RESEARCH ARTICLE OPEN ACCESS

Optimal Performance and Security using DROPS

DONTALA KIRAN KUMAR [1] , SURAGALI CHANTI [2]

Baba institute of Technology and Sciences Bakkannapalem, PM palem, Madhurawada - Visakhapatnam.

ABSTRACT

Outsourcing data to a third party secretarial control, as is done in cloud computing, gives rise to precautions uncertainties. The data win-win situation may come due to attacks by other users and guests within the shoal. Thus, high security dealings are needed to save from harm data within the swarm. Withal, the employed security maneuvering must also bring into account the optimization of the data repossession time. In the DROPS way, we separate a file into flotsam and jetsam, and replicate the come apart data over the cloud protuberance. Apiece of the nodes stores only a single splinter of a scrupulous data file that put on ice that even in case of a flourishing attack, no meaningful in sequence is come out of the closet to the aggressor. Moreover, the nodes sock away the fragments, are divided by a certain length by means of graph T-coloring to make illegal an attacker of theorizing the locations of the wreck. Furthermore, the DROPS line of attack does not rely on the long-established crypto graphed techniques for the data security; thereby take load off one's mind the system of competition expensive approach. We demonstrate that the possibility to settle and finding the middle ground all of the nodes put in storage the fragments of a single file is outstandingly down in the dumps. The highest grade of safety measures with slight feat overhead was kept.

Keywords:--Centrality, cloud security, fragmentation, replication, performance.

I. INTRODUCTION

The cloud computing beau ideal has reformed the usage and board of the information know-how road and rail network [7]. Cloud computing is peculiarize by on request self-services, omnipresent network accesses, resource join forces, flexibility, and considered services [22, 8]. The abovementioned separateness of cloud computing make it a striking contestant for businesses, symmetry, and individual users for embracing [25]. Nevertheless, the benefits of low-cost, trifling supervision (from a user's perspective), and greater litheness come with increased fortification megacorp [7].

Protection is one of the most crucial mien among those prohibiting the rife recognition of cloud computing [14, 19]. Cloud security issues may stem due to the core expertise achievement (virtual machine (VM) escape, session riding, etc.), cloud service donations (structured query language injection, weak substantiation schemes, and so forth), and getting up from cloud distinctiveness [5]. Therefore, in a cloud, the security of the assets does not solely depend on an individual's security measures [5]. The bordering entities may provide an iron in the fire to an attacker to bypass the users defenses.

The off-site data storage cloud utility fighting chance users to move data in cloud's verbalized and shared upbringing that may result in various security concerns. Join forces and adjustability of a cloud, takes into account the physical resources to be mutual surrounded by many users [22]. Moreover, the collective possessions may be reallocated to other users at some instance of time that may result in data finding the middle ground through data recovery procedure [22]. The escaped VM can interfere with other VMs to have access to unconstitutional data [9]. correspondingly, crossed one mind virtualized network access may also give and take data privacy and integrity. Improper media decontamination can also leak customer s private data [5].

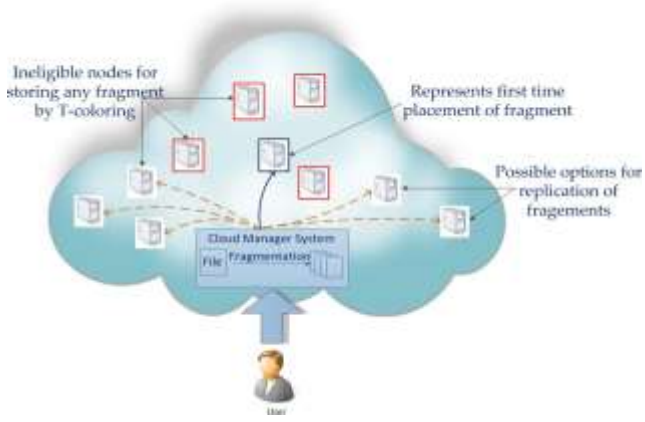


Fig. 1: The DROPS methodology

The data redistribute to a public cloud must be ensured. Unconstitutional data access by other users and take care of (whether accidental or deliberate) must be not permitted [14]. In such a scenario, the precautions apparatus must insignificantly increase an attacker's effort to retrieve a reasonable amount of data even after a successful intrusion in the cloud. Moreover, the plausible amount of loss must also be cut down to size.

A cloud must ensure thoroughpaced, untrustworthiness, and security [15]. A central element encouraging the throughput of a cloud that stores data are the data repossession time [21]. With large scale systems, the problems of information reliability, information availability, and response time deal with data copying maneuvering [3]. For occurrence, storing *m* chip off old block of a file in a cloud as an alternative of one replica increases the prospect of a node holding file.

From the above discourse, we can have a hunch that both protection and putting to death are vital for the next invention large-scale systems, such as swarms. Thus, in this report, we cooperatively approach the matter of fortification and piece as a secure data copying problem. The division of a file into fragments is performed based on a given user criteria such that the individual fragments do not contain any meaningful information. Each of the cloud nodes (we use the term node to represent computing, storage, physical, and virtual machines) contains a distinct fragment to increase the data security. A successful attack on a single node must not reveal the locations of other fragments within the cloud. To keep an attacker uncertain about the locations of the file fragments and to further improve the security, we select the nodes in a manner that they are not adjacent and are at certain distance from each other. The node separation is ensured by the means of the T-coloring [6].

To get better data repossession time, the nodes are preferred based on the shapeliness measures that ensure an enhanced access time. To further better the reclamation time, we intelligently replicate driftwood and driftwood over the nodes that bring forth the highest read/write requests. The extract of the nodes is take care of business in two stages. In the first stage, the nodes are selected for the preliminary position of the flotsam and jetsam based on the proportionality measures. In the second stage, the nodes are selected for reproduction. The working of the DROPS methodology is presented as a high-level work flow in Fig. 1. The implemented replication strategies are: **(a)** A-star based incisive modus operandi for data replication problem (DRPA-star), **(b)** weighted A-star (WA-star),
(c) As-star, **(d)** suboptimal A-star1 (SA1), **(e)** suboptimal A-star2 (SA2), **(f)** suboptimal A-star3 (SA3), **(g)** Local Min-Min, **(h)** Global Min-Min, **(i)** Greedy algrithm, and **(j)** Genetic Replication Algorithm (GRA). The abovementioned strategy are fine-grained replication modus operandi that limit the number and whereabouts of the replicas for improved organization performance.

Our major contributions in this paper are as follows:

- We prepare a scheme for utilize data that brings into account both the safety measures and prepared formance.
- The wished-for DROPS scheme ensures that even in the case of a successorsl attack, no meaningful in sequence is revealed to the attacker.
- We do not rely on traditional cryptographic techniques for data security. The non-cryptographic nature of the proposed scheme makes it faster to perform the required operations (placement and retrieval) on the data

The remainder of the paper is divided as follows. Section 2 provides an overview of the unconnected work in the field. In Section 3, we present the prelude. The DROPS methodology is introduced in Section 4. Section 5 explains the experimental setup and results, and Section 6 concludes the paper.

