



# Hyper-FTT: A Food Supply-Chain Trading and Traceability System Based on Hyperledger Fabric

Kui Gao, Yang Liu<sup>(✉)</sup>, Heyang Xu, and Tingting Han

College of Information Science and Engineering, Henan University of Technology,  
Zhengzhou 450000, China  
liu\_yang@haut.edu.cn

**Abstract.** Building a food traceability system provides an effective way solving the problems arising in the food safety domain. Blockchain is preferred to construct a commodity traceability system due to its innate immutability and consistency of stored data, maintained through cryptographic means and consensus mechanism. While current blockchain based traceability system just collects and stores verified commodity information in the chain, which is composed of a decentralized ledger architecture. But the information is less accurate or intact which cannot trace the real food source along the food supply-chain. We propose to establish a Hyperledger-based Food Trading and Traceability system called Hyper-FTT, by aggregating all the providers including food warehousing enterprises, food processing enterprises and food retails to reach agreements and conclude business transactions on the chain, then an unbroken food supply-chain can be formed to provide trusted food tracing. Implementation and experiments are conducted to evaluate the performance of the proposed demonstration system.

**Keywords:** Food traceability · Food trading · Food supply-chain · Blockchain · Hyperledger fabric

## 1 Introduction

With the improvement of living standards, people pay more and more attention to food safety and quality. However, food safety incidents frequently happen in recent years. For example, Europe explored “horsemeat scandal” in 2013 [1]. In China, reports of “trench oil” happened from time to time [2]. These food safety issues not only endanger human health but also bring food panic. The food supply chain is an integrated functional network structure model, which is including food production, processing, warehousing, transportation, and retail [3]. All suppliers involved in food supply chain are autonomous with their own profit demands, and there is a game relation of them. At the same time, there is also a game relation between suppliers and regulatory authorities in the food supply chain. How to accurate find the source of the food crises and the perpetrators belongs to food traceability, which is a complex and challenging problem [4].

Traditional technologies, such as database, cloud computing and big data, can achieve information sharing and collaboration in game relation of subjects. However, the technologies will cause many other problems. For example, it is difficult to transport data between multiple centralized data centers. Blockchain technology is a transparent and verifiable system, which is seen as a new way to provide trusted service [5]. Some popular blockchain platform, such as Hyperledger Fabric [6] and Ethereum [7], are widely used in. When a transaction occurs between two participants, the network will automatically execute the invoked smart contract (SC) to complete the transaction [8]. Recently, blockchain technology is frequently adopted to resolve the problem of commodity traceability. For example, Watanabe et al. [9] developed a decentralized rights management system, in which to management the copyrights of videos. Tian et al. [10] proposed an agri-food supply chain traceability system by blockchain and RFID technologies. The proposed system can gather all the information in agri-food circulation and record it in blockchain to guarantee food safety. Toyoda et al. [11] developed a product ownership management system to anti-counterfeits in the supply chain. Caro et al. [12] proposed an agricultural food supply chain traceability management approach and implemented it under two blockchain platforms, i.e. Ethereum and Hyperledger Sawtooth. Lin et al. [4] constructed a food safety traceability system based on blockchain and EPC information services (EPCIS), and proposed two data management architectures with on-chain and off-chain ways to solve the issue of data explosion caused by the application of the blockchain for Internet of Things.

Food traceability is different from other traceability sources, not only in the aspect of food circulation, but also in the changes of food's properties and states. Although many researchers have explored food traceability, using blockchain technology to solve this issue, which is a new field and still exists some following challenges. First, most traceability systems only accentuate the information traceability instead of guarding against the information falsification in food circulation and processing, and so the information authenticity is difficulty to guarantee [13]. Second, existing traceability systems ignore the supply chain manufacturers or just intercept a small supply chain information [11], which are incomplete and difficulty to trace the source. Third, existing systems are lack of transparency and often produce some information islands. The enterprise resource planning (ERP) systems between enterprises in the same supply chain are not interoperable, and the enterprise autonomous ERP system cannot be connected to the traceability system.

