iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** IV **Month of publication:** April 2021

DOI: https://doi.org/10.22214/ijraset.2021.33733

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IV Apr 2021- Available at www.ijraset.com

Repair Pipelining Mechanism with Hybrid Erasure Coding for Block Chain

Ms. M. Bavithra[1], Ms. P. Subithra[2]
[1]*Assistant Professor, Department of MCA, Dhnalakshmi Srinivasan EngineeringCollege, Perambalur, Tamil Nadu.*
[2]*Master of computer Applications, Dhanalakshmi Srinivasan EngineeringCollege, Perambalur, Tamil Nadu.*

***Abstract:** Yet another drawback is that this mechanism may limit the throughput in permissioned block chain. Moreover, due to the existence of Byzantine nodes, existing partitioning methods, though widely adopted in distributed systems for decades, cannot suit for block chain systems directly, so that it is critical to devise new storage mechanism for block chain systems. In existing system first property of BFT-store is that the storage consumption per block can be reduced to O (1) for the first time, which enlarges overall storage capability when more nodes attend the block chain. Second, an efficient online re-encoding protocol for storage scale-out and a hybrid replication scheme to enhance reading performance. This erasure coding is not that much efficient due to repair time for decoding is high. In proposed system a repair pipelining mechanism is used in byzantine environment to reduce repair time for decoding. This increases the bit storage performance rapidly.*
***Keywords:** Block chain, Byzantine fault tolerance, Hybrid erasure coding and replicating*

I. INTRODUCTION

Blockchain is a distributed append-only ledger maintained by all nodes in an untrustworthy environment. In order to avoid being tampered by malicious nodes, blocks in the blockchain are chained by cryptograph hash values in the block headers. In general, each node holds a complete copy of block data, and consensus protocol, such as PoW in permission less blockchain and PBFT (Practical Byzantine Fault Tolerance) in permissioned blockchain, ensures the data consistency among whole network. Thus the overall storage consumption per block is O(n), where n is the numberof nodes. However, with this full-replication manner, each node has to devote huge storage space to preserving blocks, e.g., the total amount of block data of Bitcoin is over 200 GB right now and the Ethereum network generates about 0.2 GB data every day 1. Permissioned blockchains are designed to support applications beyond crypto currency such as finance management, banking and insurance, where participants are authenticated and do not fully trust each other and everyone has the potential to do malicious things for its own benefit. Commonly, permissioned blockchains adopt PBFT protocol to achieve higher transaction throughput ,thus the amount of block data will break through the storage capacity of single node. A natural, but challenging issue arises to be: can we lower the storage consumption but still be capable of recovering the whole blockchain at any time? To address above, some techniques are introduced, such as light-weight client and lightning network. Each light-weight client just saves block headers rather than entire blockchain to verify block bodies received from full nodes. However, this technique just reduces the overhead 1. of client while the full node still reserves a complete copy of all block data. The basic idea of lightning network is to restrain the volume of block data by gathering multiple micropayments between two accounts and rarely uploading the final balances to blockchain. Although this approach eases the stress of storage for each node, it does not subvert the full-replication mechanism fundamentally.

The sharding technique divides entire network into multiple groups, each of which stores a part of data. However, in same group, each node retains a full copy of partial block data as well. Furthermore, due to the scale of blockchain becomes smaller, the throughput of each group increases correspondingly and blocks are generated at a faster speed, which adds the storage overhead to each node. A straightforward concern is whether the partitioning manner (e.g., hash or range partitioning), which is widely adopted in distributed systems, suits for blockchain systems or not. In this mechanism, the original data is divided into a number of partitions and dispatched to all nodes. Particularly, for high availability, each partition is replicated for several times to tolerate the fault of single node. However, this mechanism cannot suit for blockchain due to the existence of the malicious nodes who may keep silent or tamper data deliberately. For example, if all replicas of a block happen to be dispatched to malicious nodes, this block may be tempered, or even lost. To ensure each block can be recovered, the only way is to generate more replicas, at least more than the potential malicious nodes. However, it will be too expensive as well. Alternatively, erasure coding(EC) provides significantly lower storage overhead than multiple-replication mannerat the same fault tolerance level.

