Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Accelerating Smith-Waterman Alignment Based on GPU

Komal B. Deshmukh
Computer Department, MET BKC
Savitribai Phule Pune University, India,

M. U. Kharat
Computer Department, MET BKC
Savitribai Phule Pune University, India,

Abstract - Bioinformatics is interdisciplinary research area. In various genome projects, huge biological sequences are available and analysis of these sequences is fundamental operation in Bioinformatics. The ultimate goal behind this is to find the similarity region based on comparing method is called as sequence alignment. Sequence alignment for large biological sequence often consumes most of processing time.

Recently several attempts are made to use parallel processing capabilities of GPU, to offload the time consuming sequence alignment task. This paper describes implementation of Smith-Waterman (SW) algorithm on the Graphics Processing Unit (GPU) with parallel scan approach, aiming to accelerate SW alignment. Also optimization technique are introduces like reduce accessing global memory so that it can helps to reduce a latency time. Proposed system developed using Compute Unified Device Architecture (CUDA), which provides by nvidia GPU. In sequences alignment two methods are reported that are local alignment and global alignment. Smith-Waterman algorithm is providing optimal local alignment which has quadratic time and space complexity. In several real time applications huge sequences are aligned to discover structural, functional and evolutionary characteristics of newly generated sequence. To evaluate performance, proposed system uses sequence retrieved from the NCBI site. Proposed algorithm implemented on NVIDIA GTX 680 Graphics card using CUDA.

Keywords -Bioinformatics, sequence alignment, Smith- waterman (SW) algorithm, GPU, CUDA

I. INTRODUCTION

In recent years, biological data explosion has occurred and also a great acceleration in the accumulation of biological knowledge. Collecting and analyzing complex biological data with the help of computer technology is nothing but Bioinformatics. Bioinformatics is one of integrative and important research area. It is interdisciplinary field includes computer science, biology, mathematics and statistics. It means standing with foot in the two worlds that is world of computer science and world of biological science. This is the phenomenon of collecting and analyzing complex biological data that uses computer technology for management of huge genetic data. The data is in terms of biological sequence. Computers are used to gather, store, integrate and analyze the large scale biological data and genetic information. The emphasis is on the make use of of computers because most of the part in genetic data analysis are highly monotonous or mathematically multifaceted. General activities in Bioinformatics include mapping and analyzing sequences, aligning sequences to compare them in order to discover function, structure and evolutionary information of particular sequence [1].

The alignment of biological sequences is most important component in the field of bioinformatics. The ultimate goal is to discover the similarity between different sequences based on comparing method is called as sequence alignment. If there is any new sequence is discovered, then the newly generated sequence is compared with other sequences available in genome database. Similarity search procedure is executed by an alignment process between query sequence and targeted sequences. The obtained similarity degree can be used to infer the functionality or evolution history of the sequences. The best possible way to achieve the optimal result of similarity between sequences is using Smith-Waterman algorithm which based on dynamic programming method [15].

Sequence comparison is one of the key operations in Bioinformatics. The outcome of sequence comparison is measured in terms of score and alignment. Score is nothing but the similarity between the sequences. That is it provides one integer number indicate that how much similarity between two sequences and alignment highlights the similarity between two sequences.

In various real time application alignments are reported in two ways that are local and global alignment. A global method is aiming from end-to-end of the sequences and there are two methods that functional as global method; Dot Plot and Needleman-Wunsch (NW) algorithm meanwhile, another alignment also carried two methods which are local alignment method. The methods are known as an exact method like Smith-Waterman algorithm and heuristic based approximate method like FASTA and BLAST. In local alignment method, both methods are attempted to identify the most similar region between pair or more sequences [15].

Smith-Waterman algorithm is providing optimal local alignment which has quadratic time and space complexity. This algorithm uses the dynamic programming approach. Using this approach it creates the dynamic programming matrices called DP (Dynamic Programming) matrix. There are some restrictions related to SW algorithm. One of the important restrictions related to SW algorithm is that it requires quadratic space to store DP matrices. And another restrictive property is it requires quadratic time complexity. So far many efforts have been taken to reduce the time and space complexity of SW algorithm.