In this paper, we propose a hyperledger-based food supply-chain trading and traceability system called Hyper-FTT. The system can aggregate all the food warehousing enterprises, food processing enterprises and food retails to trade on the blockchain. The supply chain is complete, food circulation and processing can be regulated, in which all information is traceable. At the same time, Hyper-FTT provides an interface for the autonomous ERP of each trader to interface with the blockchain and thus can break the information island. There the proposed system can form a "complete chain, complete cycle, and complete coverage" of the food supply chain to achieve accurate and reliable traceability. Finally, we implement Hyper-FTT on the base of Hyperledger Fabric v1.1 and the system performance and throughput are evaluated under different operating modes and system configurations.

The rest of this paper is organized as follows. Section 2 explains the scenario of Hyper-FTT. Section 3 describes the Hyper-FTT system architecture. The block data structure and smart contract are given in Sect. 4. The implementation and experimental analysis are presented in Sect. 5. Finally, we conclude this paper in Sect. 6.

## 2 Hyper-FTT Application Scenario

The Hyper-FTT application scenario is shown in Fig. 1, in which traders represent all suppliers and manufacturers in the food supply-chain, administration of traders and administration of quality to represent the administrations.

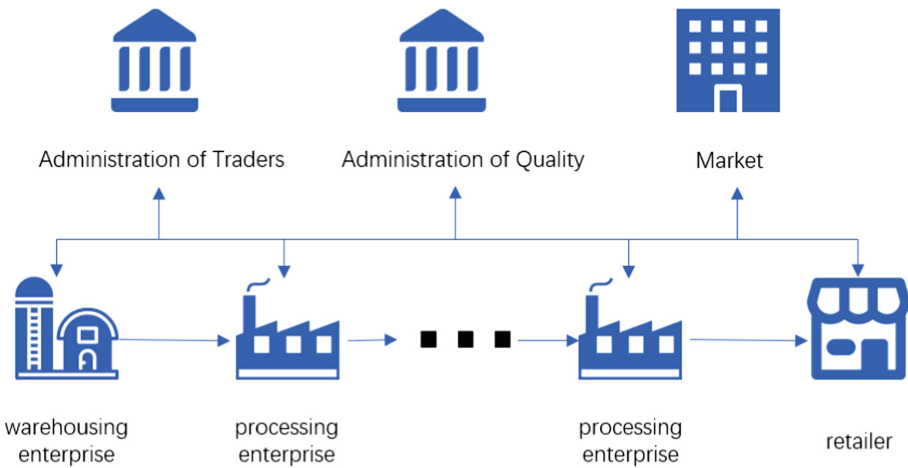


Fig. 1. Hyper-FTT application scenario.

**Traders:** All suppliers and manufacturers (food warehousing enterprises, food processing enterprises and food retails, etc.) of food supply chain. Traders must submit qualification certificates to administration of traders. Only those who pass the qualification examination can participate in the food supply chain. They are autonomous entities and can make deals on the blockchain.

**Administration of Traders:** Who is responsible for examining and verifying the qualification certificates submitted by traders and then issuing a certificate ID to the approved trader. The administration of traders also undertakes the verification of the assets in the real world before uploading the information onto the blockchain network. It means that you can transform real-world assets into blockchain assets at this organization and vice versa. The role of this part can be undertaken by the bureau of industry and commerce.

**Administration of Quality:** This organization examining and verifying quality of all kinds of foods and storing the approved food information on the blockchain ledger. The stored information includes raw information of food, processing and circulations

of food information. It can establish the relationship of food information associations in the supply chain. The role of this part of the system can be undertaken by the quality and technology supervision bureau.

**Market:** This organization collecting food information owned by each food manufacturer and providing a decentralized trading platform. A trader can view all foods information on blockchain ledger and select the food that they want to trade. We create a secondary index of food information in the market, which can help to search and analyze food information.

The processes for a trader to join Hyper-FTT and start a transaction are as follows.

Step 1: The trader submits a qualification certificate to administration of traders. If approved, then it will get a certificate ID; otherwise, it will be rejected.

Step 2: Administration of quality verifies the food information submitted by traders and records it in the blockchain ledger.

