小型端末におけるブロックチェーンを用いた分散ストレージ方式の研究

KOBAYASHI, Reiji / 小林, 澪司

(出版者 / Publisher)
法政大学大学院理工学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 理工学研究科編
(巻 / Volume)
64
(開始ページ / Start Page)
1
(終了ページ / End Page)
6
(発行年 / Year)
2023-03-24
(URL)
https://doi.org/10.15002/00026396

小型端末におけるブロックチェーンを用いた分散ストレージ方式の研究

RESEARCH ON DISTRIBUTED STORAGE SCHEMES USING BLOCKCHAIN IN IOT DEVICES

小林澪司
Reiji KOBAYASHI
金井敦

法政大学大学院理工学研究科応用情報工学専攻

In recent years, cloud storage has begun to be used by consumers. However, the major cloud storage services currently available have the potentials to be hacked by a comapany's servers, or even by an insider to steal or extract the data they have been asked to store. There are systems that use blockchain to solve this problem, but they require the use of high-performance terminals such as PCs, since blockchain nodes need to be set up. In this study, we propose a method that allows a small terminal to communicate with distributed storage via a PC at home.

***KeyWords**: Blockchain, cloudstorage*

１．はじめに

近年において、回線技術の発展により、消費者は画像を始めとしたファイルの永続的な保存先としてクラウドストレージを利用し始めている。しかし、現座主要なクラウドストレージサービスにおいてはGoogle Driveや企業のサーバがハッキングされたり、アクセスが集中することによるサーバダウン、さらには内部の人間に保存を依頼したデータがデータを流出させたり抜き出される可能性もある。そして、それらの問題の解決を行うためにP2P通信を利用した分散ストレージシステムが提案されてきたが、経済的インセンティブが存在しないため誰かに保存してもらう際に責任を持たせることが出来ない。これに対してブロックチェーンを利用して経済的インセンティブを暗号通貨として解決しているシステムが存在するが、ブロックチェーンノードを立てる必要があるため、PCのような高性能な端末を利用する必要がある。この研究では自宅にあるPCを経由させ、小型端末でも分散ストレージとのやりとりが出来る様な方式について提案する。

２．使用技術

（１）ブロックチェーンとは

ブロックチェーンとは、分散型ネットワークを構成する複数のノートに対してデータを同期する手法である。データのやり取りの最小単位をトランザクションと呼び、トランザクションのまとまりをブロックと呼ぶ。ブロックをブロックチェーンに格納する方法については、大量のブロックが同時にブロックチェーンに追加されて順番が一致しない問題を避けるためにコンセンサスアルゴリズムという名前で研究されている。利用するブロックチェーンの種類に依るが、ビットコインやイーサリアムで使われているアルゴリズムはProof of Workと呼ばれる方式のため、トランザクションの承認には膨大な計算を必要とすることで悪意のあるノードが安易に不正なブロックを格納することを拒むことを可能にする。ブロックチェーンはネットワーク内の取引内容の公開範囲によって、パブリック型とプライベート型に分かれる。

a）パブリック型ブロックチェーン

パブリック型のブロックチェーンはインターネットに接続している誰もが自由に取引に参加できるブロックチェーンである。管理者がいないことが多く、ブロックをブロックチェーンに取り込む際の合意形成にはコンセンサスアルゴリズムが一般的である。コンセンサスアルゴリズムについてはProof of Workと呼ばれる計算能力を利用したものが多く用いられてきたが最近ではマイニングによる多くのエネルギー消費の問題が指摘されてきおり、さらにコンセンサスアルゴリズムの研究が進むことでProof of Importanceなどの様々なアルゴリズムが提案されている。

b）プライベート型ブロックチェーン

プライベート型のブロックチェーンはパブリック型のブロックチェーンとは異なり、誰にでも公開されておらず、何かしらの承認が必要であることが多い。また、プラ

イベート型ブロックチェーンは全員が公平な立場であることが多いパブリック型ブロックチェーンとは異なり、管理者が存在する場合が多い。管理者が存在するブロックチェーンでは管理者が常に正しいという原則を利用することでブロックの取り込みに必要なマイニングを省くことが出来る。また、管理者が存在しない場合においても、参加者が少ないので合意形成にかかる時間やコストを抑えることができる。しかし、この場合においては透明性は参加者の中でしか確保されないという問題がある。プライベート型のブロックチェーンの例としてはHyperledgerFabric などがある。

３．関連研究

（１）Filecoin

Filecoin[1]とはファイルの保存と取得が可能なP2Pなネットワークである。経済的なインセンティブが既に組み込まれていて、ユーザが指定した期間ファイルをほぼ確実に保存されることを可能としている。独自のブロックチェーンをもち、IPFS を利用したアーキテクチャとなっている。ほとんどのブロックチェーンプロトコルではマイナーはネットワークに参加しており、ブロックチェーンを発展させるために計算能力の提供を行い、暗号通貨を得ている。Filecoin では計算能力を提供する代わりにストレージの提供や検索機能などを提供する。そしてクライアントと取引を行い、データを保存および取得し、その見返りに Filecoin でのネィティブ通貨である FIL を受け取る。

（２）Sia

Sia[2]は 2013 年に HackMIT で David Vorick と Luke Champine によって考案された、ブロックチェーン技術によって保護されているクラウドストレージプラットフォームである。P2P や企業レベルのクラウドストレージサービスとの競合することを意図して取り組まれた研究であり、本研究と同様にクラウドストレージサービスからストレージを借りるのではなく、ピア同士が互いにストレージを借り合う仕組みを提案している。ブロックの取り込みなどについては言及されておらず、ビットコインをベースにしたプロトコル拡張されたアルトコインとして実装される予定がされていた。そのためファイル周りのコンセンサス以外はビットコインのものと変わらず、マイニングなどについても言及されていない。Sia と Bitcoin のプロトコルの違いで大きく異なるのはトランザクションの内容であり、Bitcoin は Script と呼ばれる様々な命令の集合を組み合わせることで取引の種類の幅を持たせているが、Sia はスクリプトシステムを完全に排除し、その代わりに M 対 N のマルチシグネチャ方式を使用している。Script を排除することにより複雑さを減らしつつ、攻撃対象を減少させている。さらに、ストレージ保存の契約内容の作成と実際の契約の執行を可能にするためにトランザクションを拡張し、契約、証明、契約更新という 3 つの機能を追加している。Sia ではトランザクションの発行時に通常のフィールドに加えて FileContract フィールドを追加することが出来る。FileContract には期間やチャレンジ頻度、報酬、見逃すことのできる証明の回数などが含まれる。チャレンジ頻度とはファイルの保存証明を提出しなければならない頻度を指定して、保存証明の提出期間のウィンドウを作成する。1 つのウィンドウごとに一回の保存証明の提出を必要とする仕組みとなっており、提出先をブロックチェーンとする。本研究と異なる点ではストレージ提供者とストレージ利用者がブロックチェーンに保存証明の提出をし続ける部分と、プロトコルレベルでストレージ提供者、ストレージ利用者、ブロックチェーンノードが分離されていない点、そしてチャレンジのタイミングがあらかじめ指定されている点である。本研究ではチャレンジのタイミングを動的に指定しておくことでストレージ利用者がオンラインでなければならない時間を削減していることと、ブロックチェーン上で保存証明の提出をし続けるのではなく、プライベートチェーンで保存証明の提出を行うことで必要な負担を削減している。加えて本研究では、ファイルの保存期間中でのパブリックなネットワークと繋がっているブロックチェーンノードとの接続は不要となっている。

４．背景

近年、IT 技術の進歩により消費者は情報通信端末としてスマートフォンを持つようになり、誰もが Web に接続するようになってきた。さらにスマートフォン自体の性能も向上するに連れ、より高画質な写真や高品質な動画を保存出来るようになり、処理能力の向上による様々な 3D 表現を持つアプリケーションの作成が可能となり、消費者が利用するストレージの容量が多くなってきている。これによりユーザは内部ストレージとは別の保存先としてクラウドストレージを利用する様になってきている。しかし、現在の大企業がクラウドストレージ事業を進め、一般消費者がそれらを利用することによるリスクがある。最初に挙げられるのは価格であり、主要のクラウドストレージサービスはいくつかの大企業によってほとんどの利用者を囲い込んでいる。これは消費者が信頼性がありそうという観点で大企業を自分のデータの保存先に選ぶため自然だが、これによりクラウドストレージ産業自体への参入障壁が高くなってしまい、クラウドストレージ提供業者間での価格競争が行われにくい問題があった自分の預けていたデータを持つ企業が値上げなどを行なってしまう可能性が考えられる。また、企業が所属する国家の検閲を受けるリスクや、その企業の従業員が漏洩させてしまうなどのプライバシーの問題がある。これに対してブロックチェーンを利用した非中央集権で誰もがストレージを提供できて誰もがストレージを利用できるシステムについて提案されてきたがパブリックなブロックチ

ェーン上でのやり取りのためノードとしてブロックチェーンに参加しストレージ提供者側になるために求められるコンピュータのスペックが高い問題があった。今回の研究ではスマートフォンなどの家庭用コンピュータに比べて低性能なデバイスにおいてもストレージ提供者側に立つことを可能にし、ある程度ストレージ提供者側がファイルを適切に保存しないなどの不適切な振る舞いを行なっても問題無く預けたファイルが復元できるような、方式について提案を行い、性能についての検証を行う。さらに非中央集権型を維持するために既存の大規模なスマートコントラクトが利用可能なブロックチェーンである Ethereum に提案方式の一部をプロトタイプ実装し、性能検証を行う。

５．提案手法

前提として自宅のプライベートなネットワークが以下のように接続されていると仮定する。ここで PC とはパブリックなブロックチェーンネットワークに接続できる性能を持ったコンピュータとし、外部のネットワークと通信が出来るとする。また、スマートフォンなどの PC に比べて大幅に性能差があるようなコンピュータを IoT と呼び、IoT は PC と同一のネットワークに存在するが、必要に応じて外部と通信が出来るとする。そして、IoT は自身への接続先を持つ。本研究では例としてグローバル IP アドレスを持っていると仮定する。その時の構成図を図 1 に示す。

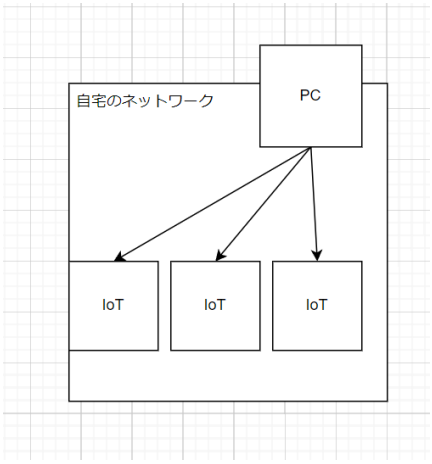

図 1 自宅のプライベートネットワーク

PC は他の PC と接続してパブリックなブロックチェーンを構成する。パブリックなブロックチェーンでは、ファイルの保存依頼やその保存依頼への参加など、公開された掲示板として機能し、ストレージの提供者(StorageProvider)とファイルの保存依頼主(Customer)を取り合わせる。また、その中に適切にストレージの提供が行われているかを監視する監視者(Observer)を含める。パブリックブロックチェーン上で StorageProvider と Customer の取り合わせが完了した際、Customer は自身の接続先情報を StorageProvider と Observer に共有する。ストレージの共有の際、ファイルを暗号化した後、リード・ソロモン符号方式で符号化する。ここでシャードがストレージ提供者の数になるように符号化を行い、それぞれのシャードをストレージ提供者に割り振る。これにより、ストレージ提供者を増やすことによってファイルの可用性を高めることが出来る。そして、Customer と StorageProvider と Observer はプライベートブロックチェーンを構成する。構成したときの構成図を図 2 に示す。

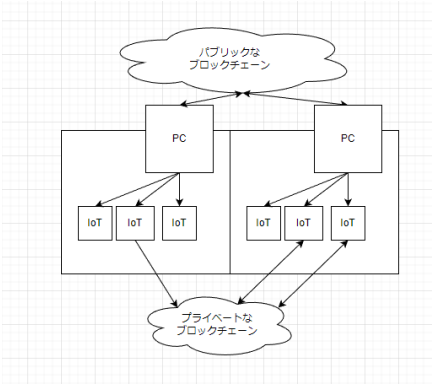

図 2 全体の構成図

プライベートチェーンでは Customer があらかじめ決められた、望んだタイミングで StorageProvider に保存証明の提出を求めることが可能であり、StorageProvider が正しい証明の提出が出来なかった場合、Observer の承認を得て正しい振る舞いを行わなかった StorageProvider を追放することが出来る。
ファイルの保存期間が終了した際、StorageProvider と Observer は Customer から報酬を受ける。

６．評価概要

提案方式についてプロトタイプを作成し評価を行う

（１）リードソロモン符号のプロトタイプ概要

今回は IoT デバイスとして Raspberry Pi を利用してリード・ソロモン符号でファイル分割と復元をしたときの CPU とメモリ負荷、計算速度について算出を行う。また、パブリックブロックチェーン上でのトランザクションについて Ethereum 上で構成した際の消費された ether と経過時間について計測 を行う。ファイルの分割操作では、テスト用の画像ファイルを AES 暗号化した後、リードソロモン符号化して符号化したバイナリをファイルに保存を行い、復元操作については分割操作で保存したファイルを読み込み、リードソロモン符号をコードし、AES 暗号方式で複合し、元の画像ファイルに復元を行った後ファイルを保存する。この操作をテスト用画像を 100x100～3000x3000 のサイズに拡大縮小したものをテストデータ

として利用する。

（2）パブリックなブロックチェーンのプロトタイプ概要

提案手法のコスト計測のため、分散型アプリケーション開発プラットフォームである Ethereum を利用して市マートコントラクトを作成した。Ethereum はプログラムの実行環境として Ethereum Virtua lMachine と呼ばれる実行環境を持ち、ステートマシンの役割を担っているためトランザクションに書かれたプログラムを実行することで状態を書き換えることが出来る。Ethereum ではオペコードごとに消費される ether の量が設定されていて、それらは gas と呼ばれる。今回は提案手法をスマートコント楽に起こしたと際、スマートコントラクトの作成時と参加関数を呼び出したときの gas の消費量について計測する。また、計測に使用する

7． 評価結果

（1）リードソロモン符号のプロトタイプ概要の評価結果

ファイルの分割操作に経過した時間について横軸を画像サイズとしてプロットしたものを図 3 に示す。

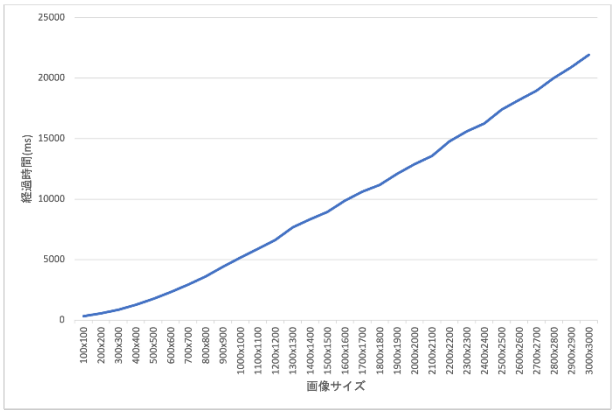

図 3 テストデータの分割操作をした際の経過時間のグラフ

ファイルの分割操作で使用された最大メモリ使用量について横軸を画像サイズとしてプロットしたものを図 4 に示す。

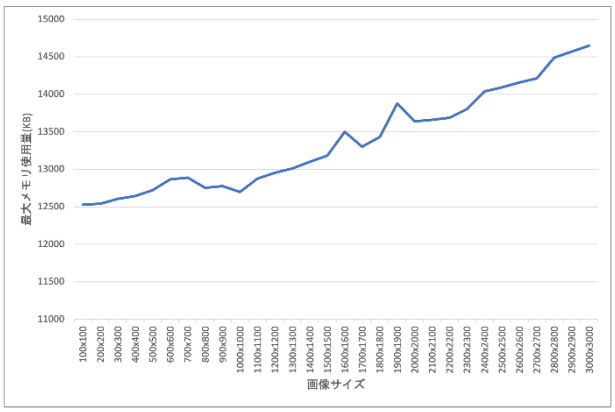

図 4 テストデータの分割操作をした際の最大メモリ使用量のグラフ

ファイルの復元操作に経過した時間について横軸を画像サイズとしてプロットしたものを図 5 に示す。

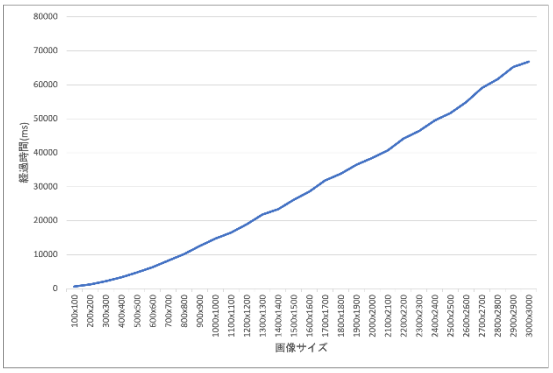

図 5 テストデータの復元操作をした際の経過時間のグラフ

ファイルの復元操作で使用された最大メモリ使用量について横軸を画像サイズとしてプロットしたものを図 6 に示す。

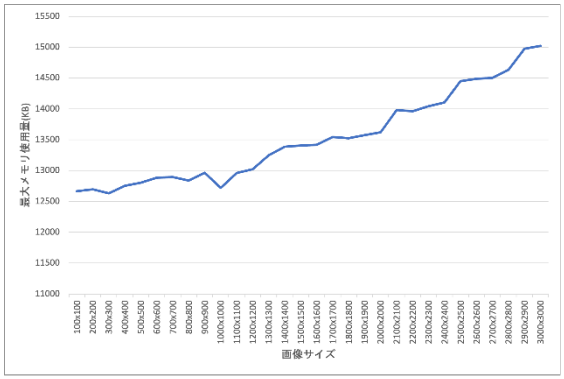

図 6 テストデータの復元操作をした際の最大メモリ使用量のグラフ

（2）パブリックなブロックチェーンのプロトタイプ概要の評価結果

StorageProvider の数は 5~10 とする。ここで評価時に利用する StorageProvider の数と使用した gas の量

(gasUsed)についての結果を表 1 に記載する。

表 1 StorageProvider の数と使用した gas の変化

StorageProvider	gasUsed
5	1105417
6	1130633
7	1155849
8	1181065
9	1206269
10	1231497

次に StorageProvider の参加、observer の参加関数を読んだときの gas の消費量について表 2 に示す。

表 2 参加関数を呼んだ時の gas の消費量

StorageProvider として参加	Observer として参加
66927	66926

８．考察

評価の結果を元に考察を行う

（１）消費される gas について

表 7 により StorageProvider の数を増やせば増やすほど消費される gas の量は増えていった。これはシャードのハッシュを StorageProvider ごとにブロックチェーンに保存する処理が含まれているためであり、ブロックチェーンに保存するデータの量が増えれば増えるほど gas の量は増えていくため、結果 StorageProvider の数を増やせば増やすほど消費される gas の量は増えていくと考えられる。

（２）StorageProvider がファイルを失った時について

StorageProvider がファイルを失った時、Customer が元のファイルを保持していなかった場合、そのファイルは永久に失われることになる。そのため、storagePovider にはファイルを保存する際に責任を持たせる必要がある。StorageProvider はパブリックチェーンに展開されたスマートコントラクトの参加関数を呼び出すときに一定の掛金(premium)を必要とすれば良く、StorageProvider が振る舞いが適切でない場合、つまり保存の証明が取れないの時にファイルを Customer に送ることができなかった場合に報酬と共に掛金を失う仕組みにすれば良い。

（３）StorageProvider が Observer と結託した時について

StorageProvider が Observer と結託したとき、つまり requestTiming の時に Customer にファイルが送られなかったがために追放投票されたが、Observer の一人が StorageProvider と結託した際、StorageProvider の追放を阻止することが可能となってしまう。この問題に対して部分的に解決するためには Observer の追放機構を追加する必要がある。StorageProvider の追放の手順同様に Observer に対しても通報を可能にする。ここで投票者は対象者以外の Observer 全員と Customer にすれば良いと考えられる。

（４）実用上必要な端末数について

ファイルを正しく復元できる確率を 99.9999%を目標とした際、StorageProvider がどこまでファイルのロストを行っても良いのか、99.9999% を満たせるような StorageProvider の数はいくつなのかについて検証する。リードソロモン符号におけるシンボルをシャードとすると、StorageProvider が保管するファイルはリードソロモン符号におけるシンボルになる。リードソロモン符号ではある一定のビット数をまとめて 1 シンボルとして扱う。ここでは、kビットとする。そしてリードソロモン符号は情報シンボルiと冗長シンボル$2r$で構成される。冗長シンボルが$2r$に設定されているとき、リードソロモン符号で符号化された情報シンボルと冗長シンボルをまとめたシンボル$s = i + 2r$の内、rシンボルまで修正が可能となる。ここで、kが十分に大きいとし、$i = 5$として、StorageProvider が Customer にファイルを送らない確率を 5%~30%まで 5%刻みで変化させていった時に Customer がファイルを正しく復元できる確率 h を 99.9999%以上になるような StorageProvider の数について表 9 に記載する。ここで StorageProvider のファイルロスト率を p とした時に Customer がファイルを正しく復元できる確率 f は$f = \sum_{j=0}^{Floor(\frac{h-i}{2})} {}_hC_j(1-p)^{h-j}p^j$で表すことが出来る。また、StorageProvider がファイルを送信しない確率を「ファイルロスト率」、Customer がファイルを復元するのに必要なだけのシンボルが集まる確率を「復元率」、復元率が 99%を超える時の最小の StorageProvider 数を「StorageProvider 数」と呼ぶ。

表 3 ファイルロスト率と StorageProvider 数の変化

ファイルロスト率	StorageProvider 数
5%	21
10%	31
15%	43
20%	63
25%	97
30%	153

表 3 ファイルロスト率と StorageProvider 数の変化より StorageProvider のファイルロスト率が上がれば上がるほど、99.9999%の復元率を保つために必要な StorageProvider の数が膨らむことが分かった。また、20%を超えると

StorageProvider の必要な数が大幅に増えるため、StorageProvider が 80%以上の確率でファイルを正しく Customer に送ることが可能であることが必要だと考えられる。

９．結論

これにより、非中央集権を利用したクラウドストレージ方式についてよりストレージ提供者の参加ハードルを下げる方式について提案した。さらに、提案方式についてブロックチェーンを用いた分散型アプリケーション開発プラットフォームである Ethereum を用いてスマートコントラクトを作成し、実際の gas の消費量について計測を行い、またストレージ提供者がファイルを失う頻度ごとにほぼ正しくファイルを復元するために必要なストレージ提供者についての算出を行い、提案手法の性能について示すことができた。

謝辞：

本論文を作成するにあたり、ご指導を頂いた指導教員の金井敦教授、呉助教授に心より感謝致します。また、日常の議論を通じて多くの知識やスキルを頂戴いたしました金井研究室の皆様に深く感謝致します。

参考文献

1) ProtocolLabs, Filecoin: A Decentralized Storage Network, 2017.

2) David Vorick, Decentralized Storage, 2014, Sia: Simple Decentralized Storage.