In today's era the conventional method of improving the performance of CPUs is accomplished by escalating the number of computing cores and enlarging the size of cache. In current generation traditional CPUs are built with four to five cores but present generation of graphic cards, e.g. Nvidia GeForce contain hundreds of computing cores. The performance of these cards higher than that of conventional CPUs and they and are capable of performing large scale computations for all problems where data parallel approach is applicable. Recently several attempts are made to use parallel processing capabilities of GPU, to offload the time consuming task. Algorithms for number of suitable problems have been already implemented in many case and showing good results. The examples from different disciplines, such as quantum engineering , chemistry, fluid dynamics, astrophysics, computer science and many more .In the similar way to increase the efficiency of biological sequences alignment operation in our system we make use of Nvidia Graphical Processing Unit(GPU) that helps to accelerate sequence alignment operation. This paper describes implementation of Smith-Waterman (SW) algorithm on the Graphics Processing Unit (GPU) with parallel scan approach, aiming to accelerate SW alignment [1]

The paper is organized as follows. Section II describes related methods of sequence analysis. In section III we describe CUDA programming language along with nvidia GTX680 GPU. Section IV describe basic Smith-Waterman algorithm. Section V describes proposed methodology for parallel scan approach on SW algorithm. In section VI we discuss our Dataset and performance comparison of SW alignment and conclusion in section VII

A. Sequence Analysis – Score and Alignment

Sequence Comparison is core function of Bioinformatics analysis. When the two sequences are compared, as output result we get score and alignment between the two sequences. Fig. 1 shows how to align two sequences.[1]

Fig. 1 Sequence Alignment.

Score is measure of similarity between two sequences. Consider two sequences as seq1 and seq2, while calculating score need to consider following values for each instance of sequence..

1. val= +1, if both character of sequence are match (ma= +1)
2. val= -1, if both character of sequence are not match. (mi= -1)
3. val= -2 , if one of the character is space (G=-2)

Total score between the two sequences are calculated by summing up all val of each instance. Another measure of sequence analysis is Alignment. It basically highlights the similarity between two sequences. Sequences can be aligned locally or globally.

- Global Alignment - In this method two sequences are assume to be of same length. Two sequences are aligned from beginning to the end of sequence s to find the best alignment. This approach provides very precise solution by considering whole length of sequence. But in some cases global alignment technique is insufficient because there is need to compare smaller sequence with larger sequence. Myers-Millers Algorithm is used to get global alignment.
- Local Alignment - This method does not assume to be two sequences are of same length because at the time alignment it considers the local region of sequence and not whole sequence. It only finds local regions which has highest level of similarity between two sequences without considering remaining part of sequences. The two sequences to be aligned can be varying in lengths. Smith-Waterman algorithm is used to find out the local alignment.[15]

II. RELATED WORK

There are various techniques are available for retrieving score and alignment between biological sequences. The result of analysis used to discover functional, structural, evolutionary characteristic. The various biological sequence are huge in nature and for analysis of such huge sequences can be done by various algorithm like Smith-Waterman algorithm, Myers-millers algorithm, Fast LSA, CUDAlign 1.0, CUDAlign 2.0. To increase the efficiency of these algorithm there is use of GPU along with CPU in these algorithm.

T. F. Smith and M. S. Waterman [2] proposed Smith Waterman Algorithm, used to retrieves the optimal score and local alignment between two sequences. This algorithm totally based on Dynamic programming approach. Gotoh [3] modified the SW algorithm to include affine gap penalties. Because using a general form of gap penalty function slows down the algorithm, an affine gap penalty function is preferred. When using affine gap penalty function in dynamic programming, it only needs to differentiate between the case that the gap is first being introduced and the case that the

gap is being extended. Hirschberg [4] presented an algorithm which solve the problem of quadratic time and space complexity of SW algorithm and proposed a linear space algorithm to compute the alignment of Longest Common Subsequence (LCS). Myers and Millers (MM)[5] algorithm that calculate the optimal global alignment in linear space. This algorithm of sequence alignment based on the Hirschberg's algorithm which uses divide and conquer procedure. The idea is to find the midpoint of Longest Common Sequence (LCS).

