コンピュータ科学科情報システム系卒業論文

DTVMを用いたオフチェーン型スマートコントラクト実行基盤の組込みシステムへの適用

2026 年 2 月

102230045　岩見 一輝

概要

近年,IoT 技術の普及に伴い, 組込み機器間での自律的な取引やデータ管理を実現する手段としてブロックチェーンおよびスマートコントラクトの活用が期待されている [1]。しかし, すべてをノード上で実行する従来のフルオンチェーン実行方式では, 膨大な計算コストや実行遅延, およびガスコストの発生が課題となり, 計算資源や電力に制約のある組込み機器への適用は困難であった。また現在のブロックチェーンはクラウドサーバ等の外部リソースに依存した構成が一般的であるが, これは通信遅延や単一障害点のリスクを増大させる要因となっている。本研究では, 組込み機器上でもスマートコントラクトを実行できるようにするために, スマートコントラクトの計算処理自体をデバイス近傍のオフチェーン環境で実行し, その実行結果および正当性の証跡のみをブロックチェーンに記録する手法を提案する。この構成により, ブロックチェーンの持つ透明性と改ざん耐性を維持しつつ, 組込み機器における計算負荷の大幅な低減とリアルタイム性の向上を図る。またオフチェーン実行には DTVM（DeTerministic Virtual Machine）というスマートコントラクト専用言語である solidity を実行可能かつ高い実行速度を誇る仮想マシンを使用する。本論文では, 提案手法に基づいたプロトタイプシステムを構築し, 一般的な PC を用いた評価実験を行う。既存のフルオンチェーン実行の実行方式と比較し, 処理時間, 消費電力, ガスコストの観点から組込み機器上に適用可能であるか, またどのような問題点が発生するか検討する。

目次

第 1 章

はじめに

1.1 研究背景

近年,IoT 技術の急速な普及により, センサーデバイスやアクチュエータなどの組込み機器がネットワークを介して相互に接続される社会が実現しつつある。IoT 技術は, スマートシティ, 自動運転, サプライチェーン管理, 産業用ロボットなど, 多岐にわたる分野で基盤技術として活用されている。これらのシステムでは, 膨大な数のデバイス間でデータの授受や価値の取引が自律的に行われることが期待されており, その信頼性を担保する技術としてブロックチェーンおよびスマートコントラクトが注目を集めている [1]。スマートコントラクトは, あらかじめ定義された契約条件をプログラムとして記述し, 特定の条件が満たされた際に自動的に実行する仕組みである。中央集権的な管理者を介さず, 分散ネットワーク上で実行されるため, プロセスの透明性や高い改ざん耐性を有している。しかし, 現在のスマートコントラクトの実装, 特に Ethereum に代表されるパブリックブロックチェーンにおいては, 契約の実行（計算処理）とデータの記録（合意形成）が不可分な「オンチェーン実行」が主流である。オンチェーン実行には, 大きく分けて 3 つの課題が存在する。第一に, 計算資源の制約である。ブロックチェーンに参加する全ノードが同一の計算を重複して行う必要があるため, 計算コストが極めて高い。第二に, 処理性能の限界である。現在主流である Etherum のブロックチェーン上ではスマートコントラクトを実行する EVM という仮想マシンが存在しているが, ブロックチェーンが普及するにつれその処理性能はより高いものが求められる。第三に, 経済的コストである。実行のたびに「ガス代」と呼ばれる手数料が発生し, 頻繁にデータを更新する IoT デバイスにとって大きな負担となる。これらの課題から, 現状の IoT システムにおけるスマートコントラクトの活用は, 潤沢な計算リソースを持つクラウドサーバ上で代理実行されるケースが一般的である。

しかし, クラウド依存の構成は, 通信遅延の増大や, 特定のサーバがダウンした際にシステム全体が停止する単一障害点の問題を内包している。真の意味で自律的な IoT エコシステムを構築するためには, ネットワークの末端に位置する組込み機器そのものが, スマートコントラクトの実行能力を持つことが不可欠である。

