What-If 分析フレームワークによる 改ざんデータ復元が可能なデータ連携基盤の提案

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　社会課題の解決に向け，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・管理するようなデータ管理基盤の実現が期待される．我々はこれまでに自動車部品製造業におけるカーボンフットプリント (CFP) 管理を題材に，CFP データの改ざん発生を検知・特定できるような手法を提案してきた．一方で，システムに改ざんデータが残る限りは，信頼性と可用性が保たれない．そこで本稿ではこれに加え，特定した改ざん箇所を What-If 分析フレームワークによって復元可能となるようなシステムデザインを提案する．What-If 分析フレームワークは,「特定の事象が発生した場合にどうなるか (What-If)」という仮定に基づく影響を事前にシミュレートし，遡及処理によってその影響を反映させる手法である．サプライチェーンというトランザクションが複雑に関係し合う環境下において，What-If 分析フレームワークを導入することで，無関係なトランザクションのリプレイを防ぎながら，意味論的に矛盾のないデータ復元を実現する．評価実験では単純なフルリプレイによるデータ復元手法と比較して，7,310 倍も高速という結果が得られ，CFP 管理基盤における提案手法の有効性が示された．

キーワード　分散データベース，データ連携基盤，データ復元，What-If 分析，ブロックチェーン

1　はじめに

データが核となった現代社会では，多様なデータを安全に蓄積・活用するような新しいデータ管理基盤による社会課題の解決が期待されている．その一例として，脱炭素社会に向けたサプライチェーン全体でのカーボンフットプリント管理基盤の実現が挙げられる．カーボンフットプリント (以降，CFP：Carbon Footprint of Products) とは，図 1 のように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 CO_2 排出量に換算し，製品単位で表示する仕組みである [1]．各構成部品の CFP には，その製造工程で直接排出される温室効果ガスだけでなく，全ての下位構成部品から蓄積された排出量も含まれており，サプライチェーン全体に渡る追跡と算出が必要となる．そこで，複数の企業を横断する分散 DB システムでトレーサビリティを表現し，実際のサプライチェーンを遡りながら CFP データを安全に集約・算定・保管するような CFP 管理基盤の実現が重要となる．CFP 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により，効率的な削減につながること，二点目に表示により低炭素・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．

CFP 管理基盤の実現に際して，次の課題解決が求められる．第一に，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．第二に，データ改ざんリスクへの対応が求められる．改ざんは外部攻撃だけでなく，環境規制圧力による社内不正の可能性もあり，システムの改ざん検知機能の実現が必須である．第三に，サプライチェーンのような階層関係は非常に複雑であり，高速かつ正確な処理を行うための工夫が求められる．我々はこれまでに，複雑な関係を持つ仕組みの CFP データの改ざんを検知し，具体的な発生箇所が特定可能であるようなデータ連携基盤を提案してきた [2]．システムの実装にあたっては，実世界で進められている CFP 管理基盤を参考に，プライベートブロックチェーンプラットフォーム Hyperledger Iroha を基盤とした．自動車部品製造業での運用を想定した設計・実証実験を行なっている．

図 1　カーボンフットプリントの概要

一方で，有害な改ざんデータがシステムに含まれている限りは，部品の階層関係により，サプライチェーンの全体的に悪影響を及ぼす可能性がある．特定された不正箇所を正しい状態へ復元し，システムの可用性を維持するために，本稿では特定された改ざんデータが復元可能であるようなシステムデザインを提案する．

本システム実現にあたり，改ざん復元のアプローチはこれまでに様々な研究がなされてきた．例えば Ammann ら [3] は，悪意あるトランザクションからの回復において，依存関係に基づく局所的な復元アルゴリズムを提案した．しかし，彼らの手法はトランザクション間の依存関係をすべてログから解析する必要があり，大規模・高頻度なシステムにおいては依然として計算コストや即応性の面で課題が残る．また，意味論に基づく依

存解析はなされておらず，意味論的矛盾が生じる可能性がある．そのため，サプライチェーンのように膨大なトランザクションが複雑に関係し合う環境下で，ある特定のデータ修正を行う場合，修正とは無関係なトランザクションまで含めた広範な再実行を余儀なくされることが懸念される．

そこで本稿では，先行研究の改ざん特定可能な CFP 管理基盤への What-If 分析フレームワーク導入を検討する．What-If 分析フレームワークは，「特定の事象が発生した場合にどうなるか (What-If)」という仮定に基づく影響を事前にシミュレートし，遡及処理によってその影響を反映させる手法である．これによって，「ある CFP データが特定の値であったと仮定した場合，上位部品の CFP 値がどのように変化し，整合性が取れるか」という仮定を生成・解析することで，無関係なトランザクションのリプレイを省略することが可能となる．

本稿は以下の通り構成される．第 2 節では関連研究として先行研究と本研究で採用した What-If 分析フレームワークを紹介する．第 3 節では，CFP データの改ざんを特定，データ復元可能であるようなシステムを提案する．第 4 節では，提案手法の有効性を確認するための実験とその結果を述べる．第 5 節では提案手法の実用性に関する考察を行い，第 6 節でまとめる．

2 関 連 研 究

2.1 先 行 研 究

先行研究 [2] では，CFP データに発生した改ざんを特定可能にすべく，**ハッシュ部品木**を提案した．ハッシュ部品木はハッシュ値をツリー状に組み合わせたものである．各部品の CFP データをハッシュ化し，部品の構造関係に従って複数のハッシュ値を XOR で統合することで，部品ごとのハッシュ値を決定する．具体的なハッシュ値の定義は以下の通りである．ここで，$\boldsymbol{Hash}(x)$ は x にハッシュ関数を適用した結果を出力し，$\boldsymbol{Path}(x,y)$ は部品 x から部品 y までのパスを出力する．

