An Intelligent Data Backup And Restore Mechanism For Using ABE

Sowmiya S
Student/M.E-Computer Science and Engineering
P.S.V College of Engineering & Technology
Krishnagiri, India
sowmijohn@gmail.com

Chandra Sekeran S
Associate professor/ Department of Computer Science and Engineering
P.S.V College of Engineering & Technology
Krishnagiri,India
chandrudpi@gmail.com

Abstract— **Data backup and restore operations can be resource-intensive and lead to performance degradation or may require the system to be offline entirely. The backup server is the cloud storage where the backup of all the data is taken and when ever the data is needed it is data is recovered. This will lead to server failure and heavy data loss. Once user is requested to the data server, it is carried by apache server sends the request file server memory, it is not present then forwarded to server disk. Generator compares the file name with corresponding file size and time of last modifications are compared then it is updated in the server as well as backup is taken. We implement Attribute Based Encryption for file access.**

Keywords- ***ABE,BIFS,RPO,FRM,SRN***

I. INTRODUCTION

Data backup plays an important role in the protection of IT system data. By storing copies of system data, backup defends the data against possible loss caused by hardware and software failures, human errors and natural disasters. While data backup is essential for data protection, its deployment can negatively affect system performance and availability. For example, backup execution may bring the system offline(offline backup) and incur additional downtime. Even if the system is accessible during a backup, i.e., by using ―online backup techniques, system performance is still likely to be affected. This is because the backup process may compete with normal service processes for system resources and can be intensive on some types of resources, for example, disk I/O.

In order to reduce the impact of the backup process on system performance and availability, a suitable backup policy needs to be devised which specifies the technique, scope and schedule of the backup. For example, a commonly adopted backup approach is to combine full backups, where all the data is copied to the backup storage, with partial backups [1] which only copy data change that has occurred since an earlier backup. Partial backups can be classified as differential and incremental backups. The former processes data changes since the last full backup and the latter processes changes since the most recent backup (full or incremental). The tradeoff here is that an incremental backup takes the least time to perform, but restoration tends to be slower as it requires all the partial backups since the last full backup.

On the other end of the spectrum, the full backup is slow to perform but restores the system faster. Finally, the differential backup sits in between. Thus, depending on system failure characteristics, a mixed backup strategy may reduce the time spent on backup/recovery and improve system availability and performance. Such a strategy is coupled with a backup schedule that defines the frequencies of full and partial backups. The backup frequencies affect the system recovery point objective (RPO), the maximum time period in which data could be lost due to a system failure. Hence, a tradeoff exists where higher backup frequencies achieve lower RPOs but potentially decreases system availability and/or performance. To devise a suitable backup policy, it is important to properly evaluate system performance and availability under different policies. Our previous paper [2] focused on automated model composition and availability of an offline backup system. In this paper, we extend the previous paper by: 1) considering online backup and workload priorities, and 2) deriving new availability and performance metrics based on more detailed analytical models. The models are developed with a variety of model types including Markov chains, queuing networks and Stochastic Reward Nets (SRN), and capture the details of both the normal system operation and the backup process. Notably, the rejection rate and ratio capture the system availability as perceived by the users.Then investigate these metric values under several backup policies. Our results provide insights into the interactions between the backup

process and the applications running on the system, and the tradeoffs needed to provide data protection with data backup while minimizing the impact on system availability and performance.

This paper is organized as follows, In section 2,we describe some related work. In section 3,we present the system model and data backup are provided. In section 4,proposed and its module description are explained. Finally,we conclude tracability report and test case report in section 5.

II. RELATED WORK

Protecting File Systems a Survey of Backup Techniques[1] This paper presents a survey of backup techniques for protecting file systems.These include such choices as device-based or file-based backup schemes, full vs incremental backups, & optional data compression.Fullfill request management [2]In this paper we introduce the term FRM (Fulfillment Request Management). According to the FRM in a BSS/OSS environment we can use a unified approach to implement a SOA in order to integrate BSS with OSS and handle.Availability Modeling and Analysis for Data Backup and Restore Operations[3] This paper considers technological approaches to meeting disaster recovery needs, and emerging configuration alternatives and technologies that provide cost-effective solutions for disaster recovery and business resilience. This paper does not address traditional tape backup technology or disk-based backup options, but focuses on remote replication approaches.

Disaster Recovery Issues and Solutions Author[4] This paper considers technological approaches to meeting disaster recovery needs, and emerging configuration alternatives and technologies that provide cost-effective solutions for disaster recovery and business resilience. It reviews a variety of remote copy technologies that are widely used for in-region and out-of-region replication, offered by Hitachi and other major storage suppliers. And it outlines cost-effective replication approaches—including two data center and three data center configurations that meet a wide range of business needs.

A Privacy-Protecting File System on Public Cloud Storage[5] With the development of cloud-based systems and applications, a number of major technical firms have started to provide public cloud storage services, and store user data in datacenters strategically positioned across the Internet.To provide strong protection on user data, we design a new file system called BIFS (Bit-Interleaving File System). Focusing on the privacy protection of the on-disk state, BIFS re-orders data in user files at the bit level, and stores bit slices at distributed locations in the storage system.We implement BIFS on the Amazon Simple Storage Service (S3), and examine its performance characteristics. The comparison with several existing network or Internet-based file systems shows that BIFS provides robust file system functions with satisfactory throughput on S3.

