A Comparative Analysis of Hashing-Based Algorithm for Data Replication Control in the Cloud computing environment

Dr. MD. Rafeeq[1], Dr. K. Rajeshwar Rao[2],Dr. Subhash Chandrda[3]

1. Associate Professor, Department of Computer Science and Engineering, CMR Engineering College,Medchal Rd, Kandlakoya, Medchal(D),TS.
2. Professor, Department of Computer Science and Engineering, Siddhartha Institute of Engineering and Technology, Ibrahimpatnam Hyderabad.
3. Professor, Department of Computer Science and Engineering, CVR College of Engineering, Manglapalli, Ibrahimpatnam R.R(D) TS.

Abstract

Management of a huge assortment of replicated information incentralized or distributed environments is very importantfor many systems that give data processing,mirroring, storage, and content distribution. In itssimplest kind, the documents are generated,duplicated, and updated by emails and websites.Although redundancy might increase the dependableness at alevel, uncontrolled redundancy aggravates theretrieval performance and may well be useless if thereturned documents are obsolete. Documentsimilarity matching algorithms don't give information on the variations of documents, and filesynchronization algorithms are sometimes inefficient andignore the structural and syntactical organization ofdocuments.Data deduplication is one of the techniques used to solve the repetition of data. The deduplication techniques are generally used in the cloud server for reducing the space of the server.

Keywords: Data Deduplication, Processing, Replication control, Hashing, Distributed computing.

DOI Number: 10.14704/nq.2022.20.11.NQ66267 **NeuroQuantology 2022; 20(11): 2652-2658**

Introduction

The WWW is an easy way to obtain access to a wide variety of outlets. The congestion and surcharge are typical issues with WWW. The deduplication of information should be done in a way to boost the overall performance of the internet and minimize the loading time of the user. A framework is provided in to determine the efficiency of the website's proxy cache in reducing internet redundancies.

This also supported the protocol-impartial method of eliminating redundant visitors to the network and masking the absence of web proxy cache. The protocol supplied is independent of the float records application protocol, the protocol supplied will facilitate the flow of FTP data, stream media, emails, etc.

Reduction in data size on storage devices: while in recent times, cassette libraries with accurate deduplication have been commonly used for backup information garages. The disk-based, fully deduplicated backup systems compete with record deduplication in the tape libraries. Since they have higher runtime and greater efficiency of use.

In addition, lower data recovery from mainly disk-based alternate equipment takes lower time and is quicker and more productive. The data domain deduplication file system (DDFS). The paper gives a five-layer gadget and uses the deduction of the variable bite level.

The interface layer with various protocols to access different data kinds, including: Network File System (NFS), CIFS and the Virtual Tapes Library (VTL). The interface layer is a network file system.

eISSN 1303-5150

Analysis

A string-matching algorithm is used to find matches between specified string or text T and pattern P, where T is longer or equal to P. It is either precisely matched or partially matched with the example, in view of this, string matching algorithms are classified as Exact string-matching algorithms and Approximate string-matching algorithms.

Exact string-matching algorithms are worried about the number of events of the pattern in a given text, while Approximate string-matching algorithms are worried about the likeness rate between the pattern and the text or any part of the text.

Hashing (Rabin-Karp) matcher is used to find a numeric pattern P from a given text T. It firstly divides the pattern with a predefined prime number q to calculate the remainder of pattern P.

Then it takes the first m characters from text T at first shift s to compute remainder of m characters from text T. If the remainder of the pattern P and remainder of the text T are equal, only then we compare the text with pattern otherwise there is no need for comparison. We will repeat the process for next set of characters from text for all possible shifts which are from s=0 to n-m.

- **Naïve** pattern looking is the least complex technique among other example looking through calculations. It checks for all the characters of the primary string to the example. This calculation is useful for simple words. It need not bother with any pre-handling stages. it can check substring by checking once for the string. In figure 1 shows the approach of pattern matching.
- Let us consider given string example is C O M P U T E R D A T A B AS E, Pattern is D A T A, in this method it maps with first string character and pattern, if not moves to the next like that map with all the character. In the below diagram shown the process. In worst cases the time complexity of Search process can be O (m*n), where n is the size of string and n is the size of the pattern.

