

5K-8

エキスパート・システム構築ツールEXTKERNEL (4)
 - MELCOM70 MXでの高速処理 -

黒田正博 松下和隆 加藤幸男 小川義高
 (三菱電機株式会社 三菱電機東部コンピュータシステム 株式会社)

1. はじめに

エキスパート・システム構築ツールEXTKERNELは、MELCOM-PSI, MELCOM MX シリーズ上で動作する。

MELCOM MX シリーズ上の本ツール(以降EXT on MXと呼ぶ)は、prolog言語をベースに実現しているため単体性能では専用マシンMELCOM-PSIに及ばない。

そこで、EXT on MXで作成したエキスパート・システムを含んだトータル性能の向上という面から検討したのでそれを報告する。

2. 高速化

エキスパート・システムの高速化を考える場合、そのシステムの機能が固定の時は、低位言語で書ききることにより高速化をはかることができる。しかしながら、このようなシステムは機能拡張に弱く、記述言語の高速化の恩恵を受けないという面がある。エキスパート・システム(含エキスパートシステム構築ツール)は、まだ機能的に発展段階にあることを考えると、高位言語をベースとし必要な所のみ低位言語化していく事が良いといえる。

以下に順次、ソフトウェア・システムレベル、アルゴリズムレベル、言語レベルそして計算機レベルでの高速化を報告していく。

2.1 ソフトウェア・システムとしての高速化

実際のシステムは、1つのエキスパート・システム

ムからできているのではなく、複数のエキスパート・システム、複数の非エキスパート・システムからなっていることが多い。このため、システムの応答性能を上げるには、そのシステムを構成している個々のシステムの並列動作を得ることが有効になってくる。

EXT on MXは、前発表の共有データベース機能に加えて、

- (1) エキスパート・システム間の動作制御機能 (図1参照)
- (2) エキスパート・システムの並列動作機能 (H/Wの並列動作、参考文献(2))

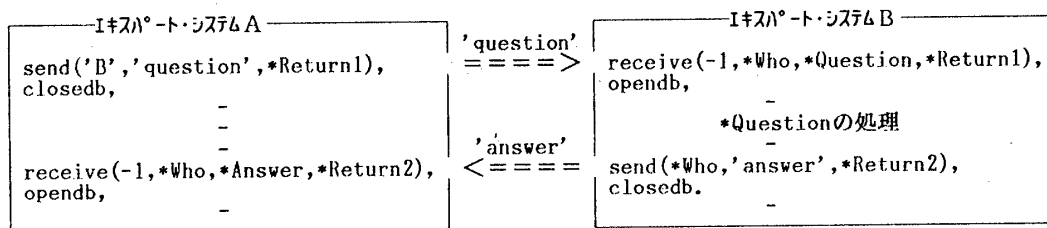
を持っている。これにより、1つの計算機内でのエキスパート・システムの並列動作、さらには複数計算機システムの並行動作を実現している。

2.2 アルゴリズムレベルの高速化

アルゴリズムレベルの高速化として、Reteアルゴリズムがよく知られている。これは、発火可能ルールの左辺が非常に類似したパターンを持っているという構造的類似性がある時に有効である。

しかしながら、実際の産業システムでは、システムを階層的に記述しルールを作成するため、Reteアルゴリズムの効果は小さい。

このため、本ツールでは現在Reteアルゴリズムを採用していない。



send (相手エキスパート・システム名、送信メッセージ、終了情報) : 相手エキスパート・システムを起動あるいは停止を解き、送信メッセージを送る。
 receive(モード、相手エキスパート・システム名、受信メッセージ、終了情報) : 送信メッセージを受信する。モードは、本述語が呼び出された時点でメッセージがない場合の動作を指定する。

図1 エキスパート・システム間の動作制御

2.3 言語レベルの高速化

EXT on MXの前向き推論は、図2の最初のルール記述を構文解析し、prolog言語の単位節に変換する。そして、prolog言語の持つ照合機能を利用し、作業領域要素を書き換えていく事により推論していく形式をとっている。

