

Hyperledger Iroha を用いたデータ検証可能な分散データベースの実現手法の一検討

堀 遥 (指導教員：小口 正人)

1 はじめに

データ駆動型社会の実現に向け、クライアント端末により収集されたヘテロなデータを、一元的に保存し、その活用を促進する分散データ管理基盤の開発が期待されている。そのようなシステムにおいて、収集・蓄積されたデータの安全性を保証すべく、データ検証プロセスの実装が求められる。

この課題に対し、本稿では検証対象データのメタデータをブロックチェーンで管理し、データの検証可能性を保証するというアプローチを提案する。また、OSS のブロックチェーンプラットフォーム Hyperledger Iroha を用い、データ検証プロセスを備える分散 DB システムの実装手法を検討する。

システムの実証実験として、自動車製造過程におけるカーボンフットプリント管理への応用を想定する。具体的には、自動車を構成する部品の製造・組み立て工程にフォーカスし、各部品の製造時に排出された CO₂ 量を開発システムで管理する。部品間には階層関係があり、部品製造時の CO₂ 排出量を *EMISSIONS* とすると、その部品を構成する全下位部品らの *EMISSIONS* を合算した値は部品製造にあたり排出された CO₂ の総排出量となる。本稿では、これを *TotalEMISSIONS* と呼び、データ検証の対象データとする。

2 関連研究

2.1 ブロックチェーン

2008 年に Satoshi Nakamoto により投稿された論文 [1] により暗号通貨ビットコイン [2] の公開取引台帳としての役割を果たすために発明された。

トランザクションを記録する単位をブロックと呼び、これがチェーンのように連鎖するデータ構造をとる。ブロックチェーンの厳格性を担保する仕組みはチェーンにある。各ブロックに含まれるハッシュ値は、前のブロックのハッシュ値と新規トランザクションの内容などから算出される。すなわち、チェーンとは各ブロック間のハッシュ値による関連であり、これによりブロックチェーンは耐改ざん性に優れるとされる。また、Peer to Peer ネットワーク上の各ノードはブロックチェーンを分散して保持するため、耐障害性と高い可用性を持つ。

2.2 Hyperledger Iroha

Hyperledger Iroha は、The Linux Foundation の Hyperledger プロジェクト [3] にて GA リリースされているパーミッション型ブロックチェーンプラットフォームである。パーミッション型ブロックチェーンは、透明性はないもののプライバシー保護に優れ、大量処理を迅速に行うことが可能であるという特徴を持つブロックチェーンである。Hyperledger Iroha は、簡単な導入、開発の高い自由度と柔軟性、少ないリソースでの高速動作といった特徴を備える。

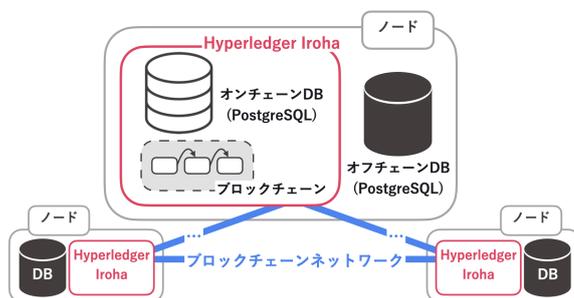


図 1: 提案システム全体の構成

表 1: テーブル Metadata の概要

Metadata	
PK	<u>PartsID</u>
	Link
	ChildPartsID
	TimeStamp

Hyperledger Iroha は起動時にブロックごとのハッシュ値を再計算を行う。もし改ざんが発生した場合どのような挙動を示すか、ブロック欠損時、ブロックチェーン改ざん時について説明する。まずブロック欠損時、起動時のハッシュ値再計算にてブロックの欠損を知らせるエラーメッセージが出力され、Hyperledger Iroha は起動しなくなる。複数 Peer 構成で運用する場合は、自動的に他の Peer から欠損ブロックが読み込まれる。次にブロックチェーン改ざん時、ブロック中のトランザクション処理内容が改ざんされた場合は、ハッシュ値に変化が生じる。そのため、他の Peer との比較により改ざんが検知される。また、トランザクションの制作者が改ざんされた場合には、欠損時同様に Hyperledger Iroha は起動せず、改ざんが検知される。

3 提案手法

図 1 には本研究の提案システムの構成を示した。各ノードは Hyperledger Iroha に加え、オフチェーンデータベースを持つ。検証対象データを含む実データはオフチェーンデータベースで保管し、検証対象データのメタデータを Hyperledger Iroha 上で管理する。

