エキスパートシステムのための実時間推論データベースシステムの実装と評価

2P－8

小野 智弘　西山 智　小花 貞夫
国際電信電話株式会社 研究所

1. はじめに

筆者らは、網管理分野等におけるエキスパートシステムを構築するために、大量のデータを扱え、外界の変化に実時間で応答可能な推論機能を持つデータベースシステム (以下、推論 DB と呼ぶ) の開発を行なっている[1][2]。本稿では、推論 DB の実装と性能評価を述べる。

2. 推論 DB の概要

推論 DB は、(1)ES の推論機能をデータベースに組み込むことで大量のデータを用いた推論を可能とし、(2)LEAPS[3]を改良したアルゴリズム[2]を使用することで外界の変化に実時間で応答を可能にしたデータベースシステムである。

推論 DB は図 1に示すように、ルールコンパイラと推論ライブラリを用いて生成した、推論プロセス、外部 I/F プロセス、DB サーバプロセスの 3 つの実行プロセスからなる。実装は全て C++を用いた。

3. 推論 DB の実装

3.1 ルールコンパイラ

OPS83[4]準拠の言語で書かれた応用プログラムを入力とし、実行プロセスを生成する際に必要な応用プログラムに依存したコードを出力するコンパイラである。パーザ部分には LEX、YACC を使用した。ソフトウエアの規模は全体で 7K 行程度である。

入力言語は OPS83 に対して以下の 2 点の変更を加えたものである。

- ディスク上に書かれる永続的な WME(Working Memory Element:推論に用いられる事実) を作成するために、エレメント宣言に以下のようにキーワード persistent を指定可能とした。
  <elementdef>::=persistent element (<要素の並び>)
- OPS83 の組み込み関数のうち、LEAPS アルゴリズムでは使用しない競合集合に関するものを削除し、代わりに発火すべきルールと対応する WME の組を求める関数を追加した。

また出力については、下記の推論ライブラリにおいてデータの永続／非永続を C++の仮想関数を用いて隠蔽させているため、ルールコンパイラはデータの永続／非永続に依存しないコードを生成する。

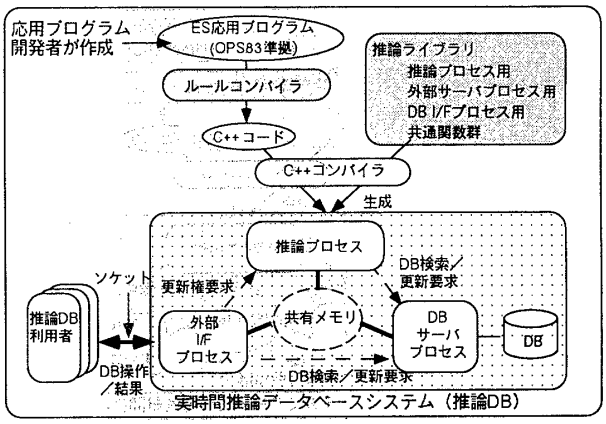

図 1: ソフトウエア構成

3.2 推論ライブラリ

推論ライブラリは実行プロセスのメインルーチンや推論機能などのライブラリであり、推論プロセス用、DB サーバプロセス用、外部 I/F プロセス用、共通関数群の 4 つのライブラリからなる。ソフトウエアの規模は全体で 24K 行程度である。

推論プロセス用 LEAPS アルゴリズムを実現する WME や索引 (αメモリと呼ぶ) 等のクラス定義である。C++の仮想関数を用いて永続／非永続の区別のないクラスとクラス毎のメンバ関数を定義する。例えば WME を格納する WM(Working Memory) クラスの場合、図 2に示す WME を操作するための関数がある。

```
class WM{
    ....
     (データ定義)
    ....
public:
    void        add_wme()       WMEの追加
    void        remove_wme()    WMEの削除
    void        clear_wm()      WM内の全てのWMEの削除（初期化）
    }
```

図 2: WM クラスが提供するメンバ関数

DB サーバプロセス用 DB への格納機能を実現するライブラリであり、以下の特徴をもつ。

- 永続と指定された WME と、その WME を推論する際に必要となるαメモリを DB に格納する。
- 推論を行なうために、DB 上の WME 及びαメモリを共有メモリ上にバッファリングする。バッファリングはページ単位 (2Kbyte) とし、LRU アルゴリ

"Implementation and Evaluation of Database System with Real-time Inference Mechanizm for Expert Systems " by Chihiro ONO, Satoshi NISHIYAMA and Sadao OBANA, KDD R & D Laboratories

ズムを用いる。

- DB 上では、WME をページ番号＋ OID（各データにユニークに付与した識別子）で管理する。
- 外部からの DB 検索のための索引も WME に付与可能とする。
- DB のデータ管理機能には拡張可能 DBMS:ASSIST /M[5]を用いた。

外部 I/F プロセス用　推論 DB 利用者に対して DB 操作を提供する。操作は永続 WME、非永続 WME のいずれに対しても可能であり、永続 WME に対しては DB サーバプロセスに検索を依頼し、非永続 WME に対しては共有メモリ上の WME を直接検索する。

