



HAL
open science

A Distributed Supply Chain Architecture Based on Blockchain Technology

Peng Zhao, Shiren Ye

► **To cite this version:**

Peng Zhao, Shiren Ye. A Distributed Supply Chain Architecture Based on Blockchain Technology. 12th International Conference on Intelligent Information Processing (IIP), May 2022, Qingdao, China. pp.260-273, 10.1007/978-3-031-03948-5_21 . hal-04178753

HAL Id: hal-04178753

<https://inria.hal.science/hal-04178753v1>

Submitted on 8 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

A Distributed Supply Chain Architecture Based on Blockchain Technology

Peng Zhao¹ and Shiren Ye²

¹Changzhou University, Jiangsu Changzhou 213000, China

²Changzhou University, Jiangsu Changzhou 213000, China
yes@cczu.edu.cn

Abstract. Traditional supply chain management systems use centralized servers and centralized databases, which exhibit disadvantages such as opaque transaction information, difficult information sharing, high maintenance costs, low database security, and poor anti-risk capabilities. Here we develop a supply chain transaction architecture based on blockchain technology. It stores data on decentralized blockchains, and these distributed data are traceable and antitampering under low maintenance costs. We take the alliance chain as the underlying architecture and adopts the "blockchain + business database" double-layer data storage structure to ensure the safety and convenience of transaction data, which will protect the security of transaction data and ensure the speed of data processing. We conduct simulation experiments and performance tests on the system. The experimental results show that the supply chain system using this architecture achieves high efficiency, transparency and security, as well as it could be utilized in the real applications.

Keywords: Blockchain, Supply Chain, Consortium Blockchain, Hyperledger Fabric, Smart Contract, Chaincode

1 Introduction

The concept of supply chain began to be proposed in the 1980s, with the advent of economic globalization, the flow of information between enterprises, the interaction between capital flow and logistics has become more frequent and complex, the data scale of the supply chain is growing geometrically in the competition of enterprises^[1]. The traditional supply chain model dominated by core enterprises has problems such as poor information sharing, low collaboration efficiency, difficulty in product traceability, lack of trust between enterprises, and excessive voice of core enterprises. At the same time, most of the supply chain management and control system load limit is severely restricted by the performance of the central server. It is very expensive to maintain a high-performance server. Once the centralized server and database are compromised, it will bring serious security threats to the entire supply chain^[2].

The essence of blockchain is a distributed database (ledger), which has the characteristics of decentralization, tamper-proof and traceability. It has been widely used in

digital currency, anti-counterfeiting and traceability, electronic deposit certificates, intellectual property protection, Internet of Things and other fields. The data stored on the blockchain is tamper-resistant, providing a new mode for information sharing between enterprises in the supply chain. Companies on the chain can build a decentralized or multi-center supply chain network to avoid the potential risk of excessive power of the core company.

Based on summarizing the problems of traditional supply chain management systems and analyzing some deficiencies in the existing blockchain-based supply chain management architecture, this paper constructs a supply chain management architecture based on alliance chains. The architecture uses Hyperledger Fabric as the underlying framework to isolate private data through channel isolation. Under the premise of protecting privacy, companies in the supply chain can quickly reach consensus and trusted transactions, and achieve massive transactions that meet actual application scenarios. The "blockchain + business database" distributed storage accounting mode overcomes many of the problems of low transaction speed and low data throughput of classic blockchain technology.

2 Related work

Blockchain was proposed as the underlying technology of Bitcoin^[3]. Taking Bitcoin as an example of the blockchain model, the blockchain is divided into data layer, network layer, consensus layer, incentive layer, contract layer and application layer^[4]. Each block is composed of a block header and a block body. The blocks that are generated one after another are linked together to form a block chain. Smart contract^[5] is a kind of script running on the blockchain, which realizes the automation of contract content and rule execution by computer coding the script language. On the basis of Bitcoin, the addition of Turing's complete smart contract is the technical core of Ethereum^[6].

