A Comparative Testing on Performance of Blockchain and Relational Database: Foundation for Applying Smart Technology into Current Business Systems

Si Chen, Jinyu Zhang, Rui Shi(✉), Jiaqi Yan, and Qing Ke

Nanjing University, Nanjing 210023, Jiangsu, China
{si.chen,zhjinyu,jiaqiyan,keqing}@nju.edu.cn,
141070049@smail.nju.edu.cn

Abstract. Blockchain technologies have been developing very fast recently. Ethereum, as the second generation of blockchain, can support smart contracts with various functions. Blockchain, combined with IoT technologies like RFID, can bring more security into IoT systems. It has been noted that just storing hash values, addresses and transaction records cannot meet the needs of various blockchain applications. Thus, how to test capacity of reading/writing data for blockchain and further make comparisons with the traditional relational databases needs to be paid attention. Although there has been some research on testing and analyzing the performance of blockchains, few studies focus on analyzing blockchains and relational databases. This test was conducted to analyze the capacities of reading/writing data and processing/recording transactions of blockchains and relational databases. Ethereum and MySQL are chosen as the representatives of blockchain and relational databases. This test has three aims: (1) getting the detailed data of the blockchain's capacity of processing transactions; (2) identifying the bottleneck or potential bottleneck in blockchain systems; (3) putting forward a testing method and testing indexes which are reasonable, practical and compatible with the current development situations of blockchain systems. With the results data, it was concluded that the maximum data volume in single transaction on blockchain network was about 1/10 of MySQL. As for the time spent in processing single transaction, the blockchain network was 80–2000 times as much as MySQL. Thus, it's recommended to store little-size data into blockchain. For more detailed testing, these indexes and testing method can be referenced.

Keywords: Blockchain · Relational database · Performance testing Ethereum · MySQL

1 Introduction

Recently, blockchain has attracted lots of attention and interests from both academia and various industries. The original idea of Bitcoin is put forward by Nakamoto in 2008 [1]. And since then, blockchain, as the underlying technology framework, has been studied and developed greatly [2]. Blockchain has no standard technical definition. And it is

N. Streitz and S. Konomi (Eds.): DAPI 2018, LNCS 10921, pp. 21–34, 2018.
https://doi.org/10.1007/978-3-319-91125-0_2

usually a loose 'umbrella' term used by people in various fields when referring to systems which are similar to bitcoin and its ledger in some respects [3]. Blockchain has three main characteristics, (1) decentralized, (2) recording transactions by some consensus mechanism and (3) tamper-resistant. With these characteristics, blockchain has been honored with 'trust machine' [4]. It's because blockchain provides a practical technical solution to the problem of lack of trust in business world and our society.

Usually, Bitcoin has been considered as the original implementation of blockchain. Bitcoin supports electronic cash transactions peer to peer in its system. And Ethereum, as the second generation of blockchain, supports smart contracts to be deployed and executed on it [5]. With smart contracts, Ethereum blockchain is able to support many functions besides electronic cash transactions, such as digital identity, electronic voting, string data processing and so on. Smart contracts are computer programs deployed on blockchains, which can execute automatically under some specific conditions [6]. In the perspective of data reading and writing, Bitcoin can just read and store numeric values and strings with fixed length. This limitation is constrained to the design of Bitcoin: a design of a peer to peer electronic cash transaction system. Therefore, Ethereum has widely extended the functions of blockchain. In order to support various functions, Ethereum allows data to be read and written in various length. Apart from Ethereum, Hyperledger Fabric project also supports smart contracts [7]. Specifically, Fabric is designed for blockchain applications in industries and there are amount of data with different length in various industries.

IoT technology can connect things in various industries and empower smart industries. And blockchain provide a probability of a secure IoT world. The problem of data security limits the wider applications of IoT technology in industries. However, based on blockchain, smart contracts can be used to implement digital identity and access control of IoT data. Azaria et al. applied blockchain into medical data access and permission management [8]. Li et al. exploited consortium blockchain technology and proposed a secure energy trading system in industrial IoT [9]. Besides academia, there exists examples of such application in industries. IBM and Samsung announced ADEPT Project, which uses blockchain, specifically Ethereum, to build a decentralized and secure IoT [10]. In ADEPT project, blockchain technology can bring trust into supply chain management and help improve their manufacture security and service quality. Microsoft and ConsenSys announced EBaaS Project to offer Ethereum Blockchain as a Service (BaaS) on Microsoft Azure [11]. By offering such BaaS, Microsoft and Consensys can empower a number of developers and companies to develop and deploy blockchain-based applications and systems, which will bring trust and security into their business and profits to Microsoft and Consensys. And Deloitte opened its own BaaS platform named Rubix [12]. Such BaaS projects will empower industries to applying blockchain into their own business more quickly and conveniently. And blockchain binds with IoT more closely.

