ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

FAULT-TOLERANT DATA DISTRIBUTION IN EDGE COMPUTING VIA ERASURE CODING: THE EDGE HYDRA APPROACH

M. RAMAN KUMAR[1], KOLIPAKA HARITA[2], AMBAREESHA SRI VAISHNAVI[3], THAMMULURI AMULYA DEEPIKA[4], YANMANGANDLA JASWANTH[5]

[1]Assistant Professor, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

[2, 3, 4, 5] B.Tech, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

ABSTRACT— In edge computing environments, distributing popular data from the cloud to edge servers efficiently is critical for providing low-latency access. Existing distribution schemes divide a file into multiple data blocks and transmit them in parallel to target edge servers, which then reconstruct the file upon receiving all blocks. Although this approach accelerates data delivery, it is vulnerable to transmission delays and failures caused by network fluctuations or server outages, as the successful reconstruction depends on receiving every data block. To address these limitations, this paper proposes EdgeHydra, a fault-tolerant data distribution scheme based on erasure coding. EdgeHydra encodes a file into both data and parity blocks, allowing edge servers to reconstruct the original file from any sufficient subset of these blocks without waiting for all transmissions to complete. Furthermore, it employs a leaderless block supplement mechanism, enabling distributed coordination among edge servers to recover missing blocks without relying on a centralized controller. Experimental results demonstrate that EdgeHydra significantly improves robustness against delays and failures, achieving up to 50.54% faster distribution times compared to state-of-the-art methods.

Index Terms – Fault-Tolerant Data Distribution, Edge Computing, Erasure Coding, Distributed Storage, Data Availability, Fault Tolerance, Storage Optimization, Cloud-Edge Computing, Distributed Systems, Data Reliability.

I. INTRODUCTION

WITH the rapid development of modern technologies such as the 5G communication and artificial intelligence (AI), the number of smart devices and services has increased dramatically over the last decade. The requirements for real-time data processing have also increased profoundly. In the traditional cloud computing paradigm, data is stored in remote data centers for user access. The distance between users and data centers introduces inevitable latency when users access these data. This significantly impacts the performance of applications especially those requiring real-time data access Edge computing has emerged as a promising solution to this fundamental challenge by extending storage and computing resources from the cloud to the network edge. Based on users' data demands, popular data can be cached on edge servers by app vendors, enabling their users to access these data with ultralow latency. An edge caching system serving a specific geographic region consists of a set of edge servers covering different geographical are as in the region. Recently, a series of studies have focused on the design of edge caching strategies to meet app vendors' diverse needs. To implement these caching strategies, app vendors must distribute data from cloud servers to targeted servers within the edge caching system. However, data transfer from the cloud to the edge takes time and incurs financial costs. Given a set of target edge servers in an edge caching system, denoted by S = {s1,...,sn}, edge data distribution (EDD) aims to transmit a file f from the cloud server to the system for caching on each of the edge servers in S. There are different ways to transmit f to S. DataSync: DataSync is an end-to-end data transfer ser vice provided by Amazon. Data Sync ensures low data distribution latency. However, considering that the cost of transferring data from the cloud to edge is much more than the cost of transferring data between edge servers ($0.05-$0.09 per GB versus $0.01 per GB),1,2 DataSync-based edge data distribution is often not economic. Gossip is widely used for data transfer across storage nodes in distributed systems. To minimize distribution costs, only a single copy of the file is transmitted from the cloud to the edge caching system under the Gossip scheme. Based on Gossip, transferred to a selected entry server. Then, these edge servers transfer f process completes when all the target servers receive f. Compared with DataSync, the Gossip scheme reduces the transmission cost but requires gossiping among edge servers, which often causes substantial delays in the distribution process. EDD-A EDD-A is a scheme specifically designed for edge data distribution, considering both transmission delay and cost. EDD-A constructs a minimum Steiner tree with the cloud server serving as the root node, while target servers function as either leaf nodes or intermediate nodes. The cloud server then transfers f along the edges of the Steiner tree. EDD-A reduces the amount of data transferred from the cloud server to the edge caching system, lowering the transmission cost incurred. However, under the EDD-A scheme, the distribution of f is sensitive to runtime exceptions like network fluctuations and server failures. A network delay or a server failure occurring over any path in the Steiner tree will impact the entire data distribution process significantly. EdgeDis Edgedis is the state-of-the-art solution designed for the EDD problem. Under the Edgedis scheme, the app vendor first slices f into some data blocks and transfers each of the blocks simultaneously to a selected entry server. These entry servers will then forward their block to the target servers, which can combine received blocks to reconstruct f. To improve the robustness of the EDD process, EdgeDis selects an edge server in the system as a coordinator to supply any missing blocks to target servers, ensuring that all the target servers will receive f eventually.