- 任意の単品部品 P_s のハッシュ値 H_s :

$$H_s = Hash(CFP_s)$$

- n 個の子部品 $\{P_{c-1}, P_{c-2}, \cdots, P_{c-n}\}$ $(n \geqq 2)$ を持つ任意の構成部品 P_c のハッシュ値 H_c :

$$H_c = Hash(CFP_c) \oplus H_{c-1} \oplus H_{c-2} \oplus \cdots \oplus H_{c-n}$$

$P_{c-i}, (2 \leqq i \leqq n)$ が重複部品の場合，H_{c-i} を以下に置換する．

$$Hash(H_{c-i}||Path(P_c, root))$$

ハッシュ部品木における XOR の性質の貢献は二点ある．第一に，交換法則 $(a \oplus b = b \oplus a)$ によって，ハッシュ値の統合順序に非依存となる．ハッシュ値を統合するデータ構造の代表にハッシュ木が挙げられる．ハッシュ木はハッシュ値同士を連結していくが，連結順序を一意に定める必要がある．本シナリオにおいて部品の順序は重要ではなく，XOR の採用によって順序情報を省略可能となる．第二に，自己反転性 $(a \oplus a = 0)$ によって，改ざん特定が実現可能となる．詳細は第 2.1 節で述べる．一方で，ハッシュ部品木に重複部品が含まれる場合，特別な処理が必要となる．重複部品とは，複数の構成部品に使用される部品のことを指す．重複部品は単純な XOR の結合を行うと，上位ノードにて重複部品同士で自己反転性が働いてしまう．このような意図しない自己反転性を防ぐために，一意に定まるようなハッシュ値への置換が必要となる．

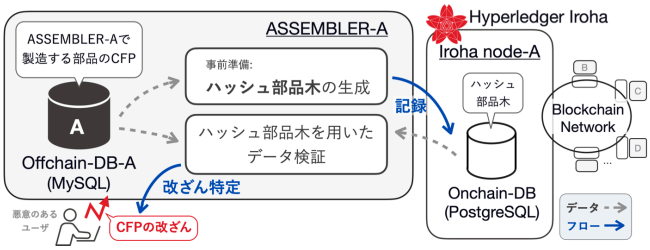

図 2　改ざんが特定可能な CFP 管理基盤の概要

ハッシュ部品木を導入した改ざん特定可能な CFP データ管理システムの全体像を図 2 に示す．サプライチェーンに属する企業を ASSEMBLER と呼ぶ．ASSEMBLER はローカル環境に Offchain-DB を持つ．Offchain-DB には自社の製造部品の CFP が平文で保管される．ASSEMBLER はそれぞれ，プライベートブロックチェーンプラットフォーム Hyperledger Iroha のノード (Iroha node) を持ち，Blockchain Network で接続される．Hyperledger Iroha は軽量かつ導入が容易で，ビザンチン耐性 (BFT：Byzantine Fault Tolerance) を備える YAC コンセンサスアルゴリズムを採用している．Iroha node 上には Blockchain の状態を反映したスナップショットである Onchain-DB が動作しており，これらは全 Iroha node で同期される．Onchain-DB にはルートを自動車とするハッシュ部品木が保管される．

以降の節で改ざん特定可能な CFP 管理基盤の処理手順を記す．

ステップ 1 改ざん特定の事前準備: ハッシュ部品木の生成

改ざんを特定可能にするために，あらかじめハッシュ部品木を生成しておく必要がある．具体的な手順は以下の通りである．

1) 全 ASSEMBLER の Offchain-DB から CFP を集約．
2) ハッシュ部品木の定義に従い部品ごとのハッシュ値を決定．
3) Hyperledger Iroha のスマートコントラクトで 2) を Onchain-DB に格納．

ここで，CFP を更新する場合，ハッシュ部品木の更新も必要となる．更新部品からルート部品 (自動車) までのパス上に存在する部品らのハッシュ値を再算出し，Hyperledger Iroha のスマートコントラクトで Onchain-DB を更新する．この一連の処理を CFP 更新処理という．

ステップ 2 データ検証による改ざん特定

ハッシュ部品木を Onchain-DB に格納することで，Offchain-DB の CFP データに改ざんが発生しても，「どの部品に改ざんが発生したのか」という具体的な特定が可能となる．以下にデータ検証の手順を示す．

1) 検証部品のハッシュ値を現在のデータから再算出．
2) Onchain-DB に保管したハッシュ値と比較．
3) 2) が不一致なら改ざん特定処理を実行．

i. 検証部品からハッシュの不一致を探索.
ii. 親 (H'_P) から改ざんされた子の影響 (H'_C) を除去したものに，Onchain-DB から取得した値 (H'_C) を XOR した $Check$ を算出.

$$Check = H'_P \oplus (H'_C \oplus H_C)$$

iii. $Check$ と正しいハッシュ値 (H_C) を比較し改ざんの再判定を行う．一致なら改ざんは子部品のみ，不一致なら親も改ざんがあると判定.