S. Aluru, N. Futamura, and K. Mehrotra [6] presented parallel algorithm which is based on prefix computations and handles pair wise comparison of biological sequences and able to handle constant as well as affine gap penalty, full-sequence and subsequence matching of sequences. Fast LSA [7] is nothing but variant in MM algorithm that uses divide and conquer method. It performs parallelization in linear space.

Graphical Processing Unit (GPU) is one in which many cores are available that allows parallel execution of task. Several algorithms were design based on GPU helps to accelerate alignment of biological sequences. Y. Liu, W. Huang, J. Johnson and S. Vaidya [8] present hardware implementation of double affine Smith Waterman (DASW) algorithm uses dynamic programming and implemented on a commodity graphics card. Other GPU based implementation [9], [10], [11] speed up alignment task but one of the restrictive characteristic of these algorithm that they cannot align two huge sequences.

CUDAlign 1.0 [12] algorithm able to handle large huge sequences to retrieve optimal score but not alignment. Main focus is on handling pair wise biological sequences. CUDAlign 2.0 is extended part of CUDAlign 1.0. This algorithm allows calculating both score and alignment for huge sequences. CUDAlign 2.1 [1] includes optimization techniques to speed up the performance. Blockpurning is optimization step and goal of this step is to eliminate the calculation of cell that surely not belongs to optimal alignment.

CUDAlign 2.1 follows algorithm introduced in CUDAlign 1.0 as base algorithm and then optimization techniques were introduced. As first part is to create DP matrix for two sequences as seq1 and seq2 of size m and n respectively. Matrix is divided into grid with n/rt *CB blocks where CB represent CUDA blocks, t is number of threads in each CUDA block and each thread is able to process r rows that means each block process rt rows. CUDAlign 2.1 divided into 6 stages-

1. Create DP matrix and obtain optimal score.
2. Partial traceback –to generate optimal alignment.
3. Splitting partition to create more crosspoints.
4. Apply MM algorithm with balance splitting
5. Concatenate all result of partition to obtain full alignment.
6. Representation of alignment.

Courtesy-1[Edans Flavius de O. Sandes and Alba Cristina M.A. de Melo, "Retrieving Smith-Waterman Alignments with Optimizations for Megabase Biological Sequences Using GPU"]

III. COMPUTE UNIFIED DEVICE ARCHITECTURE(CUDA)

CUDA programming language was first proposed by Nvidia, offering versions for both the Windows and Linux operating systems. CUDA is a programming environment having provision of writing and running general-purpose applications on the Nvidia GPU. This environment allows us to efficiently run highly-threaded applications on the GPU. CUDA follows a Single Instruction Multiple Thread (SIMT) programming model. The massive parallelism in the CUDA programming model is achieved through its multithreaded architecture. This thread parallelism allows the programmer to partition the problem into sub problems which can be processed in parallel by number of available of threads, and each sub problem is further divided into smaller pieces that are solved cooperatively in parallel by all threads in a block. The kernel, namely the program running for every thread but each with a different thread ID, can be written in the C-like language [14]. The GPU CUDA support different types of memory's are as follows:

- Global memory - This is largest amount of memory on the device is read-and write global memory: 500 megabytes. Although far slower than other memory types, it is relatively constraint free and the easiest to use. The global memory is accessible to all threads means all threads can read/write memory content directly.
- Shared memory- Threads in one block can access some common memory which cannot be accessible to threads of other blocks.
- Constant memory- Constant memory is read-only memory that does not change over the course of kernel execution. This memory can accessible to all threads but in read only manner.
- Texture memory- It is type of global memory. The only difference is that texture memory is accessed through a provided read-only cache. There is difference between cache and conventional cache, it is optimized for spatial locality and not locality in memory[14].

In this paper, proposed algorithm implemented on NVIDIA GTX 680 Graphics card using CUDA is the example of great performance/watt Streaming Multiprocessor. In GTX680 total no of CUDA core is 1536 .GTX Core work on Base Clock 1006 and Boost Clock 1058. Each CUDA core in GTX 680 support 1024 number of active thread. The GTX 680 has 2048MB memory which work on 6 MHz Speed. Block diagram of GPU Architecture [14] is shown in Fig.2.

