ブロックチェーンベースのカーボンフットプリントデータ管理基盤の構築と自動車部品製造業への応用における評価

堀　　遥† Le Hieu Hanh† 小口　正人†

† お茶の水女子大学　〒 112–8610 東京都文京区大塚 2–1–1
E-mail: † haruka-h@ogl.is.ocha.ac.jp, {le,oguchi}@is.ocha.ac.jp

あらまし　脱炭素社会実現に向け，複数企業を横断する分散データベースシステムでトレーサビリティを表現し，カーボンフットプリントを管理するようなデータ管理基盤の実現が期待される．我々は自動車部品製造業におけるカーボンフットプリントの管理を題材に，部品ごとの製造時 CO_2 排出量の可視化，およびデータの改ざんを検知し，安全性を担保することを目的とした検証機能の設計と実装を試みた．一方でこのようなシステムでは，検証側と各ノードにおけるデータ量増加によるパフォーマンスの低下が懸念される．特にこの応用例の場合，各部品は階層関係にあるため，検証時には最下位部品まで再帰的に遡って検証していく必要がある．この課題に対し，我々は検証結果の一部をブロックチェーンに記録しておき，再検証の際にこれを活用することで再帰処理を省略するような簡易化した検証プロセスを検討した．提案手法の評価実験では，2 つの検証プロセス，従来手法と提案手法の実行時間を調査した．従来手法に比べ提案手法は最大で 1/6 程度まで抑えられ，高速であることが示された．一方で，提案手法の検証可能範囲は従来手法に比べて限定的である事から，実行時間と検証精度はトレードオフとなった．

キーワード　分散データベース、ブロックチェーン

1 はじめに

1.1 研究背景

2050 年までにカーボンニュートラル達成という目標を表明する 150 以上の国々において，多様な取り組みが推進されており，その一つにカーボンフットプリントがあげられる．カーボンフットプリントとは，製品のライフサイクルの各過程で直接的・間接的に排出された温室効果ガスを CO2 排出量に換算し，製品単位で表示する仕組みである．そして現在，サプラーチェーン全体でカーボンフットプリントを管理して，製品単体のカーボンフットプリントが可視化できるような分散データ基盤の開発が期待されている．このようなデータ基盤実現のメリットととして，一点目にホットスポットの特定より，効率的な削減につながること，二点目に可視化により低炭素・脱炭素製品が積極的に選ばれること，三点目にサプライチェーン上の他事業者による排出削減が自社の削減とみなされるため，各社の削減の可能性が広がることが挙げられる．

CREST にて進行中のプロジェクト「検証可能なデータエコシステム」[1] では，データに付随する信頼度や来歴を第一級のデータとしてサポートし，任意のデータが検証可能であるようなデータエコシステムの研究開発を目指しており，実証実験として自動車製造業におけるカーボンフットプリント管理に取り組んでいる．我々は中でも，データエコシステムの基盤となる分散データベースシステムに，安全性を提供する役割を担当する．

1.2 データ連携基盤に関連する各国の動向

サプライチェーンを横断して連携するようなデータ基盤の構築により，社会課題や経済課題の解決を目指す動きは，各国で推進されている．例えばドイツでは，Catena–X [2] が自動車業界を中心とした企業向けに安全で標準化された企業間データ連携サービスを提供している．欧州では 2023 年 8 月に施行された欧州バッテリー規則にて，バッテリーのライフサイクル全体にわたる CO2 排出量や資源リサイクル率の欧州委員会への開示が求められており，Catena–X は規則への対応を支援している．さらにアメリカでは，MOBI [3] がブロックチェーン・DLT を基盤としたデータ連携基盤で自動車産業に挙げられる 6 つの課題の解決を目指す．6 つの各テーマで標準デジタルインフラの企画・開発が推進されている．

一方日本では，経済産業省が独立行政法人情報処理推進機構とともに Ouranos Ecosystem [4] [5] プロジェクトが進む．Ouranos Ecosystem のもとでの業種横断的なシステム連携の実現を目指し，人流・物流 DX 及び商流・金流 DX に先行的に着手している．本システムの基盤にはブロックチェーンが用いられており，スマートコントラクト技術によるデータ主権の確保を実現している．

1.3 研究目的

自動車製造業のカーボンフットプリント管理というシナリオにおいて，以下の三つの課題が挙げられる．一点目は複数の異なる企業が分散システムに参加するため，異なる企業間の連携の信頼性を担保する必要がある．二点目に改ざんは外部攻撃だけでなく，自社内においても発生する可能性がある．近年，自動車の品質不正の事例も多く，カーボンフットプリントも例外ではない事から，改ざん検知機能の実現が求められる．三点目は部品は複数の部品を組み立てて作られるため，階層的な構成をしており，製品単位のカーボンフットプリント値の計算には

