An Enhanced Boyer-Moore Algorithm for Worst Case Running Time

Manjit Jaiswal[1], Dr. Nilay Khare[2],
Department of Computer Science and information Technology,
[1&2]*Maulana Azad National Institute of Technology, Bhopal,462051,India*
[1]manjit.jaiswal222@gmail.com
[2]nilay.khare@radiffmail.com

***Abstract*—This article adderesses the exact string matching problem which consists in finding all occurrences of a given pattern in a text.It is an extensively studied problem in the field of computer science mainly due to despite its popularity in diverse area of application such as cluster computing, image and signal processing, speech analysis and recognition, information retrieval, data compression,computational biology,intrusion detection and virus scanning detection.In the last decade several new algorithm has been proposed.In this paper we compares all improved of the Boyer-Moore algorithm with my enhanced Boyer-Moore algorithm practically and theoretically result.It is not only generate the largest distance but also produces the minimum shifting and frequency of comparisons steps.By this enhanced algorithm we can reduce the number of comparisons frequency and number of shifting steps during the searching process.Moreover result of this enhanced Boyer-Moore algorithm reveals the efficiency is higher than of previous improved Boyer-Moore algorithms and time complexity is reduced in the concept of worst case analysis and lower than BM algorithm.Our enhanced algorithm 16% boost-up than previous improved Boyer-Moore algorithm when executed on the CPU.This enhanced Boyer-Moore algorithm can be plays an important role in finding extremely fast genetic moleculer and complex sequence pattern of interested database alignment of DNA.**

***Index Terms*—Moving distance,BM,Index value,Linear worst case Searching pattern,Enhanced algorithm,CUDA.**

I. Introduction

The fastest known exact string matching algorithm are based on the Boyer-Moore idea[BoM 77].The Boyer –Moore algorithm is a string searching algorithm.It was developed by Bob Boyer and J strother algorithm in 1977 as in [13].
Bob Boyer and J Strother Moore are both part of the computer science faculty team at the University of Texas at Austin.
BM algorithm has high efficiency and sublinear on the average due to it generally does not compare each character of the text with the search string.

Menuscrip received February 24,2012.The author are with the Computer Science and Information Technology,Bhopal,India.
This work was supported by the Maulana Azad National Institute of Technology,Bhopal(India).

DOI: 10.5176_2010-2283_2.1.143

Moreover it uses the preprocessing phase about the search string to help eliminate unwanted comparisons.BM algorithm works fast when larger is the alphabet and longer is the pattern.Boyer-Moore algorithm an important role play towards the network security as in [3].It combines the network security audit system and improves the original BM algorithm to detect the intrusion.Our enhanced algorithm is based on the running time executed on the CPU which means the running time of that enhanced algorithm is less than as in [8].BMH algorithm is an improved part of BM algorithm.Thus BMH in the concept of practically is more suitable than BM algorithm.The BMHS algorithm has enhanced the BMH algorithm as in [7] which is used the bad character features of BMH algorithm.Like the BM and BMH the BMHS always assumes its the best case time complexity i.e if every time in the first comparison a text symbol is found that does not occur at all in the pattern.BM algorithm does not use the current matching information but in the literature as in [2] it used.This paper is organized as follows.Section two briefly describes the related work in BM based string searching algorithm.Section third describes our enhanced BM algorithm and section fourth explain the working of our enhanced BM algorithm in the term of worst case analysis.Section fifth draws the experiment result of my enhanced BM algorithm based on the as in [8]-[7]. Eventually section VI contains the conclusion of our work.

II. Related Work

A. Principle of Boyer-Moore algorithm

The BM algorithm successively aligns P with T and then checks whether P matches the opposing characters of T. However BM algorithm contains three clever ideas not contained in the KMP or Naïve algorithm.

- Right to left scan
- The bad character shift rule
- Good suffix shift rule.

