Reduction of Spatial Overhead in Decentralized Cloud Storage using IDA

P Suresh Babu, K Madhavi

Abstract: ***Decentralized cloud storage reflects a significant change in large-scale storage performance and economy. Removing central control enables users to store and exchange data without depending on third-party storage [16] providers. Decentralization reduces the possibility of data failures and outages while increasing object storage protection and privacy at the same time. Decentralized cloud storage is where individuals or groups encouraged to enter, store, and keep data available are stored on a decentralized network through several locations. The servers used, rather than a single organization, are hosted by individuals. In this paper, an information dispersal algorithm is applied on decentralized cloud storage to reduce spatial overhead, which provides more efficient performance compared to the existing methodologies.***

Keywords: ***Decentralized Cloud Storage, Information Dispersal Algorithm, Spatial Overhead***

I. INTRODUCTION

In cloud computing, secure data storage and data retrieval are the basic characteristics. Besides, maintaining confidentiality, reliability, and data availability is also an important goal. Cloud storage relies almost entirely on large [3] cloud storage providers (CSP) which act as trusted third parties to send and store data. In decentralized cloud storage (DCS) this can be achieved by dispersing the data into pieces and storing them in various cloud storage platform. A DCS has many advantages, such as cloud storage based on datacenters. Data security is be maintained by using client-side encryption and data integrity is ensured by redudancy .This helps to significantly minimise security breaches and infrastructure failures. Decentralized cloud storage means that each file/resource is broken into multiple chunks/slices and then sent to multiple cloud servers for storage, in simple terms maintaining same file chunks in different cloud storage is called as decentralized cloud storage. In DCS the data is not stored in single location (centralized storage) rather it is stored in different locations termed as nodes.These nodes are the available storage at different locations it may be an organaization, group of computers or an individual PC.If individual nodes does not function due to problems the chunks store at those non-working clouds may not be accessible to users, then decentralized clouds will use resource replication to fix such failover issues. To overcome the failover problem, replication helps maintain multiple copies of the same slice in different nodes. If one node fails, the user can approach another node and download it. To achieve security the data is stored in multiple locations where all nodes will not have all slices. If a node is compromised, the original data still cannot be retrieved. Only the owner of the data can retrieve the original data.

Manuscript received on February 20, 2021.
Revised Manuscript received on February 26, 2021.
Manuscript published on February 28, 2021
* Correspondence Author

P Suresh Babu*, Department of Computer Science, JNTUA College of Engineering, Ananthapuram, Andhra Pradesh, India.
Dr. Madhavi Kasa, Department of Computer Science & Engineering, Jawaharlal Nehru Technological University, Anantapur of Anantapuramu, Andhra Pradesh, India.

II. RELATED WORK

The existing work is concentrated on data owners to store securely, delete resources in DCS services and it is threefold technique first protection is ensured by [6] All-Or-Nothing-Transform (AONT) which controls slicing of the resources. Second allocation of the resource with Min_slice and Min_node techniques, third guaranteed availability and security of the resource. In DCS the resource is split into number of slices and transferred to different locations, to protect data we need to encrypt for that AONT is used .The AONT is encryption algorithm which follows symmetric key cryptography. AONT transfroms the plaintext into ciphertext again to get back original file the whole resource need to be decrypted with key.To ensure more security MIX&SLICE is used which performs different rounds,first round mixes continuous mini-blocks and second round mixes mini-blocks of different computation in first round.The allocation is done by using Min_slice and Min_node in Min_slice it select the minimum number of slices and Min_node select the minimum number of nodes.The resource then transferred to different locations in DCS system. Figure 1 illustrates the resource slicing and allocation of existing work, which focused on the proper slicing and allocation of the resources to various nodes present in the DCS system using AONT.

