Sequence Similarity Search on Reconfigurable Computing System

Sajish Chandrababu, Yogindra Abhyankar, and Rajendra Joshi

***Abstract*—Genome sequencing has been one of the great achievements of biotechnology. The future of bioinformatics lies in the analysis of the vast ocean of data generated by whole genome sequencing. There are tremendous challenges in building a high throughput environment for whole genome analysis of multiple organisms. The Smith-Waterman algorithm provides sequence alignments with high sensitivity but suffers from high computational requirement. We report an accelerated Smith-Waterman solution using FPGA based Reconfigurable Computing (RC) that significantly brings down the search time. Our solution handles large lengths up to 1 M characters each for the query as well as database sequences and do not restrict on using a particular brand of processor. The experimental benchmarks show 90X speedup in the execution time against a purely software solution. Reported performance figures include cases, where reduction in search time from 4 days to merely 59 minutes is obtained. Multiple queries with an estimated search time of 74 days were completed in just 20 hours by our RC solution.**

***Index Terms*—FPGA, reconfigurbale computing, smith-waterman.**

I. Introduction

Genomics has been one of the main drivers in the area of biotechnology. The human genome project along with the numerous other sequencing projects has opened the floodgates of the genomic data. Recent years have seen an explosion in the amount of biological information that is available. Various sequence databases are doubling in size every 15 months and we now have the complete genome sequences of ~ 400 organisms [1]. It appears that the challenges have shifted from the ability to generate vast quantities of data to analyzing this data and deriving knowledge from it. More than sixty percent of the human diseases are genetic in nature and therefore genomic technologies have a major role to play. There are a large number of diseases where genomic and bioinformatics technologies can play an important role in finding a cure. The pharmaceutical industry has embraced genomics as a source for drug targets. It also recognizes that the field of bioinformatics is crucial for validating these potential drug targets and for determining which ones are the most suitable for entering the drug development pipeline. Today, it is no longer a challenge to analyze a single or multiple genes, but to analyze and compare multiple genomes. Building a high throughput environment to analyze and model such large data is the key to the future.

For carrying out genome sequence analysis, many popular methods like Smith-Waterman (S-W) [2], BLAST [3] and FASTA [4] exist. The BLAST and FASTA algorithms are based on heuristics, while the Smith-Waterman algorithm is based on dynamic programming and highly compute intensive in nature. Though the S-W algorithm gives the best alignments due to its high degree of sensitivity, its use has been limited due to high demand of computational resources. When searching long queries over large databases, it may take hours and days of computation time, making it necessary to use high-end servers and clusters or use other algorithms by compromising on sensitivity. To speed up searching with S-W, several approaches such as using heuristic searches of S-W algorithm like ParAlign [5], use of parallel processing methods [5]-[9] and the use of specialized hardware [10]-[11] and Graphics Processing Units [12] have been tried. Also commercial products e.g. TimeLogicDeCypher (www.timelogic.com), GeneMatcher2 (www.paracel.com), Progeniq (www.progeniq.com) and Cray XD1 with application accelerator (www.cray.com) are available.

Reconfigurable Computing (RC) is an emerging field that blurs boundaries between software and hardware. Usually, software is considered as a flexible entity while the hardware is considered to be of fixed in nature. RC explores solutions where the underlying hardware is also flexible and is modified at runtime to boost the application performance. Typically from an application, the compute intensive routines are off-loaded on the RC hardware as hardware libraries, resulting in application acceleration. One of the enabling technologies for RC is the Field Programmable Gate Array (FPGA). RC is now recognized as one of the key technology for enhancing supercomputing performance.

We present an accelerated Smith-Waterman solution using RC, designed and developed in-house, that significantly shortens the search time compared to a purely software solution. The performance results with standard protein and DNA databases are presented with speedup up to 90X.

Some of the high-end systems use internal networks especially designed for low latency and high bandwidth to connect their application accelerator directly to the AMD Opteron processors [13]. Our solution uses generic PCI interface, without restricting to a specific processor. Despite the additional latency caused by this interface, our RC based Smith-Waterman solution shows better results in terms of performance; further our design can handle longer lengths of query and database sequences which were a limitation in other implementations [14]. The Performance benchmarks for our solution are obtained by comparing the execution time of the Smith-Waterman searches on a server machine attached with the RC hardware to a purely software based

Manuscript received September 13, 2012; revised October 24, 2012. This work was supported by Ministry of Communications and Information Technology, Government of India.

The authors are with the Hardware Technology Development Group,Centre for Development of Advanced Computing,Pune 411007,India (e-mail: yogindra@cdac.in).

search, running on the same machine without RC.

