Multiple Pairwise Sequences Alignments with Needleman-Wunsch Algorithm on GPU

Da Li
Dept. of Electrical and Computer Engineering
University of Missouri
Columbia, Missouri, USA
dlx7f@mail.missouri.edu

Michela Becchi
Dept. of Electrical and Computer Engineering
University of Missouri
Columbia, Missouri, USA
becchim@missouri.edu

***Abstract*—Pairwise sequence alignment is the method to find the best matching piece for two sequences. It can be either local which return the most similar subsequence or global which return the alignment of whole sequence. Nowadays, the number of genetic sequences increases exponentially so that it becomes a challenge to analyze and understand those data. The cutting-edge parallel architectures provide lots of opportunities to ace the problem. The Graphical Processing Unit (GPU) is becoming an appealing choice for accelerating these processes. In this paper, we will explore multiple pairwise sequence alignment with Needleman-Wunsch algorithm, which is the fundamental method for global alignment. Our proposed method levers the parallelism between different pairs and void the unnecessary data copy to achieve 2X speedup for data transfer and 2.9X acceleration ratio for kernel execution.**

***Index Terms*—sequence alignment, GPU.**

I. Introduction

Sequence alignment is one of the fundamental problems in bioinformatics. Needleman-Wunsch algorithm [1] and Smith-Waterman algorithm [2] are two widely used approaches for pairwise alignment. The former is for global alignment which measures the similarity between two sequences while the latter is aimed at find the longest similar piece within two sequences. These two algorithms are very alike. Basically, a 2-D matrix will be filled with different scores according to some rules and the whole process goes from up-left to the bottom-right. Taking Needleman-Wunsch algorithm as an example, each element in the score matrix will be filled according to equation (1). In the equation, the M(i,j) means the value in ith row and jth column of score matrix. G is the penalty score for a gap and S(x_i,y_j) is the substitution score of current position. The 1st row and 1st column of the score matrix will be initialized and the value in remaining position is determined by the values in its left, up and diagonal elements.

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + S(x_i, y_j) \\ M(i-1,j) + G \\ M(i,j-1) + G \end{cases} \quad (1)$$

Researchers have proposed some implementations of Needleman-Wunsch algorithm on different parallel architecture [3-5]. The most famous one on GPU is from Rodinia [5] which is a heterogeneous bench mark.

In this paper, we propose a practical implementation of Needleman-Wunsch on GPU for multiple pairwise alignments. Compared with existing solution, the contributions of our paper lie on the following aspects:

- Support multiple pairwise sequence alignments at the same time which is the common case in practice
- Perform initialization on GPU
- Optimize memory usage to avoid unnecessary data transfer
- Support alignment between the sequences in different lengths

II. Motivation

The CUDA implementation of Needleman-Wunsch in Rodinia benchmark is the first parallel implementation on GPU. The optimized final version can achieve an 8.0x speedup over single thread CPU code [3]. The score matrix is processed in a diagonal strip manner from top-left to bottom-right. This implementation utilizes a hierarchical parallelism (grid level and thread-block level) to reduce global memory access and kernel launch overhead.

After carefully analyze the CUDA implementation in Rodinia, we find there are several drawbacks:

A. Practical Issues

Due to increasing amount of sequences in genetic database, usually the operations are querying and comparing multiple pairwise alignments concurrently. There is rare chance that only one pair of sequences will be considered. The fact of multiple pairwise alignments can introduce another dimension of parallelism since there are no dependences between different pairs. Also, the sequences are usually in different length while the CUDA implementation in Rodinia only supports sequences in the same length so that padding is needed.

B. Communication overhead

in Rodinia's implementation, three data transfers are existed. Before kernel launch, the score matrix and reference matrix will be copied from CPU to GPU. And after processed by GPU, the score matrix should be copied back from GPU to CPU. If the sequences in one pairwise alignment are in length N and M separately, the matrix will be in size of N*M.

actually, the first two data transfers can be avoided. For score matrix, the 1st row and 1st column should be initialized. If the initialization can be performed on GPU, there is no need to copy score matrix. Also, for the reference matrix, we don't need one in the same size as score matrix. It can be encoded

This work has been supported by National Science Foundation grant CNS-1216756, and by equipment donation from Nvidia Corporation.

into a small table which is square of character set size and by indexing we can get the value when producing the score matrix.

C. Kernel Launch overhead and Configuration

The CUDA programming model doesn't support global synchronization between blocks. The implementation in Rodinia has two levels of parallelism and in the grid level, the global synchronization is needed. This leads to multiple kernel launches from the host side. Also, to achieve maximum occupancy, only 16 threads are configured for each block [3] which is against the fact that in GPU 32 threads are grouped into one warp and perform an SIMT behavior. Half of the threads in one warp are wasted.

III. Design and Implementation

Our proposed method is aimed at multiple pairwise alignments at the same time. So each pair of sequences will be assigned to one block and within the block, multiple threads will process the score matrix in diagonal strip manner. Figure 1 shows how each of score matrices is processed in different block. Four pairs are processed in four different blocks. They are independent to each other.

