並列化Prolog処理系「LONLI+」(1)
—基本設計方針—

4Q-8

広瀬 正* 車谷 博之* 加藤 整**
*(株)日立製作所システム開発研究所 **(株)日立マイクロコンピュータエンジニアリング

1．はじめに

通常の逐次型PROLOGプログラムをマルチプロセッサ上で並列に実行する言語処理系「LONLI+」を開発している。本稿ではその設計方針について述べる。

2．前提と目標

スケジューリングエキスパートシステムなど、PROLOGで記述された計算量の大きい計画型知識処理システムの開発数が多くなりつつある。一方、マルチプロセッサシステムがコスト的にも実用段階に入りつつある。PROLOGの並列実行については、ICOTなど、数多くの研究実験が行われているが、我々の目標は、両者を結び付ける実用的な処理系を開発することにあり、

(1) 数台から数十台レベルのメモリ共有型(密結合)汎用マルチプロセッサを対象とする。
(2) (逐次型)PROLOGとの互換性を確保する。
(3) マルチプロセッサ資源活用のためのユーザ指示も可能とする。

を前提とした。

(2)のPROLOGとの互換性の確保は既存のPROLOGプログラムがそのまま利用可能とするるためのほかに、デバッグが逐次型PROLOGの場合と同様に行なえる様にするためでもある。

(3)はマルチプロセッサシステムのためのシステム記述言語であるための必須条件である。もしユーザが望むならば、マルチプロセッサ上で実行することを意識し、マルチプロセッサ資源を有効に活用するプログラムも書けるように配慮しておかねばならない。

3．設計方針

以下の方針に従って設計を進めた。

(1) 並列化制御プリミティブによる実行制御
ソースプログラムの静的解析を行い、並列化効果の大きい並列処理部分の抽出を試みる。
並列化処理は、逐次PROLOGプログラム内に特別な組込述語(並列化制御プリミティブと呼ぶ)の挿入によって実現する。

(2) 有効な並列性の抽出
ソースプログラムの静的解析を行い、並列化効果の大きい並列処理部分、カットの影響を受けない部分を抽出し、並列化オーバヘッド、プロセス間同期オーバヘッドを極力圧縮する。

(3) 「王位継承型スケジューリング方式」の採用
できるだけ逐次型PROLOGの処理順序を尊重した深さ優先型の動的スケジューリング方式も選択可能とし、PROLOG仕様順守のために避けられない並列処理間待合せ時間を短くする。

次節以降に、それぞれの基本的な方式を述べる。

4．並列化制御プリミティブ

並列実行のために必要な実行制御機能は「並列化制御プリミティブ」と呼ぶ組込述語として実現する。通常のPROLOGプログラム内にこの組込述語を挿入するだけで並列実行が実現される。挿入は、システム(並列化コンパイラ)が自動的に行うが、ユーザが自ら修正しても良い。

並列化制御プリミティブとしては、

(1) 並列処理部分と逐次処理部分を分ける機能(並列実行ブロック指定機能)のための並列化制御プリミティブparaとdfg(depth first gateの略)、
(2) 並列実行を開始する部分を指定する機能(並列化機能)のための並列化制御プリミティブdivideを設ける。

図1は、上述の並列化制御プリミティブをソースプログラムに挿入した例を示している。並列処理ブロックはネストが許される。

dfgプリミティブがpara以降で発生した並列処理プロセス間の同期を取り、dfg以降は逐次的に実行

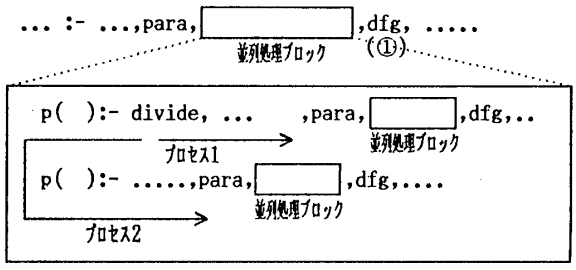

図1. 並列化制御プリミティブの働き
divide述語でプロセス1とプロセス2が生成され、以降並列に処理される。
プロセス1とプロセス2はdfg述語(①)で同期制御を受ける。

An parallel Prolog processor LONLI+ - design concept
Tadashi HIROSE*,Hiroyuki KURUMATANI*,Osamu KATOU**
*HITACHI, Ltd., **HITACHI Micro Computer Engineering, Inc.

