Journal of Theoretical and Applied Information Technology
15th March 2022. Vol.100. No 5
2022 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DATA DISTRIBUTION OPTIMIZATION OVER MULTI CLOUD STORAGE

SAIF SAAD ALNUAIMI[1], ELANKOVAN A SUNDARARAJAN[2], AND ABDUL HADI ABD RAHMAN [3]

[1,2]Center for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia
[3]Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia
Email: [1]p92264@siswa.uk.edu.my, [2]elan@ukm.edu.my, [3]abdulhadi@ukm.edu.my

ABSTRACT

Cloud storage is an essential matter for people's organization and growth. Unfortunately, it is too risky if the data and files are hosted only on a single cloud storage provider. Meanwhile, a possibility exists for insider attack to steal or corrupt the data. Using multi-cloud storage providers and distributing the data over is a possible solution to improve data security in such a context. However, the performance of uploading speed of a cloud server provider plays an influential role. In this study, we used multi-cloud storage with the optimization parameters to speed up the uploading time spent to store data in several cloud storage services. Slicing the data and sending the contrasting amount of data over multi-cloud storage according to the optimization result can provide better security features and upload faster. This work considers the upload time and access latency parameter to implement the optimization model. Our finding shows a 12% enhancement in distribution performance compared to traditional data slicing without optimization, if equally sized slices are sent over multi-cloud storage. In future work, the effectiveness of bandwidth should be included, especially on the optimization parameters.

Keywords: *Distribution Optimization, Cloud Computing, Data Slicing, Multi-Cloud Storage*.

1. INTRODUCTION

Organizations are choosing the multi-cloud environment to avoid vendor monopoly, to have more services to choose from, and to drive more innovation. The main problem with clouds is security issues. Before adopting this technology, we should know that we will be handing over all of our company's sensitive information to a third-party cloud service provider. This could put any company at great risk. Hence, we have to make absolutely sure that we choose the most reliable service provider that will keep your information completely safe. Storing information in the cloud can leave our company vulnerable to external threats and threats. As we well know, nothing on the internet is completely secure and therefore, there is always a hidden latent potential for sensitive data. Cloud services for storage, computing services, and collaboration platforms are becoming more relevant, meaningful, and pervasive [1]. Organizations' and people's survival and growth require storing and retrieving information swiftly by the multi-cloud [2, 3]. Various solutions are available to store files efficiently, such as the growth of parallel file systems that are scalable and are robust cloud computing platforms [2]. Each file is divided into many chunks and transmitted to a different server in the cluster in the most scalable parallel file systems. [4]. For data storage and information security, parallel processing can be used because of data fragmentation and dispersion. The ability to save files on third-party servers is evolving into a more common occurrence. [5]. Data storage in the cloud has a fantastic potential for service providers to develop new services, thereby resulting in new income opportunities [6]. The National Security Agency's PRISM program (US Surveillance Program) in a multi-cloud motivates the business of cloud storage services since cloud storage

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

providers may see data saved on their systems. [7].

Data integrity is a major security concern in cloud computing with several clouds' scenarios. The significance of data integrity is taken into account. Furthermore, the merits and disadvantages of various traditional data integrity solutions are investigated [8]. Data integrity can be improved by using multi-cloud data systems, splitting the original data, and dispersing data slices across multiple clouds. This element will be of tremendous assistance to secure data consumers. Given the sensitivity of the user's data or information preserved on the cloud, security is the most important feature of any cloud computing architecture [9].

Among all the security requirements for cloud computing, access control is one of the basic requirements to avoid unauthorized access to systems and to protect user data. Access control is a security technique that organizes the process of regulating access to resources, issuing permissions to users, and then deciding whether to enter or not. Controlling access to data is one of the biggest challenges in the cloud because of its flexible nature that allows a huge number of users to access data in different geographic regions, time and validity. In addition, the nature of data stored in the cloud is also different because it is huge, which makes access control very complicated. On the other hand, protecting data in the cloud is not only limited to protecting it from the user, but even from the service provider. The amount of trust between the service provider and the user should be partial and not total; This data is the property of the user and not of the service provider. In our proposed work slicing data maintains data validity this is because data is not stored on a single cloud, as each piece of data is stored on a certain cloud according to the efficiency of the cloud.

The primary benefit of storage across many clouds is that it enhances data security and performance. The data from a single cloud storage account is kept on a centralized server that is vulnerable to malevolent insiders [10]. Companies should start thinking of collaborating with multiple cloud providers simultaneously. Cost-cutting, performance, disaster recovery, dependability, security, and privacy are all important considerations. Providers and consumers must work together [11]. Several businesses share the majority of personal information with their customers or suppliers, and data sharing is a top priority for them [12]. Higher productivity levels are achieved through data exchange if the organizations use a multi-cloud approach with dynamic data slicing to store data instead of a single cloud. The cost and time spent on the cloud data would be cheaper than those for the conventional methods of manually delivering and storing data, which sometimes results in out-of-date and redundant papers [12]. Many researchers have discussed the protection of data in cloud storage by encrypting data to provide protection for data in general. Using encryption algorithms is a suitable way to protect data and store it in a single cloud, while this may cost time to perform the encryption and decryption process when the data is stored and retrieved. Therefore, the research idea that we propose is to slice the data and send it to several clouds according to the preference of the selected clouds. In this work, we present two significant contributions:

- Optimal slicing usage for data, and
- Distribution of slices in parallel over multiple clouds.

