The Exchange Center: A Case Study of Hybrid Decentralized and Centralized Applications in Blockchain

Yung-Chen Hsieh Chih-Wen Hsueh Ja-Ling Wu

will@cmlab.csie.ntu.edu.tw cwhsueh@csie.ntu.edu.tw wjl@cmlab.csie.ntu.edu.tw

Abstract—
Blockchain and smart contract provide a convenient vehicle to build decentralize applications. Among varies Decenrtalized applications (DAPPs), exchange is one of the most mentioned ones. To fully-decentralized an exchange looks somehow like an utopia. Most works tried to use different methods to solve different problems on this subject. According to our observations, the problems caused by blockchain include 1) Long confirmation time, 2) Vulnerable to front running attack, 3) wasting on-chain resource, all these problems make a blockchain based exchange not user friendly. To deal with the above-mentioned problems, a hybrid centralized and decentralized exchange is presented in this work. By simulating the transaction execution processes and eliminating all the unwanted uncertainties, we are confident that the transactions in the proposed exchange center (EC) will be confirmed and executed well on the blockchain, that is, our system allows users to conclude their trades without need of waiting for a long blockchain confirmation time. In conclusion, a Hybrid Centralized and Decentralized Application (HCDAPP) provides a better-quality service, just like a centralized server; at the same time, it also provides a higher-level security, like a decentralized smart contract.

1 Introduction

Blockchain provides a convenient vehicle and great opportunity to decentralize services over the internet. Bitcoin, the first blockchain project started at 2008, is a decentralized cash transaction system [1]. Bitcoin protocols defined rules on the coin supply, coin transfer, and structures to store the related information. Without a centralized control entity, Bitcoin is able to be run autonomously on a distributed network with the aid of a decentralized consensus algorithm. After Bitcoin, Ethereum provides a virtual environment allowing users to upload a program onto the blockchain and running the program in autonomous sense. This attractive feature booms DAPPs up, this fact can be evidenced by observing various kinds of services have been built on the basis of Ethereum blockchain recently.

Among the existing blockchain based DAPPs, Decentralized Exchange (DEX) is one of the most mentioned and best fitted ones. However, due to the limitations of current blockchain technology, there still have lots of problems if a DEX is running in a fully decentralized sense. Some of the well-known challenges for DEX include: front running [2] of the transactions, long waiting of confirmation, and the waste on-chain resource. For providing better user experience, in this work, a novel approach is proposed and realized to hybrid the involved centralized matching service and the decentralized smart contract.

2 Background

2.1 Decentralized exchange (DEX)

To decentralize an exchange center is one of the most discussed and well fit issues of the blockchain's practical usage. Blockchain provides a public billboard for people to place their orders. Smart contracts, most of them are built on Ethereum, can define the rules for exchanging the tokens among multiple parties, without fetching a trusted third party to hold their funds.

DEX differs from traditional Centralized exchange (CEX) mainly in that DEX enabling users to remain in control of their funds by operating their critical functions on the blockchain. In other words, DEX leverages the technology advantages behind cryptocurrencies themselves to enable a safer and more transparent trading. It solves major limitations faced by cryptocurrency markets, since there is no single point of failure and it aligned the markets with what has made the blockchain technology so powerful. Exchange market is centralized because it is the simplest way to proceed, and it is either too costly or technically too complex to build a fully decentralized platform. Most of the existing DEXs are not fully decentralized, but semi-decentralized.

EtherDelta [5] is one of the oldest projects in the field. It is a DAPP providing with a simple user-interface and basic trading features. A server is used to collect the well-formatted order and provide the related trading information. In EitherDelta, the first step for a user is to deposit funds into the EitherDelta's smart contract. And then the user can make an order by providing the trading volume and price with a signature to ensure the user's identity. Or, one can select an order from the order-book (which contains the list of orders received) and included it into a transaction sent to the EtherDelta's smart contract. The contract will make sure that the order is well-formatted and is attached with a correct signature. If anything goes well, the contract will update the balances in both trading sides.

3 Problems

Since the status of a contract will change per transaction, different transaction orders will produce different status contents.

As we may need to wait for at least one-block confirmation

978-1-5386-4870-4/18/$31.00 ©2018 IEEE

time to make sure that our transaction has been put on the blockchain and well-executed. Unfortunately, for human, even just wait for only 15 seconds, say in Ethereum, will produce bad user experience.