II. RELATED WORK

A. *Edge Dis: Enabling fast, economical, and reliable data dissemination for mobile edge computing.*

Mobile edge computing (MEC) enables web data caching in close geographic proximity to end users. Popular data can be cached on edge servers located less than hundreds of meters away from end users. This ensures bounded latency guarantees for various latency-sensitive web applications. However, transmitting a large volume of data out of the cloud onto many geographically-distributed web servers individually can be expensive. In addition, web content dissemination may be interrupted by various intentional and accidental events in the volatile MEC environment, which undermines dissemination efficiency and subsequently incurs extra transmission costs. To tackle the above challenges, we present a novel scheme named EdgeDis that coordinates data dissemination by distributed consensus among those servers. We analyze EdgeDis's validity theoretically and evaluate its performance experimentally. Results demonstrate that compared with baseline and state-of-the-art schemes, EdgeDis: 1) is 5.97x - 7.52x faster; 2) reduces dissemination costs by 48.21% to 91.87%; and 3) reduces performance loss caused by dissemination failures by up to 97.30% in time and 96.35% in costs.

B. *Age-aware data selection and aggregator placement for timely federated continual learning in mobile edge computing*

Federated continual learning (FCL) is emerging as a key technology for time-sensitive applications in highly adaptive environments including autonomous driving and industrial digital twin. Each FCL trains machine learning models using newly-generated datasets as soon as possible, to obtain a highly accurate machine learning model for new event predictions. The age of data, defined as the time difference between the generation time of a dataset and the current time, is widely adopted as a key criterion to evaluate both timeline and quality of training. In this paper, we study the problem of age-aware FCL in a mobile edge computing (MEC) network. We not only investigate optimization techniques that optimize the data selection and aggregator placement for FCL but also implement a real system as a prototype for age-aware FCL. Specifically, we first propose an approximation algorithm with a provable approximation ratio for the age-aware data selection and aggregator placement problem for FCL with a single request. In real application scenarios, there are usually multiple FCL requests that require to train models, and delays in the MEC network are usually uncertain. We then study the problem of age-aware data selection and aggregator placement problem for FCL with uncertain delays and multiple requests, by devising an online learning algorithm with a bounded regret based on contextual bandits. We finally implement a prototype for FCL in an MEC network, with various heterogeneous user equipments (UEs) and cloudlets with different computing capabilities in the network. Experiment results show that the performance of the proposed algorithms outperform existing studies, by achieving 47% lower age of data and 12% higher model accuracy.

C. *Enabling age-aware Big Data analytics in serverless edge clouds*

With the fast development of artificial intelligence applications, large-volume big data generated in the edge of networks are waiting for real-time analysis, such that the valuable information is unveiled. Analytic developers for big data applications usually face the burden of managing the underlying cloud resources, which greatly drags the speed of analytic development. Serverless Computing is envisioned as an enabling technology to release the management burden of developers and to enable agile big data analytics. That is, big data analytics can be implemented in short-lived functions via the Function-as-a-Service (FaaS) programming paradigm. In this paper, we aim to fill the gap between serverless computing and mobile edge computing, via enabling query evaluations for big data analytics in short-lived functions of a serverless edge cloud (SEC). Specifically, we formulate novel age-aware big data query evaluation problems in an SEC so that the age of data is minimized, where the age of data is defined as the time difference between the current time and the generation time of the dataset. We propose approximation algorithms for the age-aware big data query evaluation problem with a single query, by proposing a novel parameterized virtualization technique that strives for a fine trade-off between short-lived functions and large resource demands of big data queries. We also devise an online learning algorithm with a bounded regret for the problem with multiple queries arriving dynamically and without prior knowledge of resource demands of the queries. We finally evaluate the performance of the proposed algorithms by extensive simulations. Simulation results show that the performance of our algorithms is promising.

