

ACCELERATED COMPLEX EVENT PROCESSING WITH GRAPHICS PROCESSING UNITS

Prabodha Srimal Rodrigo

Registration No. : 138230V



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk
Degree of Master of Science

Department of Computer Science & Engineering

University of Moratuwa
Sri Lanka

April 2015

ACCELERATED COMPLEX EVENT PROCESSING WITH GRAPHICS PROCESSING UNITS

Prabodha Srimal Rodrigo

Registration No. : 138230V



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Dissertation submitted in partial fulfillment of the requirements for the
degree Master of Science in Computer Science

Department of Computer Science & Engineering

University of Moratuwa
Sri Lanka

April 2015

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a 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 University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Candidate

.....

Date

.....

U.P.S. Rodrigo

The above candidate has carried out research for the Masters Dissertation under my supervision.

Supervisors

.....

Dr. H.M.N. Dilum Bandara

.....

Dr. Srinath Perera

.....

Date

.....

Date

Acknowledgments

This project would not have been possible without the support of many people. Specially thanks to my supervisors, Dr. Dilum Bandara and Dr. Srinath Perera, for their valuable guidance. Many thanks to our MSc research project coordinator, Dr. Malaka Silva, for his dedication and support. Thanks to all the lecturers at the Faculty of Computer Science and Engineering, University of Moratuwa, for their valuable advice. Also thanks to my wife, Malmi Amadoru and my parents always offering support and love. Finally, I thank to my numerous friends who endured this long process with me.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Abstract

As Big Data scenarios increasingly become common, a large number of distributed data processing systems require timely processing of high volumes of real-time data streams. Detecting complex correlations between incoming data streams in near real-time is at the heart of these data processing systems. Complex Event Processing (CEP) have been dominating in this domain since inception a decade back. But, growth of Big Data volumes demands for more performance and faster processing. CEP operators like stream join and event patterns require considerable processing power and have huge impact on the overall query processing performance. In some use cases these operators have to operate on lots of events simultaneously. Making parallel algorithms for these operators is a common approach for improving the individual operator performance.

A Graphics Processing Unit (GPU) provides a vast number of parallel computing cores and leverage new parallel algorithms which enables novel problem solving approaches for existing problems. But the challenge is combining complex event processing and GPUs in the right way to get the maximum performance out of the this parallel hardware. There had been attempts to use parallel hardware in improving CEP performance in both commercial and academic implementations, and most of them uses multi-core approach. Only a very few researches had used GPUs for CEP. We believe the lack of GPU related CEP researches is that they are not designed to benefit from parallel processing in GPUs.

In this research we investigate how and when GPUs can be used to improve the query processing performance of a popular open source CEP implementation, Siddhi CEP. Siddhi, by design, supports for parallel query processing in multi-core CPUs. This work propose a novel approach for parallel event processing in GPUs with several GPU event processing algorithms. Performance evaluation on our implemented algorithms shows, for a mix of complex queries, parallel event processing on GPUs achieve more than ten times event processing throughput than the sequential processing in CPUs. Moreover, our approach helped to reduce event queuing at the incoming event queue when there are high frequent input event stream and several complex queries.

Keywords. Complex Event Processing, Parallel Hardware, GPGPU, Siddhi CEP.

Contents

Declaration	i
Acknowledgments	ii
Abstract	iii
List Of Figures	vii
List Of Tables	ix
List of Abbreviations	1
1 Introduction	1
1.1 Big Data and Data at Move	1
1.2 Complex Event Processing	2
1.3 Parallel Hardware	3
1.4 CEP on Parallel Hardware	4
1.5 Problem Statement	5
1.6 Contributions	5
1.7 Challenges	6
1.8 Organization of the Thesis	7
2 Literature Review	8
2.1 Event Processing	8
2.1.1 Data Stream Processing	11
2.1.2 Complex Event Processing	12
2.1.2.1 Primitive Events and Complex Events	12
2.1.2.2 Event Stream Processing	13
2.1.2.3 CEP Use Cases	15
2.2 Siddhi CEP Engine	16
2.2.1 Siddhi Architecture	16
2.2.2 Siddhi Internal Data Model	17
2.2.2.1 Event and Event Streams	18
2.2.2.2 Query Data Model	19
2.2.3 Processor Architecture	24
2.3 Parallel Hardware Architectures	25

