A Two-Stage Multi-Objective Optimization of Erasure Coding in Overlay Networks

Nishant Saurabh*, Dragi Kimovski*, Francesco Gaetano‡ and Radu Prodan*
Institute of Computer Science, University of Innsbruck, Austria.
Email: *{nishant, dragi, radu}@dps.uibk.ac.at, ‡francesco.gaetano@student.uibk.ac.at

***Abstract*—In the recent years, overlay networks have emerged as a crucial platform for deployment of various distributed applications. Many of these applications rely on data redundancy techniques, such as erasure coding, to achieve higher fault tolerance. However, erasure coding applied in large scale overlay networks entails various overheads in terms of storage, latency and data rebuilding costs. These overheads are largely attributed to the selected erasure coding scheme and the encoded chunk placement in the overlay network. This paper explores a multi-objective optimization approach for identifying appropriate erasure coding schemes and encoded chunk placement in overlay networks. The uniqueness of our approach lies in the consideration of multiple erasure coding objectives such as encoding rate and redundancy factor, with overlay network performance characteristics like storage consumption, latency and system reliability. Our approach enables a variety of tradeoff solutions with respect to these objectives to be identified in the form of a Pareto front. To solve this problem, we propose a novel two stage multi-objective evolutionary algorithm, where the first stage determines the optimal set of encoding schemes, while the second stage optimizes placement of the corresponding encoded data chunks in overlay networks of varying sizes. We study the performance of our method by generating and analyzing the Pareto optimal sets of tradeoff solutions. Experimental results demonstrate that the Pareto optimal set produced by our multi-objective approach includes and even dominates the chunk placements delivered by a related state-of-the-art weighted sum method.**

***Keywords*—Erasure coding, peer-to-peer, overlay network, multi-objective optimization, Pareto optimal set.**

I. Introduction

Overlay networks [2] [18] [16] recently emerged as a crucial platform for deployment of distributed applications, such as file sharing, content distribution, or real-time communication systems. An overlay composed of hosting machines, however, is not as efficient as the dedicated servers and suffers from a resource handicap in terms of storage and bandwidth constraints. Moreover, the non-uniformity of the overlay topology, composed of heterogeneous hosts as overlay nodes, can induce various reliability issues, typically resolved by using data redundancy-based techniques. One such technique is *replication* [8], where a data item is replicated over N overlay nodes at a rate $R \leq N$. While concept of replication allows increased system scalability and reliability at the cost of creating multiple copies of a single item that can induce huge storage costs.

Another alternative redundancy technique for distributed applications in overlay networks is *erasure coding* [19]. Initially used for secured information dispersal [10], erasure coding has been nowadays adopted to enhance fault tolerance without incurring high storage overheads, as in the case of replication. In general, an appropriate erasure coding mechanism applied to a data item of size S is split it into m equally sized chunks, further encoded into n chunks of size b each [1]. The original data item can be then reconstructed from any m out of n chunks, where $1 < m < n$. For example, in a $(16, 64)$ encoding scheme, the data item of size S is initially split into $m = 16$ chunks of equal size. The initial 16 chunks are then used to create additional 48 chunks, such that any 16 out of 64 chunks can be used to retrieve the original data item. Hence, an erasure code with $(16, 64)$ encoding scheme can survive the loss of 48 chunks denoted by its fault tolerance level $k = 48$.

Regardless of the benefits over replication, erasure coding imposes performance overheads in terms of finding the appropriate encoding scheme (m, n) for data items of varying sizes. On one hand, adding more data redundancy to erasure codes enhances the fault tolerance incurring increased storage cost. On the other hand, minimizing redundancy increases the encoding rate in terms of the processing time. Moreover, reducing redundancy increases the data rebuilding cost when the number of lost chunks is approaching the fault tolerance level. Another relevant issue is the systematic placement of the n encoded chunks in the overlay network, since the selection of specific encoding scheme affects its performance characteristics. Increasing the value of n for a specific m in an erasure coding enhances the overlay network reliability at the cost of a high storage consumption at each node in the overlay network. Similarly, a higher value of n induces higher latencies in read and write operations over a data item in the overlay network.

To overcome these barriers, we address in this paper two important optimization problems for storing data items of varying sizes in overlay networks: (1) identifying the appropriate (m, n) encoding scheme for a data item and (2) selecting the optimized placement for n encoded chunks in the overlay network.

To achieve these goals, we designed a corresponding two-stage optimization approach for erasure coded-based storage for data items of varying sizes in peer-to-peer (P2P) overlay networks. The first stage focuses on identifying the appropriate encoding scheme by considering three erasure coding objectives: encoding rate, redundancy factor and rebuilding cost. The second stage selects the optimal chunk placements for the encoded chunks by considering latency, storage consumption and reliability performance characteristics. The central aspect of our method is the use of a multi-objective optimization algorithm that approximates the Pareto optimal set in the three dimensional space of tradeoff encoding scheme and chunk

placement solutions in each stage. We performed an extensive series of experiments to study the benefits of applying our two-stage erasure coded-based multi-objective approach in large overlay networks of varying sizes. Producing and visualizing the Pareto set of tradeoff solutions with a good variety and distribution gives decision makers the flexibility of choosing the "best" encoding scheme and chunk placement that satisfies their storage, reliability or latency requirements with varying fault tolerance levels. Experimental results demonstrate that the Pareto optimal set produced by our multi-objective approach includes and even dominates the chunk placements delivered by a related state-of-the-art weighted sum method [17].

The paper is organized as follows. Section II summarizes the related work. Section III explains the architectural model and formulates the erasure coding objectives in overlay networks. Section IV presents the two stage multi-objective optimization approach for encoding scheme and chunk placement optimization for data items in overlay networks. Section V provides implementation details of the simulated overlay network and optimization algorithms. Section VI presents experimental results and Section VII concludes the paper.

II. Related Work

Recently, erasure codes [4], [7], [9] are increasingly adopted as an alternate to replication, primarily owing to lower storage cost and finer control over redundancy level.

Hakim et al. [20] provide a comparative analysis between erasure coded and replication-based systems. They determine the availability and durability gains in an erasure-resilient system using reduced bandwidth and storage, while maintaining similar mean time to failure as in replication systems.

With respect to overlay systems, George et al. [13] study erasure coding in P2P backup systems, focusing on performance objectives in terms of network utilization, CPU cost, storage overhead and fault tolerance. This study also puts forward the effect of varying encoding schemes (m, n) on the performance objectives.

