维普资讯 http://www.cqvip.com

数据流管理系统中适应性查询机制的研究*)

宋宝燕[1] 武珊珊[2] 于 戈[2]

(辽宁大学信息科学与技术学院 沈阳110036)[1] (东北大学信息科学与工程学院 沈阳110004)[2]

摘 要 介绍了数据流技术的发展现状,然后讨论了适应性查询在数据管理中的发展演变,特别是在数据流管理中的特殊性。最后,在此基础上,提出了一个支持适应性查询的数据流管理系统RealStream,并详细介绍了其适应性查询处理机制。

关键词 数据流,适应性查询,查询处理

The Study on Adaptive Query Processing Mechanism in DSMS

SONG Bao-Yan[1] WU Shan-Shan[2] YU Ge[2]

(Department of Information Science and Technology, Liaoning University, Shenyang 110036)[1]

(Department of Information Science and Engineering, Northeastern University, Shenyang 110004)[2]

Abstract This paper presents the state of art of data stream technology, and discusses the evolvement of adaptive query processing mechanism in data management, with an emphasis on the charateristics of data stream management. Based on this, RealStream, a prototype system of dada stream management system for adative query, is proposed and implemented focusing on adaptive query processing mechanism.

Keywords Data stream, Adaptive query, Query processing

1 引言

随着信息处理在通信、工业生产、经济信息处理等领域的广泛应用,数据已不仅仅拘泥于文件、数据库等传统的静态形式,一种连续、无界、不定速度的流式数据(即数据流)已经出现在越来越多的应用领域,如:网络监控、网络流量管理、入侵检测、传感器的数据处理、生产线管理、股市信息分析等等。如何对数据流实施有效管理是这些应用领域中的核心问题。对数据流管理技术的研究目前已经成为数据库领域又一新的研究热点[1~6]。在传统的数据库管理系统中,所有的数据都是以持久的数据集合形式出现的,而近年来在越来越多的应用中,数据是以流的形式在线到达的。在这些应用中,不仅原始数据是以流的形式到达,对这些数据流进行各种处理(主要是查询操作)之后得到的结果也同样是以流的形式输出的。针对数据流的特征,传统的数据处理技术(如数据库技术)已不能满足其处理要求,所以研究合适的数据流处理技术是非常必要的。

数据流的查询处理有两个特性:实时性和适应性。简单地说,实时性是指系统需在规定的时间内对查询处理任务有所响应;适应性是指系统能随着数据特性和系统属性的变化动态地调整查询执行的行为,以在满足实时性要求的前提下完成查询任务。适应性查询一直是传统数据管理中的一个热点和难点,随着数据库系统的演变、发展以及查询需求的不断提高,适应性查询的内涵也不断丰富,特别是在数据流这一新型的数据形式下,适应性查询具有更重要的涵义和作用。

本文首先介绍了数据流技术研究与开发的发展状况,然后讨论了适应性查询在数据管理中的发展演变,特别是在数据流管理中的特殊性。最后,在此基础上,给出了一个支持适应性查询的数据流管理系统RealStream,并详细介绍了其适应性查询处理机制。

2 相关工作

近年来,有关流式数据的研究已经成为当前数据库领域新的研究热点。这种流式数据一般被称作为数据流(data stream),其定义如下[2]:

*)本课题得到辽宁省自然科学基金(编号:20022027),国家"863"高技术计划GIMS主题(编号:2002AA1Z2308,2002AA118030)资助。宋宝燕 博士,教授,研究方向为数据流技术、数据库技术;武珊珊 博士研究生,研究方向为数据流技术;于 戈 教授,博士生导师,研究方向为数据库技术、数据仓库技术以及网格计算技术。

