

プロダクションシステム並列化の解析評価手法と TWIN 並列化方式の提案†

湯 川 高 志†† 石 川 勉††

本論文では、プロダクションシステム (PS) の照合並列化による速度向上度を解析的に評価する手法を提案する。また、新しい並列化方式として TWIN 方式を提案し、本解析手法により他方式と比較する。解析評価法は、対象とする並列化方式とエキスパートシステム (ES) の特徴を表すパラメータから、計算により速度向上度を推定する手法である。本解析手法では、PS 照合に用いられる Rete ネットをモデル化し、このモデルに基づいて速度向上推定の一般式を構築する。この推定式に並列化方式に関するパラメータを代入することにより速度向上度を推定する。本評価法を用いて、いくつかの並列化方式に対し広い ES 特性にわたる概略的な速度向上評価を行い、各方式の適用領域を明確にする。つぎに、上の評価結果の考察に基づき、簡易なアーキテクチャで 10 倍程度の速度向上が得られる TWIN 方式を提案する。本方式は、Rete ネットの構造的な並列性と、Rete ネットノードにおける照合処理のデータ並列性を融合した方式であり、照合処理中のプロセッサ間通信を不要としたまま、負荷ばらつきの少ない並列化を実現している。また、解析評価と並列プロセッサシステムを用いた実験評価の両方を行い、本方式の性能を確認し、解析評価の妥当性を示す。

1. はじめに

エキスパートシステム (ES) 開発の本格化に伴い、その推論機構であるプロダクションシステム (PS)¹⁾の高速化が要求されており、高速化の一方法として、推論処理を並列化する研究が数多くなされている。PS 推論処理のうち最も多くの時間を消費するルール照合処理には、Rete と呼ばれる高速アルゴリズムが最も一般に用いられているため、この Rete アルゴリズムを対象とした並列化方式が多く提案されている^{2)~7)}。

Rete アルゴリズムでは、図 1 (a) に示すようなルールを、図 1 (b) に示す Rete ネットと呼ばれるデータフローネットワークに展開する。事実知識を表すワーキングメモリ (WM) が、トークンとして Rete ネットの枝に沿って流れ、個々のノードで部分的な照合が行われることによって、全体の照合が処理される。この Rete ネットの構造、ネットを構成するノードでの処理に着目して、さまざまな粒度での並列化が考えられる。Rete ネットの構造的並列性に着目した方式として、第一段目の 2 入力ノード (トップノード) とそれに連なるすべてのノードを粒とするトップノード方式²⁾、個々のルールを粒とするルール方式、個々のノードを粒とするノード方式³⁾などが、また、ノード処理の並列性に着目した方式として、個々の

トークン比較を粒とするジョインスライス方式⁴⁾などが提案されている (図 2)。一般に、細粒度化すれば、並列度が向上しプロセッサ負荷が均等化されるが⁴⁾、プロセッサ間通信量が増加する。

PS 並列化による速度向上度は対象 ES の問題特性に大きく依存するため、方式比較のためには、広範囲の ES 特性にわたる評価が望まれる。しかし、従来の報告はシミュレーション評価であり、報告ごとにサンプル ES が異なるため、相互の比較は困難であった。

本稿では、まず、PS 照合並列化方式の速度向上度を解析的に評価する方法について述べる。本評価法は、Rete ネットのモデル化により並列度や処理量を ES の問題特性で表現し、このモデルに基づいて、様々な並列化方式を広範囲の ES 特性にわたり概略的に評価する。本評価法をもちいて、いくつかの代表的な並列化方式の比較を行い、各方式の適用領域、ハードウェアの要求条件を明確にする。

次に、評価結果の考察に基づき、簡易なハードウェアで広い適用領域を持つ TWIN 方式を提案する。本方式ではトップノード方式と同様の Rete ネット分割により、照合処理中のプロセッサ間通信を不要とし、さらに、Rete ネットの個々のノード内でのデータ処理も並列化して負荷ばらつきを低減している。本方式に対しては、解析評価法と実際の ES を用いた実験との両方により評価し、性能を他方式と比較するとともに、解析評価法の妥当性も確認する。

† An Analytical Evaluation Method for Parallel Production Systems and Proposal of a Parallel Scheme Named TWIN by TAKASHI YUKAWA and TSUTOMU ISHIKAWA (NTT Network Information Systems Laboratories).

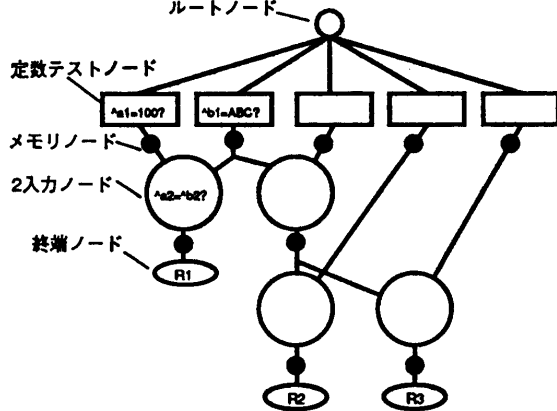
†† NTT 情報通信網研究所

```

(R1 (C1 (^a1 100) (^a2 ?x))
  (C2 (^b1 ABC) (^b2 ?x))
  -->
  (create C3 (^c1 ?x) (^c3 ABC)))

```

(a) ルールの一例
(a) An example of a rule
ルートノード



(b) 上記ルールに対するReteネット
(b) Rete network for a rule

図 1 Rete ネット
Fig. 1 Rete network.