A. Proposed Algorithm

In proposed algorithm a repair pipelining mechanism is used in byzantine environment to reduce repair time for decoding. This increases the bit storage performance rapidly.

Focal Point

1) This erasure coding is enhanced and effective.
2) The repair time for decoding is reduced.
3) Pipelining mechanism performance is high.

II. ENVIRONMENT DETAILS

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use.

1) Software Specification
a) Front end = java 8 and above
b) Back end = mysql 5 and above
c) Tool = netbeans 8.0 and above
2) Hardware Specification
a) Processor = intel i3 and above
b) Ram = 4 gb and above
c) Hard disk = 250 gb and above

A. System Architecture

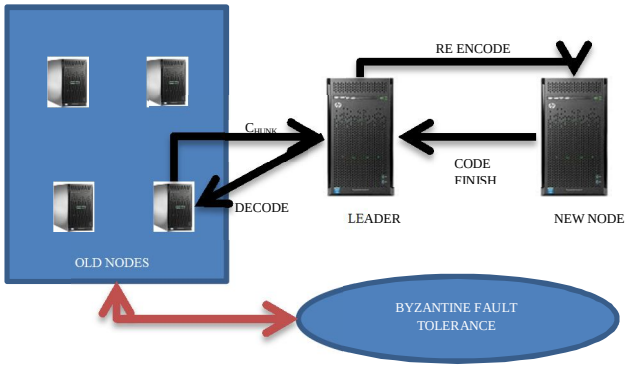
1) Block Chain Creation
a) RS ENGINE IS CREATED
b) Encoder buffers the blocks received from consensus, and encodes them into chunks. Each node stores a fragment of all chunks and discards the rest chunks.
c) Decoder pulls chunks from others through underlying network and recovers blocks by RS decoding.

2) Replication And Erasure Coding Technique
a) All nodes are sorted by their public keys and the position in this list is treated as the global index for each node.
b) In this way, the sorted node list is identical among all honest nodes because each node knows all others' public keys.
c) Erasure code is done to re-encode data.

3) Recovery Engine And Scale Out Engine: The arrival of a new node means redistribution of current data, i.e, historical blocks need to be recoded among all nodes. However, since no node preserves entire blockchain, it is challenging to ensure that all correct nodes have accomplished reencodingin distributed system with Byzantine fault.
a) Crashed nodes are recovered through recovery engine.
b) The crashed node will take the initiative to recover block data once it is restarted.
c) Scale-out engine starts to work once new nodes join in blockchain system

4) Pipeline Repairing In Bft-Store: BFT-Store into an open-source blockchain system Tendermint, and the extensive experiments confirm the scalability, availability and efficiency of BFT-Store compared with full replication manner.
a) Byzantine fault tolerance is used to store data.
b) Bft reduce storage consumption in each block.
c) Online re-encoding protocol is done
d) Pipeline repairing is carried out for fast decoding.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IV Apr 2021- Available at www.ijraset.com



B. Software Overview

Java is a platform Independent. Java is a high level programming language Introduced by Sun Microsystems in June 1995 Java is becoming a standard for Internet Applications. It provides for interactive processing and for the use of graphics and animation on the Internet. Since the Internet consists of different types of computers and operating systems, a common language was needed to enable computers to run programs that run on multiple platforms. Java is an object oriented language built upon C and C++.It derives its syntax from C and its object-oriented features are influenced by C++. Java can be used to create applications and applets. An application is a program that runs on the user's computer, under its operating system. An applet is a small window based program that runs on HTML page using Java enabled browser like Internet Explorer, Netscape Navigator, Hot Java or an applet view

