OPTIMIZED MACHINE LEARNING-BASED STORAGE ARCHITECTURE FOR SECURE DYNAMIC FILE ACCESS IN HADOOP BIG DATA ENVIRONMENTS

[1]Dr.P.L.N. Ramesh., [2]R.Ushadevi and [3]Mr.M. Viswanathan

[1]Dept. of Mechanical Engineering, Prathyusha Engineering College, Tiruvallur- 602025

[2,3] Dept. of Computer Science, SriKrishnasamy Arts & Science College, Sattur, Tamil Nadu.

ABSTRACT

The rapid expansion of big data applications demands more efficient and secure file storage solutions. Traditional file management techniques often fail to adequately handle both structured and unstructured data, especially in dynamic environments. This research introduces an advanced storage model integrated with machine learning and deep learning techniques to enhance Hadoop's Distributed File System (HDFS). The model incorporates a secure MapReduce framework with improved data trustworthiness, leveraging Apriori-based deep learning for data prediction and visualization techniques for dynamic access optimization.

The proposed system addresses the limitations of static authentication and rigid data storage policies by implementing a flexible, scalable solution within the MapReduce cloud environment. Additionally, it incorporates elastic net regression and infrastructure handling to support large-scale, distributed storage. This architecture aims to reduce computational complexity, enhance energy efficiency, and ensure data reliability through probability-based replication strategies. Ultimately, the model promotes secure, reliable, and scalable storage suitable for big data analytics.

Keywords: Hadoop Distributed File System (HDFS), Dynamic File Access, Machine Learning, MapReduce Architecture

INTRODUCTION

The process of turning data into science is difficult. One key issue is that today's data is big, dynamic, and diversified, originating from various sources and frequently lacking in

organization. The Hadoop Distributed File System (HDFS) is used as a data warehouse by the bulk of modern data analytics, management, and service tools and services; these tools may utilize Hadoop ecosystem services for processing. In terms of price/performance, Hadoop outperforms the competition. As a result of Hadoop's freedom to scale on data management issues, users operate inefficiently. According to Huang et al., users are paying less attention to how their scripts consume resources as a result of the manner it add machines to handle computational challenges, and many HDFS users assume the file system was created for batch processing. As a result, it's acceptable to leave scripts running in the background for an extended amount of time without considering the resources It take. The authors of Bajda- Pawlikowski et al. Adapt used structured data as an example of inefficiency and showed how It decreased it by a factor of 50. The bulk of enterprise data is semi-structured, multi-structured, and unstructured, according to Michael Walker. According to the International Data Corporation, the volume of digital data will increase by 40 to 50 percent per year (IDC). According to IDC, by 2020, the number will have risen to 40 Zettabytes. (ZB)

OVERVIEW OF THE PROJECT

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. Visual Studio.NET, there is no need to open the Enterprise Manager from SQL Server. Visual Studio.NET has the SQL Servers tab within the Server Explorer that gives a list of all the servers that are connected to those having SQL Server on them.

AIM OF THE PROJECT

We've been provided an estimate of the efficacy of the recommended model optimise state-of-the-art by using a data-aware Hadoop Distributed File System, as well as services operating on top of that MapReduce Distributed File A structure for storing resource descriptions Using Hadoop Distributed File System cluster-based data partitioning, the researchers altered the actual location of data in Hadoop Distributed File System processes in order to fit the graph Using Hadoop Distributed File System, researchers discovered that concurrent data queries required fewer resources.

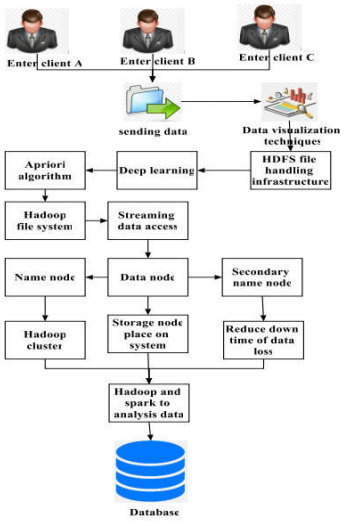
EXISTING SYSTEM

The file system's unit of work is the block size. Every read and write operation is performed in multiples of the block size. The block size is also the smallest file size that can be stored on disc. A file of 16 bytes fills a whole block if the block size is 16 bytes. If the replication files are higher than the number of block, the replication will continue according to block size. Otherwise the replication process is stopped. So the files are does not store in HDFS.

