够获得任一时刻的小波参数,即如果流中元素已经排好序,则仅需 O(B + log$N$) 的存储空间,就能够获得 B 个最大的小波参数。

2.2.6　哈希方法(Hashing)

计算机领域的一个常用手段是定义一组哈希函数,将数据从一个范围映射到另一个范围中去。其中 Bloom Filter 方法是

使用一小块远小于数据集数据范围的内存空间表示数据集。假设所申请的内存大小为 m 比特位,创建 h 个相互独立的哈希函数,能将数据集均匀映射到[1..m]中去。对任何元素,利用哈希函数进行计算,得到 h 个[1..m]之间的数,并将内存空间中这 h 个对应比特位都置为1。这样,就可以通过检查一个元素经过 h 次哈希操作后,是否所有对应的比特位都被置1来判断该元素是否存在。这种判断方法可能会产生错误——虽然某元素并不存在,但是它所对应的 h 个比特位都已经被其他元素所设置了。文献[15]改进了上述方法,在每一个位置上都用计数器代替比特位,从而不仅能够判断元素是否存在,而且能够估算元素的值。

以上介绍各种构造概要数据结构的算法,根据不同的产生概要数据结构的方法而分类的近似数据流算法如表1所示。

表1 近似数据流算法及功能

方 法	功 能
随机抽样	次序统计、频率项(热门元素)
写生技术	频率因素、不同值计数、频率项、检测近排序性
直方图	频率计数、冰山查询、分位点
滑动窗口	频率项、频率计数、不同值计数
小波技术	聚合
哈希方法	不同值计数、频率项

3 数据流的热点问题

由于数据流的连续的、实时的、无限制的特性决定了数据流的查询是基于连续查询或长期查询(Long-run)。下面讨论的是连续查询的语义、连续查询处理中的热点问题。

3.1 连续查询语义

假定时间被表示为自然数集,而所有的连续查询在每个时间被重估算,$A(Q,t)$ 为时间 t 的连续查询的应答集,τ 为当前时间,而0为起始时间,则一个单调连续查询 Q 在时间 τ 的应答集为 $A(Q,\tau)=\bigcup_{t=1}^{\tau}(A(Q,t)-A(Q,t-1))\cup A(Q,0)$ 即对新到来的数据项的查询可重估算并将满足的元组追加到结果中。相反,非单调的连续查询在新数据项到来时需要全部重新计算。它的语义为 $A(Q,\tau)=\bigcup_{t=0}^{\tau}A(Q,t)$

3.2 连续查询计划

在关系数据库中,所有操作都是基于提取式的,即仅当需要时,操作项才从数据表中提取数据。相反,流操作处理的数据是由数据源推入系统。在 Stream 中使用一种方法来协调它们的关系。即使查询操作项形成队列,允许数据源把数据推入队列,操作项取回所需的数据。问题在于如何调度操作项使队列大小最小,以及在突发数据流出现时的队列延迟最小,并且还要维持 QoS 服务质量保证。在连续查询计划中另一个挑战性的问题是如何支持历史查询。设计基于磁盘的数据结构和索引来探索数据流文档模式的访问还是一个开放的问题。

3.3 多查询处理

NiagaraCQ 和 Psoup[13] 分别提出两种方法来处理多查询问题,即共享查询计划和索引查询谓词。在共享查询计划中,属于同一组的查询共享一个计划,产生组中每一查询所需的结果的并集,再运用选择操作来得到最后的结果。问题还包括当新查询加入系统时如何动态地重组、各种窗口大小的窗口连接的共享估算和复杂查询的计划共享。在索引方法中,查询谓词存于一个表中,当新元组到达时,它的属性值被抽取并查看查询表,看是否满足某个查询。数据和查询通过查询谓词表和数据表的多方式的连接,一起当作两个约减的查询处理。索引方法当前不适用于窗口聚合等查询。

3.4 自适应

一个查询计划的代价可以由三方面原因而改变,即一个操作项处理时间的改变、一个谓词选择的改变和一个数据流速率的改变。最初的自适应查询计划是中间查询的重估算。为了进一步增加自适应性,Eddies 方法[15] 取代了维护一个生硬的树结构的查询计划,而是执行每个元组的调度,并通过补偿查询计划的操作项来路由。实际上,查询计划动态地重新排序以匹配当前的系统条件。这是通过元组路由策略试图去发现哪一个操作项可选并运行快,哪个操作项就先调度来实现。然而,在结果的自适应和分别路由每一个元组的开销之间存在重要的平衡问题。