2.3.1	Multi-Core Processor Systems	27
2.3.2	Field-Programmable Gate Arrays	28
2.3.3	Accelerator Co-processors	28
2.3.3.1	Cell Broadband Engine	29
2.3.3.2	Graphics Processing Units (GPUs)	29
2.3.4	GPU Programming Environments	33
2.3.4.1	GPU Programming with CUDA	33
2.3.5	Java in GPU Programming	36
2.4	Related Work	40
2.4.1	Event Processing on Parallel Hardware	41
2.4.2	Pub-Sub Middleware on Parallel Hardware	44
2.4.3	CEP on Other Hardware Architectures	47
3	Siddhi GPU Query Processors	48
3.1	Siddhi CEP Architecture	49
3.1.1	Siddhi Architectural Components	49
3.2	Siddhi GPU Query Runtime	53
3.3	Internals of GpuStreamRuntime	56
3.3.1	Event Serialization	57
3.3.1.1	Java NIO ByteBuffer	58
3.3.1.2	Java Wrapper for Event Processing Library	59
3.3.2	Increasing Performance in GpuStreamRuntime	60
4	GPU Event Processing Library	61
4.1	CUDA Access for Java Code	61
4.2	Event Processing Library for GPU	63
4.2.1	GPU Event Processors	66
4.2.2	Event Processor Basic Concepts	66
4.3	GPU Filter Event Processor	68
4.4	GPU Sliding Window Event Processor	73
4.4.1	Generating Window Output Events	74
4.4.2	Set Post-process Window State	75
4.5	GPU Event Stream Join Processor	81
4.5.1	Concurrent Event Window Access	85
4.6	Improving Performance in Event Processing Library	88
5	Evaluation	90
5.1	Analysis of the Workload	91
5.2	Experiment Setup and Methodology	92
5.3	Filter Query Performance	94
5.3.1	Event Consume Speedup Analysis	96

5.3.2	GPU Processing Time Analysis	97
5.4	Join Query Performance	100
5.4.1	Event Consume Speedup Analysis	102
5.4.2	GPU Processing Time Analysis	104
5.4.3	Input Event Processing Throughput Analysis	106
5.5	Query Mix Analysis	106
5.5.1	Event Consume Speedup Analysis	108
6	Conclusions	112
6.1	Conclusion	112
6.2	Future Work	115
6.2.1	Implementing GPU Algorithms for Other CEP Operators	115
6.2.2	GPU Aware CEP Architecture	115
6.2.3	Distributed GPU Servers	115
6.2.4	Automatic Query Configure to Run on GPUs	116
6.2.5	Runtime GPU Kernel Generation	116
	References	117



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Figures

2.1	Event Processing Architecture.	10
2.2	Siddhi High Level Architecture.	16
2.3	Siddhi Event and Event Stream representation.	18
2.4	High level Siddhi query architecture.	20
2.5	Siddhi filter condition data model.	20
2.6	Siddhi time window architecture.	22
2.7	Siddhi Join query data model.	24
2.8	Siddhi Processor Architecture.	24
2.9	CPU and GPU Architecture.	30
2.10	Basic modern GPU architecture.	32
2.11	CUDA threading model.	34
2.12	CUDA memory model.	36
2.13	CDF algorithm data structures and processing for listing 2.4.	42
2.14	Publish-Subscribe network.	45
2.15	CUDA Content-based Matcher (CCM) algorithm data structures.	46
3.1	Higher Level Architecture of Proposed Solution.	48
3.2	Siddhi Query Processors.	50
3.3	Siddhi Event Processing Constructs.	52
3.4	Siddhi QueryRuntime organization based on threading model and input stream count.	52
3.5	Proposed solution for GPU event processing.	53
3.6	Interconnecting GpuStreamRuntime and SingleStreamRuntime with output event queue.	54
3.7	Internals of GpuStreamRuntime.	57
3.8	Event serialization in to memory buffer.	59
4.1	High-level design of GPU Event Processing Library.	64
4.2	Basic concept of GPU event processors.	67
4.3	Filter operator expression tree conversion to executor array for query defined in Listing 4.1.	69

