

**COMPLEX EVENT PROCESSING OVER OUT-OF-
ORDER EVENT STREAMS**

Sinthuja Rajendran

(168259U)

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

February 2019

COMPLEX EVENT PROCESSING OVER OUT-OF-ORDER EVENT STREAMS

Sinthuja Rajendran

(168259U)

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

February 2019

DECLARATION

I declare that this is my own work and this MSc project report does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

Name: Sinthuja Rajendran

We certify that the declaration above by the candidate is true to the best of our knowledge and that this report is acceptable for evaluation for the CS6997 MSc Research Project qualifying evaluation.

Supervisors

.....
Dr. H. M. N. Dilum Bandara

.....
Dr. Srinath Perera

.....
Date

.....
Date

ABSTRACT

Complex Event Processing (CEP) enables real-time inferring of events and patterns of interest. Aggregation on a time window of events and pattern matching are two of the core functionalities of CEP. Accuracy of these CEP operations depend on the order of the events received at the CEP engine. However, due to network delay, environmental differences in event producing sources, and distributed CEP systems, event arrival order at the CEP engine maybe different from the order of event generation at the source. Such out-of-order events may lead to incorrect output events by the CEP engine.

We propose a novel solution to handle the out-of-order events in three steps, namely (a) ordering events from the same source, (b) ordering events from multiple sources, and (c) optimizing query operator to further improve the accuracy after applying former steps. Sequence numbers are used to order events from a single source, whereas estimated time drift of each event source is used to order event from multiple event sources. Finally, the query operators are optimized to reduce the error of remaining out-of-order events. Performance of the proposed solution is evaluated using the DEBS 2013 Football dataset. The performance analysis shows that the proposed techniques result in 9600% to 21300% and 1200% to 2500% reduction in latency compared to MP-K-Slack and AQ-K-Slack techniques, respectively. Further, the proposed solution was able to order the events with 99.97% - 99.99% accuracy. While it is comparatively lower than MP-K-Slack which had an accuracy of 99.99% and better than AQ-K-Slack which had an accuracy of 99.02%. Therefore, the proposed solution provides a good balance between latency and accuracy. The additional optimizations carried out in aggregator and pattern matching operators further increased the accuracy of the results by 50% compared to the final results obtained without these query optimizations.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deep sense of gratitude and profound feeling of admiration to my project supervisors. Many thanks go to all those who helped me in this work. My special thanks to University of Moratuwa for giving an opportunity to carry out this research project.

I would like to gratefully acknowledge to Dr. Dilum Bandara, the internal supervisor of the project, for sharing the experiences and expertise with the project matters. I would like to extend my heartfelt gratitude to Dr. Srinath Perera, the external project supervisor, for his continuous guidance and support throughout the whole duration of the project. Last but not least, I thank to Dr. Miyuru Dayarathna, for sharing the experience in the related work carried out with Siddhi engine, and all those who like to remain anonymous although the help they provided me was valuable.

Thank you.

TABLE OF CONTENTS

DECLARATION	i
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vii
LIST OF TABLES	ix
1. INTRODUCTION	1
1.1. Background	1
1.2. Motivation	2
1.3. Research Question	4
1.4. Objectives	5
1.5. Outline	5
2. LITERATURE REVIEW	6
2.1. Complex Event Processor Functionalities	6
2.2. Out-of-order Event Handling Approaches	7
2.2.1. Buffer-based Approach	7
2.2.2. Punctuation-based Approach	8
2.2.3. Speculation-based Approach	9
2.2.4. Approximation-based Approach	9
2.3. Buffer-based Techniques	9
2.3.1. K-Slack Approach	10
2.3.1.1. Out-of-order Event Processing in SASE	12
2.3.2. MP-K-Slack Approach	16

2.3.3. AQ-K-Slack Approach	18
2.3.4. Latency Distance and Purging Time Based out-of-order Event Processing (LDOP)	22
2.3.5. K-Slack Chain Approach	23
2.3.6. Summary	27
3. METHODOLOGY	29
3.1. Definitions	29
3.2. Proposed Solution	31
3.2.1. Handling Events Produced from Single Event Source	31
3.2.2. Handling Events Produced from Multiple Event Sources	36
3.2.3. Query Operators with Out-of-order Events	40
3.2.3.1. Time Batch Window and Aggregation Operators	40
3.2.3.2. Pattern Matching	42
3.4. Summary	43
4. IMPLEMENTATION	45
4.1. Summary	53
5. PERFORMANCE EVALUATION	55
5.1. Dataset	55
5.1.1. Dataset for Single Source	59
5.1.2. Dataset for Multiple Sources	59
5.1.3. Dataset with Time Drift for Event Source	60
5.1.4. Dataset for Pattern Matching	60
5.2. Experimental Setup	61
5.2.1. Prototype	61
5.2.2. Analysis of Out-of-order Events Handling Solution	61