Designing Dependable Storage Solutions for Shared Application Environments[6] The costs of data loss and unavailability can be large, so businesses use many data protection techniques,such as remote mirroring, snapshots, and backups, to guard against failures. Choosing an appropriate combination of techniques is difficult because there are numerous approaches for protecting data and allocating resources.

III. PROPOSED SYSTEM

The problem could be overcome by using Attribute Based Encryption algorithm for providing security. User is requested to the data server, it is carried by server sends the request to file server memory, it is not present then forwarded to server disk. Generator compares the file name with corresponding file size & time of last modifications are compared then it is updated in the server as well as backup is taken. In the modification, Attribute Based Encryption (ABE) is used for File Access. Only Authorized Users can Edit the Data, and can Upload the Data. Updated Files has to get Approval for the change from the Owner or Admin, only then the Files are updated.

SYSTEM MODEL

To implement a system successfully, a large number of inter-related tasks need to be carried out in an appropriate sequence. Utilising a well-proven implementation methodology and enlisting professional advice can help but often it is the number of tasks, poor planning and inadequate resourcing that causes problems with an implementation project, rather than any of the tasks being particularly difficult.

3.1. User Enrollment

If the User wants to access the data from the server, they should have an account with that server. Without having an account they aren't able to access the files are view the details. So first the patient will create an account with that server by providing the necessary information like Username, Password, Address , Phone number, etc. Once this information are provided by the user, server will get those information and stored it into the database for future purpose.

3.2. Server Construction

The Server will monitor the entire User's information in their database and verify them if required and the Server will store the entire User's information in their database. Then the Server will establish the connection to communicate with the Users. The Server will update the each User's activities in its database. The Server will authenticate each user before they access the Application. That will prevent the Unauthorized User from accessing the application on the server.

3.3. Backup Server Construction

A server farm or server cluster is a collection of computer servers usually maintained by an enterprise to accomplish server needs far beyond the capability of one machine. Server farms often consist of thousands of computers which require a large amount of power to run At the optimum performance level, a server farm has enormous costs associated with it, both financially and environmentally. Server farms often have backup servers, which can take over the function of primary servers in the event of a primary server failure. Server farms are typically connected with the network switches and routers which enable communication between the different parts of the cluster and the users of the cluster.

3.4. Data Generator/ Verifier

When user requested to the data server it is carried by apache server the request to file server memory.It is not present then forwarded to server disk. Generator compares the file name with corresponding file size and time of last modification are compared then it is updated in the server as well as backup is taken 1.The Generator compares the file list with its local contents and decides which files are necessary to obtain from the file server based on file metadata. After this decision process, the Generator walks the list of files to be transferred and sends each file name with its block checksums to the Sender. 2.If the backup being performed is a full backup, the Sender will directly send the whole requested file to the Receiver; otherwise, it will perform more complicated file block checksum computations and only send file blocks whose checksum values are different from those provided by the Generator.Steps 2 and 3 in the above process will repeat until the Generator finishes walking the file list, which also signals the completion of the backup procedure. Verifier is the Component which checks the previous occurrence of the file with Date/Time and with the Size of the File is being modified.

3.5. Abe Verification Scheme

This provides end-to-end encryption and ABE-based tokens to enable authorization by both authorities and owners and to move policy enforcement from cloud to destinations. With our user-centric approach, owners can take control of their data when it rests in semi entrusted cloud storage. Moreover, with most cryptographic functions delegated from owners to authorities, owners can gain computation power from clouds. All the required auxiliary building blocks and compared the computational weight that each of them adds to the overall performance of this protocol. In particular, single pairing and multi-pairing implementations achieve state-of-the-art time performance at the 126-bit security level.

3.6. Owner Approval System & Updation In Backup Server

This Module implements Attribute based encryption for file access.Only authorized user can edit the data, and can upload the data. Updated files have to get approval for the changes from the owner or admin, only then the files are updated.

3.7 Attribute Based Encryption Algorithm

Attribute-based encryption (ABE) is a relatively recent approach that reconsiders the concept of public-key cryptography. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver's public-key. Identity-based cryptography and in particular identity-based encryption (IBE) changed the traditional understanding of public-key cryptography by allowing the public-key to be an arbitrary string, e.g., the email address of the receiver. ABE goes one step further and defines the identity not atomic but as a set of attributes, e.g., roles, and messages can be encrypted with respect to subsets of attributes (key-policy ABE - KP-ABE) or policies defined over a set of attributes (ciphertext-policy ABE - CP-ABE). The key issue is, that someone should only be able to decrypt a ciphertext if the person holds a key for "matching attributes" (more below) where user keys are always issued by some trusted party.