4.4	Length Sliding Window Processor.	74
4.5	Length Sliding Window Processor - output event calculation.	75
4.6	Event window state set algorithm: scenario 1.	79
4.7	Event window state set algorithm: scenario 2.	80
4.8	Event window state set algorithm: scenario 3.	80
4.9	Event replace mechanism of EventWindow state update GPU algorithm.	81
4.10	Event stream join processor joins two event streams based on a join condition.	83
4.11	Siddhi GPU event processor model for stream join processor.	84
4.12	GPU thread allocation for join output event calculation phase.	85
4.13	Event window double buffering.	87
5.1	Architecture of the experiment setup.	93
5.2	Filter query event consume rate speedup for different GPU thread block sizes.	95
5.3	Filter query event consume rate speedup against concurrent query count (event batch size 2048 events).	97
5.4	Filter query event consume rate speedup for different event batch sizes.	98
5.5	GPU Filter processor: GPU processing time breakdown (event batch size 2048 events).	99
5.6	GPU Filter processor: GPU processing time as a percentage of total processing time (event batch size 2048 events).	100
5.7	Join Query: Average event consume rate speedup against concurrent query count (event batch size 2048 events).	102
5.8	Join Query: Average event consume rate speedup for multiple GPU devices (event batch size 2048 events).	103
5.9	Join Query: Input event queue publish latency against concurrent query count (event batch size 2048 events).	104
5.10	GPU Join processor: Processing time breakdown (event batch size 2048 events).	105
5.11	Join Query: Input event processing throughput against concurrent query count (event batch size 2048 events).	107
5.12	Query Mix: Average event consume rate speedup against concurrent use case count (batch size 2048 events).	108
5.13	Query Mix: Input event queue publish latency against concurrent use case count (batch size 2048 events).	109
5.14	Query Mix: Input event processing throughput against concurrent use case count (batch size 2048 events).	110

List of Tables

2.1	Similarities between Event Driven Architecture, Complex Event Processing and Human Body.	14
2.2	Summary of CUDA memory hierarchy.	36
2.3	Comparison between CUDA and OpenCL.	39
3.1	Siddhi Query Annotations for GpuStreamRuntime.	55
5.1	Filter query test setup parameters.	95
5.2	Filter query GPU processing time breakdown.	99
5.3	Join query test setup parameters.	100
5.4	Join query GPU processing time breakdown.	105
5.5	Query _{mix} test setup parameters.	108



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Abbreviations

CEP	Complex Event Processing
CMP	Chip Multi-processors
CUDA	Compute Unified Device Architecture
DBMS	Database Management Systems
DSMS	Data Stream Management System
EDA	Event Driven Architecture
FPGA	Field-Programmable Gate Array
GPGPU	General Purpose Computing on the Graphics Processing Unit
GPU	Graphic Processing Units
HPC	High Performance Computing
JSON	Javascript Object Notation
MIMD	Multiple Instruction, Multiple Data streams
MISD	Multiple Instruction, Single Data stream
POJO	Plain Old Java Object
SIMD	Single Instruction, Multiple Data stream
SISD	Single Instruction, Single Data stream
SMP	Symmetric Multiprocessing



University of Moratuwa, Sri Lanka.

Electronic Theses & Dissertations

www.lib.mrt.ac.lk