1.2 研究目的

本研究の目的は, リソース制限の厳しい組込み機器上において, スマートコントラクトを効率的かつ安全に実行可能とするオフチェーン実行基盤を構築することである。本研究では, スマートコントラクトの複雑な論理演算をブロックチェーンの外側（オフチェーン）で, かつデバイスに近い組込み環境で実行し, その実行結果および実行が正しく行われたことを示す証跡のみをブロックチェーンへ接続・保存する仕組みを提案する。またオフチェーンでの実行には EVM と比較して実行速度の速い DTVM を採用することで, より処理性能を向上させる。これにより, ブロックチェーンが持つ高いセキュリティ性を維持しつつ, オンチェーン実行のボトルネックを解消することを目指す。本研究を通じて, 組込み機器が自律的に契約を履行できる環境を示すことで,IoT 社会における高度な分散型自動化システムの実現に寄与する。

1.3 本論文の構成

第 2 章では本研究に必要な前提知識について詳しく述べる。第 3 章ではオフチェーン実行とオンチェーンでの保存を組み合わせる仕組みについて述べる。第 4 章では作成した仕組みを使用した比較して評価実験を行い, 組込み機器への適用性を評価する。第 5 章では本研究のまとめと今後の課題について述べる。

第 2 章

前提知識

本章では, 本研究に関連する専門用語について解説する。

2.1 ブロックチェーン

ブロックチェーンとは, 分散型台帳技術の一種であり, 複数の参加者によって取引履歴を共有・管理する仕組みである。ブロックチェーンの特徴としてまず図 2.1 のようなデータの保存方法である。取引データはブロックと呼ばれる単位にまとめられ, 各ブロックは暗号学的ハッシュ関数によって直前のブロックと連結される。この構造により, 過去のデータを改ざんすることが極めて困難となり, 高い耐改ざん性が実現されている。

ブロックチェーンの 2 つ目の特徴として, 図 2.2 のように中央管理者を必要としない点が挙げられる。従来の中央集権型システムでは, 単一の管理主体がデータを管理していたが, ブロックチェーンではネットワーク参加者が全員取引を実行して確認することで取引の正当性を検証する。また中央集権型システムでは中央のサーバがエラーで止まってしまった場合, システム全体が停止してしまうが, ブロックチェーンの場合は一部のシステムが停止しても, システム全体は動き続けることができる。

このような特性から, ブロックチェーンは暗号資産だけでなく, サプライチェーン管理や IoT 分野など, 改ざん耐性や透明性が求められる分野への応用が進められている。現在主流なブロックチェーンプラットフォームは Ethereum である [4]。