Step 3: The admitted traders can view the food information and negotiate the transaction with other traders in the market. And then, a transaction can be concluded.

### 3 Hyper-FTT System Architecture

The system architecture of Hyper-FTT is illustrated in Fig. 2, which can be divided into two parts: one is the configuration of blockchain network, the other is the client application.

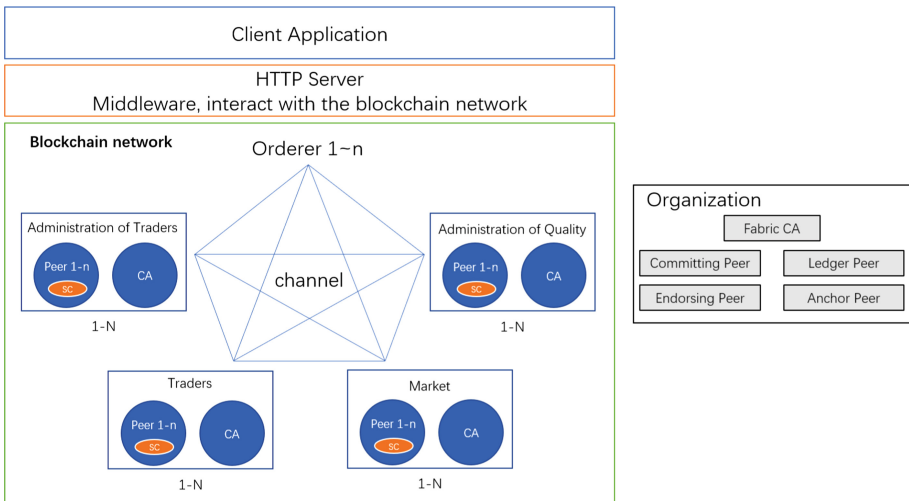


Fig. 2. Hyper-FTT system architecture.

### 3.1 Blockchain Network

The blockchain network of Hyper-FTT contains following features, namely distributed and chained storage of transaction records, fine-grained access control based on identity authentication, hot-pluggable consensus protocol in Orderer cluster and rich smart contracts to support blockchain network business logic. Besides, the system uses CouchDB database, which supports rich query instructions, to store blockchain ledger data. The functions of each component are described as follows.

**Orderer:** This component is responsible to receive all the transaction requests in the blockchain network on the specified channel, sort these transaction requests by consensus protocol, then package the verified transaction requests into blocks and finally broadcast them to the blockchain network through gossip communication.

**Organization:** Hyperledger fabric is managed by all organizations. Administration of traders, administration of quality, traders and market are embedded in the blockchain network as organizations. In the hyperledger fabric, an organization can have a local fabric CA as well as a set of nodes, which have multiple roles, such as committing peer, endorsing peer, ledger peer and anchor peer. Local fabric CA can manage certificates of all nodes in an organization. Only the node that the administrator has authenticated through the fabric CA can install the smart contract. A node can install multiple smart contracts and the operations completed by calling SC will be irreversible stored in the blockchain ledger.

**Channel:** A channel provides a dedicated bridge for a set of specific organizations. In Hyper-FTT, the channel maintains communication between administration of traders, administration of quality, traders and market.

**Smart Contract:** A smart contract is a program that implements a specified interface. The state of the blockchain ledger can be irreversible updated by calling a smart contract. In this paper, we propose a set of smart contracts to achieve the Hyper-FTT business logic. After the smart contract is instantiated in the channel, all nodes that installed the smart contract can interact with blockchain ledger through the blockchain network by calling the interface defined in the smart contract.

### 3.2 Client Application and HTTP Server

Hyper-FTT provides applications for administration of traders, administration of quality, traders and market to facilitate them interaction with the blockchain network. HTTP server acts as a middleware in the system, providing RESTful APIs to interact with the blockchain network. In addition to receiving and processing requests from the client application, the HTTP server must directly interact with the blockchain network to complete specific business logic by invoking a specific smart contract. In this way, the HTTP server can decouple the client application from the blockchain network.



which are untrading, trading and traded. When a raw food is stored in the blockchain ledger, the value of Previous\_Key is null and the Transaction\_State is set as untrading. When the food is trade between traders, the Transaction\_State is changed to trading. When a trade is finished, the food Transaction\_State is set to traded. Then the new owner anew records the food with Previous\_Key and sets the new food Transaction\_State as untrading. Figure 3 shows the changing food information of food processing. Figure 4 shows the information changing in food transaction.