2. RELATED WORKS

Managing multi-cloud storage is an essential task for most people and researchers. The introduction of cloud computing technologies has recently fueled a surge of interest in this area. However, many works currently focus on data encryption on multi-cloud storage, and others concentrate on the data transfer to multi-cloud storage. Subramanian and John [13] present a framework for data-sharing security operations in a cloud storage with several providers. Their research shows how to use cryptographic index-based data slicing techniques to secure data exchange. This technique aims to keep hostile insiders out by encrypting the entire file with 3DES and encrypting the private key with RSA. Furthermore, by leveraging dynamic file slicing, this approach improves the anonymity of secure data sharing. The framework interface allows customers to specify the number of file sections to slice. Su [14] introduced Triones, a systematic paradigm that uses erasure coding to allow storing data in many clouds in a formal manner.

To begin, it leverages geometric space and non-linear programming abstraction to solve the challenge of data placement optimization, Triones might be able to handle optimization with several goal requirements. Second, Triones can efficiently balance many optimization targets and is also scalable to include newcomers. Extensive testing on several suppliers of cloud storage in the actual world has verified the model's usefulness. Erradi and Mansouri [15] developed two viable online object placement

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithms according to the assumption that data access in the future is unknown. The first online cost optimization technique, which uses no replication, is placed in the hot tier at first. This technique may decide to transfer the item to the cool tier to reduce storage service costs on the basis of the read/write access pattern following an extended tail distribution. When read/write requests are received, a second algorithm that replicates the object is placed in the cool tier and is then replicated in the hot tier.

Celesti et al. [7] offers a unique approach that, allows end-users to rely on many suppliers of cloud storage while also enforcing long-term availability, obfuscation, and encryption. The authors developed a system in which end-users can safely access their data even if the service provider is unavailable temporarily or permanently. Furthermore, only end-users have complete control over all data security, and no critical information is shared with cloud storage providers. Lalith Singh, Jyoti Malhotra, and Sayalee Narkhede [16] aimed to provide an architecture that decreases malicious insiders and file risks in multi-cloud storage services, as well as an algorithm that enhances the security of data storage. This approach creates a secure environment in which the data owner can store and access data from a multi-cloud environment without having to worry about file merging conflicts, as well as prevents insider assaults from acquiring useful information. The proposed architecture lowers the risk of dangerous insider attacks, and the approach protects the provider's resources against malicious files. Encryption is possible for all types of media, including video files using the index-based cryptographic. Nelder, J.A. and R. Mead [17] developed a method that allows users to make cloud storage decisions at low cost while providing high data availability and efficient resource utilization using Apache Hadoop. The data are distributed among numerous servers in the cloud overlap at a single point of failure of a centralized system. For user data security, the shared key encryption technique meets two security principles: secrecy and privacy. This technique serves as a cost-effective solution to vendor lock-in that is suitable for critical applications. Latency or storage limits are not mentioned in this article.

In a multi-cloud environment, slice-based secure data storage means that a file is divided into several slices. The use of fixed-size slices to slice data could simplify Slice-based Secure Data Storage in MultiCloudEnvironment(SSDSMC) implementation. When the slice size is too tiny, the file is sliced into numerous slices, thereby wasting transmission resources and requiring additional cloud provider connections. When data slices become too large, processing the data slices will consume too much memory. As a result, depending on the file size, different slice sizes should be used. When uploading a user file, the data slicing policy is used to determine the slice size [18]. In our proposed method, two parameters, access latency and upload time, decide the slice size after the optimization operation. Providing a cost effective and high-availability data placement for users is a research hotspot in multi-cloud storage. In Wang [19] paper, they first define the multi-objective optimization problem, which is to maximize data availability and to minimize the monetary cost, under the erasure coding mode in multi-cloud storage. Erasure coding is used to reduce storage cost and to improve availability, as compared to data replication [19]. The problem is solvable using a method in line with the evolutionary algorithm for non-dominated sorting (NSGA-II) to provide the Pareto-optimal set of non-dominated solutions. Subsequently, a method according to the entropy approach is proposed to discover the best answer for users who are unable to choose straight from the Pareto-optimal collection [19]. The focus of the article was on placing data in the most cost-effective way possible and high-availability storage across many clouds.

In a recent work in this area [19] to cut total costs and increase data availability, a problem is created in the multi-objective optimization. The authors suggested approaches according to NSGA-II to solve a multi-objective optimization problem effectively and identify a set of non-dominated solutions and erasure coding parameters.