One big obstacle and challenge for blockchain deployment in different industries is that blockchain system cannot process data in an acceptable way now. A practical solution is to combine blockchain with current relational databases to provide speed-accepted service for users. Blockchain applications are usually combined with electronic cash and financial systems, medical data, digital assets transaction systems and supply chain management systems [8, 13–15]. If all these systems need to implement all the functions that blockchain provides, the capabilities to read and write the data with various length are required. Especially in supply chains, an amount of real-time industrial is generated during the process of manufacturing, logistics and retailing. For example, there are thousands of components in a car and in a supply chain quality management system for the production of cars, how to process such a lot of real-time quality data for these thousands of components is crucial. Therefore, there is a need to test the capabilities of reading and writing data in blockchain and make comparisons between blockchain and mainstream relational databases. Furthermore, considering the differences of capabilities of reading and writing data between blockchain and relational databases, a combination of above two ways may be a practical solution to applying blockchain technologies in various industries. Based on this solution, how to make the distribution of blockchain and relational databases so as to obtain the best balance between efficiency and security, is a realistic question. Thus, it is meaningful to offer a testing framework to compare their own ability of reading and writing data. With such a testing framework, blockchain systems developers are able to carry out necessary test accordingly. And after getting the exact test result, they can make the decision about how to combine the two methods of storing data to reach the balance of efficiency and security.

2 Literature Review

With the rapid development of blockchain technology, investigations on testing various blockchain have been made. However, these studies have mainly focused on just testing blockchains or comparing different blockchains. Dinh et al. has put forward a framework for analyzing private blockchains, called Blockbench [16]. They choose three representative private blockchains as the sample. Private blockchain is such a blockchain that all nodes are authenticated in some way, for example, some specific initial parameters in Ethereum private blockchain or some membership service in Fabric. Data is shared and synchronized among the authenticated nodes in private blockchain. So, private blockchain is not open to the public. It's not like Bitcoin, in which system everyone can participant in or withdraw freely and can access data with no right control. Dinh et al. pointed out that recent private blockchains have made improvement on their consensus mechanism. Considering that the traditional PoW consensus mechanism uses an amount of computational resources to ensure the security, which results in a waste of computational resource and electricity, they have adopted PBFT algorithms or developed their own variants to improve the performance of the consensus mechanism instead of PoW. Dinh et al. divided private blockchains into three layers: consensus layer, data model layer and execution layer. They first measure the overall performance indexes of the three private blockchain, which indexes are throughput, latency scalability and fault-tolerance. Then test perform micro-test on the above three layers. The testing results demonstrate that there exist big performance among these private blockchains, and that these private blockchains are still far from replacing the current database systems to support various applications.

In September 2017, European central bank and Bank of Japan published their joint research project report on testing of blockchain, which project is called Stella [17]. The

Stella project's test object is the blockchain application in the area of financial market infrastructure. Stella project used smart contracts to implement the business logic of transactions and executed these smart contacts in both a virtualized and restricted in-house test environment and cloud computing environment. It concluded that the blockchain based solutions could meet the performance needs in a real-time gross settlement system. Although favorable results had been reached, they still suggested that it's too earlier to conclude that blockchain technologies could be used in realistic production.

Croman et al. studied the performance of bitcoin regarding to the scalability of decentralized blockchains, including maximum throughput, latency, bootstrap time and cost per confirmed transaction [18]. They concluded that a basic rethinking of technical approaches is required to advance current blockchains. Gervais et al. studied the proof of work (PoW) blockchains' security and performance and their influence on each other. They put forward a framework, in which they can capture the current PoW blockchains and develop variants to compare the tradeoffs between their security and performance [19]. Aniello et al. put forward a method of prototype evaluation based on the implementation of their previous work on architect of blockchain-based redo logs [20]. Their prototype adopts a specific consensus algorithm based on a three-phase commit protocol [21]. Their analysis focused on the throughput, stability and latency of two operations, get operation and set operation. Suankaewmanee et al. designed an application of mobile blockchain [22]. And their performance analysis focused on memory utilization, PoW process and chain verification process. They set different numbers of transactions contained in one block and observe the different execution time and energy consumption correspondingly. Spasovski and Eklund conducted testing to analyze performance and scalability of PoS (Proof of Stake) blockchain [23]. They designed and implemented blockchain-based and non-blockchain-based groupware communication applications for comparison testing. Their analysis focused on response time, throughput and network topology. Although their work compared blockchain-based and non-blockchain-based applications, they still focused on the overall performance instead of just data reading and writing module. Walker et al. put forward a platform for trans active IoT blockchain applications repeatable testing [24]. This platform focused on the execution time of various manipulations including miners create, blockchain make, distribute to miners and so on.