Although a large number of research has been conducted in erasure coding and some of the corresponding benefits over the replication based overlay systems have been identified, there is no study that focuses on the selection of the appropriate encoding scheme and coded chunk placement in a large scale P2P system. Moreover, due to the varying and conflicting nature of the performance objectives involved in the erasure coded systems, the problem of identifying an appropriate encoding scheme and mapping is aggravated even more.

Recently, Maomeng et al. [17] studied the systematic placement of erasure coded chunks in a multi-Cloud storage system optimizing the fault tolerance, vendor lock-in and access latency using a non-linear programming model. Although this method reports interesting results, it uses a weighted sum optimization method that has limited applicability in large scale environments where setting the proper weights to objectives becomes unclear. Combining tradeoff objectives of different and conflicting nature in a single objective is unnatural resulting in unclear discrepancies between the set of weights and the identified solutions.

To the best of our knowledge, there exist no work that approached the erasure coded-based placement of data items in overlay networks using an evolutionary multi-objective optimization method that approximates the Pareto optimal set of encoding schemes and chunk placement solutions.

III. Model

We present in this section the architectural model of our multi-objective erasure coded storage in overlay networks, together with the corresponding parameter notations and conflicting objective functions.

A. Architectural Background

Our architecture is based upon a structured P2P network overlay. Unlike the unstructured networks, the structured overlay networks use a *distributed hash table (DHT)* for searching of stored data items. For this purpose, DHTs maintain a unique hashing in terms of *(key, value)* pairs enabling joined peer nodes to retrieve data associated to a key.

In this model, the peer nodes and correlated data items, together with the corresponding erasure coded chunks, are represented as a logical tree. The root of this tree is represented by the root peer node, which handles the process of storing or retrieving the data items, while the lower child level peer nodes hold the individual erasure coded chunks. In case of storage, the root peer node applies the erasure coding with a randomly selected (m, n) scheme, such that any m out of n chunks are required to reconstruct the data item, $\forall\ m \in [2, n)$. Consequently, the data chunks are forwarded to the random child peer nodes for storage, such that each distinct peer node has a unique individual chunk of a data item. The peer node representing the root of the logical tree may hold a chunk of the data item depending upon the random placement of chunks.

An essential characteristic of the structured peer-to-peer overlay is that the ownership of the stored data item is shared among all the peer nodes storing the corresponding chunks. Hence, the data retrieval initiated by any peer node in the network overlay proceeds by accessing the minimum number of chunks m required to reconstruct the original data item from the closest peers in the logical tree. The updates or repairs of lost data chunks can also be initiated through any peer node in the network, which is then propagated to the child peer nodes of the logical tree storing the chunks of the data item.

However, inducting an erasure coding-based storage with an (m, n) encoding scheme into a P2P-based overlay system affects the overall performance with respect to the peer nodes where the n chunks are placed. The performance measures are analyzed based on the chunk placing and consider various characteristic objectives, such as storage, reliability, latency, or data rebuilding costs. To properly measure the overall system performance, it is important to identify the objectives which are directly correlated to the selection of encoding (m, n) scheme and the objectives affecting the performance of P2P overlay system based on the distinctive data chunks placing. To this end, we divide our model in two distinctive stages: (1) optimization of the performance objectives affected by varying erasure coding parameters such as m and n, and (2) optimization of the performance objectives affected by the encoded chunk placement over the P2P overlay system.

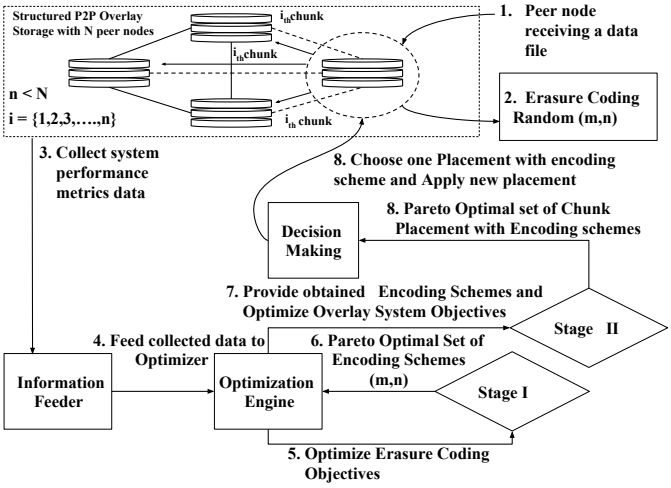


Fig. 1: Multi-objective erasure coding storage architecture in overlay networks.

B. Architectural Model

We present the architectural representation of our model in Figure 1, where the peer node denoted by a dashed circle receives a data item for storage. The circled peer node becomes the root of the logical tree, applies an erasure coding mechanism with a random (m, n) scheme on the data item and computes n data chunks. The n chunks are forwarded to the randomly selected child peer nodes in the overlay. The performance metrics of the overlay system are monitored over the storage life-cycle of the data item by initiating operations over the data item chunks such as read and write requests from any peer node, the availability of the individual peer nodes, the uptime interval of each peer node, and so on. The information feeder collects performance monitoring, which is further supplied to the optimization engine. The optimization engine is divided in two distinctive stages specifically tailored for our model, discussed in detail in Section IV:

1) The first optimization stage computes the Pareto set of *encoding schemes* for the instantiated data item;
2) The second optimization stage computes the Pareto set of *chunks placements* starting from the Pareto set of encoding schemes produced in the first stage.

Once, the placement solution set is computed, a decision making module (manual or automated) chooses an appropriate solution representing the encoding scheme and the chunk placement for the data item. The optimized encoding scheme and the chunk placement is further propagated to the root peer node within the dotted circle shown in Figure 1. Based on the selected solution, the root peer node applies the newly selected encoding scheme and chunk placement identified by the optimization algorithm resulting in a new logical tree.

C. Erasure Coding Parameters in a P2P Overlay Network

The proper definition of the erasure coding parameters essential for correct modeling of its application over the P2P overlay network is defined.

a) Storage peer nodes: We denote by N the number of unique storage peer nodes, physically located at different geographical locations. We base our overlay network upon the assumption that any peer node may join and leave the network at any time.

b) Disk capacity: We denote by $\overrightarrow{EC}$ the disk storage capacity vector, where EC_i represents disk capacity of the i^{th} peer node in the P2P network. In an overlay network, the attached N peer nodes differ in storage characteristics and hence, the number of data items or encoded chunks to be stored depends on the available disk storage at each peer node.

c) Width: We denote by n the width of an erasure coded system, representing the configured number of encoded chunks generated during data item encoding. As the width of an erasure coded system increases, more peer nodes are required in the overlay network. A width factor of $n < N$ results in some peer nodes not having any chunk of a data item stored, while $n > N$ requires attaching more peer nodes or some peer nodes storing more than one chunk. For simplicity, we assume in this paper that $n < N$, where every peer node stores individual chunks of the same data item.

d) Threshold: We denote by m the number of chunks required to reconstruct the original data item that defines the threshold in an erasure coded system, which is a subset of the width ($|m| < |n|$). As the threshold m increases, so does the rebuilding cost in terms of recovery of lost or corrupted chunks corresponding to a data item.

e) Fault tolerance level: We denote by k the fault tolerance level in an erasure coded system, defined as the difference between the width and the threshold ($k = n - m$), where $|k| < |n|$. Failing to retrieve $k+1$ chunks results in the loss of the stored data item.

