Technical Overview: block Archiver feature

Introduction of Ledger Archiving

Table of Contents

Introduction

This document provides a high-level overview of **the block Archiver** feature for Hyperledger Fabric - Why it was developed, how it works and what the advantages are. This document is aimed primarily at Blockchain consortium administrators who have been running their own network and have resource shortage problems.

Blockchain solutions are suited well for long term operation due to their fundamental characteristics such as traceability, immutability, transparency and provenance which provide significant business benefits.

When considering the long term operation, there are several factors that need to be considered. Scalability is one of the most important factors for future business expansion while resource planning is critical in determining future capability.

Calculating the required disk space is one of the key challenges faced by users deploying large scale solutions that use Hyperledger Fabric.
In the Blockchain community, there are a lot options to address this issue. In Bitcoin, Lightweight Node has already been introduced as an option for non-miners which download just the block headers so as to reduce disk space usage. Ethereum, the community has defined the Light client protocol for almost the same purpose as bitcoin's one. While in Hyperledger Fabric, there is a feature request for this issue as well.

The lack of a block archiving feature in Hyperledger Fabric makes capacity planning of resources difficult and is a significant barrier for the adoption of Hyperledger technology in large scale business applications. This is one of the main reasons why most of the business stakeholders have considered making use of Blockchain as a Service (BaaS) managed by the cloud service providers despite it incurring more costs.

Reference:
Lightweight node - Bitcoin Wiki
https://en.bitcoin.it/wiki/Lightweight_node
Light client protocol · ethereum/wiki Wiki
https://etherworld.co/2018/03/13/understanding-ethereum-light-node/

-> https://github.com/ethereum/wiki/wiki/Light-client-protocol

[FAB-106] ledger checkpoint and pruning/archiving - Hyperledger JIRA
https://jira.hyperledger.org/browse/FAB-106

Solution

We are introducing a **block Archiver** feature in Hyperledger Fabric to addresses the ever expanding disk storage issue in large scale applications.

In this solution, data is handled in the unit of a certain amount of chunk of blocks called blockfiles. By default in Hyperledger Fabric, each peer node stores block data into the blockfiles one by one from the genesis block to the latest one. So they have the exact same series of blockfiles through all peer nodes which are connecting to the same channel. The **block Archiver** feature uses this characteristic to realize our concept.

This functionality makes redundant the notion that every peer node is required to retain all blockfiles in the local file system indefinitely. And more importantly this preserves the Blockchain characteristics even after archiving the blockfiles.

By enabling the block archiving feature not all peer nodes need to keep maintaining all the blockfiles. Simply, even devices which don't have much disk space could be a peer node of the blockchain network. For example, in an IoT system, you would be able to construct a blockchain network using edge devices with attached sensors in order to store sensor data and track information. This feature will make the target areas of the blockchain system broader than ever. Therefore the **block Archiver** feature provides new possibilities for Blockchain applications

Functionality

In essence, the **block Archiver** feature provides the functionality to reduce storage consumption in a Blockchain network. The **block Archiver** feature is based on the following four central ideas:

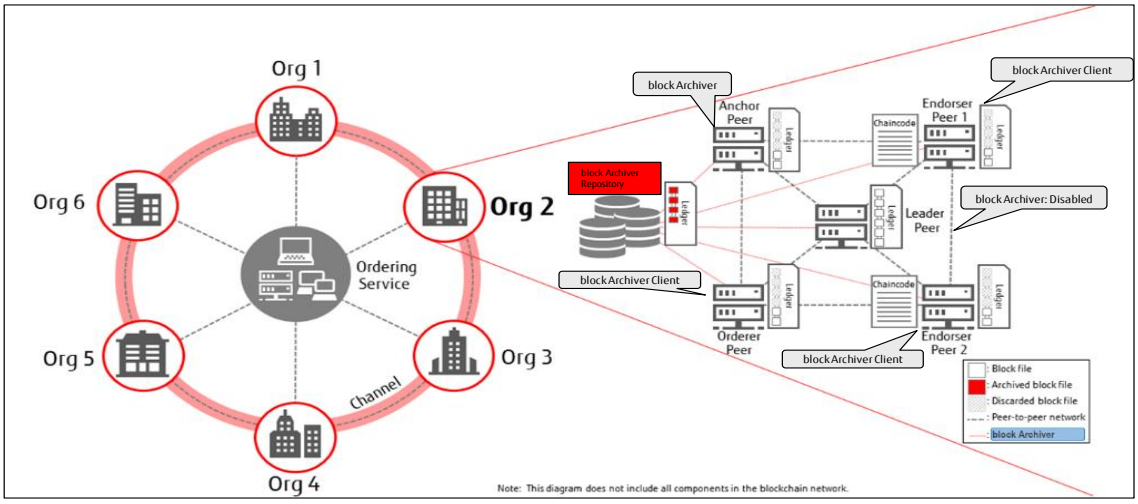
1. In a blockchain network made of multiple organizations where each organization has several peer nodes, the organizations which have enabled the **block Archiver** feature will have its own repository to store archived blockfiles.
2. Whenever a new blockfile is created in the file system and certain pre-configured criteria are satisfied, the data archiving mechanism is triggered automatically. The data archiving mechanism involves simply transferring the blockfiles in bulk to the repository without any data compression.
3. On each peer, archived blockfiles are discarded to increase available disk space on the local file system.
4. It's configurable whether blockfiles are to be discarded or not on each peer node.

