ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 6.047

Volume 5, Issue 8, August 2017

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Data Reduction using A Dedplication Aware Resemblance Detection & Elimination Scheme

K. Lavanya[1]
M.Tech ,CSE Department
JNTUA College of engineering, Ananthapur
Anantapuramu – India

Dr. A. Sureshbabu[2]
Associate Professor,CSE Department
JNTUA College of Engineering Ananthapur
Anantapuramu – India

Abstract: DARE, a low-overhead Deduplication-Aware Resemblance detection and Elimination scheme that effectively exploits existing duplicate-adjacency information for highly efficient resemblance detection in data deduplication based backup/archiving storage systems. The main idea behind DARE is to employ a scheme, Duplicate-Adjacency based Resemblance Detection (DupAdj), by considering any two data chunks to be similar (i.e., candidates for delta compression) if their respective adjacent data chunks are duplicate in a deduplication system, and then further enhance the resemblance detection efficiency by an improved super-feature approach. Our experimental results based on real-world and synthetic backup datasets show that DARE only consumes about 1/4 and 1/2 respectively of the computation and indexing overheads required by the traditional super-feature approaches while detecting 2-10% more redundancy and achieving a higher throughput, by exploitingexisting duplicate-adjacency information for resemblance detection. In the proposed system we are trying to overcome that the data-restore performance suggest that supplementing delta compression to deduplication can effectively enlarge the logical space of the restoration cache.

Keywords: data deduplication, delta compression, storage system, performance evaluation, duplicate adjacency.

I. Introduction

From the IDC study, 80% of corporations survey indicated that they were exploring data deduplication technologies in their storage systems to increase storage efficient. In general, a chunk-level data deduplication scheme splits data blocks of a data stream into multiple data chunks that are each uniquely identified and duplicate-detection by a secure SHA-1 or MD5 hash signature While data deduplication has been widely deployed in storage systems for space savings, the fingerprint-based deduplication approaches have an inherent drawback: they often fail to detect the similar chunks that are largely identical except for a few modified bytes, because their secure hash digest will be totally different even only one byte of a data chunk was changed. It becomes a big challenge when applying data deduplication to storage datasets, which demands an effective and efficient way to eliminate redundancy among frequently modified and thus similar data. Delta compression, a technique to remove similarity among similar data chunks has gained increasing attention in storage systems.

II. Related Work

Data deduplication is becoming increasingly popular in data-intensive storage systems as one of the most efficient data reduction approaches in recent years. Fingerprint-based deduplication techniques eliminate duplicate chunks by checking their secure-fingerprints. The fingerprints of a multi-TBscale storage system will be too large to fit in memory and must be moved to the disk, which causes long latencies of random disk I/Os for fingerprint index-lookup. Most existing solutions to this problem aim to make full use of RAM. DDFS and Sparse Indexing attempt to avoid the disk bottleneck for deduplication indexing by exploiting the inherent locality of the backup streams and preserving this locality in the memory to increase cache hit ratio.

III. Proposed System

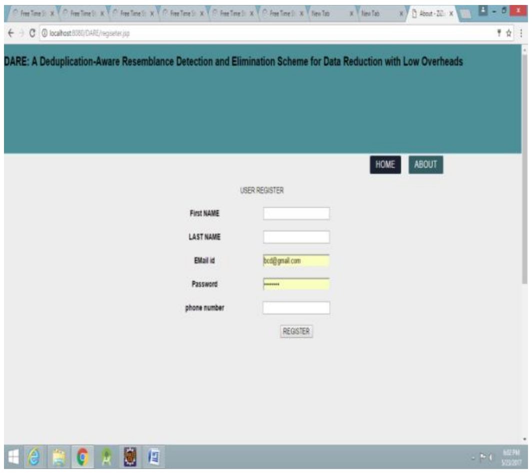
When we are uploading the files in to the system, if that file is already existed in that system then that file will not be uploaded and instead of that the reference will be created so that if number of times one file referenced to many files if by chance that file has deleted then we will loss the reference of the all files so for that reason we are creating the copies of that files in the multiple locations of the system memory. So if one file is deleted from the system memory other locations will maintain the copy of that file. By using Secure Hash Table Technique.

Secure Hash Table Algorithm:

```
1. Start
2. Declare Variable
3. Initialize variable
4. Read 1024 bytes from file in tone iteration
5. Read from file until reach EOF
  5.1 Generate Hash Value from strBuff[BLOCKSIZE]
  5.2 if (FirstBlock)
        Consider node as root element
        Inc BlockCtr
      else
        search the generated Hash in BST
        if (Find Hash == True)
          Compute the Node
          Add the Node to a linked List
          Change the EndLink of SLL
        else
          Add the node in BST
          Inc The BLockCounter
6. Calculate Deduplication Ratio
7. Display the Result for each iteration
8. END
```

IV. Experiments and Works

Here we are restricting the users to upload same file more than one time. Here and when user wants to upload the same file again then that file reference will be stored in the system in multiple locations so that if one file deleted in other locations file data will be there.

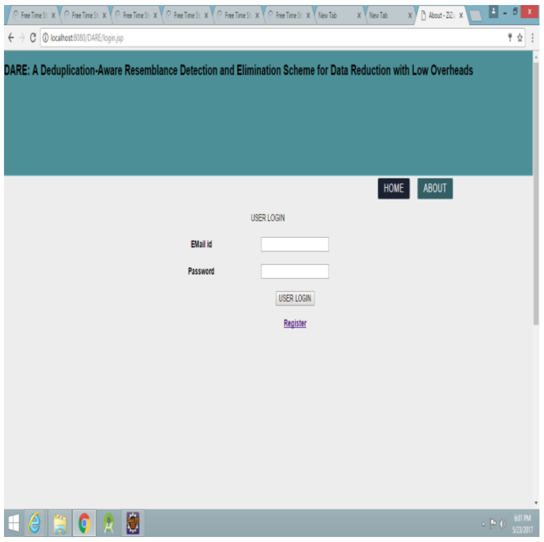

localhost:8080/DARE/register.jsp

DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads

USER REGISTER

First NAME	
LAST NAME	
EMail id	bcd@gmail.com
Password	•••••••
phone number	

When user enters into the system user has to register into the system to access the application the above one is the Registration page where user registers his details to enter into the system.

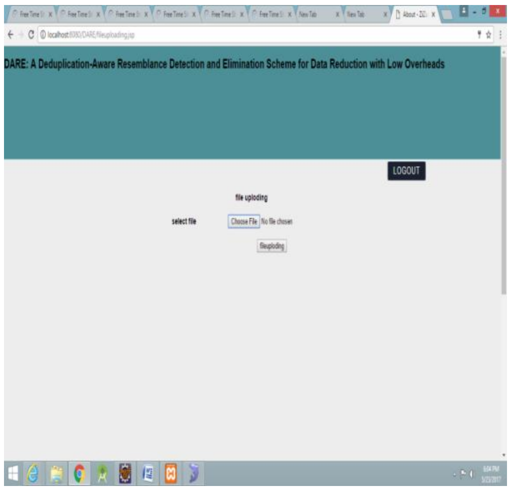

localhost:8080/DARE/login.jsp

DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads

USER LOGIN

EMail id	
Password	

After the successful registration of the user .By using this login page user has to login to his account by entering the valid credentials.

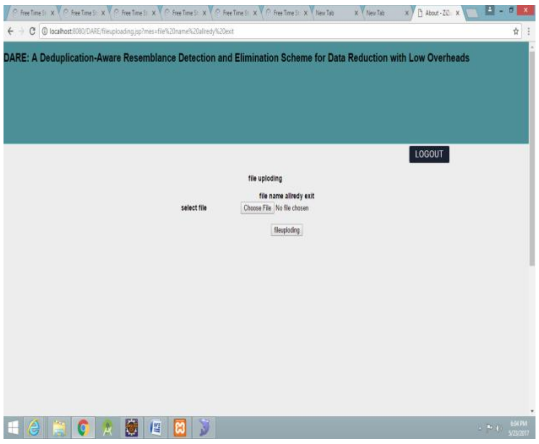

localhost:8080/DARE/fileuploading.jsp

DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads

file uploding

select file Choose File No file chosen

After successful Login into the application user can upload his file or data into the system. By using the above file uploading page.



localhost:8080/DARE/fileuploading.jsp?mes=file%20name%20allredy%20exit

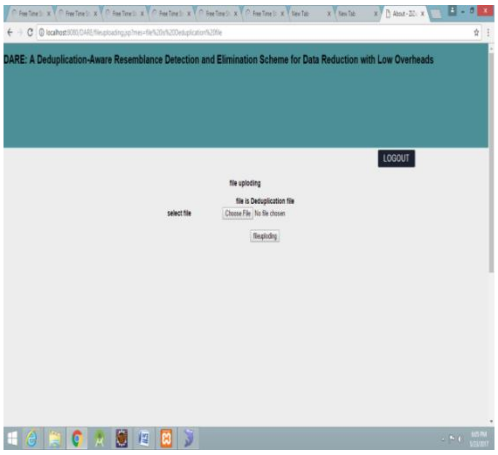
DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads

file uploding

file name allredy exit

select file

If the same data file already existed in the system then it will display the same page as above. By showing the Message "file name already exist".



localhost:8080/DARE/fileuploading.jsp?mes=file%20is%20Deduplication%20file

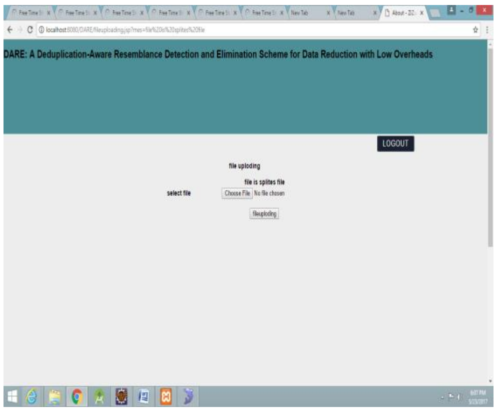
DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads

file uploding

file is Deduplication file

select file

When the user tries to upload the same file then it wills stores the reference of that file. As that the file is already stored in the system.



localhost:8080/DARE/fileuploading.jsp?mes=file%20is%20splites%20file

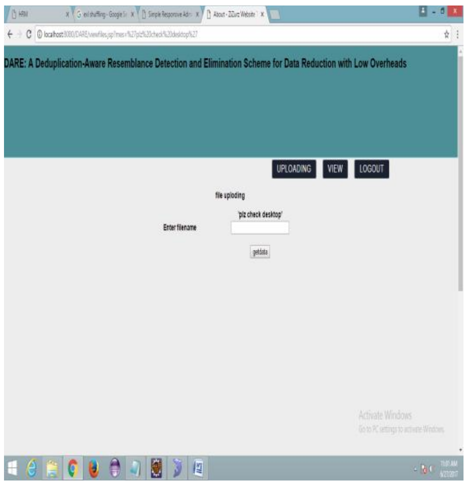
DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads

file uploding

file is splites file

select file

If By chance that file deleted from the system memory then the reference files cannot be retrieved so that we have to maintain the initial copy so that when the reference added it will be stored in the multiple drives of the system.


DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads

file uploding

'plz check desktop'

Enter filename

localhost:8080/DARE/viewfiles.jsp?mes=%27plz%20check%20desktop%27
UPLOADING
VIEW
LOGOUT
getdata

In the above page we are retrieving the file form the desktop .when the file splits the n it will be stored in the multiple locations of the system memory .

V. Conclusion

In this paper, we present DARE a deduplication-aware and detecting the files which hare like and eliminating scheme for data reduction in backup/archiving storage systems.and also DARE enhanced the capable of improving the data restore performance .and here we have improved the data performance of storage systems based on dedeupilication and delta compression for this we are storing the data files in the multiple locations of the system for the performance increasing and preventing the data loss.

References

1. M. A. L. DuBois and E. Sheppard, "Key considerations as deduplication evolves into primary storage," White Paper 223310, Mar 2011.
2. W. J. Bolosky, S. Corbin, D. Goebel, and et al, "Single instance storage in windows 2000," in the 4th USENIX Windows Systems Symposium. Seattle,WA, USA: USENIX Association, August 2000,pp. 13–24.
3. S. Quinlan and S. Dorward, "Venti: a new approach to archival storage," in USENIX Conference on File and Storage Technologies (FAST'02). Monterey, CA, USA: USENIX Association, January 2002, pp. 89–101.
4. B. Zhu, K. Li, and R. H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system." in the 6th USENIX Conference on File and Storage Technologies (FAST'08), vol. 8. San Jose, CA, USA: USENIX Association, February 2008, pp. 1–14.
5. D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," ACM Transactions on Storage (TOS), vol. 7, no. 4, p. 14, 2012.
6. G. Wallace, F. Douglis, H. Qian, and et al, "Characteristics of backup workloads in production systems," in the Tenth USENIX Conference on File and Storage Technologies (FAST'12). San Jose, CA: USENIX Association, February 2012, pp. 33–48.
7. A. El-Shimi, R. Kalach, A. Kumar, and et al, "Primary data deduplication-large scale study and system design," in the 2012 conference on USENIX Annual Technical Conference. Boston, MA, USA: USENIX Association, June 2012, pp. 285–296.
8. L. L. You, K. T. Pollack, and D. D. Long, "Deep store: An archival storage system architecture," in the 21st International Conference on Data Engineering (ICDE'05). Tokyo, Japan: IEEE Computer Society Press, April 2005, pp. 804–815.
9. A. Muthitacharoen, B. Chen, and D. Mazieres, "A low-bandwidth network file system," in the ACM Symposium on Operating Systems Principles (SOSP'01). Banff, Canada: ACM Association, October 2001, pp. 1–14.
10. P. Shilane, M. Huang, G. Wallace, and et al, "WAN optimized replication of backup datasets using stream-informed delta compression," in the Tenth USENIX Conference on File and Storage Technologies (FAST'12). San Jose, CA, USA: USENIX Association, February 2012, pp. 49–64.