Fast Hybrid String Matching Algorithms

Jamuna Bhandari[1]and Anil Kumar[2]

[1]Dept. of CSE, Manipal University Jaipur, INDIA

[2]Dept of CSE, Manipal University Jaipur, INDIA

ABSTRACT
Various Hybrid algorithms have been proposed using early string matching algorithms such as BM, KMP, Horspool, Quick searchand many others. Designing of fast string matching algorithms for different applications has started growing rapidly. In this area, many hybrid algorithms came up with new techniques of searching and rapid up the string matching process. This paper focus on five fast hybrid algorithms proposed in the year 2010-2012.

Keywords: hybrid, pattern, bad char, good suffix, string, prefix, text window.

1. INTRODUCTION

Hybrids algorithms receive the properties of original algorithms and perform string matching operations. Hybrids algorithms are projected for exact string matching[1],[3],[4] as well as approximate string matching[2],[5]. These algorithms are used for applications such as intrusion detection[6][7]biological sequence analysis[8]indeterminate string[9]virus scanning[10].The main aims behind the pattern matching/string matching algorithms are to reduce number of comparisons of characters of text and to reduce the time required mainly for worst case and average case. To strength in this area numerous hybrid algorithms have been proposed.

The paper is organized as follows; section 2 includes the working principles of fast hybrid string matching algorithms along with performance analysis and then conclude with section 3.

2. HYBRIDS STRING MATCHING ALGORITHMS

This section discussed the working principle of fast hybrid pattern matching algorithms and performance comparison of hybrid algorithm shows the performance difference with its parent's algorithms.

2.1. Hybrid pattern-matching algorithm based on Boyer-Moore and Knuth-Morris-Pratt algorithm.

In 2010, Hou Xian-feng et alproposed a hybrid algorithm using Boyer-Moore and Knuth-Morris-Pratt algorithms, this is known as KMPBS algorithm[11].

The hybrid KMPBS algorithm, search for the pattern P of length m from left to right within the text T of length n. Firstly, the last character of pattern P and the corresponding character of text T are compared. If matched, KMP algorithm used to match from left to right for rest of the characters.

If character does not match in first compare, then the characters under the text T of position (i + m-1)and character of position (i + m) of text T to determine the pattern string P of the moving location.
KMPBS algorithm greatly reduces the number of pattern shifting over the text T. The length of pattern P = 9.

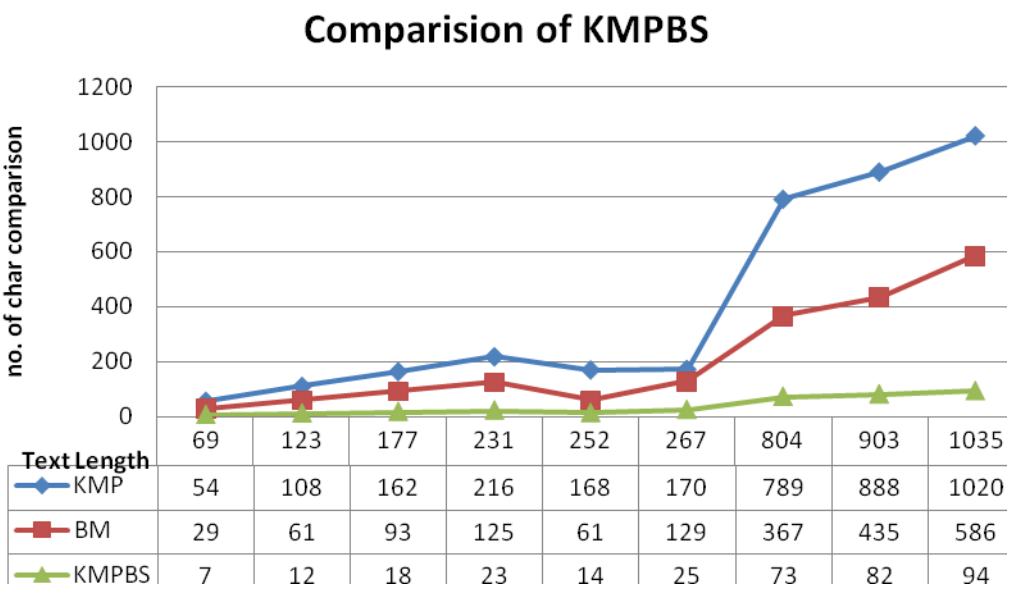


Figure 1: Pattern shifting of KMP, BM and KMPBS

Figure 1 show KMP algorithm and BM algorithm were compared with KMPBS. Here, KMPBS takes much less number of character comparison as compared with its parent algorithm KMP and BM. The comparison data and graph shows that the hybrid algorithm of KMP and BM algorithms KMPBS algorithm improves the matching efficiency comparing with its parent algorithm. The KMPBS algorithms achieve maximum shifting length, so that number of comparisons becomes less and work with less time complexity

