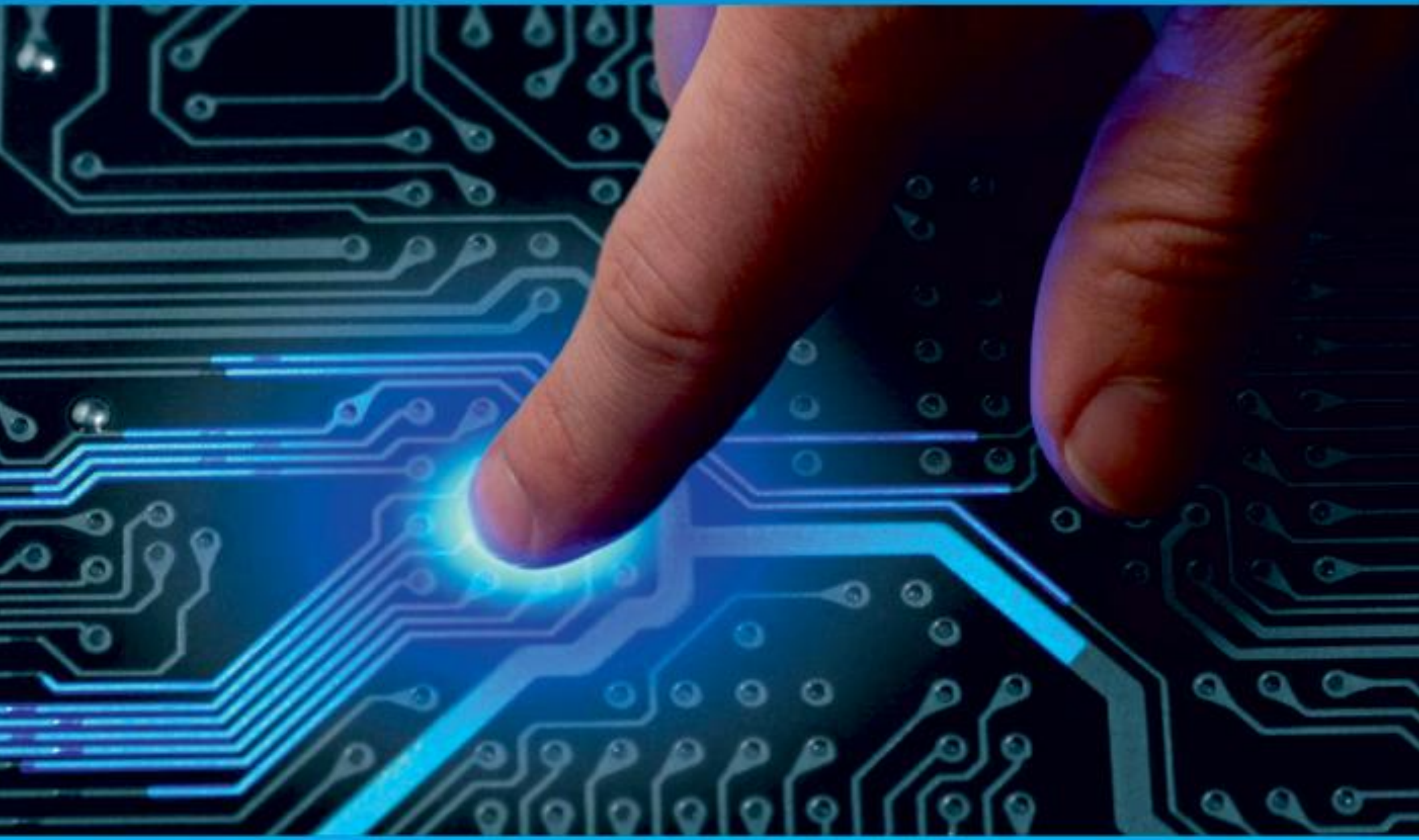
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204248 |

Similarity and Location Aware Scalable Fog Computing System for Storage Systems

A.ABISHA , A.ABITHA , K.DHANALAKSHMI , S.SHIYAMALADEVI , Mrs.S.ELAMATHI

UG Student, Dept. of CSE., Sir Issac Newton College of Engineering and Technology, Nagapattinam, TamilNadu, India

UG Student, Dept. of CSE., Sir Issac Newton College of Engineering and Technology, Nagapattinam, TamilNadu, India

UG Student, Dept. of CSE., Sir Issac Newton College of Engineering and Technology, Nagapattinam, TamilNadu, India