As for testing on relational databases, a number of works have been done. Vicknair et al. compared a graph database and a relational database before [27]. They adopted Neo4j as the representative for graph databases and MySQL for relational databases. In their paper, the evaluation was divided into objective measures, execution time for example, and subjective measures including maturity/level of support, ease of programming, flexibility and security. They constructed twelve MySQL databases and twelve Neo4j databases and each database stored a directed acyclic graph (DAG) consisting of some number of nodes and edges. Then they used SQL queries and manipulation sentences to test the capacity of these two kinds of databases. They concluded that both systems performed acceptably and graph database Neo4j performed better than relational database MySQL in structural type queries and full-text character searches and relational database works more efficiently than graph databases in numeric queries. Jing et al. made a comparison between Lucene and relational database [28]. They prepared a test data set and executed queries separately in unindexed relational database and Lucene. An operation audit log was used as the original data source for their performance tests. And the result data was analyzed in aspects of exact query, wildcard query, influence of results set size, combinational query and performance influence of record complexity. Chays et al. put forward a set of tools named AGENDA (A test GENerator for Database Applications) for testing database applications [29]. AGENDA is written for PostgreSQL and can parse a schema, generate a programmatically identifiable database state from that schema, generate test cases, and test the resulting state and output from executing those test cases. While many studies have been done to test blockchain's and relational databases' performance and functions separately, there are just few work on comparing blockchain with relational databases. Regarding to the importance of comparing them mentioned in Sect. 1, we're offering a framework for comparing their capacities of reading and writing data.

3 Testing Design

For this testing, we chose Ethereum to stand for blockchains and MySQL database for relational databases. As for such arrangement, we have three standards to depend on. The first is whether the objects tested are functionally complete or not. Ethereum is Turing-Complete and can support smart contracts to implement various functions and executions [16]. And MySQL can be used to store various kinds of data. Both are functionally complete. The second standard is that objects for testing should be used or adopted widely in their own fields. Ethereum is one of the most mainstream blockchains and has been supporting one of the largest public blockchains since 2014. Also, Ethereum is open source and often adopted to power various private blockchains. There are nearly 900 decentralized apps built on Ethereum now. As for MySQL database, it is the second most popular databases among hundreds of databases according to DB-engines ranking, only a little less popular than Oracle databases [25]. The last standard is that open source is more suitable than not open source. This is out of the needs to know its inner implementation for the research and testing in our next step. And thanks for the work of Blockbench, we could just pick one blockchain for comparison testing without loss of reliability.

Ethereum blockchain based systems' core part can be divided into three layers: data layer, consensus layer, smart contract layer. Data layer is responsible for data storing, direct reading and writing. This layer consists of each node's local database keeping a copy from blockchain. Actually, the reading and writing at data layer is just the same as these same manipulations in traditional databases. Consensus layer is usually responsible for synchronizing blocks and reaching consensus among authenticated nodes to record transactions. Comparing to centralized databases, blockchain systems can spend more time because of the consensus layer recording transactions. Consensus layer may perhaps become the bottleneck of system's performance. Another potential bottleneck is smart contract layer. This layer may perform complex computation tasks, which can also result to the increasing of time spent. However, smart contract just implements the business logic the same as in a centralized system. Reasonably, this part should not be the main point to test. Figure 1 shows the manipulation relations between the general four layers. Users send a transaction request from the application layer to the smart contract layer. After executing some tasks, the smart contract layer will send new transaction to the consensus layer. With reaching the consensus about this new transaction, the consensus layer will admit and record this new transaction and store it into block in the data layer. To sum up, we should focus on the testing for consensus layer's performance. With this in mind, we need to design such smart contracts that are as simple as they can to lower the effect on the testing of consensus layer.