再帰的な探索が必要となる．いずれの課題も自動車製造業において提案システムを動作させる上で重要であり，課題解決のアプローチを検討することが求められる．

本研究では，異なる企業間の連携の信頼性を担保とデータ改ざんの検知という課題に対し，Peer to Peer ネットワーク上でのデータ検証機能の導入を検討する．さらに，製品単位のカーボンフットプリント値に計算コストを要するという課題に対しては，一度算出したカーボンフットプリントの値を保管しておき，検証時に利用することで再帰計算の省略を狙う．本アプローチ実装にあたり，実世界で進められているカーボンフットプリント管理基盤を参考に，ブロックチェーンプラットフォーム Hyperledger Iroha を基盤とする．自動車部品製造業におけるカーボンフットプリント管理システムを目指し，カーボンフットプリントデータを検証可能な形で保管するプロセスと効率的なデータ検証機能を提案する．

我々はこれまでに，スマートコントラクト技術の組み込みコマンドを用いた分散システム上でのデータ検証機能とその効率化手法を実装してきた [6]．データ検証プロセスの評価実験では期待される動作が得られたが，処理のオーバーヘッドが高さが課題であった．また，更なるデータ検証プロセスの信頼性評価が不可欠であった．

本稿では，プロセスの実装そのものや組み込みコマンドの内容を見直し，更なる高速化を図る．さらに，提案手法の信頼性の評価のため，改ざん検知範囲と検知速度の調査を行う．この際には，実社会でのサプライチェーンは様々であることが想定されることを考慮し，部品同士の関係について条件をつけて複数生成し，プロセスの実行時間を調査する．

1.4 本稿の構成

本稿は以下の通り構成される．第 2 節では本研究の前提となるブロックチェーン及び Hyperledger Iroha の概要を説明する．第 3 節では，自動車部品製造業における製品単体のカーボンフットプリントデータを集約・算定・検証する手法を提案する．第 4 節では，提案手法の有効性を確認するため実装システムの評価実験とその結果を述べる．第 5 節では第 4 節の結果を踏まえたシステムの考察を行い，第 6 節でまとめる．

2 関連知識と関連研究

本節では，提案手法の基盤となるブロックチェーンの技術的概要を説明したのちに，本稿で採用したブロックチェーンプラットフォーム Hyperledger Iroha を紹介する．最後に，ブロックチェーンベースのデータ管理基盤に関する関連研究を取り上げる．

2.1 ブロックチェーン

ブロックチェーンは，ブロックと呼ばれるトランザクションを記録する単位を生成し，これをチェーンのように連鎖するデータ構造である．2008 年に Satoshi Nakamoto により投稿された論文 [7] に基づき，暗号通貨 Bitcoin [8] 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．仮想通貨の基礎技術としての厳格性を担保する仕組みはチェーンにある．一つのブロックにはハッシュ値が付与される．このハッシュ値は，前のブロックのハッシュ値と新規のトランザクションの内容やタイムスタンプなどから算出される．すなわち，ハッシュ値によりブロックの関連が生まれ，これがチェーンとなる．よってブロックチェーンは高い改ざん耐性を持つと言われる．また，Peer to Peer 型のブロックチェーンネットワーク上に参加する各ノードがブロックチェーンを管理するるため，欠損ブロックを他のノードから補うことができ，耐障害性と可用性に優れるといった特徴を持つ．

Etherum [9] に代表される一部のブロックチェーンには，スマートコントラクトが実装されている．スマートコントラクトとは，ブロックチェーン上で事前に指定されたルールを満たすと，自動的に契約が締結される仕組みである．仲介者を必要とせず，あらかじめプログラミングされた契約条件や内容をもとに実行されるため，非常に効率的であるほか，高い透明性と耐改ざん性を持つ．また，スマートコントラクトの内容はブロックチェーン上に記録される．本研究で使用する Hyperledger Iroha では組み込みコマンドとしてスマートコントラクトが導入されており，API を通して実行される．

2.2 Hyperledger Iroha

Hyperledger Iroha [10](以下，Iroha という) は 2019 年 5 月に Hyperledger [11] により GA リリースされた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，透明性には欠ける一方で、中央集権型ネットワークでプライバシーが確保され、マイニング不要のため高いパフォーマンスを発揮する．こうした特徴から，パーミッション型は単一の企業や組織内での運用に有効であり，特に銀行間の取引や証券取引での活用が促進されている．

他のパーミッション型ブロックチェーンと比較した Iroha の特徴として以下の 3 点があげられる．

- YAC コンセンサスアルゴリズムを採用
- 組み込みコマンドによるスマートコントラクトの実現
- 汎用的で柔軟な権限機構の実装

YAC アルゴリズムは，BFT：ビザンチン耐性を備えており，高速かつ低遅延である．

Iroha のブロックチェーンネットワークを Iroha Network といい，Iroha Network に参加するノードは Iroha ノードという．各 Iroha ノードは Iroha のサービスを提供する IROHA と，Iroha の最新情報を保持するオンチェーンデータベース，ブロックチェーンで構成される．

