ChainDB: A Peer-to-Peer Database System

Part 1 - Consensus

BitPay
bitpay.com

Abstract. Bitcoin [1] has demonstrated a new approach to securely storing data in the cloud. The Bitcoin database services millions of users around the planet and has thus far shown itself to be essentially invulnerable to attack. Bitcoin does not require boundary defenses to protect the integrity of the database, instead it relies on a consensus of peers to build an independently verifiable historical record. To alter past revisions to the database, an attacker must wield computing power in excess of the combined power of honest participants in network. ChainDB seeks to apply that same approach to securing arbitrary sets of data in the cloud. However, rather than create yet another Bitcoin clone that requires its own mining network, ChainDB uses the existing Bitcoin mining network. In this paper, we outline the ChainDB consensus mechanism.

1. Introduction

As of this writing, the Bitcoin mining network exceeds 350 PH/s [2], making it the most powerful decentralized computing network ever created (aside from the whole of the Internet itself). The power of the Bitcoin network makes Bitcoin the most counterfeit proof form of payment ever invented. It also makes it the most secure database to have ever existed. Any data or reference to data embedded in Bitcoin's block chain is the best available proof that something existed at a given time in the past (to an approximate 10 minute resolution). We seek to harness this computing power to secure other sets of data and we call this system ChainDB.

A ChainDB organizes transactions (in the database sense) into blocks that are then referenced by transactions in the Bitcoin block chain. Validation rules specific to a ChainDB govern whether a given chain of blocks and transactions is valid. Participants in a chain create blocks and append them to the chain by competing to get their block defining transaction into the Bitcoin block chain. Block selection rules govern which participant successfully defines the next block. As with Bitcoin, the winner of a block receives rewards in the form of newly created assets or transaction fees. The exact nature and composition of the reward is specific to a chain.

2. Transactions

A ChainDB transaction is an atomic operation that mutates the state of the database. If it is a financial related database with transferable assets, a transaction might look very similar to a

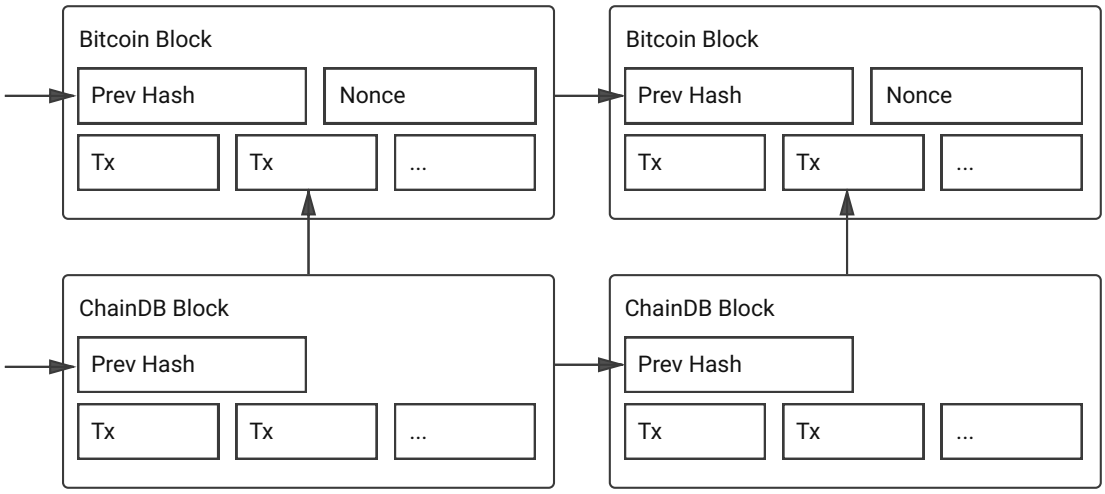
Bitcoin transaction, but that need not be the case. The semantics of a ChainDB transaction can be fixed and determined a priori, or they might employ a scripting system similar or identical to that of Bitcoin. Transactions in a ChainDB must adhere to a set of transaction validation rules that are specific to the chain. The structure and composition of transactions in a ChainDB is unrelated to the consensus process that organizes the transactions into a semantically consistent history.