共通関数群　共通関数は共有メモリ管理、プロセス間通信機能等を持つ。共有メモリ上では、永続 WME はページ単位で管理し、一方、非永続 WME は高速化を図るために WME 用、αメモリ用等の大きさ毎に管理する。

4. 評価

実装した推論 DB を用いて、大量データ (=WME) を扱う際の性能評価を以下の要領で行なった。

[評価 1] OPS83（RETE アルゴリズム[6]を使用）と、非永続データを用いた推論 DB の推論速度を比較した。結果を図 3に示す。

[評価 2] 推論 DB での、非永続データを用いた場合と永続データを用いた場合の推論速度を比較した。結果を図 4に示す。

なお、ここでの評価条件は以下の通りである。

- SUN SPARC Server 1000(Solalis 2.3) 上で、測定を行なった。
- データの検索、更新を順次行なう 5 つのルールからなるルールシステムを作成し、これに適用するデータの件数を 10 件〜10,000 件と変化させて推論を行なった。ルールの発火回数は約 (データ件数×2) 回である。
- 共有メモリのバッファサイズは、非永続データ用 10Mbyte、永続データ用 3Mbyte とした。

5. 考察

評価 1 について　OPS83 が直接アセンブラコードを出力するのに対し、推論 DB では C++を介して実行系を作成しているにも関わらず、データ件数の少ない間では、推論 DB は OPS83 と同程度の推論速度を達成しており、効率的な実装ができたといえる。また、データ件数が増加するに従って推論 DB の推論速度が優れる傾向にあり、これは RETE と比較して計算量の少ない LEAPS アルゴリズムが性能に寄与したからと考えられる。

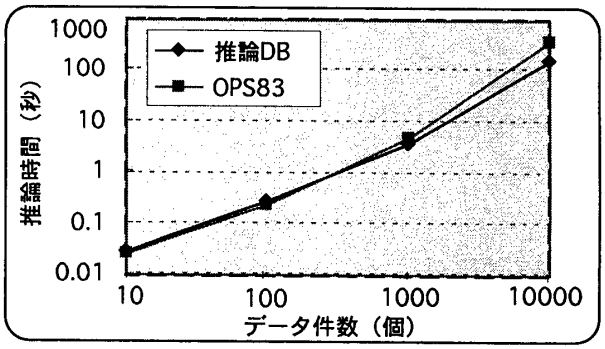

図 3: 推論 DB (非永続データ) と OPS83 との比較

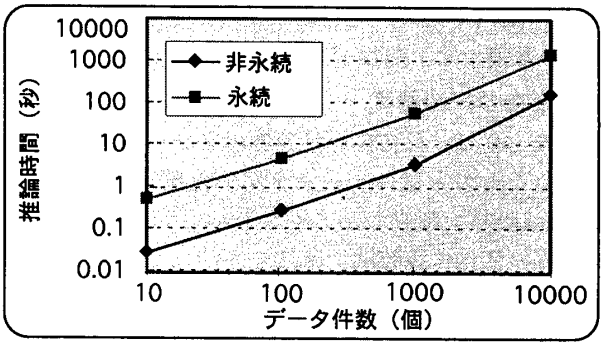

図 4: 非永続データと永続データの比較

評価 2 について　永続データを扱う場合は DB への格納のために推論時間が増加するが、それでも非永続データを扱う場合に比べて 10 倍〜20 倍程度の推論時間に抑えることができた。また、データ件数が増加するに従ってその差は減少しており、これは、共有メモリ上のバッファが効果的に作用したことと、大量データになるほど LEAPS の計算量の増加と比較して DB の格納コストの増加が少ないことが原因と考えられる。

6. おわりに

本稿では、推論 DB を生成するためのルールコンパイラと推論ライブラリの実装について述べた。また、実装した推論 DB を用いて大量データを扱う際の性能を評価し、システムの有用性を確認した。今後、外界の変化への実時間応答についての評価を行なっていく予定である。最後に日頃御指導頂く KDD 研究所 浦野所長、鈴木次長に感謝します。

参考文献

[1] 西山 他 :“エキスパートシステムのための実時間推論データベースシステムの設計－全体概要－,” 第 51 回情処全大,3D-9,(1995)
[2] 小野 他 :“エキスパートシステムのための実時間推論データベースシステムの設計－推論アルゴリズム－,” 第 51 回情処全大,3D-10,(1995)
[3] D.P.Miranker et.al.:“On the Performance of Lazy Matching in Production Systems,” Proc.of 8th AAAI,(1990).
[4] Forgy C. :”The OPS83 User's Manual System ,”(1986)
[5] 西山 他 :“OSI 管理情報ベース (MIB) の設計と実装,” 情処研究会資料,DBS 95-7 (1993)
[6] C.Forgy et al.:“RETE: A Fast Match Algorithm for the Many Pattern / Many Object Pattern Match Problem,” Artificial Intelligence, no.19, pp17-37, (1982)