Boyer-Moore inspired by the KMP algorithm.Practically ,BM algorithm is faster than KMP algorithm about 2 to 4 times as in [11] and efficiency of the BM is higher than KMP.
We define the string matching process by pattern string P=P[0],P[1],P[2]……P[j] in the text string T=T[0],T[1],T[2]……T[i],Where T[i] (0≤i≤n-1) and P[j]

$(0\le j\le m-1) \in \sum$ and $m\le n$.$\sum$ is the set of the character $\sum$=C[0],C[1],C[3]……C[k-1].Now if the given pattern string exists in the text string then we got match otherwise match unsuccessful.Boyer-Moore's algorithm preprocess define by the pattern *P* and the alphabet $\sum$ to build the index function (Index) mapping $\sum$ to integers,where Index[c] is defined as:

$$\text{Index}[c]=\begin{cases} j, \text{ if } j \text{ of } c \text{ in the pattern P is the last occurrence position value where } 0\le j\le m-1 \\ -1, \text{otherwise} \end{cases} \quad (1)$$

where m is the number of character in the pattern i.e. length of the pattern and n is the length of the text.

B. Idea

The algorithm of BM compares the pattern with the text from right to left.If the symbol that is compared with rightmost pattern symbol does not occur in the pattern at all then the pattern shifted by m positions towards the right of text symbol as in [10].These instance show in table I below illustrates above situation.

Instance: Table I. Boyer-Moore

0	1	2	3	4	5	6	7	8	9
B	A	A	B	C	B	A	B	D	A
A	B	A	B	D					
					A	B	A	B	D

The first comparison C-D at position 4 produces a mismatch.The text symbol C does not occur in the pattern.Hence the pattern can not match at any positions of the current text windows,therefore the pattern will be shifted at position 5.The best case for the Boyer-Moore algorithm requires only *O*(n/m) iff each attempt the first compared text symbol does not occur in the pattern.

C. Bad character heuristics

This method is called bad character heuristics.It can also be applied if the bad character i.e. the text symbol that causes a mismatch,occurs somewhere else in the pattern.Then the pattern will be shifted so that it is aligned to this text symbol.

For the bad character rule describes here by the formula is given as in [3] below:

$$\text{Skip}(c)=\begin{cases} n; k\ne P[j], \text{where } 0\le j\le n; k\notin P \\ \text{else } n-j, j=\max(k);\{j|P[j]=k, 0\le j\le n \end{cases} \quad (2)$$

The following example describes this situation in tableII.

Instances: Table II. Bad character rule

0	1	2	3	4	5	6	7	8	9
A	B	E	B	C	B	A	C	D	A
A	C	A	E	D					
			A	C	A	E	D		

Comparison C-D causes a mismatch .Text symbol C occurs in the pattern at the positions 1,thus pattern will shifted so that rightmost C in the pattern is aligned to text symbol C.

D. Good suffix heuristics

When bad character heuristics fails then in this situation the comparison A-B causes a mismatch. An alignment of the rightmost occurence of the pattern symbol A with the text symbol A would produce a negative shift. Instead, a shift by 1 would be possible. However, in this case it is better to derive the maximum possible shift distance from the structure of the pattern. This method is called good suffix heuristics as in [3].

Instance: Table III. Good suffix rule

0	1	2	3	4	5	6	7	8	9
A	B	A	A	B	A	B	A	C	B
C	A	B	A	B					
		C	A	B	A	B			

The suffix AB has matched. The pattern can be shifted until the next occurence of AB in the pattern is aligned to the text symbols AB, i.e. to position 2.In the table III shows following situation the suffix AB has matched. There is no other occurence of AB in the pattern.Therefore, the pattern can be shifted behind AB, i.e. to position 5. For instance: Table IV. Case 2 of good suffix rule

0	1	2	3	4	5	6	7	8	9
A	B	C	A	B	A	B	A	C	B
C	B	A	A	B					
					C	B	A	A	B

In the following situation the suffix BAB has matched. There is no other occurence of BAB in the pattern. But in this case the pattern cannot be shifted to position 5 as before, but only to position 3, since a prefix of the pattern (AB) matches the end of BAB. We refer to this situation as case 2 of the good suffix heuristics.For good suffix rule there is formula as in [3] follows:

$$\text{Shift}(j)=\min\{S|(P[j+1\ldots.n]=P[j-s+1\ldots..n-s])\&\&(P[j]\ne P[j-s])$$
$$(j>s),P[s+1\ldots n]=P[1\ldots n-s](j\le s)\} \quad (3)$$