PROPOSED SYSTEM

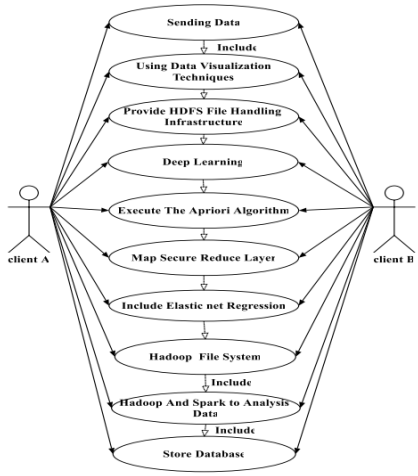
The proposed Applications for huge data storage are managed by modern technologies, which enhances application storage. File storage is made easier by big data surveys. There are no workable file management systems found in the survey. Current methods store sorted and unorganized files insecurely. Complex file management is necessary for big data analytics. This study uses secure reduction layers on maps. Particularly, the data trustworthiness of the Hadoop Distributed File System, its clear data storage policy, and its over-reliance on authentication procedures are becoming more and more apparent. For utilizing data visualization techniques and the apriori algorithm for Deep Learning prediction. Also used is the Map Reduce Distributed File System complexity reduction paradigm. To the best of our knowledge, this is an attempt to move the Hadoop Map Reduce framework to the Map Reduce cloud in order to handle the issues that come with a dynamic network environment Hadoop. An erasure code is employed to guarantee the reliability of the few working files that remain. Build an adaptive replication management system to provide high availability for data in cloud storage and raise the data locality measure. As a result, the extremely local data available boosts the Hadoop system's performance. Test the practicality of the proposed methodology by implementing adaptive replication management in cloud storage and estimating its effectiveness.

SYSTEM ARCHITECTURE



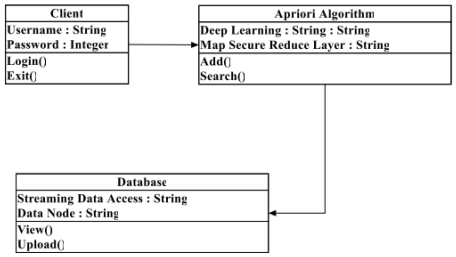
USECASE DIAGRAMS

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. That use cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.



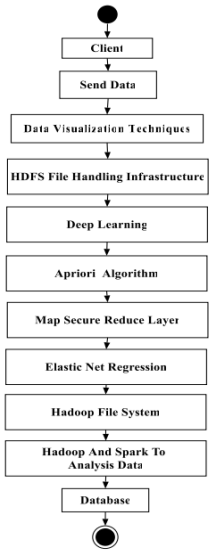
CLASS DIAGRAM

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.



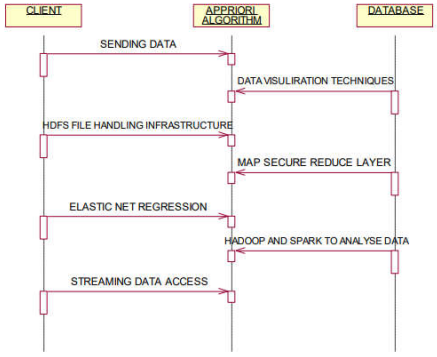
ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.



SEQUENCE DIAGRAM

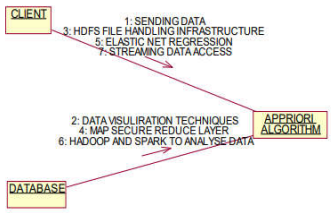
The Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case. With the advanced visual modeling capability, it can create complex sequence diagram in few clicks. Besides, VP-UML can generate sequence diagram from the flow of events which it has defined in the use case description. The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.



COLLABORATION DIAGRAM

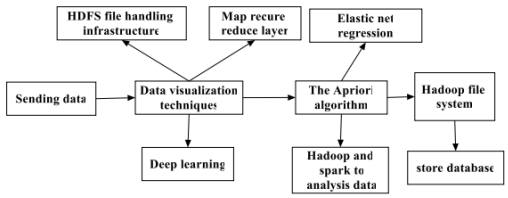
A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. A Communication diagram models the interactions between objects or parts in terms of sequenced messages. Communication diagrams represent a combination of

information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.



