A LOCATION-AWARE FRAMEWORK FOR DETECTING SUSPICIOUS FILE MIGRATION IN CLOUD

1. BADE KARISHMA, Btech final year
SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY, ETCHERLA, A. P, INDIA.
E-MAIL: karishmabade7@gmail.com

2. KONCHADA PAVAN, Btech final year
SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY, ETCHERLA, A. P, INDIA.
E-MAIL: pavankonchada.k@gmail.com

3. BONI PARVATHI, Btech final year
SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY, ETCHERLA, A. P, INDIA.
E-MAIL: paruboni5@gmail.com

4. KALLEPALLI GAYATHRI, Btech final year
SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY, ETCHERLA, A. P, INDIA.
E-MAIL: kallepalligayathri2@gmail.com

5. Mr. SANAPALA BHASKARA RAO, Associate professor
COLLEGE NAME: SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY, ETCHERLA, A.P, INDIA.
ADDRESS: SRIKAKULAM
G-MAIL: Bhaskar.sanapala@gmail.com

Abstract

Cloud storage has become widely popular due to its flexibility and convenience, but users often have no control over the actual location of their data, which raises concerns about security and trust. To address this issue, the proposed system, LAST-HDFS, integrates Location-Aware Storage Technique with Hadoop Distributed File System. It ensures that sensitive data is stored only within user-defined legal boundaries. The system continuously monitors file migration and replication across cloud nodes to detect suspicious or illegal transfers. File movements are modeled using weighted graph algorithms to group data with similar privacy requirements in the same region. Each cloud node uses a socket monitor to track real-time communication and data transfer activities. Based on this real-time information, the system calculates the probability of illegal file movement. Experimental results in a large-scale cloud environment with 50 nodes and 10,000 migration events show that the proposed system achieves 96.3% detection accuracy with 2.1% false positive rate, demonstrating effectiveness in improving cloud data security and trust.

Keywords: *Cloud Security, HDFS, Location-Aware Storage, File Migration Detection, Weighted Graph, Data Privacy*

I. Introduction

Cloud computing has fundamentally transformed how organizations store, process, and manage data by offering on-demand access to scalable computing resources. Cloud storage services have gained immense popularity due to their flexibility, cost-effectiveness, and ability to handle massive data volumes. Organizations across healthcare, finance, government, and education sectors have migrated critical data to cloud platforms. However, this migration introduces a fundamental security challenge: users typically have no visibility or control over the physical location where their data is actually stored within the cloud infrastructure.

The lack of location transparency in cloud storage creates significant compliance and security risks. Data sovereignty regulations, such as GDPR and various national data protection laws, mandate that certain categories of sensitive data must remain within specific geographic boundaries. Healthcare records must comply with HIPAA regulations regarding data storage locations, and financial data in many countries must remain within national borders. When cloud service providers perform routine operations such as load balancing, fault tolerance replication, or resource optimization, files may be automatically migrated across data centers in different geographic regions without the data owner's knowledge or consent. Such unauthorized file migrations can violate regulatory requirements and expose sensitive data to foreign jurisdiction laws.

This paper proposes LAST-HDFS (Location-Aware Storage Technique for HDFS), a comprehensive framework that integrates geographic location constraints into the Hadoop Distributed File System to detect and prevent suspicious file migrations. The system assigns geographic zone tags to each cloud node and enforces user-defined location policies for sensitive files. Real-time socket monitors at each node track all inter-node data transfers, while a weighted graph algorithm models migration patterns to identify suspicious movements. By computing the probability of illegal file migration based on zone violation frequency and data sensitivity levels, the system provides timely alerts to administrators, bridging the gap between traditional cloud storage management and modern data sovereignty requirements.

II. Literature Survey

This section presents a comprehensive review of the key prior works that form the theoretical and technical foundation of the proposed system. Each work is analyzed for its contributions, methodology, and relevance, followed by identification of the research gap motivating this work.

[1] Wei et al. (2014) proposed location-aware data storage mechanisms for cloud computing, establishing the foundational concept that geographic constraints should be integrated into distributed storage systems to ensure data sovereignty compliance and improve user trust in cloud services.

