**AIN SHAMS UNIVERSITY**
**FACULTY OF ENGINEERING**
**Computer and Systems Engineering**

# Parallel Acceleration of Bioinformatics Algorithms

A Thesis submitted in partial fulfillment of the requirements of
Master of Science in Electrical Engineering
(Computer and Systems Engineering)

by
**Mai Ahmed Mahmoud Said**
Bachelor of Science of Electrical Engineering
(Computer and Systems Engineering)
Faculty of Engineering, Ain Shams University, 2011

Supervised By
**Prof. Dr. Ayman Mohamed Mohamed Hassan Wahba**
**Dr. Mona Mohamed Hassan Safar**

Cairo, 2017

**AIN SHAMS UNIVERSITY**
**FACULTY OF ENGINEERING**
**Computer and Systems Engineering**

# Parallel Acceleration of Bioinformatics Algorithms

by

**Mai Ahmed Mahmoud Said**

Master of Science in Electrical Engineering
(Computer and Systems Engineering)
Faculty of Engineering, Ain Shams University, 2017

**Examiners' Committee**

| **Name and affiliation** | **Signature** |
|---|---|
| **Prof.Dr. Amr Ahmed Nabil Ahmed El-Kadi**<br>Department Of Computer Engineering<br>Faculty of Engineering, The American University in Cairo. | ...................... |
| **Prof.Dr. Ashraf Mohamed Mohamed El-Farghly Salem**<br>Computer and Systems Engineering<br>Faculty of Engineering, Ain Shams University. | ...................... |
| **Prof.Dr. Ayman Mohamed Mohamed Hassan Wahba**<br>Computer and Systems Engineering<br>Faculty of Engineering, Ain Shams University. | ...................... |

Date: 24 January 2017

# Statement

This thesis is submitted as a partial fulfillment of degree of Master of Science in Computer and Systems Engineering, Faculty of Engineering, Ain shams University. The author carried out the work included in this thesis, and no part of it has been submitted for a degree or a qualification at any other scientific entity.

**Mai Ahmed Mahmoud Said**

Signature

..................................................................................

**Date: 24 January 2017**

# Researcher Data

**Name: Mai Ahmed Mahmoud Said**

**Date of Birth: 11/12/1988**

**Place of Birth: Cairo, Egypt**

**Last academic degree: Bachelor of Science**

**Field of specialization: Computer and Systems Engineering**

**University issued the degree: Ain Shams University**

**Date of issued degree: June 2011**

**Current job: Teaching assistant**

# Thesis Summary

## Summary

Biological sequences alignment is an urgent algorithm applied in bioinformatics to explore the functional similarities between two sequences based on their similar structures. Protein sequences alignment is an essential operation used in multiple applications such as; structure based drug design, evolution tracing, prediction of future gene mutation and possible sources of diseases, and classifying newly discovered species.

The sizes of biological sequences databases are growing intensively urging researchers to design accelerated sequences alignment solutions. Recent studies suggest using GPU computation in accelerating such algorithms. GPU computation model suits such data intensive operations.

In this thesis, we present a survey on various GPU accelerated sequences alignment solutions, comparing them with respect to the algorithm stages implemented on the GPU, the interaction between GPU and CPU execution paths, the parallelization granularity applied, the memory structures allocation design and other effective aspects.

We propose a GPU-CPU multithreaded solution accelerating a protein sequences alignment program; PSI-BLAST. Our work addresses accelerating the iterative protein alignment tool, PSI-BLAST, for its higher accuracy results compared to other protein sequences alignment programs.

PSI-BLAST algorithm is divided to seven stages. First, user inputs are parsed including the query sequence and the database of sequences to be aligned. Next, the look up table filling stage is performed to reduce the time needed later to perform the alignment of each database sequence. Then, the alignment is performed in three stages applied to each database sequence. The first stage is seeds collection, where we find word sized alignments between the query and the database sequence. The second and third stages extend these alignments using ungapped then gapped extension. Alignments that have scores above a certain threshold are saved. The traceback stage recomputes the alignments generating bookkeeping information. Finally, the output is formatted following a user option format. The algorithm is repeated in iterations computing a complex scoring matrix based on the results of the previous iteration.

The three stages that align database sequences with the query sequence are performed on each subject sequence in a totally independent manner. Thus, accelerating these stages using parallel approaches is appealing. Each subject sequence can be aligned using different processing unit.