Many researchers are concentrated on developing the security mechanism for big data storage. A recent study by [20] concentrated on developing the encryption algorithm for storing big data in the multi cloud storage. proposed framework contains data uploading, slicing, indexing, encryption, distribution, decryption, retrieval and merging process. The hybrid encryption algorithm was developed to provide the security to the big data before storing it in to the multi cloud. While in our proposed work, data slicing was based on the efficiency parameters of cloud storage for each cloud.The primary purpose of the present study is to optimize data slicing and upload execution time by sending chunks of the data rapidly over a multi-cloud environment.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. MOTIVATION

A multi-cloud environment means that organizations will have to operate in a multi-platform environment that will often include traditional on-premises or components in conjunction with Microsoft Azure, Amazon Web Services (AWS), and other OpenStack cloud technologies. With the generation of big data in various fields such as social media, e-commerce, healthcare, smart transportation, telecom operations, finance and smart cities, the technology of big data analysis and applied research has made big data of unlimited economic and social value. The big data security problems faced by data information in many links are becoming increasingly prominent, which has become the bottleneck restricting the development of big data applications. Here we should think cloud data security for cloud storage security. After all, the development of cloud computing technology has brought about the security threats that big data faces in the process of collection, storage, sharing, and use. Big data leakage of companies' personal privacy information has brought users a huge loss. In fact, the main thing to do on the client is to see the data. The main security issue remains on the server. All data is on the server. The server receives and verifies the data, depending on whether it is important. launch attacks.

This work deals with the data slicing, multi-cloud storage, and optimization process. The idea of the optimization process posits that we want to slice data and store data over multiple clouds in line with the optimization parameters and Euclidian distance. The input parameters for the optimization are the upload time and access latency for each cloud storage provider. The optimization process includes multi-steps of mathematical operations. The result from the optimization process determines the size of the slices. Finally, the slices are distributed to the multi-cloud storage.

4. METHODOLOGY

In this section, we explain the steps in the optimization of slice sizes for distribution to the multi-cloud storage. We used non-linear programming to optimize data slicing under challenging requirements in multi-cloud storage. We also designed the goal function using the Euclidean distance measure and geometric space abstraction, an approach which aids in obtaining the final optimum data slice configurations. The variables in non-linear programming involve access latency and upload time. To obtain accurate test results several tests were performed about 10 times and mean value calculated from data transmission of different sizes 10MB, 20MB, 30MB, 40MB, 50MB, 60MB, 70MB, 80MB, 90MB, and 100MB at various periods.An average transmission time was deduced as shown in Fig. 3. The same tests were performed for access latency calculation for multi-cloud storage.

4.1. Experimental Setup

To assess the success of the optimization, we deployed a prototype interface system. We used four cloud storage providers in our work (Google Drive, MediaFire, PCloud, and MEGA) to perform the experimental setup. The multi-cloud storage was randomly chosen and not based on any impact factor like cost or cloud provider speed. We did not consider paid cloud storage because we used all four cloud storage providers for free; therefore, all four clouds are free plans for cost storage. In our proposed method, we consider two parameters, namely, upload time and access latency, for multi-cloud optimization. A wireless network connection was used with an upload speed of 10MB and a download speed of 10MB. Different regions can be used, and the region where the tests were performed is at Duhok, Iraq. All cloud providers offer 99.9999 percentage availability and 99.9999 percentage durability [21]

4.2. Architecture

In the proposed method, the system architectural framework model (Fig. 1) consists of five steps: upload time and latency measurement from the clouds, optimization process, file selection, and slicing the file on the basis of the optimization results, and distributing all the slices in parallel. The architectural framework starts from the system interface. The optimization process is the initial process for optimizing the access latency time and upload time for each cloud storage provider. The seven optimization steps are presented in Fig. 2.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

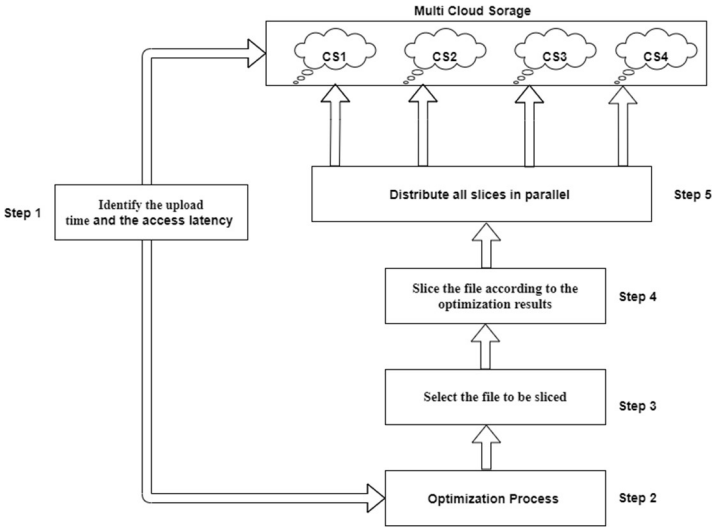

Figure 1: Architectural Framework Model.

Step 1: Identify the upload time for each cloud. The characteristics of underlying cloud storage providers will differ from time to time. Thus, we used an average value for each parameter's upload time and access latency. We used 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100MB file sizes for testing and the NetBalancer utility which can record the upload and download traffic. After we uploaded the file to Google Drive, MediaFire, pCloud and MEGA, the NetBalancer runs and records the traffic. We measured the upload speed with the NetBalancer utility and Windows 10's task manager (performance tab).

