Optimization of Data Placement using Principal Component Analysis based Pareto-optimal method for Multi-Cloud Storage Environment

V. L. Padma Latha †, Dr. N. Sudhakar Reddy ††, and Dr. A. Suresh Babu †††

†Research Scholar, Department of CSE, SVCE Tirupati, JNTUA University, Ananthapur, India
††Professor, Department of CSE, SVCE, Tirupati, Andhra Pradesh, India
†††Professor, Department of CSE, JNTUA University, Ananthapur, India

Abstract
Now that we're in the big data era, data has taken on a new significance as the storage capacity has exploded from trillion bytes to petabytes at breakneck pace. As the use of cloud computing expands and becomes more commonly accepted, several businesses and institutions are opting to store their requests and data there. Cloud storage's concept of a nearly infinite storage resource pool makes data storage and access scalable and readily available. The majority of them, on the other hand, favour a single cloud because of the simplicity and inexpensive storage costs it offers in the near run. Cloud-based data storage, on the other hand, has concerns such as vendor lock-in, privacy leakage and unavailability. With geographically dispersed cloud storage providers, multicloud storage can alleviate these dangers. One of the key challenges in this storage system is to arrange user data in a cost-effective and high-availability manner. A multicloud storage architecture is given in this study. Next, a multi-objective optimization problem is defined to minimise total costs and maximise data availability at the same time, which can be solved using a technique based on the non-dominated sorting genetic algorithm II (NSGA-II) and obtain a set of non-dominated solutions known as the Pareto-optimal set.. When consumers can't pick from the Pareto-optimal set directly, a method based on Principal Component Analysis (PCA) is presented to find the best answer. To sum it all up, thorough tests based on a variety of real-world cloud storage scenarios have proven that the proposed method performs as expected.

Keywords: *Cloud Computing; Non-Dominated Sorting Genetic Algorithm II. Privacy Leakage; Vendor Lock-in; Multi-Cloud; Principal Component Analysis;*

1. Introduction

"Cloud computing" means renting specified quantities of CPU time and storage to many customers according to their individual needs [1]. It's been a while since cloud computing principles expanded to include pre-configured computing surroundings, as well as software that's fully installed and ready for use. Data resources and big data sets such as GenBank, the 1000 Genomes Project [2] and ExAC [3] are also kept in clouds. As far as services go, different cloud concepts can be categorized as Data as a Service,Software as SaaS,IaAS, and PaaS. "Elasticity" is the defining characteristic of cloud computing models. In IT infrastructure, under-provisioning and over-provisioning are two common problems [4]. Cloud-based solutions can overcome these concerns because just the resources that are needed are hired and paid for.

As a way to lessen the burden of maintaining storage and computing necessities in-house, and to have access to sufficient resources when needed[5], cloud computing is the preferred method. It has the benefit of giving you full switch over the computing schemes. Cloud computing models may make it harder to deploy and utilize new software [6]. Another advantage is that all data is kept within the company's network, thus there is no need to transfer huge data sets, which reduces the cost and necessities for a cable. These advantages come at a price. An upsurge in sequencing capacity often requires novel investments in the IT infrastructure for storage and examination because this approach is not easily scalable. Data storage problems are only one aspect of efficient data handling [7-8].

Applications that deal with data management could be deployed in the cloud. As a result of the high initial hardware and software costs associated with on-premises business database systems [9], it is difficult to justify their utilization. The pay-as-you-go cloud computing approach, as well as having someone else worry about maintaining the hardware, is particularly tempting to many firms (especially start-ups and medium-sized corporations). It's in this sense that cloud computing resembles the ASP model and database as a service paradigm [10]. These platforms work differently from ASPs and DaaS in practice. Vendors of cloud computing often do not own, install, or manage the database software for their clients (often in a multi-tenancy design), but instead provide virtual computers on which customers can install their own software [11-12]. Ressource availability is often elastic, since compute power and storage are readily available on demand, with the pricing model paying only for what is used.

From the point of view of the user, the most important problem is to increase the availability of data while

Manuscript received December 5, 2021
Manuscript revised December 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.12.36

lowering the cost of data management, which includes storage and network costs. To get the most use of cloud storage, optimising multi-objective functions is necessary. First and foremost, a multi-cloud storage architecture is discussed in this work. The next step is to formulate a multi-objective optimization problem that aims to reduce monetary expenditures while also improving data accessibility. Another method is then provided that uses PCA in conjunction the new method of erasure coding in order to solve the multi-objective optimization problem and arrive at a list of non-dominated solutions (i.e., a list of providers (CSP)). For this reason, it's difficult to balance data management cost vs data availability when using CSPs with low (or high) storage costs. For certain users, the Pareto optimum set has only one alternative for data location. However, when confronted with the Pareto-optimal set, the majority of users remain perplexed. A PCA-based method is being considered to help offer a solution to such consumers. Finally, we use a computer simulation to demonstrate the effectiveness of the suggested approach while dealing with CloudHarmony's real-world cloud storage providers.

