IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. VI (Mar – Apr. 2015), PP 48-52
www.iosrjournals.org

A survey of Parallel models for Sequence Alignment using Smith Waterman Algorithm

Jyoti Pandey[1], Dr. Nilay Khare[2], Rahul Pandey[3]
[1,2]*(Computer Science & Engineering, Maulana Azad National Institute of Technology, India)*
[3]*(Computer Science & Engineering, International Institute of Technology, Hyderabad, India)*

Abstract: *Nowadays stack of biological data growing steeply, so there is need of smart way to handle and process these data to extract meaningful information related to biological life. The purpose of this survey is to study various parallel models which perform alignment of the sequences as fast as possible, which is a big challenge for both engineer and biologist. The various parallel models discussed in this paper are: implementation using associative massive parallelism contain architecture such as associative computing, ClearSpeed coprocessor and Convey Computer. Some parallel programming models such as MPI, OpenMP and hybrid (combination of both). Then the implementation of alignment using systolic array and lastly uses single and multi-graphics processors, that is, using graphics processing units.*

Keywords: *ASC, MTAP, PE, recursive variable expansion, Smith Waterman Algorithm, SIMD, scoring matrix.*

I. Introduction

Whenever a new gene is found, its functionalities are inferred by finding the percentage of similarity to one of the known sequence. The general problem is to find the degree of similarity between two separate sequences and best match of the query string inside much large database sequence. The focus is mainly on the sequence of the nucleic acid which contain Adenine ('A'), Cytosine ('C'), Guanine ('G') and Thymine ('T') for DNA or the sequence of amino acid form an alphabet of 20 possible amino acid.

There are various algorithms for the alignment of two sequences which are classified as local and global alignment. The Smith Waterman algorithm is a local alignment algorithm, which finds a query sequence into the large sized database sequence. It uses the standard dynamic programming since it incorporates both overlapping subproblem and optimal substructure property. The main extract of the algorithm has been explained as mathematically below:

H (i, 0) =0 for 0<=i<=m
H (0, j) =0 for 0<=j<=n

$$H(i,j) = max\begin{cases} 0 \\ H(i-1,j-1)+s(i,j) \\ H(i-1,j)+d \\ H(i,j-1)+e \end{cases} \qquad 1<=i<=m,\ 1<=j<=n$$

H (i-1, j-1) +s (i, j) is to match or mismatch, H (i-1, j) +d is for deletion and H (i, j-1) +e is for insertion.
where
a, b = strings over the alphabet
m=length (a)
n=length (b)
S (a, b) =similarity function on a, b
H (i, j) =maximum similarity score
d=penalty for deletion
e=penalty for insertion

There are various accelerated versions of the algorithm like implementation on GPU, FPGA, SIMD, Cell Broadband engine. Also the various models have been developed like Convey Computer uses HC-1, Clear speed coprocessor and so on. The various modes for parallelizing the sequence alignment algorithm are discussed below.

II. Parallel Models

2.1 Associative Massive Parallelism

The implementation of the Smith Waterman algorithm on three different parallel architectures using associative massive parallelism [1] where different architectures include: Associative Computing (ASC), the ClearSpeed coprocessor, and the Convey Computer FPGA coprocessor [2]. These three architectures allow non-overlapping, non-intersecting subalignments.

2.1.1 Associating computing model

Associative Computing (ASC) [3] is an associative model of parallel computation can be used to implement and extend local sequence alignment algorithm such as a Smith Waterman algorithm. As the name indicates, it does not use associative memory. In associative computing, data searching is performed by content rather than the using address of memory. The Associative computing model finds the best local alignment in O(m+n) time using (m+1) processing element, where m and n are size of comparing sequences.

The ASC model consists of an array of cells where each cell contains processing element and local memory, it also contains an instruction stream (IS). As seen in Fig1 every processing element has access to the memory local to the processing element. Here, each processing element can perform the task of the sequential processor, but not of issuing instruction.

Rather than using a 2-D contiguous matrix of sequential computer, m+1 PE's is used where each PE hold one row of data in sequential Smith Waterman algorithm.