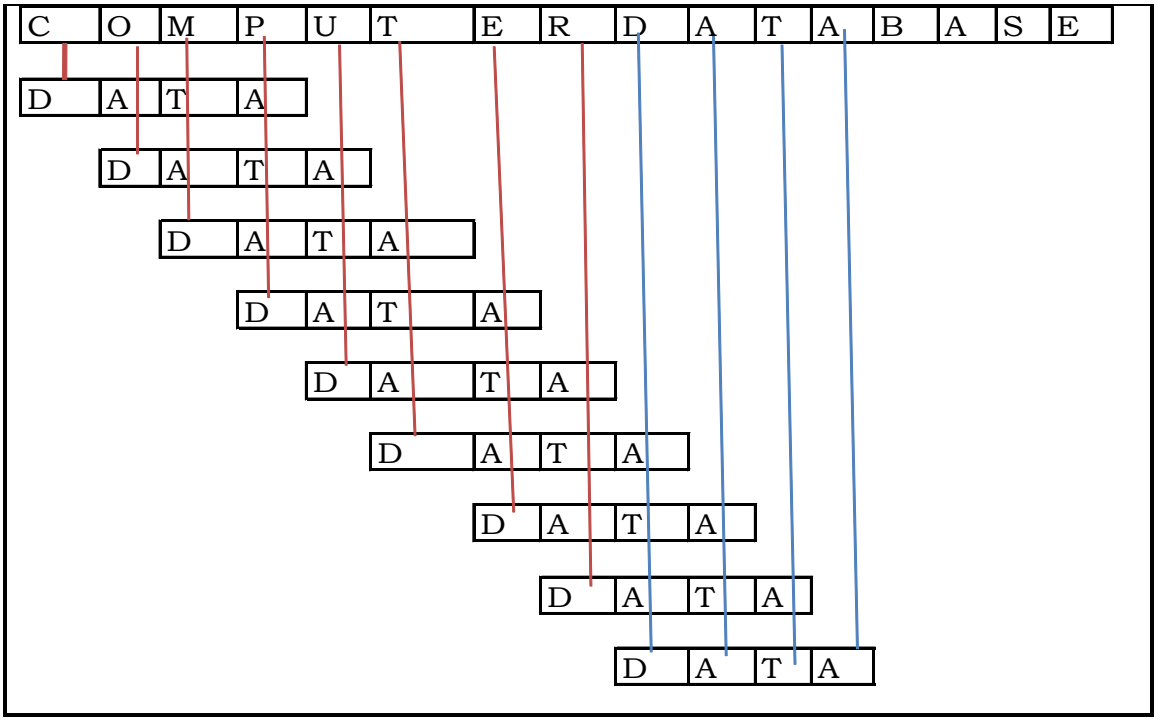

Fig. 1. Naive algorithm string search process

eISSN 1303-5150

- **KMP** algorithms on String matching. A string-matching algorithm is optimized for one or more events in one string. The algorithm returns the first character location in the text of the desired substring. Knuth-Morris-Pratt String Matching Algorithm (KMP), It contrasts the example and the content from left to right.

- If there should arise an occurrence of a mismatch or entire match it utilizes the idea outskirt of the string.

It diminishes the hour of looking contrasted with the Brute Force calculation. KMP calculation utilizes automata to discover all the occurrences of a pattern. Fig 2 presents the example, let us consider the string example D A D A T E D A T A B A S E, and pattern is D A T A. this method pattern will map with the given string, till find the exact patterns.