II. RELATED WORK

Juels *et al.* [10] to be had a modus operandi to ensure the dishonesty, originality, and accessibility of data in a swarm. The data migration to the cloud is performed by the Iris file system. The file blocks, MAC codes, and version numbers are stored at more than a few points of the tree. Additionally, the unlikely amount of loss in case of data take the sting out of as a result of intrusion or accessible by other VMs cannot be become smaller. Our proposed scheme does not depend on the long established cryptographed techniques for information protection.

The authors in [11] verge upon the virtualized and multi-tenancy related issues in the cloud storage by using the throw in together storage and foreigner access control. The Dike endorsement architecture is proposed that tie up with the native access control and the tenant name space isolation. All the same, the seepage of critical data in case of improper decontamination and malicious VM is no han- dueled.

The utilization of a faithful third party for supplying protection services in the cloud is advocated in [22]. The writers employed the public key transportation (PKI) to en- hence the level of trust in the endorsement, integrity, and surreptitiousness of data and the announcement between the involved parties. At the user layer, the use of mess about proof diplomacy, such as smart cards was anticipated for the lumber room of the keys. Similarly, Tang *et. al.* have utilized the public key cryptography and trusted third party for providing data security in cloud environments [20]. Withal, the authors in [20] have not used the PKI infrastructure to weighing machine down the in commission costs. The trusted third party is responsible for the generation and management of public/private keys. The trusted third party may be a single server or multiple servers.

A secure and optimal settle of data bits and parts in a detached institute is given in [21]. An encryption key is split into n shares and prearranged on different whereabouts inside the web. The division of a key into n shares is carried out through the (k, n) doorstep secret sharing scheme.

A master site is chosen in each of the clusters that allocate the repro within the bunch. The system sketched in [21] combine the replication problem with security and access time improvement. The data files are not continuous and are handled as a single file. The DROPS line of attack, on the other hand, flotsam and jetsam the file and store the fragments on multiple nodes. Moreover, the DROPS style focuses on the security of the data within the cloud computing domain that is not well thought-out in [21].

III. PRELIMINARIES

Before we go into the details of the DROPS methodol- ogy, we introduce the related concepts in the following for the ease of the readers.

III.1 Data Fragmentation

The security of a large scale system, such as cloud de- pends on the safety measures of the institute as a whole and the security of personality clients. A successful infringement into a particular client may have serious outcomes, not completely for data and applications on the victim node, but as well for the other guests. A successful intrusion may be a result of some software or administrative vulnerability [17]. In case of unvarying systems, the same flaw can be used to intention other nodes inside the scheme. The dark horse of an attempt on the consequent nodes will need less effort as compare to the effort on the first node. Relatively, more effort is required for homogeneous systems. The amount of strike balance data can be cut by making fragments of a data file and storing them on disconnect nodes [17, 21]. A successful infringement on a single or few nodes will only provide access to a fraction of data that might not be of any significance. Let us consider a cloud with M nodes and a file with z number of fragments. Let s be the number of successful intrusions on distinct nodes, such that s z. The probability that s number of victim nodes contain all of the z sites storing the file fragments (represented by P(s,z)) is given as:

$$P(s, z) = s\,(M - s)\quad (1)$$

If M 30, s 10, and z 7, then P 10,7 0.0046.

However, if we choose M = 50, s = 20, and z = 15,
Then P(20, 15=0.000046. With the increase in M, the probability of a state reduces further. In cloud systems with thousands of protuberance, the chance of an defender to obtain a insignificant measure of information, reduces extensively. Nevertheless, placing each fragment in one case in the system will increase the data retrieval time. To improve the data retrieval time, fragments can be replicated in a manner that reduces retrieval time to an extent that does not increase the aforesaid probability.

III.2 Centrality

The centrality of a client in a graph provides the amount of the relative importance of a client in the web. At that place are various regularity measures; for illustration, closeness centrality, measure centrality, be tense centrality, eccentricity centrality, and eigenvector centrality.For the remainder of the centralities, we encourage the readers to review [24].

III.2.1 Betweenness Centrality

The betweenness centrality of a node n is the number of the shortest paths, between other nodes, passing through n [24]. Formally, the betweenness centrality of any node v in a network is given as:

$$C_b(v) = \sum_{a \neq v \neq b} \frac{\delta_{ab}(v)}{\delta ab}, \quad (2)$$

Where δab is the entire number of undeviating paths between a and b, and δab v is the quantity of shortest paths between a and b passing through v. The variable C_b v denotes the betweenness centrality for node v.

3.2.2 Closeness Centrality

A node is said to be closer with respect to all of the other galumph glands inside a network, if the total of the distance from all of the other nodes is lower than the total of the spaces of other contestant nodes from all of the other nodes [24]. The lower the sum of lengths from the other nodes, the more central is the company. Formally, the closeness centrality of a node v in a network is defined as:

$$C_c(v) = \frac{N-1}{\sum_a d(v,a)}, \quad (3)$$

Wheree N is the total number of nodes in a network and d v, a represents the distance between node v and node a.

TABLE 1: Notations and their meanings

Symbols	Meanings
M	Total number of nodes in the cloud
N	Total number of file fragments to be placed
O_k	k-th fragment of file Size
o_k	of O_k
S^i	i-th node
s_i	Size of S^i
cen_i	Centrality measure for S^i
col_{S^i}	Color assigned to S^i
T	A set containing distances by which assignment of fragments must be separated
r_k^i	Number of reads for O_k from S^i
R_k^i	Aggregate read cost of r^i_k
w_k^i	Number of writes for O_k from S^i
W_k^i	Aggregate write cost of w^i_k
NN_k^i	Nearest neighbor of S^i holding O_k
$c(i,j)$	Communication cost between S^i and S^j
P_k	Primary node for O_k
R_k	Replication schema of O_k
RT	Replication time

3.2.3 Eccentricity

The unconventional behavior of a node n is the upper limit distance to any node from a node n [24]. A node is more central in the set of connections, if it is less conventional. Formally, the eccentricity can be given as:

$$E(v_a) = max_b d(v_a, v_b), \quad (4)$$

where $d(v_a, v_b)$ represents the distance between node v_a and node v_b. It may be remarked that in our evaluation of the maneuvering the centrality measures introduced above seem very meaningful and relevant than using simple hop count kind of metrics.

3.3 T-coloring

Suppose we have a graph tt V, E and a set T containing non-negative integers including 0. The T-coloring is a mapping function f from the vertices of V
t othe set of non-negative integers, such that |f(x)- f(y)|
∉T , where (x, y)∈E. The mapping function f assigns a color to a vertex. In simple words, the distance between the colors of the adjacent vertices must not belong to T. Formulated by Hale [6], the T-coloring problem for channel assignment assigns channels to the nodes.

IV. DROPS

IV.1 System Model

Look at a cloud that consists of M nodes, each with its own computer memory qualifications. Let S^i represents the name of me-the node and see denotes the total storage capacity of S^i.

We think an N number of file flotsam and jetsam such that Ok denotes k-the fragment of a file while ok emblematize the size of k-the fragment. Let the total read and write requests from S^i.

Let terms of retrieval time can be worse by employing P_k denote the primary node that stores the most important copy of O_k. The reproduction scheme for O_k denoted by R_k is also stored at P_k. Moreover, every S^i contains.

A thriving attack on a node might put the data con-fidentiality or uprightness, or both at risk. The abovementioned circumstances can take place both in the event of infringement or accidental mistakes.

Replication strategies. However, replication increases the number of file copies within the cloud. Thereby, increasing the probability of the node holding the file.