The remaining paper consists of literature review that contains the study of existing techniques in Section 2. The system model and problem definition is given in Section 3, where the proposed method is given in Section 4. The validation of proposed method with existing techniques is described in Section 5. Lastly, Section 6holds the conclusion of the research work.

2. Literature Survey

Putting all data in one cloud has been a source of great worry. [14] predicts that customers will become less interested in a single cloud strategy due to service unreliability problems and the threat of harmful insiders. Several new research on data storage in multi-cloud setups have recently been published as a result of the movement toward multi-cloud. It was described in [15] as a multi-cloud technique that develops the availability and integrity of cloud-stored data. Replication and erasure coding are the two major redundant strategies to classify data distributed storage. The authors of [16] [17] contrasted these two approaches. Erasure coding was adopted by the developers of [18] to increase the availability of grid data storage. The link between availability and replica number was captured in [19].

Data storage in multi-cloud systems must take into account a number of aspects, including cost, data availability, security, and latency. "SCMCS" is a multi-cloud storage model with a high level of availability and security, as described in [20]. To achieve this, several unjustified assumptions are made in order to arrive at an optimal solution that minimizes storage costs while maximizing QoS. These experiments have shown no conclusive results. A data storage strategy that adapts to user access patterns was proposed in [21]. According to [22], data hosting and storage mode transitions were included in the CHARM scheme. Data availability and cost reduction in multi-cloud setups are the major goals. As a result of this, their study does not properly evaluate the trade-off between availability and cost.

Redundant Array of Cloud Storage (RACS), a proxy that strips user data across various suppliers to reduce the cost of switching providers, is proposed by Abu-Libdeh et al. However, no solution to the data location challenge is offered in order to satisfy any optimization objectives. Scalia was influenced by RACS, according to Papaioannou et al. Adaptive data placement in the cloud is made possible by this cloud storage brokerage solution, which reduces storage costs. Mansouri et al. [25] also mention this. Describe a method for selecting subsets of data centres where the original data and its copies may be safely kept to save storage costs while ensuring projected availability.

By using a commodity flow solution, Hadji reduces the cost of storing data and the time it takes to reach data centres in [26]. When consumers access their data, the expenses of the network and operation are completely ignored. They use an ensemble of replication and erasure coding to reduce storage, bandwidth, and latency costs, but they overlook the importance of selecting the right CSPs.

An ant colony algorithm-based solution is proposed by Wang et al. [28] to reduce financial costs and maximise data availability. For the sake of simplicity, the authors, however, employ the weights to calculate the end optimization target of the integrated QoS value. A multi-objective optimization is one that seeks to maximise more than one desirable result. Using erasure coding, Su et al. [29] provide an organised paradigm for representing data placement in multi-cloud storage. It is capable of resolving the location of data under complicated conditions. Instead of using Euclidean distance to find the optimum solution to a multi-objective optimization problem, they use optimization weights that are selected based on subjective criteria.

In this research, we compare the cost and data availability optimization problems and develop the PCA technique to tackle both.

3. System and Problem Description

In this section, we concisely discuss problem statement, and then based on that, we express a data management model. Later, we define a multi-objective optimization problem based on data management formulation.

3.1 Problem Statement

User Demand Statistic, Cloud Storage Information Collection, Data Retrieving and Hosting are the four components of multi-cloud storage. The Data Demand Statistic gathers information on user requirements, such as the amount of data needed, when it needs to be available, and how often it needs to be accessed. Using Cloud Storage Information Gathering, Cloud Harmony, a third-party website that collects and monitors information on cloud service providers, such as charges, characteristics, service status and more, is used to gather information about cloud storage providers. The framework's two most important parts are Data Hosting and Data Retrieving.

Data Hosting chooses which clouds to use for storing and distributing the data. To get the data of a particular user, Data Retrieving chooses which clouds to utilise. In order to achieve high availability, these two components are dependent on storage methods that extensively employ erasure coding (EC). The data item may be broken into m equal-sized chunks using (m, n)-erasure coding, and the (n - m) pieces can be encoded using m data chunks. Erasure coding's fundamental feature is the ability to retrieve original data from any number of m data pieces. The fundamental goal of the scenario described above is to determine CSPs and erasure coding parameters to optimise data placement based on user demands.

3.2 Problem Definition

To well describe a data management model, we introduce the subsequentdescriptions.