III. PROPOSED METHODOLOGY

To mitigate the impacts caused by these faulty servers, this paper presents EdgeHydra, a new EDD scheme that distributes blocks encoded with erasure coding instead of data blocks. Under EdgeHydra, the cloud server first encodes file f into a set of coded blocks with erasure coding. Then, the cloud server sends these coded blocks to entry servers for distribution in the edge caching system. In this way, an edge server does not need all the coded blocks to reconstruct f. Instead, it can reconstruct f with only a proportion of these coded blocks. This allows EdgeHydra to tolerate slow or failed data transmission from faulty servers. It is the key idea of EdgeHydra.

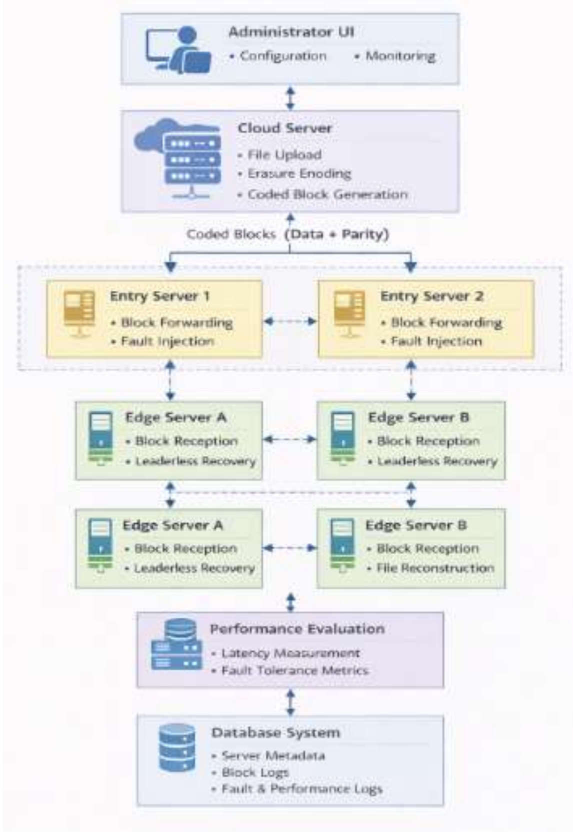
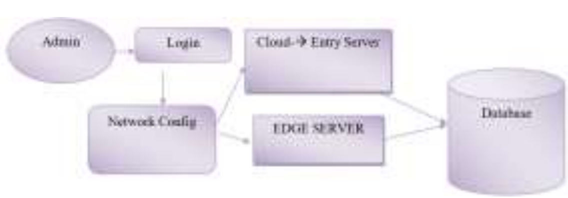


Fig. 1: System Overview

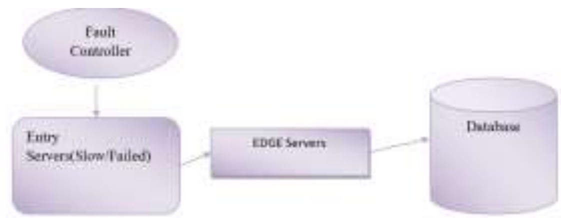
1. Network Initialization Module

This module initializes the edge computing environment by registering the cloud server, entry servers, and edge servers. Each node is assigned a unique identity, and communication links are established to prepare the system for edge data distribution. The network topology and server roles are defined in this phase.



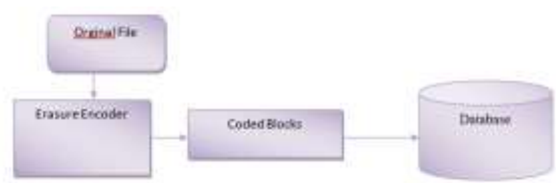
2. Fault Modeling Module

This module models faulty servers by simulating slow or failed entry servers. It introduces network delays, packet loss, and server failures to analyze how such faults affect data transmission. These fault scenarios represent real-world runtime issues in edge environments.



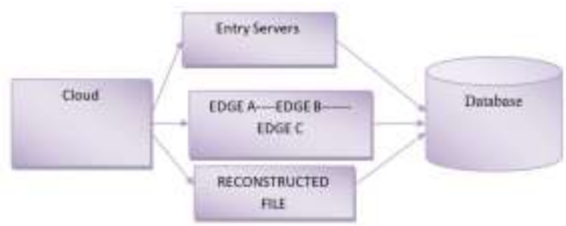
3. ERASURE CODING MODULE

In this module, the cloud server encodes the original file into multiple coded blocks using erasure coding. These blocks include both data and parity blocks, allowing edge servers to reconstruct the file even if some blocks are missing or delayed.