D. Performance Objectives

We identify in this section the objectives affecting the performance of our P2P overlay network (similar to overlay network objectives in [8]), and model them with respect to the involved erasure coding metrics. The important notations used in this section are listed in Table I.

1) Encoding: represents the amount of processing time required to encode a data item into n chunks by applying specific erasure coding algorithms, achieved at a cost associated to the *encoding rate* ER. The decrease in encoding rate for a data item increases the replication rate of erasure coding and the storage cost. Moreover, a large-sized data item has a higher encoding time making the encoding rate a relevant objective. For every data item of size S, m data blocks must be read to encode it into n data blocks. Hence, we model the encoding rate as in Equation 1, assuming the width factor n to be less than the number of available peer nodes N ($n < N$):

$$ER = \frac{m^2 \cdot b}{S \cdot n}, \ \forall \ m \in [2, n). \tag{1}$$

2) Data redundancy: is described in terms of the replication factor in an erasure coded system, enhancing its fault tolerance at an extra storage cost owing to the size of the data item. As the threshold m approaches 1 ($m \rightarrow 1$), the width$-$threshold k approaches n, which in turn increases

TABLE I: Notation summary.

Notation	*Semantic*
N	Number of peer nodes in overlay network
M	Number of stored data items
S	Size of a data item
n	Number of encoded chunks
m	Minimum number of chunks for reconstructing a data item
b	Size of each encoded chunk
$\left\|\overrightarrow{EC}\right\|$	Overlay network disk storage capacity
ER	Encoding rate
RF	Redundancy factor
RC	Rebuilding cost
SC	Storage consumption
SA	System reliability
L	Latency

the redundancy with enhanced fault tolerance and increased storage overhead. Hence, every data item of size S with m data blocks encoded into n blocks of size b increases the redundancy by factor $\frac{n}{m}$. Considering the involved parameters, the *redundancy factor* denoted by RF can be represented as in Equation 2:

$$RF = \frac{S \cdot n}{m^2 \cdot b}, \ \forall \ m \in [2, n). \tag{2}$$

3) Rebuilding cost: in an erasure coded system is defined as a rate at which corrupted or lost chunks, due to the failure of individual peer nodes in the overlay system, can be recovered. This cost factor is directly proportional to the threshold m. Hence, the minimum number of chunks required to reconstruct the original data holds the information corresponding to lost ones. As the threshold m approaches n ($m \rightarrow n$), the rebuilding cost RC increases, as modeled in Equation 3:

$$RC = \frac{m \cdot \left|\overrightarrow{EC}\right|}{\sum_{i=1}^{N} \frac{u_t(i) - d_t(i)}{N_{down}(i)}}, \ \forall m \in [2, n), \tag{3}$$

where $u_t(i)$ and $d_t(i)$ are the uptime and the downtime of the peer node i, and N_{down} is number of down times per peer node. The numerator in Equation 3 represents m chunks distributed over the total disk capacity $|\overrightarrow{EC}|$ and the denominator represents the total time to failure of all reparable peer nodes in the overlay network.

4) Storage consumption: in a P2P network is bound by an upper limit preventing the storage of data items chunks of varying sizes beyond the existing disk storage capacity at individual peer nodes. In an erasure coded system where every peer node does not store chunks of every data item, the objective is to minimize the storage consumption at each individual peer node. We estimate the storage consumption SC objective for a data item over a P2P network with N peer nodes by adding size of n chunks and dividing it by total disk capacity of peer nodes where the n chunks are placed:

$$SC = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{l=1}^{n} \frac{(1 - E) \cdot b_j}{EC_i}, \tag{4}$$

where $E : [1, N] \times [1, M] \times [1, n] \mapsto [0, 1]$ is a boolean function defined as follows:

$$E = \begin{cases} 0, & l^{th} \text{ chunk of data item j is stored on node i;} \\ 1, & \text{otherwise,} \end{cases} \tag{5}$$

and b_j is the size of the chunks of the data item j and EC is the storage disk capacity at peer node i.

5) System reliability: in a structured P2P network overlay assumes that peer nodes with unique inherent characteristics have the ability to join or leave the network at any time. This feature corresponds to the tendency of individual nodes to fail in large scale systems. In our model, reliability of stored data item chunks is expressed through the failure of k out of n peer nodes hosting the unique chunks of data item. Therefore, the system reliability SA maximization is achieved by minimizing failure of k or more peer nodes hosting n chunks, as defined in Equation 6:

$$SA = \prod_{i=1}^{N} \prod_{j=1}^{M} \prod_{l=1}^{n} E \cdot F(k, n), \tag{6}$$

where E is defined in Equation 5 and $F(k, n)$ represents the probability of failure of k or more peer nodes out of n peer nodes following the binomial distribution[1] [11]:

$$F(k, n) = \sum_{i=0}^{k-1} \binom{n}{i} \cdot a^i \cdot f^{n-i}, \tag{7}$$

where a and f are the availability and failure probability of the i^{th} peer node.

6) Latency: for a data item with n encoded chunks over the P2P network can be particularly high, as performing the read and write requests initiated by the peer nodes in the overlay network pertaining to a data item requires processing of n chunks instead of one sequential data item. Furthermore, it is essential to utilize high bandwidth channels to avoid peer nodes with low bandwidth network connections for chunk storage and retrieval. In order to model latency L, it is imperative to consider the total number of read and write requests for a data item chunk of size b made by every peer node in the overlay network, and divide it by the minimum bandwidth across the network path of source peer node requesting the chunk and destination peer node storing the chunk:

$$L = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{l=1}^{n} (1 - E) \cdot \frac{b_j}{B\left(i, H\left(j, l\right)\right)} \cdot \left(R(i, j) + W\left(i, j\right)\right), \tag{8}$$

where E is defined in Equation 5, $H(j, l)$ is the i^{th} peer node hosting an encoded chunk l of the data item j, $R(i, j)$ are the number of read requests from peer node i for any chunk of data item j, $W(i, j)$ are the number of write requests from peer node i for any chunk of data item j, and $B(i, H(j, l))$ is the minimum bandwidth from the peer node i to the peer node hosting the chunk l of the data item j.