Definition 1 (Cloud Service Provider)**.** The data management model is signified as a set of independent cloud service providers $C = \{SP_1, SP_2, .., SP_N\}$where each cloud service provider the storage service. Each CSP has tuple: $CSP = \{P_{si}, P_{bi}, P_{oi}, a_i\}$ where: P_{si} signifies the storage cost per unit size in CSP i;

1. P_{bi} is the out-bandwith cost per unit size in CSP i;
2. P_{oi} defines the cost for GET request in CSP i; and
3. $a_i$represents the probability of CSP i being available (i.e. $availability$).

Definition 2 (Data File). We assume that a data file is related with a triples: $DF = \{S, \tau, A_{req}\}$, where:

1. S is the size of a data file user stores;
2. τ denotes user data access frequency, which is equal the data access count during a time period; and
3. A_{req}defines user's required data file availability.

The goal is to choose CSP and erasure coding settings (m; n) that minimise storage and data GET costs, as well as network expenses, while increasing data availability. This will reduce overall costs while maintaining data availability. We'll make the assumption for the sake of simplicity that each CSP stores only one data piece. Note that the ensuing definitions for availability and cost are equivalent when using erasure-coding mode to host data.

Definition 3 (Erasure Coding Parameters). When using a (m; n)-erasure coding, a data file is divided into measurably smaller chunks, and the smaller chunks are then encoded into larger (n m) chunks, which comprise the original m steady portions and the smaller (n - m) parity chunks (see Figure 2). Users are able to instantly shut down in the presence of any 0(n - m) clouds**.**

Definition 4 (Data Obtainability)**.** Based on erasure coding, data availability is the sum of all cases that k CSPs are simultaneously available, where $k\epsilon\ [m;\ n]$. This depends on the fact that outage occurrences are independent among CSPs. We define $C' = fSP_1 \times \mu_1;\ SP_2 \times \mu_2, \dots, SP_N \times \mu_N\ (|C_0| = n)$ as the service list of the n block choices, where $\{\mu_i \epsilon\ \{0,1\}\ |i = 1,2, \dots N\}$, and μ_i is used to mark whether the ith$SP$is selected. $\Omega = \binom{|C'|}{k}$means the amount:

$$A = \sum_{k=m}^{n} \sum_{j=1}^{\Omega} \left[\prod_{i \epsilon S_j^{\Omega}} a_i \prod_{i \varepsilon C'/S_j^{\Omega}} (1 - a_i) \right] \quad (1)$$

whereC'/S_j^{Ω} represents the CSPs that are not in S_j^{Ω}.

Definition 5 (Storage Cost)**.** The storage cost of a data file is equivalent to the storage cost of all data chunks in nCSPs. Since each CSP stores the data chunk of size S/m, it can be defined as follows:

$$P_{stor} = \sum_{i \epsilon C'} \frac{S}{m} P_{si} \quad (2)$$

There are CSPs that charge differently depending on how much storage you utilise. To put this into perspective, the storage price for AWS S3 in USA East in the USA East region is $ 0.023 if the data is less than 50 TB, while the storage price is $ 0.022 when the data is between the range of 50-450 TB. Because we're using erasure coding to break up the data in this project, the individual chunks aren't huge. The threshold of each tier can be used to determine the storage cost, similar to the piecewise functions, if the data size is big.

Definition 6 (Network Cost)**.** Data fragments from various clouds can be used to restore a deleted file. Data retrieval is

performed using the m-cheapest clouds to minimise network costs to a minimum. Several options exist for dealing with this problem:

$$P_{net} = \min_{j\epsilon[1,\Omega]} \left(\sum_{i\epsilon S_j^{\Omega}} \frac{S}{m} \tau_t P_{bi}\right) \quad (3)$$

Definition 7 (Operation Cost)**.** The operation cost is the price customers pay for GET requests to the cheapest m CSPs to retrieve the data file. This is how it's calculated:

$$P_{op} = \min_{j\epsilon[1,\Omega]} (\sum_{i\epsilon S_j^{\Omega}} \tau_t P_{oi}) \quad (4)$$

It is worth noting that the value of j in Equation (3) is equal with that in Equation (4).

Definition 8 (Total Cost)**.** The total cost of a data file $C_T$is the sum of *storage cost*, *operation cost*, and *network cost* and is defined as follows:

$$C_T = P_{stor} + P_{net} + P_{op} \quad (5)$$

3.3. Optimization Problem

We formalise the problem of data placement optimization based on a data management model. Maximize data file availability while minimising overall costs are the goals of this strategy. Here's how you characterise the whole optimization challenge:

$$\begin{cases} Maximize\ A \\ Minimize\ C_T \end{cases}$$

Subject to:

$$\boldsymbol{A \geq A_{req}}$$

As previously mentioned, constraint 1 ensures that the availability of a data file will not be less than the level of availability requested by the user.

4. Proposed Methodology

If the Pareto-optimal front for an M-objective issue is less than M-dimensional, some of the objectives become redundant. The NSGA-II approach and PCA were used to target these. After a series of iterations in which each step progresses towards the Pareto optimal zone, the suggested technique adapts to locate the correct lower-dimensional interactions.

