Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 10, October : 2024

Blockchain-Based Decentralized Storage Design for Data Confidence Over Cloud-Native Edge Infrastructure

Dr. Abdul Khadeer, Associate Professor, Department of CSE, Deccan College of Engineering and Technology, Osmania University, Hyderabad, Telangana :: abdulkhadeer@deccancollege.ac.in

Mohd Yousuf Ahmed, PG Scholar, Department of CSE, Deccan College of Engineering and Technology, Osmania University, Hyderabad, Telangana :: mohd.yousuf.ahmed@gmail.com

Abstract:- As the modern computing market experiences a surge in demand for efficient data-management solutions, challenges posed by centralized storage systems become more pronounced, especially with the proliferation of Internet of Things devices. Centralized storage, although cost-effective, faces issues of scalability, performance bottlenecks, and security vulnerabilities. With decentralized storage, data are distributed across nodes, offering redundancy, data availability, and enhanced security. Unfortunately, decentralized storage introduces its own challenges, such as complex data retrieval processes, potential inconsistencies in data versions, and difficulties in ensuring data privacy and integrity in a distributed setup. Effectively managing these challenges calls for innovative techniques. In response, this paper introduces a decentralized storage system that melds cloud-native concepts with blockchain technology. The proposed design delivers enhanced scalability, data security, and privacy. When operating on a containerized edge infrastructure, this storage system provides higher data-transfer speeds than the interplanetary file system. This research thus blends the advantages of cloud-native frameworks with the security mechanisms of blockchain, crafting a storage system that addresses the present-day challenges of data management in decentralized settings.

***Index Terms*— Edge computing, distributed storage, cloud-native orchestration, blockchain.**

INTRODUCTION:-

In today's computing landscape, the proliferation of Internet of Things (IoT) devices and data-intensive applications has driven a significant increase in demand for efficient and responsive data management solutions. Centralized storage, characterized by the concentration of data in a single location or system, has long been a popular choice because of its simplicity and cost-effectiveness. Such a centralized approach offers immediate data consistency, simplified management, The associate editor coordinating the review of this manuscript and approving it for publication was Nitin Gupta and centralized resource allocation, reducing hardware and maintenance costs. Unfortunately, the increasing scale and complexity of modern computing environments have raised significant challenges in meeting the storage needs of today's dynamic data landscape. . Centralized storage systems face notable limitations, particularly in scenarios with numerous IoT devices, which can generate a staggering amount of data. IoT devices are estimated to number approximately 25.44 billion by 2030, which will drive an exponential growth in data volume and present considerable scalability and performance bottlenecks for centralized storage. Some of the challenges of centralized storage include a single point of failure, which can disrupt operations when the system malfunctions; performance bottlenecks due to latency when users or devices accessing the data are geographically distributed; scalability limitations as data volumes continue to grow; and data security and privacy concerns due to the concentration of large volumes of data in one location, creating a prime target for security breaches. Decentralized storage is a complementary computing paradigm that addresses these

challenges by distributing data across multiple nodes or devices instead of relying on a central location. This architecture replicates and spreads data throughout a network of nodes, providing redundancy, improved data availability, and enhanced security. To realize these advantages, a decentralized storage system must incorporate secure access and sharing mechanisms for distributed data. Existing storage systems use metadata and metadata servers to manage the physical addresses and access privileges of remote data. Notable companies such as Cohesity, VAST Data, and Hammer space offer enterprise solutions that leverage metadata to simplify access and retrieval of raw data. However, managing metadata necessitates periodic validation and updates to ensure seamless access across decentralized storage nodes. The benefits of additional servers for metadata management must be balanced with the potential impact on system complexity and performance.

LITERATURE REVIEW:-

The paper "A survey on Proof of Retrievability for cloud data integrity and availability: Cloud storage state-of-the-art, issues, solutions and future trends " Cloud storage has emerged as the latest trend for data storage over the traditional storage method which consume more storage spaces of data owner resources for backup and disaster recovery purposes. Due to the openness nature of cloud storage, trustworthy to the storage providers remains a critical issue amongst data owners. Hence, a huge number of businesses around the world remains choosing traditional storage method over cloud storage. This indicates a need for cloud storage providers to adopt cloud integrity schemes to ensure the outsourced data is secured to gain trustworthiness from clients. There are two main cloud integrity schemes available to ensure data integrity and availability: (i) Provable Data Possession (PDP) and (ii) Proof of Retrievability (PoR). PDP and PoR are protocols designed for cloud storage to proof to clients that the stored data is intact. Although PDP and PoR have similar functionality for providing cloud data integrity and availability, PoR is found to be much better than PDP with respect to full data retrievability as PoR provides recovery to faulty or corrupted outsourced data in which PDP does not cover. The objective of this paper is to examine the state-of-the-art of PoR and subsequently to identify the issues of employing PoR on cloud storage and suggest possible solutions. We analyze available PoR schemes. Then, the issues and challenges because of employing PoR specifically and cloud storage generally are described. Some possible countermeasures to address the identified issues are suggested. Finally, the potential future work of PoR schemes and future trends of cloud storage are presented.