E. BMH

Boyer-Moore Horspool in the practical application is much more suitable than BM algorithm in literature as in [9].BMH algorithm is not only used the bad character shift but also right most character of the current text window i.e. distance of the shift to right is determined by the character in the text string which is aligned to the last one of the pattern string. BMH uses the less space and has a simpler inner loop with less constant overhead per iteration.Its average performance is O(n) but it has worst case performance of O(m*n).In the best case performance time complexity is O(n/m).

For instance: Table V. BMH rule

0	1	2	3	4	5	6	7	8	9
A	B	C	A	E	E	A	A	C	B
E	C	A	D	E					
				E	C	A	D	E	

F. BMHS

Like the BM and the BMH algorithm the BMHS algorithm assumes its best case,if every time in the first comparisons a text symbol is found that does not occur at all in the pattern.BMHS is an improved algorithm and it is based on BMH algorithm.Pattern matching intrusion detection BMHS algorithm maximum moving distance times of comparing.Generally BMHS algorithm faster than the BM and

BMH algorithm as in [7].Moreover the shiftin distance of the BMHS algorithm is estimulated by the character in the text string which is under the character aligned to the last one of the pattern string,if it is not find in the pattern string then distance to the move to right side is m+1. BMHS algorithm worst case time complexity is O (mn), the best case time complexity is O (n / m +1).

For instance Table VI. BMHS rule

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
A	*B*	*C*	*A*	*B*	*D*	*A*	*A*	*C*	*B*
B	*C*	*A*	*A*	*B*					
						B	*C*	*A*	*A*

At all position of this pattern D does not occur,thus the pattern will be shifted past this position.

III. DESIGN OF OUR ENHANCED BOYER-MOORE ALGORITHM

First some make some assumption and definition:There are two string text(T) and pattern(P) needed to be matching process where,

P=P[1],P[2]…….P[j] and |P|=j where 0≤j≤ m |m=length of the pattern.

T=T[1],T[2]……..T[i] and |T|=i and 0≤i≤ n |n=length of the text.

The preprocessor table which is described refer to (1) always used by the enhanced BM algorithm.We will take an experimental example here for mention all about string matching process by enhanced BM algorithm.

The main key concept of the BM algorithm is the MovDist function which is keep the numeric value called as movement distance(MovDist) and it is defined as follows:

MovDist={i+m-MIN(j,1+Index[T[i]])} (4)

MovDist shows and provide the an index value such as if at any position mismatch occurs between the character of text and pattern then MovDist calculate the how many move the pattern toward the right direction of the text.

Where,Function Index[c] have mentioned refer to (1) and where

0≤j≤m-1|m=length of the pattern string.

0≤i≤n-1|n=lengh of the text string.

Preprocessing phase is calculated by taken the an array for Index[c].Moreover there is not only preprocessing phase for pattern but also for some time required to text string .MovDist function is consist by the bad characteristics rule and addition of an integer type value taken from at which text position we will start the pattern matching process.We will start the searching process with the text from there, where that character of the text should be available in the pattern string which is described in the table X and this process is achieved by taken an integer type variable to keep the such type of information that how many we moved towards the right side of the text from initially position to search the available character in the pattern.The improved algorithm and searching process described here:

Enhanced BM Algorithm (T,P)