The blockchain network based on the Proof of Work consensus mechanism has problems such as low transaction speed, low data throughput, and high hardware cost^[7], which will cause energy waste and environmental pollution, and cannot meet the supply chain. The needs of a large number of business scenarios. Pedrosa^[8] et al. designed the lightning network and used it as a side chain of Bitcoin to increase the transaction speed of the Bitcoin network. With the introduction of Byzantine Fault Tolerance into the blockchain, the hardware and energy costs required to achieve consensus have been greatly reduced. However, as the number of nodes increases, the time required for the Byzantine algorithm to reach a consensus has also greatly increased. For this issue, Androulaki^[9] et al. proposed an expandable Hyperledger Fabric alliance chain system, which achieved an end-to-end throughput of more than 3,500 transactions per second, and can be well extended to more than 100 peers.

Today's competition among enterprises has shifted from individual product competition to supply chain competition. In the industry, Wal-Mart uses barcode (UPC) technology and radio frequency data communication (RFDC) technology to build a low-cost and high-efficiency replenishment system^[10], which has greatly increased its

gross profit margin. JD.com completed the electronic transformation of the entire supply chain, and data shows that only in the warehousing handover stage, it saves about 100 million yuan in expenses every year. Real-time dynamics and precise arrangement of logistics are the core of supply chain development.

The traditional supply chain system integrates transaction, transportation and warehousing, spans multiple steps and hundreds of locations, and generates huge amounts of data. This may cause data to be tampered with, illegal transactions cannot be monitored, and default costs are low. The application of blockchain technology to the supply chain system helps to solve the problems of supply chain decentralization, traceability, information security, and maintenance costs. For example, Feng Tian^[11] combined the traceable supply chain with RFID tags and blockchain technology, and classified and stored the RFID tag data generated during production, processing, transportation and other processes in the blockchain, which improved the traceability of the supply chain. Figorilli^[12] et al. implemented a blockchain-based timber chain traceability system through RFID sensor technology and blockchain technology, which use the characteristics of decentralization and distributed storage of blockchain technology to safely store data information and transaction records, and realize the electronic traceability of timber from standing timber to end users. However, this solution does not support the two-way update of the blockchain and the database, and it has poor support for data persistence. Caro^[13] et al. combined the Internet of Things technology to propose a blockchain solution for agricultural product supply chain management, AgriBlockIoT, which realizes the traceability of information in the whole process of production and sales of agricultural products, but this solution has the problem of insufficient computing power at the edge nodes. Resulting in low overall system throughput. This paper adopts the "blockchain + business database" distributed storage accounting mode to control the size of the unit block. Compared with the literature [12], the efficiency of data storage is improved, and the Kafka consensus mechanism is introduced to achieve rapid consensus of trustless nodes. It reduces the computing overhead of nodes, overcomes the problem of insufficient computing power of edge nodes in the literature [13], and improves the availability of low computing power computing nodes.

3 System Design

3.1 Supply Chain Transaction Architecture Design

The supply chain transaction architecture of this article is developed based on Hyperledger Fabric 1.4, as shown in Figure 1. It consists of a data layer, a consensus layer, and an application layer. The main function of the data layer is to store business data such as business registration information and transaction data in the blockchain and the hash value of these data. In the consensus layer, each organization in the alliance has a peer node for information exchange and a node for sorting, and Kafka consensus is used to package and sort transactions. The application layer allows web

services to interact with the blockchain network through the SDK interface to facilitate data query and product traceability.

Consensus Layer	Consensus Algorithm			
	Kafka Consensus		Smart Contract	
Application layer	P2P Network			
	SDK Interface		Web Service	
Data Layer	Blockchain		Database	
	Merkle Tree	Digital Abstract	Transaction Data	Block Index

Fig. 1. Supply chain transaction architecture

3.2 Multi-Tier Database Design

With the increase of network nodes, the speed of consensus on the blockchain becomes slow. In order to accelerate the speed of consensus and increase the throughput of data per unit time of the system, the block size is often limited to a small range. In a supply chain transaction system, a transaction often involves a large amount of data, and it is obviously difficult to store all the data involved in the transaction on the blockchain network. In order to solve this problem and improve the throughput of the network, this paper adopts a two-layer database design of "blockchain + business database". As shown in Figure 2, in the supply chain, the data generated during the procurement of raw and auxiliary materials, production, circulation and use of finished products is submitted to the smart contract for verification by the node. The smart contract authenticates and processes the data and then hashes the data. After that it upload the data summary generated by the hash operation to the blockchain network, and store the original data and the block location information stored in the data summary in the business database. This paper reduces the block size through the design of a multi-layer database, improves the system throughput per unit time, and ensures the reliability of the data through smart contract verification.