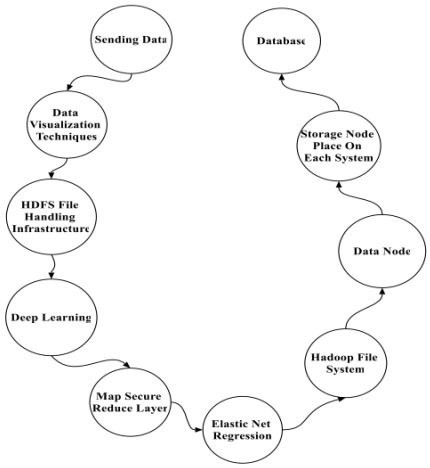
ER DIAGRAM

An entity–relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types). In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database.



DATAFLOW DIAGRAM

A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.



PROPOSED ALGORITHM

DEEP LEARNING IN APRIORI ALGORITHM

Active Datanodes DNA{DNA1, DNA2, … , DNAp}

Standby Datanodes DNS{DNS1, DNS2, …, DNSq}

Default replication factor rD (0 < rD < p)

Replica number currently in use r (0 < r < p+q)

Block B ∈ Data D

To place B, select Datanode DN.

if B = Coding Block

for DNAi in DNA

If DNAi has the fewest number of D's blocks,

DN DNAi

return DN

end if

end for

else if B = Data Block

if r < rD

for DNAi in DNA

if DNAi is appropriate for the replica placement approach by default

return DN

end if

end for

else if r >= rD

for DNSi in DNS

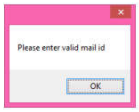
return DN

end if

end for and end if

UNIT TESTING

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.



CONCLUSION

Also used is the Map Reduce Distributed File System complexity reduction paradigm. To the best of our knowledge, this is an attempt to move the Hadoop Map Reduce framework to the Map Reduce cloud in order to handle the issues that come with a dynamic network environment. The principal storage system utilized by Hadoop applications is the Hadoop Distributed File System. The quick data flow between nodes is how this open-source architecture operates. Businesses that need to process and store large amounts of data frequently employ it. Elastic net regression and infrastructure handling are involved. Hadoop is an open-source Java-based framework that controls how big data is processed and stored for use in applications. Hadoop handles big data and analytics tasks by utilizing distributed storage and parallel processing, which reduces workloads. This open source framework works by rapidly transferring data between nodes. It often used by companies who need to handle and store big data. It is Handling Infrastructure and elastic net Regression. Hadoop is an open source framework based on Java that manages the storage and processing of large amounts of data for applications. Hadoop uses distributed storage and parallel processing to handle big data and analytics jobs, breaking workloads down into smaller workloads that can be run at the same time. Map Reduce is a distributed computing architecture that guarantees significant energy savings, data dependability, and security guarantees for processing large datasets.

REFERENCES

1) An Efficient Method of Data Encryption for Securing HDFS Shivani Awasthi; Narendra Kohli_2023

2) A Distributed Cache Mechanism of HDFS to Improve Learning Performance for Deep Reinforcement Learning Yongqiang Gao_2022

3) Applications of Machine Learning Algorithms for HDFS Big Data Security K. Rajesh Kumar; S. Dhanasekaran_2022

4) An Approach To Secure Sensitive Attributes Stored On HDFS Using Blowfish Anju Kaushik; Vinod Kumar Srivastava_2021

5) Performance Analysis Of AES And DESede On The Sensitive Data Stored In HDFS Anju Kaushik_2020

6) CORE: Cross-Object Redundancy for Efficient Data Repair in Storage Systems: Kyumars Sheykh Esmaili and Lluis Pamies-Juarez, March 2016

7) Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling: Matei Zaharia, March 2016

8) Map-reduce Implementations: Survey and Performance Comparison: Zeba Khanam and Shafali Agarwal, December 2015

9) Gaussian process for predicting CPU utilization and its application to energy efficiency: Dinh-Mao Bui , Huu-Quoc Nguyen, YongIk Yoon and SungIk Jun, October 2015

10) Big Data and Hadoop: Rakesh Rathi, Sandhya Lohiya 2014 11) Efficient Updates in Cross-Object Erasure-Coded Storage Systems: Kyumars Sheykh Esmaili, Aatish Chiniah and Anwitaman Datt, March 2013