

5W-2

Datarol プログラムの評価

立花 徹 谷口 倫一郎 雨宮 真人
九州大学大学院 総合理工学研究科

1. はじめに

著者らはデータフローアーキテクチャの問題点を解決するアーキテクチャとして、Datarolマシンアーキテクチャを提案している。Datarolとは、関数型プログラムからデータ依存関係を基に抽出されたマルチスレッド・コントロールフローであり、Datarolに沿って並列演算実行を制御するハードウェアがDatarolマシンである⁽¹⁾⁽²⁾。Datarolプログラムは、データフロー・プログラムからペア・オペランドの存在チェックやゲート演算などの余分なデータフロー制御を取り除いて最適化したプログラムと考えることができる。

本論文では関数型プログラムのような非手続き的なプログラムからデータ依存解析により抽出されたDatarolプログラムを、データフロー・プログラム並びに手続き的プログラム(LISP,C)のコンパイルコードと比較し、Datarolプログラムが有効であることを示す。特に、関数型プログラムからデータ依存解析によって抽出したDatarolプログラムが同一処理を行なう手続き型プログラムのコンパイルコードと比肩できることを明らかにする。

2. コンパイラの概要

関数型プログラムは、次の3つのステップでDatarolプログラムにコンパイルされる⁽³⁾。

- 関数型プログラムを中間言語表現⁽¹⁾に変換する。
- (A)で得られたデータ依存関係からデータ依存解析により、マルチスレッド・コントロールフローを抽出する。
- それぞれの変数にレジスタを割付ける。

(B)のコントロールフローの抽出によって、データフローグラフでは必要であるが、Datarolプログラムでは必要ないパスを除去し、オペランドマッチングを減らすことができる。また、(C)のレジスタ割付けでは、それぞれの変数がその命令の実行以後では使われないということが保証できるような命令を求め、その命令以降ではその変数に割付けられたレジスタを他の変数に再割付けすることにより必要レジスタ数を最小化する。

3. Datarolコード vs データフロープログラム

この節ではDatarolプログラムとデータフロープログラムの命令数、及びペアオペランドマッチング数の比較を行なう。例として、要素aが集合bに含まれるならtrue(1)、含まれないならfalse(0)を返す関数memberを挙げる。memberは関数型言語Valid⁽⁴⁾で図1(a)のように書ける。これからDatarolプログラムを抽出するとノード数は25、ペアオペランドマッチング数は3となる。一方データフロープログラムの命令数は17、ペアオペランドマッチング数は8であり、ともにDatarolプログラムの方が少ない。ただし、

データフロープログラムにはsink命令に伴うゴースト・トークンの処理のための(同期の)命令を含めていないので、実際はこのデータより更に大きい値である。Datarolプログラムではレジスタを用いるのでこのような処理を必要としない。この他にも、集合演算、N-Queen、行列の演算、フィボナッチ数列、階乗、総和等を求める問題を解く関数型プログラムからDatarolプログラムを抽出し、それぞれの関数型プログラムに対応するデータフロー・プログラムとの命令数(図2(b))、ペアオペランド・マッチング数(図2(c))について比較を行なった。データフロープログラムでは条件分岐のtrue partとfalse partへのデータの供給に必要なTFスイッチ(及びそれに伴うSINKノード)のために命令数が多い。従ってTFスイッチを多く含むデータフロープログラムはDatarolプログラムより命令数が多く、TFスイッチが少ないデータフロープログラムはDatarolプログラムと命令数の差は小さくなっている。

4. Datarolコード vs 逐次コード

この節では簡単なValidのプログラムについて、Datarolプログラムと手続き型プログラムのコンパイルコードとを比較するために、同様のプログラムをLISP及びCで手続き的に書き、それを68020用コードにコンパイルしたものの命令数、必要レジスタ数の比較を行なう。

memberをLispで図1(b)のように書き、HCL(Hokkaido Common Lisp⁽⁵⁾)のコンパイラにかけるとその命令数は26となる。また、memberをCで図1(c)のように書き、それをコンパイルすると(E-7300,Cコンパイラ)命令数は24になる。しかしここではnull, car, cdrを関数として呼び出

```
function member(a,b:list) return (bool)
= if null(b) then false
  elseif a = car(b) then true
  else member(a,cdr(b));
```

(a) Valid

```
(defun member (a b)
  (if null(b) nil
    (if (= a (car b)) t
        (member a (cdr b))))))
```

(b) LISP

```
member(a,b){
  while (!(null(b))) {
    if (a == car(b) return 1;
    else b = cdr(b)}}
```

(c) C

図1 memberのプログラム

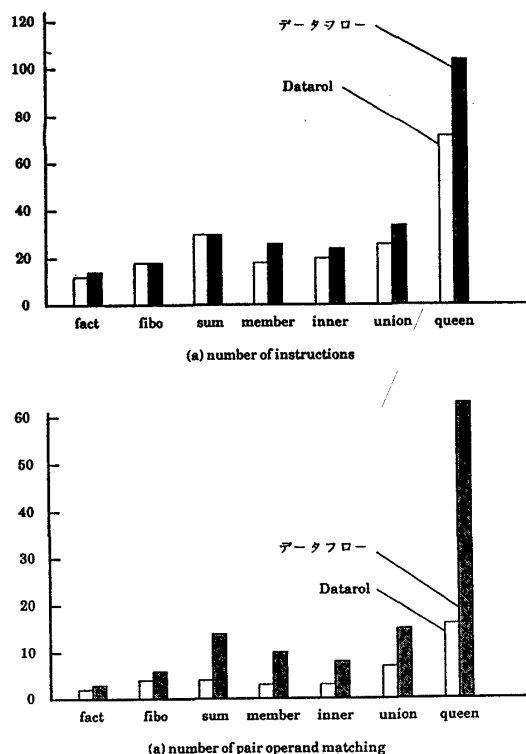


図2 Datarolとデータフロープログラムの比較

している（Datarolおよびデータフロー・プログラムではこれらはプリミティブな命令として扱っている）Datarolプログラムと比較するにはそのサブルーチン呼び出しのためのスタック操作等（*で示してある）を除外しなければならない。そうすると命令数は19となる。