2 Clarke M, Orna Jr, Peled A. Model Checking. The Mit Press Cambridge Massachusetts London, England
3 McMillan M L. Symbolic Model Checking: An Approach to the state Explosion Problem. Kluwer Academic, 1993
4 van der Hoek W, Wooldridge M. Model Checking Knowledge and Time. In: Model Checking Software—— Proceedings of SPIN, 2002
5 Su Kaile. Model Checking Temporal Logics of Knowledge in Distributed Systems. In: The Nineteenth National Conf. on Artificial Intelligence, 2004
6 吴立军,苏开乐.多智体系统时态认知规范的模型检测算法.软件学报,2004,15(7):1012~1020
7 吴立军,苏开乐.安全协议认证的形式化方法研究.计算机工程与应用,2004,40(17):152~155
8 Paulson I. The inductive approach to verifying cryptographic protocol. Journal of Computer Security, 1998, 6(1): 85~128
9 Millen J, Rue β H. Protocol-independent secrecy. In: Proc. of the 2000 IEEE Symposium on Security and Privacy, Berkeley California, USA, 2000. 100~119
10 Thayer J, Herzog J, Guttman J. Honest ideals on strand space. In: Proc. of the 11th IEEE Computer Security Foundations Workshop, Rockport massachusets, USA, 1998. 66~78
11 胡成军,郑援,沈昌祥.密码协议的秘密性证明.计算机学报,2003,26(3): 367~372
12 Tatebayashi M, Matsuzaki, Newman D B. Key distribution protocol for digital mobile communication systems. In: Proc of Crypto'89, JNCS vol 435. Berlin: Springer Verlag, 1990. 324~333

维普资讯 http://www.cqvip.com

定义1(数据流) 一个由有先后顺序关系且个数随时间不断增加的元组构成的数据集。形式化描述为:$S:=\{\langle s,\tau\rangle \mid s$ 为满足某一关系模式的元组,$\tau\in T$ 为该元组的时间戳$\}$。其中,T 为数据流系统中的时间域。

从数据管理的角度来看,对传统数据集合的查询是一次性查询(one-time queries),即将数据集合加载到静态的数据库或文件中后,对某个时间点上的数据集进行离线的查询处理,并且每个查询仅执行一次;而对于数据流来说,数据连续不断地到达,要求对数据流的每个查询也连续不断地进行,即随着数据的不断到达而得到相应的查询结果。在数据流上的查询一般为连续查询(continuous queries)[2]。

定义2(连续查询) 只提交一次,而对不断到达的数据连续不断执行的查询。

由于数据流应用的实际需求往往需要对输入流进行实时或者近乎实时的响应,而数据流的流速是不确定的,不可避免地会出现流爆发或间歇的情况,另外流上的块操作(block-operator)(如数据流上的连接,聚集操作等)给数据流上连续查询增加了难度。可见,数据流及其上的连续查询对传统的数据管理方案提出了极大的挑战。所以,在实际应用需求和数据管理研究发展的驱动下,数据管理领域中提出了一种功能基本等同于数据库管理系统 DBMS(DataBase Management System)的数据流管理系统 DSMS(Data Stream Management System),提供对数据流的高效处理。目前,有关数据流的研究已经得到了越来越多的关注,并出现了一些初具规模的研究原型系统:

1) Brandeis,MIT 和 Brown 几所大学联合设计开发的 Aurora 系统[3]。Aurora 是一个为监控应用设计的原型系统,它适用于监控来源于物理传感器的数据、网络流量信息、GPS 设备获取的数据等等。Aurora 系统采用了工作流系统中常用的 box & arrow 模型,由用户直接从多种操作符中进行选择并组合成由操作符构成的网络模型。输入的数据流经过操作符序列的处理,将结果输出给应用程序。系统还利用 Historical Storage 结构保存历史数据以支持 ad-hoc 查询。Aurora 的特殊之处在于,它给每个输出关联一个或多个简单 QoS 信息反馈,系统运行时可以根据这些 QoS 信息对动态的资源分配提出一定的指导建议。

