



Ain Shams University
Faculty of Science
Department of Mathematics

Designing Fast Algorithms Using Parallel Models

Thesis

**Submitted for the partial fulfillment of
the requirements of the M.Sc. degree
in computer science**

Submitted by

Ibrahim Mohamed Abd Elmokatder Gad

Supervised by

Ass. Prof.

Sameh Sami Daoud

Department of Mathematics,
Faculty of Science,
Ain Shams University.

Ass. Prof.

Hazem Mohamed Bahig

Department of Mathematics,
Faculty of Science,
Ain Shams University.

Submitted to

**Department of Mathematics,
Faculty of Science,
Ain Shams University,
Cairo-Egypt**

2014

Contents

List of Tables	iv
List of Figures	v
Acknowledgements	vii
Abstract	viii
Summary	ix
Some of notations	xii
1 Introduction	1
2 Parallel Computation	4
2.1 Introduction	4
2.2 Classification of Parallel Architectures	8
2.2.1 Flynn's Taxonomy	8
2.2.2 Parallel Computer Memory Architectures	9
2.2.2.1 Shared Memory	9
2.2.2.2 Distributed Memory	10
2.2.2.3 Hybrid Distributed-Shared Memory	11
2.3 Parallel Algorithms and its complexity	12
2.3.1 Running Time	13
2.3.2 Number of Processors	14
2.3.3 Cost and Work	14
2.3.4 Speedup	15
2.3.5 Efficiency	15
2.3.6 Scalability	15
2.4 Parallel Programming Models	16
2.4.1 Shared Memory Model	16
2.4.2 Message Passing Model	17

2.4.3	Hybrid Model	19
2.4.4	Message Passing Interface (MPI)	19
2.5	Parallel Operating System	22
3	Integer Factorization Methods	23
3.1	Preliminaries	23
3.2	RSA	26
3.3	Factoring Algorithms	28
3.3.1	Special Purpose Algorithms	29
3.3.1.1	Trial Division	29
3.3.1.2	Pollard's Rho Algorithm	29
3.3.1.3	Pollard's $P - 1$ Algorithm	30
3.3.1.4	The Elliptic Curve Algorithm (EC)	31
3.3.1.5	The Special Number Field Sieve Algorithm (SNFS)	31
3.3.2	General Purpose Algorithms	32
3.3.2.1	Congruent Squares	32
3.3.2.2	The Quadratic Sieve (QS)	33
3.3.2.3	The General Number Field Sieve (GNFS)	34
3.3.2.4	Comparison between methods	35
3.4	The GNFS Algorithm	36
3.4.1	Polynomial Selection	36
3.4.2	Factor Bases	37
3.4.3	Sieving	40
3.4.4	Linear Algebra	43
3.4.5	Square Roots	47
3.4.6	Summary of GNFS Algorithm	47
3.5	An Example for the GNFS algorithm	49
3.5.1	Selecting the polynomial $f(x)$	49
3.5.2	Setting up factor bases	50
3.5.3	Sieving	51
3.5.4	Perfect Squares	53
3.5.5	Square roots	57
4	Parallel Line Sieve	58
4.1	Why Sieving Step	58
4.2	Serial Sieving	59
4.3	Previous Parallel Sieving	59
4.4	The New Methods	61
4.4.1	The First Method	62
4.4.2	The Second Method	62
4.5	Hardware and Software Programming Environment	64
4.6	Performance Evaluation	67
4.6.1	Test Cases	67

4.6.2	Timing Results	68
4.6.3	Speed-Up	68
4.6.4	Sieving Efficiency	72
4.7	Discussions	75
5	Conclusions	76
A	Algorithms for Division	78
A.1	Trial Division Algorithm	78
A.2	Pollard's Rho Algorithm	78
A.3	Pollard's $P - 1$ Algorithm	79
A.4	Lenstra's EC Algorithm	79
A.5	The Quadratic Sieve (QS) Algorithm	80
B	Compile and Run Parallel Program	82
B.1	Running a parallel program at Bibliotheca Alexandria	82
B.1.1	How to access the system ?	82
B.1.2	Compile a parallel program at Bibliotheca Alexandria	82
B.1.3	Run a parallel program at Bibliotheca Alexandria	84
	Bibliography	85

List of Tables

2.1	Flynn's taxonomy	8
2.2	The minimal set of MPI routines.	22
2.3	Shared memory and distribute memory.	22
3.1	The possible results for p and q dividing $s + r$ and $s - r$ [1].	33
3.2	Comparison of some factorization methods [2, 3, 4].	35
3.3	Choosing degree d of $f(x)$ [5, 6].	37
3.4	Rational Factor Base For $n = 45113$	50
3.5	Algebraic Factor Base For $n = 45113$	50
3.6	U : list of smooth pairs (a, b) found by sieving	52
4.1	GNFS integer factorization records [5]	59
4.2	Test cases and number of processors	67
4.3	The values of a and b for different n	67

