IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 12 (December. 2018), ||V (III) || PP 41-49

Cost-Aware Cloud Storage Architecture Using Information Dispersal Algorithm

Makhan Singh[1], Sarbjeet Singh[2]
Department of Computer Science & Engineering. UIET, Panjab University, Chandigarh. India[1]
Department of Computer Science & Engineering. UIET, Panjab University, Chandigarh. India[2]
Corresponding Author: Makhan Singh

Abstract: A Cloud storage service model generally demands security of data with least cost. Different Cloud computing security threats expected to be resolved Cloud environment incorporating Data Access Controllability, Data Confidentiality, and Data Integrity. This paper proposes the cost-aware Cloud storage system using information dispersal algorithm. This proposed system does not use the same as existing methodologies of replication to address data availability and security issues. The proposed system stores the file parts using redundant data encoded with erasure code on multi-cloud environment. This proposed system helps in recreating the user file even when certain threshold numbers of file parts are not available. The proposed approach provides storage of data with-in user's budget and empowers to recover user data if some natural or men made disaster occurs.

Keywords: Cloud computing, Information dispersal, Erasure code, Cloud data storage plan chart.

Date of Submission: 13-12-2018 Date of acceptance: 28-12-2018

I. INTRODUCTION

Cloud computing is a distributed architecture for delivering information technology services that enables ubiquitous, on-demand access to shared resources like computer networks, servers, databases, analytics and many more that can be rapidly provisioned and released with negligible management effort. In Cloud computing different resources are provided to the customers as per Service Level Agreement (SLA). Failures in Cloud are still a common scenario. Therefore, high availability, adequate performance and self-contained fault tolerant backup system are necessary to sustain and improve this technology.

The concept of Information Dispersal is used in order to achieve high availability, adequate performance and self-contained fault tolerant backup system. Information Dispersal is the process of splitting original data into various pieces and storing them on different nodes in encoded form and the original file then can be retrieved by reassembling the pieces at the receiving device. Information Dispersal has the ability to disperse data securely across a number of nodes so that compromising one node will not result in any data being compromised. Our approach ensures that maximum availability of user file is assured by dispersing the data parts as well as redundant information on multiple data centres rather than explicitly storing on single Cloud service provider. User sets his budget above certain minimum predefined limit and expects maximum availability of data as per budget.

We proposed a architecture for heterogeneous Cloud system. In this algorithm we tackle several issues concerning the dispersal of information on multiple data centres while also making cloud storage more cost-effective. As the data centres in heterogeneous Cloud system have different availability rates and storage costs, optimising algorithm is used to disperse information on various such data centres so as to meet the data availability requirements of end user defined in SLA.

In this paper related work is discussed in section 2. System model is presented in section 3. Section 4 discusses the use case of Erasure codes and Reed Solomon coding with solved example. The proposed algorithm described in section 5 and results are discussed in section 6. At last Conclusion is done in section 7 followed by references.

II. RELATED WORK

Information Dispersal is the process of splitting original data into various unrecognizable parts dispersed in different network locations, so that redundant parts protects the information in the event of a location disruption, but unauthorized access at any location does not provide useable information. This Data can