3. Blocks

Like Bitcoin, ChainDB transactions are organized into blocks. ChainDB blocks are linked to the Bitcoin block chain by embedding the hash of the ChainDB block in a Bitcoin transaction. By rule, no more than one ChainDB block can be defined in a Bitcoin block. It is not necessary for a ChainDB block to be defined in every Bitcoin block if the requirements of the database do not demand frequent timestamping. Timestamping ChainDB blocks with less frequency will incur less cost. Block validation rules for a ChainDB govern the frequency at which a chain can be timestamped in the Bitcoin block chain. If more than one block-defining transaction appears in a Bitcoin block, we employ the following criteria to select the best chain:

1. Longest chain (of ChainDB blocks)
2. A deterministic function that incorporates the hash of the Bitcoin block and produces an ordering of transactions in the block weighted by transaction fee (higher fee transactions have a greater probability of appearing earlier in the order than lower fee transactions)

The first rule is analogous to Bitcoin's best chain selection algorithm. In the second rule, applying an unpredictable order for selecting the best chain mitigates certain attacks by Bitcoin miners. Were we to simply choose the first block defining transaction, mining pools could always win ChainDB blocks using minimal or no transaction fees. A comprehensive ordering of all transactions also makes it possible to select a block even when some blocks are being withheld.



A single ChainDB block can contain hundreds, thousands, or even millions of transactions, all of which are linked to the Bitcoin block chain in a single Bitcoin transaction. This makes ChainDB a scalable extension of Bitcoin without causing additional load on the Bitcoin network. Assuming a Bitcoin transaction fee of 0.0001 BTC, at today's exchange rate a chain with a 1:1 block rate to bitcoin would incur a base cost of roughly $3.40 per day. However, for private databases, a much lower frequency of block creation may be perfectly acceptable.

4. Bidding

The Bitcoin system is successful largely because it enables honest participants to pool their resources(computing power) to thwart an attacker or colluding group of attackers. The incentives for Bitcoin mining ensure that honest participation is more profitable than attack. We seek to achieve the same incentives and dynamics with ChainDB. A process called bidding ensures that honest participants can pool resources to build the best chain.

Nodes that produce ChainDB blocks are called builders. A chain builder organizes valid transactions into blocks and creates a Bitcoin transaction that references the chain and ChainDB block. The Bitcoin transaction includes a miner fee that reflects the value of defining a ChainDB block to the bidder. To minimize the risk that a builder loses bitcoin in the bidding process, builders coordinate to select a common UTXO that all bid transactions use as an input. In so doing, bid transactions are created such that they deliberately conflict. It is therefore impossible for more than one to be included in the Bitcoin block chain. Honest participants are incentivized to coordinate the bidding process such that losing bids do not cost the builder. In normal operation, a Bitcoin block would only have one ChainDB block defining transaction for a given chain. Only in cases where there is a network split or deliberate attack on a chain would more than one ChainDB block defining transaction appear in a single Bitcoin block (nLockTime is used in the bidding process to mitigate the risk of multiple block defining transactions ending up in the same Bitcoin block).

While miners do not need to directly validate the transactions and blocks in a ChainDB, they will be rewarded for securing ChainDB's blocks via transaction fees associated with bids. It is possible that as a ChainDB becomes increasingly popular, Bitcoin miners may directly participate in a chain and create their own ChainDB block defining transactions. Miners could collect the ChainDB rewards directly for themselves. However, with nearly all of the value associated with a ChainDB block manifesting itself as a Bitcoin transaction fee, there is little reason for a miner to do so.

5. Network

A ChainDB network functions much like the Bitcoin network (with the additional requirement that fully validating ChainDB nodes must also be fully validating Bitcoin nodes). Like Bitcoin, ChainDB nodes broadcast ChainDB transactions and blocks to all nodes, each of which performs independent validation. The steps to run a ChainDB network node are as follows:

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. Each node produces a Bitcoin "bid" transaction to define the block (using an unspent output that it attempted to coordinate with other nodes)
4. The bid transaction is broadcast to Bitcoin miners and other ChainDB nodes
5. Nodes accept a new Bitcoin block and select the next ChainDB block from the transactions in the block (if present)
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash

The coordination of an unspent output is a process of nodes both creating and listening for bids. A bid consists of a new unspent output (of zero value), the private key needed to spend the output, and the transaction that defines the next ChainDB block. Initial bids will carry zero fee (this ensures they have zero chance of being selected based on the weighted selection algorithm). Each node creates and broadcasts a bid. As a node hears about other bids, it will update the unspent output that it uses for bids if another bid's unspent output has a lower value transaction id. Future bids will use this new unspent output. New bids replace old bids only when they carry a higher transaction fee, or the unspent output has a lower value transaction id. Bitcoin miners are incentivized to include bid transactions carrying the highest transaction fee.