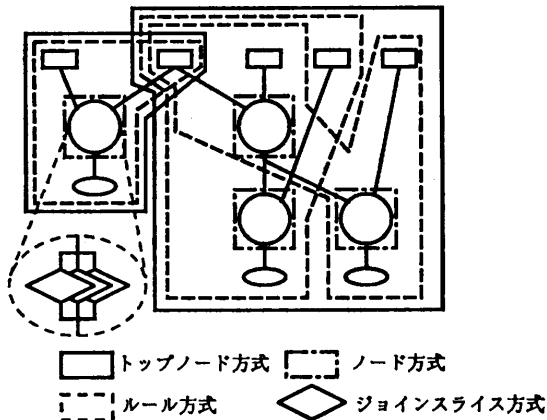


図 2 Rete アルゴリズムの並列化方式

Fig. 2 Parallel schemes for the Rete algorithm.

2. プロダクションシステムと Rete アルゴリズム

PS は、if-then 型のルール知識を用いて、事実知識を格納したワーキングメモリ (WM) の更新を繰り返すことにより推論を進める。推論の一連の動作は認知サイクルと呼ばれ、照合・競合解消・ルール実行の 3 フェーズから構成される。このうち、照合フェーズが最も処理量が多いため、これを効率化する方法として

Rete アルゴリズムが提案され⁸⁾広く用いられている。

Rete アルゴリズムでは、ES を構成するすべてのルール条件部を Rete ネットと呼ばれるデータフローグラフに展開し、WM の変化分 (トークン) をこれに流す事により照合を行う。図 1 にルールとそれに対応する Rete ネットの例を示す。ルール条件部において括弧でくくられた、C1, C2 等のクラス指定と (^a1 100) 等の属性名-属性値条件の組は条件要素と呼ばれる。WM が全条件要素を満たす場合にルールはマッチする。マッチしたすべてのルールは競合集合に登録され、競合解消フェーズにおいて競合集合から一つが選択される。ルール実行フェーズにおいて、選択されたルールのアクション部が実行される。アクション部には、一般に WM の生成、削除、更新が記述されている。個々の WM の生成、削除はそれぞれ 1 個のトークンを発生し、次の認知サイクルでの照合の対象となる。また、WM 更新は、削除と生成の組み合わせとして処理されるため 2 個のトークンを発生する。

Rete ネットは、ルートノード、定数テストノード、2入力ノード、終端ノードから構成されている。また、各ノードには、照合の中間結果を蓄えるメモリノードが付属している。新たに発生したトークンはまずルートノードに入力され、次に定数テストノードにおいて、条件要素内の定数値との比較が行われる。このノードでの比較が成功すると、2入力ノードにおいて、トークンはメモリノードに蓄えられ、条件要素にまたがる変数の照合を行うために、対向する枝のメモリノードに蓄積されたすべてのトークンとの突き合わせが行われる (図 3)。従属に接続された 2 入力ノードをすべて通過して終端ノードに到達した場合に、そのトークンは照合に成功し、競合集合に登録される。また、Rete ネットでは、異なるルールでも、同一の照合条件のものはひとつのノードとして共有する。

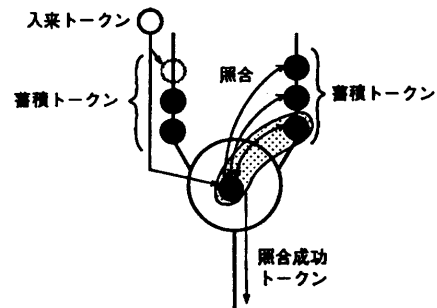


図 3 2入力ノードでの処理

Fig. 3 Token comparison in two-input nodes.

Rete アルゴリズムでは、上述のようにメモリノードに中間結果を蓄積することで不変の WM に対する再照合を不要としている。一般に認知サイクルごとに変化する WM は少数のため⁹⁾、これにより照合が飛躍的に高速化される。また、同一条件照合ノードの共有化により、さらに処理量を削減している。しかし、これを用いても照合フェーズが推論時間の 90% 以上を消費すると報告されている⁹⁾。

3. 速度向上度の解析的評価法

本章では、照合特性を支配する Rete ネットをモデル化して ES 特性と Rete ネットとの関係を明らかにし、これに基づいてさまざまな並列化方式の速度向上を推定可能な速度向上推定一般式を導出する。

3.1 エキスパートシステムの問題特性

推論並列化による速度向上は、Rete ネットの構造および処理の特性により規定される。Rete ネットは ES によってそれぞれ異なるため、速度向上の推定には ES の問題特性から Rete ネットの構造を推定することが必要となる。

Rete ネット構造の推定に必要な問題特性には表 1 に示すものがある。個々の条件要素に対し定数テストノードが生成され、また、個々のルールに含まれる条件要素数に応じて従属接続された 2 入力ノードが生成されるため、これらのパラメータから、Rete ネットのノード数、接続枝数等が推定できる。同一定数テストの比率 S_c は ES に含まれる条件要素の総数に対する同一定数テストの割合である。同一の定数テストはひとつのノードを共有するため、定数テストノード数の推定にこの値が必要である。

