



Cairo University

HETEROGENEOUS BIG-DATA CLUSTER FOR COMPUTER VISION APPLICATIONS

By

Hazem Abdelmegeed Elsayed Abdelhafez

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Electronics and Communications Engineering

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016

HETEROGENEOUS BIG-DATA CLUSTER FOR COMPUTER VISION APPLICATIONS

By
Hazem Abdelmegeed Elsayed Abdelhafez

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Electronics and Communications Engineering

Under the Supervision of

Prof. Dr. Hossam Ali Hassan Fahmy

Prof. Dr. Ameen Mohamed Nassar

.....
Professor
Electronics and Communications
Engineering Department
Faculty of Engineering, Cairo
University

.....
Professor
Electronics and Communications
Engineering Department
Faculty of Engineering, Cairo
University

Dr. Mohamed Mohamed Rehan

.....
Chief Technical Officer
AvidBeam

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016

HETEROGENEOUS BIG-DATA CLUSTER FOR COMPUTER VISION APPLICATIONS

By
Hazem Abdelmegeed Elsayed Abdelhafez

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Electronics and Communications Engineering

Approved by the
Examining Committee

Prof. Dr. Hossam Ali Hassan Fahmy, Thesis Main Advisor

Prof. Dr. Ameen Mohamed Nassar, Member

Dr. Mohamed Mohamed Rehan, Member, Chief Technical Officer,
AvidBeam

Prof. Dr Elsayed Eissa Abdo Hemayed, Internal Examiner

Prof. Dr. Khaled Mostafa Elsayed, External Examiner, Faculty of
Computers and Information, Cairo University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016

Engineer's Name: Hazem Abdelmegeed Elsayed Abdelhafez
Date of Birth: 26/09/1990
Nationality: Egyptian
E-mail: hazem.abdelmegeed@gmail.com
Phone: +2 01001128853
Address: 4, Street 100, Maadi, Cairo, Egypt
Registration Date: 01/03/2014
Awarding Date: / /2016
Degree: Master of Science
Department: Electronics and Communications Engineering



Supervisors:

Prof. Dr. Hossam Ali Hassan Fahmy
Prof. Dr. Ameen Mohamed Nassar
Dr. Mohamed Mohamed Rehan

Examiners:

Prof. Dr. Hossam Ali Hassan Fahmy
Prof. Dr. Ameen Mohamed Nassar
Dr. Mohamed Mohamed Rehan, Chief Technology Officer, AvidBeam
Prof. Dr. Elsayed Eissa Abdo Hemayed
Prof. Dr. Khaled Mostafa Elsayed, Faculty of Computers and Information, Cairo University

Title of Thesis:

HETEROGENEOUS BIG-DATA CLUSTER FOR COMPUTER VISION APPLICATIONS

Key Words:

Big-data; Parallel Computing; Graphical Processing Units; Computer Vision; Heterogeneous Computing

Summary:

A token-based scheduler is developed to enable efficient utilization of graphics processing unit in big-data clusters specifically for computer vision applications. The scheduler addresses the racing conditions that occur on the graphics processing unit due to simultaneous access by the parallel instances of the running application. The presented scheduler enables the porting of computer vision applications to big-data cluster with heterogeneous computing capabilities where multi-core central processing units exist alongside graphical processing unit.

Acknowledgments

I would like to thank my mother and my family for their support and understanding during the time working on the MScs, without their support and love I wouldn't have crossed this far in my academic life and career. I would like to thank Dr. Hossam and Dr. Rehan for their endless support, patience, sincerity and dedication during my journey in this thesis. I would like also to thank all my colleagues at Avidbeam, Mohamed Fouad, Ahmed Sobhi and Ahmed Abdelsamad for helping me prepare the hardware setup I used in my thesis. Also I would like to thank Menna Ghoneim, Dina Helal and Rana Morsi for helping me understand computer vision algorithms better and for reviewing my thesis to make sure it is in perfect shape. I would like also to thank the management at Avidbeam, Dr. Hani El-Gebaly and Hossam Samy for providing all the help I needed and supporting me during my MScs journey. Finally, I am also thankful for the free, Open-Source community that provided freely most of the tools I used in this work and we gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

Dedication

Dedicated to my mother and my family...