This system of bidding will benefit from miners that adopt a replace-by-fee strategy for ChainDB bid transactions. Replace-by-fee has been the subject of recent debate and controversy. Some have argued that replace-by-fee should be applied to all Bitcoin transactions. Opponents argue that replace-by-fee makes zero confirmation double spending much easier and thus reduces the utility and value of Bitcoin. Replace-by-fee could be adopted only for ChainDB bid transactions and not affect normal Bitcoin transactions. It would be necessary to ensure a ChainDB transaction couldn't be used to make a normal payment (i.e. payments to older Bitcoin clients that may be unaware of ChainDB bids). There are also a few non-standard aspects to a ChainDB bid transaction that would result in the Bitcoin mesh network not relaying bids. The mesh network rules around relaying could be adapted, or an alternate network for transmitting bids to miners could be employed.

6. Application

While ChainDB is designed with open participation, public databases in mind, private or shared private databases can also benefit. An individual or organization could store critical information on machines that are readily accessible online and benefit from the counterfeit resistant properties of ChainDB. The keys that are necessary to create transactions can remain on systems that are more difficult for an attacker to penetrate. A consortium of individuals or organizations can jointly operate a private ChainDB without any participant needing to trust any other entity as a single source of truth. The nodes that power a private chain could be loaded with small amounts of bitcoin for the purpose of building the chain. Alternatively, they could generate

the bitcoin they need if they have mining hardware. ChainDB nodes could be configured to obtain a certain percentage of blocks up to some maximum rate of expenditure.

7. Conclusion

In this paper we propose a mechanism for creating counterfeit resistant databases. By utilizing the Bitcoin mining network, a ChainDB benefits from the full power of the Bitcoin mining network to create an unforgeable history of database mutations. To the extent a ChainDB has value and participants are willing to extend its chain, that value will be reflected in the Bitcoin transaction fees that participants are willing to bid. One of the most promising aspects of this project and similar projects is that the existing Bitcoin network can be leveraged for new applications with little or no change to the core Bitcoin protocol. There is a substantial amount of risk involved in any material change to the core Bitcoin protocol. Avoiding such risk where possible is highly desirable.

8. Acknowledgements

Much of this work has evolved from the need to secure BitPay's payment system and many hours spent brainstorming how to accomplish that task. Many brilliant developers (and non--developers) have contributed various ideas reflected in this paper (and many ideas that will be covered in future papers on ChainDB).

Outside of BitPay, we must also recognize the many projects and experiments that have informed our thought process. Namecoin [3], which uses merged-mining was one of the earliest attempts to apply the Bitcoin consensus mechanism to other domains. While merged-mining works well, it is a technique that we felt would not scale to hundreds or thousands of databases. Only the most popular databases would enjoy the benefits of widespread merged mining.

The Pegged Sidechains [4] white paper has sparked a lot of enthusiasm about the prospect of using Bitcoin as an asset that can move between chains. The BitShares project [5] has a compelling model for asset issuance and exchange. Ethereum [6] has made progress in the areas of smart contracts and more advanced scripting (among other advancements). Truthcoin [7] lays out a template for embedding external data into a block chain (where it can be referenced and used by smart contracts). And Factom [8] has advanced the concept of using the Bitcoin block chain directly for timestamping data. These are just a few of the many projects that are advancing the state of the art in Bitcoin.

References

[1] S. Nakamoto, "Bitcoin," https://bitcoin.org/bitcoin.pdf, 2008.

[2] Bitcoin Difficulty, https://bitcoinwisdom.com/bitcoin/difficulty

[3] Namecoin, https://namecoin.info/

[4] A. Back, et al, "Enabling Blockchain Innovations with Pegged Sidechains", http://www.blockstream.com/sidechains.pdf, 2014.

[5] BitShares, https://bitshares.org/

[6] Ethereum, https://www.ethereum.org/

[7] P. Sztorc, "Truthcoin", http://www.truthcoin.info/papers/truthcoinwhitepaper.pdf

[8] P. Snow, et al, "Factom", https://github.com/FactomProject/FactomDocs/blob/master/Factom_Whitepaper.pdf