Journal of Theoretical and Applied Information Technology
30th November 2022. Vol.100. No 22

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

BANDWIDTH REALLOCATION APPROACH FOR OPTIMIZED DATA SLICES DISTRIBUTION 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.ukm.edu.my, [2]elan@ukm.edu.my, [3]abdulhadi@ukm.edu.my

ABSTRACT

This paper sheds light on cloud computing and the importance of storing data in multiple clouds. Here it becomes an obvious necessity as when large data is transferred to a single cloud; transmission time becomes excessive. A large data distribution on more than one cloud is better for solving a time-consuming problem. However, some factors affect the transmission process as the speed of data transmission varies from one cloud to another and the size of the Internet data packet provided by the service provider. This paper proposed an optimization method to improve the transmission process and optimally redistribute the data packet to prevent data bottlenecks during transmission. It was evident from the research results that the proposed optimization method for distributing data over several clouds with optimal bandwidth reallocation is 13.58% better when compared with the method of equal distribution of data over several clouds without taking into account the optimal bandwidth reallocation.
Keywords: *Bandwidth Allocation, Transmission Optimization, Multi-Cloud Storage.*

1. INTRODUCTION

Nowadays, with the fast development of network technology and cloud computing, many internet users use cloud-based storage services to safely keep their data and essential files online, about 55% of Internet consumers embraced cloud storage [1]. A cloud always needs a good internet connection and flexible access and it is the responsibility of the CSPs to see that the user uses the existing bandwidth to get connectivity [2]. However, most users have limited bandwidth to upload the files to a multi-cloud server. This is the point of view where from we will be looking at the problem of network congestion and data bottleneck, especially when transferring large file sizes. It takes a lot of time and consumes a large amount of bandwidth. The research on this topic focuses on the following aspects: Uploading performance, bandwidth reallocation, and transmission optimization. Due to the many cloud storage service providers available, one has to wisely choose a cloud storage provider that provides the maximum bandwidth while keeping data secure. Let us illustrate the problem with a simple example: Slicing data and moving data slices around in the same cloud infrastructure is slower than having that information across the Internet. This problem is posed in terms of network bandwidth and latency rates that need to be considered when working with multi-cloud architectures. If we are using a multi-cloud approach, this bottleneck is unavoidable. Network connectivity is the only way for the various clouds to communicate. Fortunately, we can use the approaches to keep connectivity issues to a minimum by avoiding having large amounts of data stored in one cloud and processed in another. While one cloud storage service might cost less, it is not worth the potential performance issues. In this paper, we will analyze the effect of the bandwidth on the distribution of optimal data slicing. This is a growing and competitive area of research which is very important to make a better decision about multi-cloud storage performance.

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

In this research, we will describe the approach in relocating the bandwidth optimally.

2. RELATED WORKS

This section outlines the existing methods available in the literature for bandwidth slicing. Al-Dulaimy et al. [3] propose bwSlicer, a framework for bandwidth slicing in cloud data centers that sheds light on the virtues of effective dynamic bandwidth allocation to improve system performance and energy efficiency. The proposed bandwidth slicing algorithms are emulated in a virtualized networking environment using the Mininet network emulator. The emulation results demonstrated an excellent improvement ratio in execution time and energy consumption, reaching 30%. These results present a call for further research into bandwidth slicing and reallocation as a viable complement to other energy-saving techniques for enhancing energy consumption in cloud data centers. While Chen, K., and N. Yang [4] said that Bandwidth guarantee (BwG) is a highly desired feature in cloud data centers for enabling tenants (i.e., users) to achieve predictable performances. However, such a function currently is not commonly available in clouds since it potentially lowers the utilization efficiency of the network fabric (i.e., denoted network efficiency). The reasons lie in two aspects. First, tenants often present time-varying and/or spatially-varying bandwidth demands. Second, the current cloud network architecture makes it hard to enable tenants to reuse unused bandwidth guarantees efficiently. Here Chen, K., and N. Yang proppproposeshare, a novel bandwidth guarantee scheme in cloud data centers that can effectively improve the network efficiency at both the guaranteed level and the best-effort level in a synergized manner. At the same time, current approaches only adopt one of them.