3.5 分布式查询处理

在传感器网络、Internet 流量分析和 Web 日志使用等应用中,大量数据流从远程数据源发送而来。根据应用,分布式的查询策略分为两类,即通用的和专为传感器设计的分布式查询处理。通用的分布式查询策略目的在于通过执行数据源间的计算来减少通信的代价,包括站点间的查询操作项的重排序和在本地的传感器或网络路由执行简单的查询功能(如过滤、聚合和信号压缩)[11]。例如,如果每一个远程节点预聚合它的值并将求和的值和元组数发送给一个公共操作器。公共操作器可以累积这些值并计算一个总平均值。此外,还有其他技术如挑选超节点来处理预处理值、缓存和发送更新值给公共操作器仅当新达到的数据与原先数据有很大差别。对传感器网络的分布式技术处理的是通过一个共享的传感器通道沿着路由树进行查询分发和在一个无线传感器网的结果收集。文献[11]为了延长电池的寿命和解决差的无线连接性,分布式查询的目标是减少传输的数量。例如,如果一个传感器在响应一个 MAX 查询时报告一个本地的最大值,一个相邻传感器监听该传输值时,若它的本地值比该值小则不必响应。解决差的连接性的方法是发送数据包的备份,如传感器可以将它的最大值广播给其他节点。而不仅仅是根路径的节点。但是,这个方法不适用于诸如 Sum 和 Count 这样的聚合操作。在这种情况下,传感器可以拆分本地的求和值,把部分求和值发给它相邻的每一个节点。即使一个包丢失,剩余的求和值仍可以到达根节点。

4 应用发展

4.1 传感器网络

传感器网络可用于地理位置监控、公路拥塞监控、运动物追踪、生命信号的医疗监控和制造过程的超级监控。这些应用都涉及复杂的过滤和发现数据中异常模式的被激活的报警设置。分析从多个数据源到达的大量流数据的聚合和连接。当某些传感器失败时,它的流数据的聚合需要补偿。传感器的数据挖掘可能需要访问一些历史数据,代表性的查询操作有:激活触发器。当同一地域的几个传感器报告测量值超过给定阈值;在气象图上绘制温度控制线,执行由气象监控站产生的温度流的连接,连接结果形成带有每个站的经度和纬度的静态

表,将报告同一温度的点连成线;数据流当前能量使用统计分析,必要时调整能量产生率。

4.2　网络流量分析

实时分析 Internet 流量的特殊网络(Ad hoc Network)已经被使用。与传感器网络功能相同的是:多个数据源到达的数据连接、包监控、包过滤、检测异常情况和支持历史查询。此外,它的功能还包括监控对热点 URL 的需求或发现用户消耗的最大带宽。在网络流量分析中典型的查询是:流量许可。决定源 - 目的节点对以及不同 IP 地址组、子网掩码和协议类型站点的总带宽,IP 流量是多元统计的,因此,流量数据流必须能逻辑分解以便能重构基本的 TCP/IP 会话[8]。而且,分解进入会话的流涉及临时语义问题。比较不同的源 - 目的站点对的数量:比较在 TCP 三次握手中分别包含第二第三步的逻辑流中不同的源 - 目的站点对的数量。若计数值超过一个大的极限,那么一个拒绝服务事件会产生,连接许可不被哄骗的用户承认。

4.3　金融分析

在线股票价格分析涉及发现关联、鉴别趋势、仲裁机会和预测未来值。典型的基于 Web 金融分析器允许用户作如下查询:High Volatility with Recent Volume Surge,即查询所有股票价格在 20 美元到 200 美元,其最高价和最低价之间在过去 30min 高过三个百分点的股票分布图,以及最后 5min 其震荡平均量超过 300% 的股票;NASDAQ Large Cap Grainer,即查询 NASDAQ 指数高于 200 天的平均变动,以及当天开盘起价格盈利在 2 ~10 个百分点,买卖量超过 50 亿的股票;Trading Near 52-week High on Higher Volume,即查询所有价格在 52 周高度的两个百分点,每天成交量至少一百万的股票。

4.4　事务日志分析

在线挖掘 Web 使用日志、电话记录和自动银行取款交易也符合数据流的模型。其目标在于发现有趣的客户行为模式,鉴别可疑的开销行为以便防止欺诈和预测未来数据。与其他流应用一样,它需要多个数据流的连接、复杂的过滤和统计分析。常见的查询有:查询某一服务器的最近 15min 被访问的、其速率至少高于日平均速率 40% 的所有 Web 页;如果主服务器过载,实时检测 Web 服务器日志和重新路由的用户;漫游直径,挖掘无线电话和每个客户,决定每个电话使用的不同基站的最大数量。