Input:text length n and pattern length (m) Compute Index[c] function that is preprocessing phase.

```
Main()
{                                   Preprocessing phase,match the character of the text(t) with the pattern(p) starting with the zero position from line 1 to 2.
  1. for i=0 to n-1
{
    1.1.  for j=0 to m-1
        {
    1.2.    if( P[j]=T[i]) then
    1.3.    goto label;
            }
    1.4.      t=i+1;
        }
  2.   label:
    2.1.    if t=n then
              {
    2.2.      print "we did not find the match".
        }
    3.      i=m+t-1;
     3.1.   j=m-1;
            Repeat                  //searching process start
     3.2.  if P[j]==T[i] then
     {
     3.3.    if j=0 then
     3.4.      Return i;  //we got match
        }
     3.5.   else
          {
     3.6.       i=i-1;
     3.7.       j=j-1;
        }
  4.    else
        {
     4.1.       for i=i+1 to  n
    {
     4.2.       if Index[i]=-1then      //refer to "( 1)"
      {
     4.3    i=i+1;
      }
     4.4   else
     4.5.          break;
}
  5.   i=i+m-Min(j,1+Index[T[i]]);  //refer  to " (4)"
    5.1       j=m-1;
}
   5.2          until  i> n-1;
}
          print "we did not get match".
```

Enhanced BM algorithm describes as following:

- First implement the preprocessing table as Index[c] function reveals in the table VII.In preprocessing phase,time complexity is $O(m+\sum)$.Our preprocessing phase is estimation and searching the position at the text where we start the string searching process of the pattern.

- Begin with the matching character of the text string with the pattern string ,if there are successive characters which are not in the pattern string they are ignored.We start the comparison from at which position of the text that are available in pattern string.For instance table X reveals that how to ignored those type of the character .

- After then we start the searching process,compare the character of the pattern with text string and when P[m-1] and T[k] match successfully,compare the character P[m-2] with character T[k-1].If the match successfully ,continue to comparing P[m-3] and T[k-2] and so on.if the any instant of time mismatch occur at position T[k-3]and P[m-4] then we should decide the distance to move according to MovDist function refer to (4).

- But we also check the index value of the position T[k-2] by Index[c] function and if there is value -1 then value of MovDist would be changed and by addition of the number of the occurences value which have Index function value is -1 otherwise the MovDist value will be according to the MovDist function refer to enhanced algorithm. Here is a flow chart of enhanced algorithm show above in the figure one:

- But we also check the index value of the position T[k-2] by Index[c] function and if there is value -1 then value of MovDist would be changed and by addition of the number of the occurences value which have Index function value is -1 otherwise the MovDist value will be according to the MovDist function refer to enhanced algorithm. Here is a flow chart of enhanced algorithm show below in the figure one:

IV. WORKING OF ENHANCED BM ALGORITHM

First,we take an example for experiment and according to that example define a function Index[c] then put the results into an array Index[c]. The function Index[c] are described in table VII as follows:
For instance :

Text string: THIS IS A VERY GOOD BOY
Pattern string: BOY

TABLE VII. VALUE OF Index[c]

C	*T*	*H*	*I*	*S*	*A*	*V*	*E*	*R*	*Y*	*G*	*O*	*D*	*B*
Index[c]	*-1*	*-1*	*-1*	*-1*	*-1*	*-1*	*-1*	*-1*	*2*	*-1*	*1*	*-1*	*0*

