ISSN: 2366-1313

DRDCOPS: Division and Replication of Data in Cloud for Optimal Performance and Security

BAYYANA MRUDULA, Assistant professor
Dr. SIKHAKOLLI GOPI KRISHNA, professor
CSE Department, Sri Mittapalli College of Engineering, Guntur, Andhra Pradesh-522233

Abstract: As increased Database the Data security and storage of data is very big issue in the database technology to overcome from this, the cloud computing comes in front. User shared the sensitive data over the cloud which gives rise to security issues in cloud computing. Therefore, high security area required protecting datain cloud. In this methodology,when data owner wants to send file on cloud server, it gets splitted into small chunks and for every upload of file a secret file key is also generated .This provides security at client level as well as in network level.Which is used to minimize the total data transfer cost.To achieve reliability, performance, balanced storage capacity and security, fragmentation plays a vital role. Fragmentation is a process which cuts every sensitive file into several fragments in such a way that it is impossible to achieve total file in one try,and for every registered user a secret key is generated so that we can secure our data.we use T-coloring concept for storing the fragments in nodes and For better reliability and performance, resources are replicated at the redundant locations and using redundant infrastructures. Number of data replication methods have been proposed to address an exponential increase in Internet data traffic and optimize energy and bandwidth in datacenter systems.

Keywords: Centrality, Cloud Security, Fragmentation, Replication, Performance.

I. INTRODUCTION

Cloud computing is characterized by on-demand, self-services, network accesses, resource pooling, elasticity, and measured services. The goal of cloud computing is to cut down the cost and allow users to take benefit from all the services provided by the cloud and helps them to focus on their core business.Cloudcomputing associates the computing and storage resources controlled by different operating systems to make available services such as large-scaled data storage and high performance computing to users.The aforementioned characteristics of cloud computing make it a striking candidate for businesses, organizations, and individual users for adoption. The benefits of low-cost, negligible management (from a user's perspective), and greater flexibility come with increased security concerns is one of the most crucial aspects among those prohibiting the wide-spread adoption of cloud computing. The data outsourced to a public cloud must be secured. Unauthorized data access by other users and processes must be prevented. Any weak entity can put the whole cloud at risk. In such a scenario, the security mechanism must substantially increase an attacker's effort to retrieve a reasonable amount of data even after a successful intrusion in the cloud.The division method is used to distribute the data which prevents the system from single point failure situation. This paper also discussed the previous existing systems. This system checks for authorized user,the user is authenticated only by entering a secret key and then user uploads the file. This file is divided into smaller fragments and for each file a secret file key is generated, so dual security is provided.

II. LITERATURE SURVEY

A body of literature has been conducted by several authors and a list of them is given below;

1. Energy-Efficient Data Replication in Cloud Computing Datacenters. Cloud computing is an emerging paradigm that provides computing resources as a service over a network. Communication resources often become a bottleneck in service provisioning for many cloud applications. Therefore, data replication, which brings data closer to data consumers, is seen as a promising solution. It allows minimizing network delays and bandwidth usage. In this paper we study data replication in cloud computing data centers. Unlike other approaches available in the literature, we consider both energy efficiency and bandwidth consumption of the system, in addition to the improved Quality of Service as a result of the reduced communication delays. The evaluation results obtained during extensive simulations help to unveil performance and energy efficiency tradeoffs and guide the design of future data replication solutions.

2. Data Security Issues in Cloud Computing. Cloud computing is an enticing technology which is a combination of many existing technologies such as parallel computing, grid computing, distributed computing and others. It offers services like data storage, computing power, shared resources

ISSN: 2366-1313

at low cost to its users over internet at anytime from anywhere. Costing model on cloud computing is based on pay as you go method; hence companies are saving millions by adopting this technology. As more and more individuals and companies are relying on cloud for their data, the question arises here is how secure cloud environment though cloud computing has many advantages, it also have some security problems.

3. On the Characterization of the Structural Robustness of Data center Networks. A Data Center Network (DCN) constitutes the communicational backbone of a data center, ascertaining the performance boundaries for cloud infrastructure. The DCN needs to be robust to failures and uncertainties to deliver the required Quality of Service (QoS) level and satisfy Service Level Agreement (SLA). In this paper, analyze robustness of the state-of-the-art DCNs. Our major contributions are: (a) we present multi-layered graph modeling of various DCNs; (b) we study the classical robustness metrics considering various failure scenarios to perform a comparative analysis; (c) The present the inadequacy of the classical network robustness metrics to appropriately evaluate the DCN robustness; and (d) The propose new procedures to quantify the DCN robustnessCurrently, there is no detailed study available centering the DCN robustness. Therefore, we believe that this study will lay a firm foundation for the future DCN robustness research. Motivated by the question of access control in cloud storage, we consider the problem using Attribute-Based Encryption (ABE) in a setting where users' credentials may change and cipher may be stored by a third party.