2.2 What-If 分析フレームワークに関する研究

2.2.1 What-If 分析の技術的概要

What-If 分析では，データ記録や取引履歴における仮想的な変更をシミュレートし，変更された条件下での結果を予測する．サプライチェーン管理の分野においては，需要変動，物流の遅延，あるいは構成部品の変更といったリスク要因が，全体のリードタイムやコストに与える影響を評価するために活用される．

本研究の実証実験シナリオである CFP 管理基盤においては，CFP データの改ざんが疑われる場合や，欠損した CFP データを復元する際に有用である．具体的には,「ある CFP データが特定の値であったと仮定した場合，上位部品の CFP 値がどのように変化し，整合性が取れるか」というシナリオを生成・解析することで，最適値を探索することが可能となる．

2.2.2 Ultraverse

従来の What-If 分析フレームワークは，ソフトウェアのアプリケーション層またはデータベース層のいずれかのみを対象とした設計であった [4]．現存の大半のソフトウェアアプリケーションが両レイヤーのデータフローを伴っていることから，最適ではない．これに対し，本稿で採用する Ultraverse はこれらの層をシームレスに統合する手法を実現しており，効率性と正確性の要件にも対応した [5].

Ultraverse は SQL トランスパイラと遡及的操作プラグインという二つの構成要素から成る．SQL トランスパイラでは，静的解析である抽象構文木と動的解析である動的シンボリック実行によって，アプリケーションコードを解析し，同等の SQL プロシージャに変換する．SQL プロシージャとは，複数の SQL 命令をカプセル化して DBMS に保存し，呼び出せるようにするものである．これにより，通信回数の削減や高い再利用性が確保され，処理の高速化が実現される．遡及的操作プラグインでは，列・行単位のクエリの依存解析や並行リプレイ，ハッシュを用いた枝刈りによって SQL プロシージャ上で効率的に What-If 分析を実行する．

Ultraverse の処理手順は以下の 3 段階で構成される．

1) 事前準備 : SQL トランスパイラがアプリケーションコードを解析し，SQL プロシージャ化してデータベースに登録.
2) ログ解析とグラフ構築 : データベースのトランザクションログを取り込み，解析可能な形式へ変換．トランザクション間の依存関係を解析し，以降の処理を高速化するためのグラフ構造をメモリ上に構築.
3) 遡及処理 : ユーザーによる「仮定」のリクエストに対し，2) の依存グラフを用いて影響が及ぶ範囲を特定．1) の SQL プロシージャと遡及的操作プラグインによる高速な選択的リプレイを実行し,「仮定」が実行されたデータベースの状態を提示.

マルチ DBMS ベンチマークツール BenchBase [6] を用いた評価実験では，先行研究 [4] に比べて，What-If 分析時間が 6,480 倍高速，メモリ効率が 1,370 倍高いという結果が得られている．また，先行研究は意味論的正確性を維持できておらず，Ultraverse は What-If 分析の速度と正確性において顕著な改善が実証された.

2.3 CFP データ管理基盤に関する既存研究

近年の研究では，CFP 管理およびサプライチェーン全体のサステナビリティ向上に向けた基盤技術として，ブロックチェーンの活用が検討されている．Pekel と Yayla [7] は，CFP を検証可能な製品属性として扱うようなブロックチェーンベースのカーボンフットプリント管理フレームワークを提案した．彼らは CFP をブロックチェーン上の記録に組み込むことで，製品レベルの排出量を安全に追跡・開示し，組織境界を越えてデータの完全性を保証できることを示した．Wang ら [8] は，ブロックチェーンがいかにしてサプライチェーンの統合能力を向上させ，炭素排出量の削減に寄与するかを示す概念フレームワークを提示した．彼らの研究は，企業間の低炭素協力を実現する上で，透明性，信頼，およびスマートコントラクトが果たす役割を強調している．Kumar ら [9] は，サプライチェーン管理におけるブロックチェーン技術に関する包括的なレビューを行い，技術革新，応用事例，および課題を特定した．彼らはトレーサビリティ，偽造防止，環境パフォーマンスなどを含むユースケースの分類を示し，サプライチェーンの持続可能性課題に対処する上でのブロックチェーンの汎用性を強調している．Kouhizadeh ら [10] は，多階層の持続可能なサプライチェーンに向けたブロックチェーンベースのアプローチを提案した．彼らは，分散台帳がサプライチェーンの異なる階層においてどのようにサステナビリティを支援できるかを分析し，特に調整メカニズムと情報共有メカニズムに注目した．最後に，Gu [11] は，自動車産業サプライチェーンにおけるブロックチェーンを活用した炭素排出量の統計および管理について調査した．彼らの研究は，排出量データを安全かつ透明性を持って集約するブロックチェーンの可能性を示し，CFP モニタリングの実用的な基盤となることを強調している.

これらの既存研究は，サプライチェーンにおける透明性，トレーサビリティ，サステナビリティの向上に対するブロックチェーンの有用性を示しているものの，その主軸は CFP 量の算出と管理に置かれている．これに対し，我々はこれまでに，データ検証による改ざん特定機能の導入を検討してきた．さらに，What-If 分析フレームワークを用いた無駄のないデータ復元を実現する点において，既存研究と一線を画すものである.

