ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Energy efficiency of sequence alignment tools—Software and hardware perspectives

Michał Kierzynka [a,b,*], Lars Kosmann [c], Micha vor dem Berge [d], Stefan Krupop [d], Jens Hagemeyer [e], René Griessl [e], Meysam Peykanu [e], Ariel Oleksiak [a,b]

[a] Poznań Supercomputing and Networking Center, Poland
[b] Poznań University of Technology, Poland
[c] OFFIS, Oldenburg, Germany
[d] christmann informationstechnik + medien GmbH & Co. KG, Germany
[e] CITEC, Bielefeld University, Germany

HIGHLIGHTS

- A comparative study of pairwise sequence alignment tools is presented.
- Considerable differences in energy efficiency between individual tools are reported.
- The proposed FPGA implementation outperforms other tools on energy efficiency front.
- Newly developed RECS® |Box hardware is presented with its monitoring infrastructure.

ARTICLE INFO

Article history:
Received 13 August 2015
Received in revised form
30 April 2016
Accepted 4 May 2016
Available online xxxx

Keywords:
Sequence alignment
Energy efficiency
FiPS project
Heterogeneous hardware
Bioinformatics
FPGA

ABSTRACT

Pairwise sequence alignment is ubiquitous in modern bioinformatics. It may be performed either explicitly, e.g. to find the most similar sequences in a database, or implicitly as a hidden building block of more complex methods, e.g. for reads mapping. The alignment algorithms have been widely investigated over the last few years, mainly with respect to their speed. However, no attention was given to their energy efficiency, which is becoming critical in high performance computing and cloud environment. We compare the energy efficiency of the most established software tools performing exact pairwise sequence alignment on various computational architectures: CPU, GPU and Intel Xeon Phi. The results show that the energy consumption may differ as much as nearly 5 times. Substantial differences are reported even for different implementations running on the same hardware. Moreover, we present an FPGA implementation of one of the tested tools—G-DNA, and show how it outperforms all the others on the energy efficiency front. Finally, some details regarding the special RECS® |Box servers used in our study are outlined. This hardware is designed and manufactured within the FiPS project by the Bielefeld University and christmann informationstechnik + medien with a special purpose to deliver highly heterogeneous computational environment supporting energy efficiency and green ICT.

1. Introduction

Sequence alignment is one of the most common and the most frequently applied operations in computational biology. This statement becomes even more authorized if we realize that many higher level algorithms, e.g. sequence mapping or phylogenetic tree construction, use this simple operation as a building block. There are a number of algorithms performing the alignment procedure, both heuristic and exact. The former were often used in the past when limited computational power was the main limiting factor. On the other hand, exact algorithms have become more popular in the recent years, mainly with the advent of high performance software implementations. However, the exponential growth of the number of sequences in databases and from individual experiments is still posing a real challenge, especially in the context of next-generation sequencing (NGS). Moreover, the increase in the amount of data is faster than the

* Corresponding author at: Poznań Supercomputing and Networking Center, Poland.
E-mail address: michal.kierzynka@man.poznan.pl (M. Kierzynka).

http://dx.doi.org/10.1016/j.future.2016.05.006
0167-739X/

rate of improvement of microprocessors. Therefore, scientists are constantly working on ways to improve the existing software tools. This also includes the research associated with different hardware architectures which was deeply explored in the recent years. The most popular accelerating hardware in this area is probably GPU (graphics processing unit), with multiple implementations of different alignment scenarios, e.g. [1–5]. Other hardware architectures that were found useful include: FPGAs (field-programmable gate arrays) [6,7], IBM's Cell BE [8,9], Intel Xeon Phi [10] alongside with traditional CPUs (central processing units) with their SIMD capabilities [11,12].

One common goal of all the high-throughput implementations of sequence alignment is obviously to maximize the performance. Therefore, scientists tend to compare their software tools in this respect. However, from a data center or cloud provider perspective it is also crucial to maximize the performance achieved per energy used, i.e. the energy efficiency of the software. From this standpoint, sequence alignment is a prime example to look at. First, because it is ubiquitous in modern bioinformatics analysis, which nowadays is carried out more frequently in high performance computing (HPC) centers. Second, as there are already multiple high quality implementations of this building block algorithm on multiple architectures. In this paper we compare the energy efficiency of various state-of-the-art pairwise sequence alignment tools alongside with their pure performance. The results may be of key importance not only for those managing data centers, but also for scientists implementing complex processing pipelines and programmers starting to consider their platform of choice. One of the goals of this work is also to raise awareness within the community about the energy-efficiency driven development and tools selection.

