Amazon Backed File system with Enhanced Storage Feature

Shilpa Malhotra ,Ms.Shakti Arora, Mr. Surjeet Singh
M.Tech Scholar, Assistant Professor, Assistant Professor
CSE Department
PIET, Samalkha, Haryana, India

Abstract : In the world of IT, end user data is considered to be the base of everything. Huge amount of data is added everyday into the system and a far greater amount of data is being modified every second. Everyone expects that the storing of that data should be on the go and that their data should be safe and secure. While keeping the data secure, the procurement of the same should be seamless and fast. To serve this purpose, one of the, dominant and reliable, technologies of the 20th century has been provided by one of the many products of Amazon Web Services by the name of Simple Storage Service (hereafter referred to as S3) which can store and retrieve huge amount of data from any form of application, be it web based or mobile, at an incredible speed. The data stored in S3 uses a flat file system which is easy to understand and implement. However, there are some drawback for the same when the size of the database increases. The proposed methodology targets few of the limitations of the flat file system of AWS S3 and provides a solution in the form of directory structure which on implementation can result in more security and can tackle inconsistency even for huge databases.

Index Terms - Cloud Computing, Amazon Workspace, Bucket, Objects, Hierarchy, Files, Folders

I.INTRODUCTION

Amazon S3 provides a storage management and administration capability that is highly flexible at its very least. Data can be transferred from or to S3 using reliable API calls provided by the S3 over the internet.Amazon S3 runs on a global cloud infrastructure which is built around region and availability zones thus providing a more effective and highly available way to create databases, design and build applications that are fault tolerant and very scalable.

The security behind download and upload of data to Amazon S3 is backed up by SSL-encrypted endpoints that run on HTTPS protocol. This helps in automatically encrypting data at rest and gives a good number of choices for management of key. Furthermore, user can configure their S3 instances to automatically encrypt their data before storing them in S3 in case the incoming request for storage is not already encrypted. Amazon S3 also provides library for the same.

Amazon S3 has another protection mode in the form of versioning. Every data stored or retrieved can be versioned to preserve, retrieve and restore the object. This helps in case of intentional or unintentional user actions or applications failure. However, requests made to the server will retrieve data from the latest versions unless mentioned otherwise. [1]

The core roots S3 infrastructure is based on buckets and objects which are then worked upon by the Amazon S3 API. The Amazon S3 key feature are:

1. The building block of Amazon S3 which is composed of object data and metadata are called Objects. The format in which data is saved is governed by key-value pairs. The unique identification of any object is key and version ID.

2. Buckets are the wrapping layer for the objects stored on Amazon S3 cloud storage. They behave and serve as the organizational structure since they help in organizing the Amazon S3 namespace at the highest level. For the S3 to work as expected, each object has to be wrapped up in a bucket.

3. To uniquely identify an object inside a S3 bucket, Keys are implemented. However each key can be mapped to only one object. Therefore to access an object inside Amazon S3, it has to be mapped or addressed by using a combination of keys, bucket name and a version ID that can be optional.

4. Regions represent the geographical region where an Amazon S3 bucket created by the user will be stored. As of this date, 14 regions are being supported by the Amazon for use of S3. Objects always stay in that region in which they are initially assigned until explicitly transferred to any other region.

5. Consistency Model defines a read-after-write consistency of data for PUTS of objects in a bucket in any region. To ensure the safety of the data, a success of PUT request is made. Any update operations made to a key are atomic. By performing a PUT operation on a key that is followed by any read operation might return either of the old data or a new updated one but will never return or result in corrupted data.

Objects on Amazon S3 can be archived onto the Amazon Glacier which is a low cost storage service for keeping the archived data secure. This makes Amazon S3 perfect for a wide variety of of cases like websites, content distribution, gaming and mobile by providing low latency and very high throughput performance.