動作特性の推定に必要な問題特性を表 2 に示す。ルール当たりのアクション数 A は、一つのルールに含まれる WM 操作アクションの数の平均値である。また、アクション部更新動作比率 β_m は、WM 操作アクション中の WM 更新動作（2 個のトークンを発生）の割合で、この 2 者からルール実行で生成されるトークン数が計算できる。クラス数 L は ES に含まれるクラスの種類である。また、WM 要素 (WME) の数 W

表 1 Rete ネット構造を決定する ES 特性パラメータ
Table 1 The structural parameters of ES programs.

パラメータ	記号
ルール数	R
ルール当たりの条件要素数	E
同一定数テストの比率	S_c

表 2 Rete ネットの動作を決定する ES 特性パラメータ
Table 2 The dynamic parameters of ES programs.

パラメータ	記号
ルール当たりのアクション数	A
アクション部の更新動作比率	β_m
WME 数	W
クラス数	L

は、生成、削除動作により推論中に変化するが、ここではその平均値を用いる。これらは、ES ソースプログラムの分析、ES 設計段階での見積りによって得られる。

3.2 Rete ネットのモデル化

2 入力ノードのふるまいに着目し、Rete ネットのモデルを構築する。2 入力ノードは、定数テストノードを通過したトークンとメモリノードに蓄積されたすべてのトークンとの照合を行うため、処理量が高ノードに比べ非常に大きい。さらに、全種類のノードは必ず 2 入力ノードに接続されるため、このモデル化により他ノードの特性も付随的に求まる。

代表的な ES についての測定⁹⁾がなされており、2 入力ノードの出力枝の分岐数、ルール当たりの条件要素数の度数分布が示されている。これらの測定結果から、Rete ネットは以下の特性を持つと考えられる。

Rete ネットの特性：

- 2 入力ノードの出力枝の分岐数は比較的少数。
- ルール当たりの条件要素数の度数分布は、条件要素の増大に伴い指数関数的に減少。

この特性より、2 入力ノードを以下のようにモデル化する (図 4)。

2 入力ノードモデル：

- 出力枝は平均的に B 本に分岐している。
- 出力枝のうち平均 P 本が終端ノードに接続されている。



図 4 2 入力ノードのモデル

Fig. 4 Model for a two-input node.

出力枝が平均 B 本に分岐し、そのうちの平均 P 本が終端ノードに接続されるとすれば、次段の 2 入力ノードに接続される出力枝は平均 $B-P$ 本となる。これは、2 入力ノードの数が一段ごとに $B-P$ 倍ずつ ($B-P < 1$ なら) 減少することを意味する。すなわち、2 入力ノードの従属段数 (条件要素数 - 1 に等しい) の度数分布は指数関数的に減少し、上記特性に良くあう。

平均条件要素数 E とノードの終端割合 P の関係:

1 段目の 2 入力ノード数を d_0 としたときに、 n 段目で終端するノード数 d_n は、このモデルより、 $d_n = d_0 \times P(B-P)^{n-1}$ となる。これより、全終端ノード数 D_i は、

$$D_i = \sum_n d_n = \sum_n d_0 P(B-P)^{n-1} = d_0 \frac{P}{1-(B-P)} \quad (1)$$

となり、その ES における従属段数の平均値 e は、

$$e = \sum_n n \times \frac{d_n}{D_i} = \sum_n n \times \frac{d_0 P(B-P)^{n-1}}{d_0 \frac{P}{1-(B-P)}} = \frac{1}{1-(B-P)} \quad (2)$$

で計算できる。ところで、 $E = e + 1$ であるから、

$$P = (B-1) + \frac{1}{E-1} \quad (3)$$

平均条件要素数 E は ES のプログラムより容易に得られるから、残る分岐数 B が決まれば Rete ネット特性は一義的に決定できる。出力枝の分岐数 B は ES のルール記述を分析すれば正確に求まるが、代表的な ES に対する測定⁹⁾において 2 入力ノード出力枝の分岐数として報告されている値は 1.1~1.9 の範囲である。

以上のモデルと問題特性に基づき導出した、Rete ネットの構造的な特性および処理量特性を表 3、表 4 に示す (導出は付録参照)。これらの値を用いることにより、平均活性化ノード数や各ノードでの平均照合回数、認知サイクルでの総照合回数などが計算できる。例えば、1 認知サイクル当りの総照合回数は、1 認知サイクルで発生するトークン数 M と 1 トークンにより活性化される 2 入力ノード数 O_2 、そして

表 3 Rete ネットの構造的な特性

Table 3 The structural features of the Rete network.

項 目	記号	計 算 式
定数テストノードの総数	D_1	$R(E-1)(1-S_2)$
2 入力ノードの総数	D_2	R/P
定数テストノードから 2 入力ノードへの接続枝数	S	$D_2\{2-(B-P)\}$

表 4 Rete ネットの処理量特性

Table 4 The dynamic features of the Rete network.