Reference [15], has reported 160-folds scaleup for the Smith-Waterman searches. However, this speedup are based on a comparison between their FPGA implementation vs. software execution on an embedded FPGA processor, running at much lower speed i.e. not on a standard state of the art processor. Performance comparison with TimeLogicDeCypher is not done due to lack of information.

Recently, GPU based accelerators are becoming popular for accelerating applications. The latest performance figures on GPU [12] reports 45 times faster results than the purely software version; however our results with RC are twice as fast as the GPU solution. Additionally the RC solution just takes around 25 watts of power that is substantially less as compared to the GPU solution.

II. Implementation

The FASTA sequence alignment package from the University of Virginia has SSEARCH module, implementing the Smith-Waterman algorithm. The SSEARCH module from this package was used for this work. Profiling of SSEARCH was done to identify the compute intensive sub-routines that are good candidates for hardware acceleration using RC. Accordingly, the matrix building routine was identified and an equivalent functionality of it was developed as synthesizable hardware library. These were seamlessly integrated with the application.

The Smith–Waterman algorithm finds the optimal alignment by constructing a solution table. The database sequence is placed along the top row, and the query sequence is placed in the first column. For two molecular sequences A= A_1 A_2 A_3…A_n and B= B_1 B_2 B_3…B_m, the table values $H_{i,j}$ are calculated, where i is the row index and j is the column index. The algorithm is initiated by placing zeros in the first row and first column of the table. The other entries in the table are then calculated via the equation,where, S (A_i, B_j) gives the similarity between the sequence elements A_i and B_j. W_i is the penalty for the first residue in a gap and W_e is the penalty for additional residues in a gap. Generating the table values is the computationally intensive part of the Smith–Waterman algorithm. Once these values are obtained, the optimal alignment is determined by performing a 'traceback'.

$$E(i,j) = max\begin{cases} E(i-1,j) - W_e \\ H(i-1,j)E - W_i - W_e \end{cases}$$

$$F(i,j) = max\begin{cases} F(i,j-1) - W_e \\ H(i,j-1) - W_i - W_e \end{cases} \quad (1)$$

$$H(i,j) = max\begin{cases} 0 \\ E(i,j) \\ F(i,j) \\ H(i-1,j-1) + S(A_i, B_j) \end{cases}$$

The RC hardware was developed in-house that can be plugged into a PC via a 64-bit, 66 MHz PCI bus. It has a 6 million gate, XC2V6000 FPGA from Xilinx and 256MB of SDRAM as well as 4MB of ZBT RAM. Fig. 1 shows the block diagram of the card.

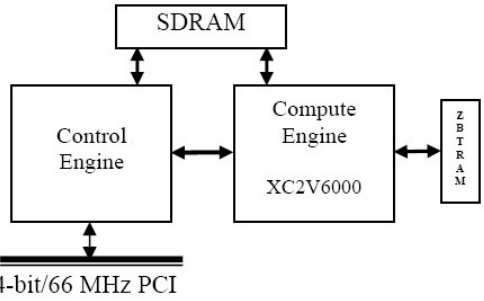

Fig. 1. RC card with 6 million gate compute FPGA.

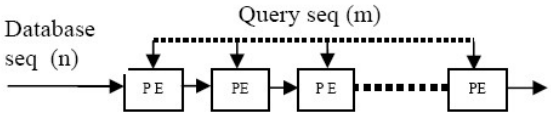

Fig. 2. Systolic array of PEs on the compute FPGA having database and query sequences as inputs.

The Smith-Waterman hardware library uses the Smith-Waterman algorithm for calculating the best local alignments between two sequences. The inherent parallel nature of the algorithm is well suited for FPGA implementation. The hardware library was realized into Processing Element (PE), with multiple such units connected to form a systolic array (Fig. 2). Database sequence, query sequence and the scoring matrix are input to the PE array. Each PE calculates the score corresponding to each cell of the Smith-Waterman algorithm.

The RC-enabled version of the code along with the required parameters was run on a PC, having the RC hardware. These parameters were passed to the Smith-Waterman hardware library that was mapped to the FPGA. The results from the FPGA were post-processed on the PC. The RC-enabled SSEARCH supports both DNA (nucleotide) and protein sequences. The hardware library modules were written in VHDL hardware description language simulated using ModelSim 6.2g simulator and synthesized using Xilinx ISE 11.1 tool. These modules are highly optimized and utilize around 95% of the FPGA resources. The design can work on query and database sequences of lengths up to 1 Million characters each.