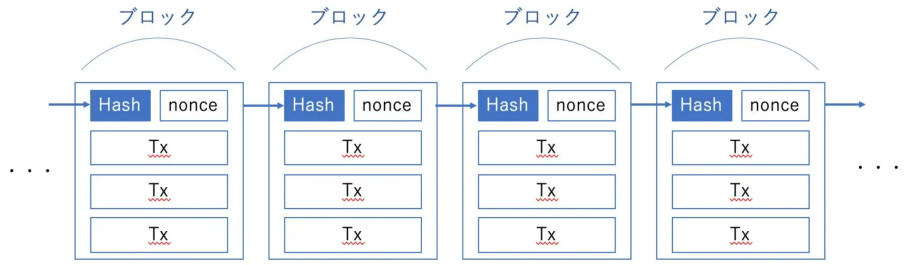


図 2.1: ブロックチェーンのデータの保存方法

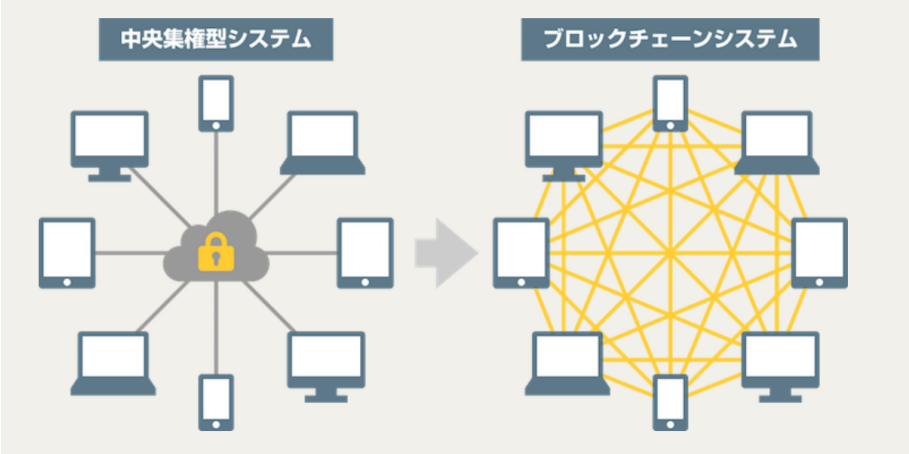


図 2.2: ブロックチェーンの概念図

2.2 スマートコントラクト

スマートコントラクトとは, ブロックチェーン上で実行されるプログラムのことであり, あらかじめ定義された条件が満たされた場合に, 自動的に処理を実行する仕組みである。Ethereum に代表されるブロックチェーンプラットフォームでは, スマートコントラクトが EVM という仮想マシン上で実行され, 取引の自動化や信頼性の高い処理を実現している。開発にはスマートコントラクト専用言語である Solidity が使われている。

従来の中央集権型のようなシステム上での契約処理では, 第三者機関やサーバが仲介する必要があったが, スマートコントラクトを用いることで, 契約条件の検証から実行までをプログラムによって自動化できる。これにより, 契約の間で他の人物が入り込むことがなく人為的ミスの削減やコスト削減が可能となる。また契約が直接行われるため, 契約の透明性も高くなる。

一方で, スマートコントラクトは通常, ブロックチェーン上のノードやサーバ環境で実行されるため, 計算資源や実行コスト（Gas）に制約がある。実行コスト (Gas) とはスマートコント

ラクトの命令ごとにあらかじめ設定された数値であり, 複雑な処理ほどそのコストは高くなっている。これによってネットワーク上で一部の処理を無限ループさせたり, 複雑で重い処理を複数回じっこうさせたりといった悪意のあるスマートコントラクトの実行を防ぎ, ネットワーク利用者の公平性を維持してる。このガスコストを計算する処理はスマートコントラクトを実行する際に自動的に追加される。そのため組込み機器のようなリソース制約の厳しい環境で直接実行することを考えた場合, このガスコスト計算処理によってより必要なリソース量が増加し, 実行するのが困難となる可能性がある。また組込み機器が高頻度で外部のデータを取得する場合, スマートコントラクトもそれだけ高頻度で実行する必要があるため, ガスコストによって処理が実行されない場合がある。これらの点が本研究の課題背景の一つとなっている。

2.3 DTVM

DTVM（Deterministic Trusted Virtual Machine）は, 決定論的な実行を保証する WebAssembly（Wasm）ベースの仮想マシンである [5]。決定論とは異なる環境においても同一の入力に対して同一の実行結果を得られることであり, これによってどのような機器を使用したとしても同じ結果をえることができる。ブロックチェーンはネットワークにつながってる機器同士がお互いの実行結果を検証するためスマートコントラクトを実行するマシンは決定論的である必要がある。そのため DTVM のこの性質は, ブロックチェーンや分散システムにおいて,EVM に変わってスマートコントラクトを実行し, 実行結果の検証や再現性を確保する上で重要である。

また WebAssembly（Wasm）という軽量な実行形式を用いることで, 組込み機器のような計算資源が限られた環境でも動作可能である。これにより, 従来はサーバ上で実行されていた処理を, 組込み機器側で安全かつ決定論的に実行することが可能となる。そして wasm ベースであることからその実行可能な言語数も多く,c++,Rust,Go など複数の言語で開発可能であり, スマートコントラクト開発用言語である Solidity でも開発をすることが出来る。