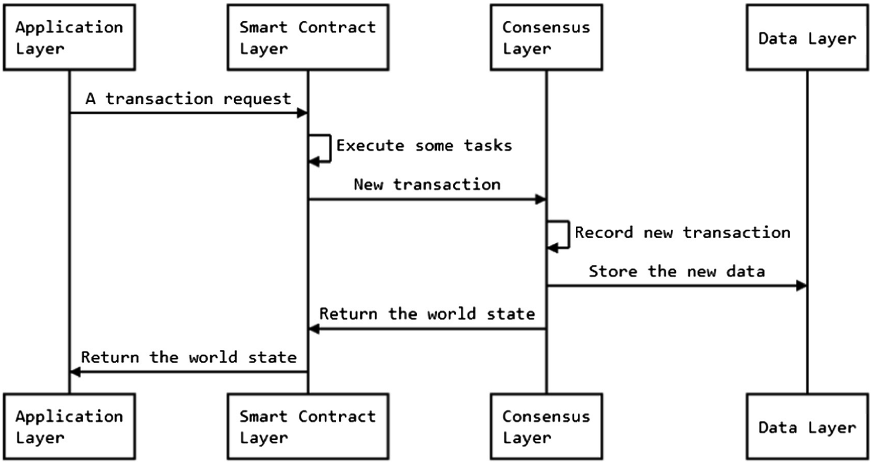

Fig. 1. Figure of manipulation relations between the general four layers.

To assess the ability of reading and writing data, these three main perspectives are necessary to consider: the ability of reaction, throughput and loading capacity. These indexes are also key to a successful service provided by blockchain based systems. For the purpose of exactness, the above indexes were divided into nine more specific and observable indexes (Table 1):

Table 1. Table of the nine specific and observable indexes.

Number	Indexes
T01	The average reaction time of transactions
T02	The longest reaction time of transactions
T03	The rate of successful transactions
T04	The amount of transactions processed per second
T05	The amount of transactions processed in unit time
T06	The most time-consuming module in each transaction
T07	The biggest volume of data processed in each transaction
T08	The execution time of each transaction
T09	Throughput

4 Testing Implementation

A six-nodes private blockchain has been prepared for the test, which was initialized with our own parameters in genesis.json file. The detailed information about our testing machines is as below (Table 2):

Table 2. Table of detailed information of the machines for testing.

Machine number	Operating system	CPU	Disk	Memory
1	Win 10	i7-6700HQ 2.60 GHz 4 cores	M2S3138E-128GM-B	8G
2	Mac OS	2.9 GHz Intel Core i5	APPLE SSD AP0256J	8G
3	Win 10	i5-4210H 2.9 GHZ 4 cores	128G SSD	8G
4	Win 10	i5-4210U 1.70 GHz 4 cores	250G SSD	4G
5	Ubuntu 16.0.4	AMD A8-4500M APU with Radeon (tm) HD Graphics	WDC WD10JPV X-00JC3T 0 (1T)	8G
6	Win 7	i7-5500U 2.40 Gz 4 cores	1 TB/5400 rotation per minute	8G
7	Win 10	i5-4210H 2.9 GHZ 4 cores	1 TB/5400 rotation per minute	8G

The Ethereum applications (or called Dapps, Decentralized applications) development and testing framework Truffle was adopted to perform this testing [26]. Dapps are applications based on blockchain like Ethereum. Dapps consist mainly of smart contracts and other interactive modules. They use smart contracts to read data from blockchain and execute many kinds of tasks like transactions, voting, analyzing and so on. And what we want to test is the capacity of reading and writing data of blockchain. The smart contracts for the test use can be viewed as a very simple Dapps with few other modules. Thus, the Truffle development and testing framework can be used to undertake such tasks in this testing: deploying smart contracts onto blockchain, providing a testing framework to write the testing programs and providing small tools to improve the efficiency of testing.

Based on the testing indexes and analysis on layers in blockchain based systems, two smart contracts were used. The first smart contract Transaction.sol can initialize an account with fixed number of altcoin (represented by numeric value). Transaction.sol can also send some number of altcoin to other accounts and query the balance that some specific account still has. Transaction.sol can cover the indexes from T01 to T08. The second smart contract KvStorage.sol can just put or get some value into or out from our private blockchains, which covers the indexes from T07 to T09. Then two test scripts were used to deploy these two smart contracts and execute the manipulations of

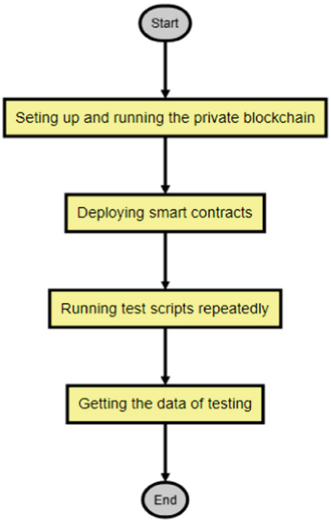

Fig. 2. Figure of the workflow of this testing.