TABLE I: Overall Speedup for Search Time Using Protein of Length Ranging from 500 AA to 32000 AA Against Swiss Rot Database

Query sequence (Length)	Total Time (Sec)		Speed-Up
	SSEARCH	RC	
NP_055494 (500 aa)	762.48	13.63	55.9
NP_009121 (1000 aa)	1490.79	21.03	70.88
NP_035737 (2019 aa)	2974.55	37.70	78.8
XP_343570 (4000 aa)	5908.75	72.40	81.6
ABF87402 (11939 aa)	17250.50	207.85	82.9
AAN10358 (23015 aa)	32152.35	390.22	82.3
NP_597681 (27118 aa)	39173	485.38	80.7
XP_001065955 (31920 aa)	45832.99	563.69	81.3

III. Results

The performance comparisons were carried out by executing database searches on a 2.8 GHz Intel Xeon

machine with 4 GB memory under Red-Hat Enterprise Linux. Same executions where run on the system with RC and also the system without RC. While executing the code on the machine with RC, whenever there was a reference to the matrix building routine, a subroutine call was made to the RC hardware. The Smith-Waterman algorithm developed as the hardware library, and mapped on RC provided the application acceleration.

TABLE II: OVERALL SPEEDUP FOR SEARCH TIME USING NUCLEOTIDE OF LENGTH RANGING FROM 500 BP TO 4000 BP AGAINST A 2.9 GB EST DATABASE OF MOUSE

Query sequence (Length)	Total Time (Sec)		Speed-Up
	SSEARCH	RC	
XR_019014 (500 bp)	42014	576.82	72.8
XM_084868 (1000 bp)	81408	997.15	81.6
NM_003004 (2019 bp)	161835	1842.96	87.8
NM_003259 (3000 bp)	241779	2698.35	89.6
NM_148414 (4000 bp)	321759	3556.71	90.4

For the protein sequence performance comparison, query sequences of length 500 amino acids to 32,000 amino acids were searched against the Swissprot database (73396406 residues in 195650 sequences). While for nucleotide sequence, query sequences of length 500 base pairs (bp) to 4000 bp nucleotide sequence were searched against databases of mouse ESTs (2193789456 base pairs in 4720045 library sequences).

An acceleration of 55 - 81 folds in execution time was observed for protein sequences (Table I); similarly for nucleotide sequences, an acceleration of 72 - 90 folds in execution time was observed (Table II). The speedup obtained using RC, depends upon the lengths of query, database sequences and the time taken by the non compute-intensive portion of the application.

When searching a nucleotide query of size 4000 bp against the EST mouse database, the execution time was nearly 4 days for software version, while with theRC,it was reduced to nearly 59 minutes (Last entry in Table II). This clearly shows the benefits of using RC technology while executing searches of large queries over large databases.

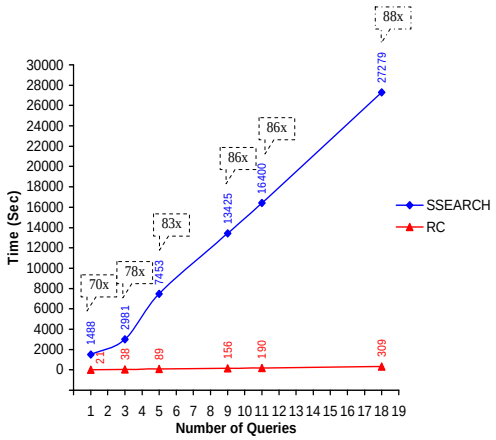

Fig. 3. Time for searching multiple protein queries, each of length 1000 aa against the swissprot database, using SSEARCH and RC.

Multiple queries of protein and nucleotide sequences were also searched, including 1000 queries of Oligo sequences against the human unigene database (7000248 sequences, 5.2 GByte in size).

Results for searching multiple protein queries, each having lengths of 1K residues against the Swissprot database are shown in Fig. 3. By using RC, Speedup in the range of 70-88 times was observed for variable number of queries.

Speedup results for multiple nucleotide queries are shown in Table III, having each query of length 4K bases. For a single query, the software based solution takes around 4 days (89 hours) while the RC enabled solution takes 1 hour, resulting in a 90X speedup. In case of searching with 20 such queries, the purely software solution will take approximately 74 days while the RC based solution completed this task in nearly 20 hours.

TABLE III: OVERALL SPEEDUP FOR SEARCHING MULTIPLE NUCLEOTIDE QUERIES, EACH OF LENGTH 4K BASES AGAINST THE EST MOUSE DATABASE
* DENOTES ESTIMATED TIME

Number of queries	Total Time (Sec)		Speed-Up
	SSEARCH	RC	
1	321759	3556.716	90.4
20	6435180*	69408.26	92.7

Searching of 1K query sequences of Oligo (50-mer each) against the human unigene database was also completed by RC in 33 ½ hours. For this same search, the estimated search time for software solution would be around 52 days.

