Research on Sequence Query Processing Techniques over Data Streams

Edgar Chia-Han Lin
Dept. of Information Communication, Asia University, Taichung
Email: edgar.chlin@gmail.com

Abstract: Due to the great progress of computer technology and mature development of network, more and more data are generated and distributed through the network, which is called data streams. During the last couple of years, a number of researchers have paid their attention to data stream management, which is different from the conventional database management. At present, the new type of data management system, called data stream management system (DSMS), has become one of the most popular research areas in data engineering field. Lots of research projects have made great progress in this area. Since the current DSMS does not support queries on sequence data, this project will study the issues related to two types of data. First, we will focus on the content filtering on single-attribute streams, such as sensor data. Second, we will focus on multi-attribute streams, such as video films. We will discuss the related issues such as how to build an efficient index for all queries of different streams and the corresponding query processing mechanisms.

Keywords: data stream management system; multi-attribute stream; index structure; query processing

数据串流上序列查询处理技术之研究

林佳汉
亚洲大学信息传播学系，台中，41354
Email: edgar.chlin@gmail.com

摘　要：随着计算机科技的进步与因特网技术的成熟发展，越来越多的数据可以透过网络以各种不同的形式散播出去，形成一种新兴的数据型态，一般称之为数据串流。目前已有许多数据串流管理系统（DSMS）的相关研究，然而，由于目前已开发之数据串流管理系统基本上是承袭传统的数据库管理系统而来，并未完全的支持时序性数据的查询。因此，本论文将依照数据串流的特性以及应用的需求，区分成两个不同的方向来进行研究。其一是针对单一属性序列形式的数据串流（例如各式探测器所收集的信息），研究其内容筛选技术；其二则是将目标扩充到多属性序列形式的数据串流（例如视讯或是音乐数据），研究其内容筛选技术。我们将依序探讨如何在这两个研究方向之下，快速而有效率的建立与维护查询的索引结构，并且设计查询处理机制以便快速的筛选出资料串流之中满足查询的序列出来。

关键词：数据串流管理系统; 多属性数据串流; 索引结构; 查询处理

1. 前言

由于计算机科技的蓬勃发展再加上因特网的成熟，越来越多的多媒体数据透过因特网散播出去。在众多的多媒体数据之中，时序性数据像是音乐、动画或是视讯数据包含了相当丰富的信息内容，而从过去的研究中可以发现，多媒体数据可以用许多不同的特征数据来表示。对于一些时序性的多媒体数据而言，其特征值将会随着时间产生不同的变化。因此，多媒体数据的内涵信息将可以利用特征值变化产生的序列来表示。传统的内涵式多媒体查询系统主要是针对记录在数据库当中的多媒体数据（例如音乐或视讯数据）分析其内涵信息并建立索引，提供使用者查询之用。

然而，随着网络传输技术的成熟发展，越来越多

的各式数据会透过网络随着时间快速的以串流（stream）的形式产生或散播出去。要如何从网络上各个不同的多媒体数据来源快速的过滤与寻找使用者期望的多媒体数据是一个新兴而重要的研究课题。

目前已开发之数据串流管理系统（DSMS）基本上是承袭传统的数据库管理系统而来，因此并未完全的支持时序性数据的查询，例如在视讯数据（包括视讯影片、监视录像带、计算机动画等）播放时搜寻是否有满足使用者心中所期待的视讯内容出现或是在音乐电台频道播放音乐时搜寻是否出现某些特定的音乐片段；这些应用都必须要能够进行序列数据的比对与查询处理，同时更需要能够提供容许些为误差的近似比对（approximate matching）功能。

在过去的研究之中，针对序列数据库（sequence database）已经提出许多不同类型的索引方法以及查询处理的技术，藉以提供快速而有效率的查询比对能力。除了找到完全相符的序列片段之外，另外有些索引方法可以配合相似度衡量的方法（similarity measurement）来找到与查询相近的近似答案。然而，过去在序列数据库查询的研究多半是着重在静态数据库的查询，只要对于存放在数据库中的数据建立索引结构，再配合查询数据从索引结构之中找到相符的数据即可。在 DSMS 之中，由于数据是不断的流入系统之中，其终止的时间并不确定，甚至没有终止结束的时候，在这样的环境下，如果要利用传统的索引方式为流入 DSMS 中的所有序列数据均建立索引结构，不管是储存空间或是建立索引的时间都会对 DSMS 造成相当沉重的负担；对于使用者的查询也难以快速的响应。对于 DSMS 而言，使用者注册查询之后，系统会对于输入的数据不断的检查是否符合使用者的查询。基于这样的情况，我们可以从另外的一个角度来处理这个问题。我们可以为使用者的查询建立索引结构，并且将不断流入系统的序列数据串流当做是「查询」，然后透过索引结构来看看目前接收到的串流数据片段是否满足某些使用者的查询。这样的处理方式可以视为对于序列数据串流进行内容筛选（content filtering）的动作。过去对于数据筛选的应用多半考虑数据中的单一字段，并未考虑时序性的筛选条件，此外也没有考虑如何因应快速流入的串流数据。因此，我们必须要重新的定义要处理的问题，并且提出一套合适地相似度衡量、索引结构以及查询处理机制。