New methods have recently emerged to address bandwidth allocation by Cao, J. et al. [3], which proposed an auto pre-allocation strategy to solve the bandwidth oversubscription issue in the cloud data center. Their proposal aims to design and implement a bandwidth allocation system embedded in a cloud platform using the technology of software-defined networking (SDN) technology. An exciting methodology in a paper by Kolhar, M., F. Al-Turjman, and A. Alameen [5] incorporates dynamic bandwidth allocation on the diversified characteristics of services such as high definition video, audio cloud computing resource sharing, and virtual reality games is required, especially in point-to-point and multipoint-to-point architectures. They propose an integrated Unified Dynamic Bandwidth Allocation Scheme (UDBAS) framework for wired and wireless network devices that can provide superior execution of bandwidth partitioning concerning the acceptance probability and bandwidth utilization, particularly when short-term variations in bandwidth requirements occur. More specifically, the present scheme is inefficient for scheduling bandwidth management due to bandwidth requirements at each customer premise. Notably, some recent theoretical work by Wu, H. et al. [6] supports a practical and efficient offloading framework with intelligent decision-making capabilities. It is clear from previous research undertaken that to optimize the system utility jointly, and the bandwidth allocation for each mobile device, [6] establish a hybrid offloading model including the collaboration of mobile cloud computing and mobile edge computing.

To enhance the response time for the person (userbase) and processing time of the records center [7], the authors Kalp Kalpesh H Wandra, V.B.G, proposed throttled modified set of rules (TMA) using effective reallocation of the tasks. It had deployment at the *vmloadbalancer* in the data center controller, which progressed primarily based on a throttled set of rules used by the Cloud Analyst simulation toolkit to simulate and evaluate the proposed algorithm with two algorithms: Round-Robin and Throttled. Authors consider the parameters such as the overall response time of the cloud system and the data center's processing time. In the work of Khatavkar, B. And P. Boopathy [8], the idea is to combine the weighted round-robin and max-min algorithms to shape a green load balancing algorithm weighted maximin, and this algorithm has decreased two critical parameters: ready time and response instances. Recent work by Tang, L. et al. [9] used the joint optimization of network selection, and task offloading and proposed an adaptive task offloading strategy. Authors Tang, L. et al. [9] have proposed a TUs pre-allocation algorithm in the cell. Also, their work has often ignored the influence of network access point selection on task execution latency. Since the access network and edge servers are often overloaded, the commonly used task offloading method cannot guarantee the user's QoS.

A more efficient implementation has been proposed relatively recently by [10] authors to implement an optimal method of distributing data over multi-cloud storage by optimizing the data placement parameters, the upload time, and access latency. Compared to typical data slicing without optimization, the authors' findings reveal a 12 percent improvement in distribution performance. Authors in this work based on the optimization parameters, while in our work, we consider the optimization parameters with bandwidth optimization.

In more recent work in this area [11], the authors extend the methods by using the biogeography-based optimization (BBO) technique to enrich virtual network functions (VNFs) to utilize available bandwidth resources in an energy-efficient and cost-effective manner. The cooperative bandwidth sharing approach using BBO reduces delay mean at the time of supervision of the impending requests in a multi-cloud environment. Bandwidth demand is a common resource to achieve computational optimization [12],[13] using a cloud platform. In-network measurement and monitoring, prediction of bandwidth requirement, and end-to-end path also influence exploiting available bandwidth resources [14], [15, 16]. Some current research works [16],[17] provide several bandwidth aggregation solutions to construct dynamic optimization techniques over bandwidth allocation.