This paper "Object storage: the future building block for storage systems" The concept of object storage was introduced in the early 1990's by the research community. Since then, it has greatly matured and is now in its early stages of adoption by the industry. Yet, object storage is still not widely accepted. Viewing object store technology as the future building block particularly for large storage systems, our team in IBM Haifa Research Lab has invested substantial efforts in this area. In this position paper we survey the latest developments in object store technology, focusing on standardization, research prototypes, and technology adoption and deployment. A major step has been the approval of the TIO OSD protocol (version I) as an OSD standard in late 2004. We also report on prototyping efforts that are carried out in IBM Haifa Research Lab in building an object store. Our latest prototype is compliant with a large subset of the TIO standard. To facilitate deployment of the new technology and protocol in the community at large, our team also implemented a TIO-compliant OSD (iSCSI) initiator for Linux. The initiator is interoperable with object disks of other vendors. The initiator is available as an open-source driver for Linux.

This paper "Object storage in the cloud and multi-cloud: State of the art and the research challenges
" The cloud has been one of the hottest topics of discussion in the information technology field, both in terms of research and on a commercial level, since its inception in the late 2000s. The ever-increasing

need for better, faster, and more powerful computers to process the ever-increasing amount of data has made the cloud an indispensable feature of modern IT. Due to its massive storage power the cloud has allowed corporate users, private users, and researchers to centralize all their data and have access to it almost constantly so long as they have an internet connection. However, despite the obvious need for more storage capacity and the ever-increasing reliance on storing files in the cloud no mainstream unified storage solution is available. This paper attempts to review and analyze the current cloud offerings on the market, the storage solutions provided both on a commercial level and on a theoretical level, the challenges of the multi-cloud and the challenges of any future research in this field.

In this paper "Autonomic Management Framework for Cloud-Native Applications" In order to meet the rapidly changing requirements of the Cloud-native dynamic execution environment, without human support and without the need to continually improve one's skills, autonomic features need to be added. Embracing automation at every layer of performance management enables us to reduce costs while improving outcomes. The main contribution of this paper is the definition of autonomic management requirements of Cloud-native applications. We propose that the automation is achieved via high-level policies. In turn autonomy features are accomplished via the rule engine support. First, the paper presents the engineering perspective of building a framework for Autonomic Management of Cloud-Native Applications, namely AMoCNA, in accordance with Model Driven Architecture (MDA) concepts. AMoCNA has many desirable features whose main goal is to reduce the complexity of managing Cloud-native applications. The presented models are, in fact, meta-models, being technology agnostic. Secondly, the paper demonstrates one possibility of implementing the afore mentioned design procedures. The presented AMoCNA implementation is also evaluated to identify the potential overhead introduced by the framework.

In this paper "Ananke: A framework for Cloud-Native Applications smart orchestration" Micro-service architecture enables smarter management of applications life-cycle. However, the increasing of the number of components also increases complexity, especially on operations like migration and horizontal scaling. While operations, in monolithic systems, involve only one component, operations in micro-services-based applications can get complex and should involve parameters and properties like connection throughput, resources usage, robustness or consistency and reliability. To perform this kind of operations and optimization strategies in micro-services applications, we propose Ananke, a framework consisting of a time-varying multi-layer graph-based model and architecture to profile micro-services and their interactions in a platform-as-a-service environment. The aim of Ananke is to provide support and facilities for optimization strategies that a Cloud Provider can exploit to guarantee quality of service and service-level agreements better.

Customizing and deploying an edge system are time-consuming and complex tasks because of hardware heterogeneity, third-party software compatibility, diverse performance requirements, and so on. In this article, we present TinyEdge, a holistic framework for the low-code development of edge systems. The key idea of TinyEdge is to use a top-down approach for designing edge systems. Developers select and configure TinyEdge modules to specify their interaction logic without dealing with the specific hardware or software. Taking the configuration as input, TinyEdge automatically generates the deployment package and estimates the performance with sufficient profiling. TinyEdge provides a unified development toolkit to specify module dependencies, functionalities, interactions, and configurations. We implement TinyEdge and evaluate its performance using real-world edge systems. Results show that: (1) TinyEdge achieves rapid customization of edge systems, reducing 44.15% of development time and 67.79% of lines of code on average compared with the state-of-the-art edge computing platforms; (2) TinyEdge builds compact modules and optimizes the latent circular dependency detection and message routing efficiency; (3) TinyEdge performance estimation has low absolute errors in various settings.

