International Journal of Current Research and Modern Education (IJCRME)
ISSN (Online): 2455 - 5428
(www.rdmodernresearch.com) Volume I, Issue I, 2016

FLEXIBLE IN FILE RETRIEVING USING LT BASED REQUEST IN CLOUD STORAGE

R. Vinoth Kumar* & K. Adlin Suji**

* PG Scholar, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

** Associate Professor, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

Abstract:

File System is one of the most popular applications in different environment. But Effective file handling on the cloud storage is very less. It leads the data leakage. For this purpose to split the documents into different parts and then stored into different node servers are clusters of cloud server. In this environment files are accessed by FIFO method. But in FIFO model data users are want to wait more times for getting a file from cloud server, this is the biggest problem in existing system. To Avoid This implement the easier file accessing method by using Luby-Transform.With this file parts are downloaded as crossly to file requested user. So FIFO method based file accessing is totally avoided by using this method.

Introduction:

Cloud storage systems provide a scalable online storage solution to end users who require flexible amount of storage space but do not wish to own and maintain storage infrastructure. Compared with traditional data storage, cloud storage has several advantages. For example, end users can access their data anywhere through Internet without bothering about carrying physical storage media. Also, different users can collaboratively contribute to the data stored in cloud storage with permission from the data owner. Due to its high popularity in industry, cloud storage has been a hot topic in cloud computing community generally, cloud storage systems rely on thousands of storage nodes. Content is often fragmented and distributed into a set of storage nodes. To offer high reliability and availability of storage services, redundancy of contents may be employed. Fragments of contents may be simply replicated and stored in different storage nodes to achieve redundancy. Other than naive replication proposed two QoS aware data replication algorithms to reduce storage cost while maintaining QoS for the applications. Analyzed two schemes for redundancy: replication and erasure coding, concluding that erasure-coded systems can provide higher availability with lower bandwidth and less storage space. Since then, there are plenty of works on designing erasure codes for storage systems, and they mainly focus on the reliability and availability of storage systems. Motivated by the great success of rate less codes or fountain codes, which have very low decoding complexity and can generate infinite number of encoded packets, some works have applied the popular rate less code or LT code, into cloud storage systems and have achieved promising performance. The main advantage for a rate less code based cloud storage system is that it significantly simplifies the challenging content placement and content recovery problems that need to be addressed in erasure code based systems. This is because a rate less code based system can potentially generate infinite number of encoded packets to be placed across the storage system and to replace those unavailable packets due to node failure. However, the disadvantage of a rate less code based cloud storage system is that it incurs longer data retrieval delay, since it requires more coded packets due to its uncertainty in decoding different from its erasure code counterpart which is deterministic in decoding.

Modules Description:

- ✓ Add data to the cloud
- ✓ Process of Storage node
- ✓ Request type
- ✓ Packet Receiving
- ✓ Key Request
- ✓ Download the file

Add Data to the Cloud:

First data owner's will register and upload the all the data to the cloud while uploading the all the file main file would be encrypt and upload it to the cloud. Every user's can upload their data from their login only.

Process of Storage Node:

In this module the uploaded original data will be in the cloud and also that will be splited in to three file and will be encrypted and also will be store in to the storage node main file would be in the cloud and the splited files only will be in the storage nod's, all the splited file's will be in encrypted format only.

Request Type:

While giving the request user can select the request type like normal request and the LT request, in normal request every user want wait at the queue and they wanted to download. Compare with normal file request. LT based request little bit faster than LT based request. User no need to wait at the queue to access their file and for the normal file service will be coming from the cloud directly and for the LT request service will coming from storage nodes.

Packet Receiving:

In this module user will receive the files as packets, while uploading the files in the cloud also it will get automatically three files, while the retrieval time user has to enter the secret key if the secret key matched user will receive the all the files as packet .

Key Request:

In this module user will view the all the files publicly and they can give the request for needed files. While giving the request user has to select the request type, after selecting the type that will be directly goes to portal control, portal will check all the request and will forward into the particular data owner, that data owner will forward the key to the data user.

Download the File:

In this module after getting all the permission and key from the data owner user's can download their favorite file. If normal request user file will get delay to download, and if LT request only less amount of traffic will occur so files will get fast download comparing with normal request.

