

Contents

<i>Preface</i>	<i>page</i> xiii
<i>Foreword</i>	xix
<i>Acknowledgements</i>	xxi
<i>List of acronyms</i>	xxii
Part I Fundamentals	1
1 What brought us here?	3
1.1 Overview	3
1.2 Towards continuous data processing: the requirements	3
1.3 Stream processing foundations	6
1.3.1 Data management technologies	8
1.3.2 Parallel and distributed systems	13
1.3.3 Signal processing, statistics, and data mining	16
1.3.4 Optimization theory	18
1.4 Stream processing – tying it all together	22
References	24
2 Introduction to stream processing	33
2.1 Overview	33
2.2 Stream Processing Applications	33
2.2.1 Network monitoring for cybersecurity	34
2.2.2 Transportation grid monitoring and optimization	36
2.2.3 Healthcare and patient monitoring	38
2.2.4 Discussion	40
2.3 Information flow processing technologies	40
2.3.1 Active databases	41
2.3.2 Continuous queries	42
2.3.3 Publish–subscribe systems	42
2.3.4 Complex event processing systems	43
2.3.5 ETL and SCADA systems	44
2.4 Stream Processing Systems	45
2.4.1 Data	45
2.4.2 Processing	49
2.4.3 System architecture	53

	2.4.4	Implementations	56
	2.4.5	Discussion	66
	2.5	Concluding remarks	68
	2.6	Exercises	69
		References	70
Part II	Application development		75
3	Application development – the basics		77
	3.1	Overview	77
	3.2	Characteristics of SPAs	77
	3.3	Stream processing languages	80
	3.3.1	Features of stream processing languages	80
	3.3.2	Approaches to stream processing language design	83
	3.4	Introduction to SPL	86
	3.4.1	Language origins	86
	3.4.2	A “Hello World” application in SPL	87
	3.5	Common stream processing operators	92
	3.5.1	Stream relational operators	92
	3.5.2	Utility operators	96
	3.5.3	Edge adapter operators	97
	3.6	Concluding remarks	101
	3.7	Programming exercises	101
		References	103
4	Application development – data flow programming		106
	4.1	Overview	106
	4.2	Flow composition	106
	4.2.1	Static composition	108
	4.2.2	Dynamic composition	112
	4.2.3	Nested composition	122
	4.3	Flow manipulation	128
	4.3.1	Operator state	128
	4.3.2	Selectivity and arity	131
	4.3.3	Using parameters	132
	4.3.4	Output assignments and output functions	134
	4.3.5	Punctuations	136
	4.3.6	Windowing	138
	4.4	Concluding remarks	144
	4.5	Programming exercises	144
		References	147
5	Large-scale development – modularity, extensibility, and distribution		148
	5.1	Overview	148

	Contents	ix
5.2	Modularity and extensibility	148
5.2.1	Types	149
5.2.2	Functions	151
5.2.3	Primitive operators	153
5.2.4	Composite and custom operators	161
5.3	Distributed programming	164
5.3.1	Logical versus physical flow graphs	164
5.3.2	Placement	166
5.3.3	Transport	170
5.4	Concluding remarks	172
5.5	Programming exercises	173
	References	176
6	Visualization and debugging	178
6.1	Overview	178
6.2	Visualization	178
6.2.1	Topology visualization	179
6.2.2	Metrics visualization	184
6.2.3	Status visualization	185
6.2.4	Data visualization	186
6.3	Debugging	188
6.3.1	Semantic debugging	189
6.3.2	User-defined operator debugging	194
6.3.3	Deployment debugging	194
6.3.4	Performance debugging	195
6.4	Concluding remarks	199
	References	200
Part III	System architecture	201
7	Architecture of a stream processing system	203
7.1	Overview	203
7.2	Architectural building blocks	203
7.2.1	Computational environment	204
7.2.2	Entities	204
7.2.3	Services	206
7.3	Architecture overview	207
7.3.1	Job management	207
7.3.2	Resource management	208
7.3.3	Scheduling	209
7.3.4	Monitoring	210
7.3.5	Data transport	211
7.3.6	Fault tolerance	212
7.3.7	Logging and error reporting	213