Table of Contents

Acknowledgments	i
Dedication	ii
Table of Contents	iii
List of Tables	v
List of Figures	vi
List of Abbreviations	ix
Abstract	x
1 INTRODUCTION	1
1.1 Motivation and Overview	1
1.2 Problem Statement	4
1.3 Objective and Challenges	4
1.4 Thesis Organization	5
2 BIG-DATA TECHNOLOGY	6
2.1 Introduction	6
2.2 Hadoop	6
2.2.1 Background and Overview	7
2.2.1.1 Hadoop Distributed File System	7
2.2.1.2 Hadoop Map-Reduce Framework	9
2.2.1.3 Yet Another Resource Negotiator	9
2.2.2 Operation	10
2.3 Spark	12
2.3.1 Background and Overview	12
2.3.2 Operation	13
2.4 Storm	15
2.4.1 Background and Overview	15
2.4.2 Operation	17
2.5 Big-data Framework for Computer Vision	19

2.6	Conclusion	20
3	GRAPHICS PROCESSING UNIT	21
3.1	History and Background.....	21
3.2	Architecture and Operation	22
3.3	GPU Programming Frameworks	27
3.4	GPUs Role in Computer Vision Applications	30
3.5	Conclusion	33
4	LITERATURE REVIEW	34
4.1	Introduction	34
4.2	Map-reduce Related Work	34
4.2.1	CUDA Framework	34
4.2.2	Other Frameworks	40
4.3	Generic Heterogeneous Computing with GPUs	40
4.4	Conclusion	43
5	DESIGN AND IMPLEMENTATION	44
5.1	Introduction	44
5.2	Computer Vision Processing on Big-data Frameworks	44
5.3	Computer Vision Processing on Big-data Frameworks with GPU Support	48
5.3.1	Priority-based Static Scheduler	48
5.3.2	GPU Utilization-based Scheduler	50
5.3.3	Single Allocation Scheduler	52
5.3.4	Token-based Scheduler	54
5.4	Conclusion	55
6	EXPERIMENTAL RESULTS	57
6.1	Introduction	57
6.1.1	Hardware Setup	57
6.1.2	Software Setup	58
6.1.3	Evaluation Procedure	61

6.1.3.1	GPU Scheduler for OpenCL framework performance	
	evaluation	62
6.1.3.2	GPU Scheduler for CUDA framework performance eval-uation	65
6.1.3.3	Token-based Scheduler Versus Nvidia MPS for CUDA	
	Framework	68
6.2	Conclusion	70
7	CONCLUSION AND FUTURE WORK	71
7.1	Conclusion	71
7.2	Future Work	72
	References	73

List of Tables

6.1	Master and slave nodes hardware specs	57
6.2	OpenCL framework performance metrics	65
6.3	Frame processing latency enhancement for OpenCL framework	65
6.4	CUDA framework performance metrics	67
6.5	Frame processing latency enhancement for CUDA framework	68
6.6	MPS performance metrics	70
6.7	MPS latency enhancement	70

List of Figures

1.1	The Digital Universe: 50-fold growth from the beginning of 2010 to the end of 2020 [6]	2
2.1	HDFS architecture [17]	8
2.2	Data management in HDFS [17]	8
2.3	YARN architecture [20]	10
2.4	Hadoop word-count example	11
2.5	Spark software stack	12
2.6	Spark cluster overview [25]	14
2.7	Spark streaming workflow [27]	15
2.8	Storm cluster overview	16
2.9	Storm parallelism components	17
2.10	Storm single layer topology	18
2.11	Storm multiple layers topology	18
3.1	Graphics rendering tasks migration from CPU to GPU [33]	21
3.2	GeForce3 programmable architecture [34]	23
3.3	Unified GPU architecture [33]	24
3.4	Stream processor architecture [33]	25
3.5	Nvidia Multi-process Service architecture [39]	26
3.6	CUDA thread hierarchy [33]	28
3.7	OpenCL architecture	29
3.8	OpenCV performance comparison [33]	32
3.9	CUDA speedup versus CPU [33]	32
3.10	OpenCL speedup versus CPU [33]	33
4.1	GPMR multi-stage map-reduce pipeline [53]	36
4.2	MGMR workflow on multi-GPU system [55]	38
4.3	OCLScheduler client-server model [65]	41
4.4	Hyperflow architecture [66]	42
5.1	Map-reduce workflow in computer vision applications	46
5.2	CPU based Storm topology for computer vision processing	47
5.3	GPU priority-based scheduler	49
5.4	GPU utilization based scheduler workflow	50
5.5	Storm topology with utilization-based GPU scheduler	51
5.6	Single allocation GPU scheduler algorithm	53
5.7	Storm topology with single allocation GPU scheduler	54
5.8	Token-based GPU scheduler algorithm	56
6.1	Evaluation test-bed setup	58
6.2	Example of rectangular features	59

