

3F-6

生産順序キューマシン命令コード生成手法の提案

川島 祐介 繁田 聡一 吉永 努 曾和 将容
電気通信大学 大学院情報システム学研究所

1 はじめに

我々は、式評価の中間結果を FIFO(First In First Out) メモリに貯えるキューマシンの研究を行っている。キューマシンは命令に対しオペランドの指定が必要ないためプログラムサイズが小さくなる。また、隣接する命令間に依存関係が発生しにくいいため、命令レベルでの高い並列性が期待できる。しかし、このモデルは取り出して使用するキューの特性から、演算における中間値を複数回使用するプログラムには不向きである。

そこで、我々は参照の概念を含めることでキュー計算方法の制約を緩和する計算モデルを提案している。その1つが「生産順序キューマシン」である。生産順序キューマシンとは、生産されたデータを順番にキューに格納し、過去に生産されたデータをキューヘッド以外でも参照可能にすることで、キューデータを複数回使用できるようにしたモデルである。このキューヘッド以外を参照できる命令を参照命令と呼ぶ。本稿ではこの生産順序キューマシン用の命令コード、特に参照命令の生成手法を提案する。

2 キュープロセッサ

キューマシンとは、FIFO の記憶装置キューを中間結果格納用として使って計算を行うモデルである。キューマシンで動くプログラムはキューレジスタを用いて実行される。キューレジスタとはデータをキューテール(QT) から入れ、キューヘッド(QH) から取り出す高速記憶装置である。キュープログラムの例として、図1に構文木とそこから得られるプログラムを示す。命令オペランドにレジスタ指定がないのが特徴であり、命令の実行順序は構文木を幅優先探索を行うことで求めることができる。

また、図1のプログラムを実行した場合のキューは図2のようになる。キューマシンにおいて、キューレジスタのデータを取り出す位置と格納する位置は、プ

ロセッサ内部で動的に求めることができる為、データの依存関係がなければ命令の並列実行が可能になる。

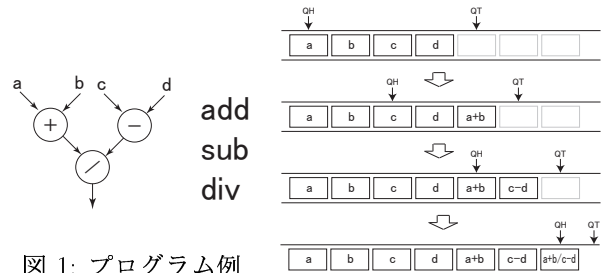


図 1: プログラム例

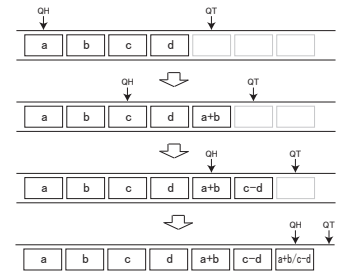


図 2: キューレジスタの変移

2.1 生産型キューマシン

同一のデータを生成している複数の命令がある場合、参照を利用した命令に変更することで同一データを生成している命令を削除できる。

そこで、QH 以外のデータも参照して命令実行できるモデルを提案した。これを生産型キューマシンと呼ぶ。参照をおこなう例を図3に示す。

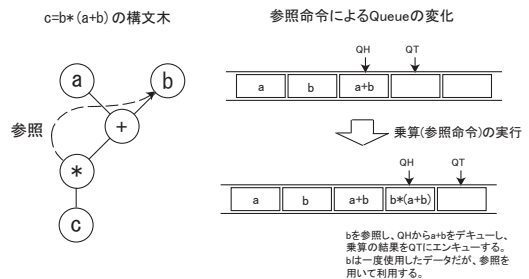


図 3: 参照命令の例

3 生産順序型キューマシン命令コードの生成手法

本稿では、生産消費型の命令コードを生産型の命令コードに変換する手法を提案する。生産型キューマシン命令の生成は、生産消費型の構文木を変換することで行い、工程として、冗長命令を発見、削除し、参照距離を設定する。