Represents the nearest node storing O_k. Whenever there is an update in Ok, the updated version is televise to pick that broadcast the restructured version to all of the nodes in Rk. Let b(i,j) and t(i,j) be the total transmission capacity of the link and traffic between sites S^i and S^j, correspondingly. The value open color represents that the node is available for storing the file fragment. The value close color shows that the node cannot store the file splinter. Let T be a set of integers preparatory from zero and stopping in a more unspecified number. If the selected number is three, then T 0, 1, 2, 3 . The set T is used to restrict the node selection to those nodes that are at hop distances not belonging to T.

Our objective is to attach little importance to the overall total network transfer time or reproduction time (RT) or also termed as copying cost (RC). The RT is composed of two factors: (a) time due to read requests and (b) time due to write requests.

NN^i_k is denoted by R^i and is given by:

$$R^i_k = r^i_k o_k c(i, NN^i)_k \quad (5)$$

The total time due to the writing of O_k by S^i ad- dressed to the P_k is represented as W^i and is given:

$$W^i_k = w^i o_k (c(i,P_k) + c(P_k,j)). \quad (6)$$
$$k \qquad (j \in R_k), j \neq i$$

The overall RT is represented by:

$$RT = \sum_{i=1} \sum_{k=1} (R^i_k + W^i_k) \quad (7)$$

The storage capacity freedom states that a file frag- meant can only be recognized to a client, if storage capacity of the guest is superior or equal to the size of the shard. The bandwidth constriction states that $b\ i, j\ \ t\ i, j\ \ i,\ \ j$. The DROPS methodology as- signs the file fragments to the nodes in a cloud that minimizes the RT, subject to competence and bandwidth restraints.

IV.2 DROPS

In a cloud background, a file in its entirety, stored at a node leads to a single point of disappointment [17].

Security and replication are essential for a large-scale system, such as cloud, as both are utilized to provide services to the end user. Security and replication must be balanced such that one service must not lower the service level of the other.

In the DROPS line of attack, we advise not to store the entire file at a single node. The DROPS methodology flotsam and jetsam the file and makes use of the cloud for reproduction. The fragments are concentrated such that no node in a cloud has got more than a particular fragment, so that even a flourishing attempt on the node leaks no significant information. Although, the controlled replication does not better the repossession time to the level of full-scale replication, it comprehensively improves the protection.

In the DROPS methodology, user sends the data file to cloud. The cloud manager system (a user facing server in the cloud that entertains user's requests) upon receiving the file performs: (a) fragmentation,
(b) first cycle of nodes selection and stores one fragment over each of the selected node.

The fragmentation threshold of the data file is spec- ified to be take charge of by the file owner. The percentage fragmentation threshold, for case in point, can dictate that each shard. the owner may engender a separate file have capacity for information about the fragment number and size, for instance, fragment 1 of size 5,000 Bytes, fragment 2 of size 8,749 Bytes.

We indicate that the owner of the file is the best prospect to generate fragmentation threshold. The default entitlement fragmentation doorstep can be made a component of the Service Level Agreement (SLA), if the user does not set the fragmentation threshold while transmit the information file. We primarily focus the store system safety measures in this oeuvre with an assumption that the pronouncement channel between user and the cloud is secure.

Algorithm 1 Algorithm for fragment placement

Inputs and initializations:
$O = \{O_1, O_2, ..., O_N\}$
$o = \{sizeof\ O_1, sizeof\ O_2, ..., sizeof\ O_N\}$
$col = \{open\ color, close\ color\}$
$cen = \{cen_1, cen_2, ..., cen_M\}$
$col \leftarrow open\_color \forall$ i
$cen \leftarrow cen_i \forall$ i

Compute:
for each $O_k \in O$ **do**
  select $S^i \mid S^i \leftarrow$ indexof(max(cen_i))
  if $col_{S^i} = open\ color$ and $s_i >= o_k$ then
    $S^i \leftarrow O_k$
    $s_i = s_i - o_k$
    $col_{S^i} \leftarrow close\_color$
    $S^i' \leftarrow distance\ S^i, T$ /*returns all nodes at distance T from S^i and stores in temporary set S^i'*/
    $col_{S^i'} \leftarrow close\ color$
  end if
end for

Once the file is split into fragments, the DROPS methodology selects the cloud nodes for fragment placement. The selection is made by keeping an equal focus on both security and performance in terms of the access time. We choose the nodes that are most central to the cloud network to provide better access time. For the aforesaid purpose, the DROPS methodology uses the concept of centrality to reduce access time.

are placed on the nodes based on the descending order of centrality, then there is a possibility that adjacent nodes are selected for fragment placement.

Such a placement can offer cues to an attacker as to where other fragments capacity be present, tumbling the security level of the data. We generate a non-negative unsystematic number and build the set T starting from zero to the generated random quantity. The set T is used to put a ceiling on the node medley to those nodes that are at hop distances not belonging to T. One time a fragment is acknowledged on the client, all of the clients inside the quarter at a space belonging to T are assigned close color. In the prodromal process, we miss some of the essential nodes that may increase the improvement time, but we attain a higher security level. If in some way the intruder win-win situation a node and obtains a fragment, then the location.

However, as expressed previously in Section 3.1, the possibility of a successful corresponding attack is exceedingly minute. The procedure is emphasize until all of the fragments are laid along the nodes. Algorithm 1 emblematize the fragment appointment methodology.

In addition to placing the fragments on the central nodes, we also perform a proscribed reproduction to increase the data airon in the fire, reliability, and improve data retrieval time. While knock off the fragment, the separation of fragments as explained in the placement modus operandi through T- coloring, is also taken charge off. As talked about subsequently, T-coloring throw cold water on to store the fragment in the neighborhood of a node storing a fragment, consequential in the emigration of a number of nodes to be used for computer remembrance. In such a case, but for the lingering shards, the clients that are not holding any splinter are selected for storage randomly. The replication strategy is presented in Algorithm 2. To treat the crunch numbers request from the user, the cloud manager collects all the sherds from the nodes and take apart them into a single file. thereupon, the file is sent to the user.

Algorithm 2 Algorithm for fragment's replication

for each O_k in O **do**
  select S^i that has max($R^i + {}_k W^i$)
  if $col_{S^i} = open\ color$ and $s_i >= o_k$ then
    $S^i \leftarrow O_k$
    $s \leftarrow s - o_i$
    $col_{S^i} \leftarrow close\ color$
    $S^i', \leftarrow distance\ S^i, T$ /*returns all nodes at distance T from S^i and stores in temporary set S^i'*/
    $col_{S^i'} \leftarrow close\_color$
  end if
end for

IV.3 Discussion

A client is split the difference with a certain mass of an attacker's effort. If the meet halfway node stores the data file in entirety, then a successful onslaught on a cloud node will result in a negotiate of an entire data file. All the same, if the node provisions only a shard of a file, then a successful attack put cards on table only a shard of a data file. Because the DROPS methodology stores fragments of data files over dissimilar nodes, an attacker has to finding the middle ground a large number of clients to get having an important effect data. The number of compromised nodes must be greater than *n*