4. Secure Overlay Cloud Storage with Access Control and Assured Deletion This paper describes outsource data backups off-site to third-party cloud storage services so as to reduce data management costs. However, we must provide security guarantees for the outsourced data, which is now maintained by third parties. We design and implement FADE, a secure overlay cloud storage system that achieves fine-grained, policy-based access control and file assured deletion. It associates outsourced files with file access policies, and assuredly deletes files to make them unrecoverable to anyone upon revocations of file access policies. To achieve such security goals, FADE is built upon a set of cryptographic key operations that are self-maintained by a quorum of key managers that are independent of third-party clouds. In particular, FADE acts as an overlay system that works seamlessly atop today's cloud storage services. We implement a proof-of-concept prototype of FADE atop Amazon S3, one of today's cloud storage services.

5. Security and Privacy Issues in Cloud Computing Environment Cloud computing is emerging as a powerful architecture to perform large-scale and complex computing. It extends the information technology (IT) capability by providing on-demand access to computer resources for dedicated use. The information security and privacy are the major concerns over the cloud from user perspective. This paper surveys and evaluates the architecture, data security and privacy issues in cloud computing like data confidentiality, integrity, authentication, trust, service level agreements and regulatory issues. The objective of this paper is to review comprehensively the current challenges of data security and privacy being faced by cloud computing and critically analyze these issues.

6.Dike: Virtualization-aware Access Control for Multi-tenant File systems This paper describes in a virtualization environment that serves multiple customers (or tenants), storage consolidation at the file system level is desirable because it enables data sharing, administration efficiency, and performance optimization .The scalable deployment of file systems in such environments is challenging due to intermediate translation layers required for purposes of networked file access or identity management. Analyzes the security requirements in multitenant file systems then we introduce the Dike authorization architecture, which combines native access control with tenant namespace isolation that is backwards compatible to object-based file systems. We experimentally evaluate a prototype implementation that we developed, and show that our solution incurs limited added performance overhead.

7. Static and adaptive distributed data replication using genetic algorithms Fast dissemination and access of information in large distributed systems, such as the Internet, has become a norm of our daily life. However, undesired long delays experienced by end-users, especially during the peak hours, continue to be a common problem. Replicating some of the objects at multiple sites is one possible solution in decreasing network traffic. The decision of what to replicate where, requires solving a constraint optimization problem which is NP-complete in general. Such problems are known to stretch the capacity of a Genetic Algorithm (GA) to its limits. Nevertheless, we propose a GA to solve the problem when the read/write demands remain static and experimentally prove the superior solution quality obtained compared to an intuitive greedy method. Unfortunately, the static GA approach involves high running time and may not be useful when read/write demands continuously change, as is the case with breaking news. To tackle such case we propose a hybrid GA that takes as input the current replica distribution and computes a new one using knowledge about the network attributes and the changes occurred. Evaluate these algorithms with respect to the storage capacity constraint of each site as well as variations in the popularity of objects, and also examine the trade-off between running time and solution quality.

8. Addressing cloud computing security issues. The recent emergence of cloud computing has drastically altered everyone's perception of infrastructure architectures, software delivery and development models. Projecting as an evolutionary step, following the transition from mainframe computers to client/server deployment models, cloud computing encompasses elements from grid computing, utility computing and autonomic computing, into an

ISSN: 2366-1313

innovative deployment architecture. From a security perspective, a number of unchartered risks and challenges have been introduced from this relocation to the clouds, deteriorating much of the effectiveness of traditional protection mechanisms. As a result the aim of this paper is twofold; firstly to evaluate cloud security by identifying unique security requirements and secondly to attempt to present a viable solution that eliminates these potential threats. This paper proposes introducing a Trusted Third Party, tasked with assuring specific security characteristics within a cloud environment.

9. Comparison and analysis of ten static heuristics-based Internet data replication techniques Compares and analyses 10 heuristics to solve the fine-grained data replication problem over the Internet. In fine-grained replication, frequently accessed data objects (as opposed to the entire website contents) are replicated onto a set of selected sites so as to minimize the average access time perceived by the end users. The paper presents a unified cost model that captures the minimization of the total object transfer cost in the system, which in turn leads to effective utilization of storage space, replica consistency, fault-tolerance, and load-balancing. The set of heuristics include six A-Star based algorithms, two bin packing algorithms, one greedy and one genetic algorithm. The heuristics are extensively simulated and compared using an experimental test-bed that closely mimics the Internet infrastructure and user access patterns. GTITM and Inlet topology generators are used to obtain 80 well-defined network topologies based on flat, link distance, power-law and hierarchical transit–stub models. The user access patterns are derived from real access logs collected at the websites of Soccer World Cup 1998 and NASA Kennedy Space Centre. The heuristics are evaluated by analyzing the communication cost incurred due to object transfers under the variance of server capacity, object size, read access, write access, number of objects and sites. The main benefit of this study is to facilitate readers with the choice of algorithms that guarantee fast or optimal or both types of solutions.

10. Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing. In this paper, to improve the resource limitation of mobile devices, mobile users may utilize cloud-computational and storage services. Although the utilization of the cloud services improves the processing and storage capacity of mobile devices, the migration of confidential information on untrusted cloud raises security and privacy issues. Considering the security of mobile-cloud-computing subscribers' information, a mechanism to authenticate legitimate mobile users in the cloud environment is sought. Usually, the mobile users are authenticated in the cloud environment through digital credential methods, such as password. Once the users' credential information theft occurs, the adversary can use the hacked information for impersonating the mobile user later on. The alarming situation is that the mobile user is unaware about adversary's malicious activities. In this paper, a light-weight security scheme is proposed for mobile user in cloud environment to protect the mobile user's identity with dynamic credentials. The proposed scheme offloads the frequently occurring dynamic credential generation operations on a trusted entity to keep minimum processing burden on the mobile device. To enhance the security and reliability of the scheme, the credential information is updated frequently on the basis of mobile-cloud packets exchange.

III. PROPOSED METHODOLOGY AND DISCUSSION

When data owner wants to send file on cloud server,first the user should register,for each registered user a unique secret key is generated. If all credentials are valid then only the user can send file in cloud. After that file is splitted, Splitting is used to minimize the total data transfer cost .To achieve reliability, performance, balanced storage capacity and security, fragmentation plays a vital role. Fragmentation is a process which cuts every sensitive file into several fragments in such a way that it is impossible to achieve total file in one try. For every upload of file a unique secret file key is also generated, so that we can secure ourdata. The probabilities to find whole fragments are also very low. Thus, this system uses a fragmentation technique by using T-coloring method. Fragmentation is divided into horizontal, vertical and mixed fragmentation. Data replication methodology is very important in today's popular systems for problems such as data reliability, availability and response time. Data replication means keeping a number of replicas on the same server or on dissimilar servers. In replication data is copied and distributed from one database to another. So, it reduces the workload from the original server and the data on the server where it is copied are always active which is not present in mirroring technique. Replication decreases the chance of data loss, increases the performance, availability, reliability. [5].The user can download the file by entering a secret file key,then all the splitted file get merged and can be downloaded

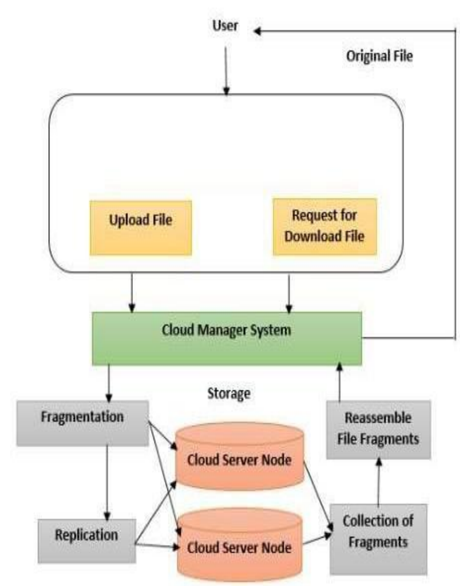

Fig.1. System Architecture.

IV. EXPERIMENTAL SETUP AND RESULTS

The communicational backbone of cloud computing isthe Data Center Network (DCN) [3]. In this paper, weuse three DCN architectures namely: **(a)** Three tier, **(b)**Fat tree, and **(c)** Dcell [2]. The Three tiers is the legacyDCN architecture. However, to meet the growing demandsof the cloud computing, the Fat tree and Dcellarchitectures were proposed [3]. Therefore, we usethe aforementioned three architectures to evaluate theperformance of our scheme on legacy as well as stateof the art architectures. The Fat tree and three tierarchitectures are switch-centric networks. The nodesare connected with the access layer switches. Multipleaccess layer switches are connected using aggregatelayer switches. Core layers switches interconnect theaggregate layerswitches. The Dcell is a server centricnetwork architecture that uses servers in additionto switches to perform the communication processwithin the network [2]. A server in the Dcell architectureis connected to other servers and a switch. Thelower level dcells recursively build the higher leveldcells. The dcells at the same level are fully connected.For details about the aforesaid architectures and theirperformance analysis, the readers are encouraged to read [2] and [3].

