

MPPMを用いた知識ベースマシン(4)

2C-8

—ソフトウェア・シミュレーションによる評価(第2報)—

酒井 浩, 柴山 茂樹, 物井 秀俊, 森田 幸伯

勸業総合研究所 勸新世代コンピュータ技術開発機構

1 はじめに

筆者らは、第5世代コンピュータ・プロジェクトの知識ベースマシン研究の一環として、単一化エンジン(UE)とマルチポート・ページメモリ(MPPM)を主要な要素とするマシンを提案している[1]。

このマシンでは Prolog のルールやファクトに相当する各知識を関係のタプルとして格納することができ、与えられたゴールについての入力演繹を、2つの関係の間で項の単一化を条件とする結合演算で実現できる。そして、この単一化結合演算は複数の UE を用いて並列処理することができ、処理の高速化が可能である。本稿では、この単一化結合演算の並列処理の性能について考察する。

2 エンジン性能のモデル化

UE は第1図に示す要素から構成され、以下に述べる方式で単一化結合演算を行なう。まず2組のソータが各々の関係について、比較属性をキーとしてタプルの並べ換えを行なう。次にペア生成部が並べ換えたタプルの並びを順に入力しながら単一化の可能性のあるタプル対を生成する。そして単一化部がタプルの対ごとに単一化を試み、実際に単一化できるものは演算の結果として出力し、そうでないものは落し落とす。詳細は[2]を参照されたい。

2つの関係PとQの間の単一化結合に要する時間は、およそ式(1)で表わすことができる。

p: 関係Pの大きさ, $\leq c$

q: 関係Qの大きさ, $\leq c$

r: 演算結果の大きさ

c: ハードウェアによる制限

$$T(p, q, r) = \alpha p + \beta q + \tau r + \delta pq \quad (1)$$

ただし、右辺の各項は下記の処理に要する時間である。

αp : 関係Pのソーティング, およびペア生成部でのPの入力

βq : 関係Qのソーティング, およびペア生成部でのQの入力

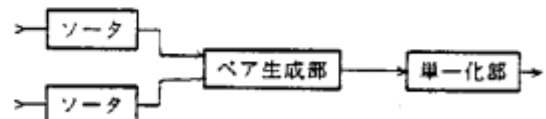
τr : 演算結果の出力

δpq : 実際には単一化しないタプル対の生成と、その落し落とし

(1) 式のパラメータ $\alpha \sim \tau$ はデータの性質に殆ど依存しないが、パラメータ δ は単一化しそうでしない項の出現する割合が大きい場合は大きな値となり、そうでない場合、小さな値となる。パラメータ δ を非選択率と呼ぶことにする。

式(1)から不等式(2)を導くことができる。この式は、エンジンの台数が変わらない場合に単一化結合を分割して実行すると、処理時間が増大することを示している。

$$T(p_1, q, r_1) + T(p_2, q, r_2) \geq T(p_1 + p_2, q, r_1 + r_2) \quad (2)$$



第1図 単一化エンジンの構成

3 単一化結合演算の並列処理

関係データベースマシンの分野では、結合演算の処理を高速化するために種々のクラスタリング手法が考案されている。これらの手法は、一般的には単一化結合演算においても、項の最も外側の関数子について適用可能である。しかし、前述の入力演繹を行なう場合、[1]に示すルールやファクトの表現では、最も外側の関数子は一種類だけとなり、処理の高速化は図れない。そこで単一化結合演算は、マルチプロセッサ・ネステッドループで実現することにした。これは関係PとQの間の単一化結合演算を行なう場合、PとQを各々 $P_1, \dots, P_n$ および $Q_1, \dots, Q_n$ に分割し、 P_i と Q_j の間の単一化結合演算をそれぞれ別の UE に処理させる方法である。

次に、この処理に要する時間Tを考える。いま、

$$\begin{aligned} p_i &= P_i \text{ の大きさ} & p &= \sum p_i \\ q_j &= Q_j \text{ の大きさ} & q &= \sum q_j \\ r_{ij} &= P_i \text{ と } Q_j \text{ の間の単一化結合の結果の大きさ} & r &= \sum r_{ij} \\ & & k &= \text{UE台数} \end{aligned}$$

とすると、

$$T \geq \sum_{i,j} T(p_i, q_j, r_{ij}) / k \quad (4a)$$

等号はUEの稼働率が100%のとき成立

$$= (\alpha np + \beta nq + \tau r + \delta pq) / k \quad (4b)$$

A Knowledge Base Machine with an MPPM(4) -A Simulation Study-

Hiroshi SAKAI, Shigeki SHIBAYAMA (Toshiba R & D Center)

Hidetoshi MONOI, Yukihiro MORITA (ICOT Research Center)

$$\geq (2\sqrt{\alpha\beta npq} + \tau r + \delta pq) / k \quad (4c)$$

等号は $\alpha np = \beta nq$ のとき成立

$$\geq 2\sqrt{\alpha\beta pq} / \sqrt{k} + (\tau r + \delta pq) / k \quad (4d)$$

等号は $np = k$ のとき成立

このうち不等式 (4d) については、すべての UE を使用するには「 $np \geq k$ 」でなければならないことによる。関係の大きさ p, q と UE の台数 k を与えられたものとし、関係の分割方法の制御により、性能を向上させるという立場にたつと、これらの不等式から次のことがいえる。