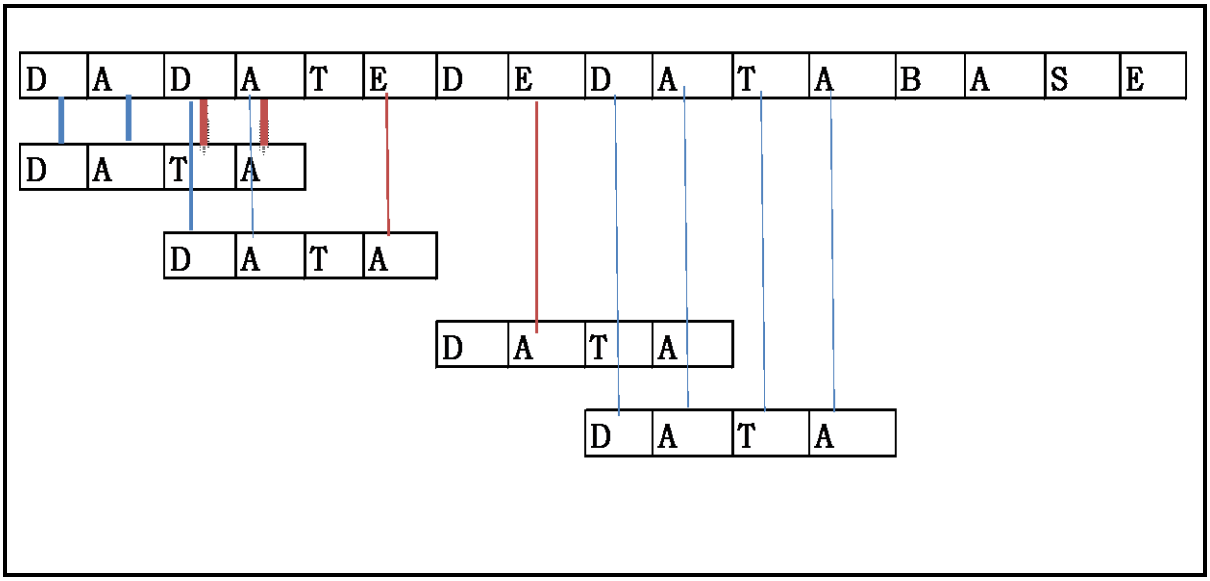

Figure. 2. KMP algorithm searching process.

- **Boyer-Moore** matching an algorithm for very similar sequences. " is a highly efficient search algorithm for a string. The algorithm pre-processes the search goal string (key), but not the search string. The algorithm typically gets faster as the search key becomes longer. The reliability of this is since ```it looks for a connection between the search string and text with each unsuccessful attempt.

- Boyer Moore String technique utilizes good-suffix operation and bad-char operation to check the new comparing position movingrightward.

- C O M P U T E R D A T A B A S E and pattern is D A T A. This technique maps complete pattern with given text, like that map only the entire files, shown in figure.3. BA technique does the similarity checking from right to left.n text char comparisons in the worst case when searching for non-periodic pattern.