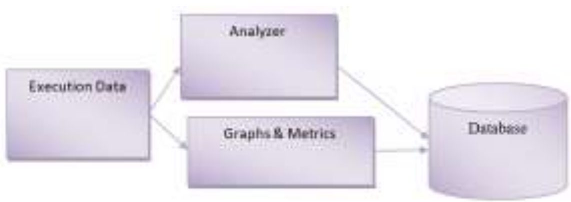
4. BLOCK DISTRIBUTION AND RECOVERY MODULE

This module distributes coded blocks from the cloud to entry servers and then to edge servers. If some blocks are missing due to faulty servers, edge servers cooperate using a leaderless block supplement mechanism to exchange blocks and recover the required data without a central controller.



5. Performance Evaluation Module

This module evaluates the performance of EdgeHydra by measuring distribution time, fault tolerance, and communication overhead. The results are compared with traditional edge data distribution schemes to demonstrate improvements in robustness and efficiency.



IV. EXPERIMENTAL RESULTS AND ANALYSIS

Experimental results demonstrated that the erasure coding mechanism efficiently distributed data across multiple edge nodes while ensuring successful data recovery even when several nodes became unavailable or experienced failures. This improved data availability without requiring excessive storage redundancy.

The analysis showed that the proposed framework achieved lower storage overhead and higher fault tolerance compared to traditional replication-based data distribution methods.

Fig. 2: Edge Cluster1

Fig. 3: Distributed Files

Fig. 4: Distributed Files

Fig. 5: Edge Cluster

Fig. 6: Benchmark

V. CONCLUSION

This paper presented EdgeHydra, an erasure coding–based edge data distribution (EDD) scheme designed to address the limitations of traditional parallel block distribution methods in edge computing environments. By encoding files into coded blocks, EdgeHydra enables edge servers to reconstruct data using only a sufficient subset of blocks, thereby eliminating dependency on slow or failed servers. The introduction of a decentralized block supplement mechanism further ensures balanced transmission workloads and removes reliance on centralized coordination. Experimental evaluations demonstrate that EdgeHydra significantly improves robustness and efficiency under dynamic network conditions, achieving up to 50.54% reduction in distribution time compared to state-of-the-art EDD approaches. Overall, EdgeHydra proves to be an effective, scalable, and fault-tolerant solution for reliable data distribution in real-world edge computing systems.

REFERENCES

[1] A. Osseiran et al., “The foundation of the mobile and wireless communi cations system for 2020 and beyond: Challenges, enablers and technology solutions,” in Proc. IEEE Veh. Technol. Conf., 2013, pp. 1–5.

[2] Z.Xuetal., “Age-aware datas election and aggregator placement for timely federated continual learning in mobile edge computing,” IEEE Trans. Comput., vol. 73, no. 2, pp. 466–480, Feb. 2024.

[3] R. Bhardwaj et al., “Ekya: Continuous learning of video analytics models on edge compute servers,” in Proc. 19th USENIX Symp. Netw. Syst. Des. Implementation, 2022, pp. 119–135.

[4] X. Hou and S. Dey, “Motion prediction and pre-rendering at the edge to enable

ultra-low latency mobile 6DoF experiences,” IEEE Open J. Commun. Soc., vol. 1, pp. 1674–1690, 2020.

[5] F. Nawab, D.Agrawal, and A. El Abbadi, “DPaxos: Managing data closer to users for low-latency and mobile applications,” in Proc. Int. Conf. Manage. Data, 2018, pp. 1221–1236.

[6] Amazon, “AWS wavelength for media & entertainment,” 2021. [On line].

AUTHORS Profile

Mr. M. RAMAN KUMAR received his B.Tech degree in Computer Science and Engineering from Ranchi University in 2010 and his M.Tech degree in Computer Science and Engineering from Bhagwant University, Ajmer, in 2020. He is presently pursuing his Ph.D. from Atal Bihari Vajpayee Indian Institute of Information Technology and Management (ABV-IIITM), Gwalior. He has been dedicated to the teaching profession for the past 2 years. At present, he is working as an Assistant Professor in the Department of Computer Science and Engineering at Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India**.**

Ms. KOLIPAKA HARITA pursuing B. Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.

Ms. AMBAREESHA SRI VAISHNAVI pursuing B. Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.

Ms. THAMMULURI AMULYA DEEPIKA pursuing B. Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.

Mr. YANMANGANDLA JASWANTH pursuing B. Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.