The simulation platform in this paper uses CouchDB database to store actual business data. CouchDB is a document-based database system. Data is stored in a key-value pair format to store specific transaction information in the ledger. The CouchDB database supports two-way synchronization. When the data in the blockchain is changed through transactions, the corresponding data in CouchDB will also be updated synchronously. When adding, modifying and deleting data in CouchDB, all operations will be simultaneously appended to the blockchain ledger.

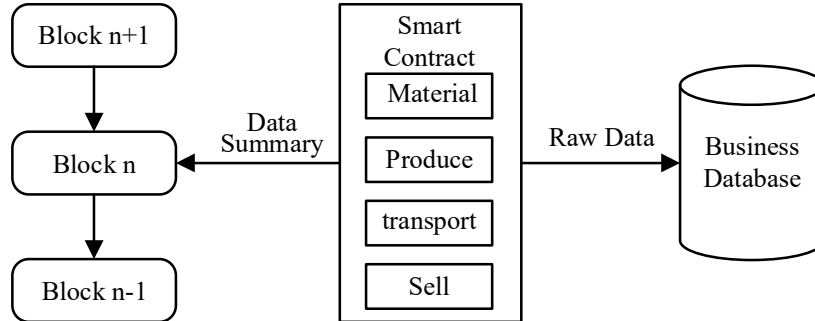


Fig. 2. Blockchain + relational database storage model

The data verification process of this article is as follows, when the data needs to be verified, the original data and the block location information stored in the data summary are retrieved from the database. Hash the data in the database and compare it with the data on the block. If it is consistent with the data on the block, the data has not been tampered with, otherwise, the data has tampered.

3.3 Kafka Consensus Configuration

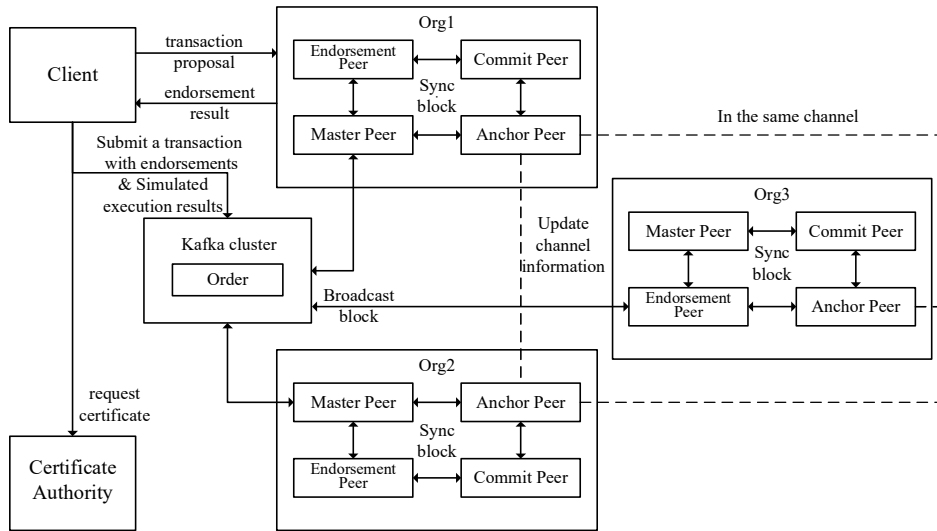
This paper adopts Apache Kafka consensus mechanism. Compared with the traditional consensus mechanism based on workload proof, Kafka consensus speed is faster, and the hardware overhead is small and will not cause energy waste.

Compared with the consensus algorithm based on Byzantine fault tolerance, Kafka consensus still has better system throughput when it accommodates more nodes. Compared with Proof of Work and Proof of Stake, Kafka consensus has sufficient fault tolerance but poor resistance to malicious attacks. Considering that general enterprise databases are deployed in the intranet, and only enterprises with a foundation of trust can join. Therefore, this paper selects the more powerful Kafka consensus mechanism without considering the existence of external attacks.

