TSUBAME 共同利用　平成28年度　学術利用　成果報告書

利用課題名　先進的ステンシルコード技術
英文：Advance Stencil-Code Engineering

利用課題責任者
千葉　滋
Shigeru Chiba

所属
東京大学　情報理工学系研究科創造情報学専攻
Grad. School of Information Science and Technology
The University of Tokyo
http://www.csg.ci.i.u-tokyo.ac.jp

邦文抄録（300 字程度）
高度なステンシル計算のためのドメイン専用言語（DSL）の開発を行い、その実行時性能を TSUBAME 上で検証するとともに性能改善をおこなった。本研究課題は JST CREST/独 DFG による日独共同研究 SPPEXA のフレームワークの中で実施している。本研究課題では、埋め込み型ドメイン専用言語を開発するための手法として我々が研究している deep reification と呼ぶ手法の評価を TSUBAME 上でおこなった。実験にあたり deep reification を Java 言語で可能にするように我々が開発した Bytespresso を用いた。

英文抄録（100 words 程度）
We have developed a domain specific language (DSL) for high-performance stencil computing and examined and improved its execution performance on TSUBAME. This work was performed under the framework of the joint research program SPPEXA of JST CREST and German DFG. This work evaluated our technique called deep reification for developing an embedded domain-specific language. For our experiment, we used Bytespresso, which is our software that enables deep reification in Java.

Keywords: domain specific languages, reflection, deep reification, stencil computing, Java.

背景と目的

ステンシル計算は多くの科学技術分野で頻繁に使われる計算でこれまでも様々なドメイン専用言語やライブラリが開発されてきた。一方で、ドメイン専用言語はせまい応用範囲（ドメイン）のために新規にコンパイラやライブラリ、開発環境を整備しなければならず、開発コストの大きさが課題である。多数のアプリケーションプログラムが見込めるドメインであれば大きな開発コストをかけてドメイン専用言語を開発しても見合うが、少数のアプリケーションプログラムしか見込めないドメインの場合、ドメイン専用言語の開発コストが見合わないことがある。

開発コストの総和
＝ドメイン専用言語の開発コスト
＋アプリケーションの開発コストの和

であり、アプリケーションプログラムの開発コストはドメイン専用言語を用いることで低減することが期待されるが、低減された開発コストがドメイン専用言語の開発コストを上回っていなければならない。このため、より応用範囲の広いドメイン専用言語を開発することになるが、応用範囲を広く取れば取るほどその言語は汎用言語に近づいていき、ドメイン専用言語の利点を活かしにくくなる。つまりそのドメインのプログラマから見て自然な抽象機構の提供や、そのドメイン特有の知見を生かした実行の最適化などの利点である。

ドメイン専用言語の開発コストという課題を解消する手法として注目されているのが埋め込み型ドメ

イン専用言語（Embedded Domain Specific Languages）と呼ばれる手法である。この手法では、ドメイン専用言語は汎用プログラミング言語（ホスト言語とよぶ）のライブラリとして実装される。ライブラリではあるが、そのライブラリを用いた書かれたホスト言語のプログラムが、あたかも専用の言語で書かれたかのように見えるよう、ライブラリのプログラミング・インタフェースを注意深く設計する。プログラミング・インタフェースの自由度は、そのホスト言語の構文の範囲内に制限されるが、近年登場した新しい汎用プログラミング言語は構文の自由度が高く、そのようなライブラリの設計が可能になってきた。

埋め込み型ドメイン専用言語は、ホスト言語のライブラリであるが故に、開発コストが比較的小さくできるが、実行時性能は必ずしも満足いくものではなかった。そのドメイン固有の実行時性能最適化技法などの適用が難しくなりがちだからである。汎用言語では、プログラムがどのように書かれていても、言語仕様に定められたとおりの計算をおこなわなければならない。このため適用できる最適化に制限が加わりがちである。

本研究課題は、最新のプログラミング言語の技術を適用することで、高い実行時性能をもつ埋め込み型ドメイン専用言語の開発手法を明らかにすることを狙っている。ドメイン専用言語には、ドメインのプログラマから見て自然な抽象機構を提供する利点と、ドメイン知識を用いた高速実行を可能にする利点があり、従来技法では前者を達成していたものの、後者の利点が不十分だった。本研究課題では、埋め込み型ドメイン専用言語で両方の利点を生かせるようにすることを狙い、ホスト言語に Java 言語を用い、対象ドメインとしてステンシル計算を例として研究をおこない、手書きの C 言語プログラムに匹敵する性能を達成できることを示した。

概要

高度なステンシル計算のためのドメイン専用言語 (DSL) の開発を行い、その実行時性能を TSUBAME 上で検証するとともに性能改善をおこなった。ステンシル計算の利用者にとってなじみやすい行列表現など抽象度の高い記述によるプログラムを、計算機に近い表現で記述されたプログラムに匹敵する速度で実行可能にすることを目指す。本研究課題は JST CREST/独 DFG による日独共同研究 SPPEXA のフレームワークの中で実施している。このフレームワークでは、ステンシル計算のための外部 DSL と内部 DSL を開発している。主に外部 DSL では記述能力と性能を、内部 DSL では利用の容易さを追求する一方、それぞれが苦手する部分の改善をおこなう。TSUBAME を用いることで、日本最大級の GPU 搭載スーパーコンピュータで実用的な性能を達成する DSL の実現手法の研究をおこなう。