E. Optimization Objectives

While previous research primarily focused on optimizing a single objective, the performance of an erasure coded system in an overlay network depends on multiple objectives. In such cases, the optimization of one objective leads to a degraded performance of the others. Hence, we define in our model the conflicts between the objectives discussed in Section III-D.

[1] http://www.ewp.rpi.edu/hartford/~ernesto/S2008/SMRE/Papers/Kuo-Zuo-koon.pdf

1) Encoding – redundancy – rebuilding: The first three conflicting objectives, encoding rate, redundancy factor, and rebuilding cost, directly correspond to the chosen *(m,n)* erasure coding scheme with minimal overlay characteristics. On one hand, as the threshold factor m approaches the width factor n ($m \rightarrow n$), the encoding rate increases, while the redundancy factor decreases and viceversa. On the other hand, the rebuilding cost is directly proportional to threshold factor m, which is in direct conflict with the redundancy factor.

2) Storage – latency – reliability: The second conflict corresponds to the storage, reliability and latency objectives. As the width n increases with respect to the threshold m, more chunks are required to be stored adding to the storage consumption of the system with enhanced reliability. Furthermore, in the case of latency and reliability conflict, the higher width n means an increase in the read and write costs with respect to storing and propagating data item chunk updates over geographically distributed peer locations.

IV. Two-Stage Multi-Objective Optimization

Multi-objective optimization involves the identification of one or several solutions that optimize specific objective functions within given constraints. In the case when the optimization involves several conflicting objectives, typically results in a set of alternate tradeoff solutions. The real-life multi-objective optimization problems usually have high complexity and involve multiple $q \geq 2$ objective functions $f_1(\overrightarrow{x}), f_2(\overrightarrow{x}), \ldots, f_q(\overrightarrow{x})$ to be minimized or maximized, where $\overrightarrow{x} = (x_1, x_2, \ldots, x_n)$ is a vector over a set of decision variables within a search space X ($\overrightarrow{x} \in X$). Furthermore, a solution $\overrightarrow{x_1}$ is said to dominate another solution $\overrightarrow{x_2}$ if it is better with respect to at least one objective: $f_i(\overrightarrow{x_1}) \leq f_i(\overrightarrow{x_2}), \forall i \in [1,p]$, and $\exists j \in [1,p]$ such that $f_j(\overrightarrow{x_1}) < f_j(\overrightarrow{x_2})$. The set of non-dominated solutions called *Pareto optimal set* in the search space X represents tradeoff values between the objective functions. The Pareto optimal set defines the *Pareto front* of a infinite points of tradeoff solutions.

In this work, we use the NSGA-II [3] multi-objective optimization algorithm to compute the optimal tradeoffs between our target objectives in two optimization stages: the first stage optimizes the encoding scheme with the encoding rate, data redundancy, rebuilding cost as objectives, while the second stage optimizes the encoded chunk placement in the overlay network with storage, reliability and latency conflicts.

A. Stage-1: Encoding Scheme Optimization

Algorithm 1 describes the step-wise representation of the first stage optimization process, which identifies the Pareto optimal set of (m,n) encoding schemes, modeled based on the encoding rate, data redundancy, and rebuilding cost as objective functions. The search space of this optimization is bounded by considering two decision variables: the width n constrained in the range $n \in [3,N)$ and the threshold m in the range $m \in [2,n)$. The decision vector $\overrightarrow{x}$ containing the two variables is represented as a single individual within the population pool. As with any other evolutionary algorithm, the input parameters are the maximal number of individual evaluations $Eval_{max}$ and the population size P. Additionally, the number of peer nodes N is also required by our model.

Algorithm 1: Encoding scheme optimization algorithm.

```
   Input : N;                                   // Number of peer nodes
   Input : Eval_max;                 // Maximal number of evaluations
   Input : P;                                     // Population size
1  Eval ← 0;                      // Initialize number of evaluations
2  X ← ∅;                           // Create empty population set
3  while Eval < |P| do
4      m ← rand(2, n);                  // Generate random threshold
5      n ← rand(3, N);                     // Generate random width
6      x⃗ ← (m, n);     // Generate encoding scheme individual
7      (x⃗, ER, RF, RC) ← evaluate_encoding_scheme(x⃗)
8      X ← X ∪ (x⃗, ER, RF, RC);                  // Add individual
9      Eval ← Eval + 1
10 end
11 while Eval < Eval_max do
12     x⃗ ← crossover(X);  // Crossover 2 random individuals
13     x⃗ ← mutation(x⃗);                  // Mutate new individual
14     (m, n, ER, RF, RC) ← evaluate_encoding_scheme(x⃗)
15     X ← X ∪ (m, n, ER, RF, RC);          // Add new individual
16     Eval ← Eval + 1
17 end
18 X_s ← sort(X);                          // Non-dominated sorting
19 return pareto_set(X_s);                // Return Pareto optimal set
```

We initiate the optimization process by creating an empty population set, where the candidate individuals are stored (line 2). After initialization, we fill the population set with randomly generated (m,n) encoding schemes, evaluations based on the three objective functions (lines 3 – 10). If the maximum number of evaluations is higher than the population size, we select two random individuals for crossover creating a new child solution (line 12), as described in Section V-B. Afterwards, we apply a mutation operator on the created solution with a certain probability (line 13) and merge the newly generated encoding scheme $\overrightarrow{x}$, which is evaluated (line 7) with the population set within the defined constraints (line 15). We repeat this process until the maximal number of evaluations is reached, and eliminate in each iteration the lower quality solutions from the population to improve its quality by accommodating the better newly identified solutions (lines 11 – 17). Finally, we sort all solutions in the population set based on dominance (line 18) and the return the Pareto optimal set of encoding schemes (line 19). As all the attained non-dominated solutions are used as input in the next optimization stage, no decision making is required.

B. Stage-2: Chunk Placement Optimization

The second stage of the optimization process corresponds to the identification of the optimal trade-off placement solutions for n encoded chunks in an overlay network with N peer nodes by simultaneously minimizing the storage consumption, system reliability and latency. In this stage, the decision variables $\overrightarrow{y}$ are represented by the peer nodes where the n encoded chunks can be placed. The i^{th} component of the solution vector $\overrightarrow{y}$ contains the peer node where the i^{th} encoded chunk is mapped with the corresponding encoding scheme.