2. 文献探讨

数据串流的管理是目前数据工程领域中一项新兴的研究主题，目前也有着相当多的研究成果发表。然而，目前相关的研究多着重于传统数据库应用与数据串流的整合，例如：连续型查询处理相关之研究[11][12][14][17]、查询与数据串流之监控[16][19][20]；这些研究并未深入探讨时序性数据例如序列数据或是多媒体数据的处理。然而，随着串流应用的普及，有许多的序列数据与多媒体数据都以串流的形式出现。因此，我们必须进一步研究在资料串流上序列数据查询处理的技术。

由于序列数据之中包含了大量的信息，为了要加速查询处理，可以将序列数据简化成符号序列，亦即相近的序列值以相同的符号来代表。因此，传统的字符串比对技术将可以应用于串流序列之查询处理上。

而从过去多媒体数据库的研究当中亦可发现，有许多不同的特征可以用来表示数据的内涵。因此，内涵式视讯数据的撷取问题也可以转换成为一个字符串比对的问题。

在过去的研究中已开发出许多不同的字串比对算法，其中，KMP 算法[3]与Boyer-Moor 算法[1]是兩个广为人知的精确字串比对算法。然而，要比对一个查询字串，所有储存在资料库当中的字串必须要一一的撷取出來与查询字串作比较，此等比对过程效率明显不佳。字尾树（suffix tree） [13][18]是一个针对部分字串比对问题所提出來的索引结构。然而，建立字尾树索引的过程是十分费时的。此外，字尾树索引所占的记忆空间也较庞大。1D-List结构[10]是针对音樂资料建立索引的一个链接串列结构，记忆空间的需求也较少。除了精确字串比对之外，由于多媒体资料的特性，近似比对也就成为了一个重要的考量。在过去的研究之中，通常以兩个字串之间的编辑距離（editing distance）作为兩个字串之间的相似程度[15]。

然而，在大部分的应用之中，使用者往往会同时考虑多种不同的属性，因此，要如何针对多属性字串来进行比对是一个值得研究的问题，在过去的研究中曾经考虑若干种不同的方式來解决之。不过，这些方法都是着重在属性數目较少的情况之下。[2][5][9]均以单一属性字串的角度出发，将不同的属性信息整合到单一的结构上，当考虑的属性數目较少时，这几种方法都可以有效的解决多属性字串比对的问题，然而，当属性數目增加时，结构的复杂度将会大幅增加。

[3][4][6]提出了几种多维字符串比对的方法上述的这几种方法在比对的过程之中，同时考虑了所有的属性。然而，由于使用者查询可能不会包含所有的属性，举例而言，使用者的查询可能只包含了移动对象的速度与方向，并没有指定加速度的变化。这几种方法并无法处理这种類型的查询。

我们过去曾经提出一个以字尾树为基础的查询比对算法來解决Q属性字串比对的问题[7]。此外，为了要解决近似比对的问题，我们以编辑距離为基础定义了多属性符号相似度以及多属性字串相似度的计算方式，并且修改索引结构，同时考虑N种属性之间的关

978-1-935068-15-0 © 2010 SciRes.

聯性。以单一的索引结构，配合查询处理的机制与加速技术可以有效率的解决近似查询的问题[8]。

有鉴于资料串流的环境与传统资料库的基本差异，前述的索引结构与比对技术将无法直接套用到资料串流上有关筛选与查询比对的问题，因此，在本篇论文中中，我们分别探讨单属性序列与多属性序列之内容筛选技术，开发出新的索引结构，以便在串流式的环境之下有效率的解决精确与近似比对的问题。