A. Comparative techniques

We compared the results of the DROPS methodology with fine-grained replication strategies, namely:**(a)** DRPA-star, **(b)** WA-star, **(c)** Aϵ-star, **(d)** SA1, **(e)**SA2, **(f)** SA3, **(g)** Local Min-Min, **(h)** Global Min-Min, **(i)** Greedy algorithm, and **(j)** Genetic ReplicationAlgorithm (GRA). The DRPA-star is a data replicationalgorithm based on the A-star best-first search algorithm.The DRPA-star starts from the null solutionthat is called a root node. The communication costat each node n is computed as: $cost(n) = g(n) + h(n)$,where $g(n)$ is the path cost for reaching n and $h(n)$ iscalled the heuristic cost and is the estimate of costfrom n to the goal node. The DRPA-star searchesall of the solutions of allocating a fragment to anode. The solution that minimizes the cost withinthe constraints is explored while others are discarded.The selected solution is inserted into a list called the OPEN list. The list is ordered in the ascendingorder so that the solution with the minimum costis expanded first. The heuristic used by the DRPA-staris given as $h(n) = max(0,(mmk(n)g(n)))$, where$mmk(n)$ is the least cost replica allocation or the max-minRC. Readers are encouraged to see the detailsabout DRPA-star. The WA-Star is a refinementof the DRPA-star that implements a weighted functionto evaluate the cost. The function is given as:$f(n) = f(n) + h(n) + \epsilon(1 - (d(n)\sim D)h(n)$. The variabled(n) represents the depth of the node n and D denotesthe expected depth of the goal node. The Aϵ-staris also a variation of the DRPA-star that uses two lists,OPEN and FOCAL. The FOCAL list contains onlythose nodes from the OPEN list that have f greaterthan or equal to the lowest f by a factor of $1 + \epsilon$.The node expansion is performed from the FOCAL listinstead of the OPEN list. Further details about WA-Starand Aϵ-star can be found. The SA1 (suboptimalassignments), SA2, and SA3 are DRPA-starbased heuristics. In SA1, at level R or below, only thebest successors of node n having the least expansioncosts are selected.

The SA2 selects the best successorsof node n only for the first time when it reachesthe depth level R. All other successors are discarded.The SA3 works similar to the SA2, except that thenodes are removed from OPEN list except the onewith the lowest cost. Readers are encouraged to read for further details about SA1, SA2, and SA3. TheLMM can be considered as a special case of the binpacking algorithm. The LMM sorts the file fragmentsbased on the RC of the fragments to be stored at anode. The LMM then assigns the fragments in theascending order. In case of a tie, the file fragmentwith minimum size is selected for assignment (namelocal Min-Min is derived from such a policy). TheGMM selects the file fragment with global minimumof all the RC associated with a file fragment. In caseof a tie, the file fragment is selected at random. TheGreedy algorithm first iterates through all of the Mcloud nodes to find the best node for allocating afile fragment. The node with the lowest replicationcost is selected. The second node for the fragmentis selected in the second iteration. However, in thesecond iteration that node is selected that producesthe lowest RC in combination with node alreadyselected. The process is repeated for all of the filefragments. Details of the greedy algorithm can befound. The GRA consists of chromosomes representingvarious schemes for storing file fragmentsover cloud nodes. Every chromosome consists of Mgenes, each representing a node. Every gene is an$N$bit string. If the k-th file fragment is to be assignedto Si, then the k-th bit of i-th gene holds the valueof one. Genetic algorithms perform the operations ofselection, crossover, and mutation. The value for thecrossover rate (μ_c) was selected as 0.9, while for themutation rate (μ_m) the value was 0.01. The use of the values

B. Workload

The sizes of files were generated using a uniform distribution between 10Kb and 60 Kb. The primary nodeswere randomly selected for replication algorithms. Forthe DROPS methodology, the S^i's selected during thefirst cycle of the nodes selection by Algorithm 1 wereconsidered as the primary nodes. The read/write (R/W) ratio for the simulationsthat used fixed value was selected to be 0.25 (TheR/W ratio reflecting 25% reads and 75% writes withinthe cloud). The reason for choosing a high workload(lower percentage of reads and higher percentageof writes) was to evaluate the performance of thetechniques under extreme cases. The simulations thatstudied the impact of change in the R/W ratio usedvarious workloads in terms of R/W ratios. The R/Wratios selected were in the range of 0.10 to 0.90. Theselected range covered the effect of high, medium, andlow workloads with respect to the R/W ratio.

C. Results and Discussion

We compared the performance of the DROPS methodology with the algorithms discussed in Section 3.1.The behavior of the algorithms was studied by: **(a)**increasing the number of nodes in the system, **(b)**increasing the number of objects keeping numberof nodes constant, **(c)** changing the nodes storagecapacity, and **(d)** varying the read/write ratio.

ISSN: 2366-1313

Theaforesaid parameters are significant as they affect theproblem size and the performance of algorithms.

1. Impact of increase in number of cloud nodes

We studied the performance of the placement techniques and the DROPS methodology by increasing thenumber of nodes. The performance was studied forthe three discussed cloud architectures. The numbersof nodes selected for the simulations were 100, 500,1,024, 2,400, and 30,000. The number of nodes inthe Dcell architecture increases exponentially [3]. ForDcell architecture, with two nodes in the Dcell0,the architecture consists of 2,400 nodes. However,increasing a single node in the Dcell0, the total nodesincreases to 30, 000 [3]. The number of file fragments

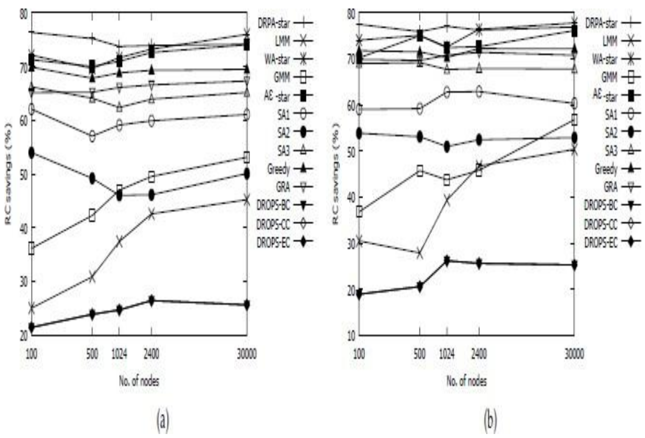

Fig.1. (a) RC versus number of notes (Three tier) (b) RC versus number of nodes (Fat tier)

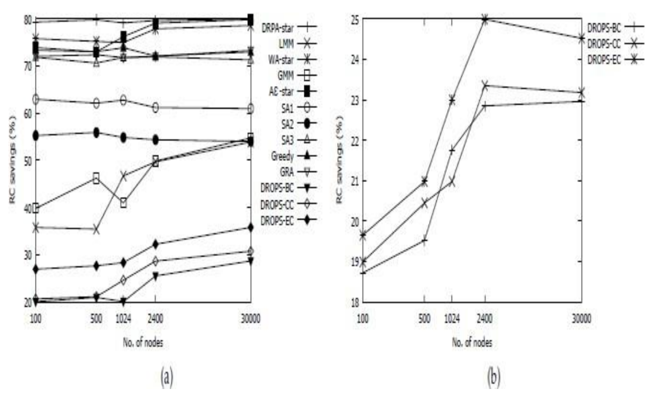

Fig.2. (a) RC versus number of nodes (Dcell) (b) RC versus number of nodes for DROPS variations with maximum available capacity constraint (Three tier)

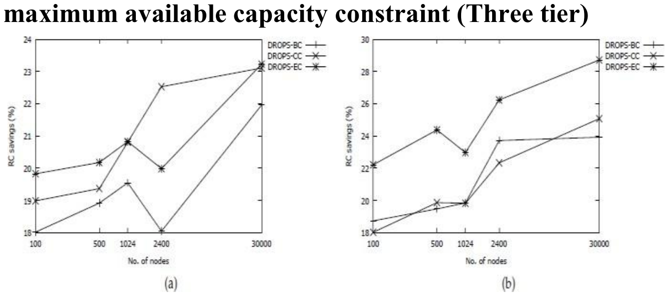

Fig.3. RC versus number of nodes for DROPS variations with maximum available capacity constraints (a) Fattree (b) Dcell

Was set to 50. For the first experiment we used$C$ = 0.2. Fig. 1 (a), Fig. 1 (b), and Fig. 2 (a) showthe results for the three tier, Fat tree, and Dcellarchitectures, respectively. The reduction in networktransfer time for a file is termed as RC. In the figures,the BC stands for the between's centrality, the CCstands for closeness centrality, and the EC stands foreccentricity centrality,the performance of the algorithms was better in theDcell architecture as compared to three tier and fattree architectures. This is because the Dcell architecture exhibits better inter node connectivity and robustness [3]. The DRPA-star gave best solutions as compared to other techniques and registered consistent performance with the increase in the numberof nodes. Similarly, WA-star, Aϵ-star, GRA, greedy,and SA3 showed almost consistent performance withvarious numbers of nodes. The performance of LMMand GMM gradually increased with the increase innumber of nodes since the increase in the number ofnodes increased the number of bins. The SA1 and SA2also showed almost constant performance in all of thethree architectures. However, it is important to notethat SA2 ended up with a decrease in performance.

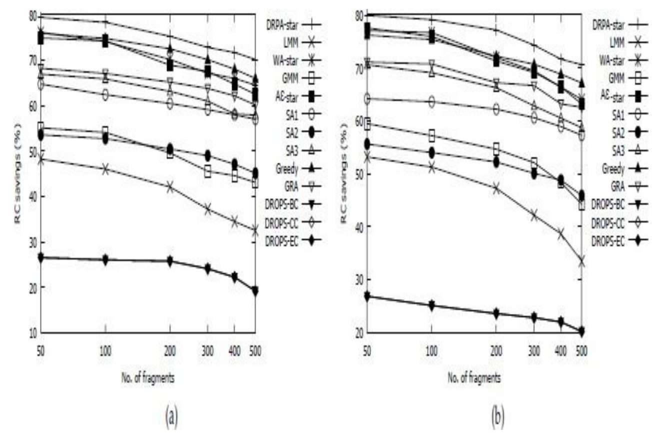

Fig.4. (a) RC versus number of file fragments (Three tier) (b) RC versus number of file fragments (Fat tier)

As compared to the initial performance. This may bedue to the fact that SA2 only expands the node withminimum cost when it reaches at certain depth forthe first time. Such a pruning for the first time mighthave purged nodes by providing better global accesstime. The DROPS methodology did not employ fullscalereplication. Every fragment is replicated onlyonce in the system. The smaller number of replicas ofany fragment and separation of nodes by T-coloringdecreased the probability of finding that fragment byan attacker. Therefore, the increase in the securitylevel of the data is accompanied by the drop inperformance as compared to the comparative techniquesdiscussed in this paper. It is important to notethat the DROPS methodology was implemented usingthree centrality measures namely: **(a)** between's, **(b)**closeness, and **(c)** eccentricity. However, Fig. 1(a) andFig. 1(b) show only a single plot. Due to the inherentstructure of the threetiers and Fat tree architectures,all of the nodes in the network are at the samedistance from each other or exist at the same level.Therefore, the centrality measure is the same for all ofthe nodes.

