ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

# Efficient Scheduling for MultiBlock Updates in file Distributed based Storage Systems

P Mallika[1], M Aanandhi[2], K Gugapriya[3], P Kowsalya[4], S Poongodi[5]

[1] Assitant Professor Department of CSE, Jai Shriram Engineering College, Tirupur, Tamilnadu, India

[2,3,4,5] Students Department of CSE, Jai Shriram Engineering College, Tirupur, Tamilnadu, India

**ABSTRACT:** High-speed networks and ubiquitous Internet access become available to users for access anywhere at any time. Cloud computing is a concept that treats the resources on the Internet as a unified entity, a cloud. Cloud storage is a model of networked online storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Hosting companies operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them. The data center operators, in the background, virtualizes the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers. Data robustness is a major requirement for storage systems. There have been many proposals of storing data over storage servers. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. A decentralized erasure code is suitable for use in a distributed storage system. We construct a secure cloud storage system that supports the function of secure data forwarding by using an AES and Proxy re encryption. In this model initial phase owner will upload the data with AES Encryption. Next phase, inside of cloud again the data has divided into small pieces, for this process we will apply a dividing key. Data will place in different storage lactations. The information of data storage will monitor by a unique data distributors. If the valid user accessing the data cloud will retrieve the data as reversible manner.

**KEYWORDS:** Cloud Service Provider (CSP), Load Balancing.

## I. INTRODUCTION

The Paper describes Cloud computing provides an enormous amount of virtual storage to the users. Cloud storage mainly helps to small and medium scale industries to reduce their investments and maintenance of storage servers. Cloud storage is efficient for data storage. Users' data are sent to the cloud is to be stored in the public cloud environment. Data stored in the cloud storage might mingle with other users' data. This will lead to the data protection issue in cloud storage. If the confidentiality of cloud data is broken, then it will cause loss of data to the industry. Security of cloud storage is ensured through confidentiality parameter. To ensure the confidentiality, the most common used technique is encryption. But encryption alone doesn't give maximum protection to the data in the cloud storage. To have efficient cloud storage confidentiality, this paper uses encryption and obfuscation as two different techniques to protect the data in the cloud storage. Encryption is the process of converting the readable text into unreadable form using an algorithm and a key. Obfuscation is same like encryption. Obfuscation is a process which disguises illegal users by implementing a particular mathematical function or using programming techniques. Based on the type of data, encryption and obfuscation can be applied. Encryption can be applied to alphabets and alphanumeric type of data and obfuscationcan be applied to a numeric type of data. Applying encryption and obfuscation techniques on the cloud data will provide more protection against unauthorized usage. Confidentiality could be achieved with a combination of encryption and obfuscation.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