この処理では、照合機能 (unification) と作業領域要素の書き換え (assert/retract 述語) が全処理時間の中の大部分を占めている。

このうち、assert/retract述語処理のprolog処理アルゴリズム高速化を考え、

- (1) 対象を単位節とする、
- (2) 引数を一般変数ではなく匿名変数までとする、
- (3) 引数の個数の最大値を明確にする、

ことにより、通常の assert/retract 述語の性能を落すことなく、このような単位節のassert/retract述語を約4倍高速にする事ができた。

つぎに、後向き推論であるが、これは prolog 言語の持つ照合機能をそのまま用いているため、prolog言語処理系のアルゴリズム的高速化をはかることができない。

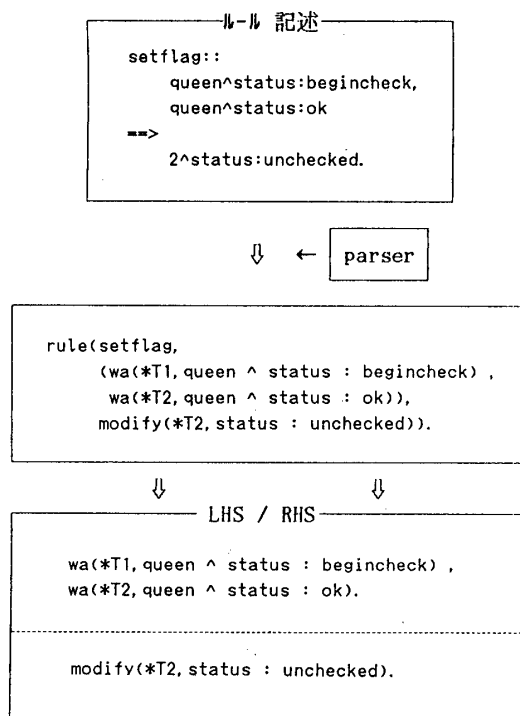


図2 前向き推論の実行形式変換

2.4 計算機レベルの高速化

EXT on MXの前向き推論は、作業領域要素の書き換えという処理が基本であるため、後向き推論

のようにEXTコンパイラの恩恵を受けにくい。

前向き推論は、2.3節の高速化は図れるが、基本処理が作業領域要素の処理系による解釈実行であるため、十分とは言えない。

そこで、処理系の何等かの高速化が必要になってくる。つまり、

- (1) 照合処理自体の高速化、
- (2) assert/retract, global assert/retract 述語自体の高速化、
- (3) スコア・ルールを越えて保存しているデータ更新の高速化、

を現在は低位言語で記述しているが、これをregisterの割り付け・H/W特性の考慮を行なったさらに低位の言語に書き換える必要がでてきた。

この書き換えにより、

- (1) 照合処理自体は3.3倍、
- (2) assert/retract, global assert/retractは2.3倍、
- (3) スコア・ルールを越えて保存しているデータの更新は2.2倍、

の性能向上をはかることができた。

3 おわりに

本報告では、prolog言語をベースにしたエキスパート・システム構築ツールで共有データベース機能をprolog言語に拡張することにより、均質なシステムから構造を持ったシステムにすることができた。しかも性能を落さず、EXT on PSIとの統一性をとることができることを示した。

本報告により、EXT on PSIと統合性のあるprolog言語ベースのエキスパート・システムで、小・中規模エキスパート・システムとしての応答性能が得られるメドがついた。

今回の報告では、Reteアルゴリズムを中心としたアルゴリズムレベルでの高速化を考えなかったが、今後条件を満すルール取り出し処理の高速化(整数演算・データの高速処理)を考えていく予定である。

謝辞

prolog言語の機能拡張で多大な協力をいただいた三菱電機情報電子研究所の根岸道郎氏、計算機製作所の伊藤孝之氏にそれぞれ深謝します。

参考文献

- (1) Forgy, C., "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem", Artificial Intelligence 19, 1982
- (2) 辻他、『エキスパート・システム構築ツール EXTKERNEL (1)~(2)』情処学会第33回全国大会予稿集5K-4, 5
- (3) 高井他、『MELCOM70 MX/3000 マルチプロセッサ』情処学会第30回全国大会 pp. 97 (1985).