

Sharma Chakravarthy
Qingchun Jiang

Stream Data Processing: A Quality of Service Perspective

Modeling, Scheduling, Load Shedding,
and Complex Event Processing

Contents

List of Figures	XIX
List of Tables	XXIII
List of Algorithms	XXV
1 INTRODUCTION	1
1.1 Paradigm Shift	3
1.2 Data Stream Applications	5
1.3 Book Organization	6
2 OVERVIEW OF DATA STREAM PROCESSING	9
2.1 Data Stream Characteristics	9
2.2 Data Stream Application Characteristics	10
2.3 Continuous Queries	12
2.3.1 Window Specification	14
2.3.2 Examples of Continuous Queries	16
2.3.3 QoS Metrics	18
2.4 Data Stream Management System Architecture	19
2.5 Summary of Chapter 2	20
3 DSMS CHALLENGES	23
3.1 QoS-Related Challenges	23
3.1.1 Capacity Planning and QoS Verification	23
3.1.2 Scheduling Strategies for CQs	24
3.1.3 Load Shedding and Run-Time Optimization	25
3.1.4 Complex Event and Rule Processing	26
3.1.5 Design and Implementation of a DSMS with CEP	27
3.2 Concise Overview of Book Chapters	27
3.2.1 Literature Review	27
3.2.2 Continuous Query Modeling	28

3.2.3	Scheduling Strategies for CQs	28
3.2.4	Load Shedding in a DSMS	29
3.2.5	<i>NFMⁱ</i> : A Motivating Application	30
3.2.6	DSMS and Complex Event Processing	30
3.2.7	Design and Implementation of Prototypes	31
4	LITERATURE REVIEW	33
4.1	Data Stream Management Systems	33
4.1.1	Aurora and Borealis	33
4.1.2	STREAM	34
4.1.3	<i>TelegraphCQ</i>	35
4.1.4	MavStream	36
4.1.5	Others	37
4.2	QoS-Related Issues	38
4.2.1	Continuous Query Modeling for Capacity Planning	38
4.2.2	Scheduling Strategies for CQs	39
4.2.3	Load Shedding in a DSMS	40
4.2.4	Design and Implementation of Prototypes	41
4.3	Complex Event Processing	41
4.3.1	Mid- to Late-Eighties: Active Databases	41
4.3.2	Nineties: Active Object-Oriented Databases	42
4.3.3	Beyond 2000: (Distributed) Complex Event Processing	45
4.4	Commercial and Open Source Stream and CEP Systems	47
5	MODELING CONTINUOUS QUERIES OVER DATA STREAMS	49
5.1	Continuous Query Processing	50
5.1.1	Operator Path	50
5.1.2	Operator Modeling	52
5.1.3	Scheduling and Service Discipline	53
5.2	Problem Definition	54
5.2.1	Notations and Assumptions	56
5.2.2	Stability and Performance Metrics	57
5.3	Modeling Relational Operators	57
5.3.1	Modeling <code>SELECT</code> and <code>PROJECT</code> Operators	58
5.3.2	Modeling Window-Based Symmetric Hash <code>JOIN</code>	60
5.3.3	Steady State Processing Cost	60
5.3.4	Handling Bursty Inputs and Disk-Resident Data	68
5.4	Modeling Continuous Queries	69
5.4.1	Modeling Operators with External Input(s)	71
5.4.2	Modeling Operators with Internal Input(s)	75
5.4.3	Modeling Operators with External and Internal Inputs	79
5.4.4	Scheduling Strategy and Vacation Period	79
5.4.5	Computing Memory Usage and Tuple Latency	82
5.5	Intuitive Observations	82

5.5.1	Tuple Latency	82
5.5.2	Service Discipline	82
5.5.3	Scheduling Algorithms	83
5.5.4	Choice of Query Plans	84
5.5.5	Input Rate	84
5.6	Experimental Validation	85
5.6.1	Validation of Operator Models	86
5.6.2	Validation of Continuous Query Plan Models	89
5.7	Summary of Chapter 5	93
6	SCHEDULING STRATEGIES FOR CQs	95
6.1	Scheduling Model and Terminology	96
6.1.1	Scheduling Model	97
6.1.2	Notations	99
6.2	Impact of Scheduling Strategies on QoS	103
6.3	Novel Scheduling Strategies for CQs	105
6.3.1	Path Capacity Strategy	106
6.3.2	Analysis of CQ Scheduling Strategies	108
6.3.3	Segment Strategy and Its Variants	111
6.3.4	Hybrid Threshold Scheduling Strategy	122
6.3.5	CQ Plan Characteristics	124
6.3.6	Starvation-Free Scheduling	125
6.4	Experimental Validation	126
6.4.1	Setup	126
6.4.2	Evaluation of Scheduling Strategies	127
6.5	Summary of Chapter 6	136
7	LOAD SHEDDING IN DATA STREAM MANAGEMENT SYSTEMS	137
7.1	The Load Shedding Problem	138
7.2	Integrating Load Shedders	140
7.2.1	Load Shedder as Part of a Buffer	142
7.2.2	Types of Load Shedders	143
7.3	Load Shedding Framework	143
7.3.1	Prediction of Query Processing Congestion	144
7.3.2	Placement of Load Shedders	151
7.3.3	Allocation of Load for Shedding	156
7.3.4	Load Shedding Overhead	157
7.4	Experimental Validation	158
7.4.1	Prototype Implementation	158
7.4.2	Experiment Setup	158
7.4.3	Load Shedding with Path capacity strategy	160
7.4.4	Load Shedding with EDF scheduling strategy	163
7.5	Summary of Chapter 7	165