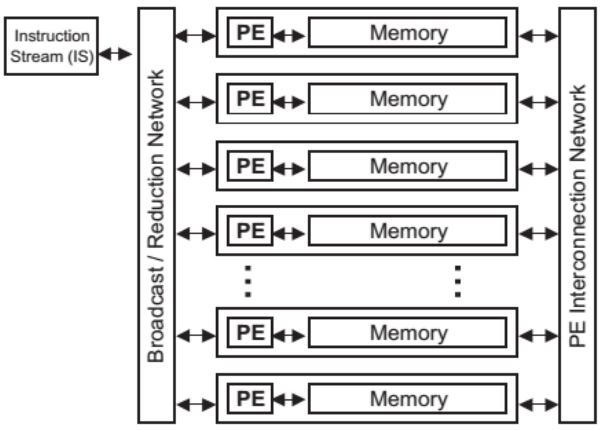

Fig1: ASC model.

2.1.2 ClearSpeed Coprocessor

To deploy Smith Waterman alignment ClearSpeed SIMD CSX620 PCI-X accelerator board is used as platform, which has 96 processing elements per chip with two chips per board so 196 processing elements, the MTAP architecture [2] is used by the board. Since similarity to SIMD like board so CSX620 was chosen, but it lacks associative function, so associative functionality is handled at the software level by CSX620. The functions were written in ClearSpeed assembly language for speed and efficiency.

The Smith Waterman alignment code implemented on ClearSpeed accelerator contain essentially two parts, firstly, the parallel computation of score matrix and secondly, the sequential traceback. The parallel part of the code has an optimal running time of O(m+n) on m processing elements. The performance is excellent for small input size, but it is not good to handle large strings. So to overcome the limitation, the Convey Computer HC1 was chosen to perform the Smith Waterman alignment.

2.1.3 Convey Computer

The Convey Computer combines both host processor and coprocessor in a single hardware system, both processors share global memory and program can be executed concurrently on both processors. FPGA technology has been employed on Convey coprocessor which makes it reconfigurable computing.

Convey Computer's Smith Waterman algorithm run on the x86/64-bit CPU processor, coprocessor or on the both processors concurrently. The implementation on both the processor and coprocessor achieve very high throughput as compared to earlier generation of HC-1 coprocessor, one of the fastest GCUPS rating at its release. The Convey computer is more robust, scalable that can even implement the large data set alignment as compared to the earlier ones.

2.2 Parallel programming model

2.2.1 Open MP

All processors can access data from shared memory without explicit communication. A joint effort by compiler vendors to establish standards in this field, known as OpenMP [3]. Open multiprocessing (OpenMP) is a shared memory architecture API, having a multithreaded architecture, which is an open specification for shared memory parallelism. Its API consist Compiler Directives, Runtime library routines and Environment variables. The main idea of OpenMP is data-shared parallel execution.

In Smith Waterman algorithm implementation, the OpenMP based parallelism using various sizes of the blocks for filling score matrix by different processors in parallel.

2.2.2 MPI

The message passing interface is a standardization of message passing interface library specification. Its central focus is on distributed processing. An MPI program contains a collection of processes, those exchange messages. All MPI processes execute the same program in an SPMD style, that is, the single program executes on multiple data, which enhances the execution speed.

2.2.3 Hybrid (OpenMP and MPI)

By using mixed mode programming, we can take advantage of both shared memory and message passing interface. In mixed mode programming two levels of communication are used, that is, inter-node communication and intra-node communication. In inter-node communication, the message passing method is used between various processors and intra-node uses the shared memory method, where various processors access shared memory

