

Ain Shams University
Faculty of Computer & Information Sciences
Computer Systems Department



Solving Computationally Intensive Problem Using GPUs- Solving Motif Finding Problem as a Case Study

A Thesis submitted to Computer Systems Department,
Faculty of Computer and Information Sciences, Ain Shams University,
in partial fulfillment of the requirements for
the Ph.D. of Science in Computer and Information Sciences

By

Mirvat Mahmoud Ahmed Al-Qutt

M.Sc. in Computer and Information Sciences,
Teaching Assistant at Computer Systems Department,
Faculty of Computer and Information Sciences, Ain Shams University.

Under Supervision of

Prof. Dr. Hossam El-Deen Mostafa Fahim

Prof. of Computer Systems,
Department of Computer Systems,
Faculty of Computer and Information Sciences, Ain Shams University.

Dr. Rania Abdel Rahman El Gohary

Associate Professor,
Department of Information Systems,
Faculty of Computer and Information Sciences, Ain Shams University.

Dr. Heba Khaled Ahmed Mahmoud

Assistant Professor,
Department of Computer Systems,
Faculty of Computer and Information Sciences, Ain Shams University.

Acknowledgement

First and foremost, thanks to Allah, the most merciful, the most graceful, for His great help throughout this work.

I sincerely acknowledge and express my appreciation to prof. Dr. Hossam Faheem for his valuable advice during the study and fulfillment of this thesis and for his support to get an access to High Performance Computing Center in King Abdulaziz University (Aziz Supercomputer) to run the work described in this thesis.

I owe my deepest gratitude and appreciation to my supervisors for their great care, valuable advices, and helpful guidance to carry out this work.

Thanks also are extended to my colleague in the Computer System department Mahmoud Fayez for his kind support and help.

My special gratitude is due to my beloved late father, my warmhearted mother, my all of my beloved family members who supported me throughout this work with all their love and faith, without their encouragement and understanding it would have been impossible for me to finish this work.

Mirvat Al-Qutt.

Abstract

With scientific problems getting larger and more complicated, level of parallelization needed to be increased to provide the needed amount of computational power, so more adequate, efficient and robust parallel environments became an essential need for scientific research. This thesis proposes hybrid parallel paradigms to solve computationally intensive problem, taking into consideration one of the most common problems in Bioinformatics, Motif Finding as a case study. Motifs are defined as the short patterns these consist of nucleotide; they are usually positioned close to the genes contained in Deoxyribo Nucleic Acid (DNA). They occur in a frequent manner within the sequence, these patterns are not identical but it comes with some mutations in several of their nucleotide positions. Motif Finding Problem aims to discover unknown motifs that are expected to be common in a set of sequences. Generally it can be viewed as a large-length sequence matching problem.

There are a numerous number of algorithms available to solve this problem; these can either be exact or approximate. Motif finding problem is categorized as one of the most computationally intensive problems in the field of bioinformatics and it requires a large amount of memory. It has been categorized as Nondeterministic Polynomial Time Order Problem. Both software and hardware accelerators have been proposed and deployed to accelerate Motif finding problem algorithms, Software based acceleration solutions are easier to implement and do not require hardware experience.

This research is kicked-off by exploring the existing algorithms and parallelization techniques for accelerating Motif Finding problem and identifying the main points that could represent a contribution.

In this research, SKIP brute-force algorithm is accelerated using different High Performance Computing implementations. The results showed that GPU significantly reduced the execution time and improved the performance better than using Multicore architecture. In Addition, it has been noticed that different sequence lengths affect the speedup and power consumption. These implementations are considered a step towards building a complete parallel architecture for solving computationally intensive problems of bioinformatics.

Consequently, machine learning techniques have been exploited to develop an intelligent recommender system that advices the optimal HPC architecture for specific Motif finding problem size and other parameters. A neural network-based multi-objective optimization approach is employed (Neural Network Inversion), which is used for direct problem approximation by mean of a neural network. The objective functions are maximizing the speedup ratio and minimizing the power consumption. The importance of this system is clear as it employs an automatic decision regarding optimal number of processors in terms the optimal hardware configuration that can efficiently solve computationally intensive problems taking into consideration the resources availability and a set of requirements and environment restrictions that might be conflicting sometimes. The proposed system achieved prediction accuracy reached over 89 % with “390” iterations on average for CPU based hardware configuration prediction system, and 87% with “306” iterations on average for GPU based hardware configuration prediction system.

Finally, considering the solution scalability, cloud platform services have been explored for storage and analytics purposes, taking into consideration the huge volume of data, tremendous cloud storage platforms that currently exist and provide a scalable, distributed storage

service. Therefore, the challenges of implementing Motif Finding based on employing the services provided by the cloud storage platform are addressed. In this research, Apache Hadoop is picked for Big Data processing which is an open source Platform that provides enormous number of clusters that is used for parallel implementations. MF solution was implemented using two different Big Data frameworks: MapReduce and Apache Spark, they have different implementation schemes and resources usage plans. The results are collected and analyzed against different parameters such as the speedup value and power consumption. From experiments Spark achieves much more speedup than MapReduce. This finding is expected due to MapReduce Input/Output operations overheads. The main purpose of this step is to accomplish effective parallelization and evaluates the performance of such an integrated solution.

