

令和元年6月27日現在

機関番号：26402

研究種目：基盤研究(C) (一般)

研究期間：2016～2018

課題番号：16K00103

研究課題名(和文) IoTデバイス向けの軽量でモジュラーなJavaScript処理系

研究課題名(英文) Lightweight and Modular JavaScript System for Internet of Things

研究代表者

鶴川 始陽 (Ugawa, Tomoharu)

高知工科大学・情報学群・准教授

研究者番号：50423017

交付決定額(研究期間全体)：(直接経費) 3,500,000円

研究成果の概要(和文)：Internet of Things(IoT)で用いられる組み込みシステム向けの軽量のJavaScript仮想機械(VM)を開発するために、アプリケーションに特化したJavaScriptサブセットのVMを自動生成するという方針で研究を行い、次の成果を得た。(1)演算子のオペランドの一部のデータ型だけを受け取るように特化したVMを自動生成する仕組みおよび最適化アルゴリズム。(2)コンパクションを行うごみ集めのアルゴリズムとその実装の検査手法。(3)弱いメモリモデルのCPU上での動作を検証するためのモデル検査ライブラリ。これらの成果はJavaScript VM生成系eJSTKに実装した。

研究成果の学術的意義や社会的意義

近年、アプリケーションの実行中にプロファイリングを行い、それに従って最適化しながらコンパイルするjust-in-time (JIT)コンパイルという技術を、VM自身に適用してアプリケーションに特化されたVMを実行時に構築する研究が成功を収めている。本研究は、それを組み込みシステムに応用するために、アプリケーション実行前にVMを特化させるという可能性を示した。また、VMの開発にいくつかのソフトウェア検証技術を導入し、それらの有用性を示した。

研究成果の概要(英文)：We developed lightweight JavaScript virtual machine (VM) suitable for embedded systems used in Internet of Things (IoT). Our approach was to generate a VM of a subset of JavaScript specialised for each application automatically. Our achievements include followings. (1) The mechanism and optimisation algorithms to generate a specialised VM that accepts restricted datatypes as operands of operators. (2) A compacting garbage collection algorithm and a verification method of its implementation. (3) A model checking library to check behaviour of algorithms on weak memory consistency model CPUs. We implemented those results in a JavaScript VM generator eJSTK.

研究分野：計算機科学

キーワード：仮想機械 組み込みシステム モデル検査 ごみ集め JavaScript

## 様式 C - 19、F - 19 - 1、Z - 19、CK - 19 (共通)

### 1. 研究開始当初の背景

ネットワーク接続されたセンサやコントローラを様々な物に取り付けてデータを収集したり制御したりする IoT(Internet of Things)という技術が注目され始めていた。IoT では小型デバイス(以下、IoT デバイス)を使って制御や通信を行う。具体的には、Raspberry Pi のような小型計算機や、マイコンが使われている。IoT デバイスのプログラムは、標準的には C 言語で開発する。しかし、C 言語は低級な言語であり IoT 応用のアイデアを試作する障害の原因となっていた。そのうえ、高級言語が普及し、日常的に C 言語でプログラミングするプログラマは限られていたため、参入のハードルが高かった。趣味のプログラミングを含めた、多くの人が参入して、様々なアイデアを試すことができれば、その中から良い製品が多く生み出される可能性があった。以上の理由から高級言語で IoT デバイスのソフトウェアを開発できるようにすることが有用であると考えた。

高級言語の中でも JavaScript はスクリプト言語であり、プログラムの試作に適していた。また、JavaScript は Web アプリケーションの記述言語として広く受け入れられており、プログラマが多い。加えて、JavaScript のイベント駆動型のプログラミングモデルは IoT デバイスのようなハードウェアの制御と相性がよいという利点もあった。

平成 27 年 10 月の国際会議で IoT デバイスに特化した JavaScript 処理系のデモンストレーションが行われており IoT デバイス向け JavaScript 処理系の研究は、世界的に見ても黎明期であると考えていた。

### 2. 研究の目的

IoT デバイスで JavaScript を利用するには、小型化が不可欠である。JavaScript は Web アプリケーションを手軽に開発できるようにするために、大きな仕様を持っている。その仕様を完全に実装する仮想機械 (VM) は大きくなってしまふ。しかし、IoT で使われるような組み込みシステムでは、ソフトウェアは一度システムが出来上がってしまうと変更されることがない。さらに、個々のアプリケーションは、JavaScript の仕様の一部しか使わない。そこで、各アプリケーションに特化した、JavaScript のサブセットだけをサポートする VM を作ることで VM を小さくする。このような VM をアプリケーション開発者自らが作れるような仕組みを作るために必要な技術開発の研究を行った。

