**FINDING ABNORMAL STORAGE FILE TRANSFER OR COPY**

*1Punnam Dikshitha Reddy , 2Semsani Srividya ,3 Adepu Shravani,*

*4DR.G.Ramkrishna Reddy*

# *1,2,3bachelor Degree, Cse,* *Jayamukhi Institute Of Technological Sciences Narsampet, Warangal.*

*4Dr.G.Ramkrishna Reddy, Associate Professor, bachelor Degree, Cse, Jayamukhi Institute Of Technological Sciences Narsampet, Warangal.*

**ABSTRACT:**

In recent years, cloud storage has experienced a rapid increase in popularity. Although cloud storage has numerous benefits, like flexibility and convenience, users sometimes have no idea where their data is actually located. This restriction can make consumers less confident and less inclined to trust the storage provider, or it might even make cloud storage inappropriate for data with precise location needs. We suggest a system called LAST-HDFS that incorporates the Location-Aware Storage Technique (LAST) into the free and free to use Hadoop distributed file system (HDFS) in order to solve this problem. In order to identify possibly unlawful transfers in the cloud, the LAST-HDFS system continually monitors transfers of files and enforces location-aware file permissions. Here, attempts to transmit private information outside of the ("legal") restrictions imposed by the file owner and its regulations are referred to as "illegal transfers." Our core algorithms maximise the likelihood of keeping data items with identical privacy preferences in the same location and describe file transfers across nodes as a weighted network. We provide each cloud node a socket monitor so it can keep track of the communication between them in real time. Our technology determines the likelihood that a particular transfer is unlawful based on the real-time data transfer data that socket monitors record. In order to show the usefulness and efficiency of our suggested system, we constructed our recommended framework and conducted a thorough experimental assessment in a sizable genuine cloud environment.

**INTRODUCTION:**

The need for cloud storage has grown dramatically along with cloud computing's rising popularity.

The use of cloud storage and cloud computing are no longer solely used by computer companies; instead, common enterprises and even end users are benefiting from the vast capabilities that cloud services may offer. Users of cloud storage give up ownership of their data while benefiting from its flexibility and convenience, and they frequently struggle to find their real data;

This could be local, national, or even international. For cloud users (such as institutions) that store important information (such as medical records) that must adhere to legal requirements to remain inside specific geographic limits and borders, a lack of geographical control may result in privacy violations. Another situation where this issue occurs is when governmental organisations mandate that all data be kept in the nation in which the authority in question is based; this challenge has seen issues with CSPs (cloud service providers) covertly moving data abroad or being acquired by foreign firms.

For instance, Canadian legislation mandates that personally identifiable information be maintained there.

The Amazon Cloud, a massive cloud infrastructure, has more than 40 zones spread out globally [1], making it exceedingly difficult to assure obedience to regulatory compliance. Likewise Hadoop, which was formerly maintained as a distributed file system that was geographically constrained, is now widely implemented across several locations (see Facebook Prism [2] or a recent patent [3]).

To date, various tools have been proposed to help users verify the exact location of data stored in the cloud [4]–[6],with emphasis on post-allocation compliance. However, recent work has acknowledged the importance of a proactive location control for data placement consistent with adopters’ location requirements [4], [7], [8], to allow users to have stronger control over their data and to guarantee the location where the data is stored.

In this study, we hack into the Apache Hadoop Distributed File System (HDFS), among the most extensively used online storage systems, and create LAST-HDFS, an improved HDFS system. In order to provide location-aware file allocations or file transfer surveillance, the LAST-HDFS expands the abilities of HDFS. Particularly, LAST-HDFS offers the following new features:

(i) constantly requires that data be loaded and stored with location in mind by allocating data nodes in accordance with user preferences. The experimental results are reported in Section 6. Section 7 then evaluates relevant work stored in a secure cloud.

The report is finally concluded in Section 8, which also identifies potential future research areas.