later be reassembled at the receiving device. H. Xu et al. Presented approach for secure and reliable data storage on the Cloud using reed Solomon codes. In this paper, they also discuss the calculation of optimal number of checksum parts required to add redundant information to user data files [1]. M.O. Rabin presented an approach for effectual dispersal of information for security, load balancing, and fault tolerance [2]. Gomez et al. presented scalable Erasure Coding algorithm for low computational overhead cost and a minimum measure of communication [3]. Khan et al. guided for load balancing and incremental scalability in data centres by applying erasure coding techniques in Cloud systems [4].Though these approaches can significantly enhance Cloud data reliability, the end user gets no support to deal with performance of service providers. Plenty of work has been done in the field of Cloud data security, to which our approach is closely related. Hwang and Li proposed to protect shared Cloud data objects with use of data colouring and software watermarking techniques [5]. Their approach provides sole access to user to their Cloud data and also prevents data from being damaged, stolen etc. Shue et al. presented a Cloud-based system which distributes workload evenly across various virtual machines in order to enhance the system performance and maximize utilization [6]. Adi Shamir proposes another technique which enables robust key management schemes construction for cryptographic systems [7]. Hugo krawczyk presented m-threshold scheme, where m shares recover the secret but m - 1 share give no (computational) information on the secret [8]. These techniques ensure security and reliability even if half of the pieces are destroyed and all but one of the remaining pieces is exposed. A significant amount of research has been conducted cantered on secure and reliable Information Dispersal but very little has been done considering the storage cost paid by the user for dispersing information onto several data centres. To establish the effectiveness of proposed approach, we represented an architecture which uses datacenters of three types, which allow cost effective, secure and reliable data storage on the Cloud.

III. SYSTEM ARCHITECTURE

In this paper a architecture is proposed for heterogeneous Cloud environment. In our multi-tier architecture, each tier has data centres with different configurations than others as shown in figure 1. All parity pieces and data pieces of user's file are dispersed on multiple data centres so as to achieve maximum availability.

As shown in figure below, the user sends a request to the Cloud broker. The Cloud broker then interacts with the Cloud Information System (CIS) in order to retrieve a catalogue of registered data centres along with the other required details concerning the availability and cost factor of data centres. The broker then passes on the user's file and information from the CIS to the scheduler along with the budget. Then it is used for the ordering of those data centres so as to minimize the cost of storage and maximise the availability factor under the user's budget. Ultimately, the Scheduler provides this list to the broker so to disperse user's data on these data centres efficiently. The basic unit of storage is considering as block.

As shown in the diagram, there are three types of data centres in our architecture and let $\{dc_1, dc_2 \ldots dc_i\}$ be a set of 'i' data centres and $i \geq 2$. Suppose the block size be fixed on all data centres be 'b', the storage cost per block be $\{c_1, c_2, c_3\}$ and value be $\{v_1, v_2, v_3\}$.The value factor of data centres is basically used for ranking the system and is decided by the cloud service provider.

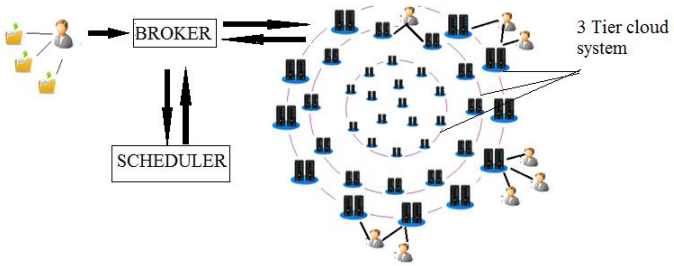

Figure 1: Architecture for Cloud storage based on Information Dispersal

Let the set of j different files and the set of r registered users be $\{f_1, f_2 \ldots f_j\}$ and $\{u_1, u_2, \ldots u_r\}$ respectively. Suppose any user wants to store a file f_a of size S_a on cloud with n_a data parts and m_a optimal parity parts are calculated as explained in section 4. Then the total numbers of parts of file f_a are (n_a+m_a). The user can use cloud data storage plan chart in order to distribute (n_a+m_a) parts on multiple data canters. The User can choose a suitable plan and submits his budget B_a. The Cloud broker then passes the user's budget as well as selected plan to scheduler. The scheduler gives the list of data canters for the distribution of data, with total storage cost under user's budget B_a. Ultimately all the data files (n_a+m_a) are allotted to data canters according

the provided list by scheduler, as discussed in section 5. Since none of the data centre has the complete information about the user's file, this approach provides effective security to the threats.