When a food transaction occurs, both sides of the transaction need to submit the necessary information to trigger the execution of trading smart contract. The data structure for stored transaction information is shown in Table 3. When a transaction is initiated, the State value is purchasing and when the transaction is finished, the State value becomes purchased.

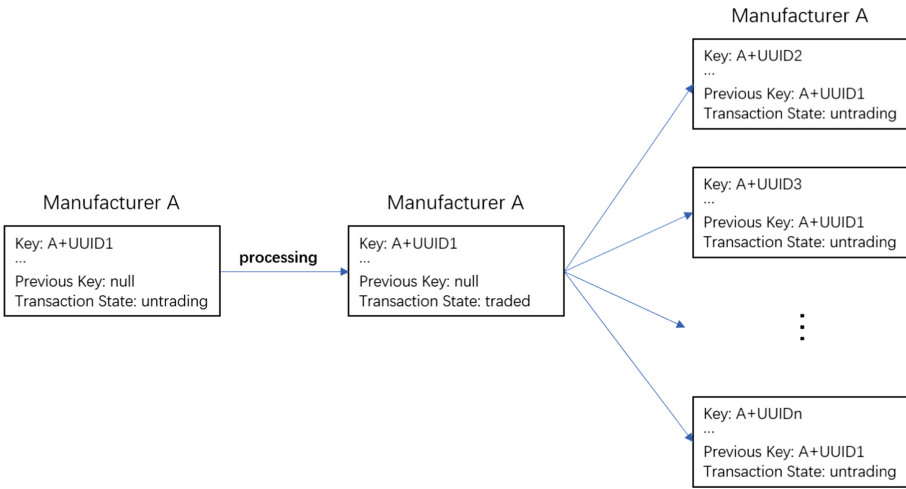


Fig. 3. Food processing and renew record in blockchain ledger at same manufacturer.

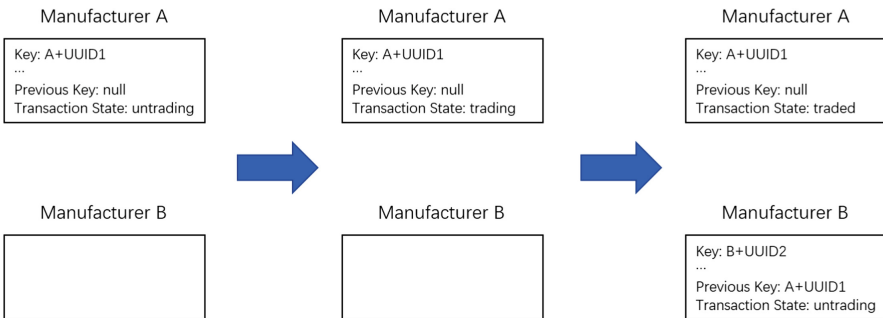


Fig. 4. Manufacturer A trades with Manufacturer B.

**Table 3.** The data structure of transaction.

Prefix	Transaction
Key	Composite(Seller,Buyer,TxId)
Value	<pre> {   Seller: Seller.ID,   Buyer: Buyer.ID,   UnitPrice: Unit price of food,   Foods: The key value of the foods to be traded,   Transacted_Foods: Successfully transacted food,   Asset: Total assets required for this transaction,   Transacted_Asset: Actual realized trading assets,   State: purchasing/purchased } </pre>

## 4.2 Smart Contract Design

In this paper, we design a variety of smart contracts to implement the business logic of Hyper-FTT. According to their functions, they can be divided into three categories.