2.2. A fast hybrid algorithm for the exact string matching problem.

In 2010 again, another fast hybrid algorithm have been proposed by Abdulwahab Ali et al using Berry-Ravindran[12] and Skip Search algorithm[13]. This algorithm is also called BRSS algorithm[14]. The hybrid algorithm build the two pre-processing phase first is built for bad charusing Berry-Ravindran method. Second phase created is bucket list, having all the locations of the characters that exist in the pattern and the text. If the character is not present in the bucket, then calculate the bad character shift value. If the bad character shift value is maximum than pattern length then use the bad character value for shifting, otherwise, shift by pattern length. If the character is present in the bucket then arrange the corresponding matched characters and then begin the comparing of characters from left to right.

When there is a match or mismatch occurs, figure out the shift value of the Skip Search in the first occurrence and secondly, calculate the bad character shift value from the two rightmost consecutive characters immediately after the text window.

If the bucket shift value is maximum than bad character shift value, then use the bucket shift value, otherwise, use the bad character shift value.

If the corresponding character is in the last position in the bucket, first calculate the bad character shift value from the rightmost two consecutive characters immediately after the window.
If the bad character shift value is maximum than pattern length than use the bad character shift value, otherwise, apply the shift value of the pattern length.

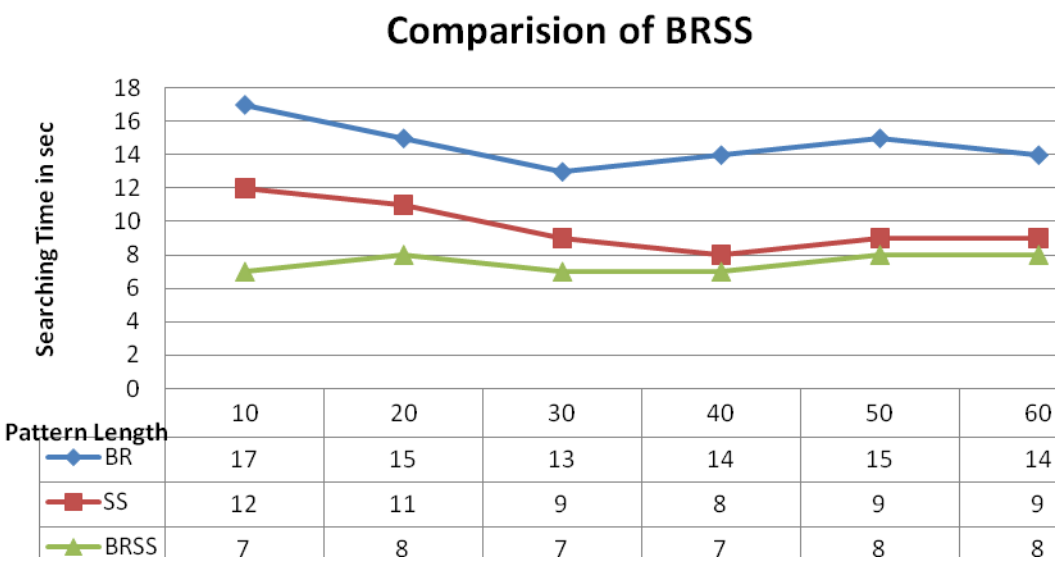


Figure 2: Searching time of BRSS with BR and SS

Figure 2 shows the performance difference of BRSS hybrid algorithm tested with BR algorithm and SS. The hybrid BRSS algorithm achieves good performance with number of character comparisons, number of attempts and searching time in applications pattern matching in DNA, Protein and English text. BRSS did character comparison, number of attempts and search of pattern in minimum time as compared to BR and SS algorithms.

2.3. A fast hybrid algorithm approach for the exact string matching problem via Berry Ravindran and Alpha skip search algorithms.