The bandwidth consumption and cost of data center networks (DCNs) are growing sharply with the extensions of network size. Thus, keeping the traffic balanced is a key and challenging issue. However, the traditional load balancing algorithms such as Equal-Cost Multi-Path routing (ECMP) are not suitable for high dynamic traffic in cloud DCNs. [18] The authors propose a port-based forwarding load balancing scheduling (PFLBS) approach for Fat-tree-based DCNs with some new features which can overcome the disadvantages of the existing load balancing methods in the following aspects. Firstly, the authors define a port-based source-routing addressing scheme, which decreases the switching complexity and makes the table-lookup operation unnecessary. Secondly, based on this addressing scheme, the authors proposed an effective routing mechanism to obtain multiple available paths for flow scheduling based on Fat-tree. All the path information is saved in servers, and each server only needs to maintain its path information. Thirdly, they propose an efficient algorithm to implement large flow scheduling dynamically in terms of the current link utilization ratio. This method is suitable for cloud DCNs and edge computing, reducing the complexity of the switches and the power consumption of the whole network.

Authors in the paper [19] discussed how to optimize the overall system in terms of data storage and retrieval by testing and validating a Multi-Cloud storage system composed of three major Cloud Storage providers that are Dropbox, Google D,rive and Copy. Their Experiments have proved that the choice of the Cloud storage providers where to store files depends on the data transfer performance according to the chunk size.

In [20], the authors presented a flexible bandwidth allocation strategy where operators can prioritize bandwidth functions to meet service-level objectives. The authors considered two objective functions, which are maximizing network utilization and optimal relative rate allocation.

In general, the past research focus on bandwidth resource utilization through bandwidth reallocation and methods in distributing data over multi-cloud storage optimally. While in our proposed work we are focusing on bandwidth optimization by relocating bandwidth according to optimal data distribution. We found in comparison to the approach of equitable distribution of data over several clouds without taking into consideration the optimal bandwidth reallocation, the suggested optimization method for the proposed paper is 13.58% better.

3. METHODOLOGY

The main objective of this work is to investigate methods for improving bandwidth allocation when multi slices need to be uploaded over multi-cloud storage. On the other hand, we are interested in examining multi-cloud performance, considering the upload speed and access latency while distributing data slices over multi-cloud. This study hypothesized that a significant correlation exists in bandwidth allocation with optimal data slicing and distribution over multi-cloud storage. We tested this hypothesis by performing Euclidean distance measurement for bandwidth reallocation to get the best weights for each cloud bandwidth.

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

The total available bandwidth was divided in such a way as to prevent throttling during optimized data distribution over multiple cloud storage.

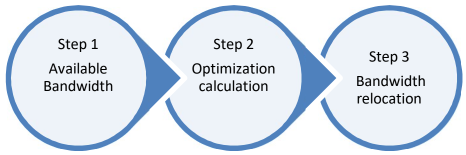

Figure 1. Steps of work.

STEPS OF WORK

Step 1: Determine the available bandwidth

In computer networks, bandwidth refers to the measurement of data that is transferred between two points during a specified per, and the bandwidth is usually expressed in bits, megabits, or gigabits per second between devices connected to the same network; This means that activities such as streaming video content or downloading large files can use up a large amount of bandwidth and slow down connections to other devices on the network. Bandwidth can also be related to some of the data transmission devices themselves, as in the case of I/O devices. For example, a low bandwidth bus can hamper a fast drive, which is the main reason behind the development of buses like AGP for personal computers. Authors [21] distinguish between the bottleneck bandwidth and the available bandwidth of the route.

Bandwidth is measured as the amount of data that can be transferred from one point to another within the network in a specified odperioi. Bandwidth is usually expressed as a bit rate and is measured in bits per second (bits per second). Bandwidth refers to the transmission capacity of the communication, which is an essential factor when determining the quality and speed of a network or Internet connection There are several different methods of measuring bandwidth. Some measurements calculate current data flow, while others measure maximum flow, typical flow, or good flow.