本研究では, スマートコントラクトの処理をすべてブロックチェーン上で実行するのではなく,DTVM を用いてオフチェーンで実行し, その結果のみをブロックチェーンに記録する方式を採用する。このアプローチにより, 計算コストの削減や, 組込み機器におけるスマートコントラクト実行環境の実現を目指す。

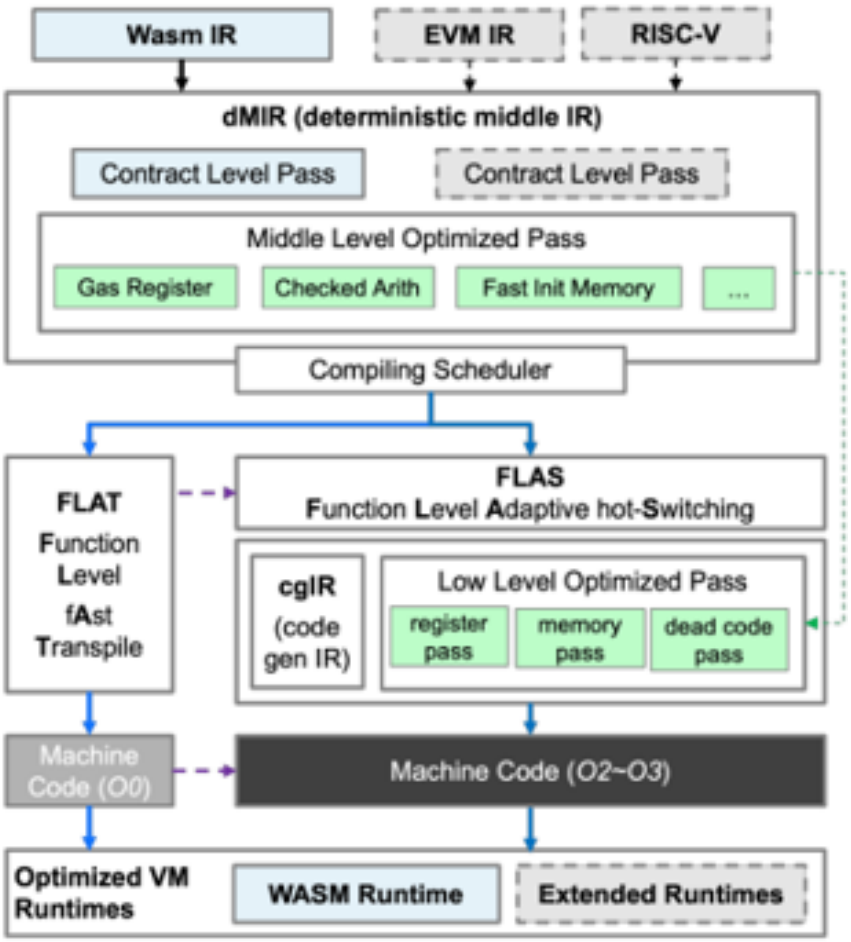

Wasm IR
EVM IR
RISC-V
dMIR (deterministic middle IR)
Contract Level Pass
Contract Level Pass
Middle Level Optimized Pass
Gas Register
Checked Arith
Fast Init Memory
...
Compiling Scheduler
FLAT
Function
Level
fAst
Transpile
FLAS
Function Level Adaptive hot-Switching
cgIR
(code
gen IR)
Low Level Optimized Pass
register
pass
memory
pass
dead code
pass
Machine
Code (O0)
Machine Code (O2~O3)
Optimized VM
Runtimes
WASM Runtime
Extended Runtimes

図 2.3: DTVM

第 3 章

提案手法

本章では, 本研究で提案する手法について述べる。

3.1 オフチェーン実行 + オンライン同期

本研究ではスマートコントラクトの処理をすべてブロックチェーン上で実行するのではなく, オフチェーンで処理を行い, その結果のみをブロックチェーンに送信して保存する方式を採用する。