4.1 PCA Analysis for Multi-Objective Optimization

Initial data matrix for M-objective optimization problem with N population members, say X will be of size $M \times N$. In PCA terminology, each of the M objectives here represents a 'measurement type' while each of the N solutions - a 'time sample/experimental trial'. The covariancematrix is work out as shown below.

$$Covariance\ Matrix\ (V) V_{ij} = \frac{X_i X_j^T}{M-1} \quad (6)$$

$$\text{Correlation matrix (R)} R_{ij} = \frac{X_i X_j^T{}_i}{\sqrt{V_{ii} \cdot V_{jj}}} \quad (7)$$

whereX_i is the i-th row of X. A three-objective (M=3) problem DTLZ5 is used to illustrate this method (2,3). R, the correlation matrix, is illustrated in Table 1.

Table 1: A 10 data points used for DTLZ5(2,3) PCA analysis

Correlation Matrix		
1.0000	0.9997	-0.9182
-0.9182	-0.9190	1.0000
0.9997	1.0000	-0.9190
EigenValue		
2.8918	0.0003	0.1078
Eigenvalues by proportion		
0.9639	0.0001	0.0359
Eigenvectors		
0.5828	0.7055	0.4033
PCA1	PCA2	PCA2
0.5830	-0.7087	0.3973
-0.5661	-0.0035	0.8243

These matrices show a negative correlation between the first and third aims - they are at odds with one other! The second and third objectives are no exception. Because of this, while the first and second objectives do not clash with each other, the third aim does. It is clear from this matrix that one or both of the first and second objectives are redundant.

This may not be the case, however, if there are a high number of objectives and the situation is complex. Below, we propose a PCA-based approach for focusing on fewer objectives. Let's go back and cross-check our objectives' contradictory nature with our correlation matrix to further decrease the problem's dimensions.

4.2 Dimensionality ReductionEigenvalue Analysis:

R is a correlation matrix, and its eigenvalues are computed for the example problem in Table 1. They are also listed in decreasing order of magnitude. The table also contains the corresponding Eigenvectors. 'PCA1' is the name given to the first principal component (eigenvector (0.5828,0.5830,-0.5661)T). For example, the contribution of the first impartial function to this vector is represented by the first component of this vector, and so on. An objective space defined by the direction cosines of a directed-ray could be used to represent the three contributions to the problem. As the major component (axes) moves in the direction of the positive value, the objective value decreases. We can deduce from this fact that if we look at the goals associated with the maximum positive and maximum negative elements of this vector, they are the ones which

contribute most the principal component. This means that we can achieve two important conflicting objectives by selecting the most negative and positive elements from a PCA. In the example above, f 2 and f 3 are experiential to be the two ideas that are in the most direct conflict.

4.3 Multiple Principal ComponentsEffect:

It is possible to obtain information on the conflicting objectives by looking at each principal component in turn. It is suggested that the first principal component be examined, followed by a look at the second principle component and so forth, until all important components have been examined. When the cumulative influence of all previously primary components exceeds the threshold cut, we stop analysing principal components. In terms of a battle, this will not bring in less vital aims. The relative contribution of each principal component's eigenvalue is considered. For each primary component, a percentage contribution is computed in Table 1. PCA1 contributes 96.39 percent of all Principal Components, hence just the first principal component will be evaluated when the TC is 95 percent. As a result, we do not examine any further PCAs and deem the second and third points to be essential objectives for this problem in order to proceed. f 2 and f 3 can be solved using NSGA-II instead of all three objectives.

Table 1 shows that the eigenvalues are squared and more highlighted when using RRT instead of R, while the eigenvectors stay unchanged. This is due to the fact that the variance contribution of each principal component depends on the ratio of the principal component's eigenvalue to the total (TC). We chose to use RRT for the suggested scheme because it logically would make the analysis more succinct.

It is important to choose a threshold cut (TC) parameter that fits your needs and your budget. This analysis may choose numerous redundant objectives if the threshold is set too high (almost 100 percent), undermining the point of performing the PCA. Another problem arises when crucial objectives are disregarded because the value is too tiny. This results in an error in the entire research. A TC value of roughly 95 percent or higher, on the other hand, may be more dependable for a study to be reliable. In addition, the relative magnitudes of the eigenvalues can be used to determine the choice of time constant (TC). No further principal component may be evaluated if the lessening in two consecutive Eigenvalues is more than a set percentage. Also, the decision-preference maker's can be taken into account when making this decision (DM). It is possible for the DM to continue the main component analysis until all desired objectives have been selected. A number of such possibilities exist and should be explored, but we'll stick with 0.95 for now. After that, a dependable methodology may be established by adopting an iterative scheme that uses a united PCA and NSGA-II procedure (explained in the next part).

We offer the following extra process to make the dimensionality-reduction process effective and adaptable to a variety of contexts. This component includes both positive and negative objectives. After that, we check whether or not the eigenvalue is bigger than 0.01. If not, we choose the aim that corresponds to the eigenvector's highest absolute value. Different situations are considered in this case, as well as in the case where the total contribution of the eigenvalues is smaller than TC. The aim that corresponds to the highest element of the eigenvector is chosen if all members of the vector are positive. A negative Eigenvector means that all objectives are chosen. We analyze two different possibilities if, on the other hand, the absolute value of the most positive element (p) is greater than the absolute value of the most negative element (n). In the case where the ratio of p to n is less than 0.90, we select two targets. We choose n's objective, on the other hand, if $p > 0.9|n|$. The same is true if $p > |n|$. Both objectives are chosen in the event that the ratio of p to n is less than 0.80. As an alternative, if p 0.8|n|, we select the objective that corresponds to p.

4.4 Final Lessening Using the Correlation Matrix:

We can only hope that this process detects a large majority of the data set's redundant aims. This can be determined by using a reduced correlation matrix (only columns and rows that correspond to non-redundant objectives). An examination of the existence of objectives with exactly equal positive or negative correlation between them, is then conducted. It follows that any individual in such a group may build relationships with the remaining aims that were incompatible. A PCA analysis may have identified a candidate as early as possible, in which case we retain the candidate with the largest eigenvalue. As long as they are from the same PCA, however, they are chosen based on their contribution to the next (lower) PCA. No additional consideration is given to any of the other objectives in the collection. Note that after a sufficient amount of generations, the correlation matrix steadies and the correlation patterns become invariant across time.

5. Results and Discussion

CloudHarmony [30] is a third-party platform that provides credible and impartial performance analysis, reports, commentary, metrics, and tools to facilitate cloud service comparisons. In the study, we employ 35 CSPs, including 12 from Amazon S3, 4 from Microsoft Azure, 3 from Google, 7 from Alibaba, and 5 from CenturyLink (SL). As an illustration, the CSP AZ-EUN refers to Microsoft Azure's northern European cloud provider. For instance, we see that Amazon S3 has two data centres, with one located

in the USA-West area, namely Northern California's (N), Oregon's, and Ohio's (O), and the other in the USA-East. AWS-USW-N, for example, signifies that Amazon S3's CSP is located in Northern California, in the Eastern United States. It's important to note that all of the CSPs are referred to by a specification that includes information on storage

Algorithm for proposed PCA-NSGA-II

***Step* 1**: *Set an iteration counter* $t = 0$ *and initial set of objectives* $I_0 = \{1, 2, \ldots, M\}$.
***Step* 2**: *Initialize a random population for all objectives in the set and obtain a population* P_t.
***Step* 3**: *Perform a PCA analysis on* P_t *using It to choose a reduced set of objectives* I_{t+1} *using the predefined TC. Steps of the PCA analysis are as follows*:
1) *Compute the correlation matrix using equation* 5.
2) *Compute eigenvalues and eigenvectors and choose non* $-$ *redundant objectives using the procedure as discussed above.*
3) *Reduce the number of objectives further, if possible, by using the correlation coefficients of the non* $-$ *redundant objectives found in item* 2 *above.*
***Step* 4**: *If* $I_{t+1} = I_t$, *stop and declare the obtained front. Else set* $t = t + 1$ *and go to Step* 2.

outgoing bandwidth, and operating costs. Since each cloud provider's SLA guarantees the availability of their services, we've included the values of each CSP's uptime in the range of [95 percent, 99.9 percent].Algorithm developed in Java and running on a 3.40GHz CoreTM i7-6700 processor with 16GB of RAM Parameter settings have a direct impact on an algorithm's performance. Multiple experiments are required to determine the appropriate parameter value.

5.1. Storage Mode Changing

By changing DAF, we can determine the data hosting plan using 35 cloud service providers. A customer's data availability restriction is set at 99 percent, and the access frequency ranges from 0.1 to 1 with an interpolation interval of 0.01. Storage modes are identical for DAFs greater than 0.4, as indicated in Table 2. A specific case of Erasure Coding occurs when the replication mode (m = 1) is selected. The frequency ranges from 0.3 to 0.4 with a 0.01 interval. Table 3 displays the results, while Figure 1 shows a graphical representation of them.