For retrieval the user u_a then can download the stored file f_a from the cloud by requesting the cloud broker. Initially the broker will make an attempt to download 'n' data pieces of f_a from all the available data canters in the set £ where the files parts were stored. If all 'n' data parts are available, then the entire original file f_a is reconstructed.. But in case if one or more data canters fail, then automatically all available data pieces n_a' and parity parts m_a' are downloaded. The original file f_a can be reconstructed by decoding lost data pieces through available n_a' data pieces as long as $n_a'+m_a' >= n_a$. It should be noted that here the parity pieces acts as redundant data and hence our architecture is fault tolerant and reliable.

IV. ERASURE CODE AND REEDSOLOMON CODING

A. Erasure code

Data replication was used as a fault tolerant technique in earlier days. Multiple versions of the same data were stored on multiple servers in order to increase the reliability of the data. Replication of data results in increasingly larger amounts of Cloud data, and eventually becomes impractical to implement Erasure codes, also called as forward error codes offer a method of data protection wherein using mathematical functions, the data is fragmented, expanded and encoded with redundant data pieces and are stored across the group of different locations on storage media [9].The purpose of erasure coding is to facilitate in reconstruction of data that is being corrupted during the process of storage with information that is stored elsewhere in the vectors. Advantage of Erasure codes over traditional RAID is its capability to minimize the time period and overhead needed to reconstruct the data.

B. Reed soloman code

All n data parts of original file are encoded into m checksum pieces using Reed solomon algorithm such that any n parts available out of total n+m parts are suffice to reconstruct original file [9]. The process of Reed soloman coding is explained using following example.

$[A'] \times [\text{Data}] = [B']$

$$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\1&2&2\\2&1&2\end{bmatrix} \times \begin{bmatrix}1&2&3\\2&3&1\\1&0&0\end{bmatrix} = \begin{bmatrix}1&2&3\\2&3&1\\1&0&0\\7&8&5\\6&7&7\end{bmatrix} \quad (1)$$

Now suppose the2nd and 3rd rows of matrices are deleted in equation 1, we get equation 2 as

$$\begin{bmatrix}1&0&0\\0&1&0\\2&1&2\end{bmatrix} \times \begin{bmatrix}1&2&3\\2&3&1\\1&0&0\end{bmatrix} = \begin{bmatrix}1&2&3\\2&3&1\\6&7&7\end{bmatrix} \quad (2)$$

Where $A^{-1} \times A = \text{Identity}$

Therefore on Multiplying A^{-1} on both sides of equation 2 we get data as follows

$[\text{Data}] = [A^{-1}] \times [B]$

$$[\text{Data}] = \begin{bmatrix}1&0&0\\0&1&0\\-1&-1/2&1/2\end{bmatrix} \times \begin{bmatrix}1&2&3\\2&3&1\\6&7&7\end{bmatrix}$$

$$[\text{Data}] = \begin{bmatrix}1&2&3\\2&3&1\\1&0&0\end{bmatrix}$$

V. COST AWARE INFORMATION DISPERSAL ALGORITHM

Cost aware Information dispersal architecture has 3 phases, which are discussed below:-

Phase 1:- Obtaining total number of data parts of file

Consider file f_a of user ua having size S_a is divided into n_a data parts and $n_a \geq 2$ then total number of data parts for file fa is calculated as

$$Na = \lceil S_a/b \rceil \quad (3)$$

where b is block size in all data canters.

Calculating optimal number of checksum pieces

File parts ($n_a + m_a$) could be stored on N = {1,2,3.....n_a+1} data canters and the combination of N data canters could be any of three types of data canters available in system as per Table I. The cost and value range for each data centre is provided to the user. The numbers of parity parts to be stored are required for calculating storage cost and value. Suppose m_a is number of parity parts and N data canters in set £ = {dc_1,dc_2,…….dc_N} are

allotted for storage of data. It should be noted that n_a remains fixed for f_a and number of m_a checksum parts depends on N in each plan.
From total n_a+m_a parts, any n_a parts are needed to reconstruct the original file and let the maximum number of data centers out of all N be X that become unavailable or are allowed to fail simultaneously.