項 目	記号	計 算 式
活性化される定数テストノード数	O_1	D_1/L
定数テストノードを通過するトークン数	T_1	$S\alpha_c/L$
活性化される 2 入力ノード数	O_2	$T_1(E-1)$
2 入力ノードでの平均照合回数	j	$(W/L)\alpha_j$
1 認知サイクルで発生するトークン数	M	$A(1+\beta_m)$

個々の 2 入力ノードでの照合回数 j の積として計算できる。

ただし、速度向上の計算には、処理量の平均、分散等統計量がわかれば十分なため、導出に際し以下の仮定を設けた。

仮定:

- 各クラスに対する条件要素の数は均一、また、トークンもこれに対し均等に発生。
- 各ノードの照合成功率は一定 (定数テスト: α_c , 変数照合: α_j とする)。

ここで、 α_c , α_j はルールの書き方に依存するが、文献 9) によれば、 α_c は定数テストノードの成功率として報告されており 10~数 10% である。また、2 入力ノードにおいてトークン入来に対し対向するメモリノードが空でない比率が 10~30%、行われたテストが成功する比率が 5~10% 程度と報告されているため、 α_j は数 % 以下である。これら値は、定数テストを粗くし、変数照合で最終的に絞り込む一般的なルールの記述法からも納得のできる数値である。また、認知サイクルごとの照合成功ルール数 (設計時に見当がつく) から逆算することもできる。

3.3 速度向上推定の一般式

上記の Rete ネットモデルに基づいて、ES 特性と並列化方式を代入するだけで速度向上度を推定できる一般式を導出する。

1 認知サイクルを逐次処理した場合の処理時間を T_0 とすると、並列化時の同サイクルの処理時間 T は、

$$T = \frac{QT_0}{N} \gamma + T_c \quad (4)$$

で表される。ここで、

N : プロセッシングエレメント (PE) 数

Q : 処理の分割による総処理量の増加係数

γ : 負荷ばらつきによる処理時間の増加係数

T_c : PE 当りの通信時間

である。ところで、Rete アルゴリズムの処理は、ノードでの照合とその結果であるトークンの伝搬であ

るため、トークン伝搬に伴うデータ転送量は、各 PE で処理するトークン数、ノード数に比例すると考えられる。したがって、通信時間 T_c は、PE 当りの処理量に比例し、

$$T_c = \frac{QT_0}{N} C_0 \quad (5)$$

と書ける。 C_0 は方式に固有の通信オーバーヘッド係数である。以上より、速度向上 U は、

$$U = \frac{N}{Q} \times \frac{1}{\gamma + C_0} \quad (6)$$

処理量増加係数 Q は、Rete ネットの分割法により容易に決定できる。例えばルール並列の場合、2入力ノードを共有しなくなるため、定数テストノードと2入力ノードとの接続枝はルール数 $R \times$ 平均条件要素数 E 本となる。2入力ノード数は枝数に比例するため、オリジナルの Rete ネットの枝数 S との比率が処理量増加となり、この場合は、 $Q = RE/S$ となる。

また、通信オーバーヘッド係数 C_0 は、総照合回数に対する通信回数の割合を C_R 、一对のトークン照合の時間に対する1トークン転送時間の比を C_c とすると、

$$C_0 = C_R \times C_c \quad (7)$$

となる。

負荷ばらつき係数 γ については、以下に詳細に述べる。

通信オーバーヘッドを除いた並列処理時間 T_P は最も時間のかかる PE により支配され、

$$T_P = \max(T_1, T_2, \dots, T_N) \quad (8)$$

となる。ただし、 T_i は PE_i の処理時間である。負荷ばらつき係数は、負荷均等と想定した処理時間と T_P との比と考えられるから、

$$\gamma = \frac{\max(T_1, T_2, \dots, T_N)}{\sum T_i / N} \quad (9)$$

この T_i を詳細化すれば、照合特性により γ を定式化できる。いま、分割した個々の処理 g_i の処理時間を t_i と定義する。これらの中には互いに逐次的に処理されるべきものもあるから、そのような処理を集めて並列化の粒を構成する。それらの粒を $G_1, G_2, \dots, G_m$ と表す。また、 G_i の処理時間を T_{Gi} と書く事にする。 G_i を各 PE に均等に割り付ければ、PE 当り m/N 個となるから、PE の処理時間 T_i は m/N 個の T_{Gi} を加算したものとなる。すなわち、

$$T_i = \sum_{j=x}^{j=x+m/N} T_{Gi}, \quad (10)$$

ただし、 $x = (i-1)m/N + 1$ 。

上記定式化に基づき、以下の手順で γ を計算できる。

- 1) モデルにより T_{Gi} の分布を計算。例えば、ルール方式の場合、処理時間は粒に含まれる2入力ノード数に比例すると考えられるので、2入力ノード従属段数と相似な分布（指数分布）となる。
- 2) 上で求めた分布を持つ母集団から m/N 個取り出した標本値の総和の分布（すなわち T_i の分布）を求め、その平均値 $E(T_i)$ を計算¹⁰⁾。
- 3) T_i の分布を持つ母集団から標本を N 個取り出した

表 5 並列化粒度と基本アーキテクチャ

Table 5 Parallel schemes for the Rete algorithm and suitable architectures.

