

Optimizing Traceback in the Smith-Waterman Algorithm for GPUs

LeAnn Lindsey
leann.lindsey@utah.edu
University of Utah
Salt Lake City, Utah, USA

Hari Sundar
Advisor
University of Utah
Salt Lake City, Utah, USA

Muaaz Awan
Advisor
Lawrence Berkeley National Labs
Berkeley, California, USA

ABSTRACT

The traceback phase of the Smith-Waterman (SW) algorithm requires a significant amount of memory and introduces an irregular memory access pattern which makes it challenging to be performant on GPU architectures. We developed an efficient traceback algorithm, using diagonal major indexing and binary representation of the traceback matrix. This implementation was integrated into the ADEPT [2] sequence alignment library. To demonstrate its efficacy in high performance software, we integrated our traceback implementation into Metahipmer2 [3], which is a widely used metagenomic assembler. Our implementation is 3.6x faster than traceback in GASAL2 [1], and 51x faster than traceback in Striped Smith Waterman [5], the current state-of-the-art SW libraries on GPU and CPU respectively. It sped up the final alignment step in Metahipmer2 by an average of 44% and improved the overall execution time of MHM2 by an average of 13%.

CCS CONCEPTS

• **Applied computing** → **Computational genomics**.

KEYWORDS

gpu, metagenomics, memory hierarchy, Smith-Waterman

ACM Reference Format:

LeAnn Lindsey, Hari Sundar, and Muaaz Awan. 2022. Optimizing Traceback in the Smith-Waterman Algorithm for GPUs. In *Proceedings of ACM Conference (SC'20)*. ACM, New York, NY, USA, 3 pages.

1 INTRODUCTION

Sequence alignment is fundamental to many bioinformatics algorithms and is essential for studying evolution, genetic diversity, and for the development of vaccines. Advancements in genome sequencing technologies have resulted in exponential growth of genomic data. High performance algorithms are needed to make large scale analysis of this data possible.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SC'20, Nov 2022, Dallas, TX, USA

© 2022 Association for Computing Machinery.

2 BACKGROUND

2.1 Smith-Waterman

The Smith-Waterman algorithm (SW) is a dynamic programming algorithm that finds the optimal local sequence alignment between two genetic sequences. It has a time and space complexity of $O(mn)$. SW with affine gap scoring uses three matrices, E , F , and H , to calculate the score according to formulas listed below.

$$E_{i,j} = \max \begin{cases} E_{i,j-1} + G_{ext}, \\ H_{i,j-1} + G_{init} \end{cases} \quad F_{i,j} = \max \begin{cases} F_{i-1,j} + G_{ext}, \\ H_{i-1,j} + G_{init} \end{cases} \quad (1)$$

$$H_{i,j} = \max \begin{cases} E_{i,j}, \\ F_{i,j}, \\ H_{i-1,j-1} + S(q_i, r_i) \\ 0 \end{cases} \quad (2)$$

Where G_{init} is the gap initiation score, G_{ext} is the gap extension score and $S(q_i, r_i)$ is the match or mismatch score between residues q_i and r_i . We refer you to the original ADEPT [2] paper for a more thorough description of the details of the algorithm.

2.2 Graphical Processing Units

GPUs have been used to significantly accelerate many scientific computing algorithms, but they also come with constraints. Data must be transferred to and from the device, and both shared and global memory on the device is extremely limited. To optimize performance, consideration must be given to how these resources are used.

3 IMPLEMENTATION

Binary Representation of Scoring Matrix

To perform traceback, pointers back to the cell that gave the maximum value must be recorded. Each cell in the E and F matrices can contain only two pointers, and thus can be represented in 1 bit of data. There are 4 possible pointers in the H matrix, which can be represented in 2 bits of data. In order to minimize global memory writes, we compress the values of all three matrices for one cell into the lower 4 bits of one byte of data, resulting in a 3x space reduction.

Diagonal Major Indexing of the Scoring Matrix

The pattern of dependencies in the SW algorithm allows parallelism to progress only along the minor diagonals [2] resulting in a memory access pattern along the diagonals. Normally a matrix would be stored in memory using either row-major or column-major indexing, which would result in many uncoalesced memory accesses on GPU. To tackle this, we restructured the matrix so that it is laid out

in memory with "diagonal" major indexing. This type of indexing requires maintaining a lookup table to keep track of diagonals and an offset variable to keep track of element within a diagonal.

4 RESULTS

We compared our traceback algorithm with the Striped-Smith-Waterman [5], SeqAn3 [4], and Gasal2 [5] Libraries.