3.7.1**. Traceability Report**

Attributes /Modules	User Enrollment	Login Form	Upload	Download	Admin Approval
User Name	✓	✓	✓	✓	✓
Password	✓	✓	X	X	✓
Designation	✓	X	✓	✓	X
Year of Experience	✓	X	✓	✓	X
Secret Key	X	X	X	✓	X
Group Key	X	X	✓	X	X
File Name	X	X	✓	✓	✓

Report 3.7.1. Traceability Report

3.7.2. Test Case Report

S.No.	Function	Description	Expected Output	Actual Output	Status
1.	Registration	Registering New User	When New User Registered 3Different Key is Generated and Sent Via E-Mail	User is Registered and Keys Sent Via E-mail	Success
2.	Upload	User File Uploading	Users File must be Uploaded in Server in Encrypted	Uploaded in Server in Encrypted format	Success
3.	Admin Approval	Approval From Admin	Admin has to approve user file to store in disc server	File Moved to Server Disc	Success
4.	Download	Authenticated user Download	File Downloaded once the User Authenticated	File Downloaded	Success
5.	Download	Unauthenticated User Download	File not downloaded	File Not Downloaded	Failed

Report 3.7.2. Test Case Report

IV. CONCLUSION

The project presents an analytical modeling framework in this paper to evaluate the availability and performance of a storage system with periodic data backup. The models could be extended to capture the effects of different backup policies on system metrics under a variety of system configurations and work loads. The recovery of the file can be done by authorize user from the backup server. The models and formulas is to extend existing work on storage backup modeling [4] and provide a basis for future work on more detailed and comprehensive models for storage system operations. The current work adopted a variety of assumptions on

system operation scenarios and parameter settings. The most of the assumptions are reasonable to further improve confidence in model accuracy. To perform model validation using either simulation and experimental data in the future, and obtain information about configurations and operational patterns of systems deployed in practice. It is to apply/extend the technique presented to real systems of much practical interest. When user requested to the data server it is carried by apache server the request to file server memory.It is not present then forwarded to server disk. Generator compares the file name with corresponding file size and time of last modification are compared then it is updated in the server as well as backup is taken 1.The Generator compares the file list with its local contents and decides which files are necessary to obtain from the file server based on file metadata. After this decision process, the Generator walks the list of files to be transferred and sends each file name with its block checksums to the Sender. 2.If the backup being performed is a full backup, the Sender will directly send the whole requested file to the Receiver; otherwise, it will perform more complicated file block checksum computations and only send file blocks whose checksum values are different from those provided by the Generator.Steps 2 and 3 in the above process will repeat until the Generator finishes walking the file list, which also signals the completion of the backup procedure. Verifier is the Component which checks the previous occurrence of the file with Date/Time and with the Size of the File is being modified.

ACKNOWLEDGMENT

The authors would like to thank Prof. Lorenzo Alvisi of the University of Texas at Austin for his constructive comments on preliminary versions of this paper

REFERENCES

[1] Chervenak .A, Vellanki.V, and Kurmas.Z,“Protecting the File Systems: A Survey of Backup Techniques”, Proc. Joint NASA and IEEE Mass Storage Conf., 2012.
[2] CAARCserve,http://www.arcserve.com/solutions/backup-andarchiving.aspx,2012.
[3] Cherkasova .L, Zhang.A, and Li.X, “Design of Efficient Backup Scheduling, Proc. Int’l Conf. Network and the Service Management”(CNSM), pp. 118-125, 2010.
[4] EMCBackupAdvisor,http://www.emc.com/products/detail/software/backupadvisor.htm, 2013.
[5] Gaonkar.S, Keeton.K, Merchant.A, and the Sanders.W.H,“Designing Dependable Storage Solutions for Application for shared Environments”,Proc.Int’l Conf. Dependable Systems and Networks (DSN’06), pp. 371-382, 2006.
[6] K. Keeton, C. Santos, D. Beyer, J. Chase, and J. Wilkes, “Designing for Disasters, Proc. Third Conf. File and Storage Technologies” (FAST ’04), pp. 59-72, 2004.
[7] Keeton.K and Merchant.A,“A Framework for Evaluating Storage System Dependability”,Proc. Int’l Conf. Dependable Systems and Networks (DSN’04), pp. 877-886, 2004.
[8] E. Rozier, W. Sanders, P. Zhou, N. Mandagere, S. Uttamchandani, and M. Yakushev, “Modeling the Fault Consequences Deduplication,” Proc.IEEE Symp. Reliable Distributed Systems, 2011.
[9] D. Geer, “Reducing the Storage Burden via Data De-Duplication,” Computer, vol. 41, no. 12, pp. 15-17, Dec. 2008.
[10] K. Renuga, S. Tan, Y. Zhu, T.C. Low, and Y. Wang, “Balanced and Efficient Data Placement and Replication of the Strategy for Distributed Backup Storage Systems,”Proc.Int’l Conf.of the Computational Science and the Engineering (CSE’09), pp. 87-94, 2009.
[11] H. Wang, K. Zhou, and L. Yuan, “Fault-Tolerant Online Backup Service: Formal Modeling and Reasoning,” Proc. IEEE Int’l Conf.Networking, Architecture and Storage (NAS), pp. 452-460, 2009.