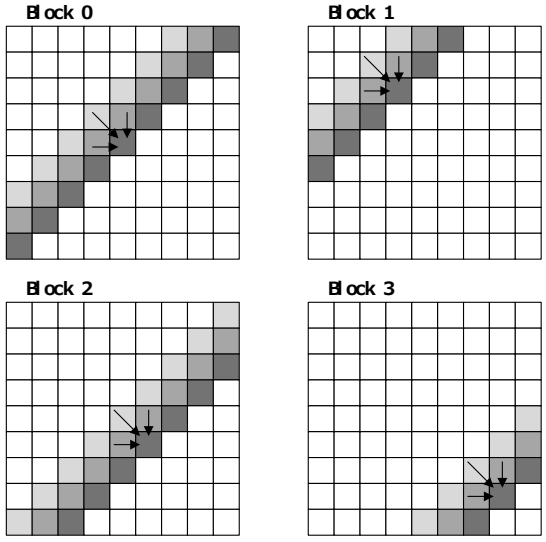

Fig. 1. Four pairs are processed in different GPU blocks concurrently

To utilize the shared memory, two diagonal lines of data will be loaded from global memory to shared memory. According to equation (1), using these two diagonal lines, the following diagonal line of data can be calculated and then stored to shared memory and global memory. After that, the first diagonal line of data can be discarded and the shared memory can be used for the next diagonal line of data. So to calculate one score matrix, only three lines of shared memory are needed. Two of them will hold the previous results and the remaining one will store the new results.

IV. Experimental Results

We implement our method with CUDA 4.0 runtime API and test it on GeForce GTX 480. The GPU consists of 480 CUDA cores (15 Multiprocessors and 32 CUDA Cores/MP). The GPU clock speed is 1.40GHz and the memory clock rate is 1.848 GHz with 384-bit bandwidth. The length of sequences is 2000 which is the common case. Figure 2 and 3 show the comparison of data transfer time and kernel execution time between Rodinia implementation and our approach. The data transfer time is roughly half of the original one in Rodinia and the kernel execution achieves at most 2.9X speed up.

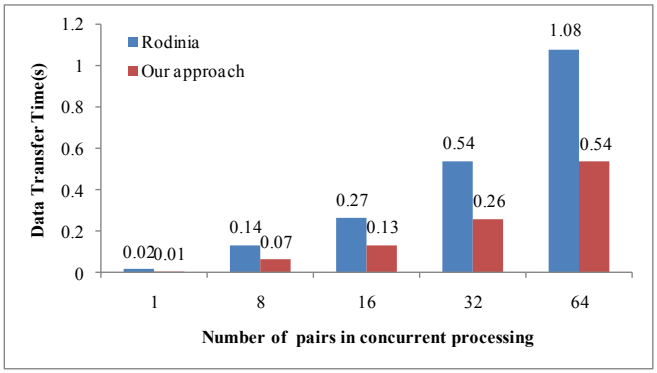

Fig 2. Comparison of data transfer time

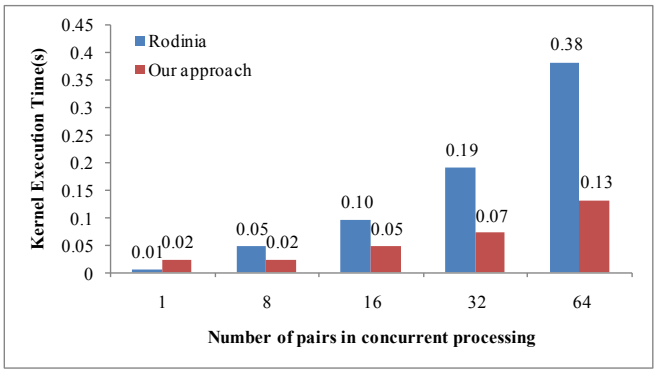

Fig 3. Comparison of kernel execution time

V. Conclusion

We implement a practical GPU version of Needleman-Wunsch that supports multiple pairwise sequence alignment at the same time. The results show that both the data transfer time and the kernel execution time can be reduced to roughly 1/2 of the previous implementation.

References

[1] Saul B. Needleman, Christian D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", Journal of Molecular Biology 48 (3): 443-53, 1970.

[2] Temple F. Smith, Michael S. Waterman, "Identification of Common molecular Subsequences", Journal of Molecular Biology 147: 195-197, 1981.

[3] Tahir Naveed, Imitaz Saeed Siddiqui, Shaftab Ahmed, "Parallel Needleman-Wunsch Algorithm for Grid", Proceedings of the PAK-US International Symposium on High Capacity Optical Networks and Enabling Technologies(HONET 2005), Islamabad, Pakistan, 2005.

[4] Shuai Che, Jie Li, Jeremy W. Sheaffer, Kevin Skadron, John Lach, "Accelerating Compute-Intensive Applications with GPUs and FPGAs", SASP, 2008.

[5] Shuai Che, Michael Boyer et al, "Rodinia: A Benchmark Suite for Heterogeneous computing", In Proceedings of the IEEE International Symposium on Workload Charactertization (IISWC), pp. 44-54. 2009.