Table of Contents

| | |
|--|------|
| <i>Acknowledgement</i> | ii |
| <i>Abstract</i> | iii |
| List of Figures | viii |
| List of Tables | xi |
| List of Abbreviations..... | xiv |
| Chapter 1: Introduction | |
| 1.1 Overview | 2 |
| 1.2 Motif Finding as a Case-Study..... | 3 |
| 1.3 Aim | 6 |
| 1.4 Motivation..... | 6 |
| 1.5 Objectives..... | 8 |
| 1.6 Research Points of Investigation | 9 |
| 1.7 Contribution | 10 |
| 1.7 Thesis Organization..... | 11 |
| 1.8 Summary..... | 12 |
| Chapter 2: Literature Review | |
| 2.1 Overview | 14 |
| 2.2 Motif Finding Algorithms..... | 14 |
| 2.2.1 Word-Based Methods..... | 16 |
| 2.2.2 Probability-Based Methods | 17 |
| 2.3 Related Work..... | 19 |
| 2.4 System Specification and Analysis | 22 |
| 2.5 Summary..... | 24 |
| Chapter 3: Proposed System for Solving Motif Finding Problem | |
| 3.1 Overview | 26 |
| 3.2 High Performance Computing Paradigms | 27 |
| 3.3 Programming Models | 29 |
| 3.4 Proposed HPC Implementation..... | 32 |
| 3.4.1 CPU Based Implementation Using (OpenMP-MPI)..... | 34 |
| 3.4.2 GPU Based Implementation Using (CUDA-MPI) | 36 |
| 3.5 Experimental Results | 47 |
| 3.6 Summary..... | 54 |
| Chapter 4: A Proposed System for Predicting Optimal Hardware Configuration for Motif Finding Acceleration Based on Neural Network Inversion | |
| 4.1 Overview..... | 57 |

| | |
|--|----|
| 4.2 Background | 57 |
| 3.4 Proposed Prediction System for Optimal Hardware Configuration | 60 |
| 4.4 Experimental Results | 64 |
| 4.5 Summary | 73 |
| Chapter 5: A Proposed MapReduce and Spark Solutions for Motif Finding Problem | |
| 5.1 Overview | 75 |
| 5.2 Background | 75 |
| 5.2.1 MapReduce | 79 |
| 5.2.2 Spark | 80 |
| 5.3 Motif Finding Acceleration on Hadoop Cluster | 82 |
| 5.3.1 MapReduce Based Motif Finding Solution | 83 |
| 5.3.2 Spark Based Motif Finding Solution | 85 |
| 5.4 Experimental Results | 87 |
| 5.5 Summary | 91 |
| Chapter 6: Conclusion and Future Work | |
| 6.1 Conclusion | 93 |
| 6.2 Future Work | 94 |
| <i>References</i> | 97 |

List of Figures

| Figure | Caption | Page |
|--------|--|------|
| 1.1 | (MF) Problem Definition | 5 |
| 1.2 | Planted (l,d) (MF) Problem Definition | 6 |
| 2.1 | SKIP Brute-Force Algorithm | 23 |
| 3.1 | The most common paradigms of HPC systems | 28 |
| 3.2 | point-to-point MPI | 30 |
| 3.3 | point-to-many MPI | 30 |
| 3.4 | An example of the file's structure of 20 sequences each is 1200 nucleotide | 33 |
| 3.5 | MPI-OpenMP implementation of the MFP using the CPUs based architecture | 35 |
| 3.6 | Multi-Core CPUs based architecture | 36 |
| 3.7 | Kepler GK110 Full chip block diagram Each SMX has 192 single precision CUDA cores, 64 double precision units, 32 special function units (SFU), and 32 load/store units (LD/ST). | 39 |
| 3.8 | CUDA implementation of the MFP solution using the GPUs based architecture | 43 |
| 3.9 | Pseudo code for CUDA kernel function that runs on the GPU | 43 |

| | | |
|------|---|----|
| 3.10 | Flowchart of CUDA implementation of the MFP solution using the GPUs based architecture | 45 |
| 3.11 | Flowchart of CUDA kernel function that runs on the GPU | 46 |
| 3.12 | Score Function Example | 46 |
| 3.13 | Execution Time for Solving MFP on different parallel paradigms | 48 |
| 3.14 | Number of sequences and Sequences' lengths against Number of comparisons to solve (MF). | 49 |
| 3.15 | Execution Time for Solving (MF) on different number of K20 GPUs CUDA cores. | 50 |
| 3.16 | Power Consumption for Solving MFP by different parallel paradigm | 53 |
| 4.1 | ANN model for functions optimization | 58 |
| 4.2 | Proposed prediction system | 61 |
| 4.3 | Pareto front construction steps | 63 |
| 4.4 | Fully connected ANN | 65 |
| 5.1 | HDFS Architecture | 79 |
| 5.2 | MapReduce Architecture | 80 |
| 5.3 | Proposed MapReduce Architecture | 84 |
| 5.4 | Map Function | 85 |

| | | |
|-----|---|----|
| 5.5 | Spark Implementation | 86 |
| 5.6 | MapReduce and Spark Speedup values for different data set values 2 cluster nodes number | 90 |
| 5.7 | MapReduce and Spark Speedup values for different data set values 5 cluster nodes number | 90 |
| 5.8 | MapReduce and Spark Speedup values for different data set values 8 cluster nodes number | 91 |

