

## 手続き間定数伝播処理の検討

6E-3

草野 和寛<sup>†</sup>, 妹尾 義樹<sup>†</sup>, 島村 正司<sup>††</sup><sup>†</sup>NEC C&C システム研究所,<sup>††</sup>NEC コンピュータシステム(株)

## 1 はじめに

手続き間解析は、プログラム全体の情報を用いて最適化を行なうために用いられる手法である。この中の一つである手続き間定数伝播[1]は、プログラム中で静的に確定可能な値を持つ変数とその値を見つけるための手法である。これにより、並列化で重要なループ長などをコンパイル時に決定することが可能になる。

本稿では、従来の手続き間解析の問題点とそれを解決できるようにした試作システムについて述べる。

## 2 手続き間解析

本章では試作システムにおける手続き間解析処理の概要と、その中の手続き間定数伝播処理について述べる。

## 2.1 全体構成

従来の手続き間解析は次の手順で処理されている(図1参照)。

1. 手続き毎の解析を行ない、その解析情報をデータベースにファイルとして格納する。
2. 手続き毎の解析結果であるデータベースファイルから、手続き間のインタフェース情報(以下、手続き間解析情報)を作成する。
3. 手続き毎のデータベースファイルに手続き間解析情報を加えて、手続き毎の再解析を行なう。

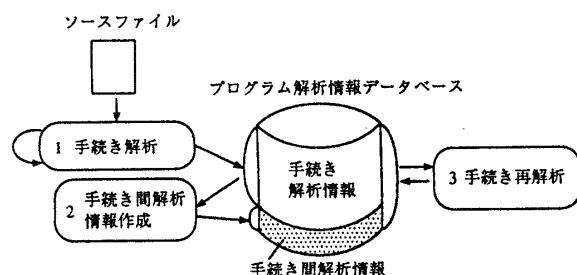


図1: 手続き間解析の解析手順

手続き間解析の効果としては、CALL文を含むDOループの並列化が可能になること、全ての手続き呼び出

し点で常に同じ値を持つ引数(またはコモン変数)とその値がわかることなどがある。

本システムで行なっている手続き間解析は、1) 定数伝播、2) 依存関係解析、3) ワーク変数の認識、の3つである。この解析のために、以下の情報をプログラム全体から収集する。

- 手続きの呼び出し点での引数とコモン変数の値の定数情報
- 手続き中の配列のアクセス範囲情報
- 手続き中の変数のUSE, MOD 情報

## 2.2 手続き間定数伝播

手続き間定数伝播では、プログラム中のある手続きを呼び出している全ての呼び出し点において同一の値を持つ引数とコモン変数を伝播対象にする。伝播値は、手続きの入口点での確定値として手続き内部の定数伝播処理で利用する。この解析により、ループ長、配列のサイズなどを引数で渡している場合でも、全ての呼び出し点で同じ値の場合には、その値が決定可能になる。

また、手続きの出口(リターン文など)において定数である引数とコモン変数の値を手続き間解析情報の一部として、呼び出し手続き側で利用可能にしている。

## 3 手続き間解析での提案

## 3.1 手続き間解析の反復適用

手続き間解析の情報には、手続きの呼び出し関係をアークで示したコールグラフを葉(leaf)から根(root)の方向(以下、上位レベル方向)へ伝達される情報と根(root)から葉(leaf)の方向(以下、下位レベル方向)へ伝達される情報がある。上位レベル方向へ向かう情報には変数や配列のアクセス情報があり、下位レベル方向へ向かう情報には呼び出し点の定数伝播情報がある。これらの情報は相互に影響しており、ある情報が変更された場合、それに対応して他の情報が変更される可能性が存在する。

したがって、手続き間の解析をコールグラフの上位、または下位レベル方向に順に処理する方が、ランダムに処理するよりも効率が良い。これは、同方向に伝達される情報は、直前に解析した手続きの解析結果が利用できるためである。しかし、前章の手順では、逆方向に伝達される情報の解析結果が他の手続きで利用できない[2]。

そこで我々は手続き間解析を繰り返して行なう手法

を開発し、手続き間解析の結果を、再び他の手続きにフィードバックできるようにした。本システムでは図2のように収束判定を行ない、手続き間解析を繰り返す。収束判定は、手続き毎の再解析前に集めた手続き間解析情報と、再解析後に集めた情報を比較して、変更されていなければ収束したと判定する。