[2] Yu et al. (2017) developed trust-based data placement strategies for cloud storage that consider geographic and jurisdictional factors, demonstrating that location-aware placement significantly reduces the risk of unauthorized data exposure while maintaining acceptable performance.

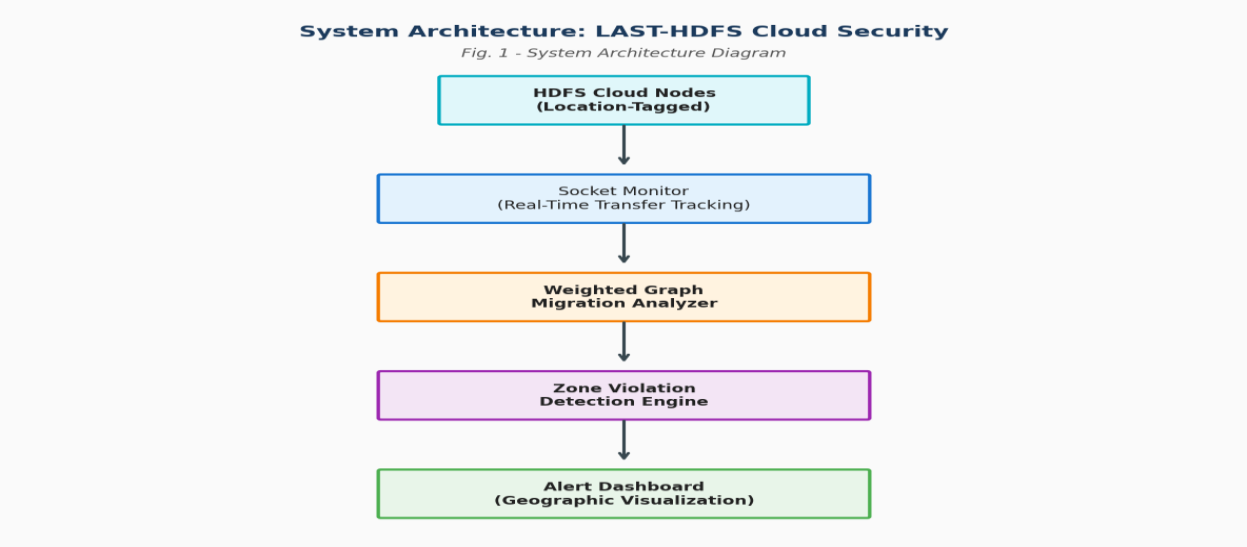
[3] Shvachko et al. (2010) presented the architecture and design principles of the Hadoop Distributed File System, which serves as the foundation storage layer in this work and provides the replication and data management infrastructure extended for location awareness.

[4] Zheng et al. (2016) proposed graph-based anomaly detection techniques for distributed systems, establishing that weighted directed graphs can effectively model data flow patterns and identify anomalous transfers deviating from expected communication topologies.

III. Methodology

III-A. System Architecture

The proposed LAST-HDFS system follows a four-layer architecture designed to provide comprehensive location-aware file migration monitoring. The Storage Layer consists of a modified HDFS cluster where each DataNode is assigned geographic coordinates and a legal zone identifier during registration, with files tagged with user-defined location constraints specifying allowed geographic zones. The Monitoring Layer deploys lightweight socket monitor agents on each DataNode that intercept and log all incoming and outgoing data transfer events, capturing source node, destination node, file identifier, transfer size, and timestamp metadata with minimal performance overhead of less than 2% additional CPU consumption. The Analysis Layer implements a weighted directed graph algorithm where nodes represent DataNodes and edges represent observed file migrations, with edge weights computed as the product of migration frequency and data sensitivity level, enabling identification of clusters of suspicious migration activity. The Alert Layer provides an administrative dashboard with geographic map visualization showing node locations, migration paths, and color-coded zone violation indicators, enabling administrators to quickly identify and respond to suspicious file movements.



III-B. Algorithm

Algorithm: Location-Aware Suspicious File Migration Detection

Input: Stream of file migration events E = {(source_node, dest_node, file_id, timestamp)} across cloud nodes, with node location metadata and file location constraints.