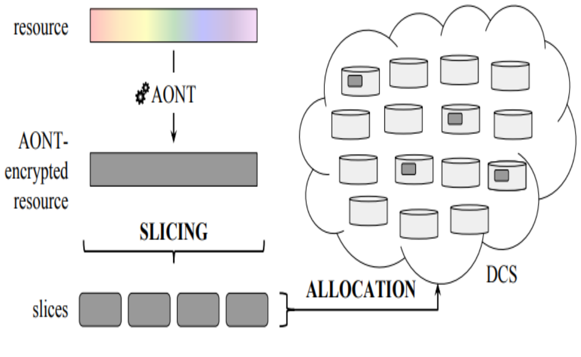

Figure 1: Slicing and allocation of resources

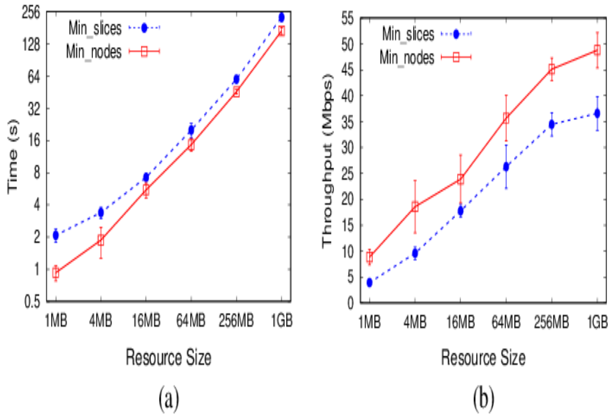

Figure 2: Time completion and throughput graph of existing work

In figure 1 the resource is encrypted and then the encrypted resource is partitioned into slices. The slices are allocated in different locations using Min_slices and Min_node techniques in DCS system. Figure 2 displays the time completion and throughput of the existing work. It took 256 seconds to allocate a file size of 1GB using Min_slice and Min_node allocation techniques.

III. PROPOSED WORK

In proposed work Information dispersal algorithm is used to reduce the spatial overhead. A spatial overhead genarally occurs in allocating space or occupying space for storing a data. In Decentralized cloud storage such overhead arises because data is not stored in centralized cloud servers. The data is stored across multiple locations it may be either an organization, individual etc., to reduce such overhead IDA is used. In the proposed work the file is stored across multiple storage locations with redudant copies of the file fragments. Each file chunk is encrypted using AES algorithm.A different cloud storage platform has been used namely Drivehq and aws. Whenever the file is uploaded it first divided into n number of chunks with which it can construct the original file m.The n chunks are stored at different nodes of the cloud storage platform. The file is stored in such way that it atleast misses a chunk from the actual file so that if one cloud server comprimises the information still cannot access by any others. Original file can be accesed only by the particular user who has uploaded the file.It improves the confidentiallity of the file which inturn results increase in the security.The proposed work guarantees the high availability of the resource since the data is stored in different locations with redundant copies

Information Dispersal Algorithm

The information dispersal algorithm (IDA) is an algorithm used to split a file into chunks/fragments provides a technique for storing data in pieces across various locations, so that redundancy preserves the data in the case of a location outage, unauthorized access does not provide useful information at any particular location. IDA divides the file into n parts so that each part of the file can be stored at different number of nodes. To get the actual file a process is needed that requires m file chunks (m<n). [2] Information dispersal algorithm breaks a file F of length L = |F| into n pieces F_i, $1 \leq i \leq n$, each of length |F i | = L/m, so that every m is sufficient to reconstruct F. Figure 2 displays the functioning of the IDA.

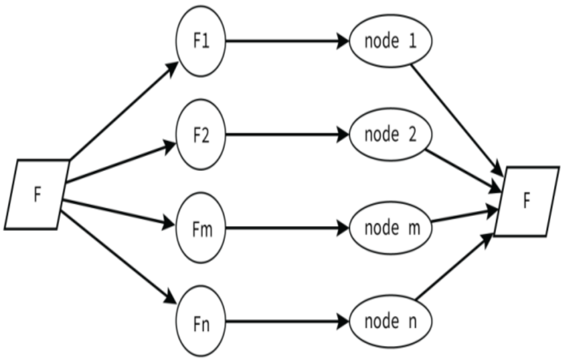

Figure 3: Information dispersal algorithm

Information dispersal algorithm consists mainly two phases, first file split and second file reconstruction. In file split the file f is divided into n fragments and those n fragments assist to form actual file f in file reconstruction. Inbetween the two phase file is encrypted and decrypted using AES algorithm.

File splitting

The file is split down into n chunks in such way it can retrieve the original data. [1] Then the text is converted to an ASCII value from the encoding file. The ASCII value of the file is defined as a matrix F with dimensions of m x k, with m < n and m showing the minimum number of file fragments needed for the file to be reconstructed. As a result some m file chunks, each file chunks are stored at different nodes. Figure 3 displays the File Splitting in IDA.

