



2009 年度上期未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当PM

首藤 一幸 PM(東京工業大学 大学院情報理工学研究科
数理・計算科学専攻 准教授)

2. 採択者氏名

チーフクリエイター: 小室 直(電気通信大学大学院
電気通信学研究科 情報工学専攻)
コクリエイター : なし

3. プロジェクト管理組織

株式会社オープンテクノロジーズ

4. 委託金支払額

3,000,000 円

5. テーマ名

スクリプト言語 Lua の高速化と省メモリ化

6. 関連Webサイト

<http://hogel.org/>

<http://github.com/hogelog/ostacklua/>

7. テーマ概要

私はスクリプト言語 Lua のメモリ管理機構を改良し、高速性と省メモリ性を向上させアプリケーション組み込み言語の利用用途を広げることを提案する。

Lua はアプリケーション組み込みスクリプト言語としての利用を想定している比較的シンプルで小さなプログラミング言語である。高速性、移植性において優れていることから商用・非商用のビデオゲームなどでの利用も多い。しかしそれでも C や C++といった言語により丁寧に記述されたプログラムにその性能が及ぶものではない。そのため速度や省メモリ性能が厳しく問われ Lua のようなスクリプト言語を利用できない場面も多い。よって本提案では Lua をより広い場面で有効に活用するため、Lua 処理系の高速化と省メモリ化に取り組む。

GC による自動メモリ管理を前提としたプログラミング環境ではほとんどのオブジェクトはヒープに格納されるため、手動でスタックに割り当てたオブジェクトと比較すると実行コストは大きくなりやすい。ヒープ割り当てオブジェクトの中から安全にスタック割り当て可能なオブジェクトを探すエスケープ解析は動的な言語においては実装が非常に困難である。

よって本提案ではまずは全てのオブジェクトをスタックに格納し、自分よりも底に近いオブジェクトかヒープから参照されたときにヒープにオブジェクトを移動するスタックベースのメモリ管理機構を実装する。

Lua のアプリケーション組み込み言語としての利用が広まっているのはクローズドソフトウェアでも問題無く組み込むことのできる緩いライセンス形態が大きな理由の一つだと考えられる。よって提案ソフトウェアそのものも元の Lua 処理系本体と同様 MIT ライセンスの OSS として公開・整備していく予定である。

8. 採択理由

組み込み用途でよく使われているプログラミング言語 Lua の処理系を改良して、高速化と省メモリ化を図ろうという提案である。

具体的には、もともとはヒープに割り当てるオブジェクトを楽観的にスタックに割り当てることで、ガーベジコレクションの処理軽減とヒープ消費量の削減を図ろうという、論文[Eric06]の手法を実装する。

ビデオゲーム等の中で実際によく使われるようになった Lua 処理系を効率化できれば、恩恵を受ける人も多いことだろう。オープンソースソフトウェアである Lua 処理系の本流にその成果を貢献していく過程で提案者が得るものもとても大きいだろう。

一方で、見方によっては、この提案は既存・既知の手法の1つを Lua 処理系に実装するというものに過ぎない。ヒープ割り当てを減らすという一点でどの程度の高速化

が可能か、という疑問も残る。

なので、提案書にあるスタックへの割り当て手法[Eric06]の実装と効果検証などはさっさと済ませてしまい、その先、つまり、手法の改良、別の手法の調査・実装・検証、自分なりの手法の提案など、+α を打ち出すことに力を注いで欲しい。どこまで行けるか、楽しみにしている。

9. 開発目標

既存の Lua 実行系を対象として、スタックベースメモリ管理機構 [1] を実装し、メモリ管理コストおよびプログラム実行性能の向上を図る。

10. 進捗概要

途中で実装の方法を変更するなど、紆余曲折あったが、期間中に機構の実装を完了した。性能については、現状、実行性能が向上するプログラムもある、という状況である。今後、自身で取り組むとともに他の開発者を巻き込んで、向上を図っていく。

11. 成果

ファイルディスクリプタを多く使うマイクロベンチマークとして作成した FileLoop でのメモリ使用量、ファイルディスクリプタ使用量のグラフを図2と図3に示す。



図2:メモリ使用量



図3:ファイルディスクリプタ使用量

また、資源の量が厳しい組み込み環境の代表として、ゲーム機 ニンテンドーDS 上での動作実験を行った。通常の Lua 実行系ではファイルディスクリプタ不足で動作しないプログラムが動作した、という結果を得た。ニンテンドーDS で動作している様子を図4に示す。



図4:ニンテンドーDS で動作している様子

12. プロジェクト評価

7 月に未踏ユースとしてのプロジェクトが始まった。開始当初は、9 月中には動作させるという予定が示されていた。しかし 12 月に入っても動作に至らない。クリエイタに実装するだけの腕があることは PM として確信しており、そうでなければ採択はできなかった。年が明けた頃によりやく動き始め、それから 2 月半ばの成果報告会までは発表の準備、特にデモの準備に力を入れてもらった。

開発が遅れた理由は、実装方針の模索に時間がかかり、開発開始から数カ月経っ

て方針を変更したためである。9 月中に実装可能性を見極めよ、という指示を出し、それに対して、およそ目処が付いたという回答があった。しかし結果として、年末にかけて実装方針を変更することになったのは、クリエイター、PM ともに見通しが甘かったと言わざるを得ない。

当初の提案プラス α を模索するという目論見は達成できなかったが、クリエイターは持ち前の実装力を活かして、期間内には提案通りの機能を実装し終えた。また、すでにオープンソースソフトウェアとしての公開も済んでいる。なんとか、最低限の目標は達成した。

13. 今後の課題

より多くのプログラムの性能を向上させられるように、実装の改善を行っていく。そのためには Lua 開発者への広報活動も重要となる。

[1] Erik Corry. Optimistic stack allocation for Java-like languages. In J. Eliot B. Moss, editor, ISMM'06 Proceedings of the Fourth International Symposium on Memory Management, pp. 162—173, Ottawa, Canada, June 2006. ACM Press.