### A. Storage Smart Contract

The storage smart contract can be divided into two categories. One is used to store raw information (such as traders information and raw food information), which can be stored directly after being authenticated. The other is used to store processing food information, which requires that its raw food information should first pass the examination of administration of quality. The process of the later is described in Algorithm 1. First, read raw food information through RFK (line 1 Algorithm 1), if raw food exists in blockchain ledger, then update raw food Transaction\_State to traded (line 2–5, Algorithm 1). Base data structure of food, RFK, pname and nCharacters set new foods information (line 6–8, Algorithm 1), then according to processing foods number create new foods and update blockchain ledger (line 9–14, Algorithm 1).

---

#### Algorithm 1 Restorage processing food

---

##### Inputs:

Key values of raw food (RFK), name of the processing food (pname), characters of the processing foods (nCharacters), number of the processing foods (number).

##### Outputs:

```

1  rf = GetState(RFK);
2  if rf != null
3    nf = rf;
4    rf.Transaction_State = traded;
5    PutState(RFK,rf);
6    nf.Transaction_State = untrading;
7    nf.Previous_Key = RFK;
8    update nf's Food_Name and Characters use pname and nCharacters;
9    while number != 0
10     uuid = CreateUUID();
11     nf.Id = uuid;
12     PutState(Composite(nf.Owner, uuid), nf);
13     number--;
14   end while
15   endif

```

---



## B. Trading Smart Contract (TSC)

A transaction mainly involves two entities, i.e., a seller and a buyer. In order to make a transaction, the buyer first needs to get all food information from the market, selects the needed food and submits the required quantity and the acceptable price. After the seller receiving the buyer's information, they will negotiate the final quantity. Finally, involved foods will be transported from seller to buyer in the real world: the buyer purchases the involved foods, confirms their information and returns the other foods that not pass qualification examination. The execution of TSC contains two steps. First, both parties initiate a transaction. Second, involved foods are transported from the seller to the buyer in the real world. Then a transaction is completed.

In the first step of TSC buyers and sellers need to submit identity information, such as the keys of the foods to be trade, which can be described in Algorithm 2. Initially, the transaction state of all the traded foods will be set as trading (line 1–5, Algorithm 2), temporarily deduct all costs of the transaction from buyer's assets (line 6–7, Algorithm 2). Finally, base data structure of transaction, seller identity, buyer identity, etc., create transaction information and update blockchain ledger (line 8–12, Algorithm 2).

---

### Algorithm 2 Trading smart contract part 1

---

**Inputs:**

Seller.ID (STID), Buyer.ID (BTID), key of the foods to be trade (FoodsKey), unit price of food (UnitPrice), number of transaction foods (FoodsQuantity).

**Outputs:**

```

1   for FK in FoodsKey
2     fk = GetState(FK);
3     fk.Transaction_State = trading;
4     PutState(FK, fk);
5   endfor
6   buyer = GetState(BTID);
7   buyer.Asset -= UnitPrice*FoodsQuantity;
8   nt = new Transaction;
9   update nt's Seller, Buyer, UnitPrice, Foods and Asset use STID, BTID, UnitPrice,
    FoodsKey, UnitPrice*FoodsQuantity;
10  nt.State = purchasing;
11  PutState(BTID, buyer);
12  PutState(Composite(STID,BTID,TxId, nt);
```

---

When the involved foods are transported from the seller to the buyer in the real world, it will trigger the second step of the TSC. The process of this step is described in Algorithm 3. In this step, the buyer needs to submit the transaction key and the verified foods keys to TSC. First, read transaction information, seller information and buyer information through the key value of transaction (line 1–3, Algorithm 3). Then the verified foods will change the owner and other foods will return back to the seller with their corresponding assets back to the buyer (line 4–19, Algorithm 3). Finally, update the blockchain ledger (line 20–23, Algorithm 3).

---

**Algorithm 4** Trace part

---

**Inputs:**

The food key (TFK)

**Outputs:**

```

1  tf = GetState(TFY);
2  push(TraceInformation, tf);
3  while tf.Previous_Key != null
4      TFK = tf.Previous_Key;
5      tf = GetState(TFY);
6      push(TraceInformation, tf);
7  endwhile
8  return TraceInformation;
```

---

**C. Traceable Smart Contract**

Hyper-FTT supports a verify of traditional query methods, including block number query, certificate ID query, HASH query, etc. In addition, it also provides a function of convenient traceability searching. A transaction can be quickly queried based on the designed transaction structure. Via the Previous\_Key in the date structure of food, we can trace the source of food at any point in the food supply chain. The trace process of food source is shown in Algorithm 4.