2.3 関 連 研 究

高い機密性が求められるデータ管理システムにおけるブロックチェーンの技術的な可能性について，これまでに幾つかの論文が述べられている [12–15]．管理データのハッシュ値やポインタ，アクセス権などを含めたメタデータでブロックを構成し，ブロックチェーンに追加することで，安全性の担保されたメタデータを用いたデータ管理を実現している．

中でも，Neha Mishra らの研究 [12] では，PDV：Personal Data Vault と呼ばれる個人の生涯にわたるデジタルドキュメントを，検証可能かつ安全な方法で保管，保存，保護，共有するためのフレームワークの開発し，これにブロックチェーンプラットフォーム Hyperledgr Iroha を採用している．各文書は暗号化，圧縮され，クラウドに安全に保存され，文書の保存先 URL がメタデータとして Hyperledger Iroha に入力されるほか，システムに書類を追加する際のユーザ認証などにスマートコントラクトを利用している．また，開発システムは Markov Tree を用いた予測プリフェッチ機能を備えており，次に発生する要求を予測，事前実行する．

以上の関連研究と 1.3 節で消化した事例より，業種を横断しながら安心安全なデータ流通を叶えるためにはブロックチェーンは有効であると考えられる．

3　提案システム

提案システムは自動車部品製造業におけるカーボンフットプリントをサプライチェーン全体で算定・管理を行う．本システム算定した製品単体のカーボンフットプリントは，第三者機関による評価や削減シミュレーションに利用され，サプライチェーン全体での CO_2 削減をサポートする．

本節では，まず本稿における自動車部品製造業のカーボンフットプリントの定義を 3.1 節で紹介し，以降で提案システムの詳細な設計を述べる．

3.1　カーボンフットプリントの定義

図 1 は，本稿で定めた自動車部品製造業におけるスキーマであり，供給元 (SUPPLIER) から子部品 (PART) が供給され，組み立て工場 (ASSEMBLER) による組み立て工程 (ASSEMBLE PROCESS) を経て，親部品 (PARENTS PART) を構築する．子部品 (PART) から親部品 (PARENTS PART) を階層的に組み立てて，最上位部品の自動車を製造する．

ここで，スキーマに沿って部品 X が製造された時の CO_2 排出量を *CFP(X)* とする．さらに，スキーマを重ね合わせて最下位の子部品までサプライチェーンを遡ることで，部品 X のカーボンフットプリントを表すことができ，これを *TotalCFP(X)* とする．例を図 2 を用いて説明する．これは，部品 P_A を構成する子部品らの関係を階層的に表したものである．本稿ではこれを P_A の部品木と呼ぶ．親部品 P_A は子部品 $P_B, P_C, \cdots, P_N$ で構成され，親部品 P_B は子部品 $P_S, P_T, \cdots, P_X$ で構成... という関係がある．この時，*CFP(P_A)* は，$P_B, P_C, \cdots, P_N$ が供給，組み立てされる P_A の製造工場で発生した CO_2 排出量である．*TotalCFP(P_A)* は，サプライチェーンを遡って求めるため，部品木に属する全部品の *CFP* の総和となる．

提案システムでは，自動車構成部品の部品単体のカーボンフットプリントである *TotalCFP* を検証可能な形で保管し，データの信頼性を担保する．そして，企業・ユーザにとって安全なカーボンフットプリント管理への貢献を目指す．