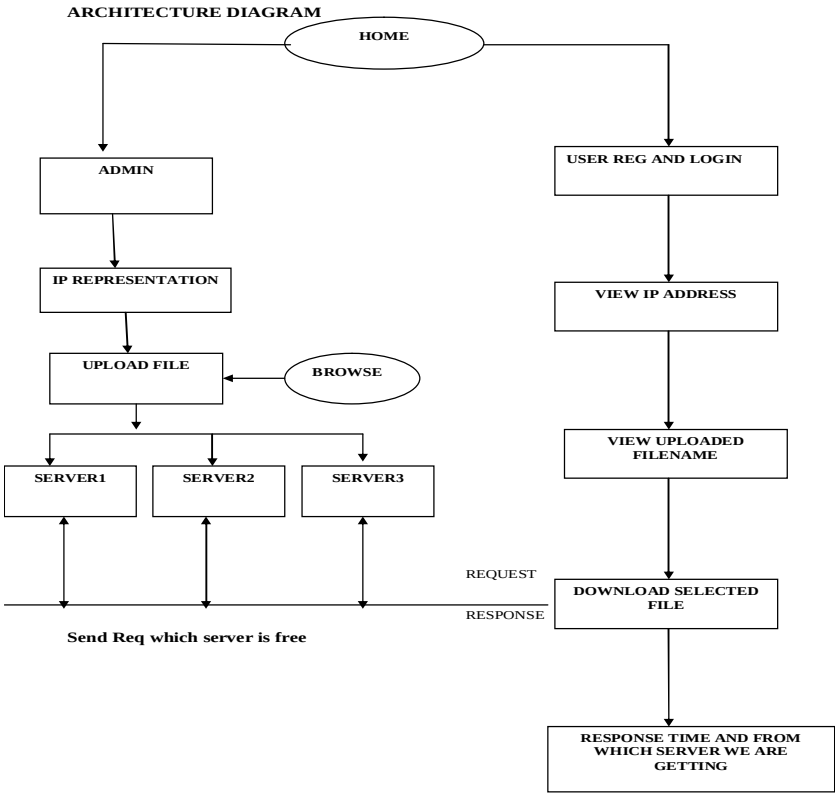
## II. EXISTING SYSTEMS

The popular file system for networked computers is the Network File System (NFS). It is a way to share files between machines on a network as if the files were located on the client's local hard drive. Scalable distributed file system that manages a collection of disks on multiple machines as a single shared pool of storage. The machines are required to be under a common administrator and be able to communicate securely.This poses a huge barrier to their wide adoption in online applications (e.g., social networks) where the latency is a critical concern.

## III. PROPOSED SYSTEM

In our Proposed System to implement the Round robin Scheduling (Algorithm) is to overcome the problem of existing system. This model will reduce the latency and increase the throughput than the existing system.
The Round robin model is very helpful for load balancing of the server. This will reduce the load of the server while the server is being busy. These are the advantages of our proposed system. This scheme can minimize the average latency by about 40 percent and improve throughput across a variety of workloads.

**ARCHITECTURE:**




**MODULES:**

- Authentication module
- IP Address Representation
- Sub server
- Load balancer

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

- Security
- Response time

**Authentication Module:**
The authentication module is to register the new users and previously registered users can enter into our project. The admin only can enter and do the uploading files into the servers. After login by every user and the admin the sql server checks the login id and password is valid or not. If the login is not valid it displays that the login is not correct.

**IP Address Representation Module:**
The IP Address Representation module is to give the IP addresses which we are going to assign those as servers. The enter and view IP addresses from this module. In load balancing system we can connect the three servers [system]. The connection has to be represented by the IP Address representation only.

**Sub Server:**
The main server divided in to three sub servers. IP addresses are assigning to those sub servers. the enter IP address, from the load balancing system we can connect the three sub servers [system]. The connection has to be represented by the IP Address representation only. Manager upload files in the system, those files are also in three sub servers.

**Load balancer:**
In this load balancing get benefits of distributing the workload includes increased resource utilization ratio which further leads to enhancing the overall performance thereby achieving maximum client satisfaction. The Authentication users can enter into the upload page and can view the file name which the manager stored into the servers. The user can select the file from the list and can download from the server which is in idle state. We will get the response time and from which server we are getting the file. Finally we can get the decrypted file through using key pairs.

**Security:**
Authentication, Admin store the files in encrypted format using RSA algorithm. User can view the files after decryption only. Only Admin allow, user can view the encrypted files.

**Response Time:**
Response time is the total time it takes from when a user makes a request until they receive a response. Here, user can calculate the response time for the whole system .Latency+ processing time=response time. In our project amount of time takes for downloading files from system.

**AES ALGORITHM:**
The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES. A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.
The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES Provide full specification and design details
- Software implementable in C and Java

**Operation of AES:**
AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).Interestingly, AES performs all its computations on bytes rather than bits. Hence,

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

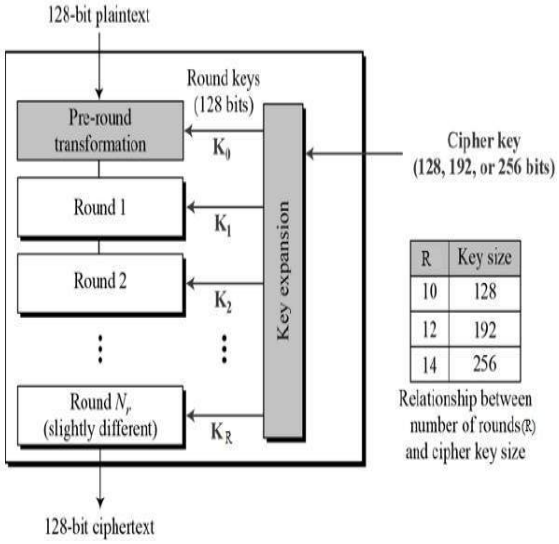
International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