Bandwidth is usually measured using software or firmware, and a network interface and standard bandwidth measurement tools include the Test TCP (TTCP) utility and PRTG Network Monitor. TTCP measures the throughput on an IP network between two hosts, one host is the recipient, the other one is the transmitter, and each side displays the number of bytes sent and the time of each packet to complete the journey in one direction. PRTG provides a graphical interface and charts for measuring bandwidth trends over longer periods, and it can measure traffic between different interfaces. Usually, to measure the bandwidth, the total amount is calculated for transmitted and received traffic over a specified period. The resulting measurements are expressed as a number per second. Another way to measure the bandwidth is to transfer a file or several files of a known size and count the time the transfer takes, and the result is converted into bits per second by dividing the size of the files by the amount of time required to transfer. Most Internet speed tests use this method to calculate the speed of a user's computer connected to the Internet.

The difference between bandwidth, speed, and throughput: There are many ways to think about network data flow. Network speed is defined as circuitd bit rate and is determined by the speed of the physical signal of the medium. Bandwidth is the amount of physical circuit capacity used to transfer data and determine by the network capacity available based on the connection. While a Gigabit Ethernet connection allows 1 gigabit per second, the available bandwidth for a computer connected by a Fast Ethernet card will be 100 Only megabits per second. Transfer rate is the rate of successful transmission. At the same time, bandwidth calculates the amount of data passing through the network interface, regardless of whether the data leads to successful transmission. As such, the transmission rate is always less than the bandwidth.

We have tested the bandwidth in 5 different intervals to find the average upload speed in megabits per second, and we found that the average is 94 Mbps. The tests were performed in succession. Notably, successive tests yield different measurements presented in Figure 3.

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

Figure 2. Example metrics from an Ookla Speedtest, a canonical Internet speed test.[22]

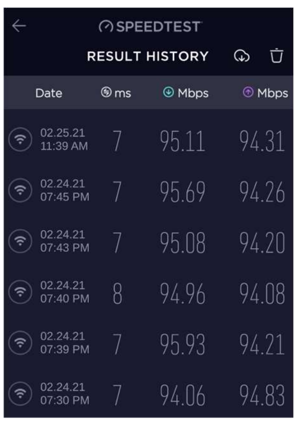

Figure 3. Successive runs of different throughput tests.

Step 2: Calculation of Optimization

We used Euclidean distance measurement to calculate each cloud storage's efficiency for optimization. Euclidean distance measurement in our proposed work is based on two parameters which are upload time and access latency. We got the experimental result for upload time by sending 100MB for multi-cloud storage, and we have calculated the time spent for sending process, and we found the time for clouds (Gdrive Cloud Storage 85s,4shared Cloud Storage 700s, mega Cloud Storage 102s, Pcloud Cloud Storage 610s). While for the access latency, we obtained the experimental result by using the ping command. The ping command was run 20 times and calculated the average we found (Gdrive Cloud Storage 44ms,4shared Cloud Storage 201ms, mega Cloud Storage 121s, Pcloud Cloud Storage 81ms).

The first step of the optimization process is to calculate the best point for upload time and access latency from multi-cloud storage. Moreover, the best point for upload time from multi-cloud storage represents the lost value of 85s, while the best point for access latency from multi-cloud storage represents the lost value of 44ms.

From the two dimensions' geometric space above, we can see 3 Euclidean distances only, and the fourth one is zero because the best point is located at the same place as the gdrive cloud storage. Therefore, when the best point value is located at the same cloud storage, the cloud storage represents the best point.

The Euclidean distance measure formula is

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

We calculate the clouds' weight to identify each cloud's weight percentage. The following formula is applied to find the cloud weight, CW_i.

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

Cloud weight here reflects the actual percentage, and based on the proposed system, we are supposed to slice a large amount of data to be sent for low percentage cloud storage. In this case, we should find out the inverse value of the actual cloud percentage; therefore, first, we calculate the fail function and the actual cloud percentage. The calculation of the failure process used the formula:

$$FF_i = 1 - \left(\frac{ED_i}{ED_t}\right) \qquad (3)$$

The cloud percentage or actual cloud percentage in this step determines the percentage for each cloud storage, and based on these percentages, the slicing can start in the next step. The cloud percentage formula is shown below.