The results were similar, but NetBalancer also offers a grid shown in Fig. 3 (in the picture, the vertical lines are 60ms apart). We used the PsPing utility on the Azure website for a latency speed test on multi-cloud storage. This tool provides many features like ping, latency, and a bandwidth measurement utility. The testing was conducted 10 different times per day for an averaged 93 ms for Google Drive cloud, 85 ms for the MediaFire cloud, 205 ms for the pCloud, and 125 ms for the MEGA cloud (Fig. 4).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

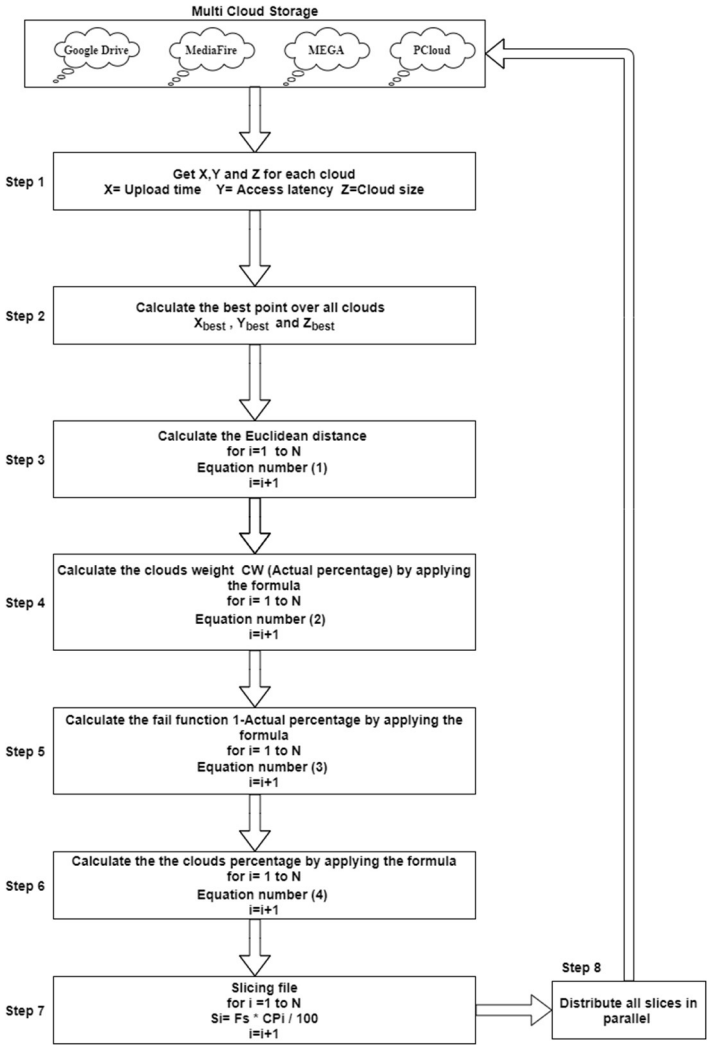

Figure 2: Optimization Steps.

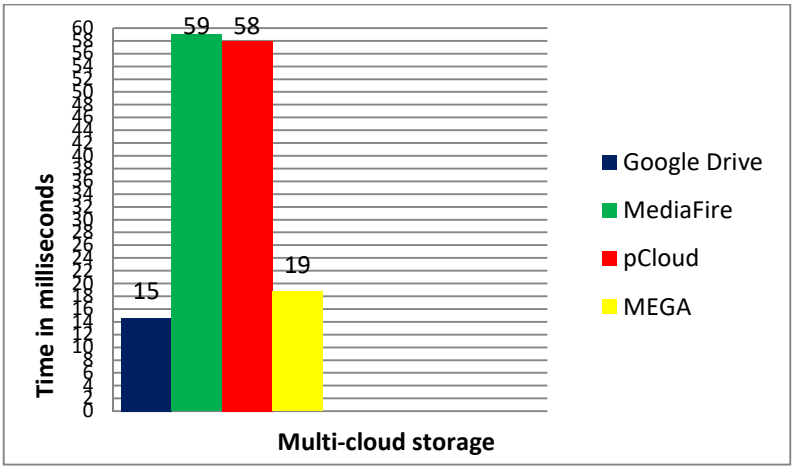

Figure 3: Average Upload Time for Multi-Cloud Storage

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

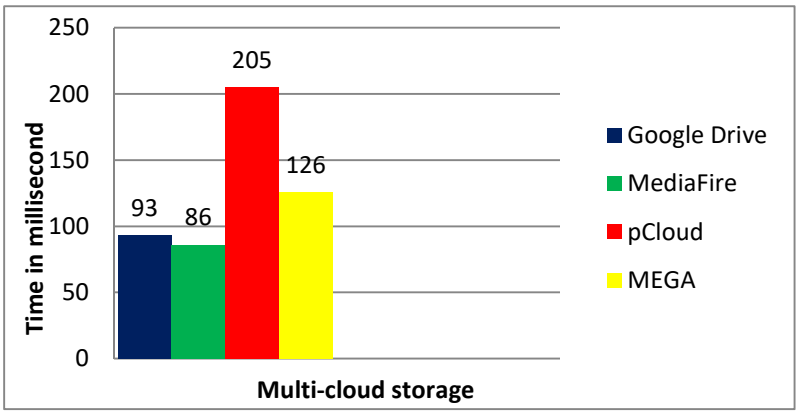

Figure 4: Access Latency for Multi-Cloud Storage

Step 2: Identify the best point among the upload times as the minimum value (time) to send the data to the cloud. The ranking of transfer speed in descending order for the clouds is as follows: Google Drive, MediaFire, MEGA, and PCloud. The recorded average time was 15, 59, 58, and 19 ms, respectively. The best point for upload time among the cloud storages is 15 ms for the Google Drive.