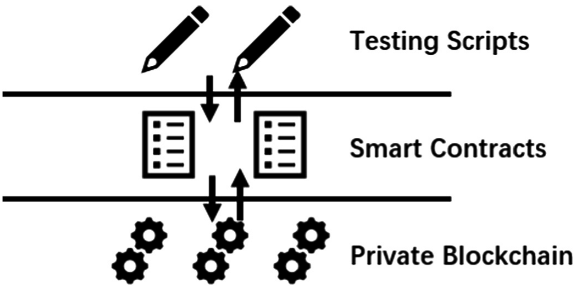

Fig. 3. Figure of the manipulation relationship between testing scripts, smart contracts and private blockchain.

putting data, getting data and sending altcoins to other accounts repeatedly. When testing, a increasing pressure policy was adopted, and the seven kinds of data volume are 128 B, 256 B, 512 B, 1 KB, 2 KB, 3 KB and 6 KB.

The outline of testing on private blockchain part is as follows. A private blockchain of six-nodes was set up firstly. Then the two smart contracts, Transaction.sol and KvStorage.sol, were deployed onto this blockchain. After deploying, the test scripts were executed by some of these seven machines/nodes to get the testing data for analysis (Figs. 2 and 3).

As for MySQL part, a Python script was written to manipulate reading and writing data from and into MySQL database. A Python package named pymysql was imported to implement these functions. Similar to Ethereum part, seven kinds of data volume which are 128 B, 256 B, 512 B, 1 KB, 2 KB, 3 KB and 6 KB were used to test MySQL's performance under the same indexes.

5 Results Evaluation

The testing result data varies among different machines. Considering that the comparison results vary from machines, and the simple average is not enough to stand for the common situations, machine No. 2 was chosen as the standard machine because of its highest complex performance among all the testing machines. The reason is that in the real industrial environment, computing machines' performance is usually higher than these seven portable computers. It's reasonable to choose the machine No. 2 as the standard (Tables 3 and 4).

Table 3. Table of testing results data of the indexes T07, T08 and T09 of MySQL database.

Number	Index	Results
T01	The average reaction time of transactions	125.08 ms
T02	The longest reaction time of transactions	322 ms
T03	The rate of successful transactions	100%
T04	The amount of transactions processed per second	8.00 times
T05	The amount of transactions processed in unit time	8.00 times/s
T06	The most time-consuming module in each transaction	Write/read data: 76.4%
T07	The biggest volume of data processed in each transaction	6,688 bytes
T08	The execution time of each transaction	Refer to Fig. 4
T09	Throughput	Refer to Fig. 6
Number	Index	Results
T07	The biggest volume of data processed in each transaction	65,535 bytes
T08	The execution time of each transaction	Refer to Fig. 5
T09	Throughput	Refer to Fig. 7

Table 4. Table of the comparison between the result data of the index T07 of the private blockchain and MySQL database.

Testing objects	The biggest volume of data processed in each transaction
Private blockchain	6,688 bytes
MySQL database	65,535 bytes

From the table above, it's obvious that the private blockchain's biggest volume of data processed in each transaction, which is 6,688 bytes, is far less than that of MySQL database, which is 65,535 bytes because of the limitation of data type varchar in MySQL. This implies that this six-nodes private blockchain is not suitable to store too large volume of data in a system, or that would lower the performance of the whole system. Considering such a fact, it's natural to choose to store just some hash values into blockchain systems. In the next step, a more specific and reasonable arrangement would be made with the results data. The T06 index's result shows that the potential bottleneck of this private blockchain can be the module responsible for reading and writing data, which consumes about 76.4% of the testing time.