Each peer node is still able to access all the blockfiles (including archived/discarded ones) seamlessly. Even after archiving blockfiles, blockchain characteristics are still maintained on the Blockchain network. For example, if a Blockchain network of 20 organizations enable the **block Archiver** feature, the Blockchain data will continue to be distributed among those 20 organizations.

The **block Archiver feature consist of the following three types of node roles:**

1. **block Archiver**
   This role needs to coexist with the fabric original peer node roles (anchor / leader) - It is responsible for archiving the growing blockfiles from its local file system to the repository and notifies via gossip message protocol to the other **block Archiver** client nodes within the organization when completing each transfer.
2. **block Archiver Client**
   This role also needs to coexist with the fabric original peer node roles
   When notified by the **block Archiver**, it discards the archived blockfiles from its local file system
3. **block Archiver Repository**
   The repository that the archived blockfiles are stored
   It's running off the chain and is totally independent of the fabric original peer node roles.

Fig 1: Overview of the network topology regarding **block Archiver** enabled Blockchain



Technical Advantages

The **block Archiver feature** drastically reduces the disk usage in Blockchain network while maintaining the Blockchain characteristics simultaneously.

In this section, we'll explain it from the following 3 aspects of performance, security, and robustness.

1. **Performance**
   Since only finalised blockfiles are archived, there is no effect on the performance in updating the current blockfile. Similarly, there is no effect on the performance in querying the blockchain as state-db is not touched by the **block Archiver**. Of course, there is network overhead to retrieve the archived blockfiles when you want to verify data consistency of the entire Blockchain data or when a brand new peer node joins to the blockchain network. In such cases, we can look to improve performance by combining with other potential solutions, like a checkpoint concept.

2. **Security**
   In Hyperledger Fabric network, the number of peer nodes within each organization does not affect the tamper-proof capability of the channel (or consortium) - It only provides high resilience and high reliability to access the Blockchain from the client application within the organization. So partially relocating blockfiles and decreasing redundancy of them does not provide any degradation in term of the tamper-proof capabilities.

3. **Robustness**
   As mentioned earlier, with **block Archiver**, the blockfiles are still distributed among the blockchain network at the organization-level. Even if archived blockfiles are lost for an organization, they could be restored by obtaining them from other organizations.

Business Advantages

Starting on a Small Scale

You can choose anywhere as a **block Archiver** repository as long as the place is reachable from each peer node. The **block Archiver** repository doesn't need to work as part of the blockchain. It runs in off chain.

Therefore, when launching the Blockchain service, you don't need to plan the future storage capacity of your network in detail and assign the storage resources required for future scaling - You can start your service on the Blockchain without this complex resource planning.

A critical advantage of the **block Archiver** is to remove the barriers that can impede a simple start to a blockchain business and thereby enables the option of starting your business on a small scale.

Reducing running costs

The long-term costs required to keep operating your blockchain is reduced since the required disk space for the system is drastically reduced.