Edge computing constitutes a promising paradigm of managing and processing the massive amounts of data generated by Internet of Things (IoT) devices. Data and computation are moved closer to the client, thus enabling latency- and bandwidth-sensitive applications. However, the distributed and heterogeneous nature of the edge as well as its limited resource capabilities pose several challenges in implementing or choosing an efficient edge-enabled storage system. Therefore, it is imperative for the research community to contribute to the clarification of the purposes and highlight the advantages and disadvantages of various edge-enabled storage systems. This work aspires to contribute toward this direction by presenting a performance analysis of three different storage systems, namely MinIO, BigchainDB, and the IPFS. We selected these three systems as they have been proven to be valid candidates for edge computing infrastructures. In addition, as the three evaluated systems belong to different types of storage, we evaluated a wide range of storage systems, increasing the variability of the results. The performance evaluation is performed using a set of resource utilization and Quality of Service (QoS) metrics. Each storage system is deployed and installed on a Raspberry Pi (small single-board computers), which serves as an edge device, able to optimize the overall efficiency with minimum power and minimum cost. The experimental results revealed that MinIO has the best overall performance regarding query response times, RAM consumption, disk IO time, and transaction rate. The results presented in this paper are intended for researchers in the field of edge computing and database systems.

Edge computing responds to users' requests with low latency by storing the relevant files at the network edge. Various data deduplication technologies are currently employed at edge to eliminate redundant data chunks for space saving. However, the lookup for the global huge-volume fingerprint indexes imposed by detecting redundancies can significantly degrade the data processing performance. Besides, we envision a novel file storage strategy that realizes the following rationales simultaneously: 1) space efficiency, 2) access efficiency, and 3) load balance, while the existing methods fail to achieve them at one shot. To this end, we report LOFS, a Lightweight Online File Storage strategy, which aims at eliminating redundancies through maximizing the probability of successful data deduplication, while realizing the three design rationales simultaneously. LOFS leverages a lightweight three-layer hash mapping scheme to solve this problem with constant-time complexity. To be specific, LOFS employs the Bloom filter to generate a sketch for each file, and thereafter feeds the sketches to the Locality Sensitivity hash (LSH) such that similar files are likely to be projected nearby in LSH tablespace. At last, LOFS assigns the files to real-world edge servers with the joint consideration of the LSH load distribution and the edge server capacity. Trace-driven experiments show that LOFS closely tracks the global deduplication ratio and generates a relatively low load std compared with the comparison methods.

The rapid growth of computing capabilities at network edge calls for efficient management frameworks that not only considers placing hot data on edge storage for best accessibility and performance, but also makes optimal utilization of edge storage space. In this paper, we solve a joint optimization problem by exploiting both data popularity (for optimal data access performance) and data similarity (for optimal storage space efficiency). We show that the proposed optimization is NP- hard and develop a $2\lceil 2\Gamma \rceil - 1 + \epsilon$-approximation algorithm by (i) making novel use of δ-similarity graph to capture pairwise data similarity and (ii) leveraging the k-MST algorithm to solve a Prize Collecting Steiner Tree problem on the graph. The proposed algorithm is prototyped using an open-source distributed storage system, Cassandra. We evaluate its performance extensively on a real-world testbed and with respect to real-world IoT datasets. The algorithm is shown to achieve over 55% higher edge service rate and reduces request response time by about 30%.

EXISTING METHODS:-

Cloud-native edge storage has received considerable attention, with various open-source projects and standards exploring cloud-native computing. Applying cloud-native concepts to edge computing has led to the development of software frameworks that support autonomic configuration, management of containerized edge infrastructures, and workflow monitoring. However, the absence of standard guidelines for cloud-native edge storage and the reliance on limited virtualization support in previous studies create challenges in this area.

Many studies neglect to specify the assumed edge computing environment, often resorting to virtualized environments established on singular personal computer-level apparatuses for experimentation.

Existing research often relies on storage solutions such as IPFS, which may underperform compared with traditional protocols such as FTP [36]. Although IPFS supports a novel storing and retrieving mechanism and is widely used to host over 1000 websites, it has notable performance drawbacks in writing and reading files

PROPOSED SYSTEM:-

We propose a blockchain-based decentralized storage system that ensures a secure storage environment. The blockchain manages data encryption keys for secure data sharing and handles data descriptions such as version, modification date, and data ownership for data validation. Conversely, non-sharable data, including user accounts, access rights, and data location, are maintained in a distinct database. • We implement an open-source software framework that supports cloud-native orchestration to obtain a fully containerized storage system. The software framework allows each node to assume any role in the storage cluster, addressing scalability and flexibility challenges in blockchain-based storage systems. • We implement the proposed storage system in an edge computing environment with real physical boxes equipped with 100 GbE smart network interface cards (NICs) and nonvolatile memory express (NVMe) solid-state drives (SSDs), ensuring all-flash data storage. These physical boxes relate to 100 GbE smart switches, and this environment serves to validate the storage system's performance. The system performance is further compared with that of IPFS to highlight the system's efficiency.