**RELETED WORK:**

We give some basic knowledge on the Hadoop Distributed File System (HDFS), the system's foundation, in this part. An crucial part of the free Hadoop software is HDFS [10]. In order to facilitate distributed data storage and access by applications built on top of it with high throughput and fault tolerance, HDFS is a distributed file system intended to run on commodity hardware. It has a master-slave architecture with one name node serving as the master node and several data nodes serving as the slave nodes.

The namenode controls file accesses and oversees file system meta-data management. Clients submit read/write requests to the datanodes, which then carry out the requested read/write operations on disc blocks under the direction of the namenode. In the sections that follow, we quickly go through the load-balancing and data storage strategies used by the present HDFS because our suggested solution would change these two features to provide location-aware storage.

HDFS's Write Mechanism

To upload a file to HDFS, a data owner (client) must first send a write request to the namenode requesting it to create a new file in HDFS. The client will start publishing data to the stream where data is divided into packets after the namenode grants the request. A data block from the file that will be written to the data nodes is represented by each packet. A list of potential data nodes will be sent to the client by the name node once a separate thread in the client receives a packet and contacts it.

The client will then send a write packet to the first data node in the list, which will store the data block.

The information from that block will then be copied in a pipeline fashion to the subsequent data nodes in the list.

3.2 HDFS Load Balancing

When a new data node is introduced to the cluster or the disc space of certain data nodes is full, load balancing is crucial to the overall performance of HDFS systems. A Hadoop balancer utility enables a cloud administrator to balance the HDFS cluster's use of disc space. The following is a description of the load balancing procedure:

1) Based on their data block consumption statistics, the balancer divides all the datanodes into two groups: (1) underutilised node group and (2) overutilized node group.

2) To create a pair of nodes whose load will be balanced by sending a specified quantity of data from one to the other, the balancer randomly chooses one datanode from each group.

3) The balancer transfers the data from the over-utilized datanode to the under-utilized datanode in the same pair by randomly selecting a list of data blocks.

4) The balancer repeats the first three phases until the cluster's datanodes all reach a specific level of utilisation, or the system reaches balanced load.

**EXISTING SYSTEM:**

The existing system focuses on integration systems at web scale that automate the matching of records from different sources, identify true matching records, and transform them into a standard record for user consumption. The system addresses the challenges of record matching (duplicate record detection, record linkage, etc.) and truth discovery (fact finding, truth finding) in data fusion. However, it does not provide detailed strategies for record normalization.

Disadvantages: The truth discovery problem and record matching problem are not explicitly addressed, leading to potential issues in data accuracy and reliability.