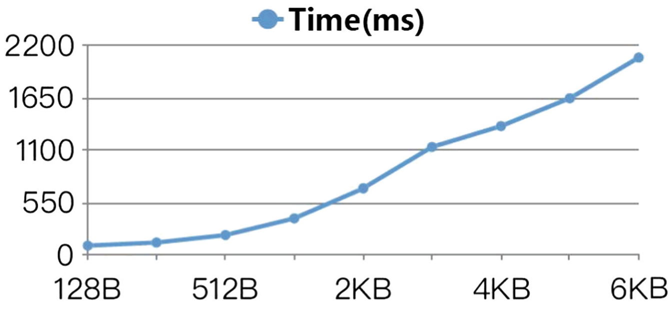

Fig. 4. Figure of relationship between data volume and time for reading/writing per transaction of the six-nodes private blockchain.

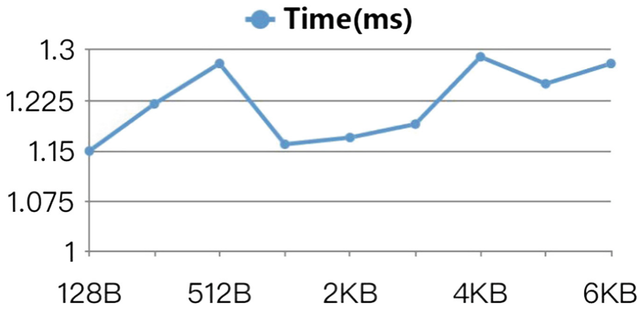

Fig. 5. Figure of relationship between data volume and time for reading/writing per transaction of MySQL database.

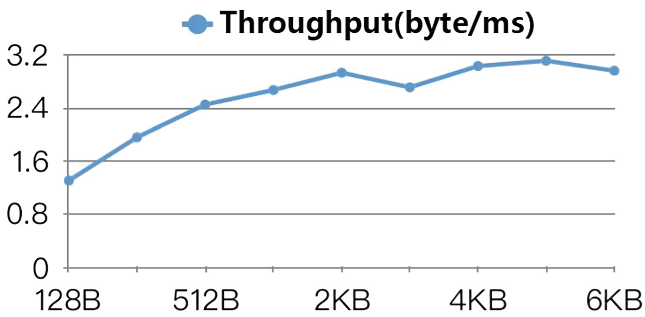

Fig. 6. Figure of relationship between throughput and data volume per transaction of the six-nodes private blockchain.

With the growing data volume of one transaction, the time consumed increase. The used time and data volume are in an exponential-like relationship. Notice that there are six nodes in this private blockchain, the exponential-like relationship may be a result of the number of nodes. This needs more accurate testing to verify (Table 5).

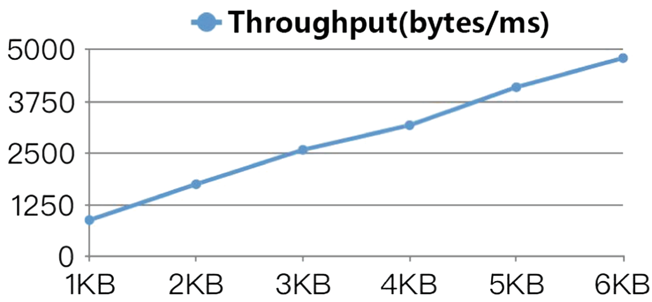

Fig. 7. Figure of relationship between throughput and data volume per transaction of the MySQL database.

Table 5. Table of the comparison between the result data of the index T09 of the private blockchain and MySQL database.

Testing objects	Throughput
Private blockchain	Varies with different data volume and around 3 bytes/ms
MySQL database	Linearly related to data volume

Table 6. Table of the comparison between the result data of the index T08 of the private blockchain and MySQL database.

Testing objects	The execution time of each transaction
Private blockchain	Varies with different data volume, less than 2027 ms
MySQL database	1.22 ms in average

When the data volume keeps increasing and is more than some threshold, here it is 2 KB, the throughput of the private blockchain keeps relatively fixed. Comparing to the six-nodes private blockchain, MySQL shows more high performance with far more bigger data volume to process and linear-like relationship with time consumed (Table 6).

As for the execution time of each transaction, this six-nodes private blockchain spends more time on bigger volume of data, reaching 2027 ms with 6 KB of data. However, the time MySQL spends on different volume has no apparent relation to the data volume. And on machine No. 2, it's average time of reading and writing is 1.22 ms. The time used for reading and writing data which this private blockchain spends is more than 1660 times as long as that of MySQL (Table 7).

Table 7. Table of the comparison between the result data of the index T04 of the private blockchain and MySQL database.