An important aspect is also the fact that this research was conducted as part of FiPS—an EU-founded project entitled: "Developing Hardware and Design Methodologies for Heterogeneous Low Power Field Programmable Servers". The main goal of the project is to develop highly heterogeneous low power servers (called RECS®|Box) for HPC centers and cloud applications, with a special emphasis placed onto FPGA modules. Alongside with the hardware, the project partners have developed tools helping the programmers to port their applications to heterogeneous environment [13]. Interestingly, the hardware is developed in close collaboration with application partners, who adapt their domain-specific software tools to achieve more energy-efficient implementations. One of the applications that are considered in the project is G-DNA [5]—a GPU-based software for pairwise sequence alignment. We investigate whether any improvement in its energy efficiency is possible, given that the application was already highly optimized for GPU. In particular, an FPGA implementation of this tool is proposed. Since the comparative results are very promising, the article also presents some of the experience and results achieved in this context. Furthermore, selected details regarding the RECS®|Box hardware design are outlined too. This hardware deserves special attention as it facilitates development of energy efficient applications due to its hardware configuration and software monitoring solutions.

The rest of the paper is organized in the following way. Section 2 shortly outlines the basic idea behind sequence alignment algorithms. Section 3 first briefly presents the software tools for pairwise sequence alignment that are compared in our study, and then refers to literature related to energy efficiency. Section 4 presents the methodology and results of the comparative study. Section 5 describes the FPGA implementation of G-DNA as well as details regarding the architecture of energy efficient RECS®|Box hardware. Finally, conclusions are outlined in Section 6.

2. Dynamic programming algorithms

There are three basic sequence alignment methods based on the dynamic programming, namely: the Needleman–Wunsch algorithm (NW) [14] for global alignment, its semi-global version, and the Smith–Waterman algorithm (SW) [15] for local alignment. These algorithms work in a similar way, and differ only with respect to the boundary conditions. Additionally, each algorithm may compute only the so called alignment score, i.e. quantitative information about the similarity of sequences, or the full alignment by performing the backtracking step. Moreover, the algorithms may use either linear or affine gap penalties. This section explains the basic idea behind these methods on the example of NW with affine gap penalties. For more detailed analysis of alignment algorithms see [3].

2.1. The Needleman–Wunsch algorithm

Let us first define the following notation:

- A—an alphabet, i.e. a set of characters (nucleotides or amino acids),
- $s_i(k) \in A$—kth character of the ith sequence,
- $SM(s_i(k) \in A, s_j(l) \in A)$—substitution value for a given pair of characters,
- G_{open}, G_{ext}—gap opening and gap extension penalties,
- H—a matrix with partial alignment scores,
- E, F—auxiliary matrices with partial alignment scores indicating vertical and horizontal gap continuation, respectively.

The Needleman–Wunsch algorithm fills the dynamic programming matrix H according to a similarity function expressed in Formula (1). The matrix is of size $n+1 \times m+1$, where n is the number of characters in the first sequence s_1 and m—in the second sequence s_2. The similarity function is based on a score of substitution between any two characters which is typically defined by a *substitution matrix*. Gap penalties, i.e. G_{open} and G_{ext}, are applied to reflect the cost associated with insertion/deletion mutations.

$$H_{i,j} = \max \begin{Bmatrix} E_{i,j} \\ F_{i,j} \\ H_{i-1,j-1} + SM(s_1(i), s_2(j)) \end{Bmatrix} \tag{1}$$

$$E_{i,j} = \max \begin{Bmatrix} E_{i,j-1} - G_{ext} \\ H_{i,j-1} - G_{open} \end{Bmatrix} \tag{2}$$

$$F_{i,j} = \max \begin{Bmatrix} F_{i-1,j} - G_{ext} \\ H_{i-1,j} - G_{open} \end{Bmatrix} \tag{3}$$

where $i = 1..n$ and $j = 1..m$. Additionally, the first row and the first column of matrix H are filled according to the following formulas:

$$H_{0,0} = 0 \tag{4}$$

$$H_{i,0} = -G_{open} - (i-1) \cdot G_{ext} \tag{5}$$

$$H_{0,j} = -G_{open} - (j-1) \cdot G_{ext} \tag{6}$$

where $i = 1..n$ and $j = 1..m$. Moreover, the first rows and columns of matrices E and F are initialized with $-\infty$.

At this point the optimal alignment score is already known and can be found in cell $H(n, m)$. However, the actual alignment of the two sequences, i.e. the relative arrangement of subsequent characters, is not known yet. This may be computed by the optional *backtracking* procedure which performs backward moves starting from $H(n, m)$ until the first cell, i.e. $H(0, 0)$, is reached. A single move is performed to the neighboring cell that contributed to maximum value in Formula (1). If the algorithm moves diagonally, two characters are aligned. If the move is performed to the upper cell, a gap character is inserted into sequence s_1 in the alignment. In a similar way a gap is added to sequence s_2 every time the algorithm moves to the left.

Please cite this article in press as: M. Kierzynka, et al., Energy efficiency of sequence alignment tools—Software and hardware perspectives, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.05.006