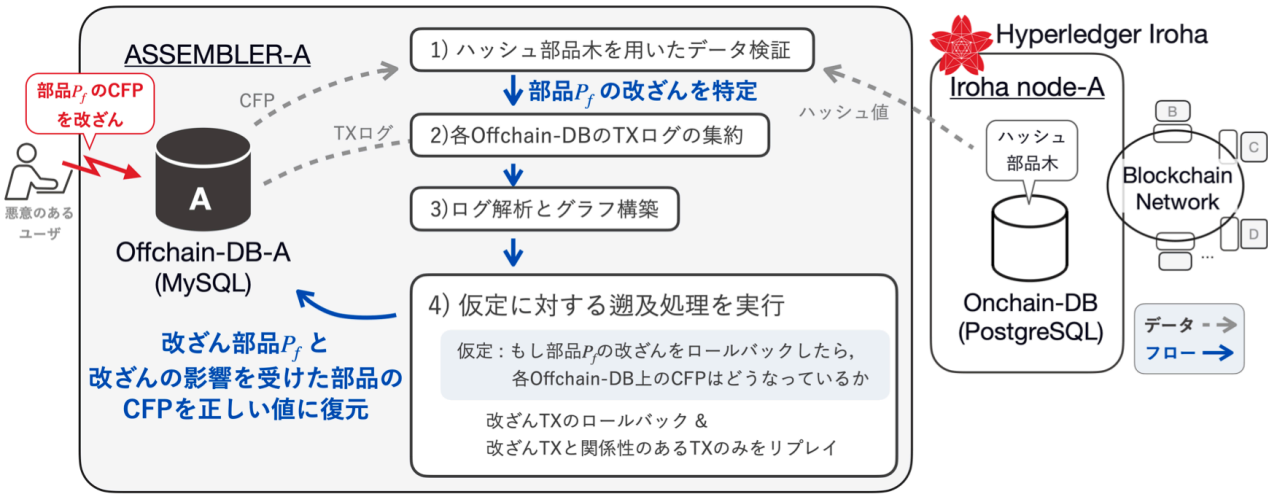

図 3　提 案 手 法

3　提 案 手 法

サプライチェーンに参加する企業間が複雑に連携し合う CFP 管理基盤において，改ざんが発生するとその影響は他企業や他製品にも及ぶ．そのため，先行研究で特定した改ざんは復元する必要がある．さらに，改ざんデータを復元する際には，整合性が保たれており，かつ意味論的な矛盾があってはならない．また，改ざんが発生したシステムにおいて，最新のチェックポイント以降のトランザクションは「改ざんトランザクション」「改ざんの影響が伝搬するトランザクション」「改ざんとは無関係なトランザクション」に分類される．先の 2 つに分類されるトランザクションのみのロールバックやリプレイが効率的である一方で，複数のトランザクションが複雑に関係し合う環境下での実現は容易ではない．

そこで本稿では，先行研究 [2] 改ざんが特定可能な CFP 管理基盤に What-If 分析フレームワークを導入し，「もし過去のあの時点で改ざんが発生していなかったら，現在の Offchain-DB 上の CFP データはどうなっているか.」という仮定をシュミレート，結果を Offchain-DB に反映させることで，改ざんデータを復元する手法を提案する．使用する What-If 分析フレームワークは，第 2.2.2 節で述べた Ultraverse を用いる．Ultraverse は与えられた仮定に対し，依存グラフを用いた影響範囲特定によって，改ざんとは無関係なトランザクションのリプレイを省略する．単一 Peer 設計である Ultraverse を分散環境への導入するにあたり，本稿ではシステムに参加する全ての ASSEMBLER からのトランザクションログの集約・解析・遡及処理が逐次的に行われるようなスクリプトを用意した．Ultraverse の基本の処理手順は第 2.2.2 節の通りであるため，ここでは本提案システムにおける具体的な処理手順と，改ざん特定手法との連携に焦点を当てる．以降の節に，データ復元の事前準備およびデータ復元手順を述べる．

3.1　事 前 準 備

Ultraverse による効果的に遡及処理を実現すべく，次の三点の事前準備が必要となる．第一に，ハッシュ部品木の生成である．第 2.1 節の手順 1 に示した手順に従い，システムで扱う製品のハッシュ部品木を構築する．ハッシュ部品木を用いたデータ検証によって改ざんの具体的な発生箇所が特定可能となる．Ultraverse に与える「仮定」は特定された改ざんの情報によって厳密となり，より無駄のない遡及処理が期待される．第二に，Ultraverse による CFP 更新スクリプトの SQL プロシージャ化である．CFP 管理基盤において，最頻の Write 処理は CFP 値の更新である．すなわち，遡及処理時においても繰り返し実行されることから，更なる高速化に貢献する．第三に，定期的なチェックポイントの収集とログリセットである．一定時間ごと，あるいは一定ログサイズごとに各 Offchain-DB のチェックポイントを取得し，同時にバイナリログをリセットする．これにより，遡及処理時の探索空間を最小化すると同時に，ストレージ効率を維持する．

3.2　データ復元手順

特定された改ざん CFP データの復元を実現する手順を図 3 と以下に示す．

1) ハッシュ部品木を用いたデータ検証で改ざん部品 P_f を特定．
2) 分散する各 ASSEMBLER の Offhchain-DB のトランザクションログを Ultraverse に集約．
3) Ultraverse の処理手順「2) ログ解析とグラフ構築」の実行．
4) 部品 P_f に改ざんが発生したトランザクションをロールバックする命令を Ultraverse に与え，Offchain-DB の遡及処理を実行．すなわち，Ultraverse は「もし部品 P_f の改ざんをロールバックしたら，各 Offchain-DB 上の CFP の値はどのように整合が取れるか」という仮定に対するシミュレーションを行い，その結果を Offchain-DB に反映する．

手順 2)，3) を遡及準備，手順 4) を遡及処理とカテゴライズする．

ここで，Ultravserse は分散環境における遡及処理を並列実行する設計ではないことに注意されたい．そのため，分散システムである「改ざん特定可能な CFP 管理基盤」への導入に