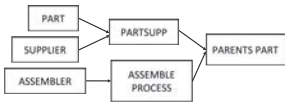


図 1: カーボンフットプリント応用のスキーマ

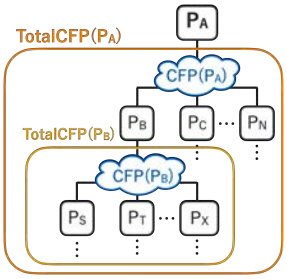


図 2: 部品木と定義

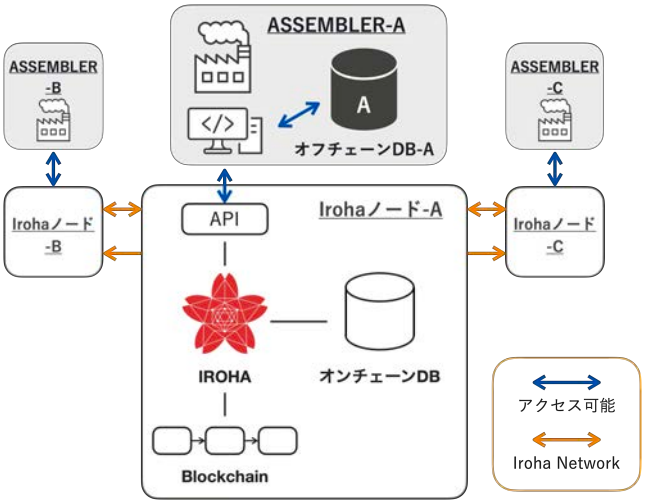


図 3: 提案システム全体像

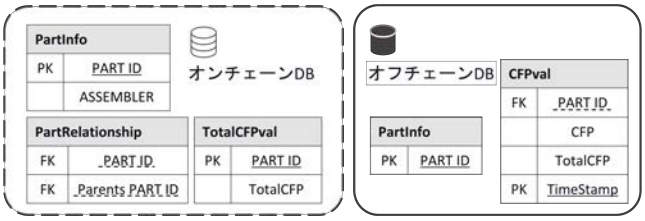


図 4: DB テーブル

3.2　提案システム概要

図 3 は提案システムの全体像である．サプライチェーンに属する ASSEMBLER はローカル環境にオフチェーン DB を持つ．そして，ASSEMBLER はそれぞれ Iroha ノードを持っており，これらは他 ASSEMBLER の Iroha ノードと Iroha Network で接続される．各社は Iroha Network を介して連携を行うため，ASSEMBLER のローカル環境で直接的な連携はなく，他社のオフチェーン DB へのアクセス権限は持たない．

図 4 にはオフチェーン DB とオンチェーン DB 上のテーブルを示した．図中には，テーブル名，属性名，キー設定を含めており，PK は主キー (Primary Key)，FK は外部キー (Foreign Key) を表す．オフチェーン DB は各 ASSEMBLER 専用の DB であり，そこに保管されるデータは ASSEMBLER 内で製造する部品に関するものに限定される．例えば，部品 $P_{A1}, P_{A2}, \cdots, P_{An}$ を製造する ASSEMBLER-A では，オフチェーン DB-A にだけ部品 $P_{A1}, P_{A2}, \cdots, P_{An}$ の *TotaalCFP, CFP* データが格納される．他 ASSEMBLER で製造される部品の *TotaalCFP, CFP*

データは持たない．一方でオンチェーン DB は全 Iroha ノードで同期されるものであり，全部品のデータが集約される．特に，テーブル：PartRelationship は部品の親子関係を格納している．またテーブル：TotalCFPval には，組み込みコマンドによって算出された全部品の最新 *TotalCFP* が保管される．

提案システムは 2 つのメインプロセスを持つ．一つ目は，*TotalCFP* 算出プロセスである．各部品の *TotalCFP* を検証可能な形で保管するためのプロセスである．二つ目は，*TotalCFP* 検証プロセスである．本研究の挑戦の一つである，Peer to Peer 上でのデータ検証を行うプロセスで，検証可能範囲の異なる 2 つの手法を提案する．次の節でそれぞれのプロセスの詳細を述べる．

3.3 *TotalCFP* 算出プロセス

TotalCFP 算出プロセスの流れを図 5 と以下に示す．例として，ASSEMBLER-A で製造される部品 P_A の *TotalCFP*(P_A) を算出する．

(1) **集約**：サプライチェーンに参加する全 ASSEMBLER のオフチェーン DB から全部品の *CFP* を一時的に集約する．
(2) **CTE 再帰クエリ実行**：CTE 再帰クエリで新規 *TotalCFP* を算出する．
(3) **BC に追加**：組み込みコマンドの実行結果を含めたブロックをブロックチェーンに追加する．
(4) **DB 更新**：新規 *TotalCFP* で ASSEMBLER-A のオフチェーン DB と全オンチェーン DB を更新する．

TotalCFP 算出プロセスは Iroha の組み込みコマンドで処理が実行される．組み込みコマンドの実行条件は，任意部品の新規 *CFP* の取得や *TotalCFP* の再算出が要求された場合である．組み込みコマンドで実行することにより，オンチェーン DB 上に新規 *TotalCFP* が記録されるが，オンチェーン DB 上のデータは耐改ざん性に優れ，高い信頼性を持つため，データ検証機能に活用される．

本プロセスは，システムに部品を新たに登録する場合や，各 ASSEMBLER の *CFP* が更新された場合，*TotalCFP* 検証プロセス内で実行される．

3.4 *TotalCFP* 検証プロセス

プロセスの流れを図 6 と以下に示す．ASSEMBLER-A で製造される部品 P_A の *TotalCFP*(P_A) を検証する．

(1) **既存値取得**：ASSEMBLER-A のオフチェーン DB-A から既に保管されている *TotalCFP*(P_A) の最新値を取得する．
(2) **再算出**：*TotalCFP*(P_A) を再算出する．
(3) **比較**：(1) 既存の *TotalCFP*(P_A) と (2) 新規 *TotalCFP*(P_A) を比較する．一致した場合は検証成功であり，ユーザは信頼性の担保されたデータ利用が可能である．一方，不一致の場合は検証失敗であり，データは改ざんデータまたは旧データである可能性がある．これをユーザに告知し，ユーザに利用の有無の判断を委ねることができる．