UG Student, Dept. of CSE., Sir Issac Newton College of Engineering and Technology, Nagapattinam, TamilNadu, India

Assistant Professor, Dept. of CSE., Sir Issac Newton College of Engineering and Technology, Nagapattinam, TamilNadu, India

ABSTRACT: Big data is widely considered as potentially the next dominant technology in IT industry. It offers simplified system maintenance and scalable resource management with storage systems. As a fundamental technology of Fog computing, storage has been a hot research topic in recent years. The high overhead of virtualization has been well addressed by hardware advancement in CPU industry, and by software implementation improvement in hypervisors themselves. However, the high demand on storage image storage remains a challenging problem. Existing systems have made efforts to reduce storage image storage consumption by means of deduplication within a storage area network system. Nevertheless, storage area network cannot satisfy the increasing demand of large-scale storage hosting for Fog computing because of its cost limitation. In this project, we propose SILO, a scalable deduplication file system that has been particularly designed for large-scale storage deployment. Its design provides fast storage deployment with similarity and locality based fingerprint index for data transfer and low storage consumption by means of deduplication on storage images. It also provides a comprehensive set of storage features including instant cloning for storage images, on-demand fetching through a network, and caching with local disks by copy-on-read techniques. Experiments show that SILO features perform well and introduce minor performance overhead.

I. INTRODUCTION

Storing large amounts of data efficient, in terms of both time and space, is of paramount concern in the design of backup and restore systems. Users might wish to periodically (e.g., hourly, daily or weekly) backup data which is stored on their computers as a precaution against possible crashes, corruption or accidental deletion of important data. It commonly occurs that most of the data has not changed since the last backup has been performed, and therefore much of the current data can already be found in the backup repository, with only minor changes. If the data, in the repository, that is similar to the current backup data, can be located efficient, then there is no need to store the data again, rather, only the changes need be recorded. This process of storing common data once only is known as data deduplication. Data deduplication is much easier to achieve with disk based storage than with tape backup. The technology bridges the price gap between disk based backup and tape based backup, making disk based backup affordable. Disk based backup has several distinctive advantages over tape backup in terms of reducing backup windows and improving restore reliability and speed.

II. LITERATURE REVIEW

1.Dynamic Placement of Virtual Machines for Managing SLA Violations

Dynamic resource management is an active area of research.The authors of employ prediction techniques and queuing theory results to allocate resources efficiently within a single server serving a web workload.

Advantage:

The goal of the algorithm is to minimize the cost of running the data center.

Disadvantage:

It also has quickly decaying autocorrelation function and no distinct periodic component. This workload is not likely to benefit from dynamic management. It exhibits low variability.

2. Performance and Power Management for Cloud Infrastructures

The first approach aims for an optimal tradeoff between performance and power consumption while ignoring adaption costs. Analogue to the previous solution, A heuristic bin packing algorithm with a gradient based search is used.

Advantage:

- The management frameworks to reduce the energy consumption .
- The user has a single access point from which different services are available.

Disadvantage:

- Although the algorithm isn't optimal, experimental data are needed .
- Better heuristics are possible, this overview shows that energy saving is possible in general.

3. Efficient Resource Provisioning in Compute Clouds via VM Multiplexing

These control methods include providing resource guarantees for VMs in the form of reservations or mins, enforcing resource limits with limits or maximums and manipulating dynamic resource scheduling priorities with shares. With reservations, the service providers or end users can explicitly specify the resources that are reserved for the deployed VMs.

Advantage:

A systematic method to estimate the total amount of capacity for provisioning multiplexed VMs. The estimated aggregate capacity ensures that the SLAs for individual VMs are still preserved. VM combinations lead to high capacity savings if they are multiplexed and provisioned together.

Disadvantage:

While not the scope of our work, there are several avenues of improvement for our VM selection technique.

III. EXISTING SYSTEM

For storage snapshot backup, file level semantics are normally not provided. Snapshot operations take place at the virtual device driver level, which means no fine-grained file system metadata can be used to determine the changed data. Backup systems have been developed to use content fingerprints to identify duplicate content. Offline deduplication is used to remove previously written duplicate blocks during idle time. Several techniques have been proposed to speedup searching of duplicate fingerprints. Existing approaches have focused on such inline duplicate detection in which deduplication of an individual block is on the critical write path.

3.1 ALGORITHM:

Whole File Hashing: In a whole file hashing (WFH) technique, the whole file is directed to a hashing function. The hashing function is always cryptographic hash like MD5 or SHA-1. The cryptographic hash is used to find entire replicate files. This approach is speedy with low computation and low additional metadata overhead.