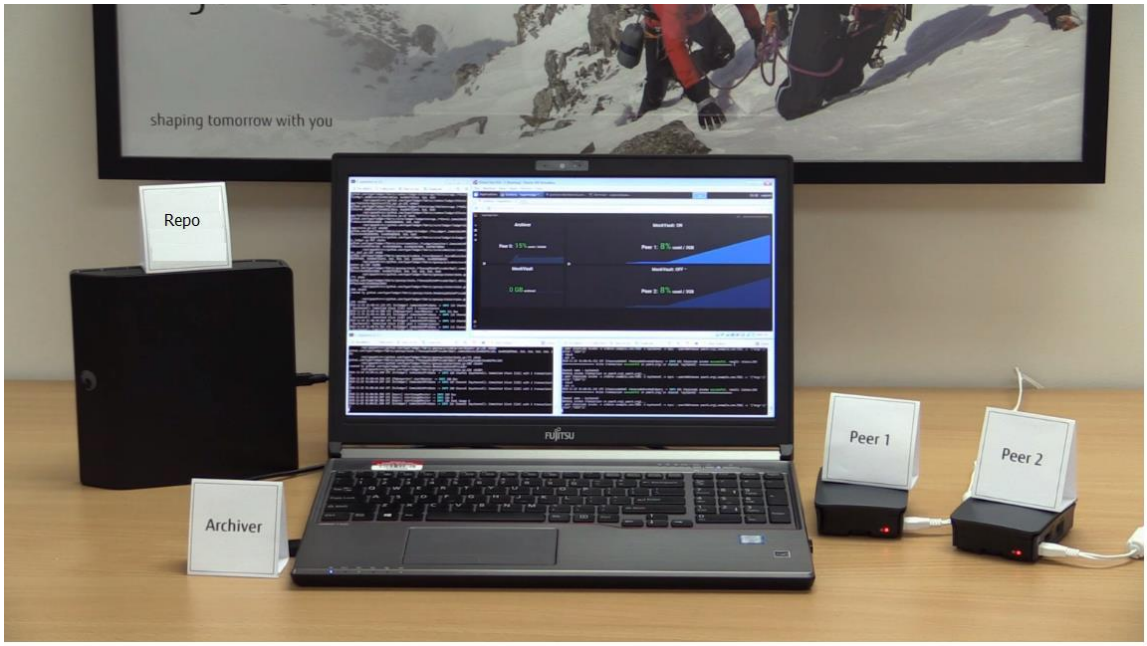
In the following environment, we can reduce 50% disk usage in the organization compared to the environment without the **block Archiver**.

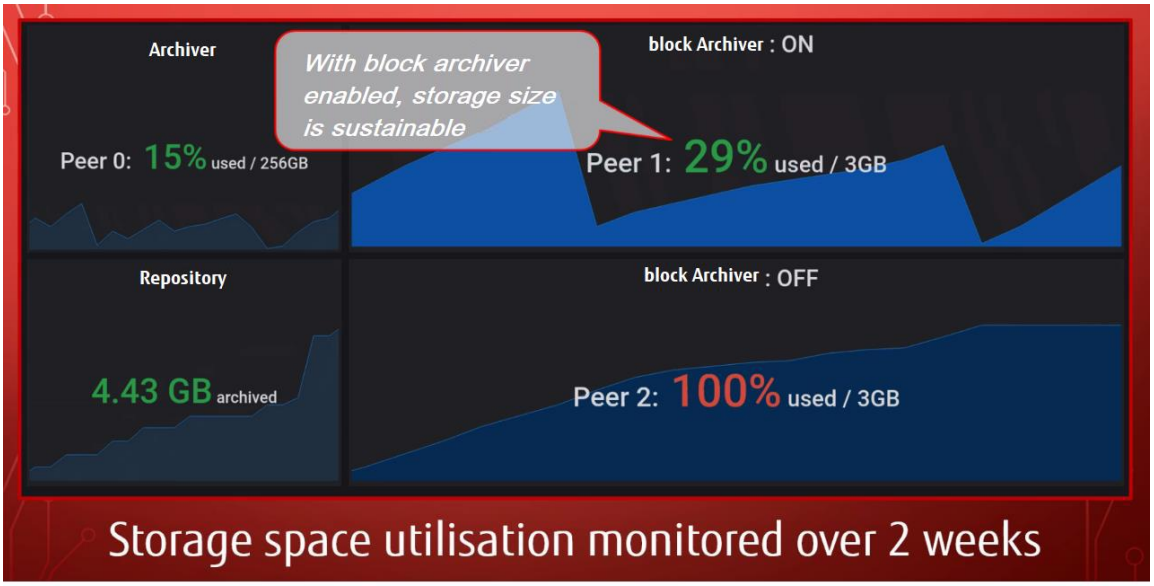
Node type	Number of nodes	disk usage before	disk usage after
Orderer	1	1	1
Peer node with **block Archiver**	4	4	0 (*)
Peer node without **block Archiver**	1	1	1
block Archiver Repository	1	0	1

(*) Fig 2: The disk usage of a **block Archiver** enabled peer node is approximated to 0 in long term running.

Increasing Applicability

- Blockchain applicability to be expanded across environments which have a constraint with disk storage. For example, in IoT system environment, the edge devices don't have large disk resource. If they utilised the **block Archiver** it would enable deployment of Blockchain across a wider range of environments including small devices like Raspberry Pi which can be a Blockchain peer node by simply using the **block Archiver**.





Conclusion

In this document, we have outlined one of the key demands community users have and our solution to this issue. The **block Archiver** solution demonstrates how effective it is for reducing storage consumption which can lead to increasing options for applying Blockchain to a wide range of use cases. While challenges may still exist **block Archiver** is helping to resolve one of those key challenges, thereby by helping to increase the adoption of Blockchain.