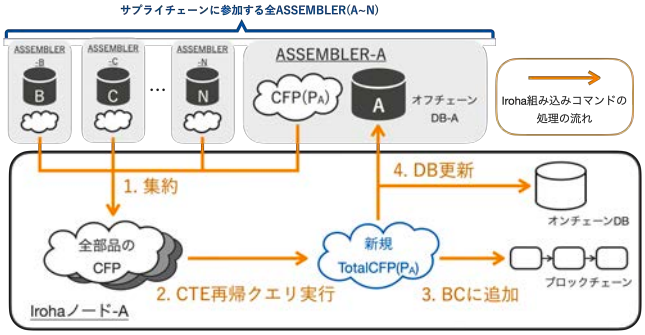


図 5: *TotalCFP* 算出プロセス

図 6: *TotalCFP* 検証プロセス

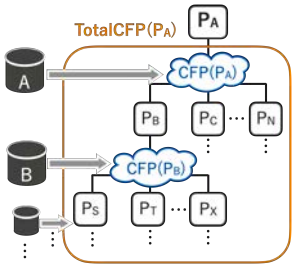


図 7: Fine-grained 手法

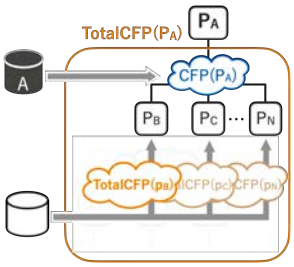


図 8: Coarse-grained 手法

さらに，本稿では検証可能範囲の異なる 2 つのデータ検証として，Fine-grained 手法と Coarse-grained 手法を提案する．これらの違いはプロセスの「(2) 再算出 」で新規 *TotalCFP* を算出する際に注目する部品にある．次の項で詳細を説明する．図 7 と図 8 には部品 P_A の部品木におけるそれぞれの概要を示した．

3.4.1 Fine-grained 手法

Fine-grained 手法は，検証対象の部品を構成する全ての部品，すなわち部品木全体に注目することで，高い粒度で検証を行う．図 7 に部品 P_A の部品木における概要を示した．実装は，*TotalCFP* 算出プロセス通りであり，全部品分の *CFP* を集約して CTE 再帰問い合わせをすることにより新規 *TotalCFP* を算出する．この時，WITH 句の仮想テーブルはメモリに乗るほか，子部品の数がデータ量に影響するため，検証側に負担がかかるという問題点を持つ．

3.4.2 Coarse-grained 手法

Coarse-grained 手法では，部品木で検証対象の部品の直下に位置する子部品のみに注目し，低い粒度で検証を行う．図 8 に部品 P_A の部品木におけるそれぞれの概要を示した．実装の具

体的な手順は以下の通りである．

(2) **再算出**:

(2)-i オフチェーン DB-A から $CFP(P_A)$ を取得する．

(2)-ii 親部品が P_A である子部品，P_B，P_C，$\cdots P_N$ の *TotalCFP* をオンチェーン DB から集約する．

(2)-iii (2)-i の総和と $CFP(P_A)$ を加算する．

本プロセスは，*TotalCFP* 算出プロセスを経てオンチェーン DB が構築されていることが前提に実行できる．

Fine-grained 手法の *TotalCFP* 算出プロセスと比較すると，まず，接続するオフチェーン DB の数が 1 つだけに減少する．また，CTE 再帰クエリによる再帰計算がなくなり，加算処理が一度だけとなる．一連の処理は組み込みコマンドを使用しないため，Iroha 上での処理もない．主にこれらの変更点を踏まえて，よりライトで高速な処理の実現を図った．

4 実　　験

本節では，提案手法の性能を調査する．

4.1 実験概要

提案する 2 つのメインプロセスについて，*TotalCFP* 算出プロセスは C++で記述した組み込みコマンドで，*TotalCFP* 検証プロセスは Python プログラムで実装した．提案手法の性能を調査すべく，*TotalCFP* 検証プロセスの Fine-grained 手法と Coarse-grained 手法をそれぞれ実行し，処理時間を計測した．複数の部品木を制約のもとで生成し，根の親部品の *TotalCFP* を検証する．

ここで，Coarse-grained 手法を実行するために，*TotalCFP* 算出プロセスを実行して全部品の *TotalCFP* がオンチェーン DB に登録している．この際に，*TotalCFP* 算出プロセスの動作確認を行い，正しい計算結果と期待される動作を確認した．

4.2 実験設定

実装の構築環境とマシンの性能は図 9 に示した．より実運用に近い環境を求める場合は実ネットワークでの評価が妥当であるが，Iroha ネットワークの性能は実ネットワークと仮想ネットワークで大きな差が見られないため [16]，本稿では Docker 環境を採用している．また，参加する ASSEMBLER ごとに Iroha をビルドする Ubuntu 22.04 LTS コンテナとオンチェーン/オフチェーン DB の機能を提供する PostgreSQL コンテナをセットで構築する．本実験では，システムに参加する ASSEMBLER 数を 3 社とした．