Sub File Hashing: Sub file hashing (SFH) is appropriately named. Whenever SFH is being used, it means the file is broken into a number of smaller sections before data de-duplication. The number of sections depends on the type of SFH that is being used. The two most common types of SFH are fixed size chunking and variable-length chunking. In a fixed-size chunking approach, a file is divided up into a number of fixed-size pieces called “chunks”.

Delta Encoding: The term delta encoding (DE) is comes from the mathematical use of the delta symbol. In math and science, delta is used to calculate the “change” or “rate of change” in an object.

3.2 DISADVANTAGES:

- There is no scalability in distributed data sharing systems.
- Difficult to implement fault tolerance mechanisms when the number of nodes keeps changing.
- Provide huge volume of garbage data collector.

IV. PROPOSED SYSTEM

In deduplication framework, propose system implement block level deduplication system and named as similarity and locality based deduplication (SILO) framework that is a scalable and short overhead near-exact deduplication system, to defeat the aforementioned shortcomings of existing schemes. The main idea of SiLo is to consider both similarity and locality in the backup stream concurrently. Specifically, expose and utilize more similarity through grouping strongly correlated small files into a division and segmenting large files, and leverage locality in the backup stream by grouping closest segments into blocks to confine similar and duplicate data missed by the probabilistic similarity detection.

4.1 ALGORITHM

4.1.1 Secure Hash Algorithm

- SHA was designed by NIST & NSA and is the US federal standard for use with the DSA signature scheme (nb the algorithm is SHA, the standard is SHS).
- It produces 160-bit hash values.
- SHA is a close relative of MD5, sharing much common design, but each having differences.
- SHA has very recently been subject to modification following NIST identification of some concerns, the exact nature of which is not public

4.1.2 METHODOLOGY:

Step 1: Append Padding Bits….

Message is “padded” with a 1 and as many 0’s as necessary to bring the message length to 64 bits less than an even multiple of 512.

Step 2: Append Length....

64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message.

Step 3: Prepare Processing Functions….

SHA1 requires 80 processing functions defined as:

f(t;B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)
f(t;B,C,D) = B XOR C XOR D (20 <= t <= 39)
f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)
f(t;B,C,D) = B XOR C XOR D (60 <= t <= 79)

Step 4: Prepare Processing Constants....

SHA1 requires 80 processing constant words defined as:

K(t) = 0x5A827999 (0 <= t <= 19)
K(t) = 0x6ED9EBA1 (20 <= t <= 39)
K(t) = 0x8F1BBCDC (40 <= t <= 59)
K(t) = 0xCA62C1D6 (60 <= t <= 79)

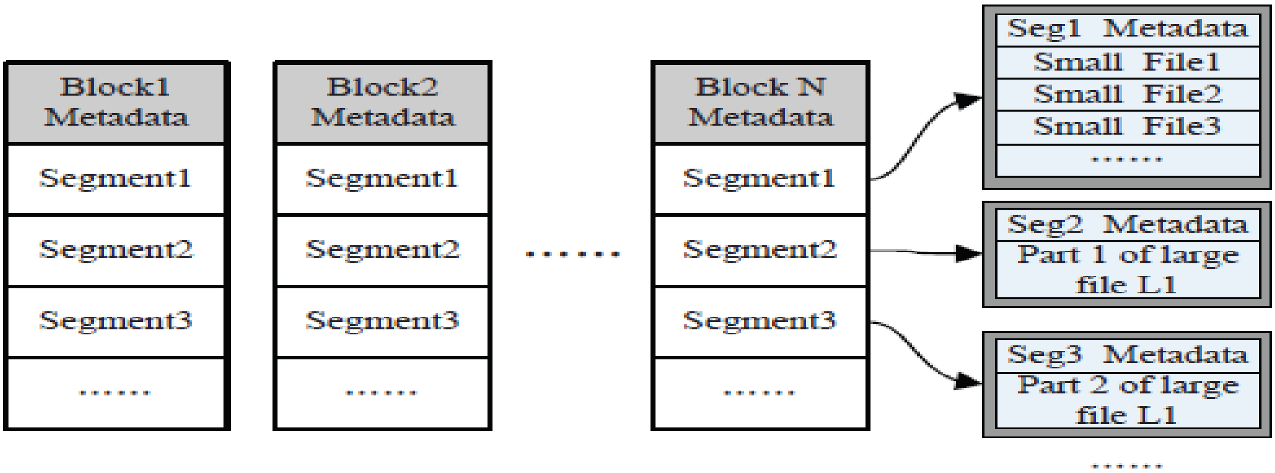


FIGURE 4.1.2.1 : Basic data structure

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204248 |

4.3 ADVANTAGES:

- SiLo is able to remove large amounts of redundant data, dramatically reduce the numbers of accesses to on-disk index.
- Maintain a very high deduplication throughput.