List of Figures

2.1	Sequential computer	5
2.2	Solve a problem in sequential computer	5
2.3	Solve a problem in parallel computer	6
2.4	Shared memory	10
2.5	Distributed memory	11
2.6	Hybrid distributed-shared memory	12
2.7	Shared memory model	17
2.8	Message passing model	18
3.1	RSA Public-Key Cryptography	28
3.2	Generic procedure for factoring integer n	36
3.3	The five major steps of the GNFS algorithm	37
3.4	Two sieving arrays, $a + bm$ and $a + b\theta$, for certain b	42
3.5	The sieve step of the GNFS algorithm	43
3.6	Row vector for (a, b) in matrix X	45
4.1	Flow Chart for Serial Sieving	60
4.2	The flowchart of first method	63
4.3	The flowchart of second method	65
4.4	The parallel implementation for the first method $n = 61, 76, 80,$ 100, 110, 120, 130, 140 digits	69
4.5	The parallel implementation for the second method $n = 61, 76, 80,$ 100, 110, 120, 130, 140 digits	70
4.6	The parallel implementation for the first and the second method n $= 61, 76, 80, 100, 110, 120, 130, 140$ digits	71
4.7	The comparison between the first method, the second method, and the previous method $n = 61, 76$ digits	72
4.8	Sieve Speed-up for the first method and the second method $n = 61,$ 76, 80, 100, 110, 120, 130, 140 digits	73
4.9	Sieve efficiency for the first method and the second method $n = 61,$ 76, 80, 100, 110, 120, 130, 140 digits	74
B.1	Putty configuration	83
B.2	Compile steps	83
B.3	The path contain executable files	84

B.4 Submit the job file	84
-----------------------------------	----



Acknowledgements

Praise be to Allah who favored me with capability and patience to complete this work.

I would like to thank **Prof . Dr. Sameh S. Daoud** Ass. Professor Emeritus of Mathematics, Computer Science Division, Faculty of Science, Ain Shams University for guiding me throughout the thesis work. I am grateful for his kind help and valuable discussions. I would like to express my sincere thank and great gratitude to **prof. Dr. Hazem M. Bahig** Associate professor, Department of Mathematics, Faculty of Science, Ain Shams University for his continuous help, valuable discussion and supervisor. I always admire with his experience, quiet manner, and life logic.

I am also grateful *my family* specially my parents for their patience, understanding and encouragement. All appropriations for the academic staff of Computer Science, Department of Mathematics, Faculty of Science, Ain Shams University. Many thanks also go to the staff members of Department of Mathematics, Faculty of Science, Tanta University.



Abstract

RSA is one of the most important public key cryptosystems for information security. The security of RSA depends on integer factorization problem, it relies on the difficulty of factoring large integers. The General Number Field Sieve algorithm (GNFS) is currently the best known method for factoring large numbers over than 100 digits. The algorithm consists of five main steps. The third step, which is sieving step, takes the most time consumed in GNFS algorithm as proved in previous researches.

In this thesis, we improve the running time of this step by parallels it on a cluster system. We proposed the different techniques to improve the running time of this step. The experimental results have shown that the algorithm has achieved a good speedup and can be used for factoring a large integers.



Summary

Factoring is very important in the field of cryptography, specifically in the RSA cryptosystem. RSA algorithm is one of the most important public key cryptosystems for information security. RSA is used in real world applications such as: internet explorer, email systems, online banking, digitally signing, and encrypting email.

Due to the importance of this algorithm, cryptanalysts have been working for decades to identify weaknesses in the algorithm. The security of RSA depends on integer factorization problem, preciously relies on the difficulty of factoring large integers. Till now the problem of factoring integers is still not solved practically, which means there is no deterministic polynomial time algorithm for factoring integers. Much research has gone into the problem of factoring a large number. Due to advances in factoring algorithms and computing hardware the size of the number that can be factorized increases year by year.

There are many integer factorization algorithms used to factor large numbers. For example: Trial division, Pollards p-1 algorithm, Lenstra Elliptic Curve Factorization (ECM), Quadratic Sieve (QS) and General Number Field Sieve (GNFS) algorithm. The General Number Field Sieve algorithm (GNFS) is currently the best known method for factoring large composite numbers over than 100 digits.

Although the GNFS algorithm is efficient, it still takes a long time to factor a large integer such as an integer with 150-digits or larger. In order to reduce the execution time, one natural solution is to distribute jobs to parallel computers. The GNFS algorithm contains several time consuming steps. The most time consuming step is the sieving step which is used to generate enough relations. This step is very suitable for parallelization because the relation generations are independent. This thesis describes an implementation of the General Number Field Sieve (GNFS) algorithm using C language. This thesis, presents a new two methods for a parallel sieving step on GNFS implemented on a BA-cluster.

The thesis consists of five chapters, two appendices, and the bibliography.

Chapter One: It briefly shows the importance of cryptography and the RSA cryptosystem.

Chapter Two: This chapter introduces the basic concepts related to parallel computing. We give an overview for an introduction about why we need parallel computing, definition of parallel computers, motivation of parallel computing, applications area for parallel computing, and the keys that effect on parallelism. We give a brief description about the parallel architectures, also an introduction to parallel algorithms and the complexity measures for any parallel algorithm, the different types of parallel programming models. We also give a brief description about the library MPI that is used in the thesis.

Chapter Three: The aim of this chapter is to give an overview about integer factorization algorithms. We will focus on one of these algorithms that is used to factor large integers. We give mathematical definitions related to factorization problem. We give the importance of factorization problem in RSA, and different algorithms for factorization. We introduce, in details, one of the best algorithm for integer factorization which is the general number field sieve (GNFS). We give an example of the steps of GNFS algorithm.

Chapter Four: This chapter gives the two improved algorithms for sieving step in a cluster system. We give the reasons for selecting the sieve step as a main object for our research, and an overview for serial sieve step. We give an overview for previous works for parallel sieve step. We propose new method for parallel sieve step on a cluster system. Also we describe the configuration of hardware and software, used to implement the parallel sieving step on the cluster system, the experimental results for the proposed methods are given.

Chapter Five: This chapter gives the conclusions and the expected future work that will improve the performance of GNFS.

Appendix A: It includes algorithms for division.

Appendix B: It includes the steps for how to run a parallel program at BA cluster.

