Stream Data Processing: A Quality of Service Perspective

Modeling, Scheduling, Load Shedding, and Complex Event Processing

ADVANCES IN DATABASE SYSTEMS

Volume 36

Series Editors

Ahmed K. Elmagarmid
Purdue University
West Lafayette, IN 47907

Amit P. Sheth
Wright State University
Dayton, Ohio 45435

For other titles published in this series, go to
www.springer.com/series/5573

Stream Data Processing: A Quality of Service Perspective

Modeling, Scheduling, Load Shedding, and Complex Event Processing

by

Sharma Chakravarthy
The University of Texas at Arlington
Arlington, TX, USA

Qingchun Jiang
Oracle Corporation
Redwood Shores, CA, USA

Springer

Sharma Chakravarthy
Department of Computer Science
& Engineering
University of Texas at Arlington
Arlington, TX 76019, USA
sharmac@uta.edu

Qingchun Jiang
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065, USA
dustin.jiang@gmail.com

ISBN: 978-0-387-71002-0 e-ISBN: 978-0-387-71003-7
DOI: 10.1007/978-0-387-71003-7

Library of Congress Control Number: 2009920957

Printed on acid-free paper

springer.com

In Loving Memory of my father **Vidwan Chakravarthy Ananthachar** (27^{th} June 1916 to 9^{th} December 2000) who instilled in me an appreciation for learning and a thirst for knowledge

– Sharma Chakravarthy

To my wife **Ailing Wang**, and kids **George and Kevin** for their unconditional love and support

– Qingchun Jiang

Preface

In recent years, a new class of applications has come to the forefront – primarily due to the advancement in our ability to collect data from multitudes of devices, and process them efficiently. These include homeland security applications, sensor/pervasive computing applications, various kinds of monitoring applications, and even traditional applications belonging to financial, computer network management, and telecommunication domains. These applications need to process data continuously (and as long as data is available) from one or more sources. The sequence of data items continuously generated by sources is termed a data stream. Because of the possible never-ending nature of a data stream, the amount of data to be processed is likely to be unbounded. In addition, timely detection of interesting changes or patterns or aggregations over incoming data is critical for many of these applications. Furthermore, the data arrival rates may fluctuate over a period of time and may be bursty at times. For most of these applications, Quality of Service (or QoS) requirements, such as response time, memory usage, and throughput are extremely important. These application requirements make it infeasible to simply load the incoming data streams into a persistent store and process them effectively using currently available database management techniques.

As the currently used data processing paradigm does not fully meet the new requirements of these stream applications, the paradigm shift needed for the effective management of stream data needs to be understood, and new techniques/approaches have to be developed. As stream data is handled by applications in disparate domains, stream data processing can be addressed at different levels and in different contexts: processing of sensor data, pervasive computing, situation monitoring, real-time response, approximate algorithms, on-the-fly mining, complex event processing, or a combination thereof. Although the applications are diverse, some of the fundamental characteristics of stream data and their processing requirements remain common to most applications. Time-critical processing which can be generalized to a QoS requirement is one of the fundamental requirements and forms the thrust of this book. Prior to the advent of stream processing systems as a research area,

database management systems (DBMSs) and their variants were used for some of the above applications. Hence, stream processing can also be related to earlier work on main-memory (or in-memory) databases, embedded databases, and real-time transaction processing systems. This book focuses on the QoS aspects of stream processing and concomitant extensions/modifications that are needed in comparison to the features supported by traditional DBMSs.

In this book, we first discuss the characteristics of stream data and its processing requirements to arrive at a paradigm shift, cull out some of the challenges of a data stream processing system (or a DSMS), and propose our solutions to the challenges. We develop an architecture along with the functional components required for processing data streams and meet the requirements of stream applications. After a broad literature survey to provide a bird's-eye view of the research and development efforts in the thrust areas of stream and complex event processing, the book focuses on functionalities of a DSMS from a *QoS* perspective. In other words, techniques and approaches required for guaranteeing QoS requirements are presented and elaborated in this book. This leads us to focus on the topics of: (*a*) continuous query modeling, (*b*) scheduling strategies for improving QoS, (*c*) load shedding to satisfy QoS requirements with minimal deviation in accuracy of results, and (*d*) synergistic integration of stream and complex event processing to satisfy end-to-end QoS requirements. In addition, we analyze a complex, real-life network fault management application in detail to infer the extensions needed for integrating stream and complex event processing. Finally, we describe the design and implementation of a stream processing system and its synergistic integration with an existing Complex Event Processing (CEP) system.