Existing System:

In existing system there were many issues in file retrieval, if any user want to download any files from the cloud storage it will get time to download in most of the time when n number of user try to get download the same file at the same time server will get hang and also it will take long time to download.

Disadvantages

- ✓ Server may hang
- ✓ Delay while downloading

Proposed System:

In proposed we have introduced LT (Luby Transform) based request, by using this techniques. we can overcome the existing problems, and the while n-number of

user try to get the download the same file at the same time , file will get the download from the server node , all the file would come from the splited server nodes. While uploading the file into the cloud that will automatically will get splited into three files.

Advantages:

- ✓ Server will not get hang
- ✓ Optimizing Delay timing while downloading

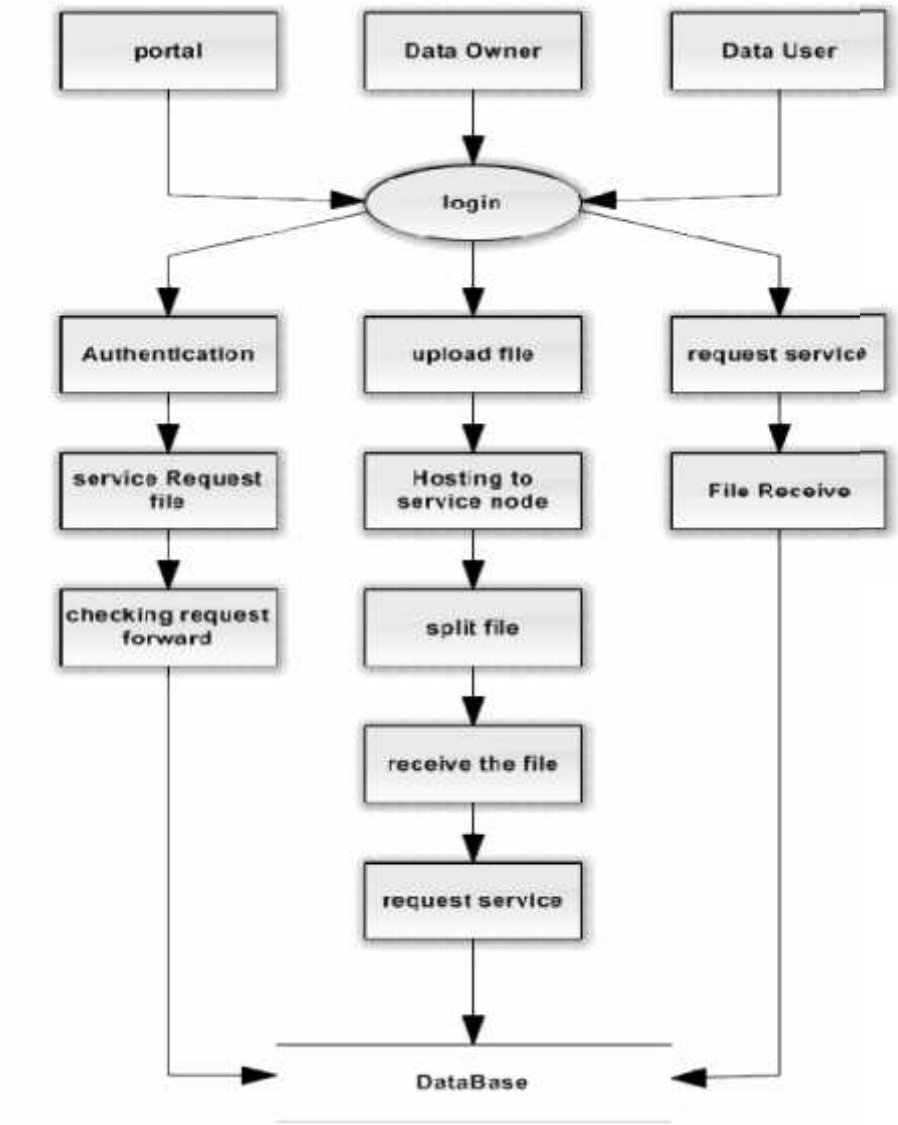


Figure 1: Data Flow Diagrams

Experimental Result:

A software application in general is implemented after navigating the complete life cycle method of a project. Various life cycle processes such as requirement analysis, design phase, verification, testing and finally followed by the implementation phase result in a successful project management. System implementation is an important stage of theoretical design is turned into practical system.