Testing objects	The amount of transactions processed per second
Private blockchain	Varies with different data volume, more than 0.49 times
MySQL database	819.67 times in average

Another important index is the amount of transactions processed per second. The private blockchain's execution amount was affected by data volume and can be low to 0.49 per second. On the contrast, MySQL has a much higher average number of 819.67 per second. And MySQL's performance on the index of throughput is more than 1000 times better than this six-nodes private blockchain.

These results imply that private blockchain may be more suitable for some modules which is not data-intensive in the systems. In this section, three aspects and nine indexes were used to comparing the private blockchain and relational databases. And detailed results data was reached by testing. This work provides an example for comparing the two ways of data reading and writing. Such comparisons are necessary for deciding how to combine blockchain technology and centralized databases in a practical way to gain a balance. And more complex tests are needed for more detailed evaluation between the two ways.

6 Discussion

Although blockchain is an emerging technology filed, and it can bring trust into business world and human society. Successful real-world business use cases based on blockchain are still rare. One big obstacle and challenge for blockchain deployment in different industries is that blockchain system cannot process data in an acceptable way up to now. Thus, a pure-blockchain system may come across failure in real-world business systems and applications. A practical solution to apply blockchain technology into various industries is to combine the two different ways of reading and writing data in systems. Then it's necessary to divide the whole system into data-intensive modules and non-data-intensive modules. The data-intensive modules can be built on relational databases for the high requirement of data reading and writing. And the non-data-intensive modules can be built on blockchain. Another practical solution is to reorganize their business system's architecture. The system can be divided into trust-related part and non-trust-related part. The part regarding trust and security should be simplified in data volume, using hash values for example, to meet the blockchain's data reading and writing performance. So, they can make use of blockchain technology to secure their business. And the part not related to trust and security can store more detailed information and be built on relational databases. With the testing results data and business requirements, a reasonable solution would be reached to obtain the best balance between performance and security.

Reviewing on the above testing, there are three points to improve in the future research. Firstly, the smart contracts are not fine enough to testing each layer's performance. To design blockchain based systems, more exact data is needed. Secondly, a tool named testrpc provided by Truffle framework was used in the testing. This tool will fasten the reaction time of this private blockchain. In the future testing research, such tools should be excluded for a more accurate result. Another important point is that blockchain-based systems and centralized systems are not only different in architecture, but also in the business processing mechanism. More complex tests are needed to evaluate the comprehensive performance of the two types of systems in various industries.

7 Conclusion

Blockchain technology will bring trust, security and decentralization into current business systems and applications. However, the performance of blockchain limits its better application in various industries. Because of blockchain's low capacity of reading and writing data compared to current relational databases, based on pure-blockchain's business systems and applications' speed of processing business cannot be accepted. This is surely a big challenge for blockchain deployment in various industries and our society, where cost trust can be reduced by blockchain technology. If a real-world business system which combine blockchain technology and current relational databases can provide acceptable service for users, it will be a solution for this challenge. A practical solution may be combining blockchain with relational databases to implement industrial functions. IoT technology can connect different kinds of things in real world and bring smart functions into our life and industries. And blockchain will bring security and decentralization. The method and indexes used in this paper provide a foundation for applying blockchain into current IoT systems. With testing data, decisions can be made reasonably about how the design better architecture including blockchain and relational databases, reaching better balance between security and performance. In other words, the method and indexes here will provide testing data to reduce the limit of blockchain's performance and can make contributions to more applications in various industries and IoT world, in a way of combining blockchain with relational databases.

Acknowledgement. This work was supported by Natural Science Foundation of China (NSFC No. 71701091) and the Chinese Ministry of Education Project of Humanities and Social Science (No. 17YJC870020).

References

1. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008). Consulted
2. Romano, D., Schmid, G.: Beyond Bitcoin: a critical look at blockchain-based systems. Cryptography **1**(2), 15 (2017)
3. Narayanan, A., Clark, J.: Bitcoin's academic pedigree. Commun. ACM **60**(12), 36–45 (2017)
4. Katz, D.M., Bommarito, M., Zelner, J.: The trust machine. The Economist, October 2015. https://www.economist.com/news/leaders/21677198-technology-behind-bitcoin-could-transform-how-economy-works-trust-machine. Accessed 28 Dec 2017
5. Ethereum Project Homepage. https://www.ethereum.org/. Accessed 28 Dec 2017
6. Buterin, V.: A next-generation smart contract and decentralized application platform. White paper (2014). http://www.the-blockchain.com/docs/Ethereum_white_paper-a_next_generatsion_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf. Accessed 28 Dec 2017
7. Fabric Project Homepage. https://www.hyperledger.org/projects/fabric. Accessed 28 Dec 2017
8. Azaria, A., Ekblaw, A., Vieira, T., et al.: MedRec: using blockchain for medical data access and permission management. In: International Conference on Open and Big Data, pp. 25–30. IEEE (2016)