eISSN 1303-5150

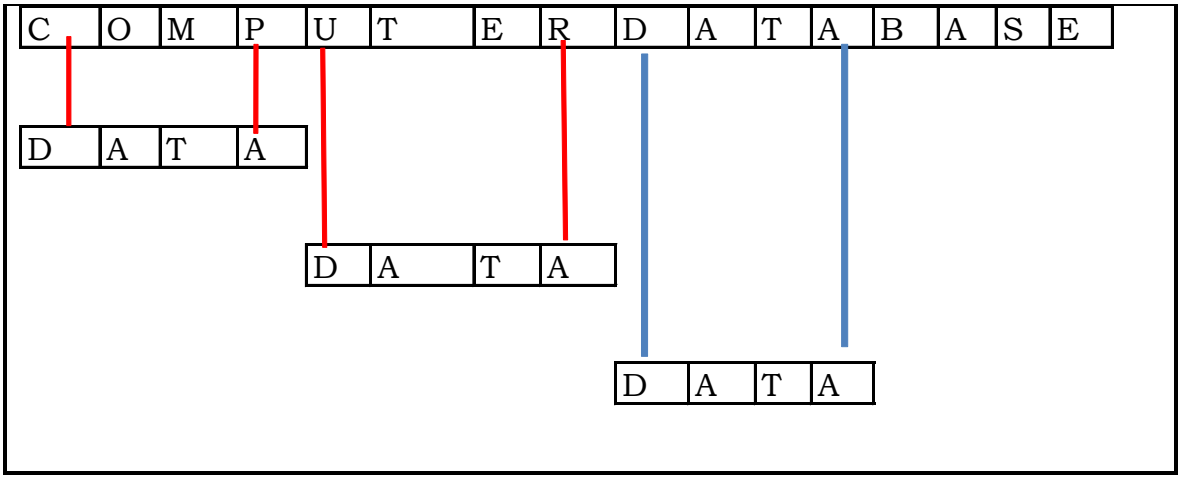

Figure. 3.BM Algorithm search process

For applying large data base records, it will take more time to response. the search algorithm for "Word" W occurrences of a key "Text string" T uses a remark that, in cases of mismatch, the word it contains sufficient content to complete where the next match the start by avoiding the re-examination of earlier similar chars.

Also find that the benefit of BM algorithms is totally lost when the pattern is very small. The result is like that of the algorithm but KMP be still a little stronger.
KMP can be a decent prime when the distance of the design be actually petite and the naïve alternative and in situations where Boyce-Moore is not adjusted it is certainly the best choice. In these tests, the result of Rabin-Karp is very good, much better than KMP.

Hashing Methodology

- The **Rabin-Karp rolling hash algorithm** is excellent at finding a pattern in a large file or stream, but there is an even more interesting use case: creating content based chunks of a file to detect the changed blocks without doing a full byte-by-byte comparison.)

•0000 – Window start
0001
…
1111
0000 – New window start

Start a new window when the bits of the hash are all zeroes! If we are aiming for a ~8k window size, we'd need to use the lowest 13 bits of the hash to decide when to start a new window. This method gives very reasonable window sizes (tested on a ~300mb binary installer file):

Min window: 1
Max window: 81287
Average window: 8132
Median window: 5658

Eg: •: Let us consider alphabet order instead of ASCII VALUES,
for an easy way of calculation)

eISSN 1303-5150

Text = A B E D A B C
Pattern: A B C
Prime no = 3
Step 1: H(ABC) = 1 * 30 + 2 * 31 + 3 * 32 = 34,
Hash value (ABC)= 34.

Step 2: Now compare the Hash value with the remaining text
Step 3: A B E
1 + 2 * 3 1 + 5 * 32 , 1 +6+45= 52
52 does not match with 34 , next shift
Step 4 : B E D (A B E D A B C)
Rule 1: subtract with
old hash (52) – old char (i.e. A B E A)
52-1 =51
Rule 2: x =x /prime (51/3=17)
Now add the value new char (A B E D A B C)
17+ 4 * 32 = 53

Here 53 does not match with 34
•Shift(Rolling by next Text)

E D A , (A B E D A B C)
(5 4 1)
Last hash value 53
53 – 2(value of B) = 51
51/3 = 17
Then followed by: 17 + 1 + 32 , = 26
26 also does not match with 34
•Again shift next text:

D A B (A B E D A B C)
(4 1 2)
26- 5 = 21
21/3 = 7
7 + 2 * 32 = 25 , This one also not match with 34.
•Shift(Rolling by next Text)
A B C (A B E D A B C)
25 - 4 = 21
21/3 = 7
Add new char (i.e ;C)
7 + 3 * 32 = 34
34 matching with pattern Hash, so keeps as a reference of the index.
TEXT = (A B E D A B C)
INDEX= 0 1 2 3 4 5 6
HASH PATTERN = ABC
MACTHING PATTERN IS AT LOACTION= 4.

eISSN 1303-5150

After presenting experimental results between the existing methods and proposed techniques. Now, describe the analysis comparative and complexity of the approaches. Table1. showsthe classification among the algorithms by specifying their advantages and disadvantages.