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

Slicing the file based on the percentage of the results is the essential part of the proposed optimization model because based on the actual

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

percentage for each cloud, the slicing process of a different amount of data related to the cloud.

Table 1. Mathematical notations defined and their descriptions.

Notations	Description
EDi	The Euclidean distance from the point corresponding (i) to the best point in the multi-dimension geometric space.
Xb	The best upload time over multi cloud.
Yb	The best access latency over multi cloud
i	Represent the cloud storage number from 1 to 4 in our proposed work.
CWi	Cloud weight for each corresponding i.
FFi	The failure function for each cloud storage.
CPi	Cloud percentage for each corresponding i.

Step 3: Reallocate the bandwidth according to optimization result percentages

These were performed using commercial software from NetLimiter is an ultimate internet traffic control and monitoring tool. NetLimiter allows to decide where our applications are allowed to connect and how fast these connections should be. Here for bandwidth reallocation, we set the optimization percentage value for each cloud storage according to the results presented in Table 2. In this research, six practical tests were performed for the three cases A, B and C shown in Table 5 for two file sizes, 512 MB and 1 GB.

We performed five experiments for each case, result data represented by graph chart in Figures 4-10. While the other four test result are represented by the numbers in Table 3.

Table 2. Cloud's storage percentages.

Cloud storage	Cloud percentage (%)
Gdrive Cloud	33.33
4shared Cloud	16.27
mega Cloud	31.21
Pcloud Cloud	19.18

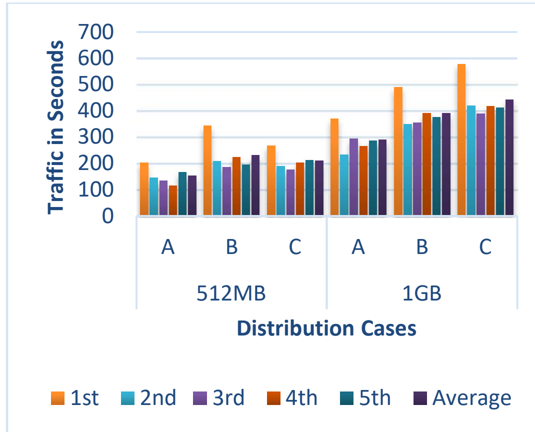

Figure 4. The average time of five distribution cases for 512MB and 1GB.

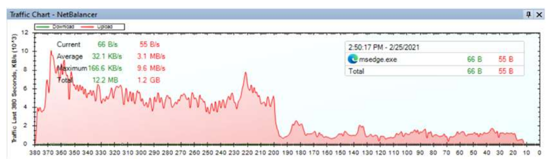

Figure 5. 1GB bandwidth reallocation with different file slice size

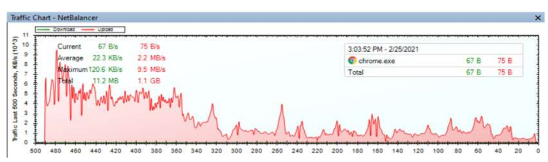

Figure 6. 1GB bandwidth optimization with same file slice size.