The organization of the book is designed to facilitate reading of groups of chapters depending upon reader's interest. By choosing appropriate chapters, readers of this book can obtain: an overview of the thrust areas (stream and complex event processing), technical details on important issues of focus topics, and insights into the architecture, design, and implementation of stream & CEP systems. The discussions are cross-referenced at the chapter and section level to help the reader.

Objectives

The objective of this book is to provide a comprehensive understanding of stream processing and its relationship with complex event processing. We start with an overview of these areas, a clear understanding of the processing requirements, and challenges leading to a review of the large body of work that exists in these two areas. We then provide a technical discussion of topics from a QoS perspective. This book also introduces readers to a new paradigm, its relevance, importance, and the theoretical underpinnings of the abstractions needed for the new paradigm. The book also addresses the transition of techniques into prototypes along with design and implementation details. The

contents of this book have been chosen to highlight the QoS aspects of stream processing. The techniques presented here can be applied to other domains and applications as well. In summary, this book provides the state of the art in the areas of stream and complex event processing, presents technical details on selected topics, and discusses design and implementation of prototypes to make the book useful for a diverse audience.

Intended Audience

What background should one have to appreciate this book and benefit the most? Someone who has an undergraduate or a graduate degree in computer science has that background. Those who have taken a course on or have worked with database management systems can readily relate to the paradigm shift and its details. An understanding of relational databases and mathematical inclination will help the reader in enjoying the technical contents of this book. We provide adequate discussion of related work so that the reader can understand and appreciate the approaches and techniques presented in the book. The discussion of prototypes provides insights into the design and implementation issues of a data stream management system that integrates complex event processing in a seamless manner.

This book would be of interest to: senior undergraduate and graduate students, computer professionals, IT managers, database management system users & administrators, time-critical or event-driven application developers, and developers dealing with publish/subscribe systems.

Acknowledgements

This book owes its existence to a large number of people who have played an active role throughout the development of the work presented in the book. The support from National Science Foundation (NSF) for the `MavHome` project (ITR grant 0121297) at The University of Texas at Arlington (UTA) was instrumental in getting us started on this problem. Qingchun's background in managing network applications and the frustrations he faced in the absence of a data stream processing system provided some of the impetus for this work. Another NSF grant (IIS 0534611) for the integration of stream and event processing helped us continue the work and complete the `MavEStream` prototype.

The following students at UTA enthusiastically worked on the `MavStream` and `MavEStream` prototypes as part of their theses: Altaf Gilani, Satyajeet Sonune, Vamshi Pajjuri, Bala Kumar Kendai, and Vihang Garg. Their contributions to this book are very much appreciated. Raman Adaikkalavan has contributed extensively to the contents of chapters on the integration of events with streams and prototype implementations. We sincerely thank him for his

contributions and for his time during the semester on this book. Raman has also edited the entire book, added most of the index, and has improved its technical presentation significantly. It is also our pleasure to thank our colleagues at UTA who were part of the `MavHome` project, especially Diane Cook and Larry Holder.

Proofreading is an arduous and time-consuming task. We were lucky to have several volunteers helping us out on this. We want to thank the following for improving the readability and the quality of presentation of the book: Raman Adaikkalavan, Reuben Brooks, Tejas Chakravarthy, Mohan Kumar, Jonathan Ndede, Roochi Mishra, Charu Puri, and Aditya Telang. Sincere thanks to Aditya Telang for also helping us extensively with figures and references.

Sharma Chakravarthy would like to thank his wife *Shubha* and son *Tejas*, and Qingchun Jiang would like to thank his wife *Ailing* and kids *Kevin* and *George* for their patience, love, encouragement, and support during the preparation of this book.