V.FLOWCHART

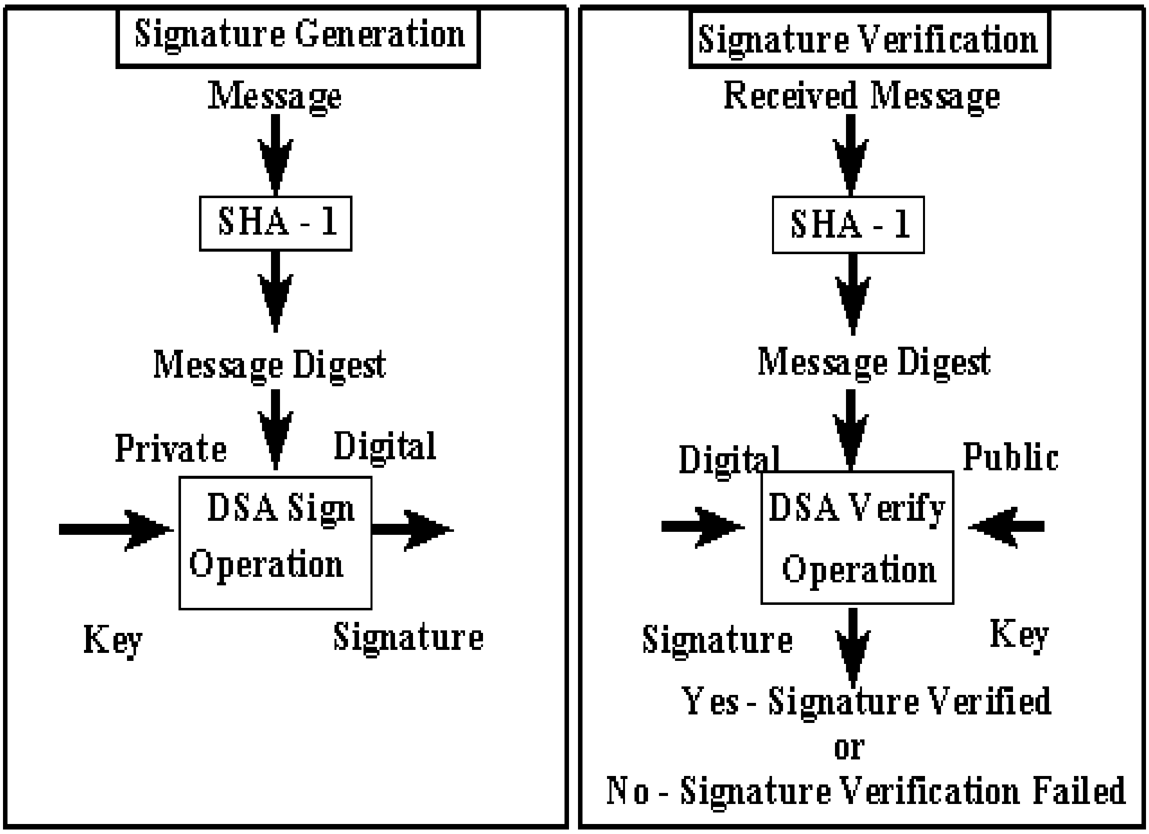
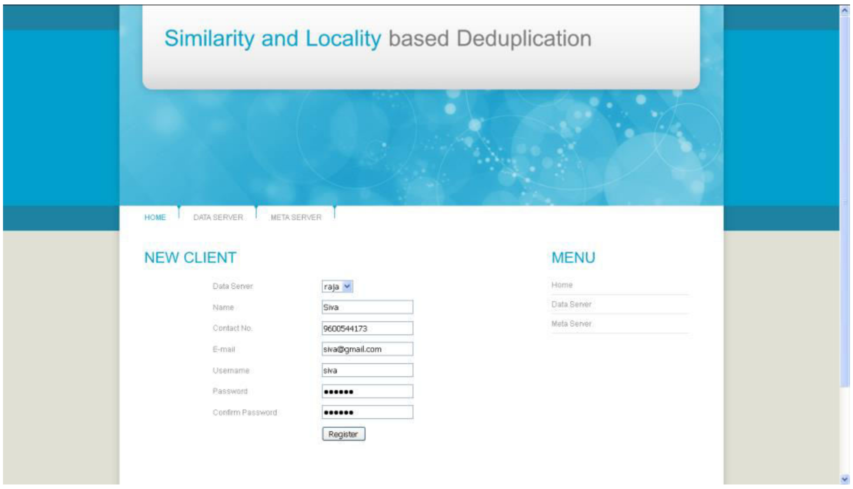


FIGURE 5.1 : Working of proposed system

VI. RESULT


Similarity and Locality based Deduplication

NEW CLIENT

Data Server: raja

Name: Siva

Contact No.: 9600544173

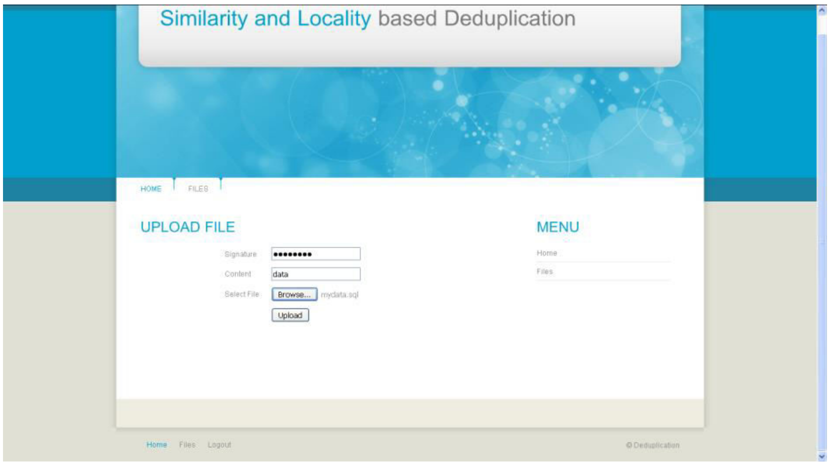
E-mail: siva@gmail.com

Username: siva

Password: ••••••

Confirm Password: ••••••

FIGURE 6.1 : Client Registration



UPLOAD FILE

Signature ●●●●●●●●●

Content data

Select File Browse... mydata.sql

Upload

FIGURE 6.2 : User uploads a file

VII. CONCLUSION

In Fog many data are stored again and again by user. So the user need more spaces store another data. That will reduce the memory space of the Fog for the users. To overcome this problem uses the deduplication concept. Data deduplication is a method for sinking the amount of storage space an organization wants to save its data. In many associations, the storage systems surround duplicate copies of many sections of data. For instance, the similar file might be keep in several dissimilar places by dissimilar users, two or extra files that aren't the same may still include much of the similar data. Deduplication remove these extra copies by saving just one copy of the data and replace the other copies with pointers that lead reverse to the unique copy. So we proposed Block-level deduplication frees up more spaces and exacting category recognized as variable block or variable length deduplication has become very popular. In Fog using the SHT and LHT tables the user easily searches the data and retrieves the searched data from the Fog. And implemented heart beat protocol to recover the data from corrupted Fog server. Experimental metrics are proved that our proposed approach provide improved results in deduplication process.

VIII. FUTURE WORK

In future we can extend our work to handle multimedia data for deduplication storage. The multimedia data includes audio, image and videos. And also implement heart beat protocol recover each data server and increase scalability process of system.

REFERENCES

[1] D. Meyer and W. Bolosky, “A study of practical deduplication,” in Proceedings of the 9th USENIX Conference on File and StorageTechnologies, 2019.
[2] B. Debnath, S. Sengupta, and J. Li, “Chunkstash: speeding up inline storage deduplication using flash memory,” in Proceedings ofthe 2010 USENIX conference on USENIX annual technical conference. USENIX Association, 2015.
[3] W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, and P. Shilane, “Tradeoffs in scalable data routing for deduplication clusters,” in Proceedings of the 9th USENIX conference on File and storagetechnologies. USENIX Association, 2011.
[4] E. Kruus, C. Ungureanu, and C. Dubnicki, “Bimodal content defined chunking for backup streams,” in Proceedings of the 8thUSENIX conference on File and storage technologies. USENIX Association, 2010.
[5] G.Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu, “Characteristics of backup workloads in production systems,” in Proceedings of the Tenth USENIX Conferenceon File and Storage Technologies, 2012.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

☎ 9940 572 462 6381 907 438 ✉ ijircce@gmail.com

www.ijircce.com

Scan to save the contact details