---

**Algorithm 3** Trading smart contract part 2

---

**Inputs:**

Key value of transaction (TS), key of verified foods (VerFoods), quantity of verified foods (VerQuantity).

**Outputs:**

```

1  ts = GetState(TS);
2  BTID = ts.Buyer; btid = GetState(BTID);
3  STID = ts.Seller; stid = GetState(STID);
4  for TF in ts.Foods
5      tf = GetState(TF);
6      if TF in VerFoods
7          tf.Transaction_ID = traded;
8          nf = tf;
9          nf.Owner = BTID;
10         nf.Id = CreateUUID();
11         nf.Transaction_State = untrading;
12         PutState(Composite(nf.Owner, nf.Id), nf);
13         stid.Asset += ts.UnitPrice;
14     else
15         tf.Transaction_State = trading;
16         btid.Asset += ts.UnitPrice;
17     endif
18     PutState(TF, tf);
19 endfor
20 PutState(BTID, btid);
21 PutState(STID, stid);
22 update ts's Transacted_Foods, Transacted_Asset use VerFoods, VerQuantity*ts.UnitPrice and set the State is purchased;
23 PutState(Ts, ts);
```

---

## 5 Experimental Configurations and Result Analyses

### 5.1 System Configuration

System throughput is generally used to evaluate the performance of all kinds of blockchain systems [14–16]. In order to achieve accurate results, all submitted food information used in the experiment is authentic. In the experiment, the system has two traders, each trader contains one peer. The designed smart contract is installed on the peer to complete the business logic in the food supply chain. At the same time, this system uses the CouchDB database to store transaction data. The Orderer peer uses solo consensus protocol. All nodes (peer, order, fabric ca) run as containers. A series of experiments are performed on CentOS 7 with core i3-7100 processors and 4G RAM. The throughput of Hyper-FTT is evaluated by requests per second (RPS), the rate at which requests are completely processed. We use the locust [17], a load-based open source test framework for evaluating system throughput under increasing request arrival rates. We conduct two experiments to test the system transaction read and write.

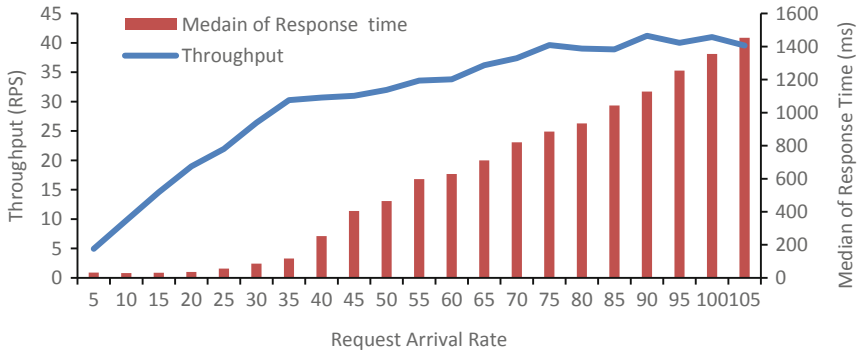
### 5.2 Result Analyses

#### A. Reading Test

The system throughput is related to the real-time performance of the machine, so the system throughput will fluctuate when testing on different times. In this paper, we take the mean value of the results of three independent experiments to smooth the system throughput fluctuation caused by the machine performance. Figure 5 shows the results of the Hyper-FTT reading test. The figure shows the relationship between the throughput of the query operation and the median of response time. In the case of a low request arrival rate, the throughput of the system increases linearly with the increase of request arrival rate. After the request arrival rate reaches 35, as the request arrival rate increases, the throughput of the system grows slowly and the median of response time increases rapidly. When the request rate reaches 90, the throughput of the system decreases with the number of requests increase. This shows that when the request rate reaches a certain level, the performance of the server becomes the bottleneck of system throughput growth.