検証対象となる部品木は，総部品数，部品木の高さ，1 ノードの子部品数，など条件を設定して複数パターン生成する．また，部品木生成には以下の制約を設けた．

- 5 分木
- 子部品を持つ場合は 2 つ以上 5 つ以下
- 部品木中の部品は重複しない

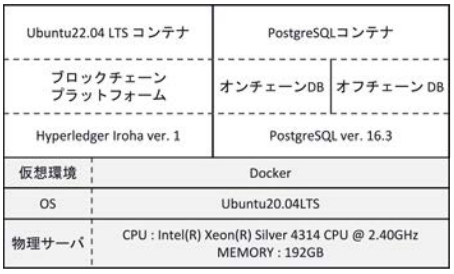


図 9: 構築環境

表 1: 第 4.3 節 実験条件

高さ	2	3	4	5	6	7
総部品数	5	30	155	780	3,905	19,530

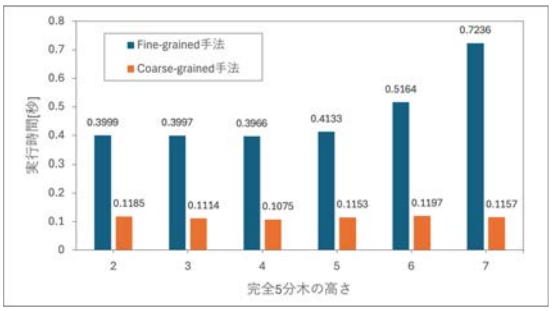


図 10: 完全 5 分木における実行時間

4.3 完全木の比較

完全 5 分木で表される 6 つの部品木を生成した．表 1 に示すように，総部品数は $\sum_{i=1}^{N} 5^i$, $\{N = 1, \cdots, 6\}$となる．各部品木を対象に 2 つのデータ検証の実行時間を 10 回測定し，1 回の平均を求めた．結果を図 10 に示す．

まず Fine-grained 手法であるが，高さ 2 から 4 までは実行時間は横ばいであるが，以降は増加傾向にある．総部品数は増加，高さは高くなり，単純に木のサイズと扱うデータ量は多くなるため，処処理時間の増加は妥当と考えられる．次に Coarse-grained 手法では，部品木のサイズに関わらず，0.11-0.12 秒程度となった．その差は最大となる高さ 7 にて約 1/6 であり，非常に高速である事が示された．

4.4 総部品数一定での比較

総部品数が 19,530 個という条件と第 4.2 節で述べた制約のもと，ランダムに部品木を 10 個生成した．第 4.3 節と同様に，各部品木を対象に 2 つのデータ検証の実行時間を 10 回測定し，1 回の平均を求めた．高さ 7 の完全 5 分木は総部品数が 19,530 個であるため，ランダム生成の結果と併せて図 11 に示した．図 11(a) が Fine-grained 手法，図 11(b) が Coarse-grained 手法である．

まず Fine-grained 手法であるが，完全 5 分木の結果と合わせてみても，木の高さに対して実行時間が増加している．線形近似では $y = 0.0113x + 0.643$ となった．これは *TotalCFP* 算出プロセスの「(2) CTE 再帰クエリ実行」にて再帰探索する回数が増えるため，処理時間もまた，増加していると考えられる．

一方で Coarse-grained 手法においては，第 4.3 節の結果と同様に，高さに関わらず 0.11-0.13 秒程度で安定している．

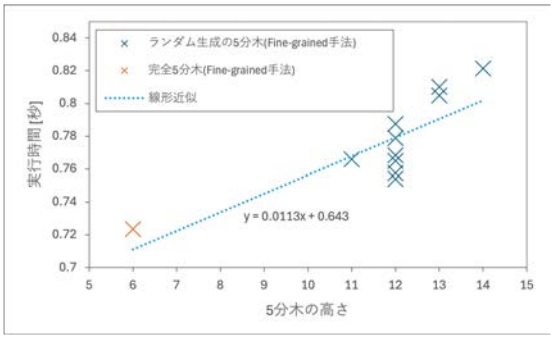
5 考　　察

本節では，提案システムの実装についての考察を行う．

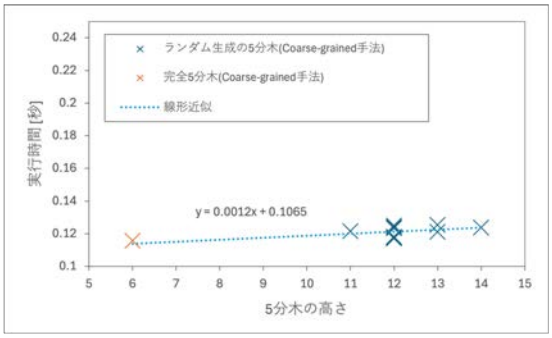
まず，第 4 節の評価実験の結果から議論を行う．2 つの評価実験より，第一に Fine-grained 手法は部品木のサイズ (総ノード数と高さ) に影響を受けること，第二に Coarse-grained 手法は安定的に高速な処理であるという結論が得られた．第 4.3 節で行った完全 5 分木で比較実験の結果について，その内訳を調査し，図 12 に表示した．*TotalCFP* 算出プロセスの (1)，(2) に該当する部分を算出過程，*TotalCFP* 算出プロセスの組み込みコマンドに関連するその他の Iroha の処理を Iroha 処理，*TotalCFP* 検証プロセスの (3) に該当する部分を検証過程に分類した．Fine-grained 手法では高さ 2 から 4 で実行時間はほぼ横ばいになると述べたが，図 12 の通りその内訳も横ばいであり，現在の実装では最低でも 0.3 秒程度の時間がかかるとわかる．また，算出過程は高さ 2 から徐々に増加しているが，高さ 4 と 5 でもその差は 0.011 秒程度であることから，*TotalCFP* 算出プロセスの (1) 集約がその大部分を占めていると考えられる．この部分の実装は，各オフチェーン DB に PostgreSQL の DBlink を利用して接続して行っている．本稿での実験では ASSEMBLER 数を 3 つとしているが，実運用では 100 社以上であると想定されるため，この過程のオーバーヘッド増大が懸念される．