Step 3: Calculate the Euclidean distance of the objective function of the system. In this step, two parameters can be included, namely, the simple or complex needs that reflected in the objective function. However, as discussed, the factors under consideration have different definitions. Triones solves this problem by abstracting multi-dimensional geometric space [17, 22]. To perform the optimization process and apply Euclidean distance measurement, we consider one optimal point and one point for each cloud storage provider (Google Drive, MediaFire, MEGA, and PCloud) Thus, we must determine the distance between each pair of cloud point and the best spot. From the two dimensions geometric space in Fig. 5, we observed four Euclidean distances and one value for the best point located. Therefore, when the best-point value is located at the same cloud storage, the cloud storage represents the best point. Eq. (1) shows the Euclidean distance measure formula.

The Euclidean distance requires the same units, and we used milliseconds for access latency and upload time parameters. In our proposed method with a description of all notations, Table 1 shows a list of the most important mathematical notations defined in Euclidean distance equations.

Table 1. The Definitions Of Mathematical Expressions.

Notations	Descriptions
ED_i	The Euclidean distance between the best point in the multi-dimensional geometric space and the point corresponding to (i).
ED_t	The sum of the total Euclidean distances' points.
X_b	The best upload time over the multi-cloud.
Y_b	The best value of access latency over the multi-cloud
i	The cloud storage number, which is from 1 to 4 in our proposed method.
CW_i	Cloud weight for each corresponding (i).
FF_i	The failure function for each cloud storage.
CP_i	Cloud percentage for each corresponding (i).
FF_t	The total of the failure function.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$ED_i = \sqrt{(x_i - x_{best})^2 + (y_i - y_{best})^2} \quad (1)$$

Step 4: Calculate the weight of the cloud. To identify each cloud, weight percentage is computed using the equation,

$$CW_i = ED_i + ED_t * 100. \quad (2)$$

The cloud weight from this step reflects the actual percentage, and we must ascertain the real percentage. According to the proposed system to achieve the slicing optimization, we should send a large amount of sliced data to lower the percentage cloud storage, which means more efficient cloud storage and the shortest cloud storage distance. Therefore, we should identify the inverse value of the actual cloud percentage because we do not need to send the smallest slice to the shortest cloud storage. First, we compute the failure function and the actual cloud percentage.

Step 5: Calculate the failure function using the equation

$$FF_i = 1 - \frac{ED_i}{ED_t}. \quad (3)$$

Step 6: The cloud percentage is calculated to determine the percentage for each cloud storage for the purpose of data slicing in the next step. The cloud percentage formula is shown in Eq (2).

$$CP_i = 1 - \left(\frac{FF_i}{FF_t}\right) * 100. \quad (4)$$

Step 7: Slice the file on the basis of percentage. The results from Step 6 are the most important part of the proposed optimization model because they are in line with the actual percentage. For each cloud, the slicing process of different amounts of data is related to the cloud.

Step 8: Distribute all slices to the multi-cloud storage where slices are distributed according to the percentage computed in Step 6. The smallest slice size was transferred to the cloud storage location with the highest priority. Meanwhile, according to the optimization results, the greatest slice size is transferred to the lowest cloud storage priority. To examine the upload time capabilities of cloud storage providers Google Drive, MediaFire, pCloud, and MEGA, we conducted several experiments using real file sizes of 10KB, 100KB, and 1MB, which we sliced into four and eight slices, respectively. The upload time for Google Drive was dramatically decreased when the file was divided into eight slices. Thus, when a file is split into too many pieces, slicing the file takes longer but the transfer speed increases. Nonetheless, we did not see a significant reduction in duration when the file was chucked into a small number of chunks. A relatively appropriate fixed size occurs for a specific computing environment when jointly considering the system performance and safety requirements. Figs. 6, 7, and 8 represent the mean upload times considering the four and eight chunks per cloud storage provider. The transferring file size (10KB, 100KB, and 1MB) did not substantially affect the distribution performance for the analysis of the mean upload time. However, when we transferred bigger file sizes (100, 200, and 300MB), we found that as the data size doubles, the upload time does not increase two times..

From this outcome, we conclude that when sending data of minimal size, the percentage of improvement does not appear significantly. The percentage of improvement becomes more apparent when sending data of a larger size.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