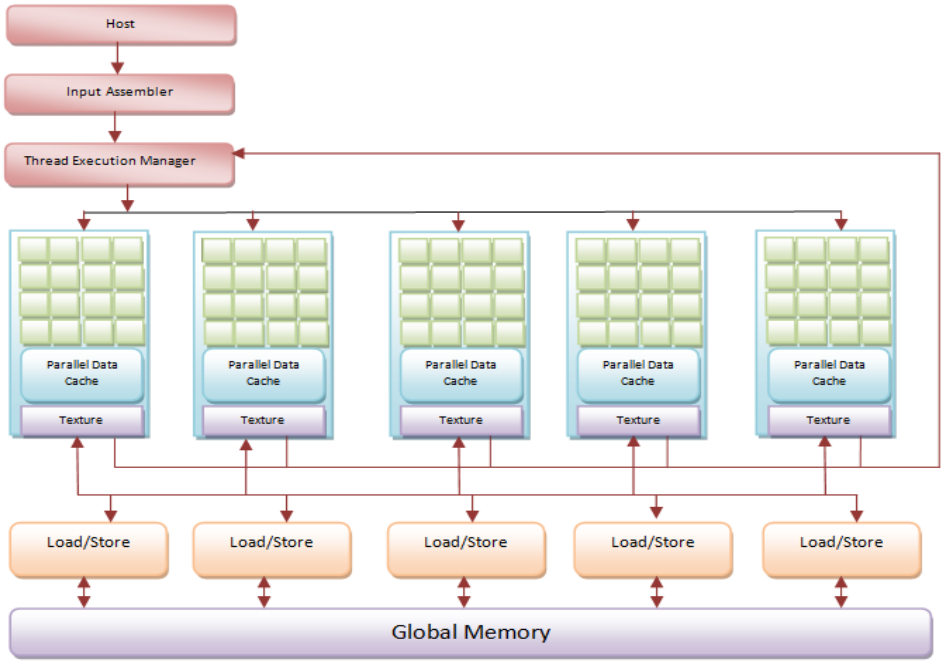

Fig. 2. Architecture of GPU

IV. SMITH-WATERMAN ALGORITHM

T. F. Smith and M. S. Waterman [2] proposed Smith Waterman Algorithm, used to retrieves the optimal score and local alignment between two sequences. This algorithm totally based on Dynamic programming approach. In dynamic programming method applied to complex problem to solve that problem and get optimal solution by dividing the problem into small subproblems then find the solution for each sub problem. Finally combine solution of all sub-problems to get overall solution. This algorithm is divides into three steps-

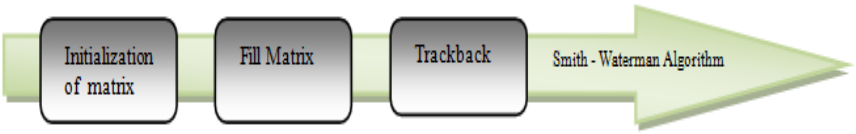

Fig. 3 SW algorithm flow.

First part of SW algorithm is to create the dynamic programming (DP) matrix and initialize it. Suppose the two sequences are seq1 = {A T C G } and seq2 = { T C G } of length m=4 and n=3 respectively then create DP of m+1 rows and n+1 column. Then initialize all cells the first row and column of matrix by 0 as shown in Fig. 4.

		T	C	G
	0	0	0	0
A	0			
T	0			
C	0			
G	0			

Fig. 4 Initialization Step.

Next part is to fill each cell of matrix and it can be done by using Equation (1). $H_{i,j}$ represents each cell of matrix. $S_{i,j}$ denote the similarity score according to match or mismatch for each instance and G is consider as Gap penalty.

$$H_{i,j} = \max \begin{Bmatrix} 0 \\ H_{i-1,j-1} + S_{i,j} \\ H_{i-1,j} + G \\ H_{i,j-1} + G \end{Bmatrix} \quad (1)$$

Final stage of SW algorithm Trace back. It is used to find out optimal alignment which has maximum score. In trace back step, it starts from the highest score and continues until the minimum score as shown in Fig. 5.

		T	C	G
	0	0	0	0
A	0	0	0	0
T	0	1	0	0
C	0	0	2	0
G	0	0	0	3

Fig. 5 Trace back

Finally with the use of Trace-back step we get best alignment in seq1 and seq2 that is {T, C, G} of seq1 is aligned with {T, C, G} of seq2. In this way smith waterman algorithm helps to obtain best alignment for complex biological sequence.