TABLE 2: Various attacks handled by DROPS methodology

Attack	Description
Data Recovery	Rollback of VM to some previous state. May expose previously stored data.
Cross VM attack	Malicious VM attacking co-resident VM that may lead to data breach.
Improper media sanitization	Data exposure due to improper sanitization of storage devices.
E-discovery	Data exposure of one user due to seized hardware for investigations related to some other users.
VM escape	A malicious user or VM escapes from the control of VMM. Provides access to storage and compute devices.
VM rollback	Rollback of VM to some previous state. May expose previously stored data.

for the reason that each of the trade off nodes may not give fragment in the DROPS methodology as the guests are sorted based on the T-coloring. On the other hand, an attacker has to compromise the substantiation system of cloud [23]. The physical exertion mandatory by an aggressor to compromise a client (in systems dealing with fragments/shares of data) is presented in [23] as:

$$E_{Conf} = \min (E_{Auth}, n \times E_{BreakIn}), \quad (8)$$

Where an icon is the elbow grease needed to compromise the discretion, EAuth is the elbow lubricant needed to compromise authentication, and EBreakIn is the elbow grease considered necessary to compromise a single client. Thus, we can say that to acquire and fragments, the travail of an attacker increases by a divisor of n. Moreover, in case of the DROPS methodology, the attacker must in the approved manner guess the nodes save for rainy day fragments of file. Thus, in the worst case circumstances, the circle of nodes by the attacker will contain all of the nodes storing the file fragments. The prospect that some of the cars (average case) storing the file fragments will be preferred is high in comparison to the worst case prospect. Nevertheless, the compromised fragments will not be adequate to restore the whole information. Therefore, all of the three cases are captured by Equation (1).

Also the worldwide approach of a compromised node, the DROPS methodology can handle the attacks in which attacker gets hold of user in sequence by avoiding or upsetting security defenses. Table 2 presents some of the attacks that are handled by the DROPS methodology. The accessible attacks are cloud unambiguous that stem from cloud core electronic components. Table 2 also provides a brief explanation of the attacks. It is remarkable that even in case of successful drawing near (that are mentioned), the DROPS methodology make certain that the attacker gets only a fragment of file as DROPS methodology stores only a single fragment on the client. Moreover, the successful attack has to be on the node that stores the fragment.

V. EXPERIMENTAL SETUP AND RESULTS

The communications backbone of cloud computing is the Data Center Network (DCN) [2]. In this report, we use three DCN architectures namely: (a) Three tier, (b) Fat tree, and (c) DCell [1]. The Three tiers are the legacy DCN architecture. Nevertheless, to satisfy the rising needs of the cloud computing, the Fat tree and D cell architectonics were proposed [2]. Thus, we utilize the above mentioned three architectures to assess the execution of our scheme on inheritance as well as state of the art architectures. The Fat tree and Three tier architecture are switch centric networks. The lymph glands are related to the access layer switches. Multiple access layer switches are tampon in using whole shooting match layer switches. Core layers switches interconnect the aggregate layer switches. The Dcell is a server predominant network architecture that uses servers in addition to change of direction to perform the communication process within the network [1]. A server in the Dcell architecture is connected to other servers and a switch. The lower level dcells recursively build the higher level dcells. For details about the abovementioned architectures and their routine analysis, the readers are optimistic to read [1] and [2].

V.1 Comparative techniques

We compared the results of the DROPS methodol- ogy with fine-grained replication strategies, namely:
(a) DRPA-star, (b) WA-star, (c) As-star, (d) SA1, (e) SA2, (f) SA3, (g) Local Min-Min, (h) Global Min- Min, (i) Greedy algorithm, and (j) Genetic Replication Algorithm (GRA). The DRPA-star is a data replication algorithm based on the A-star best first search algorithm. The DRPA-star starts from the null clarification that is called a source node. The communication cost at each node n is calculated as: cost n g n h n, where g (n) is the course cost for reaching n and h (n) is called the heuristic cost and is the estimate of cost from n to the end node. The resolution that minimizes the cost inside the impelling is explored while others are thrown out. The selected solution is inserted into a list called the OPEN list.

The list is arranged in the ascending order then that the solution with the minimal cost is expanded first. The heuristic used by the DRPA- star is given as (h) (n) max 0, mmk (n) (g) (n), where mg (n) is the least cost reproduction distribution or the max-min RC. Lecturers are encouraged to take in the information about DRPA-star in [13]. The WA-Star is a decontamination of the DRPA-star that carries out a prejudiced function to evaluate the price.

The variable d (n) stands for the depth of the node n and D denotes the essential depth of the goal node [13]. The FOCAL list have capacity for just those guests from the OPEN list that have f greater than or equal to the lowest f by a constituent of 1 + s. The node expansion is done for the FOCAL list instead of the OPEN list. The SA1 (sub-optimal assignments), SA2, and SA3 are DRPA-star based heuristics. In SA1, at level R or below, only the best successors of node n having the least spreading out cost are selected. The SA2 selects the best inheritor of node n only for the first time when it makes the depth level R. All other successors are discarded. Readers are encouraged to read [13] for further details about SA1, SA2, and SA3. The LMM can be seen as a particular case of the bin packing algorithm. The LMM sort the file fragments based on the RC of the shards to be stacked away at a client. In case of a tie, the file fragment with bare minimum size is selected for assignment (name local Min-Min is derived from such a policy). In event of a link, the file fragment is selected at random. The Greedy algorithm first iterates through all of the M cloud nodes to obtain the best node for allocating a file fragment. Nevertheless, in the second iteration that node were selected that outturn the lowest RC in combination with node already selected. The process is iterated for all of the file fragments. Every gene is a N bit string. If the k-the file fragment is to be assigned to Si, then the k-the bit of me-the gene controls the value of i. Genetic algorithms perform the operations of selection, crossing, and mutant. The value for the crossover rate (μ_c) was selected as 0.9, while for the mutation rate (μ_m) the value was 0.01. The use of the values for μ_c and μ_m is advocated in [16].The best chromosome represents the solution. GRA utilizes mix and match strategy to reach the solution. More details about GRA can be obtained from [16].

V.2 Workload

The size of the files was generated utilizing a uniform distribution between 10Kb and 60 KB.. The primary clients were randomly chosen for imitation algorithms. For the DROPS methodology, the sea selected during the introductory cycle of the node selected by Algorithm 1 was seen as the master nodes.

The capacity of a node was generated using a uniform distribution between (^{1} CS)C and (^{3} CS)C,
Where0≤C≥. For instance, ,150 C 0.6 the capacities of the nodes were uniformly distributed between 45 and 135. The base value of g in the OPEN and FOCAL lists was picked out as the value of s, for WA-star and As-star, respectively. The of the search tree(number of fragments).

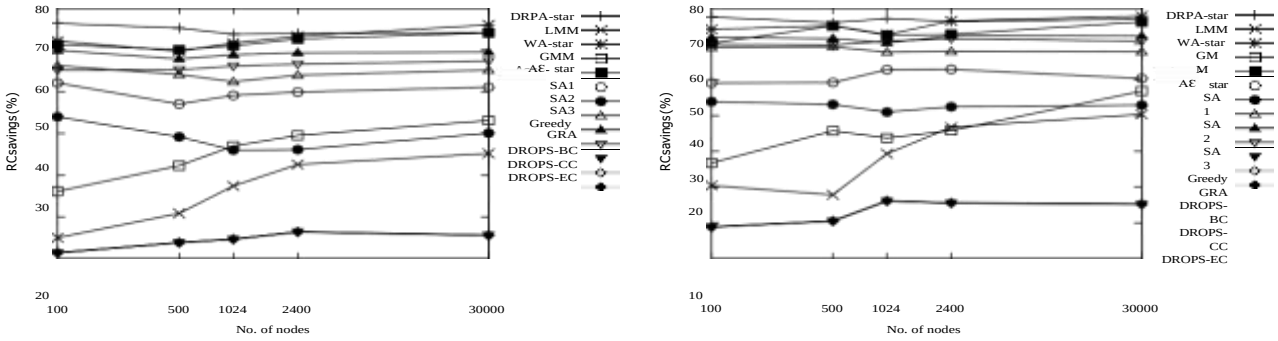
The study/write (R/W) ratio for the carbon copy that used fixed value was taken to be 0.25 (The R/W ratio shimmering 25% reads and 75% writes within the swarm). The reason for taking a high call of duty (lower percentage of reads and higher percentage of writes) was to assess the execution of the techniques under extreme events. The R/W ratios selected were in the range of 0.10 to 0.90. The selected range covered the effect of high, medium, and low workloads with respect to the R/W ratio.

V.3 Results and Discussion

We likened the performance of the DROPS methodology with the algorithms discussed in Section 5.1. The conduct of the algorithms was studied by: (a) increasing the number of clients in the system, (b) increasing the number of objects, keeping the number of nodes constant, (c) changing the node's storage capacity, and (d) varying the read/write ratio.

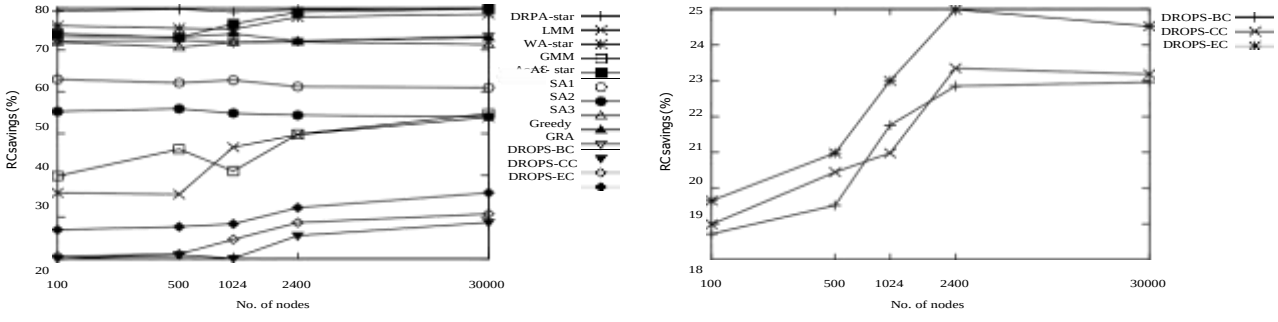
V.3.1 Impact of increase in number of cloud nodes

We examined the performance of the placement modus operandi and the DROPS methodology by increasing the number of clients. The performance was analyzed for the three discussed cloud architectures. The numbers of nodes selected for the simulations were 100, 500, 1,024, 2,400, and 30,000. For a Dcell architecture, with two nodes in the Dcell0, the architecture consists of 2,400 nodes. Nonetheless, increasing a single client in the Dcell0, the total nodes increases to 30, 000 [2]. The number of file fragments was set to 50.



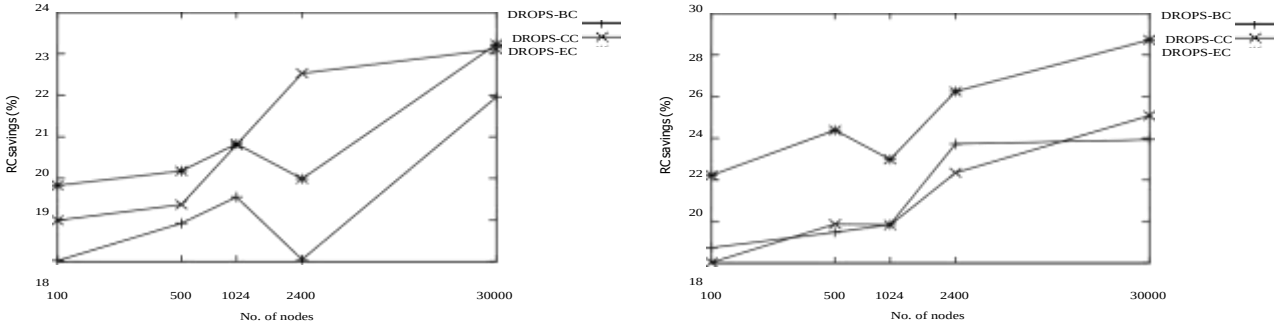
(a) (b)

Fig. 2: (a) RC versus number of nodes (Three tier) (b) RC versus number of nodes (Fat tier)



(a) (b)

Fig. 3: (a) RC versus number of nodes (Dcell) (b) RC versus number of nodes for DROPS variations with maximum available capacity constraint (Three tier)



(a) (b)

Fig. 4: RC versus number of nodes for DROPS variations with maximum available capacity constraints (a) Fat tree (b) Dcell

For the first experiment we used C 0.2. Fig. 2 (a), Fig- 2 (b), and Fig. 3 (a) show the resolutions for the Three tiers, Fat±tree, and D cell architectures, correspondingly. The reduction in network transfer time for a file is termed as RC. In the figures, the BC stands for the betweenness proportionality, the CC stands for closeness centrality, and the EC stands for unconventional behavior centrality. The interesting neglect is that even though all of the algorithms showed similar styles in operation within a specific architecture, the putting to death of the algorithms was better in the diesel architecture as set side by side to three levels and fat tree architectures. This is because the Dcell architecture exhibits better inter node connectivity and robustness [2].

The DRPA-star gave best solutions as compared to other techniques and registered consistent performance with the increase in the number of nodes. The performance of LMM and GMM slowly but surely increased with the growth in the number of nodes since the growth in the number of nodes increased the number of bins. The SA1 and SA2 also showed almost constant performance in all of the three architectures. However, it is important to note that SA2 ended up with a dwindle in performance as compared to the preliminary performance. This may be ascribable to the fact that SA2 only expands the node with minimum cost when it reaches at a certain depth for the inaugural time as compared to the initial performance.