**PROPOSED SYSTEM:**

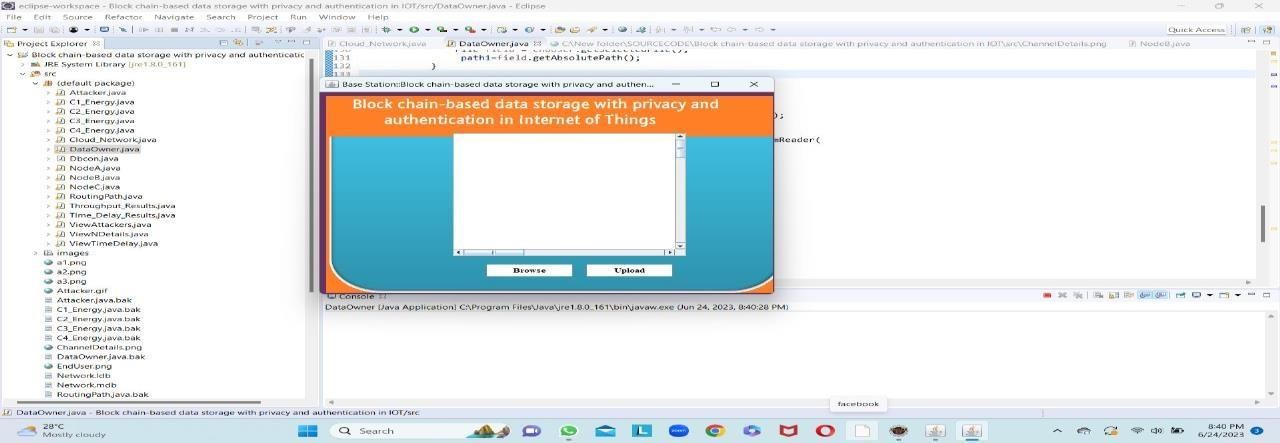
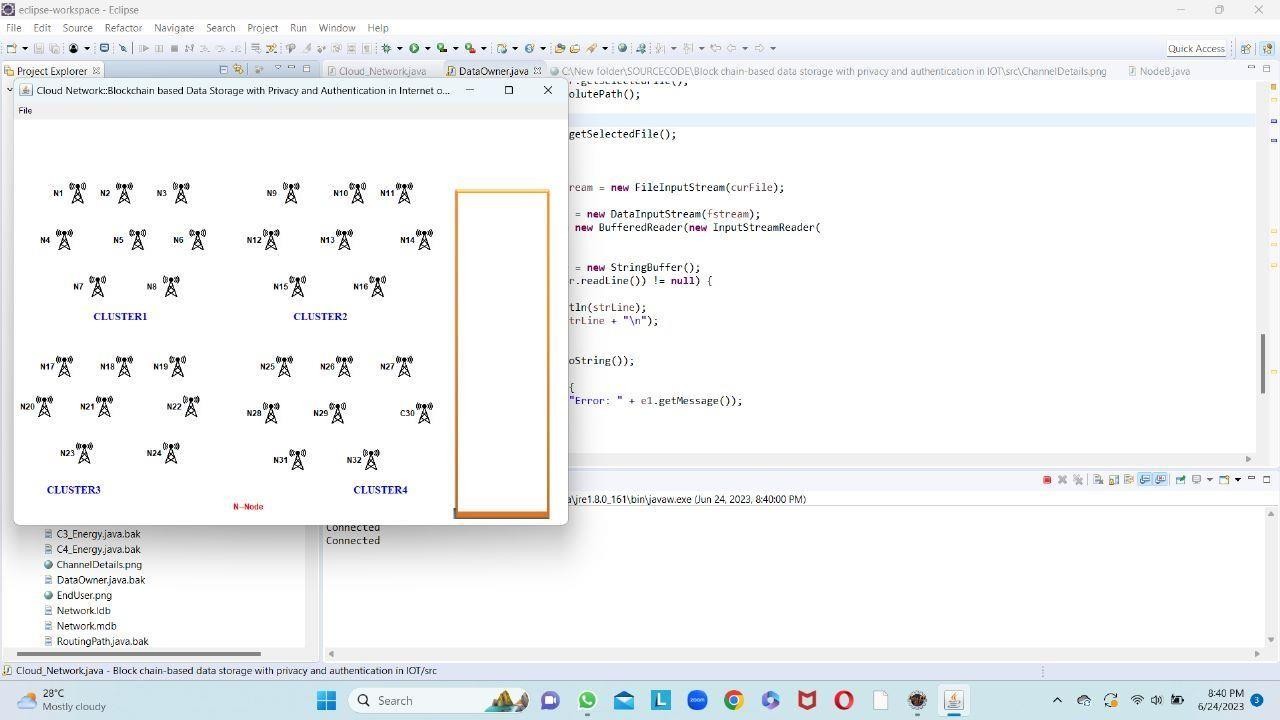
In the proposed system, we introduce a comprehensive framework for record normalization, offering three levels of granularities and construction methods for normalized records. This framework allows for flexibility and the easy addition of new strategies. Our proposed strategies include frequency, length, centroid, and feature-based approaches, as well as more advanced methods like (weighted) Borda that utilize result merging models from information retrieval.