### 3. 研究の方法

IoT デバイスで動作する軽量な JavaScript VM を自動生成する仕組みを開発するのに必要な、次の研究を行った。

- (1) JavaScript の仕様を記述するためのドメイン特化言語(DSL)の研究と、それを使って、アプリケーションが必要とする JavaScript の仕様の記述から、特化した VM を合成する方法およびその最適化の研究
- (2) 動的に生成されるデータを保持するためのメモリを削減するためのごみ集めの研究と、それをモジュール化して VM に実装する研究
- (3) IoT デバイスで用いられる ARM などの、弱いメモリモデルを持つ CPU でごみ集めのアルゴリズムが正しく動作することを確認するためのモデル検査の研究

これらの研究成果は、JavaScript VM 生成系である eJSTK として実装して評価した。eJSTK は研究代表者と共同研究者が科研費補助金(23500038)の補助で開発したサーバサイドアプリケーション向け JavaScript VM を組み込みシステム向けに改修して開発した。

### 4. 研究成果

本研究の成果は、以下に述べる各研究項目に対する研究成果、それを実装した eJSTK 処理系、および、その開発の中で用いたソフトウェアの検査手法の有用性を示したことである。

- (1) JavaScript の仕様を記述するためのドメイン特化言語(DSL)の研究と、それを使って、アプリケーションが必要とする JavaScript の仕様の記述から、特化した VM を合成する方法およびその最適化の研究

本研究では、JavaScript の演算子に与えられるデータ型に注目して研究を進めた。

動的データ型を持つ JavaScript では、どのデータ型の値を持つのが実行してみるまで分からない。そのため、VM を実装する際には、全てのデータ型の可能性を考慮して、(1)どのデータ型かを調べて適切な処理に分岐する処理(型ディスパッチ)と、(2)それぞれの具体的なデータ型に対する処理を各所で行う必要がある。特に JavaScript の演算子は、オペランドのデータ型によって異なる処理をする個所の代表例である。例えば、加算演算子(+)は、「1 + 5」のように数値をオペランドとして与えた場合、数値の加算を行うが、「"temperature: "+25」のように文字列と数値をオペランドとして与えた場合は数値を文字列に変換した後にそれらを連結する処理をする。この他にも真偽値や配列など、あらゆる型の組み合わせに対して動作が定義されている。しかし、個々のアプリケーションは、これらの全ての組み合わせを実行することはほとんどない。

そこで、次に挙げる技術を開発して、アプリケーション実行時に使われないデータ型に対す

る処理を省くとともに、簡素でコンパクトな型ディスパッチを持つ VM を自動生成する仕組みである eJSTK を開発した。

アプリケーションのユニットテストを利用して、アプリケーションが実行時にどのようなデータ型の演算をする可能性があるかを自動的に調べる技術

全てのデータ型を受けとる JavaScript VM の VM 命令(演算子は VM 命令にコンパイルされる)の仕様と、アプリケーションが必要とするデータ型の組み合わせを列挙した「オペランド仕様」から、VM を自動生成する技術

が生成する VM の型ディスパッチの最適化アルゴリズム

で用いる、VM 命令の仕様を記述するための DSL

これらの成果は、国内外のプログラミング言語に関する査読付きの会議や論文誌で発表した。特に、の成果は ACM 学会 SIGAPP 研究会の最も重要な国際会議 Symposium on Applied Computing (SAC 2018)に採択された。

この他にも、演算子に与えるデータ型の他にも次の点で VM をカスタマイズする機能を持つ。

- 対象のアプリケーションで頻繁に使われるデータ型を効率よく扱えるようにするために、ユーザが指定する「データ型仕様」を使ってデータ型の表現方法をカスタマイズする機能
- 対象のアプリケーションで頻繁に使われる命令の組み合わせを高速化に実行し、また、中間コードの圧縮をするために、ユーザが指定する「合成命令仕様」に基づいて命令を選択的に合成する機能

また、VM 命令の仕様を記述するための DSL は、VM 全体を記述できる新しい DSL の研究に発展している。

## (2) 動的に生成されるデータを保持するためのメモリを削減するためのごみ集めの研究と、それをモジュール化して VM に実装する研究

本研究では、eJSTK にコンパクションの仕組みを導入した。また、コンパクションで必要になる「ルートの登録」が正しく行われていることを、制御フローグラフに対するパターンマッチで検査できることを示した。

プログラムを長時間実行すると、動的にデータを生成する領域であるヒープの空き領域が徐々に断片化してしまう。その結果、プログラムによっては、使えない空き領域が増えてメモリ効率が悪くなる。そのため、長時間動作するシステムでは、コンパクションによって断片化を解消する必要がある。本研究では、コンパクションを行わないごみ集めと選択できるような形で、eJSTK にコンパクションを行うごみ集めを実装した。