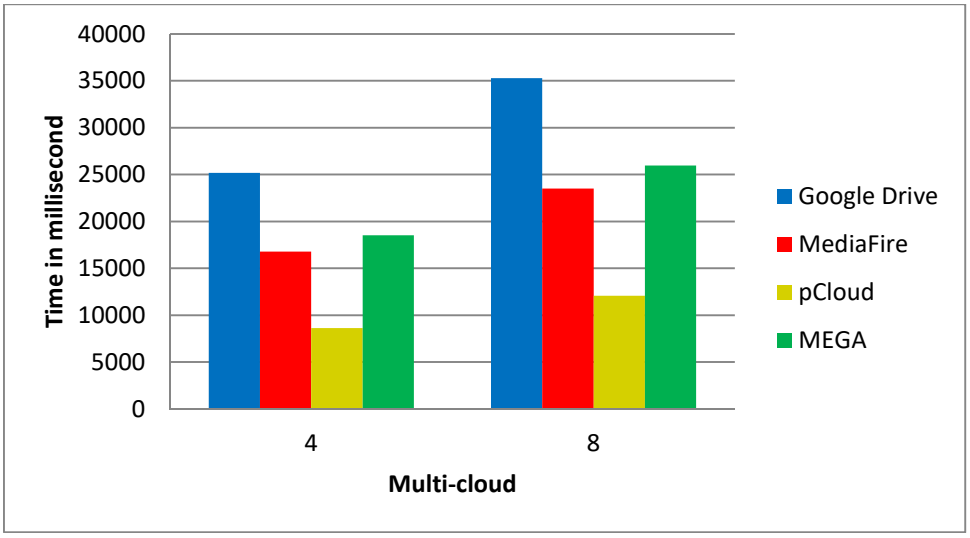

Figure 6: Mean Upload Time for Four and Eight Slices Of 100 Megabytes.

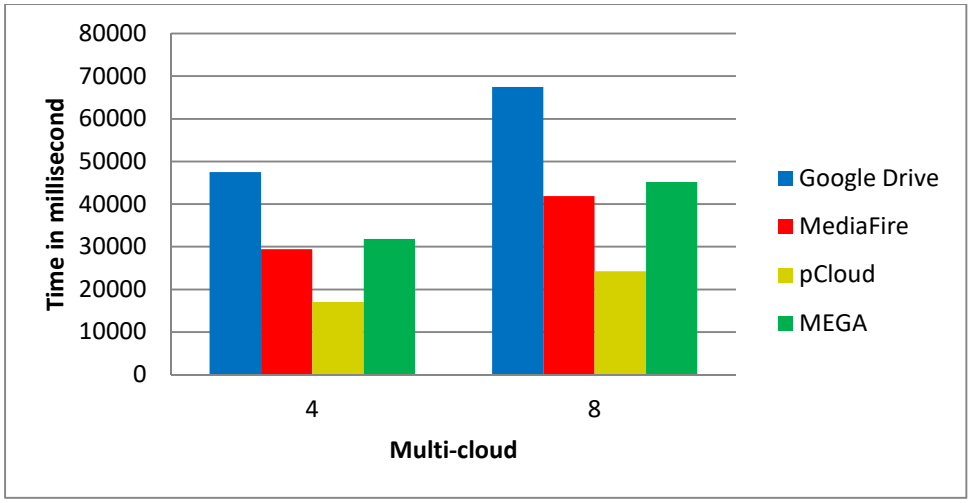

Figure 7: Mean Upload Time for Four and Eight Slices Of 200 Megabytes.

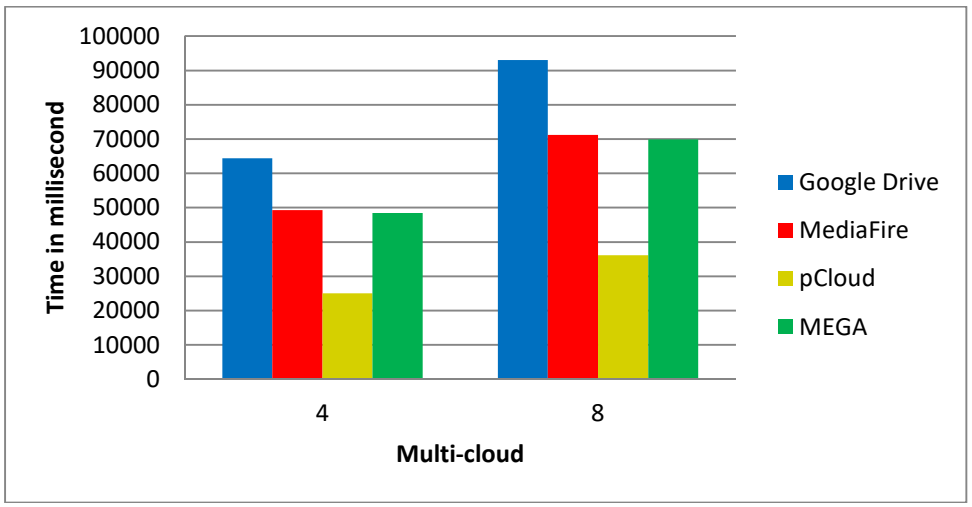

Figure 8: Mean Upload Time for Four and Eight Slices Of 300 Megabytes.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. RESULTS

The optimization results related to Euclidean distance in the proposed method for Google Drive, MediaFire, pCloud, and MEGA showed the distances of 7, 44, 126, and 40 respectively. In the optimization process, a higher slicing rate is obtained if the Euclidean distance is less than zero . Thus, for Google Drive, the Euclidean distance was 7 and the slicing percentage was 32.26%. The largest Euclidean distance determined the lowest slicing percentage rate. We investigated files of various sizes, upload durations, and access latency parameters to better understand the system's behavior in a real-world scenario. More specifically, we examined the file sizes of 100, 200, and 300MB. We concurrently uploaded all slices of each file to multi-cloud storage. Table 2 summarizes the optimization process results of different cloud storage types for Euclidean distance, cloud weight, cloud fail-function, and slicing percentage.

Table 2: Multi-Cloud Optimization Results.