Algorithm	**Advantages**	**Disadvantages**
• Naive string search algorithm (Linear Search)	• It checks for all the characters of the main string to the pattern. • Easy and not inherently pre-processing method. The cumulative running time is therefore the same as the preceding time.	• Often it can be too late. If the text length is n and the pattern length m • Very inefficient method. Because this method takes only one position movement in each time.
• Knuth–Morris–Pratt algorithm (Heuristic-based)	• The KMP algorithm have of two parts: search division, where instance complication be O(N), search part, which consists of the search for the true shifts in the text and which is accomplished by comparing the pattern and the changes of the text.	• . If the pattern length is very short, KMP may be a good alternative. • KMP works based on a failure function not rolling hash.
• Boyer–Moore string Search algorithm (Heuristic – based)	• The procedure pre-processes the given string being examined for (the pattern), but not the string being examined in (the text). • Uses good suffix shift and bad character shift.	• The main characteristics of this procedure are to match on the end of the pattern rather than the starting, and to avoid along the text in jumps of numerous letters rather than examining all single letters available in the text. • it gives better results in less time when the alphabet is reasonably sized, and the pattern is reasonably lengthy.
• Rabin Karp Rolling Hash (Hashing based)	• Compare the Text and the patterns from their hash functions. Rabin Karp is easier to implement than KMP it works based on a rolling hash	• RKRH you need to verify the match, which means you might suffer quadratic blow up, due to that decreases performance in parallel computing and centralized.

Table 1. Merits and Demerits of String Searching Algorithms

eISSN 1303-5150

Conclusion

This hashing-based algorithm works in two phases, i.e. pre-processing phase (time complexity Θ(m)), matchingphase (time complexity average Θ (n+m),worst Θ((n-m+1) m)).The rules for the Rabin-Karp rolling hash are utilized for the deduplication of variable block length. String matching has affected the field of Information technology and will play an important role infuture technology. The importance of memory andA well-functioning algorithm will last a long time majoring in a cloud environment. Precisely the string-Matching algorithms present different problems and states can fix them. Rabin-Karp algorithm is utilized to identifyplagiarism and it requiresextra space for matching. Rabin-Karp Hashing obtains very good results, its results are a lot better than KMPand the naive solution and it is the best choice in the situations where Boyce-Mooreis not adapted.The time performance of exact string pattern matching can be greatly improved if an efficient algorithm is used.

References

[1]. Ricardo A. Baeza-Yates. ” algorithms for string searching: A survey”. ACM SIGIR Forum, 23, pages 34–58, 1989.

[2]. Robert S. Boyer and J. Strother Moore. ”a fast string searching algorithm”. Communications of the
ACM, Volume 20, Number 10, pages 762–772, october 1977.

[3]. H. Wang, Identity-Based Distributed Provable Data Possession in Multicloud Storage, IEEE,
8(2), 328-340 (2015).

[4].Dr N. Subhash Chandra Md. Rafeeq, Dr. C. Sunil Kumar, Erasure-based improved replication control mechanism and fault tolerance for road traffic data, International Conference on Electrical, Electronics, Computers, Communication, Mechanical and Computing, 2018.

[5]. J. M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono and N. Marnau, Security and Privacy-Enhancing Multicloud Architectures, 10(4), 212-224 (2013).

[6]. Data Deduplication in Cloud Computing Systems, International Workshop on Cloud Computing and
Information Security (CCIS) (2013).

[7]. Cloud Computing Security: From Single to Multi-Clouds using Digital Signature, Int. J. Engg. Technol., Manage. Appl. Sci. www.ijetmas.com, 2(6), (2014).

[8]. Md. Rafeeq, Dr. C. Sunil kumar,A Novel Shorten Erasure Based Reed Solomon Fault Tolerance Code
for Road Traffic Data Fault Tolerance DNSC International journals of soft computing- 13 (Issue 1),
Pages: 25-30.

eISSN 1303-5150