Needleman Wunsch Implementation for SPAM/UCE Inline Filter

E.M.Tamil[1], M.Y.I.Idris[1], C.M.Thong[1], M.M.Saudi[2] and M.Z.Jali[2]

[1]System-on-chip Research Group, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia
[2]Faculty of Science & Technology, Universiti Sains Islam Malaysia,
Bandar Baru Nilai, Negeri Sembilan.
e-mail: emran@um.edu.my

Abstract

In this paper, the author(s) propose a new technique in spam detection from another discipline and propose an implementation of the underlying algorithm based on FPGA. The choice of algorithm are Needleman-Wunsch that are previously used in bioinformatics. By using Needleman-Wunsch as the main engine, real network traffic will be used as query and compared with spam signature to detect the real spam. Needleman-Wunsch algorithm is one of the earliest algorithm from the family of dynamic programming in approximate string matching. Applying Needleman-Wunsch algorithm in FPGA will greatly speeds the performance of this algorithm in spam scanning as it operate in hardware level instead of software level.

Keywords

Spam, Needleman-Wunsch, FPGA

1. Introduction

Approximate string matching shares a history of being used in bioinformatics for the purpose of sequence alignment. As traditional string matching techniques are only able to detect identical sequences, approximate string matching using dynamic programming principles tend to be more suitable as it capture similarities based on match and mismatch among two sequences. This is more suitable as sometimes genomic sequences that have the same biological purpose are no longer exactly the same or identical because of mutation (Liu et al, 2006).

The application of approximate string matching algorithms from bioinformatics in computer science have gained interest among computer scientists in recent years. There are researches on using algorithms from bioinformatics for network protocol analysis(Beddoe), musical information retrieval using vocal input (Kline and Glinert, 2003) and spam (Sanpakdee et al, 2006).

Spam has becoming a major problem in the technology world. According to Ferris Research, spam costs $100 billion US dollar in term of productivity loss worldwide for the year 2007 alone (Ferris Research, 2007). Besides, as the volume of spam increases to a level higher than ordinary mail, it leads to a similar effect of denial of service attacks (DOS) (WU, 2005) on computer servers. Spam amount to around 90% of today's network traffics (Spamhaus, 2007). Spam consumes bandwidth (Hoanca, 2006), reduce efficiency and dependability of the network (WU, 2005). In the wake of the spam issue, many techniques and approaches are being introduced to counter and reduce the problem. One of the popular method is the application of Bayesian Naives algorithm on email server. However, spammers nowaday are able to find ways to evade detection through Bayesian poisoning (Cumming, 2006). Spammers tend to use words that are not associated with spam in their spam mail to reduce the effectiveness of detection by Bayesian. Beside the Bayesian algorithm, there are other ways to counter spam problem like spam filtering (Hoanca, 2006), rate throttling (Hoanca, 2006), alliance-based approach (Chiu, 2007), multi-faceted approach (WU, 2005) and CAPTCHA (WU, 2005). However, every approach and algorithm used in countering spam have their own pros and cons. As there are no silver bullet in countering the threat of spam (Hoanca, 2006), the implementation of Needleman-Wunsch FPGA will co-exist with different algorithm that are implemented in software level to block spam.

The remainder of this paper is organized as follows. Section 2 review about the calculation of Needleman-Wunsch algorithm. In section 3, the implementation of the algorithm will be described in details and section 4 concludes with the results.

2. The Needleman-Wunsch Algorithm

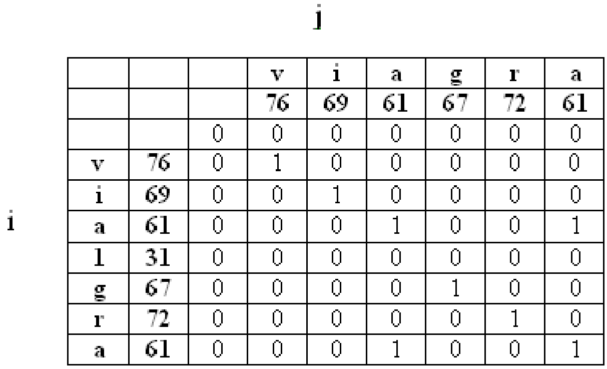
Needleman-Wunsch is one of the earliest algorithm implemented in bioinformatics. It is being selected because of its ability to detect slight differences in the string. The example below demonstrates how the slightly modified string S_1 which is commonly found in spam could be matched with signatures in the database. The algorithm consist three parts of calculation which is step (1) compute scores, step (2) compute main table and step (3) traceback alignment (Needleman and Wunsch, 1970).