List of Tables

| Table | Title | Page |
|-------|--|------|
| 1.1 | The Research Points of Investigation. | 9 |
| 1.2 | Relation Matrix of Publication Contributions and Research Points of Investigation | 11 |
| 3.1 | Generated sequences | 33 |
| 3.2 | The Tesla K20 GPU (GK110) Configuration | 37 |
| 3.3 | Minimum number of motif space batches needed for different MFP sizes | 41 |
| 3.4 | Execution Time for Different batches number for solving MFP of size (T=240,N=25 and l=15) | 42 |
| 3.5 | Execution Time for Solving (MF) on different parallel paradigms | 48 |
| 3.6 | Number of Comparisons needed for Solving MFP for various sequence lengths. | 49 |
| 3.7 | Execution Time for Solving (MF) on different number of K20 GPUs CUDA cores. | 50 |
| 3.8 | Execution Time and Power Consumption of Sequential SKIP-brute force | 51 |
| 3.9 | Average speed up of implementing SKIP-brute force for Solving (MF) on different parallel paradigms | 51 |
| 3.10 | Power consumption for different number of comparisons on different parallel architectures | 53 |

| | | |
|-------|--|----|
| 4.1.1 | 1x24 CPUs architecture execution time and power consumptions for MFP (15,4) problems of different size | 66 |
| 4.1.2 | 2x24 CPUs architecture execution time and power consumptions for MFP (15,4) problems of different size | 67 |
| 4.1.3 | 4x24 CPUs architecture execution time and power consumptions for MFP (15,4) problems of different size | 67 |
| 4.1.4 | 8x24 CPUs architecture execution time and power consumptions for MFP (15,4) problems of different size | 68 |
| 4.1.5 | 16x24 CPUs architecture execution time and power consumptions for MFP (15,4) problems of different size | 68 |
| 4.2.1 | 1xk20 GPU architecture execution time and power consumptions for MFP (15,4) problems of different size | 69 |
| 4.2.2 | 2xk20 GPU architecture execution time and power consumptions for MFP (15,4) problems of different size | 69 |
| 4.3 | The regression accuracy against the number of epochs and number of neurons in hidden layers for CPU implementation. | 70 |
| 4.4 | The regression accuracy against the number of epochs and number of neurons in hidden layers for GPU implementation. | 70 |
| 4.5 | The prediction accuracy and the number or required iterations to predict the optimal CPU based hardware configuration of different function inverse methods. | 72 |
| 4.6 | The prediction accuracy and the number or required iterations to predict the optimal GPU based hardware configuration of different function inverse methods. | 72 |
| 5.1 | MapReduce and Spark Speedup values for acceleration of motif finding problem on different cluster nodes number | 89 |

List of Abbreviations

| | |
|------|---|
| ANN | Artificial Neural network |
| API | Application Program Interface |
| CPU | Central Processing Unit |
| CUDA | Computer Unified Device Architecture |
| DNA | Deoxyribo Nucleic Acid |
| FPGA | Field-Programmable Gate Array |
| GPU | Graphics Processing Unit |
| HDFS | Hadoop Distributed File System |
| HPC | High Performance Computing |
| HW | Hardware |
| IaaS | Infrastructure as a Service |
| IT | Information Technology |
| MEME | Multiple Expectation Maximization for Motif Elicitation |
| MFP | Motif Finding Problem |
| MIC | Intel Many Integrated Cores |
| MIMD | Multiple Instruction Multiple Data |
| ML | Machine Learning |

| | |
|-------------|--|
| MOOF | Multi-Objective Optimization Function |
| MPI | Message Passing Interface |
| MSE | Mean Square Error |
| MSP | Median String Problem |
| NBP | Nonamer-Binding Protein |
| NGS | Next Generation Sequencing |
| NP-Complete | Nondeterministic Polynomial Time Order Problem |
| NUMA | None Uniform Memory Access |
| OpenHMPP | Open Hybrid Multicore Parallel Programming |
| PaaS | Platform as a Service |
| PMP | Planted motif problem |
| PMSE | Power Mean Square Error |
| PSSM | Position Specific Scoring Matrix |
| RDBMS | Relational Database Management Systems |
| RDD | Resilient Distributed Datasets |
| RNA | RiboNucleic Acid |
| SaaS | Software as a Service |
| SKIP BF | SKIP Brute Force |