Phase 2: Generation of the Cloud data storage plan chart
For storage of all file parts ($n_a + m_a$), a chart is prepared corresponding to allotted number of N data canters on which information is dispersed. This chart displays cost range and value range corresponding to N where $2 \leq N \leq (n_a+1)$. According to the data centre chosen i.e ordinary data centre, super data centre, main data centre, the cost and value varies. Following algorithm can be used to calculate cost and value range for each distinct value of N where $2 \leq N \leq (n_a+1)$.

Input :- c_1,c_2,c_3,v_1,v_2,v_3
Output :- min, max, min_avail, max_avail
Initialise min=∞ ,max = -∞,z=0;
If(N≥2) then
For i=N to 0
For j=N-i to 0
set K=N-(i+j)
$C_N[z]=(i\times c_1 + j\times c_2 + k\times c_3)\times y$ and store i||j||k pattern for each iteration
$V_N[Z]= (i\times v_1 + j\times v_2 + k\times v_3)$
If($C_N[z]$ <min) then
min=$C_N[z]$
min_avail = $i\times v_1 + j\times v_2 + k\times v_3$
end if
If($C_N[z]$ >max)
max= $C_N[z]$
max_avail = $i\times v_1 + j\times v_2 + k\times v_3$
increment z , end if
end for
end for
end if
return (min, max, min_avail, max_avail)

The value, minimum and maximum cost is calculated for all possibilities C_N^{N+2} for selecting N data canters. The vector $C_N[]$ is used for storing cost of storage and the vector $V_N[]$ is used for storing value of each possibility. The plan chart displays the cost range and value range for each distinct data centre. As file parts are evenly distributed on allotted N data canters, hence the total storage cost is calculated by multiplying storage cost per block for each data centre with y.

Phase 3: Dispersal of data and checksum pieces over multiple data canters
User u_a can select one suitable plan from the constructed cloud storage plan chart and submits his budget B_a to broker within the range of the plan the user has opted. The broker provides budget B_a, opted plan id to the scheduler. Cost and value for all possibilities corresponding to N are already stored in vector $C_N[],V_N[]$.Ordinary data canters are the least expensive, are closest to user and have the least value. On the other hand, main data canters are more expensive than ordinary data centre and possess lesser value. Super data canters are the most expensive but still have highest value due to maximum availability. Thus optimal allotment amongst opted plan with minimum cost and maximum availability factor within budget B_a is calculated using following algorithm.

Input :- B_k
Output :- d_allot
Initialise temp = -∞
For i=0 to length($C_N[]$) and $C_N[] \leq B_K$
If($V_N[i]$>temp)
temp= $V_N[i]$
d_allot = i||j||k pattern for datacenter allotment
end if
end for
return (d_allot)

Scheduler calculates d_allot = i||j||k in same plan id. Accordingly i super data canters, j main data canters and k ordinary data canters are allotted on which file parts (n_a+m_a) are stored where $C_N[]$ is vector storing cost of

storage for all possibilities of N data canters as per opted plan.V_N[] is vector storing corresponding availability factor for all possibilities of N data canters as per opted plan.

VI. RESULTS

Creation of 6 datacenters in virtual cloud environment is shown in Table 1. Three types of datcenteres are specified. Here in the created environment, 2 ordinary datacenters, 2 main datacenters and 2 super datacenter are created in all making sum total of 6 datacenters.

Table 1: Shows description of created virtual environment

Datacenter type	Count of datacenters created	Cost of data storage (each datacenter)	Availability proportion
ordinary Datacenter	2	100	2
main Datacenter	2	300	4
super Datacenter	2	500	6

Table 2: Details of datacenters

S No.	Datacenter category	Number of datacenters	Storage cost per block	Value per datacenter
1	Super Datacenter	2	100	5
2	Main Datacenter	2	60	3
3	Ordinary Datacenter	2	30	2

Characteristics of 6 datacenters in created environment including 2 ordinary datacenters, 2 main datacenters and 2 super datacenter is shown in Table 2.