Let S_1 and S_2 represent two string that will be used for comparison. The input of the two string which will be used in this example will be:

String S_1 – via1gra
String S_2 – viagra
S_1 in ASCII representation characters(hexadecimal) – 76 69 61 31 67 72 61
S_2 in ASCII representation characters(hexadecimal) – 76 69 61 67 72 61

The score table S and main table E is constructed with *i* representing string S_1 in vertical of the table and *j* representing string S_2 in horizontal of the table. In step 1, a match will be given a value of 1 on the S table and a mismatch with a value of 0.

To compute the main table E in step 2, the value of $E_{(i,0)}$ and $E_{(0,j)}$ is initialized to 0 where $0 \leq i < a-1$ and $0 \leq j < b-1$. *a* represent the length of string S_1 and b represent the length of string S_2.

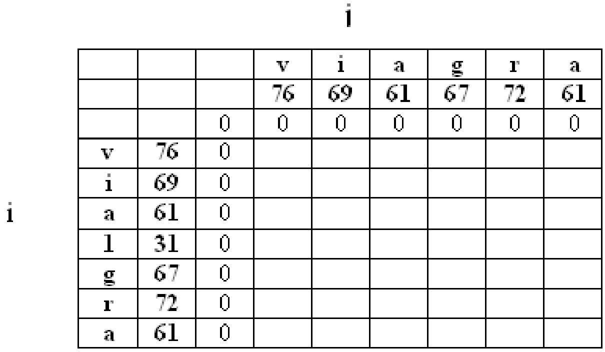


j

i

			v	i	a	g	r	a
			76	69	61	67	72	61
		0	0	0	0	0	0	0
v	76	0	1	0	0	0	0	0
i	69	0	0	1	0	0	0	0
a	61	0	0	0	1	0	0	1
1	31	0	0	0	0	0	0	0
g	67	0	0	0	0	1	0	0
r	72	0	0	0	0	0	1	0
a	61	0	0	0	1	0	0	1

Figure 1: score table S



j

i

			v	i	a	g	r	a
			76	69	61	67	72	61
		0	0	0	0	0	0	0
v	76	0						
i	69	0						
a	61	0						
1	31	0						
g	67	0						
r	72	0						
a	61	0						

Figure 2: initialization of table E(i,0) = E (0,j) = 0

Using the formula below, the table is contructed. To compute the value for position $E_{i,j}$, value from three other position will be considered which is $E_{i-1,\ j-1} + S_{i,j}$, $E_{i,j-1}+W$ and $E_{i-1,j}+W$. The highest value among this three position will be used for $E_{i,j}$. W represent the gap penalty for the alignment. Gap penalty is used to improve alignments between more distance sequences sometimes but for the purpose of spam analysis, a gap penalty of 0 produce accurate results (Beddoe).

$$E_{i,j} = \text{Max}\{$$

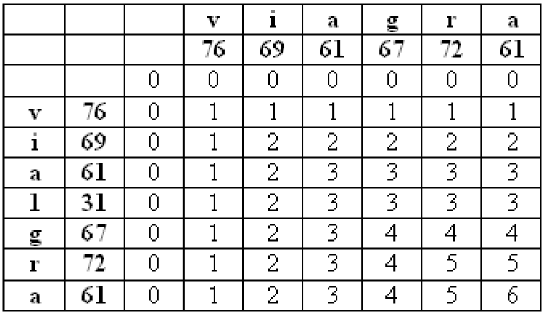
$$E_{i-1,\ j-1}+S_{i,,j},$$

$$E_{i,j-1}+W,$$

$$E_{i-1,j}+W$$

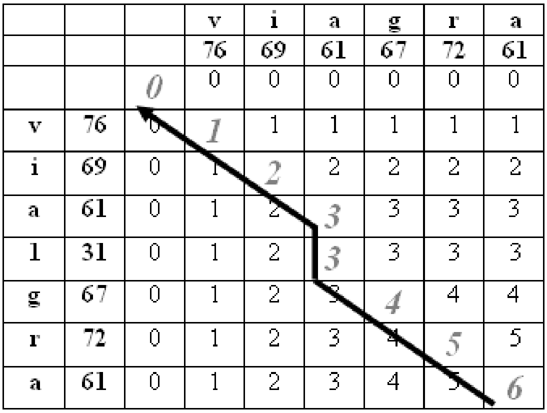
$$\}$$

The fully constructed table is shown in figure 3.



			v	i	a	g	r	a
			76	69	61	67	72	61
		0	0	0	0	0	0	0
v	76	0	1	1	1	1	1	1
i	69	0	1	2	2	2	2	2
a	61	0	1	2	3	3	3	3
l	31	0	1	2	3	3	3	3
g	67	0	1	2	3	4	4	4
r	72	0	1	2	3	4	5	5
a	61	0	1	2	3	4	5	6

Figure 3: main table with the computed value.



			v	i	a	g	r	a
			76	69	61	67	72	61
		0	0	0	0	0	0	0
v	76	0	*1*	1	1	1	1	1
i	69	0	1	*2*	2	2	2	2
a	61	0	1	2	*3*	3	3	3
l	31	0	1	2	*3*	3	3	3
g	67	0	1	2	3	*4*	4	4
r	72	0	1	2	3	4	*5*	5
a	61	0	1	2	3	4	5	*6*

Figure 4: main table E with the computed value and traceback.