次に，先行研究 [6] との比較を行う．先行研究でも 2 つのデータ検証手法を提案している．先行研究で提案した Naive 手法は従来手法であり，本稿の Fine-grained 手法に該当する．また，先行研究で提案した Simplified 手法は簡易化手法であり，本稿の Coarse-grained 手法に該当する．先行研究からの実装の主な各手法ごとにあり，Fine-grained 手法では先行研究の Naive 手法の組み込みコマンドのプログラムの最適化を行っている．そして Coarse-grained 手法であるが，先行研究の Simplified 手法を組み込みコマンドで実装していたところを，Hyperledger Iroha を返さずアプリケーション上の処理のみで実装し直した．これにより，コンセンサス形成といった Hyperledger Iroha 上での処理を省略でき高速化を図れるほか，ブロックの生成頻度も減らすことができ，システムの負荷を軽減することができる．図 13 には，性能比較のため 4.3 節で行った実験を同様に先行研究の手法でも実施し，4 つのデータ検証手法の実験結果を一つのグラフに示した．本稿で提案した手法はいずれも先行研究より高速に改善されていることが確認された．特に Fine-grained 手法と Naive 手法の比較では実行時間の増加傾向より，データ検証処理のオーバヘッドそのものを改善できたとわかる．

最後に，2 つのデータ検証手法の改ざん検知範囲について議論する．2 つのデータ検証手法は検証可能範囲が異なり，Fine-grained 手法は部品木全体，Coarse-grained 手法は直下の子部品までである．Fine-grained 手法では，検証対象の部品を構成する全子部品の *CFP* の値を参照して新規 *TotalCFP* を得る．



(a) Fine-grained 手法



(b) Coarse-grained 手法

図 11: 総部品数:19530 の部品木における実行時間

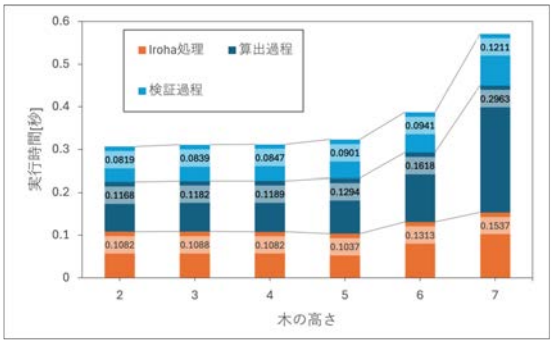


図 12: Fine-grained 手法 実行時間の内訳

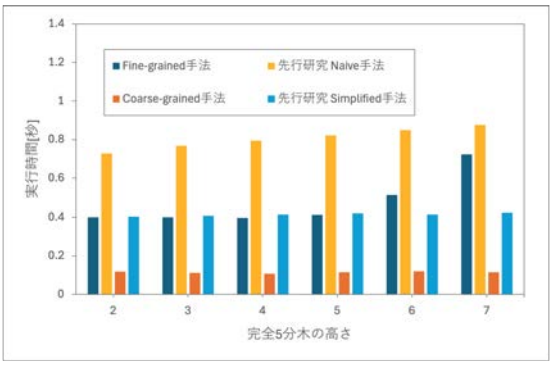


図 13: 先行研究との比較

そのため，各オフチェーン DB において，検証対象部品の既存 *TotalCFP* や子部品の *CFP* らに改ざんが発生していた場合，既存 *TotalCFP* と新規 *TotalCFP* が一致せず，改ざんが検知される．これに対する Coarse-grained 手法では，オンチェーン DB から検証対象部品の直下の子部品の *TotalCFP* を取得し

て，新規 *TotalCFP* を得る．よって，オフチェーン DB において検証対象部品の既存の *TotalCFP*，*CFP* に改ざんが発生していた場合は改ざんが検知されるが，検証対象部品の部品木の子部品らの改ざんは検知することができない．