図 4　データ復元手法の具体例

むけ，手順 2) で全ての Offchain-DB のトランザクションログの集約が求められる．Ultraverse は，手順 3) において全ての ASSEMBLER のトランザクションログを解析し，依存グラフを構築する設計とした．また，手順 4) においても，逐次的に各 ASSEMBLER の遡及処理が実行される．データ復元実行中においては，ダーティリードや DB の不整合を防ぐために，Offchain-DB の一時的なロックが求められる．

挙動の具体例を図 4 に示す．同じの親を持つ部品 P_p と P_o，そして部品 P_p を親として持つ部品 P_c という構成がある時，部品 P_c に発生した改ざんを復元していく．まず，初期状態にて部品 P_p の CFP が 10，部品 P_c の CFP が 1，部品 P_o の CFP が 3 であったとする．TX1 で部品 P_c の CFP が 1 から 100 に改ざんされ，TX2 ではその改ざんの影響が部品 P_p に伝搬する．TX3 では新規の温室効果ガスの排出量データ取得などにより，部品 P_o の CFP が+2 される．TX4 でも同様に，部品 P_c と部品 P_p の CFP が+1 される．この状況において，提案手法により，改ざんの特定やグラフ構築などが行われる．最後に Ultraverse が What-If 分析を行い，改ざんの発生した TX1 をロールバックして，関係のある TX4 のみをリプレイされることで，部品 P_p や部品 P_c はあるべき値に復元される．この時，ロールバックした TX1 とは無関係な部品 P_o の更新トランザクションの TX3 のリプレイは省略される．

4　評　　価

本稿では，データ復元手法の時間的コストと正確性の調査，および What-If 分析フレームワークを用いない改ざん復元アプローチとの比較実験を行った．ここで本評価は，先行研究 [2] で提案した「CFP データの改ざんが特定可能であるシステム」上に改ざん復元機能を追加する目的を前提として実施される．

4.1　実　　装

実装の構築環境とマシンの性能は図 5 に示す．物理サーバ

Ubuntuコンテナ 22.04 LTS				PostgreSQL コンテナ
Blockchain Platform	API	Ultraverse	Offchain-DB	Onchain-DB
Hyperledger Iroha ver. 1	Python	C++	MySQL ver.8.01	PostgreSQL ver.16.3
仮想環境	Docker			
OS	Ubuntu (20.04 LTS)			
物理サーバ	CPU : Intel(R) Xeon(R) Silver 4314 CPU @ 2.40GHz MEMORY : 192GB, CORE: 16, THREAD: 34			

図 5　構 築 環 境

上の Docker 環境で，Ubuntu コンテナと PostgreSQL コンテナを用意した．Ubuntu コンテナには，プライベートブロックチェーンプラットフォームの Hyperledger Iroha，What-If 分析フレームワークの Ultraverse，MySQL の Offchain-DB を構築する．PostgreSQL コンテナには Onchain-DB が構築される．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 [12]，本稿では Docker 環境を採用している．ここで，本実験では，システムに参加する ASSEMBLER 数を 3 社とした．

実装に 2 種の DBMS が使用される理由としては，Hyperledger Iroha は PostgreSQL または RocksDB を Onchain-DB として利用できる仕様であり，Ultraverse は MySQL をメイン DBMS として採用しているためである．これらのインターフェースやハッシュ部品木生成におけるデータ加工らはすべて Python で行う．ここで，データ復元手順 4) における，「改ざんが発生したトランザクション」の抽出は，本稿での実験段階では手作業で行なわれている．

4.2　時間的コストの調査

本節では，第 3.2 節に示したデータ復元手順 2) から 4) の実行時間を計測した．調査の目的は 2 点ある．

目的 1　評価指標の変化が実行時間に与える影響を明らかにする．

目的 2　分散対応していない Ultraverse を採用するにあたり，実運用での実行時間を見積もることで，有効性を調査する．

ここで，評価指標は，チェックポイント取得以降に発生した CFP 更新処理数，即ちトランザクション数，システムの規模として部品の総数および部品の重複率，部品の改ざん率を用いる．具体的な値は以下の通りであり，全改ざんデータの復元に要する時間を調査した．

- CFP 更新処理の実行回数 (回)：10,000
- 部品の総数 (点)：300/3,000/30,000
- 部品の重複率 (%)：0/10/20/30
- 部品の改ざん率 (%)：1/5/10/15