The simulation platform of this paper adopts the consensus mechanism based on Kafka cluster as shown in Figure 7, and the nodes in the following Table 2 are configured on 16 virtual machines. The hardware platform of the system is as follows: Intel Core i9 10900k 4.7GHz processor, NVIDIA RTX3070 graphics card, DDR4 32G memory, 1T solid state drive. The software platform is as follows: Ubuntu 16.04 operating system, Hyperledger Fabric1.4, Golang 1.8, Docker 2.0. The server configuration information is shown in the table below. A total of 16 nodes run on 16 virtual machines. It includes three organizations: Organization 1/2/3. Each organization has two nodes. The first node of each organization is the anchor node that joins the public channel and interacts with other organizations. In addition, 3 Zookeeper clusters, 4 Kafka clusters, and 3 Orderer nodes for sorting are included.

Table 1. Kafka cluster configuration

Name	IP Address	Quantity	Organization
Zookeeper	192.168.247.1/2/3	3	/
Kafka	192.168.247.4/5/6/7	4	/
Orderer	192.168.247.8/9/10	3	/
Org1	192.168.247.11/12	2	Org1
Org2	192.168.247.13/14	2	Org2
Org3	192.168.247.13/14	2	Org3

**Fig. 3.** Kafka consensus architecture

3.4 Smart Contract Design

The smart contract in this simulation experiment is written in Golang language and uses the shim API interface provided by the Hyperledger Fabric team. The data structure of the smart contract defines a user information table and a product information table. The user information table is shown in Table 3, which is used to maintain the user's personal information and a list of products owned by the user. The product information table is shown in Table 4, which is used to maintain various information of the product. When the product is made of chain materials, the product information table also maintains the raw material table of the product. Table 5 is the function description in the following smart contract algorithm 1-3.

Table 2. User Information Form

Serial number	Field name	Description
1	userName	User's name
2	userId	User Unique Id number
3	productArr	productArrUser's product list

Table 3. Product Information Sheet

Serial number	Field name	Description
1	productName	Product name
2	productId	Product unique Id number
3	madeInChian	Whether the product is made from on-chain materials
4	productInfo	Product information such as production date, quantity, etc.
5	rMaterialArr	productArrUser's product list

Table 4. Function Description in the Algorithm

Serial number	Field name	Description
1	isExist	Check if the input exists
2	getState	Get the current value of a state variable
3	putState	Update the specified state variable
4	delateState	Delete the specified state variable
5	madeInChain	Determine whether it is made from raw materials on the chain

In this paper, three contracts are designed, among which algorithms 1 and 2 are the registration and transaction algorithms of smart contracts, and algorithm 3 is used to verify the reliability of data on the supply chain. Algorithm 1 is a smart contract algorithm used by companies in the blockchain for product registration. Before product registration, the user needs to define the structure of the product and store it in a json file in string format. After the definition is completed, the smart contract will determine the format of the written data. After the determination is passed, the product will be registered under the corresponding owner's name and written into the blockchain network.

```
Algorithm_1 Product_Grade
Input productId, productInfo, ownerId
Output return Shim.Success else Except
Begin:
1.type Product struct
2.if input is empty or error struct
```

8

```
3.    return input error
4.else
5.    if isExist(productId) is true
6.        return productId is exist
7.    else
8.        if isExist(ownerId) is false
9.            return user not exist
10.       else
11.           ownerId.product.PutSate(productId)
12.           productId.putState(productId)
13.           return put product success
End
```

Algorithm 2 introduces product transaction between enterprises. During the transaction, the smart contract will determine the input data. When all the determination conditions are passed, the smart contract will change the asset owner and put the transaction on the chain.

```
Algorithm_2 Product_Trading
Input preownerId, productId, ownerId
Output return Shim.Success else Except
Begin:
1.if isExist(preownerId or productId or ownerId)
is false
2. return input error
3.else
4.  getState(preownerId)
4.  for i in perownerId.productArr
5.      if perownerId.product[i] == productId
6.          preownerId.deleteState(productId)
7.          ownerId.putState(productId)
8.      else
9.          return preownerId not have this productId
End
```