本研究課題では、埋め込み型ドメイン専用言語を開発するための手法として我々が研究している deep reification と呼ぶ手法の評価を TSUBAME 上でおこなった。実行時性能の高いドメイン専用言語を実現するには、プログラムを抽象構文木ないしはより低レベルな中間表現に変換し、それを用いてより高速実行できる形に変換することが必要である。埋め込みドメイン専用言語でもこれを可能にするため、deep embedding のような技術が開発されてきたが、プログラマの視点からは自然な形のプログラムになりにくいという欠点があった。これを補う手法 LMS も提案されているが、Scala 言語のような高度な型システムを持った言語でなければ利用できない。Deep reification は実行中にラムダ式などの形で与えられたプログラムの断片に対して、対応する構文木を返す reflection 機構を用意し、それを用いて埋め込み型ドメイン専用言語を実現しようというアイデアである。Lisp 系の言語に見られるマクロ機構によく似ているが、そのプログラムの断片の中で呼ばれている関数、メソッドについても構文木を芋づる式に取り出せる点が異なる。

我々は deep reification を備えた処理系 Bytespresso を Java 言語をホスト言語として実装した。Bytespresso も Java 言語で書かれたライブラリであり、Reifier クラスを提供する。このクラ

スの snap メソッドを下のように呼ぶと、引数 cm で渡されたメソッド・オブジェクトを与えられた引数 args の下で実行した場合の抽象構文木が返される。

```
CtMethod cm = ...
Object[] args = { ... };
Reifier reifier = new Reifier(cm, args);
Reifier.Snapshot image = reifier.snap();
```

変数 image は、snap が取り出した抽象構文木の他に内部で使われている型や、そこから参照されているオブジェクトを保持する。抽象構文木はその中から呼ばれている他のメソッドの抽象構文木を含むので実際には木ではなく森となる。Bytespresso は抽象構文木の取り出しにあたって Java バイトコードを逆コンパイルして抽象構文木を作成する。したがってソースコードがなくとも抽象構文木を取り出すことができる。

Bytespresso を用いた埋め込み型ドメイン専用言語として、ステンシル計算をおこなう簡単なドメイン専用言語を開発した。この言語は Java をホスト言語とし、次のようなプログラムを書くことができる。

```
grid.initialize(cInitPressure)
  .each(Reduction.SUM, 1, Predicate.FOREVER,
      new Kernel() {
          public float newValue(FloatArray3D p,
                  Cursor cur, int t, Reduction r) {
            float s0 = cur.self(a0) * cur.east(p)
                    + 略
            float ss = 略
            r.apply(ss * ss);
            return cur.self(p) + omega * ss;
        }
      }).repeat(N, drv);
```

書かれたプログラムは C 言語プログラムへと変換され、MPI 環境で実行できる。ドメイン専用言語であるので、MPI 環境での実装の詳細はプログラマからは隠されている。

結果および考察

Bytespresso で実装したドメイン専用言語の実行時性能を評価するために、上述のステンシル計算用の言語を TSUBAME 上で実行した。ベンチマークプログラムとして Himeno ベンチマークの問題サイズ XL (512x512x1024)を用いた。我々はこのベンチマークプログラムを我々のステンシル計算言語に移植し、比較には C 言語で書かれた元のベンチーマークプログラムを用いた。それぞれの MPI ノードはシングルスレッドで単精度計算を行った。コンパイルには Intel C compiler を -Ofast オプションを付けて用い、MPI 環境には OpenMPI 1.8.2 を用いた。最大で 256 ノードを用いた。

実行結果を図 1 （Shigeru Chiba, YungYu Zhuang, Maximilian Scherr, “Deeply reifying running code for constructing a domain-specific language”, PPPJ’16, ACM 2016 より引用）に示す。図の横軸は MPI プロセス数を表す。8x8x4 のような注釈は 3 次元のグリッドの分割数を表す。図が示すように Bytespresso が Java プログラムから生成する C 言語プログラムの実行速度は非常によい。元の C 言語プログラムよりも高速なのは、Bytespresso が採用している配列のメモリ配置が本ベンチマークの計算により適していたためと思われる。

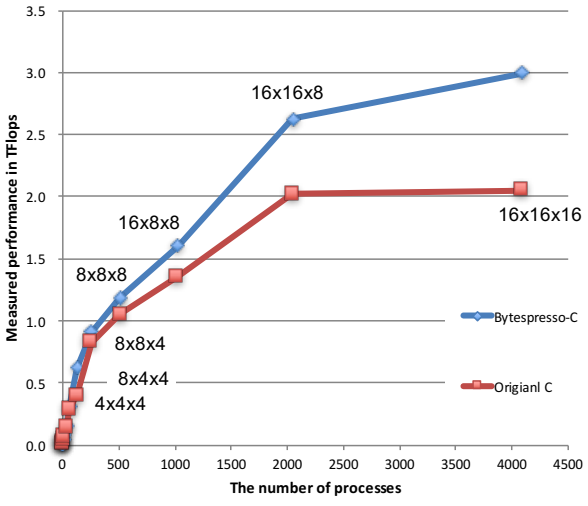

図 1. Strong scale performance of the Jacobi kernel of Himeno on TSUBAME2.5.

まとめ、今後の課題

本研究課題では、我々が研究開発した deep reification の実効性を確かめるため、TSUBAME を用いた性能測定をおこなった。Java 言語プログラムの中で deep reification を可能にするシステムとして Bytespresso を開発し、簡単なステンシル計算用のドメイン専用言語を作成し、実行速度を測定した。その結果、人手で書いたプログラムに遜色ない実行速度が達成できることを示した。

今後の課題はドイツ側の共同研究者が開発している外部ドメイン専用言語である ExaSlang を deep reification を用いて埋め込み型ドメイン専用言語に移植することである。移植したものの実行時性能を元の ExaSlang の実行速度と比較し、より実用的な状況での deep reification の効果を議論する。