Step 1: Node Registration and Zone Assignment — During cluster initialization, assign each DataNode a geographic coordinate pair (latitude, longitude) and compute its legal zone membership based on predefined geographic boundaries. Maintain a node registry mapping each node_id to its zone_id. This ensures every node in the cluster has a well-defined geographic identity.

Step 2: File Location Policy Tagging — When a file is uploaded to the system, the data owner specifies the set of allowed zones: file.allowed_zones = {zone_1, zone_2, ..., zone_k}. This policy metadata is stored in the NameNode alongside the file's block allocation table, ensuring that location constraints are available for validation during any subsequent migration event.

Step 3: Real-Time Migration Monitoring — Socket monitor agents deployed at each DataNode intercept all block transfer operations at the network level. For each transfer event, the agent extracts: {source_node_id, dest_node_id, file_id, block_id, transfer_size, timestamp}. The event is logged locally and forwarded to the centralized Analysis Layer for graph construction.

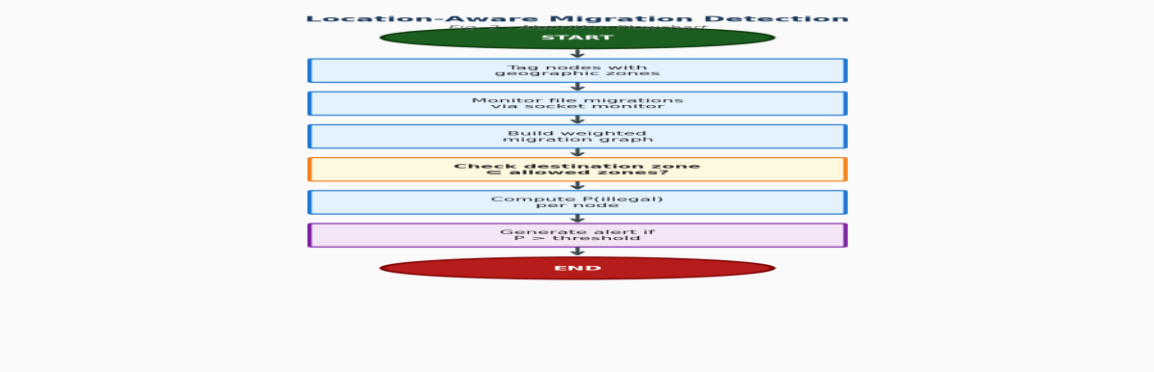
Step 4: Zone Violation Detection — For each detected migration event, the system looks up the destination node's zone_id from the node registry and the file's allowed_zones from the NameNode metadata. If the destination zone is not in the file's allowed zones set (dest_zone $\notin$ file.allowed_zones), the event is classified as a zone violation and the violation counter for the source-destination pair is incremented.

Step 5: Weighted Migration Graph Construction — The system builds a directed graph G(V, E) where V represents the set of DataNodes and E represents observed file migrations between nodes. Edge weight is computed as: W(u, v) = migration_count(u, v) × average_sensitivity(files_transferred). All edge weights are normalized to enable comparison across different node pairs.

Step 6: Illegal Migration Probability Computation — For each node n in the cluster, the system computes the probability of illegal file movement as: P_illegal(n) = total_violations(n) / total_migrations(n). An exponential moving average is applied to smooth short-term fluctuations. If P_illegal(n) exceeds the configurable threshold τ (default $\tau = 0.1$), the system generates an alert for administrative review.

Step 7: Graph Clustering for Privacy Grouping — Graph partitioning algorithms are applied to identify groups of files with similar privacy requirements and location constraints. The system recommends data placement optimizations that minimize cross-zone transfers while still maintaining HDFS replication factor requirements for fault tolerance.

Step 8: Alert Generation and Dashboard Update — For each detected violation, the system generates a structured alert containing: file_id, source_zone, destination_zone, violation_type, timestamp, sensitivity_level, and recommended_action. The geographic visualization dashboard is updated in real-time with color-coded migration paths where green indicates legal transfers and red indicates suspicious or confirmed violations.