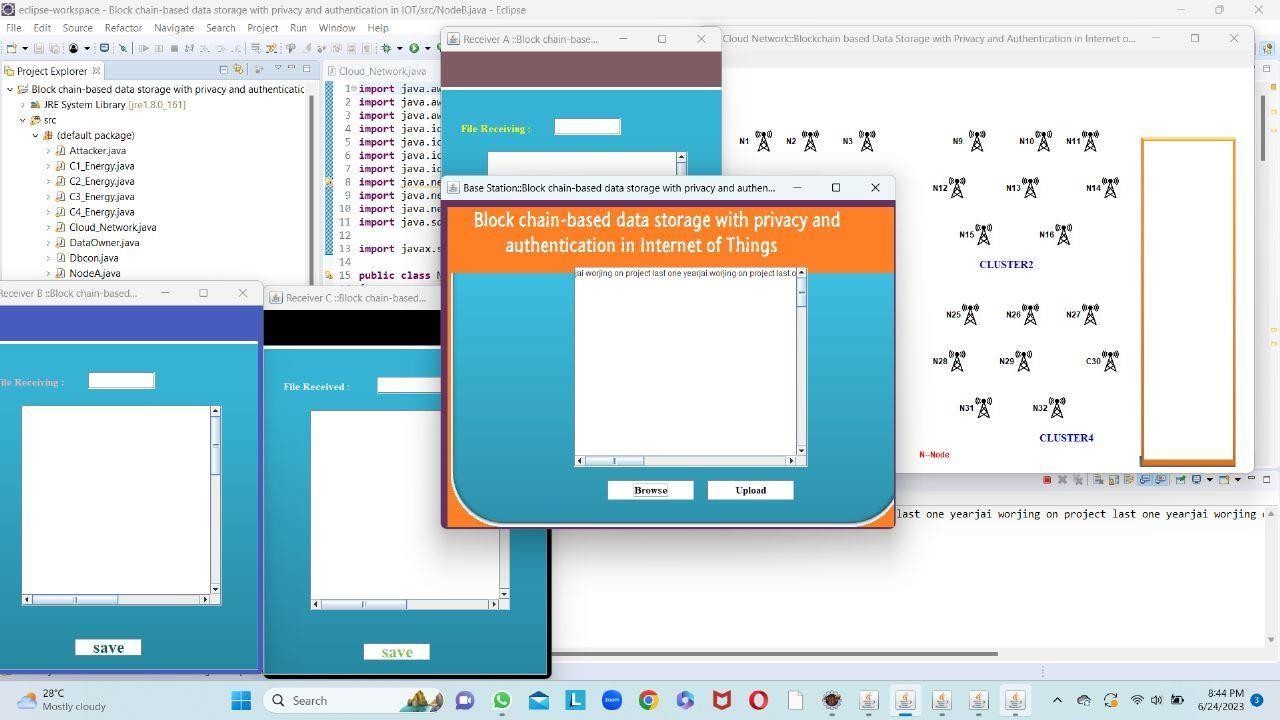
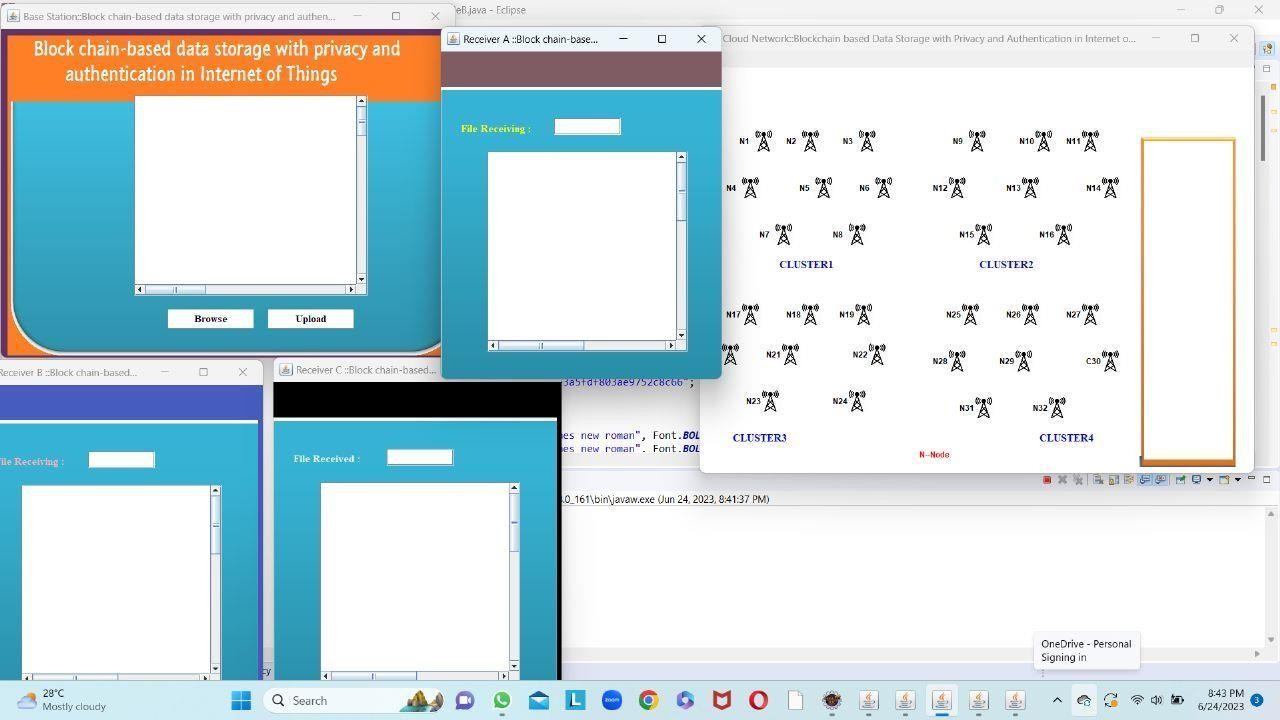
Advantages:

* High Accuracy: The proposed system aims to improve the accuracy of record normalization by implementing advanced strategies and algorithms.
* Best Performance: By comparing different normalization strategies, including the proposed weighted-Borda-based approach, we aim to achieve the best performance in generating normalized records.
* Analysis of Record and Field Level Normalization: Our system provides a comprehensive analysis of both record and field level normalization, addressing specific challenges at each level.

Improved Normalized Field Values: We introduce heuristic rules for acronym expansion and value component mining, resulting in significantly improved normalized field values.

**RESULT:**





**CONCLUTION :**

In this study, we develop a brand-new LAST-HDFS system on top of the current HDFS to handle the issue of cloud data placement control. Policy-driven file loading is supported by LAST-HDFS, enabling location-aware storage in cloud locations.

It also makes sure that the location policy is applied independently of data replication and load balancing procedures that can influence policy compliance, which is even more crucial. In particular, an effective LP-tree and Legal File Transfer graph were created to aid in efficiently allocating files with similar location preferences to the best cloud nodes, increasing the likelihood of spotting unlawful file transfers. Both a large-scale simulated cloud environment and a genuine cloud testbed have been used by us to undertake comprehensive experimental experiments.

The usefulness and efficiency of the suggested LAST-HDFS system have been demonstrated by our experimental findings.

We intend to incorporate more complex regulations in the future to capture privacy concerns beyond geography. To hasten policy comparison and node selection for freshly uploaded files, we will use a more complex policy analysis method [21] and calculate the integrated policy as the representative policy [22] for each node. Additionally, we want to use Intel SGX technology to protect socket monitors from being corrupted.

