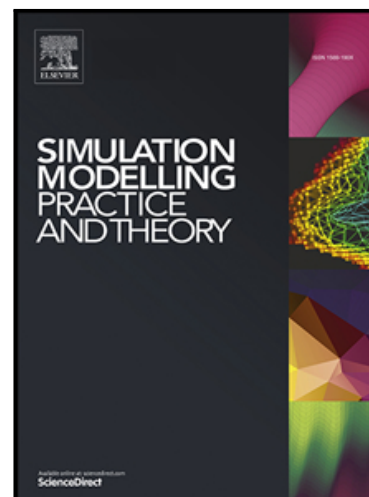


Accepted Manuscript

GPU Parallelization of Sequence Segmentation Using Information Theoretic Models

Luay Alawneh, Emad Rawashdeh, Mahmoud Al-Ayyoub, Yaser Jararweh

PII: S1569-190X(18)30056-X
DOI: [10.1016/j.simpat.2018.04.007](https://doi.org/10.1016/j.simpat.2018.04.007)
Reference: SIMPAT 1804



To appear in: *Simulation Modelling Practice and Theory*

Received date: 11 February 2018
Revised date: 30 April 2018
Accepted date: 30 April 2018

Please cite this article as: Luay Alawneh, Emad Rawashdeh, Mahmoud Al-Ayyoub, Yaser Jararweh, GPU Parallelization of Sequence Segmentation Using Information Theoretic Models, *Simulation Modelling Practice and Theory* (2018), doi: [10.1016/j.simpat.2018.04.007](https://doi.org/10.1016/j.simpat.2018.04.007)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

GPU Parallelization of Sequence Segmentation Using Information Theoretic Models

Luay Alawneh^a, Emad Rawashdeh^b, Mahmoud Al-Ayyoub^b, Yaser Jararweh^b

Jordan University of Science and Technology, Irbid, Jordan

^a*Department of Software Engineering*

^b*Department of Computer Science*

Abstract

Sequence segmentation has gained popularity in bioinformatics and particularly in studying DNA sequences. Information theoretic models have been used in providing accurate solutions in the segmentation of DNA sequences. Existing dynamic programming approaches provide optimal solution to the segmentation problem. However, their quadratic time complexity prohibits their applicability to long sequences. In this paper, we propose a parallel approach to improve the performance of a quasilinear sequence segmentation algorithm. The target segmentation technique is a divide-and-conquer recursive algorithm that is based on information theory principles and models. We present three parallel implementations that aim at reducing the segmentation time. The first implementation uses the multithreading capabilities of CPUs. The second one is a hybrid implementation that utilizes the synergy between the CPU and the multithreading power of GPUs. The third implementation is a variation of the hybrid approach where it utilizes the concept of unified memory between the CPU and the GPU instead of the standard memory copy approach. We demonstrate the applicability of the parallel implementations by testing them on real DNA sequences and randomly generated sequences with different lengths and different number of unique elements. The results show that the hybrid CPU-GPU approach outperforms the sequential implementation with a speedup of up to 5.9X while the CPU parallel implementation provides a poor speedup of only 1.7X.

Keywords: CUDA, OpenMP, Recursive Sequence Segmentation, Unified Memory

1. Introduction

The analysis and understanding of sequential data are very crucial in various applications and domains such as bioinformatics, stock markets, telecommunications, and weather forecasting. However, mining of sequential data has become very cumbersome due to the huge amounts of data which limits the performance of traditional data mining techniques [1]. One of the solutions to enhance the performance of data mining techniques is parallel processing. Parallel processing is a computing strategy for speeding up the execution of computationally complex problems by dividing the problem into smaller tasks that run in parallel on different processing units. It emerged as a consequence of the inability of single processing to meet the requirements of highly complex scientific and commercial problems.

Recently, a new parallel processing power has emerged from contemporary Graphics Processing Units (GPUs) [2]. GPUs, which were originally specialized in 3D graphics processing, are capable of executing thousands of threads simultaneously due to the large number of cores which improves the parallelization efficiency. Thus, in many applications, GPUs outperform the parallel processing power of modern CPUs, which makes them more practical in solving complex problems in bioinformatics, simulation and modeling, and many high performance computing (HPC) applications [3, 4, 5]. One of the most famous parallel computing platforms and programming models for GPUs is Compute Unified Device Architecture (CUDA) [6]. CUDA has been widely used to enhance the performance of many algorithms in data mining and various research areas [7, 8, 9].

Email addresses: lmalawneh@just.edu.jo (Luay Alawneh), efrawashdeh13@cit.just.edu.jo (Emad Rawashdeh), maalshbool@just.edu.jo (Mahmoud Al-Ayyoub), yijararweh@just.edu.jo (Yaser Jararweh)

Download English Version:

<https://daneshyari.com/en/article/6902493>

Download Persian Version:

<https://daneshyari.com/article/6902493>

[Daneshyari.com](https://daneshyari.com)