2) Stanford 大学设计开发的 STREAM (STanford stREam datA Manager)系统[1]。STREAM 系统是一个通用的数据流管理系统。它支持一种描述性查询语言 CQL(Continuous Query Language)[4],用户可以用 CQL 注册查询,也可以直接输入查询计划。系统对数据流按照用户的查询需求进行实时的连续查询,并为各种形式的连续查询提供连续的和实时的结果。STREAM 系统可以直接以 HTTP 方式提供 Web 接口。系统提供一个基于 Web 的图形用户接口,通过该接口远程应用程序可以不受开发平台和编程语言的限制,直接注册查询并以 HTTP 流的形式获得 XML 格式的查询结果。此外,该接口还向用户提供了一个交互式监控系统运行的途径。

3) UC Berkeley 大学设计开发的 TelegraphCQ 系统[5]。TelegraphCQ 是在开放的关系数据库管理系统 PostgreSQL 基础上开发的一个数据流管理系统。它继承了一些 UC Berkeley 以前的数据流开发成果,如适应性查询实体 Eddy,Psoup 等模块。Eddy 对操作符次序的排列主要是根据它的调度策略,元组在不同的调度策略下所经操作符的次序不尽相同而且也会有不同的优化效果。在 Eddy 使用的调度策略中,需要考虑到操作符的处理能力、处理速度和选择度等等。

4) Worcester Polytechnic Instituted 设计开发的 RainDrop 系统[6]。RainDrop 系统提供分布式环境下信息提供者和信息消费者之间的信息连接。系统特别适用于处理以 XML 数据格式发布的连续的数据流形式的股市信息监控和传感器网络数据。用户可以使用 XQuery 按照要求定制特定的分类的 XML 数据。XML 数据流入系统后结果信息会自动分布到相应的用户。

3 适应性查询与数据流管理

适应性查询不是一个新概念,它随着关系数据库系统的发展和演变,经历了不同的发展阶段。从关系数据库的查询优化到数据流中的适应性查询,适应性的粒度从整个查询计划到对数据流中的每个数据单元,适应性从人为的手动到系统的自动化等等。可见,随着数据库技术的蓬勃发展,适应性也不断丰富了其自身的内涵。

文[7]中指出,具有如下属性的系统可以称作为适应性查询处理系统:(1)系统从周围环境中获得信息;(2)系统根据获得的信息确定自身的行为;(3)这个处理过程是周而复始的进行的,构成了一个环境和系统行为之间的反馈环。其中第三点是适应性系统区别于一般做静态优化的系统的不同之处。由此可以看出适应性系统的三个特殊的属性:(1)适应性的频度,即系统获取信息和更新行为的频获取信息和更新行为的频度;(2)适应性的影响,即为提高适应性系统行为有哪些变化;(3)适应性的程度,即反馈环持续的时间。

早期关系数据库系统中所显露出的适应性概念通常是通过获取数据库中数据的分布情况来估算查询中间结果的大小,并以此估计评价操作符的执行代价。著名的 System R 系统[8]的查询优化器维护了一个系统信息库(catalog of statistic)来记录关系表的大小和各属性值的分布情况。这些统计信息是由系统管理员周期性的手动获取的,而不是明显的系统反馈信息,所以这种方式还不能算作真正的适应性查询,但是这种方法确实能够比较有效地指导查询计划的选择,并且给查询优化技术在以后的发展奠定了坚实的基础。

System R 中统计信息的获取是粗粒度的,需要系统管理员周期性获取的,这样消耗了不小的系统资源。在文[9]中提出了适应性选择度估计的策略,把统计信息的收集绑定到查询处理上。在一个查询执行中,记录中间结果的大小并以此作为优化的依据,这就提出了一个有机的信息反馈机制。这种反馈机制同时也影响了存取方法的选择以及连接顺序,连接方法的选择,并且把适应性发展到了长期的(long-term)、以每个查询为粒度的新阶段。

