

STREAM DATA MANAGEMENT

ADVANCES IN DATABASE SYSTEMS

Series Editor

Ahmed K. Elmagarmid

*Purdue University
West Lafayette, IN 47907*

Other books in the Series:

FUZZY DATABASE MODELING WITH XML, Zongmin Ma, ISBN 0-387-24248-1; e-ISBN 0-387-24249-X

MINING SEQUENTIAL PATTERNS FROM LARGE DATA SETS, Wei Wang and Jiong Yang; ISBN 0-387-24246-5; e-ISBN 0-387-24247-3

ADVANCED SIGNATURE INDEXING FOR MULTIMEDIA AND WEB APPLICATIONS, Yannis Manolopoulos, Alexandros Nanopoulos, Eleni Tousidou; ISBN: 1-4020-7425-5

ADVANCES IN DIGITAL GOVERNMENT, Technology, Human Factors, and Policy, edited by William J. McIver, Jr. and Ahmed K. Elmagarmid; ISBN: 1-4020-7067-5

INFORMATION AND DATABASE QUALITY, Mario Piattini, Coral Calero and Marcela Genero; ISBN: 0-7923-7599-8

DATA QUALITY, Richard Y. Wang, Mostapha Ziad, Yang W. Lee; ISBN: 0-7923-7215-8

THE FRACTAL STRUCTURE OF DATA REFERENCE: Applications to the Memory Hierarchy, Bruce McNutt; ISBN: 0-7923-7945-4

SEMANTIC MODELS FOR MULTIMEDIA DATABASE SEARCHING AND BROWSING, Shu-Ching Chen, R.L. Kashyap, and Arif Ghaffoor; ISBN: 0-7923-7888-1

INFORMATION BROKERING ACROSS HETEROGENEOUS DIGITAL DATA: A Metadata-based Approach, Vipul Kashyap, Amit Sheth; ISBN: 0-7923-7883-0

DATA DISSEMINATION IN WIRELESS COMPUTING ENVIRONMENTS, Kian-Lee Tan and Beng Chin Ooi; ISBN: 0-7923-7866-0

MIDDLEWARE NETWORKS: Concept, Design and Deployment of Internet Infrastructure, Michah Lerner, George Vanecek, Nino Vidovic, Dad Vrsalovic; ISBN: 0-7923-7840-7

ADVANCED DATABASE INDEXING, Yannis Manolopoulos, Yannis Theodoridis, Vassilis J. Tsotras; ISBN: 0-7923-7716-8

MULTILEVEL SECURE TRANSACTION PROCESSING, Vijay Atluri, Sushil Jagodia, Binto George ISBN: 0-7923-7702-8

FUZZY LOGIC IN DATA MODELING, Guoqing Chen ISBN: 0-7923-8253-6

INTERCONNECTING HETEROGENEOUS INFORMATION SYSTEMS, Athman Bouguettaya, Boualem Benatallah, Ahmed Elmagarmid ISBN: 0-7923-8216-1

FOUNDATIONS OF KNOWLEDGE SYSTEMS: With Applications to Databases and Agents, Gerd Wagner ISBN: 0-7923-8212-9

DATABASE RECOVERY, Vijay Kumar, Sang H. Son ISBN: 0-7923-8192-0

For a complete listing of books in this series, go to <http://www.springeronline.com>

STREAM DATA MANAGEMENT

edited by

Nauman A. Chaudhry
University of New Orleans, USA

Kevin Shaw
Naval Research Lab, USA

Mahdi Abdelguerfi
University of New Orleans, USA



Springer

Nauman A. Chaudhry
University of New Orleans
USA

Kevin Shaw
Naval Research Lab
USA

Mahdi Abdelguerfi
University of New Orleans
USA

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available from the
Library of Congress.

STREAM DATA MANAGEMENT

edited by

Nauman A. Chaudhry

Kevin Shaw

Mahdi Abdelguerfi

Advances in Database Systems

Volume 30

ISBN 0-387-24393-3

e-ISBN 0-387-25229-0

Cover by Will Ladd, NRL Mapping, Charting and Geodesy Branch
utilizing NRL's GIDB® Portal System that can be utilized at
<http://dmap.nrlssc.navy.mil>

Printed on acid-free paper.

© 2005 Springer Science+Business Media, Inc.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, Inc., 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

SPIN 11054597, 11403999

springeronline.com

Contents

List of Figures	ix
List of Tables	xi
Preface	xiii
1	
Introduction to Stream Data Management	1
<i>Nauman A. Chaudhry</i>	
1. Why Stream Data Management?	1
1.1 Streaming Applications	2
1.2 Traditional Database Management Systems and Streaming Applications	3
1.3 Towards Stream Data Management Systems	4
1.4 Outline of the Rest of the Chapter	5
2. Stream Data Models and Query Languages	6
2.1 Timestamps	6
2.2 Windows	6
2.3 Proposed Stream Query Languages	7
3. Implementing Stream Query Operators	8
3.1 Query Operators and Optimization	8
3.2 Performance Measurement	8
4. Prototype Stream Data Management Systems	9
5. Tour of the Book	10
Acknowledgements	11
References	11
2	
Query Execution and Optimization	15
<i>Stratis D. Viglas</i>	
1. Introduction	15
2. Query Execution	16
2.1 Projections and Selections	17
2.2 Join Evaluation	18
3. Static Optimization	22
3.1 Rate-based Query Optimization	23
3.2 Resource Allocation and Operator Scheduling	24
3.3 Quality of Service and Load Shedding	26
4. Adaptive Evaluation	28
4.1 Query Scrambling	28
4.2 Eddies and Stems	29
5. Summary	31

References	32
3	
Filtering, Punctuation, Windows and Synopses	35
<i>David Maier, Peter A. Tucker, and Minos Garofalakis</i>	
1. Introduction: Challenges for Processing Data Streams	36
2. Stream Filtering: Volume Reduction	37
2.1 Precise Filtering	37
2.2 Data Merging	38
2.3 Data Dropping	38
2.4 Filtering with Multiple Queries	40
3. Punctuations: Handling Unbounded Behavior by Exploiting Stream Semantics	40
3.1 Punctuated Data Streams	41
3.2 Exploiting Punctuations	41
3.3 Using Punctuations in the Example Query	43
3.4 Sources of Punctuations	44
3.5 Open Issues	45
3.6 Summary	46
4. Windows: Handling Unbounded Behavior by Modifying Queries	46
5. Dealing with Disorder	47
5.1 Sources of Disorder	47
5.2 Handling Disorder	48
5.3 Summary	50
6. Synopses: Processing with Bounded Memory	50
6.1 Data-Stream Processing Model	51
6.2 Sketching Streams by Random Linear Projections: AMS Sketches	51
6.3 Sketching Streams by Hashing: FM Sketches	54
6.4 Summary	55
7. Discussion	55
Acknowledgments	56
References	56
4	
XML & Data Streams	59
<i>Nicolas Bruno, Luis Gravano, Nick Koudas, and Divesh Srivastava</i>	
1. Introduction	60
1.1 XML Databases	60
1.2 Streaming XML	61
1.3 Contributions	62
2. Models and Problem Statement	63
2.1 XML Documents	63
2.2 Query Language	64
2.3 Streaming Model	65
2.4 Problem Statement	65
3. XML Multiple Query Processing	66
3.1 Prefix Sharing	66
3.2 Y-Filter: A Navigation-Based Approach	67
3.3 Index-Filter: An Index-Based Approach	69
3.4 Summary of Experimental Results	75
4. Related Work	76
4.1 XML Databases	76
4.2 Streaming XML	77

4.3	Relational Stream Query Processing	78
5.	Conclusions	78
	References	79
5		
	CAPE: A Constraint-Aware Adaptive Stream Processing Engine	83
	<i>Elke A. Rundensteiner, Luping Ding, Yali Zhu, Timothy Sutherland and Bradford Pi- elech</i>	
1.	Introduction	83
1.1	Challenges in Streaming Data Processing	83
1.2	State-of-the-Art Stream Processing Systems	84
1.3	CAPE: Adaptivity and Constraint Exploitation	85
2.	CAPE System Overview	85
3.	Constraint-Exploiting Reactive Query Operators	87
3.1	Issues with Stream Join Algorithm	88
3.2	Constraint-Exploiting Join Algorithm	88
3.3	Optimizations Enabled by Combined Constraints	90
3.4	Adaptive Component-Based Execution Logic	91
3.5	Summary of Performance Evaluation	93
4.	Adaptive Execution Scheduling	93
4.1	State-of-the-Art Operator Scheduling	94
4.2	The ASSA Framework	94
4.3	The ASSA Strategy: Metrics, Scoring and Selection	95
4.4	Summary of Performance Evaluation	98
5.	Run-time Plan Optimization and Migration	98
5.1	Timing of Plan Re-optimization	99
5.2	Optimization Opportunities and Heuristics	99
5.3	New Issues for Dynamic Plan Migration	101
5.4	Migration Strategies in CAPE	102
6.	Self-Adjusting Plan Distribution across Machines	104
6.1	Distributed Stream Processing Architecture	104
6.2	Strategies for Query Operator Distribution	106
6.3	Static Distribution Evaluation	107
6.4	Self-Adaptive Redistribution Strategies	107
6.5	Run-Time Redistribution Evaluation	108
7.	Conclusion	109
	References	109
6		
	Time Series Queries in Data Stream Management Systems	113
	<i>Yijian Bai, Chang R. Luo, Hetal Thakkar, and Carlo Zaniolo</i>	
1.	Introduction	113
2.	The ESL-TS Language	116
2.1	Repeating Patterns and Aggregates	117
2.2	Comparison with other Languages	120
3.	ESL and User Defined Aggregates	121
4.	ESL-TS Implementation	125
5.	Optimization	127
6.	Conclusion	129
	Acknowledgments	130
	References	130

7		
Managing Distributed Geographical Data Streams with the GIDB Portal System		133
<i>John T. Sample, Frank P. McCreedy, and Michael Thomas</i>		
1. Introduction		133
2. Geographic Data Servers		134
2.1 Types of Geographic Data		134
2.2 Types of Geographic Data Servers		136
2.3 Transport Mechanisms		137
2.4 Geographic Data Standards		138
2.5 Geographic Data Streams		139
3. The Geospatial Information Database Portal System		139
3.1 GIDB Data Sources		139
3.2 GIDB Internals		140
3.3 GIDB Access Methods		142
3.4 GIDB Thematic Layer Server		144
4. Example Scenarios		147
4.1 Serving Moving Objects		147
4.2 Serving Meteorological and Oceanographic Data		149
Acknowledgements		150
References		150
8		
Streaming Data Dissemination using Peer-Peer Systems		153
<i>Shetal Shah, and Krithi Ramamritham</i>		
1. Introduction		153
2. Information-based Peer-Peer systems		154
2.1 Summary of Issues in Information-Based Peer-Peer Systems		154
2.2 Some Existing Peer-Peer Systems		156
2.3 Napster		157
2.4 Gnutella		157
2.5 Gia		157
2.6 Semantic Overlay Networks		158
2.7 Distributed Hash Tables		158
3. Multimedia Streaming Using Peer-Peer Systems		160
4. Peer-Peer Systems for Dynamic Data Dissemination		161
4.1 Overview of Data Dissemination Techniques		162
4.2 Coherence Requirement		163
4.3 A Peer-Peer Repository Framework		164
5. Conclusions		166
References		167
Index		169

List of Figures

2.1	The symmetric hash join operator for memory-fitting finite streaming sources.	19
2.2	A breakdown of the effects taking place for the evaluation of $R \bowtie_p S$ during time-unit t .	19
2.3	A traditional binary join execution tree.	22
2.4	A multiple input join operator.	22
2.5	An execution plan in the presence of queues; q_S denotes a queue for stream S .	25
2.6	Progress chart used in Chain scheduling.	25
2.7	Example utility functions; the x -axis is the percentage of dropped tuples, while the y -axis is the achieved utility.	26
2.8	A distributed query execution tree over four participating sites.	29
2.9	The decision process for query scrambling; the initiation of the scrambling phases is denoted by 'P1' for the first one and 'P2' for the second one.	29
2.10	Combination of an Eddy and four Stems in a three-way join query; solid lines indicate tuple routes, while dashed lines indicate Stem accesses used for evaluation.	31
3.1	Possible query tree for the environment sensor query.	44
3.2	Synopsis-based stream query processing architecture.	52
4.1	A fragment XML document.	64
4.2	Query model used in this chapter.	64
4.3	Using prefix sharing to represent path queries.	66
4.4	Y-Filter algorithm.	67
4.5	Compact solution representation.	68
4.6	Algorithm Index-Filter.	71
4.7	Possible scenarios in the execution of Index-Filter.	73
4.8	Materializing the positional representation of XML nodes.	74
5.1	CAPE System Architecture.	86
5.2	Heterogeneous-grained Adaptation Schema.	87