Amazon Simple Storage Service commonly known as Amazon S3, is an Infrastructure as a Service (IaaS) solution which is provided by Amazon Web Services (AWS). Amazon S3 has amplified the IT industry/development activities as it facilitates highly secured and low-latency data storage from the cloud. [2,3]

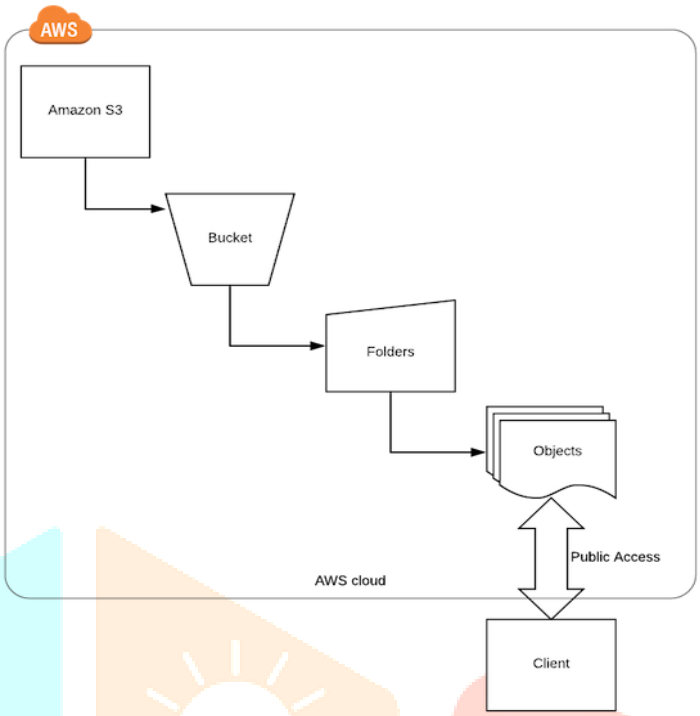


Figure 1: AWS Storage Infrastructure

Amazon Web Services provides choices for enterprise productivity applications that run as services on the AWS Cloud. Applications such as corporate e-mail, calendar, document collaboration, virtual desktop, etc. meet the ease of use, performance and reliability of employees easily while at the same time realizing the security and compliance requirements of the most demanding enterprise IT organization to do.

Amazon S3 has its roots in Cloud Computing. Cloud computing systems are categorized into two sections namely the front end and the back end. The two sections are connected to each other via Internet. Front end refers to the end user or the client. It is not necessary that all cloud systems have the same front-end or the user interface. Front-end usually consists of computer networks, client computers or the applications that are required to access any cloud computing system. The back end section lies in the cloud section of the system which consists of data warehouses, servers and computers that collectively make up the "cloud". [4]

The cloud computing is actually moving the physically stored data to huge data clusters that can be accessed by any authorized user from anywhere in the world. It hence takes away all the heavy lifting that is involved in crunching and processing of data. It is not one piece of technology but a mesh of three services:

Infrastructure as a Service (IaaS)
It eliminates the need of clients to purchase servers or the software but users can procure these resources in an outsourced and on-demand service. Everything from Operating Systems to storage or software are delivered via IP based connectivity as an on-demand service.

Software as a Service (SaaS)
It is concerned with licensure of software applications to clients. Licenses are provided on the basis of pay as you demand and hence it becomes an excellent investment opportunity because the customers pay for what they demand or require.
Platform as a Service (PaaS)
This is the most complex and complicated aspect of cloud computing. It shares some similarities with SaaS; the major primary difference between the two lies in terms of the software. SaaS delivers the software online while PaaS provides a platform for creating the software that is provided online.

These services can either be used together or separately depending upon the need of the organization. Cloud computing has thus become a boon to IT enterprise as it extends the scalability and storage feature on a large scale.

1.1 Object Keys and Meta Data
Each Object at Amazon S3 is composed of two components:
- Key
- Metadata

A key is used to unqiuely identify an object inside a bucket and Object Metadata is further consituted of a set of name-value pairs. While uploading any object, its metadata can be set. Once the object is uploaded, metadata can not be modified. The alternative way to modify it is by making a copy of the object and set the updated metadata. [5]

When an object is created, it is always necessary to specify the key name as it identifies the object uniquely. For instance, when a bucket is highlighted in AWS Management Console, we can see a list of objects in the bucket and these are Object Keys. The name of any key consists of a sequence of Unicode Chars.