User of cloud willing to store data on cloud and submits file abc.txt of size 240KB and fixed block size of all datacenters is taken as 60KB. Size of each data part is taken equal to size of block i.e. 60 KB. Total number of data parts of 240KB file will be 4, each of 60 KB and the optimal parity parts corresponding to distinct value of N, cost and value are calculated. The cloud data storage plan chart describes various plans and gives provision to user to choose number of network locations on which user wants to disperse data is shown in Table 3.

Table 3: Cloud data storage plan chart

Plan id	No. of allotted datacenter	Data parts distribution	Parity parts distribution	Cost range	Value range
1	2	(2,2)	(2,2)	240-800	4-10
2	3	(1,1,2)	(1,1,0)	180-600	6-15
3	4	(1,1,1,1)	(1,1,0,0)	240-800	8-20
4	5	(0,1,1,1,1)	(1,1,1,0,0)	300-1000	10-25

Also this cloud data storage plan chart gives an idea of cost expenditure and value range of each plan using which, user can decide budget .

Table 4: Description of all parity parts and data parts for user1's file

S No.	Name of file part	Type of filepart
1	abc.txt.0	data part
2	abc.txt.1	data part
3	abc.txt.2	data part
4	abc.txt.3	data part
5	abc.txt.4	parity part
6	abc.txt.5	parity part
7	abc.txt.6	parity part

Suppose user 1 submits file "abc.txt" and choses plan 2 with budget of 500. The various file parts created for file abc.txt are shown in Table 4 description for each part is provided that whether the part is data part or paarity part.The optimisation algorithm results out 3 datacenters out of all available 6 datacenters. Datacenter1, Datacenter2, Datacenter4 are the alloted datacenters to user 1 with maximum availability proportion in provided budget.

Now suppose user 2 stores data on cloud with different plan id.User of cloud willing to store data on cloud submits file 'file2.txt' of size 230KB and fixed block size of all datacenters is taken as 60KB . Size of each data part is taken equal to size of block i.e 60KB .The cloud data storage plan chart describes various plans and gives provision to user to chose number of network locations on which user wants to disperse data. Also cloud data storage plan chart gives an idea of cost expenditure and value range of each plan using which user can decide budget .

Total number of data parts of 230KB file will be 4 each of 60 KB and the optimal parity parts corresponding to distinct value of N ,cost and value are calculated as in Table 5.

Table 5: Description of all parity parts and data parts for user2's file

S No.	Name of file part	Type of file parts
1	file2.txt.0	data part
2	file2.txt.1	data part
3	file2.txt.2	data part
4	file2.txt.3	data part
5	file2.txt.4	parity part
6	file2.txt.5	parity part
7	file2.txt.6	parity part

Suppose user 2 submits file "file2.txt" and choses plan 3 with budget of 800. The optimisation algorithm results out 4 datacenters out of all available 6 datacenters. Description for each part is provided that whether the part is data part or paarity part. The optimisation algorithm results out 4 datacenters out of all available 6 datacenters. Datacenter1, Datacenter2, Datacenter4,datacenter 5 are the alloted datacenters to user 2 with maximum availability proportion in provided budget.

Let user 3 stores data on cloud with different plan id. User of cloud willing to store data on cloud submits file 'file3.txt' of size 220KB and fixed block size of all datacenters is taken as 60KB . Size of each data part is taken equal to size of block i.e 60KB. Total number of data parts of 220KB file will be 4 , each of 60 KB and the optimal parity parts corresponding to distinct value of N, cost and value are calculated as in Table 6.

Table 6: Description of all parity parts and data parts for user3's file

S No.	Name of file part	Type of filepart
1	file3.txt.0	data part
2	file3.txt.1	data part
3	file3.txt.2	data part
4	file3.txt.3	data part
5	file3.txt.4	parity part
6	file3.txt.5	parity part
7	file3.txt.6	parity part

Observations have been made in situation of 2 datacenters failure. Table 7 shows allotment of data parts and parity parts to corresponding datacenters and file retrieval probability (when any 2 allotted dc fails) at budget amount 800. Here file could be retrieved in 2 cases of dataparts, parity parts combination as 3, 2 and 4, 2.