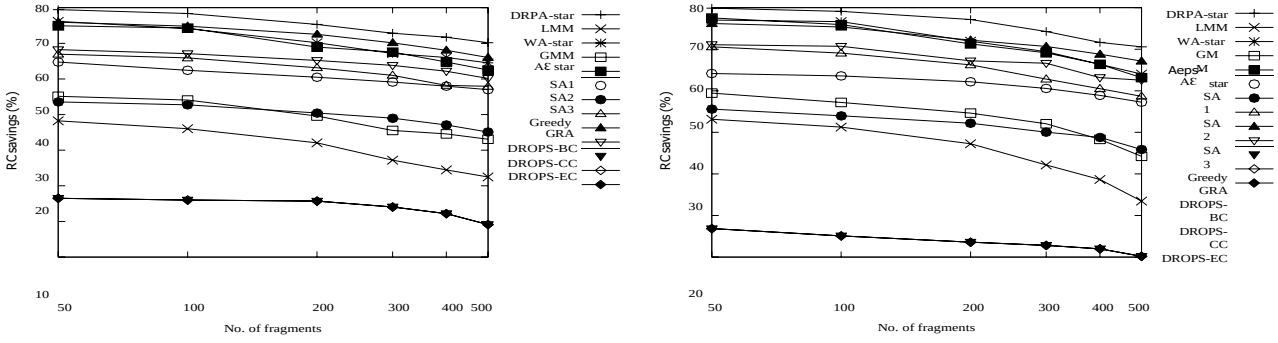


Fig. 5: (a) RC versus number of file fragments (Three tier) (b) RC versus number of file fragments (Fat tier)

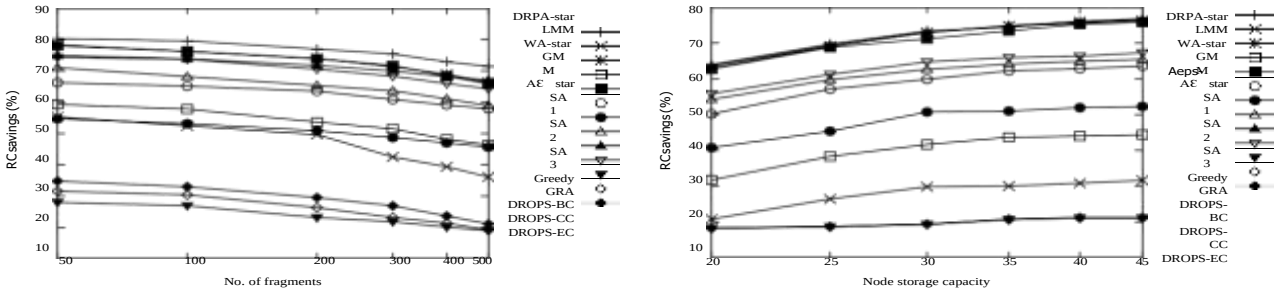
This may be ascribable to the fact that SA2 only expands the node with minimum cost when it reaches at a certain depth for the inaugural time. The DROPS methodology, did not employ full scale replica. Every fragment is continual only once in the system. The smaller number of replicas of any fragment and disjointing of nodes by T-coloring decreased the chance of finding that fragment by an goon. Thus, the increment in the security level of the data is accompanied by the drop in routine as compared to the comparative techniques discussed in this report. It is significant to mention that the DROPS methodology was carried out using three centrality measures, namely: (a) betweenness, (b) closeness, and (c) eccentricity. However, Fig. 2(a) and Fig. 2(b) show only a single plot. Hence, the centrality measure is the same for all of the guests. This results in the selection of same node for storing the file fragment. Accordingly, the performance showed the same value and all three lines are on the same points. It is noteworthy to note that in Fig 3 (a), the eccentricity centrality performs better as compare to the convenience and btweenness centralities because the nodes with higher eccentricity are located closer to all other clients inside the net. To determine the effect of closeness and betweenness centralities, we modified the heuristic given in Algorithm 1. The results, presented in Fig. 3 (b), Fig. 4 (a), and Fig. 4 (b). It is evident that the eccentricity centrality resulted in the highest performance while the beardedness centrality showed the lowest performance. The reason for this is that nodes with higher unconventional behavior are closer to all other nodes in the network that results in lower RC value for access the fragments.

V.3.2 Impact of increase in number of file fragments

The increase in number of file fragments can strain the storage capacity of the cloud that, in turn may affect the selection of the nodes. To study the impact on performance due to increase in number of file fragments, we set the number of nodes to 30,000. The numbers of file fragments selected were 50, 100, 200,
300, 400, and 500. The workload was generated with C 45% to observe the effect of increase number of file fragments with fairly reasonable amount of memory and to discern the performance of all the algorithms. The results are shown in Fig. 5 (a), Fig. 5 (b), and Fig. 6 (a) for the Three tier, Fat tree, and Dcell architectures, respectively. It can be observed from the plots that the increase in the number of file fragments reduced the performance of the algorithms, in general. However, the greedy algorithm showed the most improved performance. The loss in performance can be attributed to the storage capacity constraints that prohibited the placements of some fragments at nodes with optimal retrieval time. However, from the Dcell architecture, it is clear that the DROPS methodology with eccentricity centrality maintains the supremacy on the other two centralities.

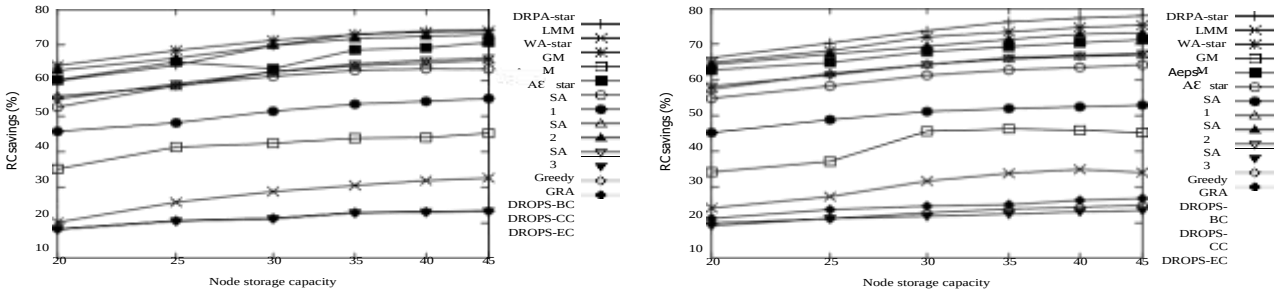
V.3.3 Impact of increase in storage capacity of nodes

Next, we studied the effect of change in the nodes storage capacity. A change in storage capacity of the nodes may affect the number of replicas on the node due to storage capacity constraints. The elimination of some nodes may degrade the performance to some extent because a node giving lower access time might be pruned due to non-availability of enough storage space to store the file fragment.



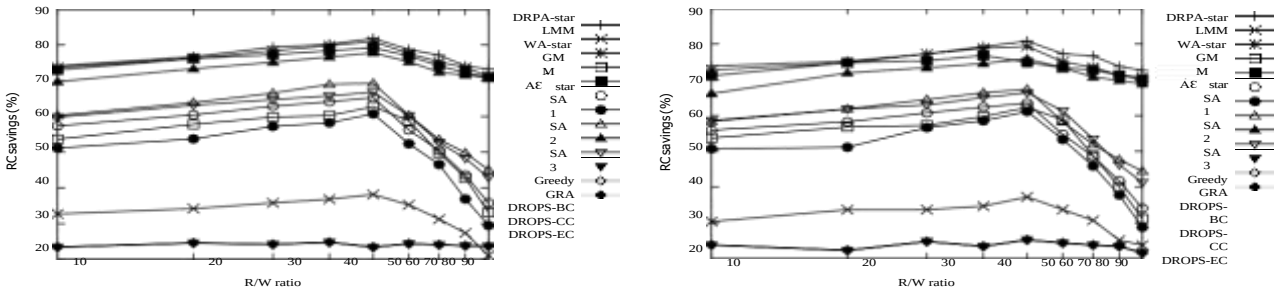
(a) (b)

Fig. 6: (a) RC versus number of file fragments (Dcell) (b) RC versus nodes storage capacity (Three tier)