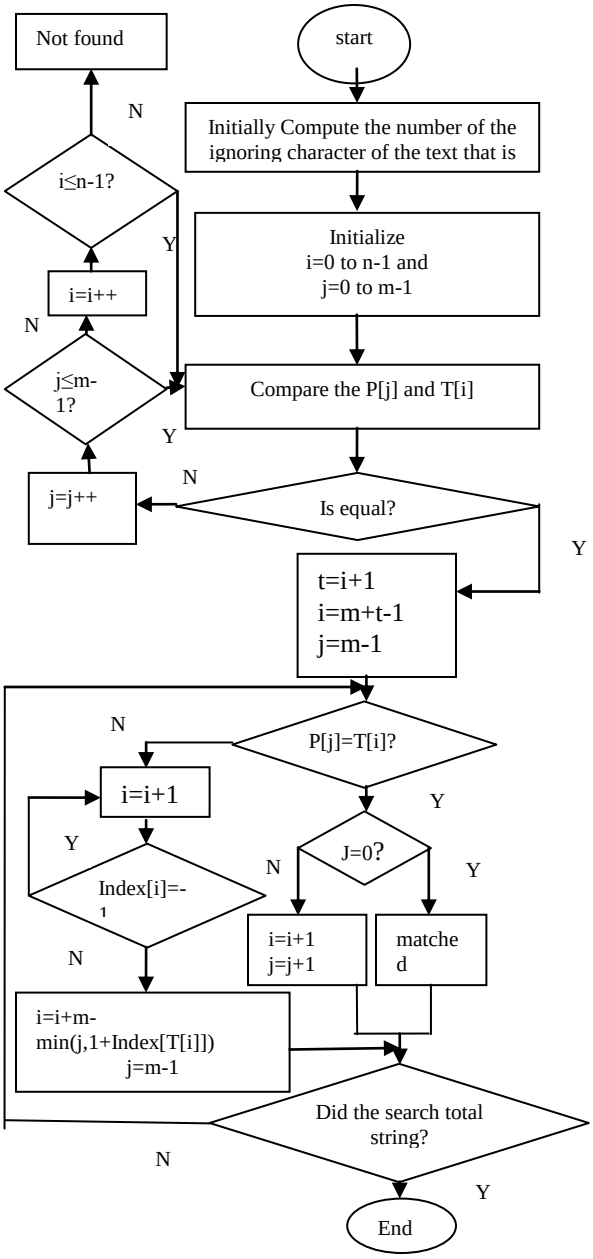

Figure 1. flow chart of enhanced BM

In the above table VII of Index[c] indicates that whose character which have -1 Index value then these all character did not appeared in the pattern string.
Table VIII to table X shows that how to matching processes of BM, improved BM and enhanced BM algorithm being.

TABLE VIII. BM ALGORITHM

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*
Text	*T*	*H*	*I*	*S*		*I*	*S*		*A*		*V*	*E*	*R*	*Y*		*G*	*O*	*O*	*D*		*B*	*O*	*Y*
pattern	*B*	*O*	*Y*																				
				B	*O*	*Y*																	
							B	*O*	*Y*														
										B	*O*	*Y*											
													B	*O*	*Y*								
																B	*O*	*Y*					
																	B	*O*	*Y*				

																			B	*O*	*Y*	
																				B	*O*	*Y*

In this BM method the pattern string shift 8 time and compares of character 11 times as in [12].

TABLE IX. IMPROVED BM ALGORITHM

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*
Text	*T*	*H*	*I*	*S*		*I*	*S*		*A*		*V*	*E*	*R*	*Y*		*G*	*O*	*O*	*D*		*B*	*O*	*Y*
Pattern	*B*	*O*	*Y*																				
														B	*O*	*Y*							
																	B	*O*	*Y*				
																					B	*O*	*Y*

In this method the pattern string shift 3time and compares of character 6 times as in [9].

Table X. OUR ENHANCED BM ALGORITH

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*
Text	*T*	*H*	*I*	*S*		*I*	*S*		*A*		*V*	*E*	*R*	*Y*		*G*	*O*	*O*	*D*		*B*	*O*	*Y*
Pattern														*B*	*O*	*Y*							
																	B	*O*	*Y*				
																					B	*O*	*Y*

In this method the character T,H,I,S,A,V,E,R are not in pattern string that why these are ignored and starting the search process from as in [13].Thus shifted 2 time and compares of character 5times.

A. Analysis of worst case time complexity of enhanced algorithm

According the BM algorithm if the longer pattern then there are less comparison needed.In the enhanced algorithm average time taken atmost O(n).In the worst case analysis if there are constant number of matches of the pattern in the text then Boyer moore algorithm takes O(n) comparisons in the searching phase.Moreover if the pattern did not match with the text string then in this situation comparisons are till the last character of the pattern and displacement is only one then time taken to comparisons in worst case analysis of the algorithm with the pattern is (n-m+1)*m as in [8].In preprocessing phase time required atleast $O(m+\sum)$,where $\sum$ is the alphabet.In the searching phase when we consider the best case (all the characters in the pattern is completely different compared to the characters in the text),matching m characters of the pattern in the text (see example below) yields a shift of m+1 at each attempt and hence the time complexity is $O([(n/(m+1))])$.