Consider first case in Table 7 where data parts are 3 and parity parts are 2(3+2=5). To retrieve file, any of 3 out of all 5 parts should be available, otherwise file could not be retrieved. Similarly file retrieval probability is calculated for all possibilities .less is the size of block, more will be file distribution. When user sets budget as 1500 in virtual environment described in Table 4 out of available 5 datacenters, datacenter1, datacenter2, datacenter3, datacenter4, datacenter5 gets allotted as a result of optimization algorithm as shown in Table 8.

Table 7: Allotment of data parts and parity parts to corresponding datacenters and file retrieval probability (when any 2 allotted dc fails) at budget amount 800

Data, parity parts	DATA CENTER 1 (Alloted)	DATA CENTER 2 (alloted)	DATA CENTER 3 (not alloted)	DATA CENTER 4 (alloted)	DATA CENTER 5 (alloted)	File retrieval probability (when any 2 allotted dc fails)
3,2	ds[2]	ds[1],ps[2]	-	ds[3]	ps[1]	0.5
4,2	ds[2],ps[2]	ds[1],ps[1]	-	ds[3]	ds[4]	0.16
5,3	ds[2],ps[1]	ds[1],ds[2]	-	ds[3],ps[3]	ds[4],ps[8]	0
6,3	ds[2],ds[6]	ds[1],ds[5], ps[2]	-	ds[3],ps[3]	ds[4],ps[1]	0
7,4	ds[2],ps[6], ps[1]	ds[1],ds[5], ps[2]	-	ds[3],ds[7], ps[3]	ds[4],ps[4]	0

Table 8: Allotment of data parts and parity parts to corresponding datacenters and file retrieval probability (when any 2 allotted dc fails) at budget amount 1500

Data, parity parts	DATA CENTER 1 (Alloted)	DATA CENTER 2 (alloted)	DATA CENTER 3 (alloted)	DATA CENTER 4 (alloted)	DATA CENTER 5 (alloted)	File retrieval probability (when any 2 alloted dc fails)
3,2	ds[1]	ds[2]	ds[3]	ps[1]	ps[2]	1
4,2	ds[1],ps[1]	ds[2],ds[7]	ds[3]	ds[4]	ps[2]	0.6
5,3	ds[1],ps[1]	ds[2],ps[2]	ds[3],ps[3]	ds[4]	ds[5]	0.7
6,3	ds[1],ds[6]	ds[2],ps[1]	ds[3],ps[2]	ds[4],ps[3]	ds[5]	0.6
7,4	ds[1],ds[6],ps [1]	ds[2],ds[7]	ds[3],ps[2]	ds[4],ps[3]	ds[5],ps[4]	0.6

We can conclude from figure 3 that to get more distribution of data i.e. more number of data parts and parity parts as well as to get more security of data and higher probability of file retrieval in case of 2 datacenter failure, user need to increase budget. If only one datacenter fails, then in every case, Reed-Solomon is able to retrieve the original file, when number of parts accessible are greater than or equal to n(where n is the total number of data parts).