Proposal of Code Generation Scheme for a Producer-order Queue Machine
Yusuke Kawashima, Soichi Sigeta, B. A. Abderazek, Tsutomu Yoshinaga, and Masahiro Sowa
The Graduate School of Information Systems, University of Electro-communications

3.1 冗長命令の検出

同一のデータをキューに生成する冗長命令を検出する。冗長命令の例として図4のような場合がある。

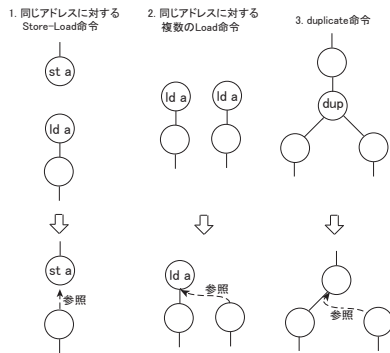


図 4: 冗長命令の検出例

冗長命令は大きくわけて2種類あり、メモリから同じデータをロードする場合と、キューで複製を作る場合である。どちらの場合も参照命令を用いることで、冗長命令を削除できる。

3.2 参照距離の算出

参照命令はオペランドとして参照場所を必要とし、その値はQHからの相対位置で指定する。これを参照距離と呼ぶ。図3の場合、乗算における参照距離は-2となる。

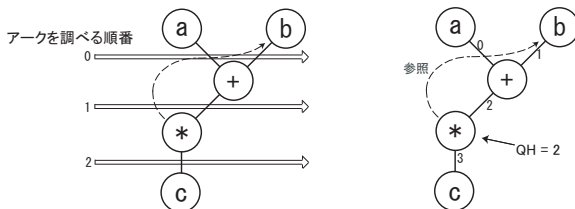


図 5: 参照距離の算出方法

参照距離を算出する為には、「参照データの位置」と「参照命令実行時のキューヘッド位置」が必要になる。

参照距離を算出する方法として、構文木の枝に番号をつける。番号は図5のように左上から右下へ1レベルずつ下がりながら左から右へと調べながらつけていく。これは、各枝の親によってデータが作成される順番であり、同時にデータのキュー位置と一致する。そのため、参照される枝番号を調べることで、参照データの位置がわかる。

次に、参照命令実行時のキューヘッド位置を求める。これは、その命令が実行される際に、何番目のデータを消費するかを調べればいいため、先に求めた枝番号を利用する。ノードが親と接続されている枝のうち、

番号が一番若いものが命令実行時のキューヘッド位置になる。

この2つの値の差分をとることで、参照距離を求めることができる。

4 評価

SciMark2のベンチマークから生産消費型と生産型のコードを生成し、静的評価を行った。その結果を1と2に示す。

FFT	生産消費型	生産型
命令数	2980	2759
キュー使用量	3157	2928

表 1: FFTによるプログラムサイズとキュー使用量の比較

LU	生産消費型	生産型
命令数	1064	1009
キュー使用量	927	872

表 2: LUによるプログラムサイズとキュー使用量の比較

参照化を施すことで、FFTでは7%、LUでは5%程度の命令を削減できた。また、FFTでは7%、LUでは5%程度キュー資源を節約できた。

5 まとめ

生産型キューマシンコード生成において、生産化の手法を提案し、検証を行った。その結果、参照命令を有する生産型キューマシンコードは生産消費型キューマシンコードと比べ命令数を削減できることが確認できた。

今後の課題として、動的評価による検証がある。命令数の削減により、プログラム実行速度の向上にもつながると考えられる。

参考文献

- [1] 菊池遊, 吉永努, 曾和将容, "キュー計算モデルを用いた並列プロセッサの設計", 信学会 情報・システムソサイエティ大会講演論文集, p.44, (2001).
- [2] Ben A. ABDERAZEK, 繁田聡一, Kirilka NIKOLOVA, 吉永努, 曾和将容, "並列キュープロセッサの基本設計", 電子情報通信学会, 信学技法 (CPSY) vol.102, No.478, pp.55-60, (2002).
- [3] A. Markovskij, M. Sowa, S. Shigeta, and B. A. Abderazek, "Producer-order Parallel Queue Processor Architecture Design", 信学技報 (CPSY) (2004).