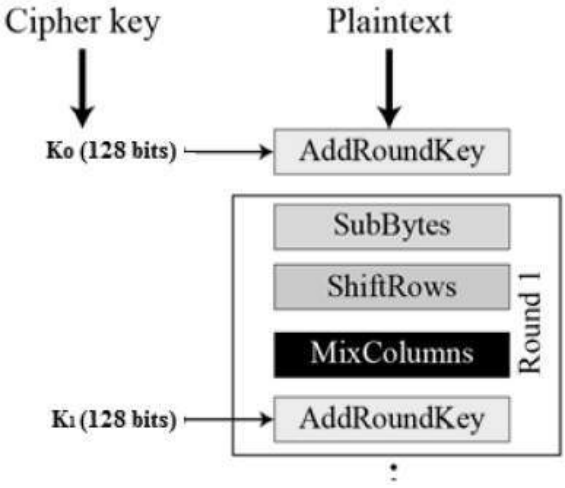
AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.The schematic of AES structure is given in the following illustration −




| R | Key size |
|---|---|
| 10 | 128 |
| 12 | 192 |
| 14 | 256 |

**Encryption Process:**

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below −




**Byte Substitution (SubBytes)**:

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

**Shiftrows:**

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows −

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 3, March 2019

- First row is not shifted.
- Second row is shifted one (byte) position to the left.
- Third row is shifted two positions to the left.
- Fourth row is shifted three positions to the left.
- The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

.

## IV. IMPLEMENTATION OF PROPOSED SYSTEM

Once the system has been designed, the next step is to convert the designed one in to actual code, so as to satisfy the user requirements as expected. If the system is approved to be error free it can be implemented. Implementation includes proper training to end-users. The implemented software should be maintained for prolonged running of the software. When the initial design was done for the system, the department was consulted for acceptance of the design so that further proceedings of the system development can be carried on. After the development of the system a demonstration was given to them about working of the system. The aim of the system illustration was to identify any malfunctioning of the system. Initially the system was run parallel with manual system. The system has been tested with data and has proved to be error-free and user-friendly. Training was given to end-user about the software and its features.

## V. CONCLUSION

Our proposal strives to balance the loads of nodes and reduce the demanded movement cost as much as possible, while taking advantage of physical network locality and node heterogeneity. In the absence of representative real workloads (the distributions of file chunks in a large scale storage system) in the public domain, we have investigated the performance of our proposal and compared it against competing algorithms through synthesized Probabilistic distributions of file chunks.

## REFERENCES

[1]B. Krishnamachari, Networking Wireless Sensors. Cambridge, U.K.: Cambridge Univ. Press, Dec. 2005.
[2] R. Shorey, A. Ananda, M. C. Chan, and W. T. Ooi, Mobile, Wireless, Sensor Networks. Piscataway, NJ, USA: IEEE Press, Mar. 2006.
[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Commun. Mag., vol. 40, no. 8, pp. 102–114, Aug. 2002.
[4] W. C. Cheng, C. Chou, L. Golubchik, S. Khuller, and Y. C. Wan, "A coordinated data collection approach: Design, evaluation, and comparison," IEEE J. Sel. Areas Commun., vol. 22, no. 10, pp. 2004– 2018, Dec. 2004.
[5] K. Xu, H. Hassanein, G. Takahara, and Q. Wang, "Relay node deployment strategies in heterogeneous wireless sensor networks," IEEE Trans. Mobile Comput., vol. 9, no. 2, pp. 145–159, Feb. 2010.
[6] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in Proc. 7th ACM Conf. Embedded Netw. Sensor Syst., 2009, pp. 1–14.
[7] E. Lee, S. Park, F. Yu, and S.-H. Kim, "Data gathering mechanism with local sink in geographic routing for wireless sensor networks," IEEE Trans. Consum. Electron., vol. 56, no. 3, pp. 1433– 1441, Aug. 2010.