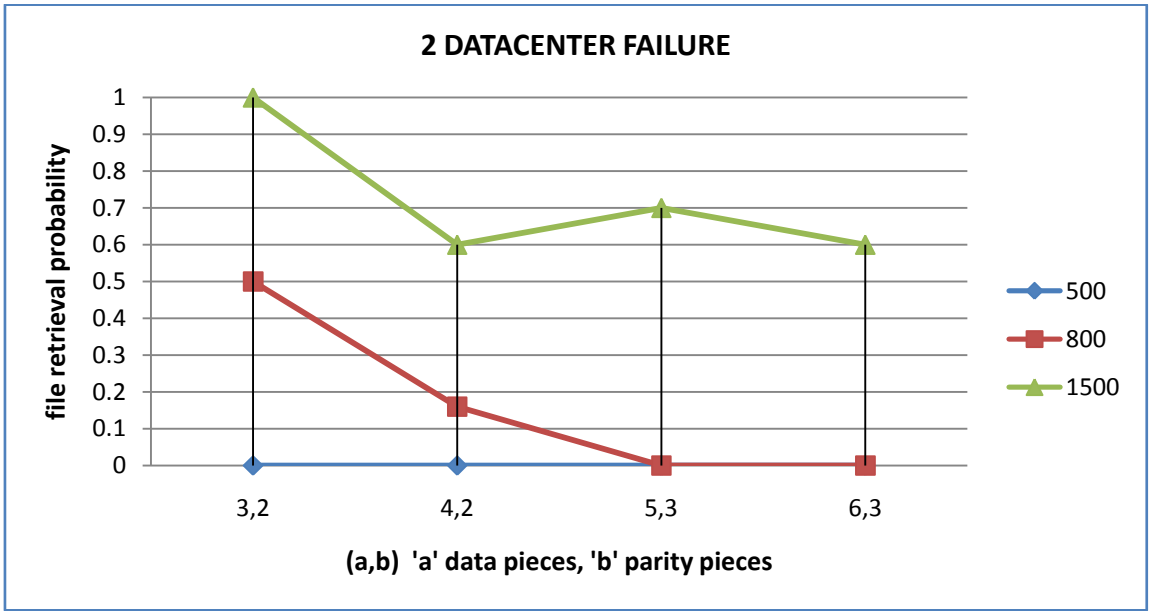

Figure 3: File retrieval probability versus data, parity parts against particular budget amount

ROLE OF COST OPTIMISATION ALGORITHM

Cost optimization algorithm is used to allot those datacenters out of all available, whose cost of storage sums up to user's budget and the distribution of users file parts could be on more number of datacenters with maximum availability proportion. Figure 4shows here the comparison of 2 cases with and without using optimization algorithm.

Without Cost Optimization Algorithm

Suppose that out of all available datacenters (1 main datacenter (cost 300 each), 3 super datacenter (cost 500 each) and one ordinary datacenter (cost 100 each)), in this case obviously user whose budget is 1500 opts to deploy data at available 3 super datacenters (3* 500=1500). Performance, availability of these super datacenters could not be challenged but as our goal is to retrieve file in case of datacenter failure that is impossible in this case where only 3 datacenters are used to deploy data and 2 of them fails.

With Cost Optimization Algorithm

Let's consider case when user sets budget 1500, all available datacenters (2 main datacenter, 2 super datacenter and one ordinary datacenter), gets allotted as a result of optimization algorithm.

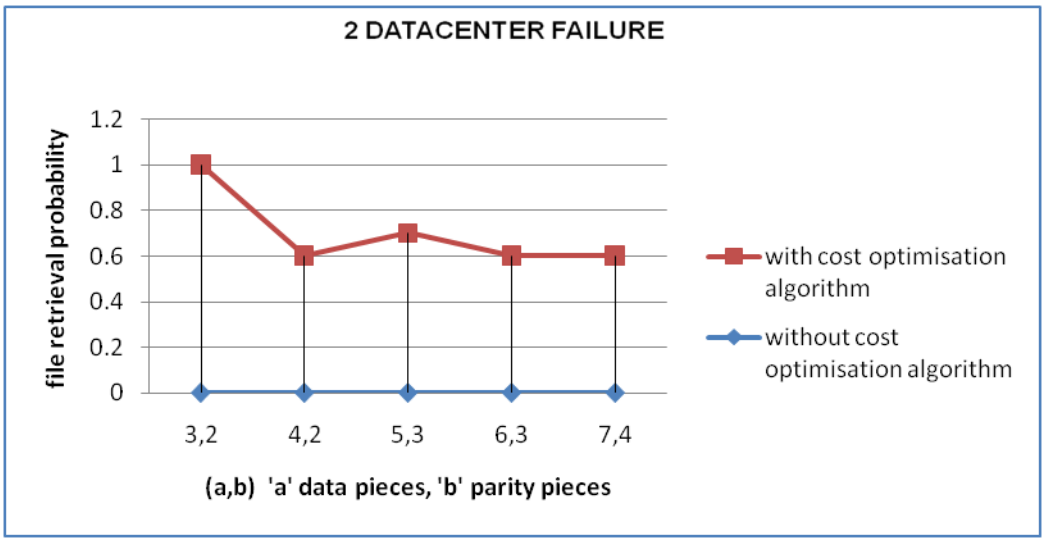

Figure 4: File retrieval probability vs data,parity parts with and without using optimisation algorithm