CFP 更新処理について補足をする．第 2.1 節にもあるように，ある部品の CFP 値を更新する場合には，その親部品の CFP 値

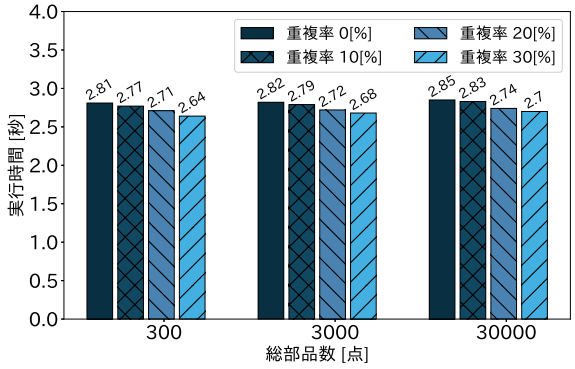

図 6　遡及準備時間

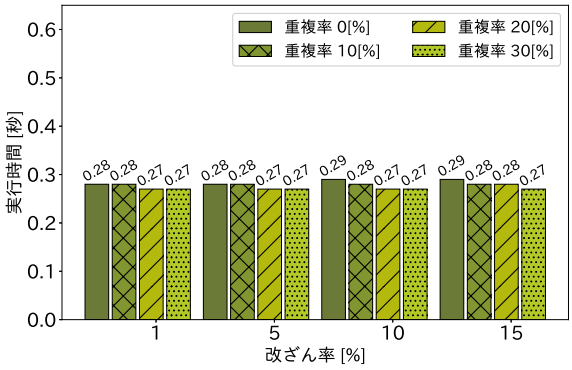

図 7　部品総数 300 点における遡及処理時間

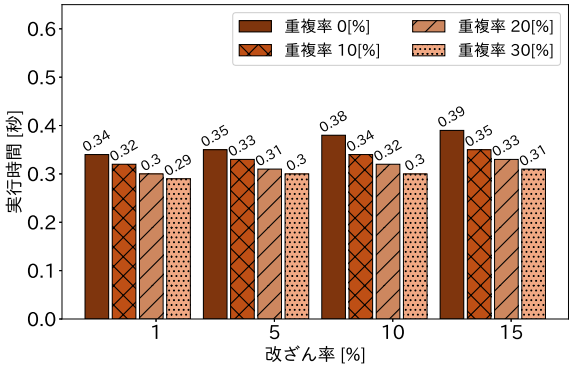

図 8　部品総数 3,000 点における遡及処理時間

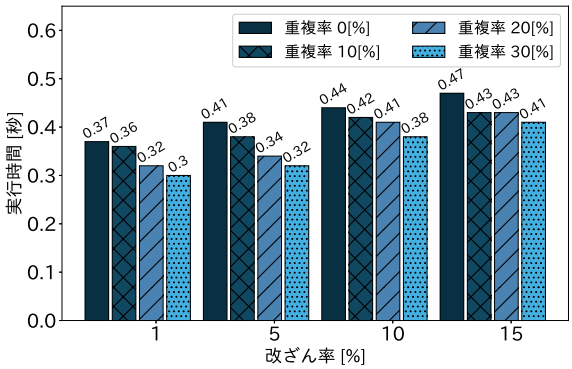

図 9　部品総数 30,000 点における遡及処理時間

の再計算とハッシュ部品木の更新も必要となる．これらの一連の処理を CFP 更新処理と呼ぶが，特に，ハッシュ部品木の更新は Hyperledger Iroha のスマートコントラクトを用いて行われることに注意されたい．また，部品の総数については，自動車一台分に相当するとされる 30,000 点とその 1/10 倍，1/100 倍規模の調査となる．

まず，データ復元手順 2), 3) の遡及準備の実行結果を図 6 に示した．部品総数が少なく，重複率は高いほど，実行時間はやや短いという結果となった．しかしながら，部品総数が 300 点から 10 倍，100 倍となっても，その処理時間は 2.64 から 2.85 秒と 1.08 倍のみの増加となった．

次に，データ復元手順 4) の遡及処理の実行結果を部品総数ごとに，図 7，8，9 に示した．先ほどの同様に，部品総数が少なく，重複率が高いほど高速であるという結果が得られた．また，改ざん率が高くなるほど，ロールバック対象のトランザクション数も増加することから，実行時間は増加傾向であった．一方で遡及処理においても，部品総数が 300 点から 100 倍となっても，その処理時間は 1.1 から 1.6 倍程度となった．

以上の評価結果を用いた Ultraverse の分散環境適用に関する有効性の議論は第 5.1 節で行う．ここで，どの条件においても，改ざん復元結果に整合性に矛盾は生じていない．

4.3　What-If 分析適用手法と非適用手法の比較

本節では，What-If 分析フレームワークの採用が CFP 管理基盤において効果的であるかを明らかにする．提案手法の What-If 分析フレームワークを用いた改ざん復元に対し，チェックポイントからのフルリプレイによる改ざん復元を比較対象とした．評価手法の概要を図 10 に，比較手法の具体的な手順は以下に示す．

1)　ハッシュ部品木を用いたデータ検証で改ざん部品 P_f を特定．
2)　最新のチェックポイントまでロールバック．
3)　ハッシュ部品木の再構築．
4)　組み込みコマンドを用いた CFP 更新処理のフルリプレイ．

手順 4) では，組み込みコマンドの実行ログの中で，使用したチェックポイントを取得した直後ものから順に CFP 更新処理を全て再実行するというものである．以上の手順の実行時間を計測した．ここで，評価パラメータは，チェックポイント取得以降に発生した CFP 更新処理数，システムの規模として部品の総数および部品の重複率，部品の改ざん率を用いる．具体的な値は以下の通りである．

- チェックポイント取得以降に発生した CFP 更新処理回数 (回) :100/1,000/10,000
- 部品の総数 (点)：300/3,000/30,000
- 部品の重複率 (%)：0
- 部品の改ざん率 (%)：15

評価結果を第 4.2 節での同条件評価結果と併せて表 1 に示す．What-If 分析適用手法と非適用手法の差は歴然であり，非適用手法は，部品総数 30,000 点で約 29 時間から 59.8 時間も要するという結果が得られた．特に，CFP 更新処理回数 10,000 回，部品総数 30,000 点の時，What-If 分析を適用した提案手法は，

