

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成24年 6月12日現在

機関番号：12601

研究種目：若手研究（B）

研究期間：2009～2011

課題番号：21700024

研究課題名（和文）組込みシステム向けスクリプト言語処理系構成手法

研究課題名（英文）Scripting Language Interpreter for Embedded systems

研究代表者

笹田 耕一（SASADA KOICHI）

東京大学・大学院情報理工学系研究科・講師

研究者番号：10436561

研究成果の概要（和文）：組込みシステム向けスクリプト言語構成手法を明らかにするために、規模の大きなソフトウェアから自動的に必要な機能の取捨選択を行うためのコンパイラ構成法や、利用資源を抑えるために必要になるプロファイラの構成法などを明らかにした。

研究成果の概要（英文）：Our goal is to invent a method to implement scripting language interpreter for embedded systems. We have proposed a compiler technique to choose only required functions. And also we have developed a profiler for making a low resource consuming programs.

交付決定額

（金額単位：円）

	直接経費	間接経費	合計
2009年度	1,400,000	420,000	1,820,000
2010年度	900,000	270,000	1,170,000
2011年度	900,000	270,000	1,170,000
総計	3,200,000	960,000	4,160,000

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：プログラム処理系，計算機システム，組込みシステム

1. 研究開始当初の背景

近年、プログラム言語として、Java や C++ などといった静的コンパイル型言語ではなく、Perl, Python, PHP, Ruby に代表されるスクリプト言語を選択することが多くなった。この理由として、スクリプト言語によるプログラム記述の容易性や、提供されている機能・ライブラリの豊富さが挙げられる。これは、特にウェブアプリケーション開発において顕著である。このように、スクリプト言語は様々な場面で利用されている。

一方、組込みシステム開発では、社会的ニーズも年々高くなってきており、短期間に様々な要求に応える必要が出てきている。こ

のような短期間に多くの要求に応えるには開発効率の高さからスクリプト言語が適している。しかし、既存のスクリプト言語処理系はデスクトップやサーバ環境のような画一的で潤沢な計算機環境向けに開発されており、組込みシステムで利用される多種多様な計算機環境でそのまま利用することができない。既存のスクリプト言語処理系の組込みシステムでの利用を阻害する要因を次にまとめる。

まず、(1) 移植性、安全性が挙げられる。既存のスクリプト言語処理系は豊富な機能を利用可能にすることで開発効率を高めているが、組込みシステムが対象とするような特定用途向けのアプリケーション開発には不

要な機能を搭載し出荷することになり、移植性や安全性についての問題が生じる。次に、(2)リソース問題がある。スクリプト言語処理系は潤沢な計算機資源を利用できることを前提しているため、例えばメモリ使用量を抑える等の配慮はない。そのため、組込みシステムで利用されることがある資源制約のある環境での実行はできない。3番目に(3)リアルタイム機能の欠如が挙げられる。スクリプト言語には実時間システムを構築するための支援がないため、リアルタイムシステムを構築することができない。最後に、(4)組込み OS との連携の欠如がある。組込みシステムでは、OS とアプリケーションが密接に連携し、時には OS を改変することで効率的なシステムを開発することができるが、現在のスクリプト言語処理系にはそのような機能はない。

これらの問題により、現状ではスクリプト言語処理系を組込みシステムに利用するのは困難である。

2. 研究の目的

前述した (1)~(4)の問題を解決するため、本研究では既存のスクリプト言語処理系を組込みシステムで利用するために、次に示す点を解決するスクリプト言語処理系構成手法を解明する。

- (1) 機能の追加および削除を容易にするプラグイン、プラグアウト機能
- (2) 処理系のコンパクト化、省メモリ化
- (3) リアルタイム機能
- (4) 組込み OS と言語処理系の連携

(1)は、実行時、ビルド時に必要な機能、環境に依存する機能を取捨選択するための組込みシステム向けフレームワーク構築手法を解明し、問題点(1)移植性、安全性を向上する。そして、計算機環境に依存する機能を容易に実装、削除する仕組みを導入することで、組込みシステムへの柔軟な応用を可能にする。(2)はスクリプト言語処理系のフットプリントを解析し省メモリ化を行う。また、(1)により利用しない機能を削除することでさらなる軽量化を目指し、問題点(2)リソース問題を解決する。(3)はスクリプト言語に実時間制約を満たすための機能をサポートする仕様を追加し、スクリプト言語処理系にてリアルタイム機能をサポートする。また、リアルタイムガーベージコレクション (GC) を追加することで、問題点(3)で述べたリアルタイム機能を実現する。(4)は、組込み OS と連携を行う仕組みを解明し、OS の機能を利用したリアルタイム機能や機能のプラグイン、プラグアウト機能を実現する。また、組