在 DEC(later Oracle)RDB 系统[10]中首先使用竞争(competition)方法来帮助选择查询计划。它指出仅仅需要很短的运行时间就能区分出两种访问方法的性能优劣。所以系统对一个给定的表同时执行多个访问方法,然后停止最有效地访问方法以外的所有方法。

为了提高分布式系统中查询的适应性,Query Scrambling 技术被用来处理分布式查询中不可预见的延迟,它在查询执行的过程中动态改变查询计划,如调整操作符的执行顺序以避免操作符空闲。

随着适应性粒度从整个查询细化到单个操作符,查询优化器对不同操作符选择适应性的算法。如对排序操作符考虑对可用内存量的适应[11],对连接操作中数据到达的延迟采取 Ripple Join[12]的算法等。

在对数据流这一不确定性强、实时处理要求高的新型数据形式处理中,适应性查询就更显得尤为重要。数据流系统只

维普资讯 http://www.cqvip.com

有周而复始地、迅速地从数据流中以及系统中获得查询性能相关的信息，并以此指导自身的后继行为，才能高效完成对流式数据的连续查询要求。可以看出，数据流上的连续查询完全符合了前面提到的适应性查询的性质。

比较典型的流处理系统中的适应性查询实例是 UC Berkeley 大学开发的 TelegraphCQ 系统中的适应性查询模块 Eddy[13]。Eddy 的特点在于它能够对每个元组动态地选择路由，使得查询适应性的粒度达到了元组的粒度。Eddy 的适应性技术主要体现在它对元组的路由是随着系统性能、查询特性等情况的动态变化而变化的。另外，还体现在通过将查询结果的计算与提交分割开来，以支持间歇式连接的数据流。其它数据流处理系统中也都从不同的角度采用一定的方法来实现适应性的目标。如 STREAM 系统的适应性查询只是从中央调度器的调度策略上来体现的。中央调度器采用了一种链式调度策略[14]，使系统运行时最大内存使用量最小化。从本质上来说，链式调度是从把操作符处理代价和操作符的选择度结合起来考虑的一种调度策略，使得操作符的输入队列使用的内存尽量的少。Aurora 系统中的优化技术着眼于调度器的调度策略[15]，调度器在系统运行过程中不断应用动态的两层调度策略：一是确定选择哪个查询进行调度，即操作符序列的选择；二是确定没个查询如何执行，即操作符序列的顺序确定。可见，这种调度策略在很大程度上采用了批处理的技术（操作符调度的批处理和元组调度的批处理）来适应性地降低调度代价和操作符代价。

4 RealStream 数据流管理系统

为了满足支持实时应用的数据流处理和管理的需要，在国家863计划 CIMS 主题，辽宁省自然科学基金和教育部优秀青年教师科研教学奖励计划的资助下，我们正在研制开发一个支持实时应用的数据流管理系统，因为该系统最初是为支持嵌入式实时应用而开发的，因此命名为 RealStream。

RealStream 系统提供如下功能：①支持嵌入式实时应用的需求；②支持数据流和传统数据库系统的统一操作；③支持对历史数据的查询，即即席查询(ad-hoc 查询)；④支持数据流上的两种时间戳机制——基于物理时间的物理时间戳和基于元组的逻辑时间戳；⑤ 对系统的用户提供标准的友好用户接口。具有良好的可伸缩性、可扩展性。