For traceback alignment in step 3, the traceback point starts from bottom right. The formula below is used for traceback.

$$\text{Max}\{ E_{i-1,\ j-1}+S_{i,j},\ E_{i,j-1},\ E_{i-1,j} \}$$

From the formula shown above, if the MAX value comes from $M_{i-1,\ j-1}+S_{i,j,}$ the trace will move diagonally to the next point. If the MAX value comes from $M_{i,j-1}$ the trace will move to the left. If the MAX value come from $M_{i-1,j}$ the trace will move to one step above. The completed trace alignment is shown in figure 4 in grey and italic font.

The traceback yields the following results:

In ASCII	In normal character
76 69 61 31 67 72 61	v i a l g r a
76 69 61 5f 67 72 61	v i a _ g r a

3. Implementation

The design of the algorithm are written using VHDL programming language. For the Microblaze softcore, C programming language is being used. The VHDL design is then synthesized and imported into Xilinx Platform Studio. Using Xilinx Platform Studio, both the C programming and VHDL design is downloaded into Xilinx Virtex 4 ML-401 experiment kit. The ethernet cable are plugged into the kit and results are obtained.

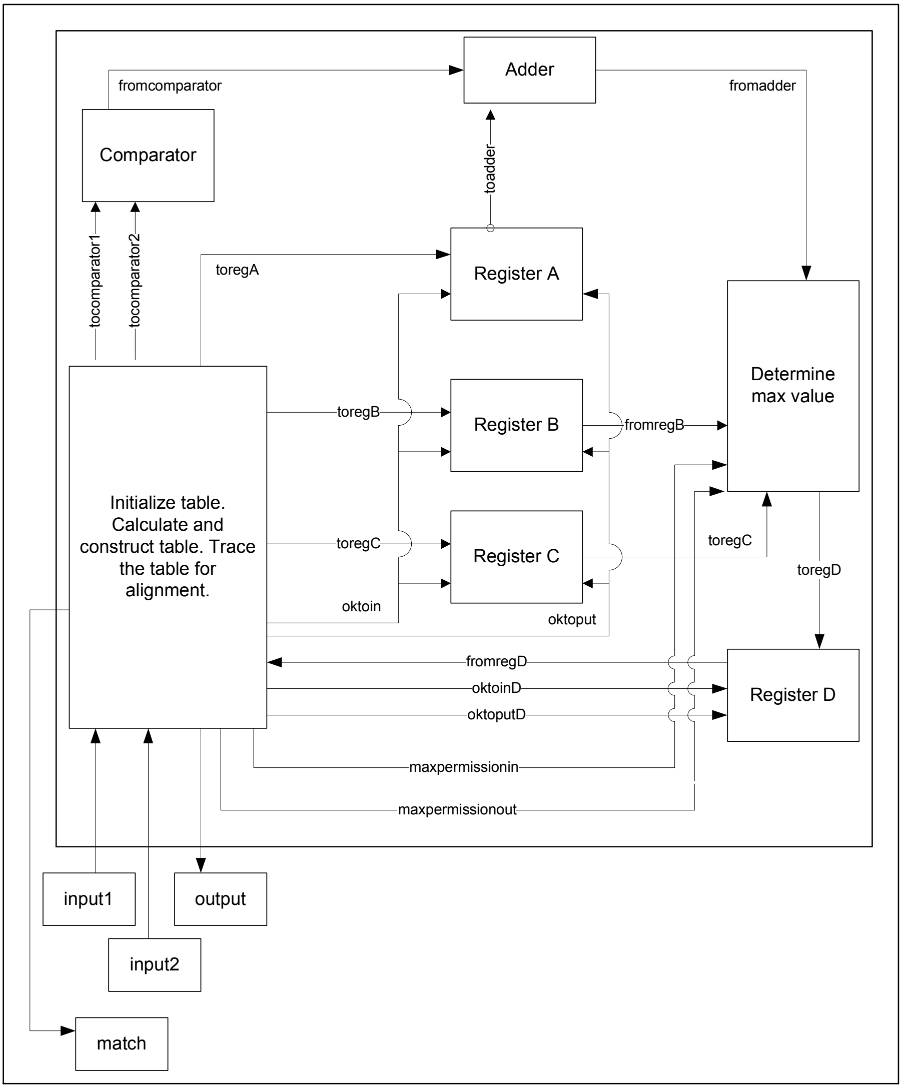


Figure 5: The block diagram for the VHDL system processor.