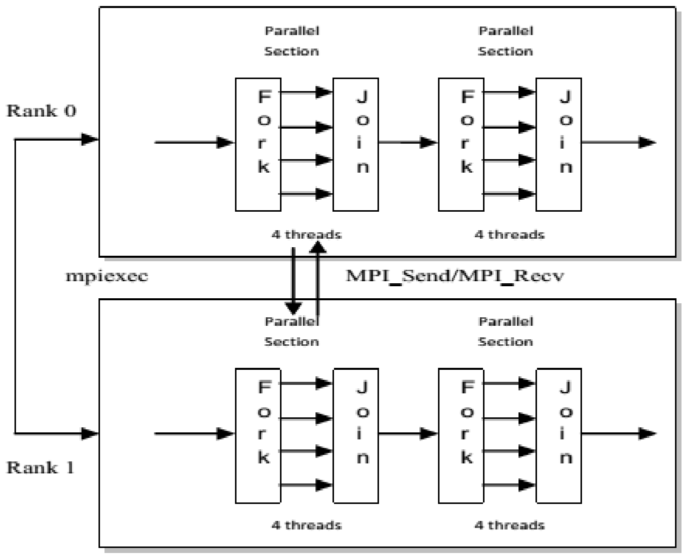

Fig2: MPI/Open MP program execution on two processors.

Fig 2 shows a hybrid system in which two nodes communicate through a message passing interface (MPI) and the processor themselves fork four threads which uses shared memory for communication.
Basically, all three models discussed above are inherited from SIMD modelling [4], [5].

2.3. Systolic Architecture

Most of the hardware implementation for sequence alignment algorithm such as Smith Waterman algorithm do not present full implementation of algorithms and the traceback process is performed on general purpose processor. There are various techniques to accelerate the algorithm like linear recursive variable expansion (RVE) implementations [5]; systolic arrays [6]; incremental approaches [7]; and other simplifications of the algorithm.
Milik and Pulka speed up the processing task by optimizing the hardware architecture and taking advantage of the properties of FPGA [8].

Using Systolic Array

Systolic Array is an arrangement of processors in 1-D or 2-D form where data flow across the array between neighbor synchronously [9]. In systolic array design, the systolic processing element array is mapped to an anti-diagonal line of the score matrix in sequence alignment algorithm like Smith Waterman algorithm. Limited number of PE can be implemented on FPGA due to hardware limitation. So, in the calculation of similarity matrix matrices needed to be divided into submatrices. The complete Smith Waterman algorithm is implemented on FPGA not like earlier ones where scoring matrix is calculated on FPGA and traceback is calculated on general purpose processor.

Smith-Waterman algorithm on an innovative reconfigurable supercomputing platform, the XD1000, for DNA takes a 384-PE systolic array working at 66.7 MHz, achieves 25.6 GCUPS peak performance [10].

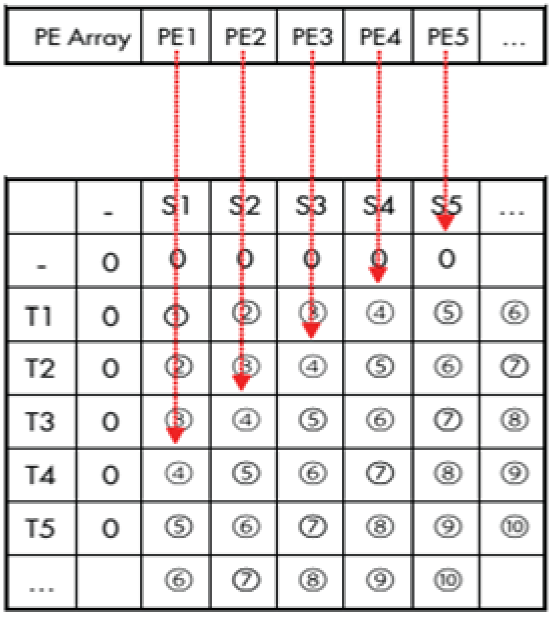

Fig3: Mapping Smith Waterman to Systolic PE Array

2.4 Single and multi graphics processor

In this paper, firstly basics of the Smith Waterman algorithm are presented, the column maximum and row maximum are calculated for each table entry causing O(L1L2(L1+L2)) computational work which is comparatively more as compared to retaining the previous column and row sums. By doing so the storage requirement increases, but it reduces work significantly.

The classic way to parallelize the Smith Waterman algorithm is to work along anti-diagonals. As each diagonal element depends upon cell above, left and diagonally above, so each diagonal element can be calculated independently of others.

The main problem with parallelizing through diagonal approach is how to access memory. Any hardware can be used to accelerate memory accesses. In Graphics processing unit, mainly it is the matter of how to fetch database and query sequence from memory, through the implementation of memory banks and memory streaming, the memory access can be enhanced with hardware. So it implies that, it is very efficient to access 32 consecutive memory locations in comparison to those scattered along the anti-diagonals [11], [12].