II. RELATED WORK

In 2017, Yibin Li, Keke Gai, Longfei Qiu, Meikang Qiu, Hui Zhao proposed a novel methodology that focused on the problem of the cloud data storage and provided an excellent approach to avoid the cloud operators reach the user's sensitive data and their approach was named as Security-Aware Efficient Distributed Storage (SA-EDS) model. The distributed storage has given rise to mass remote data storage via STaas which stands for Storage as a Service model. This model has become a satisfying approach in big data along with Web services and Networks. [6]

In 2016, Omer Y. Adam, Young Choon Lee, and Albert Y. Zomaya proposed one of the major concerns in Cloud where multiple resources are to be located via Performance Predictability. The optimization model in the paper correlated performance variability that occurred across various instances in cloud. [7]

In 2015, S. Narula described how AWS proves out to be a robust, secure and an excellent model when it comes to the security and storage of data on cloud. AWS being the most trusted provider of cloud computing not only provides the excellent performance and security but is also enriched with many other core operations and services. [8]

In 2014, Monjur Ahmed and Mohammad Ashraf Hossain presented a review on in-depth concepts of cloud computing and cloud data security issues that are inherent within cloud infrastructure and cloud computing context. He took into consideration the technical and philosophical factors while dealing with cloud data security issues. The influence of cloud computing includes both, the technical as well as social effects. [9]

In 2012, V. Michael Vrable, S. Stefan and Geoffrey M. Voelker analyzed and prospected solution to network backed file systems on cloud that how LAN based workstations are transparently served by cloud based services and how these continue to provide good performance for enterprise workloads. The paper has proposed the optimization techniques that help in achieving low-cost and good performance along with secured log-structured design. [10]

In 2010, Jesús Hernández Martin, Ioan Raicu focused on IaaS cloud platform after studying the raw performance in terms of I/O. The platform is reliable and easy to use with the increasing number of public cloud platforms and the growth in terms of computing capacity. They have described the tools that can be used for benchmarking which is actually the description of the file storage systems and solutions. [11]

In 2007, Simson L. Garfinkel analyzed the security model of Amazon and based upon which presented the user-report of Amazon of Amazon's computing EC2 and S3 services and concluded that EC2 delivers and continue to provide virtual machines at low cost. This paper deals with the details of grid computing services between November 2006 and May 2007, which also includes a detailed analysis of the overall system's application program interface. Through the findings and outcomes of the paper, it was concluded that Amazon Services are beneficial and innovative and present the customers with best business services when it comes to storing huge amount of data on cloud servers without the need of physical storage keeping in concern the security risks.[4]
S. Obrutsky and E. Erturk studied about the cloud storage products and particularly Amazon S3. The research paper discussed the case study about SmugMug's migration to the cloud hosted by Amazon. SmugMug is basically a premium online served photo and video sharing storage service platform that has numerous photos and videos from various professional photographers across the world. Further, it is analyzed how can a website be modified to use Amazon S3 to host videos on the cloud. The results and outcomes evaluate how beneficial and useful Amazon S3 is for e-learning and other business applications. [12]

III. PROPOSED METHODOLOGY

The paper proposes an efficient and resourceful technique to eradicate the storage structure limitation of AWS. This flexible distributed scheme provides explicit dynamic sustenance to ensure the data is stored in a directory structure on cloud instead of flat file system.

The proposed paper deals with storage of user's data on cloud in a flexible way such that the files and the folders can be stored in a hierarchical manner and this hierarchy can extend to a great level and user can move to any node or any folder in the hierarchy by clicking the desired node in the structure.

The Amazon S3's data model is completely a flat file structure that is when a bucket is created, the bucket stores objects but there exist no hierarchy of sub buckets or sub folders which is one of the limits of Amazon S3 Storage Infrastructure that is addressed in the research and is overcome in the methodology proposed.

Each user is first authenticated before he/she can access the data stored on the cloud to ensure the security of data. There would be one common storage environment for all users.

The data storage infrastructure of AWS has one of the limitations, the way files / folders are stored as flat file system. The users cannot store their files and folders in directory structure on cloud via AWS. The proposed methodology hence targets this limitation.

A well-managed system backed up by Amazon S3 has been developed that supports object creation, update and deletion. The flaw of Amazon S3 of not providing a directory structure and hence storing the data in the form of files only has been overcome. The authorized/authenticated users can access the system and store the data to the 100^{th} level of folder hence making the structure hierarchical rather than the flat file based system.

The research area revolves around enhancing the current file/folder structure of Amazon S3 to make it store the user data with full security and also provide a directory structure to organize and manage the files more precisely.

The research initiated with the base of login to AWS. When users create their account on AWS, they are able to create the objects. The data can be stored in the form of images or files of any type. Each user is given an access, which files they can access. Users are authorized to make sure that they do not access the confidential data that belong to users of other group. Firebase's SDK served the purpose of providing email based authentication to the users.