込み OS に独自機能を組み込むことでさらに効率化を目指し、問題点(4)組込み OS と連携した効率的なスクリプト言語処理系の実現を行う。

上記構成手法を適用したスクリプト言語処理系 Ruby を、実際に資源制限があるような組込み機器上で動作させる実験を行い、(1)~(4)の結果を実証する。

3. 研究の方法

まず、初年度は次の2つのテーマについて取り組んだ。

(1) OS の機能を利用した組込み向け機能拡張

デスクトップ OS、サーバ用 OS などとは違い、組込みシステムでは POSIX システムコールなどの一般的な機能が利用できないことがある。そこで、移植性のある機能だけではなく、OS に特化した機能を用いて特定環境下での性能向上を実現するための試作を行った。今年度はコルーチン機能を提供する Fiber というクラスの実装を、POSIX の固有の機能、Windows の固有の機能を用いて、それぞれの環境下での性能向上を図った。また、移植性を確保するために、これら以外の環境では、性能はでないが移植性の高い手法を選択するようにした。

(2) 処理系のコンパクト化、省メモリ化

組込みシステムでは扱うことができる資源が限られているため、たとえばメモリ効率を高める工夫は重要である。そこで、まずは Ruby のメモリ管理手法を根本から検討し直し、メモリ管理効率を向上する主張を開発した。開発したメモリ管理手法を用いて、いくつかの例で性能向上、メモリ利用率の向上を確認した。また、Ruby プログラムでのメモリ利用状況を詳細に調査するためのメモリプロファイラを開発し、同じくプログラミング研究会で発表した。メモリプロファイラによって、メモリリークを検出し、その原因追及ができることを確認した。

昨年度では、これまで研究を続けてきたプロファイラを、さらに進化させた高性能な性能プロファイラを開発を行った。また、Ruby のスレッド機構の改善を行った。そして、プラグイン・プラグアウトを自動的に行うためのコンパイラの検討と開発を行った。

2年目は、1年目の成果を踏まえて、より現実的、実践的なスクリプト言語処理系構成法を研究するべく、性能プロファイラの実現に取り組んだ。また、スレッド機構の検討と、コンパイラ基盤を用いたプラグイン・プラグ

アウト手法の研究に取り組んだ。

(1) 性能プロファイラの実現

実行時オーバヘッドを極力少ない、リアルタイム性能プロファイラを開発した。また、計算機資源が少ないことが多い組み込みシステムで必須となるリモートプロファイリング機能を実現した。本性能プロファイラは、メソッド呼び出しの親子関係を可視化し、その呼び出しごとのプロファイリング結果を表示することができるようにした。

(2) スレッド機構の改善

Ruby のスレッド機構を改善し、Ruby インタプリタの省電力化を行った。Ruby インタプリタはタイマスレッドというものを用いるが、これが CPU を低速状態に保つ妨げとなっていた。そこで、タイマスレッドが不要な場合は停止するような実装を行い、実際に消費電力減となったことを確認した。また、複数 CPU 上でスレッドがスイッチしないという問題点について、スケジューリングに関する問題点を、スケジューリング機構に新しい仕組みを導入することで改善した。

(3) プラグイン・プラグアウト機構の検討と開発

コンパイラインフラストラクチャである LLVM を利用して、C 言語レベルで不要な仕組みをプラグアウトし、また必要になったら自動的にプラグインするコンパイラの検討と開発を行った。現状ではプラグイン・プラグアウトの最小単位は C 言語の関数単位で行うことができる。今後、これをさらに進展させ、柔軟なプラグイン・プラグアウト機構を実用化させたいと考えている。

最終年度は、1 年目、2 年目の成果をさらに深化させるべく、プロファイラフレームワークの開発、および関数に着目した C コンパイラによるモジュール分割機構の提案を行った。

(1) 複数の処理系に対応できるスクリプト言語処理系向けプロファイラフレームワークの開発

実用的なソフトウェアを開発するためにはプロファイラが重要であるが、スクリプト言語およびその処理系向けのプロファイラは開発されてこなかった。そこで、我々は複数のスクリプト言語に対応できるプロファイラフレームワークを開発した。フレームワークは

(a) プロファイルモジュール、(b) 情報サーバ、(c) 情報ビューアの3つで構成され、プロファイルモジュールを各言語処理系向けに対応させれば、(b)、(c) は既存のものを利用することができる。(a) と (b)、(b) と (c) はネットワークで接続されており、(c) は JavaScript の動作するウェブブラウザを用いて実現しているため、例えば (a)、(b) を組み込み機器、もしくはクラウド上に配置し、(c) を手元で実行する、といったことが可能である。開発したプロファイラフレームワーク、およびいくつかのプロファイラは llprof という名前で公開した。

(2) 関数に着目したCコンパイラによるモジュール分割機構の提案

C プログラムは関数の集合として実現されているが、この関数に着目し、関数単位で分割し、再グループ化するCコンパイラの開発を行った。本機構を用いることで、従来は手動で共有リンクライブラリの構成を指定しなければならなかったが、プロファイル情報などを用いて自動的にプログラムを分割することが可能になる。我々はLLVMを用いてこの分割・再グループ化機構を実装し、いくつかのプログラムで分割の効果を確認した。