Algorithm 2 takes as input the Pareto optimal set of encoding schemes obtained in the first stage and the set of N peer nodes in the overlay network. The algorithm then spawns multiple separate optimization tasks, each corresponding to a separate encoding scheme (line 2). Similar as in the first stage, we start all these optimization processes by creating an empty population set Y (line 4), randomly filled with individual

Algorithm 2: Chunk placement optimization algorithm.

```
   Input : EncodingSchemes;                          // Encoding schemes
   Input : N;                                        // Number of peer nodes
   Input : Eval_max;                                 // Maximum number of evaluations
   Input : Population;                               // Population size
1  Z ← ∅;                                            // Create empty population set
2  while t < |EncodingSchemes| do
3      Eval ← 0;                                     // Initialize number of evaluations
4      Y ← ∅;                                        // Create empty population set
5      while Eval < |population| do
6          y⃗ ← rand(chunk_place(n));                 // Place random chunk
7          (m, n, SC, SA, L) ← evaluate_chunk_placement (y⃗)
8          Y ← Y ∪ (y⃗, m, n, SC, SA, L);             // Add individual
9          Eval ← Eval + 1
10     end
11     while Eval < Eval_max do
12         y⃗ ← crossover(Y);                         // Crossover 2 individuals
13         y⃗ ← mutation(y⃗);                          // Mutate new individual
14         (m, n, SC, SA, L) ← evaluate_chunk_placement (y⃗)
15         Y ← Y ∪ (m, n, SC, SA, L);                // Add individual
16         Eval ← Eval + 1
17     end
18     Y_s ← sort(Y);                                // Non-dominated sort
19     Z ← Z ∪ pareto_set(Y_s) ;                     // Add Pareto optimal set
20 end
21 Z_s ← sort(Z);                                    // Non-dominated sort
22 return pareto_set(Z_s);                           // Return Pareto optimal set
```

evaluated chunk placements $\overrightarrow{y}$ (lines 6 – 8). Afterwards, as long as the maximal number of evaluations $Eval_{max}$ is not reached, we generate new solutions using the crossover and mutation operators, evaluated and added to the solution set (lines 11 – 16). We present the implementation of the crossover and mutation operations in Section V-B. When the maximal number of evaluations is reached, we perform a non-domination sort to identify the set of Pareto optimal solutions Y_s. The optimal solutions from every separate optimization process are then merged into a single population set Z (line 22). Finally, we sort the aggregated population set Z based on dominance and present the final Pareto set, containing the encoding scheme and chunk placement, to the decision making entity (lines 21 – 22).

V. Implementation

In this section, we discuss the the essential implementation details of our model with respect to the P2P based overlay network and the multi-objective optimization algorithms.

A. Overlay Network

We simulated the peer-to-peer based overlay network system on top of *Hive2Hive*[2], which is a well known Java-based structured P2P file synchronization and sharing library with DHT support. Since Hive2Hive supports only replication policy, we performed changes to support erasure coding, where every peer node receiving a data item initially applies a random (m, n) scheme and distributes the n chunks to a random set of peer nodes. We further assume every peer node in the overlay network to have a specific storage disk capacity with a fixed number of stored data items of varying size, provisioning the current storage consumption at each peer node. We also defined a scheme to allow every peer node leave and join the network after a random time interval, which is a common scenario in large scale P2P networks. This allowed us to estimate the uptime of a peer node, as well as its failure frequency, thus enabling the approximation of individual availability of each peer node in the overlay network.

[2]http://hive2hive.com/

TABLE II: Simulation setup.

Parameters	*Range*
Storage disk capacity	1 GB to 10 GB
Minimum bandwidth	$128\,\mathrm{kbit\,s^{-1}}$ to $1000\,\mathrm{kbit\,s^{-1}}$
Peer node uptime	20 % to 80 %

We also implemented a policy to enable every peer node in the overlay system make at least m read and n write requests for chunks of a corresponding data item, where an i^{th} peer node making a request to j^{th} peer node receiving a request has the minimum bandwidth along the network path. The minimum bandwidth model provides our simulation with a real networked system-based scenario, where the size of any network packets to be transferred between two peer nodes is determined by dividing the size of the stored data item by the minimum bandwidth along the path of the source peer node to the destination. We collect this information with respect to our simulated P2P-based erasure coded storage model for the data item and feed it into our optimization algorithms.

B. Optimization Framework

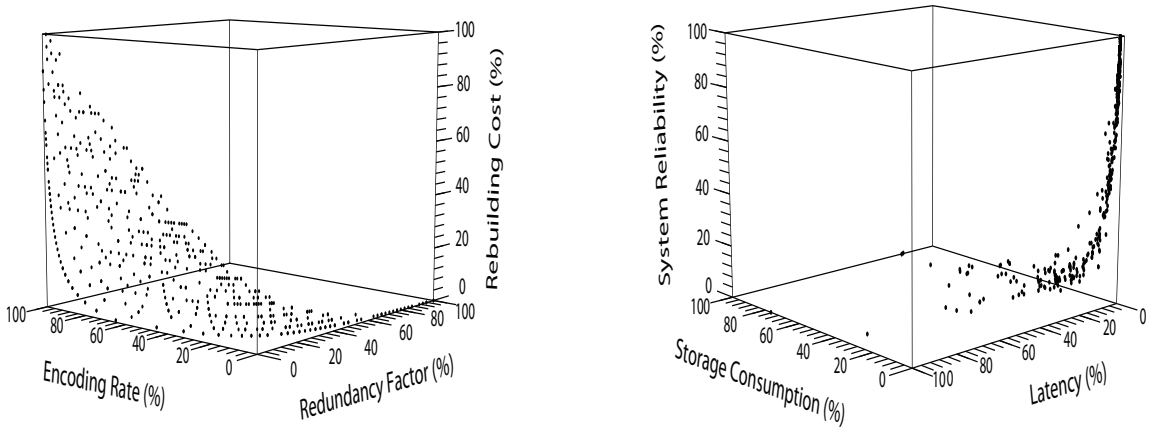
We perform the optimization of the multiple conflicting objectives (Section III-E) modeled as part of the information collected from the simulated overlay system (Section V-A) using the *NSGA-II* [3] multi-objective optimization algorithm. To instantiate NSGA-II in our optimization module, we used the *jMetal* [5] object-oriented Java framework for multi-objective optimization problems.

We implemented particular modifications in the jMetal framework to deal with the specific characteristics of our model. More concretely, we developed new crossover and mutation operators to enable the second optimization stage, as the standard crossover operators of jMetal do not guarantee the correctness of the new solution. For example, a crossover between two placement individuals with a width of $n = 10$ may produce a child with an incorrect number of chunks that will induce a mismatched placement. For this reason, we extended the jMetal to support partially-mapped crossover operations [6]. This crossover operator randomly selects two cut points on both parent individuals. For creating the child placement, the sub-mapping between the two cut points in the first parent replaces the corresponding sub-mapping in the second parent. Afterwards, the inverse replacement is applied outside of the cut-off points, thus eliminating duplicates in the mapping.

Furthermore, we modified the mutation operators included in jMetal by implementing simple bit swap mutation [12] to introduce random perturbations into the search process and diversity in the homogeneous populations. This operator works by simply switching the values between two randomly selected points in the individual.

VI. Experimental Results

In this section, we first present our experimental setup and further analyze the results of our optimization approach



(a) Encoding scheme optimization.

(b) Chunk placement optimization.

Fig. 2: Pareto front representation of the two stages optimization in a 50 peer nodes overlay network.

including comparison with the related work.

A. Experimental Setup

We simulated a P2P storage system, extended to encompass the minimum network bandwidth model and the individual peer availability in P2P networks. In terms of bandwidth and peer node availability, we based our simulation on a recent study enclosing various measurements of real-world P2P networks [15]. Based on this study, we simulated minimum bandwidth between peer nodes in the overlay network using random uniform distribution within the range as listed in Table II. Similarly, we simulated the uptime of each peer node using a random uniform distribution within the range specified in Table II. The uptime range is based on the assumption that the joining and leaving of peer nodes in a large scale P2P networks is a common scenario. We understand that the overlay network simulation with defined model involves inefficacies to that of real world overlay networks. However, in lack of access to the large overlay systems as used in our experimental setup, simulation is the only alternative.

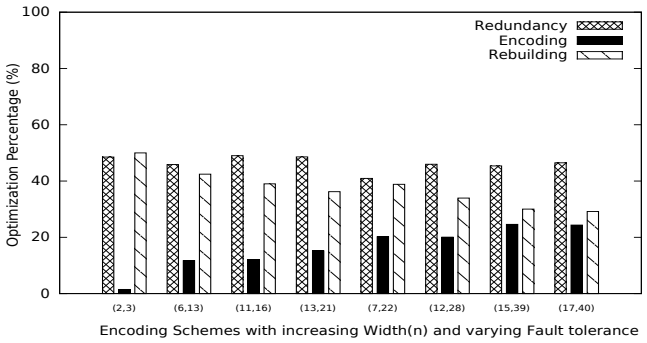
We simulated a structured P2P overlay network with 50, 100 and 200 peer nodes similar to [8] to analyze the scalability of our approach with increasing peer nodes. We presented the implementation details of the overlay network in Section V-A. We conducted our experiments over a single data item, however, the experiments using our approach can be extended to incorporate multiple data items too. Our preliminary analysis on the problem size and the specifics of the model allowed us to identify the optimized input parameters for the NSGA-II algorithm. We selected a population size of 500 individuals and 1000 separate evaluations for each case of $N = 50$, $N = 100$ and $N = 200$ peer nodes in the overlay network for the execution of both optimization stages. We configured both algorithms to use the crossover operator with a probability of 90% and the mutation operator with a probability of $\frac{1}{N}$.

We studied in each experiment the tradeoff between the conflicting objectives at both optimization stages, along with the change in each objective for the various encoding schemes in the final Pareto optimal set.

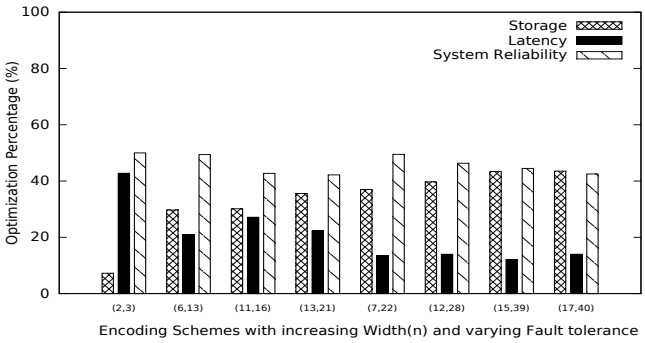
B. Result Analysis

Figure 2a and 2b shows the representation of the Pareto front of the two optimization stages over a P2P overlay network with 50 peer nodes. The axis ranges in both figures are normalized in percentage with respect to the minimum and maximum values obtained over the complete execution. We clearly observe that our approach obtains a number of non-dominated solutions. We obtained similar Pareto fronts for 100 and 200 peer nodes but unable to present them here due to space limitations. Analyzing the Pareto front produced during the first stage, we observe that the solutions are uniformly spread and their convergence is also good. The Pareto front obtained after the second stage shows a good convergence towards lower latency and storage, while offering higher reliability. The spread of the solutions, however, is not uniform and some outliers are present.

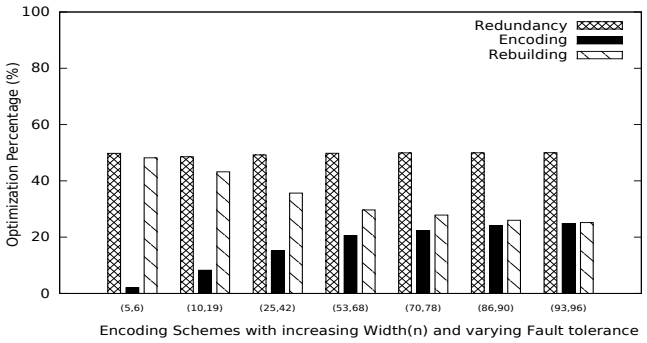
To analyze the variety of obtained solutions, we filtered a representative subset of them with encoding schemes(m, n) having varying fault-tolerance level k with increasing width n as shown in Figures 3a and 3b for a P2P overlay network with 50 nodes. The x-axis represents the selected (m, n) encoding schemes with specific chunk placements and the y-axis the optimization percentage in each objective. We computed this



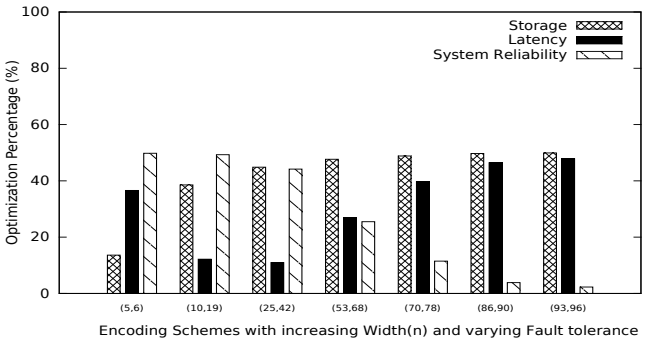
(a) Encoding scheme optimization percentage for 50 peer nodes.