5.2.2.1. Performance and Accuracy with Single Event Source	62
5.2.2.2. Performance and Accuracy with Multiple Event Sources	68
5.2.2.3. Accuracy with Time Drifted Event Sources	71
5.2.3. Analysis of Query Operators under Out-of-order Events	73
5.2.3.1 Aggregator Operator	73
5.2.3.2. Pattern Matching Operator	76
5.3. Summary	78
6. CONCLUSION	80
6.1. Summary	80
6.1. Research Limitations	82
6.2. Future Work	84
REFERENCES	86

LIST OF FIGURES

Figure 1.1: Out-of-order event arrival	2
Figure 2.1: Out-of-order event arrival – K- Slack	11
Figure 2.2: Sorting the events in the window with K-Slack	11
Figure 2.3: Event Query Plan	12
Figure 2.4: Query Evaluation of SASE	13
Figure 2.5: Problems exists in SSC and PSSC with out-of-order event arrival	14
Figure 2.6: Out-of-order event arrival handling in MP-Slack	18
Figure 2.7: Design of AQ-K-Slack	19
Figure 2.8: Adoption of α using a PD controller	21
Figure 2.9: A Global Query Graph	24
Figure 2.10: Shared disorder handling for a query graph G that has no filter operators: single K-slack versus K-slack chain	26
Figure 3.1: Event source S_l sending event stream R_l and event stream R_2 to CEP	32
Figure 3.2: Communication between the CEP receiver and event source to calculate the timestamp drift	37
Figure 3.3: Out-of-order event handling with multiple event sources with event stream R_l .	39
Figure 3.4: Multiple time batch windows to cover the events from $\pm (T_a/2)$ of original time batch window w .	41
Figure 4.1: Out-of-order event flow of an event stream within siddhi engine.	45
Figure 4.2: Sequence-based ordering in sequence-based reorder extension.	50
Figure 5.1: Playing field and its dimensions	55
Figure 5.2: Average events inter-arrival time with out-of-order event Dataset 1 and Dataset 2	57

Figure 5.3: Maximum events inter-arrival time in Dataset 1 and Dataset 2.	57
Figure 5.4: Average events inter-arrival time with out-of-order event Dataset 3	58
Figure 5.5: Maximum event inter-arrival time in seconds for Dataset 3.	58
Figure 5.6: Average and maximum latency of events for Dataset 1 with proposed sequence-based approach.	64
Figure 5.7: Average and max latency of events for Dataset 1 with MP-K-Slack, AQ-K-Slack and sequence-based approach	64
Figure 5.8: Average latency of all three approaches with respective to the dataset.	65
Figure 5.9: Total number of out-of-order events for all three approaches and the respective datasets.	67
Figure 5.10: Variation of average latency in ms of all three approaches with number of event sources.	68
Figure 5.11: Average latency for sequenced based approach across all event sources.	69
Figure 5.12: Total out-of-order events with multiple event sources for all three approaches	70
Figure 5.13: The number of out order events with amount of time drifts for sequence based approach.	72
Figure 5.14: The difference between the of average values for normal time batch window, and reorder based time batch window.	75

LIST OF TABLES

Table 2.1: The comparison of the out-of-order event handling approaches	10
Table 2.2: Comparison of buffer-based approaches	28
Table 3.1: List of Symbols	33
Table 5.1: Description of event attributes	56
Table 5.2: Input Dataset and out-of-order events	56
Table 5.3: Specification of the machine that was used for the experiment and evaluation	61
Table 5.4: Overall Summary of Latency incurred for events in Dataset1 with all out-of-order handling approaches	65
Table 5.5: Total number of out-of-order events with Dataset1 with all out-of-order handling approaches	65
Table 5.6: Average Latency for all three approaches along with datasets.	66
Table 5.7: Accuracy of all three approaches with datasets.	66
Table 5.8: Average latency for all three approaches along with number of events sources	69
Table 5.9: Accuracy of all three approaches when publishing events with multiple sources.	70
Table 5.10: The total out-of-order events produced in sequence based approach with the time drift between the event sources, with and without time syncing of event source.	71
Table 5.11: The differences between the expected and actual average values.	74
Table 5.12: The average value of the velocity after using the reorder based time batch window.	75
Table 5.13: Results of the pattern matching dataset.	77

LIST OF ABBREVIATIONS

AIS	Active Instance Stack
ATM	Automated teller Machine
CEP	Complex Event Processor
DAG	Directed Acyclic Graph
IoT	Internet of Things
LDOP	Latency Distance and Purging Time
NFA	Nondeterministic finite automaton
NTP	Network Transfer Protocol
PD	Proportional Derivative
PSSC	Purged Sequence Scan Construction
QDDH	Quality Driven Disorder Handling
RFID	Radio-Frequency Identification
SC	Sequence Construction
SL	Selection
SS	Sequence Scan
TCP	Transmission Control Protocol
TF	Transformation
UDP	User Datagram Protocol
WD	Window