RealStream 系统的主要部分包括应用程序/用户接口、查询编译器、查询执行器、负载平衡器、流包装器和结果处理器。体系结构如图1所示。

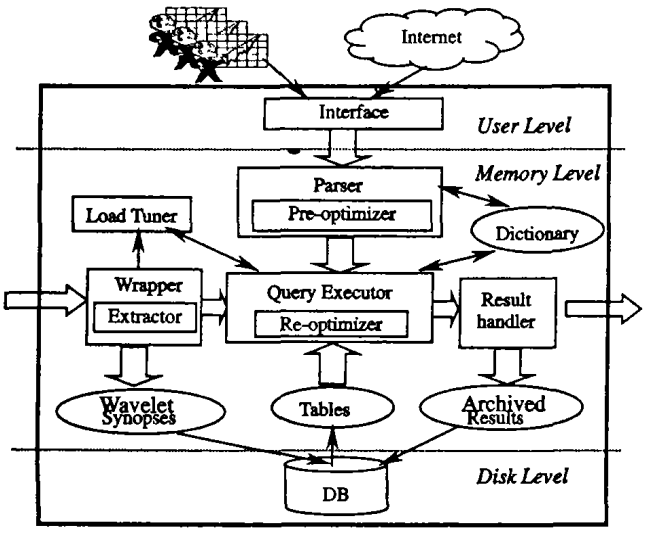

图1 RealStream 系统结构图

在 RealStream 系统中，应用程序/用户接口接收用户直接注册的或网络应用程序注册的查询，查询编译器对查询进行词法、语法分析生成初始查询计划树。在初始查询计划树上应用静态优化策略进行优化后，从树的叶子节点自底向上遍历该树生成每个元组对应的操作符序列交给查询引擎执行。在查询执行的过程中，再优化器捕捉系统性能及与查询相关的统计信息，根据适应性查询的需求动态的对当前的查询计划树进行优化，生成新的操作符队列。在查询执行的过程中，这个在优化的动作不断连续的进行着，以达到动态优化的效果。此外，负载调整器根据 QoS 要求和系统负载情况不断采取相应的措施进行负载平衡。

RealStream 系统的适应性主要体现在多重优化机制、基于触发器的动态优化策略、基于 QoS 模型的多目标优化、负载调整和元组调度粒度的适应性几个方面：

1) *多重优化* 如图2(a)所示，在传统的数据库管理系统中用户输入的查询经过语法分析器和预处理器两个查询编译模块进行语法分析和语义正确性的检查，最后由查询计划生成器在各种可选的查询计划中进行优化权衡，确定最终的查询计划。传统的查询优化器利用数据字典中的元数据，关系代数的性质和系统性能选择一个占用时间尽可能少的查询计划。这种优化是在查询执行之前的一种静态的优化。而在数据流管理系统中，由于数据流的流速，数据本身的特性，在查询执行过程中会出现流速爆发等动态特性和特殊情况，系统性能也会随着数据的特性的变化动态地波动，因此需要在查询执行过程中对查询计划进行动态的优化，以适应数据流的流速，流中数据的变化。如图2(b)所示，在数据流管理系统的查询分析器中，RealStream 系统中提出了多重优化策略：在查询执行前进行初步的预优化，之后在查询执行过程中根据变化的数据流特性和系统性能进行动态优化。

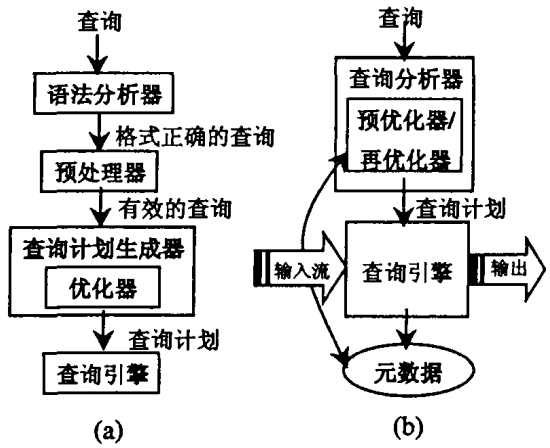

图2 查询处理器体系结构的演化

2) *基于触发器的动态优化* 如图3所示，基于触发器的动态优化器由三个主要部分构成：统计信息探测器(detector)，触发器管理器(trigger manager)和查询计划修改器(plan modifier)。其中，统计信息探测器中包含一个系统信息探测器和多个查询信息探测器。系统信息探测器负责收集系统参数，如内存的使用情况，CPU 的负载状况，缓冲区的使用情况等。显然，查询信息探测器用来收集各个查询的特性参数，如操作符的选择度变化，QoS 反馈信息，以及查询涉及到的流速的变化情况，流中数值的变化等。查询信息探测器定期更新上述统计信息并传递给触发器管理器，触发器管理器把新的统计信息和它所维护的各个触发器规则进行匹配，如果满足某个规则，即通知给查询计划修改器对当前的查询计划