3. 序列资料串流之查询处理技术

3.1 单一属性数据串流之内容筛选

单一属性的数据串流可能来自于各式不同的数据来源，例如对于气候的侦测、交通流量、网络流量的侦测等。藉由对串流资料的监控，可以发现一些异常的变化情况，进而提供实时的因应之道。串流式序列数据比对的问题跟传统序列数据库查询比对问题最大的差异在于数据以及查询的形式。传统的序列数据库查询技术，必须要针对数据库当中所有的序列数据建立索引结构。对于每一个查询而言只需要从索引之中找到符合的数据字符串即可。而对于串流式的序列数据而言，由于数据是一直不断连续的进入到系统之中，在实时性的考虑之下，并无法对完整的数据建立索引，相对的，由于数据会持续的产生出来，使用者的查询必须不断的比对目前产生出来的数据是否满足使用者需求，这也就是所谓的连续式查询（Continuous Query）。因此，为了要能快速回答目前串流内容与各查询的关系，我们必须要将所有查询建成索引并开发查询处理机制；后者会监控持续流入的资料，并于查询索引上追踪是否有该资料符合的使用者查询。为了能够快速地检验流经系统的所有资料，我们需要提供有效率的索引结构与查询处理机制，得以及时比对使用者查询与所接收字串之间的相关性。索引结构可将所有查询浓缩成一个单一结构，而查询处理机制方能达成快速比对的目标。单一属性数据串流之内容筛选主要可分成以下几个进行步骤。

3.1.1 查询字串索引之建构

有别于传统字串比对问题，数据字符串当中只须有部分字符串符合使用者查询即可；我们的查询字串必须要被完整的比对。因此，我们将以 Trie 的结构为所有查询字串建立索引，若要检验某一段资料字串满足哪些查询，只需持续追踪索引结构中具有相符查询字串的路径。

3.1.2 查询比对

当接收到一段数据串流后，系统会依其符号到索引中找寻符合的查询字串。由于资料字串的每个符号都可能对应到查询字串的某个起始符号，因此，在比对过程中，系统须持续追踪所有可能满足的查询字串。如资料字串为 abcd 时，系统须检验查询字串索引内是否存在 abcd、bcd、cd、d 等路径，因为所有以这些字串为前缀（prefix）的查询，abcd 都有可能满足。

3.1.3 字尾串接

由上述过程中我们可以发现，一旦资料字串满足某一路径时，它也一定会满足这条路径所代表字串的所有字尾（suffix），例如：满足 abcd 也必然满足 bcd、cd、d。因此，为了减少查询比对时同时追踪多条路径的成本，我们将借着字尾关系串接每条路径与其相关字尾所对应的路径。查询比对只需持续追踪一个节点，其它相关的路径均可透过字尾串接的方式来取得，藉以避免需同时追踪多条路径的负荷。

3.1.4 查询串接

然而，在上述的追踪过程中可能会有一些查询字符串会被遗漏，因为这些查询字符串位于该路径目标结点之上层，最直觉的解决方式是在透过字尾串接之后再回溯路径上所有的节点，并将节点上相符的查询回报出来。但这种做法增加了追踪的时间，因为不管上层节点是否储存查询，都必须要重新回溯一遍。为了避免这样的情况，我们采用了查询串接的形式来加速处理。也就是说，在建立字尾串接的同时，如果所串接的节点指向某个查询时，便将同时建立一个查询串接指向该查询。此时，当追踪任何一条路径时，如果路径中某个节点拥有查询串接，则该串接所连接的查询必须加以回报，因此，只要路径中的一段子字符串满足查询时，必然可以透过查询串接得到，并加以回报，因此可以避免掉必须要经常回溯的问题，并加快查询比对的速度。

3.1.5 路径结合（Path Connection）

综观前述的处理方式，我们可以发现，为了加速查询处理的速度，我们采用了字尾串接以及查询串接的方式来简化查询追踪的程序与复杂度。然而，这些额外的串接也造成了储存空间的负担，特别是当处理的查询数目增加时，整个索引结构所占的储存空间可

能会超出系统的负荷，为了要减少索引结构所占的空间，我们将利用路径结合的方式，将各条路径中部分重复的前缀与字尾加以结合，藉以减少索引结构中的节点数目进而减少数据的纪录。我们开发了一套算法来寻找最佳的路径结合策略，并设计相对应的查询处理方法，藉以正确的将符合串流数据的查询回报出来。

3.2 多属性串流之内容筛选

在不同的应用下，数据串流的来源与组成均会有所差异，以时序性的多媒体数据串流而言，其丰富的内涵信息可以藉由许多的特征来加以表示，而其中的每个特征值也会随着时间而产生出不同的变化，这样的特征值的改变历程可以反应出数据内涵的变化；为了要能够充分的掌握并且呈现出这些特征值的变化，我们可以将这些多媒体数据表示成多属性数据串流，也就是由多个属性值（attribute value）所形成的数据流。其中，每个属性会对应到一种资料特征，而每一个属性值则是代表在某一段特定的时间内某种一直保持不变的资料特征；换言之，只要有一种数据特征发生变化，亦即某一个属性值出现改变，将会有新的一笔资料出现在资料串流中，而且每笔资料都会包含相同數目的属性。因此，随着串流数据中符号的改变，多媒体数据内涵的变化将可以完整的被纪录下来。