Table 2: Storage modes with vary DAF (0.1~1)

Storage Mode (m,n)	DAF
(6,8)	0.1
(2,3)	0.2
(2,3)	0.3
(1,2)	0.4
(1,2)	0.5
…	…
(1,2)	1

Table 3: Storage mode with vary DAF (0.30~0.33)

Storage Mode (m,n)	DAF
(2,3)	0.30
(2,3)	0.31
(1,2)	0.32
(1,2)	0.33
…	…

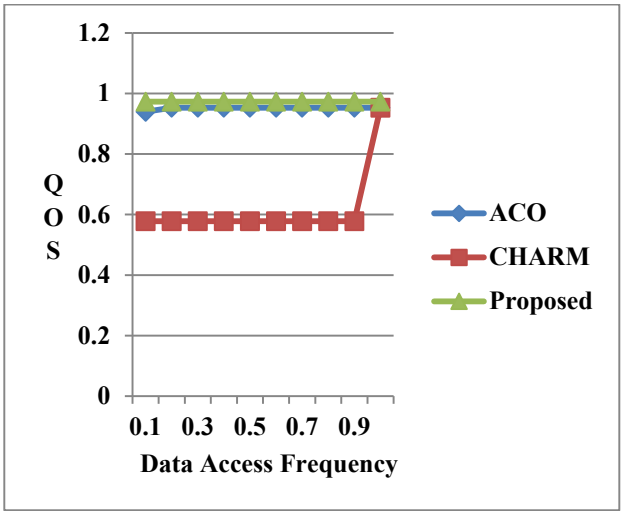

Fig 1. Comparison of three approaches regarding the integrated Qos Value of the resulting solutions

5.2 Performance of the Proposed Algorithm

In this study, we first discuss the correctness of the model before describing its performance. Provide users with cost-effective and highly available data storage is a hotspot for research in multicloud storage. In multi-cloud

storage, we first construct the multi-objective optimization challenge of increasing data availability while reducing financial expenditures. Erasure coding, as opposed to data duplication, saves money on storage while also increasing accessibility. Data hosting availability and costs in erasure coding mode are specified using this approach, which has become standard. The multiobjective optimization issue is subsequently solved using a PCA-based NSGA-II approach. Many multi-objective optimization problems may be solved successfully by utilising this approach. Since the final Pareto-optimal set frequently comprises a large number of alternatives, leaving consumers perplexed and unable to make decisions, we apply the PCA approach to identify the best option for each user. When it comes to data placement, the entropy method's placement solutions can meet the user's need for compromise across all objectives. In addition, we contrast our findings with those of two other investigations. Our proposed model's efficacy has been demonstrated by all of the data.

5.3 Cost and Availability Performance

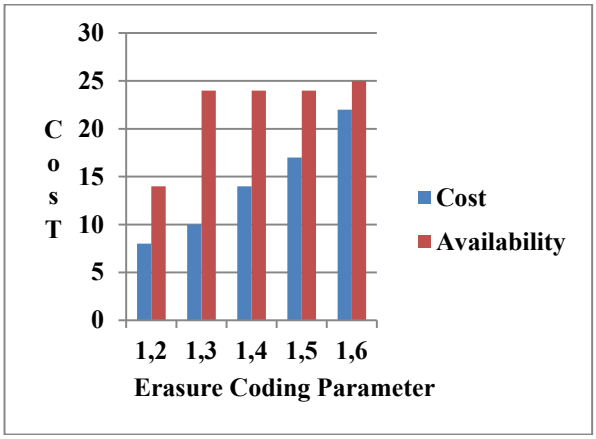

Figure 2. Cost and availability vs. erasure coding.

This section evaluates the suggested algorithms' efficiency in terms of both cost and availability. Using data sizes ranging from 200GB to 1TB and DAF 0:3, we investigate the influence of erasure coding on cost and availability by altering the erasure codes from (1; 2) to (1; 6). As can be seen in Figure 2, the availability approaches 1 as the erasure coding value n grows. For this reason, a data object's total obtainability is based on the chance that not more than (n m) CSPs crash at once. Increasing n makes it possible for the data placement scheme to withstand the failure of many CSPs at once, hence increasing the overall availability. However, the overall cost is larger when n is used. As the number of copies increases, so does the storage cost per data object.

5.4. Comparative Experiments

ACO [28] and CHARM [22] were used to compare the integrated QoS values of the resulting data placement solutions. A is calculated using the formula I = 1f1 (P) + 2f2 (A) (1). The comparative results are shown in Figure 3. Because CHARM's optimization objective is only to minimalize total cost, the proposed method outperforms both CHARM and ACO. Comparatively, the proposed method optimizes not only the total cost, but also the availability of data as well. DAF is varied from 0.01 to 1.20 with a 0.01 step size. A fixed amount of data is available, 2000 GB. Comparing CHARM and the proposed method, when DAF is 0.26 and 1.20, the proposed technique can save about 50% and 55%, respectively.

Two methods are compared with differing data sizes, ranging from 200GB up to 5TB in increments of 200GB. Fixed in 0.001 is the DAF. Figure 4 shows that the proposed technique outperforms both CHARM and ACO in terms of efficiency. When the data size is 5000GB, CHARM's solution costs 180.0$, while the proposed method costs only 90$.