V. PROPOSED METHODOLOGY

The implementation of parallel algorithm to calculate alignment of huge biological sequence is achieved to increase the efficiency of Smith Waterman algorithm on GPU. The program is executed simultaneously on CPU and GPU. The control loop is handled by CPU. Program reads subject sequence and target sequence as input for alignment. Once the query is received, GPU memory is allocated for input sequences and dynamic programming matrix of SW algorithm. SW algorithm is executed in three steps as initialization, processing for creation of dynamic programming matrix and finally traceback to generate score and alignment for subject and target sequence. CUDA Kernel returns score and alignment for sequences using Parallel scan approach of SW algorithm. The resultant data is copied from GPU to CPU thereby displaying result of alignment. Program can utilize number of cores available in GPU. Each thread should take care of complete processing of single pair of sequences. The subject query sequence is shared between threads.

GPU GTX 680 consists of 1024 number of threads per block. The device code is launched by host which organized as grid which is one or two dimensional array of block. Using these threads parallelism in Smith Waterman algorithm is achieved.

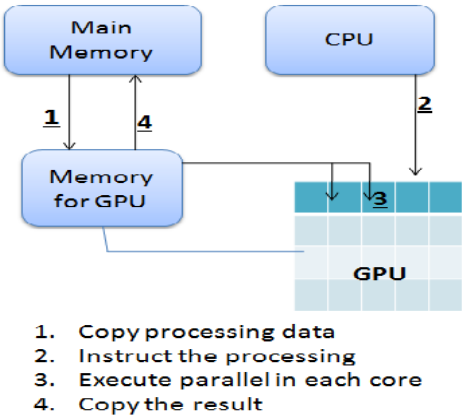

Fig. 6 CUDA Processing Flow

Figure 6 shows CUDA Processing flow. CUDA Program communicates with different device parts. It starts with, copying data from main memory to GPU memory i.e. Global Memory and then CPU instructs the GPU to perform processing. This process is carried out by GPU using Kernel Command after which the assigned tasks are executed in parallel. The results are then copied from GPU memory to main memory.

For resolving the critical requirement of memory space, parallel method for Smith-Waterman algorithm, using the strategy of divide and conquer.

- Input sequence is divided into number of segments and assign each segment independently to each thread
- Next step is to apply Smith-Waterman algorithm independently on each thread.
- Last step is to combine the results yield from each thread and the optimal local alignment is obtained.

Algorithm

An unknown sequence usually needs to compare with several known sequences in a sequence database.

1. Read input as query sequence (Unknown sequence) and subject sequence (known sequences).
2. Allocate GPU memory for input data and copy the sequences on GPU memory.
3. For allocating threads, first calculate total number of thread and thread per block then divide query sequence into segments. Assign each segment to each thread independently, at the same time broadcast subject sequence to each thread.

4. Then launch CUDA Kernel. Apply Smith Waterman algorithm independently on each thread for query sequence segment and subject sequence.
5. At Last, Combine solution of each thread to get optimal alignment of query and subject sequence. Copy final result from GPU memory to CPU memory and finally display result of alignment.

In order to make comparison of parallel scan approach of SW algorithm, an algorithm is implemented in three ways as serial implementation, OpenMP implementation and GPU based implementation. OpenMP is a programming model for shared memory architectures. It is also an Application Program Interface (API) that could be used to explicitly direct multi-threaded, shared memory parallelism for a set of processing nodes.

VI. PERFORMANCE ANALYSIS

In experimentation NCBI dataset is used. Input to the Smith waterman algorithm is query sequence for which alignment and score is calculated as output. We test the performance of serial implementation of Smith-Waterman algorithm with OpenMP implementation and finally CUDA based implementation of parallel scan approach of SW algorithm. Here we use NCBI dataset for finding best alignment over available biological sequences. Table I shows time comparison of serial and OpenMP SW algorithm for different query length. Table II shows performance comparison of serial, OpenMP and cuda SW algorithm on same length sequence.