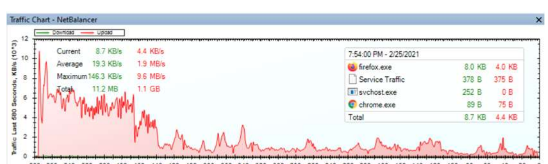

Figure 7. 1GB without bandwidth reallocation and same file slice size

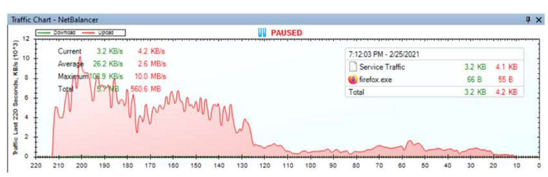

Figure 8. 512MB bandwidth optimization with different file slice size

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

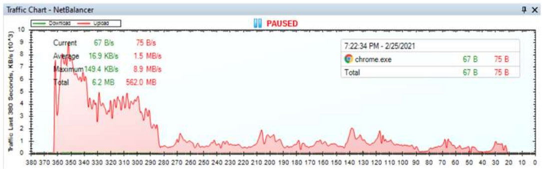

Figure 9. 512MB bandwidth optimization with same file slice size

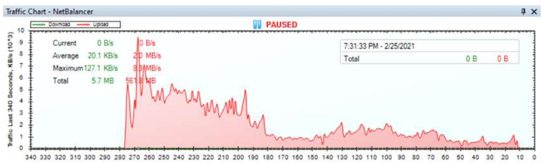

Figure 10. 512MB without bandwidth reallocation and the same file slice size

4. RESULTS

This section outlines the study results where we have sliced the file size 512MB file according to the proposed method for optimizing slicing and bandwidth reallocation with optimization. Calculate the Euclidean distance measurement of the objective function of the system. Two parameters can be put in the objective function reflecting complex or straightforward requirements in this step.

However, as discussed above, the parameters considered have distinct definitions. This issue is through multi-dimensional geometric space abstraction [19], [20]. We consider we have four cloud points and one best point found before; therefore, we need to find the distance between each cloud and the best point.

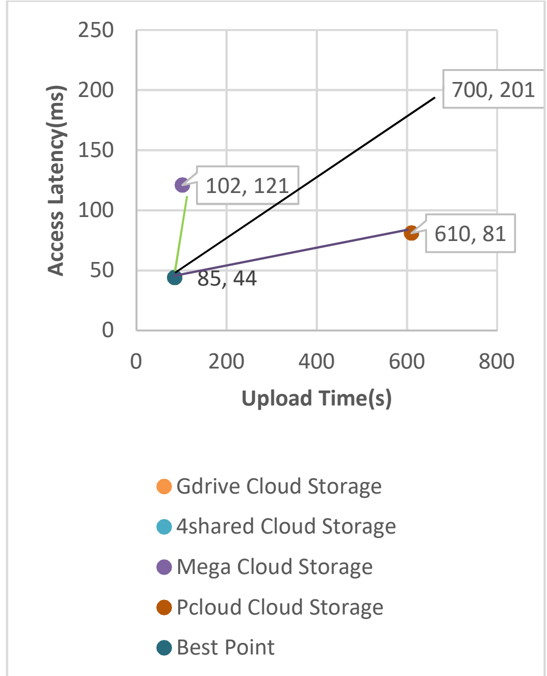

Figure 11. Two-dimension geometric space for optimizing upload time and access latency.

In this paper, we implemented several cases for distributing data to several clouds using the following cases: A) Bandwidth reallocation with different file slice sizes; B) Bandwidth reallocation with same file slice size; C) Without bandwidth reallocation and same file slice size.

The best percentages among all cases will be found by comparing the three cases, A, B, and C. The five transmission attempts shown in table 3 show the transmission time recorded in seconds. Also, we have repeated the same transmission process of the file size one gigabyte five times to get more accurate results. After that, we found the mean value from the five times to grantees more accurate percentages results, and it is presented in Table 4.

Table 3. Five successful transmission time in seconds.

File size	Distribution cases	1st	2nd	3rd	4th	5th
512 MB	A	204	146	134	114	167
	B	344	210	186	224	195
	C	268	190	176	204	213

1GB	**A**	370	234	295	265	288
	B	490	350	355	392	376
	C	577	420	390	418	412

Table 4. Average time with percentages of five successful transmissions.

File size	Distribution cases	Average(s)	Percentage (%)
512MB	**A**	153	25.71
	B	231.8	38.96
	C	210.2	35.33
1GB	**A**	290.4	25.78
	B	392.6	34.85
	C	443.4	39.36

The percentage of performance improvement of the proposed approach is shown in Table 5. We see an improvement of 13.24 percent for a file size of 512MB and 9.07 percent for a file size of 1GB when the file slice size are different compared to homogeneous file slice size. Therefore, distribution case A is better than case B.