従来のフルオンチェーン型の場合, 実行命令であるトランザクションを受け取るとブロックチェーンにつながっているすべてのノード上で対象のスマートコントラクトが実施される。しかし本研究ではその実行するスマートコントラクトの処理をオフチェーン, いわゆるローカルで DTVM を用いて実行し, その結果のみをブロックチェーン上に保存する仕組みを提案する。この仕組みによるメリットはまずスマートコントラクトで実行される部分が少なくなるため, ガスコストを減らすことが出来る。ガスコストが減ることで, 高頻度でプログラムを呼び出すことができる。またガスコストを計算する処理の数も減るためプログラム全体の実行時間が短くなる可能性がある。

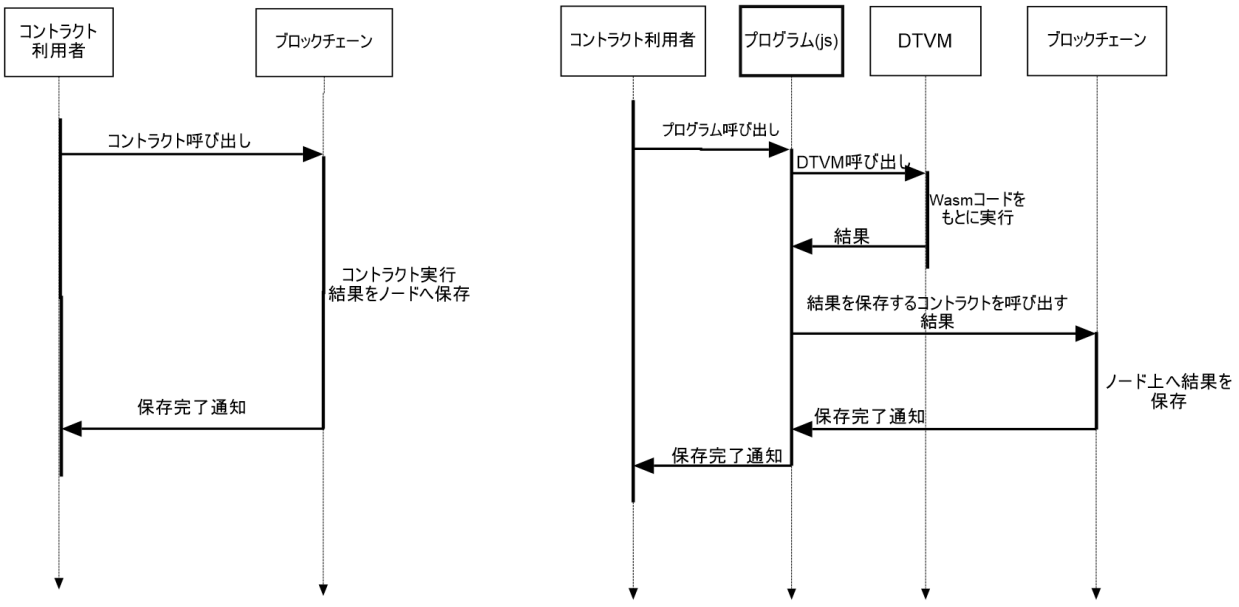
一方でデメリットとして考えられることが, まずセキュリティ性の低下である。本来ブロックチェーンとはスマートコントラクトの実行をすべてのノードで行い, その結果をお互いに監視しあうことでその正当性を検証し, データをブロックにして保存する。しかしこの仕組みではコントラクトの処理をローカルで実行し, その結果のみをブロックチェーン上で保存する。そのためローカルで処理をする部分を改ざんされたとしてもブロックチェーン上から気づくことができなくなっている。また, ローカルで実行した結果をブロックチェーンに接続する際のオーバーヘッドの大きさによっては実行時間全体が逆に長くなってしまう可能性も考えられ

