

論理型言語向き並列計算機KPRにおける FGHCの処理方式について

1 L - 1

鹿毛裕史、柴山 潔、萩原 宏

(京都大学・工学部)

1. はじめに

KPRは、論理型言語プログラム(論理プログラム)の実行モデルとして我々が提案した「並列リダクション・モデル」に基づいて設計されたマルチプロセッサ・システムである。[1]

本稿では、AND並列論理型言語FGHCのプログラムのKPRにおける処理方式について示し、その処理過程について述べる。

2. 並列リダクション・モデル(PRモデル)

KPRの実行モデルである「並列リダクション・モデル(PRモデル)」は、AND/OR木の論理的AND関係にあるノードにはStream(S)プロセスを、OR関係にあるノードにはOr(O)プロセスを割り付ける。Sプロセスは、ストリーム並列処理方式で本体ゴール列を処理し、疑似的にAND並列性を実現する。

3. KPRにおける論理型言語の処理方式

3.1 KPRの対象言語KPR-L

KPRの対象言語であるKPR-Lは、pure-Prologを基本とし、これにいくつかの並列性制御機能の付加を含む拡張を施したものである。KPR-Lは、論理プログラムのOR並列性を積極的に利用するOR並列型言語であり、本体ゴールのAND並列性に関しては、逐次評価を基本として処理する。

3.2 Flat GHC(FGHC)の言語機能

GHCはAND並列型言語であり、AND関係にある本体ゴールの評価を並列に行い、これらゴール間の通信・同期は共有変数を介して行われる。Flat GHC(FGHC)とは、GHCのガード部においてユーザ定義述語を書くことを許さず、組み込み述語のみに限定するものである。従ってガード部の環境がネスト構造を取らず、トラスト中に処理中の節を放棄させる処理が簡単に行い得るという点で、FGHCはGHCに比べてKPRで処理を実現するのが容易である。

3.3 FGHCのプログラム処理方式

FGHCのプログラムをKPR上で実行する際に重要となるのが、(1)トラスト機構の実現と、(2)AND関係にあるゴールの処理である。(1)に関し

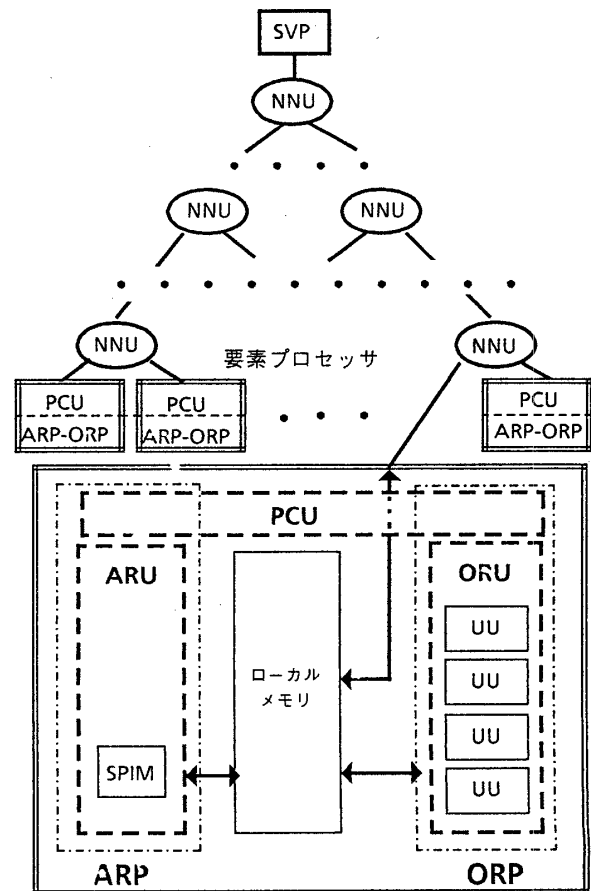


図1 KPRのシステム構成

ては、複数の候補節に対する節頭部の単一化とガード部の実行を並列に行う。つまりKPRにおける純粋なOR並列処理によってこれに対処することにする。(2)に関しては、処理すべきAND関係にあるゴールを待ち行列につないでスケジューリングする事により、AND並列性を疑似的に実現することにする。

この2つの処理を実現させるためのFGHCプログラムの処理方式として、まずFGHCのプログラムをKPR-LプログラムにコンパイルしてからKPRのマシン命令列にさらにコンパイルするか、もしくはFGHCプログラムを直接KPRのマシン命令列にコンパイルするかのいずれかの方法が考えられる

