ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. IV, Issue V, May 2018

# SCHEDULING FOR WORKFLOWS WITH SECURITY-SENSITIVE INTERMEDIATE DATA BY SELECTIVE TASKS DUPLICATION IN CLOUDS

[1] Pavithra.R, [2] Ramathilagam.A

[1] M.E. Student, Department of Computer Science and Engineering
P.S.R Engineering college, Anna University, Chennai.

[2] Assistant Professor, Department of Computer Science and Engineering, P.S.R Engineering College, Anna University, Chennai.

**ABSTRACT-**

Data deduplication is an emerging technology that introduces reduction of storage utilization and an efficient way of handling data replication in the backup environment. In cloud data storage, the deduplication technology plays a major role in the virtual machine framework, data sharing network, and structured and unstructured data handling by social media and, also, disaster recovery. In the deduplication technology, data are broken down into multiple pieces called "chunks" and every chunk is identified with a unique hash identifer. These identiers are used to compare the chunks with previously stored chunks and verified for duplication. Since the chunking algorithm is the firrst step involved in getting efficient data deduplication ratio and throughput, it is very important in the deduplication scenario. This paper proposes a high throughput hashless chunking method called Rapid Asymmetric Maximum (RAM). Instead of using hashes, RAM uses bytes value to declare the cut points. The algorithm utilizes a fix-sized window and a variable-sized window to find a maximum-valued byte which is the cut point. The maximum-valued byte is included in the chunk and located at the boundary of the chunk. This configuration allows RAM to do fewer comparisons while retaining the CDC property. We compared RAM with existing hash-based and hash-less duplication systems. In this

paper, we discuss different chunking models and algorithms with a comparison of their performances.

## 1.INTRODUCTION

The Internet is one of the important technology advancements of recent times. Knowledge and media sharing are easier than before because the Internet provides data access everywhere and anywhere, and interconnects devices and machines around the world. However, the bandwidth requirement and the amount of data to be stored increase as the number of internet users increases. This has led to challenges researchers to develop high bandwidth networks and techniques to reduce the amount of data to be transferred.

A technique to reduce the amount of data transfer is to use compression. Data compression algorithms work by eliminating redundant data within a file or multiple files. In particular, a data compression technique called differential compression eliminates the amount of data to be transferred by sending only the differences between files. Differential compression works by comparing two files and only sends the differences. For example, when a user wants to update a file, the server compares and finds the difference between the old version and the new version. The server sends the difference, and the client is required to process the differences and the old file to get the new file. In this scenario, the differential compression technique is called local differential compression.

Remote differential compression (RDC) refers to another situation where the client has a file F1 and the server has a similar file F2. In this situation, the server and the client work together to discover the differences between the two files, and only the differences are sent. RSYNC by Tridgell et al. , which is an

RDC protocol, does remote differential compression by splitting a file into multiple fixed length chunks. Checksums or hash values are calculated from the chunks and sent to the server. The server does the same thing to the file that it believes to be similar and compares the hashes. Only chunks with different hashes are sent to the client, thus reducing the amount of data to be transferred.

The fix-sized chunk used in RSYNC has a problem. Bjørner et al. stated it cannot identify two identical files, F1 and F2, where F2 is F1 with a byte inserted at the beginning of the file. When F1 and F2 are processed using the RSYNC protocol described in the previous paragraph, no checksums of F1 matches the checksums of F2. This is called the byte shifting problem. Content defined chunking (CDC) algorithms fixed this problem by chunking the file based on the internal features of a file. Similar to RDC, CDC algorithms are also used in data deduplication to split a file into chunks for detecting duplicate data.

In the cloud storage ecosystem, differential compression is known as data deduplication. Data deduplication is important because it saves storage spaces by eliminating redundant data within files and between files. Data deduplication involves three main components: chunking, hashing, and comparing hashes to detect redundancy and it works by chunking the files into chunks and comparing the chunks by using a mathematical hash function. In a deduplication system, chunking defines the characteristic of the system because chunking algorithms used in the system can make the system more effective and faster.

One of the most known chunking algorithms is Rabin based chunking algorithm. Rabin based chunking algorithm uses Rabin rolling hash by Michael O. Rabin to find the cut-point. The rolling hash works by using a sliding window, and a hash value is calculated when the sliding window is moving. Rabin rolling hash consumes a considerable amount of time for calculating the hashes. Unlike Rabin based chunking, Local Maximum Chunking (LMC), which is proposed by Bjørner et al. , uses byte values of a file and sliding window to determine the cut point. In this approach, no hash calculation is required but it uses comparisons for each byte processed. Another variant of the local maximum chunking is Asymmetric Extremum (AE) , which uses a fixed length window and a variable-sized window to determine a cut point. The extreme value is located between the two windows. Unlike local maximum, AE does not use sliding window to find the cut points.

Thus, it requires fewer comparisons because the local maximum needs to find the extreme value in the sliding window each time the window slides. Although the number of comparisons is significantly reduced for AE compared to LMC, AE still does a lot of comparisons which might be expensive for performance limited devices. The aim of this work is to reduce the computational overhead of the existing CDC algorithm. We achieved the lower computational overhead by harnessing the statistical properties of the algorithm to minimize the number of comparisons.

