

3N-4

並列 A I マシン Prodigy の応用(1)

— 並列プロダクションシステムの実現と評価 —

John R. Canfield 藤田 純一 小柳 滋
(株) 東芝 総合研究所

1. はじめに

エキスパートシステムの適用分野が拡大するのに従って、プロダクションシステム(P S)を高速に実行する手法への期待が高まっている。特にリアルタイムエキスパートシステムを実用的な速度で動作させるためには大幅な速度向上を図る必要があると言われている。この性能を達成するために、並列処理への期待が大きい。我々は並列処理により大幅な性能向上の達成を目的としてプロダクションシステムの並列処理を目指している。本稿では、プロダクションシステム O P S 5 の並列処理について述べる。

2 P S の並列処理方法

並列 P S として Rete ネットワークに基いたシステムが提案されている [1]。しかしながら Rete ネットワークのノードごとにプロセッサを割当てて並列処理を行うことを想定すると、次の理由で並列処理の効果が少なくなる [2]。

- (1) 同一クラスの 1 入力テストノードは縦に並ぶため、本来並列処理が可能であるが逐次処理となる。(図 1a,b)
- (2) 同一ルールの 2 入力テストノードが縦に並ぶため、逐次処理となる。(図 1c,d)
- (3) 各ノードの処理に要する時間にバラツキがあるため並列処理の効果が減少する。特に 2 入力テストノードは 1 入力テストノードの数倍～10 倍の時間を要する。
- (4) ノード間を流れる情報(トークン)が大きい場合、通信のオーバーヘッドが増大する。

これらの並列処理の効果を阻害する要因を解決する方策として、トークンを属性単位に分割し、全ての 1 入力ノードと 2 入力ノードの並列処理を行う方法が考えられる。しかし、この方式ではルールノードの入力リンクが複数個となり、マッチの結果より動作の対象となる要素を求める作業、すなわち全ての条件を満たす要素の番号を求める作業が困難になる。

この問題を解決するため、各クラスの操作対象となる要素をただ一つであると限定する。すなわち、複数の要素をもつクラス(B)を要素毎に展開し(B1, B2)、関連するテストノード、ルールノードも要素毎に展開してネットワークを作る。図 1 の例の展開結果を図 2 に示す。その結果、ルールの条件部を満たす W M E は一意的に定まり、上記の解の生成は容易に行える。また各テストノードから

ルールに送信する情報はテストの結果(真/偽)だけでなく、ノード間の通信量を減らすことができる。さらに 2 入力ノード内で保持するトークンも高々 1 つであるため、2 入力ノードの処理も軽減され、ノード間の処理時間のバラツキもなくなる。すなわち、前述の Rete ネットワークの 4 つの欠点をすべて取り除くことができる。

本方式では、ルールをクラスの要素毎に展開する必要がある。そこで我々はクラスの要素を予め宣言することによりコンパイル時にルールを展開する方針をとる。クラスの宣言には次の 3 種類がある。

(assert A unique) : class A に属する要素は 1 つだけである。

(assert B serial) : class B は複数個の要素をもつが、その中で操作の対象となる要素は 1 つだけである。

(assert C (key a1) (value C1 C2)) : class C は複数個の要素をもち、その要素は属性 a1 の値により識別され、その取り得る値は C1, C2 である。

Rete ネットワーク以外にもう一つの並列処理を阻害する要因は O P S 5 でルールの実行がただ 1 つと限定されていることである。これを解決するために、競合しない限りすべて並列にルールを実行するモデルを採用し、そのモデルでルール間の優先度を予め定義することにより、競合するルールの並列発火を抑止する方式をとる。

3. P S のインプリメンテーション

3. 1 ハードウェア構成

以上に述べた P S の並列処理方式に基いたソフトウェアを開発し、並列実験機 Prodigy0 上で評価を行った。Prodigy0 は現在開発中の Prodigy1 マシン [3, 4] のソフトウェア開発を目的としたマシンであり、AS3000 をホストとして 16 台の Processing Element (P E) を持つバックエンドマシンである。P E は 68000 マイクロプロセッサ、512 K のメモリ、P E 間インタフェースにより構成されている。また P E 間通信としてはマイクロプロセッサで構成されたルータが 2 次元 mesh ネットワークでメッセージを送信、転送、受信する形で行われている。

3. 2 ソフトウェア構成

本システムはプリプロセッサ、コンパイラ、ルールアナライザ、アロケータ、インタプリタの 5 つのサブシステムにより構成される。プリプロセッサ、コンパイラ、ルールアナライザはルールで記述されたソースプログラムを並列 P S ネットワークに変換し、ルールの並列発火についての情報をユーザに出力する。アロケータは P S ネットワーク記述を入力とし、ネットワークの各ノードを静的に P E 上に分散して配置する。インタプリタは P S ネットワークの

"Prodigy" Parallel AI Machine Applications(1)
A Parallel Production System Model and its
Measurement

John R. Canfield, Sumikazu Fujita, Shigeru Oyanagi
Toshiba R&D Center

記述と割当ての結果を入力とし、並列実験機上でPSを実行する。

3. 3 結果

プロダクションシステムの並列化の効果を調べるため、ネットワークのノードを193個持つモンキーバナナ問題を用いてProdigy0上で実行し、評価を行なった。この例題におけるPEの台数効果を図3に示す。