る。

本研究では実行時間に焦点を当てて, この手法について評価する。

3.2 具体的な仕組みについて

本研究では hardhat というブロックチェーン開発環境を使用する。javascript で作成したプログラム内で DTVM に実行させる Wasm ファイルを指定して実行させ, あらかじめブロックチェーンにデプロイしておいたデータを保存するコントラクトを呼び出すことで,DTVM の実行結果のみをブロックチェーンに保存する。従来のフルオンチェーン型である図 3.1(a) と今回提案する仕組みである図 3.1(b) を比較すると, ブロックチェーンで動作する部分が少なくなっていることが分かる。本研究ではこのブロックチェーン上の動作が少なくなる事によるメリット, デメリットを評価する。



(a) フルオンチェーン　　(b) オフチェーン実行＋オンチェーン同期

図 3.1: 実行方式の比較

第 4 章

評価実験

本章では, 本研究で実施した評価実験の内容, 実験環境, そしてその結果と考察について述べる。

4.1 実験概要

本実験ではフィボナッチ数列を計算するプログラムを対象として,3 つのパターンでその実行速度, メモリ使用量, そして必要なガスコストを計測した。パターンは以下の 3 つである。

4.1.1 パターン 1:EVM によるフルオンチェーンでの実行

従来のフルオンチェーン型の例として,Solidity で書いたフィボナッチ数列のスマートコントラクトをブロックチェーン上で実行する。

4.1.2 パターン 2:DTVM によるオフチェーン実行 (元の言語は Solidity)

DTVM を用いた提案手法の仕組みを使い,Solidity で書いたプログラムを Wasm バイトコードにコンパイルして実行する。

4.1.3 パターン 3:DTVM によるオフチェーン実行 (元の言語は c++)

パターン 2 と同様に DTVM に実行させるが,C++ で書いたプログラムを Wasm バイトコードにコンパイルして実行する。

4.1.4 実験環境

windows 11 Home

プロセッサ Intel(R) Core(TM) Ultra 5 125U

実装 RAM 16.0 GB

システムの種類 64 ビット オペレーティング システム,x64 ベース プロセッサ

4.2 実験内容

フィボナッチ数列 30 40 を順番に計算するプログラムを対象に実行速度, 最大メモリ使用量, また数列 40 を計算して保存する際のガスコストを計測した。それぞれのパターンで 30 回プログラムを実行してその平均値で比較した。

4.3 実験結果

4.4 考察

第 5 章

おわりに

5.1　まとめ

まとめ

5.2　今後の課題

今後の課題。

参考文献

[1] K. Christidis and M. Devetsikiotis, *Blockchains and Smart Contracts for the Internet of Things*, IEEE Access, vol. 4, pp. 2292–2303, 2016. https://ieeexplore.ieee.org/document/7467408

[2] Ali Dorri, et al. *Blockchain for IoT Security and Privacy: The Case Study of a Smart Home*, Conference Paper · March 2017 https://people.cs.pitt.edu/~mosse/courses/cs3720/Blockchain_for_IoT_Security_and_Privacy.pdf

[3] Pooja Khobragade *On-chain Off-chain Blockchain Model for IoT using IPFS*, IET Conference Proceedings, July 2023 https://www.researchgate.net/publication/373003375_On-chain_off-chain_blockchain_model_for_IoT_using_IPFS

[4] G. Wood, *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, Ethereum Yellow Paper, 2014. https://ethereum.github.io/yellowpaper/paper.pdf

[5] Wei Zhou, et al. *DTVM: REVOLUTIONIZING SMART CONTRACT EXECUTION WITH DETERMINISM AND COMPATIBILITY*, June 10, 2025 https://arxiv.org/pdf/2504.16552v2

[6] Kenta Kawai, Wu Yuxiao, Yutaka Matubara, and Hiroaki Takada *BLOCKCHAIN-BASED DEMAND-SUPPLY MATCHING SYSTEM FOR IOT DEVICE DATA DISTRIBUTION* , 2024 https://aircconline.com/csit/papers/vol14/csit142408.pdf?utm_source=chatgpt.com