一方で Fine-grained 手法で正しい検証結果が得られない場合もある．ある ASSEMBLER のオフチェーン DB 上で，改ざんした既存の *TotalCFP* の値が新規 *TotalCFP* が一致するように，算出に利用する *CFP* が改ざんされた場合である．この場合，改ざんが発生しているにも関わらず，検証成功という結果が得られる．この改ざんが検知されるためには，同じ検証対象部品に対し Coarse-grained 手法を適用，または改ざんされた部品が子部品となる別の親部品を対象に Fine-grained 手法を実行する必要がある．よって，検証結果の信頼性を高く保つためには，いずれかのデータ検証手法を偏って実行するのではなく，計画的に 2 つの手法を実行することが求められる．

6 まとめと今後の課題

本研究では，自動車部品のカーボンフットプリントを管理する分散 DB システムを開発にむけ，データを検証可能な形で保存するための *TotalCFP* 算出プロセス，および Peer to Peer 上でデータ検証を叶える *TotalCFP* 検証プロセスを提案した．特に，*TotalCFP* 検証プロセスでは Fine-grained 手法と Coarse-grained 手法という検証粒度の異なる 2 つの手法を検討した．提案手法はブロックチェーンプラットフォームの Hyperledger Iroha をベースに実装を行った．

実行時間を調査した実験により，Fine-grained 手法の実行時間は部品木の高さの影響を強く受け，Coarse-grained 手法では安定的に高速な検証時間が確認された．本稿での実験では，同じ部品木に対する実行時間は最大で約 1/6 程度にまで改善されていた．その一方で，Fine-grained 手法は検証対象部品の部品木全体を検証範囲とするが，Coarse-grained 手法は改ざん検知範囲が限定される側面があり，実行速度と検証範囲はトレードオフとなった．

今後の課題として，第一に 2 つのデータ検証手法について，様々な改ざんパターンに対する検知性能を調査し，リスク分析を詳細に行う必要がある．また，より実用性を高めるべく，改ざんを検知するだけで終わらず，発生箇所の特定も可能なシステムを目指し，アプローチの検討・実装を行っていく．第二に，部品の重複の考慮である．本稿では部品は重複なしとして実装，評価実験を行っていたが，実際の自動車には同じ部品が複数個使われている．重複部品はメモ化のように値を保管しておくことで，更なる高速化が望めるため，これについても挑戦していく．

謝　　辞

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

文　　献

[1] CREST : 検証可能なデータエコシステム. 入手先 ⟨https://www.kde.cs.tsukuba.ac.jp/crest/index.html⟩ (参照 2024-12-30).
[2] Catena–x. 入手先 ⟨https://catena-x.net/en/1⟩ (参照 2025-1-6).
[3] Mobi. 入手先 ⟨https://dlt.mobi/⟩ (参照 2025-1-6).
[4] 経済産業省. Ouranos ecosystem(ウラノス・エコシステム). 入手先 ⟨https://www.meti.go.jp/policy/mono_info_service/digital_architecture/ouranos.html⟩ (参照 2024-12-29).
[5] 経済産業省デジタルアーキテクチャ・デザインセンター (DADC). Ouranos ecosystem サプライチェーン上のデータ連携の仕組みに関するガイドライン (蓄電池 cfp·dd 関係). ⟨https://www.ipa.go.jp/digital/architecture/guidelines/scdata-guidline.html⟩(参照 2025-1-7).
[6] Haruka Hori and Masato Oguchi. A study of blockchain-based metadata management and its use for data verification. In *In Proc. the 12th International Symposium on Computing and Networking(CANDAR2024)*, 2024.
[7] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
[8] Bitcoin. 入手先 ⟨https://bitcoin.org/ja/⟩ (参照 2024-12-30).
[9] Ethereum. 入手先 ⟨https://ethereum.org/ja/⟩ (参照 2024-12-30).
[10] Hyperledger foundation project iroha. 入手先 ⟨https://www.hyperledger.org/projects/iroha⟩ (参照 2024-1230).
[11] Hyperledger foundation. 入手先 ⟨https://www.hyperledger.org/⟩ (参照 2024-12-30).
[12] N. Mishra and H. Levkowitz. Pdv: Permissioned blockchain based personal data vault using predictive prefetching. In *BIOTC '21: Proceedings of the 2021 3rd Blockchain and Internet of Things Conference*, pp. 59–69, 2021.
[13] G. Zyskind, O. Nathan, and A. S. Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pp. 180–184, 2015.
[14] 萱原正彬, 本田祐一, 山田達大, Le Hieu Hanh, 串間宗夫, 小川泰右, 松尾亮輔, 山崎友義, 荒木賢二, 横田治夫. ブロックチェーンとプロキシ再暗号化を用いた共有範囲設定可能な医療情報管理. In *DEIM Forum 2019*, 2019.
[15] N. B. Truong, K. Sun, and Y. Guo. Blockchain-based personal data management: From fiction to solution. In *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, pp. 1–8, 2019.
[16] 坂本明穂, 小口正人. 実ネットワークで接続されたデータ検証可能な分散データベースの性能に関する検討. マルチメディア，分散，協調とモバイル (DICOMO2024) シンポジウム, 2024.