Implementation Procedure:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning,

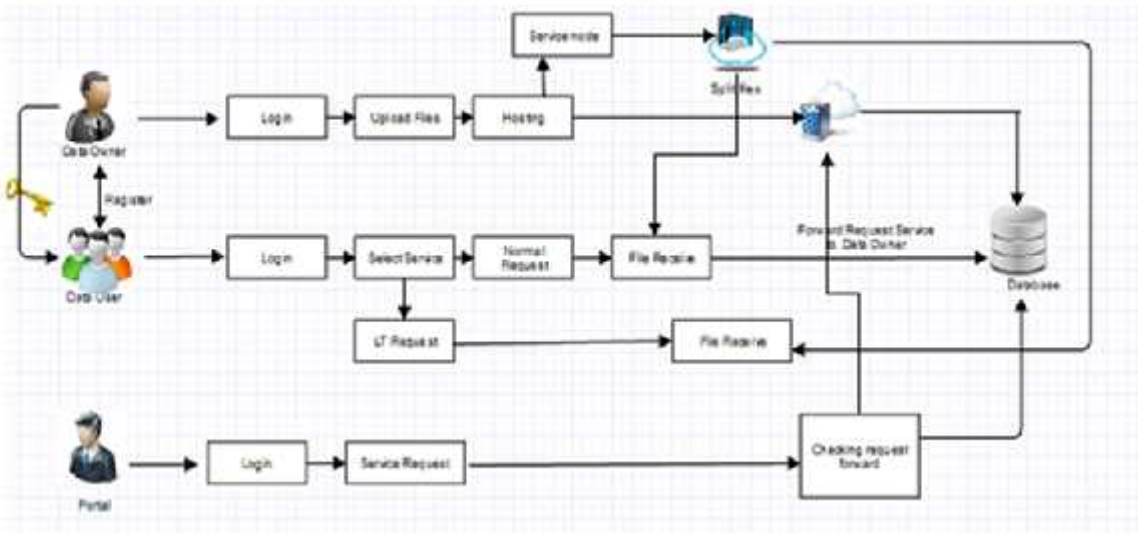
investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user.

The system that has been developed is accepted and proved to be satisfactory for the user and so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly. The final stage is to document the entire system which provides components and the operating procedures of the system.

System maintenance is the process of after the software execution. The primary goal of implementation is to write source code and internal implementation. So that conformance of code to its specification can be easily verified, So that debugging, testing and modification are eased. The source is developed with clarity, simplicity and elegance.

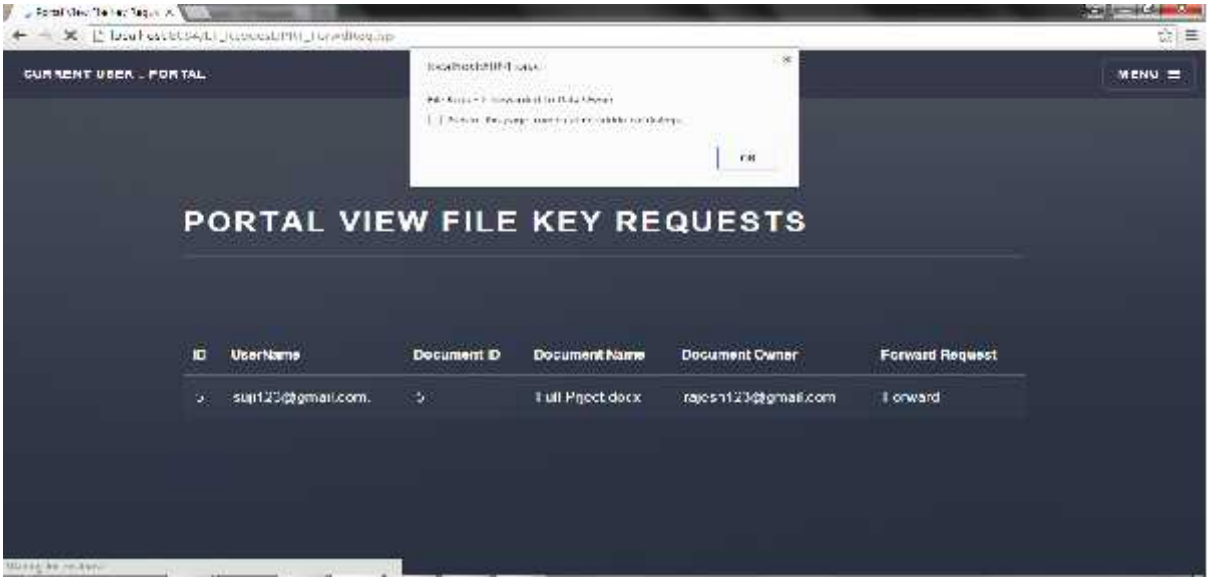
The coding is done in a modular fashion giving such importance even to the minute detail so, when hardware and storage procedures are changed or now data is added, rewriting of application programs is not necessary. To adapt or perfect use must determine new requirements, redesign generate code and test exiting software/hardware. Traditionally such task when they are applied to an existing program has been called maintenance.