In our design we followed the Assess Parallelize Optimize Deploy (APOD) design cycle proposed by NVIDIA. In the first stage, we assessed PSI-BLAST algorithm by applying a profiling study. The study aims at allocating the areas of improvement in the algorithm. Seeds collection and ungapped extension stages are found to be the most time consuming stages in the algorithm. Our design parallelizes these stages by GPU and CPU multithreading.

Various optimization techniques were applied to the design including choosing a suitable role assignment model for GPU threads. We chose the granularity of one thread to align each database sequence, since database sequences alignment is totally independent. Database sequences are sorted according to their size in a preprocessing step allowing threads of the same warp to align sequences of comparable lengths. Warp threads paths divergence is reduced as threads perform tasks of almost the same time, thus threads need not wait for each others to finish execution. At the beginning of GPU kernel execution, shared memory structures are copied from the global memory using the coalesced memory accession feature. Optimization of shared memory structures sizes is applied quartering the required memory.

The design was built using CUDA toolkit 6.5 and deployed on an NVIDIA TESLA C2070 GPU card of 448 cores running on 1030 GFLOPs. The GPU card is installed alongside an Intel Xeon E5-2609 six-core processor with 2.4 GHz clock speed and 8 GB RAM. Our solution achieves a speedup of around 4X compared to the sequential PSI-BLAST.

The thesis is divided into six chapters as listed below:

Chapter 1 is an introduction explaining the motivation for protein sequences alignment. We present a brief description of the various categories of the biological sequences algorithms. Also, a brief discussion about the use of GPUs nowadays in general purpose computing. Finally, a demonstration of the main thesis contributions is presented.

Chapter 2 explains the concepts of sequences alignment. We describe the basics of biological genes sequencing, starting with the biochemical approaches. Then, we present a description of various sequences alignment algorithms. We describe the BLAST algorithm stages in more details. Then, we emphasize the differences between protein BLAST tools. We highlight the accuracy upgrade inferred by PSI-BLAST algorithm.

Chapter 3 is a survey of the related solutions proposed recently. First, A review of the various BLAST accelerated solutions is discussed. Then, we review the use of the GPU

in accelerating sequence alignment algorithms recently. Finally, an analysis of the recent solutions accelerating BLAST programs using GPU parallel computing is presented.

Chapter 4 proposes our solution GPU-PSI-BLAST, a heterogeneous GPU-CPU acceleration of the protein alignment algorithm PSI-BLAST. The chapter starts with a description of the GPU programming design model proposed by NVIDIA. Then, we present a time profiling of PSI-BLAST algorithm showing the stages that form the bulk of execution of PSI-BLAST runs. Then, a flow chart of our solution describing the load distribution among threads and the parallelization granularity we adopted is presented. Finally, we describe our design memory architecture, showing memory structures placement, optimization efforts and other memory handling operations.

Chapter 5 presents a set of experiments we have applied to measure the speed and accuracy of our solution comparing it with other related solutions. We describe the benchmark data set we used in our experiments. Then, we define the time measurement methodology we used to get accurate timing measurements. We present a comparison of the speedup results of our solution and the multithreaded NCBI solution. We present various experiments investigating the effect of changing the database partitioning size, and the scalability of our solution when using different number of iterations of PSI-BLAST. Other presented experiments include a justification of our choice to accelerate PSI-BLAST rather than accelerating other protein alignment tools.

Chapter 6 is a brief conclusion of our work. We present multiple proposals for the possible future work extending this work.

Key words: BLAST, bioinformatics, BLASTP, computing, GPU, parallel, parallel architecture, PSI-BLAST, sequence alignment

# Acknowledgment

All praise is due to Allah, Most Merciful, the Lord of the Worlds, Who taught man what he knew not. It is my pleasure to express my deepest gratitude to my supervisors, Professor. Dr. Ayman Wahba, Dr. Mona Safar for their continued guidance, effort, and encouragement. Last but not Least, I cannot find words to thank my wonderful mother for her unlimited love, care and support, my father,who has always been my shelter and support, and my sister; my journey companion, Sara for her unconditional love and care.