另一方面，使用者查询也可以描述成一連串属性值的变化，这些变化同样也可以表示成多属性序列，然而，使用者在查询中多半只描述自己心中比较重视的一些属性，因此未必会拥有与资料串流相同數量的属性。一般而言，若资料以 N 种属性表示，而使用者查询包含 Q 种属性，则 Q ≤ N；如此一來，内容筛选的问题便转换成一个 N-属性序列与多个 Q-属性序列间近似比对的问题了。在资料串流环境下，多个查询（Q-属性序列）会持续地储存在系统中，并与陸续流入的资料（N-属性序列）进行比对，以辨认现阶段资料符合哪些查询；在本节中，我们针对查询序列建立索引结构，并开发一套可快速完成近似比对的方法。

在本篇论文中，我们以多媒体串流如视讯影片作为主要考虑的对象，其它具多属性序列特性的资料型态，将來亦可应用我们所研发的内容筛选技术。就多媒体串流而言，其数据特性与上述单一属性数据串流相似。因此，我们同样必须要将所有查询建成索引与查询处理机制。为了能够快速地检验流经系统的所有资料，我们同样的需要提供有效率的索引结构与查询处理机制，得以及时比对使用者查询与所接收字串之间的相关性。多属性串流之内容筛选分成以下几个进行步骤。

3.2.1 多属性查询字串索引之建构

在这个步骤当中，我们会将使用者的所下达的查询（Q-属性序列）拆解成 Q 个单一属性查询字符串，并且利用与单一属性字符串相同的方式为这些查询字符串依照属性分别建立索引结构。

3.2.2 多属性字串之查询比对

多属性字串比对与单一属性字串比对最大的差異在于符号相配（match）的定义。在单一属性字串比对的问题之中，只有在兩个符号完全相同的情况之下才称为相配；然而在多属性字串比对的问题之上，由于查询之中所包含的属性數目可能会小于资料当中的属性數目，因此，我们必须要重新定义符号的相配。我们将依据多媒体资料特性定义「符号包含」（symbol containment）关系，作为符号相配的依据。假设资料字串中包含了 N 种属性，而查询字串中包含了 Q 种属性，若一个 Q-属性符号包含于一个 N-属性符号，则 N-属性符号对应的 Q -属性值会和 Q-属性符号对应的属性值完全相同；此时，我们称这个 Q-属性符号和 N-属性符号相配。换言之，假如资料字串中对应于查询字串 Q 种属性其值相符的话，这段资料字串即视为符合该查询。基于这个定义，我们只需要修改单一属性字串的查询比对方法，利用新的符号相配定义来从单一属性索引结构之中找到相符的查询，进而再加以整合确认即可找到真正相符的 Q-属性序列查询。

3.2.3 路径压缩

由于在上述的符号相配定义之下，一个多属性符号会对应到多个单一属性符号，因此，在比对的过程当中要同时追踪多个符号与路径。在前一步骤之中，我们利用路径结合的方式来将具有相同前缀与字尾的路径相连，进而减少路径的数目。再配合字尾串接与查询串接可以只追踪单一路径即可找到所有相符的查询。然而，在一个多属性符号会对应到多个单一属性符号的情形之下，每一个符号进来系统时可能会对应到查询索引中的多个符号，因此还是会产生必须要同时追踪多条路径的问题。为了进一步加速查询比对的速度，我们开发出一套算法来将路径中所有的共通部分（common sub-path）均加以连接在一起，更进一步的精简索引结构的大小。此外，并提出相对应的查询

978-1-935068-15-0 © 2010 SciRes.

