静的型付けスクリプト言語 KonohaScript の HPC 利用に向けて

井出真広[†] 若松悠樹[†]
平岡祐太郎[☆☆] 倉光君郎[†,☆]

1. はじめに

近年，スクリプト言語の適用領域として大規模科学計算 (HPC) 分野にも注目が集まっている．

HPC 分野にスクリプト言語を適応するにあたって，実行速度が問題となる．MPI を実装した既存のスクリプト言語としては Ruby，Python などがあるが，いずれも動的型付けの特徴によって C 言語で記述されたプログラムにくらべ低速である．

本稿では我々が開発を行なっている静的型付けスクリプト言語 KonohaScript を用いた HPC プログラムにむけて言語処理系の拡張と MPI ライブラリのバインディングの設計と実装を行った．また，行列積プログラムを用いて他のスクリプト言語, C 言語と実行性能の比較を行った．

2. KonohaScritpt 言語処理系

KonohaScript[1)☆☆☆]は我々が開発を行なっている静的型付けオブジェクト指向スクリプト言語である．KonohaScript では，整数，浮動小数点数などの変数の型をプログラム中に明示できるため，C/C++ や，Java ユーザからも比較的書きやすい言語になっている．また KonohaScript ではコンパイル時型検査や型情報に基づきバイトコードとバイトコード評価機を実装しており，高性能な演算処理を実現することができる．

我々は KonohaScript の HPC 利用を行うためにまず MPI の通信 API をスクリプトから利用することができるよう KonohaScript による並列分散処理のための MPI 拡張 mpikonoha の設計, 実装を行った．mpikonoha では MPI の通信 API を KonohaScript の API としてバインディングし，KonohaScript の並列プロセス間でデータの送受信を可能とする．

さらにプログラムの実行性能の向上を行うために，型付けされたプログラムをプログラム実行中にネイティブコードを生成する Just-in-Time(JIT) コンパイラ KonohaJIT の設計，実装を行った．KonohaJIT では機械語の生成に LLVM[3)] を用いてコード生成を行う JIT コンパイラである．KonohaJIT ではスクリプトのソースコードから抽象構文木を生成し，さらに抽象構文木から LLVM Intermediate Representation (LLVM IR)[2)] に変換する．そして例えば共通部分式の除去といったコンパイラ最適化を LLVM のバックエンドを用いて実行し，最後に機械語を生成する．

CPU	Intel Xeon X5690 3.46GHz * 6cores * 2
RAM	24GB (DDR3 1333 4GB * 6)
インターコネクト	InfiniBand QDR x1 (8Gbits/sec)
OS	CentOS 5.6 (Kernel 2.6.18-274.17.1.el5)
C コンパイラ	gcc 4.1.2
MPI ライブラリ	OpemMPI 1.4.3
ノード数	4

表 1 計算環境

3. 評　価

本節では予備的な評価実験として他の MPI を実装した言語と，KonohaScript (VM，JIT) について実行速度を計測した．ベンチマークプログラムとして，C 言語で実装された MPI 通信を行う NxN 行列積を求めるプログラムを各言語に移植したプログラムを用いて実験を行った．実験は**表 1** に示す環境で行い，また使用した言語，MPI ライブラリについては**表 2** を用いた．なお，KonohaScript の MPI ライブラリバインディングの mpikonoha ライブラリを用いている．並列プロセス数 NP を 2 から 48 の範囲で設定し，ベンチマークプログラムを並列実行した．

実験結果を**図 1** に示す．縦軸はそれぞれの MFOLPS を示しており，また横軸は並列実行したプロセス数を示している．

実験結果より，いづれの場合も KonohaScript は

† 横浜国立大学大学院
Graduate School of Yokohama National University
☆ 現在，日本科学技術振興機構 CREST
Presently with Japan Science and Technology Agency/CREST
☆☆ 現在，富士通
☆☆☆ KonohaScript: http://konohascript.org/

測定言語	MPI バインディング
Konoha	mpikonoha
Konoha(JIT)	mpikonoha
Ruby 1.8.5	mpi_ruby 0.4
Python 2.7.2	mpi4py 1.3
C 言語	-

表 2 測定言語とバインディング

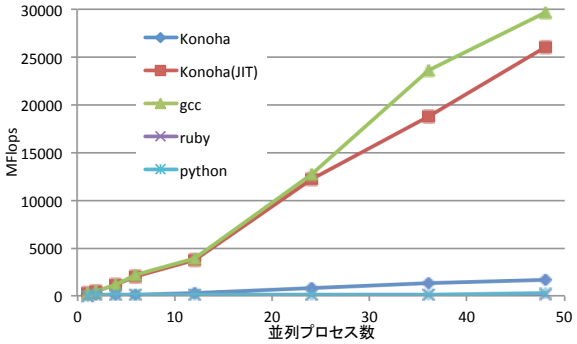

図 1 NxN 行列積 (N=1920) 実効性能の言語間の比較

JIT コンパイラを用いた場合，C 言語で実行した場合と等しい実行性能を実現することが確認でき，また，並列プロセス数が 48 の時には 85% の実行性能を確認することができた．

4. ま と め

本稿ではスクリプト言語 KonohaScript の HPC 利用に向けて MPI ライブラリのバインド，JIT コンパイラの設計，実装を行った．

その結果，行列積計算において C 言語で記述した場合と同等の性能が得られることが確認できた．今後は得られた実験結果の分析を行い、また中規模，大規模プログラムに対しても KonohaScript で実装を行い，性能評価を行いたい．

謝　　辞

本研究は，JST/CREST「実用化を目指した組込みシステム用ディペンタブル・オペレーティングシステム」領域の研究課題「実行時の安全性を確保する SecurityWeaver と P-SCRIPT」の一部として行われた．

参 考 文 献

1) Kimio Kuramitsu. Konoha - implementing a static scripting language with dynamic behaviors. In *Workshop on Self-sustaining Systems 2010*, S3, The University of Tokyo, Japan, September 2010.
2) Chris Lattner. Llvm language reference manual. `http://llvm.org/docs/LangRef.html`.
3) Chris Lattner and Vikram Adve. LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. In *Proceedings of the 2004 International Symposium on Code Generation and Optimization (CGO'04)*, Palo Alto, California, Mar 2004.