To deal with slow random memory access Smith Waterman algorithm can be reformulated in which calculations can be performed in parallel one row or column at a time.

Using SLI (Nvidia) or ATI CrossFire technology, Smith Waterman algorithm can be implemented on upto 4 GPUs which are installed on a single motherboard.

If it needs to be done by implementing a large number of GPUs operated on different motherboard and communicating via the connecting network, then a smart parallelizing and implementation strategy needed. In this, the sequence is distributed in N slightly overlapping blocks and blocks distributed on different processing cores and then result is collected through Ethernet, so needed much attention while combining the result collected from GPUs on the host CPU. Here the size of overlapping need to be considered taking care as the result depends on alignment from left to right (because Smith Waterman compute from left to right) as compared to alignment in right to left.

III. Conclusion

The sequence alignment using a Smith Waterman algorithm has been the most extensively used algorithms in Bioinformatics to align two sequences for extracting information relevant to the evolutionary and biological life. So far there are various models or we can say the accelerator used for alignment of sequences which has specific advantages, according to the needs like depending on small or large dataset or just to get highest alignment or scoring value in scoring matrix. This paper discusses about the FPGA, GPU, Associative massive parallelism and various hybrid system.

We conclude that parallel models provide an excellent platform on which to a run sequence alignment, and those parallel architectures will be able to cope far more easily with the vast data produced by the high throughput sequencer.

References

[1]. S. Steinfadt, J.W. Baker, SWAMP: Smith–Waterman using associative massive parallelism, International Symposium on Parallel and Distributed Processing, 2008, 1–8.

[2]. S. Steinfadt, Fine-grained parallel implementations for SWAMP+ Smith–Waterman alignment, Parallel Computing, 39(12), 2013, 819-833.

[3]. Potter, J. W. Baker, S. Scott, A. Bansal, C. Leangsuksun, and C. Asthagiri, ASC: an associative-computing paradigm, Computer, 27 (11), 1994, 19-25.

[4]. M. Farrar, Striped Smith-Waterman speeds database searches six times over other SIMD implementations, Bioinformatics, 23(2), 2007, 156–161.

[5]. L. Hasan and Z. Al-Ars, An efficient and high performance linear recursive variable expansion implementation of the Smith-Waterman algorithm, Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society, The Netherlands, 2009, 3845–3848.

[6]. M. Gok and C. Ylmaz, Efficient cell designs for systolic smith-Waterman implementations, International Conference on, 2006, 1–4.

[7]. A. Pulka and A. Milik, Considerations on incremental approach to hardware implementation of smith-waterman algorithm, in Mixed Design of Integrated Circuits and Systems (MIXDES), Proceedings of the 18th International Conference, 2011, 283–288.

[8]. A. Milik and A. Pulka, Hardware oriented optimization of Smith Waterman algorithm, International Conference on, 2010, 319–322.

[9]. L. Hasan, Y.M. Khawaja, A. Bais, A Systolic Array Architecture for The Smith-Waterman Algorithm With High Performance Cell Design, Proc. IADIS European Conference on Data Mining, Amsterdam, The Netherlands, July 2008.

[10]. P. Zhang, G. Tan, and G. R. Gao, Implementation of the Smith Waterman algorithm on a reconfigurable supercomputing platform, Proc. 1st international workshop on High-performance reconfigurable computing technology and applications, New York, NY, USA, 2007, 39-48.

[11]. Ł. Ligowski, W. Rudnicki, An efficient implementation of Smith–Waterman algorithm on GPU using CUDA for massively parallel scanning of sequence databases, 23rd IEEE International Parallel and Distributed Processing Symposium, May 25–29, Aurelia Convention Centre and Expo Rome, Italy, 2009.

[12]. V. Chaudhary, F. Liu, V Matta, and T. Yang, Parallel implementations of local Sequence alignment: hardware and software, Parallel Computing for Bioinformatics and Computational Biology: Models, Enabling Technologies, and Case Studies, Wiley Series on Parallel and Distributed Computing , 2006, 234.