This paper proposes a new Rapid Asymmetric Maximum (RAM) chunking algorithm which is based on the AE algorithm. Similar to AE, the algorithm uses two windows: a fixed-length window and a variable-sized window. Unlike AE, RAM uses different positions for the windows. The windows position is started with the fixed-length window and followed by the variable-sized window and the maximum sized byte. The cut point is located at the end of the chunk, which puts the maximum-valued byte at the end of the chunk. This configuration has different statistical properties than AE's configurations, which reduces the number of comparisons. Thanks to the reduced number of comparisons, RAM performs better than AE in terms of deduplication throughput. In addition to the reduced number of comparisons, the algorithm produces chunks with sizes distribution similar to AE's. The contributions of this paper are as follows:

Studies on multiple chunking algorithms;

Proposes RAM, a fast and hash-less CDC algorithm;

Analyzes the properties of RAM and compares it with other CDC algorithms;

Carried out performance evaluation of RAM and compares it with other CDC algorithms to learn the performance of RAM compared to other CDC algorithms when used in a deduplication system. The rest of this paper is organized as follows. Section 2 explains CDC algorithms, its problem, and our motivation. Section 3 discusses the design and analysis of our proposed RAM algorithm. In Section 4, we compare RAM with other CDC algorithms, followed by discussion of the results in Section 5. Lastly, we conclude our paper in Section 6.

## 2.CLOUD COMPUTING

Cloud storage is a model of data storage in which the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company. Cloud storage is a service model in which data is maintained, managed, backed up remotely and made available to users over a network (typically the Internet).

Cloud storage is a cloud computing model in which data is stored on remote servers accessed from the Internet, or "cloud." It is maintained, operated and managed by a cloud storage service provider on storage servers that are built on virtualization techniques.

Cloud Files is an affordable, redundant, scalable, and dynamic storage service offering. The core storage system is designed to provide a safe, secure, automatically resizing, and network accessible way to store data. You can store an unlimited quantity of files ranging in size from a few bytes to extremely large.

The maximum storage allowance on BT Cloud depends on how much storage you get free with your BT Broadband package, and the extra storage you purchase: With limited BT Broadband or limited BT Infinity, you get 5GB of free BT Cloud storage. With Unlimited BT Broadband and Unlimited BT Infinity 1, you get 100GB free.

Cloud Storage is a service where data is remotely maintained, managed, and backed up. The service allows the users to store files online, so that they can access them from any location via the Internet. Now, let's look into some of the advantages and disadvantages of Cloud Storage.

## 3.PROPOSED ALGORITHMS

### 3.1.1 CHUNKING LGORITHM:

Chunking Algorithm In data deduplication, the basic idea is to split a file into blocks and applies hash functions to compute hash values. To check data duplication the client sends the hash key list to the server. The hash key for each chunk is used to determine if that chunk exists in the multiple locations by comparing hash keys. If there are same hash keys on another location, we assume that the chunk is duplicated. Therefore, we can prevent duplicated data blocks to be transferred. Generally, the chunking algorithms are divided into two; fixed length chunking and variable length chunking. The fixed length chunking approach achieves very fast data deduplication result but the performance is not good; because boundary shift problem degrades the deduplication performance. On other hand, variable length chunking achieves high degree of performance while causing high computation overhead and longer processing time.

### 3.1.2 DE-DUPLICATION

Cloud can store and retrieve file. De-duplication has a removing duplicate file. Its will find out duplicate file.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. IV, Issue V, May 2018

Deduplication The Admin is the data owner who performs deduplication by checking if the contents of two files are the same and stores only one of them. Here the data owner upload, download and update the files. Then the deduplication is performed by applying the RAM Algorithm

**RAPID ASYMMETRIC MAXIMUM ALGORITHM (RAM)**

RAM algorithm design with a goal of achieving low computational overhead and byte shift-resistant algorithm, we proposed a boundary version of AE, called RAM. RAM is similar to AE because it also uses two windows: fixed and variable-sized windows. The placement of the windows is, however different from AE. In RAM, the fixed-sized window is located at the beginning of The algorithm works by searching a byte with the maximum value in the fixed-sized window. If the byte next to the fixed sized window has larger value than the one in the fixed-sized window, the byte is used as the maximum-valued byte and the cut-point is found. Otherwise, the algorithm moves to the next byte until it finds the larger byte as illustrated in the pseudo code of RAM . Thus the algorithm's minimum chunk size is 1, where is the size of the fixed-sized Window. RAM reduces the computation time by searching the byte that is equal or larger than the current maximum value, while Reprocess all the bytes smaller or equal than the maximum-valued bytes. Since the probability that the next byte is smaller than the current maximum value is higher than the probability that the next byte is larger than the current maximum value, RAM enters the first condition less frequently than AE. This lowers RAM's overhead.

## 4.CONCLUSION AND FUTUREWORK

we have discussed the importance of content defined chunking for multiple applications and why it is better than fix-sized chunking. We proposed a new chunking algorithm, called Rapid Asymmetric Maximum (RAM) based on asymmetric chunking algorithm. We analyzed and compared RAM with other chunking algorithms. Our results show that RAM offers lower computational overhead compared to other CDC algorithms. The main advantage of RAM is its low computation overhead which allows high chunking throughput. The high chunking throughput comes at the cost of higher chunk variance. The higher chunk variance produced by RAM is negligible compared to the performance gain over other chunking schemes based on local maximum chunking. In some cases, RAM offers 26% to 40% higher byte saved per second compared to the other chunking algorithms.

## REFERENCES

**[1]** Towards a Self-Adaptive Data Management System for Cloud Environments

**Author:** Alexandra Carpen-Amarie

**Year** : 2011

**[2]Title** : Bringing introspection into blobseer: towards a self-adaptive distributed datamanagement system

**Author:**alexandrucostan, jingcai

**Year** : 2011

**[3]Title** :Optimizing storage performance in public cloud platforms

**Author** :Jian-zong WANG, Peter VARMAN

**Year** : 2011

**[4]Title** : Dynamic Load Balancing on Web-server Systems

**Author** : Valeria Cardellini

**Year** : 1999