1) Features of JAVA

Simple

a) Java Language constructs are easy to learn and use. It takes care of memory management. Though
b) Java was developed from C++, the complexities associated with C++ have been eliminated in Java.
2) Object-Oriented: Java is designed around the object-oriented model. In Java the focus is on the 'data' and the 'methods' that operate on the data in an application and not just on the procedures. The data and methods together describe the state and the behaviour of an object in Java.
3) Robust: Java is a robust language since it has strict compile time and run time checking of code. This minimizes programming errors. Error handling and recovery is taken care of in Java by the 'exception- handling' feature.
4) Secure: Java is language that focuses on the network. Java security features ensure that its programs that run are safe. Programmers cannot manipulate memory in Java. This is a good defense mechanism against malicious code that may flow in from the network. Java programs running on the Web cannot open, read, write or delete files on the user's system or run other programs on it.
5) Distributed: Java can be used to develop applications that are portable across multiple platforms, operating systems and graphical user interfaces. Java is designed to support network applications. Thus Java is widely used tool in an environment like the Internet where there are different platforms.
6) Multithreaded: Java programs can do many tasks simultaneously by a process called 'multithreading'. Java provides the master solution for synchronizing multiple processes. Therefore, interactive applications on the Net can run smoothly. This is made possible by the built-in support for threads.
a) Running Java File with single command
b) New utility methods in String class
c) Local-Variable Syntax for Lambda Parameters
d) Nested Based Access Control
e) HTTP Client
f) Reading/Writing Strings to and from the Files
g) Flight Recorder

C. Technology Infrastructure

Core Java: Java can be used to create two types of programs: application and applet. An application is a program that runs on your computer, under the operating system of that computer. That is, an application created by java is more or less like one created using C or C++. When used to create application, java is not much different from any other computer language. Rather, it is java's ability to create applets that makes it important. An applet is an application designed to be transmitted over the internet and executed by a java-compatible Web Browser. An applet is actually a tiny java program, dynamically downloaded across the network, just like an image, sound file, or video clip. The important difference is that an applet is an intelligent program, not just an animation or media file. In other words, an applet is a program that can react to user input and dynamically change-not just run the same animation or sound over and over.