FPGA design are being chosen for its programmability feature (Emran and Yamani, 2006). Beside being programmable, FPGA provides performance advantages of almost the same as ASIC (Emran et al, 2007; Emran and Yamani, 2006). In comparison, software version of the algorithm is much less slower and it consumes a lot of memory and precious CPU cycles. This makes software version less suitable for high speed matching especially when it involves gigabit networks. Few examples of application of dynamic programming algorithms in FPGA are the Levenshtein Distance (Emran et al, 2007) and K-difference (Emran and Yamani, 2006). There are also application of algorithm from exact string family in FPGA which is the Knuth-Morris-Pratt algorithm (Yamani et al, 2007).

Figure 5 provides the block diagram of the design of Needleman-Wunsch algorithm in FPGA.

4. Conclusion

In this paper, a new technique to scan spam is being introduced. The algorithm design greatly reduces the computation time needed for blocking spam. The experimental results shows that implementing spam scanning in FPGA are one of the efficient way to reduce the spam problem. Further work would be required to refine the algorithm and conserve memory.

5. References

Beddoe, M. A., “Network Protocol Analysis using Bioinformatics Algorithms”, http://www.4tphi.net/~awalters/PI/pi.pdf, (Accessed November 20, 2007).

Chiu, Y.F., Chen, C.M., Jeng, B. and Lin, H.C., (2007), “An Alliance-Based Anti-spam Approach” *Natural Computation, 2007. ICNC 2007. Third International Conference*, Volume 4, 24-27 Aug. 2007 pp203-207 Digital Object Identifier 10.1109/ICNC.2007.173

Cumming, J.G. (2006), “Does Bayesian poisoning exist?”, http://www.virusbtn.com/spambulletin/archive/2006/02/sb200602-poison, (Accessed December 27, 2007).

Ferris Research, (2007), http://www.ferris.com/research-library/industry-statistics/, (Accessed January 2, 2007).

Hoanca, B., (2006), “How Good Are Our Weapons in the Spam Wars?”, *Technology and Society Magazine, IEEE,* Volume 25, Issue 1, Spring 2006 pp22-30 Digital Object Identifier 10.1109/MTAS.2006.1607720.

Idris, M.Y.I., Teng, Y.G. and Tamil, E.M., (2007), “Hardware-Based Worm Detection Design Using Knuth-Morris-Pratt Algorithm”, *Proceedings of the Conference on IT Research and Application (CITRA 2007)*, 4th April 2007, Selangor, MALAYSIA.

Kline, R.L. and Glinert, E.P., (2003), “Approximate Matching Algorithms for Music Information Retrieval Using Vocal Input”, *Proceedings of the eleventh ACM international conference on Multimedia MULTIMEDIA '03.*

Liu, Y., Johnson, J. and Vaidya, S. (2006), “GPU Accelerated Smith-Waterman”, *International Conference on Computational Science Reading*, United Kingdom.

Needleman, S.B. and Wunsch, C.D., (1970), "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of Molecular Biology,* 48:443-453.

Sanpakdee, U., Walairacht, A. and Walairacht, S. (2006), "Adaptive Spam Mail Filtering Using Genetic Algorithm", *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference,* Volume 1, 20-22 Feb. 2006 pp441–445.

SPAMHAUS.org, (2007) "The SPAMHAUS Project", http://www.spamhaus.org/ effective_filtering.html (Accessed January 4, 2007).

Tamil, E.M. and Idris, M.Y.I., (2006), "FPGA Based Approximate String Search Algorithm Implementation To Detect Polymorphic Worm", *In Proceedings of 3rd International Conference on Artificial Intelligence in Engineering and Technology (ICAIET 2006), 22-24 Nov 2006,* Sabah, MALAYSIA

Tamil, E.M., Idris, M.Y.I. and Heng, T.H., (2007), "FPGA Design of Spyware Inline Filter Using Levenshtein Distance Approximate String Search Algorithm", *In Proceedings of the SCORED 2007,14-15 May 2007,* Universiti Tenaga Nasional, MALAYSIA.

Wu, M.W., Huang, Y., Lu, S.K., Chen, I.Y. and Kuo, S.Y., (2005), "A Multi-Faceted Approach towards Spam-Resistible Mail Dependable Computing", *Proceedings. 11th Pacific Rim International Symposium on 12-14 Dec. 2005* Page(s):9 Digital Object Identifier 10.1109/PRDC.2005.8