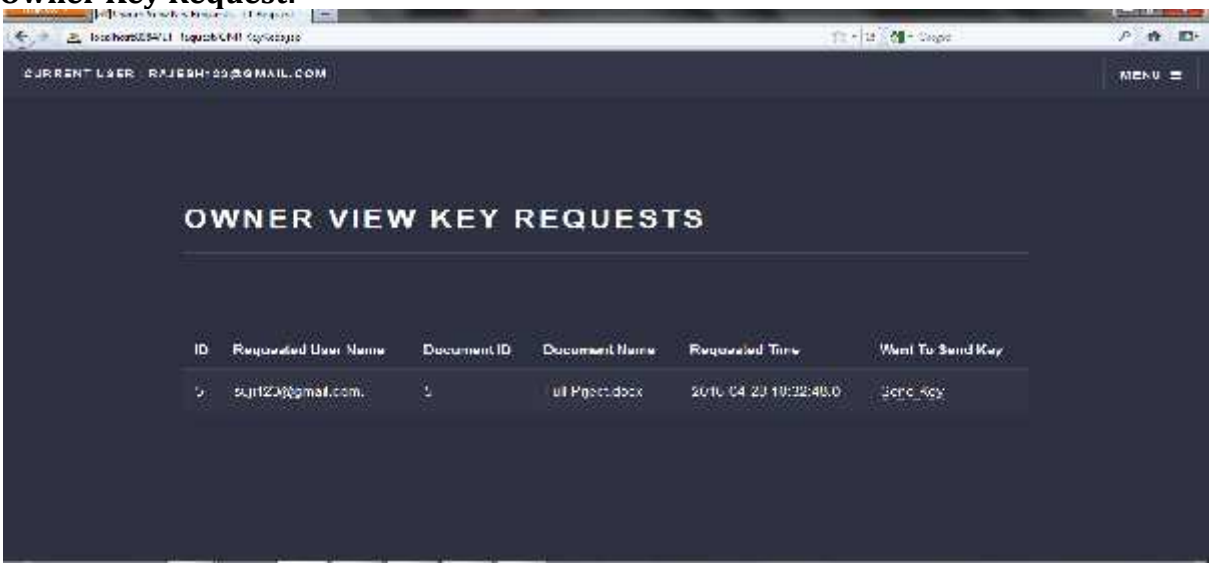
System maintenance is the process of after the software execution. The primary goal of implementation is to write source code and internal implementation. So that conformance of code to its specification can be easily verified, So that debugging, testing and modification are eased. The source is developed with clarity, simplicity and elegance.

The coding is done in a modular fashion giving such importance even to the minute detail so, when hardware and storage procedures are changed or now data is added, rewriting of application programs is not necessary. To adapt or perfect use must determine new requirements, redesign generate code and test exiting software/hardware. Traditionally such task when they are applied to an existing program has been called maintenance.