コンパクションの実装では、VM のプログラム全体で JavaScript のデータを持つ変数を漏れなく GC が見える表に登録する必要がある。これは通常のプログラミングでは必要がない特殊なルールであり、忘れやすい。特に、eJSTK では、ユーザ自身が利用するハードウェアにアクセスするための組込み関数を追加することも想定しているが、eJSTK の開発者ではないユーザにとっては、ルールを間違えずに守ることはさらに難しい。そこで、ソースコードの制御フローグラフに対するパターンマッチで間違いを発見することを試した。その結果、非常に高い精度で誤りを発見することができた。

## (3) IoT デバイスで用いられる ARM などの、弱いメモリモデルを持つ CPU でごみ集めのアルゴリズムが正しく動作することを確認するためのモデル検査の研究

IoT デバイスは ARM など、デスクトップ用に普及している x86 とは異なる CPU を持つ。このような CPU では、弱いメモリモデルを採用しており、VM の、特にごみ集めの実装において x86 よりも注意を払う必要がある。

本研究では、モデル検査により、ごみ集めのアルゴリズムが弱いメモリモデル上で正しく動作するかを検査するためのライブラリである MMLib を開発した。このライブラリは、モデル検査器 SPIN 用のモデル記述言語 Promela で記述された、モデルの部品である。SPIN で弱いメモリモデルの CPU 上でのアルゴリズムの動作を検査するためには、CPU アルゴリズムだけでなく、CPU のモデルも作る必要がある。MMLib は CPU をモデル化しており、これを用いると、通常モデル検査で記述するモデルとほぼ同等の記述で、弱いメモリモデルの CPU 上での動作を検査できる。本研究では、MMLib のライブラリ本体に加えて、これにより発見した誤りを可視化し、原因を提案するシステムも開発した。この研究成果は情報処理学会の英文論文誌で発表した。

本研究によって、モデル検査を用いて弱いメモリモデル上のごみ集めアルゴリズムの動作を検証することが有効であることが示された。特に、ごみ集めの教科書にもとりあげられているアルゴリズムに誤りを発見した。その成果は、プログラミング言語の最高峰の国際会議である Object-oriented Programming, Systems, Languages, and Applications (OOPSLA 2017)に採択された。

さらに、本研究の手法を、デスクトップ向けのごみ集めとして開発していた Transactional Sapphire に適用し、いくつかの誤りを発見した。この経験を主要な貢献として含む論文は、

ACM 学会 SIGPLAN 研究会の最高峰の論文誌 Transactions on Programming Languages and Systems (TOPLAS) に採録された。

## 5 . 主な発表論文等

### [ 雑誌論文 ] ( 計 6 件 )

野中 智矢、鶴川 始陽: 組み込みシステム向け JavaScript 仮想機械生成系 eJSTK における合成命令の設計と実装、情報処理学会論文誌 プログラミング、情報処理学会、2019 (採録決定) (査読有り)

Kosuke Matsumoto, Tomoharu Ugawa: Visualization of Counterexamples of Memory Model-Aware Model Checking Using SPIN, Journal of Information Processing, IPSJ, 2019 (to appear) (査読有り)

Tomoharu Ugawa, Hideya Iwasaki, Takafumi Kataoka, eJSTK: Building JavaScript Virtual Machines with Customized Datatypes for Embedded Systems, Journal of Computer Languages, Vol. 51, pp. 261—279, Elsevier, 2019 (査読有り)

DOI: 10.1016/j.cola.2019.01.003

Tomoharu Ugawa, Carl G. Ritson, Richard E. Jones: Transactional Sapphire: Lessons in High Performance, On-the-fly Garbage Collection, Transactions on Programming Languages and Systems (TOPLAS), Vol. 40, Issue 4, Article No. 15, pp. 1—56, ACM, 2018 (査読有り)

DOI: 10.1145/3226225

Kosuke Matsumoto, Tomoharu Ugawa, Tatsuya Abe: Improvement of a Library for Model Checking under Weakly Ordered Memory Model with SPIN, Journal of Information Processing, Vol. 26, pp. 314—326, IPSJ, 2018 (査読有り)

DOI: 10.2197/ipsjip.26.314

Tomoharu Ugawa, Tatsuya Abe, Toshiyuki Maeda: Model checking copy phases of concurrent copying garbage collection with various memory models, Proceedings of the ACM on Programming Languages (PACMPL), Vol. 1, Issue OOPSLA, Article No. 53, pp. 1—26, ACM, 2017 (査読有り)

DOI: 10.1145/3133877

### [ 学会発表 ] ( 計 20 件 )

Tomoharu Ugawa, Hideya Iwasaki, Takafumi Kataoka: Collecting Type Information Using Unit Tests for Customization of JavaScript Virtual Machines, 14th Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS '19), 2019 (to appear) (査読有り)

高野 保真、鶴川 始陽、岩崎 英哉: JavaScript 仮想機械記述のためのドメイン特化言語、第 21 回プログラミングおよびプログラミング言語ワークショップ (PPL '19)、2019 (査読有り)

