PERFORMANCE ESTIMATION AND MAPPING OF APPLICATIONS ONTO GPUs

ARUN PARAKH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY DELHI

AUGUST 2016

PERFORMANCE ESTIMATION AND MAPPING OF APPLICATIONS ONTO GPUs

by

ARUN PARAKH

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

to the

Indian Institute of Technology Delhi
AUGUST 2016

Certificate

This is to certify that the thesis titled **Performance Estimation and Mapping of Applications onto GPUs** being submitted by **Mr. Arun Parakh** for the award of **Doctor of Philosophy** in Computer Science and Engineering is a record of bona-fide work carried out by him under our guidance and supervision at the Computer Science and Engineering Department, Indian Institute of Technology Delhi. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

M. Balakrishnan
Professor
Dept. of Computer Science & Engineering
Indian Institute of Technology Delhi
New Delhi, India 110 016

Kolin Paul
Associate Professor
Dept. of Computer Science & Engineering
Indian Institute of Technology Delhi
New Delhi, India 110 016

This thesis is dedicated to..

Parent

&

Teacher

&

Student

Acknowledgments

Though only my name appears on the cover of this thesis, a great many people have contributed to its production. I owe my gratitude to all those people who have made this thesis possible and because of whom my Phd experience has been one that I will cherish forever.

My deepest gratitude is to my supervisor, Prof. M. Balakrishnan. I have been amazingly fortunate to have an guide who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered. I hope that one day I would become as good a supervisor to my students as Prof. Bala has been to me.

My co-advisor, Dr. Kolin Paul, has been always there to listen and give advice. I am deeply grateful to him for the long discussions that helped me sort out the technical details of my work. Dr. Kolin always inspired me to improve my knowledge for Computer Architecture with his sound and clear concepts of this domain.

This research would not be possible without support of Mayura. She constantly monitored the work progress and feedback honestly. I am thankful to her to let me work around the clock. Writing skill of Vayam and Noopur helped me alot to prepare draft of the thesis as well as research papers so I am thankful to them.

I would like to thank my colleague Vaibhav Jain, Sarat Chandra Varma, Sohan Lal Sharma, Atul Sharma, Lava Bhargava, Rajesh Kumar Pal, Yamuna Prasad and all the other colleagues for their cooperation and support. I wish to thank Prof. Anshul Kumar, Prof. G. S. Visweswaran, and Prof. Preeti Ranjan Panada for their valuable feedback and encouragement. I would also like to thank the staff members of Computer Science department especially Mr. S. D. Sharma and Mrs. Vandana Ahluwalia for their help. They have opened their hearts and their doors, supplying me with resources I never could have found without them.

No one walks alone on the journey of life. I devote my deepest gratitude to my father and mother for their unlimited love and support. They have encouraged me throughout the years of my study. The most special thanks belong to my wife, Vineeta, for her understanding about my leaving during all these years, selfless love and support all along and encouragement and company during the preparation of this thesis. I am also grateful to my kids, Aaradhya and Aashman, for their support to keep me happy and relaxed in many tense situations. I have to give a special mention for the support given by Sanjay, Sangeeta and Manish. Their worry about me inspired me to finish this research successfully as soon as possible. I warmly appreciate the generosity and understanding of my in-laws. My father-in-law and mother-in-law has supported even during the toughest period of their life.

Arun Parakh

Abstract

Researchers strive to develop new techniques and tools to efficiently and effectively utilize hardware resources that are made available for compute intensive applications. The Graphics Processing Unit (GPU) is one such hardware, which is very popular in the domain of High Performance Computing (HPC). Our work here presents a model to estimate performance of various applications on a modern GPU with Fermi architecture from NVIDIA. First we try to estimate computation time and then follow it up with an estimation of memory access time. We have found that our average estimation errors for these applications range from -7.76% to 55%. We have analyzed the existing Map-Reduce (MR) framework and enhanced the same for new GPU architectures. Our experiments show an average of 2.5x speedup of MR framework. We have achieved performance benefits ranging from 10% to 100% for various cache sizes. Based on the above analysis, three techniques have been developed for the performance enhancement of MR framework. Using these techniques, we have saved over 32% cache miss per thread. We have reduced the number of comparisons per thread in the group phase and gained an average of 1.5x speed up in the group phase. Use of delayed writing, reduces significant cache misses and improves the execution time by 10% to 25%. The extension of this work is to reduce the performance gap between MR and native implementation of applications in CUDA ($onlyCUDA$). This work reports deployment of three complete algorithms (SW, NB and BF) and achieves penalty reduction of 5x, 6.45x and 15.87x respectively. Finally, we try to execute applications with data size larger than memory capacity, on heterogeneous platforms. This data needs to be broken into partitions of different sizes and processed in parallel on different GPUs with different characteristics. We propose a technique to improve resource utilization, faster execution and also ensure simultaneous completion of such large data applications on heterogeneous platforms.

Contents

List of Figures

List of Tables

List of Abbreviations

AC	Accelerator Cluster
AMD	Advanced Micro Devices
BF	BlowFish Algorithm
BLAST	Basic Local Alignment Search Tool
BPM	Bank-level Partition Mechanism
CPU	Central Processing Unit
CSW	CALC1 subroutine from SWIM
CUDA	Compute Unified Device Architecture
DEGIMA	DEstination for Gpu Intensive MAchine
DES	Data Encryption Standard
DG	Distribute Grep
FDTD	Finite-Difference Time-Domain
FFT	Fast Fourier Transform
FPGAs	Field Programmable Gate Arrays
FPM	Functional Performance Model
GFLOPS	Giga Floating-point Operations per Second
GPGPU	General Purpose Computing on GPU
GPMR	GPU Map Reduce
GPU	Graphics Processing Unit
GSI	Gauss-Seidel Iterative Method
HG	Histogram
HPC	High Performance Computing
II	Inverted Index
IP	Internet Protocol
JIM	Jacobi Iterative Method
KFLOPS	Kilo Floating-point Operations per Second
KM	K-Means
KNN	K-nearest Neighbor Search
LRU	Last Recent Used

LT	Laplace Transformation
MFLOPS	Mega Floating-point Operations per Second
MFPM	multi-core Functional Performance Model
MM	Matrix Multiplication
MPI	Message Passing Interface
MR	Map-Reduce
MROH	Map-Reduce overheads
MRQ	memory request queue
MT	Matrix Transpose
MV	Matrix Vector
NB	N-Body Simulation
NBC	Naive Bayes Classifier
PCA	Principle Component Analysis
PCI	Peripheral Component Interconnect
PSS	Prefix Sum Scan
PTX	Parallel Thread Execution
PVC	Page View Count
PVR	Page View Rank
rCUDA	Remote CUDA
SHM	shared memory
SIMD	Single Instruction Multiple Data
SIMT	Single Instruction Multiple Thread
SIO	Sparse Integer Occurrence
SM	String Matching
SMV	Sparse Matrix-Vector
SMX	Streaming Multiprocessors
SORI	Successive Over Relaxation Iterative
SP	Saxpy
SRC	Speed Ratio Constant
SS	Similarity Scores
SW	Smith-Waterman Algorithm
TFLOPS	Tera Floating-point Operations per Second
URL	Universal Resource Locater
WC	Word Count