	Google Drive	MediaFire	MEGA	PCloud
Euclidean distance	7	44	126	40
Cloud weight	3.21	20.21	58.11	18.64
Cloud Fail Function	0.97	0.80	0.42	0.82
Slicing percentages	32.26	26.60	13.96	27.18

We attempt to investigate the phenomenon further by identifying the improvement percentage which is represented by calculating the upload execution time for the optimized sliced data and comparing it to that of sliced data without optimization. Our optimization approach improves the upload process, and the various sizes of files used in this case are from 10 to 100MB. The method was run for more than 10 times. Figures 9 and 10 present the upload times for different file sizes.

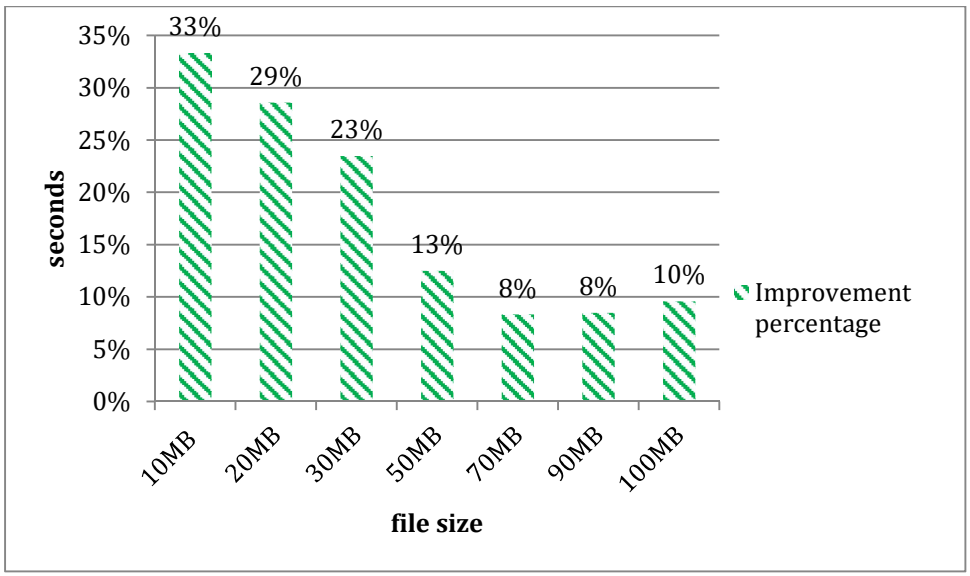

Figure 9: Relative Advantage Of Optimization.

Additionally, slicing can be done statically or dynamically [23]. We define without optimization as slicing the file size to four chunks and each chunk will be upload to cloud which is Google Drive, MediaFire, pCloud, and MEGA. We investigated the sizes of 10, 20, 30, 50, 70, 90, and 100MB once with optimization and once without optimization for a total of four clouds. Thus, the data

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

were sliced by the number of cloud storage providers (four) used in the experimental arrangement. The results from the slicing of files without optimization are presented in Fig. 10.

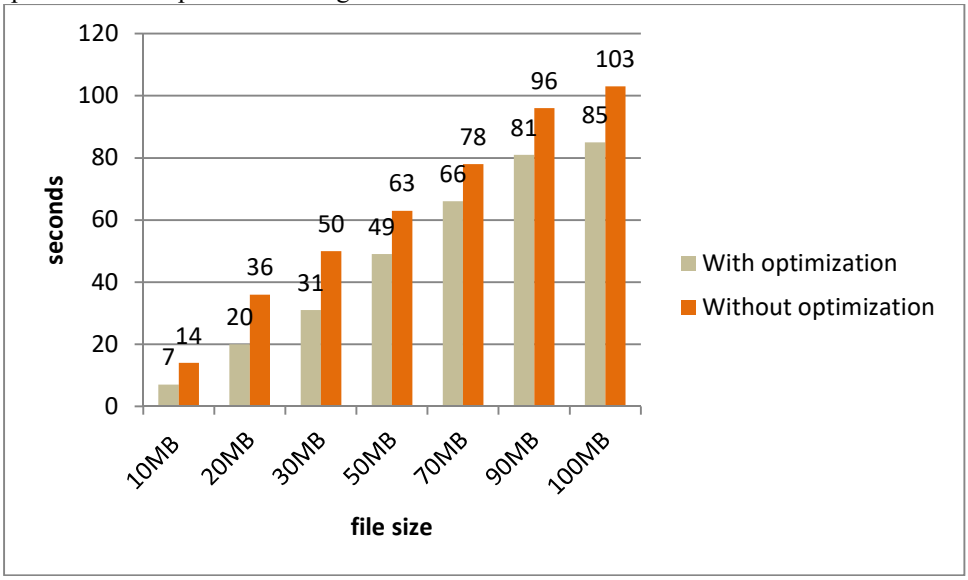

Figure 10: Difference In The Upload Times Of The Optimized And Non-Optimized Slices.