In 2011,Abdulwahab Ali proposed another hybrid using Berry Ravindran(BR) and Alpha Skip Search (ASS)[15]. This algorithm is commonly known as BRASS algorithm[16]. The pre-processing phase of this hybrid BRASS algorithm uses BR bad character shift values.BRASS algorithm observes the last three characters If they are not present in the text window then the pattern is shifted by BR shift value. If the characters are found within the pattern, positioned the matched characters correspondingly and then the next process is that the character comparisons are imitated from left to right of the window. At last, there will be a probability of a match or mismatch, shift the pattern by computing the BR shift value from the two right most consecutive characters immediately after the text window.

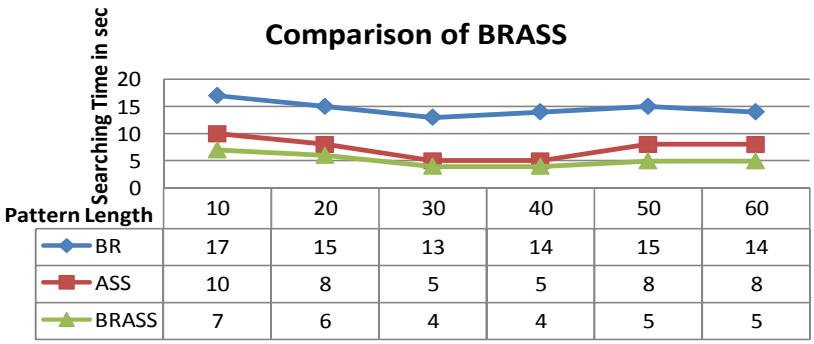


Figure 3: Comparison of BRASS algorithm with BR and ASS

The figure 3 shows the character comparison of BRASS algorithm with BR algorithm, ASS algorithm. The hybrid BRSS algorithm achieves good performance as compares with its parent algorithms.

2.4. Hybrid exact string matching algorithm for intrusion detection system

In 2012, Awsan Abdulrahman et al proposed Hybrid string matching for network security using Karp-Rabin[17] and Horspool algorithm[18]. This hybrid algorithm also work with two preprocessing phase. It uses the same preprocessing phases of its parent algorithm KP algorithm and Horspool algorithm for preprocessing. The need for KP hash function is to reduce the number of pre-processing character comparisons which decelerate of Horspool algorithm process. On the other hand the need for bmBc table is to reduce the number of attempts that consider the drawback in Karp-Rabin algorithm in long text searching. In the searching phase, the hybrid algorithm performs the comparison between the pattern and the text by utilizing the advantages of the hybrid algorithms. After the preprocessing phase, the comparison of pattern P and text T is done by comparing the hash value of pattern and text window T.

If the hash value of text window T and pattern P are not matched then the hybrid algorithms perform the shifting. Hybrid shifting uses the Horspool algorithm to avoid the sluggishness of Karp-Rabin algorithm(move to right side only by one character in every shift process). The hybrid algorithm shift to the right based on the values of right most character for the text window T in the bmBc table.

The process of hybrid algorithm continues until all characters in the text T are being compared and whether the mismatching or matching is found.