される。具体的には、並列処理ブロック内で並列実行中の複数のプロセスの実行を、深さ優先の順(逐次処理時と同じ順序)となるようにする。次のプロセスは、先に通過したプロセスが終了するまで実行が待たされる。

並列化制御プリミティブdivideは並列化する部分の指示である。divideが実行されると、そのプロセスはこのプリミティブが成功した場合の処理と失敗した場合の処理を行う二つのプロセスとなり、並列に実行される。成功側のプロセスは、divide述語までバックトラックしてきた時点で消滅する。

制御プリミティブはすべて通常のPROLOGプログラムに追加される形で挿入する。これらのプリミティブの機能を無動作(true動作)とすれば従来通り逐次型PROLOGとして動作させることができ、従って、そのプログラムをそのまま従来どうり逐次実行し、改良、デバッグすることができる。

5．ソースプログラム静的解析と並列化制御プリミティブの自動挿入

PROLOGプログラムの場合、全ての節の並びがOR並列処理の候補であるが、分割オーバヘッドを圧縮するためには十分に処理時間の長い(粒度の大きい)処理部分だけを選択的に並列処理する必要がある。

また、並列処理プロセス間でカット機能実現のために必要となる処理量の大きい同期処理(他プロセスの実行を停止させる処理)を省くために、カット処理を含む部分を並列化対象から外す。

そのため、応用プログラムをトップレベル(スタート述語は与える)から順に解析し、この種の述語内のOR節は並列化対象から外し、選択的に並列化処理部分を抽出し、並列化制御プリミティブを挿入する。詳細は[1]を参照されたい。

6．王位継承型動的スケジューリング方式

実行時にプロセッサ台数以上の並列プロセスが生成された場合、必要以上のプロセス生成の抑止のための動的なスケジューリングを行う。スケジューリング方式はユーザ選択項目の一つだが、我々は、前で述べたプロセス間同期処理(dfg述語)発生時のプロセス待ち時間を小さくするために、逐次実行時の処理順序を尊重するスケジューリング方式を用意した。具体的には、探索木の中で、深さ優先順位の高いプロセスにだけ別プロセス生成権を与える。これはちょうど長子が優先権をもつ王位の継承規則にあてはまることから、「王位継承方式」と呼ぶ。

図2に王位継承方式の概念図を示す。図は考え方を示しているにすぎないが、左側の処理部分(逐次処理時に先に処理される部分)が、図のように優先的に処理される。

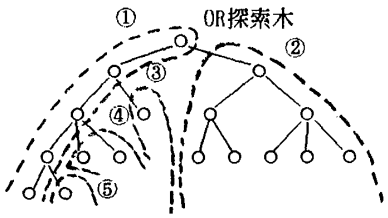

図2．王位継承型スケジューリング
①②③...が生成プロセスの処理範囲

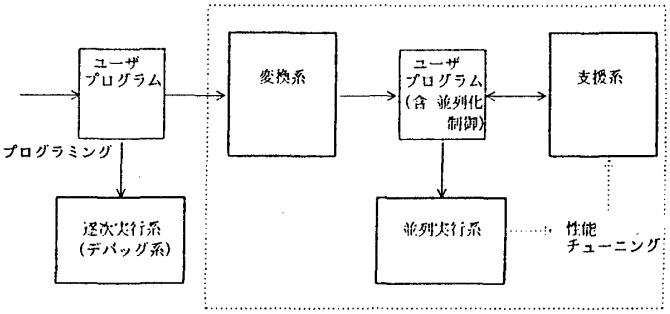

図3 LONLI+ システム構成

7．システム構成

図3にシステム構成を示す。LONLI+システムは従来の逐次型PROLOG処理系LONLIに加えて

(1) 変換系 ： ユーザプログラムを静的に解析し、並列化制御プリミティブを自動挿入するシステム。

(2) 並列化実行系 ： マルチプロセッサ上で、並列制御プリミティブの指示に従い、ユーザプログラムを並列実行させるシステム。実体は、並列制御プリミティブの組込述語群。

(3) 支援系 ： より効率的な並列実行を実現するために、プログラム内の並列化制御プリミティブを対話的に変更するための支援システム。

の三つのサブシステムから構成される。従来の逐次処理系はユーザプログラムのデバッグシステムとして使用する。

8．まとめ

通常の逐次型PROLOGプログラムをそのままマルチプロセッサ上で並列に実行する言語処理系「LONLI+」を提案した。提案した並列化制御プリミティブdivideをプログラム内の適当な箇所に挿入することで、簡単に経験則を踏まえたbest first探索が実現できる。我々の実験では、この手軽な性能チューニング機能が実応用の世界でかなり有効であった。

参考文献

[1]車谷、他:並列化PROLOG処理系「LONLI+」 - 並列化変換と実行方式、情報処理第39回全国大会