ISSN: 2366-1313

This results in the selection of same nodefor storing the file fragment. Consequently, the performanceshowed the same value and all three linesare on the same points. However, this is not the casefor the Dcell architecture. In the Dcell architecture,nodes have different centrality measures resulting inthe selection of different nodes. It is noteworthy to mention that in Fig.2(a), the eccentricity centralityperforms better as compared to the closeness and between'scentralities because the nodes with highereccentricity are located closer to all other nodes withinthe network. To check the effect of closeness andbetween's centralities, we modified the heuristicpresented in Algorithm 1. Instead of selecting thenode with criteria of only maximum centrality, weselected the node with: **(a)** maximum centrality and**(b)** maximum available storage capacity. The resultsare presented in Fig. 2 (b), Fig. 3 (a), and Fig. 3 (b). It isevident that the eccentricity centrality resulted in thehighest performance while the between's centralityshowed the lowest performance. The reason for thisis that nodes with higher eccentricity are closer to allother nodes in the network that results in lower RCvalue for accessing the fragments.

2. Impact of increase in number of file fragments

The increase in number of file fragments can strainthe storage capacity of the cloud that, in turn mayaffect the selection of the nodes. To study the impacton performance due to increase in number of filefragments, we set the number of nodes to 30,000. Thenumbers 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 numberof file fragments with fairly reasonable amount ofmemory and to discern the performance of all thealgorithms. The results are shown in Fig. 4 (a), Fig.4 (b), and Fig.5 (a) for the three tier, Fat tree, and Dcell architectures, respectively. It can be observed from theplots that the increase in the number of file fragmentsreduced the performance of the algorithms, in general.However, the greedy algorithm showed the mostimproved performance. The LMM showed the highestloss in performance that is little above 16%. The loss inperformance can be attributed to the storage capacityconstraints that prohibited the placements of somefragments at nodes with optimal retrieval time.

3. Impact of increase in storage capacity of nodes

A change in storage capacity of the nodes may affect the number of replicas on the node due to storage capacity constraints. Intuitively, a lower node storage capacity may result in the elimination of some optimal nodes to be selected for replication because of violation of 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.Higher node storage capacity allows full-scale replicationof fragments, increasing the performance gain.However, node capacity above certain level will notchange the performance significantly as replicatingthe already replicated fragments will not produce considerableperformance increase. If the storage nodeshave 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.

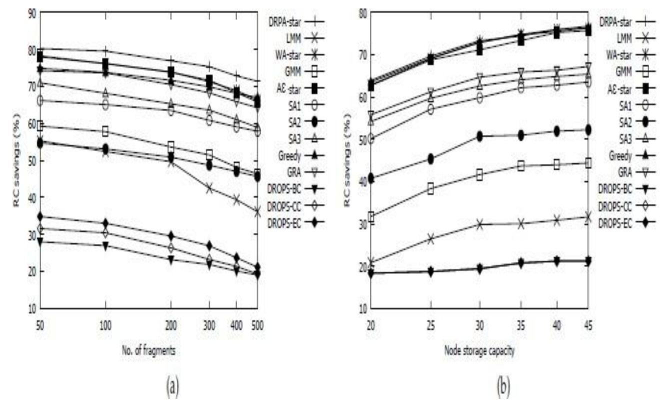

Fig.5. (a) RC versus number of file fragments (Dcell) (b) RC versus nodes storage capacity (Three tier)

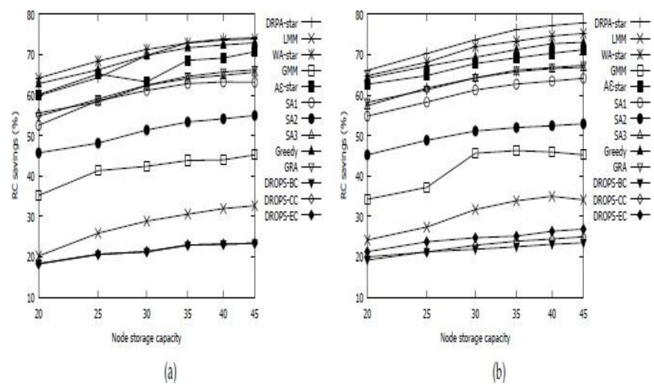

Fig.6. (a) RC versus nodes storage capacity (Fat tree) (b) RC versus nodes storage capacity (Dcell)