この他にも、3節と同じ例題のDatarolプログラムと68020コード（ソース言語:Lisp,C）との命令数（図3(a)）、必要レジスタ数（図3(b)）の比較を行った。

<Lispのコンパイル・コードとの比較> 全ての関数についてDatarolプログラムの方が命令数が少ない。これは、関数型プログラムを逐次型のコードにコンパイルすると関数呼び出しの際のスタック操作が必要となるためである。また、レジスタ数は、ほぼ同程度になっている。

<Cのコンパイル・コードとの比較> sum, queenを除きほぼ同程度の命令数になっていることが分かる。sumはDatarolプログラムでは分割統治法を用いているため命令数が多くなっている。また、queenはCでも再起呼び出しを用いているのでスタック操作のためにCのコンパイル結果の命令数が多くなっている。これらの例から、Datarolプログラムの命令数が、Cのコンパイルコードと同程度であることがわかる。また、レジスタ数は、2から3倍になっており、十分現実的な数字であると考えられる。

5. おわりに

本論文では、Datarolプログラムとデータフロー・プログラムについてノード数、オペランド・マッチング数を比較し、Datarolプログラムはデータフロー・プログラムの欠点を大幅に改善したコードであることを示した。また、Datarolプログラムと、(1)関数型プログラムを逐次型マシン用にコンパイルしたコード、(2)手続的プログラムを逐次型マシン用にコンパイルしたコード、との比較を行ない、Datarolの命令数が(1),(2)の命令数と同程度であることを

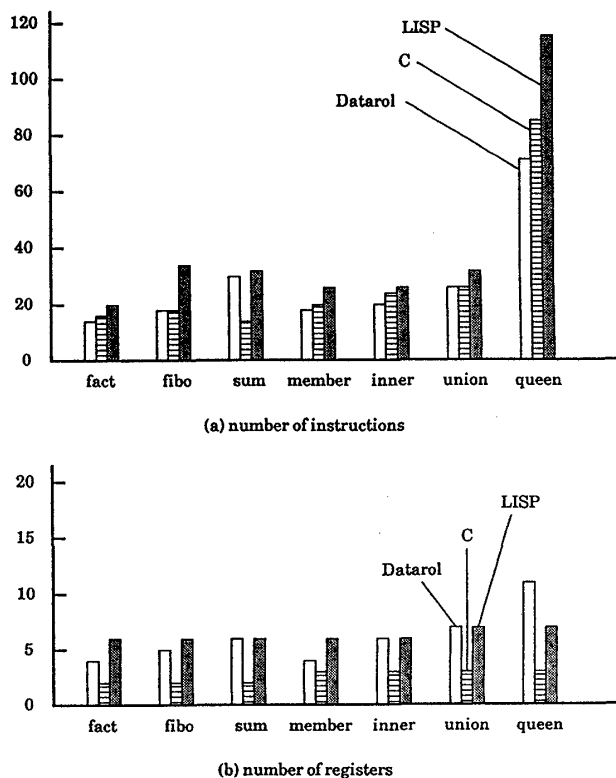


図3 DatarolとLISP,Cの比較

示した。このことは、Datarolマシンに於ける各関数インスタンスの実行速度が逐次型マシンと同程度であることを示す。従って、Datarolマシンは関数インスタンスを並列に実行するという点で逐次型のマシンに比べて有利であるので、全体的にみてもDatarolマシンが高い実行速度を持つ可能性があることを示している。即ち、Datarolマシンは、比較的遅いと言われている関数型言語の処理系の中で、逐次型のマシンに比肩し得る能力を持つといえる。また必要レジスタ数は、Cのコンパイル結果と比較しても2から3倍程度に収まっており、必要レジスタ数という点からみても現実的であるといえる。

今後はコンパイラを完成させ、より多くの現実的なプログラムについてコンパイルのアルゴリズム、及びDatarolプログラムの評価を行なっていく。特に比較的大規模な関数における必要レジスタ数を調べ、各レジスタファイルに必要なレジスタ数を決定しなければならない。

【参考文献】

- 1) 雨宮:超多重並列処理のためのプロセッサ・アーキテクチャ, 情報処理学会「コンピュータアーキテクチャ」シンポジウム, pp.99-108(1988).
- 2) 上田,谷口,雨宮:Datarolプロセッサのアーキテクチャについて, 情報処理学会第38回全国大会, pp.1406-1407(1989).
- 3) 立花,谷口,雨宮:データフロー解析による関数型プログラムのコンパイル法, 情報処理学会「並列処理」シンポジウム(1989).
- 4) 雨宮,長谷川,小野:データフロー計算機用高級言語Valid, 研究実用化報告, Vol.33, No.6, pp.1167-1181(1984).
- 5) 山本:北海道大学コモン・リスプ(HCL)リリース・ノート, version 1.0(1988).