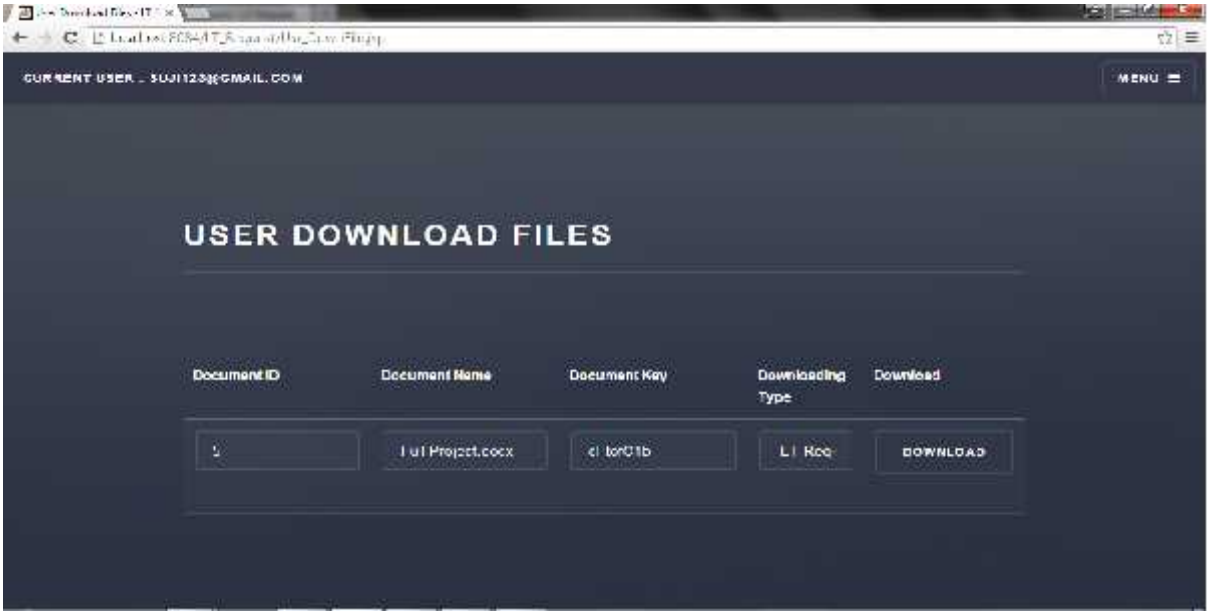
File Forward to Data Owner:



Owner Key Request:



User Download File:



Conclusion and Future Enhancement:

Finally optimal retrieval schemes given requirements on success decidability. Our numerical results suggest a fundamental tradeoff between the file-retrieval delay and the target probability of successful file decoding, and that the file-retrieval delay can be significantly reduced by optimally scheduling packet requests in a multi-stage fashion.

References:

1. Amazon S3. http://aws.amazon.com/s3/.
2. EMC Atmos Online Storage. http://www.atmosonline.com/.
3. L. Heilig and S. Voss, "A scientometric analysis of cloud computing literature," IEEE Transactions on Cloud Computing vol. 2, no. 3, pp. 266–278, July 2014.
4. J.-W. Lin, C.-H. Chen, and J. Chang, "Qos-aware data replication for data-intensive applications in cloud computing systems," IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 101–115, Jan 2013.
5. H. Weatherspoon and J. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," Peer-to-Peer Systems, pp. 328–337, 2002.
6. J. Hafner, "Weaver codes: highly fault tolerant erasure codes for storage systems," in Proc. the 4th conference on USENIX Conference on File and Storage Technologies. USENIX Association, 2005, pp. 16–16.
7. J. Plank and L. Xu, "Optimizing cauchy Reed-Solomon codes for fault-tolerant network storage applications," in IEEE Int. Symp. Network Computing and Applications. IEEE, 2006, pp. 173–180.
8. C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," in IEEE Int. Symp. Network Computing and Applications. IEEE, 2007, pp. 79–86.
9. F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in Proc. IEEE INFOCOM. IEEE, 2010, pp. 1215–1223.
10. H. Xia and A. Chien, "Robustore: a distributed storage architecture with robust and high performance," in Proceedings of the 2007 ACM/IEEE conference on Supercomputing. ACM, 2007, p. 44.