维普资讯 http://www.cqvip.com

进行优化调整。由上述可见，基于触发器的动态优化方法原理简单，而且可以根据实际需求灵活地增减，更改触发器规则。而触发器规则的确定是个关键。我们正在研究不同情况下动态调整查询计划的代价和可行性。

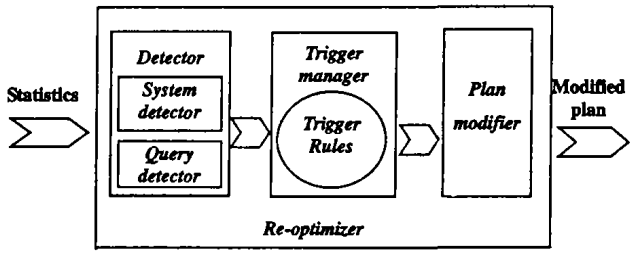

图3 基于触发器的再优化器

3）基于QoS模型的多目标优化 RealStream系统中我们把QoS定义为一个多维函数，每维参数就是一个优化目标，系统优化的最终目标就是试图使得输出结果有最大的QoS函数值。应用领域工程师会协助系统开发人员确定QoS的某些影响因素。影响QoS的因素很多，而且根据应用领域的不同会有所不同，这里我们给出几个典型的因素：

·响应时间：结果元组的输出快慢。若结果输出有延迟，则响应的QoS值会下降。

·丢弃元组：如果有元组被丢弃，则相应输出的QoS值会下降。

·结果值：QoS值还取决于结果值的重要程度。例如在某些实时监控应用中，某些输出数值要比其他的重要。通常在一个临界区附近的值就比那些远离临界区的值重要得多。所以，可以利用基于数值的QoS信息来小心的处理临界区附近的值。

4）负载调整器 对于突发的、大容量的数据流负载，系统应能够通过调整查询处理策略，适应性地满足它的实时处理要求。为了提高系统处理能力对输入数据流的流速和流量的适应性，RealStream系统由负载调整器负责解决数据流处理中的过载问题。负载调整器以多维QoS函数为前提，主要采用两种方法对系统的负载进行调整：一是Load shedding[16]技术。即在保证满足查询要求的QoS前提下，对数据进行随机的丢弃；二是采用无损的或有损的数据压缩技术，通过减少数据容量，达到减轻负载的目标。从系统结构图中可以看出，流包装器中的压缩器可以对流数据采用小波压缩的方法形成数据梗概，一方面可以满足对历史数据的查询，也可以在流爆发时减少数据容量，达到减轻负载的目标。

5）元组调度粒度的适应性 此外，在系统性能和QoS比较稳定的情况下，可以把优化的粒度放大些，对元组进行批处理形式的动态调度。即不对每个元组进行动态路由选择，而是一次对多个元组进行相同的动态路由选择。这样做可以降低元组处理代价，代价的减少主要体现在如下几个方面：

· 一次调度多个元组，对输入队列的拥塞能起到较快的缓解作用；

· 每计算一次元组路由队列，可以给多个元组共享，减少了单位元组动态优化时的计算量。

但是每次批处理多少个元组是需要确定的。简单地可以采用一个静态的批处理量。在以后的研究中，希望能够根据系统性能和QoS响应等因素，动态地决定每次批处理调度的元组数，以获得更灵活的适应性查询效果。

综上所述，RealStream系统在满足嵌入式应用的实时要求的前提下，从不同的角度出发，在系统结构、算法设计、调度策略等方面做出研究和尝试，以尽可能地提高系统的适应性。