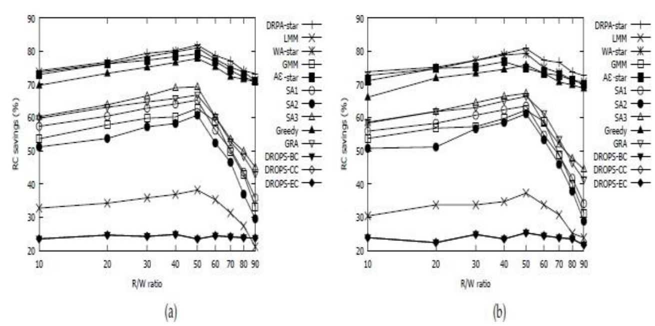

Fig.7. (a) RC versus R/W ratio (Three tree) (b) RC versus R/W ratio (Fat tree)

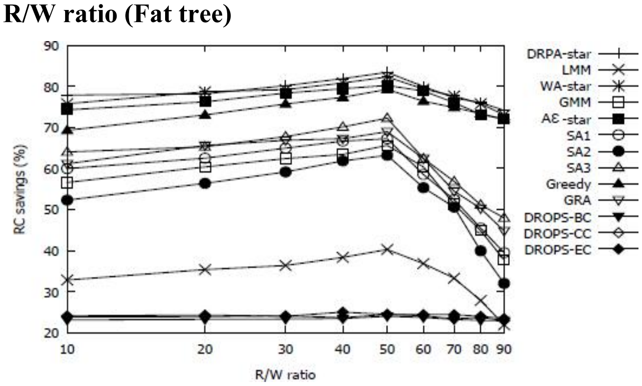

Fig.8. RC versus R/W ratio (Dcell)

ISSN: 2366-1313

Moreover, the T-coloring allows only a singlereplica to be stored on any node. Therefore, after acertain point, the increase in storage capacity mightnot affect the performance. We increase the nodes storage capacity incrementallyfrom 20% to 40%. The results are shown in Fig.5 (b), Fig. 6 (a), and Fig. 6(b). It is observable from The plots that initially, all of the algorithms showedsignificant increase in performance with an increasein the storage capacity. Afterwards, the marginal increasein the performance reduces with the increase inthe storage capacity. The DRPA-star, greedy, WA-star,and Aϵ-star showed nearly similar performance andrecorded higher performance. The DROPS methodologydid not show any considerable change in resultswhen compared to previously discussed experiments(change in number of nodes and files). This is becausethe DROPS methodology does not go for a full-scalereplication of file fragments rather they are replicatedonly once and a single node only stores a singlefragment. Single time replication does not requirehigh storage capacity. Therefore, the change in nodesstorage capacity did not affect the performance ofDROPS to a notable extent.

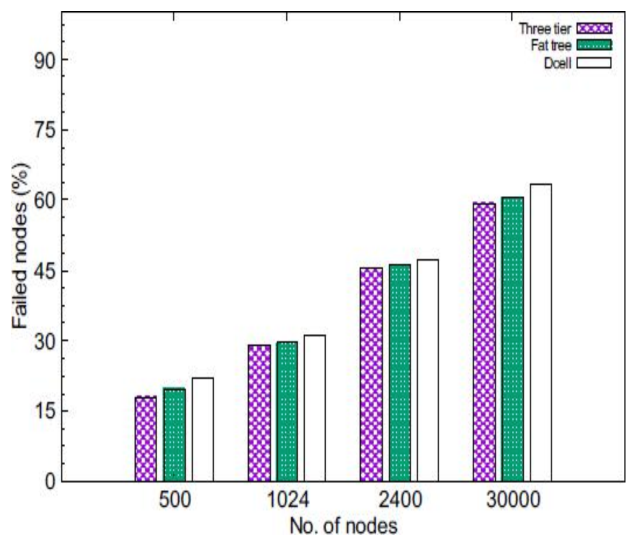

Fig.9. Fault tolerance level of DROPS

4. Impact of increase in the read/write ratio

The change in R/W ratio affects the performance ofthe discussed comparative techniques. An increase inthe number of reads would lead to a need of morereplicas of the fragments in the cloud. The increasednumber of replicas decreases the communication costassociated with the reading of fragments. However,the increased number of writes demands that thereplicas be placed closer to the primary node. Thepresence of replicas closer to the primary node resultsin decreased RC associated with updating replicas.The higher write ratios may increase the traffic on thenetwork for updating the replicas. Fig7(a), Fig7(b), and Fig. 8 show the performanceof the comparative techniques and the DROPSmethodology under varying R/W ratios. It is observedthat all of the comparative techniques showedan increase in the RC savings up to the R/W ratio of0.50. The decrease in the number of writes caused thereduction of cost associated with updating the replicasof the fragments. However, all of the comparativetechniques showed some sort of decrease in RC savingfor R/W ratios above 0.50. This may be attributed tothe fact that an increase in the number of reads causedmore replicas of fragments resulting in increased costof updating the replicas. Therefore, the increased costof updating replicas underpins the advantage of decreasedcost of reading with higher number of replicasat R/W ratio above 0.50. It is also important to mentionthat even at higher R/W ratio values the DRPA-star,WA-star, Aϵ-star, and Greedy algorithms almostmaintained their initial RC saving values. The highperformance of the aforesaid algorithms is due to thefact that these algorithms focus on the global RC valuewhile replicating the fragments.

Therefore, the globalperception of these algorithms resulted in high performance.Alternatively, LMM and GMM did not showsubstantial performance due to their local RC viewwhile assigning a fragment to a node. The SA1, SA2,and SA3 suffered due to their restricted search treethat probably ignored some globally high performingnodes during expansion. The DROPS methodologymaintained almost consistent performance as is observablefrom the plots. The reason for this is that theDROPS methodology replicates the fragments onlyonce, so varying R/W ratios did not affect the resultsconsiderably. However, the slight changes in the RCvalue are observed. This might be due to the reasonthat different nodes generate high cost for R/W offragments with different R/W ratio. As discussed earlier, the comparative techniquesfocus on the performance and try to reduce the RCas much as possible. The DROPS methodology, onthe other hand, is proposed to collectively approachthe security and performance. To increase the securitylevel of the data, the DROPS methodology sacrificesthe performance to certain extent. Therefore, we see adrop in the performance of the DROPS methodologyas compared to discussed comparative techniques.However, the drop in performance is accompanied bymuch needed increase in security level.

Moreover, it is noteworthy that the difference in performance level of the DROPS methodology andthe comparative techniques is least with the reducedstorage capacity of the nodes (see Fig. 5 (b), Fig. 6 (a), and Fig. 6 (b)). The reduced storage capacity proscribesthe comparative techniques to place as manyreplicas as required for the optimized performance. Afurther reduction in the storage capacity will tend toeven lower the performance of the comparative techniques.Therefore, we conclude that the difference inperformance level of the DROPS methodology and the comparative techniques is least when the comparative techniques reduce the extensiveness of replication forany reason. Due to the fact that the DROPS methodology reducesthe number of replicas, we have also investigatesthe fault tolerance of the DROPS methodology.If two nodes storing the same file fragment fail, theresult will be incomplete or faulty file. We randomlypicked and failed the nodes to check that what percentageof failed nodes will result in loss of data or Selection of two nodes storing same file fragment.The numbers of nodes used in aforesaid experimentwere 500, 1,024, 2,400, and 30, 000. The number

ISSN: 2366-1313

of filefragments was set to 50. The results are shown in Fig.9. As can be seen in Fig. 9, the increase in number ofnodes increases the fault tolerance level. The randomfailure has generated a reasonable percentage for asoundly decent number of nodes.

TABLE 1. Average RC (%) savings for increase in number of nodes

Architecture	DRPA	LMM	wa-star	GMM	Aϵ-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 2. Average RC (%) savings for increase in number of fragments

Architecture	DRPA	LMM	wa-star	GMM	Aϵ-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 3. Average RC (%) savings for increase in storage capacity

Architecture	DRPA	LMM	wa-star	GMM	Aϵ-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 4. Average RC (%) savings for increase in R/W ratio

Architecture	DRPA	LMM	wa-star	GMM	Aϵ-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

We report the average RC (%) savings in Table 1, Table 2, Table 3, and Table 4. The averages are computedover all of the RC (%) savings within a certain class ofexperiments. Table 1 reveals the average results of allof the experiments conducted to observe the impact ofincrease in the number of nodes in the cloud for all ofthe three discussed cloud architectures. Table 2 depictsthe average RC (%) savings for the increase in thenumber of fragments. Table 3 and Table 4 describe theaverage results for the increase the storage capacityand R/W ratio, respectively. It is evident from theaverage results that the Dcell architecture showedbetter results due to its higher connectivity ratio.

V. CONCLUSION

The user has to register in cloud, for each registered user, a unique secret key is generated .the user when wants to upload the file,it gets splitted into small chunks and for every upload of file a secret file key is also generated when user wants to download a file, they should enter a secret file key of their file,then splittedchunks get merged and can download the file. This provides security at client level as well as in network level. The aforesaid future work will save the time and resources utilized in downloading, updating, and uploading the file again.

VI. REFERENCES

[1] Mazhar Ali, Student Member, IEEE, Kashif Bilal, Student Member, IEEE, Samee U. Khan, SeniorMember, IEEE, BharadwajVeeravalli, Senior Member, IEEE, Keqin Li, Senior Member, IEEE, andAlbert Y. Zomaya, Fellow, IEEE, "DROPS: Division and Replication of Data inCloud for Optimal Performance and Security", IEEE Transactions on Cloud Computing.

[2] 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 datacenter architectures," Concurrency and Computation: Practice andExperience, Vol. 25, No. 12, 2013, pp. 1771-1783.

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

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

[5]Y. Deswarte, L. Blain, and J-C. Fabre, "Intrusion tolerance in distributedcomputing systems," In Proceedings of IEEE ComputerSociety Symposium on Research in Security and Privacy, OaklandCA, pp. 110-121, 1991.

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

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

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

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

[10] W. A. Jansen, "Cloud hooks: Security and privacy issues incloud computing," In 44th Hawaii IEEE International ConferenceonSystem Sciences (HICSS), 2011, pp. 1-10.

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

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