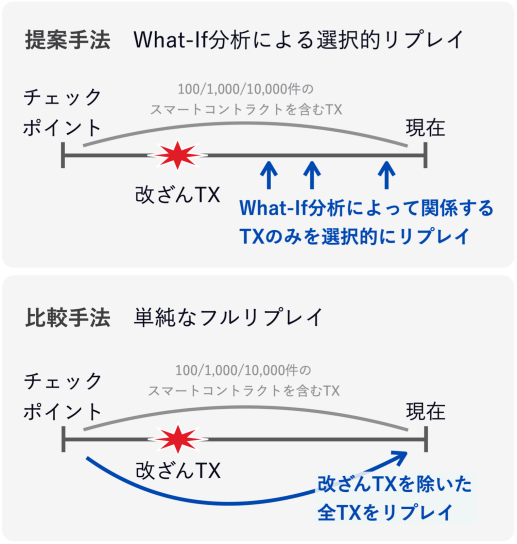

図 10　What-If 分析適用手法と非適用手法の概要

表 1　What-If 分析適用手法とフルリプレイ手法の処理時間

部品総数		300 点	3,000 点	30,000 点
CFP 更新処理回数	What-If 分析	1.14 秒	1.22 秒	1.27 秒
100 回	フルリプレイ	104 秒	105 秒	119 秒
CFP 更新処理回数	What-If 分析	1.58 秒	1.64 秒	3.21 秒
1,000 回	フルリプレイ	1,036 秒	1,040 秒	1,556 秒
CFP 更新処理回数	What-If 分析	3.10 秒	1.66 秒	3.32 秒
10,000 回	フルリプレイ	29 時間	29.1 時間	59.8 時間

非適用手法に対して，7,310 倍も高速となった．

5　考　　察

本節では，改ざん特定・復元可能な CFP 管理基盤の考察を行う．

5.1　CFP 管理基盤における What-If 分析の有効性の議論

第 4 節の評価結果をもとに，改ざん特定・復元可能な CFP 管理基盤実現に向け，What-If 分析および Ultraverse の有効性について議論する．まず，第 4.2 節の時間的コストの調査より，データ復元手順 2) から 4) の実行時間は 2.91 から 3.32 秒程度であることが確認された．ここで，部品の総数 30,000, 部品の重複率 10%，30%という条件における，データ復元手順 1) から 4) の実行時間を図 11，12 に示した．データ復元手順 1) の改ざん特定は，遡及準備や遡及処理と同様に，部品の総数が高いほど，重複率が低いほど，改ざん率が高いほど，その実行時間が長時間となる．遡及準備や遡及処理に比べ，それらの要素の影響をより強く受けていることから，処理のボトルネックになるのは，改ざん特定手法であると示唆される．一方で，最も時間を要した部品重複率 10%, 改ざん率 15%においても，データ復元手法は 31.1 秒で完了している．また，第 4.3 節のフルリプレイとの比較結果と併せても，What-If 分析による選択的なリプレイは高速であり，有効であると示唆される．

次に，CFP 管理基盤という分散システムにおいて，単一 Peer

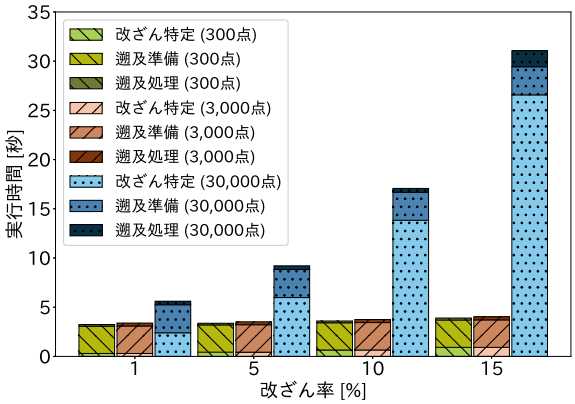

図 11　データ復元手法の実行時間 (部品総数:30,000, 重複率 10%)

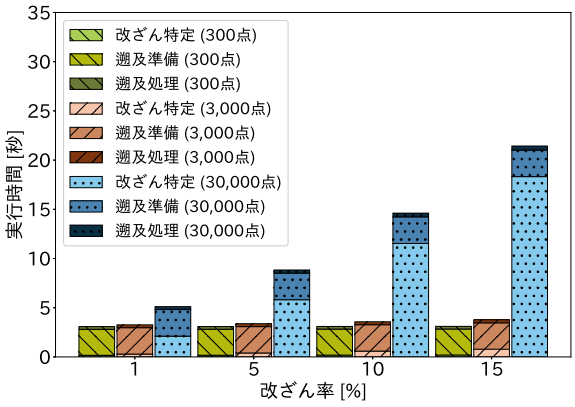

図 12　データ復元手法の実行時間 (部品総数:30,000, 重複率 30%)

設計である Ultraverse の有効性について考察する．Ultraverse は分散環境における遡及処理を並列化する設計となっていないため，データ復元手順 3) および 4) は，各 ASSEMBLER の Offchain-DB のチェックポイントを用いて直列に実行される．そのため，単純に ASSEMBLER 数が増加すれば，それに比例して解析時間も増加することが予想される．しかしながら，評価結果より，ASSEMBLER 1 つあたりの実行時間は 1 秒前後であり，仮にサプライチェーンを構成する企業数が数十社から百社規模に拡大したとしても，総処理時間は数分程度に収まる見積もりとなる．

CFP 管理シナリオにおいては，データ更新や整合性確認の頻度が相対的に低く，リアルタイム性よりもデータの正確性や信頼性が求められる．以上の議論より，提案手法の性能は，実運用において十分に許容範囲内であることが示唆された．