図3に示すように、16台のPEを用いた場合に2.1倍の性能向上が得られる。並列処理による性能向上を阻害する理由を以下に示す。

- ・通信オーバーヘッド：総計算時間（全PEでの処理時間の総和）は16PEの場合は1PEの1.5倍であった。このオーバーヘッドの84%はメッセージ送信手続きに要するオーバーヘッドである。送信手続きが遅い理由はProdigy0が特別な送信ハードウェアを持たないため、送信手続きはルータが空いていない間は待たなければならないからである。
- ・ノードの割当て：16PEの場合、最も負荷の重いPEは全負荷の1/4を占めている。サイクル毎に動的に割当てを行い負荷を完全に均等に分散できるならば、16PEの性能は現在の5.2倍向上する。

3. 4 改良

上記の問題について、以下のような改良を考えている。

- ・ルータをLSI化することにより通信を高速化する。
- ・ハードウェアとOSの改良により送信手続きのオーバーヘッドを減らす。
- ・ノードの現在の割当て方法を改良することにより16台PEシステムで2～3倍の性能向上が可能である。

4. おわりに

プロダクションシステムの並列処理方式と並列実験機上での評価について説明した。評価の結果、16台のPE上で実行されたシステムは1台のPEのシステムより2倍程度の性能が得られた。さらに通信オーバーヘッドとノードの割当て方法を改良することにより、かなり性能向上させることが可能である。

現在、通信機能をLSI化したProdigy1を設計しており、Prodigy0上でPSインタプリタとアロケータを改良中である。今後、Prodigy1上で改良したPSインタプリタを実行させて性能評価を行なう所存である。

[参考文献]

1. A.Gupta, et.al.: "Parallel Algorithms and Architectures for Rule-based Systems." Proc.13th Int. Symp. on Computer Architecture, 1986.
2. 鈴岡, 藤田, Canfield, 小柳: "プロダクションシステムの並列処理方式" 昭63人工知能学会全大(1987-7)
3. 小柳, 藤田, 中村, 鈴岡: "超並列AIマシンの構想", データフローワークショップ1987, pp.159-166 (1987-10).
4. 鈴岡, 田辺, 中村, 藤田, 小柳: "並列AIマシンProdigyのアーキテクチャ" 昭62情報処学会第35回全大, 1D-2(1987-3)

Rule 1

```

if (A ^1 1 ^2 2 ^3 <x>)
   (B ^1 3 ^2 <x> ^3 <y>)
   (C ^1 4 ^2 <y>)
then (make .... )

```

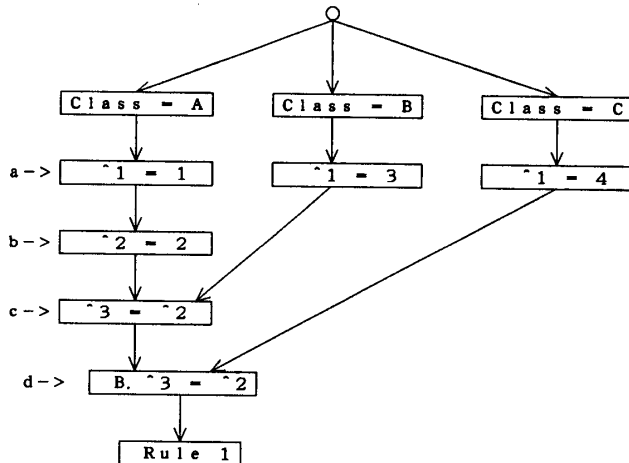


図1 Rete ネットワーク

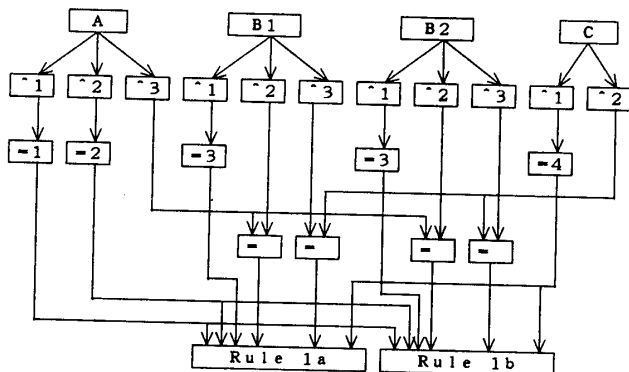


図2 展開後のネットワーク

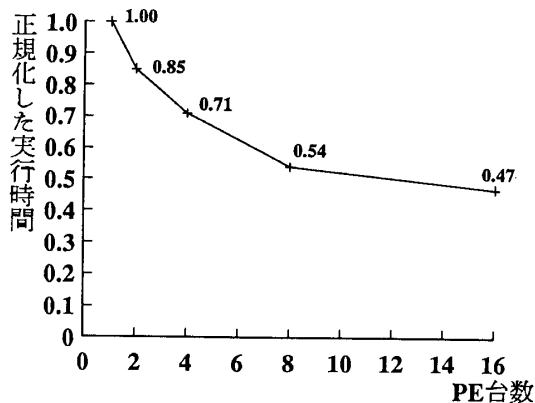


図3 モンキーバナナ問題における台数効果