结束语 数据流技术的研究和开发已经得到了学术界和产业界越来越多的关注，并出现了一些初具规模的数据流管理系统原型。随着计算机技术和通信技术的蓬勃发展，传感器网络，手持数码设备等嵌入式系统正在迅速普及，而数据流技术则是其中的关键技术，许多难题有待解决。我们在支持嵌入式应用的背景下设计并实现了一个适应性的数据流管理系统RealStream，从不同的角度力争增加系统的适应性，以更好地满足数据流上连续查询的实时性要求。

参考文献

1 Motwani B, Widom J, Arasu A, et al. Query Processing, Resource Management, and Approximation in a Data Stream Management System. In: CIDR Conf. Asilomar, CA, Jan. 2003. 245～256

2 Babcock B, Babu S, Datar M, et al. Models and issues in data stream systems. In: Proc. 21st ACM SIGACT-SIGMODSIGART Symp. on Principles of Database Systems, Madison, Wisconsin, May 2002. 1～16

3 Abadi D J, Carney D, et al. Aurora: a new model and architecture for data stream management. The VLDB Journal, 2003, 12(2): 120～139

4 Arasu A, Babu S, et al. CQL: A language for continuous queries over streams and relations. In: 9th Intl. Workshop on Database Programming Languages, Sept. 2003. 1～11

5 Chandrasekaran S, Deshpande A, Franklin M, et al. TelegraphCQ: Continuous dataflow processing for an uncertain world. In: Proc. of the 1st Biennial Conf. on Innovative Data Systems Research (CIDR '03), Asilomar, CA, Jan. 2003

6 Su H, Pielech B, Ding L, et al. Raindrop: A uniform query paradigm for processing xqueries on XML streams. Submitted for publication, 2003

7 Hellerstein J M, Franklin M J, Chandrasekaran S, et al. Adaptive query processing: Technology in evolution. IEEE Data Engineering Bulletin, 2000, 23(2): 7～18

8 Patricia G S, Morton M A, Donald D C, et al. Access path selection in a relational database management system. In: Philip A. Bernstein, ed. Proc. of the 1979 ACM SIGMOD Intl. Conf. on Management of Data, Boston, Massachusetts, May 30 - June 1, ACM, 1979. 23～24

9 Chung M C, Roussopoulos N. Adaptive selectivity estimation using query feedback. In: Richard T. Snodgrass and MarianneWinslett, eds. Proc. of the 1994 ACM SIGMOD Intl. Conf. on Management of Data, Minneapolis, Minnesota, May 24-27, ACM Press, 1994. 161～172

10 Antoshenkov G, Ziauddin M. Query Processing and Optimization in Oracle Rdb. VLDB Journal, 1996, 5(4): 229～237

11 Pang H H, Michael J C, Miron L. Memory-adaptive external sorting. In: Rakesh Agrawal, Se'an Baker, and David A. Bell, ed. 19th Intl. Conf. on Very Large Data Bases, Dublin, Ireland, Proc. Morgan Kaufmann, 1993. 618～629

12 Peter J H, Joseph M H. Ripple Joins for Online Aggregation. In: Proc. ACM-SIGMOD Intl. Conf. on Management of Data, Philadelphia, 1999. 287～298

13 Avnur R, Hellerstein J M. Eddies: Continuously adaptive query processing. In: Proc. of the ACM SIGMOD, Dallas, TX, May 2000. 261～272

14 Babcock B, Babu S, Datar M, Motwani R. Chain: Operator Scheduling for Memory Minimization in Data Stream Systems. In: Proc. of the ACM SIGMOD Int. Conf. on Management of Data, 2003. 253～264

15 Carney D, Cetintemel U, Rasin A, et al. Operator Scheduling in a Data Stream Manager. In: Proc. of the 29th VLDB Conf., Berlin, Germany, Setp. 2003

16 Babcock B, Datar M, et al. Load Shedding Techniques for Data Stream Systems. In Management and Processing of Data Streams, San Diego, California, USA. Jun. 2003