For instance:

Text: mnmnmnmnmn

Pattern: ssss

V. EXPERIMENT RESULTS AND ANALYSIS

An experimental environment (CPU: Intel core i7 with 4.0 GHz, Memory RAM: 4GB, OS: Windows 7 with 32 bit) to test the enhanced BM algorithms. Meanwhile,we used two counter variable by which it could be count the number of shifting and comparisons among the text and pattern character.we take 22 Kbyte,44 Kbyte,66 Kbyte and so on text string and 4 byte character of pattern string.We can see there that enhanced BM algorithm is doing matching process and taken approximately 13-16% less time than improved BM algorithm.Results are show below in the table XII to XIV.

TABLE XI. EXPERIMENT RESULTS

	The number of comparison frequency	*The number of shifting frequency*	*Text Size(Kbytes)*	*The run time of CPU(ms)*
BM algorithm	1832	434	22	9.8903
Improved BM algorithm	1601	293	22	7.1002
Enhanced BM algorithm	1071	223	22	6.6201

TABLE XII. EXPERIMENT RESULTS

	The number of comparison frequency	*The number of shifting frequency*	*Text Size(Kbytes)*	*The run time of CPU(ms)*
BM algorithm	1832	434	44	9.8903
Improved BM algorithm	1601	293	44	7.1002
Enhanced BM algorithm	1071	223	44	6.6201

TABLE XIII. EXPERIMENT RESULTS

	The number of comparison frequency	*The number of shifting frequency*	*Text Size(Kbytes)*	*The run time of CPU(ms)*
BM algorithm	1832	434	22	9.8903
Improved BM algorithm	1601	293	22	7.1002
Enhanced BM algorithm	1071	223	22	6.6201