To portray the above mentioned scenario, the proposed system first authorizes the user before they can login successfully. Prior to login, users have to register themselves with their email address, password and the department they belong to. The two parameters are used while logging in. Once the user is logged in, he is displayed all the files that belong to the same department, the user has opted in for. All the files that belong to the server are available at Root Vault.

Users who are logged in can upload the files to the particular department. Direct files can be loaded and removing the shortcoming of Current Amazon S3 environment, the users can upload the files, embedding in the directory structure meaning that users can store the files in the following way too:

Folder_1/Folder_2/Folder_3/Folder_4/FileName

Pseudocode of how files/folders are created :

1. Start
2. Create a S3 object.
3. Config the S3 object add region and key
4. Set accessKeyId
5. Set secretAccessKey
6. Set region
7. Set parameters to get the object from s3
8. Set Bucket name
9. Set Prefix (url part)
10. Set MaxKeys (number of object want from bucket)
11. Call listObjects() method to get object
12. If error then show error message
13. If data then Parse the response data of the listObjects() method where data.Contents is set of array which contain the list of object.
14. Split the key by "/".
15. Then check if the object is file or folder
16. If file then show in file list.
17. If folder then show in folder list.
18. Stop

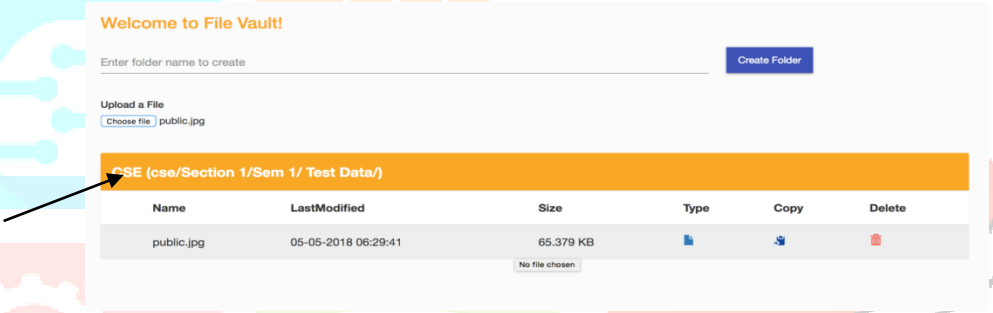
The directory structure is not provided by Amazon currently but by the proposed system. The following code snippet overcome the shortcoming mentioned above:

```
for (var i = 0; i < data.Contents.length; i++) {
   var splitLength = data.Contents[i].Key.split("/").length;
   if (splitLength == 2 && data.Contents[i].Key.split("/")[1] != ""&& data.Contents[i].Key.indexOf('.') > -1) {
      he.files.push({

         name: data.Contents[i].Key.split("/")[1],
         key: data.Contents[i].Key,
         url: he.awsUrl + data.Contents[i].Key,
         size: data.Contents[i].Size,
         lastModified: data.Contents[i].LastModified,
```

```
                type: 0
            });
        }
        else if (splitLength > 2 || (splitLength == 2 && data.Contents[i].Key.split("/")[1] != ""&&
            data.Contents[i].Key.indexOf('.') == -1)) {
            var name = data.Contents[i].Key.split("/")[1];
            var isExist = false;
            for (var j = 0; j < he.folders.length; j++) {
                if (name == he.folders[j].name) {
                    isExist = true;
                    break;
                }
            }
            if (!isExist) {
                he.folders.push({
                    name: data.Contents[i].Key.split("/")[1],
                    key: data.Contents[i].Key,
                    lastModified: data.Contents[i].LastModified,
type: 1
                });
            }
        }
    }
```

And this results into the files that are stored via directory structure and not only at the root location.



The above image displays how a file is stored at the root level of a folder directory structure.