Algorithm 3 is an algorithm called when companies consume some raw materials or materials recorded on the chain to manufacture products. It is mainly used to determine whether there are illegal operations such as raw materials that have passed the shelf life and ingredients measurement inconsistent during the production process.

```
Algorithm_3 Production_Monitoring
Input productId, productId, ownerId
Output Return Shim.Success else Except
Begin:
1.type Product struct
```

```

2.if isExist(productId) is true
3.  return productId is exist
4.else
5.  if madeInChain(productInfo) is true
6.      get rMaterialArr from productInfo
7.      for rMaterial in rmArr.num
8.          if rMaterial.Exp<Now.date
9.              return rMaterial out of date
10. else
11.     ownerId.deleteState(product. rMaterial)
12.     ownerId.putState(productId)
13.     productId.putState(productInfo)
End

```

4 System Performance Analysis

4.1 Parameter Introduction

After completing the construction of the supply chain transaction system, the supply chain transaction system needs to be tested. This article uses the load testing tool Tape to evaluate the throughput of the system by analyzing the response to the gRPC request. The test mainly calls the query function and the call function in the smart contract, and measures the system performance by testing the system transaction throughput, transactions per second, TPS, under different parameter settings. In the process of blockchain transactions, transaction data throughput, transaction success rate, and transaction delay are the key factors for measuring system performance.

In the process of transaction packaging, the ordering node needs to digitally sign each transaction. Digital signatures for a large number of transactions in a unit time occupy a large amount of system performance. Therefore, this article introduces the concept of batch signature, which is to package a group of data into one bag. The ordering node only needs to sign each packet once, thereby achieving the goal of saving system performance and improving data throughput per unit time. Also, to prevent the number of transactions in the blockchain within a period of time from being too small to meet the packaging requirements, resulting in too long a single package generation time, this article sets a maximum block time for each package. The system performance test in this article will provide the system throughput under the query operation and call operation of the system under the adjustment of packet size, maximum block time, and block size.

4.2 System Performance Analysis Under Query Operation

When performing a query operation on the blockchain, there is no need to call an additional ordering node of the system, and no record will be left on the blockchain.

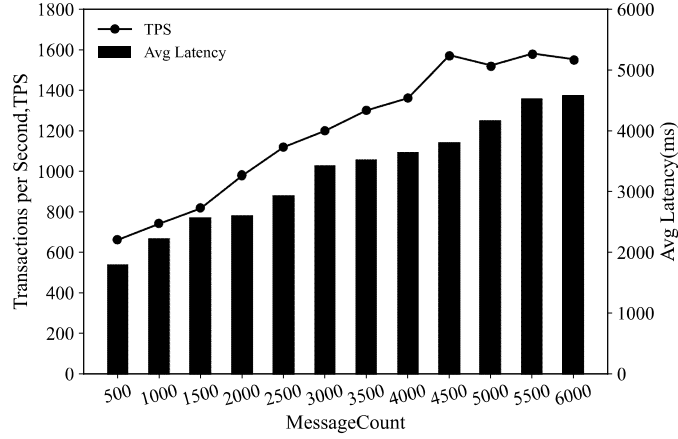


Fig. 4. System Throughput (TPS) under Query Operation

Figure 4 shows the change in system throughput with the increase of the request volume under the query operation. When the number of requests is 500, the throughput of the system is 603TPS, when the number of requests is 1000, the throughput of the system reaches 1080TPS, and when the number of requests is more than 2000, the throughput of the system reaches more than 1600TPS and tends to be stable. This shows that the blockchain-based supply chain platform proposed in this article can basically meet the needs of most small and medium-sized enterprises for supply chain management when deployed on ordinary servers.

4.3 System Performance Analysis Under Call Operation

The call operation needs to call the ordering node to order the transaction, and the endorsing node is required to endorse the transaction. Among them, the batch size, maximum block time, block size and other parameters will affect the performance of the system under the call operation. This article chooses to adjust other parameters to test the system performance based on controlling the maximum block time. Table 5 records the size of the system throughput under different parameter settings.

Table 5. System throughput under different parameter settings

BatchSize	Timeout	BlockSize	TPS
8	2s	8m	57
32	2s	32m	236
128	2s	32m	501
512	2s	64m	1458
1024	2s	128m	1539

Figure 5 shows the effect of batch size and block size on system throughput when the maximum block time is fixed.

