A Survey on Storage Techniques in Cloud Computing

Samson Akintoye
ISAT Laboratory, Department of Computer Science, University of the Western Cape, Cape Town, 7535, South Africa.

Antoine Bagula
ISAT Laboratory, Department of Computer Science, University of the Western Cape, Cape Town, 7535, South Africa.

Yacine Djemaiel
CNAS Lab, Carthage University, Tunis, Tunisia

Noureddine Bouriga
CNAS Lab, Carthage University, Tunis, Tunisia

ABSTRACT
In recent years, cloud computing is highly embraced and considered a must have in the IT world. Cloud computing is a computing paradigm, where a large pool of systems are connected in private, public or hybrid networks, to provide dynamically scalable infrastructure for computing resources. One of the services provided by cloud computing is cloud storage that enables the upload and the retrieval of different kinds of data. Accessing cloud storage service through Internet and pay as you used subscription have been the reasons for the emergence of methods and techniques to effectively store data and reduce storage security vulnerabilities in the cloud storage. This paper provides a survey of some proposed cloud storage methods and techniques, their advantages and drawbacks and makes stress on the current requirements for storage techniques in cloud computing.

Keywords
Cloud computing, Storage Technique, Cloud Storage, Cloud Security

1. INTRODUCTION
Recently, the processing and the storage of huge volumes of data have been enhanced enormously in these last decades due to the emergence of Cloud computing. This concept is defined as a computing paradigm, where a large pool of systems are connected in private, public or hybrid networks, to provide dynamically scalable infrastructure for computing resources [18]. The characteristics of cloud computing include on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. On-demand self-service means that organizations can access and manage their own computing resources. Broad network access allows services to be offered over the Internet or private networks.

The cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [11]. In SaaS model, consumer uses the provider's applications running on a cloud infrastructure. In PaaS, an operating system, hardware, and network are provided, and the customer installs or develops its own software and applications. The IaaS model provides just the hardware and network; the customer installs or develops its own operating systems, software and applications.

Cloud services are typically provided through a private cloud, community cloud, public cloud or hybrid cloud [11]. In public cloud, services are offered over the Internet and are owned and operated by a cloud provider. In a private cloud, the cloud infrastructure is operated solely for a single organization, and is managed by the organization or a third party. In a community cloud, the service is shared by several organizations and made available only to those groups. The infrastructure may be owned and operated by the organizations or by a cloud service provider (CSP). A hybrid cloud is a combination of two or more cloud infrastructures (private, community, or public) that remain unique entities, but are bounded together by standardized or proprietary technology that enables data and application portability.

One of the important services provided by cloud computing is cloud storage, which is a service where data is remotely maintained, managed, and backed up. The service is available to users over a network, that is usually the Internet. It allows the user to store files online on a "pay-as-you go" or subscription basis so that the user can access them from any location via the internet. There is no need to purchase storage before storing data. Only the amount of storage the data actually consuming is paid [12]. Furthermore, accessing cloud storage service through Internet and pay as you used subscription have been the reasons for the emergence of methods and techniques to effectively store data and reduce storage security vulnerabilities in the cloud storage. This paper reviews and surveys some proposed cloud storage techniques, their advantages and drawbacks, data management and retrieval from cloud storages in addition to security mechanisms that may be implemented to protect storage clouds.

2. STORAGE TECHNIQUES IN CLOUD COMPUTING
In this section, we review some methods and techniques to store data in cloud computing. The techniques are grouped into Storage Techniques for cloud, data management in cloud storage, data retrieval from cloud, authentication schemes for cloud storage, and data integrity and availability for cloud storage.

2.1 SETiNS: Storage Efficiency Techniques in No-SQL database for Cloud Based Design
Bhatia et al. [1] proposed a novel design for efficient storage technique in No-SQL Database for cloud storage. According to this technique, data is fetched from Hadoop Distributed file system (HDFS). Data are stored in the form of flat text files and to applying SETiNS efficiently for large data, the file is deduplicated using a MapReduce to provide the key-value pairs on the basis of which pointer table is created using MongoDB. MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document. Collection is a group of MongoDB documents which is equivalent of a table structure in an RDBMS model. A document is a set of key-value pairs

represented as a dynamic schema. In this work, key-value pairs with the pointer table are now stored as a single document. To get more efficient storage, the file is compressed and saved in HDFS to save bandwidth consumption due to the reduced size. Deduplication and compression together are used to reduce storage space for unravelling large volumes of data handling problems.

The execution phase include: deduplicated, MapReduce Paradigm, Integration of the key-value pairs and compression in Hadoop. The result shows improved storage efficiency by saving subsequent storage space and network bandwidth in the case of internodes data transfer.