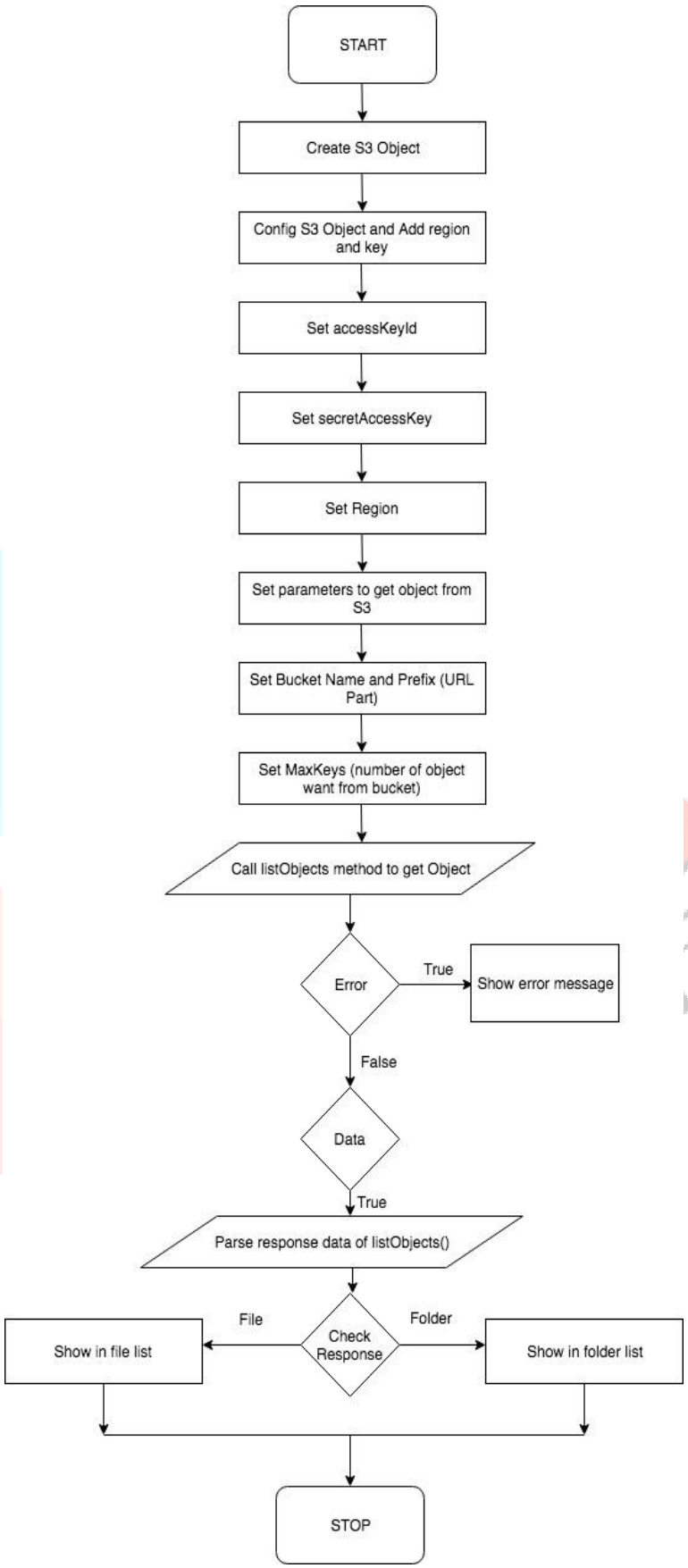


Figure 2: Flowchart of how files and folders are created

IV. CONCLUSION AND FUTURE SCOPE

The proposed system overcomes the shortcoming of Amazon S3 to provide the clients, a directory structure to their files and folders making it easier to organize the data.

Portraying the proposed system, it first authorizes the user before they can login successfully. Prior to login, users have to register themselves with their email address, password and the department they belong to. The two parameters are used while logging in. Once the user is logged in, he is displayed all the files that belong to the same department, the user has opted in for.
All the files that belong to the server are available at Root Vault.

Users who are logged in can upload the files to the particular department. Direct files can be loaded and removing the shortcoming of Current Amazon S3 environment, the users can upload the files, embedding in the directory structure.
The currently proposed work can be enhanced in the following ways:

1. Maintaining versions of the files/data that is being uploaded so that when some change is made to one of the versions, the user is able to access the other versions of the file without any loss or delay. Versioning is a means of keeping multiple variants of an object in the same bucket. The users can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, one can easily recover from both unintended user actions and application failures. Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite.

2. Archiving the files/folders that are older than say a year so that the system is always clean and optimized enough to provide good backup and recovery of data. Archiving data is a far bigger challenge than performing ordinary backups. It helps users not winding up with a collection of obsolete or irretrievable junk.

REFERENCES

[1] Amazon Web Services: Overview of SecurityProcesses. June 2014 White Papers.

[2] Shivlal Mewada, Umesh Kumar Singh and Pradeep Sharma, "Security Enhancement in Cloud Computing (CC)", ISROSET-International Journal of Scientific Research in Computer Science and Engineering, Vol.-01, Issue-01, pp (31- 37), Jan -Feb 2013.