Table 9: Observations for scenario of using IDA with cost optimization algorithm and without cost optimization algorithm

Scenario	No. of datacenters allotted	Comments
with cost optimization algorithm	1 super datacenter 2 main datacenter 2 ordinary datacenter	More distribution of data, more security, good availability proportion, file retrieval even when 2 datacenter fails
without cost optimization algorithm	3 super datacenter	Best case when none or one of datacenter fail but no file retrieval is possible when 2 datacenters fail.

Hence more data distribution occurs i.e. at 5 datacenters in same budget. More data security and higher will be the probability of data retrieval in this case.

VII. CONCLUSION

In this paper, we have considered reliability, security and cost efficiency which are some of the major issues in cloud storage. We have proposed secure, distributed cost aware cloud storage architecture for end users rather than using redundancy at server side. In this proposition erasure coding is used for calculating redundant data files and uploaded onto various network locations. Multiple failures of data centres are easily handled by adding redundancy to data files. We have proposed an algorithm that solves the issue of cost efficiency. Optimal calculation can be performed for calculating number of checksum pieces that helps to achieve space efficiency. As the user's information is distributed over different sites, all the user data cannot be accessed from single site. This is how the user data is protected from unauthorized access in Cloud and helps in resolving the security issues. Along with the benefits of security and fault tolerance, our approach also provides cost effective architecture for data storage in cloud environment.

REFERENCES

[1]. H. Xu ,D. Bhalerao, "Reliable and secure Distributed Cloud data storage using Reed Solomon codes", International Journal of Software Engineering and Knowledge Engineering ,(2015).

[2]. M.O.Rabin, "Efficient Dispersal of Information for Security,Load Balancing, and Fault Tolerance", Journal of the Association for Computing Machinery, April 1989.

[3]. L. B. Gomez, B. Nicolae, N. Maruyama, F. Cappello and S. Matsuoka, "Scalable Reed-Solomon-based reliable local storage for HPC applications on IaaS Clouds", in Proc. of the 18th International Euro-Par Conference on Parallel Processing (Euro-Par '12), Rhodes, Greece, August 2012, pp. 313-324.

[4]. O. Khan, R. Burns, J. Plank and W. Pierce, "Rethinking erasure codes for Cloud file systems: minimizing I/O for recovery and degraded reads", in Proc. of the 10th USENIX Conference on File and Storage Technologies (FAST-2012), San Jose, CA, USA, February 2012, pp. 20-33.

[5]. K. Hwang and D. Li, "Trusted Cloud computing with secure resources and data coloring", IEEE Internet Computing 14 (5) (2010) 14-22.

[6]. D. Shue, M. J. Freedman and A. Shaikh, "Performance isolation and fairness for multi-tenant Cloud storage", in Proc. of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI '12), Hollywood, CA, USA, October 8-10, 2012, pp. 349-362.

[7]. A. Shamir, "How to share a secret", Communications of ACM, November 1979.

[8]. Hugo Krawczyk, "Secret Sharing Made Short" , IBM T.J. Watson Research CenterYorktown Heights, NY 10598

[9]. Haiping Xu ,Deepti Bhalerao, "Reliable and secure distributed cloud storage using reed Solomon codes" , International Journal of Software Engineering and Knowledge Engineering © World Scientific Publishing Company

Makhan Singh. "Cost-Aware Cloud Storage Architecture Using Information Dispersal Algorithm." .IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 12, 2018, pp. 41-49.