SCATTER-GATHER BASED APPROACH IN SCALING COMPLEX EVENT PROCESSING SYSTEMS FOR STATEFUL OPERATORS

Sriskandarajah Suhothayan

(168268V)

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

February 2019

SCATTER-GATHER BASED APPROACH IN SCALING COMPLEX EVENT PROCESSING SYSTEMS FOR STATEFUL OPERATORS

Sriskandarajah Suhothayan

(168268V)

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 acknowledgment 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 acknowledgment 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: Sriskandarajah Suhothayan

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

With the introduction of Internet of Things (IoT), scalable Complex Event Processing (CEP) and stream processing on memory, CPU, and bandwidth constraint infrastructure have become essential. While several related work focuses on replication of CEP engines to enhance scalability, they do not provide expected performance while scaling stateful queries for event streams that do not have pre-defined partitions. Most of the CEP systems provide scalability for stateless queries or for the stateful queries where the event streams can be partitioned based on one or more event attributes. These systems can only scale up to the pre-defined number of partitions, limiting the number of events they can process. Meanwhile, some CEP systems do not support cloud-native and microservices features such as startup time in milliseconds.

In this research, we address the scalability of CEP systems for stateful operators such as windows, joins, and pattern by scaling data processing nodes and connecting them as a directed acyclic graph. This enabled us to scale the processing and working memory using the scatter and gather based approach. We tested the proposed technique by implementing it using a set of Siddhi CEP engines running on Docker containers managed by Kubernetes container orchestration system. The tests were carried out for a fixed data rate, on uniform capacity nodes, to understand the processing capacity of the deployment. As we scale the nodes, for all cases, the proposed system was able to scale almost linearly while producing zero errors for patterns, 0.1% for windows, and 6.6% for joins, respectively. By reordering events the error rate of window and join queries was reduced to 0.03% and 1% while introducing 54ms and 260ms of delays, respectively.

ACKNOWLEDGMENT

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

I would like to gratefully acknowledge Dr. Dilum Bandara, the internal project supervisor, for his continuous guidance and support throughout the whole duration of the project, under whose supervision that I gained a clear concept of what I should do. I would also like to extend my heartfelt gratitude to Dr. Srinath Perera, the external supervisor of the project, for sharing the experiences and expertise with the project matters. Last but not least, I thank all those who like to remain anonymous although the help they provided to me was valuable.

Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ATM	Automated teller Machine
CEP	Complex Event Processor
CPU	Central processing unit
DBMS	Database Management System
ESB	Enterprise Service Bus
GC	Garbage Collection
IoT	Internet of Things
JMS	Java Messaging Service
NFA	Non-deterministic Finite Automata
RDD	Resilient Distributed Dataset
TCP	Transmission Control Protocol
TPS	Transactions Per Second
XA	eXtended Architecture