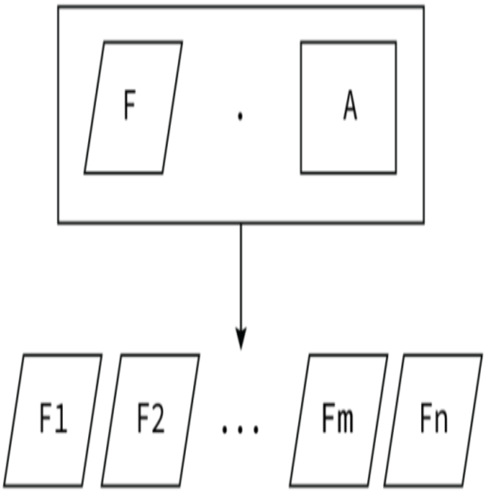

Figure 4: File Splitting

File reconstruction

The file reconstruction process is completed by selecting some file chunks from the nodes where the file chunks are stored. [1] The data for the number of file fragments m was read and then represented in a matrix b with the size of the row m. In the splitting file process, the matrix that is used is reused for its inverse. The inverse of matrix A is then multiplied by matrix b, the file reconstruction is the encoding text that the Base64 decoding process would perform to obtain the original file.Figure 4 displays the file reconstruction of IDA.

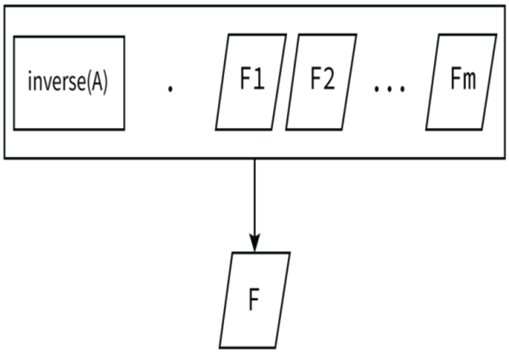

Figure 5: File reconstruction

File encryption

Data stored in the cloud need to be encrypted [1]. So, the files that are stored and secured are accessed secretly, in order not to share with others. AES algorithm is used in file encryption phase, which has sufficient encryption security. File encryption and the splitting process are shown in Figure 6.

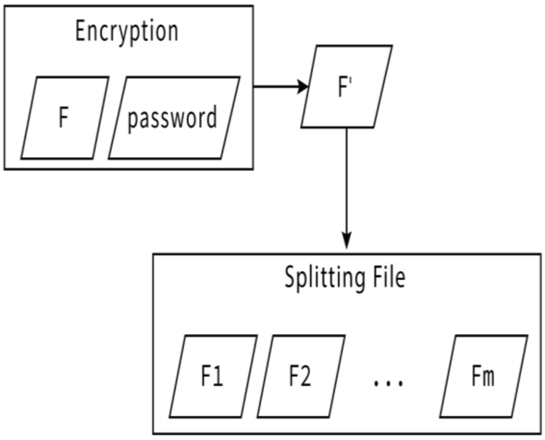

Figure 6: File Encryption

File decryption

Files acquired from the reconstruction phase will be encrypted.The fragments cannot be read instantaneously. [1] To read the data from the file, it is necessary to perform the decryption procedure for that prior step is file reconstruction because the data has been divided into chunks so the file need to be reconstructed and then decryption is done.Figure 7 displays the reconstruction and decryption of the file.

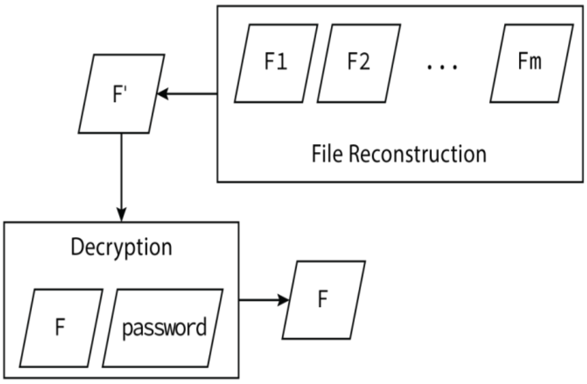

Figure 7: File Decryption

In proposed work these file splitting, file reconstruction, encryption and decryption steps are implemented on resource. The resource is first split in m fragments such that it forms a file n then it undergoes encryption and sent to different location in DCS.Finally decryption is applied on the file fragments to retrieve the original file/resource.