2.2 Dynamic Data Deduplication in Cloud Storage

Leesakul et al. [4] proposed a dynamic deduplication scheme for cloud storage, to improve storage efficiency and maintaining redundancy for fault tolerance. Data deduplication is a technique used to reduce storage space and network bandwidth. In existing deduplication systems duplicated data chunks identify, store only one replica of the data in storage and logical pointers are created for other copies instead of storing redundant data. The existing deduplication schemes may prevent the system fault tolerance since it may be that several files refer to the same data chunk which may be unavailable due to failure. The dynamic deduplication scheme was proposed to balance between storage efficiency and fault tolerance requirements and address limitation of static deduplication scheme that cannot cope with changing user behavior. For instance, data usage in cloud changes overtime; some data chunks may be read frequently in a period of time, but may not be used in another period. Dynamic deduplication scheme has the capability to adapt to various access patterns and changing user behavior in cloud storages. The proposed system is based on client-side deduplication using whole file hashing. Hashing process is performed at the client, and connects to any one of deduplicators according to their loads at that time then identifies the duplication by comparing with the existing hash values in Metadata Server. The system is composed of the following components: load balancer that requests from clients sending to any one of deduplicators according to their loads at that time; Deduplicators which identify the performed duplication; Cloud Storage, a Metadata Server to store metadata and File Servers to store actual files and their copies; and Redundancy Manager to identify the initial number of copies, and monitor the changing level of Quality of Service (QoS). The system model was stimulated using HDFS, one Namenode as Metadata server, and five Datanodes as File servers. Three events were simulated: upload, update, and delete. The upload event is when the file is first uploaded to the system. If files already exist in the system, and have been uploaded again, the number of copies of the files will be recalculated according to the highest level of QoS. For a delete file event, users can delete their files, but the files will not permanently deleted from the system if there are any other users refer to the same files. The number of copies of files was changed dynamically according to the changing level of QoS. The experimental results show that, the proposed system performs well and can deal with scalability problem.

2.3 A file-deduplicated private cloud storage service with CDMI standard

Liu et al. [5] proposed a data deduplication private cloud storage system with Cloud Data Management Interface (CDMI) standard based on the fundamental DFS. A data deduplication scheme is implemented in the proposed system to reduce cost and increase the storage efficiency, the international standard interface CDMI is implemented in the proposed system to increase the interoperability. Gluster is chosen as the basic for DFS to implement proposed private cloud storage system. The proposed private cloud storage system consists of five components as shown in Figure 1: Client which communicates with Controller in Front-end node and exchange information; Front-end node contains Apache server that redirects the requests which are sent from CDMI request sender to enhance load balance; Adaptor node receives CDMI request and stores files via Gluster client. Storage nodes contains GlusterFS server, which can create different types of volume for different purposes. The proposed system provides the following three main functionalities: upload file, download file, and delete file. During the file upload process, hash value of the file is calculated by Hash generator in Client and sent to the Controller where the hash value is compared with all file metadata stored in Database. Controller will notify Client that file doesn't exist, if the file is not duplicated. Then, the file's metadata will be sent to Controller to insert it into database and CDMI sender in Client will send upload file request. After receiving the request, Load balancer will route it to CDMI Server by the apache load balancer scheduler algorithms. Finally, Adaptor node will store the file to Storage node and will notify Client with uploading finished response message. But, if the file is duplicated, Controller will notify the Client that the file exits, then Client will send this file's metadata to Controller to insert it into database which will create an empty metadata file with the same file name Storage node.

For download functionality, Client initiate request and send a file_path to the Controller. The Controller will search database for the real_path of this file and respond to Client. Then, client's CDMI sender will send CDMI download file request and the Load balancer redirects the request to CDMI server by the apache load balancer scheduler algorithms. Finally, Adaptor nodes get the requested file from GlusterFS and transmit it back to Client.

The delete functionality is classified into two types according to the relationship among file_path, real_path and whether the file is shareable. If file_path is equal to real_path and the file is shareable, the file is moved to the other place rather than delete directly. Controller query database by file_path to find a suitable path for moving files. After removing the files, the Database will be updated and Client will receive the delete response message. But if file_path is not equal to real_path, or file path is equal to real path and the file is not shareable, the file can be deleted directly. After receiving a delete message which contains file_path, Controller will delete file according to the file_path directly and Client will receive the delete response message. The experiments are carried out to estimate the efficiency and characteristics of the proposed system. The results show that the proposed system is efficient for data transmission, even if there are overhead times caused by data deduplication mechanisms. The disadvantage of the proposed system is unable to perform very well while dealing with bigger files since the file level deduplication is adopted.