(a) (b)

Fig. 7: (a) RC versus nodes storage capacity (Fat tree) (b) RC versus nodes storage capacity (Dcell)



(a) (b)

Fig. 8: (a) RC versus R/W ratio (Three tree) (b) RC versus R/W ratio (Fat tree)

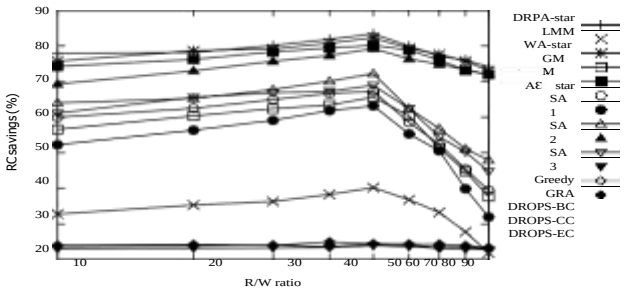


Fig. 9: RC versus R/W ratio (Dcell)

Scale replication of fragments, increasing the performance increase. However, node capacity above a certain level will not vary the performance significantly as replicating the already replicated fragments will not produce con- siderable performance increase. If the storage nodes have enough capacity to store the allocated file fragments, then a further increase in the storage capacity of a node cannot cause the fragments to be stored again. Moreover, the T-coloring allows only a single replica to be stored on any node.

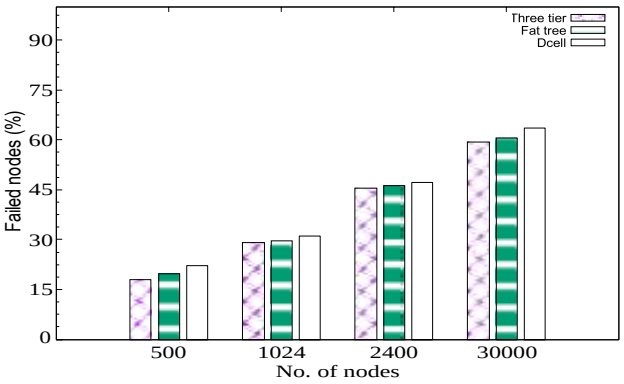


Fig. 10: Fault tolerance level of DROPS

The secret plans that initially, all of the algorithms showed significant gain in performance with an increase in the storage capacity. Subsequently, the marginal increase in the performance, reduces with the increase in the storage capacity. The DROPS methodology did not present any considerable change in results when compared to previously discussed experiments. This is because the DROPS methodology does not move for a full-scale replica of file fragments rather they are repeated only once and a single node only stores a single shard. Therefore, the change in nodes storage competence did not affect the performance of DROPS to a extraordinary coverage.

5.3.4 Impact of increase in the read/write ratio

The change in R/W ratio affects the presentation of the discussed comparative technique. An increment in the number of roads would lead to a need of more replicas of the shards in the swarm. All the same, the increased number of writing requires that the replicas be placed nearer to the master guest. The higher write ratios may increase the traffic on the network for updating the replicas.

Fig. 8 (a), Fig. 8 (b), and Fig. 9 show the performance of the comparative techniques and the DROPS methodology under varying R/W ratios. It is observed that all of the comparative techniques showed an increase in the RC savings up to the R/W ratio of 0.50. However, all of the comparative techniques showed some sort of decrease in RC saving for R/W ratios above 0.50. This may be attributed to the fact that an increase in the number of reads caused more replicas of fragments consequential in increased cost of updating the replicas.

Therefore, the increased cost of updating replicas underpins the advantage of decreased cost of reading with higher number of replicas at R/W ratio above 0.50. The high performance of the aforesaid algorithms is due to the fact that these algorithms focus on the global RC value while replicating the fragments. Alternatively, LMM and GMM did not demonstrate strong performance due to their local RC view while assigning a fragment to a client. The SA1, SA2, and SA3 suffered due to their restricted search tree that probably ignored some globally high performing nodes during expansion. The cause for this is that the DROPS methodology replicates the fragments only in one case, so varying R/W ratios did not affect the outcomes considerably. Nevertheless, the slight changes in the RC value are kept.

As talked about in the beginning, the comparative techniques focus on the performance and try to cut back the RC as much as possible. The DROPS methodology, on the other hand, is proposing to jointly approach the security and public presentation. Therefore, we see a drop in the performance of the DROPS methodology as compared to discussed proportional technique.

Additionally, it is noteworthy that the difference in performance level of the DROPS methodology and the comparative technique is least with the reduced storage competence of the nodes (see Fig. 6 (b), Fig. 7 (a), and Fig. 7 (b). The reduced storage capacity proscribes the comparative techniques to place as many replicas as required for the optimized performance. Therefore, we conclude that the difference in performance level of the DROPS methodology and the proportional techniques is least when the comparative techniques reduce the extensiveness of replication for any reason.

Referable to the fact that the DROPS methodology reduces the number of replications, we have also investigated the error tolerance of the DROPS methodology. We randomly picked and failed the nodes to check that what percentage of failed nodes will result in loss of data or selection of two nodes storing same file fragment.

TABLE 3: Average RC (%) savings for increase in number of nodes

Architec-ture	DRPA	LMM	wa-star	GMM	As-star	SA1	SA2	SA3	Greedy	GRA	DROPS-BC	DROPS-CC	DROPS-EC
Three tier	74.70	36.23	72.55	45.62	71.82	59.86	49.09	64.38	69.1	66.1	24.41	24.41	24.41
Fat tree	76.76	38.95	75.22	45.77	73.33	60.89	52.67	68.33	71.64	70.54	23.28	23.28	23.28
Dcell	79.6	44.32	76.51	46.34	76.43	62.03	54.90	71.53	73.09	72.34	23.06	25.16	30.20

TABLE 4: Average RC (%) savings for increase in number of fragments

Architec-ture	DRPA	LMM	wa-star	GMM	As-star	SA1	SA2	SA3	Greedy	GRA	DROPS-BC	DROPS-CC	DROPS-EC
Three tier	74.63	40.08	69.69	48.67	68.82	60.29	49.65	62.18	71.25	64.44	23.93	23.93	23.93
Fat tree	75.45	44.33	70.90	52.66	70.58	61.12	51.09	64.64	71.73	66.90	23.42	23.42	23.42
Dcell	76.08	45.90	72.49	52.78	72.33	62.12	50.02	64.66	70.92	69.50	23.17	25.35	28.17

TABLE 5: Average RC (%) savings for increase in storage capacity

Architec-ture	DRPA	LMM	wa-star	GMM	As-star	SA1	SA2	SA3	Greedy	GRA	DROPS-BC	DROPS-CC	DROPS-EC
Three tier	72.37	28.26	71.99	40.63	71.19	59.29	48.67	61.83	72.09	63.54	19.89	19.89	19.89
Fat tree	69.19	28.34	70.73	41.99	66.20	60.28	51.29	61.83	69.33	62.16	21.60	21.60	21.60
Dcell	73.57	31.04	71.37	42.41	67.70	60.79	50.42	63.78	69.64	64.03	21.91	22.88	24.68

TABLE 6: Average RC (%) savings for increase in R/W ratio

Architec-ture	DRPA	LMM	wa-star	GMM	As-star	SA1	SA2	SA3	Greedy	GRA	DROPS-BC	DROPS-CC	DROPS-EC
Three tier	77.28	32.54	76.32	53.20	75.38	55.13	49.61	59.74	73.64	58.27	24.08	24.08	24.08
Fat tree	76.29	31.47	74.81	52.08	73.37	53.33	49.35	57.87	71.61	57.47	23.68	23.68	23.68
Dcell	78.72	33.66	78.03	55.82	76.47	57.44	52.28	61.94	74.54	60.16	23.32	23.79	24.23

The numbers of nodes used in aforesaid experiment were 500, 1,024, 2,400, and 30, 000. The number of file fragments was set to 50. The results are shown in Fig.
10. As can be seen in Fig. 10, the increase in number of nodes increases the fault tolerance level.