IV. PERFORMANCE ANALYSIS AND RESULTS

The proposed work uses information dispersal algortihm to reduce spatial overhead in decentralized cloud storage our implemented work utilized Drivehq as one of the cloud storage platforms and aws for another storage. Testing has been performed on proposed work while uploading and downloading files of different sizes and compared to existing approach.The results are displayed in figure 8 & 9 in the form a bar graph the X-axis represents the name of the algorithm and Y-axis represents the time in milli seconds and overhead in kb.Figure 8 represents the overall time completion graph where IDA takes 62800 milli seconds to complete the uploading and downloading of a particular file whereas the AONT takes 62200 milli seconds to complete the same task and the file size for both algorithm is 54kb.Figure 9 displays the spatial overhead of the both algorithms where AONT has 38kb and IDA has 11kb here it clear says the spatial overhead is reduced.

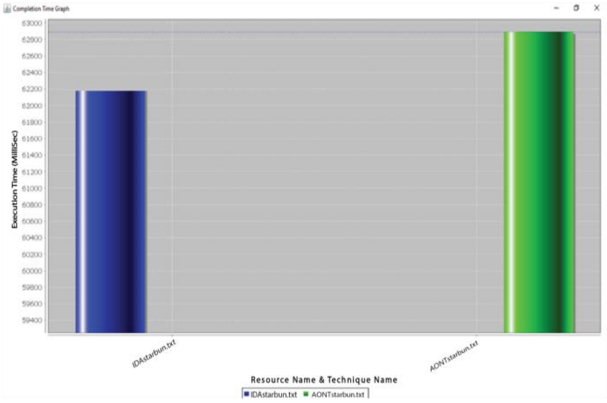

Figure 8: Time completion graph

The time completion graph displays overall time taken by the existing and proposed work to complete the upload and download of a particular file.

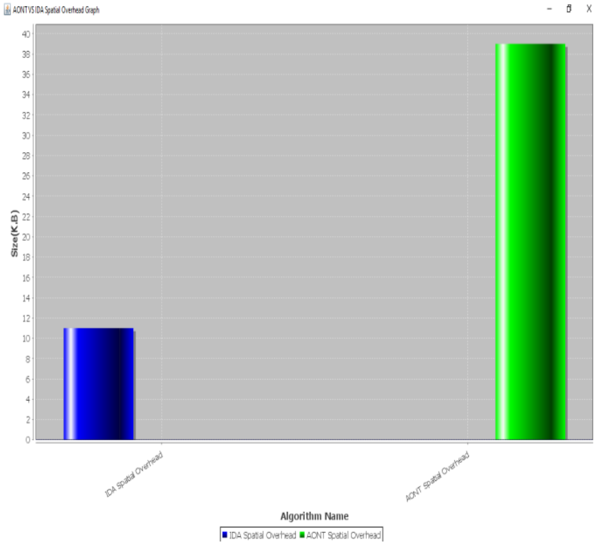

Figure 9: Comparison of Spatial Overhead

Retrieval Number: 100.1/ijeat.C22750210321
DOI:10.35940/ijeat.C2275.0210321
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Figure 9 displays the comparison graph of AONT and IDA of saptial overhead in decentralized cloud storage. The graph clearly shows the [4] IDA gives better performance compared to the existing.

Table 1: Comparison of AONT & IDA

S. N o	Algorithms name	Cloud Storage	File size	Completion time	Over head
1	AONT	AWS, Drivehq	54k b	62900ms	38kb
2	IDA	AWS, Drivehq	54k b	62800ms	11kb

The difference between the AONT and IDA is clearly seen in table 1 where the file size for both algorithms is the same but the IDA algorithm provides better results compared to AONT. The total completion time for IDA is lower and overhead in kb is also lower compared to AONT. The storage used for both algorithms is same i.e, Drivehq as one of the storage and aws as another storage to form a DCS.

V. CONCLUSION

In this paper, the implementation is designed in such a way that it offers a higher level of reliability with an optimized cost factor that takes less upload/download time. It also increases the network's performance and error-free operation for storing and downloading data. Drivehq and aws is used as cloud storage to form DCS network. The spatial overhead is reduced in Decentralized Cloud Storage using Information Dispersal Algorithm. The overhead reduction improves the uploading and downloading speed of files that are stored in decentralized cloud storage. The experimental results shows that the proposed work gives better performance in time completion graph and overhead graph. This increases the latency and reliability of the DCS.