#### B. Write Test

The blockchain ledger records the transaction data in the form of block, so the storage of the transaction data is related to the configurations of blocks. In Hyper-FTT blockchain network, each block includes three parameters, which are batch time, max message count and block size. Once the transactions in blockchain network satisfies any of these three conditions, then the blockchain network will produce a new block. The write test can be divided into two parts, the part I is to test the impact of change max message count on system throughput and the part II is change block size. The write test block configurations are shown in Table 4.



**Fig. 5.** Hyper-FTT reading test.

**Table 4.** Hyper-FTT write test block configuration.

Part	Consensus protocol	Batch timeout(s)	Max message count	Block size (KB)
I	Solo	2	1, 2, 5, 10	512
II	Solo	2	30	8, 16, 32

The result of the test part I is shown in Fig. 6. It can be seen that when the block size is constant, the throughput of write operation increases with the increasing of the number of recorded transactions. When the request arrival rate is too small, it cannot meet the max message count or block size of the block configuration when it reaches batch timeout, the throughput of the system is determined by batch time. when the request arrival rate is small, smaller configurations (max message count, block size) make it easier to publish new blocks. While with the increasing of request arrival rate, the throughput of the system is determined by max message count. Also, the throughput of the system increases with the max message count increasing, but when the max message count reaches 5, the throughput of the system reaches a limit of 13 RPS. When the number of max message count is 1, 2, 5, as the request arrival rate increase, a write failure may occur during the writing process. Therefore, we can conclude that, before the optimal max message count is reached, the throughput of the system increases with the increasing of the max message count and then the performance of the Orderer becomes the bottleneck of system throughput growth.

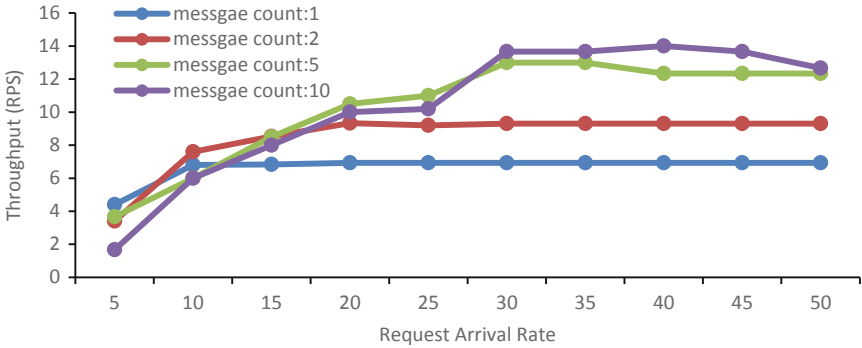


Fig. 6. Hyper-FTT write test with messages variation.

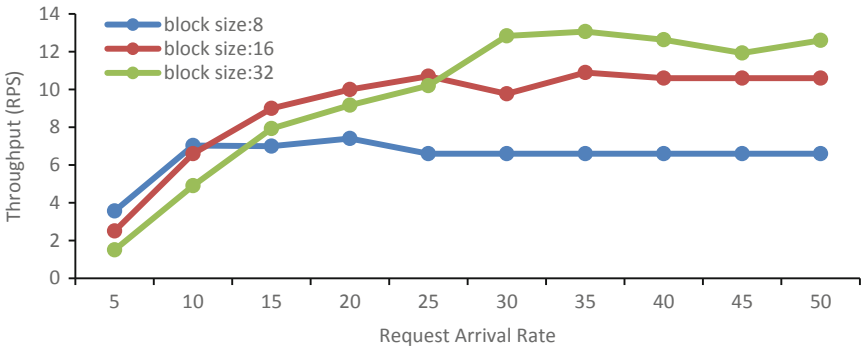


Fig. 7. Hyper-FTT write test with block size variation

Figure 7 shows the results of the throughputs under different block sizes. It can be seen that the system throughput under changing block sizes has the same trend with the changing max message counts. When the block size reaches 16 k, the throughput of the system becoming steady.