Last but not least, we would like to thank Ahmed Elmagarmid for suggesting that we consider writing a book on stream processing. This book would not have been possible without the patience, gentle nudging, and timely support of Susan Lagerstrom-Fife and Sharon Palleschi of Springer throughout the preparation of this book. Our sincere thanks to Deborah from Springer author support for resolving several issues on formatting and font usage.

Southlake, Texas — *Sharma Chakravarthy*
Foster City, California — *Qingchun Jiang*
November 2008

How to Use the Book

This book can be used in several ways. It can be used as a reference book, but the contents and organization have been successfully used for a graduate course as well. Organization of this book is meant to help readers maximize their benefit based on their interest in the areas of stream and complex event processing.

It is important to read the first three chapters (Introduction, Overview of Data Stream Processing, and DSMS Challenges) as it sets the stage for the rest of the contents of the book. These chapters introduce the problem and characteristics of stream data processing, continuous queries, need for window-based computation, and differences between stream processing & conventional data processing paradigms leading to the challenges arising from the new requirements. The DSMS challenges chapter also includes a concise overview of the material covered in each chapter of the book to provide a sneak peek into what is coming.

The chapter on literature survey (Chapter 4) provides a broad overview of the work carried out in stream and complex event processing along with an exhaustive bibliography. Most of the research systems and some commercial systems are introduced. In addition, detailed discussion of related work in focus topics is provided. The reader can gauge the interest of the research and vendor community on the thrust areas by the number of currently available prototypes and systems. The large number of references provided in the book is meant to help the reader access additional materials as needed. Internet links are also provided, where possible, for theses, technical reports, and home pages of commercial systems.

Each of the next three chapters (Modeling Continuous Queries over Data Streams, Scheduling Strategies for Continuous Queries, and Load Shedding in a DSMS) are fairly technical and self-contained with respect to its topic. These can be read selectively based on the interest of the reader. Each chapter discusses a problem and our proposed solution to that problem. Some of the theoretical aspects can be skipped without sacrificing the understandability

of the chapter contents. A concise non-technical summary of these chapters is provided in Chapter 3.

The purpose of Chapter 8 (NFM^i : An Inter-Domain Network Fault Management System) is primarily to analyze a complex, real-life application to cull out the critical requirements beyond stream processing. This chapter brings out the need for integrating complex event processing with stream processing.

Those who are mainly interested in complex event processing can read the first and the fourth chapters to obtain an understanding of CEP development, and continue with Chapters 8 on the need for complex event processing. Chapter 9 analyzes the similarities and differences between the two areas before providing a synergistic, integrated solution. If the design and implementation of systems is of interest, read Chapters 10 and 11 after reading the first four setting-the-stage chapters.

We have summarized the usage of the book into an easy-to-use format, based on the areas of interest, in the Table below.

Areas \ Chapters	1	2	3	4	5	6	7	8	9	10	11	12
Stream Processing (SP)	✓	✓	✓	✓	✓[1]	✓[1]	✓[1]					✓
SP Design & Implementation	✓	✓	✓							✓		✓
Complex Event Processing (CEP)	✓			✓				✓	✓		✓	✓
SP & CEP Integration	✓	✓	✓	✓				✓	✓		✓	✓
SP & CEP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

[1] These chapters can be read selectively

Mapping of Areas of Discussion to Chapter Contents

Sharma Chakravarthy continues to use the contents of this book (sans some of the theoretical details) to teach a one-semester graduate course on stream and complex event processing at UTA. Hands-on implementation projects are included to appreciate and apply the concepts taught, and to enhance the course experience. Students implement the same stream and event processing application using different freely available systems to analyze: expressiveness, ease of application development, QoS evaluation, and the ability to handle large data sets. Students discuss their project experience with the rest of the class through an in-class presentation. Most of the stream and complex event processing systems used for the course, such as StreamBase, Aleri, AMiT, Snoop/Sentinel, Coral8, Esper, and RuleCore are available free-of-charge for teaching purposes or as trial/lite/open source versions.

If you are interested in adapting this book for teaching, preliminary slides for the contents of the book can be obtained by sending email to `sharma@cse.uta.edu` with course information.

Contents

List of Figures

List of Tables

List of Algorithms