5.2　提案手法の実運用性能に関する議論

実用化を想定したネットワーク規模の拡大に伴う実行時間の増加について検討する．本実験でのブロックチェーンネットワークは小規模な構成であったが，実際の CFP 管理シナリオでは，システムに参加する企業数は約 100–150 社程度になると予想される．この規模拡大において実行時間増加の主な要因となりうるのは，Iroha および Ultraverse の 2 点である．第一に，Iroha については，組み込みコマンドが実行されるハッシュ部品木の生成と更新における処理時間の増加が懸念される．特に

処理時間を要するのは前者であるが，ハッシュ部品木の生成は，システムに新規部品を登録する際などに実行されるものであり，その頻度は極めて低い．一方，実行頻度の高い更新処理に関しては，現状の 30,000 点規模の実験でも約 1 秒で完了しているため，規模が拡大しても十分に許容できる範囲であると考えられる．第二に，Ultraverse であるが，前述の通り，各社 1 秒程度の見積もりであり，CFP 管理基盤のデータ復元において数分の待ち時間は実用上の障壁とはならないため，大規模運用においても十分に適用可能であると考えられる．

以上より，提案する CFP 管理基盤は，改ざんに対する高い耐性を持ちつつ，実用的な運用性能を有していると結論付けられる．

6 まとめと今後の課題

本稿では，先行研究で開発された改ざん特定可能な自動車部品 CFP データ連携システムに対し，What-If 分析フレームワークである Ultraverse を導入することで，特定された改ざんデータからの迅速な復元を可能とする手法を提案し，その有効性を検討した．What-If 分析フレームワークに「ある CFP データが特定の値であったと仮定した場合，上位部品の CFP 値がどのように変化し，整合性が取れるか」という過程を与え，シミュレーション解析を行うことで，無駄なリプレイを省略可能となるほか，意味論的にも矛盾のない処理が実現可能となった．評価実験の結果，部品総数 30,000 点のシステム規模において，全データの 15%（4,500 件）におよぶ改ざんが発生した状況下でも，データ復元処理自体は 3.32 秒程度で完了可能であることが確認された．さらに，What-If 分析フレームワークを適用しない従来実装との比較実験においては，本手法が最大で約 7,310 倍も高速であることが示され，先行研究のシステムに対する What-If 分析フレームワーク導入が有効であると結論付けられた．

今後の展望として，以下の 2 点に取り組む．第一に，実用化を想定したネットワーク規模の拡大評価である．システムの接続 ASSEMBLER 数を増加させた場合の評価実験を実施し，スケーラビリティを検証する．考察でも述べたように，ASSEMBLER 数の増加に伴い復元処理時間が線形に増加する可能性があるため，この直列実行によるボトルネックを詳細に調査するとともに，分散環境下での並列処理化などによる解消手法を検討する．第二に，ロールバック対象となる改ざんトランザクション抽出の自動化である．現状の実装では，改ざんが疑われる箇所が特定された際，データ復元のためにどのトランザクションをロールバックするかを運用者が手動で選択する必要がある．今後は，改ざんの影響範囲を正確に特定し，ロールバックすべきトランザクション群を自動的に決定するアルゴリズムを実装することで，人手を介さない完全な自動復元の実現を目指す．

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] *ISO 14067:2018 - Greenhouse gases - Carbon footprint of products Requirements and guidelines for quantification.* January 2018.

[2] H. Hori, H. H. Le, M. Oguchi. *A data integration platform with identifiable falsification detection and its evaluation in automotive parts manufacturing.* In 2026 20th International Conference on Ubiquitous Information Management and Communication (IMCOM), pp. 1–8, 2026.

[3] P. Ammann, S. Jajodia, Peng Liu. *Recovery from malicious transactions.* IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 5, pp. 1167–1185, 2002.

[4] F. S. Campbell, B. S. Arab, B. Glavic. *Efficient answering of historical what-if queries.* the 2022 InternationalConference on Management of Data, SIGMOD'22, pp. 1556 – 156, 2022.

[5] R. Ko, C. Xiao, M. Onizuka, Z. Lin, Y. Huang. *Ultraverse: An efficient what-if analysis framework for software applications interacting with database systems.* Proceedings of the ACM on Management of Data, Vol. 3, Iss. 1, Article No. 84, pp. 1–27, 2025

[6] Benchbase. `https://github.com/cmu-db/benchbase`.

[7] U. Pekel, O. Yayla. *Blockchain-based carbon footprint management.* In Cryptology ePrint Archive, Vol. 2024/1863, 2024.

[8] M. Wang, B. Wang, A. Abarehii. *Blockchain technology and its role in enhancing supply chain integration capability and reducing carbon emission: A conceptual framework.* In Sustainability, Vol. 12, Iss. 24, pp. 10550, 2020.

[9] N. Kumar, K. Kumar, A. Aeron, F. Verre. *Blockchain technology in supply chain management: Innovations, applications, and challenges.* Technical and Informatics Report, Vol. 18, Article No. 100204, 2019.

[10] V. K. Manupati, T. Schoenhrr, M. Ramkumar, S. M. Wanger, S. K. Pabba, R. I. R. Singh. *A blockchain-based approach for a multi-echelon sustainable supply chain.* International Journal of Production Research, Vol. 58, No. 7, pp. 2222–2241, 2019.

[11] X. Gu. *Exploration of carbon emission statistics and management in the supply chain of automobile industry based on blockchain technology.* Academic Journal of Management and Social Sciences, Vol. 6, No. 1, pp. 97–100, 2019.

[12] 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.