Mai Ahmed Mahmoud Said
Computer and Systems Engineering
Faculty of Engineering
Ain Shams University
Cairo, Egypt
January 2017

# Contents

**Contents** xiii

**List of Figures** xvi

**List of Tables** xviii

**Abbreviations** xix

**1 Introduction** 1

1.1 Motivation for Protein Sequences Alignment Acceleration . . . . . . 1
1.2 Sequences Alignment Algorithms . . . . . . . . . . . . . . . . . . . . 2
1.3 General Purpose GPU Programming . . . . . . . . . . . . . . . . . . 3
1.4 Thesis Contribution . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
1.5 Thesis Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

**2 Basic Local Alignment Search Tool** 8

2.1 Biological Sequences Alignment . . . . . . . . . . . . . . . . . . . . . 8
2.1.1 Biochemical Means of Genes Sequencing . . . . . . . . . . . 9
2.1.2 History of Sequences Databases . . . . . . . . . . . . . . . . 11
2.1.3 Sequences Formats . . . . . . . . . . . . . . . . . . . . . . . 11
2.1.4 Importance of Sequences Alignment . . . . . . . . . . . . . . 11
2.1.5 Sequences Alignment Algorithms . . . . . . . . . . . . . . . 12
2.2 BLAST . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
2.2.1 BLAST Algorithm Stages . . . . . . . . . . . . . . . . . . . 16
2.2.1.1 Inputs parsing . . . . . . . . . . . . . . . . . . . . 17
2.2.1.2 Lookup Table Filling . . . . . . . . . . . . . . . . . 18
2.2.1.3 Seeds Collection . . . . . . . . . . . . . . . . . . . 18
2.2.1.4 Ungapped Extension . . . . . . . . . . . . . . . . . 19
2.2.1.5 Gapped Extension . . . . . . . . . . . . . . . . . . 19
2.2.1.6 Traceback and Output Formatting . . . . . . . . . 20
2.2.1.7 Output Formatting . . . . . . . . . . . . . . . . . . 20
2.2.2 Example of BLAST Steps . . . . . . . . . . . . . . . . . . . 23
2.3 Protein BLAST Programs . . . . . . . . . . . . . . . . . . . . . . . . 26
2.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

**3 Related Work** 31

3.1 BLAST Accelerated Solutions . . . . . . . . . . . . . . . . . . . . . . 31

3.2 GPU Parallel Computing Model . . . . . . . . . . . . . . . . . . . . 33
3.2.1 GPU Memory Architecture and Work Distribution Paradigm 34
3.2.2 GPU Accelerated Sequence Alignment Algorithms . . . . . . 37
3.3 GPU Accelerated BLAST Solutions . . . . . . . . . . . . . . . . . . 37
3.3.1 Accelerated Implementations . . . . . . . . . . . . . . . . . . 38
3.3.2 Accelerated BLAST Programs . . . . . . . . . . . . . . . . . 39
3.3.3 BLAST Stages Performed by the GPU Kernel . . . . . . . . 39
3.3.4 GPU-CPU Interaction . . . . . . . . . . . . . . . . . . . . . 40
3.3.5 Load Assignment on the GPU . . . . . . . . . . . . . . . . . 41
3.3.6 Data Structures Hosting on GPU Memory Types . . . . . . 41
3.3.7 Speedups Rough Comparison . . . . . . . . . . . . . . . . . 42
3.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

**4 A Cooperative Load Balanced GPU-CPU PSI-BLAST Accelerator 45**
4.1 GPU Programming Model . . . . . . . . . . . . . . . . . . . . . . . 46
4.2 Time Profiling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
4.3 GPU-CPU Execution Flow . . . . . . . . . . . . . . . . . . . . . . . 50
4.4 Memory Architecture . . . . . . . . . . . . . . . . . . . . . . . . . 54
4.4.1 Memory Mapping . . . . . . . . . . . . . . . . . . . . . . . . 54
4.4.2 Memory Economizing . . . . . . . . . . . . . . . . . . . . . . 56
4.4.3 Memory Checking . . . . . . . . . . . . . . . . . . . . . . . . 57
4.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

**5 Experimental Results 59**
5.1 Measurement Data Set . . . . . . . . . . . . . . . . . . . . . . . . . 60
5.2 Time Measurement Methodology . . . . . . . . . . . . . . . . . . . 60
5.3 Database Partitioning . . . . . . . . . . . . . . . . . . . . . . . . . 61
5.4 GPU-PSI-BLAST versus Multithreaded PSI-BLAST . . . . . . . . 62
5.4.1 Varying Length Queries Speedup Comparison . . . . . . . . 64
5.5 GPU-PSI-BLAST versus NCBI-GPU-BLASTP . . . . . . . . . . . 66
5.6 GPU-PSI-BLAST Scalability across Multiple Iterations . . . . . . . 68
5.7 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69

**6 Conclusion and Future Work 71**
6.1 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
6.2 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
6.3 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73

**Bibliography 75**