並列化方式	並列処理の粒	アーキテクチャ
トップノード	トップノードとそれに連なるすべてのノード	放送機能を持つ（トークンを全 PE に配布するため）バス結合型
ルール	個々のルール	同 上
ノード	個々のノード	ツリー、メッシュ等のネットワーク結合型
ジョインスライス	ノード内での個々のトークン照合	トークン配布、収集機構を持つ階層的ネットワーク結合

表 6 各方式のパラメータ

Table 6 The estimated parameters for the parallel schemes.

パラメータ	トップノード	ルール	ノード	ジョインスライス
粒の数 n	T_i/L	QT_i/L	T_iM/L	T_iMj/L
Q	≈ 1	RE/S	1	1
処理量分布	粒内のノード数	粒内のノード数	無視可	無視可
逐次性	なし	なし	ノードの従属数	無視可
C_R	0	0	$1/j$	2

時の最大値の分布を求め、平均値 $E\{\max(T_i)\}$ を計算.

- 4) $\gamma = E\{\max(T_i)\}/E(T_i)$ を計算.

4. 解析手法による並列化方式の評価

代表的ないくつかの方式に対し、速度向上推定式の各項を導出し、速度向上度の比較を行う。

表5に比較評価した方式を示す。表におけるアーキテクチャは各方式に適していると考えられる基本的なものである。方式とアーキテクチャから、速度向上推定式の各項は表6のように求めることができる。トップノード、ルール方式の処理量分布は、粒に含まれる2入力ノード数に比例すると考えられ、これは条件要素数分布と相似である。また、ノード方式では個々のノードの接続関係に起因する逐次性が負荷ばらつきを支配する。ジョインスライス方式は本質的に高並列度で処理量ばらつきも小さいため、速度向上は通信オーバーヘッドで規定される。

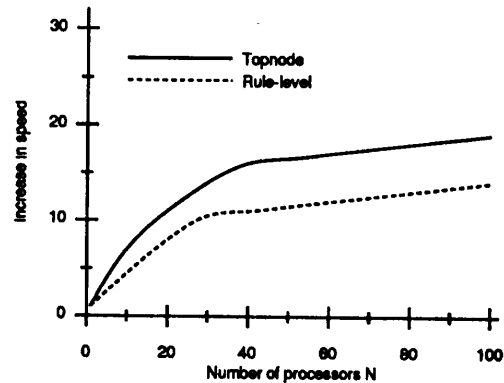
表7に示すパラメータを持った中規模ESを想定して、照合フェーズの速度向上を評価した結果を図5に示す。ノード方式およびジョインスライス方式に対しては、トークン転送速度（1トークン転送時間の1トークン照合時間に対する比 C_e ）を変化させている。この値と表6に示した通信発生割合 C_R との積が、通信オーバーヘッド係数 C_0 となる。

照合フェーズの消費する時間は推論全体の約90%のため、照合部のみ的高速化では十数倍の速度向上が限界である。この観点で図5をみると、トップノード方式は十分な速度向上を達成していることがわかる。また、ルール方式は処理量増加が原因となりトップノード方式より常に速度向上が小さい。粒度の細かいノード方式も、従属接続されたノードの逐次性により並列処理できる粒が比較的少ないため、トップノード方式やルール方式などの粗粒度方式と同程度の性能である。ただし、認知サイクルごとに多数のトークンを発生するESでは、従属接続ノードをパイプライン処理できるので、より大きな速度向上が期待できる。ジョインスライス方式は、高速のプロセッサ間通信が可能ならば非常に大きな速度向上が期待できる。しかし、通信性能が低い場合には、トップノード方式よりも速度向上が小さい。一対のトークン比較は数マシンステップで済むため、この方式の性能を十分に引き出すには、トークンを高速に転送できる特殊なアーキテクチャが要求される。

表7 評価に用いたES特性

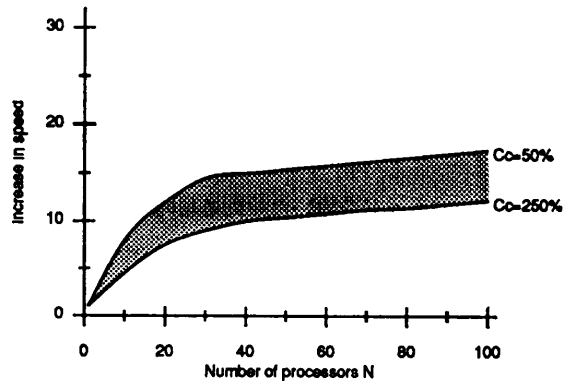
Table 7 The parameters of a sample ES program.

$R : 300$	$A : 2.0$	$\alpha_e : 0.35$
$E : 3.5$	$\beta_m : 1.0$	$\alpha_j : 0.07$
$L : 7$	$B : 1.2$	
$W : 250$	$Sc : 0.3$	



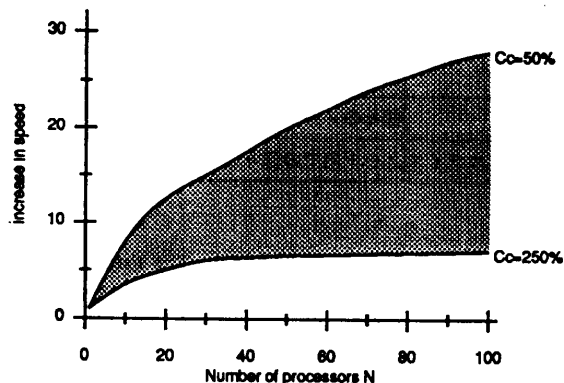
(a) トップノード方式、ルール方式

(a) Topnode scheme and Rule-level scheme.