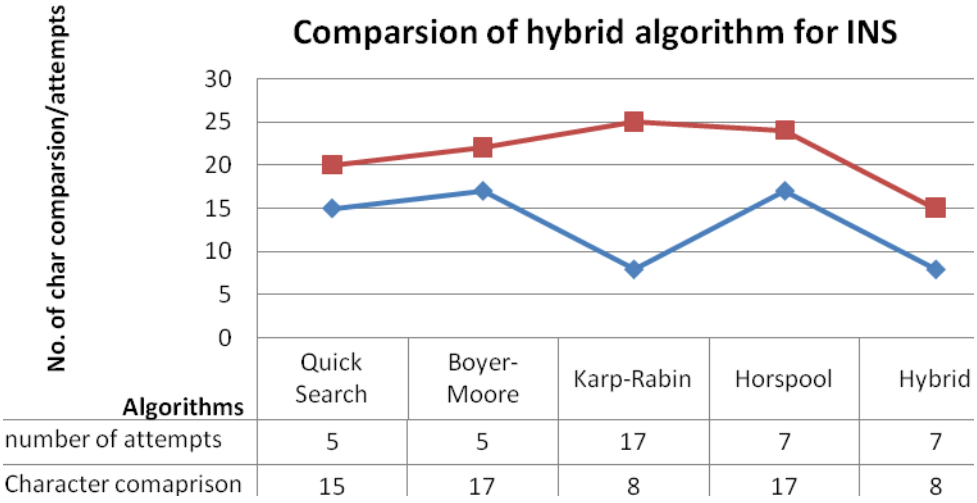


Figure 4: Character comparisons and number of attempts of Hybrid algorithm

Figure 4 shows the comparision of hybrid algorithm with Horspool, Quick search, Boyer-Moore and Karp-Rabin algorithms.Hybrid algorithm clearly shows comparison it performs better than its parent algorithm as well as some of fast string matching algorithms such as Quick search, Boyer-Moore algorithm. This improve the hybrid algorithm during the comparison process and at the same time it reduced the number of character comparison with the help of the hash function. This hybrid algorithm is introduced for network intrusion detection since this algorithm requires less time and space complexity.

2.5. Multithreaded implementation of hybrid string matching algorithm
In 2012,Akhtar Rasool et al proposed hybrid algorithm for multithreaded implementation using Boyer-Moore and Knuth-Morris-Pratt algorithms.

Firstly Multithreaded hybrid pattern matching algorithm[19] divides the string into two halve.
Secondly it merge the m-1 character (where m is the length of the pattern) from end side of first half with the m-1 character from beginning portion of second half as shown in example 1. This way total three text T will be in three sub parts T_{s1}, T_{s2} and T_3.

Example 1:

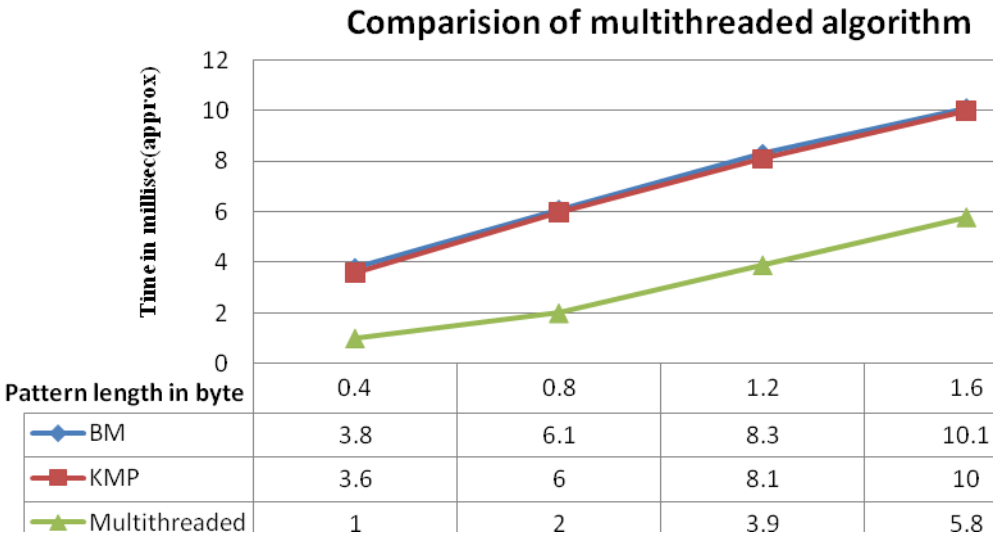
T: t_1, t_2, t_3, t_4 ……..t_n further divided into two sub parts

T_{S1}: t_1, t_2, t_3 …….. t_i T_{s2}: t_{i+1}, t_{i+2} ……..t_n now, merge m-1 character from both the sub parts

T_3: $t_j\ t_{j+1}$……t_{i+k} where $t_j \in T_{s1}$, $t_{i+k} \in T_{s2}$ and length of T_3

Thirdly, start pattern matching simultaneously with all the three sub text windows.

Lastly if pattern found, return the pattern else break the thread and stop. This hybrid algorithm creates the thread of the text and then performs comparisons of pattern and text characters.



Pattern length in byte	0.4	0.8	1.2	1.6
BM	3.8	6.1	8.3	10.1
KMP	3.6	6	8.1	10
Multithreaded	1	2	3.9	5.8

Figure 5: Performance comparisons of Hybrid algorithm with BM and KMP

Figure 5 shows the performance graph of multithreaded hybrid pattern matching algorithm with its parent algorithm BM, KMP. Multithreaded method for pattern matching is simpler than its patents algorithms. This hybrid algorithm performs better than them for short patterns. For long patterns size Knuth-Morris Pratt algorithm is a good choice.