表 1 にはメタデータを格納するテーブル Metadata の概要を示した。メタデータは各部品につき 1 つ保管する。主キーである PartsID は部品の ID、Link は *TotalEMISSIONS* 格納場所、ChildPartsID は下位部品の PartsID、TimeStamp は時刻を表す。ブロックチェーンシステムはパフォーマンスとサイズのコストが懸念点として挙げられるが、この課題に対しメタデータは有効である。その設計次第で実データに比べサイズを抑えることが可能であり、また、本システムはヘテロなデータを一元的に扱うことを想定するため、その精度や形式を考慮する必要がなくなる。

提案システムにおけるデータ格納時の手順は以下の

表 2: ブロックの構成

前のブロックのハッシュ値
データ更新トランザクションの内容
メタデータのリンク (PartsID)
タイムスタンプ
ハッシュ値
電子署名

通りである。

- 1) データ更新トランザクションのトリガー.
- 2) データ更新トランザクションの実行.
 - 2-1) 必要に応じたデータの加工処理実施.
 - 2-2) オフチェーンデータベースに格納.
- 3) 2-2) の格納場所を含めて新規メタデータ作成.
- 4) メタデータをテーブル Metadata に格納.
- 5) ブロック作成, 追加.

また, データ参照時の手順は次の通りである。

- 1) メタデータを参照し, Link を取得.
- 2) 1) からオフチェーンデータベースにアクセスし, データを参照.

データ更新トランザクションでは, 検証対象データの更新に伴い必要となる一連のデータ加工処理や演算などを行い, これをオフチェーンデータベースに格納する. 実装は Hyperledger Iroha のコマンドの開発で行う. 次はカーボンフットプリント応用におけるデータ更新トランザクションの例である. ここで本稿では, 新規 EMISSIONS の取得をデータ更新トランザクションのトリガー条件とする.

- 1-1) テーブル Metadata を参照し, 対象部品の下位部品の PartsID を取得.
- 1-2) テーブル Metadata で 1-1) の PartsID を検索し, Link を取得.
- 1-3) Link からオフチェーンデータベース上の下位部品の TotalEMISSIONS を取得. これを対象部品の全下位部品で行う.
- 1-4) 1-3) で取得した TotalEMISSIONS を合算する.
- 1-5) 1-4) で算出した値に対象部品の新規 EMISSIONS を加算し, これが新規 TotalEMISSIONS となる.
- 2) 1-5) をオフチェーンデータベースに格納.

表 2 には本研究におけるブロックの構成を示した. メタデータのリンクとはテーブル Metadata における格納場所の情報であり, 本稿では PartsID を指定する. ブロックのハッシュ値の算出はブロック中の上 4 要素から行う.

表 3: 実装環境の概要

	Hyperledger Iroha	PostgreSQL
コンテナ	Ubuntu22.04LTS イメージ: ubuntu:22.04	
仮想環境	Docker	
OS	Ubuntu20.04LTS	
サーバ	CPU: Intel(R) Xeon(R) Silver 4314 CPU @ 2.40GHz メモリ: 192GB	

以上の手順により, ブロックチェーンの特性からメタデータの安全性は保証される. このメタデータと蓄積されたブロックの情報を元に, Hyperledger Iroha のブロック検証メカニズムを利用して, 検証対象データの検証可能性を保証する方針である.

4 提案手法の実装計画と進捗

提案手法実装のため, まずはデータ検証トランザクションの実装に取り掛かる. 具体的には, データ更新トランザクションの処理を提供する Hyperledger Iroha のコマンドの作成である. そのために Hyperledger Iroha の構成ファイルの開発が必要となる.

表 3 は実装環境の詳細である. 開発した構成ファイルで Hyperledger Iroha のビルドを行うため, 既存の Docker イメージを利用した導入はできない. ビルドには Ubuntu22.04LTS が求められるため, Docker を使用して簡易的に Ubuntu22.04LTS の環境を用意した. PostgreSQL は World State View と呼ばれ, Hyperledger Iroha の最新情報を保管するオンチェーンデータベースとしての機能を提供する. 表 1 に示したテーブル Metadata は PostgreSQL で動作するデータベース上に存在する.

5 まとめと今後の課題

本稿では, Hyperledger Iroha を用いたデータ検証プロセスを叶えるための基盤システムおよびメタデータを設計した. 今後は Ubuntu サーバ上に設計システムを構築していく. 第一にデータ更新トランザクションの実装を行うべく, Hyperledger Iroha のシステムプログラムに追記する形で新規コマンドの開発を行う予定である.

謝辞

本研究は一部, JST CREST JPMJCR22M2 の支援を受けたものである.

参考文献

- [1] Japan Blockchain Association. ブロックチェーンの定義. <https://jba-web.jp/news/642>.
- [2] Bitcoin. <https://bitcoin.org/ja/>.
- [3] Hyperledger foundation. <https://www.hyperledger.org/>.