Performance improvement of 9.61 percent for file size of 512MB and 13.58 percent for file size of 1GB for the case with bandwidth reallocation with different file slice size compared to the one without bandwidth reallocation with the same file slice is observed. Thus, distribution case A is better than case C, i.e., with bandwidth reallocation with different file slice size is better than without bandwidth reallocation and same file slice size.

Performance improvement of -3.63 percent for file size of 512MB and 4.51 percent for file size of 1GB for the case with bandwidth reallocation with different file slice size compared to the one without bandwidth reallocation with the same file slice is observed. Thus, distribution case C is better than case B when file size of 512MB, i.e., with without bandwidth reallocation with same file slice is better than bandwidth reallocation and same file slice size when file size of 512MB. While distribution case B is better than case C when file size of 1GB, i.e., with bandwidth reallocation with same file slice is better than without bandwidth reallocation and same file slice size when file size of 1GB.

Experimental result in figure 9 shows that in the Pcloud cloud storage and the 4shared cloud storage when the transmission rate percentage reaches more than 90 percent, the data transfer rate increases very clearly. Gdrive and Mega clouds storage complete the process of receiving data very close to each other, this is followed by Pcloud and 4Shared in almost all attempts. In the distribution case of C, it was observed that the Mega cloud, followed by the Gdrive cloud, was consuming bandwidth intensely, which creates a bottleneck in the flow of data to the multi-clouds for both sizes, 512MB and 1GB. Here it is clear that when the data is optimally sliced, the data is received by the clouds in an ideal way, ensuring an even distribution of the data.

Table 5. Differences in performance percentages.

File size	Performance percentage difference between		
	A and B	**A and C**	**B and C**
512 MB	13.24	9.61	-3.63
1 GB	9.07	13.58	4.51

Our experiments indicate that bandwidth reallocation with different file slice sizes were better than i) bandwidth reallocation with same file slice size and ii) without bandwidth reallocation and same file slice size. Our findings show that, in the best case scenario, bandwidth relocation should be based on the upload time and access latency which we use as multi-cloud optimization parameter. Results show that our approach performs with sufficient reliability when used in real time. Also the quality of the results is satisfactory for the majority of cases.

5. CONCLUSIONS

The key contribution of this work is the solution it provides for bandwidth constraint. Various experiments conclude that bandwidth reallocation affects the distribution performance by avoiding the bottleneck when distributing multiple slices over multi-cloud storage. The distribution of optimal data slice with bandwidth reallocation is better than slicing evenly without bandwidth reallocation by 13.58 percent. Also, it is better than the distribution of slicing evenly with bandwidth reallocation by 9.07 percent by sending a file size of one gigabyte. In conclusion, this study demonstrated that distribution optimization parameters could be used as a predictor for bandwidth reallocation. This analysis leads to some useful conclusions, most important of which we proposed a solution to improve the performance of sending data slices over multi-cloud storage by bandwidth load balancing and sending data in the appropriate size for each cloud. The lack of bandwidth virtualization is another notable important limitation of this study. The obtained results justify further development of the method for researchers that deal with large files and use multiple clouds to store data. A new framework based on the proposed method with dynamic bandwidth reallocation based on the transmission efficiency should be developed.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Yang, C., X. Tao, F. Zhao, and Y. Wang, *Secure data transfer and deletion from counting bloom filter in cloud computing.* Chinese Journal of Electronics, 2020. **29**(2): p. 273-280.

[2] Shetty, S., A. Manu, V. Kumar, and C. Antony, *Need of Multi-Cloud Environment and Related Issues: A Survey.* Journal of Xi" an University of Architecture & Technology, 2020. **12**: p. 78-87.

[3] Auday Al-Dulaimy, W.I., Javid Taheri, Maha Shamseddine, *bwSlicer: A bandwidth slicing framework for cloud data centers.* Future Generation Computer Systems, 2020. **112**: p. 767-784.