METHODOLOGY:-

Owner/Client Module:

Using this module owner / client can register with application with valid details and request for public and private key is sent to data pound module where keys are generated, and public key is received to client after acceptance using this key client can login to application with username password and key. User can upload data with rights permission, version details , data description and encrypt data then send to data pond nodes which will manage different type of data. Client can modify data and update version details and send to data pond and block chain server for secure verification.

Data pond/ Nodes Module:

Using this module data pond is a storage server which will store different types of data in nodes like version , data, framework. Details are in encrypted format for secure storage and store data to drive hq cloud. Data pond can view requests sent by user module for verification and send security key for decryption.

Block Chain Server:

Using this module block chain server will improve security for generating block chain key for users' data and when user changes data or version of data then block chain keys will be changes. This sever helps in second layer authentication without data modification and improve privacy of data. Server will

view request from user module for block chain verification and give permission to download data.

User Module:
Using this module user will register with application and view encrypted data with version details and request data to data pond and block chain server and get key to download and decrypt data.

ARCHITECTURE:-

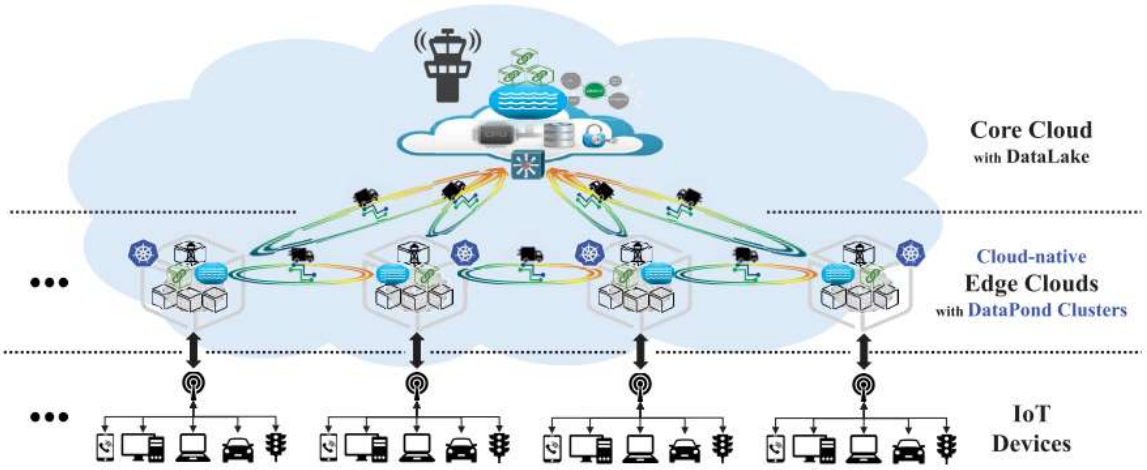


Figure 1. System Architecture

Figure 1 shows the system environment of the proposed storage. It visually depicts the key components and relationships that are crucial for understanding the present research. This paper uses the concepts of core cloud and edge cloud to describe specific computing environments. The core cloud represents traditional centralized cloud infrastructure characterized by large data centers and centralized processing and stands in contrast to the edge cloud, a decentralized computing model in which computing resources are situated closer to data sources, often involving smaller, distributed data centers. These distinctions are applied consistently throughout the paper and are essential for understanding this study of decentralized storage. In this study, we explore the implications of enhancing the security and scalability of storage in edge clouds.

ALGORITHM:-

Attention (ai ∈ M): Attention reflects the importance or priority attached to vi. The retrieval order of vi is established by ai. The storage system starts by retrieving the vi with the highest ai. For instance, if ai > aj, vi is retrieved before vj. The retrieval condition of vi is ai ≥ ci. The client determines the value of ai. • Confidence (ci ∈ M): Confidence indicates the degree of trust the storage system places in di. The storage system adjusts the value of ci to retrieve vi. As explained above, the storage system retrieves vi only if ai ≥ ci. Thus, for confidentiality, ci has a threshold to prevent some of the data from being retrieved by the storage system