Our design can handle queries up to 1 Million lengths; however, searching such large queries against the standard databases like Swiss-Prot, EST-mouse and Human unigene will require computation time typically in weeks/months on a machine without RC. Therefore the above performance comparisons were restricted to a maximum query length of 32K.

Table I-III, show up to 90X speedup for execution timefor search based on RC. If such acceleration is to be achieved using parallel computing technology, it would require at least 90 processors. Keeping in mind the cost of large compute cluster and the associated exorbitant running cost, the RC provides a cost effective alternative.

IV. SUMMARY

Presented RC based solution was found to accelerate drastically the Smith-Waterman sequence searches. Test results showing speedup up to 90 times, compared to the software solution are presented. Reported performance figures include cases, where reduction in single query search time from 4 days to merely 59 minutes and for multiple queries, from 74 days to 20 hours were obtained. Standard protein and DNA databases were used for the performance analysis using different query sequences. Our solution supports large query and database lengths. RC technology can play an important role while building a high throughput environment for analyzing and comparing multiple genomes of various organisms. Such an environment can ultimately lead to the further understanding of the basic biology of various organisms leading to better agro and healthcare products.

ACKNOWLEDGMENT

Authors are thankful to our colleague, Mr. Ashok Verma and Mr. SantoshAtanur for helpful discussions.

REFERENCES

[1] K. Liolios, N. Tavernarakis, P. Hugenholtz, and N. C. Kyrpides, "The Genomes on Line Database (GOLD) v.2: a monitor of genome projects worldwide," *Nucleic Acids Res.*, vol. 34, pp. D332-D334, 2006.

[2] T. F. Smith and M. S. Waterman, "Comparison of Biosequences," *Adv. Appl. Math*, vol. 2, pp. 482-498, 1981.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic Local Alignment Search Tool,"*J. Mol. Biol.*, vol. 215, pp. 403-410, 1990.

[4] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," in *Proc. Natl. Acad. Sci.* USA, 1988, pp. 2444-2448.

[5] T. Rognes, "Par Align: a parallel sequence alignment algorithm for rapid and sensitive database searches," *Nucleic Acids Res*, vol. 29, pp. 1647-1652, 2001.

[6] D. L. Brutlag, J. Dautricourt, R. Diaz, J. Fier, B. Moxon, and R. Stamm, "BLAZE-an implementation of the Smith-Waterman Sequence Comparison Algorithm on a Massively Parallel Computer," *Comput. Chem.*, vol. 17, pp. 203-207, 1993.

[7] C. Janaki and R. R. J oshi, "Accelerating comparative genomics using parallel computing," In *Silico Biol.*, vol. 3, pp. 429-440, 2003.

[8] A. S. Deshpande, D. S. Richards, and W. R. Pearson, "A platform for biological sequence comparison on parallel computers," *Comput. Appl. Biosci.*, vol. 7, pp. 237-247, 1993.

[9] T. Rognes and E. Seeberg, "Six-fold speed-up of Smith-Waterman sequence database searches using parallel processing on common microprocessors,"*Bioinformatics*, vol. 16, no. 8, pp. 699-706, 2000.

[10] R. Hughey, "Parallel hardware for sequence comparison and alignment," *Comput. Appl. Biosci*, vol. 12, no. 6, pp. 473-479, 1996.

[11] R. P. Jacobi, M. A. Rincon, L. G. Carvalho, C. H. Llanos, and R. W. Hartenstein, "Reconfigurable systems for sequence alignment and for general dynamic programming," *Genet Mol Res.*, vol. 4, pp. 543-552, 2005.

[12] A. K. Saeed, S. Poole, and J B. Perot, "Acceleration of the Smith–Waterman algorithm using single and multiple graphics processors," *Journal of Computational Physics*, vol. 229, pp. 4247-4258, 2010.

[13] S. Margerm, "Reconfigurable computing in real-world applications," *FPGA and Structured ASIC Journal*, 2006.

[14] M. Yim, A. Jacobs, and A. George. Performance evaluation of the Cray bioscience application package on the XD1. [Online]. Available: http://docs.hcs.ufl.edu/xd1.

[15] I. T. S. Li, W. Shum and K. Truong, "160-fold acceleration of the Smith-Waterman algorithm uses a field programmable gate array (FPGA)," *BMC Bioinformatics*, vol. 8, no. 185, 2007.