TABLE I
PERFORMANCE COMPARISON FOR SW ALGORITHM FOR SERIAL & OPENMP IMPLEMETATION

Length Of seq1 (n)	Length Of seq2 (n)	Serial Execution time for Alignment (sec)	OpenMp Execution time for Alignment(sec)
213	248	1.962	0.0037
496	496	2.387	0.0088
709	709	3.689	0.0144
993	901	6.498	0.0254
1244	1173	9.053	0.0413

TABLE III
PERFORMANCE COMPARISON FOR SW ALGORITHM ON CPU, OPENMP & GPU

Input Size (n)	Serial Execution time for Alignment (sec)	OpenMP execution time for alignment (sec)	CUDA Execution time for Alignment (sec)
2000	51.97	0.145	0.09
3000	74.27	0.215	0.12
4000	206.43	0.570	0.15
6000	461.23	1.26	0.19
8000	819.10	2.88	0.32

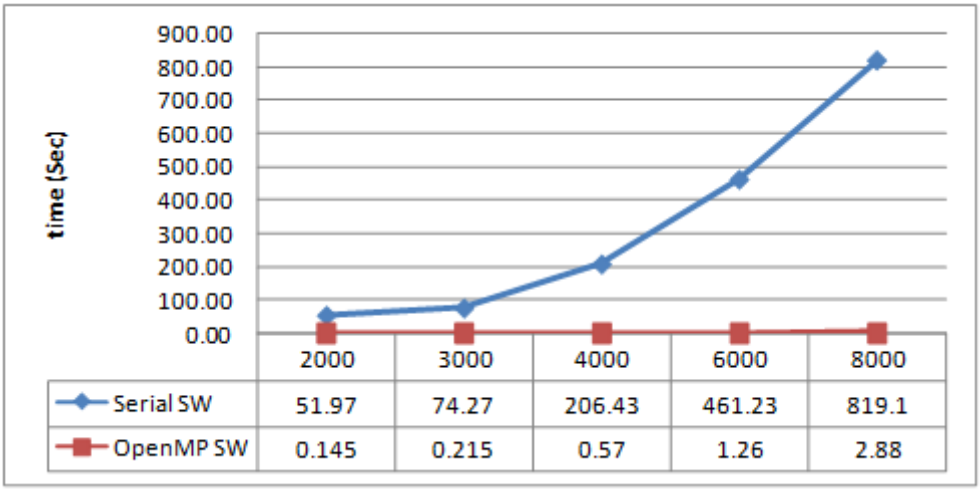

Fig. 7. Time Comparison of SW algorithm :Serial Vs OpenMP implementation

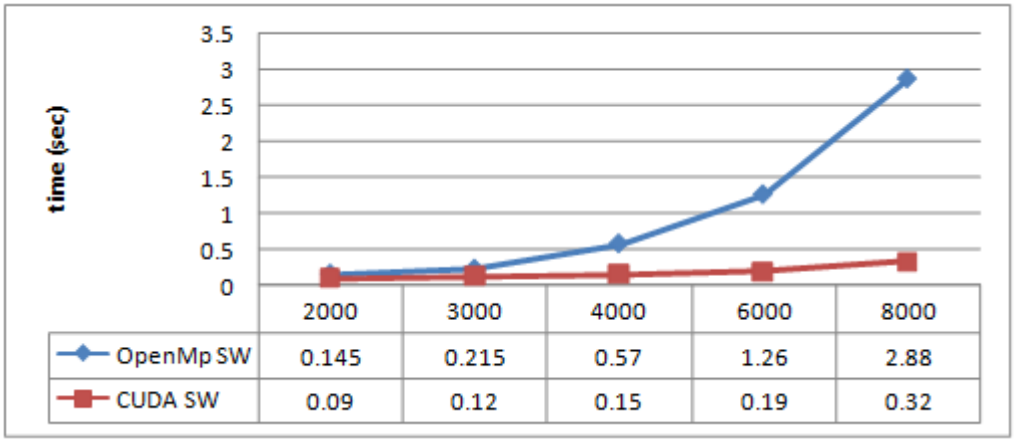

Fig. 8. Time Comparison of SW algorithm : OpenMP Vs CUDA implementation

VII. CONCLUSIONS

Sequence Analysis is core operation in Bioinformatics. The implementation of Smith-waterman algorithm with the parallel scan approach on GPU able to align large biological sequences. Now a day's GPU receive lot of attention. Use of GPU helpful to accelerate analysis task in bioinformatics in which we have make provision that the program will be executed concurrently on CPU and GPU. There are various algorithm available based on GPU but drawback is that they are able to calculate only score but not alignment for complex biological sequence. Smith waterman algorithm with parallel scan approach provides score as well as alignment for huge biological sequences and increasing efficiency of alignment calculation as compared to serial and OpenMp implementation.