4.1 Library Comparisons

4.1.1 Traceback Time. We showed a 3.6x speed up over Gasal2 and a 51x speed up over the SSW implementation.

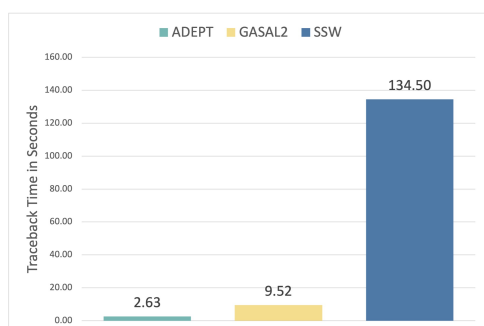


Figure 1: Traceback Time Comparison

4.1.2 Overall Execution Time. On the DNA data sets, we showed a 10.5x average speed up over SeqAn3 and an 11.1x average speed up over SSW. On the protein data sets we showed an average speed up of 6.8x over SeqAn3 and a 11.8x speedup over SSW.

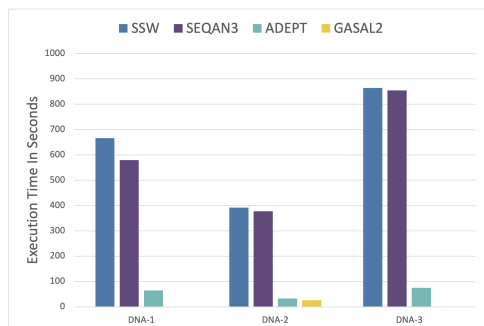


Figure 2: Overall Execution Time DNA

4.2 Metahipmer2 Integration

Metahipmer2 (MHM2) is a high performance de novo metagenomic short read assembler. In the optional final step, all reads are aligned to the contigs. Previously MHM2 used the SSW-Library on CPU to complete this last step.

ACKNOWLEDGMENTS

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy



Figure 3: Overall Execution Time Protein

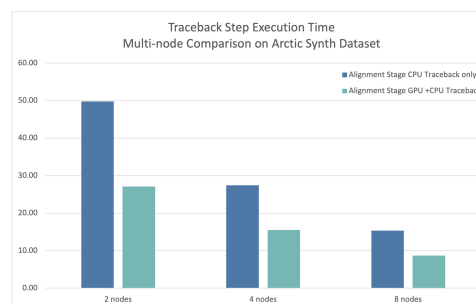


Figure 4: Average of 44% Performance Improvement in the Final Alignment Step of MHM2

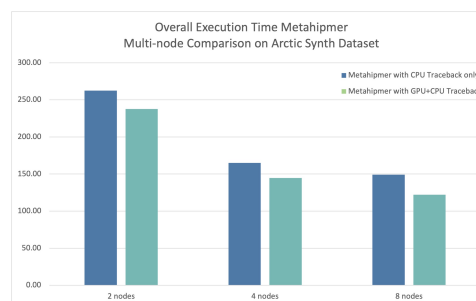


Figure 5: Average of 13% Overall Performance Improvement of MHM2

organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative. This research was also funded by National Science Foundation grants 1808652 and 2008772.

REFERENCES

- [1] Nauman Ahmed, Jonathan Lévy, Shanshan Ren, Hamid Mushtaq, Koen Bertels, and Zaid Al-Ars. 2019. GASAL2: a GPU accelerated sequence alignment library for high-throughput NGS data. *BMC Bioinformatics*, 20, 1, (Oct. 2019), 520. doi: 10.1186/s12859-019-3086-9.

- [2] Muaaz G. Awan, Jack Deslippe, Aydin Buluc, Oguz Selvitopi, Steven Hofmeyr, Leonid Oliker, and Katherine Yelick. 2020. ADEPT: a domain independent sequence alignment strategy for gpu architectures. eng. *BMC bioinformatics*, 21, 1, (Sept. 2020), 406. doi: 10.1186/s12859-020-03720-1.
- [3] Muaaz Gul Awan, Steven Hofmeyr, Rob Egan, Nan Ding, Aydin Buluc, Jack Deslippe, Leonid Oliker, and Katherine Yelick. 2021. Accelerating large scale de novo metagenome assembly using GPUs. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (SC '21). Association for Computing Machinery, New York, NY, USA, (Nov. 2021), 1–11. ISBN: 978-1-4503-8442-1. doi: 10.1145/3458817.3476212.
- [4] Andreas Döring, David Weese, Tobias Rausch, and Knut Reinert. 2008. SeqAn an efficient, generic C++ library for sequence analysis. eng. *BMC bioinformatics*, 9, (Jan. 2008), 11. doi: 10.1186/1471-2105-9-11.
- [5] Michael Farrar. 2007. Striped Smith–Waterman speeds database searches six times over other SIMD implementations. *Bioinformatics*, 23, 2, (Jan. 2007), 156–161. doi: 10.1093/bioinformatics/btl582.