4.5　需求分析

下面列举的是流数据连续查询基本操作项集和在未来的新应用中可能增加的需求:Selection——所有流数据的应用都需要支持复杂的过滤;Nested Aggregation——复杂的聚合,包括嵌套聚合,用于计算数据的趋势;Multiplexing and Demultiplexing——物理流需要被分解为一系列逻辑流,相反,逻辑流需要合成物理流(类似于 Group-by Union);Frequent Item Queries——已知的有 top-k 项或阈值查询,依赖于分离点的条件;Stream Mining——模式匹配、相似性查询和预测都需要在线的流数据挖掘[7]。目前已有的流挖掘算法有计算流信号和代表性趋势、决策树、预测、k-median 聚类、最近相邻查询和回归分析;Join——支持多个数据流连接和带静态元数据的流数据的连接;Windowed Queries——上述所有查询类型的返回结果都可能被限制在一个窗口内。

5　结论

本文着重介绍了近年来国际上数据流领域的主要研究成果,综述了数据流应用的流模型、构造概要数据结构的算法、连续查询处理和优化技术等热点问题及流数据的应用发展。如何设计更优化的单遍扫描算法,以及连续查询中多个查询的处理、不同数据源的自适应的分布式流数据的查询处理(尤其在 P2P 或网格环境下)等问题是当前数据流领域研究的主流方向。

参考文献:

[1] abcock B, *et al*. Models and Issues in Data Streams[A]. Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems[C]. Madison: ACM Press, 2002. 1-16.

[2] Garofalakis M, Gehrke J, Rastogi R. Querying and Mining Data Stream: You Only Get One Look, A Tutorial[A]. Franklin MJ, Moon B, Ailamaki A. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data[C]. Madison: ACM Press, 2002. 635.

[3] Vitter JS. Random Sampling with a Reservoir[J]. ACM Trans. on Mathematical Software, 1985, 11(1): 37-57.

[4] Gibbons PB, Matias Y. New Sampling-based Summary Statistics for Improving Approximate Query Answers[A]. Haas LM, Tiwary A. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data[C]. Seattle: ACM Press, 1998. 331-342.

[5] N Alon, P Gibbons, Y Matias, *et al*. Tracking Join and Self-join Sizes in Limited Storage[C]. Proc. of the 1999 ACM Symp. on Principles of Database Systems, 1999. 10-20.

[6] N Alon, Y Matias, M Szegedy. The Space Complexity of Approximating the Frequency Moments[C]. Proc. of the Annual ACM Symp. on Theory of Computing, 1996. 20-29.

[7] P Domingos, G Hulten. Mining High-Speed Data Streams[C]. Proc of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2000. 71-80.

[8] P Indyk. Stable Distributions, Pseudorandom Generators, Embeddings and Data Stream Computation[C]. Proc. of the 2000 Annual IEEE Symp. on Foundations of Computer Science, 2000. 437-448,

[9] Gibbons PB, Matias Y, Poosala V. Fast Incremental Maintenance of Approximate Histograms[A]. Jarke M, Carey MJ, Dittrich KR, *et al*. VLDB'97, Proc. of the 23rd Int'l Conf. on Very Large Data Bases[C]. Athens: Morgan Kaufmann, 1997. 466-475.

[10] Ioannidis Y, Poosala V. Balancing Histogram Optimality and Practicality for Query Result Size Estimation[J]. SIGMOD Record, 1995, 24(2): 233-244.

[11] Babcock B, Datar M, Motwani R. Sampling from a Moving Window over Streaming Data[A]. Proc. of the 13th Annual ACM-SIAM Symp. on Discrete Algorithms[C]. San Francisco: ACM/SIAM, 2002. 633-634.

[12] C Cortes, K Fisher, *et al*. Hancock: A Language for Extracting Signatures from Data Streams[C]. Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2000. 9-17.

[13] S Chandrasekaran, *et al*. Streaming Queries over Streaming Data[C]. Proc. of the 28th Int. Conf. on Very Large Data Bases, 2002. 203-214.

[14] Cohen S, Matias Y. Spectral Bloom Filters[A]. Halevy AY, Ives ZG, Doan AH. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data[C]. San Diego: ACM Press, 2003. 241-252.

[15] S Madden, M Shah, J Hellerstein, *et al*. Continuously Adaptive Continuous Queries over Streams[C]. Proc. of ACM Int. Conf. on Management of Data, 2002. 49-60.

[16] Charikar M, Chen K, *et al*. Finding Frequent Items in Data Streams[J]. Theoretical Computer Science, 2004, 312(1): 3-15.

作者简介:

杨颖(1969-),女,博士研究生,主要研究领域为数据库、数据仓库、软件工程;韩忠明(1972-),男,博士研究生,主要研究领域为数据库、数据仓库、软件工程;杨磊(1966-),男,研究员,主要研究领域为数据库技术、网络技术等。