その他にも組み込みシステムでも有用な VMM や分散 FS などの研究を行った。

4. 研究成果

本研究では、目標とした4つの項目のうち、次の3つ、つまり(1) 機能の追加および削除を容易にするプラグイン、プラグアウト機能は関数に着目したCコンパイラによるモジュール分割機構によって、(2) 処理系のコンパクト化、省メモリ化を行うために、必要なメモリ量の解析を行うためのとして、メモリプロファイラや複数の処理系に対応できるスクリプト言語処理系向けプロファイラフレームワークの構成手法を解明し、(4) 組み込みOSと言語処理系の連携について、OSの機能を利用したFiberやスレッドの向上方式を明らかにすることで、込みスクリプト言語処理系での構成方法について、その一部を明らかにすることができた。

残る目標である、(3)リアルタイム機能については、本期間中に明らかにすることが出来ず、今後の課題である。

本研究課題に取り組むにあたり、提案したソフトウェア構成手法を検証するため、実際に様々なソフトウェアを開発した。その成果

は、プログラミング言語 Ruby の処理系に実際に取り込む、またオープンソースソフトウェアとして公開するといった方法で、実社会への貢献を行った。

今後は、未解決の問題である動的言語処理系におけるリアルタイム性の確保をどのように行うか、またプロファイラでの解析により得られた結果をどのように省資源化に活用するか、明らかにしていく。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計7件)

- ① 荒川 淳平, 笹田 耕一: 分散ファイルシステムに適した分散アクセス制御機構, 情報処理学会論文誌 コンピューティングシステム (ACS38) (採録決定) あり, 2012 (査読有り).
- ② 高橋一志, 笹田 耕一: WinKVM : Hybrid Hypervisor 移植の一例, 情報処理学会論文誌 コンピューティングシステム (ACS36) , Vol.5, No.1 (2011), pp.27-40 (査読有り).
- ③ 須永 高浩, 笹田 耕一: Ruby 用リアルタイムプロファイラ的设计と実装, 情報処理学会論文誌 (PRO) , Vol.4, No.3 (2011), pp.1-15 (査読有り).
- ④ 荒川淳平, 笹田 耕一, 竹内郁雄: 実用的な分散アクセス制御機構, 情報処理学会コンピュータシステム・シンポジウム論文集 2010, 2010, pp.47-56 (査読有り).
- ⑤ 笹田 耕一: Ruby のスレッド実装の改善, 第 52 回プログラミングシンポジウム報告集, Vol. 1 (2011), 145-152, (査読無し).
- ⑥ 芝 哲史, 笹田 耕一: Ruby1.9 での高速な Fiber の実装, 第 51 回プログラミング・シンポジウム予稿集, pp. 21--28 (2010) (査読無し).
- ⑦ 高橋 一志, 笹田 耕一: WinKVM : 異なるホスト OS 間の VM ライブマイグレーション実現に向けて, 情報処理学会コンピュータシステム・シンポジウム論文集 (ComSys 2009), Vol. 2009, No. 13, pp. 59--66 (2009-11) (査読有り).

[学会発表] (計6件)

- ① 須永 高浩, 笹田 耕一: Ruby 用リアルタイムプロファイラ的设计と実装, 情報処理学会 第 82 回 プログラミング研究発表会, 2011 年 1 月 20 日, 宮古島市 中央公民館 (沖縄県宮古島市).
- ② 高橋一志, 笹田 耕一: KVM を利用した異種ホスト OS 間上でのライブマイグレーション, 情報処理学会「システムソフト

ウェアとオペレーティング・システム」 SWoPP 2010, 2010 年 7 月 6 日, 金沢市文化ホール (石川県金沢市).

- ③ 上野康平, 笹田 耕一: 軽量なプロセスマイグレーションを可能とするフレームワーク, 情報処理学会「システムソフトウェアとオペレーティング・システム」 SWoPP 2010, 2010 年 7 月 6 日, 金沢市文化ホール (石川県金沢市).
- ④ Tetsu Soh, Koichi Sasada, Design and Implementation of Memory Profiler for Ruby, 情報処理学会 第 78 回プログラミング研究発表会, 2010/3/16, 電気通信大学
- ⑤ Koichi Sasada: Ruby Memory Management Hacks, RubyConf 2009, 2009/11/20, Embassy Suites San Francisco Burlingame, California, USA
- ⑥ 笹田 耕一: Ruby のメモリ管理の改善, 情報処理学会 第 76 回プログラミング研究会, 2009/10/28, 石垣市商工会館.

6. 研究組織

(1) 研究代表者

笹田 耕一 (SASADA KOICHI)

東京大学・大学院情報理工学系研究科・講師
研究者番号: 10436561