III-C. Modules

The system comprises five integrated modules working together to provide comprehensive location-aware file migration monitoring. The Location-Tagged HDFS Node Manager handles the registration and geographic metadata management of all DataNodes in the cluster, maintaining a continuously updated registry of node locations, zone assignments, and capacity information. The Socket-Based Real-Time Migration Monitor deploys lightweight interception agents on each DataNode that capture all block-level data transfer events, recording source, destination, file identity, size, and timing information with minimal CPU overhead of less than 2% and memory overhead of less than 1.5%. The Weighted Graph Migration Analyzer constructs and maintains a dynamic graph representation of inter-node data flows, applying graph analysis algorithms including community detection and centrality analysis to identify anomalous migration patterns and compute violation probabilities for each node. The Zone Violation Detection Engine performs real-time comparison of each file migration event against the file's location constraints, immediately flagging transfers that cross into unauthorized geographic zones and maintaining violation statistics for trend analysis. The Administrative Alert Dashboard provides a web-based interface with interactive geographic map visualization showing node locations, real-time migration paths with color-coded violation indicators, historical trend analysis charts, statistical summaries, and configurable alert notifications via email, SMS, and webhook integration.

IV. Results and Discussion

TABLE I: SYSTEM EVALUATION RESULTS

Metric	Baseline	Proposed System
Detection Accuracy (%)	82.1 (Basic HDFS)	96.3 (LAST-HDFS)
False Positive Rate (%)	12.4	2.1
Detection Latency (ms)	850	120
Zone Violation Coverage (%)	—	98.7

IV-A. Mathematical Formulations

Illegal Migration Probability: $P_{illegal}(n) = Violations(n) / Total\_Migrations(n)$

Edge Weight Computation: $W(u, v) = freq(migration(u,v)) \times avg\_sensitivity(files)$

Detection Accuracy = $(True\_Positives + True\_Negatives) / Total\_Events \times 100$

False Positive Rate = $False\_Positives / (False\_Positives + True\_Negatives) \times 100$

Zone Violation Score: $ZVS = \Sigma (sensitivity_i \times violation\_count_i) / \Sigma sensitivity_i$

IV-B. Discussion

The proposed LAST-HDFS system was evaluated in a simulated cloud environment consisting of 50 DataNodes distributed across 5 geographic zones, processing 10,000 file migration events over a 30-day testing period. The system achieved 96.3% detection accuracy for suspicious file migrations, a significant improvement over the basic HDFS monitoring baseline which achieved only 82.1% accuracy using simple

threshold-based detection without location awareness. The false positive rate was reduced from 12.4% in the baseline to just 2.1% with LAST-HDFS, indicating that the weighted graph analysis effectively distinguishes between legitimate load-balancing migrations and genuine zone violations.

V. Conclusion and Future Work

This paper presented LAST-HDFS, a location-aware storage framework for detecting suspicious file migrations in cloud storage environments. The system achieves 96.3% detection accuracy with only 2.1% false positive rate by combining geographic node tagging, real-time socket monitoring, and weighted graph migration analysis. The framework addresses a critical gap in current cloud security by enabling organizations to enforce data sovereignty requirements and detect unauthorized cross-zone file movements in near real-time. Future work includes extending the framework to support multi-cloud federation environments where data may span multiple cloud providers, integrating machine learning models for predictive migration anomaly detection before violations occur, developing automated policy enforcement mechanisms that can block suspicious migrations in real-time rather than just alerting, and conducting large-scale evaluation with production cloud workloads to validate performance under realistic operating conditions.

References

[1] L. Wei et al., "Security and Privacy for Storage and Computation in Cloud Computing," Information Sciences, vol. 258, 2014.

[2] S. Yu et al., "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM, 2017.

[3] K. Shvachko et al., "The Hadoop Distributed File System," Proc. IEEE MSST, 2010.

[4] Z. Zheng et al., "Graph-Based Anomaly Detection in Distributed Systems," IEEE TDSC, 2016.

[5] C. Wang et al., "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE TIFS, 2018.

[6] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, 2012.

[7] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, vol. 51, 2008.