比对算法。

4. 结果与讨论

本论文之主要目的在于是针对单一属性序列形式的数据串流（single-attribute stream），研究其内容筛选技术，此外并将目标扩充到多属性序列形式的数据串流（multiple-attribute stream），并研究其内容筛选技术。

在单一属性数据串流之内容筛选方面，我们完成了索引结构之设计、查询比对算法之开发、字尾串接方法之研究、查询串接方法之开发以及路径连接方法之设计。经实验证明，我们所设计之索引结构以及查询算法将可有效的处理单一属性数据串流之内容筛选的问题。

而在多属性数据串流之内容筛选方面，我们依照预定的系统架构分析并开发合适的索引结构，并且针对其路径压缩方法加以研究探讨，最后依照我们所修正的索引结构，开发出相对应的多属性字符串查询比对算法。经实验证明，我们所设计的方法在多属性串流的环境之下的确可以达成内容筛选之目标。

因为在我们目前的索引结构开发所考虑的是提供快速而有效率的内容筛选能力，换言之，希望能够开发出有效率的查询处理技术。因此我们所开发出来的索引结构较为复杂，对于空间的需求也较大，因此未来的目标在欲持续研究各项索引结构的压缩技术，希望能利用更为精简的索引结构来提供有效率的内容过滤技术。此外，我们也将持续研究如何整合多属性的索引结构，提供更具效能的内容过滤技术。

References (参考文献)

[1] R.S. Boyer and J.S. Moore, “A Fast String Searching Algorithm,” Communications of the ACM, Vol. 20, October 1977.
[2] A.L.P. Chen, M. Chang, J. Chen, et al., “Query by Music Segments: An Efficient Approach for Song Retrieval,” IEEE Conference on Multimedia and Expo, 2000.
[3] H.V. Jagadish, N. Koudas, and D. Srivastava, “On Effective Multi-dimensional Indexing for Strings,” ACM SIGMOD Conference, pp. 403-414, 2000.
[4] T. Kahveci, A. Singh, and A. Gurel, “Similarity Searching for Multi-attribute Sequences,” International Conference on Scientific and Statistical Database Management, pp. 175-184, 2002
[5] W. Lee and A.L.P. Chen, “Efficient Multi-Feature Index Structures for Music Data Retrieval,” SPIE Conference on Storage and Retrieval for Media Databases, pp. 177-188, 2000.
[6] S.L. Lee, S.J. Chun, D.H. Kim, J.H. Lee, et al., “Similarity Search for Multidimensional Data Sequences,” IEEE Conference on Data Engineering, pp. 599-608, 2000.
[7] C.H. Lin and A.L.P. Chen, “Indexing and Matching Multiple-Attribute Strings for Efficient Multimedia Query Processing,” IEEE Transactions on Multimedia, 2005. (accepted)
[8] C.H. Lin and A.L.P. Chen, "Approximate Video Search Based on Spatio-Temporal Information of Video Objects," The First IEEE International Workshop on Multimedia Databases and Data Management 2006.
[9] C.C. Liu and A.L.P. Chen, “3D-List: A Data Structure for Efficient Video Query Processing,” IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 1, pp. 106-122, 2002.
[10] C.C. Liu, J.L. Hsu and A.L.P. Chen, “An Approximate String Matching Algorithm for Content-Based Music Data Retrieval,” IEEE Conference on Multimedia Computing and Systems, pp. 105-112, 1999.
[11] S. Madden and M.J. Franklin, “Fjording the Stream: An Architecture for Queries Over Streaming Sensor Data,” IEEE Conference on Data Engineering, 2002.
[12] S. Madden, M. Shah, J. Hellstein, and V. Raman. “Continuously Adaptive Continuous Queries Over Streams,” ACM SIGMOD Conference, pp. 49-60, 2002.
[13] E. McCreight, “A Space-Economical Suffix Tree Construction Algorithm,” Journal of Association for Computing Machinery, pp. 262-272, 1976.
[14] L. A. Moakar, T. N. Pham and P. Neophytou, “Class-based Continuous Query Scheduling for Data Streams,” 6th International Workshop on Data Management for Sensor Networks, August, 2009.
[15] G. Navarro, “A Guided Tour to Approximate String Matching,” ACM Computing Surveys, Vol. 33, No. 1, pp. 31–88, March 2001.
[16] S. Qin, S. Gu, and A. Zhou, “Detecting Bursts in Data Streams,” International APWeb Conference, 2005.
[17] U. Srivastava and J. Widom, “Memory-Limited Execution of Windowed Stream Joins,” VLDB Conference, 2004.
[18] P. Weiner, “Linear Pattern Matching Algorithms,” IEEE 14th Annual Symposium on Switching and Automata Theory, pp. 1-11, 1973.
[19] L.H. Yang, M.L. Lee, and W. Hsu, “Finding Hot Query Patterns over an XQuery Stream,” VLDB Journal Special Issue on Data Stream Processing, 2004.
[20] A. Zhou, S. Qin, and W. Qian, “Adaptively Detecting Aggregation Bursts in Data Stream,” International DASFAA Conference, 2005.