[4] Kang Chen, N.Y., *BwShare: Efficient bandwidth guarantee in cloud with transparent share adaptation.* Computer Networks, 2020. **170**: p. 107095.

[5] Manjur Kolhar, F.A.-T., Abdalla Alameen, *Unified Dynamic Bandwidth Allocation Scheme for Ethernet Passive Optical Network.* IEEE Access, 2020. **8**: p. 216176-216184.

[6] 6. Huaming Wu, Z.Z., Chang Guan, Katinka Wolter, Minxian Xu, *Collaborate Edge and Cloud Computing with Distributed Deep Learning for Smart City Internet of Things.* IEEE Internet of Things Journal, 2020.

[7] Kalpesh H Wandra, V.B.G., *optimizing performance with load balancing algorithm to reduce latency, throughput and processing time on cloud computing.* 2021.

[8] Bharat Khatavkar, P.B. *Efficient WMaxMin static algorithm for load balancing in cloud computation*. in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. 2017. IEEE.

[9] Lujie Tang, B.T., Li Zhang, Feiyan Guo, Haiwu He, *Joint optimization of network selection and task offloading for vehicular edge computing.* Journal of Cloud Computing, 2021. **10**(1): p. 1-13.

[10] Saif S. Alnuaimi, Elankovan A. Sundararajan, and A.H. Abd, *Data Distribution Optimization Over Multi Cloud Storage.* Journal of Theoretical and Applied Information Technology, 2022. **100**(5).

[11] Bipasha Mahato, D.G.R., Debashis De, *Distributed bandwidth selection approach for cooperative peer to peer multi-cloud platform.* Peer-to-Peer Networking and Applications, 2021. **14**(1): p. 177-201.

[12] Deepsubhra Guha Roy, D.D., Md Mozammil Alam, Samiran Chattopadhyay. *Multi-cloud scenario based QoS enhancing virtual resource brokering*. in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*. 2016. IEEE.

[13] Deepsubhra Guha Roy, M.D., DeDebashis De, *Cohort Assembly: A Load Balancing Grouping Approach for Traditional Wi-Fi Infrastructure Using Edge Cloud*. 2018. p. 93-108.

[14] Hassan Harb, C.A.J., Abdallah Makhoul, *An energy-efficient data prediction and processing approach for the internet of things and sensing based applications.* Peer-to-Peer Networking and Applications, 2019: p. 1-16.

[15] Roy Deepsubhra, G.M., Bipasha Ghosh, Ahona De, Debashis, *Service aware resource management into cloudlets for data offloading towards IoT.* Microsystem Technologies, 2019: p. 1-15.

[16] Siva Kumar Gavvala, C.J., G R Gangadharan, Rajkumar Buyyad, *QoS-aware cloud service composition using eagle strategy.* Future Generation Computer Systems, 2019. **90**: p. 273-290.

[17] Gang Wang, T.L., *Resource allocation for M2M-enabled cellular network using Nash bargaining game theory.* Peer-to-Peer Networking and Applications, 2018. **11**(1): p. 110-123.

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

[18] Zhiyu Liu, A.Z., Mangui Liang, *A port-based forwarding load-balancing scheduling approach for cloud datacenter networks.* Journal of Cloud Computing, 2021. **10**(1): p. 1-14.

[19] Celesti, A., A. Galletta, M. Fazio, and M. Villari, *Towards Hybrid Multi-Cloud Storage Systems: Understanding How to Perform Data Transfer.* 2019. **16**: p. 1-17.

[20] Nagaraj, K., D. Bharadia, H. Mao, S. Chinchali, M. Alizadeh, and S. Katti. *Numfabric: Fast and flexible bandwidth allocation in datacenters*. in *Proceedings of the 2016 ACM SIGCOMM Conference*. 2016.

[21] Lai Kevin, B.M. *Measuring bandwidth*. in *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*. 1999. IEEE.

[22] Nick Feamster, J.L., *Measuring internet speed: current challenges and future recommendations.* Communications of the ACM, 2020. **63**(12): p. 72-80.