(a) 実行時間を短縮するには、(4c)と(4d)から

$$n = \sqrt{\alpha kp / \beta q}, \quad n = \sqrt{\beta kq / \alpha p}$$

とすべきである。

(b) UE稼働率を高め、かつ関係をあまり細かく分割しないようにするには、単一化結合の実行時間を等しくなるように関係を分割すべきである。しかし、そのような制御は実際には不可能であるから、次善の方法として P および Q をそれぞれ等分割することが考えられる。

(c) (a), (b) から P と Q は次式に従って分割すべきである。

$$p = \sqrt{\beta pq / \alpha k}, \quad q = \sqrt{\alpha pq / \beta k}$$

(d) 実行時間は、(4d)の式の第1項の存在により、UE台数 k とは正確には反比例しない。

上記(c)は、単一化結合の並列処理では、各々の関係をその大きさおよびUE台数に応じて分割すべきであることを示している。そこで筆者らは、関係を格納する単位であるページの大きさを比較的小きな値(0.5~4Kbyte)に設定し、2つの関係それぞれについて、上記(c)で述べた大きさに近いページ数を算出し、それを各 UE に処理させる方式を考えた。これをMP (Multiple Pages at a Time)方式と呼ぶことにする。そして比較対象として、各々の関係を1ページずつ各 UE に処理させる方式もとあげた。これをSP (Single Page at a Time)方式と呼ぶことにする。

4. MP方式とSP方式の比較

MPとSP方式の比較は、ソフトウェア・シミュレータ[3]を使用し、親子関係に関するルールとファクトから、ある人の先祖を検索する問題について行なった。この問題では(1)式のパラメタの比はおおよそ次のとおりである。

$$\alpha : \beta : \tau : \delta = 2 : 2 : 1 : 0$$

MPとSP方式の処理時間を第2図に、演算結果の格納ページに実際にデータの詰まっている割合を第3図に示す。

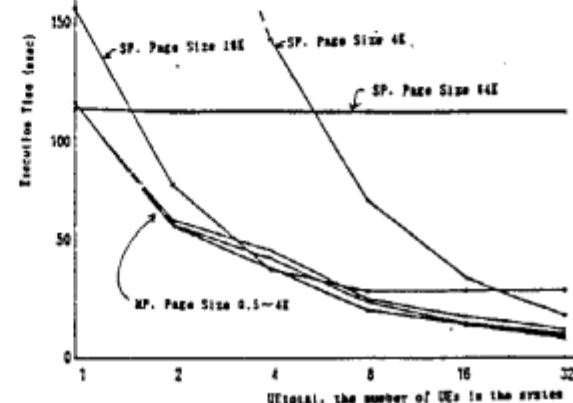
SP方式ではUE台数が少ない時はページサイズが大きいほど所要時間は短い。これは、ページサイズが大きいほど式(2)で示す分割幅が少ないためである。逆に、UE台数が多い時はページサイズがある程度小さい方が、所要時間が短くなっている。これはページサイズが小さい方がUEの稼働率が高くなるためである。SP方式とは対照的にMP方式では、種々のUE台数について安定的に良好な性能が

得られている。これは関係の分割の仕方をUE台数に応じて変えることの適切さを示している。

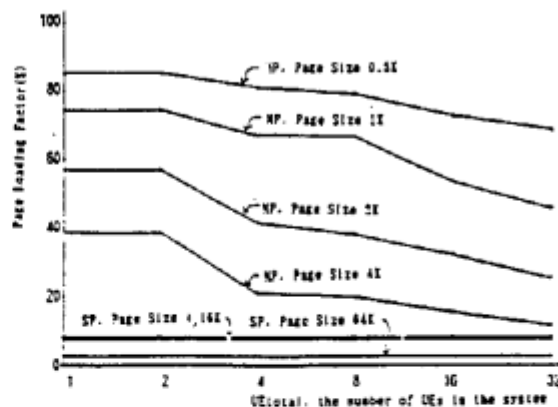
またメモリの使用効率についても、第3図よりMP方式の方が優れていることがわかる。これは、MP方式は比較的大きなデータに対する演算結果を小さなページに格納できるためである。SP方式ではメモリの使用効率を高めるためにページの詰替えを行なう必要があるが、MP方式ではそれをやらなくとも高い使用効率を得られる。

5. 結論

単一化結合演算をマルチプロセッサ・ネステッドループで並列処理する時の制御方式としてMP方式を提案した。そしてより単純なSP方式と比較して、処理時間とメモリ使用効率の両面で優れていることがわかった。この結果は関係データベースマシンについても成立つと考えられる。



第2図 MPとSP方式における先祖検索の所要時間



第3図 MPとSP方式におけるメモリの使用効率

参考文献

- [1] Yokota, H. et al.: A Model and an Architecture for a Relational Knowledge Base, ISCA, 1986.
- [2] Morita, Y. et al.: Retrieval by Unification Operation on a Relational Knowledge Base, Proc. VLDB'86.
- [3] Sakai, H. et al.: A Simulation Study of a Knowledge Base Machine Architecture, IJDM'87 (to appear).