## 6 Conclusion

In this paper, we propose a food trading and traceability system that can provide services for government, enterprises and consumers. The system can support trading among autonomous enterprises to build into the Market which can associate the relationship of food variation throughout the food circulation. So the system can form the “complete chain, complete cycle, and complete coverage” of the food supply chain to obtain accurate and reliable food traceability. Extensive experiments have been conducted to evaluate the system performance. The experimental results show that Orderer service has a great influence on the system throughput under different configurations. In the future work, the index in the market will be optimized to support more efficient query and search for food products in the consumer perspective.

**Acknowledgments.** This paper is partially supported by the National Natural and Science Foundation of China (61702162), Program for Innovative Research Team (in Science and Technology) in University of Henan Province (17IRTSTHN011), Plan for Scientific Innovation Talent of Henan University of Technology (2018RCJH07), Natural Science Project of the Education Department of Henan Province (No. 19A520021), Plan for Nature Science Fundamental Research of Henan University of Technology (No. 2018QNJH26) and the Research Foundation for Advanced Talents of Henan University of Technology (2017BS016).

## References

1. Boyaci, I.H., Temiz, H.T., et al.: A novel method for discrimination of beef and horsemeat using Raman spectroscopy. *Food Chem.* **148**, 37–41 (2014)
2. Jing, X., Ziyu, L., Beiwei, L.: Research on a food supply chain traceability management system based on RFID. *Agric. Mech. Res.* **34**(2), 181–184 (2012)
3. Lin, Q., Wang, H., et al.: Food safety traceability system based on blockchain and EPCIS. *IEEE Access* **7**, 20698–20707 (2019)
4. Creydt, M., Fischer, M.: Blockchain and more - algorithm driven food traceability. *Food Control* **105**, 45–51 (2019)
5. Yuan, Y., Wang, F.: Blockchain: the state of the art and future trends. *Acta Automatica Sinica* **42**(4), 481–494 (2016)
6. Cachin, C.: Architecture of the hyperledger blockchain fabric. In: *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310 (2016)
7. Wood, G.: Ethereum: a secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **151**, 1–32 (2014)
8. Clack, C., Bakshi, V., Braine, L.: Smart Contract Templates: essential requirements and design options. *arXiv preprint arXiv:1612.04496* (2016)
9. Watanabe, H., Fujimura, S., Nakadaira, A.: BRIGHT: a concept for a decentralized rights management system based on blockchain. In: *IEEE 5th International Conference on Consumer Electronics, ICCE-Berlin, Berlin*, pp. 345–346 (2015)
10. Tian, F.: An agri-food supply chain traceability system for China based on RFID & blockchain technology. In: *13th International Conference on Service Systems and Service, ICSSSM*, pp. 1–6 (2016)
11. Toyoda, K., Mathiopoulos, P.T., Sasase, I., Ohtsuki, T.: Anovel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. *IEEE Access* **5**, 17465–17477 (2017)
12. Caro, M.P., Ali, M.S., Vecchio, M., Giaffreda, R.: Blockchain-based traceability in agri-food supply chain management: a practical implementation. In: *IoT Vertical and Topical Summit on Agriculture - Tuscany, IOT Tuscany*, pp. 1–4 (2018)
13. Liu, P., Liu, W., et al.: Study on application of the fast digital signature algorithm in food traceability system. *Digit. Content Tech. Appl.* **6**(23), 172–178 (2012)
14. Raikwar, M., Mazumdar, S., Ruj, S., Sen Gupta, S., Chattopadhyay, A., Lam, K.: A blockchain framework for insurance processes. In: *2018 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS, Paris*, pp. 1–4 (2018)
15. Yang, Z., Yang, K., et al.: Blockchain-based decentralized trust management in vehicular networks. *IEEE Internet Things* **6**(2), 1495–1505 (2018)
16. Yuan, P., Xiong, X., et al.: Design and implementation on hyperledger-based emission trading system. *IEEE Access* **7**, 6109–6116 (2018)
17. Locust. <https://www.locust.io/>. Accessed 27 Aug 2019