REFERENCES

1. Ardhi Wijayanto, Bambang Harjito. "Reduce Rounding Off Errors in Information Dispersal Algorithm", 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2019.
2. Aisha Abdallah, Mazleena Salleh. "Secret sharing scheme security and performance analysis",2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), 2015.
3. Hoang Giang Do, Wee Keong Ng. "Blockchain-Based System for Secure Data Storage with Private Keyword Search", 2017 IEEE World Congress on Services (SERVICES), 2017.
4. Zhen Yang, Yingying Chen, Yongfeng Huang, Xing Li. "Protecting personal sensitive data security in the cloud with blockchain", Elsevier BV, 2020.
5. Gaganpreet Kaur Sehdev, Anil Kumar. “Performance Evaluation of Power Aware VM Consolidation using Live Migration", International Journal of Computer Network and Information Security, 2015.
6. Enrico Bacis, Sabrina De Capitani di Vimercati, SaraForesti, Stefano Paraboschi, Marco Rosa, Pierangela Samarati, “Securing resources in decentralized cloud storage”.IEEE Transactions on Information Forensics and Security, 2020.
7. E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa,and P. Samarati, “Mix&Slice: Efficient access revocation in the cloud,”in Proc. of ACM CCS, Vienna, Austria, October 2016.
8. C. Patterson, “Distributed content delivery and cloud storage,” https://www.smithandcrown.com/distributed-content-delivery-cloud-storage/,Smith and Crown, Tech. Rep., 2017.
9. H. Hacig¨um¨us¸, B. Iyer, C. Li, and S. Mehrotra, “Executing SQL overencrypted data in the database-service-provider model,” in Proc. of ACMSIGMOD, Madison, Wisconsin, June 2002.
10. A. Shamir, “How to share a secret,” Communications of the ACM,vol. 22, no. 11, pp. 612–613, September/December 1979.
11. M. Conti, E. S. Kumar, C. Lal, and S. Ruj, “A survey on security and privacy issues of bitcoin,” IEEE Communications Surveys & Tutorials,vol. 20, no. 4, pp. 3416–3452, 2018.
12. D. A. Patterson, G. Gibson, and R. H. Katz, “A case for redundant arraysof inexpensive disks (RAID),” ACM SIGMOD Records, vol. 17, no. 3,pp. 109–116, Jun. 1988.
13. K. D. Bowers, A. Juels, and A. Oprea, “HAIL: A high-availability andintegrity layer for cloud storage,” in Proc. of ACM CCS, Chicago, IL,USA, November 2009.
14. M. Albanese, S. Jajodia, R. Jhawar, and V. Piuri, “Dependable andresilient cloud computing,” in Proc. of IEEE SOSE, Oxford, UK, March2016.
15. A. Aldribi, I. Traore, and G. Letourneau, “Cloud slicing a new architecturefor cloud security monitoring,” in Proc. of IEEE PACRIM, Victoria,Canada, August 2015.
16. Zhen Yang,Yingying Chen,Yongfeng Huang,Xing Li.“Protecting personal sensitive data security in the cloud with blockchain”,Elsevier BV,2021.

AUTHORS PROFILE

P Suresh Babu received B.Tech in Computer Science and Engineering from Gates Institute of Technology, Gooty in 2015.Currently pursuing M.tech in Computer Science from JNTUA college of Engineering, Ananthapuramu, Andhra Pradesh, India. His areas of interests include Cloud Computing, Database systems.

Dr. Madhavi Kasa holds a Ph.D. degree in Computer Science & Engineering from the Jawaharlal Nehru Technological University Anantapur of Anantapuramu, Andhra Pradesh, India. Dr. K. Madhavi has more than 16 years of teaching experience in Computer Science & Engineering. She started her career in 2002 as Assistant Professor at SVCET, Chittoor, India. In 2003 she joined Sri Vidyanikethan Engineering College, Rangampet, India, and worked there as an Assistant Professor in Computer Science & Engineering department. Since 2006 she is an Assistant Professor at Jawaharlal Nehru Technological University Anantapur, India, and currently, she is serving as Associate Professor. She published over 70 papers in refereed journals and conferences. Her research interests include Mobile Ad hoc Networking, Big Data & Cloud Computing.