(b) ノード方式

(b) Node-level scheme.



(c) ジョインスライス方式

(c) Join-slice scheme.

図5 解析評価による速度向上特性

Fig. 5 Analytical results.

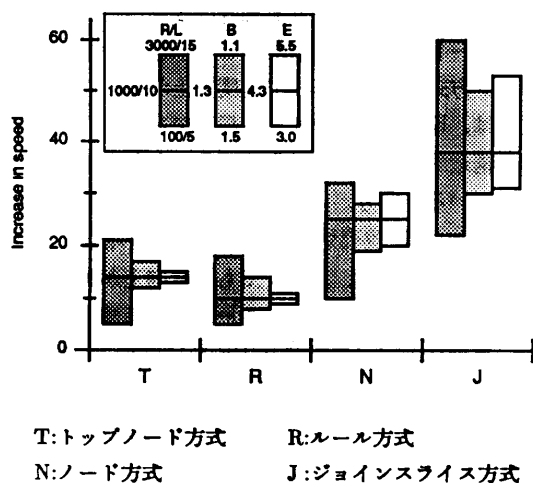


図 6 速度向上の問題依存性
Fig. 6 Sensitivities of the increase in speed.

ES 特性パラメータの変化に対する速度向上度の変動範囲を図 6 に示す。変化させたパラメータは、Rete ネット構造への影響が大きく、速度向上の規定要因となるものである。パラメータの変化範囲は各種 ES の実測結果⁹⁾を参考に決定している。この図から、並列化方式の速度向上はルール数（すなわち ES 規模）に最も大きく影響されることがわかる。また、小規模領域ではトップノード方式やルール方式などの粗粒度方式は数倍の性能しか得られないことがわかる。これは並列処理の粒が不足して負荷ばらつきが増加するためである。すなわち、ルール数が多い領域が粗粒度方式の適用領域となる。ジョインスライス方式は、本質的には、あらゆる ES 特性領域で十分な速度向上が得られるが、上述のように、この性能を引き出すには高速かつ大容量の転送機構をもつ特殊なアーキテクチャが必要となる。

5. TWIN 方式の提案

トップノード方式などの粗粒度方式では、小ルール領域での速度向上が不十分である。ただし、WME 数が小さい場合には処理量が少なく、逐次処理でも十分な速度が得られるため、問題となるのは WME 数が多い場合である。そこで、WME 数が多い場合にはノードでの照合回数も多い点に着目し、構造並列であるトップノード方式にノードでのトークン照合

の並列性を融合させた TWIN 方式を提案する。

TWIN 方式では、トップノードで分割されたサブネットを複数の PE に配置し、メモリノードへのトークン蓄積を選択的に行って照合対象のトークンを分散させることによりデータ並列性を引き出す。すなわち、2 入力ノードが対向枝上のトークンとの突合せとトークンの蓄積というふたつの動作を行っている点に着目し、複数の PE に割り付けた同一ノードにそれぞれメモリノードをもたせ、これに蓄積されるトークンを分散させることで通信なしにノード処理を並列化する。具体的には、2 入力ノードでの処理を以下のようにする（図 7）。

2 入力ノードの処理：

- トークンの突合せはすべての PE で行う。
- トークン蓄積はどれか一つの PE が行う。

トークンを蓄積する PE をランダムに選ぶことにより、各 PE に均等にトークンが分散される。

また、上記のノード並列化は最上段の 2 入力ノードのみで行う。なぜならば、Rete ネットでは、上段ノードの照合結果が下段ノードの照合データとなっており、上段での処理を分散することで、下段での処理も自然と並列に行われるからである。このようにすることで、トークンが PE に入力された後は、データの転送は不要なので、従来と同様に照合処理中のプロセッサ間通信は不要である。ただし、2 段目以降のノードの右側入力枝はすべてのトークンを蓄積する必要がある。

データ方向の並列化は、原理的にはメモリノードに蓄積されたトークン数まで可能であるが、実際には蓄積トークン数に応じて並列度を動的に変化させるのは困難であるし、2 段目以降のメモリノードに重複して蓄積されるトークンによりメモリ消費が大きくなるため、数～十数にとどめるのが得策である。

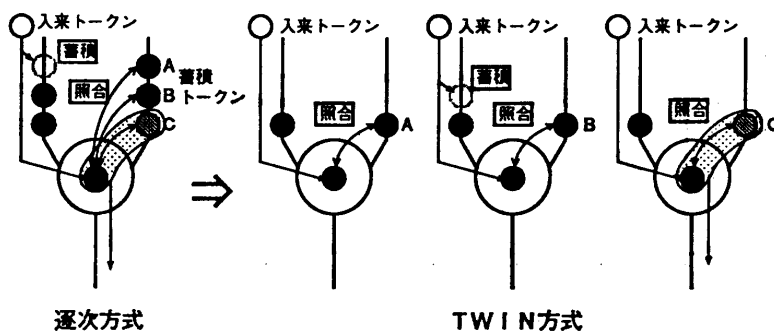


図 7 TWIN 方式
Fig. 7 The TWIN scheme.