6.3	Mutli-stage cascade classifier	60
6.4	CPU-only Storm topology	60
6.5	Hybrid CPU-GPU Storm topology	61
6.6	OpenCL performance evaluation	63
6.7	CUDA performance evaluation	66
6.8	MPS performance evaluation against token-based scheduler for CUDA framework	69

List of Abbreviations

AdaBoost	Adaptive Boost
AM	Application Master
APIs	Application Programming Interfaces
ASF	Apache Software Foundation
AVX	Advanced Vector Extension
BSD	Berkeley Software Distribution
Cgroups	Linux Control Groups
CUDA	Compute Unified Device Architecture
CUDASA	Computer Unified Device and Systems Architecture
CUs	Compute Units
DAGs	Directed Acyclic Graph
DSL	Domain-Specific Language
DSP	Digital Signal Processing
FPGA	Field Programmable Gate Array
FPS	Frames per Second
GFS	Google File System
GPGPU	General Purpose Computing on GPU
IGP	Integrated Graphics Processor
IPP	Intel Performance Primitives
JAR	Java Archive
JSON	JavaScript Object Notation
JVMs	Java Virtual Machines
LPR	License Plate Recognition
MPS	Nvidia's Multi-process Service
ms	milliseconds
NM	Node Manager
OCR	Optical Character Recognition
OpenACC	Open Accelerators
OpenCL	Open Computing Language
OpenCV	Open Computer Vision
OpenGL	Open Graphics Library
PCI	Peripheral Component Interconnect
PEs	Processing Elements
PMGMR	Pipelined Mutli-GPU Map-Reduce
RDD	Resilient Distributed Dataset
RM	Resource Manager
SDK	Software Development Kit

SIMD	Single Instruction Multiple Data
SM	Streaming Multiprocessor
SSE	Streaming SIMD Extensions
SSSE3	Supplemental Streaming SIMD Extensions
TOM	Task-oriented Modules
VPE	Virtual Processing Elements
YARN	Yet Another Resource Negotiator

Abstract

Big-data technology in recent years has become increasingly utilized to process huge amount of data in a timely manner. The growth in the amount of visual data - videos and images - generated nowadays raises the need for porting computer vision applications to big-data frameworks in order to increase the processing throughput of these applications.

On the other hand, developers and the scientific community have already ported many computer vision algorithms to the Graphics Processing Unit (GPU) that successfully accelerates these algorithms thanks to its data-parallel architecture.

Combining big-data with GPUs to scale computer vision applications both horizontally and vertically yields a promising architecture for processing the huge amounts of visual data and to fulfill the urging need to mine these data for underlying patterns and information. Unfortunately, the number of GPUs available on a typical processing node in a big-data cluster is limited and most of the time there is only one GPU on such nodes. Therefore, multiple instances of the same computer vision application running on top of any big-data framework leads to competition between these instances on the scarce GPU resource.

In this thesis, we address the challenge of combining GPUs with big-data technology in order to accelerate the processing of computer vision applications. We introduce a GPU scheduler that is responsible for assigning the GPU to multiple instances of the computer vision application efficiently with minimal competition and best performance compared to using either the GPU or the Central Processing Unit (CPU) solely. In order to achieve this we propose a token based scheduler that guarantees that no competition occurs on the GPU. The evaluation shows increased processing throughput up to 2.3x compared to CPU-only big-data processing with 24 cores, 2.1x compared to CPU-GPU big-data processing and up to 32x compared to the sequential processing on a single CPU core.