片岡 崇史、鶴川 始陽、岩崎 英哉: アプリケーションに特化した JavaScript 仮想機械開発のためのユニットテストを用いたプロファイリング、第 21 回プログラミングおよびプログラミング言語ワークショップ (PPL '19)、2019 (査読有り)

大林 建造、岩崎 英哉: eJSVM における対話的なプログラミング環境の実現、第 60 回プログラミング・シンポジウム、2019

Tomoharu Ugawa, Hideya Iwasaki: Generating a Minimum JavaScript VM Specialised for Target Applications, Workshop on Virtual Machines and Language Implementations (VMIL '18), 2018

藤本 太希、鶴川 始陽: C 言語プログラムにおけるコピーガベージコレクションの実装の誤りの検出、平成 30 年度電気関係学会四国支部連合大会、2018

野中 智矢、鶴川 始陽: 組み込みシステム向け JavaScript 仮想機械における合成命令の実装、平成 30 年度電気関係学会四国支部連合大会、2018

片岡 崇史、鶴川 始陽: アプリケーションの実行に必要な VM の機能を特定するためのプロファイラ、平成 30 年度電気関係学会四国支部連合大会、2018

Tomoharu Ugawa Tatsuya Abe, Toshiyuki Maeda: Model checking copy phases of concurrent copying garbage collection with various memory models, 日本ソフトウェア科学会 35 回大会, 2018 (招待講演)

森 翔太郎、片岡 崇史、鶴川 始陽: アプリケーション特化による組み込みシステム向け JavaScript VM 小型化の効果確認、日本ソフトウェア科学会 35 回大会、2018

Takafumi Kataoka, Tomoharu Ugawa, Hideya Iwasaki: A Framework for Constructing JavaScript Virtual Machines with Customized Datatype Representations, 33rd Annual ACM Symposium on Applied Computing (SAC '18), 2018 (査読有り)

赤澤 亮弥、鶴川 始陽、岩崎 英哉: 言語仮想機械におけるカスタマイズ可能なごみ集めモ

ジュールの実装、第 20 回プログラミングおよびプログラミング言語ワークショップ (PPL '18)、2018

野中 智矢, 片岡 崇史, 鶴川 始陽, 岩崎 英哉: arguments オブジェクトに着目した JavaScript コンパイラの最適化、第 59 回プログラミング・シンポジウム、2018

Kosuke Matsumoto, Tomoharu Ugawa: Modelling Acquire and Release Fences for Promela Models of TSO and PSO Using Store Buffers, 15th Asian Symposium on Programming Languages and Systems (APLAS '17) poster session, 2017

Takafumi Kataoka, Tomoharu Ugawa, Hideya Iwasaki: Generating Type-Based Dispatching Code for Intermediate Language Interpreter of Dynamic Typed Languages, 15th Asian Symposium on Programming Languages and Systems (APLAS '17), 2017

大石 孝史, 鶴川 始陽: 組み込みシステム用 JavaScript 仮想機械における RaspberryPi を制御する組み込み関数の実装、平成 29 年度電気関係学会四国支部連合大会、2017

片岡 崇史, 鶴川 始陽, 岩崎 英哉: 仮想機械における型ディスパッチャの自動生成、第 58 回プログラミング・シンポジウム、2017

松元 稿如, 鶴川 始陽, 安部 達也: メモリモデルを考慮したメモリアクセス命令を提供する SPIN 用ライブラリ、第 23 回ソフトウェア工学の基礎ワークショップ (FOSE 2016), 2016 (査読有り)

Tatsuya Abe, Tomoharu Ugawa Toshiyuki Maeda, Kosuke Matsumoto: Reducing State Explosion for Software Model Checking with Relaxed Memory Consistency Models, Symposium on Dependable Software Engineering Theories, Tools and Applications (SETTA '16), 2016 (査読有り)

中尾 成一, 鶴川 始陽: JavaScript の ANTLR 用文法記述のテスト、平成 28 年度電気関係学会四国支部連合大会、2016

[ その他 ]

ホームページ: <http://pl.info.kochi-tech.ac.jp/index.php/page-347/>

GitHub: <https://github.com/plasklab/ejstk>

## 6 . 研究組織

### (1)研究分担者

研究分担者氏名: 岩崎 英哉

ローマ字氏名: IWASAKI, hideya

所属研究機関名: 電気通信大学

部局名: 大学院情報理工学研究科

職名: 教授

研究者番号 (8 桁): 90203372

研究分担者氏名: 高野 保真

ローマ字氏名: TAKANO, yasunao

所属研究機関名: 青山学院大学

部局名: 理工学部

職名: 助教

研究者番号 (8 桁): 70780550

### (2)研究協力者

なし

科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。