が、前者は処理効率の面で難点があるので、後者の方式を採用することにした。

4. プログラム実行方式

4.1 KPR-Lの実行方式

KPRは2分木ネットワーク結合方式のヘテロジニアス・マルチプロセッサ・システムであり(図1)、その葉ノードには要素プロセッサが接続され、根ノードにはシステム全体の管理を行うSVP(SuperVisory Processor)が接続される。また、ネットワークの中間ノードはNNU(Network Node Unit)と呼ばれる。KPR-Lのプログラムは、そのSVPによって静的情報を抽出、コンパイルされ、定義体テンプレートと呼ばれるオブジェクト形式として各要素プロセッサに転送される。

プログラムの実行は通常、トップレベルからの質問を与えることによって行われるが、SVPはこの質問をプロセッサ間で授受される環境情報の形に翻訳して各プロセッサに実行制御を委ねることになる。各要素プロセッサのメモリに格納されている定義体テンプレートは、この環境情報を書き換えていく操作のための命令と考えることができ、この定義体テンプレートの情報を元に、各要素プロセッサで解釈・実行が行われる。トップレベルからの質問に対する解は、この環境情報の中に含まれている解指定情報及び解情報に与えられていて、解が求まった段階でSVPに環境情報が転送され、解が出力され、実行を終了する。

4.2 KPRのマシン命令

KPR-Lのプログラムをコンパイルして得られる定義体テンプレートは、その機能レベルとして、各プロセッサのマシン命令に対応していると言える。基本的なマシン命令は、大別するとWAM(Warren's Abstract Machine)の命令セットと同じ機能レベルのものと、並列性を制御するための命令の2種類に分けることができる。各プロセッサのマシン命令に関しては、ARPとORPの機能が異なるので、各プロセッサのテンプレート・メモリ・モジュールに格納される情報の性質や機能レベルが異なっている。

コンパイルされた定義体テンプレートは節に関するいろいろな情報を表現している。並列リダクション・モデルに基づくプロセス割り付けやプロセス間の関係付けなどに関する情報も、これに含まれている。

4.3 FGHCプログラムの実行

FGHCのプログラムをKPRのマシン命令列にコンパイルし、実行するのであるが、3.3で述べた2つの機能、すなわちトラスト機構やAND関係にあるゴールの処理のスケジューリングなどをいかに効率よく実現するかが重要となる。

まず、トラスト機構についてであるが、候補節の単一化やガード部のゴール実行などの処理をORPが受け持つ。ORPにおける4つの

UU(Unification Unit)により、節頭部の単一化を並列に行うことができ、またガード部の実行を行う。そして最初にsuccessメッセージを返してきたUUに対応する節を選択し、その本体部のゴール実行を要求するデマンドを送出し、他のUUの処理をsuspendすることでトラスト機構を実現することとなる。しかしどの候補節も成功せず、かつ単一化のsuspendした候補節が1つ以上存在する場合には、呼び出し元のゴールがsuspendする。また候補節の実行中に呼び出し元のゴールに含まれる変数に値を設定しようとする、単一化の中断の規則により、実行がsuspendする。ORPにおいて、このような実行のsuspendが生じた場合、呼び出し元のゴールを処理するARPにsuspendメッセージを送出することになる。

AND関係のゴールの並列処理方式としては、実行すべき複数のゴールを待ち行列につなぎ、これに対しデッドロック・フラグを持たせ、待ち行列の先頭ゴールから実行して行き、一巡したところでデッドロック・フラグを見て、ゴールがsuspendしているかどうか管理する方式を採用した。この処理は具体的にARPで管理することになる。

4.4 ARPにおける実行時のゴール管理

ARPでは、節の本体部のゴールの管理を行うが、これらのゴールに関する大域的な情報、すなわちゴールの引数情報やsuspendに関するフラグなどの環境をARU内にあるSPIMに作る。FGHCはOR並列型言語と異なり、トラスト機構により、あるゴールから環境が複数に分かれることはなく、また単一化の中断の規則によりあるゴールから呼び出し側のゴールに値を送出することはない。従って解が複数生じるためにそれだけプロセス間で多くの通信メッセージをやり取りしなければならない並列リダクション・モデルに比べると、多少は通信メッセージが単純となる。しかしゴールを待ち行列につなぐためのマシン命令や、呼び出し側のゴールに値を送出することによりsuspendメッセージなどが必要となる。

5. おわりに

以上、KPRにおけるFGHCの処理方式について述べた。FGHCのOR並列性に関しては、候補節の実行順序の規定がセマンティクスとしてないため、この点でKPRのOR並列性が生かされると期待できる。一方、AND並列性に関してはAND関係にあるゴールを待ち行列につなぐという方式をとったが、これに関しても、元の論理プログラムに対応するAND/OR木を深さ優先探索か幅優先探索のいずれでスケジューリングするかで効率も変わるので、この点に関しても検討する必要がある。

参考文献

[1] 柴山ほか:「論理型プログラミング言語向き並列計算機KPRの並列処理方式」, 情報処理学会論文誌, 30, 12 (1989).