Besides, most of the public blockchains, including Bitcoin and Ethereum, used the proof-of-work (PoW) consensus algorithm to synchronize the status between nodes. PoW is a kind of probabilistic consensus algorithm, this implies a confirmed block may be forked and be discard in the future. Because of this, we can only say that a transaction is stable and secure on the blockchain with a certain percentage of ratio. This ratio is enlarged as the block confirmation number is raised. Therefore, "long confirmation waiting time" seems a necessary evil to ensure a transaction to be stable and well-executed.

The other problem is its vulnerability to frontrunning attack. That is, with higher incentive, someone makes his transaction be processed on the blockchain before others waiting in the pool of transactions. Clearly, this kind of attack may not only make other transactions fail but, even worse, lose the funds. Finally, the Frontrunning attack makes DAPPs quite difficult to guarantee whether a transaction is processed successfully, or not. Besides providing bad user experiences, the pre-described confirmation uncertainty also makes the smart contract be executed very inefficiently.

Consequently, each matched transaction includes only one taker and one maker orders. If there is a large market trader who wants to deal with many maker orders at a time, he needs to send the matched transaction for each order and waits for confirmation each time. This would waste computational and storage resources, especially for duplicated taker orders. Beside the resource problem, transactions generated from the same sender must be executed in sequence, this means the confirmation waiting time would be even longer.

4 Proposed solution

4.1 System Architecture

The restructured DAPP, presented in this work, is a hybrid centralized and decentralized exchange (HDEX). It can be separated into several different parts, the first one is a centralized server which is responsible for collecting valid orders and posting them on the order-book; the second one is a centralized matching server which is used to match the orders stored on the order-book, and the last one is an exchange contract which is responsible for performing the trade rules and protecting the users' rights.

4.2 Eliminate the confirmation waiting time

The confirmation waiting time is designed to ensure a transaction is stable in blockchain. When interact with a smart contract, the situation is even more complex. Since the transaction's order will change the contract's status and generate different results. The confirmation is needed not only for preventing from double spending but also to ensure the transactions be executed in order, on the blockchain. That is, we must ensure our transaction will not be proceeded after other late-ordered transactions; or at least, the statuses we are going to access will not change or still available until our transaction is processed. This is difficult because blockchain is built upon a distributed network and suffers from frontrunning. To prevent from frontrunning and get back the decision power of transaction orders from miners. We make use of the account's nonce, which is defined in the protocol that must be increased by one for each transaction sent from this account. With the aid of this technique, plus access control mechanisms embedded in some specific functions of the exchange contract, we can eliminate the confirmation waiting time when interact with the contract.

4.3 Exchange contract

Smart contract defined the execution rules. In an exchange contract, the pre-defined rules are used to protect the users' rights. Since the proposed DAPP is an HDEX, we need to investigate how can a centralized server influence the trading.

When an exchange contract received a matched transaction, it will first check the transaction's sender - the sender must be one of the pre-negoitated administrators. Then the contract will check every order in the transaction. The order must provide a valid signature which can only be generated by a legal user's private key. The other necessary condition checks include: the balance check and the price check. Since limited number of orders are used in our system, the matched price should be better than that of all in the other orders. Only if all the orders passed the condition checks, the contract will update the user's balance and close the trade.

5 Conclusion

A hybrid centralized and decentralized exchange is presented in this work. By using a proper access control mechanism, we successfully link a centralized matching server with a decentralized exchange smart contract on Ethereum. By simulating the transaction execution processes and eliminating all the unwanted uncertainties, we **are confident** that the transactions in the proposed EC will be confirmed and executed well, in other words, our system allows users to conclude their trades without need of waiting for a long blockchain confirmation time. Besides the user experiences, the matching server takes back the decision power over transactions' execution order from miners, which resolves the frontrunning problem of the blockchain. An HCDAPP can provide a better-quality service, like the centralized server; at the same time, it can also provide a higher-level security, like the decentralized smart contract. It is our belief that the proposed HDEX is a better choice for constructing a blockchain based EC than its DEX and/or CEX counterparts.

6 Reference

[1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008): 28.
[2] Frontrunning explation from investopedia. https://www.investopedia.com/terms/f/frontrunning.asp
[3] Nick Szabo. The idea of smart contracts. http://szabo.best.vwh.net/smart contracts idea.html, 1997.
[4] Ethereum Foundation. Ethereum's white paper. https://github.com/ethereum/wiki/wiki/White-Paper, 2014.
[5] EtherDelta, https://etherdelta.com/