6. TWIN 方式の評価

6.1 解析による評価

本方式の解析評価結果を図8に示す。ESパラメータは4章での解析と同一のものをを用い、データ並列度は3とした。WME数が多ければ、小ルール数でも十分な速度向上が得られることがわかる。

6.2 実験による評価

本方式の速度向上効果を確認するため、実験評価を行い解析結果との比較を行った。実験システムはマイクロプロセッサボードを汎用バスで結合したものである。バスは実行フェーズで発生したトークンを全PEに配布するための放送機能を持っている。

実験用として2地点間の最短経路を深さ優先法により探索する小規模ESを構築して用いた。本ESはルール数23、WME数1000で構成される。実験結果を図9に示す。実線が実測値、破線が推定値である。

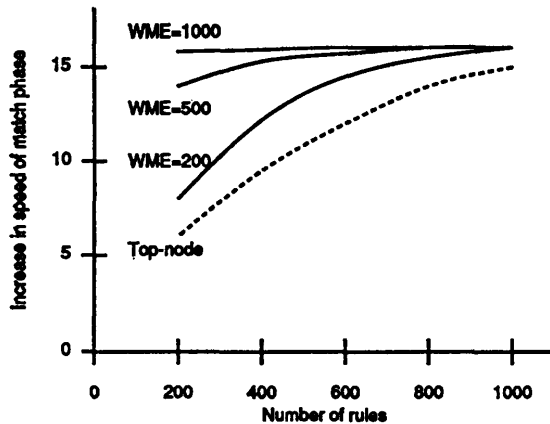


図8 TWIN方式の解析評価結果
Fig. 8 Estimated increase in speed using TWIN.

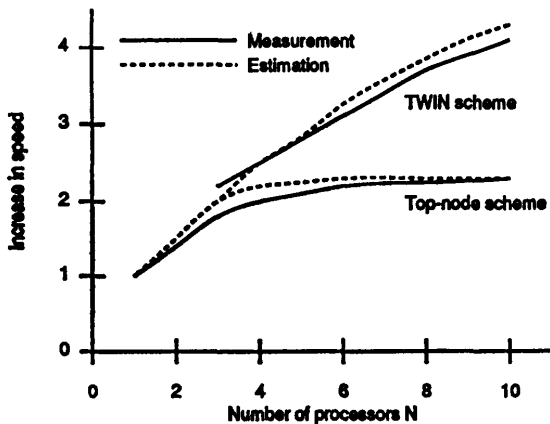


図9 実験結果
Fig. 9 Experimental results.

解析評価と同様にデータ並列度は3とした。また、トップノード方式の結果もあわせて示している。トップノード方式では2倍程度で速度向上が飽和している。これは、ES規模が20ルール程度であり、Reteネットの構造的分割だけでは負荷ばらつきが大きいことを示している。一方、TWIN方式では、データ並列性の融合により並列処理可能な粒の数が増加して負荷ばらつきが減少し、10PEでも速度向上が飽和せずに上昇している。しかし、ルール数が極端に小さいため、並列化効率(速度向上/PE数)は低めである。100ルール規模のESを用いれば、より高い並列化効率を得られ、16PEで10倍程度の速度向上が得られると予測される。

また、図からわかるように、推定結果は実測とは良く合っており、誤差は10%程度である。並列化方式の適用領域の比較や、その方式に適したアーキテクチャの探索に利用するには十分な精度である。

7. まとめ

PS並列化の速度向上を解析的に評価する手法を提案した。まず、2入力ノードのふるまいに着目して、Reteネットをモデル化し、ネット形状、処理量をESパラメータにより表現した。次にこれに基づいて、負荷ばらつき、通信オーバーヘッド等を考慮して、速度向上推定の一般式を導出した。本推定式により、評価用ESやシミュレータを構築することなく、様々な並列化方式の評価が可能となった。

つぎに、本解析評価法を用い、粒度の異なるいくつかの方式について、ESパラメータを広範囲に変化させて比較評価を行った。その結果、従来の方式では、小ルール数の領域で負荷ばらつきにより十分な速度向上が得られないか、または、非常に高速なトークン転送を行うための特殊なハードウェアが必要とされることを明らかにした。

さらに、上記結果に基づき、通信オーバーヘッドと負荷ばらつきが共に少ないTWIN方式を提案した。本方式は、Reteネットを構造的に粗く分割することにより、照合処理中のプロセッサ間通信を不要とし、2入力ノードでの比較処理を並列化することにより、小ルール時の負荷ばらつきを抑えている。