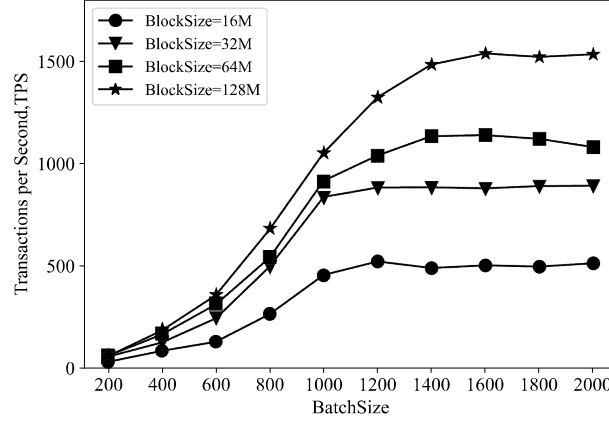


Fig. 5. System throughput under call operation

As the unit block size and batch size increase, the throughput of the network also increases. When the block size is set to 32M, as the batch size increases, the system throughput gradually rises. When the batch size reaches 2048, the system throughput reaches 880TPS. When the batch size is 1024 and the block size reaches 128M, the system throughput reaches 1500TPS. At this time, increasing the block and batch size cannot improve the system performance. Through the analysis of log files, it is found that the number of transactions per unit time processed by the sorting node has not reached the upper limit, that is, a bottleneck has appeared in the I/O performance of the computer. Limited by the computer's hardware, there is still room for improvement in the performance of this system. The final data throughput under the current hardware platform is 1500TPS, which can basically meet the transaction processing needs of medium-sized enterprises.

4.4 Analysis of the Influence of the Number of Nodes on the System Performance

As the number of nodes on the blockchain network increases, where a large number of nodes make data calls simultaneously, it will also lead to an increase in system resource overhead, furthermore affecting system latency and system throughput. We also need to consider how to perform concurrent testing among multiple nodes and how to select the appropriate number of peak nodes for the system.

We simulate the impact of the number of nodes on the system throughput when the batch size is set to 2048, the block size is set to 128M, and the call operation is performed in the public channel. Table 7 shows the number of nodes in the experiment and the system delay time and system throughput of 50,000 transactions completed by the call operation under the current settings. It can be seen from Figure 6 that with the increase in the number of nodes, the overhead of the system gradually increases, the throughput of the system does not increase indefinitely, and the system delay continues to rise. When the number of nodes exceeds 60, the performance of the system

begins to drop significantly. When there are 100 nodes in the call operation at the same time, the system throughput can still reach 1326TPS. Through experiments, it can be found that the system performance can still meet the basic application requirements when multiple nodes are called at the same time.

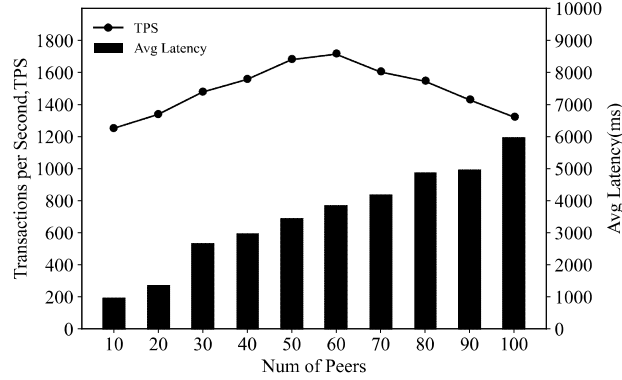


Fig. 6. The impact of the number of users on throughput

4.5 Comparison with existing systems

The scheme in this paper is a supply chain management scheme based on blockchain, which adopts Kafka consensus mechanism to reach the consensus of network nodes. Compared with the traditional supply chain management system based on centralized service, this scheme has certain advantages in data tamper proof, independent of central server and data traceability. Compared with the reference [17], which takes azure SQL blockchain platform to build supply chain management system, our scheme uses CouchDB database to store business data and supports bidirectional modification of data modification of data, which has certain advantages in data storage. Compared with the Ethereum platform and sawtooth platform adopted in document [18], the supply chain management system based on Ethereum platform has the problem of insufficient computing power of edge nodes. The Kafka consensus used in our scheme has lower requirements for node computing power, better compatibility with low configuration computers, and higher efficiency of consensus. Table 6 shows the comparison between our scheme and the above two schemes in decentralization, computing overhead, tamper proof, fault tolerance and data storage.