[3] A. Bessani, M. Correia, B. Quaresma, F. Andr´e, and P. Sousa. DepSky: Dependable and Secure Storage in a Cloud-of-Clouds. In EuroSys 2011, Apr. 2011.

[4]Simson L. Garfinkel. An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS. https://dash.harvard.edu/bitstream/handle/1/24829568/tr-08-07.pdf?sequence=1

[5] N. Zhu, J. Chen, and T.-C. Chiueh. TBBT: Scalable and Accurate Trace Replay for File Server Evaluation. In Proceedings of the 4th USENIX Conference on File and Storage Technologies (FAST), Dec. 2005.

[6] Yibin Li, Keke Gai, Longfei Qiu, Meikang Qiu, Hui Zhao, "Intelligent cryptography approach for secure distributed big data storage in cloud computing", 2017.
[7] Omer Y. Adam, Young Choon Lee, and Albert Y. Zomaya, "Constructing Performance-Predictable Clusters with Performance-Varying Resources of Clouds", 2016.

[8]S. Narula. "CLOUD COMPUTING SECURITY:
AMAZON WEB SERVICE". 2015 Fifth International Conference on Advanced Computing & Communication Technologies, 2015.

[9] M. Ahmed, M. Ashraf Hossain, "CLOUD COMPUTING AND SECURITY ISSUES IN THE CLOUD", IJNSA, Vol.6, No.1, January 2014

J. Howard, M. Kazar, S. Nichols, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West. Scale and Performance in a Distributed File System. ACM Transactions on Computer Systems (TOCS), 6(1):51–81, Feb. 1988.

[10] V. Michael Vrable, S. Stefan and Geoffrey M. Voelker, "BlueSky: A Cloud-Backed File System for the Enterprise", 2012.

[11] Jesús Hernández Martin, Ioan Raicu. Performance evaluation of AWS; Exploring storage alternatives in Amazon Web Services. http://datasys.cs.iit.edu/reports/2012_AWS-storage-benchmarks-TR.pdf.

[12] S. Obrutsky and E. Erturk, "Multimedia Storage in the Cloud using Amazon Web Services: Implications for Online Education", 2013.

[13] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish. Depot: Cloud Storage with Minimal Trust. In Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI), Oct. 2010.

[14] Meiko Jensen, JorgSehwenk et al. " 2015 Fifth International Conference on Advanced Computing & Communication Technologies, 2015.

[15] Deyan Chen, Hong Zhao. Data Security andPrivacy Protection Issues in Cloud Computing. 2012Internationals Conference on ComputerScience andElectronics Engineering.

[16] Parneet Kaur and Sachin Majithia, "Various Aspects for Data Migration in Cloud Computing and Related Reviews", International Journal of Computer Sciences and Engineering, Volume-02, Issue-07, Page No (83-85), Jul -2014, E-ISSN: 2347-2693

[17] R. Pike, D. Presotto, S. Dorward, B. Flandrena,K. Thompson, H. Trickey, and P. Winterbottom. Plan 9From Bell Labs. USENIXComputing Systems, 8(3):221–254, Summer 1995.

[18] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In Proceedings of the 1st USENIX Conference on File and Storage Technologies (FAST), 2002.

[19] P.Shanthi Bala," Intensification of Educational Cloud Computing and Crisis of Data Security in Public Clouds",IJCSE Vol. 02, No. 03, 2010, 741-745.

[21] Glen Robinson, Attila Narin, and Chris Elleman.Amazon Web Services- Using AWS for DisasterRecovery. October 2014 White Papers.

[22] C. Ruemmler and J. Wilkes. A trace-driven analysis of disk working set sizes. Technical Report HPL-OSR-93-23, HP Labs, Apr. 1993.

[23] http://aws.amazon.com/what-is-aws/L. Darrell, Unlimited cloud storage at amazon.com, inc on black friday, Url= http://www.bidnessetc.com/58232-unlimited-cloud-storage-at-amazoncominc-on- black- friday/ .

[24] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka,and E. Zeidner. Internet Small Computer Systems Interface(iSCSI), Apr. 2004. RFC 3720, http://tools.ietf.org/html/rfc3720.

[25] A. Muthitacharoen, R. Morris, T. M. Gil, and B. Chen. Ivy: A Read/Write Peer-to-Peer File System. In Proceedings of the 5th Conference on Symposium on Operating Systems Design and Implementation (OSDI), Dec. 2002.