ACKNOWLEDGMENT

The authors wish to thank MET's Institute of Engineering Bhujbal Knowledge City, Nashik, India for providing lab facilities. Authors are also thankful to Edans Flavius de O. Sandes and Alba Cristina M.A. de Melo, "Retrieving Smith-Waterman Alignments with Optimizations for Megabase Biological Sequences Using GPU" for their work in this area. The mentioned paper is major guideline for this work.

REFERENCES

[1] Edans Flavius de O. Sandes and Alba Cristina M.A. de Melo, Senior Member, IEEE, "Retrieving Smith-Waterman Alignments with Optimizations for Megabase Biological Sequences Using GPU," *IEEE transactions on parallel and distributed systems*, vol. 24, no. 5, may 2013.

[2] T.F. Smith and M.S. Waterman, "Identification of Common Molecular Subsequences," *J. Molecular Biology*, vol. 147, no. 1, pp. 195-197, Mar. 1981

[3] O. Gotoh, "An Improved Algorithm for Matching Biological Sequences," *J. Molecular Biology*, vol. 162, no. 3, pp. 705-708, Dec. 1982.

[4] D.S. Hirschberg, "A Linear Space Algorithm for Computing Maximal Common Subsequences," *Comm. ACM*, vol. 18, no. 6, pp. 341-343, 1975.

[5] E.W. Myers and W. Miller, "Optimal Alignments in Linear Space," *Computer Applications in the Biosciences*, vol. 4, no. 1, pp. 11-17, 1988.

[6] S. Aluru, N. Futamura, and K. Mehrotra," Parallel Biological Sequence Comparison Using Prefix Computations," *J. Parallel Distributed Computing*, vol. 63, no. 3, pp. 264-272, 2003.

[7] A. Driga, P. Lu, J. Schaeffer, D. Szafron, K. Charter, and I. Parsons," FastLSA: A Fast, Linear-Space, Parallel and Sequential Algorithm for Sequence Alignment," *Algorithmica*, vol. 45, no. 3, pp. 337-375, 2006.

[8] Y. Liu, W. Huang, J. Johnson, and S. Vaidya, "GPU Accelerated Smith-Waterman, " *Proc. Sixth Int'l Conf. Computational Science (ICCS)*, vol. 3994, pp. 188-195, 2006.

[9] W. Liu, B. Schmidt, G. Voss, A. Schroder, and W. Muller-Wittig, "Bio-Sequence Database Scanning on a GPU," *Proc. 20th Int'l Conf. Parallel and Distributed Processing (IPDPS)*, 2006.

[10] S. Manavski and G. Valle, "CUDA Compatible GPU Cards as Efficient Hardware Accelerators for Smith-Waterman Sequence Alignment," *BMC Bioinformatics*, 9(Suppl 2), 2008.

[11] Y. Liu, D. Maskell, and B. Schmidt, "CUDASW++: Optimizing Smith-Waterman Sequence Database Searches for CUDA-Enabled Graphics Processing Units," *BMC Research Notes*, vol. 2, no. 1, p. 73, 2009.

[12] E.F. de, O. Sandes, and A.C.M.A. de Melo," CUDAlign: Using GPU to Accelerate the Comparison of Megabase Genomic Sequences," *Proc. 15th ACM SIGPLAN Symp. Principles and Practice of Parallel Programming (PPoPP)*, pp. 137-146, 2010.

[13] www.youtube.com/watch?v=IatoWOsJ35Q

[14] http://www.techarp.com/showarticle.aspx?artno=358pgno=3

Komal B.Deshmukh, M. U. Kharat, "Review on Retrieving Biological Sequence Alignment using Smith-Waterman Algorithm",IJIRCST, Vol- 03, Issue-01, January-2015.