Table 6. Comparison with existing systems

Function	This paper	Literature [12]	Literature [13]
Decentralization	√	√	√
Tamper proof	√	√	√
Support fault tolerance	√	√	√
Low computational cost	√	×	√
Bidirectional data storage	√	×	×

5 Conclusions

With the advent of the era of blockchain 3.0, the concept of blockchain + business database has received extensive attention from all walks of life in society, and the blockchain network has been applied to various fields as a trust network. This paper designs a supply chain transaction processing architecture based on blockchain technology, and uses experiments to demonstrate the feasibility and efficiency of this architecture. This article uses the Hyperledger Fabric framework to design, realizes the transaction processing in the supply chain, uses the channel isolation method to realize the protection of private data in the transaction process, and guarantees the transaction processing speed and data storage efficiency through the multi-layer database design. The simulation experiment test results show that the final throughput of the blockchain architecture in this paper has reached 1500TPS, which proves that the solution can meet the needs of small and medium-sized enterprises' supply chain management.

This solution still has some problems to be solved later, such as how to achieve rapid product traceability on the chain under the premise of meeting privacy data protection is still difficult, and how to further improve the efficiency of the consensus algorithm. These issues are worthy of further study.

References

1. Jiang Wuxue. An intelligent supply chain information collaboration model based on Internet of Things and big data[J]. *IEEE Access*, 2019, 7:58324-58335
2. Kim H M, Laskowski M. Toward an ontology-driven blockchain design for supply-chain provenance[J]. *Intelligent Systems in Accounting, Finance and Management*, 2018, 25(1): 18-27.
3. Wang R J, Tang Y C, Zhang W Q, et al. Privacy protection scheme for Internet of vehicles based on homomorphic encryption and block chain technology[J]. *Chinese Journal of Network and Information Security*, 2020, 6(1):46-53.
4. Huang C Y, Wang Z Y, Chen H X, et al. RepChain: a reputation-based secure, fast, and high incentive blockchain system via sharing[J]. *IEEE Internet of Things Journal*, 2021, 8(6): 4291-4304.
5. [1] Hassan E R, Tahoun M, Eltaweel G S. A robust computational DRM framework for protecting multimedia contents using AES and ECC[J]. *Alexandria Engineering Journal*, 2020, 59(3): 1275-1286.
6. Luu L, Chu D H, Olickel H, et al. Making Smart Contracts Smarter[C]// *ACM SIGSAC Conference on Computer and Communications Security*, 2016: 254-269.
7. Leonhard R D. Developing Renewable Energy Credits as Cryptocurrency on Ethereum's Blockchain[J]. *Social science electronic publishing*, 2016, 14(12): 1-15.
8. Pedrosa A R, Potop-Butucaru M, Tucci-Piergiovanni S. Scalable lightning factories for Bitcoin[C]// *ACM/SIGAPP Symposium on Applied Computing*, 2019: 302-309.
9. Androulaki E, Manevich Y, Muralidharan S, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains[C]// *the 13th EuroSys Conference*. 2018: 1-15.
10. Ali S, Wang G, White B, et al. Libra critique towards global decentralized financial system[C]// *Communications in Computer and Information Science*. Springer, 2019: 661-672.

11. Tian F. An agri-food supply chain traceability system for China based on RFID & blockchain technology[C]// 2016 13th international conference on service systems and service management (ICSSSM). IEEE, 2016: 1-6.
12. Caro M P, Ali M S, Vecchio M, et al. Blockchain-based traceability in agri-food supply chain management: a practical implementation[C]// 2018 IoT Vertical and Topical Summit on Agriculture, 2018: 1-4.
13. Figorilli S, Antonucci F, Costa C, et al. A blockchain implementation prototype for the electronic open source traceability of wood along the whole supply chain[J]. Sensors, 2018, 18(9): 3133.