5.3	Example Query in Online Auction System.	89
5.4	Dropping Tuples Based on Constraints.	90
5.5	Adaptive Component-Based Join Execution Logic.	92
5.6	Architecture of ASSA Scheduler.	95
5.7	A Binary Join Tree and A Multi-way Join Operator.	100
5.8	Two Exchangeable Boxes.	102
5.9	Distribution Manager Architecture.	105
5.10	Distribution Table.	106
6.1	Finite State Machine for Sample Query.	125
7.1	Vector Features for Nations in North America.	134
7.2	Shaded Relief for North America.	135
7.3	Combined View From Figures 7.1 and 7.2.	135
7.4	GIDB Data Source Architecture.	141
7.5	Detailed View of GIDB Data Source Architecture.	143
7.6	GIDB Client Access Methods.	145
7.7	Diagram for First Scenario.	148
8.1	The Problem of Maintaining Coherence.	164
8.2	The Cooperative Repository Architecture.	165

List of Tables

2.1	Notation used in the extraction of cost expressions.	17
3.1	Non-trivial pass behaviors for blocking operators, based on punctuations that have arrived from the input(s).	42
3.2	Non-trivial propagation behaviors for query operators, based on punctuations that have arrived from the input(s).	42
3.3	Non-trivial keep behaviors for stateful query operators, based on punctuations that have arrived from the input(s).	43
3.4	Punctuation patterns.	43
5.1	Example Event-Listener Registry.	93
5.2	An example QoS specification.	96
7.1	Selected OGC Geographic Standards.	138
7.2	GIDB Client Software Packages.	144
7.3	ISO 19915 Standard Geographical Themes.	146
8.1	Overheads in Push and Pull.	163

Preface

In recent years, a new class of applications has emerged that requires managing data streams, i.e., data composed of continuous, real-time sequence of items. However, database management systems were originally developed to support business applications. The data in such applications is changed as a result of human-initiated transactions. Similarly data is queried as a result of human-initiated queries. The database management system acts as a passive repository for the data, executing the queries and transactions when these are submitted. However, this model of a database management system as a repository of relatively static data that is queried as a result of human interaction, does not meet the challenges posed by streaming applications.

A data stream is a possibly unbounded sequence of data items. Streaming applications have gained prominence due to both technical and business reasons. Technologically data is now available from a wide variety of monitoring devices, including sensors that are extremely cheap. Data from such devices is potentially unbounded and needs to be processed in real-time. Additionally businesses and Federal agencies now increasingly want to perform analysis on data much sooner than is possible with the current model of storing data in a data warehouse and performing the analysis off-line. Application domains requiring data stream management include military, homeland security, sensor networks, financial applications, network management, web site performance tracking, real-time credit card fraud detection, etc.

Streaming applications pose new and interesting challenges for data management systems. Such application domains require queries to be evaluated continuously as opposed to the one time evaluation of a query for traditional applications. Streaming data sets grow continuously and queries must be evaluated on such unbounded data sets. The monitoring aspect of many streaming applications requires support for reactive capabilities in real-time from data management systems. These, as well as other challenges, require a major rethink of almost all aspects of traditional database management systems to support streaming applications. Consequently, stream data management has been a very active area of research over the past few years.

The goal of this edited manuscript is to gather a coherent body of work spanning various aspects of stream data management. The manuscript comprises eight invited chapters by researchers active in stream data management. The collected chapters provide exposition of algorithms, languages, as well as systems proposed and implemented for managing streaming data. We expect this book will appeal to researchers already involved in stream data management, as well as to those starting out in this exciting and growing area.