**REFERENCES:**

[1] Amazon, “Aws global infrastructure,” in https://aws.amazon.com/aboutaws/global-infrastructure/, 2017.

[2] C. Metz, “Facebook tackles (really) big data with project prism,” in https://www.wired.com/2012/08/facebook-prism/, 2012.

[3] K. V. SHVACHKO, Y. Aahlad, J. Sundar, and P. Jeliazkov, “Geographically-distributed file system using coordinated namespace replication,” in https://www.google.com/patents/WO2015153045A1?cl=zh, 2014.

[4] C. Liao, A. Squicciarini, and L. Dan, “Last-hdfs: Location-aware storage technique for hadoop distributed file system,” in IEEE International Conference on Cloud Computing (CLOUD), 2016.

[5] N. Paladi and A. Michalas, ““one of our hosts in another country”: Challenges of data geolocation in cloud storage,” in International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014, pp. 1–6.

[6] Z. N. Peterson, M. Gondree, and R. Beverly, “A position paper on data sovereignty: The importance of geolocating data in the cloud.” in HotCloud, 2011.

[7] A. Squicciarini, D. Lin, S. Sundareswaran, and J. Li, “Policy driven node selection in mapreduce,” in 10th International Conference on Security and Privacy in Communication Networks (SecureComm), 2014.

[8] J. Li, A. Squicciarini, D. Lin, S. Liang, and C. Jia, “Secloc: Securing location-sensitive storage in the cloud,” in ACM symposium on access control models and technologies (SACMAT), 2015.

[9] E. Order, “Presidential executive order on strengthening the cybersecurity of federal networks and critical infrastructure,” in https://www.whitehouse.gov/the-press-office/2017/05/11/presidentialexecutive-order-strengthening-cybersecurity-federal, 2017.

[10] “Hdfs architecture,” http://hadoop.apache.org/docs/stable/ hadoop-project-dist/hadoop-hdfs/HdfsDesign.html.

[11] R. Miller, “Inside amazon cloud computing infrastructure,” in http://datacenterfrontier.com/inside-amazon-cloud-computinginfrastructure/, 2015.

[12] T. Bujlow, K. Balachandran, S. L. Hald, M. T. Riaz, and J. M. Pedersen, “Volunteer-based system for research on the internet traffic,” Telfor Journal, vol. 4, no. 1, pp. 2–7, 2012.

[13] M. Geist, “Location matters up in the cloud,” http://www.thestar.com/ business/2010/12/04/geist location matters up in the cloud.html.

[14] Z. N. Peterson, M. Gondree, and R. Beverly, “A position paper on data sovereignty: the importance of geolocating data in the cloud,” in Proceedings of the 8th USENIX conference on Networked systems design and implementation, 2011.

[15] K. Benson, R. Dowsley, and H. Shacham, “Do you know where your cloud files are?” in Proceedings of the 3rd ACM workshop on Cloud computing security workshop. ACM, 2011, pp. 73–82.

[16] M. Gondree and Z. N. Peterson, “Geolocation of data in the cloud,” in Proceedings of the third ACM conference on Data and application security and privacy. ACM, 2013, pp. 25–36.

[17] G. J. Watson, R. Safavi-Naini, M. Alimomeni, M. E. Locasto, and S. Narayan, “Lost: location based storage,” in Proceedings of the 2012 ACM Workshop on Cloud computing security workshop. ACM, 2012, pp. 59–70.

[18] A. Albeshri, C. Boyd, and J. G. Nieto, “Geoproof: proofs of geographic location for cloud computing environment,” in Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on. IEEE, 2012, pp. 506–514.

[19] A. Albeshri, C. Boyd, and J. G. Nieto, “Enhanced geoproof: improved geographic assurance for data in the cloud,” International Journal of Information Security, vol. 13, no. 2, pp. 191–198, 2014.