x	Contents	
	7.3.8 Security and access control	213
	7.3.9 Debugging	214
	7.3.10 Visualization	214
	7.4 Interaction with the system architecture	215
	7.5 Concluding remarks	215
	References	215
8	InfoSphere Streams architecture	218
	8.1 Overview	218
	8.2 Background and history	218
	8.3 A user's perspective	219
	8.4 Components	220
	8.4.1 Runtime instance	222
	8.4.2 Instance components	223
	8.4.3 Instance backbone	227
	8.4.4 Tooling	229
	8.5 Services	232
	8.5.1 Job management	232
	8.5.2 Resource management and monitoring	236
	8.5.3 Scheduling	239
	8.5.4 Data transport	241
	8.5.5 Fault tolerance	247
	8.5.6 Logging, tracing, and error reporting	248
	8.5.7 Security and access control	251
	8.5.8 Application development support	256
	8.5.9 Processing element	259
	8.5.10 Debugging	264
	8.5.11 Visualization	267
	8.6 Concluding remarks	268
	References	270
	Part IV Application design and analytics	273
9	Design principles and patterns for stream processing applications	275
	9.1 Overview	275
	9.2 Functional design patterns and principles	275
	9.2.1 Edge adaptation	275
	9.2.2 Flow manipulation	287
	9.2.3 Dynamic adaptation	301
	9.3 Non-functional principles and design patterns	310
	9.3.1 Application design and composition	310
	9.3.2 Parallelization	314
	9.3.3 Performance optimization	325
	9.3.4 Fault tolerance	333

	Contents	xi
9.4	Concluding remarks	339
	References	339
10	Stream analytics: data pre-processing and transformation	342
10.1	Overview	342
10.2	The mining process	342
10.3	Notation	344
10.4	Descriptive statistics	345
	10.4.1 Illustrative technique: BasicCounting	348
	10.4.2 Advanced reading	353
10.5	Sampling	353
	10.5.1 Illustrative technique: reservoir sampling	356
	10.5.2 Advanced reading	357
10.6	Sketches	358
	10.6.1 Illustrative technique: Count-Min sketch	360
	10.6.2 Advanced reading	363
10.7	Quantization	363
	10.7.1 Illustrative techniques: binary clipping and moment preserving quantization	366
	10.7.2 Advanced reading	369
10.8	Dimensionality reduction	370
	10.8.1 Illustrative technique: SPIRIT	373
	10.8.2 Advanced reading	375
10.9	Transforms	375
	10.9.1 Illustrative technique: the Haar transform	379
	10.9.2 Advanced reading	383
10.10	Concluding remarks	383
	References	383
11	Stream analytics: modeling and evaluation	388
11.1	Overview	388
11.2	Offline modeling and online evaluation	389
11.3	Data stream classification	394
	11.3.1 Illustrative technique: VFDT	398
	11.3.2 Advanced reading	402
11.4	Data stream clustering	403
	11.4.1 Illustrative technique: CluStream microclustering	409
	11.4.2 Advanced reading	413
11.5	Data stream regression	414
	11.5.1 Illustrative technique: linear regression with SGD	417
	11.5.2 Advanced reading	419
11.6	Data stream frequent pattern mining	420
	11.6.1 Illustrative technique: lossy counting	425
	11.6.2 Advanced reading	426

xii	Contents	
	11.7 Anomaly detection	427
	11.7.1 Illustrative technique: micro-clustering-based anomaly detection	432
	11.7.2 Advanced reading	432
	11.8 Concluding remarks	433
	References	433
	Part V Case studies	439
12	Applications	441
	12.1 Overview	441
	12.2 The Operations Monitoring application	442
	12.2.1 Motivation	442
	12.2.2 Requirements	443
	12.2.3 Design	445
	12.2.4 Analytics	451
	12.2.5 Fault tolerance	453
	12.3 The Patient Monitoring application	454
	12.3.1 Motivation	454
	12.3.2 Requirements	455
	12.3.3 Design	456
	12.3.4 Evaluation	463
	12.4 The Semiconductor Process Control application	467
	12.4.1 Motivation	467
	12.4.2 Requirements	469
	12.4.3 Design	472
	12.4.4 Evaluation	479
	12.4.5 User interface	481
	12.5 Concluding remarks	482
	References	482
	Part VI Closing notes	485
13	Conclusion	487
	13.1 Book summary	487
	13.2 Challenges and open problems	488
	13.2.1 Software engineering	488
	13.2.2 Integration	491
	13.2.3 Scaling up and distributed computing	493
	13.2.4 Analytics	495
	13.3 Where do we go from here?	496
	References	497
	<i>Keywords and identifiers index</i>	500
	<i>Index</i>	504