

MASTER'S THESIS

A multiple-precision integer arithmetic library for GPUs and its applications

Zhao, Kaiyong

Date of Award:
2011

[Link to publication](#)

General rights

Copyright and intellectual property rights for the publications made accessible in HKBU Scholars are retained by the authors and/or other copyright owners. In addition to the restrictions prescribed by the Copyright Ordinance of Hong Kong, all users and readers must also observe the following terms of use:

- Users may download and print one copy of any publication from HKBU Scholars for the purpose of private study or research
- Users cannot further distribute the material or use it for any profit-making activity or commercial gain
- To share publications in HKBU Scholars with others, users are welcome to freely distribute the permanent URL assigned to the publication

A Multiple-Precision Integer Arithmetic Library for GPUs and Its Applications

ZHAO Kaiyong

A thesis submitted in partial fulfillment of the requirements

for the degree of

Master of Philosophy

Principal Supervisor: Dr. Xiaowen CHU

Hong Kong Baptist University

January 2011

Abstract

Public-key encryption plays a critical role in our daily life. The core component of a public-key system is a set of multiple-precision integer operations. A server that relies on public-key encryption (such as an SSL server) needs to process a large number of multiple-precision integer operations, which require huge computing power. Recent advances in Graphics Processing Units (GPUs) open a new era of GPU computing. We are motivated by the fact that GPUs could be utilized to speed up multiple-precision integer operations. This is of practical importance to end users as well as application servers. However, it is not easy to achieve high performance on GPUs due to the complicated memory architecture and the relatively slow integer operations. In this thesis, we present our design, implementation, and experimental results on a highly optimized multiple-precision integer library, GPUMP. Our library achieved a significant speedup for a number of multiple-precision integer operations.

To show the effectiveness of GPUMP, we developed a practical and secure random linear network coding system, which can be applied in peer-to-peer networks and wireless networks in order to enhance the system throughput and robustness. Network coding systems are prone to pollution attacks because a single polluted data packet from a malicious peer will be encoded with other genuine data packets and propagated to the whole network at an exponential rate. Homomorphic hash functions have been proposed to defend the pollution attacks, but there remain two challenges: (1) Homomorphic hash function requires network coding be performed in $GF(q)$ where q is a very large prime number, which is computationally expensive due to the extensive large number operations; (2) Homomorphic hash function itself is computationally expensive for contemporary CPUs. By using the library of GPUMP, this thesis proposes to exploit the computing power of GPUs for network coding and homomorphic hashing, which leads to an integrated practical network coding solution for distributed systems.

Table of Contents

Declaration.....	i
Abstract	ii
Acknowledgement	iii
Table of Contents	iv
List of Figures	vii
List of Tables	x
Chapter 1. Introduction.....	1
1.1 Multiple Precision Arithmetic and Public Key System.....	1
1.1.1 Multiple Precision Arithmetic.....	1
1.1.2 Public Key System.....	1
1.2 CUDA Programming Background Knowledge.....	1
1.2.1 Graphic Processing Units.....	2
1.2.2 CUDA: a General-Purpose Parallel Computing Architecture	2
1.2.3 A Scalable Programming Model.....	3
1.2.4 CUDA Programming Model.....	4
1.2.5 Heterogeneous Programming.....	5
1.2.6 Memory Hierarchy.....	5
1.2.7 Hardware Implementation	6
1.2.8 SIMT Architecture	8
1.3 GPU Computing Application.....	9
1.3.1 GPU Computing in Finance.....	9
1.3.2 GPU Application in Biological Science.....	12
1.4 Objective.....	17
1.5 Contributions	18
1.6 Outline of the Thesis.....	18
Chapter 2. Example of GPU programming	20
2.1 Network Coding	20
2.2 Massively Parallel Network Coding.....	22
2.2.1 Encoding	22
2.2.2 Decoding.....	29
2.3 Experimental Results.....	30
2.3.1 Encoding Performance.....	32
2.3.2 Decoding Performance.....	33
2.4 Conclusions	34
Chapter 3. GPUMP Design.....	36
3.1 Multiple-precision Comparison, Addition and Subtraction.....	36
3.2 Modular Multiplication.....	39

3.3	Multiple-precision Montgomery Algorithm	41
3.4	Barrett Modular Reduction Algorithm	43
3.4.1	Analysis of Modular Reduction Algorithm	44
3.4.2	CIOS Montgomery Reduction	44
3.4.3	Karatsuba Montgomery Reduction	46
3.4.4	Improving the Montgomery Multiplication	47
3.5	Multiplicative Inversion	48
3.6	Modular Exponentiation	49
3.7	Exponentiation with Multi-Exponentiation	50
3.8	Exponentiation with Precomputation	51
3.9	Implementation and Optimization	52
3.9.1	Implementation	52
3.9.2	Experimental Montgomery Results	54
3.9.3	Conclusions	56
3.9.4	Experimental the Library Results	56
Chapter 4.	Parallel Network Coding in $GF(q)$ on GPUs	66
4.1	Introduction	66
4.2	Background and Related Work	67
4.2.1	Network Coding	67
4.3	Multiple Precision Modular Arithmetic	68
4.4	Parallel Network Coding on GPUs	68
4.4.1	Network Encoding	68
4.4.2	Network Decoding	69
4.5	Implementation Parallel Network Coding on GPUs	72
4.6	Experimental Results	73
4.6.1	Performance of Network Encoding	73
4.6.2	Performance of Network Decoding	74
4.7	Conclusions	78
Chapter 5.	Massively Parallel Homomorphic Hashing on GPUs	80
5.1	Introduction	80
5.2	Background and Related Work	81
5.2.1	Mathematical Preliminaries	81
5.2.2	Homomorphic Hash Functions	82
5.3	Parallel Homomorphic Hashing on GPUs	84
5.3.1	Homomorphic Hashing by Montgomery Multiplication with Precomputation	85
5.3.2	Implementation Details	88
5.4	Experimental Results	89
5.4.1	Evaluation of Montgomery Multiplication	90
5.4.2	Evaluation of Homomorphic Hash Function	91
5.5	Conclusions	92
Chapter 6.	Conclusion	94

References.....	96
Curriculum Vitae.....	100