D. Developing Metholodies

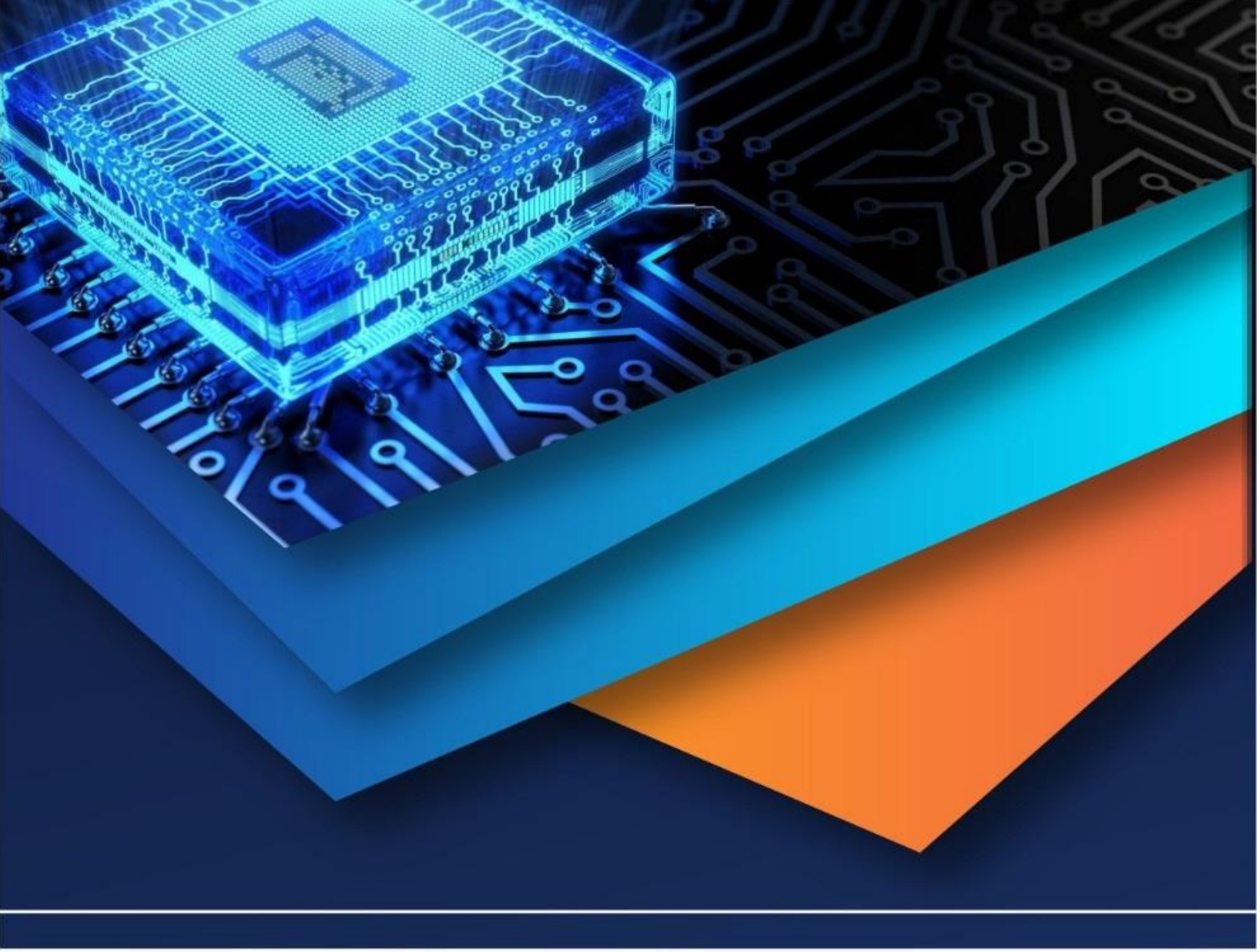
The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

III. CONCLUSION

Full-replication manner without any scalability leads to the waste of huge storage, which is a serious problem in blockchain systems. The combination of PBFT with erasure coding provides a chance to partition blocks over all node I a hostile environment. Our contributions include: (i) exploit erasure coding for blockchain to reduce storage complexity per block to O(1); (ii) propose a hybrid erasure coding and replicating to improve reading performance over encoded partition; (iii) design a re-encoding approach to deal with addition or removal of nodes. However, the RS coding employed in this study is not an efficient erasure coding. In the future, we will design repair pipelining mechanism in Byzantine environment to reduce repair time for decoding, which can improve the performance of BFT-store further.

REFERENCES

[1] Hyperledger. https://www.hyperledger.org.
[2] D. Boneh, C. Gentry, B. Lynn, H. Shacham, et al. A survey of two signature aggregation techniques. RSA cryptobytes, 6(2):1–10, 2003.
[3] E. Buchman. Tendermint: Byzantine fault tolerance in the age of blockchains. PhD thesis, 2016.
[4] N. Budhiraja, K. Marzullo, et al. The primary-backup approach. Distributed systems, 2:199–216, 1993.
[5] C. Burchert et al. Scalable funding of bitcoin micropayment channel networks. Royal Society open science, 5(8):180089, 2018
[6] V. Buterin et al. A next-generation smart contract and decentralized application platform. white paper, 2014.
[7] M. Castro, B. Liskov, et al. Practical byzantine fault tolerance. In OSDI, volume 99, pages 173–186, 1999.
[8] H. Chen, H. Zhang, M. Dong, et al. Efficient and availablein-memory kv-store with hybrid erasure coding and replication. TOS, 13(3):25, 2017.
[9] H. Dang et al. Towards scaling blockchain systems via sharding. In SIGMOD, pages 123–140. ACM, 2019.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY