ブロックチェーンを用いた拡張性を持つIPFSネットワーク方式の提案

出本, 拓海 / Izumoto, Takumi

(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)
65
(開始ページ / Start Page)
1
(終了ページ / End Page)
5
(発行年 / Year)
2024-03-24
(URL)
https://doi.org/10.15002/00030731

ブロックチェーンを用いた拡張性を持つ IPFSネットワーク方式の提案

PROPOSED IPFS NETWORK SCHEME WITH BLOCKCHAIN-BASED SCALABILITY

出本 拓海
Takumi Izumoto

法政大学大学院理工学研究科応用情報工学専攻修士課程

Blockchain has been used not only for virtual currencies but also for various other applications in recent years. In addition, InterPlanetary file system (IPFS) has been developed for distributed use in P2P networks. In this study, we propose a scheme that combines both to take advantage of their mer its. We propose a scheme to realize a private IPFS network with scalability by decentralized manage ment of private network membership using blockchain. In addition, we implement and evaluate a prot otype of the proposed scheme. We propose and implement a scheme that takes advantage of the adva ntages of both methods and show that the practicality of the proposed scheme is almost the same as that of IPFS.

Key Words *: Blockchain, IPFS, Ethereum*

１．背景

近年，ブロックチェーン(BC)[1]は従来よく知られている仮想通貨に限らず，他の目的でも利用されてきている[2].BCは複数の端末で互いにデータを検証し合い，トランザクションデータをブロックとして保存することで高いトレーサビリティ，耐改ざん性，透明性を実現でき，データの管理を非中央集権的に行える．これらのことを活かし，特定の管理者を持たず，改ざんが困難であるファイル共有システムが検討されてきた．ファイル共有を行う際に従来は特定の企業が運用する共同デジタルストレージを利用することなどが考えられる．一方BCを用いたファイル共有システムはネットワーク参加者のストレージを利用しデータを分散的に保管，参照することができ，先に述べた特定の企業のサーバにユーザがデータを預ける形のストレージサービスが持つ，サービスの終了によるデータ消失の危険性を回避することが期待できる．現在よく知られているBCを用いたファイル共有システムであるFilecoinの時価総額は現在1986億円[3]と，ファイル共有システムとして利用は認知されてきている．BC を使ってのファイル共有システムはBCが持つ特性を活かすことができるため以下のような特徴を持つ．

- ファイルの登録や取得を行う際のトレーサビリティ，透明性に優れる
- トランザクションデータと同様に耐改ざん性に優れる
- 権限が特定の組織にならない

これらはトラストレスなデータの共有を行えることを意味するため，従来用いられていたような特定の組織内のみでの使用に限らず，同一のシスムを使うもの同士であればファイルの共有なども行うことができる．

しかし,BCをファイル共有システムとして使うには，仮想通貨で行われているようなトランザクションデータではなく大きいファイルサイズのものを扱うため，ネットワーク参加者間での通信量の増加やそれに伴う通信効率の低下といった問題が考えられる[4]．例えば，ビットコインの普及初期はブロックの最大サイズは36MBであったが，ネットワーク上でのスパムやDos攻撃のリスク低減のために1MBまで縮小した．ブロックサイズを拡大する流れは今の所検討されておらず，ファイルといった大きなデータサイズはBCを用いて共有することは現実的ではない．

この問題を解決するため，BCとIPFS(InterPlanetary -FileSystem)[5]を組み合わせたファイル共有システムが検討されてきた．IPFSとはコンテンツ指向のハイパーメディア分散プロトコルで，従来のHTTPとは異なりコンテンツの内容からファイルを取得する．ファイルの断片を分散し,それらをハッシュ値に変換することで似たようなサービスであるWinnyであったような情報漏洩の問題も解消されている．IPFSはファイルの断片を分散し，特定のサーバにデータを集中させないことから耐障害性，負荷分散，耐検閲性，耐改ざん性に優れている．つまり，BC上でのデータサイズを小さくするためにファイル自体ではなくCIDのみをやりとりし，ファイルはIPFSに保存する．

そこで，本研究でもIPFSにおいてファイルを一意に決めるCID(Content Identifiers)のみをBCでやり取りし，ファイル自体はIPFS上で管理するファイル共有システムを提案する．また，既存研究ではIPFSにファイルを保存していたが，IPFSは分散ファイル共有システムであり機密なファイルをやり取りする際に不特定多数のユーザがファイルを閲覧できる可能性があることを考慮し，本研究ではプライベートなIPFSネットワークを構築する機能を持たせる．また，本手法の特徴である高い拡張性を実現するため，ネットワークに新たなユーザを参加させる際にその権限の付与とファイル一つ一つに閲覧できるユーザを指定するためのファイル名称を付与する機能を実装する．

２．関連技術

２．１　ブロックチェーン

ブロックチェーンとは，ネットワーク上にある複数の端末同士が直接P2Pで接続し，取引記録をトランザクションのまとまりであるブロックとして同期する．ブロック

をネットワークに追加する際には前のブロック情報と紐づけてチェーンとする．これにより特定のブロックを変更にはそれ以前のブロック全ての内容を変更する必要があるため高い耐改ざん性を持つ．

また，ブロックチェーンはネットワークに参加している複数の端末がそれぞれの情報の同期を行う分散型台帳という仕組みで管理されている．これは従来の管理者が管理を行う集中型のネットワークにおける問題点であるシステムの停止や故障にも耐性がある．

さらに，先に述べた分散型台帳の仕組みにより全ての取引履歴がネットワーク上の全ての端末に共有される．このことからブロックチェーンは透明性が高いといった特徴も持つ．

２．２　Ethereum

Ethereumはビットコインの有用性の普及後に，今後ビットコイン上で新たなサービスを改発するのか，新しいブロックチェーンの開発を行うのかが議論された．ビットコインを使用してのサービス開発には制限があることが認知されたことから汎用的で新たなブロックチェーンの構築がされた．その後2015年にEthereumの稼働が始まった．

Ethereumはスマートコントラクトと呼ばれるプログラムを実行するグローバルに非中央集権化されたオープンソースの基盤である．ブロックチェーンを使用してシステムの状態変化を保存し，プログラム実行のための資源コストを測定，制限するために暗号通貨であるイーサを使用する．スマートコントラクトとはブロックチェーン上で契約を自動的に実行するものであり，特定の管理者がいなくとも処理が可能である．非中央集権化できること以外にも第三者への手数料がなくなることも利点としてあげられる．

コントラクトによる処理や取引が行われるとトランザクション，そのまとまりであるブロックとして保存されるが，その際にPoW(Proof of Work)と呼ばれるコンセンサスアルゴリズムが用いられる．これによりトランザクションの承認に莫大な計算が必要になり，悪意のあるユーザによるブロックの格納や大量のデータがブロックに格納されることを防いでいる．

２．３　IPFS

IPFSとは分散ファイルシステムにデータを保存するP2Pネットワーク，またはプロトコルである．IPFSの特徴としてファイルの指定方法はコンテンツ指向のプロトコルである．よく知られているHTTPはロケーション指向，つまりファイルの場所を指定して参照を行う．コンテンツ指向はそれと異なりファイルの内容自体を指定して参照を行う点から，内容の改ざんなどに強い特徴を持つ．また，IPFSはファイルを一意に指すCID(Content Identifie-rs)の断片をネットワークに接続しているノードに分散する．つまり，ファイルを取得する際には特定の管理者がおらずとも参加者のノードからCIDを集めるため，権限が偏ってしまうことを防いでいる．さらにこれは負荷分散にも繋がっている．従来のファイル共有システムではサーバなどが停止や攻撃を受けた際にはデータの損失やシステム停止の危険性が考えられる．IPFSではデータの断片が複数ノードに存在しているため参加ノード大部分がオフラインにならなければ問題なくデータの取得が可能である．

３．関連研究

BCと分散ストレージシステムを用いた研究は複数存在する．従来のクラウドストレージシステムでは，暗号技術と属性を用いたアクセス制御を備えるABE方式を利用したものが一般的である．しかし，その技術を備えたシステムを利用しても秘密鍵を生成する権限を持つものはクラウド上に保存された全てのデータを復号できる能力を持つため，機密なデータの漏洩といった深刻な問題は解決できていない．これは先に述べた特定の企業，ユーザに権限が集中してしまっていることが問題であり，BCと分散ストレージシステムを使用してこの問題を解決する手法が研究されてきた[6]．データの所有者はデータそれぞれに対する秘密鍵を生成し，データの利用者から閲覧のリクエストが送られてきた際にその秘密鍵をBCを用いて送付する．これにより権限の集中，データ漏洩のリスクといった問題を解決し，データの保存，閲覧には分散ストレージを利用することでBCのデータサイズの肥大化を避けつつ共有することが可能である．

次に，BCを用いてIPFSのCIDを共有する手法が研究されている[7]．この研究では同様にBCとIPFSの両方を使ったデータを共有するシステムの実装を行なっている．データ登録，削除，閲覧といった機能を有したシステムの実装が主な内容で，このシステムはBCとIPFS相互に通信を行う制御プロセスとして作用する．

また，BCがもつトレーサビリティを活かした研究も存在している[8]．この研究では，BCが持つトレーサビリティを活かして食品のサプライチェーンをユーザが追跡可能にするシステムである．従来のトレーサビリティシステムはデータの可視化が困難である，改ざん，機密情報の漏洩といった問題が存在していた．本研究ではBCの特性からこれらを解決する．物流の情報取得のためにEPC(Electronic Product Code)というあらゆる商品にユニークなIDを付与するものを利用する．EPCにより商品の内容，場所，製造や取引された時間，用途といった詳細な情報を共有することが可能である．これら情報はEPCIS(EPC -information service)と呼ばれるサーバに保存される．BCのデータサイズ肥大化問題は，チェーンを生成する際にオンチェーン，オフチェーン両方を用いて管理することで解決している．

４．提案手法

本研究では，先に述べた問題点を解消するためIPFSとBC両者のメリットを活かした方式を提案する．しかし，機密な文書を扱う際にオープンなIPFSネットワークに保存するのは漏洩の危険性がある．また，IPFSでファイルの参照時にはCIDを使うがランダムな文字列であるためファイル内容の推測が難しい．よって，BCを用いてプライベートIPFSネットワークに参加し，ファイルの名称を指定して共有を行えることを目的とする．ネットワークに参加する際，ユーザが接続要求後に参加権限の有無を確認し，参加権限があれば事前に登録したネットワークに参加するための情報を返す．また，ネットワークに拡張性を持たせるために参加したユーザは参加権限の操作を行える．参加後にユーザはシステムを使用しIPFSにファイルの登録，参照を行える．登録する際にはファイルそれぞれにユーザは名称を指定し，参照する際にファイルの名称を指定することで閲覧する．以下に本システムが持つ機能を示す．

・ 参加権限の登録
・ 参加権限の確認
・ ネットワーク情報の取得
・ ファイル名称の登録
・ ファイル名称の確認
・ CIDの生成
・ CIDの登録，取得

以下に利用者が使える機能を示す．

表1：利用者が使える機能

ネットワークへの参加
参加権限の登録
ファイルと名称の登録
ファイルの取得

５．実証方式

プロトタイプを使っての実証方式を示す．実証内容は，まず本プロトタイプで使用するコントラクトをEthereumネットワークにデプロイする際に使用したGasの計測を行う．次に本プロトタイプを用いてネットワークへの参加とファイルの登録，取得を行うまでの実行時間と消費したガス量の計測を行う．

また実験は様々なファイルの共有が行われることを想定してファイルサイズを1kb, 10kb, 100kb, 1mb, 10mb, 100mbと変更して実験を行う．以下に実験の流れを示す．

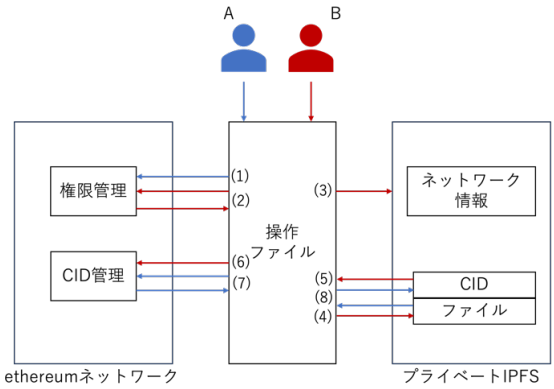

図1：本実証での各コンポーネントの流れ

実証の詳細を以下に示す．

（１）ユーザAはユーザBの参加権限を登録する
（２）ユーザBは接続要求を行い，参加権限を確認する
（３）プライベートIPFSネットワークに接続する
（４）ユーザBはプライベートIPFSネットワークで ファイルの登録を行う
（５）ユーザBはCIDとその名称をコントラクトに登 録する
（６）ユーザAは名称を指定し，対応するファイルのCIDを取得する
（７）ユーザAがファイルを参照する

６．実装

本研究では提案方式の実証のためプロトタイプの実装を行なった．本プロトタイプはユーザがシステムを操作するためのipfs-BCファイルと，Ethereumネットワーク上のスマートコントラクトで，主に権限情報の管理を行うManageAuthコントラクト，CIDの管理や名称の管理を行うManageCIDコントラクトで構成されている．以下に各コンポーネントが持つ機能をより詳細に示す．

・ ipfs-BC(ユーザが操作するファイル)
  - プライベートIPFSネットワークへの接続
  - CIDの生成
  - 各コントラクトの操作
・ ManageAuth
  - 参加権限の登録，確認
  - ネットワーク情報の登録，確認
・ ManageCID
  - 名称の登録，確認
  - CIDの登録，取得

また，本研究はBCが持つ特性を生かし，かつデータサイズの肥大化を抑えたファイル共有システムの提案を目的としている．よってIPFSネットワークの操作は基本的にipfs-BCで行い，生成されたCIDのみをコントラクトに登

録することで特性を活かしつつ肥大化を抑えている．各コンポーネントの関係を以下の図に示す．

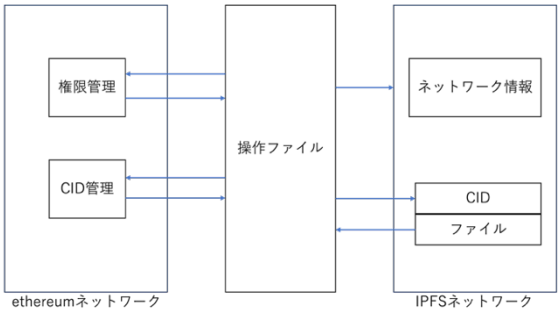

図2：提案手法の各コンポーネントの関係

７．実装

プロトタイプの評価を行う．

７．１ 実験評価

デプロイ時に消費したGas量を以下の表に示す．

表2：デプロイ時のGas消費量

ManageCID	ManageAuth
444955	574303

表1 から二つのコントラクトを比較するとManageAuthコントラクトの方が多くデプロイ時にGasを消費している．次に，様々なデータサイズでの各機能を実行した際の経過時間を以下の表に示す．

表3：経過時間とデータサイズ(KBytes)

機能(ms) ＼ データ(KBytes)	1	10	100
Ethereumと接続	9.9	10.9	10.4
ManageCIDと接続	2.4	2.6	2.3
ManageAuthと接続	0.8	0.8	0.9
setAuth	49.7	46.8	47.0
checkAuth	30.1	29.0	29.4
IPFSと接続	17.1	19.7	20.2
IPFSファイル登録	44.6	44.7	46.4
IPFSファイル取得	40.0	34.0	46.3
合計	194.7	188.7	202.8

表4：経過時間とデータサイズ(MBytes)

機能(ms) ＼ データ(KBytes)	1	10	100
Ethereumと接続	8.7	8.9	8.7
ManageCIDと接続	2.8	2.5	2.3
ManageAuthと接続	0.8	0.8	0.8
setAuth	48.2	50.1	56.9
checkAuth	35.6	34.0	36.1
IPFSと接続	17.7	17.8	17.1
IPFSファイル登録	61.2	113.3	672.1
IPFSファイル取得	74.1	372.9	3123.6
合計	249.2	600.3	3917.6

各機能で差はあるが，合計経過時間はデータサイズが大きくなるにつれて長くなっている．次に，様々なデータサイズでの各機能を実行した際のGas消費量を以下の表に示す．

表5：Gas消費量とデータサイズ(KBytes)

機能(ms) ＼ データ(KBytes)	1	10	100
IPFSファイル登録	31753	31753	37353
IPFSファイル取得	36516	36516	42116
権限登録	34851	34851	34851
権限確認	37243	37243	37243

表6：Gas消費量とデータサイズ(MBytes)

機能(ms) ＼ データ(MBytes)	1	10	100
IPFSファイル登録	37353	31753	37353
IPFSファイル取得	42116	36516	42116
権限登録	34851	34851	34851
権限確認	37243	37243	37243

データサイズによるGas消費量に差はなかったが，各機能で消費量に差がある．

８．考察

評価の結果を元に考察を行う．

８．１　デプロイ時に消費されるGas

表1より，使用する二つのコントラクトを比較するとManageAuthの方がManageCIDよりも消費されるGas 量が多いことが分かった，Ethereumにおける取引の手数料(Gas-Fee)は取引で消費されたガスの総量(GasUsed)とGas1単位あたりの価格(GasPrice)の積で表される．GasPriceはコントラクトの処理に影響せず変動するものなのでシステムの実用性などを測る際にはGasUsedを参考にする．消費されるGasはコントラクトとの接続やデプロイ時，コントラク内の関数呼び出し，変数の変更などにより決まる．二つのコントラクトを比較すると，ManageAuthではユーザの権限情報とそれぞれのノードが持つIDとIPアドレスをコントラクトに保存する．ManageCIDはファイルのCIDとファイルの名称をコントラクトに保存する．それぞれのコントラクトで使用されている関数の数，また権限情報，名称の確認はどちらもキーと値を対応付けて設定するmappingを利用しているため関数による処理の差は少ない．これらのことから，デプロイ時のGas消費量に差があったのはコントラクトに保存する変数の数により生じたものだと考えられる．

８．２　デプロイ時に消費されるGas

表2，表3で本プロトタイプを用いてファイルの共有を行なった際のデータサイズと経過時間の関係をまとめた．データサイズを変更した際に経過時間が変化しなかったものと，大きく変化したものがあった．具体的にはEther-eumとIPFSネットワーク，またそれぞれのコントラクトとの接続と権限情報の設定と確認では変化がなかった．しかし，操作ファイルからIPFSを用いてのファイル登録と名称からCIDの取得，ファイル表示はデータサイズが大きくなればなるほど経過時間も長くなることがわかる．ここで，本プロトタイプの実用性を考えるため，通常のIPFSを使用してのファイル登録とファイル表示にかかる時間を計測した．その結果を図に示す．

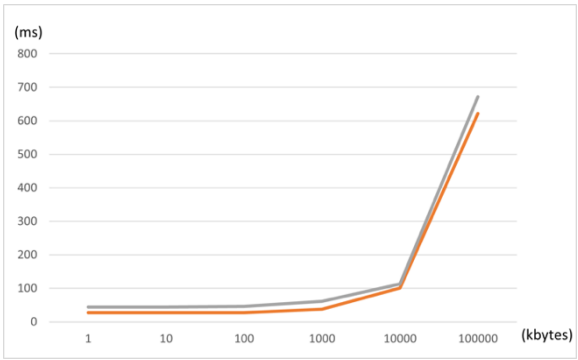

図3：IPFSと本プロトタイプにおけるファイルを登録した際の経過時間

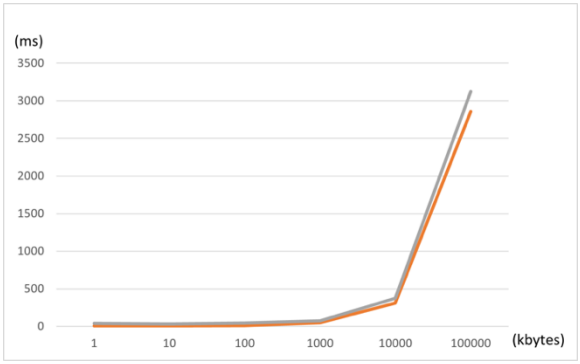

図4: IPFSと本プロトタイプにおけるファイルを取得して表示した際の経過時間

本プロトタイプを使用した際のファイル登録から表示までの時間と，IPFSを使用した際のファイル登録から表示までの時間は図から大きく変わらないことが分かる．よって，実装したプロトタイプで利用する操作ファイルや権限，ファイル名称の処理による経過時間は十分小さくIPFS同様のファイル共有システムとして利用できると考えられる．

８．３　関数呼び出し時のGas消費量

IPFSを用いてのファイル登録とファイル取得，権限の登録と確認にはそれぞれのコントラクト内の関数を呼び出しているためGasが消費された．表4，表5から比較するとどのデータサイズでもIPFSではファイルの登録よりも取得が，権限では登録よりも確認の際に比較的多くのGasが消費されていることが分かった．Gas消費量の推定として考察8.1と同様にコントラクトに保存される変数の数とコントラクトの複雑さ，つまり関数の数や内容を使って考える．それぞれの関数を比較するとどれもコントラクトに保存する変数の数は一つであり，solidityのmapp-ingを利用してキーとその値を保存している．これらのことから，mappingではキーとその値を保存するよりもキーから値を参照する際に多くのGasを消費すると考えられる．

８．４　プロトタイプのセキュリティ

本プロトタイプの特徴として参加ノード全てが新たなノードの参加権限を設定できるため，参加しているユーザと参加していないユーザが結託した際には悪意のあるユーザが参加できてしまう．その際にすでにプライベートIPFSネットワーク上に存在するファイルを取得できればセキュリティの面で大きな危険性を孕んでいることになる．結託した場合は悪意のあるユーザがネットワークに参加してしまうことは避けられないが，通常のオープンなIPFSネットワークと同様に参加後もCIDが分からなければファイルの参照はできない．

しかし，本プロトタイプは名称とファイルを紐づけており名称の設定を行う際にCIDが漏洩する危険性が高まってしまう．ファイルの名称やCIDは本プロトタイプにおけるプライベートIPFSネットワーク内でのみ使用できるものであるため，ネットワークの参加権限を与える際には本人確認のため電子署名といった認証を行う必要がある．

９．結論

本研究では，Ethereumを用いてスマートコントラクトを作成し，それらコントラクトとIPFSを操作するファイルの作成によりBCとIPFS両者のメリットを活かした方式を提案した．また，作成したプロトタイプについて様々な状況での経過時間やGas消費量を計測し，提案手法の性能について示すことができた．今後の課題としては，考察でも述べた通り本プロトタイプではファイルの属性を参加ノード間で共有してなければならない．また，ネットワーク参加前にはそれぞれのIPアドレスの共有とセキュリティ面での対策のための機能が必要である．

謝辞　本論文を作成するにあたり，ご指導を頂いた呉先生や指導教員の方々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] Satoshi Nakamoto: Bitcoin: *A Peer-to-Peer Electronic Cash System.*

[2] PankajDutta: *Blockchain technology in supply chain operations Applications, challenges and research opportunities*, Elsevier(2020).

[3] filecoin.io. ”Use Filecoin to store and retrieve data”. filecoin. 2024. https://filecoin.io/store/sectors,（参照 2022-10-31).

[4] J.Göbel: *Increased block size and Bitcoin blockchain dynamics*, IEEE(2017).

[5] IPFS.tech. ”For Developers: Navigating IPFS”. IPFS. 2024. https://ipfs.tech/developers/,（参照 2022-10-31).

[6] A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems: Shangping Wang, Yinglong Zhang, and Yaling Zhang, IEEE(2018).

[7] Tatsuro Ishida: Implementation and Evaluation of Data Sharing System Combining Blockchain and IPFS, IPSJ(2020).

[8] Food Safety Traceability System Based on Blockchain and EPCIS: Qijun Lin, Huaizhen Wang, Xiaofu Pei, and Junyu Wang, IEEE(2019).