8	<i>NFMⁱ</i>: AN INTER-DOMAIN NETWORK FAULT MANAGEMENT SYSTEM	167
8.1	Network Fault Management Problem	168
8.2	Data Processing Challenges for Fault Management	170
8.2.1	Semi-structured Text Messages	171
8.2.2	Large Number of Messages	172
8.2.3	Complex Data Processing	172
8.2.4	Online Processing and Response Time	172
8.3	Stream- and Event-Based <i>NFMⁱ</i> Architecture	173
8.3.1	Message Splitter	175
8.3.2	Message Filter and Information Extractor	175
8.3.3	Alarm Processing	178
8.4	Three-Phase Processing Model for <i>NFMⁱ</i>	178
8.4.1	Continuous Query (CQ) Processing Phase	178
8.4.2	Complex Event Processing Phase	181
8.4.3	Rule Processing Phase	182
8.4.4	Summary	183
8.5	Transactional Needs of Network Management Applications	184
8.5.1	Updates and Views	185
8.6	Summary of Chapter 8	186
9	INTEGRATING STREAM AND COMPLEX EVENT PROCESSING	187
9.1	Motivation	188
9.2	Event Processing Model	191
9.2.1	Event Detection Graphs	192
9.2.2	Event Consumption Modes	192
9.2.3	Event Detection and Rule Execution	194
9.3	Complex Event Vs. Stream Processing	195
9.3.1	Inputs and Outputs	195
9.3.2	Consumption Modes Vs. Window Types	196
9.3.3	Event Operators Vs. CQ Operators	197
9.3.4	Best-Effort Vs. QoS	197
9.3.5	Optimization and Scheduling	198
9.3.6	Buffer Management and Load Shedding	198
9.3.7	Rule Execution Semantics	199
9.3.8	Summary	199
9.4	MavEStream: An Integrated Architecture	200
9.4.1	Strengths of the Architecture	201
9.5	Stream-Side Extensions	203
9.5.1	Named Continuous Queries	203
9.5.2	Stream Modifiers	205
9.6	Event-Side Extensions	207
9.6.1	Generalization of Event Specification	207
9.6.2	Event Specification using Extended SQL	208

9.6.3	Mask Optimization	210
9.6.4	Enhanced Event Consumption Modes	210
9.6.5	Rule Processing	211
9.7	Summary of Chapter 9	213
10	MavStream: DEVELOPMENT OF A DSMS PROTOTYPE	215
10.1	MavStream Architecture	216
10.1.1	Functionality	216
10.1.2	MavStream Server Design	217
10.1.3	MavStream Server Implementation	219
10.2	Window Types	220
10.2.1	Functionality	220
10.2.2	Design	220
10.2.3	Implementation	222
10.3	Stream Operators and CQs	222
10.3.1	Functionality	222
10.3.2	Design of Operators	223
10.3.3	Implementation	225
10.4	Buffers and Archiving	229
10.4.1	Functionality	229
10.4.2	Buffer Manager Design	230
10.4.3	Buffer Manager Implementation	231
10.5	Run-time Optimizer	231
10.5.1	Functionality	231
10.5.2	Run-time Optimizer Design	232
10.5.3	Run-time Optimizer Implementation	234
10.6	QoS-Delivery Mechanisms	243
10.6.1	Functionality	243
10.6.2	Scheduler Design	243
10.6.3	Scheduler Implementation	245
10.6.4	Load Shedder Design	246
10.6.5	Load Shedder Implementation	246
10.7	System Evaluation	248
10.7.1	Single QoS Measure Violation	248
10.7.2	Multiple QoS Measures Violation	249
10.7.3	Effect of Load Shedding on QoS Measures	253
10.7.4	Effect of Load Shedding on Error in Results	257
11	INTEGRATING CEP WITH A DSMS	261
11.1	MavStream: Integration Issues	262
11.1.1	Event Generation	263
11.1.2	Continuous Event Query (CEQ) Specification	264
11.1.3	Events and Masks	264
11.1.4	Address Space Issues	265
11.1.5	Summary	265

11.2	Design of the Integrated System	266
11.2.1	Address Space	266
11.2.2	Continuous Event Queries	266
11.2.3	Events and Masks	268
11.2.4	Event Generator Interface	271
11.2.5	Need for a Common Buffer for All Events	272
11.2.6	Complex Events and Rule Management	274
11.3	Implementation Details of Integration	275
11.3.1	Input Processor	276
11.3.2	Event and Rule <i>Instantiator</i>	278
11.3.3	Event Generator Interface	278
11.4	Stream Modifiers	281
11.4.1	Tuple-Based Stream Modifiers	281
11.4.2	Window-Based Stream Modifiers	282
11.4.3	Implementation	282
11.5	Additional Benefits of CEP Integration	284
11.6	Summary of Chapter 11	285
12	CONCLUSIONS AND FUTURE DIRECTIONS	287
12.1	Looking Ahead	287
12.2	Stream Processing	288
12.2.1	Continuous Query Modeling	289
12.2.2	Scheduling	289
12.2.3	Load Shedding	290
12.3	Integration of Stream and Event Processing	291
12.4	Epilogue	293
	References	295
	Index	315