



2010 年度 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: 井出 真広(横浜国立大学大学院)

コクリエイター: なし

3. 委託金支払額

1,782,400 円

4. テーマ名

JavaScript の GCC フロントエンドの開発

5. 関連Webサイト

なし

6. テーマ概要

JavaScript は Web ブラウザ上に実装されているスクリプティング言語で、ほとんどのすべての Web ブラウザに実装されている。JavaScript の実行性能の向上により、デスクトップアプリケーションと遜色ない、大規模 Web アプリケーションを実現することが可能となったため、JavaScript は Web アプリケーションを構築する際には必要不可欠な存在となって来ている。

クライアントサイドの Web アプリケーションはそれぞれ基盤となるライブラリ、フレームワークの上に構築され、その上にそれぞれの Web アプリケーションのためのコードが記述される。ここで、各アプリケーションはブラウザ上で実行される際に、ライブラリ、フレームワークはアプリケーションごとに複製されて実行されるため、メモリ効率があまりよくないと言える。

本提案では、JavaScript の GCC フロントエンド gccjs の開発を行う。

gccjs を用いてプログラムの実行前に JavaScript コードを機械語へコンパイルすることで、アプリケーション間でコードを共有し、プログラムサイズを小さくでき、レスポンス性能が高く、ロード時間の短い Web アプリケーションの構築につながるのではないかと考えている。

7. 採択理由

今日、Web ブラウザ上で動作するプログラムは、おしなべて JavaScript 言語で書かれるようになった。

世界中が、実行時 (JIT) コンパイルという方式で、JavaScript プログラムの高速化にしのぎを削っている。

提案は、実行前にコンパイルを済ませてしまう (AOT) コンパイラを開発しようというものである。

AOT コンパイラには JIT コンパイラとは違ったメリット (例えばコードのプロセス間共有) があり、また、実際に開発することで、その性能、使い勝手等も明らかになる。JavaScript のような言語でコンパイルして性能を高めることは易しくないが、井出君はそれを理解し、対策を検討している。

達成は決して容易ではないが、言語処理系の開発経験もあり、自らの動機に基づく提案なので、期待している。

8. 開発目標

開発ソフトウェアは、大きく分けて、GCC フロントエンドとランタイムライブラリから成る。それらそれぞれが、また、次の通りの各部分から構成される。

- GCC フロントエンド (GCCJS)
 - JavaScript パーサ
JavaScript ソースコードを読み込み、GENERIC tree の生成までを行う。
- ランタイムライブラリ
 - メモリ管理機構
 - オブジェクト管理機構

- ライブラリ呼び出し機構
- コード評価器

フロントエンドはまた、若干の最適化を行う：

- 定数量み込み
- オブジェクトのフィールドを表現するハッシュ値のコンパイル時計算
- メソッド呼び出しの高速化
- 一部変数の定数への置き換え

9. 進捗概要

およその JavaScript プログラムを実行できるコンパイラおよびランタイムライブラリを開発した。ただし、対象プログラムについて次の仮定を置いている：

- 関数、メソッドは実行中には変更されない。
- プロトタイプは実行中には変更されない。
- eval によって破壊的な変更は行われない。

また、数値を含めた基本型の値も内部ではオブジェクトとして表現する。これに起因する性能上のオーバーヘッドがある。

10. プロジェクト評価

プロジェクト提案時より、JavaScript 言語では AOT より Just-in-Time (実行時) コンパイラの方が最終的な性能では勝るだろうと予測していた。しかし、AOT コンパイラにも別の利点 (例：OS によるページ共有の働きやすさ) があり、また、AOT ではどの程度の性能を達成できるかを明らかにすることには意義がある。

結果としては、AOT の性能限界を (ごく限られたプログラムについてだけでも) 示すところまでは至らず、また、AOT やこのコンパイラならではの利点を示すにも至らなかった。プロジェクトは道半ばということである。言語処理系は開発が難しい類のソフトウェアであり、未踏ユースの開発期間中にゴールに至らなかったことに無理はない。

11. 今後の課題

まずは限られたプログラムについてだけでも、AOT コンパイラの性能限界を示して欲しい。