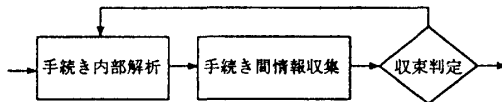


図2: 手続き間解析の繰り返し手順

本手法を用いることで、並列化などの解析で利用可能な解析情報を増やすことができる。例えばこれまで解析できなかった、引数で渡されるループ長などの定数値を新たに決定できることなどがある。次節では手続き間解析を繰り返す行なうことで解析可能になる例をあげる。

### 3.2 適用例

図3のプログラムを考えてみる。この例で、SUB2

```

PROGRAM MAIN      SUBROUTINE BB(IX, K, ...)
  IX = 0           IF (IX.EQ.0) THEN
  K = 10           K = 5
  CALL BB(IX, K, ...) CALL SUB1(...)
  ...             ELSE
  CALL SUB2(K, ...) CALL SUB2(K, ...)
  ...             ENDIF
  END              RETURN
                  END
  
```

図3: プログラム例

はMAINとサブルーチンBBから呼ばれている。BBでのSUB2の呼び出しは条件文(IF文)に含まれており、引数IXの値により実行の可否が決定される。

最初集める手続き間解析情報で、BBの引数IXとKはそれぞれ定数値0と10を持つことが判る。この時点では、SUB2の引数KはBB内部で書き換えられる可能性があり、不定である。手続き毎の再解析では、BBではSUB2は呼ばれないことと、KがBBの出口で定数値5を持つことが判る(SUB1内でKへの書き込みはないものとする)。

この後に再び手続き間解析情報を集めると、SUB2の呼び出しはMAINのみで引数Kは定数値5を持つことが新たに判る。この変更により手続き間解析情報は収束していないと判断し、もう一度手続き毎の解析を行なう。この手続き毎の解析では、直前に集めた手続き間情報である、SUB2の引数Kが定数値5を持つことをSUB2の解析に利用できる。

以上述べたように、手続き間解析を2回行なうことでSUB2の引数Kに定数値5を伝播させることができ

るようになった。この例で手続き間解析を1回行なうだけではSUB2の引数に定数値5を伝播させることができないのは、コールグラフの下位レベルへ伝達した情報により、上位レベルへ伝達する情報が変化するためである。

### 3.3 オーバヘッド

手続き間解析を繰り返す行なった場合のオーバーヘッドについて考察する。

試作システムで測定したところ(テストプログラムはLinpack)、手続き間解析処理が1回と2回の場合の時間の比率は3:5であった(EWS4800を使用)。この結果では、手続き間解析の解析回数にほぼ比例した解析時間がかかっている。これは、現在の試作システムでは解析情報データベースの量を減らすために解析情報全ては保持しておらず、手続き間解析の度に再解析するデータが存在することや、各手続きの再解析でデータベースからの入出力が必要になるためであると考えられる。

図3の場合には、2回の手続き間解析が必要であった。例と同様な呼び出し関係がもう一段深くなった場合には、もう一度手続き間解析が必要になる。また、手続き間解析に要する時間はプログラムによって大きく異なり、時間がかかる場合もある。したがって、手続き間解析を繰り返す回数は制限するべきであると考えられる。

このように手続き間解析を繰り返す場合、繰り返す回数を減らすことと処理対象の手続きを減らすことが重要になる。繰り返す回数は前に述べたように制限しておけばよい。そして、繰り返す場合の処理対象手続きを減らすには、手続きの解析で影響のある手続き間解析情報を判定するような処理が必要になる。また、必要な解析データ全てを保持し、処理量を減らすようなデータベースを構築することも重要である。

## 4 おわりに

以上本稿では、手続き間解析の反復適用する手法を述べた。今後は、提案した手法の効果検証を行なっていく予定である。

最後に、本研究の機会を下さいましたNEC C&Cシステム研究所の山本所長、小池部長、中崎課長、また、有益な助言を下さいました皆様方に感謝致します。

## 参考文献

- [1] David Callahan, Keith D. Cooper, Ken Kennedy, Linda Torczon: *Interprocedural Constant Propagation*, SIGPLAN '86 Sym. on Compiler Construction, pp.152-161, 1986.
- [2] Mark N.Wegman, F. Kenneth Zadak: *Constant Propagation with Conditional Branches*, ACM Trans. on Programming Languages and Systems, Vol.13, No.2, pp.181-210, 1991.