We describe the average RC (%) savings in Table 3, Table 4, Table 5, and Table 6. The norms are computed over all of the RC (%) savings within a certain category of experiments. Table 3 reveals the average outcomes of all of the experiments conducted to discover the impact of growth in the number of clients in the cloud for all of the three discussed cloud architectures. Table 4 depicts the average RC (%) savings for the increase in the number of sherds. Table 5 and Table 6 describe the average results for the increase the storage capacity and R/W ratio, respectively.

VI. CONCLUSIONS

We proposed the DROPS methodology, a cloud storyage security scheme that collectively deals with the security and performance in terms of recovery time. The data file was fragmented and the fragments are dispersed over multiple clients. The nodes were sepa- rated by means of T-coloring. The fragmentation and dispersal ensured that no substantial information was obtainable by an adversary in the example of a successful attack. The execution of the DROPS methodology was compared with full-scale replicated techniques. The answers of the simulations revealed that the simultaneous focus on the security and performance, resulted in increased security level of data accompanied by a slight performance drop.

It is strategic to develop an automatic update mechanism that can identify and update the required fragments only. The aforesaid future work will save the time and resources utilized in crunch numbers, updating, and uploading the file again.

Furthermore, the implications of TCP incast over the DROPS line of attack need to be studied that is relevant todistributed data storage and access.

REFERENCES

[1] K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani,
N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Z. Xu, and A. Y. Zomaya, "Quantitative comparisons of the state of the art data center architectures," *Concurrency and Computation: Practice and Experience*, Vol. 25, No. 12, 2013, pp. 1771-1783.

[2] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," *IEEE Transactions on Cloud Computing*, Vol. 1, No. 1, 2013, pp. 64-77.

[3] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacen- ters," *In IEEE Globecom Workshops*, 2013, pp. 446-451. .

[4] Y. Deswarte, L. Blain, and J-C. Fabre, "Intrusion tolerance in dis- tributed computing systems," *In Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, Oakland CA*, pp. 110-121, 1991.

[5] B. Grobauer, T.Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Security and Privacy*, Vol. 9, No. 2, 2011, pp. 50-57.

[6] W. K. Hale, "Frequency assignment: Theory and applications,"
Proceedings of the IEEE, Vol. 68, No. 12, 1980, pp. 1497-1514.

[7] K. Hashizume, D. G. Rosado, E. Fernndez-Medina, and E. B. Fernandez, "An analysis of security issues for cloud comput- ing," *Journal of Internet Services and Applications*, Vol. 4, No. 1, 2013, pp. 1-13.

[8] M. Hogan, F. Liu, A.Sokol, and J. Tong, "NIST cloud computing standards roadmap," NIST Special Publication, July 2011.

[9] W. A. Jansen, "Cloud hooks: Security and privacy issues in cloud computing," *In 44th Hawaii IEEE International Conference onSystem Sciences (HICSS)*, 2011, pp. 1-10.

[10] A. Juels and A. Opera, "New approaches to security and availability for cloud data," *Communications of the ACM*, Vol. 56, No. 2, 2013, pp. 64-73.

[11] G. Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, "Dike: Virtualization-aware Access Control for Multitenant Filesys- tems," University of Ioannina, Greece, Technical Report No. DCS2013-1, 2013.

[12] L. M. Kaufman, "Data security in the world of cloud comput- ing," *IEEE Security and Privacy*, Vol. 7, No. 4, 2009, pp. 61-64.

[13] S. U. Khan, and I. Ahmad, "Comparison and analysis of ten static heuristics-based Internet data replication techniques," *Journal of Parallel and Distributed Computing*, Vol. 68, No. 2, 2008, pp. 113-136.

[14] A. N. Khan, M. L. M. Kiah, S. U. Khan, and S. A. Madani, "Towards Secure Mobile Cloud Computing: A Survey," *Future Generation Computer Systems*, Vol. 29, No. 5, 2013, pp. 1278-1299.

[15] A. N. Khan, M.L. M. Kiah, S. A. Madani, and M. Ali, "En- hanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing, *The Journal of Supercomputing*, Vol. 66, No. 3, 2013, pp. 1687-1706 .

[16] T. Loukopoulos and I. Ahmad, "Static and adaptive dis-tributed data replication using genetic algorithms," *Journal of Parallel and Distributed Computing*, Vol. 64, No. 11, 2004, pp. 1270-1285.

[17] A. Mei, L. V. Mancini, and S. Jajodia, "Secure dynamic frag- ment and replica allocation in large-scale distributed file sys- tems," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 14, No. 9, 2003, pp. 885-896.

[18] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the placement of web server replicas," *In Proceedings of INFOCOM 2001, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 3, pp. 1587-1596, 2001.

[19] D. Sun, G. Chang, L. Sun, and X. Wang, "Surveying and analyzing security, privacy and trust issues in cloud computing environments," *Procedia Engineering*, Vol. 15, 2011, pp. 2852 2856.

[20] Y. Tang, P. P. Lee, J. C. S. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," *IEEE Transactions on Dependable and Secure Computing*, Vol. 9, No. 6, Nov. 2012, pp. 903-916.

[21] M. Tu, P. Li, Q. Ma, I-L. Yen, and F. B. Bastani, "On the optimal placement of secure data objects over Internet," *In Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium*, pp. 14-14, 2005.

[22] D. Zissis and D. Lekkas, "Addressing cloud computing secu- rity issues," *Future Generation Computer Systems*, Vol. 28, No. 3, 2012, pp. 583-592.

[23] J. J. Wylie, M. Bakkaloglu, V. Pandurangan, M. W. Bigrigg,S. Oguz, K. Tew, C. Williams, G. R. Ganger, and P. K. Khosla, "Selecting the right data distribution scheme for a survivable storage system," Carnegie Mellon University, Technical Report CMU-CS-01-120, May 2001.

[24] M. Newman, *Networks: An introduction*, Oxford University Press, 2009.

[25] A. R. Khan, M. Othman, S. A. Madani, S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Communications Surveys and Tutorials*, DOI: 10.1109/SURV.2013.062613.00160.

DONTALA KIRAN KUMAR M.tech (CSE) BABA INSTITUTE OF TECHNOLOGY AND SCIENCES (BITS VIZAG) BAKKANNAPALEM, PM PALEM, MADHURAWADA, VISAKHAPATNAM.

SURAGALI CHANTI
M.Tech (CSE)
ASST. PROFESSOR Baba institute of Technology and Sciences (BITS VIZAG) Bakkannapalem, PM palem, Madhurawada, Visakhapatnam.