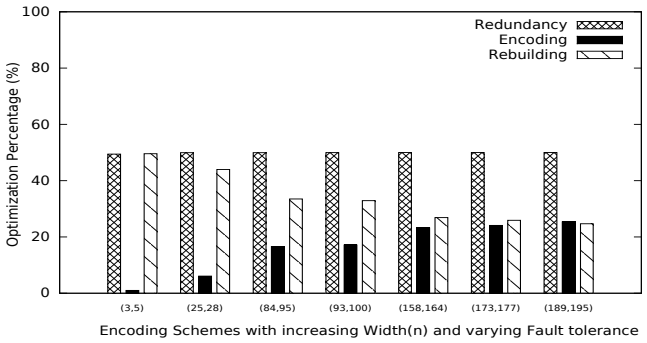
(b) Chunk placement optimization percentage for 50 peer nodes.



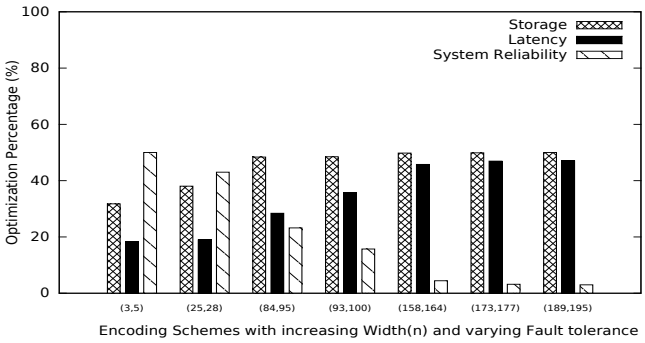
(c) Encoding scheme optimization percentage for 100 peer nodes.



(d) Chunk placement optimization percentage for 100 peer nodes.



(e) Encoding scheme optimization percentage for 200 peer nodes.



(f) Chunk placement optimization percentage for 200 peer nodes.

Fig. 3: Objectives optimization percentage for 50, 100 and 200 peer nodes.

percentage by first normalizing the objectives within their value intervals in the Pareto front, and then computing their optimized percentage with respect to the overall network such that their aggregated value amounts 100%. The higher percentage in the y-axis represents higher quality values for the objectives. One interesting observation in this solution subset is the varying fault tolerance level $k = n - m$, with the minimum value of 1 represented by the $(2, 3)$ encoding scheme, and the maximum value of 24 corresponding to the $(15, 39)$ encoding scheme.

We can draw an important observation with respect to the conflicting objectives at both optimization stages. In the first stage (Figure 3a), the increase in aggregated percentage of the encoding rate and rebuilding cost results in a decrease in the redundancy percentage. This behavior is due to the conflicting characteristic of the redundancy objective towards the encoding rate and rebuilding cost. Similarly in the second stage (Figure 3b), an increase in the aggregated percentage of the storage consumption and latency results in a decrease in system reliability.

Second observation is with respect to the optimization of erasure coding and overlay network performance objectives at distinct stages. In the second stage (Figure 3b), the encoding schemes with maximized system reliability yields minimized redundancy factor for the corresponding optimized encoding schemes (Figure 3a). However, in general, increasing data redundancy, modeled in this paper as redundancy factor, enhances system reliability. This justifies the two stage optimization approach as presented in this paper. Further, supported by identifying that increasing fault-tolerance level k of an

erasure coding system resulted by maximizing redundancy factor need not necessarily reflect enhanced reliability. The argument owes to the chunk placements over peer nodes with varying individual availability in the overlay network as regarded in second stage of the optimization approach.

Another observation with respect to the selection of the appropriate solution in an overlay with 50 peer nodes (Figure 3b) we can draw from the P2P network characteristics. For P2P networks suffering from constant peer node failures, a solution with a $(2, 3)$ encoding scheme guarantees a higher system reliability of up to 50%. However, in order to offer higher fault tolerance while maintaining a similar reliability of 49.4%, a solution with $(6, 13)$ and $(7, 22)$ encoding schemes can be chosen.

Similarly, if the P2P network suffers from high latency, a $(2, 3)$ encoding scheme with optimized placement from the solution subset of 50 peer nodes can be chosen. With respect to storage consumption, if the peer nodes suffer from a lack of storage disk capacity, the solutions with the $(17, 40)$ and $(15, 39)$ encoding schemes can be chosen.

For a limited number of optimal trade-off solutions, using a manual decision making strategy is sufficient. For large overlay networks, however, an automated decision making module, such as in [14], can be applied to select the appropriate trade-off solution. The process of automated decision making is a separate research field and it is out of the scope of this work.

Similar observations, as provided above for a 50 peer nodes solution subset, can be drawn for 100 peer nodes(Figure 3c and 3d) and 200 peer nodes(Figure 3e and 3f).

C. Related Work Comparison

We compare our work with one important related work [17] used in the Triones system that employs a weighted optimization technique by assigning weights to each objective such that their aggregated sum is 1. Since the objectives considered in [17] are different than ours, we adapted it to be comparable to our work. Since the Triones experimental results are limited to two objectives, we customized its approach assuming storage consumption and latency as one objective and the system reliability as the second.

We performed the comparative analysis over an overlay network with 50 peer nodes for the chunk placement optimization objectives only in two scenarios. In the first scenario, we assumed that the system reliability has a weight of 50%, while the storage consumption and latency have an equal aggregated weight of 50%. We executed the weighted sum optimization algorithm multiple times with varying ratios between the storage consumption and the latency weights and sorted the obtained solutions based on dominance. While the Triones approach identified only two distinctive solutions listed in Table III, our chunk placement optimization approach computed almost 500 separate trade-off solutions, including the two identified by Triones. In the second scenario, we assumed three pairs of varying weights for the aggregated storage consumption and latency as first objective and system reliability as the second: $(95\%, 5\%)$, $(75\%, 25\%)$ and $(60\%, 40\%)$,

Table IV shows that both methods identified three solutions with the encoding schemes $(27, 28), (30, 31)$ and $(34, 37)$ for the first set $(95\%, 5\%)$ of optimization weights. For the second set $(75\%, 25\%)$, one solution with the $(16, 17)$ encoding scheme is identified as optimal. Finally for the third set $(60\%, 40\%)$ of optimization weights, one solution with the $(15, 23)$ encoding scheme is represented in the sets provided by both methods. Similar to the first scenario, our method has identified hundreds of more different tradeoff solutions, in part provided in Figure 3. Considering these results, our approach provides much wider range of Pareto tradeoff solutions compared to the fraction of solutions obtained by the Triones weighted sum method. Moreover, our optimized encoding schemes and placement solutions dominate the compared approach in terms of varying fault tolerance levels.