本方式を、解析手法と小規模なESを用いた実験により評価し、小ルール数の領域でも十分な速度向上が得られることを確認した。

参 考 文 献

- 1) Brownston, L., Farell, R., Kant, E. and Martin, N.: *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming*, Addison-Wesley, Reading, Mass. (1985).
- 2) Gupta, A.: Implementing OPS5 Production Systems on DADO, *Proc. 13th Int. Conference on Parallel Processing*, pp. 83-91 (1984).
- 3) Gupta, A., Forgy, C. L., Newell, A. and Wedig, R.: Parallel Algorithm and Architectures for Rule-Based Systems, *Proc. 13th Int. Symp. on Computer Architecture*, pp. 28-37 (1986).
- 4) Kelly, M. A. and Seviaora, R. E.: A Multiprocessor Architecture for Production System Matching, *Proc. 13th Int. Symp. on Computer Architecture*, pp. 28-37 (1986).
- 5) Oshisanwo, A. O. and Dasiewicz, P. P.: A Parallel Model and Architecture for Production Systems, *Proc. 16th Int. Conference on Parallel Processing*, pp. 147-153 (1987).
- 6) 鈴岡 節, 藤田純一, Canfield, J. R., 小柳 滋: プロダクションシステムの並列処理方式, 人工知能全国大会論文集, pp. 207-210 (1988).
- 7) 湯川高志, 石川 勉: 負荷均等化を図った粗粒度 PS 並列化方式, 第 37 回情報処理学会全国大会論文集, pp. 1421-1422 (1989).
- 8) Forgy, C. L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, *Artif. Intell.*, Vol. 19, No. 1, pp. 17-37 (1982).
- 9) Gupta, A. and Forgy, C. L.: Measurements on Production Systems, Technical Report, Carnegie Mellon University (1984).
- 10) 河田龍夫: 確率論とその応用, 紀伊國屋書店 (1961).

付録 Rete ネット特性の導出

1) 定数テストノードの総数: D_1

定数テストは必ず 2 入力ノードに接続されるため, ルール間で共通の定数テストがなければ, ルール数 R とルール当りの平均 2 入力ノード数 $(E-1)$ の積 $R(E-1)$ である. しかし, 共通化されているノードが S_c の割合で存在するので,

$$D_1 = R(E-1)(1-S_c). \quad (11)$$

2) 2 入力ノードの総数: D_2

2 入力ノードは P の割合で終端するため m 段目の 2 入力ノードの終端数 b_m は, ノード数を $D_{2(m)}$ として,

$$b_m = D_{2(m)} \times P. \quad (12)$$

この総数はルール数 R に等しいから,

$$R = \sum b_m = P \sum D_{2(m)} = P D_2, \quad (13)$$

したがって, 2 入力ノードの総数 D_2 は,

$$D_2 = R/P. \quad (14)$$

また,

$$D_{2(m)} = D_2 \{1 - (B-P)\} (B-P)^{m-1}. \quad (15)$$

3) 定数テストノードから 2 入力ノードへの接続枝数: S

Rete ネットでは 2 入力ノードの右入力枝は必ず定数テストノードに接続されている. また 1 段目では左入力枝も定数テストノードに接続される. したがって

$$S = D_{2(1)} + \sum D_{2(m)} \quad (16)$$

$D_{2(1)}$ を D_2 で表して代入すれば,

$$S = D_2 \{2 - (B-P)\} \quad (17)$$

4) 活性化定数テストノード数: O_1

活性化される定数テストノードは, 入力されたトークンのクラスと等しいクラスをもつクラスごとの条件要素数が均等との仮定より,

$$O_1 = D_1/L. \quad (18)$$

5) 定数テストノード通過トークン数: T_i

照合に成功する定数テストノード数は, $O_1 \times \alpha_c$. 定数テストノードは平均 S/D_1 の分岐を持ち, これらすべてにトークンが配布されるから,

$$T_i = O_1 \times \alpha_c \times \frac{S}{D_1} = \frac{S \times \alpha_c}{L}. \quad (19)$$

6) 活性化 2 入力ノード数: O_2

2 入力ノードの平均従属段数は $E-1$ なので, これと T_i との積が活性化 2 入力ノード数となる.

7) 平均照合回数: j

照合対象となる蓄積トークンの数は定数テストを通過したトークン数に等しいためクラス当りの WME 数と定数テスト成功率との積になる.

8) 1 認知サイクルで発生するトークン数: M

トークンは生成削除動作で 1 個, 更新動作で 2 個発生するため, 平均実行要素数 A と $(1+\beta_m)$ との積となる.

(平成 3 年 10 月 25 日受付)

(平成 4 年 10 月 8 日採録)

**湯川 高志 (正会員)**

昭和 60 年長岡技術科学大学工学
部電気電子システム卒業。昭和 62
年同大学院修士課程修了。同年日
本電信電話(株)入社。以来、人工知
能向けプロセッサの研究開発に従
事。現在、NTT 情報通信網研究所研究主任。IEEE、
電子情報通信学会、人工知能学会各会員。

**石川 勉**

昭和 45 年電気通信大学電気通信
学部応用電子卒業。同年日本電信電
話公社武蔵野電気通信研究所入所。
以来、主記憶装置、高信頼化技術、
フルウェーハ LSI、並列プロセッ
サ、知識処理技術の研究に従事。工学博士。現在、
NTT 情報通信網研究所知識処理研究部グループリー
ダ。IEEE、電子情報通信学会、人工知能学会各会員。