3. CONCLUSIONS

The hybrid algorithm serve better outcome in terms of number of character comparisons and number of attempts done when searching of different data types with different pattern lengths than the original (parents) algorithms.

Acknowledgement:

This work is partially supported by HRD, Govt. of Sikkim (India), vide notification no. 166/SCH/EDN 2003, Ref. No 82/SCH/EDN, issued on 20/7/2013.

REFERENCES

1. Knuth, D. E., Morris, JR, J. H., and Pratt, V. R. 1977. Fast pattern matching in strings. SIAM J. Comput. 6, 1, 323–350, 1977
2. Patrick A. V. Hall. 1980. Approximate string matching, ACM
3. C. Charras, T. Lecroq, "Handbook of Exact String Matching Algorithms", http://www.ezdoum.com/upload/10/20020720023851/string.pdf, 2013.
4. Boyer, R. S. and Moore, J. S.1977, A fast string searching algorithm. Commun. ACM, 1977, 20, 762–772.
5. C. W. Lu, C. L. Lu, R.C.T. Lee, "A new filtration method and a hybrid strategy for approximate string matching", Theoretical Computer Science, Volume 481, 15 April 2013
6. P. S. Wheeler Techniques for improving the performance of signature based intrusion detectionsystems, Master's thesis, Universityof California Davis, 2006.
7. Awsan Abdulrahman Hasan, Nur'Aini Abdul Rashid. Hybrid Exact String Matching Algorithm for Intrusion Detection System. ICCIT, 2012.

8. Yong Huang et al, A Fast Exact Pattern Matching Algorithm for Biological Sequences. 978-0-7695-3118-2/08, IEEE DOI 10.1109/BMEI.2008.154, 2008.

9. W. F. Smyth and Shu Wang, An Adaptive Hybrid Pattern-Matching Algorithmon Indeterminate Strings. Supported in part by grants from the Natural Sciences & Engineering Research Council of Canada, 2008.
10. Po-Ching Lin et al. A Hybrid Algorithm of Backward Hashing and Automaton Tracking for Virus Scanning. IEEE transactions on computers, vol. 60, no. 4, April 2011.
11. Hou Xian-feng et al. Hybrid pattern-matching algorithm based on BM-KMP algorithm, 3rd International Conference on Advanced Computer Theory and Engineering(1CACTE), 978-1-4244-6542-2010 IEEE.
12. Berry, T. and Ravindran, S. 1999. A fast string matching algorithm and experimental results. In Pro-ceedings of the Prague Stringology Club '99, J. Holub and M. Sim ´anek, Eds. Czech Technical
13. C. Charras, T. Lecroq, and J. D. Pehoushek, "A Very Fast String Matching Algorithm for Small Alphabets and Long Patterns. Proceedings of the Ninth Annual Symposium on Combinatorial Pattern Matching," Lecture notes in computer science, vol. 1448, pp. 55-64,1998.
14. Abdulwahab Ali Al-mazroi and Nur'Aini Abdul Rashid, A Fast Hybrid Algorithm for the Exact String Matching Problem, American J. of Engineering and Applied Sciences 4 (1): 102-107, 2011.
15. Cantone, D., S. Cristo faro and S. Faro, 2004. Efficient algorithms for the δ-approximate string matching problem in musical sequences. Proceedings of the Prague Stringology Conference, (PSC'04), Universit`a di Catania, Italy, pp: 33-47.
16. Abdulwahab Ali Almazroi, A Fast Hybrid Algorithm Approach for the Exact String Matching Problem Via Berry Ravindran and Alpha Skip Search Algorithms. Journal of Computer Science 7 (5): 644-650, 2011.
17. R. M. Karp, and M. O. Rabin, "Efficient randomized pattern-matching algorithms", IBM Journal of Research and Development, Vol. 31, no. 2, pp. 249-260,1987,.
18. Horspool, R. N. 1980. Practical fast searching in strings. Softw. Pract. Exp. 10, 6, 501–506
19. Akhtar Rasool et al. Multithreaded Implementation of Hybrid String Matching Algorithm. International Journal on Computer Science and Engineering (IJCSE), Vol. 4 No. 03