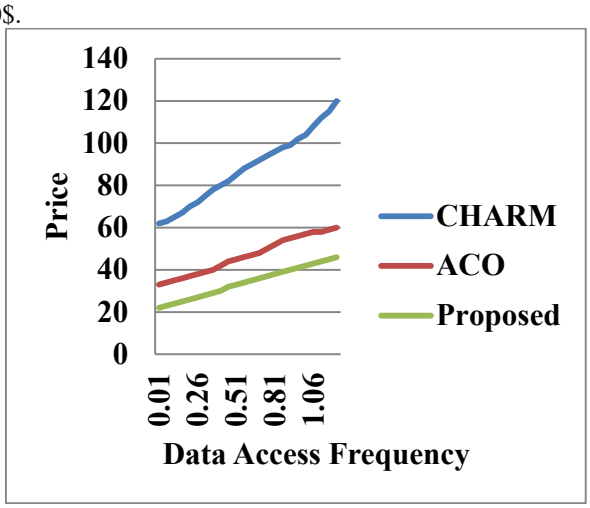

Figure 3: Comparison of costs considering different DAFs.

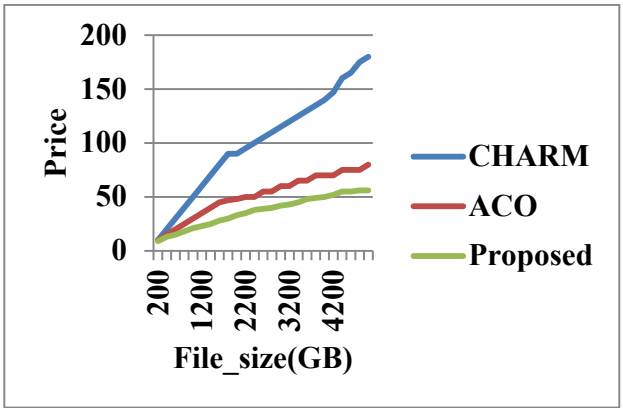

Figure 4: Comparison of costs considering different file sizes.

6. Conclusion

Vendor lock-in, data availability concerns, and privacy leaks are just a few of the risks associated with using cloud data storage. It's a new development trend to use many clouds to house your data. In multi-cloud systems, one of the most difficult difficulties is how to achieve multi-objective optimization while still maintaining a sense of balance between various components. This paper introduces a multi-cloud storage architecture for the first time. The next step is to design a multi-objective optimization problem to reduce overall costs while also increasing data availability. NSGA-II is used to locate non-dominated solutions (i.e., cloud storage providers) as well as erasure-coding parameter values in the paper's second section.. Our PCA approach recommends the best option for clients who cannot pick directly from the Pareto-optimal set. Extensive tests using real-world data from multiple cloud storage providers will be used to assess the algorithm's performance as a last stage in the research process. Our future research will cover a wide range of topics, including: I optimising the type of cloud instance used; and (ii) analysing the data hosting strategy from several angles, including security, latency, and durability.

References

[1] Langmead B, Nellore A. Cloud computing for genomic data analysis and collaboration. Nat Rev Genet 2018;19:208–19.

[2] 1000 Genomes Project Consortium, Abecasis GR, Altshuler D, Brooks LD, Durbin RM, Gibbs RA, et al. A map of human genome variation from population-scale sequencing. Nature 2010;467:1061–73.

[3] Exome Aggregation Consortium, Lek M, Karczewski KJ, Minikel EV, Samocha KE, Banks E, et al. Analysis of protein-coding genetic variation in 60,706 humans. Nature 2016;536:285–91.

[4]Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M., 2010. A view of cloud computing. Communications of the ACM, 53(4), pp.50-58.

[5] Gong, C., Liu, J., Zhang, Q., Chen, H. and Gong, Z., 2010, September. The characteristics of cloud computing. In 2010 39th International Conference on Parallel Processing Workshops (pp. 275-279). IEEE.

[6] Dillon, T., Wu, C. and Chang, E., 2010, April. Cloud computing: issues and challenges. In 2010 24th IEEE international conference on advanced information networking and applications (pp. 27-33). Ieee.

[7]Odun-Ayo, I., Ananya, M., Agono, F. and Goddy-Worlu, R., 2018, July. Cloud computing architecture: A critical analysis. In *2018 18th international conference on computational science and applications (ICCSA)* (pp. 1-7). IEEE.

[8] Clarke, R., 2010, May. User requirements for cloud computing architecture. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* (pp. 625-630). IEEE.

[9]Agrawal, D., El Abbadi, A., Antony, S. and Das, S., 2010, March. Data management challenges in cloud computing infrastructures. In *International Workshop on Databases in Networked Information Systems* (pp. 1-10). Springer, Berlin, Heidelberg.