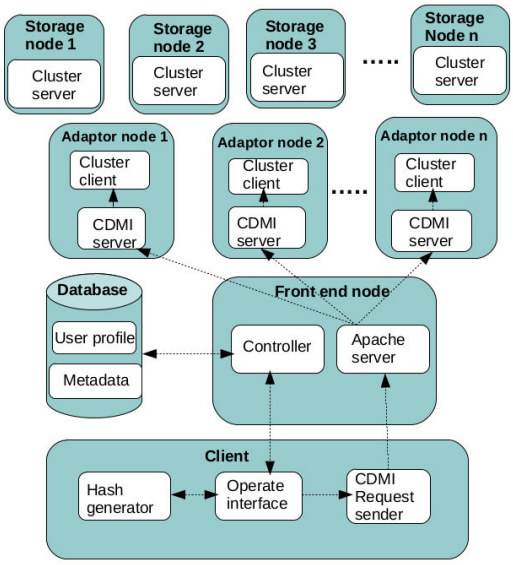

Figure 1: A file-deduplicated private cloud storage service with CDMI standard [5]

2.4 An optimized approach for storing and accessing small files on cloud storage

HDFS is an open-source software framework developed for reliable, scalable, distributed computing and storage [21]. It performs better for small size file than large file. However, there are some reasons for small file problem of native HDFS: large numbers of small files impose huge load on NameNode of HDFS, the relationships between small files are contemplated for data placement, and optimization mechanism is not provided to enhance I/O performance. In order to solve the problems of small files of HDFS, Dong et al. [13] proposed an optimized model to improve the performance of storing and accessing small files on HDFS. The cut-off point between large and small files is measured in HDFS by an experiment. This solve a 'how small is small' problem of small file and improves I/O performance. First, files are classified as independent files, structurally and logically-related files. Lastly, perfecting technique is used to improve access efficiency and considering correlations when files are stored. Performance analysis indicated that the proposed model improves the storage and access efficiencies of small files, compared with native HDFS and a Hadoop file archiving facility.

2.5 RACS (Redundant Array of Cloud Storage)

Abu-Libdeh et al. [15] proposed RACS (Redundant Array of Cloud Storage) a proxy based system that transparently stripes data across multiple cloud storage providers. The protocol allows customers to avoid vendor lock-in, reduces the cost of switching providers, and better tolerates provider failures. It retrieves data from the cloud that is about to fail and moves the data to the new cloud. It is designed as a proxy between the client application and a set of n repositories, which are cloud storage locations ideally hosted by different providers. RACS splits the incoming object into m data shares of equal size where $m< n$ are configurable parameters. It uses erasure coding to create additional $(n-1)$ redundant shares, for a total of n shares. Any subset of m shares is sufficient to reconstruct the original object. This protocol is unable to recover data lost when permanent cloud failure occurred and it does not address data integrity and confidentiality challenges for cloud storage.

3. DATA MANAGEMENT IN CLOUD STORAGE

Several techniques for data management in Cloud Storage have been proposed for different kind of clouds and also for several kind of data.

3.1 Multi-Index Technique for Metadata Management in Private Cloud Storage

Subha et al. [6] proposed a multi-index technique for metadata management of chunks stored private cloud storage to improve the performance of duplicate detection and retrieval of the file. Private cloud storage is built for an organization to offer a higher level of security and control over the data.

However, private cloud storage offers limited storage space; deduplication technique is applied to utilize the storage in an efficient manner. This technique aims to remove the redundant data and utilize the storage space in an optimized manner. It divided the file into several chunks and computes the hash value for those chunks. The hash value of a chunk is known as its chunkID. A chunk index contains chunkID and the location of the corresponding actual chunk. When a chunk of a file enters the storage, its chunkID is compared against the chunk index to determine whether it is a duplicate chunk. Chunk index entries of the entire storage cluster are divided into several indices based on the types of files. If files are mutable, content based chunking is used to divide the file to identify the duplicates. If files are less mutable, duplicates are identified by performing fixed size chunking method. Duplicates of immutable files are identified by comparing the entire files as the content of the files is not modified. The indices are distributed across different storage nodes and experiments were carried out to select a suitable index structure to organize the chunk entries for their quick retrieval. Workload consisting of different types of files with various sizes are deduplicated and stored in the storage cluster. The retrieval times for different types of files are computed with the distributed multi-index and are compared against the time taken with the sequential index. Results show that the distributed multi-index out performs the sequential index.

3.2 Indexing spatial data in cloud data managements

Wei et al. [7] proposed a novel efficient multi-dimensional index structure, the KR+-index, on cloud data managements (CDMs) for retrieving skewed spatial data efficiently. KR+-index is used for multi-dimensional range queries and nearest neighbor queries and used R+ to build the index structure and designed the key for efficient accessing of data. New efficient spatial query algorithms, range queries and k-NN queries for the proposed KR+-index are redefined. To insert a new data point, the algorithm first loops up the key of the node corresponding to the node to which the point belongs, then inserts the data point into the node. Since there is an upper bound to the number of points in the node, the insertion algorithm checks the current size of the node to determine if a split is needed. For a deletion event, the algorithm first loops up the key of the node corresponding to the node to which the point belongs, and then deletes the data point from the node. The experiments to implement the KR+-index on Cassandra using spatial data is carried out. The results show that KR+-index better than the additional methods under the skew data distributions.