9. Li, Z., Kang, J., Yu, R., et al.: Consortium blockchain for secure energy trading in Industrial Internet of Things. IEEE Trans. Ind. Inf. **PP**(99), 1 (2017)
10. IBM and Samsung: IBM ADEPT Practitioner Perspective - Pre Publication Draft. https://zh.scribd.com/doc/252917347/IBM-ADEPT-Practictioner-Perspective-Pre-Publication-Draft-7-Jan-2015. Accessed 28 Dec 2017
11. Gray, M.: Ethereum Blockchain as a Service now on Azure. https://azure.microsoft.com/en-us/blog/ethereum-blockchain-as-a-service-now-on-azure/. Accessed 28 Dec 2017
12. Deloitte's Blockchain Service Homepage. https://www2.deloitte.com/ca/en/pages/technology/solutions/deloitte-digital-blockchain.html. Accessed 28 Dec 2017
13. Tian, F.: An agri-food supply chain traceability system for China based on RFID & blockchain technology. In: International Conference on Service Systems and Service Management, pp. 1–6. IEEE, June 2016
14. Xia, Q., Sifah, E., Smahi, A., et al.: BBDS: blockchain-based data sharing for electronic medical records in cloud environments. Information **8**(2), 44 (2017)
15. Corda Project Homapage. https://www.corda.net/. Accessed 28 Dec 2017
16. Dinh, T.T.A., Wang, J., Chen, G., et al.: BLOCKBENCH: a framework for analyzing private blockchains (2017). https://arxiv.org/abs/1703.04057. Accessed 28 Dec 2017
17. Stella Project Report by European Central Bank and Bank of Japan. https://www.ecb.europa.eu/pub/pdf/other/ecb.stella_project_report_september_2017.pdf. Accessed 28 Dec 2017
18. Croman, K., et al.: On scaling decentralized blockchains. In: Clark, J., Meiklejohn, S., Ryan, P.Y.A., Wallach, D., Brenner, M., Rohloff, K. (eds.) FC 2016. LNCS, vol. 9604, pp. 106–125. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53357-4_8
19. Gervais, A., Karame, G.O., Glykantzis, V., et al.: On the security and performance of proof of work blockchains. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 3–16. ACM (2016)
20. Aniello, L., Baldoni, R., Gaetani, E., et al.: A prototype evaluation of a tamper-resistant high performance blockchain-based transaction log for a distributed database. In: European Dependable Computing Conference (2017)
21. Skeen, D., Stonebraker, M.: A formal model of crash recovery in a distributed system. IEEE Press (1983)
22. Suankaewmanee, K., Hoang, D.T., Niyato, D., Sawadsitang, S., Wang, P., Han, Z.: Performance analysis and application of mobile blockchain. arXiv preprint arXiv:1712.03659, 11 December 2017
23. Spasovski, J., Eklund, P.: Proof of stake blockchain: performance and scalability for groupware communications. In: The International Conference on Management of Digital Ecosystems (2017)
24. Walker, M.A., Dubey, A., Laszka, A., et al.: PlaTIBART: a platform for transactive IoT blockchain applications with repeatable testing (2017)
25. DB-Engines Databases Ranking. https://db-engines.com/en/ranking. Accessed 28 Dec 2017
26. Truffle Framework Homepage. http://truffleframework.com/. Accessed 28 Dec 2017
27. Vicknair, C., Macias, M., Zhao, Z., et al.: A comparison of a graph database and a relational database: a data provenance perspective. In: Southeast Regional Conference, Oxford, MS, USA, pp. 1–6. DBLP, April 2010
28. Jing, Y., Zhang, C., Wang, X.: An empirical study on performance comparison of lucene and relational database. In: International Conference on Communication Software and Networks, pp. 336–340. IEEE (2009)
29. Chays, D., Deng, Y., Frankl, P.G., et al.: An AGENDA for testing relational database applications. Softw. Test. Verif. Reliab. **14**(1), 17–44 (2010)