[10] Zhao, L., Sakr, S., Liu, A. and Bouguettaya, A., 2014. *Cloud data management* (pp. 1-189). Cham, Switzerland: Springer.

[11]Sakr, S., Liu, A., Batista, D.M. and Alomari, M., 2011. A survey of large scale data management approaches in cloud environments. *IEEE Communications Surveys & Tutorials*, *13*(3), pp.311-336.

[12] Agrawal, D., El Abbadi, A., Emekci, F., Metwally, A. and Wang, S., 2011. Secure data management service on cloud computing infrastructures. In *New Frontiers in Information and Software as Services* (pp. 57-80). Springer, Berlin, Heidelberg.

[13]Djebbar, E.I. and Belalem, G., 2016, June. Tasks scheduling and resource allocation for high data management in scientific cloud computing environment. In *International Conference on Mobile, Secure, and Programmable Networking* (pp. 16-27). Springer, Cham.

[14] M. A. Alzain, E. Pardede, B. Soh, and J. A. Thom. Cloud computing security: From single to multi-clouds. In Proceedings of the 45th IEEE Conference on System Sciences, pp. 5490–5499, Hawaii, 2012.

[15] A. Bessani, M. Correia, B. Quaresma, F. Andr, and P. Sousa. Depsky: Dependable and secure storage in a cloud-of-clouds. ACM Transactions on Storage (TOS), 9(4):12, 2013.

[16] R. Rodrigues and B. Liskov. High availability in dhts: Erasure coding vs. replication. In Proceedings of the 2005 International Conference on Peer-To-Peer Systems, pp. 226–239, 2005.

[17] H. Weatherspoon and J. Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In Proceedings of the 2002 International Workshop on Peer-To-Peer Systems, 1:328–338, 2002.

[18] R. Moussa, R. Moussa, M. Swany, and T. Niemi. Erasure codes for increasing the availability of grid data storage. In Proceedings of the 2006 International Conference on Internet and Web Applications and Services, pp. 185-185, Guadelope, French, April 2006.

[19] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng. Cdrm: A costeffective dynamic replication management scheme for cloud storage cluster. In Proceedings of the 2010 IEEE International Conference on CLUSTER Computing, pp. 188–196, Heraklion Greece, October 2010.

[20] Y. Singh, F. Kandah, and W. Zhang. A secured cost-effective multicloud storage in cloud computing. In Proceedings of the 2011 IEEE conference on Computer Communications Workshops, pp. 619–624, Shanghai China, June 2011.
[21] T. G. Papaioannou, N. Bonvin, and K. Aberer. Scalia: An adaptive scheme for efficient multi-cloud storage. In Proceedings of the 2012 International Conference on High PERFORMANCE Computing, Networking, Storage and Analysis, pp. 1–10, Utah USA, November 2012.
[22] Q. Zhang, S. Li, Z. Li, Y. Xing, Z. Yang, and Y. Dai. Charm: A cost-efficient multi-cloud data hosting scheme with high availability. IEEE Transactions on Cloud Computing, 3(3):372–386, 2015.
[23] Abu-Libdeh, H. Princehouse, L. Weatherspoon, H.: RACS: A Case for Cloud Storage Diversity. Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10), New York, NY, USA, 2010, pp. 229-240.
[24] Papaioannou, T. G. Bonvin, N. Aberer, K.: Scalia: An Adaptive Scheme for Efficient Multi-Cloud Storage. Proceedings of the 2012 International Conference on High Performance Computing, Networking, Storage and Analysis (SC '12), Utah, USA, 2012.
[25] Mansouri, Y. Toosi, A. N. Buyya, R.: Brokering Algorithms for Optimizing the Availability and Cost of Cloud Storage Services. Proceedings of 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, UK, 2013, pp. 581-589.
[26] Hadji, M., 2015, May. Scalable and cost-efficient algorithms for reliable and distributed cloud storage. In *International Conference on Cloud Computing and Services Science* (pp. 15-37). Springer, Cham.
[27] Ma, Y. Nandagopal, T. Puttaswamy, K. P. Banerjee, S.: An Ensemble of Replication and Erasure Codes for Cloud File Systems. Proceedings of 2013 INFOCOM, Turin, Italy, 2013, pp. 1276-1284.
[28] Wang, P. Zhao, C. Zhang, Z.: An Ant Colony Algorithm. Based Approach for Cost-Effective Data Hosting with High Availability in Multi-Cloud Environments. Proceedings of 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 2018.
[29] Su, M. Zhang, L. Wu, Y. Chen, K. Li, K.: Systematic Data Placement Optimization in Multi-Cloud Storage for Complex Requirements. IEEE Transactions on Computers, Vol. 65, 2016, No. 6, pp. 1964-1977.
[30] Cloudharmony, 2017. [Online] Available at: http://www.cloudharmony.com.