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 10, October : 2024

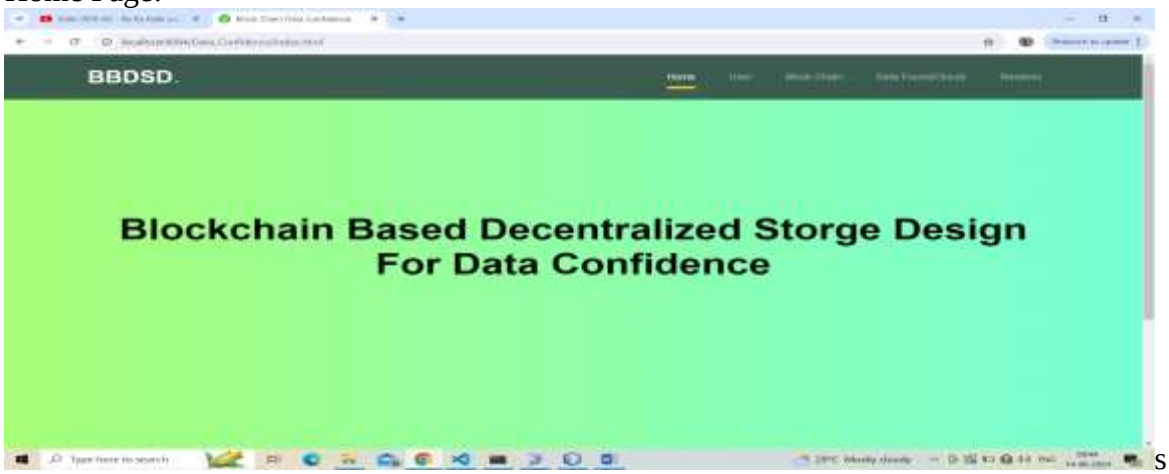
Algorithm 1: Data retrieval algorithm

Input: D, M
Output: D
for $d_i \in D$ where $a_i \neq Never$ **do**
  select i where $v_i \in \emptyset$, and $a_i = \max(a_i)$;
  // Label c_i
  if $a_i = Ever$ **then**
    Request retrieval permission from client ;
    If approved, then $c_i = Ever$;
    Otherwise, $c_i = Sometimes$;
  end
  else
    If metadata (modified date, owner, etc) of i is missing, then $c_i = Sometimes$;
    Otherwise, $c_i = Usually$;
  end
  // Retrieve v_i
  if $a_i \geq c_i$ **then**
    Retrieve v_i from DataPond storage ;
    Update result to D ;
  end
end
return D ;

Figure 2. Model Flow Diagram

RESULTS:-

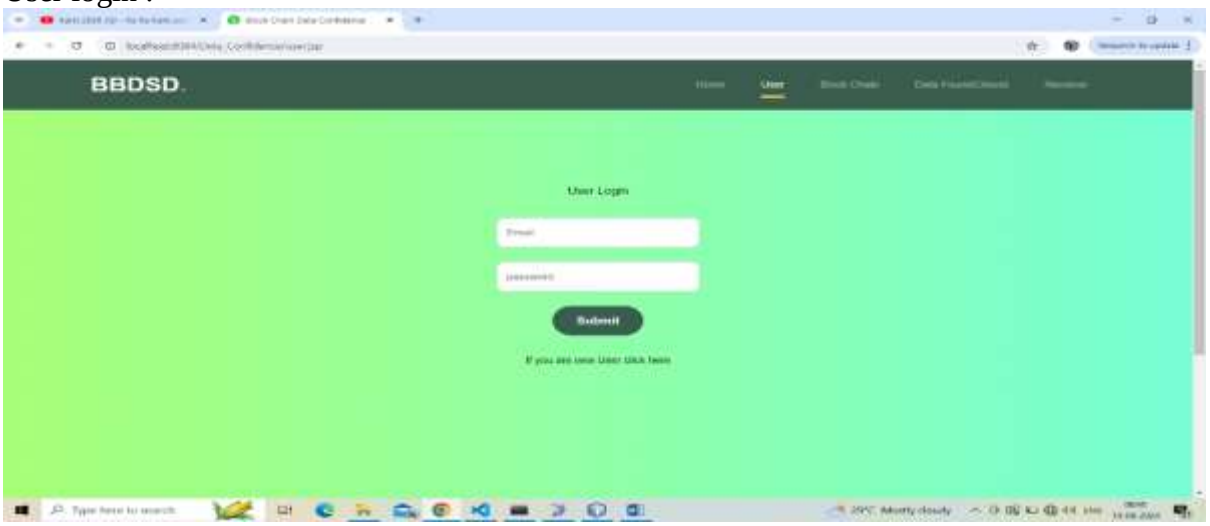
Home Page:



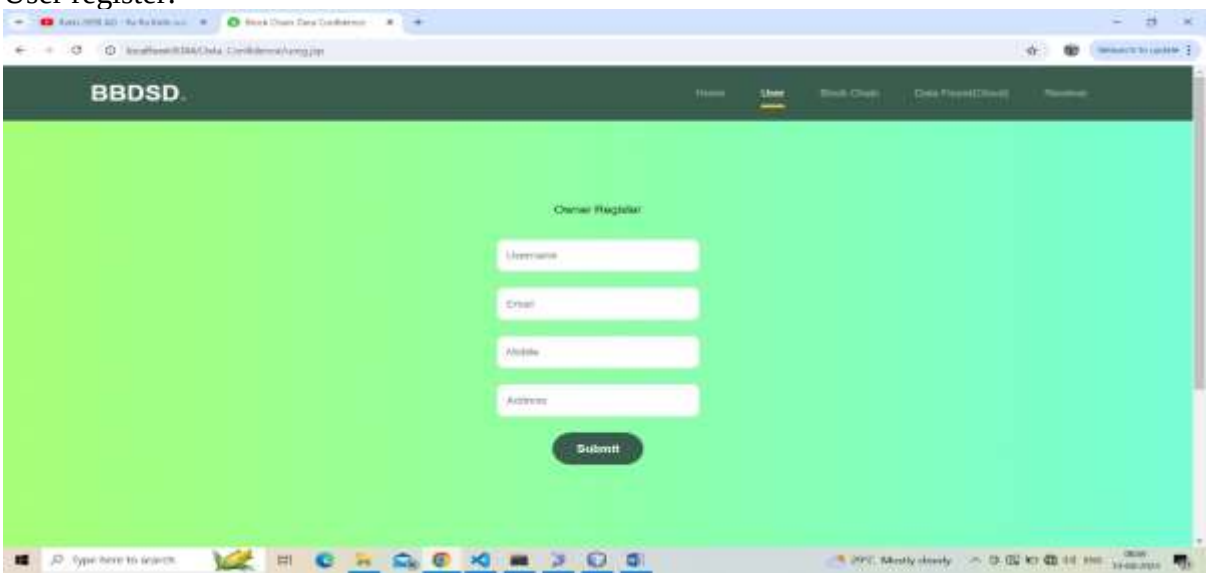
s

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 10, October : 2024

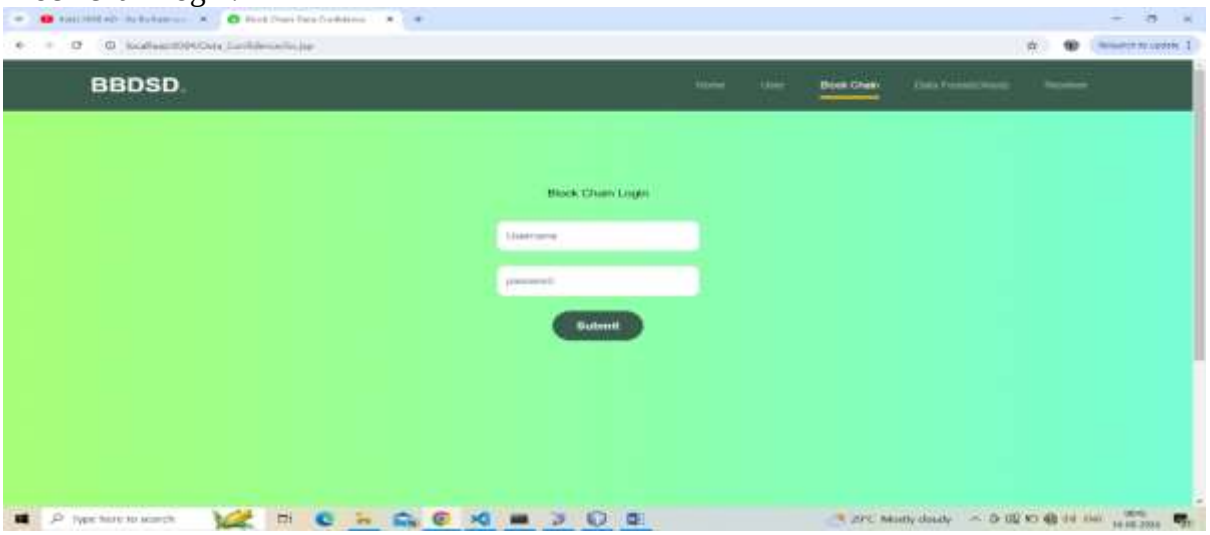
User login :



User register:

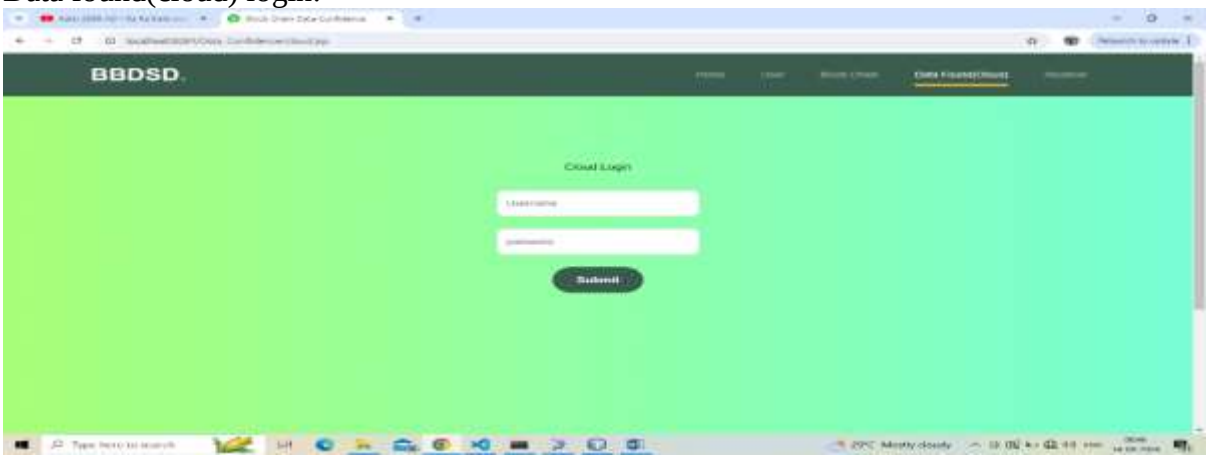


Block chain login:

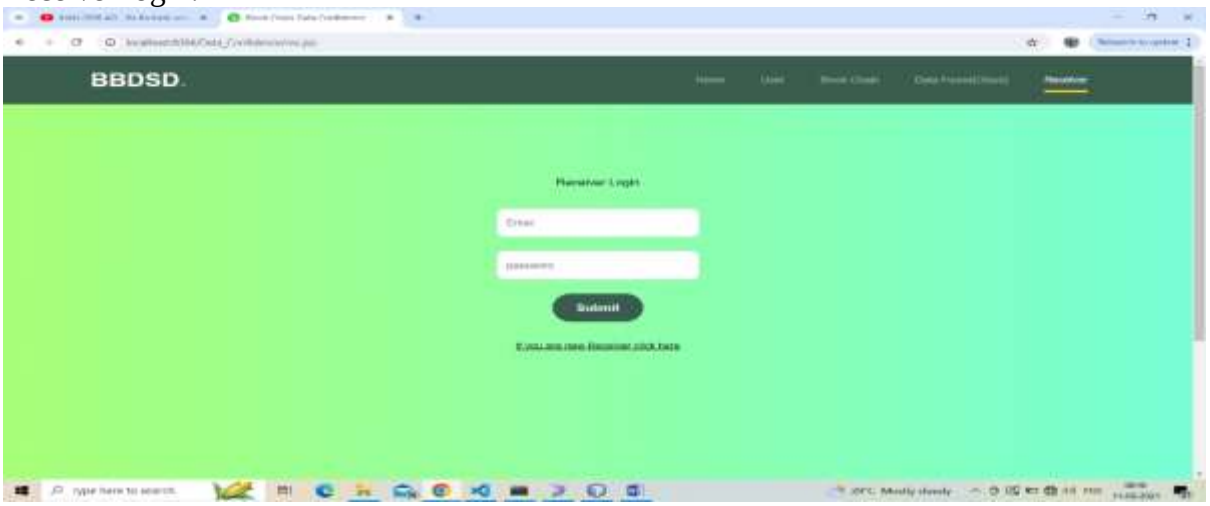


Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 10, October : 2024

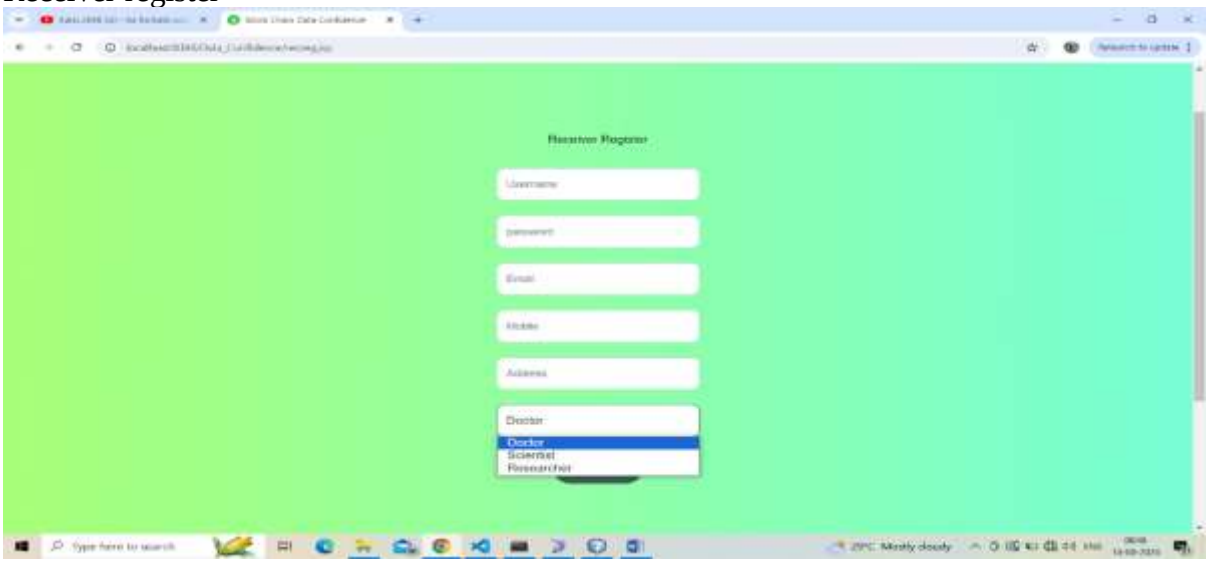
Data found(cloud) login:



Receiver login:



Receiver register



CONCLUSION:-

This paper introduces a decentralized storage system that integrates the strengths of blockchain

technology and cloud native concepts. With blockchain, we enhance data sharing security and transparent data validation. Meanwhile, cloud native concepts such as containerization and orchestration improve the system's scalability and flexibility. We evaluate the performance on a high-end edge computing infrastructure and show that the proposed system consistently outperforms IPFS in terms of data transfer speed. However, IPFS boasts of a peer-to-peer hypermedia protocol, which ensures that content is available even when nodes go offline and promotes fast and efficient data retrieval. Our intention is not to entirely replace IPFS but to offer an alternative open-source storage platform with distinct advantages. The proposed platform can work with IPFS or other systems, providing users with more flexible choices based on their specific needs.

FUTURE SCOPE:-

The future scope of blockchain-based decentralized storage for data confidence over cloud-native edge infrastructure includes:

1. **Enhanced Security**: Blockchain can provide robust encryption and immutable data records, improving security against tampering and unauthorized access.
2. **Data Sovereignty**: Decentralized storage ensures data ownership and control remain with users, aligning with emerging data protection regulations.
3. **Scalability**: As edge infrastructure grows, decentralized systems can scale dynamically without central bottlenecks, ensuring efficient data management.
4. **Interoperability**: Future advancements may enhance the ability of different blockchain systems to communicate and integrate seamlessly with various cloud-native services.
5. **Reduced Latency**: Storing data across a decentralized network can reduce latency, as data can be accessed from multiple edge locations rather than a central server.
6. **Cost Efficiency**: Leveraging decentralized storage could lower costs related to data storage and transfer by minimizing the reliance on centralized data centers.
7. **Resilience and Redundancy**: Blockchain-based storage can increase resilience through distributed nodes, reducing the risk of single points of failure and improving data availability.
8. **Smart Contracts**: The integration of smart contracts can automate and enforce data access rules and transactions, enhancing efficiency and trust in data management.
9. **Data Provenance**: Blockchain can provide transparent tracking of data origins and modifications, which is crucial for auditing and compliance.

REFERENCES:-

1. R. R. Krishna, A. Priyadarshini, A. V. Jha, B. Appasani, A. Srinivasulu, and N. Bizon, ''State-of-the-art review on IoT threats and attacks: Taxonomy, challenges and solutions,'' Sustainability, vol. 13, no. 16, p. 9463, Aug. 2021.
2. C. B. Tan, M. H. A. Hijazi, Y. Lim, and A. Gani, ''A survey on proof of retrievability for cloud data integrity and availability: Cloud storage state of-the-art, issues, solutions and future trends,'' J. Netw. Comput. Appl., vol. 110, pp. 75–86, May 2018.
3. M. Factor, K. Meth, D. Naor, O. Rodeh, and J. Satran, ''Object storage: The future building block for storage systems a position paper,'' in Proc. IEEE Int. Symp. Mass Storage Syst. Technol., Aug. 2005, pp. 119–123.
4. D. Guide, ''Amazon simple storage service,'' Tech. Rep., 2008.
5. V. Bucur, C. Dehelean, and L. Miclea, ''Object storage in the cloud and multi-cloud: State of the art and the research challenges,'' in Proc. IEEE Int. Conf. Autom., Quality Test., Robot. (AQTR),

May 2018, pp. 1–6.
6. J. Kosińska and K. Zieliński, ‘‘Autonomic management framework for cloud-native applications,’’ J. Grid Comput., vol. 18, no. 4, pp. 779–796, Dec. 2020.
7. A. Di Stefano, A. Di Stefano, and G. Morana, ‘‘Ananke: A framework for cloud-native applications smart orchestration,’’ in Proc. IEEE 29th Int. Conf. Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE), Sep. 2020, pp. 82–87.
8. W. Zhang, Y. Zhang, H. Fan, Y. Gao, and W. Dong, ‘‘A low-code development framework for cloud-native edge systems,’’ ACM Trans. Internet Technol., 2022
9. A. Makris, I. Kontopoulos, E. Psomakelis, S. N. Xyalis, T. Theodoropou los, and K. Tserpes, ‘‘Performance analysis of storage systems in edge computing infrastructures,’’ Appl. Sci., vol. 12, no. 17, p. 8923, 2022.
10. G. Cheng, D. Guo, L. Luo, J. Xia, and S. Gu, ‘‘LOFS: A lightweight online file storage strategy for effective data deduplication at network edge,’’ IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 10, pp. 2263–2276, Oct. 2022.