

# Ledger Archiving

Santanu Mukherjee  
Blockchain CoE  
Mindtree Ltd.  
Bengaluru, India  
santanu.mukherjee@mindtree.com

Teja Surisetty  
Blockchain CoE  
Mindtree Ltd.  
Bengaluru, India  
teja.surisetty2@mindtree.com

Divya Bhanu Althi  
Blockchain CoE  
Mindtree Ltd.  
Bengaluru, India  
divya.althi@mindtree.com

**Abstract**— an approach to take the backup of the ledgers in Blockchain applications. This accelerator will have a benefit of taking backup of filled data in the peer and can read the data whenever required. In Private Block chain network like Hyper Ledger Fabric there is no such methods of archiving the data, so using this approach data can be safe guarded without taking the network down. A detailed description of the same is mentioned below.

**Keywords**— Blockchain, Ledger Archiving, Hyperledger Fabric, Blockchain accelerator

## Introduction

While Blockchain may appear similar to distributed databases, they are typically implemented without a central authority and central repository. Therefore, Blockchain provide some unique differences from everything that has come before. A Blockchain survives faults and attacks by using redundant checking at multiple nodes. This resiliency goes far beyond replication, since it happens across the network without any central coordinator or intermediary.

The Main storage element for a Blockchain network is the Ledger. A Ledger contains huge number of blocks and each block has some transactions associated with it. As the block height gets increased there will be more space consumption and there is a chance of filling up the allocated space of a peer running in a network. This will lead to a major problem as that peer will not be able to store anymore data. In-order to overcome this Archiving of the data is required but as of now, there aren't any traditional methods readily available. This approach will enable for easier Backup/Archiving of data from the ledger to an external storage system and query whenever required.

The approach is to take the block file (which stores transient data) from the peer's node and store it in IPFS as chunks. The hash of the document and minimal relevant data of is stored in the ledger by performing a transaction. This way the storage capacity of a specific peer can be reduced to almost zero. Since the information for the archived data is already available in the ledger, the same can be used to fetch the archived data from the IPFS.

## I. HOW DOES IT WORK

### A. High level understanding

The implementation here is explained by setting up a simple Hyperledger fabric network which consists of two peers each from Org A and Org B. Both peers are added to a single channel named My Channel. Both peers are configured to use Couch DB as the World State DB (which stored the assets/contracts) and Level DB (to store

the block information). This is just an example implementation, as the same approach can be implemented on a much complex network as well. Once a generic network is created and at some point the storage of the ledger has filled up, the archiving process can be initiated to start backing up the data to an external network (IPFS). The ledger data stored in the location is then taken out and pushed to IPFS to safeguard and use it for future purpose.

### B. What is IPFS? and how can we make a private network in IPFS?

IPFS(interplanetary File system)is a distributed system for storing and accessing files, IPFS is a file sharing system that can be leveraged to more efficiently store and share large files. It relies on cryptographic hashes that can easily be stored on a blockchain.one more advantage with this IPFS is likewise Blockchain it has both public and private networks where we can limit users who can access the files that are stored in IPFS

Establishing a private network in IPFS: consider u have two different IP address which u want them to involve in your private IPFS network and with that IP's if u request in IPFS you will be able to retrieve the file ,you can establish this private network by installing a swarm key.

### C. Some Challenges

- The format of the block file cannot be read by anyone outside the Blockchain network but within the network have to keep in exact location otherwise from nowhere we can get the transient data
- There is no external Reader for block files in Hyperledger Fabric.
- Cannot make a peer pause action until and unless the peer is in offline state, in order to make peer pause we have to manually make the changes in Docker compose file and issue the command
- In LA we have to copy from container to local system there is a chance where system get stuck with large amount of data and sometimes may not support for that we might need an interface like cloud before adding to IPFS
- We should make sure no peer is invoking a data in world state that involves with backup information hash of IPFS, even though there is update in hash transaction we can retrieve the block file using transaction history but it involves more effort to fetch and retrieve the root transaction

#### D. Implementation

- For the implementation, consider a test-network of fabric-samples with sample configurations that can be found in the fabric-samples repository (<https://github.com/hyperledger/fabric-samples>). The smart contract used here is also an example named asset-transfer-basic which is written in JavaScript.
- The next step would be creating a custom contract to manage the archival mechanism. This can be completely customizable based on the requirement.
- This method will change the world state composite keys and make sure all the assets are same as before with different composite keys. After invoking the method, the composite keys will be updated to a newer version, and all the older composite keys as well as assets will be removed. All the newer transactions will be added with new ledger value for easy recognition between archived and non-archived data.
- Now the tricky part, were block file is extracted from the peer's ledger and stores in IPFS as chunks. A hash will be generated which then is added as an asset in the ledger. Also some relevant data based on the use case us appended to the hash for future use.
- Whenever archived data is required the same hash and relevant data can be used by the peer having appropriate authorizations to perform the query and retrieve the data.
- This entire process can be made without having a downtime at any given point of time. Just have to make sure the right commands are used when required.

- This approach was originally tested on block size ranging from 10-10000. The same can be implemented on any network regardless of the block size.

#### E. Acronyms used in this paper

- LA – Ledger Archiving
- HLF – Hyper Ledger Fabric
- IPFS – Inter planetary file system
- DB – Database

#### REFERENCES