3.3 Exploiting user metadata for energy-aware node allocation in a cloud storage system

Cloud computing enables users to store data, host applications and perform computations over the network as cost- effective platforms. Cloud storage systems have lower energy costs compared to traditional HPC clusters due to the better utilization techniques. However, the increasing energy consumption of cloud storage systems still need to be addressed as the amount of data stored and the number of computations and applications in cloud increase rapidly. In [8], several energy efficient storage node allocation methods are introduced. These methods aims to allocate a subset of cloud storage nodes for each user in order to reduce the energy by exploiting the metadata heterogeneity of cloud users and preserve uniform distribution of the system resources on demand. Three different methods to map cloud storage nodes to the cloud users were proposed in this work. The first scheme includes balancing, sequential, random and grouping which can be changed manually once one of these techniques is chosen for a cloud storage system. Secondly, the dynamic Greedy Scheme is an extension of fixed scheme method by periodically doing dynamic reallocation among one of the four aforementioned allocation techniques depending on their costs. The last method is a Correlation-Based Scheme which monitors user activities and tries to allocate the same subset of storage nodes to the users who tend to use the cloud storage system simultaneously. The proposed energy-aware node allocation methods have the capability to work for both traditional distributed storage system architecture and disk array. The proposed methods reduce not only the energy consumption, but also balance load on storage nodes depending on metrics determined by the cloud administrators.

4. DATA RETRIEVAL

4.1 Efficient data retrieval from cloud storage using data mining techniques

Pratiba et al. [2] proposed top-n multi keyword retrieval over encrypted cloud data using vector space model and Two Round Searchable Encryption (TRSE) scheme. The protocol was developed to address the limitations of existing searching schemes. In the traditional searchable encryption schemes, users are allowed to search in the encrypted cloud data through keywords, which support only Boolean search. In Single keyword ranked search, searching of data in the cloud results too coarse output and the data privacy is opposed using server side ranking based on order-preserving encryption (OPE). The aim of the proposed system are threefold: 1) protection of sensitive information by encrypting cloud data at the administrator side; 2) authentication to the multi users; and 3) retrieving top-n files matching the multi keywords and preserving privacy of encrypted data using Two round searchable encryption method. In the proposed scheme, the data owner encrypts the file using RSA encryption and also, encrypts the searchable index using Homomorphic encryption. The cloud storage server computes the scores from the encrypted index stored on cloud and returns the encrypted file with its scores to the user as the server receives a query consisting of multi-keywords from the user. Once the scores are received, the user decrypts the scores and selects the files and sends the respective file IDs to the server. Server sends the encrypted file to the user and then the user decrypts the file using the private key sent by the administrator. There are two-round communication between the cloud server and the data user to retrieve the top-n files; calculation is performed at the server side and the ranking is ensured at the user end. The proposed system avoids overloads by ranking the files at the user side, reducing bandwidth and protects document frequency.

4.2 Efficient Indexing and Query Processing of Model-View Sensor Data in the Cloud

Guo et al. [3] proposed an innovative interval index, hybrid model-view query processing and enhanced intersection search algorithm for model-view sensor data in the cloud. Model-view sensor data management stores the sensor data in the form of modelled segments which brings the additional advantages of data compression and value interpolation.. The interval index known as KVI-index is a two-tier structure consisting of one lightweight and memory-resident binary search tree and one index-model table materialized in the key-value store. Each segment of model-view sensor data consists of specific time and value range and combination of the time and value intervals are represented as keys to index each segment which allows the index to directly serve the queries proposed. The **k**ey-**v**alue denotes **i**nterval index (KVI-index) to index time and value intervals of model-view sensor data. In KVI-indexing, the virtual searching tree (vs-tree) resides in memory, while an index-model table in the key-value store is devised to materialize the secondary structure (SS) of each node in vs-tree. The combination of KVI-index, range scan and MapReduce known as KVI–Scan–MapReduce was applied to perform efficient query processing for model-view sensor data in key-value stores. The results of the experiments aiming to compare model-view sensor data query processing with conventional one over raw sensor data, showed that the proposed solution outperforms in terms of query response time and index updating efficiency.

4.3 Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage

In securing cloud data, the data owner can encrypt its data content before outsourcing rather than leaving it in the plaintext form. However, typical encryption would not be suitable for cloud information retrieval systems because the Cloud Service Provider cannot retrieve encrypted contents from a plaintext query without the decryption keys. To provide an information retrieval function while addressing the security and privacy issues, Koo et al. [21] propose a searchable encryption scheme using Attribute-Based Encryption(ABE) with scrambled attributes to handle the presence of redundant encrypted data for the same message, poor expressiveness regularly access policy and the concentration of computational overhead on the searching entity. In this scheme, the data owner can specify both of fine-grained access policy and searching keyword set which is required to retrieve its data under the access policy. To retrieve the encrypted content in cloud storage, the retriever makes index terms from its private key satisfying the access policy made up of keywords associated with the contents where these index terms are only used for data accessing in the cloud storage system. This scheme has advantage of one-to-many content distribution without a sacrifice of the nature of ABE.

5. AUTHENTICATION SCHRMES FOR CLOUD STORAGE

5.1 Identity-Based Authentication for Cloud Computing

In Cloud Computing, cloud services are accessed through Internet from different locations which exposed them to a set of security threats. Hence, there is a need to authenticate users before accessing cloud services. Li et al. [10] proposed identity-based hierarchical model for cloud computing and corresponding encryption and signature schemes. The identity-based Authentication Protocol is composed of the following steps. In step (1), the client *C* sends the server *S* a ClientHello message. The message contains a fresh random number *C n*, session identifier *ID* and *C* specification. In step (2), the server S responds with a ServerHello message which contains a new fresh random number *S n*, the session identifier *ID* and the cipher specification *S*. The ciphertext is transmitted to C as ServerKeyExchange message. Then S generates a signature *Sig S S [M]* as the IdentityVerify message to forward to *C*. In step (3), *C* verifies the signature *S Sig S S* with the help of *S ID* Being certificate-free, the authentication protocol supports the idea of cloud computing. It was observed that authentication protocol based on identity is more efficient and less weight than Secure Socket Layer (SSL) Authentication Protocol.

5.2 Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage

One of the services offer by CSPs is cloud storage with low cost, so it reduces burden of local data storage but when data is outsourced user loses control of their data which brings a security risk toward integrity and confidentiality. To address this issue, Zhou et al. [19] proposed a Role-Based Encryption (RBE) scheme that integrates the cryptographic techniques with role-based access control (RBAC) to control and prevent unauthorized access to data stored in the cloud. This work also presented a RBAC based cloud storage architecture which allows an organisation to store data securely in a public cloud, while maintaining the sensitive information related to the organization's structure in a private cloud. The proposed RBE-based architecture is implemented and the performance results show that encryption and decryption computations are efficient on the client side, and decryption time at the cloud can be reduced by having multiple processors, which is common in a cloud environment. The advantage of the proposed system is that, it has potential to be useful in commercial situations as it captures practical access policies based on roles in a flexible manner and provides secure data storage in the cloud enforcing these access policies.

5.3 Data Security scheme for Cloud Computing using Signcryption based on HyperElliptic Curves

Cloud Computing is a computing model which provides ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with little or no up-front IT infrastructure investments costs. Cloud computing moves the application and data to the cloud storage where the management of the data and services may not be fully trustworthy. Therefore, there is a need for cloud service providers to provide a sufficient level of integrity for the client's data. Akintoye et al. [22] proposed a data security scheme using Signcryption and hyper elliptic curves as a single logical step. Signcryption schemes based on hyperElliptic curves saves more computational time and communication cost.

5.4 DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems

In cloud storage system, data access control has been identified as challenging issue and existing access control schemes are not appropriate because they either produce multiple encrypted copies of the same data or require a fully trusted cloud server. For instance, Ciphertext-Policy Attribute-based Encryption (CP-ABE) requires a trusted authority that manages all the attributes and distributes keys in the system and cannot be directly applied to data access control for multi-authority cloud storage systems, due to the inefficiency of decryption and revocation. To address these problems Yang et al. [20] propose DAC-MACS (Data Access Control for Multi-Authority Cloud Storage), an effective and secure data access control scheme with efficient decryption and revocation. In this scheme, a new multi-authority CP-ABE scheme with efficient decryption is built. The access control scheme consists of four phases: System Initialization, Key Generation, Encryption and Decryption. In a second step, authors design an efficient attribute revocation method for both forward security and backward security. The attribute revocation includes three phases: Update Key Generation, Key Update and Ciphertext Update. The analysis and the simulation results show that the proposed scheme is highly efficient and provably secure. The advantage of the scheme is that it incurs less storage overhead, communication cost and computation cost compared to existing schemes.

6. DATA INTERITY AND AVAILABILITY FOR CLOUD STORAGE

6.1 HAIL (High-Availability and Integrity Layer)

Bowers et al. [14] proposed HAIL (High-Availability and Integrity Layer) that provides data integrity and availability across multiple cloud storage. It engages the use of PORs (Proofs of Retrievability) as building blocks by which storage resources can be verified and reallocated when failures are detected. It relies on a single trusted verifier that interacts with nodes to check the integrity of stored data. In HAIL, a client distributes a file F with redundancy across n servers and keeps some small (constant) state locally. The goal of HAIL is to ensure resilience against a mobile adversary. The advantages of the protocol are: Strong file-intactness assurance, Low overhead, Strong adversarial model, Direct client-server communication and Static/dynamic file protection. However, the protocol is built on erasure code which provides only fault tolerance; it does not address the recovery of the outsourced data on a failed cloud without ensuring the confidentiality of the stored data.

6.2 DEPSKY

Bessani et al. [16] proposed DEPSKY, a dependable and secure storage system to improve the availability, integrity and confidentiality of data stored in the. It addresses four limitations of individual clouds storage: loss of availability, loss and corruption of data, loss of privacy and Vendor lock-in by using Byzantine quorum system protocol, cryptography, secret sharing, erasure codes and the diversity that comes

from using several clouds. DEPSKY is designed as a virtual storage cloud, which is accessed by its users by invoking operations in several individual clouds.

6.3 NCCloud

Chen et al. [17] proposed NCCloud, a proxy-based storage system for fault-tolerant multiple-cloud storage that achieves cost-effective repair for a permanent single-cloud failure. It is built on top of a network-coding-based storage scheme called the functional minimum-storage regenerating (FMSR) codes, which maintain the same fault tolerance and data redundancy as in traditional erasure codes (e.g., RAID-6), but instead it uses less repair traffic which incur less monetary cost due to data transfer. NCCloud is a proxy based design that interconnects multiple cloud repositories. The proxy serves as an interface between client applications and the clouds. If a cloud experiences a permanent failure, the proxy activates the repair operation. The proxy reads the essential data chucks from other surviving clouds, reconstructs new data chucks, and writes these new chucks to a new cloud. It excludes the failed cloud in repair operation. However, the protocol does not guarantee the integrity and confidentiality of the data chunks stripped across multiple clouds.

6.4 Minimizing data redundancy for high reliable cloud storage systems

One of the services provided by cloud storage system is reliability of the outsourced data. In order to avoid failures, copies of data objects are distributed across set of storage nodes using redundancy schemes. In [9], a novel model for the storage allocation scheme in cloud storage systems is proposed aiming to minimize the data redundancy while achieving a given (high) data reliability. To optimize the redundancy, two schemes by finding minimum number of blocks in total with fixed number of sufficient blocks for recovery and finding maximum number of sufficient blocks for recovery with fixed number of blocks in total, which allows the schemes to be deployed for different systems. The optimal storage allocation scheme has advantages to significantly reduce the search space, simplify and accelerate the calculation process; and save redundancy process.

7. ANALYSIS AND CLASSSIFICATION OF STORAGE TECHNIQUES IN CLOUD COMPUTING

In this section, we made an analysis of storage techniques in cloud computing based on their performances and benefits. We also classified the techniques according to their usage.

Table 1: Analysis Of Storage Techniques

Reviewed papers	Descriptions and key issue handled	Benefits
Bhatia et al. [1]	Used HDFS, compression algorithm and MongoDB database for cloud storage design.	Improved storage efficiency by saving subsequent storage space and network bandwidth.
Pratiba et al. [2]	Top-n multi keyword retrieval over encrypted cloud data.	It's secure, scalable and accurate compared to the other ranked keyword search.
Guo et al. [3]	Interval index, hybrid model-view query processing and enhanced intersection search algorithm for model-view sensor data in the cloud.	It provides an optimized query response time and better index updating efficiency.
Leesakul et al. [4]	Balance between storage efficiency and fault tolerance requirements and address limitation of static deduplication scheme that cannot cope with changing user behavior.	It use to handle with scalability problem.
Liu et al. [5]	A data deduplication private cloud storage system with CDMI standard based on the fundamental DFS.	It reduces cost and increases the storage efficiency. It also increase the interoperability between storage nodes.
Subha et al. [6]	Multi-index technique for duplicate detection and retrieval of the file in private cloud storage.	Multi-index out performs the sequential index in term of retrieval time.
Wei et al. [7]	Multi-dimensional index for retrieving skewed spatial data in the cloud storage.	KR+-index better than the state-of-the-art methods under the skew data distributions.
Karakoyunlu [8]	Allocation of a subset of cloud storage nodes for each user to reduce the energy consumption.	It reduces energy consumption and balances load on storage nodes depending on metrics determine by the cloud administrators.
Huang [9]	A novel model for the storage allocation scheme in cloud storage systems to minimizes the data redundancy while achieving a given (high) data reliability.	It has advantages to significantly reduce the search space, simplified and accelerated the calculation process; and save redundancy process.

Li et al. [10]	Identity-based hierarchical model for cloud computing and corresponding encryption and signature schemes.	It more efficient and less weight than Secure socket layer (SSL)Authentication Protocol.
Dong et al. [13]	An optimized model for storing and accessing small files on HDFS.	It improves the storage and access efficiencies of small files, compared with native HDFS and a Hadoop file archiving facility
Bowers et al. [14]	High-Availability and Integrity Layer to provide data integrity and availability across multiple cloud storage.	The advantages of the protocol are: Strong file-intactness assurance, Low overhead, Strong adversarial model, Direct client-server communication and Static / dynamic file protection.
Abu-Libdeh et al. [15]	A proxy based system that transparently stripes data across multiple cloud storage providers.	It allows customers to avoid vendor lock-in, reduce the cost of switching providers, and better tolerate provider failures.
Bessani et al. [16]	A dependable and secure storage system to improve the availability, integrity and confidentiality of data stored in the cloud	It addresses four limitations of individual clouds storage: Loss of availability, Loss and corruption of data, Loss of privacy and Vendor lock-in
Zhou et al. [19]	A role-based encryption (RBE) scheme that integrates the cryptographic techniques with role-based access control (RBAC) .	It controls and prevents unauthorized access to data stored in the cloud.
Akintoye et al. [22]	A data security scheme for cloud computing using Signcryption and hyper-elliptic curves as a single logical step	The scheme saves more computational time and communication cost.
Yang et al. [20]	An effective and secure data access control scheme with efficient decryption and revocation.	It addresses inefficiency of decryption and revocation in the existing schemes. Furthermore, it incurs less storage overhead, communication cost and computation cost compared to existing scheme.
Koo et al. [21]	A searchable encryption scheme using Attribute-Based Encryption(ABE)	It provides one-to-many content distribution without a sacrifice of the nature of Attribute-Based Encryption(ABE).
Chen et al. [17]	A proxy-based storage system for fault-tolerant multiple-cloud storage that achieves cost-effective repair for a permanent single-cloud failure.	The proxy activates the repair operation, if a cloud experiences a permanent failure.

Table 2: Classfication Of Storage Techniques

Reviewed papers	Cloud Model	Data Security	Data Redundancy	Data Indexing and Retrieval	Data Storage
Bhatia et al. [1]	Both Private and public clouds	No	Yes	No	Yes
Pratiba et al. [2]	Both Private and public clouds	Yes	No	Yes	Yes
Akintoye et al. [22]	Both Private and public clouds	Yes	No	No	Yes
Guo et al. [3]	Both Private and public clouds	No	Yes	Yes	Yes
Leesakul et al. [4]	Both Private and public clouds	No	Yes	No	Yes

Liu et al. [5]	Private cloud	Yes	Yes	No	Yes
Subha et al. [6]	Private cloud	Yes	Yes	yes	Yes
Wei et al. [7]	Both Private and public clouds	No	Yes	Yes	Yes
Karakoyunlu [8]	Both Private and public clouds	No	No	No	Yes
Huang [9]	Both Private and public clouds	No	Yes	No	Yes
Li et al. [10]	Both Private and public clouds	Yes	No	No	Yes
Dong et al. [13]	Private cloud	No	Yes	Yes	Yes
Bowers et al. [14]	Public Cloud	Yes	Yes	Yes	Yes
Abu-Libdeh et al. [15]	Public Cloud	No	Yes	Yes	Yes
Bessani et al. [16]	Public Cloud	Yes	Yes	Yes	Yes
Chen et al. [17]	Public Cloud	No	Yes	Yes	Yes
Zhou et al. [19]	Both Private and public clouds	Yes	No	No	Yes
Yang et al. [20]	Both Private and public clouds	Yes	No	No	Yes
Koo et al. [21]	Both Private and public clouds	Yes	No	Yes	Yes

8. CONCLUSION

Cloud computing has become increasingly popular in the IT world as the next infrastructure for storing data and deploying software and services. It provides users with a long list of benefits, such as on-demand self-service; broad, heterogeneous network access; resource pooling and rapid elasticity with measured services. One of the important services offered in cloud computing is the cloud data storage, in which, subscribers do not have to store their own data on their servers, instead their data will be stored on the CSP's servers. In this paper, we have reviewed and surveyed some methods and techniques to provide fault-tolerant, availability, reliability, security, integrity and effective cost storage for outsourced data in the cloud. The advantages of these schemes are presented in addition to the set of issues that may be studied by the research communities to enhance the storage for cloud infrastructures.

9. REFERENCES

[1] Vandana Bhatia and Ajay Jangra "SETiNS: Storage Efficiency Techniques in No-SQL database for Cloud Based Design" EEE International Conference on Advances in Engineering & Technology Research (ICAETR 2014),August 01-02, 2014, Dr. Virendra Swarup Group of Institutions, Unnao, India.

[2] D.Pratiba , Dr.G.Shobha and Vijaya Lakshmi.P.S "EFFICIENT DATA RETRIEVAL FROM CLOUD STORAGE USING DATA MINING TECHNIQUE" international Journal on Cybernetics & Informatics (IJCI) Vol. 4, No. 2, April 2015

[3] Tian Guo , Thanasis G. Papaioannou and Karl Aberer "Efficient Indexing and Query Processing of Model-View Sensor Data in the Cloud" Big Data Research, Published by Elsevier Inc http://dx.doi.org/10.1016/j.bdr.2014.07.005 2214-5796, 2014.

[4] Waraporn Leesakul, Paul Townend, Jie Xu "Dynamic Data Deduplication in Cloud Storage" 2014 IEEE 8th International Symposium on Service Oriented System Engineering, 2014

[5] Xiao-Long Liu , Ruey-Kai Sheu , Shyan-Ming Yuan, Yu-Ning Wang "A file-deduplicated private cloud storage service with CDMI standard" Computer Standards & Interfaces, Published by Elsevier Inc http://dx.doi.org/10.1016/j.csi.2015.09.01044 (2016) 18–27

[6] Prabavathy B, Subha Devi M and Chitra Babu "Multi-Index Technique for Metadata Management in Private Cloud Storage" International Conference on Recent Trends in Information Technology (ICRTIT), IEEE, ISBN:978-1-4799-1024-3, 2013

[7] Ling-Yin Wei , Ya-Ting Hsu , Wen-Chih Peng and Wang-Chien Lee "Indexing spatial data in cloud data managements" Pervasive and Mobile Computing,

Published by Elsevier Inc, http://dx.doi.org/10.1016/j.pmcj.2013.07.001, 1574-1192, 2013

[8] Cengiz Karakoyunlu and John A. Chandy "Exploiting user metadata for energy-aware node allocation in a cloud storage system" Journal of Computer and System Sciences, Published by Elsevier Inc, http://dx.doi.org/10.1016/j.jcss.2015.09.003, 2015

[9] Zhen Huang , Jinbang Chen , Yisong Lin , Pengfei You and Yuxing Peng "Minimizing data redundancy for high reliable cloud storage systems" journal of Computer Networks, Published by Elsevier Inc, http://dx.doi.org/10.1016/j.comnet.2015.02.013, 81 (2015) 164–177

[10] Li H, Dai Y and H. Yang, "Identity-BasedAuthentication for Cloud Computing", M. G. Jaatun, G.Zhao, and C. Rong (Eds.): Cloud Computing, Lecture Notes in Computer Science, 2009.

[11] Peter Mell, Tim Grance, The NIST Definition of Cloud Computing, Version 15, October 7, 2009

[12] M. Fallah, M. G. Arani and M. Maeen, "NASLA: Novel Auto Scaling Approach based on Learning Automata for Web Application in Cloud Computing Environment", International Journal of Computer Applications, vol. 113, no. 2, (2015), pp. 18-23.

[13] Bo Dong , Qinghua Zheng , Feng Tian , Kuo-Ming Chao , Rui Ma and Rachid Anane "An optimized approach for storing and accessing small files on cloud storage" Journal of Network and Computer Applications 35 (2012) 1847–1862

[14] K. D. Bowers, A. Juels, and A. Oprea. "Hail: Ahigh-availability and integrity layer for cloud storage." In Proc. of the 16th ACM Conference on Computer and Communication Security (CCS), pages 187–198, November 2009.

[15] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon. Racs: A case for cloud storage diversity. Proc. of the 1st ACM Symposium on Cloud Computing, pages 1204–1216, June 2010.

[16] A. Bessani, M. Correia, B. Quaresma, F. Andre, , and P. Sousa. "Depsky: Dependable and secure storage in a cloud-of-clouds." In Proc. of ACM EuroSys, 2011.

[17] H. C. H. Chen, Y. Hu, P. P. C. Lee, and Y. Tang. "Nccloud:a network coding based storage system in a cloud of clouds. January 2014."

[18] L. Badger, T Grance, R. P. Comer and J. Voas, DRAFT cloud computing synopsis and recommendations, Recommendations of National Institute of Standards and Technology (NIST), May2012.

[19] L. Zhou, V. Varadharajan, and M. Hitchens "Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage" Journal of IEEE Transactions on Information Forensics and Security archive Volume 8 Issue 12, December 2013 Page 1947-1960.

[20] L. Yang, X. Jia, K. Ren and B. Zhang "DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems" 2013 Proceedings IEEE INFOCOM

[21] Hadoop Archives. Hadoop archives guide; 2011. /http://hadoop.apache.org/common/docs/current/hadoop_archives.htmlS.

[22] S. B. Akintoye, and K. A. Akintoye "Data Security scheme for Cloud Computing using Signcryption Based on HyperElliptic curves" Journal of Research and Development, Vol. 2, No. 7, Pg. 10–19, 2015.