TABLE III: Weighted sum optimization with similar weights.

Encoding Scheme	*Storage + Latency*	*Reliability*
$(2, 3)$	$(7.257 + 42.742) = 50$ %	50 %
$(3, 10)$	$(28.321 + 21.678) = 50$ %	50 %

TABLE IV: Weighted sum optimization with varying weights.

Encoding Scheme	*Storage + Latency*	*Reliability*
$(27, 28)$	$(46.25 + 48.64) = 94.89$ %	5.10 %
$(30, 31)$	$(46.13 + 48.77) = 94.9$ %	5.09 %
$(34, 37)$	$(48.13 + 46.70) = 94.83$ %	5.16 %
$(16, 17)$	$(27.24 + 47.94) = 75.18$ %	24.82 %
$(15, 23)$	$(35.91 + 23.85) = 59.76$ %	40.24 %

D. Overhead analysis

One overhead of our optimization approach is higher execution time complexity of the evolutionary NSGA-II-based algorithm [8]. Table V shows an increase in execution time for our optimization algorithms with varying peer nodes in the overlay network. The execution time can be minimized for the second chunk placement optimization stage by parallelizing the chunk placement optimization of the different encoding schemes obtained in the first optimization stage (i.e. parallelizing the outer `while` loop between lines 2 – 20 in Algorithm 2). However, as the focus of our work is to determine and analyze the quality of different encoding schemes and chunk placements for erasure coding in overlay networks, we leave its parallelization for future work.

VII. Conclusion

Finding the optimal erasure encoding scheme and encoded chunks placement is one of the key problems in overlay-based erasure coded storage. This paper proposes a novel two stage multi-objective optimization approach for identifying the appropriate encoding schemes and the optimized placement of encoded chunks in an overlay network. One of the key strengths of our approach is to distinguish between the erasure coding factors such as encoding rate, redundancy factor and rebuilding cost, which influence the selection of the encoding scheme. Furthermore, it considers other essential overlay factors, such as storage consumption, latency and system reliability affecting the encoded chunks placement. In addition, we view the performance factors as objectives instead of constraints, allowing our approach to search for solutions that optimize the categorical erasure coding and overlay objectives in two distinct stages. Specifically, we apply

TABLE V: Execution time of the optimization algorithm.

Number of Overlay Nodes	*Time in milliseconds*
50	509.696
100	1866.579
200	7230.608

the evolutionary NSGA-II algorithm at each stage to obtain high quality solutions in terms of optimized encoding schemes and corresponding encoded chunk placement. We performed an extensive set of experiments covering multiple overlay network scenarios with varying sizes and studied the optimization results in terms of the variety of the obtained tradeoff solutions. Finally, the Pareto optimal sets of encoding schemes and chunk placements produced by our method include and even dominate the fraction solutions delivered by a related weighted sum optimization method.

ACKNOWLEDGMENT

This work was accomplished as a part of project *ENTICE: "dEcentralised repositories for traNsparent and efficienT vIrtual maChine opErations"*, funded by the European Union Horizon 2020 research and innovation programme under grant agreement No 644179. The authors would also like to thank anonymous reviewers for their valuable comments.

REFERENCES

[1] M. K. Aguilera, R. Janakiraman, and L. Xu. Using erasure codes efficiently for storage in a distributed system. In *2005 International Conference on Dependable Systems and Networks (DSN'05)*, pages 336–345, June 2005.

[2] S. Bauer P. Faratin R. Sami D. Clark, B. Lehr and J. Wroclawski. Overlay networks and future of the internet. In *Communications and Strategies*, volume 63, page 109. ITS Communications and Strategies, 2006.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197, April 2002.

[4] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Inf. Theor.*, 56(9):4539–4551, September 2010.

[5] J. J. Durillo and A. J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011.

[6] D. E. Goldberg and R. Lingle. Alleles, loci, and the traveling salesman problem. In *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, volume 154, pages 154–159. Lawrence Erlbaum, Hillsdale, NJ, 1985.

[7] J. L. Hafner. Weaver codes: Highly fault tolerant erasure codes for storage systems. In *Proceedings of the 4th Conference on USENIX Conference on File and Storage Technologies - Volume 4*, FAST'05, pages 16–16, Berkeley, CA, USA, 2005. USENIX Association.

[8] O. Al-Haj Hassan, L. Ramaswamy, J. A. Miller, K. Rasheed, and E. R. Canfield. Replication in overlay networks: A multi-objective optimization approach. In *Collaborative Computing: Networking, Applications and Worksharing, 4th International Conference, CollaborateCom 2008, Orlando, FL, USA, November 13-16, 2008, Revised Selected Papers*, pages 512–528, 2008.

[9] T. Kanungo Kk. Rao J. L. Hafner, V. Deenadhayalan. Performance metrics for erasure codes in storage systems. 2004.

[10] R. Johnston and Sun il. Kim. Secure distributed storage for information dissemination and retrieval at the tactical edge. In *MILCOM 2015 - 2015 IEEE Military Communications Conference*, pages 61–66, Oct 2015.

[11] kuo Zuo-koon. Chapter 7 : "the k-out-of-n system model".

[12] M. Laumanns, L. Thiele, E. Zitzler, and K. Deb. Archiving with guaranteed convergence and diversity in multi-objective optimization. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, GECCO'02, pages 439–447, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[13] G. Nychis, A. Andreou, D. Chheda, and A. Giamas. Analysis of erasure coding in a peer to peer backup system.

[14] K. Pandiarajan and CK. Babulal. Fuzzy ranking based non-dominated sorting genetic algorithm-ii for network overload alleviation. *Archives of Electrical Engineering*, 63(3):367–384, 2014.

[15] S. Saroiu, K. P. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Multimedia Computing and Networking (MMCN)*, January 2002.

[16] R. K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain. Overlay networks: An akamai perspective. In *In Advanced Content Delivery, Streaming, and Cloud Services*. Citeseer, 2014.

[17] M. Su, L. Zhang, Y. Wu, K. Chen, and K. Li. Systematic data placement optimization in multi-cloud storage for complex requirements. *IEEE Trans. Computers*, 65(6):1964–1977, 2016.

[18] S. Tarkoma. *Overlay Networks: Toward Information Networking.* Auerbach Publications, Boston, MA, USA, 1st edition, 2010.

[19] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-coding based routing for opportunistic networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, WDTN '05, pages 229–236, New York, NY, USA, 2005. ACM.

[20] H. Weatherspoon and J. Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 328–338, London, UK, UK, 2002. Springer-Verlag.