6. CONCLUSION AND FUTURE WORK

The proposed method secure data from cloud storage insider attack. The proposed work used slicing technique based on the clouds parameters which is upload and access latency to accelerates data transfer over multi-cloud storage. The research results provide data transmission efficiency analysis to multiple cloud storage by saving different data slices on multi-cloud storage. The proposed technique allows the users to identify which cloud is best for others and send more data for it. Therefore, this work has the benefit of helping decision-making when the cloud storage capacity is limited and we decided to purchase more storage. In our future work we intend to include the effectiveness of bandwidth on the optimization parameters especially with big data. The larger files size increases the need to use high bandwidth for data transmission over multi-cloud storage. The bandwidth capacity must be in accordance with the proposed optimization process to prevent data throttling during data slices transmission. Therefore, we identify the area of bandwidth optimization and reallocation as an area requiring further investigation

The following services are provided using the proposed methodology:

- Secure and quick distribution of data across several clouds by lowering the upload execution time.
- Ability to support decision making regarding the cloud provider storage when the user is planning to extend cloud storage.
- Dynamic file slicing in various sizes according to the optimization result causes the malicious programs to encounter difficulty recognizing the data pattern.
- Providing load balancing during data transmission by sending the appropriate data segment on the basis of efficiency of cloud storage.

ACKNOWLEDGMENTS

This work was partly supported through the research university grant number TAP-K007924.

REFERENCES

[1]. Li, W. and G. Karame, Secure and efficient cloud storage with retrievability guarantees. 2020, Google Patents.

[2]. Sumathi, M. and S. Sangeetha. Survey on Sensitive Data Handling—Challenges and Solutions in Cloud Storage System. 2019. Singapore: Springer Singapore.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3]. Thakare, V.R.S., K John, Cloud Security Architecture Based on Fully Homomorphic Encryption, in Architecture and Security Issues in Fog Computing Applications. 2020, IGI Global. p. 83-89.
[4]. Ali, M., et al., Distributed File Sharing and Retrieval Model for Cloud Virtual Environment. 2019. **9**(2): p. 4062-4065.
[5] Peacock, A.L., et al., Systems and methods for monitoring globally distributed remote storage devices. 2020, Google Patents.
[6]. Fazio, M., et al., Open Issues in Scheduling Microservices in the Cloud. IEEE Cloud Computing, 2016. **3**(5): p. 81-88.
[7]. Celesti, A., et al., Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems. Journal of Network and Computer Applications, 2016. **59**: p. 208-218.
[8]. Anwarbasha, H., S.S. Kumar, and D. Dhanasekaran, An efficient and secure protocol for checking remote data integrity in multi-cloud environment. Scientific Reports, 2021. **11**(1): p. 1-8.
[9]. Subramanian, K. and J. Leo, Enhanced Security for Data Sharing in Multi Cloud Storage (SDSMC). Int. J. Adv. Comput. Sci. Appl, 2017. **8**: p. 176-185.
[10]. Doshi, N., M. Oza, and N. Gorasia, An Enhanced Scheme for PHR on Cloud Servers Using CP-ABE, in Information and Communication Technology for Competitive Strategies. 2019, Springer. p. 439-446.
[11]. Tchernykh, A., et al., Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. Journal of Computational Science, 2019. **36**: p. 100581.
[12]. Thilakanathan, D., et al., Secure Data Sharing in the Cloud, in Security, Privacy and Trust in Cloud Systems, in Security, Privacy and Trust in Cloud Systems, S. Nepal and M. Pathan, Editors. 2014, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 45-72.
[13]. Subramanian, K. and F.L. John, Dynamic and secure unstructured data sharing in multi-cloud storage using the hybrid crypto-system. International Journal of ADVANCED AND APPLIED SCIENCES, 2018. **5**(1): p. 15-23.
[14]. Su, M., et al., Systematic Data Placement Optimization in Multi-Cloud Storage for Complex Requirements. IEEE Transactions on Computers, 2016. **65**(6): p. 1964-1977.
[15]. Erradi, A.M., Yaser, Online cost optimization algorithms for tiered cloud storage services. Journal of Systems and Software, 2020. **160**: p. 110457.
[16]. Lalitha Singh, Jyoti Malhotra, and S. Narkhede., SECURE DATA STORAGE IN MULTI CLOUD ENVIRONMENT USING APACHE HADOOP. International Journal of Engineering Sciences & Research Technology, 2017. **6**(9): p. 656-666.
[17]. Nelder, J.A. and R. Mead, A Simplex Method for Function Minimization. The Computer Journal, 1965. **7**(4): p. 308-313.
[18]. Celesti, A., et al., Towards Hybrid Multi-Cloud Storage Systems: Understanding How to Perform Data Transfer. Big Data Research, 2019. **16**: p. 1-17.
[19]. Wang, P.Z., Caihui Liu, Wenqiang Chen, Zhen, Z.J.C. Zhang, and Informatics, Optimizing data placement for cost effective and high available multi-cloud storage. 2020. **39**(1-2): p. 51-82.
[20]. Viswanath, G. and P.V. Krishna, Hybrid encryption framework for securing big data storage in multi-cloud environment. Evolutionary Intelligence, 2021. **14**(2): p. 691-698.
[21]. Mete, M.O. and T. Yomralioglu, Implementation of serverless cloud GIS platform for land valuation. International Journal of Digital Earth, 2021: p. 1-15.
[22]. Hinker, P. and C. Hansen. Geometric optimization. in Proceedings Visualization '93. 1993.
[23]. Korel, B. and J. Laski, Dynamic program slicing. Information processing letters, 1988. **29**(3): p. 155-163.