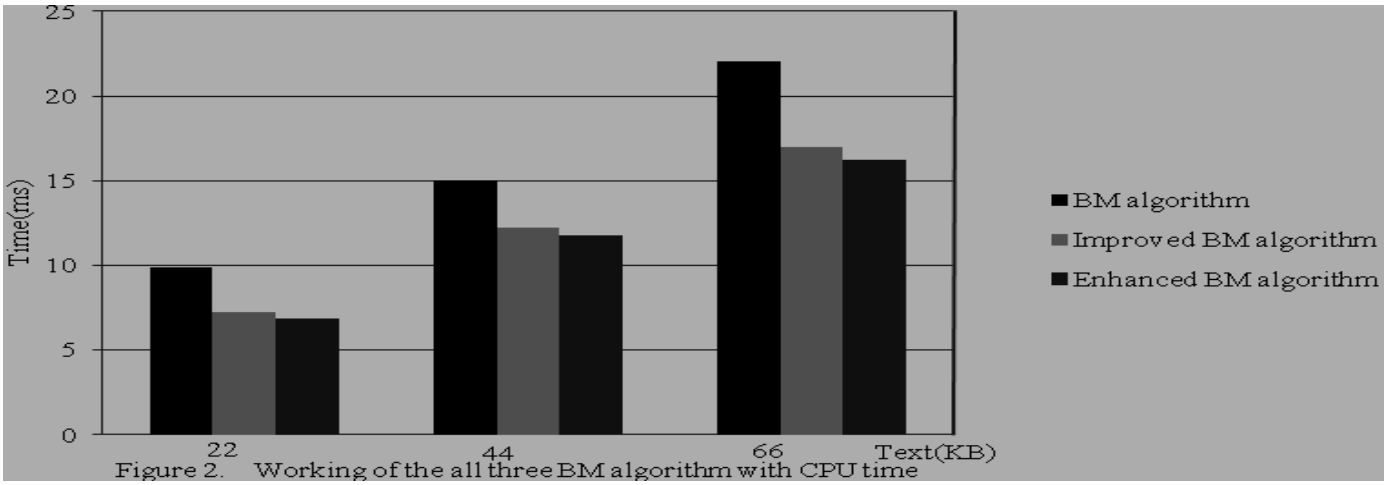

Figure 2. Working of the all three BM algorithm with CPU time

VI. CONCLUSION

In the worst case analysis actual time requires for BM O(nm+∑) but if there are only a constant number of matches of the pattern in the text,the Boyer-Moore searching algorithm perfoms *O*(*n*) comparisons.Normally best time complexity of BM,BMH and BMHS algorithm O(n/m),O(n/m) and O(n/m+1).Any string matching algorithm in the worst case must be examined at least n-m+1 character of the text.The best time complexity of proposed enhanced algorithm is O(n/m+1). But still we can improve the common preprocessing phase performance of BM algorithm by the use of another improved BM algorithm.We can also improve the performance of BM algorithm by using compute unified device architecture over(CUDA) the graphics processors unit(GPU).

Acknowledgment

This work was totally supported by the Maulana Azad National Institute of technology,Bhopal(India).Special thanks are to be Dr. Nilay Khare.He gave to many suggestions.

References

[1] Lingling yuan,"An improved algorithm for Boyer-Moore string MatchingChinese Information Processing" , IEEE.pp. 182-184,2011.
[2] Zhengda Xiong,"A composite Boyer-Moore Algorithm for the string Matching Problem" IEEE.pp. 492-496,2010.
[3] Xingxing Wang," A BM algorithm oriented on Network Security Audit System" IEEE.978-1-4244-5895,2010.
[4] Yang Tong,Qiao Xiang-dong, "Analyze and Improvement of BM Algorithm" IEEE:978-1-4244-3693,2009.
[5] Prasad JC,Dr.K.S.M. Panicker,"Single Pattern Search Implementations in a Cluster Computing Environment" , IEEE.pp391-396,2010.
[6] Knuth,D.E,Morries,Jr.,J.H.,and pratt,V.B." fast pattern matching in string SIAM J.comptng.6,2(1977),pp323-350.
[7] Yuting Han,Guoai Xu "Improved algorithm of pattern matching based on BMHS",IEEE.pp238-241,2010.
[8] Zhu Yong giang,"Two enhanced BM algorithm in pattern matching", IEEE.pp341-346,2011.
[9] Yihui SHAN, Yuming JIANG, Shiyuan TIAN, "Improved Pattern Matching Algorithm of BMHS for Intrusion Detection". Computer Engineering, vol.35, 2009, pp.170-173
[10] Zhanjun REN, Quanzhu YAO, Xiaofeng WANG, Youjiao ZOU,"Application of Pattern Matching Algorithm in Intrusion Detection Technique". Modern electronic technology, vol.2, 2009, pp.63-67
[11] Lianying MIN, Tingting ZHAO, "Improvement based on BM algorithm". Journal of Wuhan University of Technology (Transportation Science & Engineering), vol.30, 2006, pp.528-530.
[12] Baishuhong. Eason, An Impovement on BM Algorith for Chinese, fujiandiannao, pp. 90–91, october. 2009.
[13] Yi Ping, Jiang Xinghao, Wu Yue and Liu Ning. Distributed intrusion detection for mobile ad hoc networks. Journal of systems engineering and electronics, 19(4):851-859, 2008.

AUTHORS PROFILE

Associate Prof. Dr. Nilay khare is a senior faculty member at the MANIT ,Bhopal,India . He received his M.Tech. degree from Indian Institute of Technology,(IIT)Delhi,India. At, present he is head of department of Computer science and engineering and information technology in the MANIT,Bhopal,India.He had authored more than 50 research papers of national and international journal.His interests include toward of Wireless networks and theoretical computer science.He had more than 21 years work experience.

Manjit Jaiswal is pursuing Master of Technology in Computer Science and Engineering (2010-2012)from Maulana Azad National Institute of technology, Bhopal(462051),India.He did his B.E. degree in Computer Science and Engineering from Government Engineering College,Raipur(492001),India.His research interest towards the area of String matching,Single-core,Multi-core Architectures and compute unified device architecture(CUDA) programming over the GPU.